



UNIVERSITY OF NAIROBI
FACULTY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTING AND INFORMATICS

AN APPROACH TO BOOTSTRAPPING THE DEVELOPMENT OF
MULTILINGUAL RULE-BASED GRAMMARS FOR UNDER-
RESOURCED LANGUAGES USING CROSS-LINGUISTIC
SIMILARITIES: A CASE STUDY OF A SUB-SET OF KENYAN
BANTU LANGUAGES

BENSON KITUKU
Reg no. **P80/92741/2013**

November 2022

THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS OF
THE DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE DEGREE,
DEPARTMENT OF COMPUTING AND INFORMATICS, UNIVERSITY OF
NAIROBI

Declaration

This thesis is my original work. It has not been presented for a degree in any other university. No part of this thesis may be reproduced without the author's prior permission or the University of Nairobi.

STUDENT NAME:

Benson Kituku

Signature _____



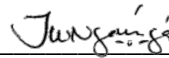
Date 20th November 2022

The thesis has been submitted for examination with our approval as the University Supervisors.

SUPERVISORS:

Dr Wanjiku Nganga

Signature _____



Date 17th December 2022

Dr. Lawrence Muchemi

Signature _____



Date 7th December 2022

UNIVERSITY OF NAIROBI

Dedication

I dedicate this thesis to my beloved wife Frida Njeru and my lovely daughters Ravine Kanini, Zarine Kanana and Zakine Kanene, who bore with my absence during the research as well as encouraged and inspired me when the going got tough. May the Almighty God richly bless you.

Acknowledgments

Though solely authored by the researcher, this thesis is a collaborative work resulting from various people's indirect contributions. Therefore, I am grateful to everyone who played a role in one way or another. Here, I shall mention a few of them.

To the supervisors: Dr. Wanjiku Nganga and Dr. Lawrence Muchemi, receive my gratitude for providing the research trajectory at the initial stage, guiding me through every stage up to the conclusion and your commitment to seeing the research come to completion.

Special thanks to the German Academic Exchange Service (DAAD) for granting me a two-year scholarship. Sincere gratitude goes to the Grammatical Framework (GF) community, especially Dr. Hans Leiß, who answered all my GF questions and encouraged me to learn it. I am also grateful to Prof. Ranta for organizing the Bantu Verb Workshop during the 6th GF summer school and the organizers of the 5th and 6th GF summer schools.

I am indebted to the linguists and informants who provided the descriptive grammar, data for elicitation, referred me to other data sources and created the test-suites by generating the gold standard for each specific Bantu language. Their criticisms and testing whether the grammar was producing the correct output have gone a long way in ensuring the success of this research study. Special mention goes to Prof. Kyallo Wamitila, Prof. Angelina Kioko, Prof. Arvi Hurskainen, Dr. Samuel Muindi, Dr. Zipporah Otiso, Rama Munara, Denis Ochachi, John Ogwae, Susan Kwamboka, Obed Mutiso, Joe Kyalo, Christopher Kithuka, Immaculate Wanza, and Christabel Lipuku.

Finally, special thanks go to the staff at the Computer Science Department at the Dedan Kimathi University of Technology, who encouraged me and made my work lighter as their team leader.

To all: Asante Sana or Asandi Muno or Mbuya mono

Abstract

Grammar development through the traditional rule-based method remains a challenge because the method is slow, time-consuming, expensive, knowledge-intensive, and laborious, particularly for under-resourced languages. Moreso, for the spoken Bantu languages. However, there is a high demand for these grammars for deep natural language processing, generation of well-formed output, or both, Controlled Natural languages Applications, and High precision machine translation. An in-depth review of previous research on improving grammar development reveals that these studies concentrated on rich-resourced languages and neglected under-resourced ones and have only concentrated on the syntax, ignoring the morphology in the shareable grammar. Therefore, there is an urgent need for cost-efficient methodologies that can accelerate grammar development to enable these languages to thrive in the digital ecosystem and minimize the language technology digital divide with the rich-resourced languages. Consequently, this research investigated an approach to reducing grammar development efforts for under-resourced languages in a rule-based multilingual environment by leveraging on cross-linguistic similarities to develop a congruent Bantu parameterized grammar and leveraging on the shared parameterized grammar to bootstrap Swahili grammar.

The descriptive analysis method was used to analyze descriptive grammar for each geolinguistics and purposively chosen Bantu languages to empirically identify the point of generalization of parameters, regular expressions and grammar rules. Furthermore, universal and individual comparative analyses were used to produce a generalized descriptive grammar for the subset of the Bantu languages. Then, quasi-experiments were set up in Grammatical Framework (GF) using the morphology-driven approach to develop the Bantu parameterized grammar utilizing grammar and to bootstrap Swahili grammar to the Bantu parameterized grammar. The GF regression method was used to test each grammar during development and reusability evaluation was done using shared and modified rules metrics for shareability and portability respectively while accuracy evaluation used a 100-English sentence test-suite.

The Bantu parameterized grammar shareability at morphology (parameters at 68.75% and paradigms at 65.3%) and syntax at 89.57%, while portability at morphology (14.29% at paradigms and 18.75% at parameter) and syntax at 10.43%. The bootstrapped

Swahili grammar had a shareability of at morphology (parameters at 68.75% and paradigms at 71.11%) and syntax at 91.41%, respectively, while portability at morphology (15.55% at paradigms and 18.75% at parameter) and syntax at 8.59%. In terms of accuracy, the grammars had 4-gram BLEU scores of 83.05%, 77.95% and 55.95% and WER of 12.82%, 13.39% and 23.90%, plus PER of 10.96%, 9.46% and 19.49% for Kikamba, Swahili and Ekegusii languages in that order. The research makes two conclusions, leveraging on the cross-linguistic similarities of principles and parameters significantly reduces multilingual grammars' development effort and leveraging on congruent grammar to bootstrap a similar grammar takes less effort since most of the rule-base will be inherited from the congruent grammar.

The study has several contributions. First, it has provided an approach of bootstrapping the development of multilingual grammar that significantly reduces the effort. Then extended GF reusability by providing standardized Swahili, Kikamba and Ekegusii grammars that are open resources. Furthermore, a hundred sentences test suite for the evaluation of grammars was created. Finally, by providing the missing parts through elicitation, mainly in the numeral, preposition fusion, and subject marker morpheme of the verb, a contribution was made to the descriptive grammar.

Keywords: Parameterized grammar, grammar engineering, bootstrapping, grammar sharing, grammar porting, complex morphology and under-resourced languages.

List of Abbreviations

AP	Adjective phrase
BLEU	Bilingual Evaluation Understudy
CN	Common Noun
CFG	context-free grammar
CSG	context-sensitive grammar
DCG	Definite clause grammar
DG	Dependency grammar
G	Formal grammar
GF	Grammatical Framework
RGL	Resource grammar library
HPSG	Head-Driven Phrase structure grammar
LE	Language Engineering
LFG	Lexical Functional Grammar
LRE	Language Resources Evaluation
MG	Montague grammar
NLP	Natural language processing
NP	Noun phrase
Oper	Operation
Param	Parameter
PER	Position Independent Error Rate
Pl	Plural
PMCFG	Parallel multiple context-free grammar
RG	regular grammar
RP	relative pronoun
Sg	Singular
SVO	subject-verb-object
TAG	Tree Adjoining Grammars
UG	Universal Grammar
VP	Verb phrase
WER	Word Error Rate

Table of Contents

Declaration.....	i
Dedication.....	ii
Acknowledgments	iii
Abstract.....	iv
List of Abbreviations	vi
Table of Contents.....	vii
List of Tables	xi
List of Example.....	xv
List of Definitions	xvi
Chapter 1 INTRODUCTION.....	1
1.1 Background.....	1
1.2 Problem Statement.....	6
1.3 Overall research question.....	7
1.4 Specific objectives:	7
1.5 Significance of the study.....	8
1.6 Scope of the study.....	9
1.7 Assumptions.....	9
1.8 Organization of the thesis	10
Chapter 2 LITERATURE REVIEW.....	12
2.1 Introduction.....	12
2.2 Universal Grammar.....	12
2.3 Comparative Descriptive Grammar	13
2.3.1 Bantu Languages Background.	13
2.3.2 Morphology.....	14
2.3.2.1 Noun	15
2.3.2.2 Adjective	17
2.3.2.3 Verbs.....	18
2.3.2.4 Closed categories.....	23
2.3.3 Syntax	24
2.4 Digital map for three Bantu languages	25

2.5 Approaches to Natural Language Processing	27
2.6 Grammar Engineering Approaches	29
2.6.1 Grammar porting.....	30
2.6.2 Grammar sharing	31
2.7 Formal Grammar.....	33
2.8 Grammar formalism.....	36
2.9 Grammatical Framework	39
2.9.1 Grammatical Framework Functor.....	43
2.10 Grammar Sharing and Porting Evaluation.....	44
2.10.1 Evaluation metrics	46
2.10.2 Error analysis	48
2.11 Conceptual Framework.....	49
2.12 Summary	51
Chapter 3 METHODOLOGY.....	52
3.1 Introduction:.....	52
3.2 Sampling.....	52
3.3 Comparative descriptive grammar development	53
3.3.1 Descriptive Case Study	55
3.3.2 Comparative Analysis.....	61
3.4. Bantu parameterized grammar development	63
3.4.1 Morphology.....	65
3. 4.1.1 Noun	67
3. 4.1.2 Adjective	71
3. 4.1.3 Verb	75
3. 4.1.4 Numerals	77
3. 4.1.5 Pronoun	80
3. 4.1.6 Preposition.....	82
3. 4.1.7 Quantifier.....	83
3. 4.1.8 Determiner	84
3. 4.1.9 Adverbs.....	85
3.4.2 Syntax	85

3.4.2.1 Common Noun (CN)	85
3.4.2.2 Determiner Phrases (Det).....	87
3.4.2.3 Adjective phrase (AP)	88
3.4.2.4 Noun phrase (NP)	90
3.4.2.5 Verb phrase (VP).....	94
3.4.2.6 Clauses.....	99
3.4.2.7 Sentences, Phrases, and Utterance.....	104
3.4.2.8 Coordination.....	107
3.4.2.9 Adverbs.....	107
3.4.3 Evaluation Test Suite	108
3.5 Bootstrapping Swahili Grammar Development.....	111
3.5.1 Morphology.....	113
3.5.1.1 Noun	114
3.5.1.2 Adjective	115
3.5.1.3 Verb and Verb phrase.....	116
3.5.1.4 Pronoun	118
3.5.1.5 Numeral.....	118
3.5.1.6 Other categories.....	119
3.5.2 Syntax	120
3.5.3 Bootstrapped Grammar Testing and Evaluation.....	125
3.6 Validation and Reliability.....	129
3.7 Summary.....	130
Chapter 4 RESULT AND DISCUSSION.....	131
4.1 Introduction.....	131
4.2 Comparative descriptive grammar.....	131
4.3 Bantu Parameterized Grammar Evaluation	133
4.3.1 Morphology shareability and portability	133
4.3.2 Syntax shareability and portability	136
4.3.3 Grammar Quality	137
4.4 Bootstrapping Swahili Grammar	140
4.5 The Generalized developed Bootstrap Approach	145
4.6 Effects of Errors on BLEU score	150

4.8 Previous studies	156
4.9 Summary	158
Chapter 5 CONCLUSION AND RECOMMENDATION	160
5.1 Introduction.....	160
5.2 Overview of the Research.....	160
5.3 Achievements.....	161
5.4 Contribution	163
5.4.1 Theoretical Contribution.....	164
5.4.2 Language technology resource tools.....	166
5.5 Conclusion	167
5.6 Recommendation	168
References.....	169
APPENDIX A Language consonants	179
APPENDIX B Resource Grammar Development	180
B.1 Noun	180
B.2 Verbs	181
B.3 Adjective	183
B.4 Numbers paradigms.....	191
APPENDIX C: TEST SUITES.....	196
C.1 Source in English	196
C.2 Gold standard for Swahili	197
C.3 Gold standard for Kikamba	198
C.4 Gold standard for Ekegusii.....	199
APPENDIX D Images	196
APPENDIX E Linearization categories.....	197
APPENDIX F Journal papers	198
APPENDIX G: Summary of ported Swahili grammar	199

List of Tables

TABLE 2.1 KENYAN BANTU GENDER.....	15
TABLE 2.2 MORPHOLOGY STRUCTURE OF KENYAN BANTU VERBS	19
TABLE 2.3 MOOD	22
TABLE 2.4 NOUN PHRASE STRUCTURE	24
TABLE 2.5 NLP TOOLS AND RESOURCES SURVEY	27
TABLE 2.6 MULTILINGUAL GRAMMAR EVALUATION METRICS	46
TABLE 3.1 LANGUAGES SAMPLING.....	53
TABLE 3.2 EKEGUSII VERB MORPHEMES PER TENSE	58
TABLE 3.3 GENERALIZED PARAMETERS AND PRINCIPLES.....	59
TABLE 3.4 KIKAMBA GENERALIZED RE AND GRAMMAR RULES	60
TABLE 3.5 EKEGUSII GENERALIZED RE AND GRAMMAR RULES.....	60
TABLE 3.6 GF CODING OF GENDERS.....	66
TABLE 3.7 NOUN GRAMMAR FRAGMENTS.....	68
TABLE 3.8 NOUN PARADIGMS	68
TABLE 3.9 FRAGMENTS OF ADJECTIVE GRAMMAR	72
TABLE 3.10 VERB PARADIGMS.....	77
TABLE 3.11 SUMMARY OF VERB GRAMMAR FRAGMENTS	77
TABLE 3.12 SAMPLE OF THE DEVELOPMENT SUITE (SOURCE GF ABSTRACT MODULES).....	109
TABLE 3.13 TREEBANK SYNTAX FUNCTIONS DISTRIBUTION	110
TABLE 3.14 TEST SUITE COVERAGE	111
TABLE 3.15 SWAHILI GENDER CODING	114
TABLE 3.16 SUMMARY OF BOOTSTRAPPING NOUN SEGMENTS	115
TABLE 3.17 ADJECTIVE PARAMETERS AND PARADIGMS	116
TABLE 3.18 SWAHILI INFLECTIONS FORMS.....	117
TABLE 3.19 SWAHILI PARADIGMS AND PARAMETERS	117
TABLE 4.1 GENERALIZED REGULAR EXPRESSION AND GRAMMAR RULES	132
TABLE 4.2 GENERALIZED PARAMETERS AND PRINCIPLES	132
TABLE 4.3 CONGRUENT GRAMMAR PARAMETERS	134
TABLE 4.4 PARADIGMS AND PARAMETERS PERCENTAGES	134
TABLE 4.5 CONGRUENT GRAMMAR PARADIGMS.....	135
TABLE 4.6 SHAREABILITY AND PORTABILITY	136
TABLE 4.7 TRANSLATION METRICS	137
TABLE 4.8 SWAHILI PARADIGMS	141

TABLE 4.9 SWAHILI PARADIGMS AND PARAMETERS	141
TABLE 4.10 BOOTSTRAPPED GRAMMAR SYNTAX RULES.....	142
TABLE 4.11 SWAHILI GRAMMAR ACCURACY.....	143
TABLE 4.12 SUMMARY OF BOOTSTRAPPED GRAMMAR	144

List of Figures

FIGURE 1.1 EXAMPLE OF A COMPLEX MORPHOLOGY WORD	3
FIGURE 1.2 LEXICAL SIMILARITY (SOURCE LEWIS, 2009)	4
FIGURE 1.3 LANGUAGE VENN DIAGRAM	5
FIGURE 1.4 SHARED AND UNIQUE GRAMMARS	5
FIGURE 1.5 SCOPE CATEGORY (SOURCE)	10
FIGURE 2.1 VAUQUOIS TRIANGLE (SOURCE DORR ET AL., 2004)	28
FIGURE 2.2 GRAMMAR PORTING	31
FIGURE 2.3 FREQUENCY OF LANGUAGE REUSE IN GRAMMAR SHARING	33
FIGURE 2.4 GF RGL MODULES	42
FIGURE 2.5 GF SYNTAXES	42
FIGURE 2.6 FUNCTOR MAPPING	44
FIGURE 2.7 TESTING PROCESS	45
FIGURE 2.8 ERRORS CLASSIFICATION IN GRAMMAR DEVELOPMENT	49
FIGURE 2.9 CONCEPTUAL FRAMEWORK	51
FIGURE 3.1 RESEARCH DESIGN	55
FIGURE 3.2 ADJECTIVE STRUCTURE	56
FIGURE 3.3 EKEGUSII VERB MORPHEMES	57
FIGURE 3.4 NEGATIVE POLARITIES FOR KIKAMBA VERB	57
FIGURE 3.5 EKEGUSII NOUN PHRASE	58
FIGURE 3.6 DESCRIPTIVE CASE STUDY METHODOLOGY	59
FIGURE 3.7 COMPARATIVE ANALYSIS PROCESS	62
FIGURE 3.8 ABSTRACT TREE	64
FIGURE 3.9 STEP BY STEP OF THE QUASI EXPERIMENT SETUP	65
FIGURE 3.10 EKEGUSII REGN PARADIGM	70
FIGURE 3.11 KIKAMBA REGN PARADIGM	70
FIGURE 3.12 WORD ALIGNMENT	78
FIGURE 3.13 CARDINAL FIVE	79
FIGURE 3.14 A LARGE CARDINAL NUMERAL EXAMPLE	79
FIGURE 3.15 NUMERAL IN EKEGUSII LANGUAGE	80
FIGURE 3.16 PRONOUN PARADIGM	81
FIGURE 3.17 PREPOSITION INFUSION	82
FIGURE 3.18 A PARSE TREE AND WORD ALIGNMENTS	87
FIGURE 3.19 DETERMINER EXAMPLE OF RULE ONE	88
FIGURE 3.20 EXAMPLE OF AP PARSE TREE AND WORD ALIGNMENT	90

FIGURE 3.21 NOUN PHRASE WORD ALIGNMENT	93
FIGURE 3.22 NP PARSE TREE IN EKEGUSII AND KIKAMBA RESPECTIVELY	94
FIGURE 3.23 KIKAMBA VP PARSE TREE	97
FIGURE 3.24 EKEGUSII VP PARSE TREE	97
FIGURE 3.25 VP PARSE TREE	98
FIGURE 3.26 SVO EXAMPLE.....	100
FIGURE 3.27 CLAUSE/SENTENCE PARSE TREE.....	102
FIGURE 3.28 DIRECT QUESTION	103
FIGURE 3.29 INTERROGATIVE QUESTION	103
FIGURE 3.30 UTTERANCE EXAMPLES.....	106
FIGURE 3.31 BOOTSTRAP STRUCTURE	112
FIGURE 3.32 BOOTSTRAP EXPERIMENT	113
FIGURE 3.33 CARDINAL NUMERAL EXAMPLE.....	119
FIGURE 3.34 CN EXAMPLE	121
FIGURE 3.35 EXAMPLE OF AP PARSE TREE AND WORD ALIGNMENT.....	121
FIGURE 3.36 NOUN PHRASE PARSE TREE AND WORD ALIGNMENT	122
FIGURE 3.37 VP EXAMPLE.....	123
FIGURE 3.38 CLAUSE EXAMPLE.....	123
FIGURE 3.39 QUESTION PARSE TREE.....	124
FIGURE 3.40 PHRASE EXAMPLE	125
FIGURE 4.1 POSITION INTERCHANGED ERROR.....	138
FIGURE 4.2 MANUAL ERROR ANALYSIS	138
FIGURE 4.3 KIKAMBA ORTHOGRAPHY ERROR	139
FIGURE 4.4 EKEGUSII ORTHOGRAPHY ERROR.....	139
FIGURE 4.5 EXAMPLE OF LEXIS ERROR	139
FIGURE 4.6 EXAMPLE OF VERB TENSES ERROR	140
FIGURE 4.7 SWAHILI GRAMMAR ERRORS.....	143
FIGURE 4.8 TENSE ERROR	144
FIGURE 4.9 WORD AMBIGUITY	144
FIGURE 4.10 ERROR CATEGORIES.....	151
FIGURE 5.1 ENGLISH TO BANTU LANGUAGES MACHINE TRANSLATION	162
FIGURE 5.2 KIKAMBA TO EKEGUSII MACHINE TRANSLATION	162
FIGURE 5.3 BANTU LANGUAGES MACHINE TRANSLATION.....	163
FIGURE 5.4 BOOTSTRAPPING SWAHILI.....	165

List of Example

EXAMPLE 2.1 NOUN STRUCTURE	16
EXAMPLE 2.2 ADJECTIVE STRUCTURE	17
EXAMPLE 2.3 POSITIVE FUTURE TENSE	19
EXAMPLE 2.4 NEGATIVE FUTURE TENSE	20
EXAMPLE 2.5 PAST TENSE	20
EXAMPLE 2.6 PAST TENSE	20
EXAMPLE 2.7 PRESENT TENSE	21
EXAMPLE 2.8 PRESENT TENSE	21
EXAMPLE 2.9 CONDITIONAL TENSE	21
EXAMPLE 2.10 CONDITIONAL TENSE	22
EXAMPLE 2.11 EKEGUSII NUMERAL	23
EXAMPLE 2.12 COMMENTS EXAMPLE	45
EXAMPLE 3.1 GENERALIZED RE OF DEMONSTRATIVE	62
EXAMPLE 3.2 LEXICON DEFINITION	67
EXAMPLE 3.3 COLOUR BLUE ADJECTIVE GF OUTPUT	73
EXAMPLE 3.4 PRONOUN "HE" OUTPUT	81
EXAMPLE 3.5 OUTPUT OF PREPOSITION "OF" USING MKPREP PARADIGM	83
EXAMPLE 3.6 QUANTIFIER OUTPUT	83
EXAMPLE 3.7 DETERMINER OUTPUT EXAMPLE	84
EXAMPLE 3.8 EXAMPLE OF CN RULES	87
EXAMPLE 3.9 PRE AND POST DETERMINER	92
EXAMPLE 3.10 PRONOUN OUTPUT	118
EXAMPLE 3.11 MANY LINEARIZATIONS OUTPUT	126
EXAMPLE 3.12 NP PARSING	126
EXAMPLE 3.13 VERB TO BE	127
EXAMPLE 3.14 AUXILIARY VERB	127
EXAMPLE 3.15 CONJUNCTION PARSING	127
EXAMPLE 3.16 DEMONSTRATIVE PARSING	127
EXAMPLE 3.17 STANDALONE NP	128
EXAMPLE 3.18 TWO SIMPLE SENTENCES	128

List of Definitions

DEFINITION 2.1 NOUN STRUCTURE	16
DEFINITION 2.2 REGULAR ADJECTIVE STRUCTURE	18
DEFINITION 2.3 NP STRUCTURE	24
DEFINITION 2.4 FORMAL GRAMMAR	34
DEFINITION 2.5 REGULAR EXPRESSION	34
DEFINITION 2.6 ABSTRACT SYNTAX	40
DEFINITION 2.7 GF DEFINITION.....	40
DEFINITION 2.8 PARALLEL CONCRETE SYNTAXES.....	40
DEFINITION 3.1 GENDER CODING IN GF	66
DEFINITION 3.2 AGREEMENT DEFINITION	67
DEFINITION 3.3 SMART PARADIGM FOR NOUN	68
DEFINITION 3.4 SHARED LOW-LEVEL PARADIGMS	69
DEFINITION 3.5 VERBS TO NOUN PARADIGMS.....	71
DEFINITION 3.6 SMART PARADIGM FOR ADJECTIVE	73
DEFINITION 3.7 VERB LINEARIZATION	75
DEFINITION 3.8 PARAMETER FOR VERB DERIVATIVE MORPHOLOGY.....	76
DEFINITION 3.9 SHARED NUMERAL DEFINITION AND PARAMETER	77
DEFINITION 3.10 NUMERAL RULES	79
DEFINITION 3.11 SHARED PRONOUN PARAMETERS AND LINEARIZATION	80
DEFINITION 3.12 PREPOSITION DEFINITION	82
DEFINITION 3.13 DETERMINER LINEARIZATION.....	84
DEFINITION 3.14 ADVERB DEFINITIONS.....	85
DEFINITION 3.15 CN DEFINITIONS.....	86
DEFINITION 3.16 CN RULE DEFINITIONS	86
DEFINITION 3.17 NP LINEARIZATION AND PARAMETER DEFINITION	91
DEFINITION 3.18 PRIMARY NOUN PHRASES RULES	91
DEFINITION 3.19 COMPLEX NOUN PHRASE RULES	92
DEFINITION 3.20 VERB LINEARIZATION AND PARADIGMS.....	95
DEFINITION 3.21 VERBS COMPLEMENTS PRODUCTIONS	96
DEFINITION 3.22 CLAUSE LINEARIZATION	100
DEFINITION 3.23 CLAUSE DEFINITION.....	101
DEFINITION 3.24 QUESTION CLAUSE.....	103
DEFINITION 3.25 SENTENCES PRODUCTIONS.....	104

DEFINITION 3.26 UTTERANCE.....	105
DEFINITION 3.27 MORE UTTERANCE PRODUCTIONS.....	105
DEFINITION 3.28 PHRASAL PRODUCTIONS.....	106
DEFINITION 3.29 CONJUNCTION PRODUCTIONS	107
DEFINITION 3.30 ADVERBS PRODUCTIONS	108
DEFINITION 3.31 EXAMPLE OF DIFFERENCE IN THE SUITES	110
DEFINITION 3.32 GENDER PARAMETER DEFINITION.....	114
DEFINITION 3.33 REGULAR NOUN PARADIGM.....	115
DEFINITION 3.34 ADJECTIVE PARADIGM DEFINITION	116
DEFINITION 3.35 PROGRESSIVE VERB DEFINITION	120
DEFINITION 4.1 THE APPROACH PSEUDOCODE.....	145
DEFINITION 4.2 SYNONYMS ERRORS	152
DEFINITION 4.3 PRONOUN PRO-DROP	152
DEFINITION 4.4 DISCONTINUOUS CONSTITUENTS DESIGN	153
DEFINITION 4.5 PROPOSE NEW RULES	154
DEFINITION 4.6 CONTEXT TRANSLATION	155

Chapter 1 INTRODUCTION

1.1 Background

Languages play a vital role in the current technological-knowledge-driven economies by enabling the creation, use, and distribution of knowledge and information. At present, there is a lot of information and knowledge in linguistic diversity due to the exponential growth of internet-connected digital devices (computers, smartphones, etc.) and the World Wide Web. Furthermore, language engineering (LE) plays a critical role in the acquisition and distribution of both diverse linguistic information and knowledge (Ghilic-Micu et al., 2011). LE is creating cost-effective, helpful and fast computer systems that recognize, understand, interpret and generate natural languages for a particular task. This happens by applying our language knowledge where both the process and output are predictable (in terms of effort and input for the process) and measurable (Cunningham, 1999; Maynard et al., 2002; Shaw & Garlan, 1996).

To build these computer systems, LE, in principle, uses two main approaches, namely, the classical rule-based approaches, also known as symbolic techniques and the state-of-the-art data driven approaches. The rule-based approaches consist of grammar rules based on the formal language of the Chomsky hierarchy, lexicon (monolingual, bilingual, or multilingual) and software to manipulate the rules. Rule-based approaches depend solely on language theory, thus providing high precision but low coverage unless an extensive dictionary of lexicons is created (Nadkarni et al., 2011; Jager & Roger, 2012). The data-driven approach is mainly a machine language model trained from a large annotated corpus (Cambria & White 2014). The corpus should be large enough to produce reliable results; thus, this approach is suitable when coverage is required within a short period. Such an approach includes but is not limited to statistical, probabilistic, neural networks, deep learning models, genetic engineering, among others.

Sometimes the two methods can be utilized at the same time resulting in a hybrid approach. The state-of-the-art data driven methodologies have been extensively relied upon in LE to create language resources and technology. However, due to their reliance on corpora to yield superb performance, such methods have been more successful when applied to richly resourced languages, such as Indo-European and Asiatic languages,

evidenced by the Language Resources Evaluation (LRE)¹² map and the white paper's series of META-NET³. This scenario offers dim hope to the under-resourced languages primarily found in the Asian and African continents, thus creating a language technology digital divide.

Under-resourced languages, also known as low density languages, have the following characteristics: they have few or no language tools and applications, no substantial presence on the Internet, have very little or no digitized corpora or digital text, no commercial interest since existing softwares have not been adapted for use, have few or no human language experts who can adequately document them. These characteristics make the languages technologically marginalized or disadvantaged (De Pauw, 2007; Kituku, 2015; Muhirwe, 2007). Muhirwe (2007) argues that most of these languages are found in developing countries where very little or no funds are allocated for natural language processing (NLP) research. Despite the above challenges, these languages have a substantial population of speakers who can form an economic hub. Therefore, the speakers should not be disenfranchised in the global language space. As a result of available digital devices and social media applications such as Facebook and WhatsApp, these languages will have a presence on the Internet. Therefore, these languages' digital visibility and viability should be enhanced by developing language resources and tools for them to compete in the technology-driven economies in equal terms with richly resourced languages (Krauwert. 2003). Additionally, UNESCO⁴ estimates that 95% of the world's languages would be extinct or endangered by the year 2100 hence the need for revitalization, preservation and documentation through language resources and technology development.

It would be an expensive affair to create language resources and tools for under-resourced languages using the data-driven approach in terms of creating large corpora, whether mono/bi/multilingual (collecting the data, translation, cleaning, annotation, alignment, etc.), human effort in annotation and time required to assemble the corpus.

¹ <http://lremap.elra.info/>

² <http://www.resourcebook.eu/> accessed 28th nov 2017

³ www.meta-net.eu/whitepapers/overview

⁴ <https://www.un.org/development/desa/indigenouspeoples/wp-content/uploads/sites/19/2018/04/Indigenous-Languages.pdf>

Besides, even if some corpus were available, a data-driven approach treats the word as the smallest unit, while in a language with complex morphology, a single word is characterized by several morphemes with distinct semantic marking. For example, the word “kilikimbia” (it ran) in the Swahili language has four morphemes, with each having a specific meaning, as exemplified in Figure 1.1. Therefore, since these data-driven approaches cannot capture the morphemes' dependence, this would lead to data sparsity.

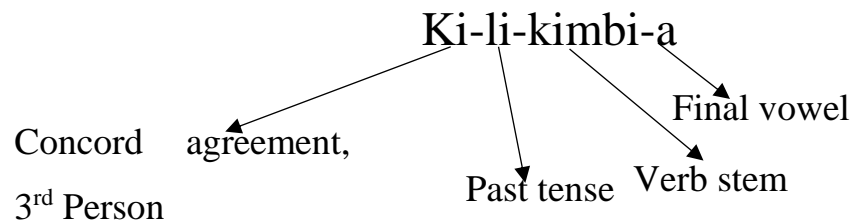


Figure 1.1 Example of a complex morphology word

The alternative is the rule-based approach that is a grammar engineering (GE) approach. GE is the process of using formal grammars to create a computational grammar with which machines can parse and/or generate. Building these language models requires a stable grammar formalism, a development toolkit for developing the grammar, an algorithm for processing the grammar and a test suite to assess developed grammar (Bender et al., 2008). However, performing GE for monolingual grammar is very slow and laborious (involves creating rules to generate the computational grammar to enable analysis and synthesis of the language(s) in question). The challenges posed by the two approaches have led to the re-thinking of innovative ways of developing computational grammars, language resources and technologies to accelerate the development cycle.

Interestingly, these under-resourced and/or complex morphology languages, especially in Africa, exist as families⁵ (Afro-Asiatic and Niger-Congo). For instance, the 42 Kenyan languages are divided into three major families, namely: Bantu⁶, Nilotic and Cushitic language families that are spoken by 65%, 32% and 3% of the population,

⁵ <https://www.ethnologue.com/statistics/family>

⁶ http://www.ijhssnet.com/journals/Vol_3_No_7_April_2013/28.pdf accessed 26th nov 2017

respectively (Obiero, 2008; Ogechi, 2003; Wamalwa, 2013). The ethnologue⁷ present 138 families spread all over the world with 54 families having over 10 languages in the set. The related languages within a family have cross-linguistic similarities and dissimilarities (Alansary, 2014; Lewis, 2009; Muhirwe, 2007) in line with the universal grammar concept of principles and parameters. The concept holds that grammars have common principles (structure features) but specific parameters with some values that control some surface phenomena (Chomsky,1981). To exemplify similarities, Figure 1.2 shows the lexical similarity between the Eastern Kenya Bantu languages. The highest lexical similarity is at least 70% between Gikuyu and Kiambu languages, while the least is at least 50% between Kikamba and Kimeru languages.

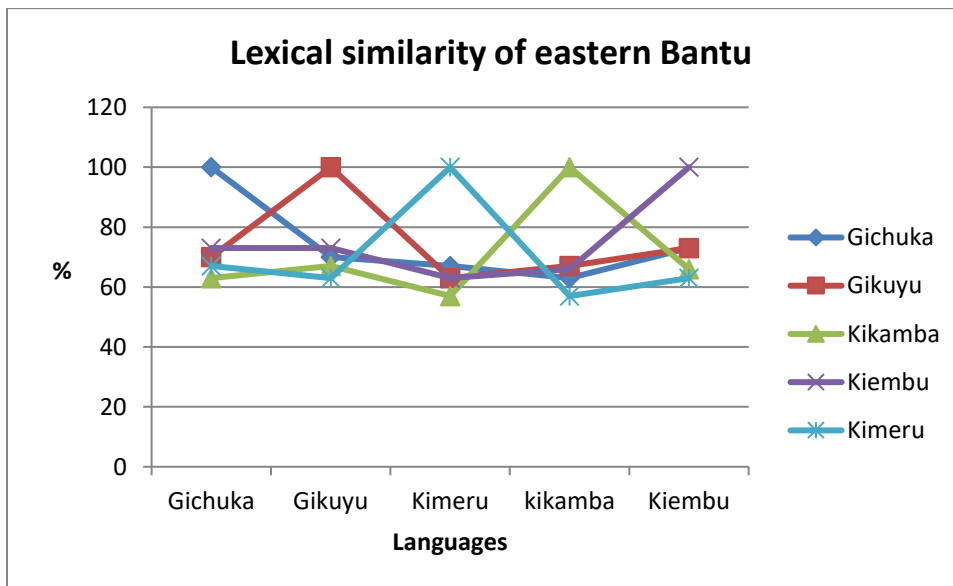


Figure 1.2 Lexical similarity (source Lewis, 2009)

These shared cross-linguistic principles and parameters can be utilized to achieve one of the main LE objectives, which is to develop shared language resources where the output becomes the foundation or support tool for the development of other Natural Language Processing (NLP) tools and resources through GE (Wright. 2002). So far, some GE attempts have been made; for example, the morphological analyzer made using the

⁷ <https://www.ethnologue.com/guides/largest-families>

rule-based approach by Pretorius and Bosch (2008) for Zulu and Xhosa languages as well as the use of grammar engineering strategies such as grammar sharing and grammar porting (Kim et al., 2003; Ranta ., 2009; Rayner et al., 2000; Santaholma, 2007). Grammar porting (also known as grammar adaptation) uses already developed grammar structures to develop a new but similar (same family) grammar. Only the structure of the grammar is shared; hence, rule modification is done to suit the new language. Grammar sharing is creating a commonly shared grammar (congruent) for all similar lexical, parameter and syntax rules of the family's languages (Santaholma., 2007). The shared grammar is formed from the union of cross-linguistic similarities in terms of syntax and morphology. In the Venn diagram shown in Figure 1.3, the intersection of the three languages $L_1 \cap L_2 \cap L_3$ in blue would represent shared grammar for the three languages. Besides, the pairs of languages L_1 and L_2 , L_1 and L_3 and L_2 and L_3 have an extra layer of shared grammar as shown by the colours orange, black, and yellow. Using extrapolation, with the addition of more languages up to language L_n , the shared grammar will be represented by $L_1 \cap L_2 \cap L_3 \cap L_4 \dots \dots \dots \cap L_n$. Therefore, developing a shared grammar using the grammar engineering strategies where additional grammar is added through bootstrapping reduces the development effort since it will inherit the congruent grammar as illustrated in Figure 1.4.

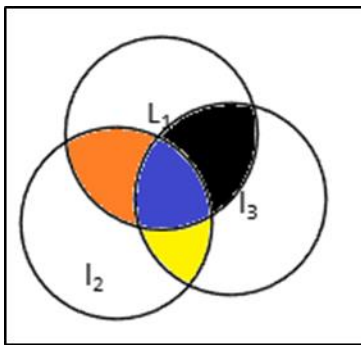


Figure 1.3 Language Venn diagram

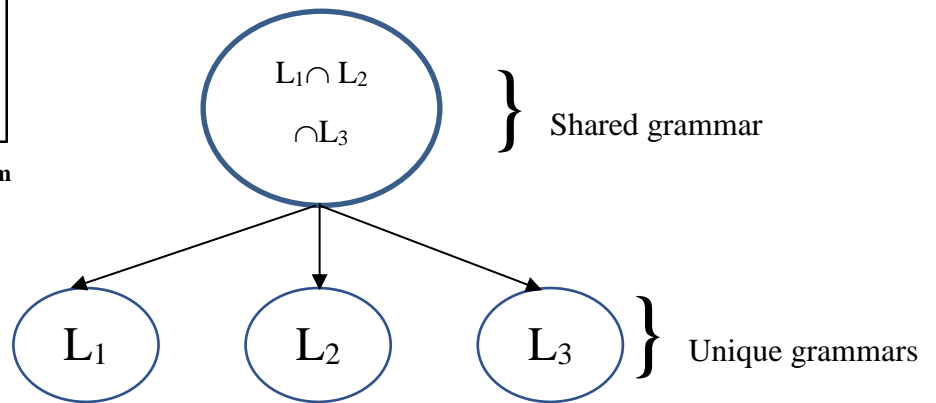


Figure 1.4 Shared and Unique grammars

The term bootstrapping is more often used in data-driven approaches and is defined as a framework for improving learning with minimal effort through leverage on a carefully chosen initial seed to find and add similar data, as training data from unlabeled data, via iterations process (Henderson, 2005; Jones et al., 1999). In using the rule-based bootstrap, the carefully chosen seed will be the shared grammar to be leveraged in bootstrapping the unique components of the grammar, thus reducing the development effort in terms of rule-base and time.

1.2 Problem Statement

Whether monolingual or multilingual, traditional rule-based grammar development is labour intensive in terms of time and knowledge requirements as well as slow, more so for under-resourced languages. Thus high development effort is an impediment to grammar development. Notwithstanding, these grammars are needed for deep natural language processing, generation of well-formed output or both. (Bender et al., 2008; Santahoma,2008), Controlled Natural languages Applications (Pretorius et al 2016, Santanloma,2008), High precision machine translation (Open et al., 2007; Lonning et al., 2006) etc. The data-driven approach does not help much in building language models for these languages because it treats the word (lexicon) as a single feature without considering morphemes with distinct meanings (multiple features of a word). Consequently, these probabilistic and statistical models do not capture dependencies in a word's morphemes resulting in data sparsity (Bender, 2009).

Furthermore, digitized corpora are a scarce commodity for under-resourced languages, especially for spoken Bantu languages rather than written languages (Ombui et al., 2014). The little available corpora may not be as helpful since they suffer from data sparseness to the extent that they cannot produce efficient and robust NLP tools. This issue may not be solved in the near future since much of the research focuses on rich-resourced languages because of economic and political aspects. In addition, these languages lack experts that can generate corpora (Gateo et al., 2006). Besides corpora, creation is a costly affair, especially for more often spoken languages with little or no digital written literature. One needs to get language experts and informants who will help in data collection, data

cleaning and annotation requires a lot of time, monetary resources, and human effort (Muhirwe, 2007; Santaholma, 2008). Therefore, rule-based approaches are more preferred for developing computational grammar, NLP tools, and applications for under-resourced languages with complex morphology.

An in-depth review of previous research on grammar engineering strategies geared towards improving grammar development reveals that these studies concentrated on rich-resourced languages and neglected under-resourced ones (Bender et al., 2008; Ranta, 2007; Rayner et al. 1996, 2000; Santaholma 2005, 2007, 2008). Secondly, they have only concentrated on the syntax, ignoring the morphology aspect in the shareable grammar. Consequently, this scenario has resulted in a wide language technology digital divide since most of the language models, language tools and applications are mostly for rich-resourced languages.

To solve the above challenge of grammar development, this study sought to find a sustainable, cost-effective and efficient methodology for developing grammars using a rule-based approach where a subset of Kenyan Bantu languages that are under-resourced, have complex morphology and many nominal classes is taken in a multilingual ecosystem. The research leveraged on cross-linguistic similarities, common principles, and parameters to develop congruent grammar (the bootstrap seed) via grammar engineering strategies. The efficiency and effectiveness of the congruent grammar was tested by bootstrapping Swahili grammar.

1.3 Overall research question

The study was guided by the overall research questions below:
What is the utility of developing Bantu parameterized grammar leveraging on cross-linguistic similarities of two or more Bantu languages? Furthermore, what is the reusability of the congruent grammar in bootstrapping a new Bantu language grammar?

1.4 Specific objectives:

In order to answer the overall research questions, the following four specific objectives were investigated.

- a. To investigate the degree of similarity of the principles and parameters between Kikamba and Ekegusii grammars

- b. To develop an approach leveraging on the shared grammar principles and parameters of Kikamba and Ekegusii grammars to produce the Bantu parameterized grammar.
- c. To bootstrap Swahili grammar into the Bantu parameterized grammar
- d. To evaluate the effectiveness and efficiency of the approach in reducing the development effort

1.5 Significance of the study

The researcher believes that the following stakeholders would benefit from the output of this research, namely: the researchers and academicians in Bantu languages, computational linguists, linguists, NLP tools developers and policymakers, as explained below.

Bantu language researchers and academic leaders would find this study useful because of the theoretical foundation of the literature review, resulting in an empirical comparative descriptive grammar for Bantu languages. The commonality established based on the universal grammar theory will be the basis for building computational grammar that can be used to validate the cross-linguistic similarities of principles and parameters.

The computational grammar developers will find this research valuable since the effectiveness and efficiency of the approach for bootstrapping the development of multilingual building grammar in terms of the rule-based effort can provide a blueprint on faster ways of developing grammars for under-resourced languages.

Moreover, NLP tools (grammars and machine translators) will provide language models for language analysis and generation in GF, especially for linguists and students of Bantu languages. Furthermore, these grammars can be used in developing domain grammars such as multilingual web gadgets, natural-language interfaces and dialogue systems. Finally, the researchers can generate a corpus for these under-resourced languages, a scarce commodity for further research using data-driven methods. The leading practitioners in this research area are translators and NLP tool developers. The study will provide a platform for multilingual translation for three less-resourced languages. Besides, it will also enable access to the over 40 languages already present in

the Grammatical Framework (GF), thus offering an extensive range of multilingual translations with high precision and reliability.

This research envisages that the NLP resource and tools developers will have been provided with a cost-efficient approach for accelerating computational grammar development for under-resourced languages that provides a base platform for making NLP tools and applications. Additionally, they will have a platform to develop controlled language tools on top of the Bantu parameterized grammar. This will create a channel through which information communication technology resources in these three languages can be actualized and interlinked with the other 40 plus languages already present within GF.

For developers interested in Bantu languages, the grammar will provide insights into how to characterize the common parts of Bantu grammar and accelerate bootstrapping a new grammar.

Policymakers in Heritage & Culture and Education ministries will get insights into how to preserve languages using technology and create localized software and applications, leading to better policies to deal with indigenous languages.

1.6 Scope of the study

The research sought to investigate Ekegusii and Kikamba descriptive grammars similarities based on universal grammar principles and parameters and thereafter develop Bantu parameterized grammar using grammar sharing and porting strategies. The sub-set of the Bantu languages is an example of a case study with aim of generalization in the future to other under-resourced language families. This grammar was then validated and generalized by bootstrapping Swahili grammar. The grammar development was limited to written text and developed up to phrase category as shown in Figure 1.5. The Bantu languages are used as a case of under resourced languages

1.7 Assumptions

This research study made the following assumptions:

That;

- languages differ in surface structure but share deep structures; hence they must have cross-linguistic similarities and common principles

- Bantu languages have dialects. Kimasaku and Rogoro dialects for Kikamba and Ekegusii languages respectively that have been used for written publication are one used in this research
- The informants and linguists who were used to elicit grammar for the categories or components that did not have the descriptive grammar and those cross-checked provided descriptive grammar to the best of their knowledge.

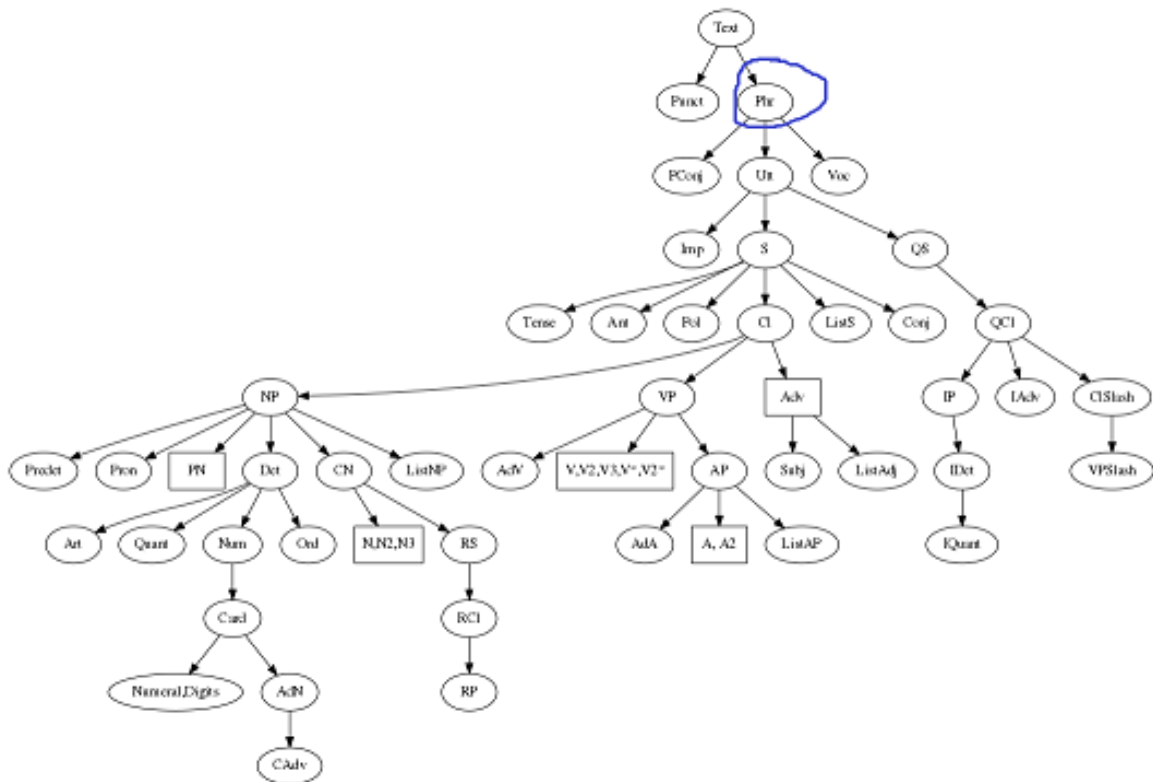


Figure 1.5 Scope category (source⁸)

1.8 Organization of the thesis

The remainder of the thesis is organized as follows.

Chapter two: This chapter extensively compares descriptive grammars establishing the nature and extent of cross-linguistic similarities among Ekegusii, Kikamba and Swahili

⁸ <http://www.grammaticalframework.org/lib/doc/synopsis/index.html>

languages. It also provides an overview of the current approaches to computational grammar and NLP resources development, clearly depicting the scenario of Bantu languages in Kenya. It also reviews grammar formalism and various bootstrapping strategies used in grammar development as well as examines the GF framework in detail and provides means of evaluating the resulting grammar. A proposed conceptual framework that guides the research is also presented.

Chapter three: The chapter explains the research design. Firstly, the chapter explains how sampling was done and how the comparative descriptive grammar was developed. Then an experiment is set up to design the Bantu parameterized grammar in GF using the bottom-up approach, How bootstrapping of the Swahili grammar was done is discussed. Finally, validity and reliability were ensured is demonstrated.

Chapter Four: The chapter presents the results of comparative descriptive grammar, evaluation of the Bantu parameterized grammar and Swahili at the morphology and syntax levels. It shows how development effort is reduced by using the grammar engineering strategies, then evaluates the resulting grammar accuracy using Bilingual Evaluation Understudy (BLEU), Word Error Rate (WER) and Position Independent Error Rate (PER) metrics, providing error analysis as well. A generalized approach for a new language is provided and Finally, a discussion of why the performance of grammars based on BLEU was not so high

Chapter Five:

The research overview is laid down with a major focus on explaining the main findings of the research and stating contributions made. Finally, a direction for future research is suggested.

Chapter 2 LITERATURE REVIEW

2.1 Introduction

This chapter extensively compares three languages' (Ekegusii, Kikamba and Swahili) descriptive grammars to establish the cross-linguistic similarities between them. The NLP resources and tools survey is undertaken to establish under-representation among the three Bantu languages. Subsequently, a review of the methodologies used to create these NLP tools and resources are examined to establish the best approach for developing the Bantu computational grammar for these under-resourced languages. A review of the Grammatical Framework is done and also the metrics for evaluating multilingual grammar. The chapter concludes by presenting a conceptual framework that guides the subsequent research steps.

2.2 Universal Grammar

Universal grammar (UG) is a language theory that is concerned with the computational systems of the mind on how to translate sound into meaning. The computational systems can map phonetic forms to logical forms. To achieve the above, the UG principles which direct structure and lexicon with all properties of words are used. Consequently, this theory brings the interaction of grammar, language and mind (Chomsky., 1981b; Cook & Newson., 2014).

The UG hypothesis is, language knowledge consists of general systems of principles that are universal and fixed parameters whose values differ from language to language. The parameters may include but are not limited to categories, rules and constraints (Dirven et al., 1982; Cook & Newson, 2014). At the abstract level languages have similar principles while at the concrete level, they differ because of the choice of the values for the parameters. Therefore, learning a new language involves applying the principles and setting the values of the parameters that apply to that particular language in question. When language data is analyzed based on Universal Grammar then the output should be the principles, parameters and lexicons in it (Cook & Newson, 2014). In the next section, We analyze the three Bantu languages' descriptive grammar based on this universal grammar theory to generate the principles and parameters.

2.3 Comparative Descriptive Grammar

This section examines the descriptive grammar of the three languages to bring up similarities between them.

2.3.1 Bantu Languages Background.

The Bantu language family consists of over 500⁹ languages spoken by over 240 million people across Africa (Van der Wal, 2015). These languages are agglutinative and tonal. In Kenya, Bantu languages are spoken by 65% of the population (Wamalwa et al., 2013). Apart from Swahili found in zone G, Kenyan Bantu languages are found in zone E of Guthrie's (1948) classification. The Great Lakes Bantu Languages (Wagner, 1970) are classified in group 40. These are Ligoli (E41), Ekegusii (E42) and Kuria (E43). The Eastern Bantu languages (McIntosh, 1968) are in group 50. They include Gikuyu (E51), Embu (E52), Meru (E53), Tharaka (E54), Kamba (E55) and Daiso (E56). The last group is the Coastal Bantu (Hinlebusch, 2007) languages that are found in two zones (E and G), according to Guthrie (1948). Zone E group 70 includes Pokomo (E71), Nika (E72 with dialects: Giriama, Rabai, Chonyi, Duruma, and Kauma), Digo (E73) and Taita (E74). Zone G has Swahili (G42), with the following dialects found on the Kenyan coast: Amu, Mvita, Mlima, and Unguja. This current study focuses on Swahili, Kikamba, and Ekegusii languages and shall represent the Kenya Bantu languages. The languages were chosen based on geolinguistics and availability of the descriptive grammar, experts and informants.

Previous research has demonstrated that the orthographies of Kenya Bantu languages are very similar. Orthography refers to the set of conventions (grapheme, diacritics, etc.) that encode a writing language (Kioko et al., 2012; Perfetti et al., 2005). In particular, the work on unifying orthographies between Gikuyu and Ekegusii languages (Mwangi., et al. 2013), Gikuyu and Kikamba languages (Kioko., et al, 2012), as well as the work on the eleven languages (Kipokomo, Mijikenda, Kuria, Gikuyu, Luhya, Dawida, Ekegusii, Kikamba, Embu, Meru, and Swahili) (Kioko., et al., 2012), clearly indicates

⁹ <https://www.britannica.com/topic/Bantu-languages> Accessed on 25th Nov 2017

similar vowels grapheme (a, e, i, o, u) for Ekegusii, Kikamba and Swahili. Besides the five vowel graphemes, Kikamba and Ekegusii languages have two extra ones that are ũ and ĩ. There are twenty-six, twenty and fifteen consonants in Swahili, Ekegusii, and Kikamba languages respectively (Iribemwangi, 2010; Kioko et al., 2012; Mwangi et al., 2013). These three languages share nine consonants. For more information on a specific language's consonants, see appendix A.1. Kikamba language has five dialects: eastern, central and north Kitui dialects, Machakos and Kilungu dialects (Mutiga, 2002). The Machakos dialect is the standard one and is mostly used in print and literature; hence, it is the dialect used for this study. There are two main dialects of Ekegusii languages: Maate and Rogoro dialects (Otiso, 2008). The research has used the Rogoro dialect by virtue of it being the standard Ekegusii dialect.

2.3.2 Morphology

Morphology can be defined as building words from morphemes or generating word forms (Jurafsky and Martin, 2009). A morpheme is a minimal unit that bears meaning in a particular language. It consists of the stem (the primary and key meaning of the word) and affixes (brings extra meaning when combined with the stem). Affixes include prefixes, suffixes, infixes, and circumfixes. The first two are highly common in Bantu languages morphology hence the tag 'prefixing and suffixing languages' by some researchers.

The Kenyan Bantu languages are agglutinative (prefixing and suffixing). Furthermore, combining the affixes and stem in some parts of speech tags is affected by the morpho-phonological transformation. The morphology uses the nominal¹⁰ class system (Ashton, 1947) that can be based on morphology (affix to a noun stem) or syntax (agreement affixes to verbs). The Ekegusii and Kikamba noun classes are based on morphology (noun prefix) (Basweti, 2005; Kaviti, 2004), while Swahili's noun classes are based on syntax agreement with verb (concord) (Njogu et al., 2006). However, it is worth noting that Swahili grammar previously had noun classes based on morphology (Ashton, 1947). Arguments have been forwarded about whether the nominal class system should be referred to as gender (way of categorizing nouns) or noun class. Some consider a pair of

¹⁰ <https://glossary.sil.org/term/noun-class>

singular and plural noun classes markers as gender (Hyman, 1979; Kihm, 2002, Di Garbo, 2014). The idea is reinforced by Demuth (2000) in suggesting that a noun class is a subset of gender. However, Ibrahim (1973) 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, thus supporting both ways. For this thesis' purpose, we adopt the view that a pair of noun classes (singular and plural) are to be regarded as gender, as shown in Table 2.1, and this approach has been adopted and used by Di Garbo (2014) in comparing over 100 languages across Africa. The nominal pairing and gender assignment have also been done by Katamba(2003). For example, in the gender *mu_a* for Kikamba language, the *mu* represents a singular noun class while the *a* represents a plural noun class and the underscore is used to pair them

Table 2.1 Kenyan Bantu Gender

Kikamba	Swahili	Ekegusii
mũ_a	a_wa	omo_aba
mu_mi	u_i	omo_eme
ĩ_ma	li_ya	e_ci
kĩ_i	ki_vi	eri_ama
ka_tũ	i_zi	ege_ebi
va_kũ	u_zi	oro_ci
n_n	u_u	aka_ebi
ũ ma	u_ya	obo_ama
u_n	ya_ya	oko_ama
kũ_ma	i_i	ama_ama
-	ku_ku	aa
	pa_pa	
	mu_mu	

The next section's discussion on morphology shall be based on open and closed categories. The former includes nouns, adjectives and verbs, while the latter comprises adverbs, prepositions, determiners (possessive and demonstrative), pronouns, interjections and numbers.

2.3.2.1 Noun

The noun structure for the three Bantu languages consists of prefixes paired to form inherent gender grammar features. For example, in Kikamba *mu_a*, the “mu”

represents the morpheme for the singular form while the “a” is for the plural form. These morphemes are obligatory.

In some cases, a gender can exist either in a singular or plural form, containing only a single morpheme. This happens with nouns that deal with locations and those that do not have singular forms like the noun *water*. Example 2.1 shows the usage of gender using the word “chair” in all the languages. Ten genders are identified for the Kikamba language (Kaviti, 2004; Mbuvi, 2005; Welmers, 1973); Swahili has thirteen genders (Deen, 2002; Njogu et al., 2006) while Ekegusii has eleven genders (Basweti, 2005; Ongarora, 2008; Osinde, 1988). They are all listed in Table 2.1. The suffix comes as a result of a nominative noun that is fused with a preposition. For example, the phrase “on the bed” is “kitandani” in Swahili, thus the suffix “ni” is concatenated with the noun. Therefore, the preposition “on” is represented by the morpheme “ni” that is fused to the noun “the bed.” Kikamba and Swahili use the morpheme “ni” and Ekegusii use “me” for this fusion suffix. Definition 2.1 represents the generalized structure of regular nouns in the three languages. The prefix encodes gender plus number, the stem also called radical, and an optional suffix for preposition fusion.

Example 2.1 Noun structure

Language	Singular	Gender	Plural	Gender
Swahili	Ki-ti	Ki_Vi	vi-ti	Ki_Vi
Kikamba	Ki-vila	Ki_i	i-vila	Ki_i
Ekegusii	Ekerogo	Ege_Ebi	Ebirogo	Ege_Ebi
Gloss	Chair		Chairs	

Definition 2.1 Noun structure

Prefix ++ Stem ++ Suffix

Morphophonemics is a process of phonological alternations and modifications when morphemes combine and do happen in Bantu languages at the noun level (Otiso., 2008). In Kikamba the class genders mu_a and mu_mi have the following two alternate rules and the same applies for Swahili in class genders a_wa and u_i. In both cases, they happen in number with the value singular while in Ekegusii it can happen in both values of the numeral. Mostly this morphophonological happens in the class gender that deals with animate and plants. (Komenda et al., 2013; Onkwani’, 2011, Njogu et al., 2006)

Kikamba language

mu_a u-a becomes wa for example Mu-anake becomes Mwanake. (gloss child).

mu_mi u-i becomes wi, for example, mu-ii becomes mwii. (gloss body)

u-e becomes we for example, Muei becomes mwei. (gloss moon)

Swahili Language

a_wa u-a becomes wa for example Mu-ana becomes Mwana. (gloss child).

Mu_Mi u-i becomes wi, for example, mu-ili becomes mwili. (gloss body)

u-e become we for example Muezi becomes mwezi. (gloss moon)

Ekegusii Language

Omo-aba o-u becomes wa for example Omu-ana becomes omwana. (gloss child).

omo_eme o-o becomes wo, for example, emo-osi becomes emwosi. (earthworm)

e-o becomes joo for example eme-osi becomes emjoosi. (gloss earthworms)

2.3.2.2 Adjective

An adjective is a noun modifier and in the three languages, it consists of a prefix (concord), which must agree with the gender of the noun to be modified and is concatenated with the adjective root. In the three languages, the concatenation is influenced by the morpho-phonological rules of the specific language. Example 2.2 below demonstrates the adjective structure, whereby the noun gender determines the adjective prefix and *adjroot* represents the radical based on the available literature (Basweti, 2005; Deen, 2002; Kaviti, 2004; Mbuvi, 2005; Njogu et al., 2006; Ongarora, 2008; Osinde, 1988; Welmers, 1973). Definition 2.2 below shows an adjective-generalized regular expression for the three Bantu languages.

Example 2.2 Adjective structure

Language	Singular	Plural
Kikamba	Mu -ti mu -nini mu _mi -gender nini-Adjroot	Mi -ti mi -nini mu _mi -gender nini-Adjroot
Swahili	M ti mu -dogo m _mi –gender dogo -Adjroot	Mi -ti mi -dogo m _mi –gender dogo -Adjroot
Ekegusii	O mo -te omo -nke O mo_eme –gender nke -Adjroot	E me-te eme -nke O mo_eme –gender nke – Adjroot
Gloss	Small tree	Small trees

Definition 2.2 regular adjective structure

Prefix (concord) ++ Adjroot

2.3.2.3 Verbs

Verbs in Bantu languages have a complex morphology with much prefixing and suffixing plus infixing for extensional morphology. Its declension involves several morphemes (several prefixes, root, extensional suffix, and final vowels representing mood) plus some grammar features such as person, number, gender, tense, and polarity. The morphemes of verbs embody all the constituents needed to make a sentence. Hence, a verb can act in place of a sentence. Table 2.2 (Basweti, 2005; Deen, 2002; Kaviti, 2004; Mbuvi, 2005; Njogu et al., 2006; Ongarora, 2008; Osinde, 1988; Welmers, 1973) summarizes all the prefixes, suffixes, roots and extensions needed to form verbs in the languages. The “-” or empty space means the suffix or the prefix does not exist in that language. The subject marker represents positive polarity, while the negation morpheme is indicative of negative polarity. Both have grammar features of gender, number, and person that form the agreement parameter. It is essential to note that the following fields are usually not obligatory: relative marker, object marker, infinitive, and extension. The focus morpheme cannot exist with negation (Munyao, 2006; Njogu et al., 2006; Ongarora, 2008).

The tense for Bantu languages is marked by a tense morpheme or no morpheme at all. Three points are needed to mark different tenses, as argued by Reichenbach (1947). These points are the speech point, the reference point and the event point in relation to time, while time is based on the speech point (Munyao, 2006). The coincidence of the three points results in the present tense. When the speech point is after the other two points, then the past tense occurs. Future tense occurs when the speech point is before other points. Finally, when the reference time proceeds to event time, the resultant is a perfect tense.

The aspect gives a view of the verb's action, such as beginning, continuing, or ending (Munyao, 2006). Most of the time, tense and aspect are combined in Bantu languages. Several tenses exist in the Ekegusii, Kikamba and Swahili languages (Basweti,

2005; Deen, 2002; Munyao, 2006; Njogu et al., 2006; Ongarora, 2008; Osinde, 1988; Otiso, 2008; Welmers, 1973). For this discussion, we focus on the 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, *Fw* for the final vowel and *Aux* for the auxiliary verb.

Table 2.2 Morphology structure of Kenyan Bantu Verbs

Structure	Morpheme	Kikamba	Swahili	Ekegusii
Prefixes	Focus	“ni”	‘ni’	“n”
	Negation	All languages as per gender, number and person		
	Subject marker	All languages as per gender, number and person		
	Tense/Aspect	All languages as per tense		
	Relative marker	-	As per class	-
	Object marker	All languages as per gender and number		
	Infinitive	“ku”	“ku”	“ko”
Root		Root	Root	Root
Extension	Applicative	“i’	“ e/i“	“er “
Suffix	Causative	” ithy”	“ ish/esh“	“i’
	Passive	” w”	“w “	“ u“
	Reversive	” u”	“u/ul”	“or“
	Reciprocal	” an”	“ an“	“ an“
	Stative	-	“ik:	“ek”
Final vowel		“a/e”	“a/e/i”	“a/e/i”

The future tense in the Kikamba language is marked by the morpheme “ka” (Munyao, 2006), while in Swahili is marked by the morpheme “ta” (Njogu et al., 2006). As for the Ekegusii language, the suffix “e” (Whitely, 1965) marks the future tense though Ongarora (2008) argues that the morpheme “e” in Ekegusii does not represent tense as shown by Examples 2.3 and 2.4 below.

Example 2.3 Positive Future tense

Language	Positive Polarity
Kikamba	A- ka- kom- a <i>Agr Tns Root Fw</i>
Swahili	A – ta - lal a <i>Agr Tns Root Fw</i>
Ekegusii	a- gocha go- rar -e <i>Agr Aux infix morpheme Root Tns</i>
Gloss	He will sleep

Example 2.4 Negative future tense

Language	Negative Polarity
Kikamba	Nda- ka- kom- a <i>Agr Tns Root Fw</i>
Swahili	ha – ta - lal a <i>Agr Tns Root Fw</i>
Ekegusii	Tari ko rar a <i>Agr Root Fw</i>
Gloss	He will not sleep

In the past tense, the morphemes “li” and “a’ are used in Swahili and Kikamba languages respectively. In the Ekegusii, morpheme “a” marks the immediate and hesternal past tense while morpheme “ete” marks distant and hodiernal past tense. Examples 2.5 and 2.6 below demonstrate the past tense. The difference within tense marked by the same morpheme is communicated through tone.

Example 2.5 Past tense

Language	Positive Polarity
Kikamba	Ni- ma- na- semb- ĩe <i>fs Agr asp Root (Fw& tns)</i>
Swahili	Wa li kimbi a <i>Agr Tns Root Fw</i>
Ekegusii	Ba- a- minyok- a <i>Agr Tns Root Fw</i> Ba- a- minyok- ete <i>Agr Tns Root Tns & Fw</i>
Gloss	They ran

Example 2.6 past tense

Language	Negative Polarity
Kikamba	Ni- ma- na- semb- ie <i>fs Agr asp Root (Fw& tns)</i>
Swahili	Hawa kumbi a <i>Agr Root Fw</i>
Ekegusii	Mba- minyog- ete <i>Agr Root Tns & Fw</i>
Gloss	They didn’t run

Though Bantu languages exhibit dichotomy regarding tenses (past versus nonpast) (Ongarora, 2008), they have present tense, exemplified by habitual tense or progressive tense. Examples 2.7 and 2.8 below show the present tense of the languages.

Example 2.7 Present tense

Language	Positive Polarity
Kikamba	Ni- u- ãs- aa Fs Agr is Root Fw \$Tns
Swahili	A- na- kul- a/ Hu- la Agr Tns Root Fw/ habitual morpheme root
Ekegusii	A- ko- raager- a Agr tns Root Fw
Gloss	He is eating /He eats

Example 2.8 Present tense

Language	Negative Polarity
Kikamba	Nda- ãs- aa Agr Root Fw \$Tns
Swahili	A- kul- i/ Ha- li Agr Root Fw/ habitual morpheme root
Ekegusii	Tari ko- rager- a Agr tns Root Fw
Gloss	He isn't eating

Finally, Examples 2.9 and 2.10 below exemplify the Conditional tense in all polarities

Example 2.9 Conditional tense

Language	Positive Polarity
Kikamba	Ni- twa- kom- a Fs Agr Root Fw
Swahili	Tu- ngali- lal a Agr Tns Root Fw
Ekegusii	Nto- ko- rar- a Agr tns Root Fw
Gloss	We have slept

Example 2.10 Conditional tense

Language	NegativePolarity
Kikamba	Tui- na- kom- a Agr tns Root Fw
Swahili	Ha- tuja - lal- a Agr Tns Root Fw
Ekegusii	Nto- ko- rar- a Agr tns Root Fw
Gloss	We haven't slept

Though much research has been done on verb morphemes, the subject and object markers are only documented for gender, which deals with animate objects. Therefore, elicitation was done to generate the subject markers for the other genders.

The final vowel of a verb represents the mood and it is defined as the speaker's attitude and belief toward the probability or actualization of an event/situation(Otiso 2008). The three languages have similar moods namely indicative, subjunctive, conditional and imperative moods (Basweti, 2005; Deen, 2002; Munyao, 2006; Njogu et al., 2006; Ongarora, 2008; Osinde, 1988; Otiso, 2008; Welmers, 1973). The indicative mood is used in declarative or assertion statements to represent a realistic situation and is the basic mood. The verb root is accompanied by a subject marker and focus and usually, the final vowel is “a” for this mood in all languages. The Subjunctive mood is an expression of permission or probability of the event, the polite form in Swahili as the three alternatives for the final vowels namely: “i” “e” and “u”. In kikamba and Ekegusii, the final vowel is “e”. In conditional mood, in kikamba, the prefix ka is used plus indicative mood final vowel, while in Ekegusii it uses the morpheme “ra” together with the infinitive morpheme. The imperative mood is a command in relation to the event. has the final vowel “a” and the verb root. The moods are exemplified in Table 2.3

Table 2.3 Mood

Mood/language	Indicative	Subjunctive	Conditional	Imperative
Kikamba	a - im-a	u - im-e	nu- ku-im-a	im-a
Ekegusii	n –a-rem-a	o-rem-e	ko ra rem- e	rem- a
Swahili	a –ka-lim-a	u –lim-e	u ta –lim-a	lim-a
Gloss	he/she dug	you dug	if you dig	Dig

2.3.2.4 Closed categories

Possessive pronouns modify a noun to show ownership. Their structures for the three languages consist of prefix morpheme, which agrees with the gender of the noun while the root has the grammar feature of number and person. Also, their structure is similar to that of adjectives. Personal pronouns stand in place of an absent noun and the animate gender has pronouns for the first, second and third person. The rest have only a third person with unique strings in Bantu languages corresponding to the personal pronoun “it” in English. When a personal noun appears as the subject of a sentence, they are dropped (pro-drop) since they are encoded in the verb's subject marker (Basweti, 2005; Kaviti, 2004; Njogu et al., 2006).

Demonstratives are noun modifiers that show how far object(s) is/are from the speaker. Indo-European languages demonstrate strings for near and distant as opposed to Bantu languages which have an extra string for the aforementioned demonstrative (Basweti, 2005; Kaviti, 2004; Njogu et al., 2006). The demonstrative string has variable features of gender and number.

Adverbs, interjections, and prepositions have a string that is independent of gender. There is an exception to the preposition “of” in all three languages in which the string agrees with the gender. Numbers, too modify the noun. Generally, cardinal numbers one to five have a prefix in Bantu languages based on gender agreement; besides, the root and their structure are similar to the adjective. The strings are independent of gender for Swahili and Kikamba languages from six to nine except for the number eight in the Swahili language. There is no numeral six to nine in the Ekegusii but a repetition similar to that of one to five, based on the gender of nouns that have been modified as exemplified in Example 2.11 below.

Example 2.11 Ekegusii Numeral	
emete etano ebere	abarwaria emerongo etano babere
trees five two	Doctors tens fifty twenty
seven trees	Seventy doctors

Therefore, it is like adding two numbers between one and five to get a number between six and nine. There is a disjunctive string for ordinal numbers before the cardinal

number except for numbers one to three, which have unique ordinal writing and are based on gender agreement. The ordinal and cardinal are available in digit and numeral form.

2.3.3 Syntax

The dominant topology for the Bantu language sentence is subject-verb-object (SVO) (Basweti, 2005; Bitutu, 1991; Deen, 2002; Kaviti, 2004; Marten, 2013; Mose, 2012; Munyao, 2006) whereby the subject is a noun phrase, followed by a verb phrase. The verb phrase is made up of a verb and object complement that can be a verb phrase, noun phrase or both. The object's presence 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 genders (Basweti, 2005; Kaviti, 2004; Marten, 2013).

A noun phrase (NP hereafter) is made of a noun and its modifiers that include adjectives (Adj), determiners (Det), both possessives (Poss) and demonstratives (Dem) and numbers (Num). Table 2.3 below shows NP's structure in the three languages (Basweti, 2005; Mbuvi, 2005; Rugemalira, 2007).

Table 2.4 Noun phrase structure

Swahili	[Dem] [Noun] [Det <Poss> <Dem>] [[Num] [Adj]]
Kikamba	[Noun] [Dem] [Poss] [Num] [Adj]
Ekegusii	[Noun] [Dem] [Poss] [Quant] [Adj]

Table 2.4 shows that from the literature, the Ekegusii language structure lacks numeral and demonstrative determiners before the noun, while the Kikamba lacks the latter. However, these were found to be necessary for the structure through the elicitation method by linguists and experts in the two languages. As a result, the overall NP structure is the one fronted by Rugemalira (2007) for Bantu languages in definition 2.3 below.

Definition 2.3 NP structure

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

The structure represents a complex NP, The symbols [] the modifiers are optional (they can occur or not but when they occur all they must follow the specified order). The

demonstratives can only occur once in a statement whether a pre or post-modifier both cannot occur at once. The symbol < > means though two options only one can occur at any time in a sentence. A simple NP can only be formed either by a noun or a pronoun. It is also possible to form a complex noun by using post modifiers to the noun phrase - mainly interrogative and past participle of a verb. The verb phrase structure is the same as a verb and carries all parameters that are integral to verbs. The VP can be used in a minimal sentence where the subject markers and object marker slots of the verb are filled resulting in a sentence with topology V. An extended sentence occur when the object after the verb is present. Therefore, the VP is made of a verb and an object. The verb phrase can also have verb and adverb, verb and noun phrases. Furthermore, it is also possible to have two verb phrases making a VP. The auxiliary verb “to be” is also used before the main verb (Marten, 2013) to form a verb phrase. Finally, extension declension also extends the valency of a basic verb to higher valency thereby requiring objects (Deen, 2002).

2.4 Digital map for three Bantu languages

In reference to the LRE¹¹ ¹² map, Bantu languages are under-represented in the digital arena. The statistics show that only Swahili appears with two resources while the top language (English) has 961 linguistic resources. However, we found other NLP resources not shown on the LRE map, as summarized in Table 2.5 Hurskainen (1992) and Lipps (2011) developed a Swahili morphology analyzer using a finite-state approach. Moreover, De Pauw et al. (2008) have also developed a morphology analyzer using a data driven-approach. Nganga (2012) developed a morphology analyzer using GF in addition to word sense disambiguation (Nganga, 2005). Four dictionaries exist for Swahili languages (De Pauw et al., 2009a) namely: Internet living Swahili dictionaries¹³, Freedict Swahili to English dictionary¹⁴, The TshwaneDJe Swahili–English Dictionary and the TUKI Swahili–English Dictionary. The following monolingual corpora exist for Swahili: Helsinki Corpus of Swahili, The TshwaneDJe Swahili Internet Corpus and The Swahili

¹¹ <http://www.resourcebook.eu/>

¹² <http://lremap.elra.info/>

¹³ <https://www.merlot.org/merlot/viewMaterial.htm?id=75705> (accessed on 7th Sept 2018)

¹⁴ <https://www.freedict.com/onldict/swa.html>

part of the parallel SAWA corpus with nine million, twenty million and a half million words respectively (De Pauw et al., 2009b). Finally, there exists a bilingual machine translation between Ekegusii and Swahili based on the Carabao framework (Ombui. et al., 2014) and the Google¹⁵ translation system available online. Speech to text system for Swahili is also available (Getao et al., 2006). An Interlingua rule-based machine translation for the language's Swahili and Ekegusii (Ombui et al., 2014) and a morphology analyzer (Elwell, 2006). Kikamba language has a part of speech tagger (Kituku. et al., 2015), name entity recognizer (Kituku et al., 2011) and Kikamba dictionary (Mwau, 2006). Despite the languages having a high-speaking population, they have few language technology resources and tools. From the survey, it is clear, the already available NLP tools and resources rotate around a few experts, which presents a human expert problem in tackling the daunting challenge of developing NLP tools and resources. Furthermore, even in our higher learning institution in Kenya, there are no established NLP laboratories that can transfer knowledge and skills and create more experts; thus, these under-resourced languages will continue to be technologically disadvantaged. If this trend continues, future internet presence and commercial interest in these languages are bleak. To bridge this language technology divide, there is a need to think about and create sustainable multilingual language technology tools that interconnect and use the languages with more NLP tools and resources as well as a development process that requires less effort. This would increase the under-resourced language's NLP tools and application usage, increase their internet presence, elicit commercial interest and finally enable the languages to compete equally in today's knowledge-driven economy.

¹⁵<https://translate.google.com/#view=home&op=translate&sl=auto&tl=en&text=wewe%20waja>

Table 2.5 NLP tools and resources survey

Language/Tool	Swahili	Ekegusii	Kikamba
Morphology analyzer	Hurskainen (1992) Nganga (2012) Lipps (2011) De Pauw et al. (2008)	Elwell, 2006	
word sense disambiguation	(Nganga, 2005).		
Dictionaries	(De Pauw et al., 2009)		(Mwau,2006)
Parallel Corpus	Helsinki Corpus SAWA corpus (De Pauw 2009).		
Translation	(Ombui et al., 2014) Google14 translation	(Ombui et al., 2014)	
Grammars			
Part of speech tagger	(Hurskainen, 2004 ; De Pauw et al., 2006)		Kituku. et al., 2015)
Name entity recognizer	Shah et al., 2010		Kituku et al., 2011).

2.5 Approaches to Natural Language Processing

Deductive and inductive approaches are the two main paradigms used to develop Natural Language Processing resources. However, recently there is a trend to harness the advantages posed by both paradigms yielding a hybrid model. The deductive model, also known as the rule-based method, uses linguistic knowledge of the specific language and consists of handcrafted grammar rules and a monolingual /bilingual /multilingual dictionary (Antony, 2013). Rule-based resources are pegged on the Vauquois triangle shown in Figure 2.1 below. The triangle has three levels of analysis and generation namely: morphology, syntax, and semantic analysis. These grammar rules (rule-based) are modeled using grammar formalism. The approach has several advantages such as: easy to maintain and extend the language, can deal with varieties of linguistic phenomena, its output has high precision because it is language-specific and uses well-formed sentences and the

linguistic knowledge gained can be used to build a new related system. However, two shortfalls are evident: first, it requires a skilled linguistic expert to provide the knowledge for crafting the rules and second, unless the dictionary is expanded, its coverage is very narrow. Additionally, investment in terms of time and effort to build the system is required.

The inductive paradigm, also known as the data-driven approach, uses the power of statistics and probability. It takes the form of machine learning algorithms to learn from (parallel) annotated corpora in order to be able to predict classes or categories. It takes a short time to develop a system that can be scaled up quickly leading to high coverage. However, it requires a large (parallel) annotated corpus for it to produce a significant performance, which requires a lot of human effort (Zeroual et al.,2018). It is also affected by the curse of sparseness, making it hard to generalize the data.

The hybrid¹⁶ model taps the high precision of rule-based approaches and wide coverage of data-driven approaches. There are two types of hybrid systems - the rule-based guided hybrid system and the data-driven guided hybrid system (Costa-Jussa et al., 2015). In a rule-based guided hybrid system, grammar rules are extracted from the corpora using data-driven methods such as deep learning. The rule-based dictionary is also enhanced by the corpora hence reducing developing time (Costa-Jussa et al., 2015; Socher, 2014). As for the data-driven guided hybrid system, grammar rules are introduced at the pre/core/post-processing stages, which involves dynamically integrating syntax and morphology knowledge in terms of rules to the data driven system.

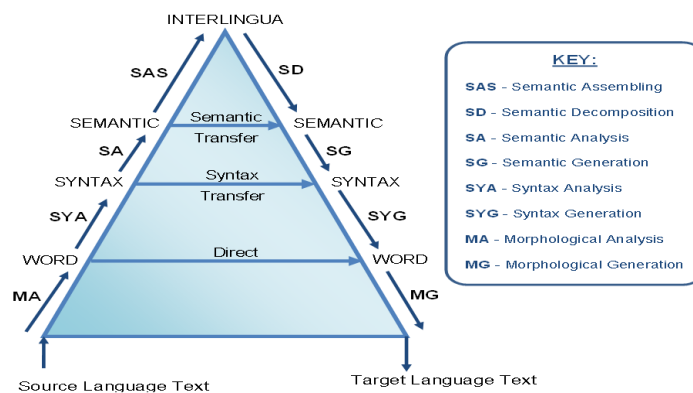


Figure 2.1 Vauquois triangle (Source Dorr et al., 2004)

¹⁶ <https://blogs.sas.com/content/subconsciousmusings/2020/09/09/nlp-the-hybrid-approach/>

Even though it takes less time to develop NLP tools and resources and it would be the best method to use, the data-driven method is not feasible/is impractical for less-resourced languages, especially the complex morphology Bantu languages. The primary reason for this is because digitized corpora are not available and the corpora which might be available suffer from data sparseness, particularly for the morphologically rich languages (Shaalán, 2010) such as Bantu languages. Zeroual et al (2018) argue that before any meaningful work can be done using the little corpora available, it takes much time, funds, and effort to collect and annotate the corpora. Additionally, Ombui et al. (2014) point out that Kenyan languages are primarily spoken as opposed to written; thus, it becomes expensive to collect enough corpora for system development. The above issues persuade full or partial adoption of rule-based approach methods by NLP researchers in developing tools for less-resourced languages and/or with complex morphology. For example, Arabic languages are leading in this venture (Shaalán, 2010). However, to reduce the development effort, in this research, we further adopt grammar engineering approaches in rule-based approaches such as grammar sharing and porting (Kim et al., 2003) in developing a multilingual NLP grammar resource. These techniques involve developing grammar for one or more language(s), then leveraging on the developed grammar(s) (acts as a bootstrap seed) to develop grammar for a new language(s). Either the rules are shared without changing any aspect, or only the structure of the rules is shared depending on the grammars' cross-linguistic similarities in consideration. This reduces the effort needed to develop the rule-base for new grammar.

2.6 Grammar Engineering Approaches

Universal Grammar (UG) theory (Bender et al., 2008; Wang, 2009) shows that the expressive capabilities of natural language grammars are equivalent and have similar basic parameters and principles, more so for a family's languages. Therefore, a lot of cross-linguistic information is shared. This shared information can be leveraged in developing multilingual grammar by developing congruent grammar instead of developing parallel monolingual grammars. This would reduce the development effort in terms of the rule-base's size, make maintenance easier and bring about standardization in the rules. The two

grammar engineering strategies that have been used to share grammar (Alshawi et al., 1992; Bender et al., 2002; Santaholma, 2007) are grammar porting, also known as grammar adaptation and grammar sharing. These strategies have the following advantages:

- When the grammar rules are shared among different languages' grammars, then the size of the code to represent the rule-base is reduced significantly, thereby reducing the time to compile or run grammars (space complexity)
- Due to the shared grammar rule-base reduction, the grammar's development time is reduced significantly since a small rule base is being developed.
- The grammars development makes use of a common features description and standard convention of naming, resulting in coherent grammar description
- The large the grammar rule-base, the harder it becomes to maintain the rules. Therefore, since the grammar engineering approaches reduce the rule-base, it also reduces the effort need to maintain the grammar rules.
- Redundancy of duplication effort is eradicated since grammars are either developed simultaneously or subsequently, therefore avoiding repeating what has already been defined.

2.6.1 Grammar porting

Grammar porting, also known as grammar adaptation, recycles grammar rules from an existing language or adapts the rules to create a new independent parallel grammar for a new language(s). The linguistic knowledge acquired while developing the existing grammar is fully exploited to develop the new language's grammar. As a result, the development time is reduced by avoiding redundancy created by repeating the rules while developing the grammar rules in parallel. Porting has been used in a variety of projects. Rayner et al. (1996, 2000) ported 80% of English syntactic rules to French in a spoken translation system. Kim et al. (2003) used lexical function grammar formalism to adapt Japanese grammar to Korean grammar, though for small grammar coverage. The Japanese grammar took two years to develop, while two months were enough to produce a good Korean grammar result. Other projects include Novello and Callaway (2003), who ported an English grammar to Italian, a process that took five months for two people to complete,

as well as Santaholma (2005), who developed a medical domain speech translator for the Finnish language by porting English grammar.

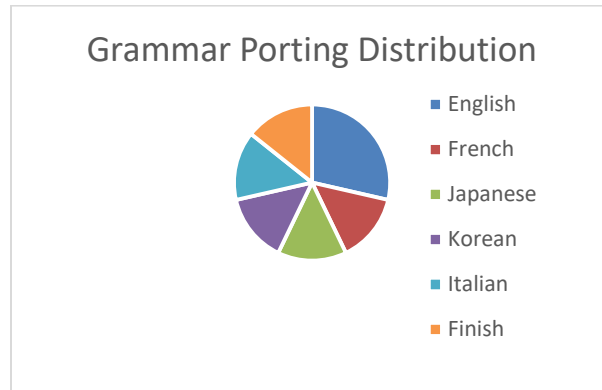


Figure 2.2 Grammar Porting

Figure 2.2 above shows that only six languages have been involved in grammar adaptation for building multilingual grammars. Four are Indo-European languages, while two are Asiatic languages. All these languages are well-resourced. This grammar engineering technique has not been applied to languages with many genders influencing speech tags' agreements, having a complex, concatenative morphology system such as Bantu languages. Therefore, there is a need to investigate whether this grammar engineering strategy can benefit under-resourced languages.

2.6.2 Grammar sharing

Grammar sharing is done either at a parallel or core grammar development approach. The parallel grammar development approach involves simultaneous grammar development for several languages while the sharing is done at the naming convention, features description and phenomena analysis (Santaholma 2007). In the core grammar development, the common grammar rules make the core engine (universal grammar) also called congruent grammar, shared by all languages. This grammar is then extended by the different grammar rules of the specific languages. Both ways of sharing are exemplified in the projects discussed below.

Kameyama (1988) was the first to prove the viability of grammar sharing at the syntax level. He presented a noun phrase expression prototype of Arabic, Japanese, English, French and German languages using categorical unification grammar in the MCC

multilingual project. Gamon et al (1997), using English as the primary grammar, reported a speedup of development time and high coverage for French, Spanish and German grammar whereby out of the 129 rules of English 10.1%, 10.7 %, 7.8% respectively of the rules were deleted and 7.8%, 8.6%, 2.3% were added for Spanish, German, and French respectively. Butt et al (2002) describe the parallel grammar sharing strategy in the parallel grammar project, which involved six languages: English, Japanese, French, Norwegian, Urdu and German. The shared grammar was developed using lexical functional grammar formalism. In terms of rules, German had 444, English 310, French 132, Japanese 50, Norwegian 46 and Urdu 25 though no quantifications of sharing capabilities were done. The grammar of Wambaya, an Australian language, was developed based on the existing languages in LinGO Grammar Matrix (Bender et al., 2008). The existing languages were English, Japanese, Modern Greek and Norwegian. It is reported that 76% of test sentences could be parsed correctly using Wambaya grammar within a short time. Bateman et al (2005), using the functionalist approach, showed that Bulgarian and Russian languages shared 76% of the features while Bulgarian, Czech, and Russian shared 92%, 84%, and 75% respectively with English grammar. Ranta (2007) worked on French, Italian, and Spanish languages' grammars using the Grammatical Framework, which resulted in 75% code sharing while the Scandinavian family (Swedish, Norwegian, and Danish) shared 90% of the syntax code. Finally, Santaholma (2008), using English, Japanese, and Finnish languages, added Greek as the new language in speech translation for a medical domain where 54% of the rules were shared in the four languages and at least 75% rule sharing for any pair of languages.

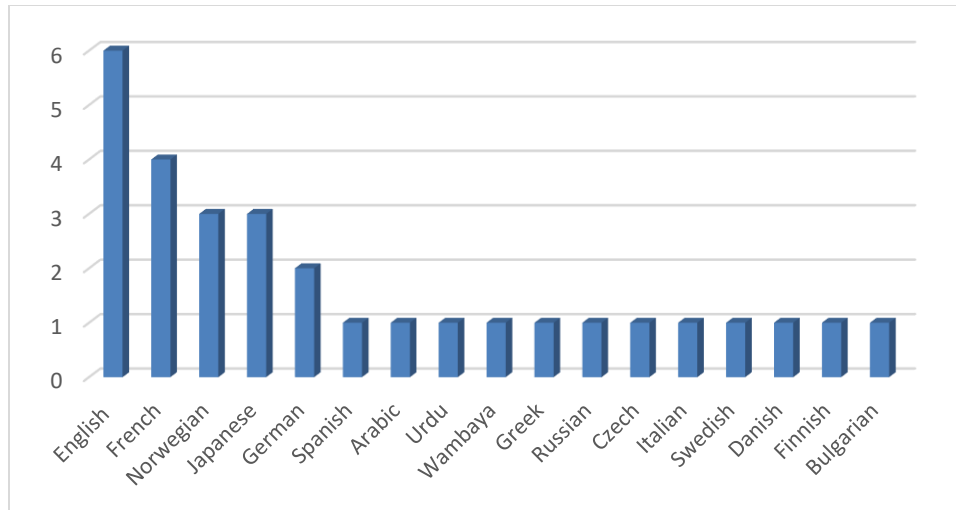


Figure 2.3 Frequency of Language reuse in Grammar sharing

Figure 2.3 above summarizes the languages that have used this approach based on the above literature on grammar sharing with a clear dominance of the well-resourced languages. Wambaya, a suffixing language, is the only under-resourced language, unlike Bantu languages that have prefixing plus suffixing and concord agreements. Therefore, there is a need to investigate how sharing grammar at the family level would accelerate its development by reducing the effort for the agglutinative languages endowed with rich, complex morphology and many genders that influence agreements.

2.7 Formal Grammar

Through grammaticalization Güldemann (1999,2003) has shown Bantoid grammar is not only limited to morphology and syntax but also lexical items especially in verb marking. When these Bantu grammars are written in a manner that computers can understand they become computational grammar and use the formal grammar representation. Kameyama (1999) proposes a three-step methodology to apply these grammar engineering strategies, namely: choosing a standard theoretical framework that is to be used to describe the language principles and parameters and secondly having a way of extracting the core grammar, and finally choosing how to generalize the grammar. This section deals with the first step, where formal grammar theory is used.

Formal language theory defines and processes formal language in a finite way (even if the language is infinite) using algorithms and mathematical means at the grammar level;

additionally, formal language is a set of strings of symbols resulting from a set of finite vocabulary Σ (Jager & Rogers, 2012). A grammar is a set of rules for forming valid sentences/strings in a language (Wang. et al., 2012). A grammar is formalized using the mathematical model in Definition 2.4 below for it to be computable.

Definition 2.4 Formal grammar

Formal grammar G is a four-tuples $G = (N, S, P, T)$

- N is a finite set of variables (Nonterminal) that can be replaced by other variables or terminals
- T is a finite set of terminals or actual words in the language
- S is a special non-terminal where all derivation starts called the start symbol
- P are production rules describing how to replace grammar symbols P

where

$$N \cap T = \emptyset$$

$$S \in N$$

and $m \rightarrow k \in P$ where m and k are in $\text{set}(N \cup T)^*$

The productions P, also known as rules, are of the form $\beta \rightarrow \mu$ where β and μ are strings belonging to $(N \cup T)^*$. A string is generated from the grammar through derivation by applying a production starting from the start symbol S and repeating through the non-terminal N until terminals are reached. In natural language words, morphemes or sounds are the terminals (Jager & Rogers, 2012). The formal grammar hierarchy is divided into four complexity classes: regular grammar (RG), context-free grammar (CFG), context-sensitive grammar (CSG), and computably enumerable grammar.

- a. Regular grammar: The productions P are of the form $\alpha \rightarrow \infty$ and $\alpha \rightarrow \beta\infty$ where α and ∞ are N while β is a T and their decidability is linear. Regular expressions defined in Definition 2.5 below are used to build regular grammar. Natural languages use regular expressions to build inflection tables.

Definition 2.5 regular expression

Ways of building a regular expression

- ϵ -use of empty morpheme
- a -use of one morpheme

- $a|b$ - a union of more than one morpheme
 - $a.b$ - a concatenation of more than one string
 - a^* recursive concatenations of zero or more of morpheme a
- b. Context-free grammar: The productions are of the form $A \rightarrow B$ where A consists of a single Non-terminal while B comprises both T and NT and the derivation of strings is decidable in cubic time and has intrinsic hierarchical nature hence sometimes referred to as phrase structure grammar.
- c. Context-sensitive grammar: In the replacement strategy, the sensitivity to the context of non-terminal must be taken into consideration; for example, in the production $\mu\alpha\beta \rightarrow \mu\infty\beta$ where μ , β , ∞ and α are arbitrary strings and the α can only be replaced with ∞ , as long it is surrounded by μ and β . The derivation of strings is decidable in polynomial time.
- d. Computably enumerable grammar: These grammars are defined by a Turing machine or any other equivalent device and the string derivation is semi-decidable. Therefore, it means that if a string w belongs to grammar G , then the TM consumes it and comes to a halt with acceptance. However, if it does not belong to G , then it runs forever or comes to a halt without acceptance.

There is a large granularity between CFG and CSG; hence the class mild context-sensitive grammar exists, representing natural language (Jager & Rogers, 2012; Ljunglöf, 2004). This class has the following characteristics:

- The length of the input is parsed in polynomial time
- Has multiple and cross and duplication agreements
- Has constant growth property and polynomial time complexity

Though mildly context-sensitive grammars usually model natural languages, they have a limitation due to the constant growth property, more so, in the case where the input strings have exponential growth. However, such can be expressed by parallel multiple context-free grammars (PMCFG) with a polynomial algorithm and tuple string linearization (Ljunglöf, 2004). The PMCFG is an extension of CFG where the right-hand side uses a tuple of strings instead of using a single string (Angelov, 2011). Many grammar formalisms with the foundation of formal theory at the Chomsky hierarchy have been developed and are discussed in the next section.

2.8 Grammar formalism

Grammar formalism¹⁷ is a mathematical model with data structures and a set of methods used to write computational grammar, especially for natural languages understandable to both humans and computers. It can either be phrase/constituents structure grammars or constrained/feature structure grammars (unification grammars). The phrase structure grammar rewrites rules to represent the constituent structure based on Chomsky's transformational grammar and it has a hierarchical organization of constituents. The main disadvantage is that natural language grammar has features (parameters) like verb agreement, person, number, case, and so on that lead to an explosion of rules due to the rules' atomic nature in the structure; consequently, implying exponential growth of parsing time. In feature structure grammar, the features are attached to the categories resulting in parameterized grammar rules (Varile et al., 1997). Unification is used to combine features and store them in a feature-value matrix. A feature could be a number with a value either singular or plural. Several grammar formalisms, both phrases and features structure-based, exist and are discussed in the following paragraphs:

Lexical Function Grammar (LFG) examines the structures of languages and their relation. It has dichotomy depictions of the syntax, Functional (f) and Constituent (c) structures. Functional structure is an abstract function representing structures such as subject, tense, case, gender and so on. The constituent structure is a form of concrete structures such as word order, phrase, etc. (Dalrymple, 2001; Austin, 2001). LFG believes more in the lexicon's rich structures and their relations, which cannot be represented by transformation or phrase structure. The phrase structures of C-structures are a regular expression. A matrix-like description is used to assemble all the F-structures, a function representing attributes – value structure, i.e., $p \rightarrow q$ where p is the attribute while q is the value.

Head-Driven Phrase structure grammar (HPSG) is a grammar formalism that is highly lexicalized, sign-based and constrain-based. It consists of lexical entries and grammar rules. The lexical entries provide phonological, part of speech tags and valence

¹⁷ http://www.alt.aasn.au/events/altss2004/course_notes/ALTSS-Asudeh-Grammar.pdf

information. Moreover, the grammar rules show the difference between immediate dominance (head and non-head) and linear precedence. Feature structures encipher the grammar information in an argument-matrix form. They are of a certain type and have been arranged using type hierarchy, thus enabling cross-cutting generation and redundancy reduction (Kahane, 2006; Levine et al., 2006). Muller (2001) notes that the subcategorization of verbs is a drawback in this formalism since they have to be encoded more than once, while Levine (2003) states that HPSG lacks non-constituent coordination.

According to Steedman (1992), in categorical grammar formalism, the lexicon carries most of the syntax information and is therefore lexicalized. The constituents (semantics and syntax) are modeled as functions, arguments and the principle of compositionality describes the relation of syntax and semantics. The type/category, whether basic or a Functor, is as well associated with the lexical item and expresses prospective amalgamation with other constituents. The use of a combinatory approach that involves operation over several functions and/or arguments such as coordination rule as an operation makes this formalism be referred to as combinatory categorical grammar.

Dependency grammar (DG) is primarily word-based. In any phrase/sentence, a word mostly depends on the neighboring word apart from the root and the relation is binary and asymmetrical (Debusmann, 2000; Debusmann et al., 2010). Dependencies are mainly based on syntactic and semantic grammatical functions plus other linguistic elements such as morphology and prosody. The dependents based on the head can be a modifier, a complementary or a specifier (Nivre, 2005). Various types of this formalism exist, such as functional generative descriptions, dependency unification grammar, meaning text theory and function dependency grammar (Nivre, 2005; Debusmann., 2000). This appeals to languages with free word order because the grammar functions (syntactic) are not affected by the permutation of words. However, they do not give explicit constituent information as compared to other formalisms. Debusmann et al. (2010) have shown that DG is a mildly context-sensitive grammar.

Montague grammar (MG) is based on formal logic and uses the compositionality principle to relate syntax and semantics. The syntax consists of rules (lexical or recursive) made of categories (sentences and entity) and functions showing how to form phrases from categories. On the other hand, semantics is derived from the formal intentional logical

translation of sentences (Kao, 2004; Kracht et al., 2012). Its implementation encompasses the tecto-grammatical and pheno-grammatical rules and, in some instances, is referred to as universal grammar (UG).

Tree Adjoining Grammars (TAG) have a set of elementary trees (finite initials and auxiliary) operated by substitution and adjunction. The substitutions involve initial tree swapping with non-terminal leaf, while adjunctions involve auxiliary trees swapped for internal nodes. Parse trees or derivation trees are derived from the history of combinations (Yoshinaga et al., 2003). TAG belongs to a mild context-sensitive grammar level in the Chomsky hierarchy of grammars.

Definite clause grammar (DCG) is a formalism used to write natural languages, especially free word order (Tanaka, 1991) and formal languages in logic programming format. DCG introduces context-dependency. Constituents have extra conditions and build structures (trees) that are not bound by grammar recursion of the rules; hence, formalism is viewed as an extension of CFG formalisms (Pereira et al., 1980). Its syntax includes terms and clauses. Terms are the data objects which include constant, variable, or compound term (Functor), whereas clause is the logic part of the grammar made of a head and a body. Prolog programming language is an example of a language using the DCG formalisms.

Generalized Phrase structure grammar (GPSG) uses CFG to capture grammar rules. The grammar features are either atomic-valued or category-valued features or use meta-rules to make generations (Jacobson, 1987).

Grammatical Framework Formalism (GF) is grounded on categorical formalism (Ranta et al., 2009). It implements generalized Montague grammar structures (Ranta, 2011) by separating abstract syntax from concrete syntax. Generally, it has one common abstract syntax and several concrete syntaxes for different languages. It also acts as a toolkit by allowing specific language grammar development at the resource's library while domain-specific grammar (application grammar) is developed on top of resource grammars. The GF grammar's expressive capability is equivalent to parallel multiple context-free grammars.

Montague grammar and abstract categorical grammar use the Curry (1961) type grammar structure of using tecto-grammatical (abstract syntax) and pheno-grammatical

(concrete structure) structure, but they fail to address the issue of multilingualism (Ranta, 2011). The abstract representation of DCG, LFG, HPSG, TAG and CCG formalisms are not comparable to the powerful Curry-based formalism (Ranta, 2011). GF possesses these two characteristics and allows further development of domain-specific grammar on top of the resource grammar. The domain grammar writer who does not have linguistic knowledge of grammar just uses the resource library grammar to reduce his/her work, thus a quick way of producing language resources (applications and tools). In addition to these, resource grammars can be used for natural language processing tasks such as machine translation, multilingual analysis, multilingual generation, software localization, natural language interfaces, spoken dialogue systems, etc. Pretorius et al. (2017) argue that GF is becoming the de facto formalism for developing controlled multilingual grammars. The GF provides mechanisms for implementing grammar porting and sharing GE strategies among related grammar (family languages) through its module known as a Functor, which is the core business of this research. The Functor uses parameters in implementing core grammar, thus referred to as parameterized modules, which augurs well with Bantu languages because of the complex morphology that utilizes many parameters, especially the genders and concord and provides a way of separating the shared and unique segments of the Bantu grammar thus providing a way to reduce development effort. The above advantages have led to GF's choice as the formalism for implementing this thesis' work.

2.9 Grammatical Framework

Grammatical Framework (GF) is a toolkit used for the rapid development of multilingual grammar resources and applications. It is based on a functional programming paradigm (types system, modules, etc.), a logic framework of abstract and concrete syntaxes and a grammar formalism grounded on categorical formalism (Paikens et al., 2012; Ranta et al., 2009; Ranta, 2011). GF has one abstract syntax that defines a set of categories (Cat) of trees, a set of functions (Fun) to implement those trees plus their type and start category (Angelov, 2011) as per Definition 2.6. Below. The framework has many concrete syntaxes, one for each language's grammar. These syntaxes define linearization of both the categories (lincat) and the function (lin) stated in abstract syntax as exemplified

using the category Noun (N) with a string “house” below (Ranta et al., 2009). Definition 2.7 below summarizes the mathematical GF definition.

Abstract syntax	Concrete syntax
Cat: N	lincat N = Str
Fun House: N	lin House = "house"

Definition 2.6 Abstract syntax

$$\text{Abstract syntax} = [N^A, F^A, S]$$

Where

- N^A is a finite set of abstract categories
- F^A is a finite set of abstract functions
- $S \in N^A$ is the start category

Definition 2.7 GF definition

$$GF = A (C_1, \dots, C_n)$$

Where

- A is the abstract syntax
- C is the concrete syntax
- 1, ..., n the number of the parallel concrete syntaxes

Each concrete syntax is of complexity parallel multiple context-free grammars (PMCFG) due to the use of tables and record data structures (Ranta et al., 2020). The definition of PMCFG is given by a 5-tuple equation as shown in definition 2.8 below.

Definition 2.8 parallel concrete syntaxes

$$\text{PMCGF} = (N^C, F^C, T, P, L)$$

Where

- N^C is a set of finite concrete categories
- F^C is a set of finite concrete functions
- T is the finite terminals symbols
- P is a finite set of production rules.

- $L \in N^C \times F^C$ is a set that defines the default linearization functions for those concrete categories that have default linearization's

All the parallel natural language computational grammars (PMCGF) reside in the GF resource grammar library (RGL), where the syntactic and morphological properties of a specific language are captured and form the multilingual grammars ecosystem (Ranta, 2006). The online repository contains¹⁸ over 48 parallel grammars. The RGL consists of several modules subdivided into three major groups: lexical, morphology, and syntax modules, as shown in Figure 2.4 below. The lexical modules are *lexicon*, *structural* and *numeral*. The *lexicon* module provides lexemes for open categories, whereas the *structural* module provided for closed categories. The *numeral* module provides lexemes for cardinal and ordinal numerals. The morphology modules use smart and low-level paradigms to implement declension. Paradigm is a function that takes lexeme word form(s) and generates the lexeme's complete word forms (inflection table). Detrez et al (2012) define, a smart paradigm as a Meta paradigm that “inspects the given base form of a lexeme and tries to infer which lower paradigm applies”. *Morpho*, *resource* and *paradigm* are the morphology modules. The syntax modules provide an ecosystem for implementing phrases, clauses, sentences, questions, and so on. In addition, the GF resource grammar library uses other modules mainly: *paramax*, *common*, and *prelude* to import functions - parameters that are common for all languages present in GF. GF provides 500 lexical items consisting of 350 content words, 100 structural words and 50 numerals for grammar testing. However, large lexical items for wide coverage are developed in the *dict* module, which is an extension of the *lexicon* module. The core syntax defines 200 functions and 60 categories, which form declarative, question and imperative sentences (Ranta. 2009).

¹⁸ <https://github.com/GrammaticalFramework/gf-rgl/tree/master/src> accessed on 6th Oct 2020

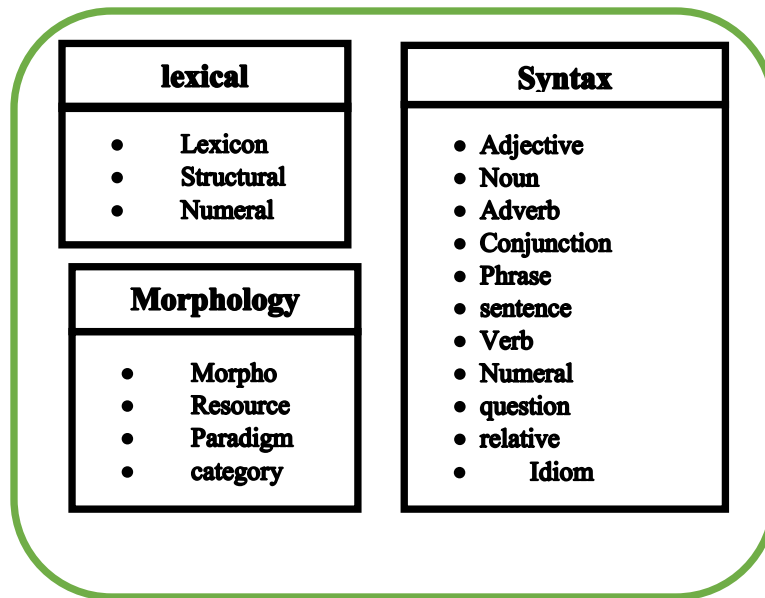


Figure 2.4 GF RGL Modules

Parsing provides a means of transforming language-specific strings to abstract trees, while linearization is a composition of homomorphic¹⁹ mapping from common abstract tree structure to specific language concrete syntax (Ranta, 2011). Machine translation would then be achieved by first parsing the source language's string to abstract trees then linearizing the tree to a string in the target language. Since the processes are reversible, GF acts as an Interlingua rule-based translation system, as shown in Figure 2.5 below for Polish, English and Spanish, enabling bi-directional translation.

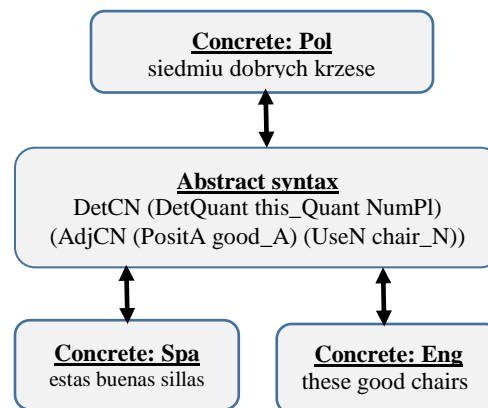


Figure 2.5 GF syntaxes

¹⁹ One to one mapping, no reanalysis of the trees

Features in GF formalism are provided via parameters that are objects of some type, defined using the keyword *param* and mostly used in table types. For example, the Noun in Bantu languages has a parameter number. It has the values: singular and plural; therefore, the definition would look like

```
param
  Number = Singular | Plural
```

A category may have more than one different parameter. In such a case, a data structure record is used to gather them. For example, the category Noun in Bantu languages has an additional parameter, gender, apart from the number; therefore, it is defined as;

```
N = {s: Number => Str; g: gender};
```

The above is a table from number to string and inherent features of gender (functions over parameters) (Ranta, 2007). GF distinguishes the function *fun* used in abstract syntax and the function operation *oper* used to implement inflection paradigms. Operation is used to implement the regular pattern in grammars to avoid redundancy of repetition. The keyword *oper* is usually of the form

```
oper function_name: function_type = function_body
```

One name can be used for different paradigms in the same category through operation overload.

2.9.1 Grammatical Framework Functor.

Functor f is a function which maps every element of a domain to an element of function f in the co-domain. It also maps every morphism of the domain to a morphism of function f in the co-domain whose type is retained, given sets A and B, which act as a domain and a co-domain respectively. In GF, a functor is called parameterized modules and it is a function that opens interface[s] which contains types of operation, not definitions (Ranta, 2007). The domain has the types of functions of a family of languages such as the Bantu family and standard definition where possible across the languages, while the co-domain has the actual definition of the functions defined as shown in Figure 2.6 below. According to Ranta (2011), a Functor has two significant advantages: it provides grammar sharing capabilities, thus reducing development effort, making it easy to add new languages in the family since only parts of the new grammar that differ from the already

developed grammar are constructed. Secondly, the shared grammar rules maintenance is cheaper since the rule-base is significantly reduced for the languages. The Ekegusii and Kikamba grammars will be used to develop the Bantu parameterized grammar in the GF Functor using the grammar engineering strategies and thereafter bootstrap Swahili.

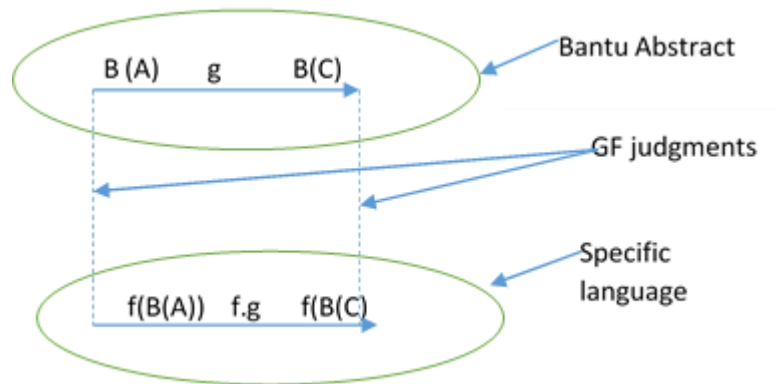


Figure 2.6 Functor mapping

2.10 Grammar Sharing and Porting Evaluation

The Bantu parameterized grammar plus the bootstrapped Swahili will need testing and evaluation. The testing aims to improve the quality of the congruent and bootstrapped grammars (reduce over generation and ensure coverage) during the development, whereas the evaluation goals are to demonstrate the reduction of development effort, correctness and accuracy of the resulting grammars. To investigate the quality, correctness and accuracy of the grammar during and after development requires test data. Bröker (2000) and Butt (2003) give three ways to get test data, namely:

- a. A grammar writer or expert writes the test suite data or uses already existing test suites.
- b. Use of a natural existing corpus or treebank.
- c. Use of the comments created for each grammar rule that shows what the rule parses in the grammar.

GF uses method *c* above to test the grammar during the development since each function or production rule has a comment (s) on the abstract syntax. The comment(s) is/are an example(s) of what the function can parse in English. Example 2.12 below shows an

extracted comment “big house” from the abstract syntax for the function AdjCN that makes an adjective phrase from a common noun.

Example 2.12 Comments example

```
AdjCN : AP -> CN -> CN ; -- big house
```

The testing will follow the GF regression testing process (Ranta, 2011; Camilleri, 2013) and will act as the development test suite, where the concrete syntax trees for grammar features/functions are implemented in a specific language and the comments are parsed in English grammar and linearized using the constructed function in the congruent and bootstrapped grammars to the specific Bantu languages (machine output). The machine output is compared with the informant translation. If the two differ, then the GF function is refined until the two translations are the same and the regression test re-run is repeated each time refinement is done to ensure no new errors are introduced. The process is summarized in Figure 2.7 below.

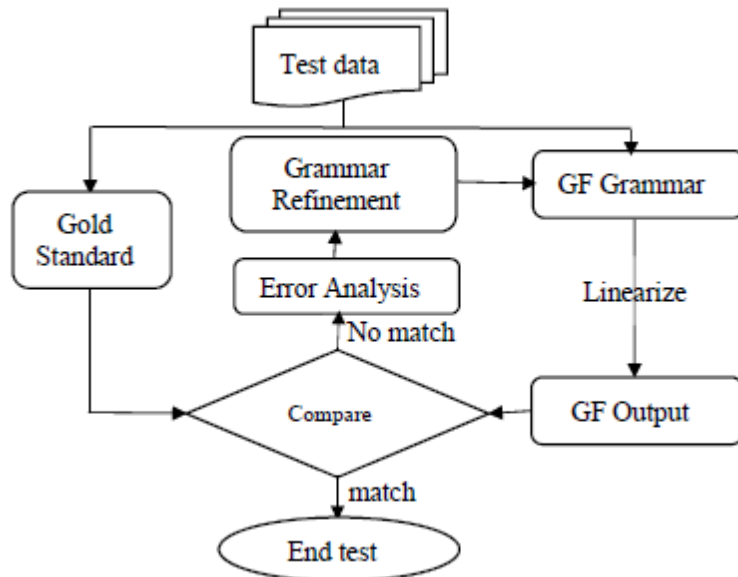


Figure 2.7 Testing process

2.10.1 Evaluation metrics

There are no standard metrics for evaluating grammar shareability and portability due to the different grammar formalisms and development tools used for each grammar (Santaholma, 2008). Table 2.6 below summarizes the metrics so far used. Shared rules measure production rules common among grammars in terms of percentage or number or line of codes. Rules modification measures the number of rules that have been modified or deleted in order to adapt a new grammar (bootstraps). Development time measures the time used to develop the grammar in terms of weeks and months per person. Finally, performance measure depends on the task being performed and the metrics for the task.

Table 2.6 Multilingual grammar Evaluation metrics

Metrics	Used by
Shared rules	(Bateman, 1997), (Santaholma, 2007) (Santaholma, 2008), (Bender et al., 2008) (Ranta et al., 2009)
Modification of rules	(Gamon et al., 1997), (Bender et al., 2008)
Development time	(Bateman, 2005), (Bender et al., 2008) (Ranta et al., 2009), (Novello and Callaway, 2003)
Performance	(Santaholma, 2010)

All researchers on shared rules used the percentage of the shared rules except for Ranta (2009), who uses the percentage of the shared line of codes. Unless aware of how the line of codes is arranged, it is hard to compare with another grammar.

In terms of development time, Ranta (2011) gives a rough estimate of five and ten months of person work to define Scandinavian and the Romance family grammars respectively. It was estimated Wambaya grammar took 5.5 person-weeks of development time to be adapted (Bender et al., 2008), which was 210 hours split into 25 hours of lexical entries, 7 hours of test suite development, 15 hours of treebanking and the rest on grammar development. Kim et al. (2003) took two years to develop the Japanese grammar but two months to port Korean (lexicon and some rules) with excellent results and expected the whole grammar to be completed in eight months. Lastly, Novello and Callaway (2003) estimated to have taken five person-months to adapt English grammar to Italian. The time was split between two people, a native speaker of English with knowledge of the

development platform and a native speaker of Italian who had no knowledge of the platform. Novello and Callaway (2003) state that different developers have different speeds and one developer might have different speeds on different days; hence, this metric cannot accurately measure time. Besides, one developer will have different competencies in the course of the grammar development cycle, meaning that the value given for time will be inappropriate and just an approximation. Therefore, the research adopts shared and modified rules as the evaluation metrics.

In this thesis, the shared rules metric was adopted to measure the shareability of the shared parameterized grammar based on the percentage of rules shared. In addition, the modification of rules metrics was also adopted, that is, the number of rules whose structures were modified to fit a new language was used to measure portability. Development time was ignored since, as argued above, it involves approximation and thus cannot give an accurate figure. However, the size of the rule-base for the congruent grammar compared with the expected monolingual grammar rule-base demonstrated reduced effort implying less time for development.

The performance was used by one researcher while doing a speech recognition task and applied word error rate (WER), sentence error rate (SER) and semantic error rate (SemEr) as the metrics for speech processing. These metrics are used to measure the accuracy of the grammar. In this thesis, grammar performance (correctness and accuracy) was measured by undertaking a machine translation task. Consequently, the Bilingual Evaluation Understudy (BLEU), Word Error Rate (WER) and Position Independent Error Rate (PER) metrics, which are commonly used metrics for evaluating machine translation (Vilar., 2006), were used. BLEU (ranges from zero to one or is expressed as a percentage) demonstrated a good correlation between machine translation to human judgment (Koehn, 2004) hence, used to measure the accuracy. Rule-based translation systems have adopted the BLEU metric to evaluate the resulting system: for example, the rule based system of Dutch to Africaans (Van & Pilon., 2009), written Spanish to Spanish Sign Language rule-base system (Porta et al., 2014), An automatic question generation rule-based system (Keklik et al., 2019), Tunisian dialect to the standard Arabic language (Sghaier & Zrigui., 2020) and English to Catalan translation system (More., 2020). Therefore, since this metric has been successfully used to evaluate rule-based systems, the research also adopt it to

evaluate the Bantu parameterized grammar. Though the performance may be affected by word order and word choice variation. PER and WER based on Levenshtein distance (Levenshtein, 1966) are excellent metrics to investigate Bantu languages' errors since these languages have a lot of nasal insertion, deletion and substitution, especially the joining of morphemes at the word level. Thus, they measured the correctness of the grammar..

2.10.2 Error analysis

There are two taxonomies used for computational grammar error analysis. First, the hierarchical taxonomy (Vilar et al., 2006; Bojar, 2011) classifies errors into five hierarchies: missing word errors, word order errors, incorrect words, unknown words errors and punctuation errors. Bojar (2011) omitted unknown errors, thus four tiers of classification. The second taxonomy is the linguistic taxonomy (Costa et al., 2015) which classifies errors as orthography, lexis, grammar, semantics and discourse. Costa et al (2015) did a comparative analysis of the two taxonomies and developed an all-inclusive taxonomy summarized in Figure 2.8 below.

The orthography errors consist of punctuation errors, capitalization, and misspelled words that can be rectified by adding, deleting, or substituting a word's letter. Lexis errors can be omitted words, added words in the target translation, and untranslated words. Grammar level errors involve misselection of words at the morphology level, such as verbal level errors; for example, tense and person, errors of agreement (gender, number, object and subject marker, negation, and so on) and misordering of words in a phrase. Semantic errors are realized when there is a sense of confusion where the word translated is out of context, the wrong choice of a word since the word chosen does not contribute to the phrase's semantics. Idiom errors as a result of wrongly constructed idiomatic expressions form semantic errors. Finally, there are discourse-level errors such as style errors (repetition of words), variety errors (where morphological changes for other languages are applied to the target language) and when words that need not be translated are translated.

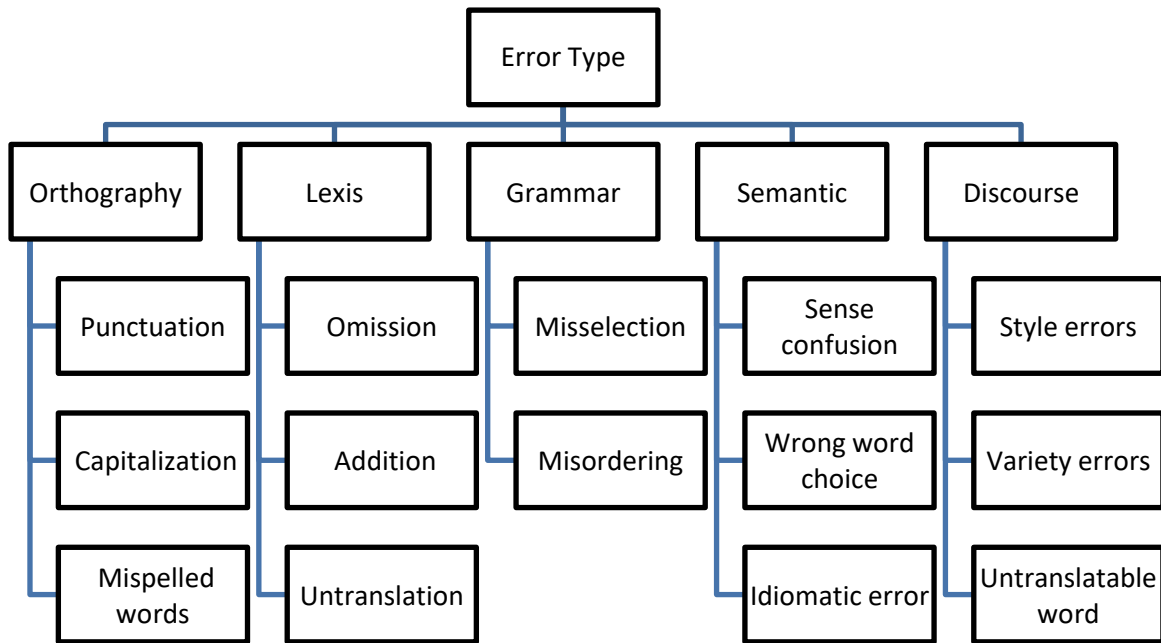


Figure 2.8 Errors classification in grammar development

This thesis adopted the comparative taxonomy to annotate and analyze the errors because it is all-inclusive manually. However, the discourse errors were excluded since they are beyond the research scope, though semantic errors were included since GF's categories and functions use the compositionality principle to relate syntax and semantics

2.11 Conceptual Framework

This research leverages on cross-linguistic similarities to develop the shared Bantu parametrized grammar, which acts as bootstraps seed grammar. The cross-linguistic similarities are based on the principles and similarities of UG. Accordingly, the research hypothesis that the higher the similarities measure (Gali et al., 2019) of the cross-linguistic similarities, the higher the percentage of congruent grammar and the similarity measure can range from 0% (variation) to 100% similarity. The grammar cross-linguistic similarities will use the concepts of lexical, morphology, and syntactic as stated in section 2.7 and their definition will follow the formal grammar illustrated in Definition 2.8. The lexical similarity was used because some lexical items bear syntax and morphology for example when prepositions and possessive demonstratives are infused in the noun as explained in section 2.3.2.1. Therefore, to model, the Bantu parameterized grammar, the

concrete syntax, regular expression for morphology and lexemes for the languages will need to be defined in GF. Consequently, the categories, functions, and linearization have to be defined to enable lexical definitions. The lexical similarity (Lewis, 2009) will measure the cross-linguistic similarity at this point. Morphology definition (inflections or derivational) involves developing the GF paradigms (regular expressions) based on Definition 2.5. In addition, the morphology is influenced by the definition of categories linearization (parameters for a part of speech) and the morphological similarity (Pretorius and Bosch, 2008) will be used to measure the cross-linguistic similarities at this point while syntactic similarities (Marten et al., 2007) will measure the similarity measure among the production rules (functions) defined.

According to Dąbrowska (2015), grammar similarities in UG result from principles and parameters that affect the cross-linguistic measure. The lexical similarity will be affected by the specific part of speech tags parameters; for example, the Bantu noun is influenced by gender and number parameters while the verb is by tense, agreement (person, number and gender) and valence, as demonstrated in the literature review. All categories need to agree with noun gender and number in terms of their inflection. Therefore these parameters affect the morphological similarity measure too. Finally, the syntactic similarity measure is moderated by topology and agreements. Higher parameters similarity across the grammar implies higher lexical, morphological and syntactic similarities, consequently, a higher percentage of the shared parametrized grammar; thus, the parameters act as moderators for the congruent grammar. Figure 2.9 below summarizes the conceptual framework. The three similarities are based on similar principles and parameter values that cut across these languages with the underlying theory being UG.

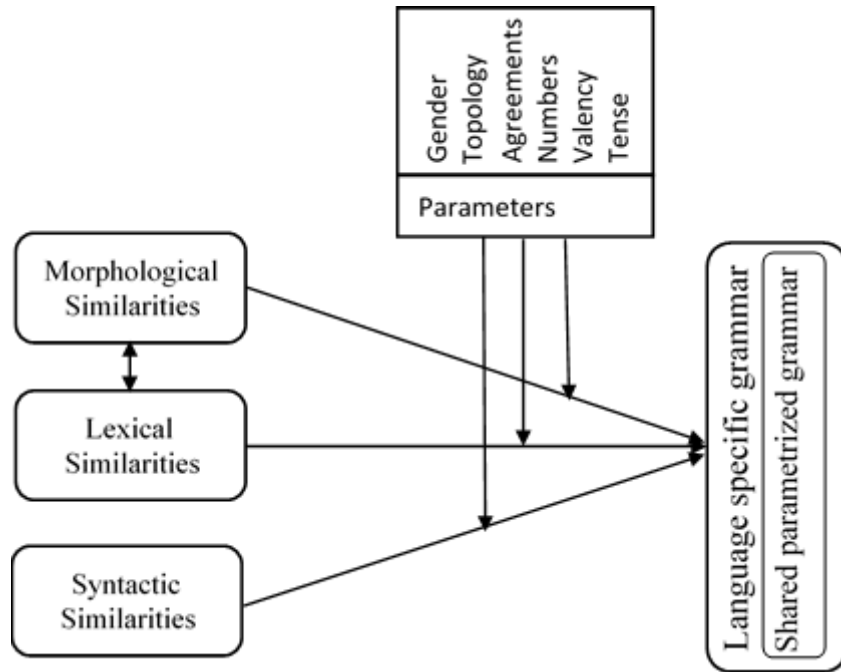


Figure 2.9 Conceptual Framework

2.12 Summary

This chapter has established a high degree of cross-linguistic similarities in the three Bantu languages across morphology, syntax, and orthography through a rigorous comparative study of respective descriptive grammars. This review has also established that these languages still lack NLP resources and tools, as confirmed by the survey on digital language resources and tools. The lack of corpora for these under-resourced languages makes data-driven approaches unsuitable for them. Though the traditional rule-based methods are expensive in terms of rule-base and knowledge required to develop them, the grammar engineering strategies (sharing and porting) reviewed have shown the capability of reducing development effort, therefore, accelerating its development. A review of the available grammar formalisms was done. GF formalism was chosen because of the separation of abstract and concrete syntaxes, having parameterized modules (Functor) that can be used to implement congruent grammar. The GF grammar regression testing method was explained and different evaluation methods were reviewed. Finally, a proposed conceptual framework was presented that guides the execution of the research.

Chapter 3 METHODOLOGY

3.1 Introduction:

This chapter introduces the research process and design. It first explains the sampling procedure for Bantu languages, descriptive grammars, experts, and linguists. The next step is developing generalized comparative descriptive grammar based on the similarities in the Ekegusii and Kikamba grammar grounded on comparative analysis. An experiment is set up in GF based on a morphology-driven approach for developing the Bantu parameterized grammar. It involves developing grammar functions using an evolutionary prototype model while testing during development and evaluating thereafter. Finally, Swahili grammar is bootstrapped to the Bantu parameterized grammar with aim of evaluating the approach of bootstrapping grammar development in a multilingual environment. At the end, the researcher demonstrates how reliability and validity have been achieved.

3.2 Sampling

There is linguistic diversity among languages at different geographical locations (Beal, 2010; Omar & Alotaibi, 2017; Trudgill & Hannah, 2008), though one of the critical research goals was to utilize cross-linguistic similarities to enable the development of a generic Bantu parameterized grammar. The languages chosen are in different geographical areas to ensure the resulting grammar can be generalized to many other Bantu languages. Consequently, language sampling used geolinguistics and Guthrie's zone plus groups among the three geographical areas identified in section 2.3.1 to include linguistic diversity. However, purposive sampling was applied to a specific group in a zone depending on the availability of descriptive grammar and experts plus the researcher's ability to understand the languages. Furthermore, the researchers' interests in the languages such as extending the work further so as to create NLP applications. The languages chosen using geolinguistics were Ekegusii E42, Kikamba E55 and Swahili G42, as shown in Table 3.1. A minimum of two languages are needed to model the congruent parameterized grammar and the hallmark of this research was based on under-resourced languages. Based on section 2.4, it was shown Ekegusii and Kikamba languages are less- resourced compared with Swahili. Consequently, these two under-resourced languages were chosen for the

development of shared grammar. This implies Swahili becomes by default the language to evaluate the shared grammar through the bootstrap process. In addition, the Swahili language having good grammar books will provide a case to evaluate the approach's effectiveness and efficiency.

Table 3.1 Languages Sampling

	Eastern Bantu	Western Bantu	Coastal Bantu
1	Gikuyu (E51)	Luhya (E30) Dialects: Bukusu, Wanga, Samia	Pokomo (E71)
2	Embu (E52)	Logoori (E41) Dialects: Tiriki, Idakho, Isukha	Nika (E72) Dialects: Giriama, Duruma, Kauma, Rabai and Chonyi
3	Meru (E53)	Ekegusii (E42)	Digo (E73)
4	Tharaka (E54)	Kuria (E43)	Taita (E74)
5	Kamba (E55)		Swahili (G72) Dialects: Amu, Mvita, Mlima and Unguja
6	Daiso (E56)		

The snowball sampling technique ²⁰(Ngau et al., 2004), a non-probability sampling technique, was used to gather the data and information and identify experts of the Bantu languages being investigated. The researcher identified the initial linguist/expert and then asked them to identify another or other potential experts who also meet the research criteria and so on until the needed descriptive grammar was available or all the experts were exhausted. The non-probability procedure does not afford any basis for estimating the probability of an item being included in the population (Kothari, 2011). This fits well for Bantu languages classified as under-resourced languages with few resources available in terms of descriptive grammar and experts.

3.3 Comparative descriptive grammar development

The research used a Hybrid research design by utilizing three research designs: descriptive case study, comparative analysis and experimental design, as summarized in Figure 3.1 below. Combining several research designs results in wealthier data/information, superior insights and detailed learning, argue Haf²¹. Two stages were

²⁰ <http://explorable.com/snowball-sampling>

²¹ http://ikmarketing.de/wp-content/uploads/2017/05/RR_Hybrid-methodology_CH.pdf

involved in the development of comparative descriptive grammar. First, a descriptive case study research design was used to understand the principles and parameters of specific language descriptive grammar and use them to model regular expression plus relationship patterns (grammar rules) for morphology and syntax, respectively, for each particular language. The second stage used a comparative analysis research design to develop congruent and portable descriptive grammar between the Ekegusii and Kikamba languages. These two stages achieved the first objective by demonstrating the degree of similarities between the languages.

The primary sources of data for the descriptive grammar were Bantu language grammar books and their dictionaries (Ashton, 1947; Kyallo, 2016; Mwau, 2006; Njogu et al., 2006); the researcher also examined postgraduate masters and Ph.D. theses for respective Bantu languages (Basweti, 2005; Kaviti, 2004; Mbuvi, 2005; Ongarora, 2008; Osinde, 1988) and several journals and conferences papers. Other sources included Bantu language linguists using the elicitation method, especially where we could not trace written materials or references to develop the missing segments' descriptive grammar. The research used two ways of doing elicitation: First, language analysis through the linguist's judgment and translation from English to the specific Bantu language (Chelliah, 2001). The linguist or informants' elicitation outputs were subjected to another linguist to ensure correctness and consistency.

In data collection, an intensive literature review and in-depth analysis were used to gather information about the categories and their syntax relation from the written literature. In addition, structured and unstructured interviews of the Bantu linguists, experts and key informants in the different languages were utilized. Face to face and telephone approaches plus emails were used for the interview.

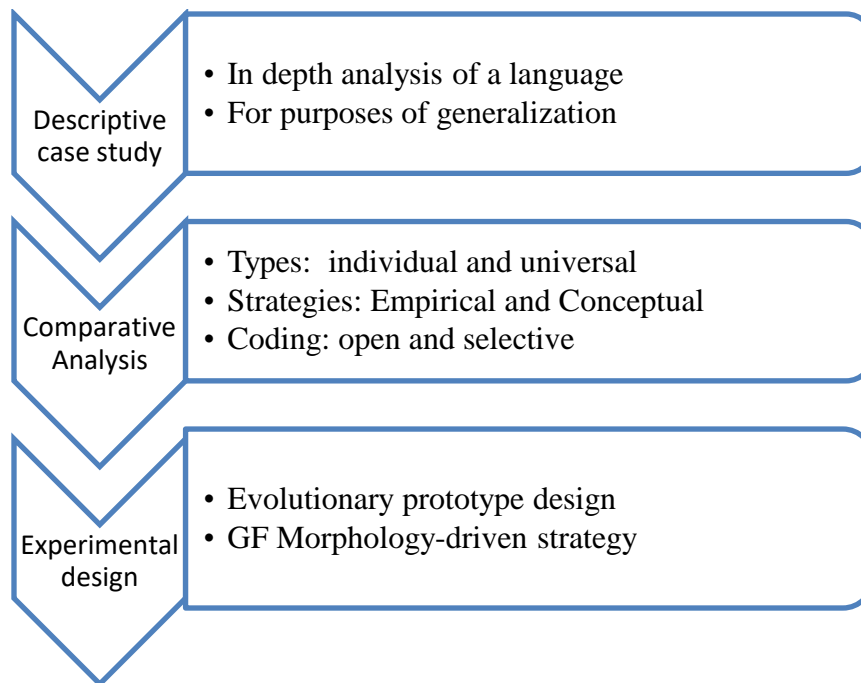


Figure 3.1 Research Design

3.3.1 Descriptive Case Study

A descriptive case study investigated categories (POS) and phrases for each specific language separately (Morpho-syntactic structures). For categories, the objective was to find out and summarize the grammar features, principles and parameters from which a generalized regular expression for morphology is generated. Strauss and Corbin's (1998) open coding was used to establish regular patterns and selective coding for the syntactic patterns. For example, Kikamba adjective category descriptive grammar has parameters concord, number and two degrees (positive and comparative) and its regular expression structure is shown below and illustrated in Figure 3.2:

- Positive degree

Concord (based on the number) ++ adjective root

- Comparative degree

Concord (based on the number) ++ adjective root ++ infix string ++ adjective root final

vowel

The ++ means the morphemes are joined conjunctively. Further, the sketch in Figure 3.3 depicts some morphemes in the Ekegusii verb, while Table 3.2 shows the morphemes' positions and types in various tenses of Ekegusii. Figure 3.4 shows morphemes in the Kikamba verb and the regular pattern in negative polarity. For the phrases, the interest was determining agreements (concord), grammar parameters and order of categories. For example, Figure 3.2 below shows that the Kikamba adjective needs a concord for agreement with a noun. Finally, Figure 3.5 shows NP's components in Ekegusii and their order. The descriptive case study steps are as follows and summarized in Figure 3.6:

1. Check and confirm that the descriptive grammar for each language is available (done at the literature review stage)
2. If any component is missing, use Bantu linguists or informants to generate it through elicitation
3. Then generate the generalized principles, parameters, regular expression and grammar rules in each language

The stage generated a generalized descriptive grammar for each category and phrase in each specific language (Yin, 2003). The generalized grammar consisted of generalized principles and parameters of each category, summarized in Table 3.3 below and generalized regular expressions and grammar rules summarized in Tables 3.4 and 3.5 for Kikamba and Ekegusii.

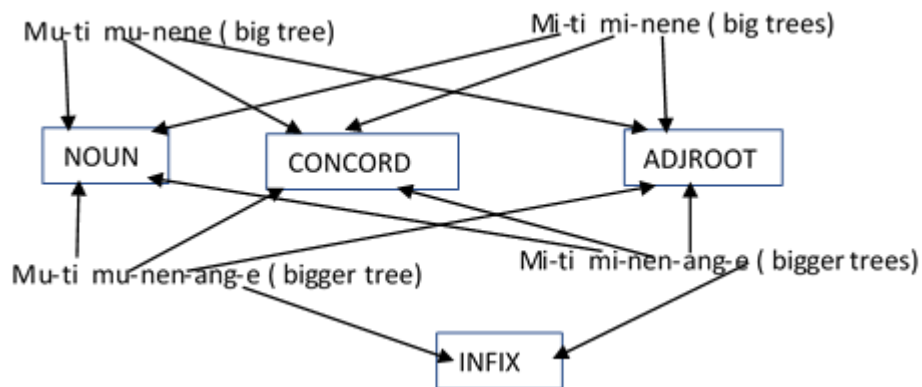


Figure 3.2 Adjective structure

N-chi - a - teera They sang
 Ti-chi - teer-et-i They did not sing
 m-ba - raager - a They ate
 Ti-ba - raager - et-i They did not eat
 N-gi - a - teng a It danced
 Ki-gi - teng - et-i
 Ti-kenigoteng - It is not singing
 Ngo teera chire
 They are not singing

Figure 3.3 Ekegusii verb Morphemes

Meyahne Present tense
 Negative tense by 'ku' for
 (AG + Neg) + Neg Tense + Verb
 Ndi ku Vuuka
 will/can not fail
 Kuuwa Kaa Kooji (K) U Pwanka nasa
 This little boy is/will not play (ing) well
 Pwani ni meku sukke
 Ague + T ma gi ku sukke.
 AG + Neg + TNS + Verb not + suffix

Figure 3.4 Negative polarities for Kikamba verb

Table 3.2 Ekegusii Verb morphemes per tense

Example and gloss	Focus	Subject	Tense	Root	Tense	vowel	Tense
Eke-busi n -ki -a -raager- a. the cat ate	N	ki	A	raager		a	past tense
Chi-mbiri n -chi -a -raar -a the goats slept	N	chi	A	Raar		a	
Chi-seese n-chi -raager-et-e the dog ate (recently)	N	chi	A	raarer	et	-e	past tense
Atandi a -ko -raager-a andandi eats		a	ko	raager		a	present tense
Aba-ana bi -go -teer -a the children sing		ba	go	Teer		a	
N -n -gend-e reero I will go today	N	n		Gend		e	future tense
Chi-sese chi -a -raager-ir-e. the dogs have eaten		chi	A	raager	Ir	e	perfect tense
Chi-nkororo chi -go -teer-a the warriors are singing		chi	go	Teer		a	progressive
to- sab- a we ask		to		Sab		a	Habitual
to-takan-a we chew		to		takan		a	Habitual

NOUN PHRASE IN EKEGUSII

INTIVE we
Personal Pronoun

enyumba yatto our house
Noun Possessive Pronoun (DEM)

enyumba erene big house
Noun Adjective

chikanisi chiano chironi itano chingya
Noun Possessive Pronoun Number Adjective

Au yar -fise gawel churules

chironi chikanisi chiano itano chingya
Number Noun Possessive Number Adjective

Summary

Personal Pronoun (Pron) or

Quantifier Noun Possessive Pronoun Demonstrative/Quantifier
Number Adjective

* Noun must appear

Figure 3.5 Ekegusii Noun Phrase

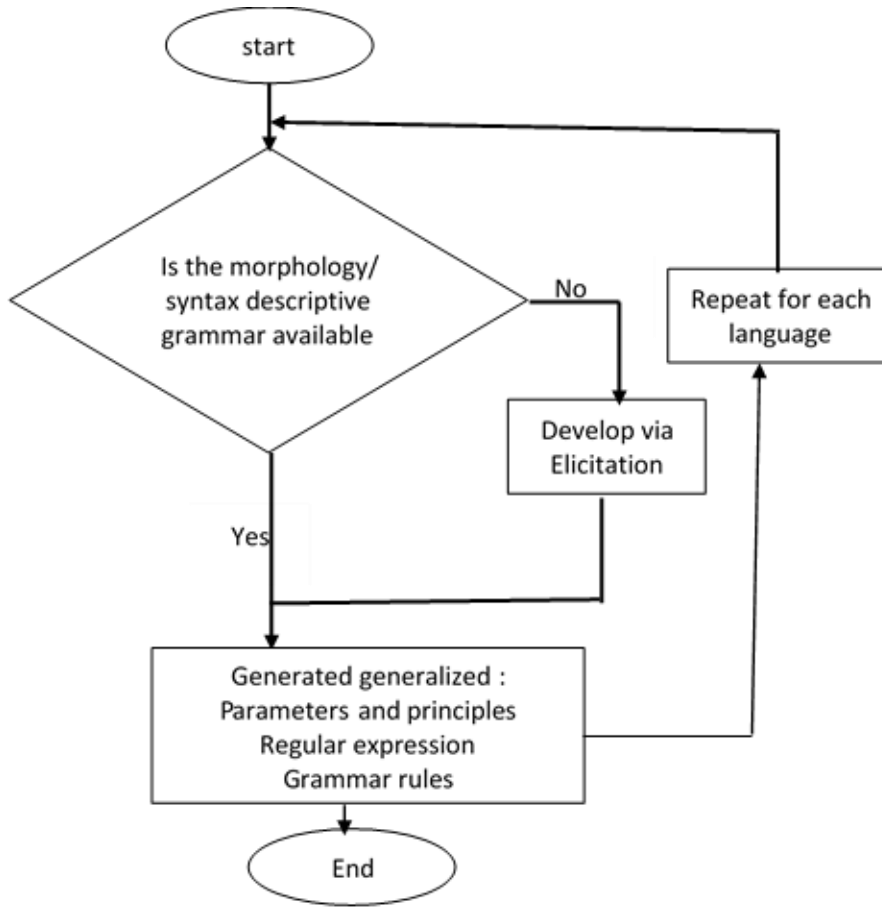


Figure 3.6 Descriptive case study methodology

Table 3.3 Generalized Parameters and Principles

Category/Phrase		Parameters and Principles	
		Kikamba grammar	Ekegusii grammar
Noun/common noun		Gender and Number	Gender and Number
Adjective/Adjective phrase		Concord, number and degree (positive and comparative)	Concord, number and degree (positive)
Verb/verb phrase	Normal	Agreement (person, number and concord), valency, mood, tense, aspect and derivation	Agreement (person, number and concord), valency, mood, tense, aspect and derivation
	Imperative	Polarity, number, command and request	Polarity, number, command and request
Pronoun	Personal	Agreement (person, number and gender)	Agreement (person, number and gender)
	Possessive	Concord and number	Concord and number
Demonstratives/Quantifier		Concord and number	Concord and number
Number(numeral, digits)		Concord, cardinal and ordinal	Concord, cardinal and ordinal
Preposition		Concord, number, infuse	Concord, number, infuse

Adverbs/Interjection	–	–
Determiner	Concord, number and position	Concord, number and position
Noun Phrase	Case and agreement	Case and agreement
Sentence and relative clause	Topology, tense and polarity	Topology, Tense and polarity
Question clause	Tense, question form(direct or indirect) and polarity	Tense, question form(direct or indirect) and polarity
Topology	SVO, VO and V	SVO, VO and V

Table 3.4 Kikamba generalized RE and grammar rules

Category		Generalized Regular expressions or grammar rules
Noun		Gender prefix(number) ++ root
Adjective	Positive	Concord prefix(number) ++ root
	Comparative	concord prefix (number) ++ root (minus final vowel) ++ infix string ++ final vowel
	Colour	concord prefix (number) + string + colour lexicon
Verb	Positive polarity	Focus(optional) ++concord(subject) ++ tense ++ concord(object) ++ derivative morpheme ++ final vowel
	Negative polarity	concord(negation) ++ tense ++ concord(object) ++ derivative morpheme ++ final vowel
Pronoun	Personal	String(based on agreement)
	Possessive	Concord ++ root
Demonstratives/quantifier		Concord prefix(number) ++ root
Preposition		Concord(number) string OR independent string OR noun ++ string"ni"
Number	Cardinal	Concord ++ root
	Ordinal	Concord + cardinal string(except 1-3)
Noun phrase (NP)		Demonstrative + Noun +possessive + Demonstrative + numeral +Adjective or personal Pronoun
Verb Phrase(VP)		Verb + post modifier
Sentence		NP + VP + NP, NP + VP + VP, NP +VP, VP+ NP, VP+ VP, VP
Conjunction		Phrase + conjunction + phrase
		++ joined conjunctively
		+ joined disjunctively

Table 3.5 Ekegusii generalized RE and grammar rules

Category		Generalized Regular expressions or grammar rules
Noun		Gender prefix(number) ++ root
Adjective	Positive	concord prefix(number) ++ root
	Colour	concord prefix (number) + string + colour lexicon
Verb	Positive polarity	Focus(optional) ++concord(subject) ++ tense ++ concord(object) ++ derivative morpheme ++ final vowel
	Negative polarity	concord(negation) ++ tense ++ concord(object) ++ derivative morpheme ++ final vowel
Pronoun	Personal	String(based on agreement)
	Possessive	Concord ++ root
Demonstratives/quantifier		Concord prefix(number) ++ root
Preposition		Concord(number) string OR independent string OR noun + string"ime"(infused)
Number	Cardinal	Multiples of 0-5 Concord ++ root
		6-8 Concord ++ root + Concord ++ root

Ordinal	Concord + cardinal string(except 1-3)
Noun phrase (NP)	Demonstrative + Noun +possessive + Demonstrative + numeral +Adjective or personal Pronoun
Verb Phrase(VP)	Verb + post modifier
Sentence	NP + VP + NP, NP + VP + VP, NP +VP, VP+ NP, VP+ VP, VP
Conjunction	Phrase + conjunction + phrase
++ joined conjunctively	
+ joined disjunctively	
The agreement involves gender, person and number	

3.3.2 Comparative Analysis

The main objective was to compare the Ekegusii and Kikamba generalized descriptive grammars to establish generality in both grammars; hence, the comparative analysis was the best research design. The comparative analysis involves discovering the principles of variation and universality between two or more cases, in our case, grammar. This research design empirically enabled the exploration of the universal grammar theory between Kikamba and Ekegusii grammars (Pickvance, 2001). The comparison was made in two stages. In the first stage, the grammars were compared and where a similarity occurred, it was noted as possible fragments for congruent (shared) grammar; the second stage involved comparing structures' similarities in remainder grammars with the matches ending up as portable grammar segments. In the end, the remainder forms a unique grammar segment. The full comparative analysis process is documented in Figure 3.7 below. The patterns and relationships established in section 3.2.1 for each descriptive grammar were compared in this section to establish the variation principle and universal principle. For example, noun patterns in Kikamba and Ekegusii descriptive grammars were compared to establish similarities or exceptions. The empirical and conceptual comparative analysis strategies (Pickvance, 2005) were used to identify and construct the two grammars' similarities (parameters, principles, features and relations). The empirical strategy used similar observable evidence; for example, a noun in every Bantu grammar has affixes (prefixes or suffixes) and a root as an established similarity. However, noun's affixes were also an abstract concept where the prefix in terms of concepts is a gender while the suffix becomes an infused preposition, thus conceptual strategy. Two comparative analysis approaches were used: differentiating comparison for understanding the differences and universalizing comparison to investigate similarities (Pickvance, 2001). Example 3.1

below shows an example of generalizing demonstrative regular expression. This category consists of a prefix and a root for both languages; the conceptual strategy was used in the prefix since it is an agreement of noun gender based on the number thus concord. The generalized regular expressions and grammar rules for each language presented in Tables 3.4 and 3.5 were compared to generate the shared and portable generalized regular expressions and grammar rules thereby achieving objective one of this thesis and a similar process was followed for parameters. Chapter four will explain the results.

Example 3.1 Generalized RE of demonstrative

Category	Kikamba grammar RE and GR	Ekegusii grammar RE and GR	Generalized regular expression
Demonstratives/ Quantifier	Concord prefix (number) ++ root	Concord prefix (number) ++ root	Concord prefix (number) ++ root

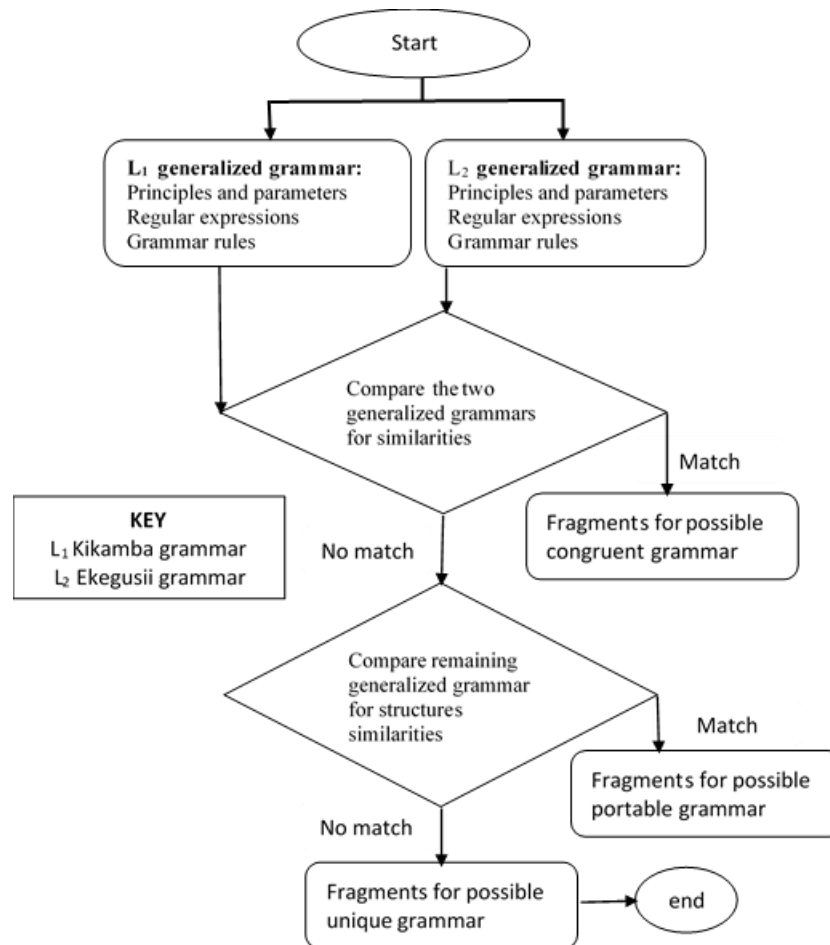


Figure 3.7 Comparative analysis process

3.4. Bantu parameterized grammar development

This stage aimed to develop the Bantu parameterized grammar with clear demarcation of the congruent, portable and unique grammar segments, thereby fulfilling the second objective of developing an approach leveraging on the shared grammar principles and parameters of Kikamba and Ekegusii grammars to produce the Bantu parameterized grammar. The development of the Bantu parameterized computational grammar (congruent grammar) employed a quasi-experimental research design. The quasi-experimental design was chosen because it was the best way of testing the theory of UG on developing congruent grammar and also since the findings were to be generalized to other under-resourced languages. In the experiment, Ekegusii and Kikamba computational grammars were developed independently to ensure no biases occurred in developing the congruent grammar. The development followed the GF modules in Figure 2.4. The lexicon definition, paradigms and the production rules for the syntax were arranged similarly and sequentially in each similar module for the two grammars (i.e., same format and order in noun module for Ekegusii and Kikamba). This kind of arrangement enabled the Linux operating system `diff`²² command to extract similarities and differences between similar GF modules of Kikamba and Ekegusii. The command has been used for similar GF work (Ranta, 2011). For example, Figure D.1 in appendix D shows the output of the comparison of adjective modules.

The shared Bantu parameterized grammar was developed using these similar productions and paradigms, parameters and linearization categories. In the remaining grammar, similar structures of the paradigms and productions were adapted based on similarities of structures to generate the portable grammars; the rest formed unique grammars. The grammar development adopted the GF morphology-driven strategy and modular-driven development, a bottom-up method. It involves first defining the lexicon, smart paradigms based on the regular expression and their respective linearization categories before working on the syntax (Ranta, 2011). The evolutionary prototype model (Carr & Verner, 1997) was used because each function developed had to be iteratively

²² Compares files line by line and output the differences

tested to ensure it worked before moving to the next function. This approach resulted in a morphology analyzer early enough, thereby validating a workable congruent grammar hypothesis. GF provides text output in the command prompt. However, to visualize the parse trees from production rules or paradigms for the grammar, the Graphviz²³ tool was used. It takes simple texts as input and converts them into diagrams. Below is the abstract parse tree for the string “the black boy” in English. Figure 3.8 below represents the same tree visualized using the Graphviz tool. Finally, all the experimental steps are shown in Figure 3.9 below.

```
Lang> p24 -lang=Eng "the black boy "
PhrUtt NoPConj (UttNP (DetCN (DetQuant DefArt NumSg) (AdjCN (PositA
black_A) (UseN boy_N)))) NoVoc
```

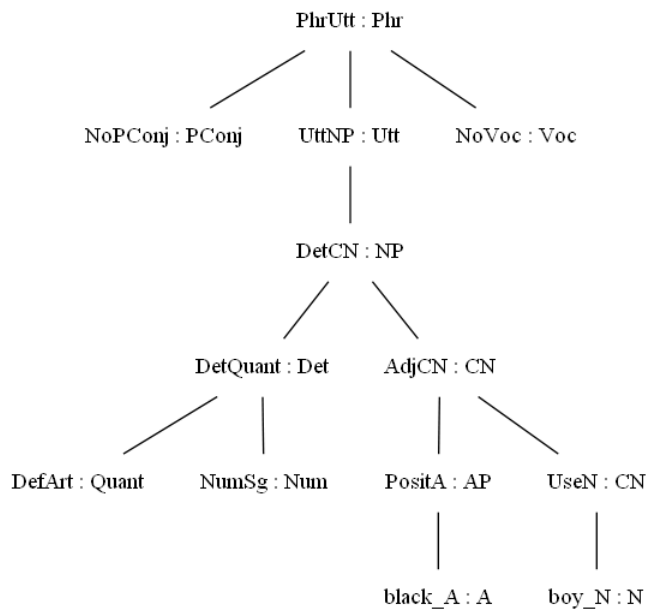


Figure 3.8 Abstract tree

²³ <https://graphviz.org/>

²⁴ p stand for parse in GF and means convert strings to abstract tree

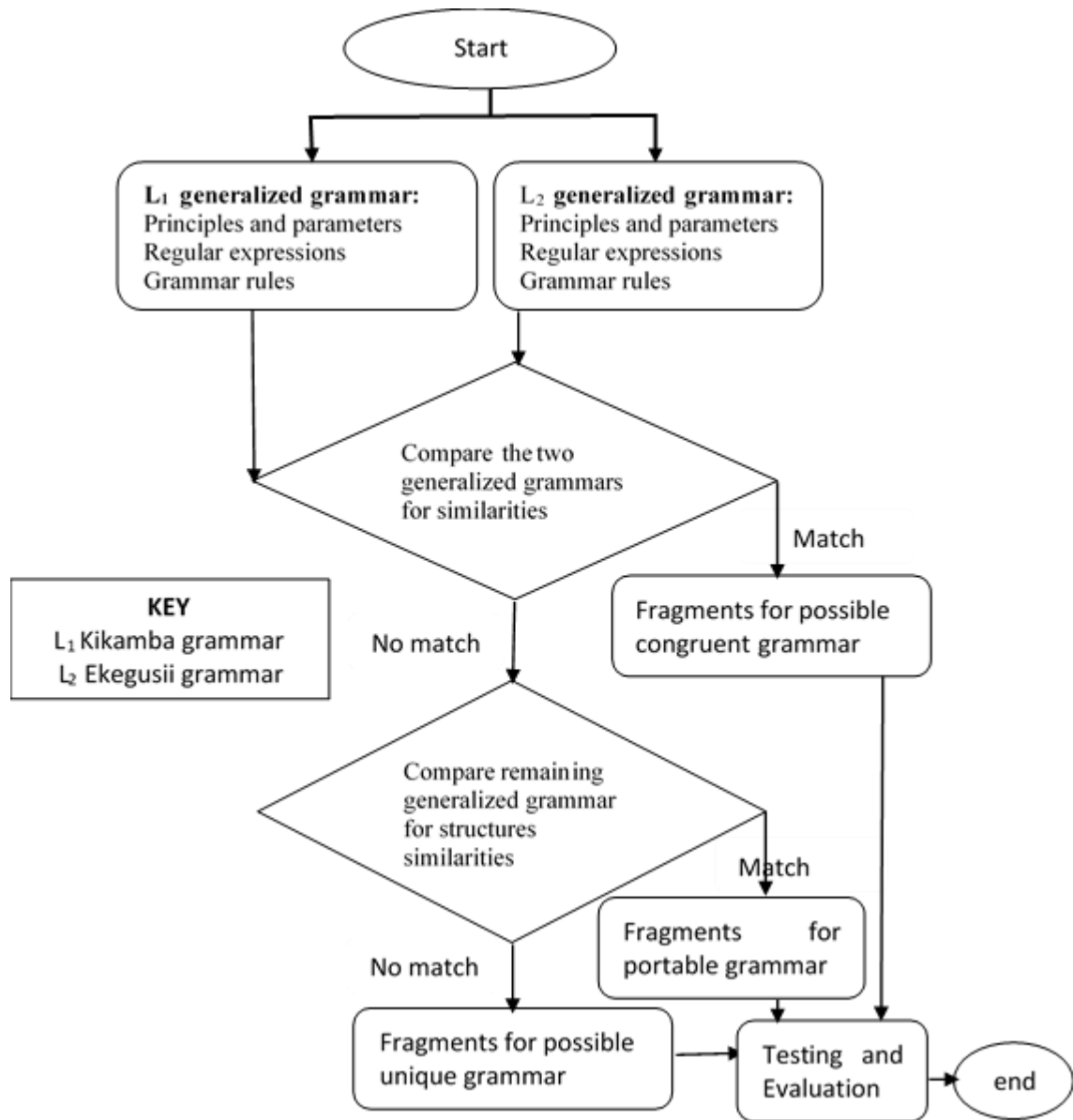


Figure 3.9 Step by step of the Quasi Experiment setup

3.4.1 Morphology

The genders presented in Table 2.1 were core in developing the Bantu parameterized grammar, thus a standardized coding as per Table 3.6. The coding is of GX, where G stands for gender and X is a number starting from one. Each gender pairs two nominal classes based on parameter number (singular and plural) and is separated by an underscore. The “blank”- means gender does not exist in a particular language. The pairing and coding were influenced by Katamba's (2003) work, which summarized 23 Bantu

nominal classes after a comparative analysis of four different studies. This coding was done to ensure uniformity, consistency, easy maintainability of the grammar, and reduce the effort required in bootstrapping a new grammar. The coding was done in the “*diff module*” (module where parameters and principles that are not shared are implemented) of the specific language because although all the languages have genders, their prefix morpheme differs for each language. Kikamba and Ekegusii have 10 and 11 genders respectively (see Table 3.6). The genders were coded in the resource grammar using the parameter *Cgender* as per Definition 3.1 below. The genders form portable grammar segments because they have a similar structure.

Table 3.6 GF coding of Genders

GF coding	Kikamba	Ekegusii
G1	mũ_a	omo_aba
G2	mu_mi	omo_eme
G3	ĩ_ma	e_ci
G4	kĩ_i	eri_ama
G5	ka_tũ	ege_ebi
G6	va_kũ	oro_ci
G7	n_n	aka_ebi
G8	ũ_ma	obo_ama
G9	u_n	oko_ama
G10	kũ_ma	aa
G11	-	ama_ama

Definition 3.1 Gender coding in GF

```
oper
  Cgender : PType ; -- DiffBantu module
oper
  Cgender = CgenderKam ; --diffKam module
  Cgender = CGenderGus ; --diffGus module
param
  CgenderKam = G1|G2|G3|G4|G5|G6|G7|G8|G9|G10 ;
  CGenderGus = G1|G2|G3|G4|G5|G6|G7|G8|G9|G10| G11 ;
```

These genders influence concordial agreements with part of speech tags; hence agreement was implemented using parameter *Agr* and its composition consists of gender, number and person as per Definition 3.2. To reduce over-generation during inflection, especially for verbs, optimization was done so that Person one (P1) and two (P2) were only

applicable to gender one (G1) because animate beings (humans) belong here. The function *toAgr* translates each person's level to the right agreement. Agreement parameters and functions are shared, thus forming a segment of the shared Bantu parameterized grammar.

Definition 3.2 Agreement definition

```
param
Agr =   AgP1  Number | AgP2  Number  | AgP3  Cgender Number  ;
oper
toAgr : Cgender -> Number -> Person -> Agr = \g, n, p ->
  case p of {
    P1 => AgP1  n  ;
    P2 => AgP2  n  ;
    P3 => AgP3  g n } ;
```

Generally, the lexeme definition for linearization of each category followed a similar structure and involved the following as exemplified by Example 3.2 below:

- Definition of the linearization category
- The low-level paradigm
- The lexeme for the category
- Parameter for the category (some had others did not have)

Example 3.2 Lexicon definition

Ekegusii Languages	Kikamba Languages	Gloss
woman_N = regN "omosubati" omo_aba;	woman_N =regN "kiveti" kĩ_ĩ ;	Woman
small_A = regA "nke" ;	small_A = regA "nini" ;	Small
play_V=regV"chiesa";	play_V=regV "thauka" ;	Play
we_Pron =mkPron "intwe" "ito" G1 Pl P1	we_Pron =mkPron "ithyi" "itu" G1 Pl P1	We
;	;	very
very_AdA = mkAdA "mono" ;	very_AdA = mkAdA "vyu" ;	

Using the example drawn from Example 3.2. *regN* is the noun paradigm where a woman belongs; “omosubati” is the lexeme for a woman in the Ekegusii language, while *oma_aba* is the parameter gender to which the noun belongs.

woman_N = regN "omosubati" omo_aba;

3. 4.1.1 Noun

The Bantu language nouns have inherent nominal genders that are key to concordial agreement with the other parts of speech tags implemented as per Example 3.2 for the two

languages. Thus, the noun inflects for number, thus parameter number with singular (Sg) and plural (Pl) as its values. Table 3.7 below summarizes the fragments of the grammar that are shared or can be adapted and it shows that the noun category linearization (*lincat*) is shared, as is the parameter number. The higher regular expressions (smart paradigm) are also shared.

Table 3.7 Noun grammar fragments

Shared grammar fragments	
Lincat	N = {s : Number => Str ; g : Cgender};
Parameters	Num =Sg Pl;
Smart paradigms	mkN,mkN2,mkN3
Low-level paradigms	compoundN, iregN
Adaptation grammar fragments	
Low-level paradigms	mkNoun, regN, verb2snoun

Definition 3.3 below exemplifies the smart paradigm *mkN* with its explanation provided in Table 3.8 below. The functions *mkN2* and *mkN3* are for the higher valency nouns N2 and N3 that take one and two prepositions respectively.

Definition 3.3 Smart paradigm for Noun

```
oper
mkN = overload {
  mkN : Str ->Cgender -> N =\n, g -> lin N (regN n g );
  mkN : (man,men : N)-> Cgender -> N =compoundN;
  mkN : V -> N = \v -> lin N (verb2snoun v G1) ;
  mkN : (man,men : Str) ->Cgender -> N = \s,p,g -> lin N ( iregN s p
g) ; } ;
```

Table 3.8 Noun Paradigms

Function	Type	Explanation
mkN	Str ->Cgender -> N	Function <i>regN</i> takes in a string and gender and returns regular words forms
mkN	(man,men : N)-> Cgender -> N	Function <i>compoundN</i> takes in two strings of nouns and gender and generates compound noun forms
mkN	V -> N	Function <i>verb2snoun</i> takes in a verb and generates a noun in the gender G1.
mkN	(man,men : N)-> Cgender -> N	Function <i>iregN</i> takes in two strings of nouns and gender, then assign one as singular and the other plural

The low-level paradigms *CompoundN* and *iregN* are shared. The *CompoundN* paradigm makes regular nouns using two strings and uses RE *regN* when supplied with two strings of a singular compound noun to generate all possible noun forms(e.g., mundu muume gloss male) and the worst-case paradigm *iregN* that takes all possible forms that are listed in the lexicon module. Paradigm *mkNoun* (make noun) is used to assign all forms of inflections to the right number maintaining the gender since it will be required in constructing common nouns (CN) and concord agreement. The paradigms are provided in Definition 3.4 below.

Definition 3.4 Shared low-level paradigms

```
oper
compoundN : N -> N ->Cgender-> N = \mundu,muume,g -> {
    s = \\n=> mundu.s! n ++ muume.s!n;
    g = g ;
    lock_N = <> } ;
iregN :Str-> Str ->Cgender -> Noun= \man,men,g ->mkNoun man men g;
mkNoun :Str-> Str ->Cgender -> Noun= \man,men,g -> {
    s = table{Sg => man ; Pl => men } ;
    g = g; } ;
```

The paradigm *regN*, as illustrated by Figure 3.10 for Ekegusii and 3.11 for Kikamba, was a grammar-specific paradigm because the gender prefix represented by the function *PrefixPINom* and the roots are very specific for each grammar. However, the structure for *regN* is similar, and thus forms part of the portable grammar segment. The morphophonological rules were implemented in the RE for the noun. The function *Predef.drop* in Figures 3.10 and 3.11 for class gender G1 and G2 is used to implement the morphophonological rules. For example, the child which is a noun in Ekegusii belongs to Omo-aba, therefore, its singular should be “omo-ana” however, the meeting of vowels o-a changes to “wa” hence the below output of “omwana”.

```
Lang> l -table child_N
s Sg : omwana
s Pl : abana
```

```

regN : Str ->Cgender -> Noun = \w, g -> let wpl = case g of {
  G1 =>case w of {
    "omwo" + _ => "aba" + Predef.drop 3 w ;
    "omw" + _ => "ab" + Predef.drop 3 w ;
    _ => PrefixPlNom G1 + Predef.drop 3 w};
  G2 =>case w of {
    "omw" + _ => "emi" + Predef.drop 3 w ;
    _ => PrefixPlNom G2 + Predef.drop 3 w};
  G3 => "chi" + Predef.drop 1 w;
  G4=> case w of { "ri" + _ => "ama" + Predef.drop 2 w ;
    _ => PrefixPlNom G4 + Predef.drop 1 w};
  G10 => [];
  G11=> w;
  _ => PrefixPlNom g + Predef.drop 3 w};
in mkNoun w wpl g ;

```

Figure 3.10 Ekegusii regN paradigm

```

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 };
  G3 => PrefixPlNom G3 + Predef.drop 1 w;
  G5 => case w of {"ka" + _ => "twa" + Predef.drop 2 w ;
    _ => PrefixPlNom G4 + Predef.drop 2 w };
  G4=> case w of {"ky" + _ => "sy" + Predef.drop 2 w ;
    _ => PrefixPlNom G4 + Predef.drop 2 w };
  G7 => w;
  G8 |G9 => PrefixPlNom g + w;
  _ => PrefixPlNom g + Predef.drop 2 w};
in mkNoun w wpl g ;

```

Figure 3.11 Kikamba regN paradigm

Finally, the paradigm *verb2snoun* that forms nouns from verbs is also specific for each grammar because the prefixes and suffixes added to the verbs are unique to each grammar, as shown in Definition 3.5 below. In the three languages, nouns from verbs are formed for class gender G1 which has animate things.

Definition 3.5 Verbs to Noun paradigms

Ekegusii regular expression for verbs to a noun

Oper

```
verb2snoun : Verb -> Cgender -> Noun = \v,g->
  let wp = "omo" + init(v.s ! VGen) + "i" ;
      wpl = "aba" + init(v.s ! VGen) + "i" in
  iregN wp wpl g ;
```

--Kikamba regular expression for verbs to a noun

Oper

```
verb2snoun : Verb -> Cgender -> Noun = \v,g->
  let wp = "mu" + init(v.s ! VGen) + "i" ;
      wpl = "a" + init(v.s ! VGen) + "i" in
  iregN wp wpl g ;
```

The noun's lexicon was defined by providing the singular lexeme and its gender (see Appendix B, B.1). These were processed by the explained paradigms depending on the nature of the defined lexicon resulting in a noun morphological inflection table that consists of a maximum of two-word forms for each number. An example in the Ekegusii language by linearization of the noun “tree” is highlighted below (for more examples, see Appendix B.1).

```
lang> linearise -table tree_N
s Sg : omote    ---gloss tree
s Pl  : emete   --- gloss trees
```

3.4.1.2 Adjective

Adjective inflects for number and gender and the parameter *AForm* was used to represent the two-variable features (parameters) for the concordial agreements, namely gender and number. Positive, comparative and adverbs are the three degrees of adjectives represented by the parameter in the form *AAdj*, *AComp* and *Advv* respectively. Only Kikamba grammar had a comparative degree formed by adding the infix “ang” before the adjective root's final vowel. Some adjectives have adverbs others do not. Table 3.9 below summarizes the different fragments for the adjective in the construction of the Bantu parameterized grammar.

Table 3.9 Fragments of Adjective grammar

Shared grammar fragments	
Lincat	A = {s : AForm => Str };
Smart paradigms	mkA,mkA2
Adaptation grammar fragments	
Low-level paradigms	regA,cregA,iregA
Unique grammar fragments	
Parameter	AForm

Even though the adjective category linearization is shared, the parameter *AForm* is defined differently for the two grammars, as shown below:

```
param
AForm = AAdj Cgender Number | Advv;-- for Ekegusii
AForm = AAdj Cgender Number | AComp Cgender Number | Advv;--Kikamba
```

The smart paradigm *mkA* has two forms, as shown in Definition 3.6. The first takes an adjective lexeme and, depending on the paradigm used to define the lexeme *regA* or *cregA*. It generates an adjective inflection table for a normal regular adjective or a colour adjective. Only white, black, and red colour adjectives are achieved via paradigm *regA*; the rest is through *cregA*. The structure consisted of a prefix specific for each number and gender, the same as the prefixes of pronouns; hence, the same operation used in pronouns is used here. The prefix is followed by a string “*color of*”, then the color lexeme. The two grammars do not have a regular pattern for forming adjectives and adverbs. Besides, there is no comparative degree in Kikamba grammar for colour adjectives; hence the fields are empty. The two languages share the structure and the definition of *cregA* paradigm is given below. The paradigm *regA* takes one string and assigns an empty string of adverbs and passes two strings to paradigm *regAdj*, which generates the adjective inflection table. However, the definition *regAdj* is too big, thus given in Appendix B B.2. The definition for *regA* is given below:

```
regA :Str->{s : AForm => Str}= \adj ->regAdj adj [];
```

Definition 3.6 Smart paradigm for Adjective

Oper

```
mkA = overload {
  mkA : Str -> A = \a -> lin A (regA a |cregA a );
  mkA : (fat,fatter : Str) -> A =\a,b -> lin A (iregA a b| regAdj a b );
} ;
```

Kikamba language cregA paradigm

```
cregA : Str-> {s : AForm => Str} = \seo -> {s = table {
  AAdj g Sg=> ProunSgprefix g + "a langi wa" ++ seo;
  AAdj g Pl=> ProunPlprefix g + "a langi wa" ++ seo;
  AComp g n => [];
  Advv =>[]}} ;
```

Ekegusii language cregA paradigm

```
cregA : Str-> {s : AForm => Str} = \seo -> {
  s = table {
    AAdj g Sg => ProunSgprefix g ++ "eragi ya" ++ seo;
    AAdj g Pl=> ProunPlprefix g ++ "eragi ya" ++ seo;
    Advv=> []} } ;
```

The output in Example 3.3 exemplifies the *cregA* paradigm given the colour blue. It outputs “colour of blue” depending on gender and the number of the noun.

Example 3.3 Colour Blue Adjective GF output

```
Lang> linearise -table blue_A
```

```
s (AAdj G1 Sg) : o eragi ya buluu
s (AAdj G1 Pl) : ba eragi ya buluu
s (AAdj G2 Sg) : o eragi ya buluu
s (AAdj G2 Pl) : ya eragi ya buluu
s (AAdj G3 Sg) : ya eragi ya buluu
s (AAdj G3 Pl) : chia eragi ya buluu
s (AAdj G4 Sg) : ria eragi ya buluu
s (AAdj G4 Pl) : a eragi ya buluu
s (AAdj G5 Sg) : kia eragi ya buluu
s (AAdj G5 Pl) : bia eragi ya buluu
s (AAdj G6 Sg) : rwa eragi ya buluu
s (AAdj G6 Pl) : chia eragi ya buluu
s (AAdj G7 Sg) : ka eragi ya buluu
s (AAdj G7 Pl) : ba eragi ya buluu
```

```

s (AAdj G8 Sg) : bwa eragi ya buluu
s (AAdj G8 Pl) : a eragi ya buluu
s (AAdj G9 Sg) : kwa eragi ya buluu
s (AAdj G9 Pl) : eragi ya buluu
s (AAdj G10 Sg) : a eragi ya buluu
s (AAdj G10 Pl) : eragi ya buluu
s (AAdj G11 Sg) : aa eragi ya buluu
s (AAdj G11 Pl) : aa eragi ya buluu

```

The second form of *mkA* takes in two strings: two irregular adjectives, one for each number, where the *iregA* paradigm is used to generate the inflection table, and a regular adjective and corresponding adverb and *regAdj* then generate the inflection table. The distinction on which paradigm to use for two strings is made at the definition of the adjective lexemes. The definition of *iregA* is the same for the two grammars.

```

iregA : Str-> Str -> {s : AForm => Str} = \seo, seoo -> {
  s = table {
    AAdj g Sg=> seo;
    AAdj g Pl => seoo;
    Advv=> [] } };

```

In a similar manner to the noun, morphophonological rules for adjectives were implemented in the RE. Below is a snippet extract from the Kikamba adjective RE for class gender *mu_mi* and singular number. Take for example the NP “a black tree” the root for the black lexeme in Kikamba is “iu”, therefore translating the above the expected outcome should be “muti mi-ui” because the agreement concord is “mi”. However in the snippet when “i” and “u” meet then “wi” is generated as shown in blue font. Consequently, the correct output is shown below the snippet. In the results also the NP is parsed from Kikamba and linearized to English, Ekegusii and Kikamba

```

AAdj G2 Sg=>case Predef.take 1 seo of {
  "i"  => "mw" + seo;
  "a"  => "my" + seo;
  "u"  => "m"  + seo;
  _    => ConsonantAdjprefix G2 Sg + seo };

```

```
Lang> p -lang=Eng -cat=NP "a black tree" | 1
muti mwiu

Lang> p -lang=Kam -cat=NP "muti mwiu" | 1
the black tree
omote omomwamu
muti mwiu
```

3. 4.1.3 Verb

The Grammatical Framework resource grammar library, by default, provides positive and negative polarities, past, present, future, and conditional tenses, as well as simultaneous and anteriority (Ranta, 2011). The positive polarity is implemented using the subject marker morpheme, while the negation morpheme is used in the negative polarity. The two morphemes require extra grammar features in order to allow agreement like gender, number, and person (first, second and third). The tense or sometimes aspect morpheme was used to implement both anterior and tense. Other morphemes, as presented in Table 2.2, were also used to implement the verbs. The record type implementation for the verb in all the languages is as defined in Definition 3.7 below:

Definition 3.7 Verb linearization

```
lincat
  V , VS, VQ, VA, VV, V2S, V2Q, V2V, V2A= Verb ;
oper
  Verb = { s :VForm => Str;
           progV:Str;
           imp : Polarity => ImpForm => Str;
           s1 : Polarity => Tense => Anteriority => Agr=> Str };

```

The operation of the verb has a record of four strings. String *s* can be defined as the various forms of verbs that can be generated in a specific language. In this string three moods (subjunctive, indicative and conditional) were implemented here. The two languages had common verb forms: infinitive, extensional or derivative morphology form and general form with a final vowel “a”. Kikamba language had extra verb forms of present progressive form, past tense form and present definitive form, while Ekegusii language had anterior, future and negation form as the extra forms.

The second record string had *progV* for the progressive verb and *imp* for an imperative verb. The imperative verb is implemented in the *sentence module* of RGL with the linearization category type shown below. It inflects for polarity and parameter *impForm* (number and Boolean with true being a polite request while false is a command) and this implemented imperative mood.

```
lincat
Imp = {s : Polarity => ImpForm => Str} ;
```

To model derivational/ extensional morphology, the parameter *VExte* was used. Both grammars shared the following verb derivations: passive, applicative, reciprocal, and causative. However, Ekegusii grammar had an extra stative and Kikamba grammar was distributive, as previously indicated in Table 2.2. Since every verb derivation had a unique morpheme, the implementation is done in the specific grammar but maintains a similar structure so that in the future, less effort will be required to bootstrap extra or fewer verb derivations. Thus, they form part of the portable segment of the Bantu parameterized grammar and are illustrated in Definition 3.8

Definition 3.8 Parameter for verb derivative morphology

```
Param ---derivative morphology for Ekegusii grammar
  VExte = EPassive | EApplicative | EReciprocal | ECausative | EStative;
Param ---- derivative morphology for Kikamba grammar
  VExte = EPassive | EApplicative | EReciprocal | ECausative
| EDistributive ;
```

The smart paradigm *mkV* took one string as input in order to generate the inflection table. Table 3.10 below shows the low-level paradigms *regV* and *mkVerb* mathematically in terms of the n-tuple input-output function. For example, The Kikamba language *mkVerb* paradigm takes as input four strings and returns an inflection table of 435-word forms. The forms are many because a string must be generated for concord agreement, number, tense, polarity and person. For more information on the actual implementation of the low-level paradigms, see Appendix B.3.

Table 3.10 Verb Paradigms

Language	Inflection function
Kikamba	regV: String ¹ → String ⁵ mkVerb: String ⁵ → String ⁴³⁵
Ekegusii	regV: String ¹ → String ⁵ mkVerb: String ⁵ → String ⁴⁰³

Table 3.11 summarizes the different fragments for the verb morphology grammar.

Table 3.11 Summary of Verb grammar fragments

Shared Bantu parameterized grammar fragments	
Lincat	As per Definition 3.7
Parameters	IMPForm = Com Pol Person = P1 P2 P3 Pol = Pos Neg Tense= Pres Fut Past Cond Agr = AgP2 Number AgP3 Cgender Number AgP1 Number
Smart paradigms	mkV,mkV2 ,mkV3
Adaptation grammar fragments	
Low-level paradigms	regV,iregV,mkVerb
Parameters	As per Definition 3.8

3. 4.1.4 Numerals

Numerals²⁵ are either cardinal or ordinal. Cardinal describes quantity while ordinal shows order and is represented in digits or words, for example in cardinal, 12 and twelve respectively. Both formats are supported in GF. The GF numeral implementation is based on Hammarström and Ranta’s (2004) work. The numeral linearization type implementation is exemplified in Definition 3.9 and was shared by the two grammars:

Definition 3.9 shared Numeral definition and Parameter

```

lincat
Numeral = {s : CardOrd => Cgender => Str ; n : Number} ;
Digits  = {s : CardOrd => Cgender => Str ; n : Number} ;
Param
    CardOrd= NCard | Nord;
    DForm  = unit | teen | ten |hund ;

```

²⁵ <https://www.englisch-hilfen.de/en/grammar/zahlen.htm>

The numeral and digit categories have gender as a variable feature and parameter *CardOrd* and inherent feature of number as shown in the linearization provided in Definition 3.9. The values of parameter *CardOrd* are cardinal (*Ncard*) and ordinal (*Nord*) numerals. The numeral one is the only digit that has the value of a number as singular; all others are plural. The values for parameter *DForm* were *unit*, *teen*, *ten* and *hund*. The *unit* is for numerals between zero and nine. The *teen* is between 11 and 19, while *ten* is for multiples of ten and *hund* for multiples of hundreds. GF²⁶ implements numbers ranging from 0 to 999,999.

In building the numeral, there was gender agreement (concord) for the cardinal numeral one to five and their multiples in the two grammars. In addition, in Ekegusii grammar, where counting ends at number five, the gender agreement was extended up to number eight. Generally, the numerals six to eight and their multiples in the Ekegusii language are constructed by recursion between one and five. For example, eighty would be constructed as (fifty thirty), as exemplified by Figure 3.12.

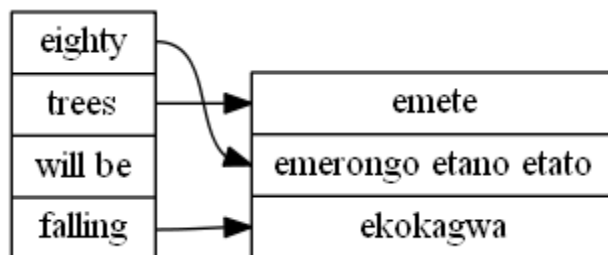


Figure 3.12 word alignment

For the numerals, three to five and their multiples of ones, tens, teens and hundreds, the low-level paradigm *mkNum* is used. *RegNum* was used for numerals six to nine and their multiples for Kikamba grammar, while *mkNum* paradigm was used in Ekegusii for numeral two to five and their multiples of ones, tens, teens and hundreds. All other numerals between one and nine and their multiples of ones, tens, teens, and hundreds had a specific function to model them. Since the numeral counting for the Ekegusii language ends at number five, the paradigms were unique for each language.

²⁶<https://www.grammaticalframework.org/lib/doc/gfdoc/Numeral.html>

The ordinal numeral was modeled by adding a disjunctive prefix of singular “of” in the specific language, while digits were modeled using a similar function for all languages. The function *IDig*, which took argument digit, returned digits 0 to 9, while function *IIDig* which took argument digit, followed by digits, returned numerals with at least two digits. The operation *mk3Dig* created the cardinal digits and ordinal digits by attaching the cardinal digits' disjunctive prefixes. For the actual implementation of the low-level paradigms, see Appendix B.4. After numerals' morphology was done at the GF *numeral module*, five rules were constructed at the Noun module for numerals as illustrated in Definition 3.10. Rules 1 and 2 represent the formation of cardinal numerals by digits and words respectively. Rule 3 represents cardinal numerals that are modified by adverbs. Rules 4 and 5 are ordinal numerals implemented in digit and word forms respectively. These rules are in shared Bantu parameterized grammar. Figures 3.13 and 3.14 below show an example of a unit and thousands of cardinal numerals for gender G1 for all languages, while Figure 3.15 exemplifies cardinal numerals in digits and words for gender G1(omo_aba) in Ekegusii. For the gloss -“four hundred and eighty-two”. As stated earlier Ekegusii does not have numbers six to nine. Thus, eighty become fifty and thirty hence “emerogo atano ne batato” as shown in Figure 3.15.

```
Lang> lin n5
Five --English
Batano --Ekegusii
Atano -- Kikamba
```

Figure 3.13 Cardinal five

```
Lang> p -lang=Eng "six hundred thousand nine hundred and forty " | l
six hundred thousand nine hundred and forty --English
chiribu amagana atano oyamo amagana kianda na emerongo ane --Ekegusii
ngili maana nthathatu maana kenda na miongo ina --Kikamba
```

Figure 3.14 A Large cardinal numeral example

Definition 3.10 Numeral rules

1. `NumDigits n = {s = n.s ! NCard ; n = n.n} ;`
2. `NumNumeral numeral = {s = numeral.s ! NCard; n = numeral.n} ;`
3. `AdNum adn num = {s = \\g => adn.s ++ num.s!g ; n = num.n} ;`
4. `OrdDigits n = { s = n.s ! NOrd} ;`

```
5. OrdNumeral numeral = {s = numeral.s ! NOrd} ;
```

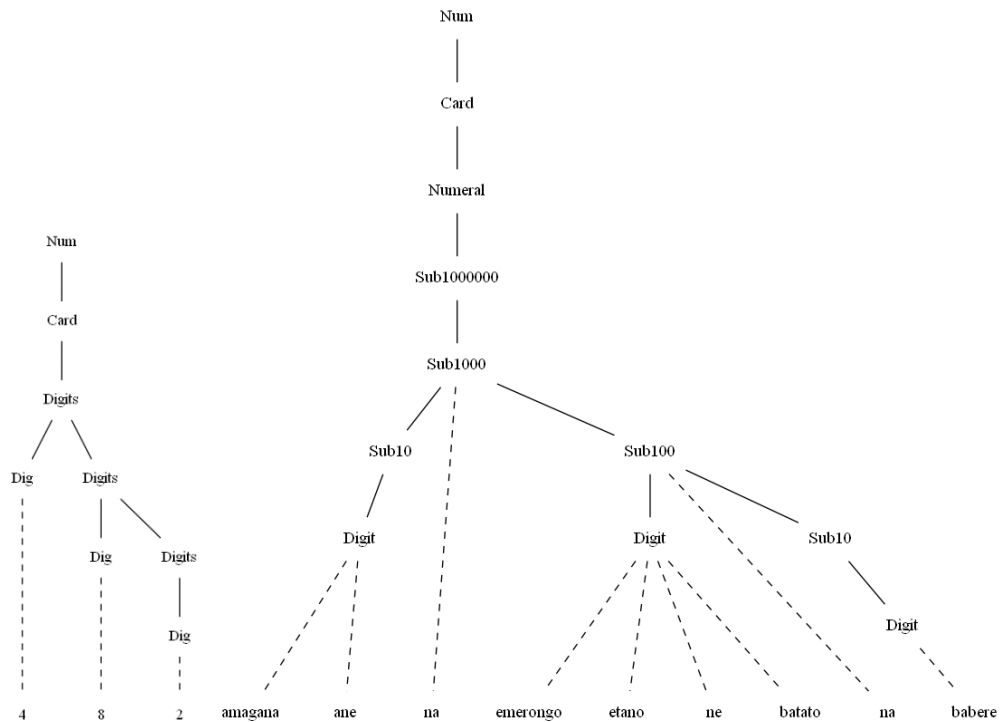


Figure 3.15 Numeral in Ekegusii Language

3. 4.1.5 Pronoun

Personal and possessive pronouns are the two forms of pronouns implemented in GF. The personal pronoun acts as a noun phrase, and thus requires agreement in terms of gender, number, and person. The possessive pronoun in GF is implemented as a quantifier. Thus, a determiner hence inflects for gender and number. The parameter *PronForm* with values *Pers* for the personal pronoun and *Poss* for possessive pronoun are used to model the linearization (*lincat*) as exemplified in Definition 3.11 that is shared between the two grammars.

Definition 3.11 Shared Pronoun parameters and linearization

```
lincat
Pron = { s: PronForm=>Str;          a : Agr  };
param
Agr = AgP2  Number | AgP3  Cgender Number | AgP1  Number ;
PronForm= Pers | Poss Number  Cgender;
```

Figure 3.16 below shows the shared paradigm *mkPron* used to implement pronouns that takes five arguments inputs (two strings- one for personal and a stem for the possessive, gender, number, and person in that order) as exemplified by the lexicon of the pronoun “he” for the Ekegusii language below. The output comprises two sets of strings, namely: the personal pronouns that act as a noun phrases and the inflection table for the possessive pronoun based on gender and number. Example 3.4 below shows the output of the pronoun “he” in Ekegusii grammar. The linearization, parameters, and paradigms for pronouns are shared, while the structure for lexemes is adapted across the grammar.

```
he_Pron = mkPron "ere" "je" G1 Sg P3 ;
```

```
oper
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 ;
      <Pl, _> => ProunPlprefix g + mine}
    } ;
  a = toAgr g n p } ;
```

Figure 3.16 Pronoun paradigm

Example 3.4 Pronoun "he" output

```
Lang> l27 -lang=Gus -table he_Pron
s Pers : ere
s (Poss Sg G1) : oje
s (Poss Sg G2) : oje
s (Poss Sg G3) : yaje
s (Poss Sg G4) : riaje
s (Poss Sg G5) : kiaje
s (Poss Sg G6) : rwaje
s (Poss Sg G7) : kaje
s (Poss Sg G8) : bwaje
s (Poss Sg G9) : kwaje
s (Poss Sg G10) : aje
s (Poss Sg G11) : aaje
s (Poss Pl G1) : baje
s (Poss Pl G2) : yaje
s (Poss Pl G3) : chiaje
s (Poss Pl G4) : aje
s (Poss Pl G5) : biaje
s (Poss Pl G6) : chiaje
s (Poss Pl G7) : baje
s (Poss Pl G8) : aje
s (Poss Pl G9) : je
s (Poss Pl G10) : je
s (Poss Pl G11) : aaje
```

²⁷ L means linearization (converting abstract tree to strings)

3. 4.1.6 Preposition

Some prepositions inflect for gender and number in Bantu languages, for example, the preposition “of” which was established through elicitation. Most of the other prepositions are just strings like in most of the other languages. It is also noted that some prepositions are fused with nouns conjunctively in Kikamba grammar and disjunctively for Ekegusii grammar resulting in a locative noun. For example, Figure 3.17 below shows the infusion of the preposition “on” to the noun “table” using the morpheme “ni” in Kikamba grammar. The string *s* was used to implement the preposition, while *s1* was the string for infusion in the *mkPrep* paradigm shown in Definition 3.12. Finally, the Boolean operator determined whether a specific preposition can be infused or not (true value meaning fused and vice versa). The linearization and paradigm *mkPrep* form a fragment of the congruent Bantu parameterized grammar segment, while the lexeme definition, because of shared structure, formed the portable grammar segment. Example 3.5 below shows how the paradigm *mkPrep* works by using the preposition “of” in Ekegusii.

Definition 3.12 Preposition definition

```
lincat
  Prep = ResBantu.Preposition;
Oper
Preposition={s: Number => Cgender => Str; s1:Str; isFused: Bool} ;

mkPrep = overload {
  mkPrep : Str ->Bool-> Prep = \str,bool ->
  lin Prep {s = \n,g => str ; s1= infusedstring;isFused = bool } ;
  mkPrep : (Number => Cgender => Str) ->Bool-> Prep = \t,bool ->
  lin Prep {s = t ; s1= infusedstring; isFused = bool} ;};
```

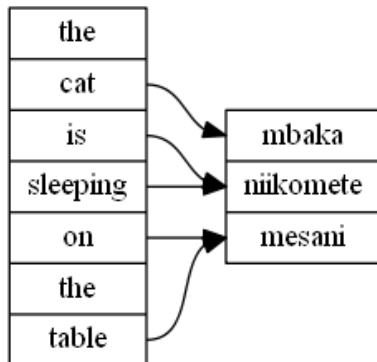


Figure 3.17 Preposition infusion

Example 3.5 Output of preposition “of” using mkPrep paradigm

```
Lang> l -table possess_Prep
s Sg G1 : bwo
s Sg G2 : bwo
s Sg G3 : ya
s Sg G4 : ria
s Sg G5 : kia
s Sg G6 : rwa
s Sg G7 : ka
s Sg G8 : bwa
s Sg G9 : gwa
s Sg G10 : a
s Sg G11 : aa
s Pl G1 : ba
s Pl G2 : ya
s Pl G3 : chia
s Pl G4 : ye
s Pl G5 : bi
s Pl G6 : chia
s Pl G7 : bia
s Pl G8 : a
s Pl G9 : a
s Pl G10 : a
s Pl G11 : aa
s1 :
```

3.4.1.7 Quantifier

Quantifiers inflect for gender and number. Bantu languages have three quantifiers namely: proximal, distant and aforementioned (Kaviti, 2004). GF library has defined the first and the second types respectively. The aforementioned quantifier was implemented in the *extra module*, where extra language-specific elements are usually implemented in GF. On the extra Bantu abstract module of the Bantu Functor, the function below was declared for the aforementioned quantifier for all languages in consideration and Example 3.6 below shows the output for Kikamba grammar.

```
fun
that_far_Quant : Quant ;
```

Example 3.6 Quantifier output

```
AllKamAbs> l -table that_far_Quant
s Sg G1 : usu
s Sg G2 : usu
s Sg G3 : yiu
s Sg G4 : kyu
s Sg G5 : kau
s Sg G6 : vau
s Sg G7 : isu
s Sg G8 : usu
s Sg G9 : usu
s Sg G10 : kuu
s Pl G1 : asu
s Pl G2 : isu
s Pl G3 : asu
s Pl G4 : isu
s Pl G5 : tuu
s Pl G6 : ku
s Pl G7 : isu
s Pl G8 : asu
s Pl G9 : isu
s Pl G10 : asu
```

The quantifier can also be formed from possessive pronouns. The rule below was developed in the *noun module* that takes an input string of possessive pronouns and produces an inflection table based on gender and number.

```
PossPron pron = {
  s = \\n,g => pron.s!Poss n g
} ;
```

3. 4.1.8 Determiner

Determiners show an indefinite number of people or objects (Mbuvi, 2006). They include but are not limited to every, much, all, and so on. They inflect for gender plus an inherent number in both grammars and its linearization is defined as per Definition 3.13. Moreover, some come before the noun they modify while others come after the noun. To show the determiner's position in relation to the noun it modifies, *isPre*, a Boolean parameter is used. *True* indicates it comes before and *false* shows it comes after.

Definition 3.13 Determiner linearization.

```
lincat
Det = { s : Cgender => Str ; n : Number ; isPre: Bool} ;
```

The lexeme definition was similar in structure as illustrated below using “much” and “many” determiners for both languages thus the linearization was shared in both grammars while the lexemes definition was ported. Example 3.7 below shows an output of the determiner “many” in both languages.

```
much_Det , many_Det = { s =\\g => Manyprefix g + "ingi" ; n = Pl; isPre
= True};-Kikamba
much_Det , many_Det = { s =\\g => Many_prefix g + "nge" ; n= Pl; isPre
=True };-Ekegusii
```

Example 3.7 Determiner output example

Kikamba grammar

```
Lang> l -table many_Det
s G1 : aingi
s G2 : miingi
s G3 : maingi
s G4 : mbiingi
s G5 : twiingi
s G6 : kwiingi
s G7 : mbiingi
s G8 : maingi
s G9 : mbiingi
s G10 : maingi
```

Ekegusii grammar

```
Lang> l -table many_Det
s G1 : abange
s G2 : emenge
s G3 : cininge
s G4 : amange
s G5 : ebinge
s G6 : cininge
s G7 : ebinge
s G8 : amange
s G9 : amange
s G10 : aninge
s G11 : amange
```

3.4.1.9 Adverbs

Adverbs do not inflect; hence are mere strings in their definition. However, in section 3.3.2.3 on adjectives, there were adverbs formed out of adjectives. Therefore, to accommodate them at the syntax phase and since the adjective inflects for gender and number, the adverb was configured to inflect for agreement (gender, number and person). Person three (P3) was used as a constant. This implementation was shared across the two grammars. Hence, their linearization and paradigm mkAdv is given in Definition 3.14 below.

Definition 3.14 Adverb definitions

```
lincat
Adv = {s : Agr => Str } ;
oper
mkAdv s = lin Adv { s= \\_ => s };
```

3.4.2 Syntax

The syntax is implemented using the dominant SVO topology shared in Bantu languages. Parameters were exchanged among the categories in order to ensure syntactic agreement (concord agreements). The *V* topology is also implemented primarily where personal pronouns are used as the subject (S), thus pro-drop of the subject since it is represented in the verb using the subject marker. Finally, *SV* is implemented where the verb does not have a complement.

3.4.2.1 Common Noun (CN)

GF has primarily been used for Indo-European languages where the CN is combined with an adjective to form a noun phrase or another CN and later, a determiner can be added as a pre-modifier or post-modifier of the CN. However, in Bantu languages, the determiner is added between the adjective and the noun. Consequently, the design of CN used two strings, *s* to hold the CN and string *s2* to hold the adjective. It would, therefore, be more comfortable to add a determiner between string *s* and *s2*. The gender was retained from the noun since it was used in agreement (concord) later. The noun and a single word *CN* are the same since Bantu languages do not have articles. Definition 3.15

below illustrates the CN's design and linearization that is part of the shared Bantu parameterized grammar.

Definition 3.15 CN Definitions

```
lincat
CN = CNoun;
oper
CNoun : Type = {S, s2 : Number => Str; g : Cgender };
```

Definition 3.16 below illustrates the nine syntax rules/production rules for CNs that are implemented. In rules one and two, the CNs are constructed from the noun of valency one and two (with a preposition), while rule three makes a CN from NP and a relational noun. Rules four to eight represent a CN modified by an adjective, relative clause, adverb, sentence and noun phrase respectively. Finally, rule nine has the usage of “of” together with a noun phrase. Everything in the modeling of CN (rules and definitions) was part of the shared Bantu parameterized grammar.

Definition 3.16 CN rule definitions

```
1. UseN n = { s = n.s ; s2 = \\_ => [] ; g = n.g }
2. UseN2 n = { s = n.s ; s2 = \\_ => [] ; g = n.g } ;
3. ComplN2 n2 np = {
    s = \\n => n2.s ! n ++ n2.c2.s!n!n2.g ++ np.s ! Nom ;
    s2 = \\_ => [] ; g = n2.g } ;
4. AdjCN ap cn = {
    s = cn.s ; g = cn.g ;
    s2 = \\n => cn.s2! n ++ ap.s ! cn.g ! n } ;
5. RelCN cn rs = {
    s = \\n => cn.s ! n ++ rs.s ! AgP3 cn.g n ;
    s2 = \\n => [] ; g = cn.g } ;
6. AdvCN cn ad = {
    s = \\n=> cn.s ! n ++ ad.s!AgP3 cn.g n ;
    s2 = \\n => [] ; g = cn.g } ;
7. SentCN cn sc = {
    s = \\n => cn.s ! n ++ sc.s ;
    s2 = \\n => [] ; g = cn.g } ;
8. ApposCN cn np = let agr = complAgr np.a in {
    s = \\n => np.s ! Nom ++ cn.s ! n ;
    s2 = \\n => "" ; g = cn.g } ;
9. PossNP cn np =let agr = detAgr np.a in
    {s = \\n,c => cn.s ! n ! Nom ++ possess_Prep.s! n!cn.g ++
    np.s ! NPoss;
    s2 = \\n => [] ; g = cn.g } ;
```


Figure 3.18 below shows a parse tree and word alignment in the Kikamba generated using Graphviz software to demonstrate the working of rules 1, 4 and 6, while Example 3.8 shows the rules' actual functions in an abstract syntax tree marked in blue. The gloss is “the brown house on the hill.”

Example 3.8 Example of CN rules

```

■ Converting the English string to an abstract syntax
Lang> p -cat=CN -lang=Eng " brown house on the hill"
AdjCN (PositA brown_A) (AdvCN (UseN house_N) (PrepNP on_Prep (DetCN
(DetQuant DefArt NumSg) (UseN hill_N))))
■ Translating the tree to strings in Kikamba and English languages
Lang> l AdjCN (PositA brown_A) (AdvCN (UseN house_N) (PrepNP on_Prep
(DetCN (DetQuant DefArt NumSg) (UseN hill_N))))
Brown house on the hill ---English
Nyumba kiimani ya langi wa kaki ---Kikamba

```

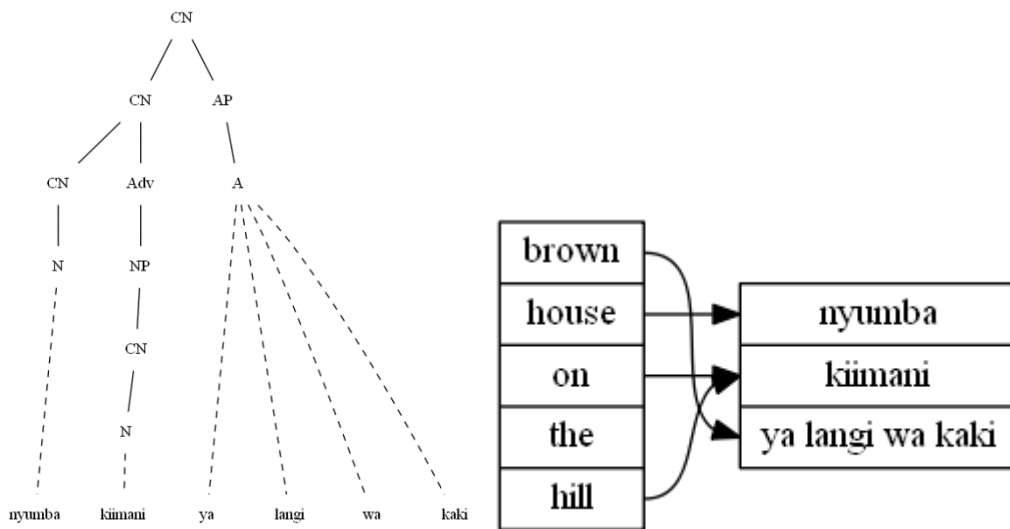


Figure 3.18 A Parse tree and word alignments

3.4.2.2 Determiner Phrases (Det)

Syntactically, determiner phrases can be formed from numerals and quantifiers with the latter being the central and the former optional. The determiner inflection was explained in morphology section 3.3.4.7. Here the focus is constructing a determiner from more than one category (speech tags). Two production rules were modeled. In the first one,

the determiner is formed from a quantifier and numeral, while in the second rule, there is the addition of an ordinal numeral. The determiner is a post modifier of a noun hence the reason the Boolean *isPre* is true. Figure 3.19 below shows an example of rule one for the two grammars using the gloss “these seven”. The two rules formed part of the congruent Bantu parameterized grammar.

```
1. DetQuant quant num = { s = \\g =>e ! num.n! g ++ num.s ! g;
                          n = num.n ; isPre =True } ;
2. DetQuantOrd quant num ord = {
   s = \\ g =>quant.s ! num.n! g ++ num.s! g ++ ord.s ! g;
   n = num.n ; isPre =True } ;
```

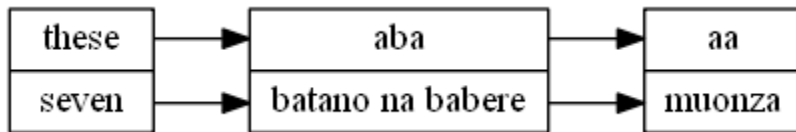


Figure 3.19 Determiner example of rule one

3.4.2.3 Adjective phrase (AP)

The AP linearisation category type was implemented with a string that inflects for gender and number plus a Boolean parameter “isPre” to determine whether AP will come before or after the noun phrase, as shown below

```
lincat
AP = {s : Cgender => Number => Str; isPre : Bool } ;.
```

The simple way to form an adjective phrase is by using a positive degree as implemented below and the rule is similar for both grammars. Thus parts of the congruent Bantu parameterized grammar.

```
PositA a = {s = \\g,n => a.s !AAdj g n ; isPre = True } ;
```

In implementing the production of the comparative adjective, two rules were crafted. The first rule used positive adjectives plus noun phrases. The second one used only positive adjectives because comparative adjectives in Bantu languages are mostly achieved via syntax unless a comparative adjective exists at morphology like in Kikamba and was implemented as a part of the unique grammar. Below is the standard implementation of the

comparative adjective. The string *conjThan* which is different for each language was implemented in the *Diff* module of GF.

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

The exception in Kikamba is exemplified here

```
ComparA a np = {
  s = \\g,n => a.s !AComp g n ++ conjThan ++ np.s ! npNom ;
  isPre = False} ;
UseComparA a = {s = \\g,n=> a.s !AComp g n ;
  isPre = False };
```

The next production deals with valency adjectives (relational adjectives), where three rules were crafted. The first rule has an adjective and noun phrase as arguments; the second one has a reflexive pronoun and the final rule has the relational adjective. The rules are common for the two grammars, thus form part of the congruent Bantu parameterized grammar.

```
ComplA2 a np = {
  s = \\g,n => a.s !AAdj g n ++ a.c2 ++ np.s ! NCase Nom;
  isPre = False } ;
ReflA2 a = {s = \\g,n =>
  a.s !AAdj g n ++ a.c2 ++ reflPron ! Ag g n P3; } ;
UseA2 a2 = {s = \\g, n => a2.s !AAdj g n ;
  isPre = True } ;
```

Lastly, an AP can also be formed using adverbs, sentence complements and noun phrases. Figure 3.20 presents a parse tree of an AP and word alignment graph for comparative adjective with NP as its modifier in Ekegusii and Kikamba, in that order, for English AP “better than some students”.

```
AdAP ada ap = {
  s = \\g,n => ap.s ! g ! n ++ ada.s ;
  isPre = ap.isPre} ;
AdvAP ap adv = {
  s = \\g,n => ap.s ! g ! n ++ adv.s ! Ag g n P3;
```

```

        isPre = False} ;
SentAP ap sc = {
    s = \\g,n => ap.s !g! n ++ sc.s ;
    isPre = False} ;
CAdvAP ad ap np = {
    s = \\g,n => ad.s ++ ap.s !g! n ++ ad.p ++ np.s ! npNom ;
    isPre = False} ;

```

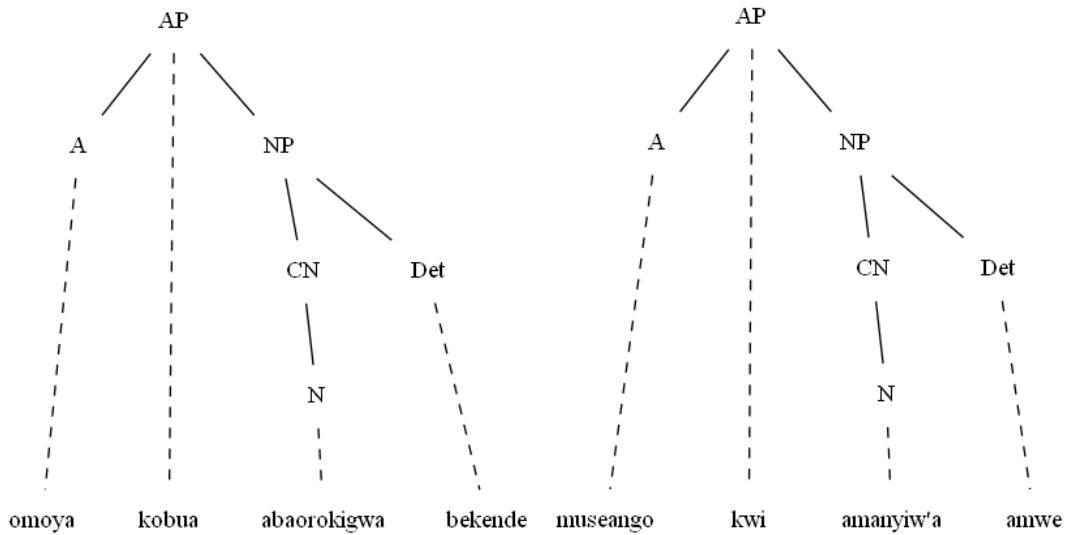


Figure 3.20 Example of AP parse tree and word alignment

3.4.2.4 Noun phrase (NP)

The NP was implemented from the common noun, proper nouns, determiners, pronouns, and recursion of NP with adverbs, pre-determiners, and determiners. NP implementation used two parameters: case and agreement (concord), which are needed when combining the NP with a verb phrase. The case's values were *nom* for nominative case, while *NPoss* was introduced to cater for noun phrases from personal and possessive pronouns. Definition 3.17 below shows how the linearization of NP is implemented and

the definition of the case parameter. The agreement had already been done earlier in section 3.3.2.

Definition 3.17 NP linearization and parameter definition

```
lincat
NP = ResBantu.NounPhrase ;
oper
NounPhrase = {s : Case => Str ; a : Agr; isPron : Bool} ;
param
    Case = Nom | NPoss ;
```

The field *s* is from case to string. Bantu languages are pro-drop languages; that is, when personal pronouns are used as a subject of a sentence with a verb, then the pronoun can be dropped (pro-drop) since it can be inferred from the subject marker morpheme of the verb. Therefore, the field *isPron* stores the information on whether the current NP is a pronoun or not to enable future pro-drop when needed. The primary ways of forming noun phrases were using proper nouns, pronouns, determiners, common nouns, and common nouns combined with determiners or adjectives, as illustrated by Definition 3.18 below.

Definition 3.18 Primary noun phrases rules

```
1.    UsePN pn = {s = \\c => pn.s ;
              a = toAgr pn.g  Sg P3;isPron=False} ;

2.    UsePron pron = let agr = nounAgr pron.a;
                  n=agr.n; g=agr.g in
                  {s = \\c => pron.s!Pers ;
                  a = toAgr agr.g agr.n agr.p;
                  isPron=True } ;

3.    DetNP det = { s = \\c => det.s!G1 ;
                  a = AgP3  G1 det.n ;
                  isPron=False} ;

4.    MassNP cn = let g = cn.g ; n = Sg | Pl in {
                  s = \\c => cn.s ! n;
                  a = AgP3 g n ;
                  isPron=False } ;

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

```

a =toAgr cn.g det.n P3 ;
isPron=False } ;

```

Example 3.9 implement rule five in Definition 3.18 above. The determiner formed from demonstrative “these” which is a post-modifier of the CN and a determiner “some” formed from quantifier which is a pre-modifier are used. The former has Boolean value TRUE for *isPre* string while the latter has FALSE. The string values are applied correctly in the output and there is no overgeneration.

Example 3.9 Pre and post determiner

```

Lang> p -lang=Eng "these horns are black and some horns are black" | 1
these horns are black and some horns are black
mbvya ii ni nzu na imwe mbvya ni nzu

```

Complex noun phrases are formed using a pre or post-modifier of the NP. The pre-modifiers are pre-determiner and determiner, while post-modifiers are past participle verbs, relative clauses, and adverbs that are implemented in productions one to five in Definition 3.19 below respectively. The inflection and all the productions of NP formed part of congruent Bantu parameterized grammar.

Definition 3.19 Complex noun phrase rules

1. **PredetNP** pred np = **let** agr = nounAgr np.a **in** {
s = \\c => np.s ! Nom ++ pred.s ! agr.g ;
a =AgP3 agr.g agr.n ;
isPron=np.isPron } ;
2. **PartNP** cn np = {
s = \\n => cn.s ! n ++ part_Prep.s! n!cn.g ++ np.s ! Nom ;
s2 =\\n => []; g = cn.g} ;
3. **CountNP** det np = **let** g = (predetAgr np.a).g **in** {
s = \\c => det.s!g ++ part_Prep.s!det.n!g ++ np.s!c ;
a = AgP3 g det.n ;
isPron=np.isPron } ;
4. **RelNP** np rs = {
s = \\c => np.s ! Nom ++ frontComma ++ rs.s ! np.a ;
a = np.a;
isPron=np.isPron } ;
5. **AdvNP** np adv = **let** agr = nounAgr np.a **in**{
s = \\c => np.s ! Nom ++ adv.s !AgP3 agr.g agr.n ;

```
a = np.a;
isPron=np.isPron } ;
```

Figure 3.21 below shows word alignment for NP “all my three red eyes” in English, Ekegusii and Kikamba. Beneath, Figure 3.21 is an example of parsing the same sentence to English from both languages. The NP consists of a pre-determiner, possessive pronoun, cardinal numeral, adjective and a noun in English and the example uses rules five and one in Definition 3.18 and 3.19 respectively. The parse trees of the same are shown in Figure 3.22 below and demonstrate how a determiner is added between an adjective and a noun, as explained in the CN section.

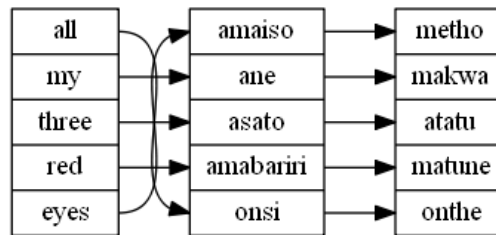


Figure 3.21 Noun phrase word alignment

```
Lang> p -cat=NP -lang=Gus " amaiso ane asato amabariri onsi"| pt -
number=1 | 1
all my three red eyes
amaiso ane asato amabariri onsi

Lang> p -cat=NP -lang=Kam " metho makwa atatu matune onthe"| pt -
number=1 | 1
all my three red eyes
metho makwa atatu matune onthe
```

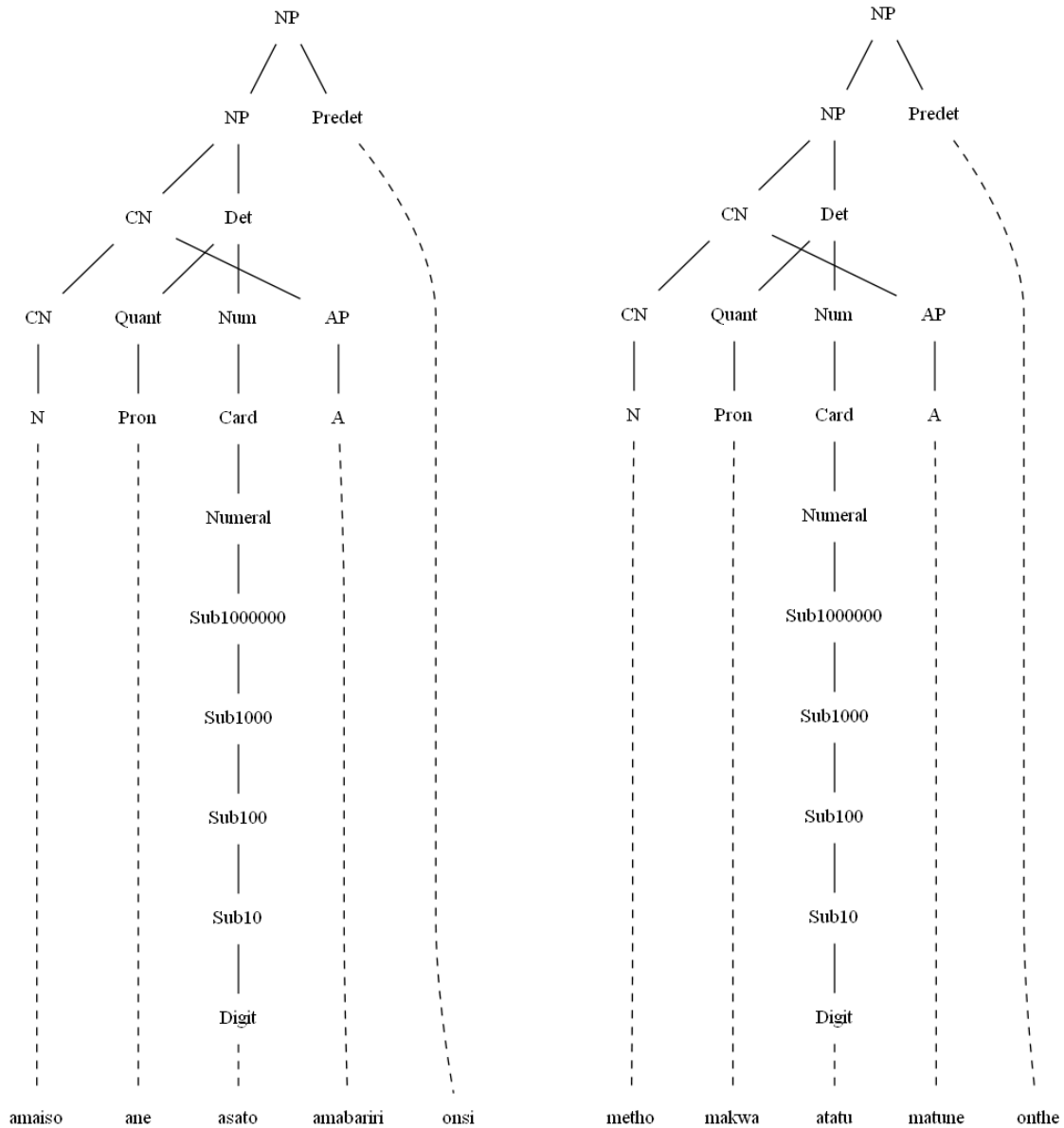


Figure 3.22 NP parse tree in Ekegusii and Kikamba respectively

3.4.2.5 Verb phrase (VP)

The Verb phrase implementation mirrored the structure of verb implementation. Therefore, its linearization is the same as that of the verb and the VP morphology paradigm *regVP* uses similar strings as shown in Definition 3.20 below. The strings are *s* for the ordinary verb, *progV* for progressive verbs, *compl* for the verb's object, *imp* for imperative verbs and *inf* for infinitive verbs. The sub-categorization of verbs was taken care of through

compl (one-place, two-place, and three-place verbs). The implementation was common for the two grammars, thus forming a congruent Bantu parameterized grammar segment.

Definition 3.20 Verb linearization and paradigms

```

lincat
VP = ResBantu.VerbPhrase ;
oper
  VerbPhrase : Type = {
    s: Agr => Polarity => Tense => Anteriority => Str;
    compl : Agr => Str;
    progV:Str;
    imp : Polarity => ImpForm => Str;
    inf: Str};
regVP run = {
  s =\\ ag,pol,tes,ant =>run.s!pol!tes!ant!ag;
  compl=\\_=> [];
  progV = run.progV;
  imp=\\po,imf => run.imp!po!imf;
  inf= run.s!VInf };

```

The object of a sentence was modeled using the VP complements that are listed below.

- Use of the verb or the verb phrase.
- Use of the verb *to be* and its complements (auxiliary verbs)
- Use of adverbs complements
- Verb passivization
- Reflexive complement.

In the first scenario, the verb or verb phrase was constructed using the smart paradigm *regVP* in all subcategorization categories (transitive, intransitive and ditransitive). The complements were noun phrases, sentence complements and adjective phrase complements, as exemplified in definition 3.21 below. The parse trees in Figure 3.23 and 3.24 demonstrate rules *SlashV2a* and *ComplSlash* (for the NP complement) for the sentence “I read the best book” in Kikamba and Ekegusii respectively.

Definition 3.21 Verbs complements productions

```
UseV verb = regVP verb ;

ComplVV vv vp = { s = \\ag , pol,tes,ant =>
(polanttense.s!Pos!tes!Simul! ag).p1 ++ cbind ++vv.s!VGen ++ vp.inf ;
    compl=\\a=> vp.compl!a ;
    imp = \\po,imf => vp.imp!po!imf;
    progV=vp.progV;
    inf= vp.inf};

SlashV2a v = mkVPSlash v.c2 (regVP v)** {n3 = \\_ => [] ;c2 = v.c2 } ;

Slash2V3 v np = insertObjc ( \\agr=>
    v.c2.s!(nounAgr agr).n !(nounAgr agr).g ++ np.s ! Nom )
    (regVP v ** {c2 = v.c3 ;isFused = False}) ;

Slash3V3 v np = insertObjc ( \\agr =>
    v.c3.s!(nounAgr agr).n !(nounAgr agr).g ++ np.s ! Nom)
    (regVP2 v) ;

ComplSlash vp np = insertObj ( \\a =>
    vp.c2.s! (nounAgr a).n! (nounAgr a).g ++ np.s! Nom ) vp;

ComplVA v ap = {
    s=\\agr,pol,tense,anter=>(polanttense.s!pol!tense!anter! agr).p1
    ++ v.s!form ++ ap.s! (nounAgr agr).g ! (nounAgr agr).n ;
    compl=\\a => [];
    imp = \\po,imf => "" ;progV=v.progV; inf= v.s!VInf};

ComplVS v s = insertObj ( \\_ =>s.s) (regVP v) ;
```

Below is a demonstration of parsing and linearization of the same sentence from kikamba to Gusii language for the gloss “I read the best book”.

```
Lang> p -lang=Kam "ninisomaa ivuku iseo vyu" |pt -number=1 | l -
lang=Gus
ngosoma egetabu egekeene mono
```

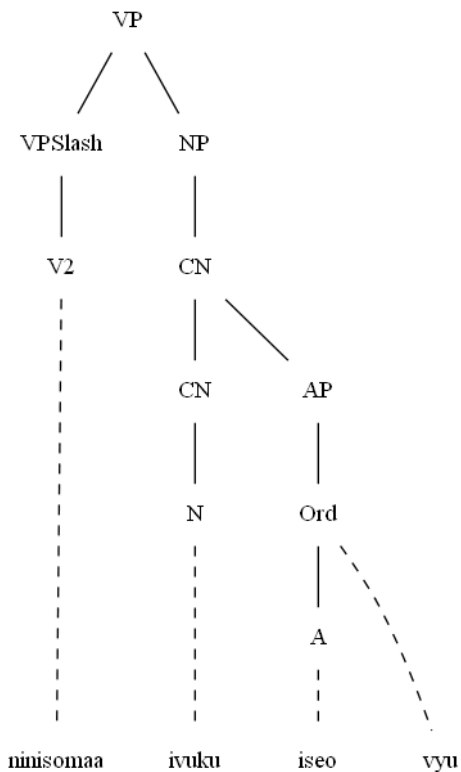


Figure 3.23 Kikamba VP parse tree

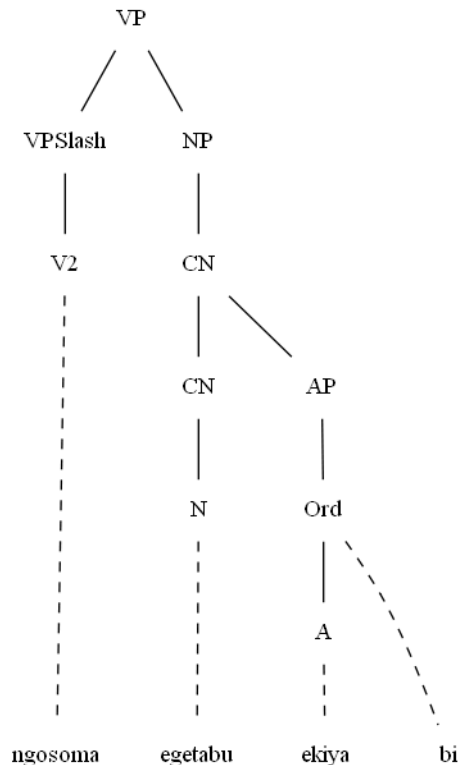


Figure 3.24 Egekusii VP parse tree

The verb *to be* was modeled using the paradigm *auxbe* in all the languages, while the complements are the adjective phrase, noun phrase, adverb, and common noun. The following productions demonstrate how they were implemented:

```

CompAdv adv = {s= adv.s};
UseComp comp = { s = \\agr , pol , tense , anter =>
    auxBe.s!agr !pol!tense!anter ++ comp.s!agr;
    compl=\\_=> [] ;progV=[]; imp =\\po,imf => "" ;inf= ""};
CompAP ap = {s=\\agr =>ap.s! (nounAgr agr).g ! (nounAgr agr).n} ;
CompNP np = {s = \\_ => np.s ! Nom } ;
CompCN cn = {s = \\a => case (nounAgr a).n of { n => cn.s ! n
    ++cn.s2!n }};
UseCopula = auxBe ;

```

The adverbs acted as complements by modifying the verb phrase as per the rules below:

```

AdvVP adv vp = insertAdv adv.s vp ;
AdvVP vp adv = insertObj (\\agr => adv.s!agr) vp;
AdvVPSlash vp adv = insertObj (\\agr => adv.s!agr) vp ** {c2 = vp.c2} ;
AdvVPSlash adv vp = insertAdv adv.s vp ** {c2 = vp.c2} ;

```

The three productions below represent verb phrases made from a preposition modifying a verb phrase, passivization of a verb, and reflexive pronoun use to make VP respectively. Figure 3.25 below shows the parse tree of a clause made using the following VP rules: *useV* and *AdvVP* in Kikamba with the gloss “the priest lives in the church.”

```
VPSlashPrep vp p = vp ** {c2 = p ; isFused=p.isFused };
PassV2 v = {s=\agr,pol,tense,anter=> (polanttense.s!pol!tense!anter!
agr).pl ++cbind ++ v.s!VExtension EPassive ;
compl=\a => [];
imp =\po,imf => ""; progV=v.progV;inf= v.s!VInf};
ReflVP vp slash = insertObjPre (\agr => vp slash.compl !agr ++
reflPron !Ag (nounAgr agr).g (nounAgr agr).n (nounAgr agr).p)
vp slash ;
```

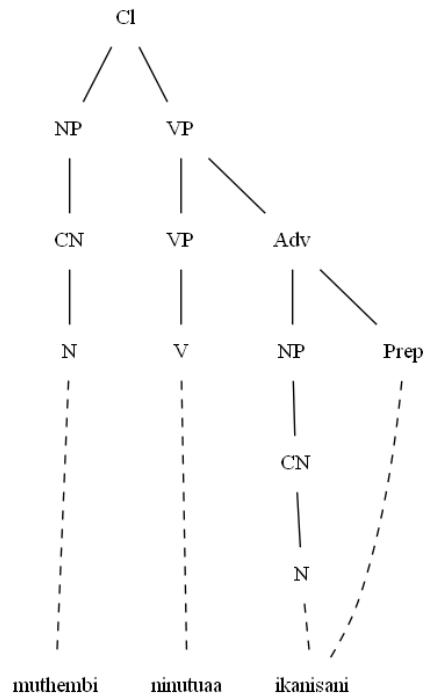


Figure 3.25 VP parse tree

Below is the parsing of the same sentence from Kikamba to Ekegusii language for the gloss “the priest lives in the church.”

```
p -lang=Kam "muthembi ninutuaa ikanisani" | pt -number=1 | l -lang=Gus
omosasiroti akomenya ase ekanisa
```

Below is a demonstration of imperative mood using the verb “sleep”. The first case is a plural command, the second a singular polite request and the last one is a plural command.

```
Lang> l UttImpPl PPos (ImpVP (UseV sleep_V))
rara
komai

Lang> l UttImpPol PPos (ImpVP (UseV sleep_V))
rara
koma

Lang> l UttImpSg PPos (ImpVP (UseV sleep_V))
rara
koma
```

To exemplify subjunctive mood which is an expression of permission or probability of an event., Let's use the sentence "or let me run please" in the Kikamba language. By parsing and linearizing in the grammar the resultant output is shown below

```
Lang> p -lang=Eng "or let me run please" | 1
kana eka nyie nisebe ame
```

Generally, any basic declarative statement uses indicative mood which is the basic mood. Figures 3.23 to 3.25 illustrate this mood in the verbs used. In summary, all the rules of the VP were shared in Bantu parameterized grammar.

3.4.2.6 Clauses

In GF, there are three types of clauses: declarative, question and relative clauses. They are constructed using categories: clause, question and relative respectively. In addition, implementation is done in sentences, questions and relative modules respectively. All clauses have undetermined polarity, tense and anteriority, which is fixed at the sentence level. The question clause uses the parameter *QForm* with values *QDir* and *QIndir* for direct and indirect questions. This parameter and the clauses linearization (inflection) were shared and thus part of the congruent Bantu parameterized grammar as shown in Definition 3.22 below:

Definition 3.22 clause linearization

```

lincat
Cl= {s : Polarity => Tense => Anteriority => Str};
QCl = QClause ;
RCl = {s : Polarity => Tense => Anteriority => Str};
oper
QClause = {s : Polarity => Tense => Anteriority => QForm => Str} ;
param
    QForm = QDir | QIndir ;

```

The Bantu languages’ direct question clause is formed by changing the declarative clause’s tone to high or using a question mark. Declarative clauses are formed using the SVO topology where the *S* is a noun phrase, while *V* is the *s* field of the Verb phrase and

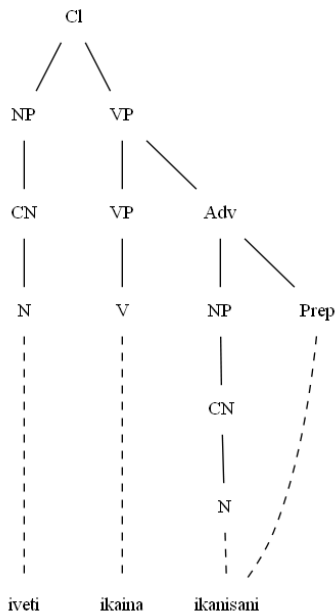


Figure 3.26 SVO example

O is the compl field of the Verb Phrase. Figure 3.26 above illustrates this topology using the sentence “iveti ikaina ikanisani” the gloss “women will sing in church” where the *S* is an NP “iveti”, *V* is the VP “ikaina” and *O* is the adverb complement that consists of noun “ikanisa” infused with a preposition “ni.” Four productions were engineered, shown in Definition 3.23 below. The parse tree in Figure 3.27 below demonstrates the production rule four *PredVP* using the clause “oronsana robwate emete ya eragi ya machani emenene Na chinyoni chigotera ororo” in Ekegusii and the gloss “The forest has big green trees and birds sing there”. The parse tree shows the combination of NP and VP makes

both clauses. The VP in the first clause is made up of the two-place verb with an NP as an object, while the second clause is made up of a one-place verb and adverb as the object.

Definition 3.23 Clause definition

```
PredVP np vp = let agr = nounAgr 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 case np.isPron of
        { True => verb ++ obj ;
          False=> np.s!npNom ++ verb ++ obj }} ;

PredSCVP sc vp= { s=\pol,tense,anter =>
    sc.s ++ vp.s!Ag G1 Sg P3 !pol!tense!anter };
SlashPrep cl prep = cl ** {c2 = prep.s} ;
SlashVP np vp = { s=\pol,tense,anter =>
    np.s!npNom ++ vp.s!np.a !pol!tense!anter};
```

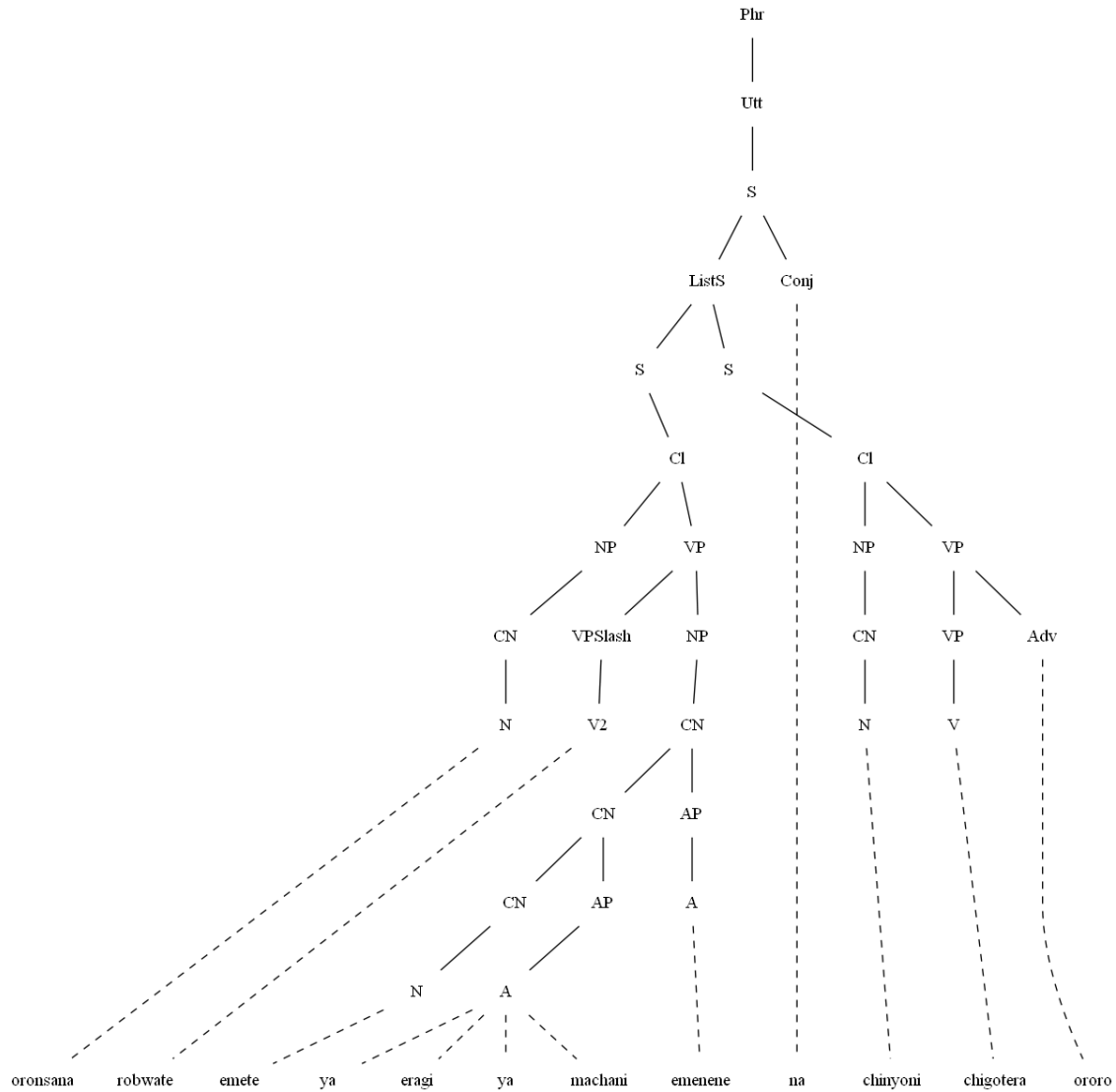


Figure 3.27 Clause/Sentence parse tree

The two ways used to form question clauses (*QCl*) are either through the yes or no answer questions or through interrogatives. The interrogatives are: interrogative Pronouns (*IP*), interrogative Adverbs (*IAdv*), Interrogative Quantifiers (*IDet*), copula interrogative complement (*IComp*) and their modifiers. All the production rules used to form question clauses and their sub-units are presented in Definition 3.24. Figure 3.28 below illustrates a direct question in Kikamba “amanyiw'a makathauka sukuluni” gloss “will the students play in school” using the production *QuestCl*. An example of an Interrogative question

is shown in Figure 3.29 below using the question “amanyiw'a makathauka sukuluni indii” gloss “when will the students play in school.” and is based on the production *QuestIAdv*.

Definition 3.24 Question clause

```

CompIAdv a = a ;
CompIP ip = {s = ip.s } ;
IdetQuant idet num = let n = num.n in {
    s = \\g => idet.s!n ! g ++ num.s !g ; n = n } ;
IdetCN idet cn = {s = cn.s ! idet.n ++idet.s ! cn.g ;
    n = idet.n } ;
PrepIP p ip = { s = p.s!ip.n !G1 ++ ip.s } ;
AdvIP ip adv = { s = ip.s ++ adv.s! Ag G1 ip.n P3 ; n= ip.n } ;
QuestCl cl = { s = \\t,a,p => let cls = cl.s ! t ! a ! p
    in table {
        QDir => dQue ++cls ;
        QIndir => inQue ++ cls } } ;
QuestIAdv iadv cl = mkQuestion iadv cl ;
QuestVP qp vp = { s = \\t,a,b,_ =>
    qp .s ++ vp.s!Ag G1 qp.n P3!t!a!b};
QuestSlash ip slash = { s = \\t,a,b,_ => ip .s ++ slash.s!t!a!b};

```

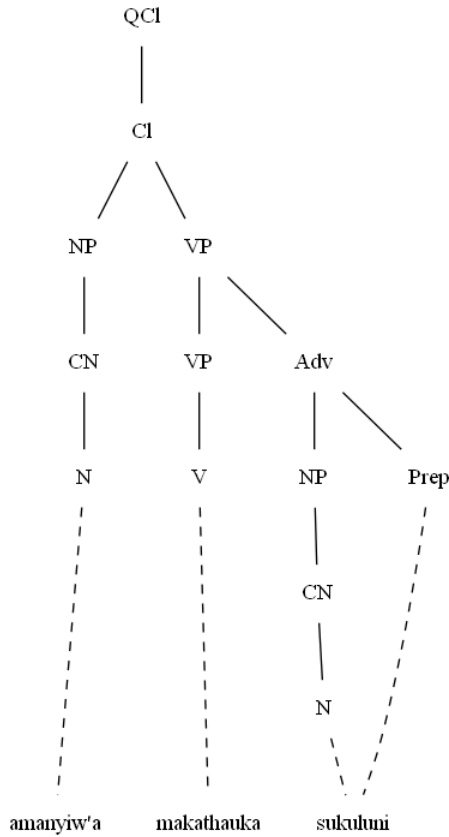


Figure 3.28 Direct question

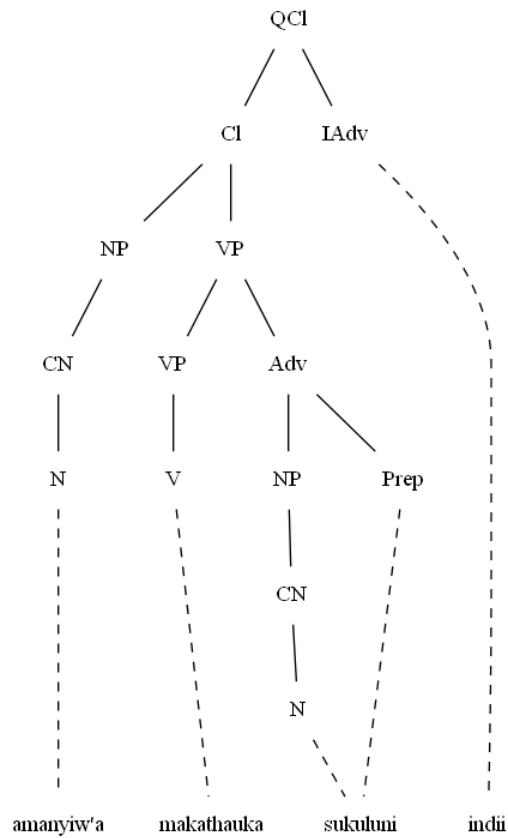


Figure 3.29 Interrogative question

The relative clause (RCI) is formed in three ways. The basic way is to use a clause. The second and third involve verb phrases and a sentence that lacks a noun phrase being modified by a relative pronoun (RP). The three ways are exemplified by rules one to three respectively. The RP is either formed from identity RP or modified by a preposition and a noun phrase shown by rules four and five respectively.

```

1. RelCl c1 = { s = \ p,t,a => such ++ that ++ cl.s! p ! t ! a };
2. RelVP rp vp = let agr = nounAgr rp.a in
   {s=\p,t,a => rp.s!agr.g!agr.n ++ vp.s!Ag agr.g agr.n agr.p
   !p!t!a ++ vp.compl!Ag agr.g agr.n agr.p};
3. RelSlash rp slash = let agr = nounAgr rp.a in
   {s=\p,t,a => rp.s!agr.g!agr.n ++ slash.s!p!t!a };
4. FunRP p np rp =let agr = nounAgr np.a in
   {s = \g,n => np.s !Nom ++ p.s!n!g ++ rp.s ! g ! n;
   a. a= np.a } ;
5. IdRP = { s = \g,n => which_IQuant.s!n!g; a=Ag G1 Sg P3};

```

The productions for declarative, relative and question clauses are shared among the two languages thus form part of the congruent Bantu parameterized grammar.

3.4.2.7 Sentences, Phrases, and Utterance

The primary way of forming a sentence is to fix the tense, anteriority and polarity to question, declarative and relative clauses as exemplified by productions one to three. Other ways include the use of embedded sentences such as question sentences and infinitive verb phrases. An adverb can modify a sentence with a comma or not. Finally, sentences can be constructed using the subjunctive, relative clause and imperative verbs. All the productions shown in Definition 3.25 formed part of the congruent grammar.

Definition 3.25 Sentences productions

```

1. UseCl temp pol c1 = {
   s = temp.s ++ pol.s ++ cl.s !pol.p ! temp.t ! temp.a } ;
2. UseRC1 t pol c1 = {
   a. s = \ag => t.s ++ pol.s ++ cl.s !pol.p ! t.t ! t.a } ;
3. UseQC1 t p c1 = { s = \q => t.s ++ p.s ++ cl.s!p.p ! t.t !
   t.a!q } ;   SlashPrep c1 prep = c1 ** {c2 = prep.s} ;
4. SSubjS a s b = {s = a.s ++ frontComma ++ s.s ++ b.s} ;
5. AdvS a s = {s = a.s!AgP3 G1 Sg ++ s.s} ;
6. EmbedQS qs = {s = qs.s ! QIndir};
7. RelS s r = {s = s.s ++ frontComma ++ r.s!AgP3 G1 Sg } ;

```

```

8. SlashVP np vp = { s=\pol,tense,anter =>np.s!npNom ++ vp.s!np.a
!pol!tense!anter};
9. ExtAdvS a s = {s = a.s!AgP3 G1 Sg ++ frontComma ++ s.s} ;
10. EmbedVP vp = { s=vp.inf};
11. ImpVP vp = { s = \pol,iform => vp.imp!pol! ImpF (getNum iform)
(getbool iform) ++ vp.compl!AgP2 (getNum iform) };

```

The primary utterance was designed from sentences, questions and imperatives in the phrase module of RGL. The imperative utterance in the Bantu languages is in both singular and plural, unlike English. GF provides production for a singular number. Thus, the reason for creating a new rule in the extra module for plural polite requests. The abstract syntax and concrete syntax were as shown below:

```

UttImpPolpl : Pol -> Imp -> Utt ; --abstract syntax
UttImpPolpl pol imp = {s = pol.s ++ imp.s ! pol.p ! ImpF Pl True };

```

The main production rules for constructing utterances are shown in Definition 3.26 below:

Definition 3.26 Utterance

```

UttS s = {s = s.s} ;
UttImpSg pol imp = {s = pol.s ++ imp.s ! pol.p ! ImpF Sg False} ;
UttImpPl pol imp = {s = pol.s ++ imp.s ! pol.p ! ImpF Pl False} ;
UttImpPol pol imp = {s = pol.s ++ imp.s ! pol.p ! ImpF Sg True} ;
UttQS qs = {s = qs.s ! QDir} ;

```

An utterance can also be formed using one word, especially where it is an answer to a question in the following categories: noun phrases, interrogative adverb, interrogative pronouns, common nouns, numerals, verb phrases, adjective phrase, adverbs, and interjections, as shown in Definition 3.27 below with the productions following the order mentioned above. Production rules four and eight are demonstrated in Figure 3.30 below with *a* and *b* in Kikamba with the gloss “why” and “alas” respectively, while *c* and *d* demonstrate productions five and eleven in Ekegusii with the gloss “who” and “to sleep” respectively.

Definition 3.27 More Utterance productions

```

1. UttNP np = {s = np.s !Nom} ;
2. NoPConj = {s = []} ;
3. NoVoc = {s = []} ;

```

4. `UttIAdv iadv = iadv ;`
5. `UttIP ip = {s = ip.s } ;`
6. `UttAP ap = {s = ap.s !G1 !Sg} ;`
7. `UttAdv adv = {s= adv.s!AgP3 G1 Sg };`
8. `UttInterj i = i ;`
9. `UttCN n = {s = n.s ! Sg !Nom};`
10. `UttCard n = {s = n.s ! G1} ;`
11. `UttVP vp = {s = vp.inf};`

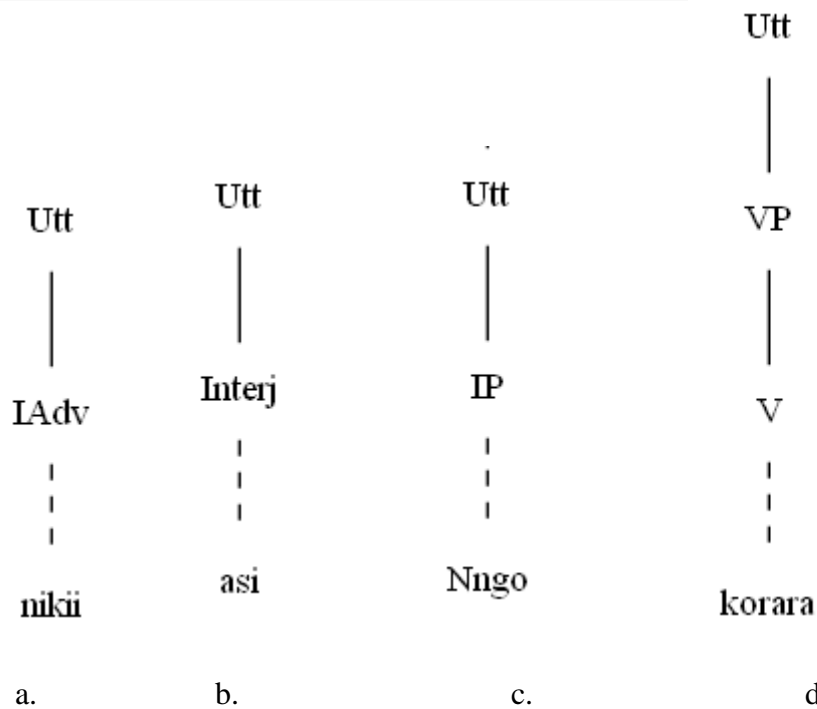


Figure 3.30 Utterance Examples

The phrase is the start category of the Bantu parameterized grammar and is formed by prefixing and suffixing utterances with a phrasal category and a noun phrase as a vocative (Voc) respectively (Ranta 2011). The phrase production rules: prefixing plus suffixing conjunction and suffixing vocative are shown in Definition 3.28 below:

Definition 3.28 Phrasal productions

1. `PhrUtt pconj utt voc = {s = pconj.s ++ utt.s ++ voc.s} ;`
2. `PConjConj conj = {s = conj.s2} ;`
3. `VocNP np = {s = "," ++ np.s !Nom} ;`

All the productions for sentences, phrases plus utterance and their linearization formed part of the congruent Bantu parameterized grammar.

3.4.2.8 Coordination

Coordination productions were implemented in the *conjunction module* of RGL for the following categories: sentences, adverbs, interrogative adverbs, noun phrases, adjectives, relative sentences, common nouns and determiner phrases in that order, as shown in Definition 3.29 below:

Definition 3.29 Conjunction productions

```
ConjS = conjunctDistrSS ;
ConjAdv = conjunctDistrTable Agr ;
ConjAdv = conjunctDistrSS ;
ConjNP conj ss = conjunctDistrTable NPCase conj ss ** {
  a = Ag (nounAgr ss.a).g (conjNumber (nounAgr ss.a).n conj.n)
  (nounAgr ss.a).p ;
  isPron = andB ss.isPron ss.isPron} ;
ConjAP conj ss = conjunctDistrTable2 Cgender Number conj ss;
ConjRS conj ss = conjunctDistrTable Agr conj ss ** { c = ss.c } ;

ConjIAdv = conjunctDistrSS ;
ConjCN conj cn = { s = \\num,c => conj.s1 ++ cn.n1.s ! num!c ++ conj.s2
  ++ cn.n2.s ! num!c ;
  g = conjGender cn.n1.g cn.n2.g ;
  s2 = \\num => [] ; } ;
ConjDet c xs = {s = \\ Cgender => xs.s1! Cgender ++
  c.s2 ++ xs.s2! Cgender ;
  n = xs.n; isPre=xs.isPre};
```

Figure 3.27 shows an example of the coordination of two sentences using the conjunction “and.”

3.4.2.9 Adverbs

Comparative Adverbs are formed using a noun phrase or sentence as a comparative object in a phrase with an adverb and adjective. Productions 1 and 2 illustrate the above-mentioned scenarios. The two rules were the default in the congruent Bantu parameterized grammar. However, the Kikamba rules were different because the language has a way of forming comparative adjectives, as discussed in section 3.3. Numeral adverbs can be formed from comparative adverbs, while the subordinate can also act as an adverb. All the adverbs productions are shown in definition 3.30 below:

Definition 3.30 Adverbs productions

```
1. ComparAdvAdj cadv a np = let ag = complAgr np.a in { s=\agr =>
  a.s !AAdj ag.g ag.n ++ cadv.s ++ np.s ! npNom};
2. ComparAdvAdjS cadv a s = { s = table{
  AgP1 n => a.s! AAdj G1 n ++ cadv.s ++ s.s;
  AgP2 n => a.s! AAdj G1 n ++ cadv.s ++ s.s;
  AgP3 g n => a.s! AAdj g n ++ cadv.s ++ s.s}} ;
3. PrepNP prep np = let ag = complAgr np.a in {s=\agr =>case
  prep.isFused of {
    True =>(np.s ! Nom ++ cBind ++ prep.s1);
    False => prep.s!ag.n!ag.g ++ (np.s ! Nom) } };
4. AdAdv sub se = { s=\agr => se.s!agr ++ sub.s } ;
5. SubjS sub se = { s=\agr => sub.s ++ se.s } ;
6. AdnCAdv cadv = {s = cadv.p ++ cadv.s } ;
```

3.4.3 Evaluation Test Suite

A development suite was used to test the correctness and accuracy of the Bantu parameterized grammar during the development process. At this point, the testing process employed the GF regression process, as summarized in Figure 2.7. The development suite consisted of the comment for each function as stated in the Abstract syntax of GF. A sample of these is provided in Table 3.12.

Most of the GF resource Library grammars are evaluated only via the GF regression testing process. Khagai (2004) goes a step further by evaluating Russian grammar using a 27 sentences treebank whose English linearizations were provided. To evaluate the Bantu parameterized grammar, a test suite was developed by a grammar writer plus the existing GF treebank based on Bröker (2000) and Butt (2003) ways of developing a test suite. The test suite development involved the following: first, a Bantu linguist generated eighty-five sentences in English using the English 500 lexemes (open and closed categories lexicons) that are provided in GF and extra fifteen sentences were drawn from GF online treebanks²⁸ and Khagai (2004) Russian treebank making the 100 sentences test suite. The online treebanks are constructed from GF RGL, such as universal dependencies documentation.

²⁸ <https://github.com/GrammaticalFramework/gf-rgl/tree/master/treebanks>

In this research, Khagai (2004) test suite was tripled thereby resulting in a 100 sentences test suite. The two approaches were used since an already existing evaluation dataset in the same environment would help simulate similar performance with already existing grammar in GF, while a Bantu linguist was used since there is no existing Bantu grammar test-suite in GF. The test suite in English is available in Appendix C, c.1.

Table 3.12 Sample of the development suite (source GF abstract Modules)

No	Categories	Functions	Development suite
1	Adjective Phrase	<i>PositA : A -> AP ;</i>	warm
		<i>UseComparA : A -> AP ;</i>	warmer
2	Adverb	<i>PrepNP : Prep -> NP -> Adv ;</i>	in the house
3	Conjunctions	<i>ConjS : Conj -> ListS -> S ;</i>	he walks and she runs
4	Noun Phrases	<i>UsePN : PN -> NP ;</i>	john
		<i>DetCN : Det -> CN -> NP ;</i>	The man
		<i>UsePron : Pron -> NP ;</i>	He
5	Determiner	<i>DetQuant : Quant -> Num -> Det ;</i>	These five
6	Numeral	<i>IDig : Dig -> Digits ;</i>	8
		<i>NumCard : Card -> Num ;</i>	five
7	Common Noun	<i>UseN : N -> CN ;</i>	house
		<i>AdjCN : AP -> CN -> CN ;</i>	big house
8	Utterance	<i>UttImpPol : Pol -> Imp -> Utt ;</i>	sleep
		<i>UttS : S -> Utt ;</i>	John walks
9	Interrogative pronouns	<i>IdetIP : IDet -> IP ;</i>	Which five
10	Relative Clause	<i>IdRP : RP ;</i>	who
11	Sentence	<i>PredVP : NP -> VP -> Cl ;</i>	John walks
12	Verb Phrase	<i>UseV : V -> VP ;</i>	sleep
		<i>AdvVP : VP -> Adv -> VP ;</i>	Sleep here
		<i>SlashV2a : V2 -> VPSlash ;</i>	Love (it)
		<i>Slash2V3 : V3 -> NP -> VPSlash ;</i>	Give it (to her)
13	Question	<i>QuestCl : Cl -> QCl ;</i>	does John walk

The treebank was created by parsing the 100 English sentences that had a total of 2854 functions, with the largest tree having 62 functions while the shortest had 11 functions. Most syntax functions came from the noun and verb modules that are in line with the research scope. The largest tree consisted of complex noun phrases and complex verb phrases and had two sentences joined together by coordination. This implies that grammar can handle complex utterances. Table 3.13 below summarizes the function distribution in the whole treebank.

Table 3.13 Treebank syntax functions Distribution

Module	Frequency	Productions
Adjective	15	AdAP, AdjOrd, ComparA, UseA2, UseComparA
Adverb	37	PrepNP, SubjS
Coordination	32	BaseAdv , BaseAP ,BaseNP ,BaseS, ConjAdv, ConjAP, ConjNP, ConjS ,ConsAdv
Idiom	5	ExistNP , ProgrVP
Noun	819	AdjCN , AdvCN , AdvNP, CountNP, DefArt, DetCN, DetNP, DetQuant, DetQuantOrd, IndefArt, MassNP, NumCard, NumNumeral, NumPl, NumSg, OrdNumeral, PartNP, PPartNP, PredetNP, SentCN, UseN, UseN2, UsePN, UsePron
Number	168	n2, n3, n4, n5, n6, n7, n9, num, pot0, pot01, pot0as1, pot1, pot110, pot1as2, pot1to19, pot2, pot2as3, pot3
Phrase	399	NoVoc, NoPConj, PhrUtt, UttAdv, UttIP, UttNP, UttS
Question	4	IdetCN, QuestVP
Sentence	215	EmbedQS, PredVP, UseCl, UseQCl, AdvS
Verb	515	PNeg, AdvVP, CompAdv, CompAP, CompCN, ComplSlash, CompNP, PassV2, PPos, SlashV2a, TFut, TPast, TPres, TTAnt, UseComp, UseCopula, UseV, VPSlashPrep

The test suite has the coverage shown in Table 3.14 implying it covers almost all the categories that the grammar has. The development and test suites were quite different, for example in testing conjunctions, Definition 3.31 shows the development suite uses a simple sentence while the test suite uses a complex sentence.

Definition 3.31 Example of difference in the suites

```

he walks and she runs -- Development suite

the teacher wrote seven books and the second book was written through
somebody -- test suite

```


Table 3.14 Test suite coverage

	Coverage
Sentence	Declarative
Tense	Present, Future, Past and Conditional
Anteriority	Positive and Negative
Verb	One-Place, Two-Place, VP, auxiliary verbs
Determiners	Quantifiers, Numbers and Possessive Pronouns
Noun	One Place Two-Place, Three Place Complex Noun
Adjective	Positive and Comparative
Noun Phrase	Personal Pronoun and NP Phrase
Numeral	Cardinal and ordinal
Mood	Indicative, Subjunctive
Others	Prepositional and Conjugation

Three Bantu experts translated the test dataset into the two Bantu languages to act as the gold standard (one for each language). The human translation was subjected again to a different set of Bantu experts to confirm the translation's correctness. These human translations are available in Appendix C: c.3, c.4 for Kikamba and Ekegusii respectively. The English dataset was transformed into abstract syntax trees through parsing (strings to abstract trees) and linearized (abstract trees to strings) to Kikamba and Ekegusii with the machine translation outputs forming the candidate or target language translations. The machine translations, human translation (gold standards) and the source language (English) were in text files. The gold standard and machine translation sets were compared using the online Tilde²⁹ software to extract the BLEU score, while WER and PER metrics were extracted using Perl scripts.

3.5 Bootstrapping Swahili Grammar Development

Developing the rule-base of monolingual computational grammar requires much effort, especially if it is to be developed from scratch. This effort is a stumbling block in grammar development more so for under-resourced languages. Therefore, an experiment was set up to evaluate how the Bantu parameterized grammar's shareability and portability can reduce the effort of bootstrapping a Swahili grammar, thereby achieving the third

²⁹ <https://www.letsmt.eu/Bleu.aspx>

objective of this research. The Bantu parameterized grammar was the leverage seed, as shown by the bootstrapping structure in Figure 3.31.

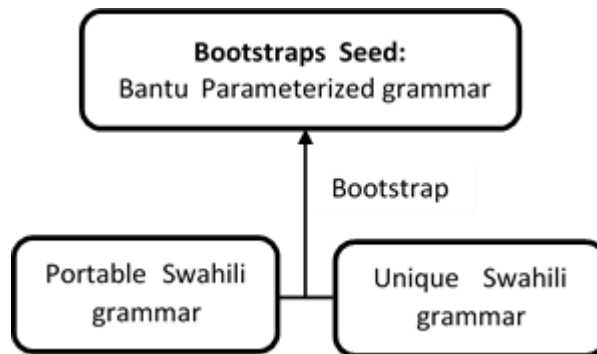


Figure 3.31 Bootstrap structure

The quasi-experiment involved defining and modifying the unique grammar and portable grammar segments, respectively. Thereafter, it was bootstrapped to the Bantu parameterized grammar and then the GF regression testing procedure was applied. If errors resulted from the process, the functions and rules were refined iteratively until the errors were resolved. However, if the errors were from the congruent grammar, the functions and/or rules were moved to portable or unique grammar depending on similarities and the testing procedure is repeated until errors were eliminated. The experiment steps summarized in Figure 3.32 below followed the GF morphology-driven approach, where the lexicon and linearization were defined first, then the regular expressions for the inflection of the different categories and finally, the syntax production rules.

Swahili language has good descriptive grammar books due to extensive years of grammar research and is widely known to many people compared to the other two languages chosen. These aspects availed a pool of different people who examined the output and validated the computational grammar.

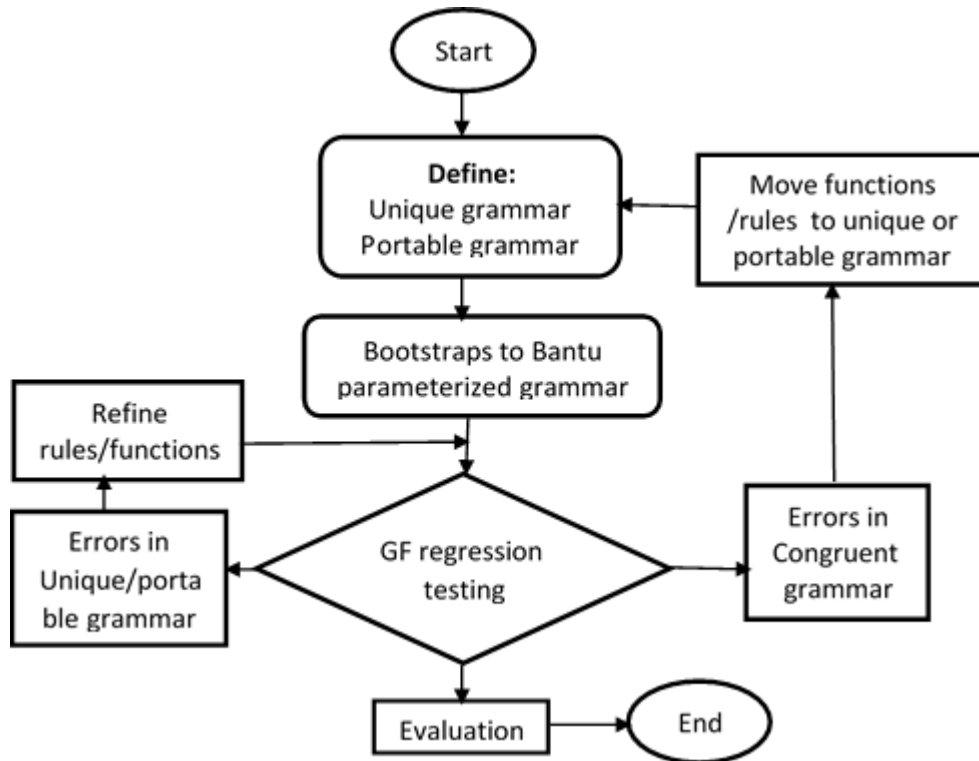


Figure 3.32 Bootstrap experiment

3.5.1 Morphology.

Though the Swahili lexicons are different from the other languages, their definition structures are similar to the definition in Example 3.2 for Ekegusii and Kikamba. Therefore, the lexicons definition followed similar structures that is the regular expression (paradigm), then the lemma and any parameter the category has. For example, the definition of “bread” below: *regN* is the paradigm for the regular noun, while “mkate” is a lexicon for bread in Swahili and finally, *u_i* is the parameter gender where “bread” belongs.

```
bread_N=regN " mkate" u_i;
```

In this case, the structure was adapted for all lexicon definitions. On inflection, all categories shared the same linearization and inflection as the congruent Bantu parameterized grammar. The gender parameter control concord in Swahili, like the other two languages hence the parameter's definition, shared the same structure and naming code, forming part of the portable grammar. However, the actual gender morphemes are unique

to Swahili and are illustrated in Table 3.15 below. Furthermore, Definition 3.32 exemplifies their GF definition.

Table 3.15 Swahili Gender coding

GF coding	Swahili
G1	a_wa
G2	u_i
G3	li_ya
G4	ki_vi
G5	i_zi
G6	u_zi
G7	u_u
G8	u_ya
G9	ya_ya
G10	i_i
G11	ku_ku
G12	pa_pa
G13	mu_mu

Definition 3.32 Gender parameter definition

```
oper
  Cgender : PType ; --in DiffBantu module
  Cgender = CgenderSwa ; in DiffSwa module
param -- in DiffSwa module
  CgenderSwa = G1|G2|G3|G4|G5|G6|G7|G8|G9|G10|G11|G12|G13 ;
```

3.5.1.1 Noun

The smart paradigms for the noun: main noun *mkN*, a relational noun with a preposition *mkN2* and three-place relational noun (with two prepositions) *mkN3* were inherited from the Bantu parameterized grammar thus shared. Also, the low-level paradigms for constructing compound nouns *compoundN* and irregular nouns *iregN* were also shared. The paradigm *mkNoun* that assigns noun strings generated by other low-level paradigms to the parameter number was shared too. The paradigm *regN* was modified to suit Swahili, as shown in Definition 3.33 below. Table 3.16 summarizes all the six nouns paradigms shared and one paradigm adapted, while gender and number parameters are adapted and shared respectively.

Definition 3.33 Regular noun paradigm

```

regN : Str ->Cgender -> Noun = \w, g -> let wpl = case g of {
    G1=>case w of {
        "mwa" + _ => PrefixPlNom G1 + Predef.drop 3 w ;
        "mwi" + _ => "we" + Predef.drop 3 w ;
        "ki" + _ => PrefixPlNom G4 + Predef.drop 2 w ;
        "m" + _ => PrefixPlNom G1 + Predef.drop 1 w ;
        _ => w };
    G2=>case w of {
        "mw" + _ => PrefixPlNom G2 + Predef.drop 2 w ;
        "mu" + _ => PrefixPlNom G2 + Predef.drop 2 w ;
        _ => PrefixPlNom G2 + Predef.drop 1 w };
    G4=> case w of {
        "ki" + _ => PrefixPlNom G4 + Predef.drop 2 w ;
        "ch" + _ => "vy" + Predef.drop 2 w ;
        _ => w };
    G6 | G8 => PrefixPlNom g + Predef.drop 1 w ;
    G11 | G12 | G13 => "" ;
    _ => PrefixPlNom g + w };
in mkNoun w wpl g ;

```

In a similar manner to the modeling of morphophonological rules in Bantu parameterized grammar, In bootstrapping the Swahili, they were done at the RE expression. For example, the word food in Swahili belongs to the class gender *ki_vi* which is coded gender G4. Therefore, it should be written as “kia-kula” and “via-kula” in singular and plural. The “kia” become “cha” while “via” become “vya” due to morphophonological rules thus “chakula and vyakula”. This rule is exemplified in the RE of definition 3.33 under gender G4.

Table 3.16 Summary of Bootstrapping Noun segments

Grammar segments	Grammar components
Shared paradigms	<i>mkN, mkN2, mkN3, compoundN, mkNoun, iregN</i>
parameter	<i>Number</i>
Adapted paradigm	<i>regN</i>
Parameter	<i>Gender</i>

3.5.1.2 Adjective

The adjective categories shared the two smart paradigms *mkA* (for normal an adjective) and *mkA2* (for adjectives with a preposition) and *regAAd*, a low-level paradigm for the adjective followed by adverbs inherited from the congruent Bantu parameterized grammar. The structure of the paradigms *regA*, *cregA*, *iregA* that generated strings for regular adjectives, for colour adjectives and irregular adjectives were modified, thus

forming portable segments. The paradigm *regAdj* for assigning strings using *AForm* was also ported. Table 3.17 below summarizes parameters and paradigms. Definition 3.34 represents the structure of the portable paradigms except *regA* given in Appendix B, B.2.

Table 3.17 Adjective parameters and paradigms

Grammar segments	Grammar components
Shared paradigm	<i>mkA, mkA2, regAAAd</i>
Adapted paradigm	<i>regA, cregA, iregA, regAdj</i>
Parameter	<i>Aform</i>

Definition 3.34 Adjective paradigm definition

```
regA : Str -> {s : AForm => Str} = \adj -> regAdj adj ("vi"+adj);

iregA : Str -> {s : AForm => Str} = \seo -> {
  s = table {
    AAdj g n => seo;
    Advv => "vi" ++ seo } };

cregA: Str -> {s : AForm => Str} = \seo -> {
  s = table {
    AAdj g Sg => ProunSgprefix g + "a rangi ya" ++ seo;
    AAdj g Pl => ProunPlprefix g + "a rangi ya" ++ seo;
    Advv => [] } };
```

3.5.1.3 Verb and Verb phrase

The smart paradigms for basic verbs (*mkV*), transitive (*mkV2*) and ditransitive verbs (*mkV3*) were shared. Besides, the paradigm *regVP* for verb phrase inflection was shared as well. The shared parameters are agreement, polarity and anteriority and the last two used the GF default. The derivational morphology parameter *VExte* was adapted because apart from passive, applicative, reciprocal, and causative already in congruent grammar, Swahili had an extra two: stative and reversive. The parameter *VForm* that enables generating verb forms was unique in Swahili. The paradigms *regV*, *iregV* and *mkVerb* for making regular verbs, irregular verbs and assigning verbs strings depending on various parameters respectively were adopted. The verb inflection table is large, consisting of 499 strings, as shown in Table 3.18 below. Table 3.19 summarizes the paradigms and parameters and their right segment in the computational grammar. The actual implementation of the category can be found in Appendix B, B.3. In VP morphology, the

paradigm *regVP* is shared while the paradigm for the auxiliary verb “*be*” *auxBe* is adapted because of unique morphemes.

Table 3.18 Swahili inflections forms

Language	Inflection function
Swahili	regV: String ¹ → String ⁴ mkVerb: String ⁴ → String ⁴⁹⁹

Table 3.19 Swahili paradigms and parameters

Grammar segments	Grammar components
Shared	paradigms <i>mkV, mkV2, mkV3, regVP</i>
	parameter <i>Agreement, polarity, anteriority</i>
Adapted	paradigm <i>mkVerb, auxBe,</i>
	Parameter <i>Vexte</i>
Unique	Paradigms <i>regV, iregV</i>
	parameter <i>Vform</i>

Normal declarative sentences and questions or relative clauses use indicative moods and this will be exemplified by the verbs used in Figures 3.37-3.40. For imperative mood, the verb “sleep” is used to illustrate it below. The first case is a plural command, the second a singular polite request and the last one is a plural command.

```
Lang> l UttImpPl PPos (ImpVP (UseV sleep_V))
lalani

Lang> l UttImpPol PPos (ImpVP (UseV sleep_V))
ulale

Lang> l UttImpSg PPos (ImpVP (UseV sleep_V))
lala
```

To exemplify subjunctive mood which is an expression of permission or probability of an event., Let's use the sentence "or let me run please" and "or let me die please". By parsing and linearizing the resultant output is shown below

```
Lang> p -lang=Eng "or let me run please" | 1
au wacha mimi nikimbie tafadhari

Lang> p -lang=Eng "or let me die please" | 1
au wacha mimi nikufe tafadhari
```

3.5.1.4 Pronoun

The parameter *PronForm* and paradigm *mkPron* for pronoun are shared with congruent Bantu parameterized grammar. To enable the bootstrap of Swahili pronouns, only lexicon definitions were done. Example 3.10 below shows the pronoun “we” output using the shared segments of pronoun grammar.

Example 3.10 Pronoun output

Lang> l -table we Pron	
s Pers : sisi	s (Poss Pl G1) : wetu
s (Poss Sg G1) : wetu	s (Poss Pl G2) : yetu
s (Poss Sg G2) : wetu	s (Poss Pl G3) : yetu
s (Poss Sg G3) : letu	s (Poss Pl G4) : vyetu
s (Poss Sg G4) : chetu	s (Poss Pl G5) : zetu
s (Poss Sg G5) : yetu	s (Poss Pl G6) : zetu
s (Poss Sg G6) : wetu	s (Poss Pl G7) : wetu
s (Poss Sg G7) : wetu	s (Poss Pl G8) : yetu
s (Poss Sg G8) : wetu	s (Poss Pl G9) : yetu
s (Poss Sg G9) : yetu	s (Poss Pl G10) : yetu
s (Poss Sg G10) : yetu	s (Poss Pl G11) : petu
s (Poss Sg G11) : petu	s (Poss Pl G12) : kwetu
s (Poss Sg G12) : kwetu	s (Poss Pl G13) : mwetu
s (Poss Sg G13) : mwetu	

3.5.1.5 Numeral

The three paradigms for constructing digits, namely: *mkDig*, *mk2Dig* and *mk3Dig* were shared beside *CardOrd* and *DForm* parameters. Swahili constructed unique paradigms *mkNum1*, *mkNum2*, *mkNum* and *regNum* for numerals 1, 2, 3 to 5 and 6 to 9 respectively and their multiples. The structures for the numeral rules were modified to suit Swahili hence becoming part of the portable grammar. Figure 3.33 below shows an example of a cardinal numeral generated with the grammar for the gloss “five hundred thousand nine hundred and thirty and three birds will swim”. The sentence is parsed from English to Bantu languages. However, in Swahili the last bit shown with blue font picks the default gender. Thus when Swahili is parsed it affects Kikamba which picks the same gender. correct as shown with blue font in the parsing of Swahili.

```
Lang> p -lang=Eng " five hundred thousand nine hundred and thirty and
three birds will swim" | 1
```



```

chinyoni chiribu amagana atano amagana kianda na emerongo etato na
isato chigocha koaka obari
nyunyi ngili maana atano maana kenda na miongo itatu na itatu
ikathambia
ndege elfu mia tano mia tisa na thelathini na watatu wataogelea

Lang> p -lang=Swa "ndege elfu mia tano mia tisa na thelathini na watatu
wataogelea " | 1
chinyoni chiribu amagana atano amagana kianda goetera emerongo etato na
basato bagocha koaka obari
nyunyi ngili maana atano maana kenda kwa miongo itatu na atatu
makathambia
ndege elfu mia tano mia tisa na thelathini na watatu wataogelea

```

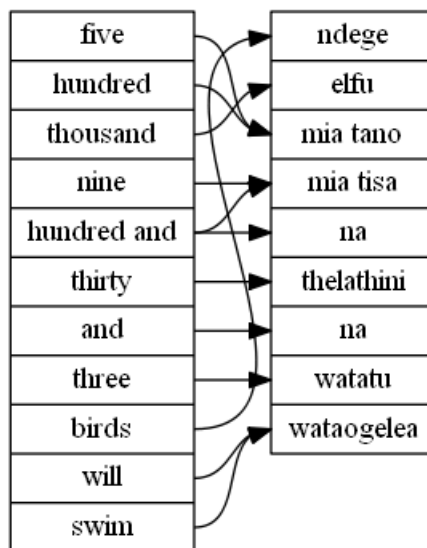


Figure 3.33 cardinal numeral example

3.5.1.6 Other categories

The paradigm *mkPrep* for prepositions and paradigms *mkAdv*, *mkAdA*, *mkCAAdv*, *mkAdN* for adverb modifying a verb, adjective, sentence and numeral respectively are shared. In addition, also shared are paradigms *mkConj* and *mkInterj* for conjunction and interjection respectively. The demonstrative determiners did not have a paradigm because they are one-string lexemes; however, their definition of lexeme structure was ported.

3.5.2 Syntax

The Bantu parameterized grammar had 163 implemented production rules at the syntax level, of which 149 rules are shared with Swahili; the rest (14) are ported. The portable rules are distributed, one on progressive verb as per Definition 3.35 while the rest are numeral and illustrated in Appendix B.4 part 4.

Definition 3.35 Progressive verb definition

```
ProgrVP vp = {s=\\ag,pol,tes,ant=>case < tes ,pol> of {  
  <Pres, _> => vp.s!ag!pol!Pres!ant;  
  <_, _> => auxBe.s!ag!pol!tes!ant ++vp.s!ag!pol!Pres!ant};  
  
  compl=\\a => vp.compl!a;  
  progV= []; imp =\\po,n =>vp.imp!po!n;inf=vp.inf};
```

To demonstrate that the bootstrapped Swahili grammar was working and producing the correct output even with only modifying 14 out of the 163 rules in syntax level, several parsing from English and linearization to Swahili were done at several categories and samples illustrated using different parse trees. Figure 3.34 below illustrates the output for category CN for the gloss “brown bread on the table” while the AP category is illustrated by Figure 3.35 below for the gloss “better than some student” The parse tree in Figure 3.36 illustrates category NP for the gloss “ all my three red eyes.

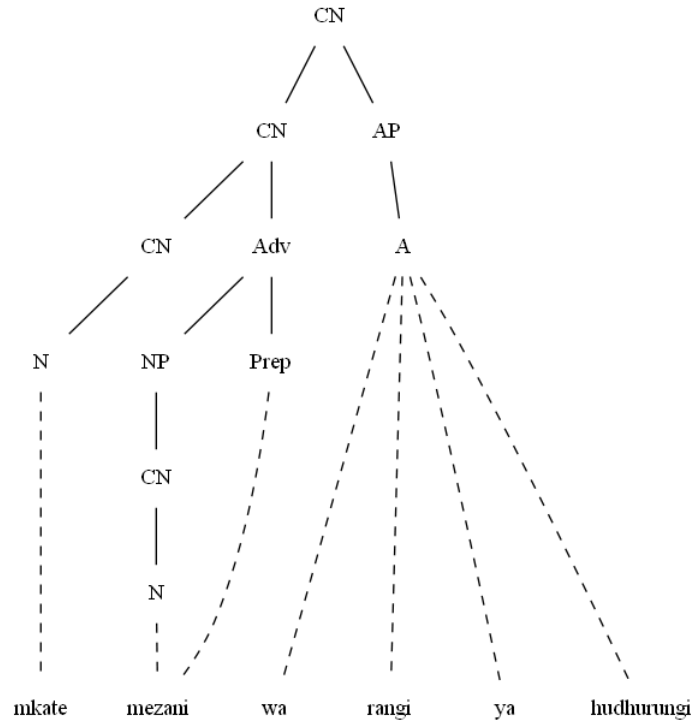


Figure 3.34 CN Example

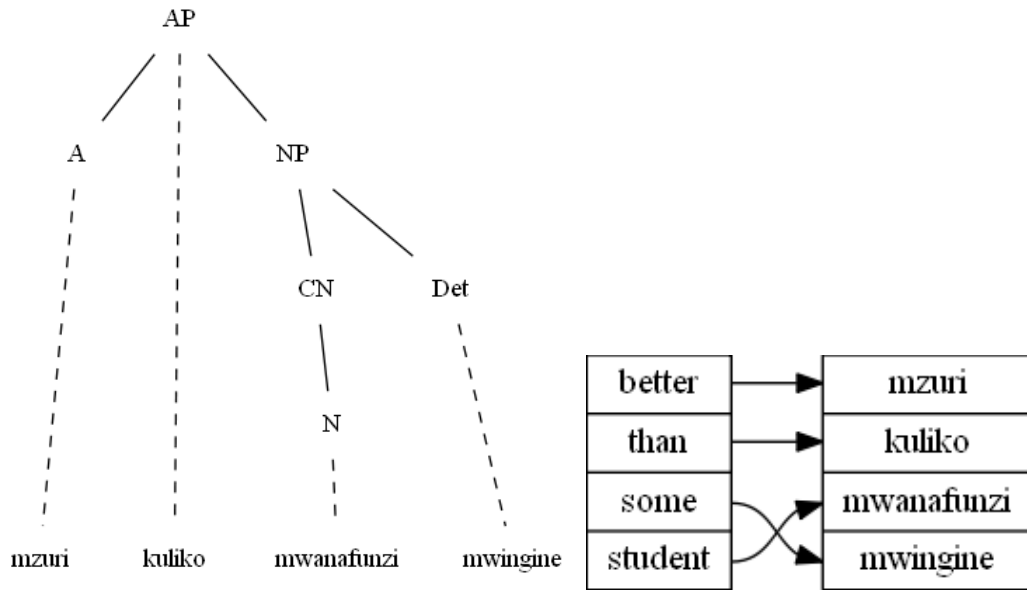


Figure 3.35 Example of AP parse tree and word alignment

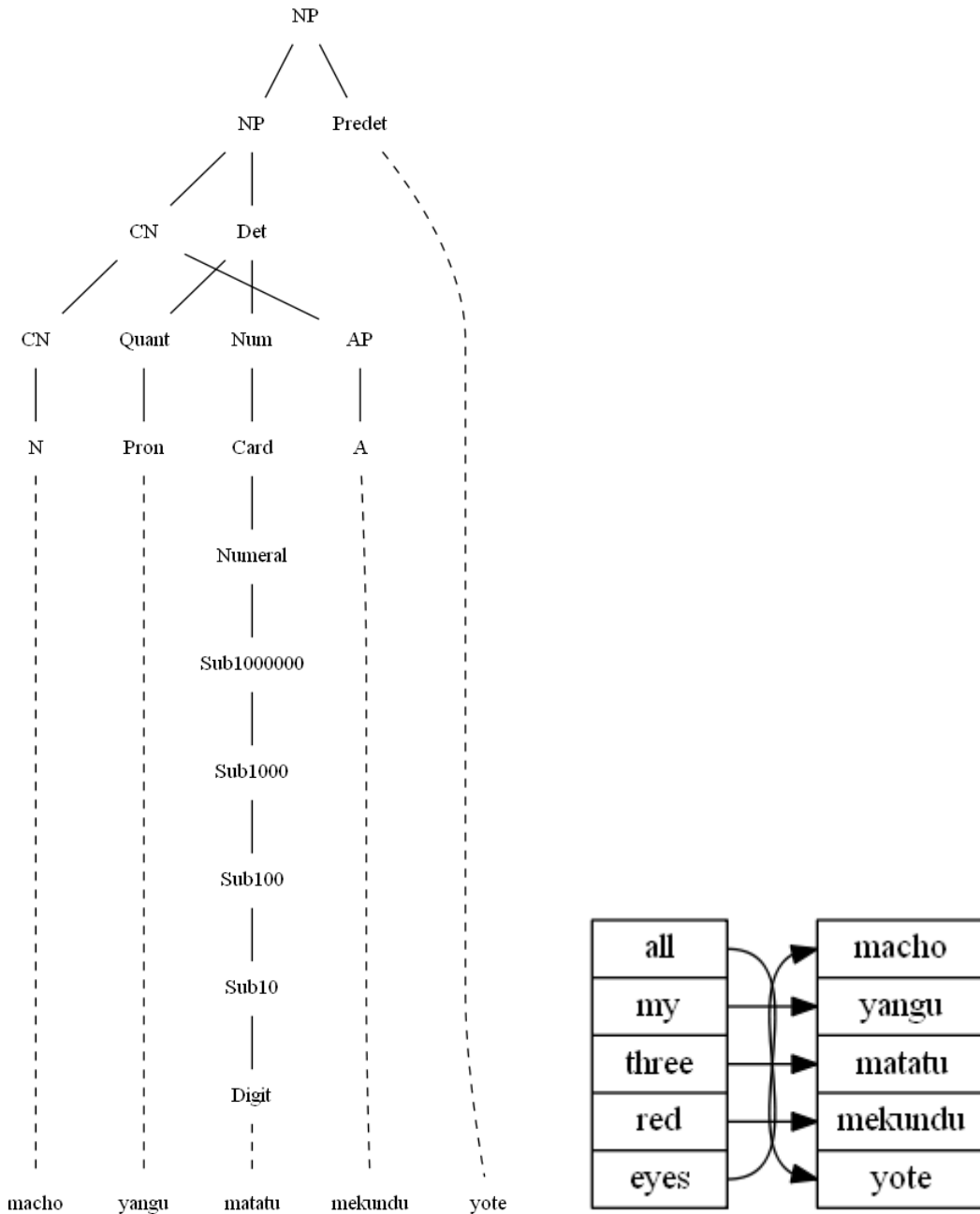


Figure 3.36 Noun phrase parse tree and word alignment

Figures 3.37 and Figure 3.38 below are examples of working VP and clause strings gloss “I read the best book” and “the children loved by the father” respectively.

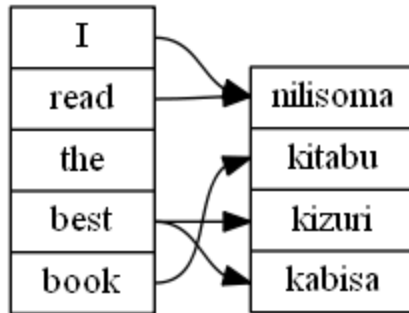
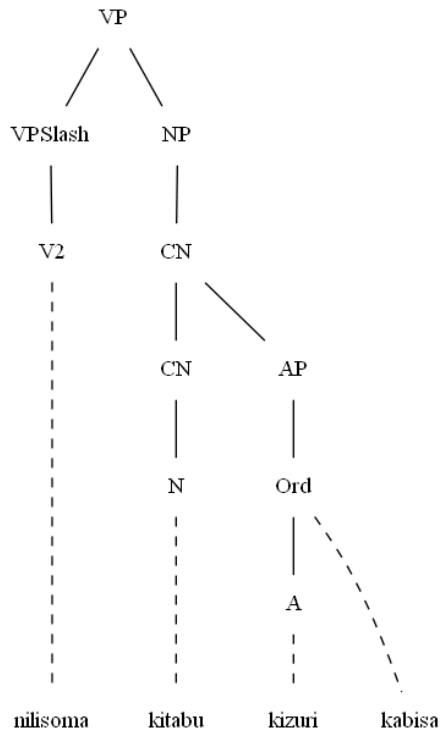


Figure 3.37 VP Example

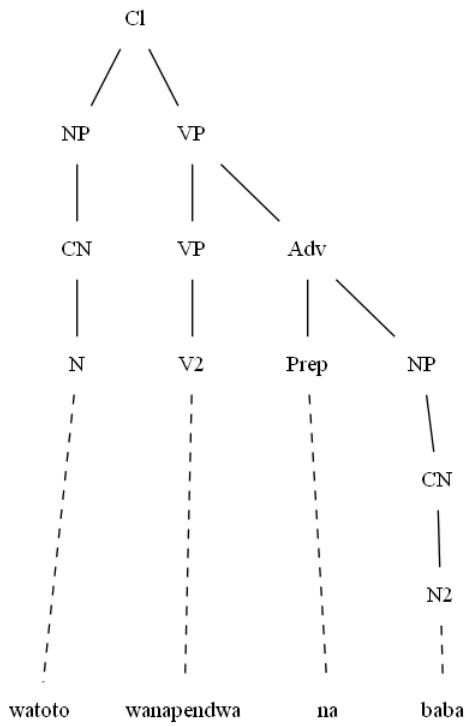


Figure 3.38 Clause Example

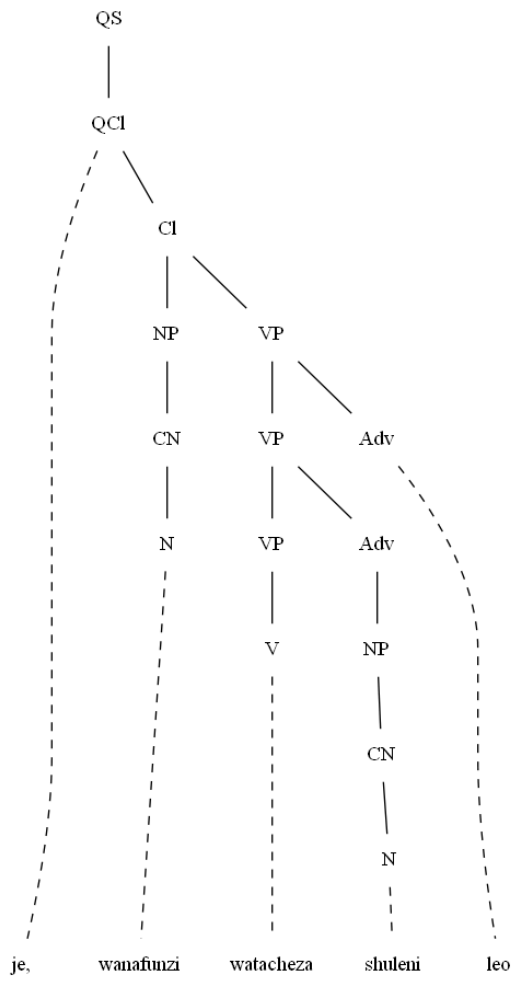


Figure 3.39 Question Parse tree

Figure 3.39 above illustrates the question clause for the gloss “when will the students play in school today”

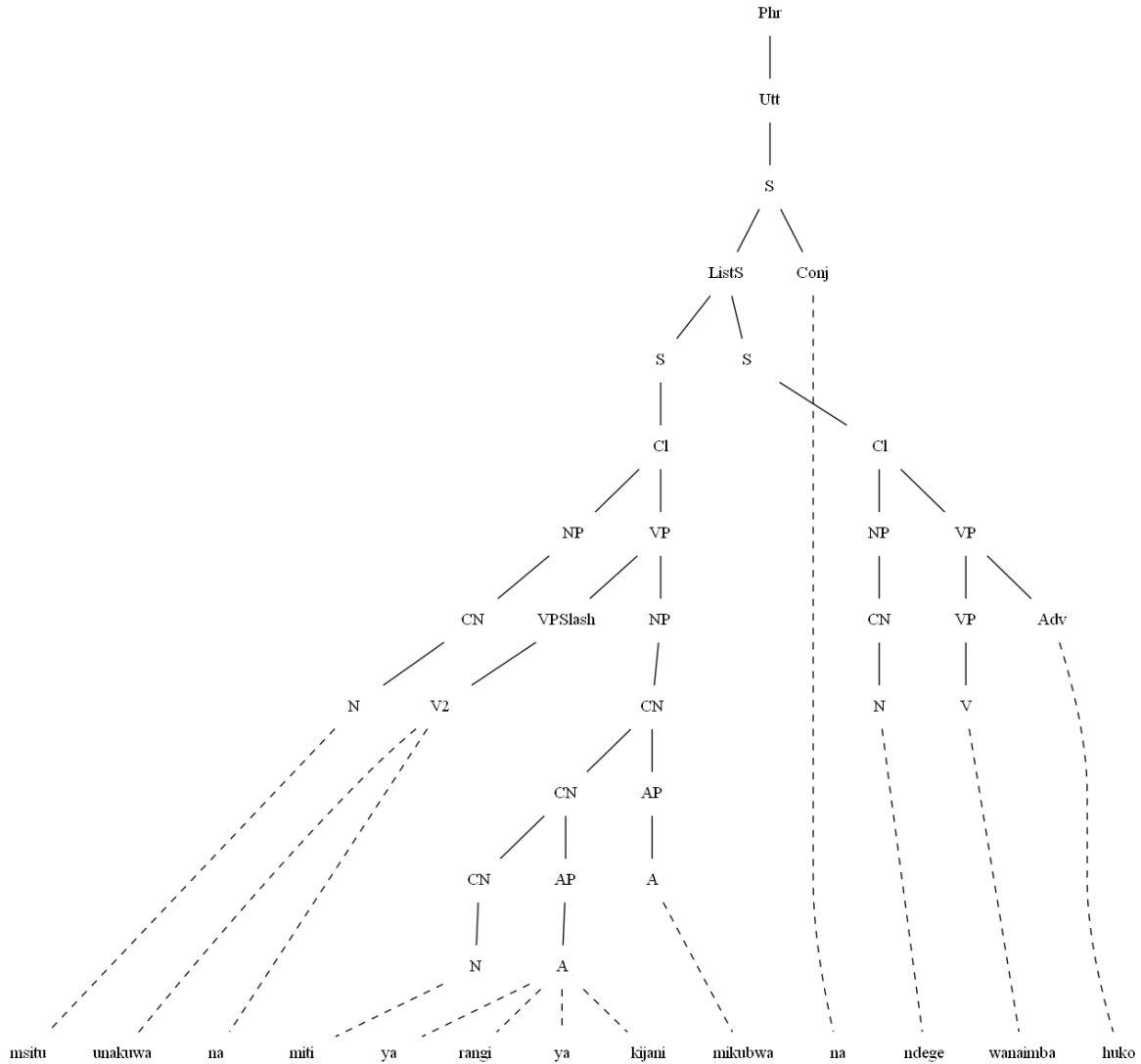


Figure 3.40 Phrase Example

Figure 3.40 above illustrates several categories: phrase, utterance, sentence and conjunction in sentences for the gloss “The forest has big green trees and birds sing there.” The above example and figures have demonstrated that the Bootstrapped Swahili grammar developed with minimal effort produces correct output.

3.5.3 Bootstrapped Grammar Testing and Evaluation

The GF regression testing process illustrated in Figure 2.7 was used to test every function (even the one inherited from Bantu parameterized grammar) during the bootstrapping process (development) to ensure accuracy in the new language. To evaluate

the grammar's performance after bootstrapping, an expert translated the test dataset (100 English sentences developed in section 3.4.3) into Swahili (gold standard) and after that, it was cross-checked by a linguist and is available in appendix C: c.2. The dataset was parsed and linearized to Swahili (machine translation) using the bootstrapped Swahili grammar. In case a sentence produced more than one linearization because of lexical variants or synonyms, then the one that best fits in reference to the gold standard was taken. For example, Example 3.11 below shows two outputs of one parsed sentence. The adjective “beautiful” is defined with stem variants “rembo” and “zuri” as shown below:

```
beautiful_A = regA "zuri" | regA "rembo" ;
```

Therefore, the two outputs are correct. However, the second one with “mrembo” is chosen as the best fit since it is used in human translation (gold standard).

Example 3.11 Many linearizations output

```
Lang> p "when everybody is young and beautiful and everything was good"
| 1 -lang=Swa -table
s : wakati kila mtu ni mchanga na mzuri na kila kitu kilikuwa kizuri
s : wakati kila mtu ni mchanga na mrembo na kila kitu kilikuwa kizuri
```

Below is a demonstration of parsing and linearization of sentences from any of the Bantu languages. The examples are drawn from the evaluation test-suite. Example 3.12 shows a Swahili sentence for the gloss: “gloss the twenty very bad men drank beer” where the NP consists of a numeral, adverb plus adjective and the sentence has past tense.

Example 3.12 NP parsing

```
Lang> p -lang=Swa "wanaume ishirini wabaya sana walikunywa pombe" | 1
abasacha emerongo ebere ababe mono bakanywa amarwa
andu aume miongo ili athuku vyu nimananyw[ ]ie nzovi
wanaume ishirini wovu sana walikunywa pombe
```

Example 3.13 show the parsing of the verb to be “were” in a Kikamba sentence with a gloss “two hundred thousand girls were good”. The sentence produces more than one linearization because of synonyms for the word “good”.

Example 3.13 verb to be

```
Lang> p -lang=Kam "eitu ngili maana eli mai aseo" | 1
abaiseke chiribu amagana ebere mbarengo abakeene --Gus
eitu ngili maana eli mai aseo --Kam
wasichana elfu mia mbili walikuwa wasahihi --Swa
abaiseke chiribu amagana ebere mbarengo abaya --Gus
eitu ngili maana eli mai aseo --Kam
wasichana elfu mia mbili walikuwa wazuri --Swa
```

Example 3.14 shows the parsing of a sentence in Ekegusii with a gloss “five skins have burned and the policemen are sleeping”. It demonstrates the use of auxiliary verbs “have” and “are” plus present tense and past tense in a sentence with a conjunction.

Example 3.14 Auxiliary verb

```
Lang> p -lang=Gus "amasangu atano asambire na abasigari bakorara" | pt
-number=1 | 1
amasangu atano asambire na abasigari bakorara
ithuma itano nisyavya na asikali nimakomete
ngozi tano zimechoma na polisi wanalala
```

Example 3.15 shows the parsing of an Ekegusii complex sentence consisting of a conjunction of AP and a sentence with gloss “those ten beautiful and clever friends have fallen now”

Example 3.15 Conjunction parsing

```
Lang> p -lang=Gus "abasani baria ikomi abasere na abang'aini bakagwire
bono" | pt -number=1 | 1
abasani baria ikomi abasere na abang'aini bakagwire bono
anyanya aya ikumi anake na oi nimavaluka yuyu
marafiki hao kumi wazuri na werevu wameanguka sasa
```

Example 3.16 is the parsing of a Swahili sentence with gloss “those boys swam and these girls ran” and intent to show the use of demonstratives

Example 3.16 Demonstrative parsing

```
Lang> p -lang=Swa "vijana hao waliogelea na wasichana hawa walikimbia"
| pt -number=1 | 1
```

```
abaisia baria bakaaka obari na abaiseke aba bakaminyoka
ivisi iya ni-nathambiie na eitu aa nimanasembie
vijana hao waliogelea na wasichana hawa walikimbia
```

Example 3.17 demonstrate parsing of standalone NP such as “everybody” and “everything” in the Kikamba sentence with gloss “when everybody is young and beautiful and everything was good”

Example 3.17 Standalone NP

```
Lang> p -lang=Kam "yila kila mundu ni wa muika na munake na undu wai
museo" | 1
ekero kera omonto nare omoke na omosere na kera egento getareng
egekeene
yila kila mundu ni wa muika na munake na undu wai museo
wakati kila mtu ni mchanga na mzuri na kila kitu kilikuwa kisahih
```

Example 3.18 is the parsing of a sentence consisting of two simple sentences from Ekegusii to Kikamba language with a minor error in the auxiliary verb “has “ instead of “wina” it gives “niuna”. For the gloss “ The forest has big green trees and birds sing there”.

Example 3.18 Two simple sentences

```
Lang> p -lang=Gus "oronsana robwate emete ya eragi ya machani emenene
na chinyoni chigotera ororo" | pt -number=1| 1 -lang=Kam
mutitu niuna miti ya langi wa matu minene na nyunyi niinaa vo
```

Figures 5.2 and 5.3 were also taken from the test-suite. In conclusion, these parsing and linearization demonstrate grammars are able to do synthesis and analysis.

The human translation and machine translation outputs were subjected to the online Tilde software and Perl scripts to extract BLEU, PER and WER. The comparative taxonomy was used to manually annotate errors for Swahili. The numbers of rules shared and adapted from the Bantu parameterized grammar were counted and converted to a percentage to demonstrate less effort to develop the bootstrapped grammar. The same was done for categories linearization, paradigms used and parameters.

3.6 Validation and Reliability

Gibbs (2007) and Creswell (2009) define validity and reliability as ways for the researcher to check the accuracy of the findings by employing specific procedures (trustworthiness, authenticity, and credibility strength of the research) and ensuring the approach to the research is consistent respectively.

Triangulation was used to ensure validity and credibility by having a variety of datasets for a language descriptive grammar. For Kikamba, the descriptive grammars by Mbuvi (2005), Kaviti (2004) and Welmers (1973) were used, while grammars by Njogu et al. (2006), Deen (2002) and Marten (2013) were utilized for Swahili. Finally, for Ekegusii languages, Osinde (1998), Ongarora (2008), Basweti (2005) and Otiso (2008) grammars were used. The research would pick a category at a time and compare all available descriptive grammars for a specific language to ensure uniformity and consistency. In case of discrepancies, the respective linguist(s) was/were requested to clarify so as to harmonize it before the actual computational grammar development.

Where elicitation was used, the output was subjected to another linguist/expert to ensure the descriptive grammar's correctness. The output on the computational grammar (especially the machine translation) using descriptive grammar from elicitation was subjected to member checking where the expert who developed the descriptive grammar would confirm the output is in tandem with what is expected. The manual translation of the test-suite from English to the specific Bantu language was also subjected to another expert to confirm the translation and to harmonize discrepancies.

Peer debriefing as a strategy for validity involved presenting the research progress in the Ph.D. seminars, conferences^{30 31,32} and summer³³ school and writing five journal papers as per Appendix F. The three grammars' machine-translation outputs were evaluated using external script/software with widely known and acceptable metrics.

³⁰ Advancing Science to Inform Sustainable Development conference. Held at Nairobi university Oct 2019

³¹ the 4th Strathmore International Mathematics Conference in 2017

³² the first Dekut international conference on science, technology and innovation in 2015

³³ sixth GF summer school held in cape town South Africa

On reliability, as Gibbs (2007) suggests, the avoidance of apparent mistakes that were observed in this research was necessary. For example, there are lexicon definitions available in the GF English language but no corresponding word in a specific Bantu language. For such a case, no corresponding definition was provided.

3.7 Summary

Geolinguistics was used to sample languages, while purposeful sampling was used in a Guthrie zone. Snowball sampling was used to identify linguists, experts, or grammar reference materials.

A hybrid research design was used, in which a descriptive case study was utilized to understand principles and parameters in each specific language. Then, a comparative analysis was used to compare Kikamba and Ekegusii similarities in terms of principles and parameters, resulting in a generalized descriptive grammar. The experimental design was used to develop the Bantu parameterized grammar in GF using the morphology-driven approach applying grammar engineering techniques. Functions of the grammar were developed using the evolutionary prototype model and the testing was done using the GF regression method. A detailed experimental process of designing Swahili grammar by bootstrapping to the Bantu parameterized grammar was discussed and how testing and evaluation were carried out. The next chapter will discuss the results of the Bantu parameterized grammar and the bootstrapped Swahili grammar and how the approach is effective and efficient.

Chapter 4 RESULT AND DISCUSSION

4.1 Introduction

The chapter presents a discussion on the results of: comparative descriptive grammar, shareability and portability of the Bantu parameterized grammar, evaluating the approach of bootstrapping using the Swahili grammar. Furthermore, the chapter demonstrates how the approach results in accurate grammar where a machine translation task was set up and BLEU, PER and WER scores extracted. It discusses how effective and efficient this approach is in adding a new Bantu language in the future and therefore provides a generalized step-by-step of how to add a new language. The chapter also discusses the reason why a 100% BLEU score could not be achieved in the right of errors reported. Finally, it compares the results with previously done work.

4.2 Comparative descriptive grammar

The results of comparative studies of Kikamba and Ekegusii grammars in terms of grammar rules plus regular expression and parameters plus principle are presented in Tables 4.1 and 4.2 respectively. These results are based on objective one. In terms of variation, Table 4.1 shows the Kikamba grammar has a comparative degree of adjective at morphology while Ekegusii at syntax hence a divergence based on this parameter. In cardinal numerals from number six to nine and its multiple does not exist in Ekegusii grammar but a repetition of one to five unlike in Kikamba. Consequently, the numeral category constitutes a large portion of the unique grammar. The grammars show high similarities in terms of parameters and RE. However, the regular expressions are constructed in the abstract, in terms of concrete context they shall differ in actual morphemes. The rest of the categories are similar and this proves empirically the concept of universal grammar. Thus, similarities in RE structures hence become a segment of portable grammar. The comparative adjective and numeral for numbers five to eight and their multiples differ. Hence each language has a unique grammar case. Finally, the NP, VP, sentences and questions are segments of shared grammar. Furthermore, the high presence of cross-linguistic similarities is an indication of a good percentage of shared grammar.

Table 4.1 Generalized regular expression and grammar rules

Category		Bantu shared RE and GR
Noun		Gender prefix(number) ++ root
Adjective	positive	concord prefix(number) ++ root
	Comparative	
	Colour	concord prefix (number) + string + colour lexicon
Verb	Positive polarity	Focus(optional) ++concord(subject) ++ tense ++ concord(object) ++ derivative morpheme ++ final vowel
	Negative polarity	concord(negation) ++ tense ++ concord(object) ++ derivative morpheme ++ final vowel
Pronoun	Personal	String(based on agreement)
	possessive	Concord ++ root
Demonstratives/quantifier		Concord prefix(number) ++ root
Preposition		Concord(number) string OR independent string OR noun + string(infused)
Number	cardinal	Multiples of 0-5 Concord ++ root
		6-8 Concord ++ root + Concord ++ root (Ekegusii only)
	ordinal	Concord + cardinal string(except 1-3)
Noun phrase (NP)		Demonstrative + Noun +Possessive + Demonstrative + Numeral +Adjective or Personal Pronoun
Verb Phrase(VP)		Verb + post modifier
Sentence		NP + VP + NP, NP + VP + VP, NP +VP, VP+NP, VP+ VP, VP
Conjunction		Phrase + conjunction + phrase

Table 4.2 Generalized parameters and principles

Category/Phrase		Parameters and Principles		
		Kikamba grammar	Ekegusii grammar	Bantu Shared grammar
Noun/common noun		Gender and Number	Gender and Number	Gender and Number
Adjective/Adjective phrase		Concord, number and degree(positive and comparative)	Concord, number and degree (positive)	Concord, number and degree (positive)
Verb/verb phrase	Normal	Agreement(person, number and concord), valence, mood, tense, aspect and derivation	Agreement(person, number and concord), valence, mood, tense, aspect and derivation	Agreement(person, number and concord), valence, mood, tense, aspect and derivation
	Imperative	Polarity, number, command and request	Polarity, number, command and request	Polarity, number, command and request
Pronoun	personal	Agreement(person, number and gender	Agreement(person, number and gender	Agreement(person, number and gender
	Possessive	Concord and number	Concord and number	Concord and number
Demonstratives		Concord and number	Concord and number	Concord and number
Number(numeral, digits)		Concord, cardinal and ordinal	Concord, cardinal and ordinal	Concord, cardinal and ordinal
Preposition		Concord,number,infuse	Concord,number,infuse	Concord,number,infuse
Adverbs/Interjection		-	-	-
Determiner		Concord, number and position	Concord, number and position	Concord, number and position

Noun Phrase	Case and agreement	Case and agreement	Case and agreement
Sentence and relative clause	Topology, tense and polarity	Topology, Tense and polarity	Topology, Tense and polarity
Question clause	Tense, question form(direct or indirect) and polarity	Tense, question form(direct or indirect) and polarity	Tense, question form(direct or indirect) and polarity

4.3 Bantu Parameterized Grammar Evaluation

The shareability of the congruent Bantu parameterized grammar at morphology involved counting the shared linearization of categories, paradigms, parameters and converting the count to percentages, while at syntax, the shared production rules were expressed in percentages. In portability, the similar structure production rules, linearization of categories, paradigms and parameters were counted and converted to a percentage.

4.3.1 Morphology shareability and portability

The Bantu parameterized grammar had thirty-seven³⁴ categories sharing their linearization (inflections) (see appendix E). The Kikamba and Ekegusii have gender systems influencing almost all categories. Furthermore, most of the inflection parameters used in the linearization are shared. However, the unique parameters at the categories (Part of speech tags) level share names due to standardization but differ in values. Accordingly, this led to 100% sharing of the linearization for congruent grammar, implying the definition of linearization categories was done once, thereby reducing the effort of definition by half.

Table 4.3 below represents the parameters used to implement the Bantu parameterized grammar. Most of the parameters are shared because of the influence of gender and its concord system; such as, *PronForm* for a pronoun, *CardOrd*, *DForm* for numerals agreements plus *polarity* for verbs etc. These parameters, *Infusion*, *Case*, *Qform*, *ImpForm*, are also shared but not influenced by the gender system. Some parameters, such as the Adjective parameter (*AForm*), derivative morphology of verbs (*VExte*) and genders, are exhibited by morphemes whose values differ from language to language hence shared at naming convention thus adapted (some modification) to suit the current grammar. Table 4.4 below demonstrates less effort needed in defining parameters because 68.75% of them

³⁴ Housed in the BantuCat module

were defined once (shared), while for 18.75% of them, values were modified to suit each specific grammar. Only 12.5% of the parameters were defined for each grammar. This means that the parameters rule-base was reduced by 68.75% in the Bantu parameters grammar, implying less time and effort in defining them, plus the standardization of the naming convention led to a modification of 12.5% of the parameters. Therefore, the benefits acquired in defining the 12.5% parameters of one grammar are transferred to the next one.

Table 4.3 Congruent grammar parameters

Category	Parameters		
	Shareable	Portable	Unique
Noun	<i>Number</i>	<i>Gender</i>	
Adjective		<i>Aform</i>	
Verb/VP	<i>Agreement, polarity, Anteriority</i>	<i>Vexte</i>	<i>Vform,tense</i>
Pronoun	<i>PronForm</i>		
Numeral	<i>CardOrd, DForm</i>		
Preposition	<i>Infusion</i>		
Noun Phrases	<i>Case</i>		
Questions	<i>Qform</i>		
Imperative	<i>ImpForm</i>		

Table 4.4 Paradigms and parameters percentages

Segment	Paradigms		Parameters	
	Count	%	Count	%
Shareable	32	65.3	11	68.75
Portable	7	14.29	3	18.75
Unique	10	20.41	2	12.5
Total	49	100	16	100

Table 4.5 below presents all paradigms used to develop the Bantu parameterized grammar. The numeral categories had the highest unique paradigms because Ekegusii words for cardinal numerals end at five rather than nine; hence, extra paradigms for constructing numerals six through nine. However, digit paradigms are shared. Verbs had unique paradigms because of unique infix morpheme strings for derivational morphology, influenced by a unique parameter. Generally, all smart paradigms are shared. Some low-level paradigms for verbs, nouns and adjectives were ported because of specific prefixes,

infixes, and suffixes morphemes in each language. Table 4.4 above shows 65.3% of the paradigms are shared, thus defined once, significantly reducing the effort of constructing morphologically regular expressions. Such reduction enables rapid and accelerated development of the overall grammar. 14.29% of paradigms were modified to be compatible with the respective specific grammar. Finally, only 20.41% was uniquely defined to be specific for each grammar. The implication is that only 34.7% of paradigms rule-based work was done, which involved defining the specific and modifying similar structure paradigms. It is possible to define a language's morphology with less effort since paradigms are the key to the inflection table.

Table 4.5 Congruent grammar paradigms

Category	Paradigms		
	shareable	portable	Unique
Noun	<i>mkN, mkN2, mkN3, compoundN, mkNoun, iregN</i>	<i>regN</i>	
Adjective	<i>mkA, mkA2, regAAAd</i>	<i>regA, cregA, iregA, regAdj</i>	
Verb/VP	<i>mkV, mkV2, mkV3, regVP, dirV2, prepV2, dirdirV3, prepPrepV3, dirV3, mkVV, mkVA</i>	<i>mkVerb, auxBe</i>	<i>regV, iregV</i>
Pronoun	<i>mkPron</i>		
Numeral	<i>mkDig, mk2Dig, mk3Dig, mkcard</i>		<i>mkNum1, mkNum2, mkNum, regNum, mkNum6, mkNum7, mkNum8, mkNum9</i>
Preposition	<i>mkPrep</i>		
Adverbs	<i>mkAdv, mkAdA, mkCAAdv, mkAdN</i>		
Others	<i>mkConj, mkSubj, mkPN, mkIP, regPN, NounPN, mkInterj</i>		

At the morphology level, the rule-base development effort is reduced by 100%, 68.75% and 65.3% at the definition of linearization categories, parameters, and paradigms respectively. The significant reduction of the rule-base implies it would take less time to develop the Bantu parameterized grammar than monologue grammars. The implication is that exploiting Bantu languages' morphological similarities helps reduce development efforts in terms of the rule-base. This is a significant development since these Bantu languages have a complex morphology (prefixing, infixing and suffixing) combined with

several genders (nominal classes) and their influence on other categories (concord) that would have complicated the grammar. Therefore, using this approach to develop the Bantu parameterized grammar helped accelerate the morphology definition in a cost-efficient manner.

4.3.2 Syntax shareability and portability

Table 4.6 shows the result of syntax rules shared and modified (portable) represented per module in GF RGL. The *adjective* and *adverb* modules difference is because Kikamba has a morphology-driven comparative adjective while Ekegusii is syntax-directed. The one modified rule in the *idiom* module results from progressive verbs whereas, the progressive verb consists of two consecutive verbs (the linking verb and the action verb), which in Kikamba are fused together. The numeral module had a significant number of modified rules because the production rules had lexemes and conjunctions in them and are unique to each grammar. Overall, 10.43% of the Bantu parameterized grammar rules are portable.

Table 4.6 shareability and portability

GF modules	Rules implemented	Shareability		Portability	
		Rules	%	Rules	%
Adverbs	7	6	85.71	1	14.29
Adjective	11	9	81.82	2	18.18
Conjunction	9	9	100.00		0.00
Idiom	10	9	90.00	1	10.00
Noun	42	42	100.00		0.00
Phrase	19	19	100.00		0.00
Question	10	10	100.00		0.00
Relative	5	5	100.00		0.00
Sentence	14	14	100.00		0.00
Numeral	15	2	13.33	13	86.67
Verb	21	21	100.00		0.00
Total	163	146	89.57	17	10.43

The shareability of the grammar at syntax stands at 89.57%. This was mainly attributed to the two languages sharing the same topology principles and having gender systems, thus similar inflection of categories and sharing most of the parameters used in

syntax. This means that at least 89% of the syntax rules were not redefined (146 rules), which significantly reduces the grammars' rule-base. The implication is that less development effort is needed to develop the Bantu parameterized grammar for similar languages if the cross-linguistic principles and parameters are exploited fully.

4.3.3 Grammar Quality

Table 4.7 below presents the extracted machine translation metrics score. The congruent grammar is evaluated by the pairs, English to Kikamba and English to Ekegusii.

Table 4.7 Translation Metrics

Language	Cumulative BLEU %				PER	WER
	1-gram	2-gram	3-gram	4-gram	%	%
Eng- Kam	93.61	89.77	86.32	83.05	10.96	12.82
Eng-Gus	80.50	69.25	61.86	55.95	19.49	23.90

The BLEU score measures the similarity index by comparing the same phrase length (n-gram) between the target (candidate) and reference (gold standard) sentences. Though, 1, 2, 3 and 4 grams (phrase length) are scored. To address the fluency of the translation, since GF is known to over-generate, the longer n-gram (4-gram) is used. The Kikamba 83.05 % 4-gram BLEU score is relatively high for such complex morphology language while Ekegusii 55.95% score is encouraging for a language with much morphophonological transformation though it should be noted that the developer is well versed with Kikamba hence much influence in the accuracy of the grammar during development. The metrics PER and WER were used to investigate errors because the former does not penalize position while the latter does and this had a huge effect on accuracy, especially where two consecutive adjectives were used in a sentence as illustrated by Figure 4.1 below, where a sentence in Ekegusii has correct translation. However, due to two consecutive adjectives: red (*chimbariri*) and small (*chinke*) which are interchanged in the target (candidate) translation, it results in 50% WER and 0% PER. The such, interchange was due to GF nature of constructing the abstract trees for the same category. Besides, the implication is high on the BLEU score since it scores partly 22.59%. Table 3.19 above shows that Kikamba had 10.96% and 12.82% and Ekegusii had 19.49% and

23.90% for PER and WER scores. An in-depth analysis of the errors by manual annotation using the comparative taxonomy is shown in Figure 4.2 below:

Source	-	-	small red seeds smell
Human	100.00	1.00	chintetere chinke chimbariri chigotiokerera
Machine	22.59	1.00	chintetere chimbariri chinke chigotiokerera

Figure 4.1 Position interchanged error

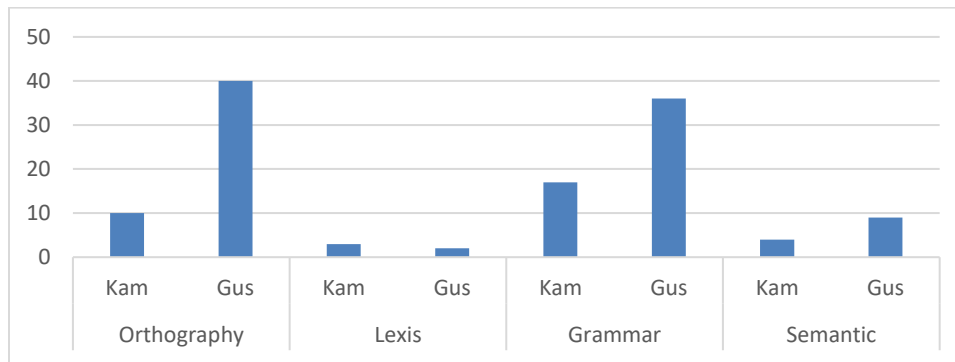


Figure 4.2 Manual error analysis

Orthography errors involved many misspelled words requiring addition, deletion, or substitution of one or more letters at the word level, especially in the Ekegusii grammar. These errors were mainly due to phonological issues resulting from nasal deletion and insertion present in the Bantu languages and Ekegusii is richer in them. Most of the phonological rules were not available in the descriptive grammars and thus not captured in the morphology since they were realized at the evaluation stage. Figure 4.3 below shows the subject marker (*ma*) and the first vowel of the verb stem come (*u*) when concatenated in Kikamba. As a result of the phonological (nasal) issue, the vowels change to double *oo* thus, the right translation is *nimooka* (gloss is “come”). Figure 4.4 below again shows when two consonants meet (r and g), the *g* is deleted in Ekegusii due to nasal issues. Therefore, these were not inflectional errors but phonological errors. These morphophonological rules can be investigated, developed and added to the grammar as future work.

Source	-	-	young doctors have come today
Human	100.00	1.00	aiiti ma muika nimooka umunthi
Machine	42.73	1.00	aiiti ma muika nimauka umunthi

Figure 4.3 Kikamba orthography error

Source	-	-	old dry seeds were bought
Human	100.00	1.00	chintetere chinkoro chinkamoku chikagorwa
Machine	21.02	1.00	chintetere chinkamoku chinkoro chikagorgwa

Figure 4.4 Ekegusii orthography error

Some words were added or subtracted in the human reference or target reference to ensure the translation was semantically correct. Such errors are at the lexis level. Figure 4.5 below shows the word “rain” is translated with two words, “mbua kua” in Kikamba, to ensure the meaning is well captured in line with the sentence.

Source	-	-	those two hundred green trees will fall after rain
Human	100.00	1.00	miti iya maana eli ya langi wa matu ikavaluka itina wa mbua kua
Machine	76.26	0.85	miti iya maana eli ya langi wa matu ikavaluka itina mbua

Figure 4.5 Example of Lexis error

Grammar errors were the highest and mostly related to the verb phrase. There was mis-selection in the verb tense. For example, Ekegusii has variants of past tense (immediately, near, far, remote). When all the variants were implemented in GF, due to the large verb inflection table, the compiler took a long time (more than 8 hours) to process. In fact, in most of the cases, the process was killed. Therefore, for testing purposes, we used only one tense in each category and this led to the scenario shown in Figure 3.36 below where the human translation is in the remote past tense while the machine translation is in the far past tense. Therefore, coping with the limitation of time complexity in GF led to several errors.

Source	-	-	the twenty very bad men drank beer
Human	100.00	1.00	abasacha emerongo ebere ababe mono banywete amarwa
Machine	64.35	1.00	abasacha emerongo ebere ababe mono bakanywa amarwa

Figure 4.6 Example of verb tenses error

Semantic errors occurred when action had to be explained by more than one word; otherwise, the word used in a specific context never made sense and Figure 4.5 above shows an example of the former. Verbs contributed most of the errors through verb tenses and phonology vowel change (morph phonological transformation). Despite the errors and leveraging the cross-linguistic principles and parameters, the research created accurate and cost-efficient Bantu parameterized grammar.

In conclusion, this approach has shown a significantly reduced rule-base size, in that at the morphology: 65.3% of the regular expression were shared plus 68.3% of the parameters. Furthermore, at the syntax level sharing was at 89.57%. This reduced significantly the effort needed to develop multilingual grammar since much of the work is already dealt at the congruent grammar level.

4.4 Bootstrapping Swahili Grammar

Swahili has genders and concord systems, at the morphology level, like grammars used to develop the Bantu parameterized grammar; thus, all the linearization categories were shared. Therefore, the thirty-seven categories were inherited from the congruent grammar, consequently reducing the linearization categories defining effort by 100%. In terms of paradigms (regular expressions), in Swahili, the numerals' unique paradigms were reduced to four compared to Ekegusii which had eight of them, as shown in Table 4.8. Overall, Swahili shared 32 paradigms with the Bantu parameterized grammar, translating to 71.11%, as shown in Table 5.2. This means that before one starts to develop (bootstrap) the Swahili grammar, over 71% of paradigms are already in place.

Moreover, 15.55% of the regular expressions were modified to suit Swahili. Therefore, paradigm structures were maintained, enabling faster and rapid development. Only 13.33% of the paradigms were uniquely defined, which is a small effort that can take less time compared with defining 100% of the paradigms.

Table 4.8 Swahili paradigms

Category	Paradigms		
	shareable	portable	Unique
Noun	<i>mkN, mkN2, mkN3, compoundN, mkNoun, iregN</i>	<i>regN</i>	
Adjective	<i>mkA, mkA2, regAAAd</i>	<i>regA, cregA, iregA, regAdj</i>	
Verb/VP	<i>mkV, mkV2, mkV3, regVP, dirV2, prepV2, dirdirV3, prepPrepV3, dirV3, mkVV, mkVA</i>	<i>mkVerb, auxBe</i>	<i>regV, iregV</i>
Pronoun	<i>mkPron</i>		
Numeral	<i>mkDig, mk2Dig, mk3Dig, mkcard</i>		<i>mkNum1, mkNum2, mkNum, regNum,</i>
Preposition	<i>mkPrep</i>		
Adverbs	<i>mkAdv, mkAdA, mkCAAdv, mkAdN</i>		
Others	<i>mkConj, mkSubj, mkPN, mkIP, regPN, NounPN, mkInterj</i>		

Table 4.9 Swahili Paradigms and Parameters

Segment	Paradigms		Parameters	
	Count	%	Count	%
Shareable	32	71.11	11	68.75
Portable	7	15.55	3	18.75
Unique	6	13.33	2	12.5
Total	45	100	16	100

Table 4.9 above shows that Swahili shared 68.75% of the parameters with Bantu parameterized grammar, meaning they were inherited from the Bantu Functor without the effort of defining them, while 18.75% of the parameters were modified to suit the bootstrapped Swahili. Finally, only 12.5% of the parameters were defined uniquely for this grammar. The Bantu parameterized grammar and bootstrapped Swahili had the same number of parameters as shown in Table 4.9. To summarize morphology, 100% of linearization categories, 71.11% of paradigms and 68.75% of parameters were not defined afresh but wholly inherited from the Bantu parameterized grammar, significantly reducing the morphology rule-base effort and development time. Consequently, this bootstrapping approach is able to achieve morphology rule-base with minimal effort (efficient). The

implication is that adding a new grammar will take less effort for the rule-base, especially if they originate from the same geographical area since the languages involved here are spoken in different geographical areas.

Table 4.10 Bootstrapped grammar syntax rules

GF modules	Rules implemented	Shareability		portability	
		Rules	%	Rules	%
Adverbs	7	7	100.00		0.00
Adjective	11	11	100.00		0.00
Conjunction	9	9	100.00		0.00
Idiom	10	9	90.00	1	10.00
Noun	42	42	100.00		0.00
Phrase	19	19	100.00		0.00
Question	10	10	100.00		0.00
Relative	5	5	100.00		0.00
Sentence	14	14	100.00		0.00
Numeral	15	2	13.33	13	86.67
Verb	21	21	100.00		0.00
Total	163	149	91.41	14	8.59

Table 4.10 above shows the distribution of syntax production rules for bootstrapped Swahili based on GF modules. Fourteen rules were ported, one and thirteen from *idiom* and *numeral* modules as the case was in the Bantu parameterized grammar. These fourteen ported rules are summarized in appendix G. GF allows defining general rules in the Functor; if it requires modification in a specific grammar, it is just excluded from being inherited. The above scenario was used to define the comparative adjective syntax rules for Kikamba in adverbs and adjective modules. Therefore bootstrapped Swahili comparative adjective rules are the same as in the Bantu parameterized grammar. This explains why all rules in adverbs and adjectives are shared, thereby increasing the shared rule-base. At the syntax phase, 91.41% of the rules (149) are shared with the Bantu parameterized grammar and the main work in bootstrapping the grammar was to modify 8.59% (14 rules) of the rules. This meant even before adding Swahili, 91.41% of the rules work was already done. This leads to faster development and scaling up of the grammar.

Table 4.11 presents the BLEU, PER and WER metrics results of the bootstrapped Swahili grammar. The 4-gram BLEU score of 77.75% is high, encouraging and demonstrates that bootstrapping can cost-effectively develop accurate grammar through exploiting similarities in already developed grammar. Figure 4.7 shows that most of the errors occurred in the categories of grammar and semantics. In grammar, verb errors contributed the most. For example, the present and habitual tenses are sometimes used interchangeably, but as explained earlier, GF takes too long to process more than one alternative tense due to the inflection table size.

Table 4.11 Swahili grammar accuracy

	Metrics	Percentage
BLEU	1-gram	91.29
	2-gram	86.00
	3-gram	81.44
	4-gram	77.95
	PER	9.46
	WER	13.39

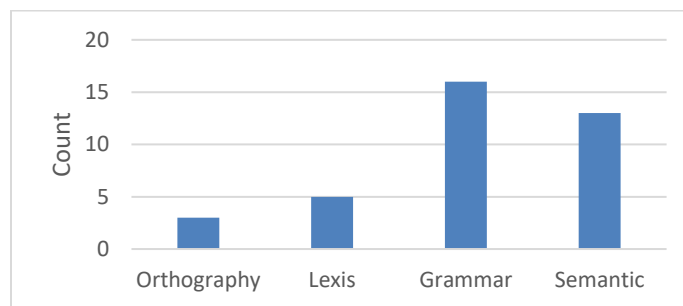


Figure 4.7 Swahili grammar errors

Figure 4.8 shows where the human translation uses habitual tense while the machine translation picks a present tense, reducing the BLEU score. Figure 4.9 shows a semantic error where the machine translation picks the word for the floor; however, regarding usage of the word in the sentence “*sakafu*” is for a building floor, but for ground, “*udongo*” would be the appropriate word.

Source	-	-	these five thousand women sing
Human	100.00	1.00	hawa wanawake elfu tano huimba
Machine	25.41	1.00	wanawake hawa elfu tano wanaimba

Figure 4.8 Tense error

Source	-	-	some big forests had wet floors and green grass
Human	100.00	1.00	misitu mingine mikubwa ilikuwa na udongo mnyevu
Machine	61.48	1.00	misitu mingine mikubwa ilikuwa na sakafu nyevu

Figure 4.9 word ambiguity

In conclusion, this approach of bootstrapping grammar significantly reduces the development effort in terms of the rule-base, as summarized in Table 4.12 below. The linearization categories, paradigms, parameters and syntax rules were reduced by 100%, 71.11%, 68.75% and 91.41% respectively. This is a large chunk of the work involved in defining Swahili grammar. In fact, the effort done on Swahili grammar was defining 13.33% and 12.5% of paradigms and parameters plus modifying 15.55%, 18.75% and 8.59% paradigms, parameters and rules respectively and finally, defining the lexicons. Therefore, bootstrapping a similar grammar to already developed Bantu parameterized grammar by exploiting the cross-linguistic similarities reduces the development effort significantly, resulting in cost-efficient, cost-effective, and accurate grammar. As a result, it enables faster development of grammar for under-resourced languages.

Table 4.12 Summary of bootstrapped grammar

Grammar section	Shareable	Portable	Unique
Linearization	100%		
Paradigms	71.11%	15.55%	13.33%
Parameters	68.75%	18.75%	12.5%
Syntax rules	91.41%	8.59%	

Therefore, to grammar developers especially for under-resourced languages, the research has provided an approach that will accelerate the development of grammar in a multilingual ecosystem with less effort. For GF users, the GF resource library has been extended by providing three concrete grammars for Ekegusii, Swahili and Kikamba. To

the Bantu linguists, the research has provided empirical evidence of UG. Finally, to the policy maker, by exploiting these cross-linguistic similarities, it is easier to develop grammar resources for even less-resourced related languages thus preserving these languages.

4.5 The Generalized developed Bootstrap Approach

The approach involves two main stages development of the congruent grammar for a particular family and then bootstrapping for similar grammars. The purpose of summarizing the steps of the approach of bootstrapping the development of rule-based grammar is to ensure it can be adapted either by bootstrapping more Bantu languages or developing another family-shared parametrized grammar. A pseudocode is provided in Definition 4.1 which details all the finer steps of the approach while Figure 4.10 summarize the flow of the steps.

Definition 4.1 the approach pseudocode

```

Approach of Bootstrapping Multilingual Grammar Development (i,n)
1 Initialize languages n -- under-resourced languages family
2 Initialize grammar formalism
3 For lang == 1 to i Do -- i languages for developing shared grammar
4     descriptive grammar analysis - for cross-linguistic similarities
5     missing gaps filling --language analysis and translation
6     Shared <-- extract shared principles and parameters
7     Portable <-- extract portable principles and parameters
8 Endfor
9 if Shared == True
10    develop congruent parameterized grammar
11 else If Portable == True
12    develop portable parameterized grammar
13    else
14        while lang < i -- no sharing or portability
15            Develop language-specific grammar
16        Endwhile
17    EndIf
18    Metrics <--evaluate congruent parameterized grammar reusability
19    Return metrics
20 EndIf
21 For lang == i+1 to n Do
22    Analysis the descriptive grammar
23    Extract portable and unique grammar
24    bootstrap grammar

```

```

25   Metrics <-- Evaluate extendibility -- to congruent grammar
26   Return metrics
27 Endfor
28 For lang == 1 to n Do
29   Metrics <-- use machine translation to evaluate performance --
    BLEU,PER,WER
30 Return metrics
31 Endfor

```

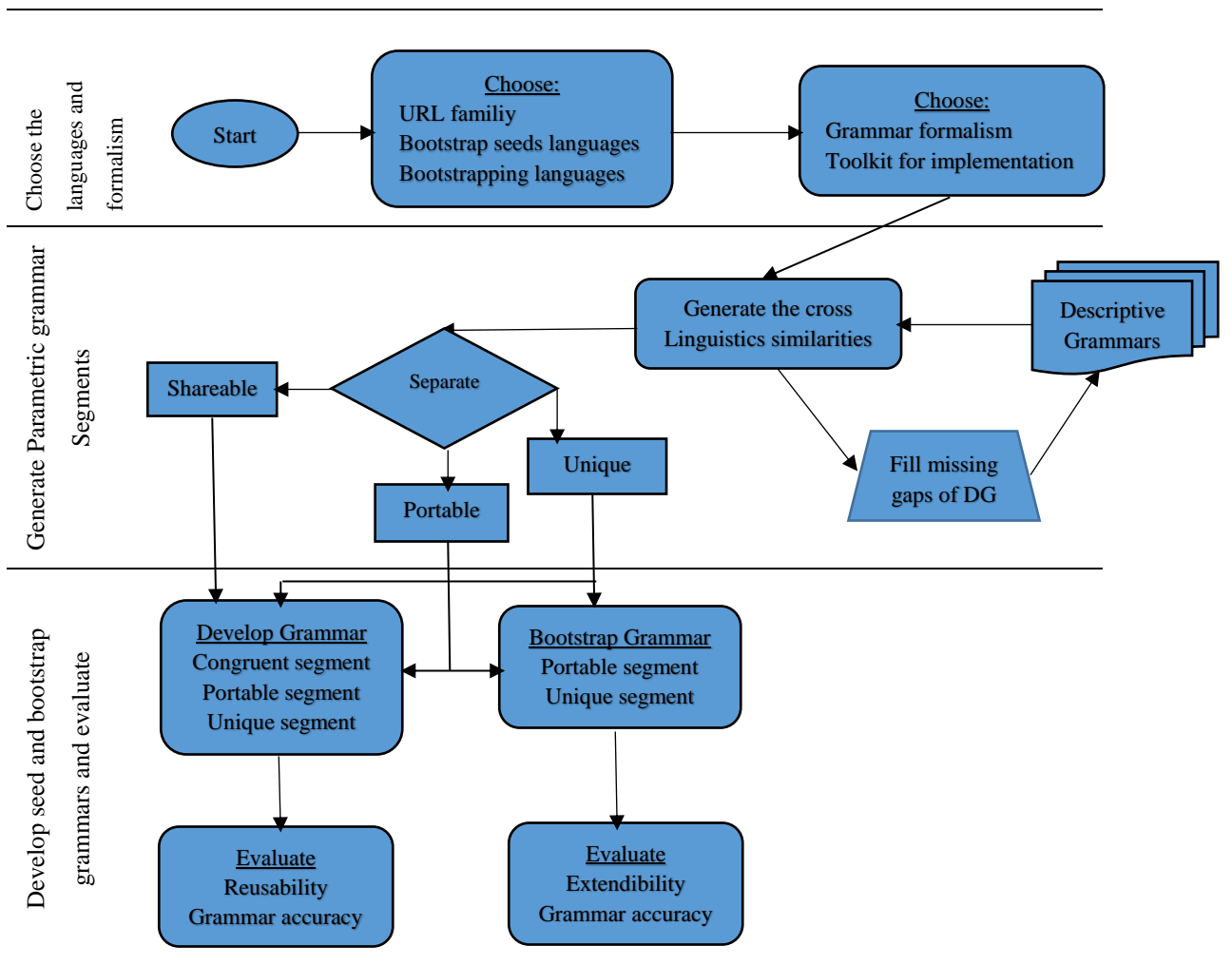


Figure 4.11 Generalized process of the approach

Step 1: identify under-resourced languages in a family

In line 1, n number of languages are selected from under-resourced languages in a specific family are entered, i number which is less than n shall be for developing the shared parameterized grammar, while the others will be for bootstrapping.

For example: in this research, in the Bantu family, the under-resourced languages chosen were Ekegusii, Kikamba and Swahili, n was three and i was two.

Step 2: identify the grammar formalism for implementation

Line 2, a choice of grammar formalism is made. For example, GF was used as the formalism and the toolkit

Step 3 Develop the cross-linguistic similarities

Lines 3 to 8, the descriptive grammar is analyzed for each language, and missing gaps are filled via language analysis and language translations by informants or experts or linguists. The cross-linguistic similarities are identified through comparative analysis then, the shareable and portable grammar are extracted. For example, one way of forming a noun phrase is by combining a noun and a determiner. The determiner (Det) is constructed from possessive (poss) and demonstrative (dem) determiners. The demonstrative can come before the noun or after the noun.

Kikamba [dem] [Noun] [Det <poss> <dem>]

Ekegusii [dem] [Noun] [Det <poss> <dem>]

The Ekegusii format was missing in the available descriptive grammar thus it was generated by linguists and informants through elicitation. In terms of cross-linguistic similarities, they share the same topology, thus form part of shareable grammar.

Bantu parametrized grammar: [dem] [Noun] [Det <poss> <dem>]

On portability, for example, to write the numeral one, ten, eleven and one hundred for cardinal or ordinal involves conjunction or disjunction of the prefix and the stem accordingly in each of the two languages. Hence the rule structure is summarized as below

```
Prefix ++/+ stem
```

However, though the pattern is similar, the morphemes are quite different thus the structure of the rule was ported.

Step 4 Develop and evaluate the congruent grammar

Lines 9-20, the shared, portable and unique grammar are developed in the grammar formalism and the reusability metrics, mainly the shareable and portability part of the grammar is expressed in terms of percentages based on the grammar formalism. The implementation is shown below

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

The parameter *isPre*, which is a boolean to allow the determiner to appear either before or after the noun. The parameter *isPron* also a boolean that finds out if the formed noun phrases come from a pronoun to allow pro-drop if used as a subject in a sentence. The generation of numeral On portability, to implement numerals one, ten, eleven and one hundred for cardinal or ordinal the following rules were used.

```
lin pot01 = mkNum1 "mwe" " yimwe" "mbee" ** {n = Sg} ; --Kikamba
lin pot01 = mkNum1 "mo" "tang'ani" ** {n = Sg} ; --Ekegusii
```

The rules followed a similar pattern of a regular expression(*mkNum1*), then stem morphemes and parameter number with value singular. The conjunction or disjunction of the prefix and the stem are implemented in each language's regular expression. The RE was unique for each language and hence formed a segment of unique grammar as illustrated below

Kikamba RE

```
mkNum1 : Str -> Str -> {s : DForm => CardOrd => Cgender => Str} =\two, second ->
{s = table {
  unit => table {NCard =>\g => Cardoneprefix g + two ;
               NOrd => \g => Ordoneprefix g + second} ;
  teen => table {NCard =>\g =>"ikomi nemo" ;
               NOrd => \g => Ordprefix g ++ "ikomi " ++ "nemo"} ;
  ten => table {NCard =>\g =>"ikomi" ;
               NOrd => \g => Ordprefix g ++ "ikomi"};
  hund => table {NCard =>\g =>"rigana erimo";
               NOrd => \g => Ordprefix g ++ "rigana erimo" } } ;
```

Ekegusii RE

```
mkNum1 : Str -> Str -> Str -> {s : DForm => CardOrd => Cgender => Str} =
  \two, twenty, second ->
  {s = table {
    unit => table {NCard =>\g => Cardoneprefix g + two ;
                  NOrd => \g => Ordprefix g ++ second} ;
    teen => table {NCard =>\g =>"ikumi na" ++ Cardoneprefix g + two ;
                  NOrd => \g => Ordprefix g ++ "ikumi na" ++ Cardoneprefix g + two} ;
    ten  => table {NCard =>\g =>"ikumi" ;
                  NOrd => \g => Ordprefix g ++ "ikumi"};
    hund => table {NCard =>\g =>"yiana " ++ twenty ;
                  NOrd => \g => Ordprefix g ++ "yiana" ++ twenty} } } ;
```

Finally, extract the metrics of sharing and portability of the parameterized grammar.

Step 5 Bootstrap and evaluate the new grammar

Lines 21-31, the descriptive grammars for bootstrapping languages are analyzed and the unique and portable grammar segments are identified and bootstrapped. For example, The NP shown in step 3 was similar to Swahili thus was part of the shareable grammar and the structure of the Swahili numeral followed a similar pattern as the case with Ekegusii and Kikamba thus formed a portable grammar segment and is illustrated below.

```
lin pot01 = mkNum1 "moja" "kwanza" ** {n = Sg} ;
```

The regular expression for the Swahili numeral was unique as illustrated below and formed part of unique grammar.

Swahili RE

```
mkNum1 : Str -> Str -> {s : DForm => CardOrd => Cgender => Str} = \two, second ->
  {s = table {
    unit => table {NCard =>\g => Cardoneprefix g + two ;
                  NOrd => \g => Ordprefix g ++ second} ;
    teen => table {NCard =>\g =>"kumi na" ++ Cardoneprefix g + two ;
                  NOrd => \g => Ordprefix g ++ "kumi na" ++ Cardoneprefix g + two} ;
    ten  => table {NCard =>\g =>"kumi" ;
                  NOrd => \g => Ordprefix g ++ "kumi"};
    hund => table {NCard =>\g =>"mia " ++ two ;
                  NOrd => \g => Ordprefix g ++ "mia" ++ two} } } ;
```

Finally, the extendibility metrics similar to the congruent grammar are extracted, plus the grammar performance via machine translation experiment. In Swahili, since this noun phrase was shared, it inherited step four implementation; after bootstrapping the

unique and portable part, the shareability and portability plus performance metrics were extracted through evaluation.

In conclusion, to apply the approach to a new Bantu language, One should start by analyzing the descriptive grammar with aim of developing parameters, regular expression structure for each POS and grammar rules in order to compare and contrast with the corresponding in the congruent grammar. The new language parameters should be compared with shared parameters in Table 4.2 while grammar rules and RE should be compared with the ones in Table 4.3. The high degree of similarity shows a corresponding shared grammar and vice versa. In non-Bantu languages, the approach may only be useful in the syntax since morphology will differ. Therefore, one needs to compare syntax parameters and structures of grammar rules in order to isolate the shareable and portable segments of the grammar.

In bootstrapping the grammar, first, define the lexicons which follow the order: regular expression, lexeme and parameter if any exist. This is followed by the actual definition of the regular expression, the low-level REs are in portable grammar since actual morphemes differ from one language to another. However, some RE will be shared like for pronouns since their arguments are supplied from lexeme definition. Finally, define the grammar rules that differ and thereafter evaluate the grammar for accuracy.

4.6 Effects of Errors on BLEU score

The BLEU score for evaluated shared and bootstrapped grammars were not high as expected due to various errors. Though, a 100% BLEU score cannot be achieved unless the development suite is the same as the test suite which was not the case here. In addition, Hovy (2007) argues that when the quality of a rule-based system improves, the BLEU score systems tend to penalize it. This is because automatic evaluation systems follow the gold standard word order while the rule-based output may not follow that order due to some rearrangements resulting in some degree of variation of the input word order which is penalized by the BLEU scorer. Therefore, in this section, the discussion will focus on some criteria of errors and how they could be remedied. The discussion shall focus on synonyms of words in the translation, pronoun pro-drop, GF complexities, morphophonological

issues in verb and the context of translation. Figure 4.10 summarizes the errors in terms of percentages.

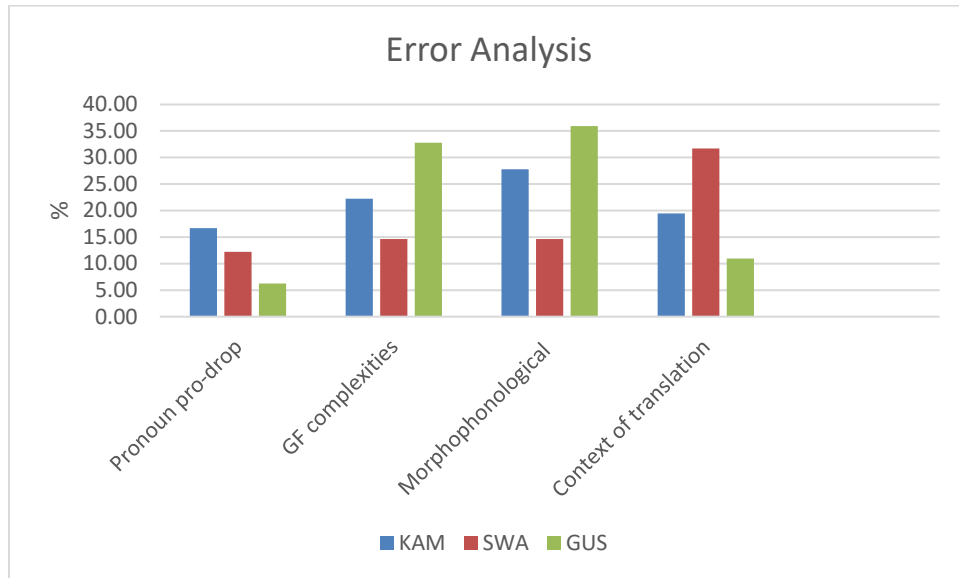


Figure 4.10 Error Categories

In terms of synonyms, the gold standard and machine translation used different lexicons for the same source word but they mean the same. For example, in Definition 4.2 in the first sentence the word “know” has been translated by the human as “nitwisi” while by the machine as “nitumanyaa” both word means “know”. Though the same meaning the choice of the synonyms has resulted in the machine translation sentence being penalized thus a BLEU score of 64.35% for the sentence. In the second sentence, the linguist translated the phrase “ in the school” using two words while the machine translation choose the infusion of the preposition to the noun. Both sentences are correct, However, the machine translation was highly penalized resulting in a 54.75% BLEU score instead of 100%. Figure 4.10 shows this category of errors contributed 13.89%, 26.83% and 14.06% for Kikamba, Swahili and Ekegusii languages which was significant thus reducing the BLEU score. Since these Bantu language lexicons were not availed to the gold standards plus the English test-suite generators, resulted was the use of synonyms and certain context of translation whose resulting words in the gold standard were not available in the GF grammar linearizations. This experimental human error was because the researcher was not

aware of the impact of such scenarios at the design stage. However, they were noted at the testing stage and had an impact on the accuracy of the grammar.

To mitigate this in the future, the researcher proposes either more than one gold standard translation be provided with all possible synonyms or the writer of the test-suite be provided with the lexicons in both source and target languages so that s/he is aware of the possible translation of the lexicons.

Definition 4.2 Synonyms errors

Source	-	we know the science on everything
Human	100.00	ithyi nitwisi sayasi iulu wa kila kindu
Machine	64.35	ithyi nitumanyaa sayasi iulu wa kila kindu
Source	-	we read three nights in the school
Human	100.00	ithyi nitusomaa iwiyo itatu vau sukulu
Machine	54.75	ithyi nitusomaa iwiyo itatu sukuluni

The pronoun pro-drop also resulted in errors, whereby either human translator did pro-drop at the beginning of the sentence like in Definition 4.3, the first case of the Kikamba language, where the machine translation did not pro-drop resulting in a 58.77% BLEU score. In the second and third cases, the opposite occurred in the Swahili and Ekegusii where the machine did pro-drop. The Swahili incurred penalization of more than 57%. In the Ekegusii language, the BLEU score is too low though beyond pro-drop other errors are contributing. Kikamba had the highest percentage of 16.67% while Ekegusii had the least at 6% as Shown in Figure 4.10 above. Therefore, this affected the overall BLEU score for the languages. The researcher proposes the pro-drop implemented for the system should be communicated to the human translator to ensure uniformity.

Definition 4.3 Pronoun pro-drop

Source	-	we bought the important newspaper from the blue shop
Human	100.00	Nitunathooie ikaseti ya vata kuma ndukani ya langi wa waiyu
Machine	58.77	ithyi nitunathooie ikaseti ya vata kuma nduka ya langi wa waiyu
Source	-	we didn't eat blood
Human	100.00	sisi hatukula damu

Machine	42.89		hatukukula	damu
Source	-		they like the rule that the books are thin	
Human	100.00	barabwo	ebanchete richiko ng	□ a ebitabu ebire ebireu
Machine	6.48		bakoancha richiko eke ebitabu bire ebireu	

GF complexities existed in two ways: Implementation of discontinuous constituents and similar tense. On the discontinuous constituents, CN was implemented with two strings *s* and *s2*, *s* to hold CN while *s2* awaits AP as per Definition 3.16 especially function AdjCN which is rule four with arguments CN and AP. This kind of implementation allows adding of a determiner in between CN and AP which is the behavior of Bantu languages as exemplified by function DetCN for NP in Definition 3.18 rule five. The recursion call of function AdjCN in higher function DetCN resulted in the interchange of two consecutive adjectives in a sentence as shown in Definition 5.3. Based on this definition, in the human translation, the two adjectives small and red followed in the same order while in the machine translation the “red “ came before “small”. This interchange was heavily penalized by the BLEU scorer as shown in Definition 4.4 where though the translation was ok, the interchange resulted in a BLEU score of 22.59%

Definition 4.4 Discontinuous constituents design

Source		small red seeds smell
Human	100.00	chintetere chinke chimbariri chigotiokerera
Machine	22.59	chintetere chimbariri chinke chigotiokerera

In the noun module, GF has two functions AdjDAP and DetDAP that takes arguments determiner plus adjective and determiner respectively and were implemented in the shared Bantu parameterized grammar though not used as exemplified below. Therefore, to mitigate the problem, the researcher proposes, an NP function as exemplified by Definition 4.5 that makes use of these two unused functions and does away with the discontinuous constituents since now an AP can be combined with a determiner and later CN added to form NP. This rule was defined in the extra abstract module of the Bantu parameterized grammar and linearization was defined in the extra module. This problem was highlighted to the author of GF as per the email in Figure D.2 in Appendix D.

```

AdjDAP det ap = { s = \\ Cgender =>det.s! Cgender ++ ap.s! Cgender !det.n;
                  n = det.n; isPre=det.isPre };
DetDAP d = { s=d.s; n=d.n; isPre=d.isPre};

```

Definition 4.5 Propose new rules

```

DapNP : DAP -> CN -> NP; -- abstract rule
DapCN dap cn= { s= dap.cn.g.cn.n ++ cn.s.dap.n }; -- in concrete module

```

There are variants of tenses in Bantu languages for example in Ekegusii grammar, the past tense has the variants immediately, near, far and remote. The verb phrase has many morphemes hence resulting in a large verb inflection table for each variant. To process all possible choices, the compiler took too long and produced no results. Thus for testing purposes, only one variant was used. This had an impact on the BLEU score since the tense used by the machine translation in a case was different from the one used by the human translator as exemplified by Figure 4.6 whereby the human translation used the distant past tense while the machine translation used the immediate past tense. The use of this different tense due to compilation complexity resulted in a 64.35% BLEU score of the machine translation. This was the second significant category that contributed to a low BLEU score, especially for Ekegusii which registered 32.81% and Kikamba at 22.22% based on Figure 4.10 above.

Semantic influenced the words chosen by the translator. Therefore, context played a role in the translation. The order of the input influenced the system, while human translation had the capability of using context in the translation based on the surrounding words. Using context in translation resulted in the gold standard having words that never existed in the lexicon definition. This is exemplified in the two sentences of Definition 4.6. In the first sentence though the lexicon definition used the word “sakafu”, but based on the context of surrounding words the translator used the word “udongo”. The choice of the noun also influenced the adjective picked due to agreement(concord). This resulted in the penalization of the BLEU score to 61.48%. In the second sentence, different words for “squeezes” results in a high penalty by scoring a mere 30.21%. This category of error was prevalent in Swahili at 31.7%, while Kikamba and Ekegusii had 19.44% and 10.94% respectively, hence impacted on the BLEU score. This could have been avoided by

providing the Bantu language lexicons to the human translator who generated the gold standard or creating more than one gold standard.

Definition 4.6 Context translation

Source	-	some big forests had wet floors and green grass
Human	100.00	misitu mingine mikubwa ilikuwa na udongo mnyevu
Machine	61.48	misitu mingine mikubwa ilikuwa na sakafu nyevu
Source	-	the wide mountain squeezes the short road
Human	100.00	mlima mpana unaibana barabara fupi
Machine	30.21	mlima mpana unafinya barabara fupi

There were morphophonological errors in particular verbs, The morphophonological rules in nouns, adjectives and quantifiers were implemented in the regular expression for each category and they worked well. However, morphophonological alternate rules in verbs were not available at the start of the research. Figures 4.3 and 4.4 in Kikamba and Ekegusii clearly demonstrate the effect of these morphophonological rules since the sentences in both Figures scores a BLEU score of 42.73 and 21.02% respectively which is quite low just due to the changes of letters in a word. This category contributed the highest impact on performance. The Ekegusii registered 35.94% which was quite high, followed by Kikamba at 27.78% and the least was Swahili at 10.94% based on Figure 4.10. To avoid this in future grammar, the researcher recommends investigation of morphophonological rules in verbs as a future study so that they can be incorporated in the grammar not only for these languages used here but other similar languages.

Based on the above discussion on errors and Figure 4.10 which summarizes each category in terms of percentages, it is evident the BLEU score was highly affected and therefore, the 4-gram BLEU score of 83.05%, 77.95% and 55.95% for Kikamba, Swahili and Ekegusii languages respectively are quite encouraging. The highest contributor to low BLEU for Kikamba and Ekegusii were the morphophonological rules and GF complexities, while for Swahili were synonyms of words and context of translation. This means the grammar did well in performance in the midst of the above limitation. Further, these grammars outperformed other rule-based grammars that have adopted the BLEU score as a measure of performance. The English-Catalan language pair reported a BLEU score of

41.52% (More, 2020), While, written Spanish to Spanish Sign Language (Porta et al., 2014) resulted in a score of 30%. In addition, a score of 25.19% was reported for the Dutch to Afrikaans rule-based system (Van & Pilon, 2009) though this was due to an introduction of external text otherwise experiment at the word level resulted in an accuracy of 71% but not based on BLEU score. Finally, translation from a Tunisian dialect to the standard Arabic language (Sghaier & Zrigui, 2020) showed a BLEU score of 55.22%. In conclusion, this performance was good, way above similar grammars in rule-based and therefore, this validates the approach as a way of accelerating the development of NLP resources and applications.

4.8 Previous studies

In comparison with previous work, the Romance (French, Italian and Spanish) languages and Scandinavian (Swedish, Norwegian, and Danish) languages showed grammar sharing of 75% and 90% at the syntax level, respectively (Ranta 2007). The Scandinavian family is in a similar range of sharing with the Bantu parameterized Grammar, while Swahili has outperformed it. However, it is important to note the latter work was quantified using rules expressed as a percentage while the former work used lines of code; thus it becomes hard to compare. The Microsoft NLP systems used 129 English Grammar rules as the sharing pivot to develop French, Spanish and German Grammar (Gamon et al. 1997). The results show that 10.1%, 10.7 %, 7.8% of the English rules were deleted and 7.8%, 8.6%, 2.3% were added for Spanish, German and French respectively at the syntax level. In our case, adding Swahili Grammar at the syntax level, no rule was added or deleted. The functionalist approach type lattice of a system (Bateman et al. 2005) showed the Grammar of Bulgarian and Russian shared 76% of the features and 72% of the systems, while Bulgarian, Czech, and Russian grammars shared 92%, 84%, 75% systems with English Grammar. A 65 rules speech translation system developed in the Regulus framework involving English, Japanese and Finnish languages (Santaholma., 2007) shared 66% of the rules and 80 rule domain-specific speech-to-speech translation systems in the same framework using three rich resourced languages (English, Japanese and Finnish) (Santaholma 2008) but Greek grammar resulted to 75% between any two pair languages (Santaholma 2008). Finally, in LinGO Grammar Matrix, Wambaya grammar

was jumpstarted with existing grammars of English, Japanese, Modern Greek and Norwegian (Bender et al. 2008). In this case, 54% of the types were shared. Only Grammar of Bulgarian and Russian with Bulgarian has performed better than The Bantu one at the syntax level. However, the remaining part of the Bantu parametrized grammar is taken care of by porting, unlike Bulgarian grammar. This being the first time statistics have been shown at the morphology level, there was no grammar to compare with. The previous studies are more focused on rich-resourced languages. Therefore, this research forms a basis for more research in under-resourced languages. The fact that the languages in this research were picked from different geographical areas and different Guthrie(1948) zones and resulting in quite high percentages implies languages in the same group and area would result in higher sharing and the generalization in different geographical areas would still significantly reduce the work of the rule-base for the grammar.

In conclusion, Tables 4.1 and 4.2 provided empirical evidence of UG, hence a firm ground for developing the grammar. Such, evidence is crucial for Bantu linguists who want to extend the theory of UG in Bantu languages. To NLP developers, these cross-linguistic similarities can be exploited in the development of shared tools which was shown by the development of the Bantu parameterized grammar resulting in significantly reduced rule-base size, in that at the morphology, 65.3% of the regular expression were shared plus 68.3% of the parameters. Furthermore, at the syntax level sharing was at 89.57%. This reduced rule-base implies less effort in development time and the number of rules required.

Therefore, in bootstrapping the Swahili grammar, this congruent grammar was inherited; hence no single effort was applied in development. The portability occurred at paradigms, parameter and syntax rules at a percentage of 15.55%, 18.75% and 8.59% respectively, which means the structures of both grammars were similar. Hence, the benefits of developing Bantu parametrized grammar were transferred to rules modification of Swahili grammar. This has demonstrated the approach of bootstrapping the development of grammar (sharing and porting) leveraging the cross-linguistic similarities significantly reduced the development, thus a faster way of scaling up grammar development for these under-resourced languages to the NLP grammar developers community.

Moreover, to the grammar evaluator, the 100 sentences Bantu test suites provide a case for evaluating similar grammar in the future.

To GF users, the GF resource library has been extended by providing three concrete grammars for Ekegusii, Swahili and Kikamba and to the policy maker, by exploiting these cross-linguistic similarities, it is easier to develop grammar resources for even less-resourced related languages thus preserving these languages.

Overall, the methodology will provide an approach for accelerating NLP resources and tools development for under-resourced languages thus reducing the language digital divide between the less and rich-resourced languages. Though the grammar had good accuracy, a limitation was noted, in the morpho-phonological errors especially, in the verb category. Therefore, the researcher recommends, linguists to develop alternate sound rules for the verbs. This will improve the accuracy greatly especially, for the Ekegusii part of the grammar

4.9 Summary

This chapter presents a discussion on comparative descriptive grammar, evaluation of the Bantu parameterized grammar and the bootstrapped Swahili grammar. It provided a generalized summary of the approach which can be used to add new languages. There is a discussion of errors that affected the performance of the grammar and finally, it places this research work in the context of previous work.

The principles and parameters are summarized and discussed, together with regular expressions and grammar rules based on cross-linguistic similarities identified in the comparative work. This work reinforces the concept of UG.

Through evaluation, the Bantu parameterized grammar shows shareability at linearization categories, parameters, paradigms and syntax rules of 100%, 68.75%, 65.3% and 89.57% respectively, while portability at paradigms, parameter plus syntax rules was at 14.29%, 18.75% and 10.43% respectively. These high shareability and portability levels demonstrated the effectiveness and efficiency of the approach in reducing the development effort. The Bantu parameterized grammar shared the rule-base between the Swahili at linearization categories, parameters, paradigms and syntax rules at 100%, 71.11%, 68.75% and 91.41% respectively..

The bootstrapping process leads to an accurate Swahili grammar of 77.95 4-gram BLEU score. The maximum effort involved defining: all categories' lexicon, 13.33% of paradigms and 12.5% of parameters and modifying 15.55%, 18.75% and 8.59% of paradigms, parameters and syntax rules respectively. Indeed, at least 68% of the parameters, paradigms and syntax rules were already catered for in the Bantu parameterized grammar.

Thus, the process has proved that adding a new Bantu grammar; one requires minimal effort due to the effectiveness and efficiency of the approach. Therefore, a generalized approach is provided with five steps. Finally, five types of errors are discussed and how they affected the performance of the grammar

Chapter 5 CONCLUSION AND RECOMMENDATION

5.1 Introduction

This chapter is an overview of the research and its achievements based on the objectives. Its primary focus is the contributions, findings and achievements of the study. In addition, the limitations of the study and the future direction of the study are given.

5.2 Overview of the Research

The research used cross-linguistic similarities between the complex morphology and less-resourced Bantu languages as leverage to build the Bantu parameterized grammar, thereby reducing the effort problem (rule-base and time) required to handcraft rules for building grammar in a multilingual ecosystem. Four objectives were used to address the challenge. To achieve the first objective of investigating the degree of similarities of the principles and parameters between the geolinguistics chosen Kikamba and Ekegusii grammars, a descriptive case study research design was used to perform an in-depth analysis of each descriptive grammar. After that, the comparative analysis research design was used to develop the shareable and portable segments of descriptive grammar. To achieve the second objective of developing an approach leveraging on the shared grammar principles and parameters of Kikamba and Ekegusii grammars to produce the Bantu parameterized, an experiment was set up in GF. The development used GF formalism and the morphology-driven approach. Each grammar function was developed and then tested using the GF regression procedure. Display of trees used Graphviz tool. Bootstrapping of the Swahili grammar to the Bantu parameterized grammar, which was the third objective, followed the same experimental setup as the Bantu parameterized grammar. The work here involved modifying similar grammar structures to suit Swahili and defining the unique grammar segment. To achieve the fourth objective of evaluating both the Bantu parameterized and bootstrapped grammars in order to demonstrate reduced effort, the grammars' shareable and portable segments were expressed as a percentage. Moreover, the grammars' precision was measured using a machine translation task where the BLEU score was obtained from Tilde software, while error metrics PER and WER were extracted from Perl scripts. The precision evaluation used machine translation of 100 sentences suite.

5.3 Achievements

The discussion on achievement is based on each specific objective.

To investigate the degree of similarity of the principles and parameters between Kikamba and Ekegusii grammars

The objective was achieved in two ways. First, through the rigorous synthesis of the literature to empirically establish: parameters, principles, regular expressions for part of speech tags and syntax rules similarities in the two languages. Additionally, linguists' performed elicitation of the grammar parts missing in the descriptive grammar references or literature. This included the comparative parameter for Kikamba adjective, a regular pattern for some prepositions fusion with a noun, the regular expression for Ekegusii numerals and subject marker plus negation marker morpheme for all gender except the first gender that is animate that was provided in the descriptive grammar for both languages.

Secondly, to empirically establish the similarities and dissimilarities between the two Bantu languages, the parameters, principles, regular expression, and syntax rules were compared, resulting in comparative descriptive grammar.

The above achievements have great significance because lacking descriptive grammar segments show gaps that linguists need to address in other related languages. Finally, the established comparative descriptive grammar laid a foundation for developing the Bantu parameterized grammar.

To develop an approach leveraging on the shared grammar principles and parameters of Kikamba and Ekegusii grammars to produce the Bantu parameterized grammar

The monolingual grammars for Kikamba and Ekegusii were used to create the Bantu parameterized grammar. The resulting grammar demonstrated a significant reduction of development effort for multilingual grammar, thereby showing the approach's effectiveness. The sharing capabilities were not only at syntax but also at the morphology level. The sharing at linearization categories, parameters, paradigms, and syntax rules was at 100%, 68.75%, 65.3% and 89.57% respectively, while portability was

exhibited at paradigms, parameters and syntax rules at 14.29%, 18.75% and 10.43% respectively. This significantly reduced the rule-base for the Bantu parameterized grammar. This is the first wide coverage of the Bantu language in the GF resource library. Additionally, it is an open resource available in the GF³⁵ repository and the principal researcher's git³⁶ account. Therefore, other researchers, especially Bantu, can utilize it for any other work. Furthermore, the grammar can be re-used to reduce the effort of creating application grammar for controlled languages besides inheriting the grammar correctness.

Machine translation tools for African low-resourced languages are very crucial for increasing online data and information consumption. The grammar can act as a machine translation tool, as shown in Figure 5.1, where a sentence is translated from English to the two Bantu languages. Figure 5.2 shows a machine translation between Kikamba and Ekegusii for the sentence “the big boy cut green grass”.

```
Lang> p -lang=Eng " the big boy cut green grass " | l
      the big boy cut green grass  --English
      omoisia omonene akanacha obonyansi -- Ekegusii
      kivisi kinene nikipatemie nyeki  --Kikamba
```

Figure 5.1 English to Bantu languages machine translation

```
Lang> p -lang=Kam " kivisi kinene nikipatemie nyeki " | pt -number=1 | l
      omoisia omonene akanacha obonyansi  -- Ekegusii
      kivisi kinene nikipatemie nyeki  --Kikamba
```

Figure 5.2 Kikamba to Ekegusii machine translation

To bootstrap Swahili grammar into the Bantu parameterized grammar

This objective demonstrates that a bootstrapping methodology using grammar engineering techniques reduces grammar development effort on a new Bantu language. Since, only 28.89%, 31.25% and 8.59% of work was done in paradigms, parameters and syntax rules respectively plus defining the Swahili lexicons. Furthermore, it demonstrates

³⁵ <https://github.com/GrammaticalFramework/gf-rgl>

³⁶ <https://github.com/kitukb/gf-rgl>

the reusability of the Bantu parameterized grammar and generalization capability to a new grammar. The process resulted in an accurate open resource, Swahili grammar, that is available for researchers' use. The grammar in GF can also be used to develop multilingual application grammars for controlled languages. Moreover, this grammar offers an opportunity for translation among Bantu languages, as demonstrated in Figure 5.3 below for the gloss “those ten beautiful and clever friends have fallen now”

```
Lang> p -lang=Swa "marafiki hao kumi wazuri wameanguka leo" |l  
abasani baria ikomi abasere bakagwire rero --Ekegusii  
anyanya aya ikumi anake nimavaluka umunthi --Kikamba  
marafiki hao kumi wazuri wameanguka leo --Swahili
```

Figure 5.3 Bantu languages machine translation

To evaluate the effectiveness and efficiency of the approach in reducing the development effort

A detailed examination of the metrics used to evaluate grammar engineering techniques in rule-based NLP tools was established to be reusability in terms of rule reuse and rule modification, development cycle and grammar performance as the key metrics. The development cycle, since it is full of approximation of time, was dropped. Therefore the other two metrics remained the most effective ones in evaluation. The reusability was demonstrated by Swahili grammar in that 100%, 71.11%, 68.75% and 91.41% of linearization categories, paradigms, parameters and syntax rules respectively were shared, resulting in the highly accurate grammar with a BLEU score of 77.95%. The implication is that bootstrapping has significantly reduced the effort needed to create accurate grammar for low-resourced languages.

Finally, the treebank created using the 100 sentences is another achievement that will serve as a test-suite for evaluating any other Bantu language that will be bootstrapped to the Bantu parameterized grammar.

5.4 Contribution

The research has provided insights into this approach for bootstrapping the development of grammar through developing the Bantu parameterized grammar and

reusing it to bootstrap a new similar grammar. This is useful for accelerating NLP grammar development for under-resourced languages by reducing development efforts (Corley and Gioia., 2011). The contribution was two-fold: theoretical and language technology resources.

5.4.1 Theoretical Contribution

First, complex morphology involves different morphemes in a word with a specific meaning. This increases the work of grammar development. The complex morphology in Bantu languages was exhibited by prefixing, infixing and suffixation of the part of speech tags morphemes, nominal genders that influence agreement (concord), pro-drops of pronouns, infusion of the preposition to nouns, and different order of noun phrase constituents, especially categories between adjectives and nouns. This would significantly increase multilingual grammar development efforts and greatly accelerate the development of NLP resources.

On the contrary, The research has contributed by providing an efficient and effective approach that can be used to bootstrap grammar development for under-resourced languages, thus reducing the effort required in ordinary settings. The approach involves two main steps developing the shared parametrized grammar (the bootstraps seed) based on the cross-linguistic similarities of chosen under-resourced languages and then bootstrapping language-specific grammar leveraging on the congruent grammar. Based on this research, using the approach to develop the shared Bantu parameterized grammar resulted in a remarkable reduction of development effort at syntax and morphology levels. The grammar shareability at linearization categories, parameters, paradigms, and syntax rules was at 100%, 68.75%, 65.3%, and 89.57%, respectively, while portability was 14.29%, 18.75% and 10.43% in paradigms, parameter and syntax rules respectively. Therefore, to bootstrap Swahili grammar for generalization purposes, the work done as illustrated in Figure 5.4 involved lexicon definition and developing 28.89%, 31.25% and 8.59% of the paradigms, parameters and syntax rules, respectively which is a significant reduction of the Swahili rule-base. Thus decreasing even the time needed to develop it. Hence, this approach has proved to be efficient and effective in accelerating the development of accurate grammars for low-resourced Bantu languages by significantly reducing the effort needed for such work.

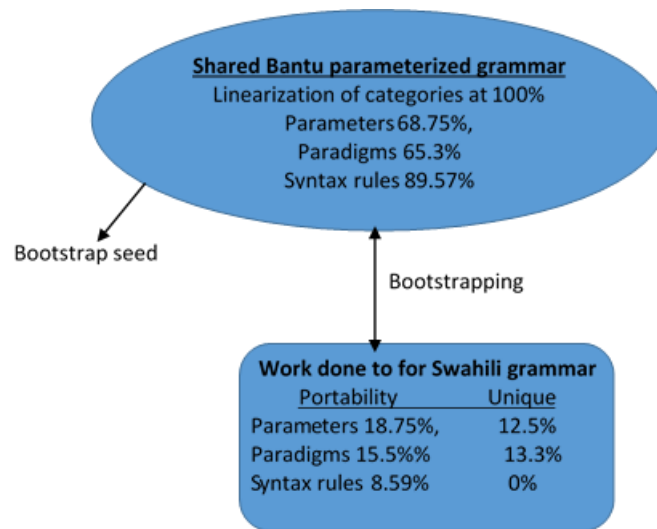


Figure 5.4 Bootstrapping Swahili

The steps are summarized below and more explanation was provided in section 4.5

- Identify under-resourced languages in a family
- Identify the grammar formalism for implementation
- Develop the cross-linguistic similarities
- Develop and evaluate the congruent grammar
- Bootstrap and evaluate the new grammar

Secondly, The research has contributed by extending GF reusability by providing a standardized Bantu parameterized grammar (Bantu functor). The standardization has been done at naming conventions, for example, the definition of genders, parameter descriptions, phenomena analysis and rules definitions. The uniformity enables Bantu grammar writers to borrow the experience accrued from developing the three grammars to accelerate new similar grammar development. Overall, it will lead to uniform Bantu grammar development, thus easy maintenance of the rule-base. We have also defined two unique functions for Bantu languages, the aforementioned quantifier and plural polite imperative request, at the abstract and concrete syntaxes. Accordingly, GF has been enabled to handle Bantu family grammars and their unique functions and we have provided a standard way of defining a new wide coverage of Bantu grammar.

The rigorous literature synthesis of Ekegusii and Kikamba descriptive grammars to derive the comparative descriptive grammar leads to empirical identification of the cross-linguistic similarities and common principles signifying grammar sharing or porting points. These grammars similarities and common principles reinforced and validated the Universal Grammar Theory (Bender et al., 2008). Where the descriptive grammar had gaps because of no reference material, the elicitation method was applied to generate it, especially in the numeral, preposition fusion, and subject marker morpheme of the verb. Therefore, the literature review contributed to the body of knowledge by providing descriptive grammar where it was missing and empirically establishing the grammars' similarities and common principles.

5.4.2 Language technology resource tools.

This study's main contribution is creating open-source resource computational grammars for Swahili, Kikamba and Ekegusii. These are the first Bantu languages to be added to GF and have taken care of various genders available and complex concatenative morphology using a standardized way to ensure faster development of similar languages in the future by borrowing the strategies used. The grammars open room for developing domain grammars such as ³⁷multilingual web gadgets, ³⁸natural-language interfaces and ³⁹dialogue systems. GF has been a multilingual ecosystem; therefore, these grammars act as translation systems; furthermore, enabling Bantu to Bantu languages machine translation, as shown in Figure 5.3. If the dictionary is expanded, they (grammars) can be used to generate corpus (a rare commodity for these low-resourced languages) to experiment with data-driven approaches. Finally, the grammars are a significant milestone towards creating a standard Basic Language Resource Kit (BLARK) (Krauer, 2003) for Kenyan Bantu languages.

The research provides a gold standard test-suite made of 100 English sentences but also transformed into 100 abstract syntax trees. This can be used to evaluate any other Bantu language that will be bootstrapped into GF, checking whether the coverage stated in

³⁷ <http://cloud.grammaticalframework.org/minibar/minibar.html>

³⁸ <https://cth.altocumulus.org/~hallgren/Alfa/Tutorial/GFplugin.html>

³⁹ <https://www.youtube.com/watch?v=1bfaYHWS6zU>

Table 3.14 has been achieved thus enabling comparative studies among Bantu languages. Furthermore, since it was established that the test-suite covers morpho-phonological issues. Once implemented in these grammars or future Bantu grammars, then it can be used to test their accuracy. In addition, the test-suite can be used to model and analyze local language translation for these under-resourced languages, especially for the Bantu languages as shown in Figure 5.3.

5.5 Conclusion

The conclusion presented in this section summarizes the two major findings of the work done and reported in this thesis.

Leveraging on the cross-linguistic similarities of principles and parameters significantly reduces multilingual grammar's development effort.

Based on the first objective, the research established high cross-linguistic similarities between Ekegusii and Kikamba languages from the rigorous review of descriptive grammars. These similarities were utilized to develop the Bantu parameterized grammar in the GF platform using the grammar engineering methodologies of sharing and porting. 89.57% of rules were shared, while 10.43% were modified for both grammars at the syntax level, thus fulfilling the second and fourth objectives. This means 89.75% of the rule-based development effort was reduced while modifying rules; the benefit accrued in creating the first grammar 10.43% rules was transferred to the second grammar. Grammar sharing was 100% at linearization of categories, 65.3% at paradigms and 68.75% at parameters at the morphology level. The sharing implies that the rule-base was defined once hence reducing the development effort by the same percentage. This is a significant reduction of the development effort. Based on the Bantu parameterized grammar work, a new Bantu grammar would only need 20.41% and 12.5% unique work on paradigms and parameters respectively, plus modification of paradigms, parameters and syntax rules at 14.29%.18.75% and 10.43% respectively, in addition to defining lexicon. Consequently, having the development effort at linearization of categories, paradigms, parameters and syntax rules already taken care of at a percentage of 100%, 65.3%, 68.75% and 89.57% respectively is a significant reduction of development effort. This means that exploiting the

cross-linguistic similarities will accelerate grammar development for these low-resourced Bantu languages and lead to accurate grammar.

Leveraging on congruent grammar to bootstrap a similar grammar takes less effort.

This research used the Swahili language as a testbed for the generalization of the research. An accurate Swahili grammar resulted after bootstrapping it to the Bantu parameterized grammar, based on the third and fourth objectives. Swahili's effort involved defining 13.33% and 12.5% of paradigms and parameters respectively and modifying 15.55%, 18.75% and 8.59% of paradigms, parameters and rules respectively and finally, defining the lexicons. This significantly reduced the work since 100% of categories linearization, 71.11% of paradigms, 68.75% of parameters and 91.41% of syntax rules were already done. It would, therefore, take a short duration to develop the grammar using the bootstrap approach compared to developing monolingual grammar by virtual of reduced effort. The implication is that this innovative way can be used to develop computational grammar for under-resourced languages with less effort and short development time as opposed to developing from scratch.

5.6 Recommendation

Since this bootstrapping methodology has proved to be effective in reducing effort for developing multilingual grammar, the research recommends the future direction be to bootstrap other Bantu languages apart from those used in the study.

Another recommended direction would be developing a congruent grammar for other family languages to enable scaling up NLP resources for these under-resourced languages.

References

- AlAnsary, S., (2014). Interlingua-based machine translation systems: UNL versus other interlinguas. In *11th International Conference on Language Engineering, Ain Shams University, Cairo, Egypt*.
- Angelov, K. (2011). The mechanics of the Grammatical Framework. Doctoral Dissertation, Chalmers University, Sweden
- Alshawi, H., Carter, D., Gambäck, B. & Rayner, M., (1992). Swedish-English QLF translation. *The Core Language Engine*, pp.277-309.
- Antony, P. J., (2013). Machine translation approaches and survey for Indian languages. *International journal of Computational Linguistics and Chinese Language Processing* 18.1 pp. 47-78.
- Ashton, E.O., (1947). *Swahili grammar: Including intonation*. (2nd ed). Longmans.
- Austin, P.K., (2001). Lexical functional grammar. *International Encyclopedia of the Social and Behavioral Sciences*, pp.8748-8754.
- Bateman, J.A., (1997). Enabling technology for multilingual natural language generation: The KPML development environment. *Natural Language Engineering*, 3(1), pp.15-55.
- Bateman, J.A., Kruijff-Korbayová, I. & Kruijff, G.J., (2005). Multilingual resource sharing across both related and unrelated languages: An implemented, open-source framework for practical natural language generation. *Research on Language and Computation*, 3(2-3), pp.191-219.
- Basweti, N.O., (2005). A morphosyntactic analysis of agreement in Ekegusii in the minimalist program. Masters Dissertation. Nairobi university. Kenya
- Beal, Joan C.(2010) *Introduction to regional Englishes*. Edinburgh University Press.
- Bender, E.M., Flickinger, D. & Oepen, S., 2008. Grammar engineering for linguistic hypothesis testing. In *Proceedings of the Texas Linguistics Society X conference: Computational linguistics for less-studied languages* (pp. 16-36).
- Bender, E.M., Flickinger, D. & Oepen, S., (2002). The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the 2002 workshop on Grammar engineering and evaluation-Volume 15* (pp. 1-7). Association for Computational Linguistics.
- Bender, E. M. (2009, March). Linguistically naïve!= language independent: Why NLP needs linguistic typology. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?* (pp. 26-32).
- Bitutu, T. (1991). The syntactic patterns of code switching in Ekegusii-English. Nairobi: Unpublished M.A. Thesis. Kenyatta University.
- Bojar, O., (2011). Analyzing error types in English-Czech machine translation. *The Prague Bulletin of Mathematical Linguistics*, 95, pp.63-76.
- Bröker, N., (2000). The use of instrumentation in grammar engineering. In *Proceedings of the 18th conference on Computational linguistics-Volume 1* (pp. 118-124). Association for Computational Linguistics.
- Butt, M., Dyvik, H., King, T.H., Masuichi, H. & Rohrer, C., (2002). The parallel grammar project. In

COLING-02: Grammar Engineering and Evaluation.

- Cambria, E. & White, B., (2014). Jumping NLP curves: A review of natural language processing research. *IEEE Computational intelligence magazine*, 9(2), pp.48-57.
- Camilleri, J.J. *A computational grammar and lexicon for Maltese*. Master's thesis, Chalmers University of Technology. Gothenburg, Sweden, 2013.
- Carr, M. & Verner, J., (1997). Prototyping and software development approaches. *Department of Information Systems, City University of Hong Kong, Hong Kong*, pp.319-338.
- Callison-Burch, C., Osborne, M., & Koehn, P. (2006, April). Re-evaluating the role of BLEU in machine translation research. In 11th conference of the european chapter of the association for computational linguistics (pp. 249-256).
- Chelliah, S.L., (2001). The role of text collection and elicitation in linguistic fieldwork. *Linguistic fieldwork*, pp.152-165.
- Chomsky, N. (1981). *Lectures on government and binding*. Foris.
- Chomsky, N. (1981b), Principles and parameters in syntactic theory. In N. Hornstein and D. Lightfoot (eds.), *Explanations in Linguistics*. London: Longman
- Corley, K.G. and Gioia, D.A., (2011). Building theory about theory building: What constitutes a theoretical contribution? *Academy of Management Review*, 36(1), pp.12-32.
- Costa-Jussa, M. R. & Fonollosa, J.A., (2015). Latest trends in hybrid machine translation and its applications. *Computer Speech & Language*, 32(1), pp.3-10.
- Costa, Â., Ling, W., Luís, T., Correia, R. & Coheur, L., (2015). A linguistically motivated taxonomy for machine translation error analysis. *Machine Translation*, 29(2), pp.127-161
- Creswell, J., (2009). *Research design, qualitative, quantitative and mixed methods approaches*. Sage Publications.
- Cook, V. & Newson, M. (2014). *Chomsky's universal grammar: An introduction*. John Wiley & Sons.
- Cunningham, H. (1999). A definition and short history of language engineering. *Natural Language Engineering*, 5(1), pp.1-16.
- Curry, H.B. (1961). Some logical aspects of grammatical structure. *Structure of language and its mathematical aspects*, 12, pp.56-68.
- Dalrymple, N. (2001). *Lexical functional grammar*. Academic Press.
- Daniel, W. (2003) A survey of bootstrapping techniques in natural language processing.
<https://www.eecis.udel.edu/~vijay/fall13/snlp/lit-survey/Bootstrapping.pdf>
- Dąbrowska, E. (2015). What exactly is Universal Grammar, and has anyone seen it?. *Frontiers in psychology*, 6, 852.
- Debusmann, R. & Kuhlmann, M. (2010). Dependency grammar: Classification and exploration. In *Resource-adaptive cognitive processes* (pp. 365-388). Springer.
- Debusmann, R. (2000). An introduction to dependency grammar. *Hausarbeit für das Hauptseminar Dependenzgrammatik SoSe*, 99, 1-16.

- Deen, K.U.D.S.U. (2002). *The acquisition of Nairobi Swahili: The morphosyntax of inflectional prefixes and subjects* (Doctoral dissertation, University of California, Los Angeles).
- Demuth, K. (2000). Bantu noun class systems: Loan word and acquisition evidence of semantic productivity. *Classification systems*, pp.270-292.
- De Pauw, G., De Schryver, G. & Wagacha, P. (2009a). A corpus-based survey of four electronic Swahili–English bilingual dictionaries. *Lexikos*, 19(1).
- De Pauw, G., Wagacha, P. W., & de Schryver, G. M. (2009, March). The SAWA corpus: a parallel corpus English-Swahili. In *Proceedings of the First Workshop on Language Technologies for African Languages* (pp. 9-16).
- De Pauw, G. & Wagacha, P.W. (2007). Bootstrapping morphological analysis of Gikuyu using unsupervised maximum entropy learning. In *In Proceedings of the eighth INTERSPEECH conference*.
- De Pauw, G. & De Schryver, G.M. (2008). Improving the computational morphological analysis of a Swahili corpus for lexicographic purposes. *Lexikos*, 18(1).
- De Pauw, G., Wagacha, P.W. & De Schryver, G.M. (2011). Towards english-Swahili machine translation. In *Research Workshop of the Israel Science Foundation*.
- Détrez, G. & Ranta, A. (2012) Smart paradigms and the predictability and complexity of inflectional morphology. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 645-653). Association for Computational Linguistics.
- Di Garbo, F. (2014). *Gender and its interaction with number and evaluative morphology: An intra-and intergenealogical typological survey of Africa* (Doctoral dissertation, Department of Linguistics, Stockholm University).
- Dirven, R. e Dirven, R. unter Radden, G. & Radden, G. (Eds.). (1982). *Issues in the theory of universal grammar* (Vol. 196). Gunter Narr Verlag.
- Dorr, B. J., Hovy, E. H. & Levin, L.S. (2004). Machine translation: Interlingual methods. *Encyclopedia of Language and Linguistics* (2nd ed.) ms. 939, Brown, Keith (ed.),
- Elwell, R.O. (2006). Finite state methods for Bantu verb morphology. *Proceedings of the Texas Linguistics Society X, Austin*
- Gali, N., Mariescu-Istodor, R., Hostettler, D., & Fränti, P. (2019). Framework for syntactic string similarity measures. *Expert Systems with Applications*, 129, 169-185.
- Gamon, M., Lozano, C., Pinkham, J. & Reutter, T. (1997). Practical experience with grammar sharing in multilingual NLP. *From Research to Commercial Applications: Making NLP Work in Practice*.
- Getao, K.W. & Miriti, E.K. (2006). Computational modelling in Bantu Language. *Special topics in computing and ICT research: Advances in Systems Modelling and ICT Applications*.
- Getao, K. & Miriti, E. (2006). Creation of a speech to text system for Swahili. In *5th World Congress of African Linguistics*.
- Gibbs, G.R. (2007). The nature of qualitative analysis. *Analyzing qualitative data*, pp.1-9.
- Ghilic-Micu, B., Mircea, M. & Stoica, M. (2011). Knowledge based economy–technological perspective:

- Implications and solutions for agility improvement and innovation achievement in higher education. *Amfiteatru Economic Journal*, 13(30), pp.404-419.
- Guthrie, M. (1948). *The classification of the Bantu languages*. Oxford Univ. Press.
- Güldemann, Tom. 1999. The ka-possessive in Southern Nguni. *Journal of African Languages and Linguistics* 20. 157–184.
- Güldemann T.(2003), Grammaticalization. In *The Bantu language* (pp. 182-194). Routledge.
- Hammarström, H. & Ranta, A. (2004). Cardinal numerals revisited in GF. In *Workshop on Numerals in the World's Languages, Leipzig, Germany*.
- Hovy, E. (2007). Investigating why BLEU penalizes non-statistical systems. *Proceedings of the eleventh MT Summit*.
- Henderson, A.R. (2005). The bootstrap: A technique for data-driven statistics. Using computer-intensive analyses to explore experimental data. *Clinica chimica acta*, 359(1-2), pp.1-26.
- Hinnebusch, T. J. (2007). *Prefixes, sound change, and subgrouping in the coastal Kenyan Bantu languages* (Doctoral dissertation, UMI Ann Arbor).
- Hurskainen, A. (2008). SALAMA dictionary compiler: A method for corpus-based dictionary compilation. *Institute for Asian and African Studies, Box, 59*.
- Hurskainen, A. (1992). A two-level computer formalism for the analysis of Bantu morphology. *Nordic journal of African studies*, 1(1), pp.87-119.
- Hyman, L.M. (1979). Aghem grammatical structure: With special reference to noun classes, tense-aspect and focus marking.
- Ibrahim, M. H. (2014). *Grammatical gender: Its origin and development* (Vol. 166). Walter de Gruyter.
- Indurkha, N. & Damerau, F. J. (2010). Natural language generation David D. McDonald. In *Handbook of Natural Language Processing, Second Edition* (pp. 137-155). Chapman and Hall/CRC.
- Intelligence, G.S.M.A. (2016). *The mobile economy Africa 2016*. GSMA.
- Iribemwangi, P. I. (2010). Swahili phonology and pronunciation guidelines.
- Jacobson, P. (1987). Generalized phrase structure grammar.
- Jäger, G. & Rogers, J. (2012). Formal language theory: refining the Chomsky hierarchy. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1598), pp.1956-1970.
- Jones, R., McCallum, A., Nigam, K. & Riloff, E. (1999). Bootstrapping for text learning tasks. In *IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications* (Vol. 1, No. 7).
- Kahane, S. (2006). On the status of phrases in head-driven phrase structure grammar: Illustration by a totally lexical treatment of extraction.
- Kameyama, M. (1988) June. Atomization in grammar sharing. In *Proceedings of the 26th annual meeting on Association for Computational Linguistics* (pp. 194-203). Association for Computational Linguistics.

- Kao, A. H. (2009). *Montague grammar*. Technical Report EECS 595, University of Michigan, 2004. 64.
- Kaviti, L.K. (2004). A minimalist perspective of the principles and parameters in Kikamba morpho-syntax. *Doctoral Dissertation, University of Nairobi*.
- Katamba, Francis. 2003. Bantu nominal morphology. In Derek Nurse & G´erard Philippson (eds.), *The Bantu languages*, 103–120. London: Routledge.
- Khegai, J. & Ranta, A. (2004). Building and using a Russian resource grammar in GF. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 38-41). Springer, Berlin, Heidelberg.
- Kyallo, W. (2016). *Kamusi pevu ya Kiswahili* Vide Muwa publishers
- Kihm, A. (2002). What's in a noun: Noun classes, gender, and nounness. *Ms. Université Paris, 7*.
- Kim, R., Dalrymple, M., Kaplan, R. M., King, T. H., Masuichi, H., & Ohkuma, T. (2003). Multilingual grammar development via grammar porting. In *ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development* (pp. 49-56).
- Kioko, A. N., Njoroge, M. C. & Kuria, P. M. (2012). Harmonizing the orthography of Gikuyu and Kikamba. In *book harmonization and standardization of Kenyan languages, orthography and other aspects*, Ogechi, O.N.; Odour, N.; Iribemwangi, P. the centre for advanced studies of African society: Cape Town, South Africa Volume 1, pp 39-63.
- Kituku, B., Wagacha, P., De Pauw, G. (2011). Memory based approach to name entity. *Proceedings of Human Language Computer Technology Conference*. Alexandria- Egypt
- Keklik, O., Tuglular, T., & Tekir, S. (2019). Rule-based automatic question generation using semantic role labeling. *IEICE TRANSACTIONS on Information and Systems*, 102(7), 1362-1373.
- Kituku, B., Musumba, G., Wagacha, P. (2015). Kamba part of speech tagger using memory-based approach. *International Journal on Natural Language Computing*. 4(2),pp 43-53.
- Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 388-395).
- Kothari, C.R. (2004). *Research methodology: Methods and techniques*. New Age International.
- Komenda, S., Maroko, G. M., & Ndung'u, R. W. (2013). The morphophonemics of vowel compensatory lengthening in Ekegusii. *International Journal of Education and Research*
- Kracht, M. (2012). Compositionality in Montague grammar. *Edouard Machery und Markus Werning Wolfram Hinzen, editor, Handbook of Compositionality*, pp.47-63.
- Krauwer, S. (2003). The basic language resource kit (BLARK) as the first milestone for the language resources roadmap. *Proceedings of SPECOM 2003*, pp.8-15.
- Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (Vol. 10, No. 8, pp. 707-710).
- Levine, R.D. & Meurers, W.D. (2006). Head-driven phrase structure grammar: Linguistic approach, formal foundations, and computational realization. *Encyclopedia of language and linguistics*, 2.

- Lewis, M.P. (2009). *Ethnologue: Languages of the world* sixteenth edition. Dallas, Tex.: SIL International. Online version: <http://www.ethnologue.com>.
- Lipps, J. (2011). XSMA: A finite-state morphological analyzer for Swahili: Multilingual grammar development via grammar porting. In *ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development* (pp. 49-56).
- Ljunglöf, P. (2004). Expressivity and complexity of the Grammatical Framework. Doctoral dissertation, Chalmers University, Sweden, 2004.
- Marten, L., Kula, N. C., & Thwala, N. (2007). Parameters of morphosyntactic variation in Bantu 1. *Transactions of the philological society*, 105(3), 253-338.
- Marten, L. (2013). Formal syntax and African languages: Dynamic syntax case studies of Swahili, Bemba, and Hadiyya. *SOAS Working Papers in Linguistics*, 16, pp.195-213.
- Jurafsky, D. and Martin, J.H. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall.
- Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K. & Wilks, Y. (2002). Architectural elements of language engineering robustness. *Natural Language Engineering*, 8(2-3), pp.257-274.
- Moré, È. G.(2020) Improvement of a rule-based machine translation system: random corpus extraction, EN. machine translation, 389, 394. Master thesis, Universitat autonoma de Barcelona.
- Mbuvu, M. K. (2005). The syntax of Kikamba noun modification. Unpublished Masters Dissertation, University of Nairobi.
- McIntosh, B. G. (1968). The eastern Bantu peoples. *Zamani: A Survey of East African History*, 198-215.
- Muhirwe, J. (2007). Towards human language technologies for under-resourced languages. *Strengthening the Role of ICT in Development*, 38.
- Müller, S. (2001). An introduction to head-driven phrase structure grammar.
- Mose, E.G. (2012). *The structure and role of the determiner phrase in Ekegusii: a minimalist approach* (Doctoral dissertation, Kenyatta University).
- Munyao, K. M. (2006). The morphosyntax of Kikamba verb derivations: A minimalist approach. *Unpublished master's thesis. The University of Nairobi, Nairobi. Kenya.*
- Mutiga, J.M. (2002) The Tone System of Kikamba: A Case Study of Mwingi Dialect. Doctoral Dissertation, University of Nairobi, Kenya.
- Mwangi, P.W., Njoroge, M. C. and Mose, E. G. (2013). Harmonizing the orthographies of Bantu languages: the case of Gikũyũ and Ekegusii in Kenya. *The University of Nairobi Journal of Language and Linguistics*, 3, pp.108-122.
- Mwau, J. (2006). Kikamba Dictionary.
- Nadkarni, P. M., Ohno-Machado, L. and Chapman, W.W., 2011. Natural language processing: An introduction. *Journal of the American Medical Informatics Association*, 18(5), pp.544-551.

- Ng'ang'a, W. (2012). Building Swahili resource grammars for the grammatical framework. *Shall We Play the Festschrift Game? Essays on the Occasion of Lauri Carlson's 60th Birthday*, pp.215-226.
- Ng'ang'a, W. (2005). Word sense disambiguation of Swahili: Extending Swahili language technology with machine learning.
- Ngau, P. & Kumssa, A. (2004). *Research design, data collection, and analysis: a training manual* (No. 12). United Nations Centre for Regional Development, Africa Office.
- Nivre, J. (2005). Dependency grammar and dependency parsing. *MSI report*, 5133(1959), pp.1-32.
- Njogu, K. & Nganje, D.K. (2006). *Swahili kwa vyuo vya ualimu*. Jomo Kenyatta Foundation.
- Novello, A. & Callaway, C.B. (2003). Porting to an Italian surface realizer: A case study. In *Proceedings of the 9th European Workshop on Natural Language Generation (ENLG-2003) at EACL 2003*.
- Obiero, J.O. (2008). *Evaluating language revitalization in Kenya: The contradictory face and place of the local community factor*.
- Ogechi, N. O. (2003). On language rights in Kenya. *Nordic Journal of African Studies*, 12(3), pp.19-19.
- Onkwani, E. (2011). *A Morphophonemic study of Ekegusii nominal derivation and pluralization* (Master dissertation).
- Omar, A. & Alotaibi, M. (2017). Geographic location and linguistic diversity: The use of intensifiers in Egyptian and Saudi Arabic. *International Journal of English Linguistics*, 7(4), pp220.
- Ombui, E. & Wagacha, P. (2014) Interlingua plus machine translation approach for local languages: Ekegusii & Swahili. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages* (pp. 68-72).
- Ongarora, D. O. (2008). Bantu morphosyntax: A study of Ekegusii. *Doctoral Dissertation*, Jawaharlal Nehru University.
- Osinde, K. N. (1988). Ekegusii morphophonology: an analysis of the major consonantal process. Masters Dissertation. University of Nairobi.
- Otiso, K, Z. (2008). The morphosyntactic analysis of Ekegusii verb derivative in the minimalist program. Masters Dissertation Kenyatta University.
- Paikens, P. & Gruzitis, N. (2012). May. An implementation of a Latvian resource grammar in Grammatical Framework. In *LREC* (pp. 1680-1685).
- Porta, J., López-Colino, F., Tejedor, J., & Colás, J. (2014). A rule-based translation from written Spanish to Spanish Sign Language glosses. *Computer Speech & Language*, 28(3), 788-811.
- Pereira, F. C. & Warren, D. H., 1980. Definite clause grammars for language analysis: A survey of the formalism and a comparison with augmented transition networks. *Artificial intelligence*, 13(3), pp.231-278
- Perfetti, C.A. & Liu, Y. (2005). Orthography to phonology and meaning: Comparisons across and within writing systems. *Reading and Writing*, 18(3), pp.193-210.

- Pickvance, C. (2001). Four varieties of comparative analysis. *Journal of Housing and the Built Environment*, 16(1), 7-28. Retrieved November 11, 2020, from <http://www.jstor.org/stable/41107161>
- Pickvance, C. (2005). The four varieties of comparative analysis: The case of environmental regulation.
- Pretorius, L. & Bosch, S. (2009). Exploiting cross-linguistic similarities in Zulu and Xhosa computational morphology. In *Proceedings of the First Workshop on Language Technologies for African Languages* (pp. 96-103). Association for Computational Linguistics.
- Pretorius, L., Marais, L. & Berg, A. (2017). A GF miniature resource grammar for Tswana: Modelling the proper verb. *Language Resources and Evaluation*, 51(1), pp.159-189.
- Ranta, A., El Dada, A. and Khagai, J. (2009). The GF resource grammar library. *Linguistic Issues in Language Technology*, 2(2).
- Ranta, A. (2011). *Grammatical framework: Programming with multilingual grammars* (Vol. 173). CSLI Publications, Center for the Study of Language and Information.
- Ranta, A. (2007.) Modular grammar engineering in GF. *Research on Language and Computation*, 5(2), pp.133-158.
- Ranta, A. (2006). Type theory and universal grammar. *Philosophia Scientia. Travaux d'histoire et de philosophie des sciences*, (CS 6), pp.115-131.
- 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>
- Ranta, A., Angelov, K., Gruzitis, N., & Kolachina, P. (2020). Abstract syntax as interlingua: Scaling up the grammatical framework from controlled languages to robust pipelines. *Computational Linguistics*, 46(2), 425-486.
- Rayner, M., Carter, D. & Bouillon, P. (1996). Adapting the core language engine to French and Spanish. *arXiv preprint cmp-lg/9605015*.
- Rayner, M., Hockey, B. A., James, F., Bratt, E. O., Goldwater, S., & Gawron, M. (2000). Compiling language models from a linguistically motivated unification grammar. *arXiv preprint cs/0006021*.
- Reichenbach, H., 1947. The tenses of verbs. *Time: From Concept to Narrative Construct: a Reader*, pp.1-12.
- Rugemalira, J.M., 2007. The structure of the Bantu noun phrase. *NC Kula and L. Marten,(eds.)*.
- Santaholma, M. E. (2005). Linguistic representation of Finnish in the medical domain spoken language translation system. *TALN-RECITAL 2005, Traitement Automatique des Langues Naturelles*, 605-614.
- Santaholma, M.E. (2007). Grammar sharing techniques for rule-based multilingual NLP systems. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA)*.
- Santaholma, M. E., (2010). *Efficient development of grammars for multilingual rule-based applications* (Doctoral dissertation, University of Geneva).
- Santaholma, M. (2008). Multilingual grammar resources in multilingual application development. In *Proceedings of the Workshop on Grammar Engineering Across Frameworks* (pp. 25-32). Association for Computational Linguistics.

- Shaalán, K., 2010. Rule-based approach in Arabic natural language processing. *The International Journal on Information and Communication Technologies (IJICT)*, 3(3), pp.11-19.
- Sghaier, M. A., & Zrigui, M. (2020). Rule-based machine translation from tunisian dialect to modern standard arabic. *Procedia Computer Science*, 176, 310-319.
- Steedman, M. (1993). Categorical grammar. *Lingua*, 90(3), pp.221-258.
- Shaw, M. & Garlan, D. (1996). *Software architecture* (Vol. 101). Prentice Hall.
- Socher, R. (2014). Recursive deep learning for natural language processing and computer vision. Stanford University.
- Strauss, A. & Corbin, J., 1998. *Basics of qualitative research techniques*. Sage publications
- Tanaka, T. (1991). Definite-clause set grammars: a formalism for problem solving. *The Journal of Logic Programming*, 10(1), pp.1-17.
- Trudgill, P. & Hannah, J. (2008). *International English: A guide to varieties of Standard English*. Routledge.
- Van der Wal, J. (2015). Evidence for abstract Case in Bantu. *Lingua*, 165, pp.109-132.
- Van Huyssteen, G. B., & Pilon, S. (2009). Rule-based conversion of closely-related languages: a Dutch-to-Afrikaans convertor.
- Vilar, D., Xu, J., Luis Fernando, D.H. & Ney, H. (2006). Error analysis of statistical machine translation output. In *LREC* (pp. 697-702).
- Varile, G. B., Cole, R., Cole, R.A., Zampolli, A., Mariani, J., Uszkoreit, H. and Zaenen, A. (eds.). (1997). *Survey of the state of the art in human language technology* (Vol. 13). Cambridge university press.
- Wagner, G. (1970). *The Bantu of Western Kenya: With special reference to the Vugusu and Logoli*. Oxford University Press.
- Wamalwa, E. W. & Stephen, O. (2013). Language endangerment and language maintenance: Can endangered indigenous languages of Kenya be electronically preserved?
- Wang, Y. (2009). A formal syntax of natural languages and the deductive grammar. *Fundamenta Informaticae*, 90(4), pp.353-368.
- Wang, Y. & Berwick, R.C. (2012). Towards a formal framework of cognitive linguistics. *Journal of Advanced Mathematics and Applications*, 1(2), pp.250-263.
- Welmers, W.E. (1973). *African language structures*. Univ of California Press.
- Whiteley W. H (1965). *An introduction to the Kisii*. EALB.
- Wright, S.E. (2002). Trends in language engineering. *Terminology in Advanced Microcomputer Applications, TAMA*, 98.
- Yin, R. K. (2003). *Case study research*. Sage Publications.
- Yoshinaga, N., Miyao, Y., Torisawa, K. & Tsujii, J. I. (2003). Parsing comparison across grammar

formalisms using strongly equivalent grammars. *Traitement Automatique des Langues*, 44(3), pp.15-39.

Zeroual, I. and Lakhouaja, A. (2018). Data science in light of natural language processing: An overview. *Procedia Computer Science*, 127, pp.82-91.

APPENDIX A Language consonants

	Swahili	Ekegusii	Kikamba
1	B	b	
2	mb	mb	
3	M	m	m
4	T	t	t
5		nt	
6	nd	nd	
7	n	n	n
8	r	r	
9	s	s	s
10	ch	ch	
11		nch	
12	ny	ny	ny
13	y	y	y
14	k	k	k
15	g	g	
16	ng'	ng'	ng'
17		nk	
18	ng	ng	
19	w	w	w
20		ns	
21	v		v
22	th		th
23	i		l
24			sy
25			ky
26			w'
28	gh		
29	h		
31	kh		
32	mv		
34	nj		
35	nz		
36	p		
37	sh		
39	z		

APPENDIX B Resource Grammar Development

B.1 Noun

An example of defining lexeme in the lexicon module for each of the languages

```
lin
cloud_N=regN"rire"eri_ama; --Ekegusii
cloud_N=regN "ithweo" i_ma ; --Kikamba
cloud_N=regN "wingu" li_ya; --Swahili
```

Inflection of a simple noun using paradigms regN

Swahili	Kikamba	Ekegusii
<pre>Lang> l-lang=Swa - table cloud_N s Sg : wingu s Pl : mawingu</pre>	<pre>Lang> l -lang=Kam - table cloud_N s Sg : ithweo s Pl : mathweo</pre>	<pre>Lang> l-lang=Swa - table cloud_N s Sg : wingu s Pl : mawingu</pre>

Defining lexemes for compound noun in lexicon modules

```
lin
university_N=compoundN (mkN"kimanyisyo" ki_i) (mkN " kinene" "nene"
ki_i) ki_i; --Kikamba language
university_N= compoundN (mkN"chuo" ki_vi) (mkN " kikuu" ki_vi) ki_vi;
--Swahili language
```

Compound noun inflection

Swahili	Kikamba
<pre>Lang> l -lang=Swa -table university_N s Sg : chuo kikuu s Pl : vyuo vikuu</pre>	<pre>Lang> l -lang=Kam -table university_N s Sg : kimanyisyo kinene s Pl : imanyisyo nene</pre>

B.2 Verbs

Kikamba language verb paradigms

```
oper
regV :Str -> Verb =\vika -> let stem = init vika in
mkVerb vika (vprogressive stem) ("ku"+vika)(stem + "ie")(stem+"aa") ;

iregV : Str -> Verb =\vika -> mkVerb vika vika vika vika vika;

mkVerb : (gen,prog,inf,past,predef : Str) -> Verb= \gen,prog,inf,past,predef ->
{ s =table{
    VGen =>gen;
    VPreProg => prog;
    VInf => inf;
    VPast => past;
    VPreDef =>predef;
    VInf => inf;
    VExtension type=> init gen + extension type + last gen
};
s1 =\ pol,tes,ant,ag => let
v_prefix = (polanttese.s!pol!tes!ant!ag).pl ; in
case < tes, ant,pol > of {
    <Pres, Simul, _> => v_prefix + predef ;
    <Cond, Simul,Pos> | <Past, Simul,Pos> => v_prefix+ past ;
    <Past, Anter,_> => v_prefix+ prog ;
    <_, _,_> => v_prefix+ gen};
progV = prog;
imp=\po,imf => case <po,imf> of {
    <Pos,ImpF Sg _> => gen;
    <Pos,ImpF Pl _> => gen + "i";
    <Neg, _> => "" } };
progressive : Str -> Str = \root ->
case Predef.dp 1 root of {
    "b"|"v"|"m" => root + "ete";
    _ => root + "ite" } ;
```

Swahili language verb paradigms

```
oper
regV :Str -> Verb =\vika -> let stem = init vika in
mkVerb vika (stem+"i") ("ku"+vika)("hu" + vika ) ;

iregV : Str -> Verb =\vika -> mkVerb vika vika vika vika ;

mkVerb : (gen,preneg,inf,habit : Str) -> Verb= \gen,preneg,inf,habit ->
{ s =table{
    VPreNeg => preneg;
    VGen => gen;
```

```

VInf => inf;
Vhabitual =>habit;
VExtension type=> init gen + extension type
};
s1 =\\ pol,tes,ant,ag => let
  v_prefix = (polanttese.s!pol!tes!ant!ag).pl ; in
  case < tes, ant,pol > of {
    <Pres, Simul, Neg> => v_prefix + preneg ;
    <Pres, Simul,Pos> => v_prefix + gen;-- | habit;
    <_, _,_> => v_prefix +gen
  };
  progV = [];
  s2=\\pol,tes,ant,ag => case < tes ,pol> of {
<Pres, Neg> =>(polanttese.s!Neg!Pres!Simul! ag).pl + preneg ;
<_, _> =>(polanttese.s!Pos!Pres!Simul! ag).pl + gen};
imp=\\po,imf => case <po,imf> of {
  <Pos,ImpF Sg False> => gen;
  <Pos,ImpF Pl False> => case last gen of {
    "a" => init gen +"eni";
    _ => gen + "ni" };
  <Pos,ImpF Sg True> => case last gen of {
    "a" => "u" + init gen +"e";
    _ => "u" + gen };
  <Pos,ImpF Pl True> => case last gen of {
    "a" => "m" + init gen +"e";
    _ => "m" + gen };
  <Neg, ImpF Sg _> => "usi" + init gen +"e" ;
  <Neg,ImpF Pl _> => "msi" + init gen +"e" }
};

```

Ekegusii language verb paradigms

```

oper
regV :Str -> Verb =\vika ->
  let stem = init vika in
  mkVerb vika (voiced_less vika) (stem + "eti")(stem + "ire") (voiced_less
vika) ;
  iregV : Str ->Str ->Str ->Str ->Str -> Verb =\gen,fut,neg,anti,inf -> mkVerb
gen fut neg anti inf ;

mkVerb :(gen,fut,neg,anti,inf : Str) -> Verb= \gen,fut,neg,anti,inf ->
{ s =table{
  VFut =>fut ;
  VNeg => neg;
  VGen => gen;
  Vanter => anti;
  VInf => inf ;
  VExtension type=> init gen + extension type + last gen
};

```



```

s1 =\ pol,tes,ant,ag => let
  v_prefix = (polanttense.s!pol!tes!ant!ag).p1 ;
  v2 = (polanttense.s!Pos!Past!Simul!ag).p1 + neg;
  v3 = (polanttense.s!Pos!Past!Simul!ag).p1 + gen;
  gene= Predef.drop 2 gen;
  in      case < tes, ant,pol > of {
<Fut, Simul, Pos> => v_prefix ++ fut ;
<Past, Simul, Neg> => v_prefix + neg;
<Past, Simul, Pos> => case Predef.take 2 gen of {
  "ka" => v_prefix + (prefixvoice gene) ;
  _ => v_prefix + (prefixvoice gen)};
<Past, Anter, Neg> |<Pres, Anter, Neg> =>v_prefix + gen ;
<Cond, Simul, _> |<Pres, Simul, _> |<Fut, Simul, Neg> => v_prefix + inf
;

<_, _, _> => v_prefix + anti };
s2=\pol,tes,ant,ag => case <tes ,ant,pol> of {
<_,_, Neg> =>(subjclitic.s!ag).p5 + inf ;
<Past,Simul,Pos> =>(subjclitic.s!ag).p1 + "reng" ++inf ;
<_,_,Pos> =>(subjclitic.s!ag).p1 +inf };
progV= [];
imp=\po,imf => case <po,imf> of {
  <Pos, _> => gen;
  <Neg, _> => "" };
prefixvoice : Str -> Str = \root ->
  case root of {
    "t"+ _|"k"+ _|"ch"+ _|"s"+ _ => "ga" + root ; ---voiceless consonants
    _ => "ka" + root} ; --voiced consonants

voiced_less : Str -> Str = \root ->
  case root of {
    "t"+ _|"k"+ _|"ch"+ _|"s"+ _ => "go" + root ; ---voiceless consonants
    _ => "ko" + root} ; --voiced consonants

```

B.3 Adjective

Kikamba low leve paradigms (regA, regAdj, iregA)

```

regA :Str->{s : AForm => Str}= \adj ->regAdj adj [];
regAAd : Str-> Str -> {s : AForm => Str} = \seo,seoo -> regAdj seo
seo;
regAdj:Str -> Str-> {s : AForm => Str} = \seo,see -> {s = table {
  AAdj G1 Sg=>case Predef.take 1 seo of {
    "a"|"e"|"i"|"o" => "mw" + seo;
    --"n" => "mwa" + seo;
    "u" => "m" + seo;
    _ => ConsonantAdjprefix G1 Sg + seo };
  AAdj G1 Pl =>case Predef.take 1 seo of {
    "u" => "o" + Predef.drop 1 seo;

```

```

    _ => ConsonantAdjprefix G1 Pl + seo };

AAAdj G2 Sg=>case Predef.take 1 seo of {
    "i" => "mw" + seo;
    "a" => "my" + seo;
    "u" => "m" + seo;
    _ => ConsonantAdjprefix G2 Sg + seo };
AAAdj G2 Pl =>case Predef.take 1 seo of {
    "u" => "my" + seo;
    "i" => "m" + seo;
    _ => ConsonantAdjprefix G2 Pl + seo };

AAAdj G3 Sg=>case Predef.take 1 seo of {
    "i"|"u"|"e" => "y" + seo;
    "a" => "yi" + seo;
    _ => ConsonantAdjprefix G3 Sg + seo };
AAAdj G3 Pl =>case Predef.take 1 seo of {
    "e" => "m"+ seo;
    "u" => "mo"+ Predef.drop 1 seo;
    _ => ConsonantAdjprefix G3 Pl + seo };

AAAdj G4 Sg=>case Predef.take 1 seo of {
    "i" => "k" + seo;
    "u"|"a" => "ky" + seo;
    _ => ConsonantAdjprefix G4 Sg + seo };
AAAdj G4 Pl =>case Predef.take 1 seo of {
    "u" => "mb"+ seo;
    "i" => "nz" + seo;
    "a" => "nd" + seo;
    --- "t" => "nd" + Predef.drop 1 seo; consider thuku
    "s" => "nz" + Predef.drop 1 seo;
    _ => ConsonantAdjprefix G4 Pl + seo };
AAAdj G5 Sg=>case Predef.take 1 seo of {
    "u" => "ko" + Predef.drop 1 seo;
    _ => ConsonantAdjprefix G5 Sg + seo };
AAAdj G5 Pl =>case Predef.take 1 seo of {
    "u" => "t"+ seo;
    "a"|"e"|"i" => "tw" + seo;
    _ => ConsonantAdjprefix G5 Pl + seo };

AAAdj G6 Sg=>case Predef.take 1 seo of {
    "u" => "vo" + Predef.drop 1 seo;
    _ => ConsonantAdjprefix G6 Sg + seo };
AAAdj G6 Pl =>case Predef.take 1 seo of {
    "u" => "k" + seo;
    "i"|"a" => "kw" + seo;
    _ => ConsonantAdjprefix G6 Pl + seo };

AAAdj G7 n =>case Predef.take 1 seo of {

```

```

"s" => "nz" + Predef.drop 1 seo;
"i" => "nz" + seo;
  "v" | "u" => "mb" + seo;
  "k" => "ng" + Predef.drop 1 seo;
  "t" | "a" => "nd" + Predef.drop 1 seo;
  _ => ConsonantAdjprefix G7 n + seo };

AAdj G9 Pl =>case Predef.take 1 seo of {
  "s" | "i" => "nz" + Predef.drop 1 seo;
  "v" | "u" => "mb" + seo;
  "k" => "ng" + seo;
  "t" | "a" => "nda" + seo;
  _ => ConsonantAdjprefix G9 Pl + seo };
AAdj G10 Sg =>case Predef.take 1 seo of {
  "u" => "k" + seo;
  "i"|"a" => "kw" + seo;
  _ => ConsonantAdjprefix G10 Sg + seo };

AAdj g Pl =>case Predef.take 1 seo of {
  "u" => "mo" + Predef.drop 1 seo;
  _ => ConsonantAdjprefix g Pl + seo };
AAdj g Sg=>case Predef.take 1 seo of {
  "i" => "mw" + seo;
  "a" => "my" + seo;
  "u" => "m" + seo;
  _ => ConsonantAdjprefix g Sg + seo };
AComp G1 Sg=>let af : Str = case Predef.take 1 seo of {
  "a"|"e"|"i"|"o" => "mw" + seo;
  "u" => "m" + seo;
  _ => ConsonantAdjprefix G1 Sg + seo };
  in init af + "ang" + last af;
AComp G1 Pl =>let af : Str = case Predef.take 1 seo of {
  "u" => "o" + Predef.drop 1 seo;
  _ => ConsonantAdjprefix G1 Pl + seo }
  in init af + "ang" + last af;

AComp G2 Sg=>let af : Str = case Predef.take 1 seo of {
  "i" => "mw" + seo;
  "a" => "my" + seo;
  "u" => "m" + seo;
  _ => ConsonantAdjprefix G2 Sg + seo };
  in init af + "ang" + last af;
AComp G2 Pl =>let af : Str = case Predef.take 1 seo of {
  "u" => "my" + seo;
  "i" => "m" + seo;
  _ => ConsonantAdjprefix G2 Pl + seo }
  in init af + "ang" + last af;

AComp G3 Sg=>let af : Str = case Predef.take 1 seo of {

```

```

        "i"|"u" => "y" + seo;
        "a" => "yi" + seo;
        _ => ConsonantAdjprefix G3 Sg + seo };
    in init af + "ang" + last af;
AComp G3 Pl =>let af : Str = case Predef.take 1 seo of {
    "u" => "mo"+ Predef.drop 1 seo;
    _ => ConsonantAdjprefix G3 Pl + seo }
in init af + "ang" + last af;

AComp G4 Sg=>let af : Str = case Predef.take 1 seo of {
    "i" => "k" + seo;
    "u" => "ky" + seo;
    _ => ConsonantAdjprefix G4 Sg + seo };
    in init af + "ang" + last af;
AComp G4 Pl =>let af : Str = case Predef.take 1 seo of {
    "u" => "mb"+ seo;
    "i" => "sy" + seo;
    "a" => "nd" + seo;
    _ => ConsonantAdjprefix G4 Pl + seo }
in init af + "ang" + last af;
AComp G5 Sg=>let af : Str = case Predef.take 1 seo of {
    "u" => "ko" + Predef.drop 1 seo;
    _ => ConsonantAdjprefix G5 Sg + seo };
    in init af + "ang" + last af;
AComp G5 Pl =>let af : Str = case Predef.take 1 seo of {
    "u" => "t"+ seo;
    "a" | "i" => "tw" + seo;
    _ => ConsonantAdjprefix G5 Pl + seo }
in init af + "ang" + last af;

AComp G6 Sg=>let af : Str = case Predef.take 1 seo of {
    "u" => "vo" + Predef.drop 1 seo;
    _ => ConsonantAdjprefix G6 Sg + seo };
    in init af + "ang" + last af;
AComp G6 Pl =>let af : Str = case Predef.take 1 seo of {
    "u" => "k" + seo;
    "i"|"a" => "kw" + seo;
    _ => ConsonantAdjprefix G6 Pl + seo }
in init af + "ang" + last af;

AComp G7 n =>let af : Str = case Predef.take 1 seo of {
    "s" |"i" => "nz" + Predef.drop 1 seo;
    "v" | "u" => "mb" + seo;
    "k" => "ng" + seo;
    "t" | "a" => "nd" + seo;
    _ => ConsonantAdjprefix G7 n + seo };
    in init af + "ang" + last af;

AComp G9 Pl =>let af : Str = case Predef.take 1 seo of {
    "s" |"i" => "nz" + Predef.drop 1 seo;

```

```

        "v" | "u" => "mb" + seo;
        "k" => "ng" + seo;
        "t" | "a" => "nd" + seo;
        _ => ConsonantAdjprefix G9 Pl + seo };
    in init af + "ang" + last af;
AComp G10 Sg => let af : Str = case Predef.take 1 seo of {
    "u" => "k" + seo;
    "i"|"a" => "kw" + seo;
    _ => ConsonantAdjprefix G10 Sg + seo }
    in init af + "ang" + last af;
AComp g Pl => let af : Str = case Predef.take 1 seo of {
    "u" => "mo" + Predef.drop 1 seo;
    _ => ConsonantAdjprefix g Pl + seo }
    in init af + "ang" + last af;

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;

Advv => see
} };

iregA : Str-> Str -> {s : AForm => Str} = \seo,seoo -> {
  s = table {
    AAdj g Sg => seo;
    AAdj g Pl=> seoo ;
    AComp g Sg => init seo + "ang" + last seo;
    AComp g Pl => init seoo + "ang" + last seoo;
    Advv =>[]
  } ;

```

Ekegusii low level paradigms

```

regA :Str->{s : AForm => Str}= \adj ->regAdj adj [];
regAAd : Str-> Str -> {s : AForm => Str} = \seo,seoo -> regAdj seo
seoo;

```

```

regAdj:Str -> Str-> {s : AForm => Str} = \seo,see -> {s = table {
  AAdj G1 Sg=>case Predef.take 1 seo of {
    "a"|"i"|"u" => "omu" + seo;
    "o" |"e" => "omw" + seo;
    _ => ConsonantAdjprefix G1 Sg + seo };
  AAdj G1 Pl =>case Predef.take 1 seo of {
    _ => ConsonantAdjprefix G1 Pl + seo };
  AAdj G2 Sg=>case Predef.take 1 seo of {
    "a"|"e"|"i"|"u" => "omu" + seo;
    "o" => "omw" + seo;

```

```

        "b" => "em" + seo;
        _ => ConsonantAdjprefix G2 Sg + seo };
AAAdj G2 Pl =>case Predef.take 1 seo of {
    "o" | "y" => "emi" + seo;
    "b" => "em" + seo;
    _ => ConsonantAdjprefix G2 Pl + seo };

AAAdj G3 Sg=>case Predef.take 1 seo of {
    "o" | "i" => "eng" + seo;
    "y" => "engi" + seo;
    "b" => "em" + seo;
    "e" => "eny" + seo;
    _ => ConsonantAdjprefix G3 Sg + seo };
AAAdj G3 Pl =>case Predef.take 1 seo of {
    "o" | "i" => "ching" + seo;
    "b" => "chim" + seo;
    "y" => "chingi" + seo;
    _ => ConsonantAdjprefix G3 Pl + seo };
AAAdj G4 Sg=>case Predef.take 1 seo of {
    "a" | "e" | "i" | "o" | "u" => "rigi" + seo;
    _ => ConsonantAdjprefix G4 Sg + seo };
AAAdj G4 Pl =>case Predef.take 1 seo of {
    _ => ConsonantAdjprefix G4 Pl + seo };
AAAdj G5 Sg=>case Predef.take 1 seo of {
    "y" | "i" => "eki" + seo;
    "g" => "eke" + seo;
    _ => ConsonantAdjprefix G5 Sg + seo };
AAAdj G5 Pl =>case Predef.take 1 seo of {
    "i" => "ebi" + seo;
    _ => ConsonantAdjprefix G5 Pl + seo };

AAAdj G6 Sg=>case Predef.take 1 seo of {
    "i" | "o" => "oru" + seo;
    _ => ConsonantAdjprefix G6 Sg + seo };
AAAdj G6 Pl =>case Predef.take 1 seo of {
    "i" | "o" => "ching'" + seo;
    _ => ConsonantAdjprefix G6 Pl + seo };
AAAdj G7 Sg=>case Predef.take 1 seo of {
    _ => ConsonantAdjprefix G7 Sg + seo };
AAAdj G7 Pl =>case Predef.take 1 seo of {
    _ => ConsonantAdjprefix G7 Pl + seo };
AAAdj G8 Sg=>case Predef.take 1 seo of {
    "i" | "o" => "obu" + seo;
    _ => ConsonantAdjprefix G8 Sg + seo };
AAAdj G8 Pl =>case Predef.take 1 seo of {
    _ => ConsonantAdjprefix G8 Pl + seo };
AAAdj G9 Sg=>case Predef.take 1 seo of {
    "i" | "o" => "oku" + seo;
    _ => ConsonantAdjprefix G9 Sg + seo };
AAAdj G9 Pl =>case Predef.take 1 seo of {

```

```

        _ => ConsonantAdjprefix G9 Pl + seo };
AAAdj G11 Sg=>case Predef.take 1 seo of {
    "e"|"o" => "am" + seo;
    _ => ConsonantAdjprefix G11 Sg + seo };
AAAdj G11 Pl =>case Predef.take 1 seo of {
    "e"|"o" => "am" + seo;
    _ => ConsonantAdjprefix G11 Pl + seo };

AAAdj G10 Sg=>case Predef.take 1 seo of {
    _ => ConsonantAdjprefix G10 Sg + seo };

AAAdj G10 Pl =>[];
Advv => see
}   };

iregA : Str-> Str -> {s : AForm => Str} = \seo,seoo -> {
  s = table {
    AAdj g Sg=> seo;
    AAdj g Pl => seoo;
    Advv=> [] } };

```

Swahili language low level paradigms

```

regA :Str->{s : AForm => Str}= \adj ->regAdj adj ("vi"+adj);
regAAd : Str-> Str -> {s : AForm => Str} = \seo,seoo -> regAdj seo
seoo;

regAdj:Str -> Str-> {s : AForm => Str} = \seo,see -> {s = table {
  AAdj G1 Sg=>case Predef.take 1 seo of {
    "a"|"e"|"i"|"u" => VowelAdjprefix G1 Sg + seo;
    _ => ConsonantAdjprefix G1 Sg + seo };
  AAdj G1 Pl =>case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G1 Pl + seo;
    "i" => VoweliAdjprefix G1 Pl + seo;
    _ => ConsonantAdjprefix G1 Pl + seo };

  AAdj G2 Sg=>case Predef.take 1 seo of {
    "a"|"e"|"i"|"o"|"u" => VowelAdjprefix G2 Sg + seo;
    _ => ConsonantAdjprefix G2 Sg + seo };
  AAdj G2 Pl =>case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G2 Pl + seo;
    "i" => VoweliAdjprefix G2 Pl + seo;
    _ => ConsonantAdjprefix G2 Pl + seo };
  AAdj G3 Sg=>case Predef.take 1 seo of {
    "a"|"e"|"i"|"o"|"u" => VowelAdjprefix G3 Sg + seo;
    _ => ConsonantAdjprefix G3 Sg + seo };
  AAdj G3 Pl =>case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G3 Pl + seo;
    "i" => VoweliAdjprefix G3 Pl + seo;
    _ => ConsonantAdjprefix G3 Pl + seo };

  AAdj G4 n =>case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G4 n + seo;

```

```

        "i" => VoweliAdjprefix G4 n + seo;
        _ => ConsonantAdjprefix G4 n + seo };
AAAdj G5 n => case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G5 n + seo;
    "i" => "ny" + Predef.drop 1 seo;
    "d"|"g"|"z" => "n" + seo;
    "b"|"p"|"v" => "m" + seo;
    _ => ConsonantAdjprefix G5 n + seo };

AAAdj G6 Sg=>case Predef.take 1 seo of {
    "a"|"e"|"i"|"o"|"u" => VowelAdjprefix G6 Sg + seo;
    _ => ConsonantAdjprefix G6 Sg + seo };
AAAdj G6 Pl =>case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G6 Pl + seo;
    "i" => "ny" + Predef.drop 1 seo;
    "d"|"g"|"z" => "n" + seo;
    "b"|"p"|"v" => "m" + seo;
    _ => ConsonantAdjprefix G6 Pl + seo };

AAAdj G7 n =>case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G7 n + seo;
    "i" => VoweliAdjprefix G7 n + seo;
    _ => ConsonantAdjprefix G7 n + seo };
AAAdj G8 n =>case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G8 n + seo;
    "i" => VoweliAdjprefix G8 n + seo;
    _ => ConsonantAdjprefix G8 n + seo };
AAAdj G9 n =>case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G9 n + seo;
    "i" => VoweliAdjprefix G9 n + seo;
    _ => ConsonantAdjprefix G9 n + seo };
AAAdj G10 n =>case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G9 n + seo;
    "i" => VoweliAdjprefix G9 n + seo;
    _ => ConsonantAdjprefix G9 n + seo };

AAAdj G11 Sg=>case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G11 Sg + seo;
    "i" => VoweliAdjprefix G11 Sg + seo;
    _ => ConsonantAdjprefix G11 Sg + seo };

AAAdj G12 Sg=>case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G12 Sg + seo;
    "i" => VoweliAdjprefix G12 Sg + seo;
    _ => ConsonantAdjprefix G12 Sg + seo };
AAAdj G13 Sg=>case Predef.take 1 seo of {
    "a"|"e"|"o"|"u" => VowelAdjprefix G13 Sg + seo;
    "i" => VoweliAdjprefix G13 Sg + seo;
    _ => ConsonantAdjprefix G13 Sg + seo };
AAAdj _ Pl =>[];

```



```

    Advv => see} };
iregA : Str -> {s : AForm => Str} =\seo -> {
    s = table {
        AAdj g n => seo;
        Advv => "vi" ++ seo} };

```

B.4 Numbers paradigms

a. Kikamba smart paradigm operation

```

oper
mkNum : Str -> Str -> Str -> Str -> {s : DForm => CardOrd => Cgender => Str} =
\two, twelve, twenty, second ->{s = table {
    unit => table {NCard =>\g => Cardprefix g + two ;
        NOrd => \g => Ordprefix g ++ second} ;
    teen => table {NCard =>\g =>"ikumi na" ++ twelve ;
        NOrd => \g => Ordprefix g ++ "ikumi na" ++ twelve} ;
    ten => table {NCard =>\g =>"miongo " ++ twelve ;
        NOrd => \g => Ordprefix g ++ "miongo" ++ twelve};
    hund => table {NCard =>\g =>"maana " ++ twenty ;
        NOrd => \g => Ordprefix g ++ "maana" ++ twenty}} };

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}} };

```

b. Swahili linearization type

```

lin num x = x ;
lin n2 = mkNum2 "li" "ili" "eli" "keli" ;
lin n3 = mkNum "tatu" "itatu" "atatu" "katatu" ;
lin n4 = mkNum "nya" "ina" "ana" "kana" ;
lin n5 = mkNum "tano" "itano" "atano" "katano" ;
lin n6 = regNum "nthathatu" ;
lin n7 = regNum "muonza" ;
lin n8 = regNum "nyanya" ;
lin n9 = regNum "kenda" ;

lin pot01 = mkNum1 "mwe" "yimwe" "mbee" ** {n = Sg} ;
lin pot0 d = d ** {n = Pl} ;
lin pot110 = regCardOrd "ikumi" ** {n = Pl} ;
lin pot111 = regCardone "ikumi na" "mwe" ** {n = Pl} ;
lin pot1to19 d = {s = d.s ! teen} ** {n = Pl} ;
lin pot0as1 n = {s = n.s ! unit} ** {n = n.n} ;

```

```

lin pot1 d = {s = d.s ! ten} ** {n = Pl} ;
lin pot1plus d e = { s = table {
    NCard => \\g => d.s ! ten ! NCard ! g ++ "na"++ e.s ! unit ! NCard ! g ;
    NOrd => \\g =>Ordprefix g++ d.s ! ten ! NCard ! g ++ "na"++ e.s ! unit !
NCard ! g } ; n = Pl} ;
lin pot1as2 n = n ;
lin pot2 d = {s = d.s ! hund} ** {n = Pl} ;
lin pot2plus d e = {s = table {
    NCard => \\g => d.s ! hund ! NCard ! g ++ "na" ++ e.s !NCard ! g ;
    NOrd => \\g =>Ordprefix g++ d.s ! hund ! NCard ! g ++ "na" ++ e.s !
NCard ! g } ; n = Pl} ;
lin pot2as3 n = n ;
lin pot3 n = { s = table {
    NCard => \\g => mkCard NCard "ngili" ! g ++ n.s ! NCard ! g ;
    NOrd => \\g =>Ordprefix g++ mkCard NCard "ngili" ! g ++ n.s ! NCard ! g }
; n = Pl} ;
lin pot3plus n m = { s = table {
    NCard => \\g => "ngili" ++ n.s ! NCard !g ++ m.s ! NCard ! g ;
    NOrd => \\g =>Ordprefix g++ "ngili" ++ n.s ! NCard !g ++ m.s ! NCard !
g} ; n = Pl} ;

```

c. Kikamba smart paradigm operation

```

oper
mkNum : Str -> Str -> Str -> Str -> {s : DForm => CardOrd => Cgender => Str} =
\two, twelve, twenty, second ->{s = table {
    unit => table {NCard =>\\g => Cardprefix g + two ;
        NOrd => \\g => Ordprefix g ++ second} ;
    teen => table {NCard =>\\g =>"ikumi na" ++ twelve ;
        NOrd => \\g => Ordprefix g ++ "ikumi na" ++ twelve} ;
    ten => table {NCard =>\\g =>"miongo " ++ twelve ;
        NOrd => \\g => Ordprefix g ++ "miongo" ++ twelve};
    hund => table {NCard =>\\g =>"maana " ++ twenty ;
        NOrd => \\g => Ordprefix g ++ "maana" ++ twenty}} } ;

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}} } ;

```

d. Swahili linearization type

```

lin num x = x ;
lin n2 = mkNum2 "ili" "ishirini" "pili" ;
lin n3 = mkNum "tatu" "thelathini" ;
lin n4 = mkNum "nne" "arobaini" ;
lin n5 = mkNum "tano" "hamsini" ;

```

```

lin n6 = regNum "sita" "sitini";
lin n7 = regNum "saba" "sabini";
lin n8 = regNum "nane" "themanini";
lin n9 = regNum "tisa" "tisini" ;

lin pot01 = mkNum1 "moja" "kwanza" ** {n = Sg} ;
lin pot0 d = d ** {n = Pl} ;
lin pot110 = regCardOrd "kumi" ** {n = Pl} ;
lin pot111 = regCardone "kumi na" "moja" ** {n = Pl} ;
lin pot1to19 d = {s = d.s ! teen} ** {n = Pl} ;
lin pot0as1 n = {s = n.s ! unit} ** {n = n.n} ;
lin pot1 d = {s = d.s ! ten} ** {n = Pl} ;
lin pot1plus d e = { s = table {
    NCard => \\g => d.s ! ten ! NCard ! g ++ "na"++ e.s ! unit !
NCard ! g ;
    NOrd => \\g =>Ordprefix g++ d.s ! ten ! NCard ! g ++ "na"++ e.s !
unit ! NCard ! g } ;
    n = Pl} ;
lin pot1as2 n = n ;
lin pot2 d = {s = d.s ! hund} ** {n = Pl} ;
lin pot2plus d e = {s = table {
    NCard => \\g => d.s ! hund ! NCard ! g ++ "na" ++ e.s !NCard ! g
;
    NOrd => \\g =>Ordprefix g++ d.s ! hund ! NCard ! g ++ "na" ++
e.s ! NCard ! g } ;
    n = Pl} ;
lin pot2as3 n = n ;
lin pot3 n = { s = table {
    NCard => \\g => mkCard NCard "elfu"!g ++ n.s ! NCard ! g ;
    NOrd => \\g =>Ordprefix g++ mkCard NCard "elfu" ! g ++ n.s !
NCard ! g } ;
    n = Pl} ;
lin pot3plus n m = { s = table {
    NCard => \\g => "elfu" ++ n.s ! NCard !g ++ m.s ! NCard ! g ;
    NOrd => \\g =>Ordprefix g++ "elfu" ++ n.s ! NCard !g ++ m.s !
NCard ! g} ;
    n = Pl} ;

```

e. Ekegusii numeral operation

```

oper
mkNum : Str -> Str -> {s : DForm => CardOrd => Cgender => Str} =
  \two, second -> {s = table {
    unit => table {NCard =>\\g =>case two of {
        "ato" => Cardprefix g +"t" +two ;
        _ => Cardprefix g +two } ;
        NOrd => \\g => Ordprefix g ++ second} ;
    teen => table {NCard =>\\g =>case two of {
        "bere" => "ikomi na" ++ Cardtwelveprefix g + two ;
        "ato" => "ikomi na" ++ CardThirteenprefix g + two ;
        "tano"=> "ikomi na" ++ Cardfifteenprefix g + two ;
        "ne"=> "ikomi na" ++ Cardfouteenprefix g + two } ;
        NOrd => \\g => Ordprefix g ++ "ikomi na" ++ second } ;
    ten => table {NCard =>\\g =>case two of {

```

```

        "ato" => "emerongo et" +two ;
        _ => "emerongo a" +two };
NOrd => \\g =>case two of {
    "ato" => Ordprefix g ++"emerongo et" +two ;
    _ => Ordprefix g ++"emerongo a" +two };;
hund => table {NCard =>\\g =>case two of {
    "bere" => "amagana e" +two ;
    "ato" => "amagana et" +two ;
    _ => "amagana a" +two };
NOrd => \\g => case two of {
    "bere" => Ordprefix g ++"amagana e" +two ;
    "ato" => Ordprefix g ++"amagana et" +two ;
    _ => Ordprefix g ++"amagana a" +two }}} } ;

```

f. Ekegusii linearization types

```

lin num x = x ;
lin n2 = mkNum "bere" "kabere" ;
lin n3 = mkNum "ato" "gatatu" ;
lin n4 = mkNum "ne" "kane" ;
lin n5 = mkNum "tano" "gatano" ;
lin n6 = mkNum6 "tano" "mo" ;
lin n7 = mkNum7 "tano" "bere" ;
lin n8 = mkNum8 "tano" "tato" ;
lin n9 = mkNum9 "kianda" ;

lin pot01 = mkNum1 "mo" "tang'ani" ** {n = Sg} ;
lin pot0 d = d ** {n = Pl} ;
lin pot110 = regCardOrd "ikomi" ** {n = Pl} ;
lin pot111 = regCardone "ikomi na" "mo" ** {n = Pl} ;
lin pot1to19 d = {s = d.s ! teen} ** {n = Pl} ;
lin pot0as1 n = {s = n.s ! unit} ** {n = n.n} ;
lin pot1 d = {s = d.s ! ten} ** {n = Pl} ;
lin pot1plus d e = { s = table {
    NCard => \\g => d.s ! ten ! NCard ! g ++ "na"++ e.s ! unit ! NCard ! g ;
    NOrd => \\g =>Ordprefix g++ d.s ! ten ! NCard ! g ++ "na"++ e.s ! unit !
NCard ! g } ; n = Pl} ;
lin pot1as2 n = n ;
lin pot2 d = {s = d.s ! hund} ** {n = Pl} ;
lin pot2plus d e = {s = table {
    NCard => \\g => d.s ! hund ! NCard ! g ++ "na" ++ e.s !NCard ! g ;
    NOrd => \\g =>Ordprefix g++ d.s ! hund ! NCard ! g ++ "na" ++ e.s !
NCard ! g } ; n = Pl} ;
lin pot2as3 n = n ;
lin pot3 n = { s = table {
    NCard => \\g => mkCard NCard "chilibu" ! g ++ n.s ! NCard ! g ;
    NOrd => \\g =>Ordprefix g++ mkCard NCard "chilibu" ! g ++ n.s ! NCard ! g
} ; n = Pl} ;
lin pot3plus n m = { s = table {
    NCard => \\g => "chilibu" ++ n.s ! NCard !g ++ m.s ! NCard ! g ;
    NOrd => \\g =>Ordprefix g++ "chilibu" ++ n.s ! NCard !g ++ m.s ! NCard !
g} ; n = Pl} ;

```

Shared Digit implementation

```
lincat
  Dig = TDigit ;

lin
  IDig d = d ;

  IIDig d i = {s = table {
    NCard => \\g => d.s! NCard ! g ++ BIND ++ i.s ! NCard ! g ;
    NOrd => \\g => d.s! NOrd! g ++ BIND ++ i.s !NCard! g } ;
    n = Pl } ;

  D_0 = mkDig "0" ;
  D_1 = mk3Dig "1" "1" Sg ;
  D_2 = mkDig "2" ;
  D_3 = mkDig "3" ;
  D_4 = mkDig "4" ;
  D_5 = mkDig "5" ;
  D_6 = mkDig "6" ;
  D_7 = mkDig "7" ;
  D_8 = mkDig "8" ;
  D_9 = mkDig "9" ;

oper
  mk2Dig : Str -> Str -> TDigit = \c,o -> mk3Dig c o Pl ;
  mkDig : Str -> TDigit = \c -> mk2Dig c (c ) ;

  mk3Dig : Str -> Str -> Number -> TDigit = \c,o,n -> {
    s = table {NCard => \\g => c ; NOrd => \\g =>Ordprefix g ++ o} ;      n
  = n} ;
  TDigit = {n : Number ; s : CardOrd => Cgender => Str } ;
```

APPENDIX C: TEST SUITES

C.1 Source in English

these five trees of the king are better
the twenty very bad men drank beer
all the ten new cars will be scratched
young doctors have come today
three beautiful clouds were flying
many very clever policemen aren't taught
two hundred thousand girls were good
the four important pens aren't lost today
the bad wind blew and one hundred trees fell
these ten ugly shoes are brown
these sixty black persons are theirs
the long rivers split many rocks
some big forests had wet floors and green grass
those ten beautiful and clever friends have fallen now
the wide mountain squeezes the short road
the hand of John is cut
the small blue horses live
very old trains came from Paris
old dry seeds were bought
those two hundred green trees will fall after rain
the eggs of snakes are white
these five thousand women sing
either the doctor killed the persons or they drank wine
somewhere there is a priest who loves policemen
small red seeds smell
my father is very old
a green long leaf floats on the river
they will write the best book
meat from a big fish was very good, we heard
one hand doesn't kill animal
seven babies with hair fear doctor
we read three nights in the school
the priests read newspaper in that white church
the clever students are from good universities
ten green carpets were long
the forest has big green trees and birds sing there
my brother vomited
we bought the important newspaper from the blue shop
the two women are married
persons with big bellies like books
those two beautiful girls swim
she is the first wife of John
the mouths of fish smell
the floor of the house is of steel
the ugly apples are from this tree
there is blood in the eyes of a dog
sea is bigger than the lake
nine hundred thousand persons live in villages
the wings of the airplane fell on the sea and all the persons died
all these twenty big women will wash black coats

we know the science on everything
the war of tongue is very bad
these bad men cut three trees
the heavy black clouds will rain
your five brothers are priests
five skins have burned and the policemen are sleeping
very big sheep have eaten grass
few white babies will jump
she has wiped the window
every animal vomited
everybody laughed
the short priest bought a house
she will eat fish
he hit the sheep
they went to the garden
some boys will play
the girl has died
they have read papers
the red blood is thick
she will buy many books
we didn't eat blood
many short children have fallen
the king played with his wife
a priest lives in a big house
a friend bought fifty books from a shop
the student came by bike to university
ten very short boots were dirty
two hundred of clever students in the school swim
the teacher wrote seven books and the second book was written through somebody
when everybody is young and beautiful and everything was good
the day such that there will be a peace on an earth
the clever students are running
the pen of John was on the table
the door of church is red
every baby is either a boy or a girl
this shortest road is from the bank to house
the science of sky is very important
either from here, there or everywhere
how many years, my doctor
the girl who hunts animals
the heart of the baby is very good
the clean leg swelled
this grammar speaks twelve languages
I am a man
these papers are dry
these fifty black dogs will sleep
they like the rule that the books are thin
John is big and clever
the baby was under the table
those boys swam and these girls ran

C.2 Gold standard for Swahili

miti hii mitano ya mfalme ni bora.
wanaume ishirini wabaya sana walikunywa pombe
magari yote kumi mapya yatakwaruzwa
daktari wachanga wamekuja leo
mawingu matatu mazuri yalikuwa yanapaa
polisi wengi werevu sana hawajafunzwa
wasichana elfu mia mbili walikuwa wazuri
kalamu nne muhimu hazijapotezwa leo
upepo mbaya ulivuma na miti mia moja ilianguka
viatu hivi kumi vibaya ni vya rangi ya hudhurungi
watu hawa sitini weusi ni wao
mito mirefu inapasua majabali mengi
misitu mingine mikubwa ilikuwa na udongo mnyevu
marafiki hao kumi warembo na werevu wameanguka
sasa
mlima mpana unaibana barabara fupi
mkono wa Yoana umekatika
farasi wa rangi ya buluu wadogo wanaishi
magari ya moshi mazee sana yalikuja kutoka Paris
mbegu nzee kavu zilinunuliwa
miti hiyo mia mbili ya rangi ya kijani itaanguka baada
ya mvua
mayai ya nyoka ni meupe
hawa wanawake elfu tano huimba
Ima daktari aliwaua watu au walikunywa mvinyo
sehemu fulani kuna kasisi ambaye anapenda polisi
mbengu nyekundu ndogo zinanuka
baba wangu ni mzee sana
jani refu la rangi ya kijani linaelea mtoni
wataandika kitabu kizuri kabisa
nyama ya samaki mkubwa ilikuwa nzuri sana, tulisikia
mkono mmoja hauwezi kumuua mnyama
watoto saba wenye nywele wanaogopa daktari
tunasoma usiku tatu shuleni
kasisi wanasoma gazeti ndani ya kanisa hilo jeupe
wanafunzi werevu hutokea vyyo vikuu vizuri
mazulia kumi ya rangi ya kijani yalikuwa marefu
msitu una miti mikubwa ya rangi ya kijani na ndege
huwa wanaimba huko
kaka wangu alitapika nyumbani
sisi tulinunua gazeti muhimu kutoka duka la rangi ya
buluu
wanawake hao wawili wameolewa
watu wenye vitambi vikubwa hupenda vitabu
wasichana hao wawili wazuri huogelea
yeye ni bibi wa kwanza wa Yoana
midomo ya samaki inanuka
sakafu ya nyumba ni ya chuma
matofaa mabaya ni ya mti huu
kuna damu kwenye macho ya mbwa
bahari ni kubwa kuliko ziwa
watu elfu mia tisa wanaishi vitongojini
mabawa ya ndege yalianguka baharini na watu wote
wakafa

wanawake ishirini watafua makoti
tunajua sayansi ya kila kitu
vita za ulimi ni vibaya sana
wanaume hawa waovu wanakata miti mitatu
mawingu mazito meusi yatayesha
kaka wenyu watano ni kasisi
ngozi tano zimeungua na polisi wanalala
kondoo makubwa sana yamekula nyasi
watoto wachache weupe wataruka
amepanguza dirisha
kila mnyama alitapika
kila mtu alicheka
yule kasisi mfupi alinunua nyumba
alimla samaki
aligonga kondoo
walienda kwenye bustani
vijana wengine watacheza
msichana mwenyewe ameaga
wao wamesoma makaratasi
damu nyekundu ni nzito
atanunua vitabu vingi
sisi hatukula damu
watoto wengi wafupi wameanguka
mfalme alicheza na bibi wake
kasisi anaishi katika nyumba kubwa
rafiki alinunua vitabu hamsini dukani
mwanafunzi alikuja chuoni kikuu na baisikeli
buti kumi fupi sana zilikuwa chafu
mia mbili ya wanafunzi werevu shuleni huogelea
mwalimu aliandika vitabu saba na cha pili kiliandikwa
na mtu mwingine
wakati kila mtu ni mchanga na mrembo na kila kitu
kilikuwa kizuri
siku ambapo kutawa kuna amani ardhini
wanafunzi werevu wanakimbia
kalamu ya Yohana ilikuwa mezani
mlango wa kanisa ni mwekundu
kila mtoto ni ima kijana au msichana
barabara hii fupi kabisa ni kutoka benki hadi nyumba
sayansi ya anga ni muhimu sana
ima kutoka hapa , hapo au kila mahali
miaka mingapi, daktari wangu
msichana ambaye anawinda wanyama
moyo wa mtoto ni mzuri sana
mguu safi ulivimba
sarufi hii inaongea lugha kumi na mbili
mimi ni mwanaume
makaratasi haya ni makavu
mbwa hawa hamsini weusi watalala
wanapenda kanuni kuwa vitabu ni vyembamba
Yoana ni mkubwa na mwerevu
mtoto alikuwa chini ya meza
vijana hao waliogelea na wasichana hawa walikimbia

C.3 Gold standard for Kikamba

Miti ii itano ya Musumbi ni miseango
andu aume miongo ili athuku muno nimananyw'ie
nzovi
Ngali syonthe ikumi nzau ikakalywa
Aiiti ma muika nimooka umunthi
Mathweo atatu manake ni nimanaulukite
Asikali aingi oi muno tiamanyisye
Eitu ngili maana eli mai aseu
iandiki inya sya vata iinawa umunthi
kiseve kithuku nikinauutanie na miti yiana yimwe
yavaluka
latu ii ikumi thuku ni sya langi wa kaki
andu aa miongo nthathatu aiu ni moo
Mbusi ndaasa nisyautanisye mavia maingi
mititu imwe minene yai na nginyo nthithu na nyeki
sya langi wa matu
anyanya aya ikumi anake na oi nimavaluka yuyu
Kiima kyaamu nikivinyiaa lelu mukuvi
kw'oko kwa Yoana nikwatemwa
Mbalasi nini sya langi wa waiyu nituaa
Ngali sya mwaki nguu vyu nisyaukie kuma Paris
Mbeu mbumu nguu syathooiwe
Miti iya maana eli ya langi wa matu ikavaluka itina wa
mbua kua
matumbi ma nzoka ni meu
iveti ii ngili itano ikaina
No kethiwa muiiti nunamoai andu kana
nimananyw'ie mbinyu
Veou vandu ve muthembi ula wendete asikali
mbeu ndune nini niinyungaa
nau wakwa ni mukuu muno
Itu yiasa ya langi wa matu niyithambalalaa usini
makaandika ivuku iseo vyu
Nyama kuma ikuyuni inene yai nzeo vyu,ithyi
nituneewie
kw'oko kumwe kuiuuua nyamu
twana muonza twina nzwii nitukiaa muiiti
ithyi nitusomaa iwiyo itatu vau sukulu
athembi nimasomaa ikaseti ikanisani yiya yeu
amanyiw'a oi ni kuma imanyisyo nene nzeo
mikeka ikumi ya langi wa matu yai miasa
Mutitu ukethethwa na miti ya langi wa matu minene
na nyunyi sikaina vo
mwana inya wakwa niunatavikie
Nitunathooie ikaseti ya vata kuma ndukani ya langi
wa waiyu
iveti ili ni ndwae
Andu mena mavu manene nimendete mavuku
eitu aya eli anake nimathambiaa
we ni muka wa mbee wa Yoana
minuka ya makuyu nunyungaa
nginyo ya nyumba ni ya kiaa
mavuu mathuku ni kuma muti uu
Ve nthakame methoni ma ngiti
ukanga ni munenange kwi iia
andu ngili maana kenda nimatuaa nduani

Nthwau sya ndeke ninavalukie iulu wa ukanga na
andu onthe nimanakw'ie
iveti ii miongo ili nene syonthe ikavua makoti maiu
ithyi nitwisi sayasi iulu wa kila kindu
kau wa uimi ni muthuku vyu
andu aume aa athuku nimatemaa miti itatu
mathweo maiu maito makaua
ana inya menyu atano ni athembi
ithuma itano nisyavya na asikali ni nimakomete
malondu manene muno nimaya nyeki
twana tunini tweu tukathaanyaka
niwavangula mbuti
kila nyamu ninatavikie
kila mundu nuunathekie
muthembi mukuvi niunathooie nyumba
we akaya ikuyu
we niunakimie ilondu
nimanaendie muundani
ivisi imwe ikathauka
mwiitu niwakw'a
nimasoma mathangu
nthakame ndune ni ngamu
we akathooa mavuku maingi
ithyi tuineeya nthakame
twana twiingi tukuvi nitwavaluka
musumbi niunathaukie na muka wake

muthembi ninutua nyumbani nene
munyanya nunauie mavuku miongo itano kuma
ndukani
mumanyiw'a anookie na kisululu nginya kimanyisyon
kinene
mbuti ikumi nguvi muno syai kiko
maana eli ma amanyiw'a oi sukuluni nimathambiaa
mumanyisya niwaandikie mavuku muonza na ivuku ya
keli niyaandikwa kwisila o mundu
yila kila mundu ni munini na mumbe nesa na kila
undu wai museo
Muthenya ula ota uu kwiithiwa muuo iulu wa nthi
amanyiw'a oi nimasembete
kiandiki kya Yoana kyai mesani
muomo wa ikanisa ni mutune
kila kana ni kavisi kana kelitu
Lelu uu mukuvi vyu umite vengi kuvika nyumba
sayasi ya yayaya ni ya vata vyu
no kethiwa kuma vaa , vau kana kila vandu
myaka yiana , muiiti wakwa
mwiitu ula ninusyimaa nyamu
ngoo ya kana ni nzeo vyu
kuu kutheu nikunaimbie
ngulama ii niineenaa ithyomo ikumi na ili
nyie ni mundu muume
Mathangu aa ni momu
ngiti ii miongo itano nziu ikakoma
nimendaa mwiao ati mavuku ni matheke
Yoana ni munene na mui
kana kai uungu wa mesa

ivisi iya ninathambiie na eitu aa nimanasembie

C.4 Gold standard for Ekegusii

emete eye etano ya omoruoti na emiya
abasacha emerongo ebere ababe mono banywete
amarwa
chigari ikomi chinyia chionsi chigocha gosigikigwa
abarwaria abake bachire rero
amare atato amasere arenge koiruruka
abasaigari abange abang'aini mono mbari gosomigwa
abasiseke chiribu amagana ebere barenge abaya
chikaramu ine chieng'encho nichitugutetu rero
embeo embe ekegusa na emete rigana erimo ekagwa
ebikoroto ebi ikomi ebibe bire maroba
abanto aba emerongo etano no' omo abamwamu na
babo
chindoche chitambe chikorusanania ebitare ebinge
chinsana chinde chinene chibwate chibaranda chinyi
na amanyansi
abasani baria ikomi abasere na abang'aini bakagwire
bono
egetunwa ekegare kemigete ebara enyeng'e
okoboko gwa Choni kobutoirwe
chibarasi chinke chia eragi ya buluu chikomenya
chitureni chinkoro mono chiarute korwa Paris
chintetere chinkoro chinkamoku chikagorwa
emete eria amagana ebere ya eragi ya machani
nkogwa egocha ekero embure yakorire gotwa
amagena ye ching'endanse are amarabu
abasubati aba chiribu batano bagotera
omarwaria omoitete abanto gose bakanywa edivai
ase gete omosasiroti naroo oanchete abasigari
chintetere chinke chimbariri chigotiokerera
tata one no omokoro mono
rito ritambe ria eragi ya machani richeng'enenete
oroche igoro
bagocha korika egetabu ekiya bi
enyama korwa enswe enene yareng'e engiya mono ,
twaigwete
okoboko okomo gotari goita eng'iti
abana batano na babere babwate etuki bakooboa
omonyagitari
intwe ntokosoma chintuko isato ase esukuru
abasasiroti basomete egaseti ase ekanisa eria
endabu
abaorokigwa abang'aini bare korwa chiyunibasiti
chingiya
chikabeti ikomi chia eragi ya machani chiare
chintambe
oronsana robwate emete ya eragi ya machani
emenene na chinyoni chigotera ororo
momura one akaroka
twagorete egaseti eng'encho korwa etuka ya eragi ya
buluu
abasubati babere banywometwe
abanto babwate chinda chinene banchete ebitabu
abasiseke baria babere abasere bakoaka obari
ere no omokungu omotang'ani o Choni

emenwa ye chinswe egotioka
ebaranda ye enyomba ere ye echuma
chiepo chimbe chire korwa omote oyo
amanyinga nare ime ye amaiso ye esese
enyancha ne enene kobua enyancha
abanto chiribu amagana kianda bamenyete ime
ebinyoro
chimbaba chie endenge chiagwete enyancha igoro na
abanto bonsi bagakwa
abasubati aba bonsi emerongo ebere abanene
bagocha gosibia chigoti chimwamu
tomanyete esayansi ya kera egento
esegi ye chimeme ne embe mono
abasacha aba ababe bakanacha emete etato
amare amamwamu amarito agocha gotwa embura
bamura bago batano na abasasiroti
amasangu atano asambirwe na abasigari ebaraire
ching'ondi chinene mono chiarire obonyansi
abana bake abarabu bagocha gocharoka
ere otinyiriri etirisa
kera eng'iti ekaroka
kera emunto agaseka
omosasiroti omweng'e agakora enyomba
ere nachiche koria enswe
ere agaaka eng'ondi
bakagenda ase omogondo
abamura bande bachiche gochesa
omoiseke ori okure
barabwo basomire amasakara
amanyinga amabariri amanetu
ere nagore ebitabu ebinge
intwe ntoretu amanyinga
abana abange abaeng'e bagure
omoruoti akagosoria na omokungu oye
omosasiroti amenyete ime ye enyomba enene
omosani akagora ebitabu emerongo etano korwa
etuka
omworokigwa ori achichete ne enyange gochia
eyunibasiti
chibuti ikomi chinyeng'e mono chiare chinchabu
abaorokigwa amagana ebere abang'aini ase esukuru
eria bagoaka obari
omorokia arigete ebitabu bitano na bibere na
egetabu kia kabere kiarigetwe goetera omento gete
ekero kera emunto are omoke na omusere na kera
egento ekiare ekiya
rituko ng'a omorembe orabe ase ense
abaorokigwa abang'aini nkominyoka bare
ekaramu ya Choni iyare emesa igoro
omorangu bwe ekanisa ore imbariri
kera omwana are omomura gose omoiseke
ebara eye enyeng'e nigo ekorwa ase ebengi gochia
nyomba
esayansi ya rire ne eeng'encho mono
igaa gose aria gose kera ase

emiaka erenga omonyagitari one
 omoiseke ori ogotwara ching'iti
 enkoro yo omwana nengiya mono
 okogoro okorabu gokabimba
 omonwa oyo okokwana chiruga ikomi na ibere
 ninde omosacha
 amasakara aya na amakamoku

chiseese echi emerongo etano chimwamu nchirare
 barabwo ebanchete richiko ng'a ebitabu ebire ebireu
 Choni no omonene na omong'aini
 omwana nigo arenge nyaro ye emesa
 abamura baria bagaaka obari na abasiseke aba
 bakaminyoka

APPENDIX D Images

```
Admin@User /c/test
$ diff --normal AdjectiveKam.gf AdjectiveGus.gf
7,8c7,8
< ComparA a np = <
  s = \\g,n => a.s !AComp g n ++ "kuvita" ++ np.s ! npNom ;
---
> ComparA a np = <
  s = \\g,n => a.s !AAdj g n ++ "kobua" ++ np.s ! npNom ;
11,13c11
< UseComparA a = <s = \\g,n=> a.s !AComp g n;isPre = True>;
---
> UseComparA a = <s = \\g,n=> a.s !AAdj g n;isPre = True>;
15c13
< AdjOrd ord = <
---
> AdjOrd ord = <
26c24
< s = \\g,n => a.s !AComp g n ++ a.c2 ++ np.s ! NCase Nom;
---
> s = \\g,n => a.s !AAdj g n ++ a.c2 ++ np.s ! NCase Nom;
29d26
<
```

Figure D.1 diff comparative

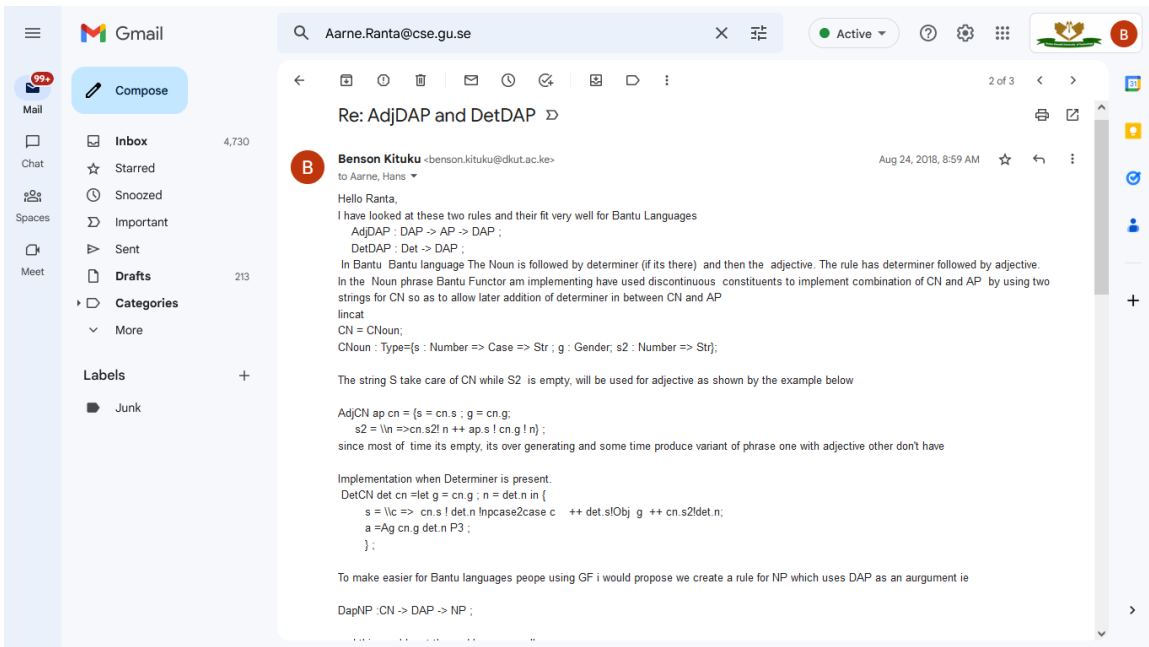


Figure D.2 Proposed new GF rules

APPENDIX E Linearization categories

```
S = {s : Str} ;
QS = {s : QForm => Str} ;
RS = {s : Agr => Str } ;
SSlash = {s : Str ; c2 : Str} ;
-- Sentence
Cl = {s : Polarity => Tense => Anteriority => Str};
ClSlash = { s : Polarity => Tense => Anteriority => Str};
Imp = {s : Polarity => ImpForm => Str} ;
-- Question
QCl = QClause ;
IP = {s : Str ; n : Number} ;
IComp = {s : Str} ;
IDet = {s : Cgender => Str ; n : Number} ;
IQuant = {s : Number => Cgender => Str } ;
-- Relative
RCl = {s : Polarity => Tense => Anteriority => Str};
RP = {s : Cgender=> Number=> Str; a : Agr} ;
-- Verb
VP = ResBantu.VerbPhrase ;
VPSlash = ResBantu.SlashVP ;
Comp = ResBantu.Comp;
-- Adjective
AP = {s : Cgender => Number => Str } ;
-- Noun
CN = CNoun;
NP = ResBantu.NounPhrase ;
Pron = {s: PronForm=>Str; a : Agr};
Det = {s : Cgender => Str ; n : Number ; isPre: Bool} ;
Predet = {s : Cgender =>Str} ;
Ord = { s : Cgender => Str } ;
Num = {s : Cgender => Str ; n : Number } ;
Card = {s : Cgender => Str ; n : Number} ;
Quant = {s : Number => Cgender => Str } ;
DAP = {s : Cgender => Str ; n : Number ; isPre: Bool} ;
-- Numeral
Numeral = {s : CardOrd => Cgender => Str ; n : Number} ;
Digits = {s : CardOrd => Cgender => Str ; n : Number} ;
-- Structural
Conj = {s1,s2 : Str ; n : Number} ;
Subj = {s : Str} ;
Prep = ResBantu.Preposition;
Adv = {s : Agr => Str } ;
-- Open lexical classes, e.g. Lexicon
```

$V, VS, VQ, VA, VV, V2S, V2Q, V2V, V2A = \text{Verb} ;$
 $V2 = \text{Verb} ** \{c2 : \text{Prep}\} ;$
 $V3 = \text{Verb} ** \{c2, c3 : \text{Prep}\} ;$
 $A = \{s : \text{AForm} \Rightarrow \text{Str}\} ;$
 $A2 = \{s : \text{AForm} \Rightarrow \text{Str}\} ** \{c2 : \text{Str}\} ;$
 $N = \{s : \text{Number} \Rightarrow \text{Str} ; g : \text{Cgender}\} ;$
 $N2 = \{s : \text{Number} \Rightarrow \text{Str} ; g : \text{Cgender}\} ** \{c2 : \text{Prep}\} ;$
 $N3 = \{s : \text{Number} \Rightarrow \text{Str} ; g : \text{Cgender}\} ** \{c2, c3 : \text{Prep}\} ;$
 $PN = \{s : \text{Str} ; g : \text{Cgender}\} ;$

APPENDIX F Journal papers

The research had four journal papers, as shown below and links provided as footnotes

1. Kituku, B., Muchemi, L. and Nganga, W., 2016. ⁴⁰A review on machine translation approaches. *Indonesian Journal of Electrical Engineering and Computer Science*, 1(1), pp.182-190.
2. Kituku, B., Nganga, W., & Muchemi, L. (2019). Towards Kikamba Computational Grammar. *Journal of Data Analysis and Information Processing*, 7(04), 250. ⁴¹
3. Kituku, B., Nganga, W., & Muchemi, L. (2021)⁴². Grammar Engineering for the Ekegusii Language in Grammatical Framework. *European Journal of Engineering and Technology Research*, 6(3), 20-29.
4. Kituku, B., Nganga, W., & Muchemi, L. (2022)⁴³. Leveraging on Cross Linguistic Similarities to Reduce Grammar Development Effort for the Under-Resourced Languages: a Case of Kenyan Bantu Languages. 2021 International Conference on Information and Communication Technology for Development for Africa (ICT4DA).

⁴⁰ <http://ijeecs.iaescore.com/index.php/IJEECS/article/view/193/4180>

⁴¹ https://www.scirp.org/html/7-2870288_95871.htm

⁴² <https://www.ejers.org/index.php/ejers/article/view/2382/1060>

⁴³ <https://ieeexplore.ieee.org/abstract/document/9672222>

APPENDIX G: Summary of ported Swahili grammar

a. Swahili rules

```
ProgrVP vp = {s=\ag,pol,tes,ant=>case < tes ,pol> of {
  <Pres, _> => vp.s!ag!pol!Pres!ant;
  <_, _> => auxBe.s!ag!pol!tes!ant ++vp.s!ag!pol!Pres!ant};
  compl=\a => vp.compl!a;
  progV= []; imp =\po,n =>vp.imp!po!n;inf=vp.inf};
lin pot01 = mkNum1 "moja" "kwanza" ** {n = Sg} ;
lin pot110 = regCardOrd "kumi" ** {n = Pl} ;
lin pot111 = regCardone "kumi na" "moja" ** {n = Pl} ;

lin pot1plus d e = { s = table {NCard => \g => d.s ! ten ! NCard ! g ++ "na"++
  e.s ! unit ! NCard ! g ;
  NOrd => \g =>Ordprefix g++ d.s ! ten ! NCard ! g ++ "na"++ e.s !
  unit ! NCard ! g } ;
  n = Pl} ;
lin pot1as2 n = n ;
lin pot2 d = {s = d.s ! hund} ** {n = Pl} ;
lin pot2plus d e = {s = table {
  NCard => \g => d.s ! hund ! NCard ! g ++ "na" ++ e.s !NCard ! g ;
  NOrd => \g =>Ordprefix g++ d.s ! hund ! NCard ! g ++ "na" ++ e.s !
  NCard ! g } ;
  n = Pl} ;
lin pot2as3 n = n ;
lin pot3 n = { s = table {
  NCard => \g => mkCard NCard "elfu"!g ++ n.s ! NCard ! g ;
  NOrd => \g =>Ordprefix g++ mkCard NCard "elfu" ! g ++ n.s ! NCard ! g }
  n = Pl} ;
lin pot3plus n m = { s = table {
  NCard => \g => "elfu" ++ n.s ! NCard !g ++ m.s ! NCard ! g ;
  NOrd => \g =>Ordprefix g++ "elfu" ++ n.s ! NCard !g ++ m.s ! NCard ! g}
  ;
  n = Pl} ;
mk2Dig : Str -> Str -> TDigit = \c,o -> mk3Dig c o Pl ;
mkDig : Str -> TDigit = \c -> mk2Dig c (c) ;

mk3Dig : Str -> Str -> Number -> TDigit = \c,o,n -> {
  s = table {NCard => \g => c ; NOrd => \g =>Ordprefix g ++ o} ;
--Ordprefix g ++
  n = n} ;
```

b. Swahili RE

- For numeral: regNum.For Adjectives: regA, regAdj, cregA
- For Verbs : RegV, iregV,regV