Evaluating the Bantu parametric grammar in grammatical framework using Swahili grammar

Article *in* Indonesian Journal of Electrical Engineering and Computer Science · May 2023

DOI: 10.11591/ijeecs.v30.i2.pp817-824

2 authors:

Benson Kituku
Dedan Kimathi University of Technology

Lawrence Muchemi
University of Nairobi

Indonesian Journal of Electrical Engineering and Computer Science
Vol. 30, No. 2, May 2023, pp. 817~824
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v30.i2.pp817-824

Evaluating the Bantu parametric grammar in grammatical framework using Swahili grammar

Benson Kituku[1], Lawrence Muchemi[2]
[1]Department of Computer Science, School of Computer Science and Information Technology, Dedan Kimathi University of technology, Nyeri, Kenya
[2]Department of Computing and Informatics, Faculty of Science and Technology, University of Nairobi, Nairobi, Kenya

Article Info

Article history:

Received Oct 28, 2021
Revised Jun 6, 2022
Accepted Nov 23, 2022

Keywords:

Bantu
Bootstrapping
Grammar portability
Grammar sharing
Grammatical framework
Swahili

ABSTRACT

Monologue grammar development for under-resourced languages is very slow and laborious (involves creating rules to generate the computational grammar to enable analysis and synthesis of the language(s) in question). However, the need for computational grammar continues to soar in this technology-driven economy for information synthesis and analysis. This paper aims to set up an experiment in the grammatical framework (GF), to evaluate the efficiency and effectiveness of the Bantu parameterized grammar to bootstrap a new grammar for Swahili. The goal is to investigate how this approach of bootstrapping grammar in a multilingual environment is effective and efficient in reducing the development effort. The bootstrapping approach uses the GF morphology-driven approach to develop portable and unique segments of Swahili grammar. The bootstrapped Swahili grammar resulted in a shareability of 100%, 71.11%, 68.75%, and 91.41% at category linearization, paradigms, parameters and syntax rules respectively. The portability was at 15.55%, 18.57%, and 8.59% at paradigms, parameters and syntax rules, respectively. Finally, this paper contributes in: first, provides an approach that leads to an effective and efficient method for developing and bootstrapping computational grammar for the under-resourced Bantu languages. Secondly, the research provided a Swahili grammar.

Corresponding Author:

Benson Kituku
Department of Computer Science, School of Computer Science and Information Technology
Dedan Kimathi University of Technology
Along Nyeri, Mweiga road, Nyeri, Kenya
Email: benson.kituku@dkut.ac.ke

1. INTRODUCTION

The need for multilingual data by information consumers has risen especially in third world countries which are 90% multilingual [1], [2] and a great percentage of the languages are under-resourced. Developing a monolingual computational grammar in such ecosystems requires much effort, especially if it is to be developed from scratch [3]. This effort is a stumbling block in grammar development for under-resourced languages especially the spoken Bantu languages [4]. Yet, in the technology-driven economy, these grammars become drivers in sharing and gathering of information [5] when used to make natural language processing (NLP) tools and applications more so for these under resourced languages where minimal corpora exist that cannot be effective in making data driven NLP tools.

To address the above challenge, grammar engineering (GE) strategies have been used to develop shared and portable grammar based on cross linguistic similarities between two or more languages [6]. So far, some GE attempts have been made; for example, the morphological analyzer made using the rule-based approach [7] for Zulu and Xhosa languages as well as the use of grammar engineering strategies such as grammar sharing and grammar porting [8]–[10]. 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; Grammar sharing is creating a commonly shared grammar (congruent) for all similar lexical, parameter and syntax rules of the family's languages [10]. In this case, a shared and portable Bantu parametric grammar was developed using the shared cross linguistic principles and parameters of Ekegusii and Kikamba languages in the grammatical framework (GF) [11]. Therefore, this paper aims to bootstrap Swahili grammar into the Bantu parameterized grammar and evaluate the effectiveness and efficiency of the approach in reducing the development effort.

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 [12], [13]. In using the rule-based bootstrap, the carefully chosen seed will be the shared Bantu parametric grammar to be leveraged in bootstrapping the unique components of the Swahili grammar, thus reducing the development effort in terms of rule-base and time.

2. RELATED RESEARCH

GF is a multilingual development toolkit that has one abstract syntax and several parallel concrete syntaxes, one for each language [14]. The abstract syntax constitutes a finite set of abstract categories with a corresponding finite set of abstract functions to implement the categories whereas the parallel concrete syntaxes are parallel multiple context-free grammars (PMCFG) [15]. The definition of PMCFG is given by a 5-tuple equation as shown in Definition 1 below. 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 ecosystem [16].

Definition 1 parallel concrete syntaxes

$PMCGF = (N^C, F^C, T, P, L)$

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 linearizations.

The GF RGL repository contains over 48 parallel grammars. The RGL resource library consists of several modules subdivided into three major groups: lexical, morphology, and syntax modules. 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) [17]. *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 and parameters that are common for all languages present in GF. GF provides 500 lexical items for grammar testing and the core syntax defines 200 functions and 60 categories, which form declarative, question and imperative sentences [14]. The Bantu parametric grammar was developed using the parametric modules also known as Functor in GF. Its development used already existing independent grammar for Kikamba [4] and Ekegusii [18].

The Swahili language belongs to the large Bantu family, it's agglutinative and tonal. The language is classified as G42 [19] and has the following dialects in the Kenyan coast: Amu, Mvita, Mlima, and Unguja. The morphology is suffixing and prefixing of the root plus affixes and is affected by Morph phonological transformation. The noun classes also known as genders affect the morphology of all categories through an agreement prefixing morpheme sometimes referred to as concord [20]–[22]. In terms of syntax, Swahili has subject verb object (SVO) as the main topology for a sentence [21], [23], [24]. The subject is a noun, while the verb phrase represents the verb. The argument of the verb phrase depending on the verb valence forms the object that can be a noun phrase or verb phrase or both. The verb can act as a sentence since it has morphemes for subject marker and object marker that stands in place of sentence subject and object respectively. Swahili language has a stable descriptive grammar and many grammar books due to extensive years of grammar research and is widely known to many people compared to the other two languages chosen [20]. These aspects availed a pool of different people who examined the output and validated the computational grammar.

3. RESEARCH METHOD

The Swahili grammar had been independently developed earlier in GF [25] to avoid biases being carried over in the bootstrapping experiment. The experiment used the shared Bantu parameterized grammar (congruent grammar) as the leverage seed, to bootstrap Swahili portable and unique segments of the grammar as shown in Figure 1.

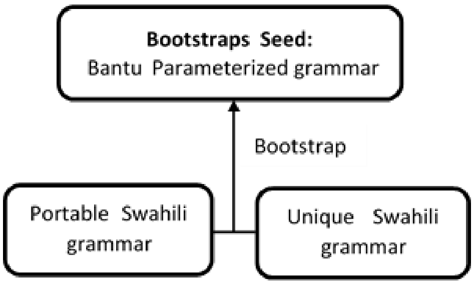

Figure 1. Bootstrap structure

The experiment involved defining and modifying the unique and portable grammar segments respectively. The GF morphology-driven approach was used, where the lexicon and categories linearizations were defined first, then the regular expressions (paradigms) for the inflection of the different categories and finally, the syntax production rules [26]. The process of defining and modifying ensured naming conventions, features/parameters description such as gender systems of nominal classes and phenomena analysis were similar to the congruent grammar so as to benefit from sharing [9]. Since each function developed had to be tested iteratively before the next function, the evolutionary prototype method was used [27]. The unique and portable grammar segments were bootstrapped to the Bantu parameterized grammar and then the GF regression testing procedure was applied as shown in Figure 2 [14], [28]. This procedure involves using the English comments in the GF abstract syntax for each function that shows what it parses as test data. The test data was translated by an expert to Swahili and cross-checked by a linguist for correctness forming the gold standard, in addition, the same comments were subjected to the developed function for machine translations and then the two outputs were compared. If errors resulted from the process, the functions and rules were refined in an iterative manner until the errors were resolved. However, if the errors were from the congruent grammar, the functions and/or rules were moved to either portable or unique grammar depending on similarities and the testing procedure repeated until errors were eliminated. The experiment steps are summarized in Figure 3.

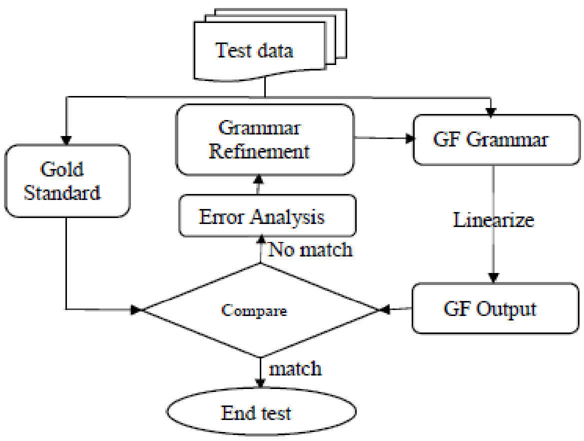

Figure 2. Testing process

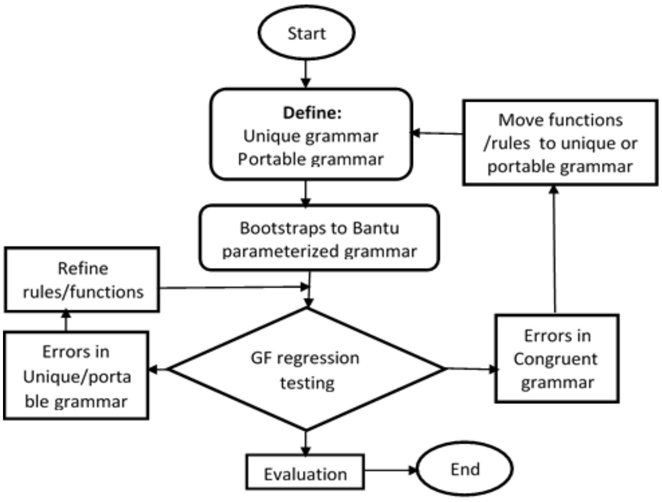

Figure 3. Bootstrap experiment

To evaluate the effectiveness and efficiency of using the Bantu parametric grammar in reducing development effort for Swahili two metrics were used shared and modified rules expressed as a percentage [10], [29], [30]. The shared rules measure production rules common among the three Bantu grammars whereas rules modification measures the number of rules that have been modified or deleted in the Bantu parametric grammar in order to adapt the Swahili grammar (bootstraps). 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.

4. RESULTS AND DISCUSSION

Swahili has the genders and concord system, at the morphology level, like the 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 has eight of them. Overall, Swahili shared 32 paradigms with the Bantu parameterized grammar, translating to 71.11%, as per Table 1. 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 more rapid development. Only 13.33% of the paradigms were uniquely defined, which is a small effort that can take little time compared with defining 100% of the paradigms.

Table 1 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. 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 1. 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 2 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. 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.

In terms of grammar sharing, past studies show Swahili has performed better as illustrated by the following. The Romance and Scandinavian families of languages though quantified work in terms of lines of codes resulting in grammar sharing of 75% and 90% at syntax, respectively [16], Swahili grammar has performed better. Only limited to features and systems, the functional approach Bulgarian and Russian grammars shared 76% of the features and 72% of the systems. A 66% grammar sharing was achieved in the regulus framework where 65 rules speech translation system involving English, Japanese and Finnish languages [8], in addition, any pairing with Greek resulted in 75% sharing [31], where our system has outperformed them. A 54% sharing of types was realized by bootstrapping Wambaya grammar [30] to the LinGO grammar matrix. Moreover, in portability of Swahili grammar no rules were deleted or added as compared with the English Microsoft NLP systems were 10.1%, 10.7 %, 7.8% of the English rules were deleted and 7.8%, 8.6%, 2.3% were added to develop Spanish, German and French grammar at syntax respectively a [29].

Table 2. 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**

Therefore, the fact that the languages used in the development of the congruent and bootstrap grammars were picked from different geographical areas and different Guthrie [19] zones and resulted 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. The research has shown this approach to be efficient and effective that can be used to bootstrap grammar development for under resourced languages, thus reducing the effort required in ordinary settings. 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 4 involved lexicon definition and development of the 28.89%, 31.25% and 8.59% paradigms, parameters and syntax rules, respectively which 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 grammar for low-resourced Bantu languages by significantly reducing the effort needed for such work. The pseudocode in Algorithm 1 summarises the approach. It has five steps namely:

- 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.

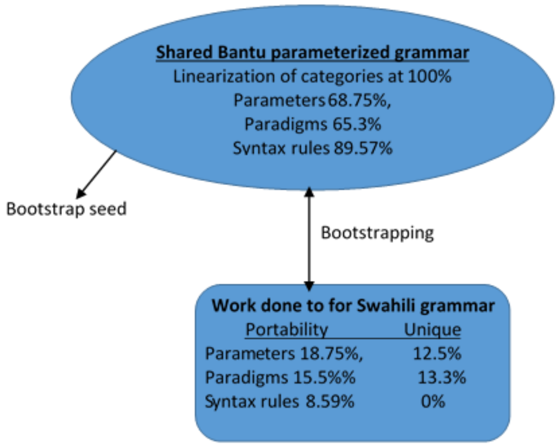

Figure 1. Bootstrapping Swahili

Algorithm 1: The approach pseudocode

```
The Approach of Bootstrapping Multilingual Grammar Development (i,n)
Initialize languages n -- under-resourced languages family
Initialize grammar formalism
For lang == 1 to i Do -- i languages for developing shared grammar
descriptive grammar analysis – for cross-linguistic similarities
missing gaps filling --language analysis and translation
Shared <-- extract shared principles and parameters
Portable <-- extract portable principles and parameters
Endfor
if Shared == True
develop congruent parameterized grammar
else If Portable == True
develop portable parameterized grammar
else
while lang < i -- no sharing or portability
Develop language-specific grammar
Endwhile
EndIf
Metrics <--evaluate congruent parameterized grammar reusability
Return metrics
EndIf
For lang == i+1 to n Do
Analysis of the descriptive grammar
Extract portable and unique grammar
bootstrap grammar
Metrics <-- Evaluate extendibility -- to congruent grammar
Return metrics
Endfor
For lang == 1 to n Do
Metrics <-- use machine translation to evaluate the performance
Return metrics
Endfor
```

5. CONCLUSION

This research used the Swahili language as a testbed for the generalization and reusability of the Bantu parametrized grammar. Swahili's effort involved defining 13.33% and 12.5% of paradigms and parameters respectively and modifying 15.55%, 18.75%, and 8.59% 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. 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.

The research has contributed by providing an open resource Swahili grammar, that is available for researchers' use. Grammars in GF can also be used to develop multilingual applications for controlled languages. Moreover, this grammar offers an opportunity for translation among Bantu languages. Again the approach has shown to be an effective and efficient approach to developing computational grammar for the Bantu languages which are under-resourced. Since this bootstrapping methodology has proved to be effective in reducing effort for developing multilingual grammar, the research recommends the future direction to involve the bootstrapping of other Bantu languages apart from those used in the study.

ACKNOWLEDGEMENTS

Authors thanks the Swahili linguist, experts who help in the testing phase of this research.

REFERENCES

[1] A. Das and D. Saha, "Improvement of electronic governance and mobile governance in multilingual countries with digital etymology using Sanskrit grammar," in *Social Transformation – Digital Way. CSI 2018. Communications in Computer and Information Science*, Singapore: Springer, 2018, pp. 523–530.

[2] S. Patankar, M. Phadke, and S. Devane, "Wiki sense bag creation using multilingual word sense disambiguation," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 1, pp. 319–326, Mar. 2022, doi: 10.11591/ijai.v11.i1.pp319-326.

[3] E. M. Bender, D. Flickinger, and S. Oepen, "Grammar engineering for linguistic hypothesis testing," in *Proceedings of the Texas Linguistics Society X Conference: Computational linguistics for less-studied languages*, 2008, pp. 1–26.

[4] B. Kituku, W. Nganga, and L. Muchemi, "Towards Kikamba computational grammar," *Journal of Data Analysis and Information Processing*, vol. 07, no. 04, pp. 250–275, 2019, doi: 10.4236/jdaip.2019.74015.

[5] B. Ghilic-Micu, M. Mircea, and M. Stoica, "Knowledge based economy–technological perspective: implications and solutions for agility improvement and innovation achievement in higher education," *The AMFITEATRU ECONOMIC journal*, vol. 13, no. 30, pp. 404–419, 2011.

[6] S. E. Wright, "Trends in language engineering," 1998.

[7] L. Pretorius and S. Bosch, "Exploiting cross-linguistic similarities in Zulu and Xhosa computational morphology," in *Proceedings of the First Workshop on Language Technologies for African Languages - AfLaT '09*, 2009, pp. 96–103, doi: 10.3115/1564508.1564526.

[8] R. Kim, M. Dalrymple, R. M. Kaplan, T. H. King, H. Masuichi, and T. Ohkuma, "Multilingual grammar development via grammar porting," in *ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development*, 2003, pp. 49–56.

[9] A. Ranta, "The GF resource grammar library," *Linguistic Issues in Language Technology*, vol. 2, Dec. 2009, doi: 10.33011/lilt.v2i.1205.

[10] M. Santaholma, "Grammar sharing techniques for rule-based multilingual NLP systems," in *Proceedings of the 16th Nordic Conference of Computational Linguistics, NODALIDA 2007*, 2007, pp. 253–260.

[11] B. Kituku, W. Nganga, and L. Muchemi, "Leveraging on cross linguistic similarities to reduce grammar development effort for the under-resourced languages: a case of Kenyan Bantu languages," in *2021 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*, Nov. 2021, pp. 83–88, doi: 10.1109/ICT4DA53266.2021.9672222.

[12] A. R. Henderson, "The bootstrap: a technique for data-driven statistics. Using computer-intensive analyses to explore experimental data," *Clinica Chimica Acta*, vol. 359, no. 1–2, pp. 1–26, Sep. 2005, doi: 10.1016/j.cccn.2005.04.002.

[13] R. Jones, A. Mccallum, K. Nigam, and E. Riloff, "Bootstrapping for text learning tasks," *In IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*, pp. 52–63, 1999.

[14] A. Ranta, *Grammatical Framework: programming with multilingual grammars*. Stanford: CSLI Publications, 2011.

[15] P. Ljunglöf, "Expressivity and complexity of the grammatical framework," Ph.D. dissertation, Chalmers University, 2004.

[16] A. Ranta, "Type theory and universal grammar," *Philosophia Scientae*, no. CS 6, pp. 115–131, Sep. 2006, doi: 10.4000/philosophiascientiae.415.

[17] G. Détrez and A. Ranta, "Smart paradigms and the predictability and complexity of inflectional morphology," in *EACL 2012 - 13th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings*, 2012, pp. 645–653.

[18] B. Kituku, W. Nganga, and L. Muchemi, "Grammar engineering for the ekegusii language in grammatical framework," *European Journal of Engineering and Technology Research*, vol. 6, no. 3, pp. 20–29, Mar. 2021, doi: 10.24018/ejeng.2021.6.3.2382.

[19] M. Guthrie, "Gender, number and person in Bantu languages," *Bulletin of the School of Oriental and African Studies*, vol. 12, no. 3–4, pp. 847–856, Oct. 1948, doi: 10.1017/S0041977X00083427.

[20] E. O. Ashton, *Swahili grammar (including intonation)*. 1944.

[21] K. Njogu and D. K. Nganje, *KiSwahili kwa vyuo vya ualimu*. Jomo Kenyatta Foundation, 2006.

[22] M. Y. Mwamzandi, "Swahili word order choices: insights from information structure," Ph.D. dissertation, The University of Texas at Arlington, 2014.

[23] K. U. D. S. U. Deen, "The acquisition of Nairobi Swahili: the morphosyntax of inflectional prefixes and subjects," Ph.D. dissertation, University of California, 2002.

[24] L. Marten, "Formal syntax and African languages: dynamic syntax case studies of Swahili, Bemba, and Hadiyya," *SOAS Working Papers in Linguistics Volumes*, vol. 16, pp. 195–213, 2013, [Online]. Available: https://www.soas.ac.uk/linguistics/research/workingpapers/volume-16/file88040.pdf.

[25] B. Kituku, "Grammar engineering for Swahili," *International Journal of Computer and Information Technology*, vol. 08, no. 06, 2019, [Online]. Available: http://repository.dkut.ac.ke:8080/xmlui/handle/123456789/1280.

[26] A. Ranta, "Modular grammar engineering in GF," *Research on Language and Computation*, vol. 5, no. 2, pp. 133–158, Sep. 2007, doi: 10.1007/s11168-007-9030-6.

[27] M. Carr and J. Verner, "Prototyping and software development approaches." 1997.

[28] J. Camilleri, "A computational grammar and lexicon for Maltese," Master's thesis, Chalmers University of Technology, 2013.

[29] M. Gamon, C. Lozano, J. Pinkham, and T. Reutter, "Practical experience with grammar sharing in multilingual NLP," in *From Research to Commercial Applications: Making NLP Work in Practice, ACL-EACL 1997 Workshop*, 1997, pp. 49–56.

[30] E. M. Bender, "Evaluating a crosslinguistic grammar resource: a case study of Wambaya," in *ACL-08: HLT - 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 2008, pp. 977–985.

[31] M. Santaholma, "Multilingual grammar resources in multilingual application development," in *Proceedings of the Workshop on Grammar Engineering Across Frameworks - GEAF '08*, 2008, pp. 25–32, doi: 10.3115/1611546.1611550.

BIOGRAPHIES OF AUTHORS

Benson Kituku is a lecturer at the department of Computer Science at the Dedan Kimathi University of Technology. He holds a B.Sc degree in computer science from Maseno University, Kenya. M.Sc and Ph.D. in Computer science from the University of Nairobi, Kenya. His research interests are natural language processing, the theory of computing and compiler complexities. He can be contacted at email: benson.kituku@dkut.ac.ke.

Lawrence Muchemi is currently a Senior Lecturer in the department of computing and Informatic at, university of Nairobi. He received B.Technology (Honors) Engineering and M.Phil. Engineering- AI systems from Moi University, Kenya. He obtained Ph.D. Computer Science from the University of Nairobi. His research area is artificial intelligence and more so natural language processing. He can be contacted at email: lmuchemi@uonbi.ac.ke.