Leveraging on Cross Linguistic Similarities to Reduce Grammar Development Effort for the Under-Resourced Languages: a Case of Kenyan Bantu Languages

Benson Kituku
Computer Science Department
Dedan Kimathi University of Technology
Nyeri, Kenya
benson.kituku@dkut.ac.ke

Wanjiku Nganga
Department of Computing and Informatic
University of Nairobi
Nairobi, Kenya
wanjiku.nganga@uonbi.ac.ke

Lawrence Muchemi
Department of Computing and Informatic
University of Nairobi
Nairobi, Kenya
lmuchemi@uonbi.ac.ke

***Abstract*— Rule-based grammar development is labor-intensive in terms of time and knowledge requirements, especially for complex morphology and under-resourced languages. Notwithstanding, these grammars are needed for deep natural language processing, generation of well-formed output, or both. To address the challenge, this paper seeks to develop shared multilingual wide-coverage grammar for a subset of Kenyan Bantu languages in Grammatical Framework (GF) by leveraging on cross linguistic similarities using the grammar engineering strategies: grammar porting and grammar sharing. The shared grammar was developed using the morphology-driven approach, where the lexicons are defined first, followed by inflection regular expression and finally the syntax production rules. The resulting congruent Bantu parameterized grammar had shareability for category linearizations, parameters, paradigms, and syntax rules of 100%, 68.75%, 65.3% and 89.57%, respectively, while portability (modification) was exhibited in paradigms, parameter plus syntax rules at 14.29%, 18.75% and 10.43% respectively. The research concludes leveraging on the cross-linguistic similarities of principles and parameters significantly reduces multilingual grammar's development effort and contributes by developing the Bantu parametrized grammar which demonstrates how the effort of developing the rule base has been significantly reduced in languages where data is a scarce commodity.**

Keywords— Grammar sharing, Grammar porting, Bantu languages, Functor, Under-resourced languages, Grammatical Framework, Complex morphology.

I. Introduction

The creation of natural language grammars using Data driven approaches such as the state of art neural approach though shown to learn cross linguistic syntactic structures and even represent them [1], they have not been effective for under resourced and complex morphology languages such as Bantu languages, Because, Firstly, the models treats complex morphology words (lexicon) as a single feature without considering morphemes with distinct meanings (multiple features of a word) thus, do not capture dependencies in the word's morphemes resulting in data sparsity [2]. Secondly, digitized corpora are a scarce commodity for the spoken Kenyan Bantu languages [3]. Furthermore, the little available corpora may not be as helpful since they suffer from data sparseness. Finally, its creation is a costly affair in terms of human effort, time and monetary resources and these languages lack experts that can generate corpora [4, 5]. The alternative is creating grammars using the traditional rule base approach that is a knowledge-intensive, expensive, slow and laborious affair in addition to limited coverage. This challenge has led to a language technology digital divide between the under-resourced and rich-resourced languages. Therefore, it is paramount to enhance languages' digital visibility and viability for the under-resourced languages by developing NLP resources, applications and tools. Thus enabling fair competition in the technology-driven economies [6].

This paper seeks to demonstrate how to reduce the effort required to develop grammars in the classical handcrafted rules in a multilingual environment using grammar sharing and grammar porting by leveraging on cross-linguistic similarities of principles and parameters. This research achieves one of the main language engineering objectives of developing shared language resources where the output becomes the foundation or support tool for the development of other Natural Language Processing (NLP) tools and resources [7]. Further, the cross linguistic morphological similarities are demonstrated and implemented for the first time. This was done in the grammatical framework (GF) using two Kenyan Bantu languages: Ekegusii and Kikamba. Grammar porting involves adapting the structure of already developed grammar rules to develop a new independent parallel grammar [5], while grammar sharing is creating a commonly shared grammar for all similar lexical, morphological and syntax rules of the languages in question enabling sharing of naming convention, features description and phenomena analysis [8]. These two grammar engineering strategies have the following advantages: reduce the size of the grammar rule base, reduce development time, ensure coherence grammar description in different languages, and reduce duplication effort [5, 7, 8].

II. Related Research

A. Related Work

Grammar sharing was used in the MCC multilingual project to [9] develop simple noun phrases shared grammar prototype for Arabic, Japanese, English, French and German languages using unification categorical grammar. Though the project concluded the concept's viability, the grammar

2021 International Conference on Information and Communication Technology for Development for Africa (ICT4DA) | 978-1-6654-3666-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICT4DA53266.2021.9672222

978-1-6654-3666-3/21/$31.00 ©2021 IEEE

coverage scope was shallow. Furthermore, the Microsoft Natural Language Processing project proved development time reduction by modeling German, Spanish and French grammars through adding or deleting rules in the initial English [10]. Lexical-function grammar formalism used Grammar sharing in parallel grammar projects [11] and it concluded the syntax parallelism within the six grammars involved could be used to bootstrap a new language. Using the functionalist approach, reference [12] showed that Bulgarian and Russian languages shared 76% of the features while Bulgarian, Czech, Russian shared 92%, 84%, and 75% with English grammar respectively. The Grammatical Framework (GF) resource grammar project [13] shows syntax coding sharing of 75% and 90% within the Romance (French, Italian, and Spanish) and Scandinavian (Swedish, Norwegian, and Danish) families grammars using the parametrized modules in GF. Santaolma [8,14] bootstrapped Greek Grammar to the multilingual spoken grammar developed in the Regulus grammar framework for Finish English and Japanese grammars, showing a 54% sharing of the rules. The researchers in [15] jump-started the development of Wambaya grammar using the grammars presented in the LinGO Grammar Matrix based on the Head-Driven Phrase Structure Grammar theory and the resultant grammar could assign correctly 76% sentences of natural text after 5.5 person-weeks work of grammar development.

The French and Spanish systems were manually ported from the English system in the core language engine system [16], where 80% of syntactic rules were ported. The Japanese ParGram grammar was ported to Korean Grammar and within two month-person work, good accuracy was achieved [17] while the Japanese took two years. English grammar was ported to Italian [18], a process that took five months for two people to complete. Finally, the Zulu morphological analyzer was ported to the Xhosa analyzer [19], where they conclude this could be a fruitful way of making resources for low-resourced languages using the Xerox lexicon compiler.

These strategies have been applied in the rich resourced languages as per the above review except for Wambaya languages, which is not a complex morphology language and there are no statistics to show the possibility of extending to wide coverage grammar in Zulu and Xhosa morphological analyzer.

B. Grammatical Framework (GF)

This work has been implemented in GF because it has a multilingual grammar development environment [13] and a powerful Curry [20] based formalism structure of using tecto-grammatical (abstract syntax) and pheno-grammatical (concrete syntax) structures. Further, apart from developing the natural language grammars (resource grammar), domain-specific grammars are developed on top of the resource grammar [13]. The domain grammar writer uses resource library grammar without linguistic knowledge of the grammar, thus speeding up the development of language resources (applications and tools). In addition, these resource grammars are also used for natural language processing tasks such as machine translation, multilingual analysis, multilingual generation, software localization, natural language interfaces, spoken dialogue systems, etc. grammar [21, 22]. Finally, GF implements Grammar porting and sharing via its module known as a Functor that uses parameters in implementing core grammar thus referred to as a parameterized module. This suits Bantu languages because of the many parameters, especially the genders and concord (agreements) resulting from Complex morphology [3].

The GF shared abstract syntax defines categories (Cat) of trees and functions (Fun) to implement those trees, while the set of concrete syntaxes (one for each language) define linearization of the categories (lincat) and linearization of the function (lin) stated in the abstract syntax [13,23] as exemplified using category Noun (N) below.

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

Parameters are objects of some type that encodes categories features and are defined using the keyword *param*. For example, the noun in Bantu languages has a parameter number with singular and plural values. Below is an example of a parameter definition.

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

When a category has more than one parameter, a data structure record gathers them. For example, the category noun in Bantu languages can be defined as shown below and means a table from number to strings and has inherent features of class gender (functions over parameters) [13].

N = {s: Number => Str; g: class_gender};

GF parametrized modules, or Functor f, is a function which maps every element of domain to an element of function f in the codomain[13]. The Bantu Functor domain will contain the type of operations or functions and shared definition of the Bantu family languages, while the codomain will contain the actual definition of the functions. The structure of the Functor is shown in Figure 1.

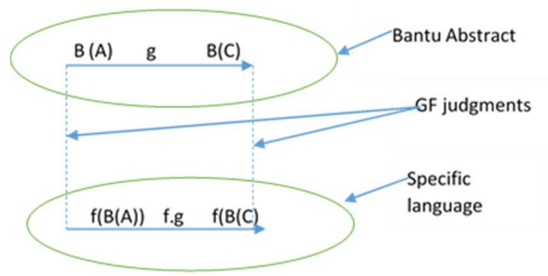

Figure 1 Bantu Functor structure

C. Kikamba and Ekegusii languages

Ekegusii and Kikamba are great Lake and Eastern Bantu languages [24, 25] and grouped E42 and E55 [26] respectively. The two are agglutinative languages and their morphology uses the nominal class system based on morphology (affix to a noun stem) [27-31]. In this research, a pair of singular and plural noun classes markers will be considered as a gender (way of categorizing nouns) based on previous arguments [32-33], Furthermore, the nominal pairing and gender assignment will follow the work in [34,35]. Combining the affixes and stem in some parts of speech tags is affected by Morpho-phonological transformation. The two languages have a similar dominant topology in a sentence that is subject-verb-object (SVO) [27, 30]. Several descriptive

Grammars exists for these languages [27-31] and were used to develop their already existing independent computational grammars in GF and for more information, see [3,36]

III. Bantu parameterized grammar development

The methodology involved four main stages namely: Developing the independent Kikamba and Ekegusii grammars, extracting the congruent, portable and unique segments of the grammar, then developing the Bantu parameterized grammar in GF plus evaluating it.

The Ekegusii and the Kikamba computational grammars had been independently developed earlier see [3,36] so as to ensure no biases were carried in developing the Bantu parameterized grammar. The lexicon definitions, paradigms(morphology) and the production rules for the syntax were arranged similarly and sequentially in each similar module for the two grammar (i.e., same format and order in noun module for Ekegusii and Kikamba). This kind of arrangement enabled the Linux operating system diff command to extract similarities and differences between similar GF modules of Kikamba and Ekegusii. The command has been used for similar GF work [22].

The shared Bantu parameterized grammar was developed using similar production rules and paradigms, parameters and linearization categories. Based on similar structures of the paradigms and production rules, the grammar remainder was adapted based to generate the portable grammar; the rest formed unique grammars. The grammar development adopted the GF morphology-driven strategy and modular-driven development, a bottom-up method

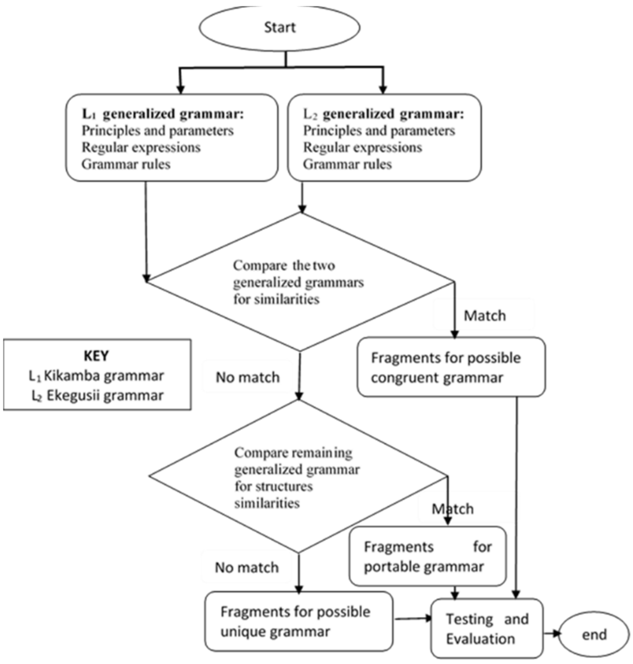

Figure 2 Experiment process

. It first defines the lexicon, then smart paradigms based on the regular expression and their respective linearization categories before working on the syntax[22]. The evolutionary prototype model [37] was used because each function developed had to be iteratively tested to ensure it works 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 [1] Graphviz tool was used. The experimental steps are shown in Figure 2.

A. Morphology

The nominal class system was coded as genders. The coding is of GX, where G stands for gender and X is a number starting from one as per Table 1. Each gender combined two nominal classes based on parameter number (singular and plural) and separated by an underscore. The "blank"- means gender does not exist in a particular language. This coding was done to ensure uniformity, consistency, easy maintainability of the grammar, and reduce the effort required to bootstrapping a new grammar. Kikamba and Ekegusii have 10 and 11 genders respectively. The genders form portable grammar segments because they have a similar structure but have different morphemes.

Table 1 Gender Coding

GF coding	Kikamba	Ekegusii
G1	mu_a	omo_aba
G2	mu_mi	omo_eme
G3	i_ma	e_ci
G4	ki_i	eri_ama
G5	ka_tu	ege_ebi
G6	va_ku	oro_ci
G7	n_n	aka_ebi
G8	u_ma	obo_ama
G9	u_n	oko_ama
G10	ku_ma	aa
G11	-	ama_ama

Generally, lexeme definition for linearization of each category followed a similar structure and involved: Definition of the linearization category, the low-level paradigm, the lexeme for the category and parameter for the category (some had others did not have) as exemplified below for category noun where woman_N is the linearization, *regN* is the noun paradigm for generating an inflection table; "omosubati" is the lexeme for a woman in Ekegusii language, while oma_aba is the parameter gender to which the noun belongs. For more details see [3, 36].

woman_N = regN "omosubati" omo_aba;

In each category, the category linearization, parameters and paradigms were compared and assigned either to shareable or portable(adaptable) segments of the grammar depending on the similarities index. Table 2 is an example of a category noun where all fragments were shared except three regular expressions.

Table 2 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

[1] https://graphviz.org/

B. Syntax

Most of the rules were shared, only in adjective and numeral modules in GF were rules were ported. The reason is that Kikamba has a comparative adjective at morphology while in Ekegusii is at syntax and Ekegusii doesn't have numerals six to nine but a repetition of one to five while that is not the case in Kikamba.

C. Testing and Evaluations

The Bantu parameterized grammar was tested using the GF regression process [22,38] summarized in Figure 3 to improve quality, correctness and accuracy.

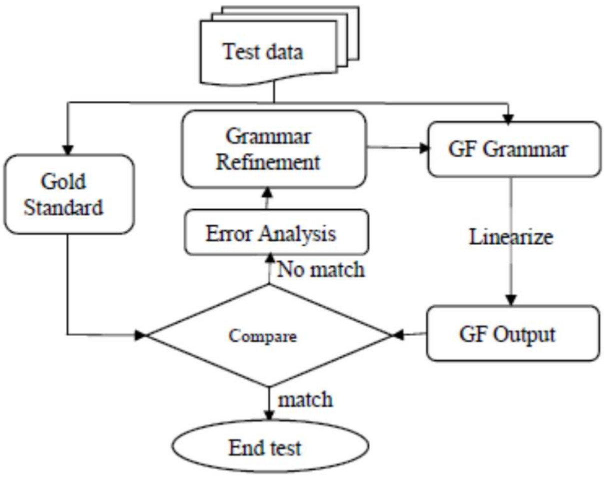

Figure 3 Testing process

In GF, 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. Figure 4 below shows an extracted comment "big house" from the abstract syntax for the function AdjCN that makes an adjective phrase from a common noun.

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

Figure 4 Comments example

A human expert translated the comments into Kikamba or Ekegusii and were cross-checked by linguists to ensure correctness. These translations formed the Gold standard. The specific features/functions were implemented either in the congruent grammar o or in the unique or portable grammar for each language. Thereafter, the comments were parsed from English and linearized using the constructed function. This GF output was compared with the gold standard and If the two differ, then the GF function was refined until the two translations were the same and the regression test re-run is repeated each time refinement is done to ensure no new errors.

D. Evaluation metrics

There are no standard metrics for evaluating grammar shareability and portability due to different grammar formalisms and development tools used for each specific development [14]. The metrics used so far are: Shared rules[7,14,15], modification of rules [10,15] and Development time [12,15,18,23]. Shared rules measure production rules common among grammars in terms of percentage or number or line of codes or number of rules. Rules modification measures the number of rules that have been modified or deleted in order to adapt a new grammar. Development time measures the time used to develop the grammar in terms of weeks and months per person.

The reference [15,17,18,22] just gives estimates of the development time not accurate time. Reference [18] argues 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, due to the above challenges, the metric was not used.

Consequently, to evaluate the Bantu parametrized grammar, shared and modified the number of rules metrics were used and expressed as a percentage

IV. RESULTS AND DISCUSSION

A. Morphology shareability and portability

The Bantu parameterized grammar had thirty-seven [2] categories sharing their linearization (inflections). 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 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.

For parameters, most of them were 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 differed hence shared at naming convention thus adapted (modification) to suit the current grammar. Table 3 below demonstrates less effort needed in defining parameters because 68.75% of them 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 uniquely for each grammar. The means parameters rule-base was reduced by 68.75% in the Bantu parameters grammar, implying less time and effort in

[2] Housed in the BantuCat module

defining them, plus the standardization of 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 3 Paradigms and parameters percentages

Segment	Paradigms Count	Paradigms %	Parameters Count	Parameters %
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 3 represents 20.41% unique grammar because, First, Ekegusii cardinal numeral end at five rather than nine; hence, extra paradigms for constructing numeral six through nine. Secondly, Verbs had unique paradigms because of unique infix morpheme strings for derivational morphology. 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 3 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. Therefore, the two grammars sharing over 65% of paradigms implies a reduction in the effort to define inflection tables.

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. This is a significant reduction of the rule-base that 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 effort 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 development. Therefore, using this approach to develop the Bantu parameterized grammar helped accelerate the morphology definition in a cost-efficiently way.

B. Syntax Shareability and Portability

Table 4 shows the result of syntax rules shared and modified (portable) represented per module in the GF resource grammar library (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 (linking verb and the action verb), which in Kikamba are fused. 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 shareability and portability

GF modules	Rules implemented	Shareability Rules	Shareability %	Portability Rules	Portability %
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), significantly reducing 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. Finally, the grammar is available at the main [3]researcher and [4]GF Github repositories.

The Bantu parametrized grammar performed better in terms of grammar sharing than the following: 75% in Romance languages [13], 76% and 72% of features and systems sharing between Bulgarian and Russian grammars [12], English, Japanese and Finnish languages rule speech translation system in Regulus framework with sharing of 65% [8], in addition to pairing Greek with the three languages in same Regulus framework still producing 75% sharing less than our congruent grammar[14]. LinGO Grammar Matrix grammars had a 54% type sharing with Wambaya less than our work [15]. Finally, Scandinavian languages had a similar sharing of 90% though quantified using the line of codes [13].

V. CONCLUSION

The paper concludes leveraging on the cross-linguistic similarities of principles and parameters significantly reduces multilingual grammar's development effort using the grammar engineering strategies in the Kenyan Bantu languages**.** The research has 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

[3] https://github.com/kitukb/gf-rgl

[4] https://github.com/GrammaticalFramework/

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. 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 in 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.

REFERENCES

[1] E.A. Chi, J. Hewitt, C.D . Manning. Finding universal grammatical relations in multilingual BERT. arXiv preprint arXiv:2005.04511. May 2020.

[2] E.M. Bender, Linguistically naïve!= language independent: Why NLP needs linguistic typology. InProceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous? pp. 26-32, Mar 2009.

[3] B. Kituku, W. Nganga, and L. Muchemi, Towards Kikamba Computational Grammar. Journal of Data Analysis and Information Processing. Vol 7(04), pp. 250-275, Oct 2019.

[4] J. Muhirwe, Towards human language technologies for under-resourced languages. Strengthening the Role of ICT in Development. 2007.

[5] M.E. Santaholma, *Efficient development of grammars for multilingual rule-based applications* (Doctoral dissertation, University of Geneva).2010.

[6] S. Krauwer The basic language resource kit (BLARK) as the first milestone for the language resources roadmap. Proceedings of SPECOM 2003. Pp 8-15,Oct 2003.

[7] S.E. Wright, "Trends in language engineering." *Terminology in Advanced Microcomputer Applications, TAMA* 98, 2002.

[8] M.E. Santaholma, Grammar sharing techniques for rule-based multilingual NLP systems. InProceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA), 2007.

[9] M. Kameyama, "Atomization in grammar sharing." 26th Annual Meeting of the Association for Computational Linguistics. pp. 194-203, 1988.

[10] M. Gamon, C. Lozano, J. Pinkham, and T. Reutter, Practical Experience with Grammar Sharing in Multilingual NLP. Proceedings of ACL/EACL, Madrid, Spain. 1997.

[11] M. Butt, H. Dyvik, T.H. King, H. Masuichi, and C. Rohrer. The parallel grammar project. In COLING-02: Grammar Engineering and Evaluation. 2002.

[12] J. A. Bateman, I. Kruijff-Korbayová, and G. Kruijff. "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 2005, pp.191-219.

[13] A. Ranta, "Modular grammar engineering in GF." *Research on Language and Computation* 5.2, pp 133-158. 2007.

[14] M.E. Santaholma, "Multilingual grammar resources in multilingual application development." *Coling 2008, Proceedings of the Workshop on Grammar Engineering Across Frameworks (GEAF)*. 2008.

[15] E. M. Bender, "Evaluating a crosslinguistic grammar resource: A case study of Wambaya." *Proceedings of ACL-08: HLT*. 2008.

[16] M. Rayner, D. Carter, and P. Bouillon,. Adapting the core language engine to french and Spanish. In Proceedings of NLP-IA-96, Moncton, New Brunswick, May 1996.

[17] R. Kim, M. Dalrymple, R.M. Kaplan, T.H. King, H. Masuichi and T. Ohkuma. Multilingual grammar development via grammar porting. InESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development 2003 (pp. 49-56).

[18] A. Novello, and C. Callaway. "Porting to an Italian surface realizer: A case study." *Proc. of the 9th European Workshop on NLG*. 2003.

[19] P. Laurette, and S. Bosch. "Exploiting cross-linguistic similarities in Zulu and Xhosa computational morphology." Association for Computational Linguistics, 2009.

[20] H. Curry. Some logical aspects of grammatical structure. Structure of language and its mathematical aspects.pp 56-68, 1961.

[21] P. Paikens and N. Gruzitis. An implementation of a Latvian resource grammar in Grammatical Framework. InLREC May 2012 ,pp. 1680-1685.

[22] A. Ranta, Grammatical Framework: Programming with multilingual grammars. Stanford: CSLI Publications, Center for the Study of Language and Information; July 2011.

[23] A. Ranta, A. El Dada , J. Khegai . The GF resource grammar library. Linguistic Issues in Language Technology. Dec 2009, pp 1-63.

[24] G. Wagner . The Bantu of Western Kenya: with special reference to the Vugusu and Logoli. Oxford University Press, London. 1970.

[25] B.G. McIntosh, "The Eastern Bantu Peoples," in B.A. Ogot and J.A. Kieran, eds. Zamani, Nairobi, 1968, pp 203-.205.

[26] M. Guthrie. Gender, number and person in Bantu languages. Bulletin of the School of Oriental and African Studies. Oct 1948, pp. 847-56.

[27] L. Kaviti, A minimalist perspective of the principles and parameters in Kikamba morpho-syntax. Doctoral Dissertation, University of Nairobi., 2004.

[28] K. Mbuvi. The syntax of Kikamba noun modification. Unpublished Masters Dissertation, University of Nairobi, 2005.

[29] N.O. Basweti, N.O, A morphosyntactic analysis of agreement in Ekegusii in the minimalist program, Masters Dissertation. Nairobi university. Kenya, 2005.

[30] D.O. Ongarora. Bantu morphosyntax: A study of Ekegusii. Doctoral Dissertation, Iawaharlal Nehru University, 2008.

[31] K.N. Osinde, K. N. Ekegusii morphophonology: an analysis of the major consonantal process. Masters Dissertation. University of Nairobi, 1998

[32] L.M. Hyman. Aghem grammatical structure with special reference to noun classes, tense-aspect and focus marking. University of Southern California, 1979.

[33] Kihm A. What's in a noun: Noun classes, gender, and nounness. Ms. Université Paris. Feb 2002.

[34] F. Di Garbo. Gender and its interaction with number and evaluative morphology: An intra-and intergenealogical typological survey of Africa (Doctoral dissertation, Department of Linguistics, Stockholm University). 2014.

[35] F. Katamba, Bantu nominal morphology. In Derek Nurse & G´erard Philippson, 2003.(eds.), The Bantu languages, 103–120. London: Routledge.

[36] B. Kituku, W. Nganga, L. Muchemi L. Grammar Engineering for the Ekegusii Language in Grammatical Framework. European Journal of Engineering and Technology Research. 2021 Mar 18;6(3):20-9.

[37] Carr M, Verner J. Prototyping and software development approaches. Department of Information Systems, City University of Hong Kong, Hong Kong. 1997,pp 319-38.

[38] J. Camilleri, A computational grammar and lexicon for Maltese. Master’s thesis, Chalmers University of Technology. Gothenburg, Sweden, 2013.