Proceedings of the Second Workshop on African Language Technology
- AfLaT 2010 -

Developing an Open source Spell-checker for Gĩkũyũ

Kamau Chege[1], **Peter Wagacha**[1], **Guy De Pauw**[1,2], **Lawrence Muchemi**[1]
Wanjiku Ng'ang'a[1], **Kenneth Ngure**[3], **Jayne Mutiga**[3]

[1]School of Computing and Informatics
University of Nairobi
Kenya
kamauchege@gmail.com
{waiganjo,lmuchemi,wanjiku.nganga}@uonbi.ac.ke

[2]CLiPS - CLG
University of Antwerp
Belgium
guy.depauw@ua.ac.be

[3]Department of Linguistics
University of Nairobi
Kenya
kamurink@yahoo.com
jaynemutiga@yahoo.co.uk

Abstract

In this paper, we describe the development of an open source spell checker for Gĩkũyũ, using the Hunspell language tools. We explore the morphology of Gĩkũyũ and highlight the inflection of various parts of speech, including verbs, nouns, and adjectives. In Hunspell, surface words are realized as a set of continuation classes, with each class providing a morpheme with a specific function. In addition, circumfixation, which is prevalent in Gĩkũyũ, is implemented. Hunspell also provides for word suggestion, using character prevalence and replacement rules. Given that the developed Gĩkũyũ spellchecker and the Hunspell tools are open source, the spell checking function developed in this work can be adopted in major open-source products such as Mozilla and OpenOffice products. The spell checker has a fairly representative Gĩkũyũ lexicon and achieves an acceptable realization of a Gĩkũyũ spellchecker. When tested on a test corpus, the spell checker attains a precision of 82%, recall of 84% and an accuracy of 75%.

1. Introduction

A grammatically and orthographically correct text is necessary in ensuring high quality textual documents for effective communication. There is therefore a need to avail tools and utilities to support electronic document preparation. A spell checker is a design feature or a program that verifies the spelling of words in a document, query, and browsers, among other contexts. The goals of developing human language technology applications can only be achieved if localization, and basic language tools or utilities like spell checkers are made publicly available for a language. This paper describes such an effort for the Kenyan language of Gĩkũyũ.

Gĩkũyũ can be classified as a resource scarce language with respect to language technology resources, tools and applications. This situation can be attributed to different factors. First, Kiswahili and English are the dominant languages in Kenya, meaning that most of all textual communication is in these languages. Second, Gĩkũyũ orthography contains two diacritically marked characters (ĩ and ũ) that require extra keystrokes to generate, a situation which often makes users opt for the diacritically unmarked equivalents, resulting in non-standard Gĩkũyũ texts. These extra characters also pose a challenge for automated corpus collection methods, such as those using optical character recognition (OCR).

However, despite such an unfavorable backdrop, Gĩkũyũ, together with a few other Kenyan languages, are steadily becoming commercial languages as evidenced by their increased use in broadcast media, publishing and advertising. These developments point to a need for language technology support for such languages to boost their use, growth and viability. The work described here makes a positive contribution to this need, by providing a word processing utility that will encourage and enhance creation of correctly spelled Gĩkũyũ texts.

1.1. Gĩkũyũ Language Technology

There exists a number of language technology research efforts on Gĩkũyũ. These include a grapheme-based approach to diacritic restoration (Wagacha et al., 2006b; De Pauw et al., 2007), morphological analysis using a maximum entropy approach (De Pauw and Wagacha, 2007) and finite-state techniques (Chege, 2009), machine translation (Chege, 2009) and speech synthesis (Maina, 2009).
The closest effort towards Gĩkũyũ spell checking is a dictionary-based system, incorporated in a Gĩkũyũ text editor (Chege, 2007). This system works well, but for only a limited number of words, as contained in the dictionary. The work described herein overcomes this limitation by using a rule-based approach for determining the correct spelling of any Gĩkũyũ word.

2. Gĩkũyũ

Gĩkũyũ is a Bantu language belonging to the Kamba-Kikuyu subgroup of the Niger-Congo family, with over seven million speakers living in Central Kenya. The language has six dialects and is lexically similar to closely related languages such as Gĩchuka, Kĩembu, Kĩmerũ, and Kĩkamba.
Gĩkũyũ is highly inflectional and is characterized by a complex word structure and phonemics. It is also highly agglutinative, with words being formed from a battery of prefixes and suffixes. Like many Bantu languages, Gĩkũyũ has fifteen noun classes and two additional locative classes. It also has a concord system formed around the noun classes. Gĩkũyũ is a tonal language, a feature that introduces ambiguity at different grammatical levels, although tonality is not represented in the standard orthography.

2.1. Noun Morphology

Gĩkũyũ nouns can be grouped into two categories, namely, derived and underived nouns. Underived nouns consist of named entities, while derived nouns can be formed in one of

two ways: by affixing diminutive, augmentative or collective prefixes to an underived noun, or through verbal nominalization. The following examples illustrate these processes:

nyũmba ⇒ kĩ-nyũmba	(a big house)
imondo ⇒ tũ-mondo	(many nice handbags)
thaaka ⇒ mũ-thaak-i	(player)
hooya ⇒ i-ho-ero	(Place of prayer)
getha ⇒ i-geth-a	(harvesting occasion)
thooma ⇒ ga-thom-i	(the small one who reads)

Membership to a noun class is determined by a concord system with agreement enforced on other sentence components, such as adjectives and verbs. All Gĩkũyũ nouns, derived or underived, can also be optionally affixed with the locative suffix *-inĩ*, which changes the meaning from a referential entity to a location, as shown in the following examples:

metha-inĩ	(on the table)
mũtĩ- inĩ	(on the tree)

2.2. Verb Morphology

A typical Gĩkũyũ verb consists of a combination of zero or more dependent morphemes, a mandatory dependent morpheme and a mandatory final vowel. The simplest verb consists of a verb root and a final vowel. These are usually commands or directives. Subjunctive verb formations, i.e. commands, can optionally take a plural marker *-i* or *–ni*. Examples include:

ma-thaak-e	(so that they play)
in-a-i	(sing)
nĩ-ci-mũ-hat-agĩr-a	(they usually sweep for him/her)
reh-e-ni	(bring)

Gĩkũyũ verbs can also undergo full or partial reduplication, depending on the number of syllables in a word. Words with two syllables or less undergo full reduplication, while words with more than two syllables undergo partial reduplication, with only the first two syllables being reduplicated. Examples are:

negena ⇒ nega-negena	(make noise a little more)
tiga ⇒ tiga-tiga	(leave a little more)

Gĩkũyũ verbs are also affected by consonantal and vowel phonemics. Meinhof's Law involves consonants *b, c, r, t, g, k* being replaced with NC composites in verbs obeying first person singular, noun classes 1, 8 and 9 concord systems. Dahl's Law is a consonantal mutation that involves the cause sound *k* appearing before trigger voiceless sounds *c,k,t*, being replaced with its equivalent voiced sound *g*. Examples include:

rathima ⇒ ndathima
uma ⇒ nyumia
kũ-theka ⇒ gũtheka
kĩ-ka-thira ⇒ gĩgathira

Vowel mutation includes vowel lengthening before prenasalized stops and vowel assimilation when some vowel combinations appear in the neighborhood of each other (Mugane, 1997).

3. Methodology

Developing a spell checker requires a method of determining the set of valid words in a given language, against which the words to be checked are compared. In addition, if a word is not valid, a list of possible suggestions or alternatives need to be returned, in some prioritized order. In this work, we use the Hunspell tools (Németh, 2010), which facilitate the definition of the valid words in a language, as well as the likely suggestions.

We relied on a Gĩkũyũ corpus to obtain valid words, as well as to determine the typical misspellings that could occur in this language. The following subsections describe the Gĩkũyũ corpus and the pre-processing that was done prior to building the spell checker. A description of how to customize the Hunspell environment for Gĩkũyũ is also given. The latter subsections discuss the actual definition of the spell checker, categorized into nouns and verbs.

3.1. Corpus Collection

The primary development corpus is from a collection of a set of 19,000 words from previous works on Gĩkũyũ at the School of Computing and Informatics, University of Nairobi (Wagacha et al., 2006a; Wagacha et al., 2006b; De Pauw et al., 2007; De Pauw and Wagacha, 2007). This corpus has a bias on religious material but also includes poems, short stories, novels and Internet sources. The corpus was pre-processed manually to eliminate non-Gĩkũyũ words, and to correct diacritics, where necessary. The corpus was then manually annotated where words were categorized into corresponding parts of speech, in line with Hunspell's defined continuation classes. Perl scripts were used for generic annotation and marking.

The test corpus was acquired from two sources: a popular Gĩkũyũ blog "Maitũ nĩ ma itũ (Our Mother is our truth)" was chosen as it contains diacritically-marked texts on a variety of contemporary topics and Ngũgĩ wa Thiong'o's novel "Mũrogi wa Kagogo", which is not diacritically marked and represents how a normal user would type on a standard keyboard.

3.2. Hunspell Language Tools

Hunspell is a set of open source tools and utilities used for development and testing of spellcheckers and morphological analyzers. The main goal of Hunspell and its predecessors is to compress the lexicon of a language into a manageable size. Hunspell is an enhancement of its predecessors *Ispell* and *MySpell*, with the latter being the official spellchecker for OpenOffice and Mozilla products.

Hunspell was built to support languages with rich morphology, including complex prefixes and compounding. In addition, Hunspell supports circumfixation where a certain suffix can only co-occur with a given prefix or set of prefixes. Hunspell makes use of two files: the affix file which defines the morphological rules of the language, and the dictionary file which contains the actual word stems. Applicable affix rules must be specified for each stem.

3.3. Defining the Gĩkũyũ Spell checker

The spellchecker is implemented using the concept of continuation classes, where a word is represented as a compo-

Foc + Subj + Neg + Cond + Tense + Obj + Redup + Verb + DvbExt + Asp + FVwl

Figure 1: Verbal Affixation in Gĩkũyũ.

sition of one or more morphemes.

3.3.1. Hunspell Language Setup for Gĩkũyũ

To handle Gĩkũyũ diacritics, it is important to set character support to Unicode (UTF-8). In addition, since Gĩkũyũ verbs generate many affix rules, Flag is set to a number so as to handle the numerous affix rules. The Gĩkũyũ alphabet includes the apostrophe and hyphen, as in *ng'ombe* and *iria-inĩ*, and the orthography set is therefore extended with these characters. This is important as it helps Hunspell determine word stops. Since Gĩkũyũ has more than one level of prefixes and suffixes, support for complex prefixes, as well as circumfixation, has to be enabled in Hunspell.

3.3.2. Suggestions Component

The suggestions component is used to generate probable suggestions for a misspelled word. It is implemented in the affix file. Hunspell uses two sections in the affix file when generating suggestions for misspelled words. The first is the TRY command. This lists the language's orthography set in order of frequency. A more frequently used character has more weight during suggestions. The TRY command is shown below:

TRY eiĩanrtocduũgmhbykw'jNRTCGDMHBEAUŨYOIĨKWJ

The second command used in the suggestion component is the REPLACE command. The command lists the most commonly misspelled n-grams and their replacements. The major n-grams are a result of influence from different dialects, foreign languages and also by differences in spoken versus written Gĩkũyũ. Examples of Gĩkũyũ replace suggestions include:

REP 35
REP s c
REP sh c
REP sh ch
REP c ch
REP f b
REP v b
REP l r
REP i ĩ

3.3.3. Noun Component

The noun component is implemented in two parts, namely the underived nouns and the derived nouns, taking into account that both noun types can take the optional locative suffix.

SFX 251 Y 1
SFX 251 –inĩ .

Underived nouns have a continuation class leading to the common locative class. The class consists of optional diminutive, augmentative and collective prefixes. Class selectors are influenced by prenasalization and Dahl's Law.

PFX 201 Y 25
PFX 201 mb kab mb
PFX 201 mb tũb mb
PFX 201 ng rũg ng
PFX 201 ng' tũg ng'
PFX 202 Y 16
PFX 202 0 tũ [abmngrw][^bdngj]
PFX 202 i ka i[^cktĩ]

 # Inside the Dictionary File
mbiira/201,251
icungwa/202,251

Derived nouns are formed through circumfixation. The CIRCUMFIX and NEEDAFFIX are used to enforce circumfixation. Examples of circumfixation include:

PFX 211 Y 20
PFX 211 0 mũ/002 .
PFX 211 0 tũ/002 .
PFX 211 0 kĩ/002 [^ckt]
PFX 211 0 gĩ/002 [ckt]
SFX 261 Y 1
SFX 261 0 i/211,002

#Inside the Dictionary File
thaaka/261
ruga/261

Example generated words are: *mũrugi, kĩrugi, karugi*

3.3.4. Verb Component

The continuation classes for verbs cater for the focus marker, concord subject, object classes, conditional, negation, tense, deverbal extensions, aspectual markers and final vowels. Since Hunspell only supports a maximum of three prefixes and three suffixes, it was impossible to directly implement the continuation classes as described, since verbs can have up to seven prefixes and four suffixes, as illustrated in Figure 1.

To handle this challenge, we combined all prefixes into one complex prefix. While it is possible to identify other prefix combinations that can simplify implementation, such combinations are subject to other problems. For instance, following from Dahl's Law, it is necessary to determine where the trigger (c, k, t) and the cause (k) appear together.

Similar problems are evident with prenasalization and Meinhof's Law. The singular complex prefix approach that was adopted for this study is shown in the following snapshot of the verb component definition. Given that the simplest verb form consists of a final vowel appended to the verb stem, the only mandatory continuation class in the verb component is therefore the final vowel continuation class.

PFX 611 Y 27705
PFX 611 0 ndĩha [^eĩoũ]
PFX 611 e ndĩhe e
PFX 611 ĩ ndĩhe ĩ
PFX 611 0 matingĩrama [^eĩoũ]
PFX 611 e matingĩrame e
PFX 611 ĩ matingĩrame ĩ

 # Negation
PFX 711 Y 1495
PFX 711 0 nda [^aeĩoũ]
PFX 711 a nda a
PFX 711 u gũtigu u[ckt]
PFX 711 e hatikwe e[^ckt]

#Aspectual/Modal Suffixes, Voiced Consonants
SFX 482 Y 3
SFX 482 a ete [^i]a
SFX 482 ia etie ia
SFX 482 wo etwo wo

#Deverbal Extensions, All Verbs
SFX 450 Y 30
SFX 450 a anga [^i]a
SFX 450 a angwo [^i]a

 # Deverbal Extensions, Verbs beginning
 # with Voiced Consonants
SFX 451 Y 12
SFX 451 a ĩka [^i]a
SFX 451 ia ĩka ia
SFX 451 a ĩra [^i]a
SFX 451 a ĩrwo [^i]

 # Deverbal Extensions, Verbs beginning
 # with Voiceless Consonants
SFX 452 Y 12
SFX 452 a eka [^i]a
SFX 452 ia eka ia
SFX 452 a era [^i]a
SFX 452 a erwo [^i]a

 # In the Dictionary File
arama/611,711,612,712,613,713,614,714,450,451,480,481
etha/611,711,612,712,613,713,614,714,450,452,480,482

Verbs in the dictionary file are categorized on the basis of applicable continuation classes. As seen in the examples above, these categories (groupings) are also influenced by whether the verb ends in a mid-low or mid-high voice, and whether it can take a causative extension, among others.

4. Results and Discussion

The developed Gĩkũyũ spellchecker and "suggester" engine was incorporated into OpenOffice Writer[1] and evaluated using the test corpus.

[1]For OpenOffice 2.x and below, integrating the spell checker requires copying files to appropriate locations and editing the dictionary.lst file accordingly, while for OpenOffice 3.x, dictionaries are added as extensions.

The results obtained are shown in Table 1 and are based on typical evaluation measures. In this study, True Positives (TP) represent those correctly spelled words that are recognized as such by the spell checker. False Positives (FP) represent misspelled words that are not flagged as such. True Negatives (TN) represent misspelled words that are flagged as such, while False Negatives (FN) represent correctly spelled Gĩkũyũ words that are flagged as misspelled.

Results	TP	FP	TN	FN	TOTAL
No. of Instances	3351	756	854	618	5579
Precision	$TP/(TP+FP) = 0.82$				
Recall	$TP/(TP+FN) = 0.84$				
Accuracy	$(TP+TN)/Total = 0.75$				

Table 1: Evaluation Results for Test Corpus

On analysing the results, it was noted that the major reason for unrecognized Gĩkũyũ words (False Negatives) is word stem missing in the dictionary file, as well as proper names, especially of people and places. Having a richer corpus from which to draw stems in addition to the inclusion of a named entity recognition heuristic would be one strategy to reduce the false negatives, thereby improving spell checking accuracy.

It was observed that, when spell checking diacritically marked texts, many misspellings (True Negatives) are generated. Subsequently, these words were easy to correct using the suggestions generated. However, the suggestion component degraded when the misspelling was a combination of a diacritic and other (one or more) characters.

An issue that is not evident in the results, but which was a major challenge, is over-generation, where the uncontrolled combination of prefixes and suffixes especially on verb morphology, generated a large number of non-Gĩkũyũ words due to semantic/meaning violations.

5. Conclusion

In this work, we have reviewed the development of an open-source spellchecker for Gĩkũyũ language using the Hunspell language tools. Results obtained in applying the developed spellchecker in OpenOffice Writer, have shown an acceptable performance with an accuracy of 75%, a precision of 82% and a recall of 84%.

The developed spell checker clearly has practical value in the spell checking of Gĩkũyũ texts. The developed tool can also provide utility in the collation and correction of additional Gĩkũyũ texts during corpus compilation. The methodology employed in this work can be easily ported to other Bantu languages that share a large percentage of similarity stems, as one approach in bootstrapping the development of spell checkers for these languages.

Availability and Acknowledgments

A beta version of the spellchecker is available from http://extensions.services.openoffice.org/en/project/Gikuyu.

The research presented in this paper was made possible through a generous grant by the Acacia Program in IDRC,

Canada, through the African Localization Network - ANLoc.

6. References

ANLoc. (2010). *The African Network for Localization*. [Online]. Available: http://www.africanlocalisation.net (accessed March 2010).

Chege, K. (2007). *Language-Independent Spellchecker: Gĩkũyũ Word Processor*. Nairobi, Kenya: School of Computing and Informatics, University of Nairobi. Unpublished Paper.

Chege, K. (2009). *Morphological Analysis of Gĩkũyũ: Towards Machine Translation*. Nairobi, Kenya: School of Computing and Informatics, University of Nairobi. Unpublished Paper.

De Pauw, G. & Wagacha, P.W. (2007). Bootstrapping morphological analysis of Gĩkũyũ using unsupervised maximum entropy learning. In H. Van hamme & R. van Son (Eds.), *Proceedings of the Eighth Annual Conference of the International Speech Communication Association*. Antwerp, Belgium.

De Pauw, G., Wagacha, P.W. & de Schryver, G-M. (2007). Automatic diacritic restoration for resource-scarce languages. In Václav Matoušek & Pavel Mautner (Eds.), *Proceedings of Text, Speech and Dialogue, Tenth International Conference*. Heidelberg, Germany: Springer Berlin / Heidelberg, pp. 170–179.

Maina, L.W. (2009). *Text-to-Speech System for Gĩkũyũ with Interactive Voice Response System*. Nairobi, Kenya: School of Computing and Informatics, University of Nairobi. Unpublished Paper.

Mugane, R. (1997). *A Paradigmatic Grammar of Gĩkũyũ*. Stanford, USA: CLSI Publications.

Németh, L. (2010). *Hunspell*. [Online]. Available: http://hunspell.sourceforge.net (accessed: March 2010).

Wa Mbugwa, G. (2010). *Maitu ni ma itu (Our Mother is our truth)*. [Online]. Available: http://gigikuyu.blogspot.com (accessed March 2010).

wa Thiong'o, N. (2007). *Mũrogi wa Kagogo*. Nairobi, Kenya: East African Educational Publishers Ltd.

Wagacha, P.W., De Pauw, G. & Getao, K. (2006)a. Development of a corpus for Gĩkũyũ using machine learning techniques. In J.C. Roux (Ed.), *Proceedings of LREC workshop - Networking the development of language resources for African languages*. Genoa, Italy: European Language Resources Association, pp. 27–30.

Wagacha, P.W., De Pauw, G. & Githinji, P.W. (2006)b. A grapheme-based approach for accent restoration in Gĩkũyũ. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*. Genoa, Italy: European Language Resources Association, pp. 1937–1940.