

Setswana Verb Analyzer and Generator

Gabofetswe Malema

*Department of Computer Science
University of Botswana
Gaborone, Botswana*

malemag@mopipi.ub.bw

Nkwebi Motlogelwa

*Department of Computer Science
University of Botswana
Gaborone, Botswana*

motlogel@mopipi.ub.bw

Boago Okgetheng

*Department of Computer Science
University of Botswana
Gaborone, Botswana*

okgethengb@gmail.com

Opelo Mogotlhwane

*Department of Computer Science
University of Botswana
Gaborone, Botswana*

mogoom@mopipi.ub.bw

Abstract

Morphological analysis is one of the first steps in natural language studies. It is a basic component in a number of natural language processing systems. There are a few attempts made with regard to the development of Setswana morphology analyzer and generator. However, these attempts are not fully developed to produce a potential multipurpose Setswana morphological analyzer and generator. This paper presents a rule-based Setswana verb morphological analysis and generation. Morphological rules are supported by a dictionary of root words. Results show that Setswana verbs could mostly be analyzed using morphological rules and the rules could also be used to generate words. The analyzer gives 87% performance rate. The rules fail when multiple words have the same intermediate word and homographs. The generator shows that Setswana verbs are very productive with an average of 89 words per root word. However, ambiguity in word generation rules leads to formation of words that are meaningless or are not used.

Keywords: Setswana, Setswana Verb Morphology, Morphological Analyzer and Generator.

1. INTRODUCTION

Setswana is an official and main language spoken in Botswana. It is also spoken in neighboring countries such as South Africa and Zimbabwe. Like many African languages not much has been developed in terms of Setswana language analytical tools. To have the explosion of natural language applications like those developed for English; basic Setswana analytical tools have to be developed. Basic tools include spell checkers, tokenization, part of speech taggers and morphological analyzers. These tools are pre-processing phases of larger systems such as machine translation information retrieval and extraction and grammar checkers [1].

This paper investigates the development of a rule-based Setswana verb morphological analyzer and generator. Morphology is the study of word formation in a language. There are different approaches to morphological analysis, the most prominent been statistical and rule-based approaches. Statistical approaches require test data to learn words formations in a language.

They are language independent and less complex compared to rule based approaches. However, statistical approaches rely heavily on available data. In scarcely resources languages such as Setswana, this approach will probably not have good results. Rule-based approaches follow morphological language rules. These rules are implemented as a program to transform the words. Unlike statistical algorithms, rule-based algorithms heavily depend on language knowledge. Setswana language morphology has been studied in a number works including [2] and [3]. We use the established rules or patterns to implement the proposed morphological analyzer and generator.

A few research works have been done in the development of a Setswana morphological analyzer and generator. K. Brits et al developed a prototype for automatic lemmatization of Setswana words in [4]. The rule based prototype used finite state automation of rules. There results were good with a performance of 94% for verbs and 93% for nouns. Similar works have been done on Setswana lemmatization in [5][6]. However, we have not seen any developments towards a fully developed and general purpose Setswana morphological analyzer and generator.

In this paper a rule-based Setswana verb analyzer and generator is presented. In this study we present the different word transformations by category and their challenges when implemented. We show why in some cases the rules fail and possible ways of minimizing such errors.

This paper is organized as follows. Section 2 presents Setswana Verb morphology by category. In Section 3 a proposed analyzer and generator architecture is described. Section 4 presents the results obtained by implementing the morphological rules in Section 2 and Section 5 concludes the paper.

2. SETSWANA VERB MORPHOLOGY

Setswana language is an agglutinative language and Setswana words can be generated from root words by adding appropriate suffixes and prefixes. A verb can be used to generate many words using derivational and inflectional morphemes. The affixes change or extend the meaning of the word[2][3][7].

In Setswana verbs prefixes and suffixes provide essential information regarding type, tense and mood. For example the verb *bu* (*speak*) could be changed in meaning by using different suffixes as below:

bu (*speak*)

buisa (*speak to*)

buisiwa (*spoken to*)

buile (*spoken*)

buisana (*speak to each other*)

Below we look at the application of prefixes and suffixes in different word categories. Although the application of prefixes and suffixes is regular for the most part there are cases where they do not give a valid word. Setswana verbs fall in different word categories which include the passive (*tirwa*), causative(*tirisa*), reflexive (*itira*), reversal (*tirolola*), applicative(*tiredi*), reciprocal(*tirana*), neuter-passive(*tiregi*), perfect tense (*paka-pheti*), extensive(*tiraka*) mood and plural.

The Passive (*tirwa*): indicated by suffix –w-

Passive verbs imply that some action is performed on the object. They are created by attaching the suffix –w- to a verb. For example:

supa >> *supiwa*(*point/to be pointed at*)

loga >> *logiwa* (*braid/to be braided*)

bopa >> *bopiwa* (*mold/to be molded*)

The reverse transformation therefore will remove *-iw-* to get the base form of the word. There are several suffixes that are used to show passivity. Below are some of the suffixes and their contracted forms.

ngwa(miwa) : loma >> longwa/lomiwa (bite/to be biten)
jwa (biwa) : leba >> lebiwa/lejwa(look/ to be looked at)
gwa (giwa) : tshega >> tshegiwa/tshegwa(laugh/to be laughed at)
nngwa (nyiwa) : senya >> senyisa/Senngwa(destroy/destroyed)
tlhwa (tliwa) : latlha >> latlhiwa/latlhwa(throw/ to be thrown/left)
lwa : lelela >> lelelwa (cry for/cried at)
swa (siwa) : lesa >> lesiwa/leswa(leave/left by)
tswa(diwa) : robala >> robadiwa/robatswa (sleep/made to sleep)
twa(tiwa) : ruta >> rutiwa/rutwa (teach/taught)

The given suffixes indicate passivity for the most part. However, there are some verbs that have the passivity suffix but are not passive verbs. Examples are *ungwa, wa, swa, nwa, lwa*. In the proposed analyzer these verbs are not a problem because they are included in the dictionary as root words.

Causative/Intensity (tirisa/tirisisa): indicated by suffixes *-is-* / *-isis-*

Causative and intensity verbs imply the object is caused or helped to do something. They are created by attaching the suffix *-is-* or *-isis-* for emphasis to the root verb. For example
supa >> supisa(point/make to point)
loga >> logisa (braid/make or help to braid)

The reverse transformation removes *-is-* to get the base form of the word. However, there are exceptions, which use the *-is-* suffix but do not mean causativity. Examples are *tataisa, itisa*. The exceptions are also not a problem in the proposed analyzer as they are part of the dictionary.

The applicative (tiredi): indicted by suffix *-el-*

The applicative verbs imply some task is performed on behalf of the object. They are created by attaching the suffix *-el-* to the root verb. Examples are
supa >> supela(point/point for)
loga >> logela (braid/braid for)

The reverse transformation removes *-el-*. Exceptions include *bela, sela, tlhatlhela*.

Reciprocal (tirana): indicated by suffix *-an-*

Reciprocal verbs imply cooperation between subjects or they are performing a task on each or for each other. They are created using the *-an-* suffix. Examples are:
supa >> supana(point/point each other)
loga >> logana (braid/ braid each other)

Exceptions include *pana* and *gana*.

The Neuter-Passive (tiregi): indicated by suffixes *-eg-*, *-al-*, *-agal-*, *-eseg-*.

Neuter-passive verbs imply something is doable. Example are
supa >> supega (point/pointable)
loga >> logega (braid/braidable)

There are also exceptions. Some verbs have these suffixes on their root form. Examples are *sega* and *bega*.

The Reversal (tirolola): indicated by suffixes *-ol-*, *-og-*, *-olog-*.

Reversal verbs imply the task is being reversed. Examples are
bofa >> bofolola (tie/untie)

soka >> *sokolola* (turn/unturn)

Extensive (tiraka): indicted by suffix –ak-

Extensive verbs imply the action is performed often, a lot, with energy or excessively. Examples are

roga >> *rogaka* (insult/insult excessively)

rutha >> *ruthaka* (hit/hit excessively)

Reflexive(itira): indicted by prefixes i-, m-, n-

Reflexive verbs imply the subject is performing a task on itself or for itself. There are different transformations when a verb is converted to reflexive depending on the starting alphabet of the verb.

Verbs starting with [a,e,i,o,u,w]

Verbs starting with these vowels introduce –k-. Example are

apaya >> *ikapaya* (cook/cook oneself)

emisa >> *ikemisa* (make to stop/stop oneself)

The reverse transformation therefore removes *ik-* to get the base form of the word. However, verbs starting with *k-* just insert the reflexive prefix *i-* without any further transformation. For example

kuka >> *ikuka* (pick/pick oneself up)

kwala >> *ikwala* (write/write oneself)

Now how do we differentiate words which start with *k-* in the base form and those that start with a vowel? There is no way of knowing if the root word starts with *k-* or with a vowel. The proposed analyzer tries both alternatives and hopes that one and only one of them produces a valid root word. Unfortunately, in some cases both cases result in valid root words. This is one of the limitations of morphological analysis rules.

Verbs starting with b-

Verbs starting with *b-* introduce –p- when converted to reflexive verbs. Examples are

botsa >> *ipotsa* (ask/ ask oneself)

bitsa >> *ipitsa* (call/call oneself)

The reverse transformation removes *b-* and replaces it with *p-*. However, verbs starting with *p-* just insert reflexive prefix *i-* without any further transformation. For example

pana >> *ipana*

penta >> *ipenta* (paint/paint oneself)

patisa >> *ipatisa* (squeeze/squeeze oneself)

Now how do we differentiate words that start with *p-* in the base form and those that start with a vowel? The proposed analyzer tries both alternatives and hope that only one produces a valid word.

Verbs starting with d- and l-

Verbs starting with *l-* and *d-* introduce *t-* when converted to reflexive verbs. For example

letsa >> *itetsa* (make to cry/make oneself cry)

dia >> *itia* (delay/delay oneself)

The reverse transformation removes *l-* or *d-* and replaces it with *t-*. However, verbs starting with *t-* just insert reflexive suffix *i-* without any further transformation. For example

tena >> *itena* (make angry/anger oneself)

tiisa >> *itiisa* (make stronger/make oneself stronger)

Now how do we differentiate words that start with *t-* in the base form and those that start with a *l-* or *d-*? The proposed analyzer tries both alternatives and hopes that only one produces a valid word.

Verbs starting with f- and h-

Verbs starting with *f-* and *h-* introduce *ph-* when converted to reflexive verbs. For example

fenya >> *iphenya* (defeat/defeat oneself)

hisa >> *iphisa* (burn/burn oneself)

The reverse transformation removes *ph-* and replaces it with *f-* or *h-*. However, verbs starting with *ph-* just insert reflexive suffix *i-* without any further transformation. For example

phimola >> *iphimola* (rub off/rub off oneself)

phutlha >> *iphutlha*

Now how do we differentiate words that start with *ph-* in the base form and those that start with an *f-* or *h-*? Not all verbs starting with *h-* follow *ph-* as in verbs starting with *f-*. Verbs starting with *h-* are transformed to *ph-* if there is an alternative verb starting with *f-*. For example,

hisa and *fisa* (same meaning)

hata and *fata* (same meaning)

Those that do not have an equivalent in *f-*, transform to *kh-*. For example,

huma >> *ikhumisa* (become rich/make oneself rich)

hemisa >> *ikhemisa* (breathe by yourself)

hupa >> *ikhupisa*

The reverse transformation removes *-kh-* and replaces it with *h-*. However, there are verbs starting with *kh-* that insert reflexive suffix without any further transformation. For example

khurula >> *ikhurumela* (cover/cover oneself)

In this paper, we transform verbs with *-ph-* to those starting with *f-* even if there is an equivalent of *h-*. There are few exceptions that start with *h-* and have equivalent words starting with *g-*. For example, *hagkamatsa* and *gakgamatsa* mean the same thing. The transformation is handled under words starting with *g-*. Words starting with *-kh-* are the once transformed to *h-*.

Verbs starting with g-

Verbs starting with *g-* introduce *kg-* when converted to reflexive verbs. For example

gana >> *ikgana*

gelela >> *ikgelela*

golola >> *ikgolola*

The reverse transformation removes *-kg-* and replaces it with *g-*. However, verbs starting with *kg-* just insert reflexive prefix *i-* without any further transformation. For example

kgalema >> *ikgalema*

kgebetla >> *ikgebetla*

kgisa >> *ikgisa*

kgobokanya >> *ikgobokanya* (collect/collect oneself)

Now how do we differentiate words that start with *kg-* in the base form and those that start with a *g-*? The proposed analyzer tries both alternatives and hopes that only one produces a valid word.

Verbs starting with r-

Verbs starting with *r-* introduce *th-* when converted to reflexive verbs. For example

rata >> *ithata* (love/love oneself)

reka >> *itheka* (buy/buy oneself)

The reverse transformation removes *th-* and replaces it with *r-*. However, verbs starting with *th-* just insert reflexive prefix *i-* without any further transformation. For example
thalosa >> *ithalosa*(*explain/explain oneself*)
tswala >> *itswala*(*close/close oneself*)

Now how do we differentiate words that start with *th-* in the base form and those that start with an *r-*? The proposed analyzer tries both alternatives and hopes that only one produces a valid word.

Verbs starting with s-

Verbs starting with *s-* introduce *tsh-* when converted to reflexive verbs. For example
senya >> *itshenya*(*destroy/destroy oneself*)
sila >> *itshila*(*grind/grind oneself*)

The reverse transformation removes *tsh-* and replaces it with *s-*. However, verbs starting with *tsh-* just insert reflexive prefix *i-* without any further transformation. For example
tshoga >> *itshoga*(*laugh/laugh at oneself*)
tshosa >> *itshosa*(*scare/scare oneself*)

Now how do we differentiate words that start with *tsh-* in the base form and those that start with an *s-*? The proposed analyzer tries both alternatives and hopes that only one produces a valid word.

Verbs starting with other alphabets other than the alphabets above and other combinations just introduce an *i-* when converted to reflexive verbs. For example
tlatsa >> *itlatsa*(*fill up/fill up oneself*)
mona >> *imona*(*lick/lick oneself*)

The *n-* prefix behaves in similar way to the reflexive prefix *i-* but does not imply reflexivity. It implies some task is performed on or for the subject by someone or something. For verbs starting with *b-*, *m-* is used instead of *n-*. Examples are
bona >> *mpona*(*see/see me*)
tena >> *ntena*(*make angry/make me angry*)
tshaba >> *ntshaba*(*fear/fear me*)

The rest of the verbs are transformed by adding the reflexive suffixes *i-* and *n-*. For example:
tlhapisa >> *itlhapisa* | *ntlhapisa*
tsamaisa >> *itsamaisa* | *ntsamaisa*

Plural Verbs

Setswana verbs could also indicated that more than one person is performing a task. Plurality is achieved by adding a suffix *-ng* to the verb. For example
rapela >> *rapelang*(*pray/pray you all*)
aga >> *aging*(*build/build you all*)

The root form is found by removing the suffix *-ng*. The plural suffix *-ng* is removed first.

Perfect verbs

Perfect verbs behave in a similar way as in infinitive tense. Perfect tense verbs use a number of suffixes depending on suffix of the original word. Below are some examples
-lwa >> *-etswe*: *bulwa* >> *butswa* (*being opened/already opened*)
-ya >> *-ile*: *tsamaya* >> *tsamaile* (*go/gone*)
-nya >> *-ntse*: *omanya* >> *omantse*(*scold/scolded*)

For a more complete list of perfect verbs suffixes refer to [3].

Mood verbs

Mood verbs are formed by changing the *-a* in the infinitive form of the verb to *-e*. Examples are
supa >> *supe* (point)
bala >> *bale* (read)

Combining multiple word categories

Some words are a result of a combination of multiple word-category transformations. For example reciprocal + passive + perfect tense + plural: *bona* >> *bonana* >> *bonanweng*
causative + reciprocal + passive + perfect tense + plural: *aga* >> *agisa* >> *agisana*
>> *agisanne* >> *agisannwe* >> *agisannweng*
neuter-passive + reciprocal: *bopa* >> *bopega* >> *bopagana*
reversal + passive: *somola* >> *somolola* >> *somololwa*
neuter-passive + applicative + passive: *roba* >> *robega* >> *robegela* >> *robegelwa*

There are many other combinations that are used to form Setswana words. However, it has to be noted that the different combinations do not make sense for all verbs.

Verbs to nouns

Setswana verbs could be used to form nouns and they are many of them. Below are some ways of forming nouns from verbs.

Case 1: *mo/ba* + verb (*-a* >> *-i/o*)

These nouns refer to a person who is performing the task just like *teach* >> *teacher* in English. Prefix *mo-* is attached to the verb and the *-a* in the verb changes to *-i*. When *-a* changes to *-o* then it does not refer to a person. Examples are
ruta >> *moruti* (teach/teacher, preacher)
thusa >> *mothusi* (help/helper)

For plural *ba-* is used instead of *mo-*. The nouns will be *baruti* and *bathusi* in the examples above.

Case 2: *se/di* + verb (*-a* >> *-i/o*) and *le/ma* + verb (*-a* >> *-i/o*)

These nouns refer to an object other than a person that is performing a task just like *point* >> *pointer* in English. Examples are

ipona >> *seiponi* (look at oneself/object used to look at oneself like mirror)
fetlha >> *lefetlho* (stir/stirrer)

Plural of *se-* is *di-* and of *le-* is *ma-*. Although this formation is largely not used to refer to people, there are a few exceptions. For example, *tagwa* >> *letagwa*.

Case 3: verb (*-a* >> *-o*)

These nouns refer to the act performed by the verb. They are similar to “a ruling” from rule and “a hearing” from hear in English. Examples are

supa >> *tshupo* (show/the show)
itse >> *kitso* (know/knowledge)

Plural of these nouns is uses *di-* as a prefix.

Case 4: *bo* + verb (*-a* >> *-i/o*)

These verbs refer to the act performed by the verb. Similar to case 3. Examples are
loa >> *boloi* (bewitch/witchcraft)
gola >> *bogolo* (grow/old age)

Case 5: verb >> noun + -ng

These verbs indicate location. It's a combination of the verbs in the cases above plus -ng. Examples are

moruti >> *moruting* (preacher/at the preacher)

potso >> *potsong* (question/in the question)

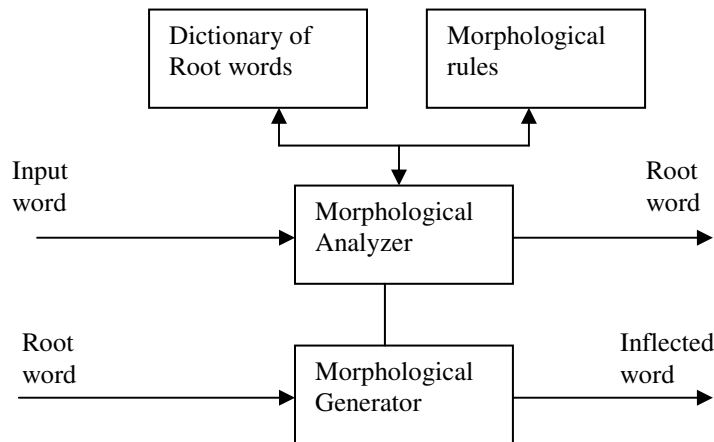


FIGURE 1: Block diagram morphological analyzer and generator.

3. EVALUATION AND PERFORMANCE

As shown in Figure 1, the proposed analyzer and generator use a database of word transformation rules. A dictionary is used to support the analyzer. The dictionary is a list of Setswana verbs in their basic form put in a hash table.

The analyzer receives a word and transforms it to its basic form. It executes the transformations discussed in the above Section in reverse. The input word is recursively transformed and checked in the dictionary until it matches a word in the dictionary or it fails. The analyzer relies on the dictionary to determine if the transformation is successful or not.

The generator on the other hand performs words transformations as in the examples given in the above Section. It transforms a given word based on the affixes of a given word and the intended word category. The rules based on the transformations discussed in the previous Section.

The proposed Setswana morphological analyzer and generator were implemented in java. The two modules are made up of functions which perform word category transformations. The main module combines several functions to reduce or generate words. In the morphological analyzer affixes are removed in sequential order a proposed in [8]. Although this sequencing does not always work, it produces the best results so far.

4. EVALUATION AND PERFORMANCE

Morphological Analyzer

The evaluation of the proposed analyzer was based on how well it maps derived forms of a word to a single basic form in the dictionary. Our dictionary contained 1500 verbs. The test data for the analyzer contained 2000 verbs mainly from [9][10]. The key thing in the test corpus is not only the number of words but also its variety. The corpus included almost all suffixes and prefixes common in Setswana. Prefixes such as *n-*, *i-*, *m-* result in different transformation when applied to verbs. We also included nouns that are derived from verbs. Such nouns include prefixes *ba-*, *bo-*, *mo-*, *ma-*, *se-* and *le*. Suffixes included covered all word categories, causative, plural, mood and others as mention in the section above.

The proposed morphological analyzer successfully reduced 1739 out of the 2000 words in the corpus giving a performance rate of 87%. Other studies have obtained similar results[4][5]. We however, could not do direct comparisons of the results because we could not the exact test data sets used in other studies. The analyzer rules may fail due to overlapping word transformations and lack of disambiguation rules in some transformations.

In the example above the verb *mpaletse* could be reduced to *balela* (*choke*) or *bala* (*read*). Given that the dictionary has *balela* as a root word it then stops. Alternatively the analyzer could check if the root word cannot be reduced further into a valid word. Although this solution may work with some words it leads to errors in others. In case the found root word is processed further and a valid word is found we still cannot determine which of the two root words the user intended.

The overlapping of word transformations can take many forms as shown below.

Case 1:

Examples: *mpaletse* >> *baletse* >> *balela* (*choke*)
>> *bala* (*read for me/read*)
>> *paletse* >> *palela* >> *pala* (*refuse?*)
itshegela >> *tshegela* >> *tshega* (*laugh*)
>> *segela* >> *sega* (*cut*)

The severity of this problem depends on the application of the analyzer. In case of a dictionary word lookup, the first example would give *balela* while the user maybe looking for *bala* which has a totally different meaning.

Case 2: In some cases the root word or even derived words have several meanings.

Example: *itshelela* >> *tshelela* >> *tshela* (*pour/live*)

In a dictionary word lookup application, the user would probably find all possible meanings of the root word. It would require context to determine the appropriate root word.

Case 3: there are cases where the analyzer cannot determine the root of the word which may lead to inappropriate transformations. That is, determining the core of the word in a given word.

Examples:

bosetlhana >> *setlhana* >> *setlha* (*scratch*)
>> *bosetlha* (*khaki*)

In the example above the prefix *bo-* is part the root word but the analyzer removes it and unfortunately the transformation ends with a valid root *setlha*. The correct transformation is the second version which removes the iterative prefix *-ana*. To overcome such cases there has to be a way to determine whether a prefix or suffix is part of the core word and therefore should not be removed in the transformation of the word.

Case 4: some prefixes can take two transformations. In some cases there is no clear determining factor on which of the two transformations is appropriate for a given word.

Examples:

bontsha >> *bona* (*see*)
>> *bonya* (*slowness*)
gantsha >> *gana* (*refuse*)
akantsha >> *akanya* (*think*)
lomagantsha >> *lomaganya* >> *lomagana* (*put together*)

In this case verbs ending with *-ntsha* could either have been derived from verbs ending with *-na* or *-nya* or both. Although most short(by length) verbs move from *-na* to *-ntsha* there are some that do not and verse versa.

Case 5: some perfect tense suffixes are the same as that of the mood tense. In such cases the analyzer has to try all alternatives. In case both alternatives produce valid words, the analyzer is unable to make a determination on the intended word. For examples are *palama* >> *palame(mount)*. *Palame* is written the same way as perfect tense and mood tense for *palama*.

Morphological Generator

The generator rules overall work well. Most Setswana verbs are productive; producing between 80 and 150 words per word. The generator can also handle complex word formations involving multiple word categories such as *tirisa + Tirana*, *tiregi + Tirana* and other combinations. From a list of 1500 Setswana verbs the generator produced 133 567 words. Although Setswana is considered to have a limited vocabulary, compared to other languages, its morphology allows for creation of new words if needed. This tool could help in generating new vocabulary.

Although the process of generating words in Setswana is relatively simpler than that of the analyzer, there is increased ambiguity in some formations. Some generated words do not make sense. They are morphological are correct but are not used or preferred.

For examples

fetlha >> *le + fetlha* >> *lefetlho(stirrer)*

>> *se + fetlha* >> *sefetlho*

ipona >> *le + bona* >> *leiponi*

>> *se + bona* >> *seiponi*

In the example above the verb *fetlha(stir)* is combined with *le-* or *se-* to form a noun. In Setswana *lefetlho* is used instead of *sefetlho*. *Se-* and *le-* are used to form nouns from verbs. We could not find any consistency in the use of *le-* or *se-*. Therefore our generator produces both forms. There are many such cases with other prefixes such as *bo-*, *i-*, *n-* and *m-*.

5. CONCLUSIONS

A rule-based morphological analyzer for Setswana verbs has been developed. The verbs' morphology is fairly regular for most categories resulting in a high lemmatization rate of 87%. The loss in performance is mainly due to overlapping words. The significance of the errors and error handling methods would heavily depend on the application of the analyzer. We believe the proposed analyzer is adaptable to many applications. We intend to develop mechanisms that could help morphological rules resolve conflicts in overlapping words. A generator was also developed. It allows transformation from verbs to other verbs and nouns by word category and multiple categories. The generator could be used to form new vocabulary.

6. REFERENCES

1. V. Balakrishnan and E. Lloyd-Yemoh "Stemming and Lemmatization: A Comparison of Retrieval Performances", *Lecturer Notes on Software Engineering*, Vol. 2 No.3, August 2014.
2. D.T. Cole, "An Introduction to Tswana grammar", Longmans and Green, Cape Town.
3. K. Mogapi, "Thuto Puo ya Setswana", Longman Botswana, 184, ISBN:0582 61903 3.
4. K. Brits, R. Petorius and G.B van Huyssteen, "Automatic lemmatization in Setswana: towards a prototype", *South African Journal of Languages*, 25:1, 27-47, 2013.
5. J.H.Brits "Outomatiese Setswana Lemma-identifisering: Automatic Setswana Lemmatization", *Master's Thesis. North West University, Potchefstroom, South Africa.2006*.
6. L. Petrorius and S.E. Bosch, "Computational aids for Zulu natural language processing", *Southern African Linguistics and Applied Language Studies* 21(4):276-282, 2003.
7. A. Chebanne, "Intersuffixing in Setswana: The case of the perfective -ile, the applicative -ela, and the causative -isa", *Pula: Botswana Journal of African Studies*. Vol. 10 No.2 pp. 83 - 94, 1996.

8. Kruger, Capser, "*Introduction to the morphology of Tswana*", Munchean, Lincon, pp314, 2006.
9. T.J. Otlogetswe, "*Poeletso-medumo ya Setswana: The Setswana Rhyming Dictionary*", Centre for Advanced Studies for African Society, 2010 ISBN: 978-1-920287-02-3.
10. T.J. Otlogetswe, "*Tlhalosi ya medi ya Setswana*", Medi Publishin, 2012. ISBN: 978-99912-921-3-7.