The effectiveness of morphological rules for an isiZulu spelling checker

Sonja E. Bosch
Department of African Languages, University of South Africa,
PO Box 392, UNISA, 0003
boschse@unisa.ac.za

Roald Eiselen
Centre for Text Technology, North West University,
Potchefstroom Campus
roaldeiselen@hotmail.com

This paper shows how morphological analysis contributes to solving the challenges posed by the development of a spelling checker for an agglutinative language like isiZulu. It demonstrates how the incremental implementation of affix removal rules can be used to derive word forms and enhance the lexical and error recall of the system. In the case of the spelling checker the strategies used are mainly based on the use of regular expressions, and more specifically on a process of stemming.

Introduction

The idea of using regular expressions for morphological analysis is not a novel one. Morphological analysis has been widely used to improve the lexical and error recall of different human language technology applications (Porter, 1980; Black et al., 1991; Kraaij & Pohlmann, 1996; Van Huyssteen & Van Zaanen, 2003). In the context of spelling checkers, lexical recall refers to the recognition of correctly spelled words by the spelling checker, whilst error recall is the accurate rejection of incorrectly spelled words (Starlander & Popescu-Belis, 2002:271). This technology is especially relevant to conjunctively written languages with an agglutinating morphological typology, where there is a high level of inflection, and prefixes and suffixes are used extensively in the formation of words.

In the development of spelling checkers, the problems encountered in the design of the programming structures that determine whether a word is correctly spelled or not will differ from language to language. Of particular interest are languages with a more complex morphology, like those belonging to the Bantu language family. These languages often require additional morphological processing during the word validation phase, since simple increase in the size of the lexicon is not only time-consuming, but also error-prone. Specific and accurate analysis at word level is therefore a necessity (Aduriz et al., 2000:2). Within the context of spellchecking the computational processing required is particularly the recognition of correctly spelled words by finding and computing the regularities of the language.

In the next section a brief overview of isiZulu morphology is given in order to shed some light on the need for morphological analysis in the development of an isiZulu spelling checker. This is followed by a discussion of the implications of morphological complexity for a spelling checker, and of how regularities in morphological complexity are modelled by means of regular expressions. Subsequently the implementation of regular expressions is explained, with specific reference to verb and noun analysis. The last section of the paper is devoted to an evaluation of the morphological analysis implemented in the isiZulu spelling checker, and also gives an indication of directions for future work.

IsiZulu morphology

The rich agglutinating morphological structure which characterizes isiZulu and other Bantu languages is based on two principles, namely the nominal classification system, and the concordial agreement system. According to the **nominal classification system**, nouns are categorized by prefixal morphemes. Noun prefixes generally indicate number, with the odd class numbers designating singular and the corresponding even class numbers designating plural. We follow Meinhof's (1932:48) numbering system, which distinguishes between 23 noun prefixes altogether in the various Bantu languages.

The **concordial agreement system** is significant in isiZulu because it forms the basis of the structure of the whole sentence. Concordial agreement is brought about by the various noun classes in the sense that their prefixes link the noun to other words in the sentence, such as verbs, adjectives, possessives, pronouns and so forth. This linking is manifested by a concordial morpheme which is derived from the noun prefix, and usually bears a close resemblance to it.

Derivation, **in morphology**, is the combination of morphemes to produce a new word in a different word category. Nouns are frequently derived from verb roots, which requires a noun prefix as well as a deverbative suffix, as illustrated in the following examples of nouns formed from the verb root *-hamb-* 'travel, go':

(1a) *u-(lu)-**hamb**-o* 'journey'

(1b) *u-m(u)-**hamb**-i* 'traveller'

The deverbative suffixes in (1) are *-o* and *-i.* Deverbative nouns may have more than one suffix if they are derived from verb roots that have been extended, e.g.:

(2) *u-m(u)-**hamb**-el-i* 'visitor'

Adverbs may be derived from nouns: for example, locative adverbs may be derived by prefixing a locative prefix *ku-* or *e-* and in some cases suffixing a locative suffix *-ini*, e.g.:

(3a) *indlu* 'house'
e-indlu-ini > *endlini* 'in the house'

(3b) *ikhanda* 'head'
e-ikhanda > *ekhanda* 'on the head'

(3c) *ubaba* 'father'
ku-ubaba > *kubaba* 'to/with/at/ father'

Inflectional morphology is the inclusion in a word of morphemes that do not change the word category, but add information such as tense, aspect, person, number and agreement. In the case of both nouns and verbs, prefixes and suffixes function as inflectional morphemes, e.g.:

(4a) *ngi-ya-**buz**-a* 'I ask' (1ps, present tense)

(4b) *ni-ya-**buz**-a* 'You ask' (2pp, present tense)

(4c) *si-zo-**buz**-a* 'We shall ask' (1pp, future tense)

(4d) *u-**buz**-ile* 'He/she asked' (3pp, class 1, perfect)

(5a) *u-m(u)-**lilo*** 'fire' (singular)

(5b) *i-mi-**lilo*** 'fires' (plural)

(5c) *u-m(u)-**lilo**-ana* > *umlilwana* 'small fire' (diminutive)

Both the derivational and inflectional processes are highly productive, especially taking into consideration that Doke and Vilakazi (1964:294) list eight nouns as being derived from the verb root *-hamb-*, and the subject concord *ngi-* (first

person singular) in the present tense verb *ngiyabuza* may be substituted by a subject concord from any of the noun classes. In addition an object concord from any of the noun classes may follow the subject concord.

Having outlined the complexities of the morphological structure of isiZulu, we now address the issues of morphological analysis and the challenges involved in building a spelling checker for the language.

Implications for an isiZulu spelling checker

The function of a spelling checker is to determine whether a word given as input is a correctly spelled word of the target language. There are, however, limitations to the computation that can be applied in spelling checking: these limitations are processing time and memory usage. They have a direct influence on the methods that are used to determine the correctness of a word. Certain methods have to be employed that will positively influence the functionality of the spelling checker.

In the past few decades different methods have been employed for this purpose, of which the most obvious and most widely used is a lexicon of correctly spelled words against which an input word is compared. If a given word occurs in the list, it is accepted as correct; otherwise it is flagged as incorrect. This method is also known as First Generation spellchecking, and is very accurate if the lexicon is properly revised and contains no incorrect words. Computationally, word lookup in the lexicon is also a faster method of determining whether a word is correct or not than other word validation processes such as morphological analysis.

For some languages like Sesotho sa Leboa and Setswana (disjunctively written Bantu languages) lexical recall of more than 98% is attained with lexica of fewer than 80,000 words, with little or no additional computation (also see De Schryver & Prinsloo, 2004a:57). In the case of Afrikaans (Van Huyssteen et al., 2004:98) for instance, a large lexicon of over 300,000 words is needed to reach the 98% lexical recall mark, because of both the inflectional nature of the morphology and the compounding features of certain word forms, which call for a more detailed analysis to determine their correctness.

In a highly agglutinative and conjunctively written language like isiZulu the need for morphological analysis is even more accentuated (cf. De Schryver & Prinsloo, 2004b:93), as there are only a limited number of uninflected words and a large number of flectional forms with a stem or root that combines with suffixes to form new lexical items. This inflectional nature of the language means that there are literally millions of possible words that can be derived from a limited number of roots and stems through the use of affixes. From the results in Table 1 it can be seen that if a lexicon of a similar size to those of disjunctively written languages is implemented, the lexical recall of the isiZulu spelling checker is significantly poorer than the recall for the two disjunctively written languages. Even with a lexicon of 225,000 words, the lexical recall of a First Generation isiZulu spelling checker reaches only 89% accuracy.

Table 1: Lexical recall in relation to size of lexicon

Language	Approximate size of lexicon	Lexical recall
Setswana	73,000 words	98%
Sesotho sa Leboa	57,000 words	98%
isiZulu	60,000 words	82%

Although it may be possible to construct a lexicon that contains all possible word forms for a language like isiZulu, this would not be a practical solution, for two reasons. Firstly such a comprehensive lexicon would be too large and time-consuming to validate manually. Secondly, and more importantly, the amount of physical memory needed to load such a lexicon would be unacceptably large, since one of the restrictions imposed on the spelling checker in

the development stage was that the entire spelling checking session should not use more than six megabytes (6 Mb) of memory. A lexicon of 225,000 words requires approximately 2.5 Mb of memory and any lexicon larger than 500,000 words would need more than 6 Mb of memory. A lexicon of this size would in all probability still not be large enough to make the spelling checker's rate of word recognition acceptable.

Second Generation spelling checkers include some form of automatic morphological analysis, supplementary to a lexicon. Such analysis makes it possible for a spelling checker to accept correctly spelled words that are not contained as entries in the lexicon and can increase the lexical recall of a spelling checker significantly (Van Huyssteen & Van Zaanen, 2003:150). The morphological analysis implemented in the spelling checker makes it possible for it to recognize a large number of words, without increasing the size of the lexicon.

There are, however, also drawbacks to the use of morphological analysis, of which two are of significance in the development of spelling checkers. The first is the problem of overgeneration, where words are recognized as correctly spelled, according to the formalism, but are in fact incorrectly spelled (cf. example 19). These errors mainly occur if the analysis module does not cover the entire morphology of the language, or if rules intended for verbs are applied to nouns. In such cases an incorrect analysis of a word will lead to the intended word being accepted as a correctly spelled word.

Secondly, morphological analysis is a much slower process than looking up a word in a lexicon. In an experiment to determine the time each word validation process in the spelling checker takes, dictionary lookup took 0.00088 seconds on average, while the morphological analysis of a word took an average of 0.003862 seconds. This means that it takes almost 44 times longer for morphological analysis to validate a word than looking up a word in the lexicon does.

The aim of the next sections is to investigate and illustrate how morphological analysis can be used to accept a large number of valid words without increasing the amount of memory needed in the processing or impairing the speed of lookup, while not significantly increasing the recognition of incorrect words either.

Recognizing regularities in morphological complexity

In order to compute all the possible word forms of a conjunctively written language it is necessary to find some form of morphological regularity in the structure of the language, and to use a computational tool to recognize such regularities (Jurafsky & Martin, 2000:87). This calls for the development of a morphological grammar of the language that can be implemented as a finite set of rules that model its regularities.

There are various ways of modelling the regularities in language, such as finite-state automata (FSA), transducers (FST) and machine learning algorithms, but one of the most widely used of these strategies is the use of regular expressions. Regular expressions can be used to determine whether a given string (or list of symbols) forms part of the set of possible strings that are included in the grammar of the language by deconstructing an input word. Regular expressions are pattern-matching devices for recognizing given patterns in the target language. Each regular expression is able to define only a restricted, or finite, set of language elements, as described by its syntax. The following Perl-style regular expression aims to match three elements in an input word:

(6) /^(u)(.*)(a)$/

Such a regular expression would match all input words that have 'u' as the first character and 'a' as last character, with one or more characters between 'u' and 'a'. This means that both input strings *ubbbbba* and *usebenza* 'he/she/it works' are defined as part of the regular language represented by the regular expression in (6) above.

In example (7) the most complex of the regular expressions used in the analysis module is outlined. Each of the components in the regular expression is implemented as a variable group representing a particular affix group, but for the example only one member of each group is given.

(7) /^(ba)(ya)?(ngi)?(.+)(el)?(a|e)(ni|phi)?$/

The regular expression in this example would recognize a wide range of words, from those as simple as *bakloloda* 'they jeer', to more complex constructions such as *bayangiklolodelani* 'why do they jeer at/ make fun of me?', where the root (*-klolod-* in the two examples) is identified among all the possible affixes. The question marks in the regular expression mean that a given part of the expression is optional, and the pipe character (|) means that either of the two final suffixes may be present. Other than the simplest and most complex constructions, 46 other word forms with *-klolod-* as the root would be matched by this regular expression. Each prefix (*ba-*, *-ya-*, *-ngi-*) in the regular expression can also be replaced by one or more other prefixes to recognize further word forms based on the same root. The subject concord *ba-* can be replaced by *u-*, *zi-*, or a number of other subject concords. The same applies to the object concord *-ngi-*. For each change in any of the prefixes, 48 different word forms are recognized.

As an extension to the regular expressions that were implemented it was necessary to implement so-called rewrite rules in some instances. These rules not only match strings, they also produce transformations of the initial input string. They are usually context-sensitive, thus mapping a given string to a new string. These implementations are especially significant for words where morphophonological changes take place, as in example (8).

(8) *ngomuntu* > *nga-umuntu*

After a particular pattern is matched (in this case, *ngo-*), the remaining pattern can be altered by concatenating new strings to the identified substring or by changing elements matched by the regular expressions to reflect these changes. The new lexical form, changed by the module, can then be used to determine whether a string is part of the isiZulu language by comparing the transformed string to lists of isiZulu verb roots and nouns. In the example, the rewrite rules change *o-* to *u-* in order to find *umuntu* in the list of nouns, as opposed to looking for **omuntu*, which is not a correct isiZulu noun citation form.

The implementation of regular expressions for the isiZulu spelling checker

The development of the isiZulu morphological analysis module in the spelling checker is exclusively based on regular expressions, and more specifically on a process of stemming through lexical decomposition. Stemming is a process which reduces morphological variants of a word to the single root or stem of the variant. This is most commonly achieved by removing suffixes, as in the Porter and Lovins Stemmer (Hull & Grefenstette, 1996), or by truncating specific character strings (Kraaij & Pohlmann, 1996; Van Huyssteen & Van Zaanen, 2003). The morphological analyser that was included in the spelling checker can be divided into two main modules, namely a verb analysis module and a noun analysis module. These modules function as truncation algorithms which remove all affixes from word variants to find common stems or roots.

Other morphologically complex constructs, specifically demonstratives, adjectives (46 stems plus prefixes in various tenses) and relatives (372 stems and prefixes in various tenses) were generated by using regular expressions that were implemented outside the spelling checker itself. The generated lists added to the lexicon were fairly small, all in all comprising 7,543 words, thus not extending the lexicon beyond the 500,000 word maximum. Therefore, the memory usage of the spelling checker was not adversely affected.

The morphological analysis was implemented in two major phases. The first phase included a component for basic verb and noun analysis, referred to as the Initial Analysis Module. During the second phase additional verb and noun constructs, such as the negative, relative and copula constructions were added; this was referred to as the Extended Analysis Module. The reasoning behind this was firstly to implement rules which would recognize the greatest number of correct forms, and to add less frequent constructions later in the development. Secondly the rules were implemented in such a way as to determine whether the impact of any rule on the error recall of the spelling checker would be negative.

Verb analysis module

The verb analysis module is probably the more comprehensive of the two modules. Its objective is to identify the root of the verb by stripping affixes from the input word until a valid verb root is extracted. This root is then compared to a list of approximately 7,500 possible verb roots extracted from an isiZulu dictionary (Doke & Vilakazi, 1964).

In total there are 14 sets of rules for identifying verb roots. Each set is distinguished by the first prefix of the verb and the corresponding final vowel. In addition, each set consists of six further rules for determining the other possible affix combinations that may be present in the input word.

Analysis starts by identifying a correct combination of a single possible prefix and a final vowel, as this is one of the simplest possible verb constructions. Although it is possible to implement the first prefix and last suffix in separate expressions, valuable computing time can be saved by omitting rules where a string does not comply with this first function, that is where it does not have a final vowel that agrees with the concord prefix in terms of the formalism.

If both the prefix and final vowel are correct according to the grammar (i.e. the regular expressions), these two affix strings are removed, leaving a string that can be checked against the verb root list as output. In the case of a simple construction like *bakloloda* 'they are jeering', *ba-* (subject concord class 2) and *-a* (verb final) are removed, leaving the module with *-klolod-*, which is in the verb root list, and therefore *bakloloda* 'they are jeering' is accepted as a correct word. The word *bayakloloda* 'they are jeering' will not be recognized on the first pass, as the output of the module will be *-yaklolod-* (i.e. verb root plus the long present tense morpheme *-ya-*). Before this word is rejected it will be matched against further regular expressions that represent all possible prefix and suffix combinations, to determine what morphemes are still present in the word. The most complex verb construction handled by the module is the following combination of verb root plus affixes:

Table 2: Possible verb prefixes and suffixes

Subject concord/ Relative concord	**Prefix** (Negative/ progressive/ present tense)	**Object concord**	**Verb root**	**Extension(s)**	**Final vowel (positive/ negative)**	**Interrogative suffix** (*-ni* / *-phi*)

For instance, the word *bayangiklolodelani* 'why do they jeer at/ make fun of me?' will be analysed as follows:

(9) *ba-ya-ngi-**klolod**-el-a-ni*

The following table explicitly explains the function and meaning of each of the morphemes in (9):

Table 3: Morphological analysis of *bayangiklolodelani*

ba	*ya*	*ngi*	*klolod*	*el*	*a*	*ni*
They		me	jeer	at		why
Subject concord, class 2	Present tense prefix	Object concord, 1ps	Verb root	Applied extension	Final vowel	Interrogative suffix

Except for the subject concord and the final vowel all other parts of the construction are optional, although their specific relations to those morphemes around them are set. For this reason the list of verb roots is checked each

time an affix is removed from the input string. There are, however, instances where a given input word might be incorrectly analysed if the word is considered in a particular semantic context. This occurs when a word is not completely analysed before the module tests true and terminates. An example is the following:

(10a) *bazothela* > *ba-zo-**thel**-a* 'they will pour'

(10b) *bazothela* > *ba-**zothel**-a* 'they are dignified for'

The simple root *-thel-* as well as the extended root *-zothel-* are included in the list of verb roots. When a form such as *bazothela* is analysed, the shorter verb root *-thel-* 'pour' is not identified because a longer extended root *-zothel-* 'be dignified for' is recognized before the full analysis is completed. In other words the analyser would first find the form *ba-**zothel**-a* and would accept this as the correct form, although this may not be the correct analysis in the context in which it appears. However, this is not a problem as far as spellchecking goes, as the purpose of spellchecking is only to determine whether an input word is correctly spelled, and not to determine the meaning of the word.

As has already been mentioned, there are some cases where morphophonological alternations take place in a word. Some of the alternation rules for verbs that were implemented in the analysis module are:

(11a) Vowel elision
*a-ngi-yi-**akh**-i* > *angiyakhi* 'I do not build it'
/^(a)(ngi)(ya)(.+)(i)$/

(11b) Consonantalization
*ba-ya-ku-**akh**-a* > *bayakwakha* 'they build it'
/^(ba)(ya)(kw)(.+)(a)$/

However, more complex morphophonological changes, such as palatalization in the case of certain passive extensions, were not implemented, as this would have opened the way for the unintended recognition of invalid words, and therefore forms like the one in example (12) are not included in the analysis module.

(12) *u-ku-**bamb**-w-a* > *ukubanjwa* 'to be caught'

Noun analysis module

The analysis of nouns is similar to that of verbs although the output of the noun analysis is not a root, but a noun stem with its class prefix attached. The main reason for only analysing to noun stem level is the dependency of each noun class on class prefixes. In order for the analysis to be accurate, the stems would have had to be divided into classes and tested to determine if a particular class prefix could combine with a particular noun stem. This would have increased the processing time significantly. Instead, a list of 27,000 noun stems with their class prefixes (Bosch & Pretorius, 2004) formed the basis of a simpler morphological analysis module, which recognizes the same number of correct words.

The noun analysis module can be divided into two sets of rules with different functions but the same output. The first part focuses on the removal of prefixes preceding the noun, e.g.:

(13) *nga-umgunya* > *ngomgunya* (adverbial prefix + noun)
'with various species of plants'

The first part of the noun analysis module consists of the prefix rules (except the rule for the locative prefix *e-*), which amount to 24 rules in all. These include two constructions, namely (i) the formation of adverbs from nouns by means of adverbial prefixes in combination with the relevant morphophonological changes as illustrated in (14); and (ii) the formation of copulatives from nouns, as exemplified in (15).

(14a) Vowel elision
nga-abantwana > *ngabantwana* 'with the children'

(14b) Vowel coalescence
na-indoda > *nendoda* 'and a/the man'

(15a) 'y' + noun commencing with *i-*
y-indoda > *yindoda* 'it is a/the man'

(15b) 'ng' + noun commencing with *a-*/*o-*/*u-*
ng-abantu > *ngabantu* 'they are people'
ng-obaba > *ngobaba* 'it is father and company'
ng-umuntu > *ngumuntu* 'it is a/the person'

In addition, subject concords (present, past and compound tenses) may be prefixed to copula constructions like those above, e.g.:

(16a) *u-y-indoda* > *uyindoda* 'he is a man'

(16b) *waye-y-indoda* > *wayeyindoda* 'he was a man'

The second part of the noun analysis module deals with the removal of the combination of the locative prefix *e-* and the locative suffix *-ini*, e.g.:

(17) *ezingazini* < *e-izingazi-ini* (locative prefix + noun + locative suffix)
'into the vitals'

This second part, consisting of 19 rules, is only concerned with the locative construction, namely *e-*[noun]*-ini*. The reason for this is that in the locative there are morphophonological changes to both the prefix, where the initial vowel of the noun is discarded, and the suffix, which can have any one of 19 realizations depending on the vowel ending of the noun, as can be seen in the following constructions:

(18a) Vowel coalescence (e.g. 'a' + *ini* > 'eni')
e-ubuntwana-ini > *ebuntwaneni* 'in the childhood'

(18b) Consonantalization (e.g. 'u' + *ini* > 'wini')
e-ubugebengu-ini > *ebugebengwini* 'at the plundering'

(18c) Palatalization (e.g. 'mo' + *ini* > 'nyeni')
e-imilomo-ini > *emlonyeni* 'in the mouth'

Both these parts of the noun analysis module are specifically concerned with the morphophonological alternations that are inherent to these constructions as demonstrated in examples (13) to (18).

Evaluation of the morphological analyser as implemented in the isiZulu spelling checker

In order to evaluate the effectiveness of the morphological analysis component in the spelling checker module, a test corpus of approximately 30,992 words was collected, consisting of various text genres. This test corpus was then used to evaluate the spelling checker in different stages of the development project, as different lexicon sizes and sets of rules were implemented. Altogether nine different spelling checkers were evaluated to determine how the increase in lexicon size, and also the addition of the different morphological rules, impacted on the recall of the spelling checker. The nine stages of the spelling checker used for the evaluations are represented in the following table.

Table 4: Different spelling checkers used for evaluation purposes

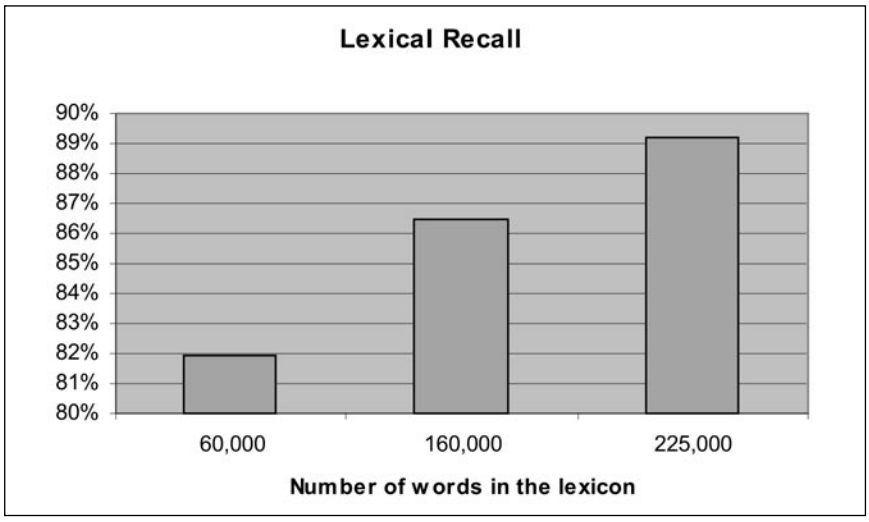
Stage	Lexicon Size (Number of words)	Morphological Analysis Implemented
1	60,000	None
2	160,000	None
3	225,000	None
4	60,000	Initial Analysis
5	160,000	Initial Analysis
6	225,000	Initial Analysis
7	60,000	Extended Analysis
8	160,000	Extended Analysis
9	225,000	Extended Analysis

By evaluating the spelling checker at different stages, the influence on the recall of the spelling checker could be accurately traced and monitored to determine specific shortcomings and problems with the modules.

From the first set of evaluations (Figure 1) with only lexicon lookup (i.e. First Generation Spelling checkers) it can be seen that the most common words can initially be recognized by a limited lexicon of approximately 60,000 words, attaining a lexical recall score of 81.91%. This means that nearly 82% of the words in the text are recognized as valid words. This reflects the words that are most commonly used, and makes up the largest number of words in any given text.

Once these initial high-frequency words are accepted, increasing the number of words in the lexicon only effects minor increases in the lexical recall. With a lexicon of approximately 160,000 words, the recall only increases to 86.44% (an increase of 4.53% in lexical recall when 100,000 words were added to the lexicon), and with an additional 65,000 words (i.e. a lexicon of 225,000 words), the recall only increases by 2.75% to 89.19%. From these results it seems that the size of the lexicon has an ever-decreasing positive impact on the lexical recall.

Figure 1: Lexical recall results of lexicon based isiZulu spelling checkers

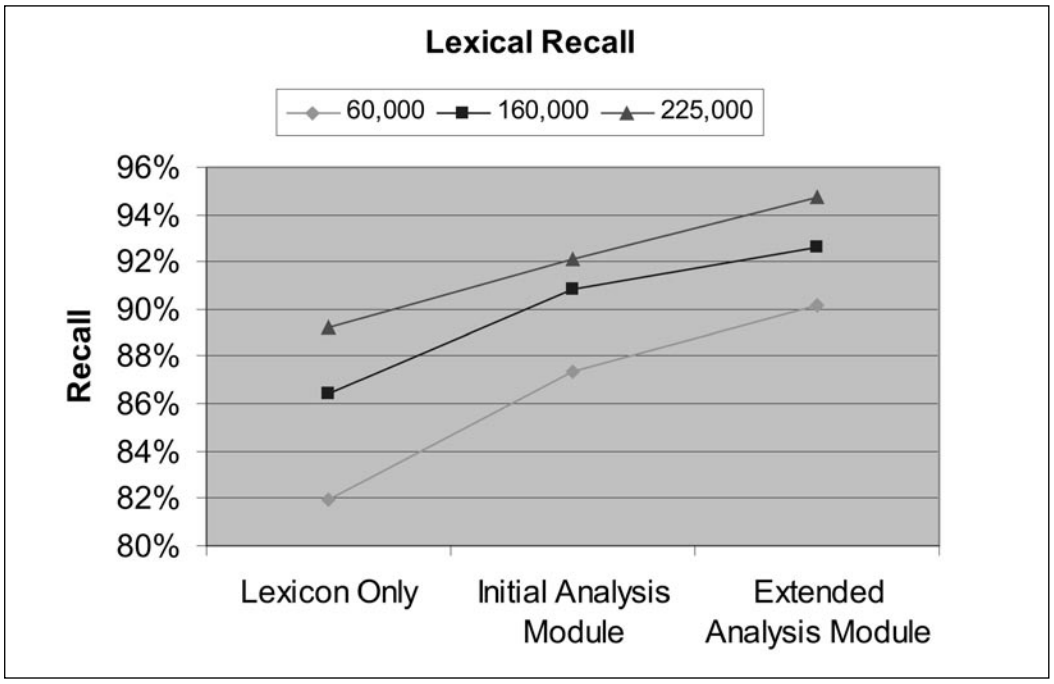


Increasing the size of the lexicon is time-consuming work, because all of the words that are included in the lexicon need to be validated, which takes approximately 17 man hours per 10,000 words. This validation process is also prone to human error, as validating lexicons is an extremely laborious process during which incorrect words are sometimes missed, and therefore included in the lexicon.

With the introduction of the Initial Analysis Module, the lexical recall of the spelling checker is increased significantly, with far less human input in the process. In these experiments an initial set of verb rules (60 regular expressions) and noun rules (28 regular expressions) was implemented. These rules increased the lexical recall to 87.35%, 90.86% and 92.08% for the respective lexicon sizes (see Figure 2). These figures relate to increases of 5.44%, 4.42% and 2.89% respectively. As an example, the increase of 4.42% on the 160,000-word lexicon, is more than the addition of another 65,000 words to the lexicon would have made.

With the implementation of the Extended Analysis Module, containing additional verb and noun constructs such as negative, relative and copulative constructions, the smallest lexicon of 60,000 words reaches 90% lexical recall. With a lexicon of 160,000 words, the recall of the spelling checker is increased by 6.15% to 92.59%. In the evaluation with the full lexicon of 225,000 words, there is an increase of 5.58%, giving the final spelling checker an overall lexical recall figure of 94.77%.

Figure 2: Impact of morphological analysis on lexical recall of an isiZulu spelling checker



Although the use of regular expressions as a morphological analysis component of the spelling checker has a positive impact on the lexical recall, some incorrectly spelled words are also recognized by the spelling checker as correct. An example is the following:

(19) **lisondelelela* < *li-s-**ondel**-elel-a*

The extended verb root *-ondel-* 'long to possess' appears in the root list, while the extension *-elel-* features in the affix list. However, the extended verb root already contains the applied extension *-el-* and is therefore incompatible with the duplicated applied extension *-elel-*.

Of a total of 848 incorrectly spelled words in the test corpus, only 27 are recognized by the spelling checker as correctly spelled words, thus showing a decrease in the error recall of the system of 3.18%. This might not be ideal, but it should be borne in mind that overgeneration is part of almost all automatic language processing systems (Van Huyssteen & Van Zaanen, 2003). During the development process a close check was done to ensure that the recognition of incorrect words by the analysis module was kept to a minimum.

Conclusion and future work

This article has shown how morphological analysis can contribute to solving the challenges posed by the development of a spelling checker for a conjunctively written agglutinative language like isiZulu. It has demonstrated that a mere increase in the size of the lexicon only effects minor increases in lexical recall after a threshold of 60,000 words, while the incremental implementation of morphological rules by means of regular expressions enhances the lexical recall of the spelling checker significantly. The morphological analysis was implemented in two phases, the first phase consisting of a component for basic verb and noun analysis (Initial Analysis Module), and the second phase including more complex verb and noun constructions (Extended Analysis Module). A test corpus was used to evaluate the spelling checker at different stages of development, as varying lexicon sizes and sets of rules were implemented. By evaluating the spelling checker at different stages, the influence on the recall of the spelling checker could be accurately traced and monitored to determine specific shortcomings and problems with the modules.

Although the morphological rules were implemented successfully, this cannot be seen as the end of the development of isiZulu spelling checkers. A relatively large number of words is still not recognized by the spelling checker, for instance place names and newly-coined words need to be dealt with. In addition, more rules need to be added to the module currently implemented, for example the palatalization rules that apply in the case of certain passive extensions.

Further attention also needs to be given to one of the main functionalities of a spelling checker, namely providing accurate suggestions for a given misspelling. At this stage suggestions can only be provided from words that are part of the lexicon that is implemented in the spelling checker. Research into the possibility of creating suggestions based on morphological analysis through the creation of similar words should be one of the next steps in the development of forthcoming spelling checkers for the isiZulu language.

References

Aduriz, E., Agirre, I., Aldezabal, I., Alegria, X., Arregi, J.M., Arriola, X., Artola, K., Gojenola, A., Maritxalar, M., Sarasola, K. & Urkia, M. 2000. A Word-Grammar based morphological analyzer for agglutinative languages. *Proceedings of the 18th International Conference on Computational Linguistics* (COLING). Saarbrucken, Germany:1−7.

Black, A.W., Van De Plassche, J. & Williams, B. 1991. Analysis of unknown words through morphological decomposition. *Proceedings of the fifth Conference on European chapter of the Association for Computational Linguistics*. Berlin, Germany:101−106.

Bosch, S.E. & Pretorius, L. 2004. Software tools for morphological tagging of Zulu corpora and lexicon development. *Proceedings of the 4th International Language Resources and Evaluation Conference* IV. Lisbon: LREC2004:1251−1254.

De Schryver, G-M. & Prinsloo, D.J.. 2004a. Spellcheckers for the South African languages Part 1: The status quo and options for improvement. *South African Journal of African Languages* 24(1):57–82.

De Schryver, G-M. & Prinsloo, D.J. 2004b. Spellcheckers for the South African languages Part 2: The utilisation of clusters of circumfixes. *South African Journal of African Languages* 24(1):83–94.

Doke, C.M. & Vilakazi, B.W. 1964. *Zulu-English dictionary*. Johannesburg: Witwatersrand University Press.

Hull, D.A. & Grefenstette, G. 1996. A detailed analysis of English stemming algorithms. Technical Report TR MLTT-023, Rank Xerox Research Centre, Meylan, France.

Jurafsky, D. & Martin, J.H. 2000. *Speech and language processing*. Upper Saddle River, NJ: Prentice-Hall.

Kraaij, W. & Pohlmann, R. 1996. Viewing stemming as recall enhancement, in *Proceedings of ACM-SIGIR-96*, edited by H.P. Frei, D. Harman, P. Schauble & R. Wilkenson. New York: ACM Press:40–48.

Meinhof, C. 1932. *Introduction to the phonology of the Bantu languages*. Berlin: Dietrich Reimer/Ernst Vohsen.

Porter, M. 1980. An algorithm for suffix stripping. *Program* 14:130–137.

Starlander, M. & Popescu-Belis, A. 2002. Corpus-based evaluation of a French spelling and grammar checker. *Proceedings of the Third International Conference on Language Resources and Evaluation*. Las Palmas, Spain:268–274.

Van Huyssteen, G.B. & Van Zaanen, M.M. 2003. A spellchecker for Afrikaans based on morphological analysis. *Proceedings of the 6th International Terminology in Advanced Management Applications Conference (TAMA2003)*. Pretoria, South Africa:189–194.

Van Huyssteen, G.B., Eiselen, E.R. & Puttkammer, M.J. 2004. Evaluating evaluation metrics for spelling checker evaluations. *1st International workshop on proofing tools and language technologies*. Patras, Greece:91–99.