

Non-word error detection in current South African spellcheckers

DJ Prinsloo^{1*} and Gilles-Maurice de Schryver^{1,2}

¹ Department of African Languages, University of Pretoria, Pretoria 0002, South Africa

² Department of African Languages and Cultures, Ghent University, Rozier 44, B-9000 Ghent, Belgium

* Corresponding author, e-mail: prinsloo@postino.up.ac.za

Abstract: The main objective of this article is to investigate the effectiveness of the current South African spellcheckers built by the authors of this article.¹ To this end translations of the same text, namely the Universal Declaration of Human Rights, will be spellchecked, and the performances compared across various sample languages. The spellcheckers themselves will be engaged in layers, so as to monitor the effect of the number of items in the spellchecker lexica. This innovative research will be preceded by a discussion of HLT (Human Language Technologies) in South Africa and the authors' philosophy in this regard, a brief theoretical conspectus of spellcheckers, a presentation of the methodology employed to create the spellcheckers, an in-depth study on how to deal with diacritics in spellcheckers, and a walkthrough of standard spelling and grammar checking functions — all with specific reference to the African languages.

Introduction

Human Language Technologies in South Africa

All efforts regarding state-of-the-art, high-tech development of especially the African languages in South Africa should be applauded. We believe, however, that such activities and strategies for their advancement ought to be sensitive to certain local realities, and should address Human Language Technology (HLT) requirements on a *priority* basis rather than according to an *ideal* HLT-development schedule. This means that major projects must be designed in such a way as to render *regular spin-offs*, i.e. usable applications that are urgently needed. This might even entail taking shortcuts in the short term in order to provide products for immediate use for which the technology is in real terms still under development.

This crucial philosophy underpins all our work, and is no different when it comes to the current creation of South African spellcheckers, the topic of the present contribution. African languages in particular require what could be called first-generation spellcheckers *now* to satisfy the immediate needs that could be described as software modules that can detect most incorrectly typed words and can suggest alternatives. This should be followed by subse-

quent, more sophisticated and improved, second-generation spellcheckers with a better performance, and ultimately even superior third-generation spellcheckers which can also check grammatical structures. We are thus convinced that if ways can be found to satisfy the immediate requirements of the users of specific languages, the process should not be delayed simply for the sake of releasing a more advanced spellchecker as the first product.

Since this is a pioneering publication on the issue for the African languages, and as spellcheckers are a relatively new research field in South Africa, it was felt that it would be useful to lay the groundwork and to devote some time to the basics first, albeit in a non-complex manner. To this end, a brief theoretical conspectus will be presented in which spellcheckers will be defined and the main strategies for their construction described. This will make it possible to place current South African endeavours in context, and will then lead to an elucidation of our own approach. This section will be followed by one in which the illusion is shattered that spellchecking is not yet feasible for African languages that are written with scores of so-called 'special characters', as is the case in, for instance, Tshivenda.

Subsequently, a walkthrough of the basic functions of spellcheckers will be presented for the benefit of the reader who might not be familiar with the use of such a tool. An exhaustive coverage of the functions of spelling and especially grammar checkers will, however, not be attempted. Although the latter are often mentioned in the same breath as spelling checkers, and although they are briefly illustrated below, grammar-checking technology lies beyond the scope of this article. The emphasis will rather be on typical functions pertinent to current and future developments in spellcheckers for the African languages. In the final section the launching of our own spellcheckers for all official South African languages, viz. for the nine African languages and Afrikaans (English already being catered for in the group of world-English spellcheckers), will be discussed. The performance of these spellcheckers will be illustrated with an in-depth evaluation for Sesotho sa Leboa, isiZulu and Afrikaans. In each case the (lexical) recall values for translations of the same text, namely the Universal Declaration of Human Rights, will be calculated. The obtained results will then be placed in a wider context, by way of a comparison with four other languages spoken on the African continent, viz. Hausa, Somali, Lingala and isiXhosa. The article will be concluded with some suggestions for improving the next-generation spellcheckers.

Brief theoretical conspectus of spellcheckers

From the early 1960s onwards, researchers have designed various methods for the computational detection of erroneous words in running electronic text. Today, four decades later, there isn't any reputable word processor that doesn't include spelling checkers, as well as spelling suggestors and/or correctors, and even thesauri and grammar checkers as integral parts. This is true for all languages with significant worldwide commercial importance, less so for those languages with a limited commercial value. Sadly, commercially available spellcheckers are unfortunately the exception rather than the rule in the case of African languages.

The term 'spellchecker' is used here to cover what the average user understands under this term today, i.e. a software application, generally integrated into a word processor

like Microsoft Word or Corel WordPerfect, which: (i) *checks* for spelling (and grammatical) errors, (ii) *automatically corrects* some typographical errors, (iii) *suggests* stylistic and punctuation replacements, and (iv) often includes a *thesaurus* (i.e. a list with synonyms and antonyms). Thus, in over-simplified terms, it can be said that the purpose of a spellchecker in word processing software is to alert the user to possibly incorrectly typed words or strings of text and to suggest options for correction. This article will be concerned with the 'word' level only, and not with 'strings of text'; and will mainly focus on the 'detection' of errors rather than on their 'correction'. Reformulated, this means that 'non-word error detection' will be treated, non-words being words that do not exist. Non-word error detection is a necessary first step towards a truly professional spellchecker, i.e. one where 'context' (and by extension 'grammar') also comes into play.

Basically there are two main approaches to the creation of spellcheckers. Firstly, a program can simply compare the spelling of typed (or scanned) words with a so-called 'spellchecker lexicon', being a stored list of valid *full orthographic words*. Secondly, one can program software with a proper description of a language, including detailed morphophonological and syntactic rules, which computes over a series of stored lists of *word roots*. Whereas the first approach only stores full orthographic words, the second approach has very few of those. In the first approach all inflections and derivations of a lemma, as well as compounds, are thus physically listed — or 'spelled out' so to say; while they are analysed/generated on purely linguistic grounds in the second. For many languages either approach will give acceptable results, for others only the second approach is feasible. Physically listing all valid orthographic words in Finnish, for example, where word roots may easily have thousands of inflectional forms each, is simply impossible.

Although it can hardly be disputed that both the development and use of spellcheckers for the African languages at large are still in their infancy, single implementations of the two types of spellcheckers already exist for these languages. Wordlist-based spellcheckers for Sesotho sa Leboa, Setswana, isiZulu and isiXhosa were developed by DJ Prinsloo for *Corel WordPerfect 9*, and have been available

since 2000 (Prinsloo & De Schryver, 2001: 129). The sizes of these spellchecking lexica are rather modest, as they run into tens of thousands of items per language only. Conversely, Arvi Hurskainen began work on a rule-based spellchecker for Kiswahili in 1987, a product that saw the light as *Orthografix 2 for Swahili* a decade later, in 1999 (Hurskainen, 1999: 139). Starting from some 45 000 word roots, tens of millions of orthographic words may be recognised thanks to the incorporation of inflection and derivation technology. This spellchecker (and hyphenator) can be used with *Microsoft Word* and *Adobe InDesign* and is distributed by *Lingsoft* (Lingsoft, 1999). Focusing on South Africa, one can also observe that several spellcheckers for Afrikaans, both wordlist-based and rule-based, are already available commercially, some of which are currently being redesigned, such as *PUK/Microsoft Speltoetser*, *Pharos Speller* or *Ispell vir Afrikaans* (Van Huyssteen & Van Zaanen, 2003).²

Our methodology for the development of spellcheckers

Given that the creation of rule-based spellcheckers can easily take up to a decade, and in the light of our drive to be able to release applications at this point in time, it should not come as a surprise that our first-generation spellcheckers for the South African languages are word-based. One may then firstly wonder: Which words? Ideally one would of course wish to list *all* orthographic words of a particular language, but this is never possible. In any language a relatively small number of words occur extremely often, while the largest section of each language's lexicon occurs much less frequently or even rarely. A spellchecker lexicon with only a limited number of valid orthographic words will therefore obviously focus on the most frequent items. Reformulated, if software would only allow the storage of 1 000 words, then choosing the top-frequent 1 000 words in the language will result in the most efficient spellchecker with this 1 000-word restriction. Fortunately, tens of thousands, even hundreds of thousands, of orthographic words can be stored in current spellchecker lexica. So the next question is: Where does one find these hundreds of thousands of words? The answer could have been expected: Electronic corpora. For more than a decade corpora have been

compiled for all South African languages (Prinsloo, 1991; De Schryver & Prinsloo, 2000), and current spellchecker lexica are based on frequency lists derived from them. Each and every list proffered by corpus query software is of course manually checked before being loaded into software — a non-trivial and extremely labour-intensive activity. As will be illustrated in the various tests below, the performance of the current spellcheckers is generally good.

Thirdly, once spellchecker lexica have been produced as outlined above, one must write the appropriate software that enables the comparison of word processor text with the items in the lexica. Here modern word processors provide a powerful feature that can be put to good use, namely the possibility of adding 'custom dictionaries' to the main dictionaries. This basically means that the spellchecker lexica must be saved in plain text with the required extension, for instance '.dic' in Microsoft Word, and be engaged as custom dictionaries. One thus effectively 'borrows' all the required spellchecking functionality by running one's own dictionary in parallel with a main dictionary. This has the huge advantage that no additional programming whatsoever is required, and that all available detection and suggestion/correction functions can be drawn upon — even though this functionality will of course be faster in truly internal dictionaries. Running, say, Xitsonga in parallel with a main dictionary like English has the additional benefit that both languages are spellchecked simultaneously. Especially in the South African context, where many documents are of a multilingual nature, this is an added value. Actually, on our own machines, all eleven South African languages are activated, one as default language, and ten as custom dictionaries. This means that one can type text in any language — Setswana, isiNdebele, Sesotho — and swap to any other language — siSwati, Afrikaans, isiXhosa — at any time; all text will be spellchecked. It must, however, be remembered that no grammar is checked in this way, only specific word forms in isolation. From the moment that a word form is listed in a spellchecker lexicon, that word will be flagged as correct. There is thus no over-generation whatsoever in a word-based spellchecker, meaning that every accepted word also exists. It is of course still possible that a valid word was

used instead of an intended one, or that the syntax is wrong. When using multiple custom dictionaries, it is also possible that a non-word in the language of, say, custom dictionary A is accepted simply as a result of the fact that it is a valid word in the language of, say, custom dictionary B. Lastly, observe that not all paradigms are always complete in a word-based spellchecker lexicon. From a dictionary angle this means that not all inflections and derivations that belong to a particular lemma are necessarily listed, and thus necessarily recognised by the spellchecker.

Spellchecking 'all' South African languages: Reality or chimera?

So far it was claimed that by making use of standard word processing software in which top-frequency wordlists are loaded as custom dictionaries, a level of non-word error detection can be achieved which may be considered high enough as to render functional spellcheckers for the South African languages. One may, however, rightfully question this claim on purely technical grounds, given that some of the South African languages employ characters not normally included in commercial software. The velar and dental symbols for Tshivenda (cf. below) immediately come to mind.

As for most African languages, the character sets of the official South African languages are based on the Latin alphabet. For some of these languages, a number of diacritics have been added to a few base symbols, in most cases contrasting them with the base symbols themselves (e.g. **e** vs. **ê** or **o** vs. **ô**). If it goes without saying that a spellchecker for, say, French or Danish must be able to spellcheck words with the symbol **ç** or **ø** respectively, it of course also stands to reason that a spellchecker for Sesotho sa Leboa must recognise **š** and differentiate this symbol from **s**, or that a Tshivenda spellchecker must be able to handle all so-called 'special characters' used in this language. The Sesotho sa Leboa **š** and **Š**, for instance, pose no problem for either compiler or user of a spellchecker, since these symbols have been assigned ASCII values, namely

0154 for **š** and 0138 for **Š**. The special characters of Tshivenda, however, are more problematic.

Indeed, the great majority of the electronic documents available in Tshivenda are prone to typographic errors. On the Internet, for example, the diacritics for the single velar and four dental symbols are rarely used, and these symbols are all simply collapsed with their respective base symbols, viz. with **n** on the one hand, and **d**, **l**, **n** and **t** on the other. Where an effort is made to present the symbols correctly, webmasters either resort to a scanned image of printed text, or to a document saved in pdf (portable document format). An example of the former is illustrated in Figure 1.

Whereas a scanned image is of course not a proper solution, pdf is not without problems either, as can be seen from the screenshots shown in Figures 2 and 3.

In Figure 2 the diacritics were added manually, which is again not an acceptable solution, while Figure 3 indicates what happens when the fonts used in the creation of the pdf document have not been correctly embedded. Only in a few rare cases does one encounter pdf documents which are displayed satisfactorily for the end user. An example can be seen in Figure 4.

This situation for Tshivenda is in a way highly surprising since, on the one hand, all *modern*

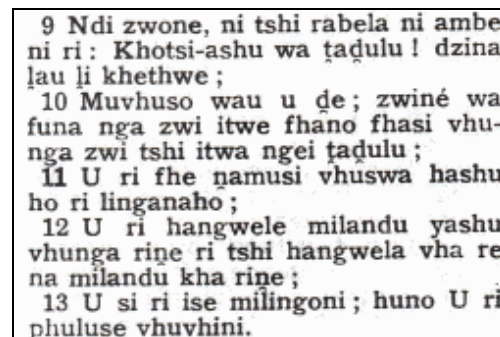


Figure 1: The Lord's Prayer in Tshivenda, uploaded as a scanned image*.³

* Due to the nature of 'screenshots' Figures 1–12 and Appendix 1 are not of the usual standard. They are included because of the value they add to the article. To view these figures in their original format consult the online version of the journal available at: <http://www.ingentaselect.com>

Nga nnda ha u tikedza mvelaphanda ya vhuandadza mafhungo, MDDA i do dovha ya ita thodisiso na u ita themendelo kha muvhuso, ndowetshumo ya vhuandadza mafhungo na mañwe madzangano. MDDA i do shumisana na madzangano othe ane a vha na dzangalelo kha mvelaphanda na thanjavhuwo ya vhuandadza mafhungo. MDDA i do fara mutangano wa ñwaha nga ñwaha hune madzangano othe ane a vha na dzangalelo a do kona u tola muvhigo wa ñwaha.

Figure 2: The *Media Development and Diversity Agency Position Paper* in Tshivenda, uploaded as a pdf with manually added diacritics⁴

Vhusimaulayo ha PANSALB ndi tsumbo khulwane ya uri hu na vhuñifhinduleli ha u ñdzhenisa na u lavhelesa mulayo wa luambo na pulane dzo ñandavhuwaho tshoñhe u katela tshipiña tshinwe na tshinwe tsha tshitshavha nahone ndi zwinezwi zwine zwa ri vhea phanña ha mañwe mashango. Ri na zwivhuya zwo engedzedziwaho zwa u kona u guda kha ññila dzo nangiwo huñwe fhethu kha ñino dzango. Huna mañwe maga a ndeme, ane a ñoña u tevhelewa kha u bula na u thoma mulayo wa luambo une wa nga konadzea na pulane.

Figure 3: *PanSALB's position on the promotion of multilingualism in South Africa* in Tshivenda, uploaded as a pdf without correct embedding of fonts⁵

- Pfesesa zwiteñwa na ñivhaipfi i tshimbilelanaho na:
 - vhuñe (tsumbo: Dzina ñanga ndi ...);
 - nomboro (tsumbo: nthihi, mbili);
 - muelo (tsumbo: khulwane, thukhu);
 - muvhala (tsumbo: tswukhu, ya ñaña).

Figure 4: The *Revised National Curriculum Statement Grades R–9 (Schools) — First Additional Language* in Tshivenda, uploaded as a pdf with correct embedding of fonts⁶

web browsers can display Unicode, while, on the other, all *recent* operating systems have a large number of Unicode fonts installed by default. In Unicode, all the world's scripts are supported:

The Unicode Standard is the universal character encoding standard used for representation of text for computer processing. ... Unicode provides a consistent way of encoding multilingual plain text and brings order to a chaotic state of affairs that has

made it difficult to exchange text files internationally. Computer users who deal with multilingual text — business people, linguists, researchers, scientists, and others — will find that the Unicode Standard greatly simplifies their work. ... The Unicode Standard provides the capacity to encode all of the characters used for the written languages of the world. (Unicode, 2003)

At present, there should thus be no problem

whatsoever in treating the African languages computationally, neither on the Internet nor on one's own PC. This is especially true for word processing software and, as illustrated for Hausa by Van der Veken and De Schryver (2003), for spellchecking. In other words, as long as fonts are used that are encoded according to Unicode, these fonts will not only be supported in a recent word processor, but also in the associated spellchecking modules.

Given that the 'special characters' required for Tshivenda are not normally included with standard PCs, the question arises: Can they be easily found? Yes they can. Firstly, since March 2002, Jako Olivier's *South African special characters font* can be downloaded from his homepage (Olivier, 2002). With this set, all South African special characters (for isiNdebele, isiZulu, Sesotho sa Leboa, Setswana and Tshivenda) can be treated in a word processor. Unfortunately, as the font doesn't comply with the Unicode standard, nor with any other standard known to us, documents created with this font are basically tied to this particular font. Documents cannot be interchanged, unless this font is installed on all computers where the documents are viewed, or unless the font itself is included with the documents.

Secondly, since September 2002, Victor Gaultney's *Gentium typeface*, encoded according to Unicode, has been online (Gaultney, 2002). As this font includes glyphs that correspond to all the Latin ranges of Unicode, this font may be used to handle Tshivenda (and the

other South African languages) in a word processor, and also in a spellchecker. Document interchange is also greatly facilitated as a result of this Unicode font character mapping. One is thus not tied to this particular font, as any other Unicode font can be 'dropped in'.

A Tshivenda spellchecker lexicon can be saved with the extension '.dic' in Microsoft Word, but now as Encoded text/Unicode, in order to keep all special characters embedded. This lexicon is then loaded as a custom dictionary. As a demonstration of the spellchecking process itself, the text from Figure 4 was retyped, albeit with three errors. These spelling errors are easily picked up by the spellchecker (and are indicated with red underscores, cf. below), as can be seen from the top half of Figure 5.

Spellchecking 'all' South African languages is thus definitely not a chimera, but has become a reality.

Since most Vhavenda and learners of Tshivenda who currently use word processing software do not make use of the special characters, however, it is advisable to prepare two spellchecker lexica for Tshivenda. One lexicon including all diacritics on **n**, **d**, **l**, **n** and **t**, and the other lexicon without any diacritics. The latter is of course easy to generate from the former by means of a straightforward search-and-replace procedure. Users who wish to employ and spellcheck the correct orthography can firstly decide to install a Unicode-based font for Tshivenda, such as Gentium, and then engage the Tshivenda spellchecker that includes the

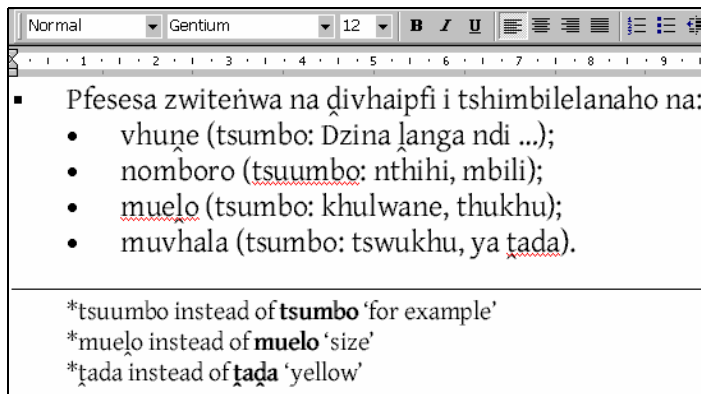


Figure 5: Spellchecking a Tshivenda text (with additional information provided for the errors under the horizontal line)

diacritics; while users who prefer to employ the Latin characters only (e.g. in e-mail correspondence — where one, in any language, typically cuts down on detail) may simply engage the simplified Tshivenda spellchecker that doesn't include any diacritics.

Standard spelling and grammar checking functions

The typical features of spelling and grammar checkers that will be outlined in this section are illustrated with Microsoft Word, since this is the word processor most widely used in South Africa. A first step for the user is to engage the spellchecker and to ensure that the correct main dictionary, as well as the correct supplementary and/or custom dictionaries (if applicable), is loaded. Clicking on the ABC-button on the navigation bar, see Figure 6, activates the spellchecking process.

If a main language was not preset, this can be done following the navigation steps in the selection boxes shown in Figure 7. (Note that no provision is made for an 'empty' default dictionary, which means that custom dictionaries have to run concurrently with an existing main dictionary.)

If supplementary and/or custom dictionaries are required (which is definitely the case in our approach), or if certain spelling and grammar preferences are to be set, these can be accessed by clicking Tools, Options and manip-

ulating the set of selection boxes shown in Figure 8.

In the automatic spelling and grammar checking mode (check spelling/grammar as you type), spelling components typically use red underscores (sometimes referred to as wavy red underlines) to indicate possible spelling errors, while grammar components use green underscores to indicate possible grammatical errors. This is illustrated in Figure 9.

The spellchecker underlined the words 'lemmatization', 'macrostructural' and 'microstructural' in red as suggested misspellings, and 'systems which' in green as a potential grammatical error. Such suggestions can be handled in two ways, either by means of the standard correction screen or a shortcut menu. The former, activated when clicking the ABC-button, is shown in Figure 10.

Figure 10 represents a typical example of the use of the standard correction screen offering the main options *ignore*, *ignore all*, *add*, *change*, *change all*, and *autocorrect*. In this

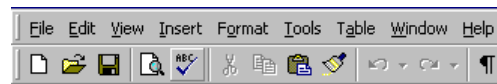


Figure 6: Clicking the ABC-button on the navigation bar activates the spellchecker

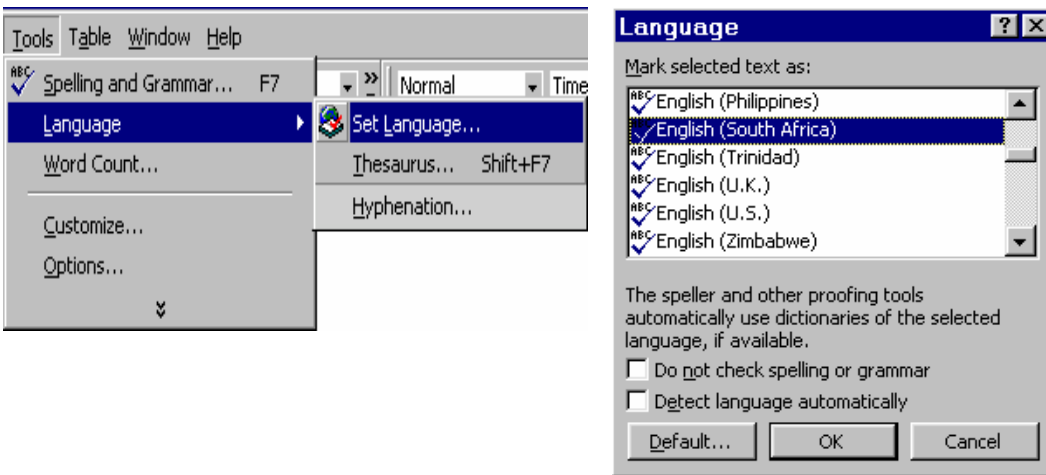


Figure 7: Selection boxes for presetting the main language

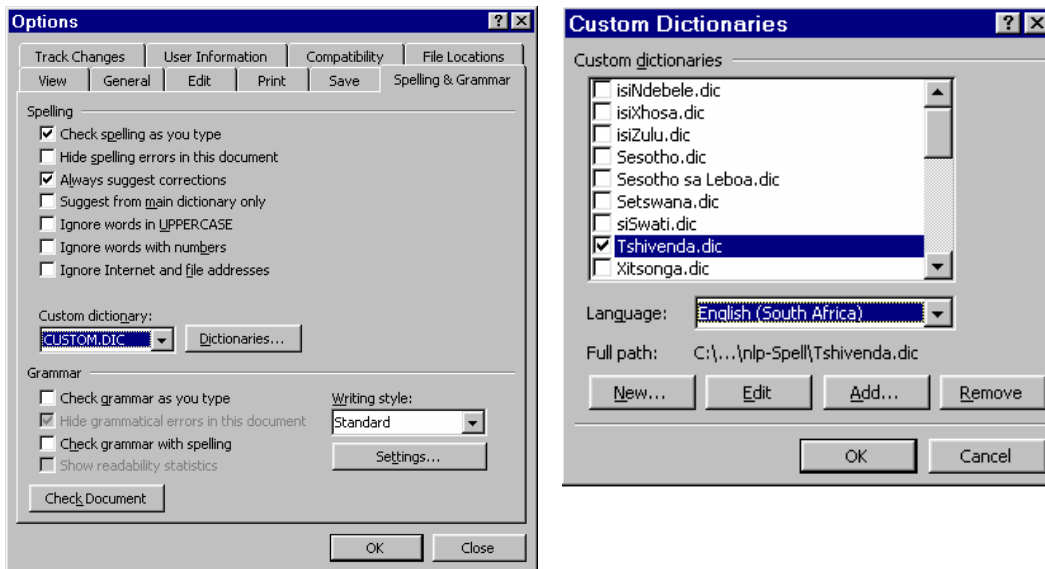


Figure 8: Selecting and engaging one or more custom dictionaries

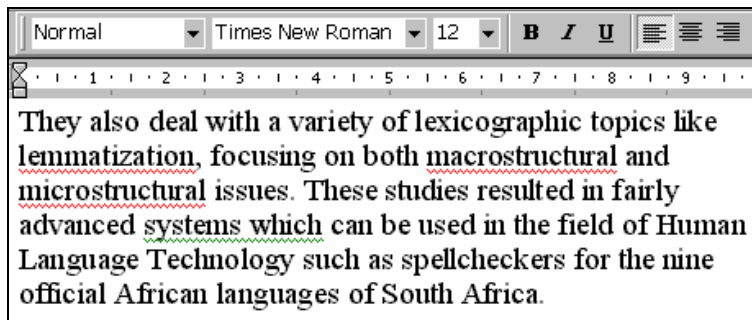


Figure 9: Red and green underscores indicating potential spelling (*lemmatization*, *macrostructural*, *microstructural*) and grammatical (*systems which*) errors respectively

case the spellchecker, which is set to South African English, suggests that 'lemmatization' should be spelled as 'lemmatisation'. Selecting *ignore* would leave the current occurrence unchanged; while *ignore all* would leave all occurrences in the document at hand unchanged, with the spellchecker not stopping at any subsequent occurrence(s) of this word. Selecting *add* would result in the word being added to the main custom dictionary (that is the top dictionary ticked off in the list of custom dictionaries). From then onwards this word will be 'known' to the spellchecker, and all future

occurrences of it, both in the current and in future documents, will be accepted as correct. (This is true as long as one does not *switch off* or *edit* this word out of the main custom dictionary — cf. Figure 8, right-hand screenshot.) The outcome of selecting *change* is that the first occurrence only would be changed to 'lemmatised', while *change all* would change all occurrences in the document to 'lemmatised'. Lastly, *autocorrect* would not only change all existing occurrences of 'lemmatized' to 'lemmatised', but would automatically change, in the current as well as all future documents, all

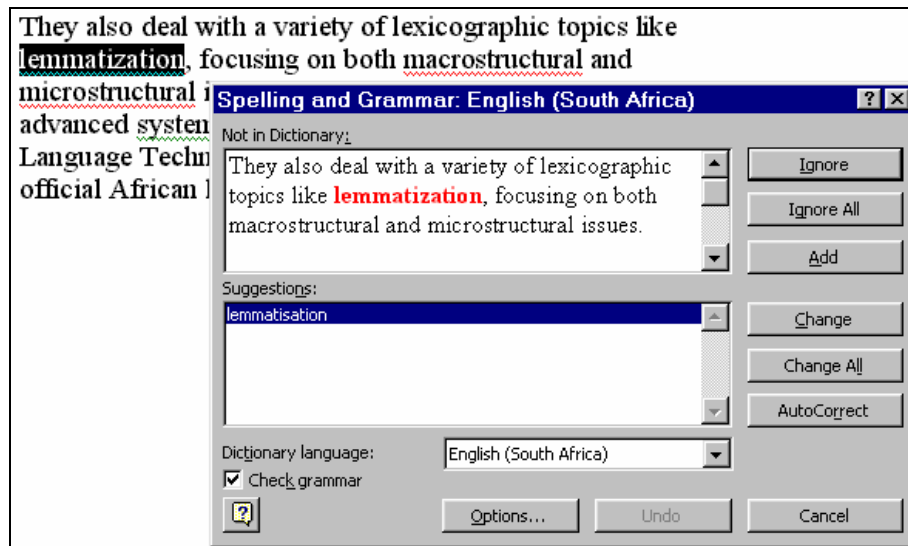


Figure 10: Standard correction screen

newly typed occurrences of 'lemmatized' to 'lemmatised' without any further notice.

The autocorrect feature can be used to automatically detect and correct typographical errors, misspelled words, grammatical errors, and incorrect capitalisation. For example, if one types 'teh' plus a space it can be immediately replaced with 'the'. Firstly, this function uses a list of built-in, so-called *autocorrect entries*. Secondly, this is of course also a very useful function to quickly insert text, graphics, or symbols in the text by simply typing the required minimum characters to trigger the desired output. For example, typing :) to obtain ☺, or i for l, or pslb instead of PanSALB, etc. (Here too, entries can be added or removed with relative ease.) Such behaviour can, unfortunately, also be counterproductive. It is believed that the autocorrect function that puts the first letter of a sentence in upper case introduces on average more mistakes in text, and especially in tables, than correcting instances where the sentence-initial letter should have been typed with a capital letter. Users do not normally know how to disengage this option.

Observe that the *add* option can of course also be used as an important tool in building and extending spellcheckers for the African languages. In writing the 200-word Sesotho sa Leboa abstract of a recent article (Nong *et al.*,

2002: 1–2), for example, the words **bangwala-pukuntšú** 'dictionary writers', **pateroneng** 'in a pattern' and **khophaseng** 'in a corpus' were not recognised by the spellchecker. These words are however all acceptable, correctly spelled Sesotho sa Leboa words. Clicking the *add* option in each case thus also instantly meant strengthening/improving the spellchecker for future non-word error detection with three new orthographic words.

In the case of grammar checking, apart from suggested corrections and improvements, grammatical *assistance* may be offered by means of pop-up windows. Compare the detailed guidance for the English phrase 'systems which' in Figure 11 in this regard.

Users can even customise the grammar checker by setting rules for grammar and writing styles. For example, a built-in style such as *casual* or *technical* may be selected, a new style can be created, or an existing style adapted.

A useful shorthand alternative to the standard correction screen is simply to right-click an underlined word, as illustrated in Figure 12.

When red underlining is right-clicked, a simplified version of the spelling correction screen appears with a number of suggested alternatives, as well as a selection of options such as to ignore all similar occurrences, to add the word to the main custom dictionary, to autocor-

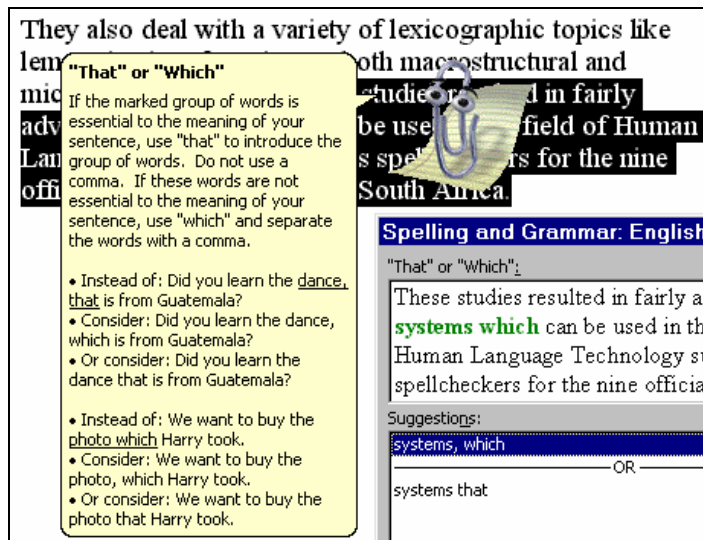


Figure 11: Grammatical pop-up window with additional information

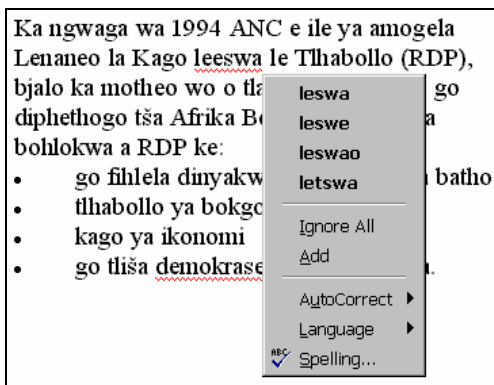


Figure 12: Shorthand correction screen (right-clicking underlined words)⁷

rect, etc. (Note that when green underlining is right-clicked, a simplified version of the grammar correction screen appears.) In Figure 12 the Sesotho sa Leboa word **leswa** 'new' was misspelled as *leeswa.⁸ In this case one may also consider the suitability of the suggested alternatives. The options **leswa**, **leswe**, **leswao** and **letswa** are quite logical suggestions reflecting cases where a single character could have been typed twice, or one or two characters mistyped. All these are very com-

mon typing errors, and one thus finds that it is possible to use built-in technology to go beyond the mere 'non-word error detection' with custom wordlists, *in casu* our African-language spellchecker lexica. As a result of the fact that the algorithms for suggesting alternatives to non-words are to a large extent language-independent, it is already possible at this point in time to perform some semi-automatic 'isolated-word error correction' for the African languages.

Not all built-in technology is useful for the African languages though. The disjunctively written African languages, in particular, require adjustments in the handling of occurrences of *sequences of identical orthographic words*. One of the typical errors made in text production in any language is the erroneous repetition of a word ('the the' is common in English, see for an authentic example the text in the bottom-right corner of Appendix 1). Therefore, a standard error-detecting function in spellcheckers is to highlight occurrences of supposedly erroneous sequences of identical orthographic words. For the disjunctively written African languages this, unfortunately, results in the highlighting of a large number of correctly typed double, triple, quadruple, etc. words. For these languages this feature is thus counterproductive because it delays the process of verification rather than contributing to it. Compare the

following random examples of concordance lines with multiple occurrences of **ba** and **le** (and **se**) culled from a 5.8-million-word Sesotho sa Leboa corpus. The words that occur twice or more in succession are highlighted as errors by the spellchecker:⁹

1. ...ba topa tša fase, baeng bao bona ba ile **ba ba** amogela ka tše pedi, **ba ba ba ba** bea fase ka a mabedi, ka gore lešago la moeng le bewa ke...
2. ...batswadi **ba ba ba ba** bea fase, dipelo tša bona di sa ngongorega. Erile mo ba...
3. ...go tšwa ka sefero a ngaya sethokgwa **se se** bego se le mokgahlo ga lapa **le le le le** le latelago. O be a tseba gabotse gore barwa ba Rre Hau o tlo ba hwetša ba...
4. ...ke yena monna yola wa mohumi le bego **le le** ka gagwe maabane. Letsogo **le le le** bonago le golofetše le, e sa **le le** gobala mohlang woo." Banna ba...

From these lines it is clear that up to four repetitions of the word **ba** and up to five repetitions of the word **le** do indeed form grammatical strings in Sesotho sa Leboa. Such strings are frequent in this language, as is evident from the corpus counts shown in Table 1 for the pairs **a a**, **ba ba**, etc.

In order for such strings not to be highlighted, one shall have to wait until such time as sophisticated grammar components will have been developed for African-language spellcheckers. The creation of such grammars will not be trivial. The four consecutive **ba**'s in the first example above, for instance, are respectively the subject concord of class 2, the auxiliary verb stem, again the subject concord of class 2, and then the

object concord of class 2; while the five consecutive **le**'s in the third example are respectively the relative pronoun of class 5, the subject concord of class 5, the copulative verb stem, and then again the relative pronoun of class 5, and the subject concord of class 5.

Evaluation of the effectiveness of our wordlist-based spellcheckers

So far we (i) expounded on our *wordlist*-based approach for the creation of South African spellcheckers, (ii) showed that no technical constraints preclude the compilation of spellcheckers for *all* South African languages, and (iii) indicated which of the standard spelling and grammar checking functionalities are already possible and useful for the South African languages in current word processing software. The proof of the pudding is in the eating, however. Reformulated: Do our spellcheckers really work? Are lists of full orthographic words really all that are needed for spellchecking the South African languages? If not, does the approach work well for some, yet less well for other languages? Here, the orthographic divide between the disjunctively written African languages on the one hand, and the conjunctively written ones on the other, suggests that there might indeed be a marked difference. Also: How does Afrikaans fit into the picture? In order to study these various aspects, a comprehensive set of interlinked tests was developed, enabling a direct comparison between the different languages involved.

Translations of the same source text, namely the Universal Declaration of Human Rights (UDHR, cf. Appendix 1), were spellchecked with our spellcheckers. As the UDHR is of a rather technical nature, spellchecking this text presents a considerable challenge. (Note that the UDHR itself was obviously not used in the compilation of the respective spellchecker lexica.) Although spellcheckers were developed and tested for all South African languages, only three sets of results will be described in detail below, i.e. for Sesotho sa Leboa, for isiZulu and for Afrikaans. The first represents the disjunctively written language group (which also includes the other Sotho languages: Sesotho and Setswana; as well as Xitsonga and Tshivenda); the second represents the conjunctively written language group (which also includes the other Nguni languages: isiXhosa,

Table 1: Typical legal occurrences of double orthographic words in Sesotho sa Leboa, with their frequency counts derived from a 5.8-million-word corpus

Double orthographic words	Frequency per million words
a a	862
ba ba	1 147
di di	17
e e	46
go go	232
le le	703
o o	54
se se	644

isiNdebele and siSwati); while the third is known to be a semi-agglutinative language with productive compound formation (like Dutch, German, etc.). For each of these three languages, the effectiveness of the cumulative build-up of spellchecker lexica will be studied in detail. It follows from the overview of the methodology above, that one expects the spellcheckers to be most effective when they are loaded with the top-frequency words of the language. Adding lower-frequency words will improve the spellcheckers still, but the power of the latter to substantially improve the performance of the spellcheckers obviously becomes smaller and smaller as the frequency of the added words decreases. In practical terms, subsequent — and cross-language comparable — ‘layers’ will be added to the spellcheckers.

For these tests we ensured that the translations of the UDHR are error-free. Actually, there were some spelling errors (which were uncovered with our spellcheckers, in combination with proofreading), and these were corrected. A perfect spellchecker is one that flags all errors, whilst leaving everything that is correct unmarked. Given that the texts are error-free, everything should thus remain unmarked. A perfect spellchecker would thus recognise or ‘recall’ all words; the (lexical) recall value would be 100%. Not all words are included in especially the smaller lexica, which means that the absent (but valid) UDHR words will be wrongly indicated as errors by the spellchecker, so the recall values will be smaller than 100%. The idea now is to see how the recall values change, and how fast, with increasing sizes of the spellchecker lexica, i.e. with an increasing number of layers. These layers have all been derived in the same way. Comparable-size corpora for Sesotho sa Leboa, isiZulu and Afrikaans of approximately five million running words (tokens) were queried and the frequency lists divided into five layers. Firstly all items occurring 10 times or more, secondly all items with a frequency from 5 to 9, thirdly those items with frequencies of 3 and 4, fourthly those with a frequency of only 2, and lastly the hapax legomena (i.e. those items occurring only once in the corpora).

The data for the first language, Sesotho sa Leboa, are looked at first. From the left section of Table 2 it can be seen that there are 5 762 549 tokens in the corpus used, but only 148 697 *dif-*

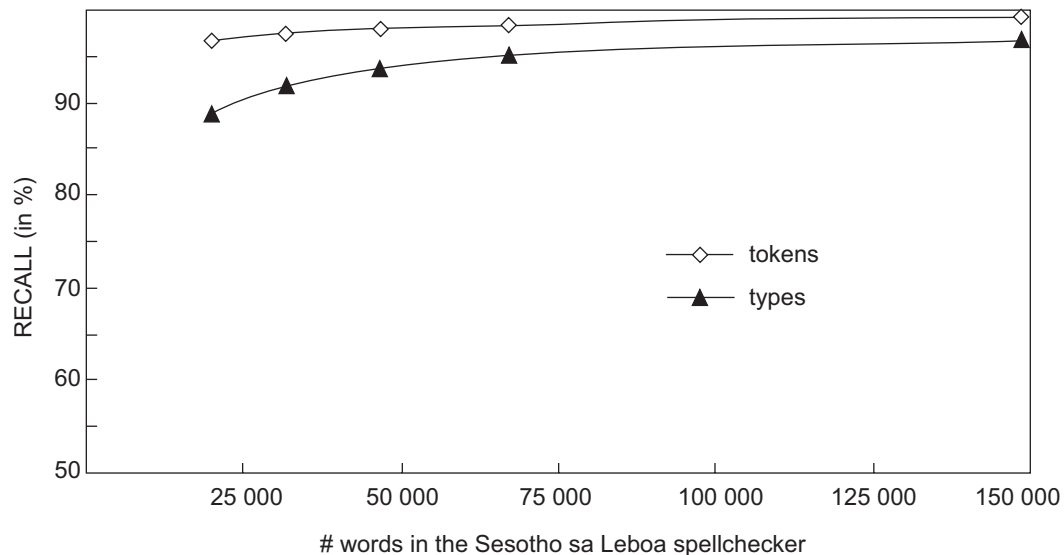
ferent orthographic words (types). The first layer has 19 823 types, which corresponds to 13.33% of all the types; the second has 12 220 types, which corresponds to 8.22% of the total; etc. The hapax value is as high as 54.73%. This means that more than half of all the types in a Sesotho sa Leboa corpus occur just once.

The right section of Table 2 shows that the UDHR in Sesotho sa Leboa consists of 2 312 tokens and 497 types. In a first test only the first corpus layer, i.e. all words with a frequency of 10 or more, of which there are roughly 20 000, was used as the spellchecker lexicon. Of the 2 312 UDHR tokens, 77 were not recognised. The ‘token recall’, with just 20 000 items in a Sesotho sa Leboa spellchecker, is thus as high as 96.67%. This is an extremely good result. One can also consider the ‘type recall’, or thus focus on the number of different words not recognised by the first layer (56) versus the number of different words in the UDHR (497). With 88.73% the type recall is not as good as the token recall. It is further possible to calculate a ‘recall value from the user’s point of view (p.v.)’, i.e. the number of types not recognised compared to the number of tokens in the UDHR. The reasoning is as follows: To a user, only the non-recognised *types* really count, no matter how often they occur in the UDHR. As seen in the description of standard spellchecker functions above, a user only needs to *add* an unrecognised type once to the main custom dictionary, after which it will be recognised by the spellchecker in all further instances. The recall value from the user’s point of view is as high as 97.58%. When the second layer is added to the first layer, the token recall gains another percent (from 96.67% to 97.53%). Again, this is a lot, even though one has actually added 50% more words to the spellchecker lexicon (from roughly 20 000 to 30 000). Adding subsequent layers to the previous ones pushes the token recall values up to 97.97%, 98.31% and finally 99.18%. In each case, the type recall values are lower, and those from the user’s point of view slightly higher. Token and type recall values for Sesotho sa Leboa are shown graphically in Figure 13.

From Figure 13 one can clearly see that token recall and type recall grow closer to one another with increasing lexicon size. This is predictable, as the addition of each spellchecker layer means the gradual accumulation of

Table 2: Building and checking the performance of a Sesotho sa Leboa spellchecker

Spellchecker for Sesotho sa Leboa (derived from 5 762 549 tokens)			Universal Declaration of Human Rights in Sesotho sa Leboa (2 312 tokens; 497 types)				
frequency	words in each layer		not recognised		recall (in %)		user's p.v.
	#	%	tokens	types	tokens	types	
10 or more	19 823	13.33	77	56	96.67	88.73	97.58
5 to 9	12 220	8.22	57	40	97.53	91.95	98.27
3 and 4	14 985	10.08	47	32	97.97	93.56	98.62
2	20 288	13.64	39	25	98.31	94.97	98.92
1 (hapaxes)	81 381	54.73	19	17	99.18	96.58	99.26
Total	148 697	100.00					

**Figure 13:** Token and type recall values for a Sesotho sa Leboa spellchecker

increasingly uncommon words, and thus by definition also words that are less frequent in the texts to be spellchecked. The difference between word form and the number of times that form occurs, or thus between type and token frequency, decreases. For very large spellchecker lexica the three types of recall (token, type and user's p.v.) thus come together.

The major finding of this first series of tests is that a Sesotho sa Leboa spellchecker with 150 000 items recognises, from the user's point of view, a stunning 99% of all words. Clearly, this is an excellent result for a first-generation spellchecker. A very different picture appears when an analogous series of tests is carried out for isiZulu, as can be seen from the data in Table 3.

Although the isiZulu corpus is somewhat smaller than the Sesotho sa Leboa one, the number of types is more than four times higher. This, of course, is a direct result of isiZulu's high degree of conjunctivism.¹⁰ This degree can be calculated, and it turns out that each isiZulu word corresponds on average with 1.60 Sesotho sa Leboa words (Prinsloo & De Schryver, 2002: 261). IsiZulu words are thus also longer, much longer, as they concatenate many formatives that are simply words in Sesotho sa Leboa. From the Sesotho sa Leboa perspective, all the possible permutations of the formatives result in a considerable increase in the number of orthographic words in isiZulu. It should thus not come as a surprise that spellchecking in isiZulu is much more complex.

Even from a user's point of view, the data in Table 3 indicate that the first layer with as many as 45 348 types recognises 'only' 72.63%. With another 42 515 types this recall climbs to 78.28%, with another 60 185 types to 81.91%, with yet another 88 911 types to 83.92%, and with all layers engaged, or thus after adding another 434 777 types, to 88.23%. Token and type recall values for isiZulu are shown in Figure 14.

A recall of 88% means that, on average, something like 40 valid isiZulu words per page will still not be recognised by our current spellchecker. Even though many users might be satisfied that around 300 isiZulu words per page are *already* recognised, it is clear that

developing techniques to increase the recall of the next-generation isiZulu spellcheckers is a high priority.

When it comes to Afrikaans one could assume, given the semi-agglutinative and productive compounding features, that the recall values would lie somewhere in-between those for Sesotho sa Leboa and isiZulu. Surprisingly, spellchecking the Afrikaans version of the UDHR shows that the Afrikaans data approach the Sesotho sa Leboa data, as is clear from Table 4.

Recall values for Afrikaans are trailing only slightly behind those for Sesotho sa Leboa, and with all spellchecker layers engaged, or thus with 284 398 types in all, the recall from the user's point of view again attains 99%. Token

Table 3: Building and checking the performance of an isiZulu spellchecker

Spellchecker for isiZulu (derived from 5 000 411 tokens)			Universal Declaration of Human Rights in isiZulu (1 045 tokens; 681 types)				
frequency	words in each layer		not recognised		recall (in %)		
	#	%	tokens	types	tokens	types	user's p.v.
10 or more	45 348	6.75	308	286	70.53	58.00	72.63
5 to 9	42 515	6.33	242	227	76.84	66.67	78.28
3 and 4	60 185	8.96	202	189	80.67	72.25	81.91
2	88 911	13.24	178	168	82.97	75.33	83.92
1 (hapaxes)	434 777	64.72	129	123	87.66	81.94	88.23
Total	671 736	100.00					

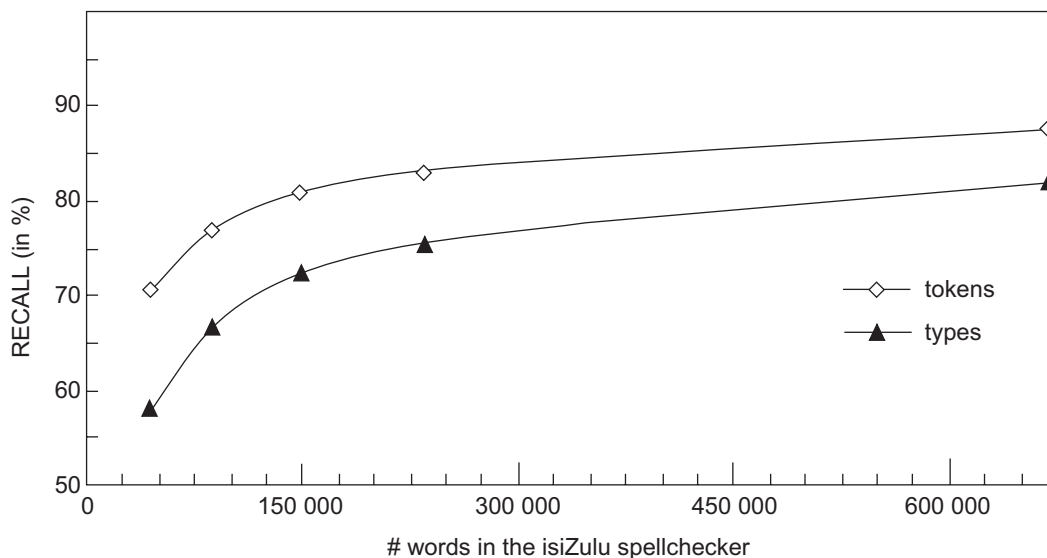


Figure 14: Token and type recall values for an isiZulu spellchecker

and type recall values for Afrikaans are shown in Figure 15.

Summarising the findings thus far, it is clear that: (i) a Sesotho sa Leboa corpus of 5 800 000 tokens, contains 150 000 types, with which a spellchecker is 99% effective; (ii) an isiZulu corpus of 5 000 000 tokens, contains 700 000 types, with which a spellchecker is 88% effective; and (iii) an Afrikaans corpus of 4 800 000 tokens, contains 300 000 types, with which a spellchecker is 99% effective.¹¹ Space-constraints unfortunately do not allow for a presentation and analysis of the tests that were done for all the other South African languages. It is nonetheless highly revealing to briefly compare the data for the three languages discussed in this article, with the data

for the four languages of the Internet study of Van der Veken and De Schryver (2003).

These scholars sampled their languages in such a way that the various regions of the African continent were covered, viz. Hausa for West Africa, Somali for East Africa, Lingala for Central Africa, and isiXhosa for southern Africa. For each of these languages they searched the Internet for four days, compiled corpora with the downloaded material, made spellcheckers, and also tested these spellcheckers on the UDHR.¹² Their results can be summarised as follows: (i) a Hausa corpus of 850 000 tokens, contains 30 000 types, with which a spellchecker is 99% effective; (ii) a Somali corpus of 300 000 tokens, contains 40 000 types, with which a spellchecker

Table 4: Building and checking the performance of an Afrikaans spellchecker

Spellchecker for Afrikaans (derived from 4 815 579 tokens)			Universal Declaration of Human Rights in Afrikaans (1 660 tokens; 468 types)				
frequency	words in each layer		not recognised		recall (in %)		
	#	%	tokens	types	tokens	types	user's p.v.
10 or more	22 654	7.97	53	43	96.81	90.81	97.41
5 to 9	16 075	5.65	32	25	98.07	94.66	98.49
3 and 4	21 269	7.48	28	22	98.31	95.30	98.67
2	30 851	10.85	25	20	98.49	95.73	98.80
1 (hapaxes)	193 549	68.06	16	13	99.04	97.22	99.22
Total	284 398	100.00					

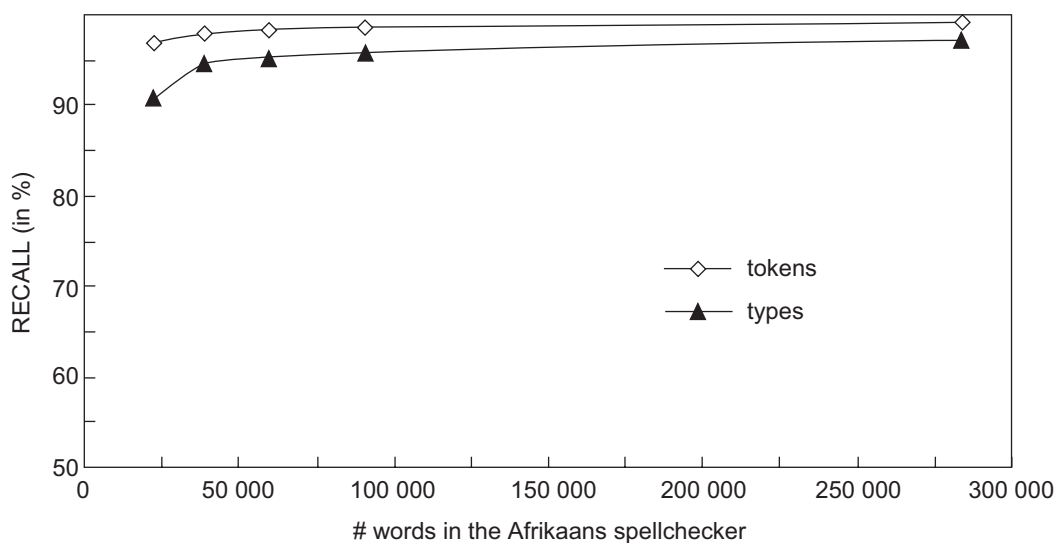


Figure 15: Token and type recall values for an Afrikaans spellchecker

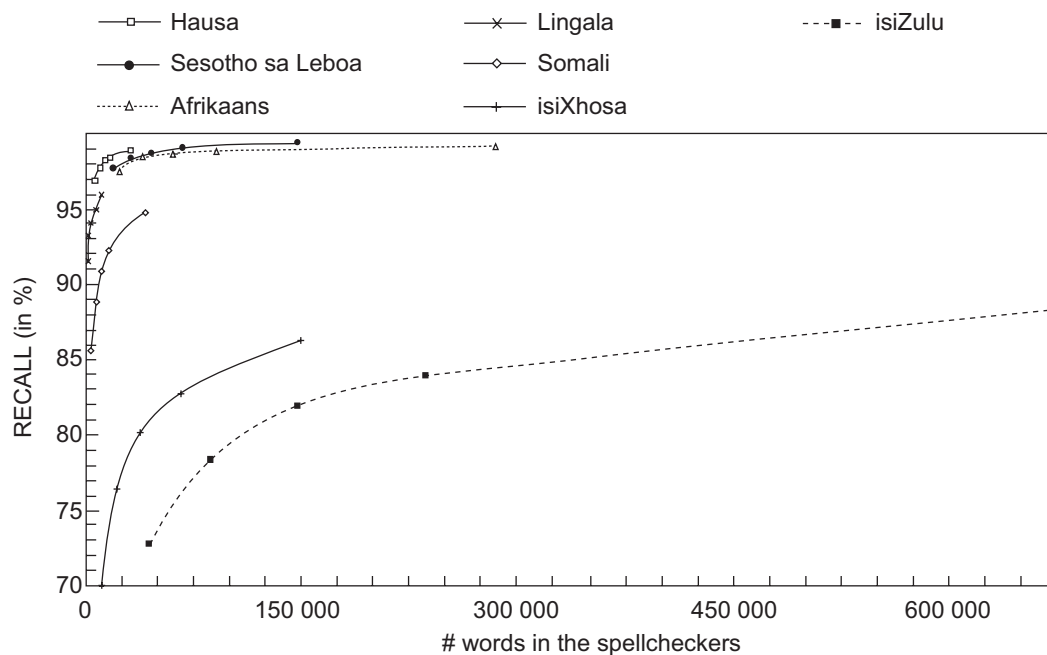


Figure 16: Comparing the effectiveness of various spellcheckers

is 95% effective; (iii) a Lingala corpus of 200 000 tokens, contains 10 000 types, with which a spellchecker is 96% effective; and (iv) an isiXhosa corpus of 950 000 tokens, contains 150 000 types, with which a spellchecker is 86% effective. Hausa and Somali are both Afro-Asiatic languages, belonging to the Chadic and the Cushitic families respectively. These languages are not related to the languages discussed in this article, but it is interesting to note that, with relatively small corpora and a very limited number of words, excellent spellcheckers can be made for these Afro-Asiatic languages. Lingala, and of course isiXhosa, do belong to the same language family as all official African languages spoken in South Africa. Lingala is written disjunctively, isiXhosa conjunctively. This orthographic difference again has direct implications for wordlist-based spellcheckers, as just 10 000 orthographic words in the Lingala lexicon pushes the recall up to 96%, while as many as 150 000 orthographic words in the isiXhosa lexicon only results in a recall of 86% — a difference of 10%. These various studies thus clearly indicate that the effectiveness of word-based

spellchecker lexica for the African languages is inversely related to the degree of conjunctivism of the orthographies of these languages.

In all tests it is also evident that the most powerful sections of the wordlists are the first few layers, or thus the top-frequent words. Especially the addition of the last layer, the hapaxes, doesn't substantially increase the recall any further. Hapaxes, by definition, just 'happen' to occur in corpora. In order to be able to single out those hapaxes that have a relatively higher occurrence likelihood, the corpus sizes must be increased even more — say, from five to ten million tokens — at which point the more 'important' hapaxes will have a higher frequency, while the accidental ones will still be genuine hapaxes. Ideally, one should reach a stage where only top-frequency wordlists are loaded as spellchecker lexica.

Conclusion

In this article spellcheckers for the South African languages were presented. We indicated what spellcheckers are, what they typically do, how they can be built, and pointed out that the technology is available to produce such

proofing tools for all South African languages, including those that have diacritics in their orthography. We characterised our top-frequency wordlist-based approach, and defended this decision. This approach was then evaluated with a series of tests applied to the spellcheckers.

The outcome of these tests can now be summarised in a graph to which the results reported by Van der Veken and De Schryver (2003) have also been added. The same text, namely the Universal Declaration of Human Rights, was spellchecked for all languages involved, and the effectiveness (here the 'recall values from the user's point of view') for all spellcheckers is compared in Figure 16.

The impact on spellchecker effectiveness of the orthographic dichotomy that exists between the disjunctively and the conjunctively written African languages is clearly visible in Figure 16. Spellchecking a disjunctively written African language such as Sesotho sa Leboa (and Lingala) is definitely feasible with a word-based approach. The recall values are located in the top-left corner. This means that few words are needed to reach a good performance, and that many words will push that performance close to 100%. Spellchecking a conjunctively written African language such as isiZulu (and isiXhosa) is much more difficult with a word-based approach, since a large number of words is required to reach a good recall. Somewhat surprisingly, a word-based approach works remarkably well for Afrikaans.

From these data one may conclude that the main focus during the creation of second-generation spellcheckers for the South African languages will have to be on the conjunctively written languages, or thus the Nguni group (isiZulu, isiXhosa, siSwati and isiNdebele). Further improving the power of the other spellcheckers (for Sesotho sa Leboa, Sesotho, Setswana, Xitsonga, Tshivenda and Afrikaans) will nonetheless still be a worthwhile venture in order to move closer to a performance of 100%, or thus spellcheckers in which all the valid words remain unflagged and in which only the non-words are detected as errors. In order to achieve this it is clear that an increase in the sizes of the corpora and, by extension, the sizes of the spellchecker lexica, would be required. More promising, especially for the Nguni group, will be to experiment with soft-

ware modules that can handle the basics of morphological decomposition.

Notes

- ¹ An earlier version of this article was presented by the authors at the *6th International Terminology in Advanced Management Applications Conference* (Prinsloo & De Schryver, 2003). Since this article has been submitted for publication in South Africa, necessary sensitivity with regard to the term 'Bantu' languages is exercised in the authors' choice rather to use the term African languages. Keep in mind, however, that the latter includes more than just the 'Bantu Language Family'.
- ² Except for the *Pharos Speller* (NB Publishers, 2003), all these spellcheckers as well as others are freely available on the Internet:
 - *PUK/Microsoft Speltoetser* (PU for CHE, 2000),
 - *Ispell vir Afrikaans* (CompuFocus, s.d.),
 - *Spell Checker for Edit Boxes* (Conradie, 2002),
 - *Afrikaanse woordelyste* (Naudé, 2000), etc.
- ³ Translated from: "Pray then, in this way: Our Father, who art in heaven, hallowed be Thy name. Thy kingdom come. Thy will be done on earth as it is in heaven. Give us this day our daily bread and forgive us our trespasses as we forgive those who trespass against us. And lead us not into temptation, but deliver us from evil." (Olivier, s.d.)
- ⁴ Translated from: "It will seek collaboration with bodies dealing with telecommunications, licensing, film and video, to achieve coordination and avoid duplication. Apart from its primary role of media support, it will commission research and make recommendations to government, the media industry and other relevant bodies. The MDDA will relate to all bodies with a direct or indirect interest in media development and diversity, amongst them the Independent Communications Authority of SA (ICASA). The MDDA will hold an Annual Review Forum where such bodies will consider the MDDA's annual report." (South Africa Government Online, 2000)
- ⁵ Translated from: "The PANSALB legislation is the most significant indicator that there is a commitment to articulate and monitor a language policy and plan broad enough to

encompass every sector of society and it is this, which places us ahead of other countries. We have the added advantage of being able to learn from the paths chosen elsewhere on this continent. There are some basic steps, which need to be followed in articulating and implementing a feasible language policy and plan." (Pan South African Language Board, 1998)

- ⁶ Translated from: "Understands concepts and some vocabulary relating to:
- identity (e.g. 'My name is...');
 - number (e.g. one, two);
 - size (e.g. big, small);
 - colour (e.g. red, yellow)." (Department of Education, 2002)
- ⁷ Translated from: "In 1994 the ANC adopted the Reconstruction and Development Programme (RDP) as the basic policy framework guiding the ANC in the transformation of South Africa. The key programmes of the RDP are:
- meeting basic needs
 - developing our human resources
 - building the economy
 - democratising the state and society." (ANC, s.d.)
- ⁸ This word was spelled correctly in the original text and the error only serves as an illustration here.
- ⁹ Translations (thanks are due to MJ Mojalefa):
1. ... they are very poor, those guests; they

then welcomed them with open arms and hosted them well because they have the responsibility of welcoming (them) ...

2. ... the parents welcomed them, and they (their hearts) did not complain (at all) when they ...

3. ... coming out of the gate he took a short-cut route which was between the woods and the courtyard. He knew very well that he would find father Hau's sons ...

4. ... it is that rich man at whose place you were yesterday. This arm that you can see is injured, was injured on that occasion." These men ...

¹⁰ See for the term 'conjunctivism', as opposed to 'disjunctivism', for instance Louwrens (1991: 1–12) or Van Wyk (1995: 83–84).

¹¹ Note that less formal tests on a few other parallel texts (of various types) indicate only marginal variation in these recall percentages.

¹² The fact that one of the authors was co-researcher in both the Van der Veken and De Schryver (2003) study and the current one explains why the methodology of the two studies is analogous. Choosing to spellcheck translations of the same document, the UDHR, was also a deliberate move, as this allows a direct comparison of the results, and enables one to place the South African data in a wider African context.

References

- ANC.** s.d. What is the African National Congress? Available at: <http://www.anc.org.za/about/anc.html> [Accessed 24 August 2002].
- CompuFocus.** s.d. Ispell vir Afrikaans. Available at: <http://www.compufocus.co.za/afrik/ispell/> [Accessed 26 April 2003].
- Conradie P.** 2002. Spell checker for edit boxes. Available at: <http://www.afrikaner.co.za/pietwerf/speltest.htm> [Accessed 26 April 2003].
- Department of Education.** 2002. Revised National Curriculum Statement Grades R–9 (Schools) — First Additional Language. Available at: http://education.pwv.gov.za/DoE_Sites/Curriculum/Final%20curriculum/policy/policy.htm [Accessed 24 August 2002].
- De Schryver G-M & Prinsloo DJ.** 2000. The compilation of electronic corpora, with special reference to the African languages. *Southern African Linguistics and Applied Language Studies* 18(1–4): 89–106.
- Gaultney V.** 2002. Gentium typeface. Available at: <http://www.sil.org/~gaultney/gentium/download.html> [Accessed 26 April 2003].
- Hurskainen A.** 1999. SALAMA: Swahili language manager. *Nordic Journal of African Studies* 8(2): 139–157.
- Lingsoft.** 1999. Orthografix 2 for Swahili. Available at: <http://www.lingsoft.fi/news/1999/o2-swahili.html> [Accessed 26 April 2003].
- Louwrens LJ.** 1991. *Aspects of Northern Sotho Grammar*. Pretoria: Via Afrika Limited.
- Naudé FJ.** 2000. Afrikaanse woordelyste. Available at: <http://www.naude.co.za/frank/afr/index.htm#woorde> [Accessed 26 April

- 2003].
- NB Publishers.** 2003. Pharos Speller: Speltoetser en Woordafbreker vir Afrikaans. Available at: http://www.nb.co.za/Pharos/phArticleDisplay.asp?iCategory_id=31 [Accessed 26 April 2003].
- Nong S, De Schryver G-M & Prinsloo DJ.** 2002. Loan Words versus Indigenous Words in Northern Sotho — A lexicographic perspective. *Lexikos* 12 (AFRILEX-reeks/series 12: 2002): 1–20.
- Office of the High Commissioner for Human Rights.** 1948–. Universal Declaration of Human Rights. Available at: <http://193.194.138.190/udhr/index.htm> [Accessed 26 April 2003].
- Olivier J.** s.d. Lord's Prayer in Tshivenda. Available at: <http://www.cyberserv.co.za/users/~jako/lang/ventexts.htm> [Accessed 26 April 2003].
- Olivier J.** 2002. South African special characters font. Available at: <http://www.cyberserv.co.za/users/~jako/lang/fonts.htm> [Accessed 26 April 2003].
- Pan South African Language Board.** 1998. PanSALB's position on the promotion of multilingualism in South Africa. Available at: <http://www.pansalb.org.za/pub.htm> [Accessed 24 August 2002].
- Prinsloo DJ.** 1991. Towards computer-assisted word frequency studies in Northern Sotho. *South African Journal of African Languages* 11(2): 54–60.
- Prinsloo DJ & De Schryver G-M.** 2001. Corpus applications for the African languages, with special reference to research, teaching, learning and software. *Southern African Linguistics and Applied Language Studies* 19(1–2): 111–131.
- Prinsloo DJ & De Schryver G-M.** 2002. Towards an 11 x 11 array for the degree of conjunctivism/disjunctivism of the South African languages. *Nordic Journal of African Studies* 11(2): 249–265.
- Prinsloo DJ & De Schryver G-M.** 2003. Towards second-generation spellcheckers for the South African languages. In: De Schryver G-M (ed) *TAMA 2003 South Africa: Conference Proceedings*. Pretoria: (SF)² Press. pp. 135–141.
- PU for CHE (Potchefstroom University for Christian Higher Education).** 2000. PUK/Microsoft Speltoetser. Available at: <http://www.puk.ac.za/spel/en/index.html> [Accessed 26 April 2003].
- South Africa Government Online.** 2000. Media Development and Diversity Agency Position Paper. Available at: <http://www.gov.za/documents/2000/mdda/> [Accessed 24 August 2002].
- Unicode.** 2003. The Unicode® Standard: A technical introduction. Available at: <http://www.unicode.org/standard/principles.html> [Accessed 26 April 2003].
- Van der Veken A & De Schryver G-M.** 2003. Les langues africaines sur la Toile. Étude des cas haoussa, somali, lingala et isixhosa. *Cahiers du Rifal* 23 (Thème: Le traitement informatique des langues africaines): 33–45.
- Van Huyssteen GB & Van Zaanen MM.** 2003. A Spellchecker for Afrikaans, Based on Morphological Analysis. In: De Schryver G-M (ed) *TAMA 2003 South Africa: Conference Proceedings*. Pretoria: (SF)² Press. pp. 189–194.
- Van Wyk EB.** 1995. Linguistic assumptions and lexicographical traditions in the African languages. *Lexikos* 5 (AFRILEX-reeks/series 5B: 1995): 82–96.

