Stellenbosch Papers in Linguistics Plus, Vol. 64, 2022, 165-187
doi: 10.5842/64-1-867

Optical Character Recognition and text cleaning in the indigenous South African languages

Danie J. Prinsloo

Department of African Languages, University of Pretoria, South Africa
E-mail: danie.prinsloo@up.ac.za

Elsabé Taljard

Department of African Languages, University of Pretoria, South Africa
E-mail: elsabe.taljard@up.ac.za

Michelle Goosen

Department of African Languages, University of Pretoria, South Africa
E-mail: michelle.goosen@up.ac.za

Abstract
This article represents follow-up work on unpublished presentations by the authors of text and corpus cleaning strategies for the African languages. In this article we provide a comparative description of cleaning of web-sourced and text-sourced material to be used for the compilation of corpora with specific attention to cleaning of text-based material, since this is particularly relevant for the indigenous South African languages. For the purposes of this study, we use the term "web-sourced material" to refer to digital data sourced from the internet, whereas "text-based material" refers to hard copy textual material. We identify the different types of errors found in such texts, looking specifically at typical scanning errors in these languages, followed by an evaluation of three commercially available Optical Character Recognition (OCR) tools. We argue that the cleanness of texts is a matter of granularity, depending on the envisaged application of the corpus comprised by the texts. Text corpora which are to be utilized for e.g. lexicographic purposes can tolerate a higher level of 'noise' than those used for the compilation of e.g. spelling and grammar checkers. We conclude with some suggestions for text cleaning for the indigenous languages of South Africa.

Keywords: text cleaning; Optical Character Recognition (OCR) tools; 'noise' in text-based corpora; scanning errors; text-sourced corpora; granularity of cleanness

1. Introduction

Having access to electronic language corpora is of the utmost importance for the indigenous languages of South Africa, since such corpora form the bases for different kinds of human language technologies, which are needed to enhance the status of these languages as languages of higher functions. These languages are, however, generally regarded as lesser-resourced with regard to electronic language resources. Corpus compilation consists *inter alia* of the collection of textual material, rendering it in machine readable format and combining such material in a structured collection of texts, i.e., a corpus. The texts harvested for corpus compilation are rarely without errors; they may contain errors in punctuation, capitalisation, spelling, grammar, and format, to name but a few. Combining such texts into a corpus results in the corpus being “dirty”, “raw” or “noisy”. Taken at face value the issue of ‘clean’ versus ‘dirty/raw’ and ‘noisy’ corpora seems to be quite simplistic — if the texts comprising the corpus are dirty, *clean them*, and save the clean copy for use by corpus query programs. In reality, however, cleaning a text corpus is much more problematic than meets the eye. The notions ‘dirty corpus’ and ‘clean corpus’ are fluid, and are influenced by a number of factors such as the intended use of the corpus. In fact, *clean corpus* is a relative concept in terms of e.g. granularity, that is, how clean the corpus should be for different purposes and which version(s) should be saved for querying and preservation purposes. A corpus may be deemed ‘clean’ when its comprising texts exactly match the original documents, or, alternatively, when it also entails (a) corrections of mistakes made by the author of the original text, (b) indication of circumflexes, diacritics and tonal patterns, (c) the removal of foreign words and paragraphs, and (d) omission of data irrelevant to the intended use of the corpus. For the purposes of this article, the terms *text cleaning* and *corpus cleaning* are used interchangeably since corpora by definition consist of electronic texts.

The corpus compiler should decide whether time and programming-sophistication should be invested in text cleaning software or whether these resources should rather be directed towards improved handling of dirty texts by corpus query programs. The corpus compiler should also consider to what extent “noise” present in the corpus would negatively impact the intended use of the corpus.

“Noise” in text is defined by Knoblock, Lopresti, Shourya and Subramaniam (2007) as “any kind of difference between the surface form of a coded representation of the text and the *intended*, *correct* or *original* text”, which implies correcting the errors made in the earlier stages of text processing. From a lexicographic perspective, the definition of noise can be expanded to also include text that might be clean but irrelevant for lexicographic purposes, such as foreign words and texts (cf. Gabrialetos 2007:6) as well as different levels of text duplication in the corpus. The aim of the present article is to provide a critical overview on corpus cleaning activities and procedures for different corpus applications such as spelling and grammar checkers, text verification, part of speech (POS) mark-up, fequency and alphabetical lists, keywords-in-context, etc. We also provide results of an experiment in which different OCR scanning tools are evaluated, focussing on text-based material. It is expected that the results presented in this article will be generalizable to other African languages, provided that languages with similar orthographical systems, i.e. disjunctive and conjunctive systems, are compared. Since the use of diacritics poses one of the challenges for the scanning of textual material, solutions and results presented here will also apply to other African languages which make use of similar diacritic signs.

2. Cleaning text material

Research on text cleaning is mostly carried out within the disciplines of Computational Linguistics and Quantitative Linguistics, with no clear indication of the value of these systems for applied disciplines such as lexicography and terminology. From the literature, it is clear that cleaning strategies vary according to the nature of the source of the material (cf. Graën, Batinic and Volk 2014:224). Secondly, the application envisaged for a specific text collection has a direct impact on the selection of cleaning methods and the level of cleaning. Bosch and Pretorius (2011), for example, refer to clean-up in the context of a practical, semi-automated procedure towards creating a clean, morphologically annotated isiZulu corpus of tractable size. Such a corpus could eventually serve both as a gold standard for isiZulu computational morphology and as a basis for further linguistic annotation. With this aim in mind, Bosch and Pretorius (2011:144) define cleaning as “the identification and appropriate processing of non-words, or equivalently, non-attested words in the Zulu language”. To this end, one needs to decide what kind of systematic clean-up is necessary. Généreux, Hendrickx and Mendes (2012) discuss the cleaning of a Portuguese corpus to be used for linguistic enquiries, whereas Kuhn Dekker, Šandrih, Zviel-Gershin, Holdt and Schoonheim (2019) report on the use of crowdsourced text cleaning for pedagogical purposes. The aim of their cleaning effort is to rid the corpus containing the texts of inappropriate and/or offensive language in order to make it appropriate for the development of language learning material. It is therefore clear that text cleaning is a matter of granularity, an issue that is discussed in section 5 below.

Textual material that are web-sourced present a different kind of noise than hard-copy texts and therefore require different cleaning strategies. Perusal of the relevant literature reveals quite extensive research on the cleaning of web-sourced material (cf. Baroni and Kilgariff 2006; Hofmann and Weerkamp 2007; Baroni Francis, Kilgarriff and Sharoff 2008; Evert 2008; Kohlschütter, Frankhauser and Nedjl 2010; Kantner, Kutter, Hildebrandt and Püttcher 2011 and Graën, Batinic and Volk 2014) but surprisingly little on cleaning of text-sourced material. In section 2.1, we give an overview on cleaning of web-sourced material, followed in section 2.2 by a discussion on strategies for the cleaning of OCR scanned text-sourced material.

2.1 Cleaning of web-sourced material

Evert (2008: 3489) refers to the web as “an amazing, almost inexhaustible and very convenient source of authentic natural language data”. However, web pages are “messier than other text sources, though, and interesting linguistic regularities may easily be lost among the countless duplicates, index and directory pages, Web spam, open or disguised advertising, and boilerplate” (Evert, op cit.). Goldhahn, Eckart and Quasthoff (2012) identify and describe five processes in their cleaning efforts of web-sourced texts. These are HTML-stripping, language identification, sentence segmentation, cleaning and sentence scrambling. In the cleaning phase, they identify non-sentences based on patterns that a normal sentence should follow, and all strings that do not comply with these patterns are removed. They also eliminate sentences that do not belong to the considered language. It is crucial to clean web-sourced texts if reliable linguistic and frequency data are to be obtained from corpora.

Web-sourced corpora usually consist of web pages, i.e. documents that are marked-up using HTML. Web pages contain:

- navigational structures, e.g. menu’s;
- headers such as logos and breadcrumbs;
- footers consisting of copyright notices and dates; and
- advertisements.

When looking at a web page, it is easy for humans to distinguish between the navigational text, advertisements and related articles on the one hand, and the actual content on the other. It is important that web-sourced texts be cleaned from non-content segments, also known as “boilerplate texts”, since these segments constitute noise and make further processing difficult, (cf. Hofmann and Weerkamp 2007). In the words of Kantner et al. (2011:5),

> Large digital text samples are promising sources for text-analytical research in the social sciences. However, they may turn out to be very troublesome when not cleaned of the 'noise' of doublets and sampling errors that induce biases and distort the reliability of content-analytical results.

Evert (2008) rightfully points out that page duplicates and boilerplate repetition may grossly inflate frequency counts for certain terms and expressions such as *click here, contents,* and *Vi@gr@* – a real concern, for example, for the lexicographer who works corpus-based. In the Media24 archive, typical boilerplate occurrences include the name of the newspaper, repetitious instructions, headings, etc. It stands to reason that cleaning of corpora needs to be maximally automatized, since the manual cleaning of especially large web-sourced corpora is not feasible – neither timewise nor in terms of human resources.

In response to the challenges of cleaning of web-based corpora, an open competition, Cleaneval, was launched in 2007 by a team comprised of four researchers, cf. Baroni et al. (2008). The challenge of this evaluative competition was the preparation of web data for use as a corpus for linguistic and language technology research and development. Participants were tasked with designing and implementing methods for the cleaning of arbitrary web pages. They were required to run their respective systems on a document set and their output was evaluated against manually cleaned documents. One aspect of the competition was the detection and stripping of boilerplate texts. The first exercise took place in 2007, under the auspices of the Association for Computational Linguistics’ (ACL) Special Interest Group on Web as Corpus. Two languages were addressed, i.e. English and Chinese. Schäfer (2016) indicates that most of the proposed solutions made use of machine-learning techniques. For a discussion of the participating systems and results, the reader is referred to Fairon, Naets, Kilgarriff and De Schryver (2007).

In follow-up work, Kohlschütter et al. (2010) point out that although a number of approaches have been introduced to automatize the detection and stripping of boilerplate texts, a systematic analysis of which features are the most salient for boilerplate content is lacking. Identifying these features would enable automatic detection and stripping of boilerplate texts. They indicate that textual content on the Web can be grouped into two classes, i.e. “long text”, which is most likely the actual content, and “short text”, which is most likely navigational boilerplate text. The results of their experiments indicate that removing words belonging to the category of short text alone is already a good strategy for cleaning boilerplate content. Up to date, no effort has been made to test the usability of the software utilized for these experiments for the African languages. NCLEANER, a corpus cleaning tool is referred to by a number of researchers, cf. Evert (2008). It is a simple tool for automatic boilerplate removal, using character-level N-gram models as classifiers. As pointed out by Généreux et al. (2012), prior to their work in Portugese, it had not yet been evaluated for languages other than English.

Although the proposed solutions may be useful for large web-sourced material, it has limited applicability, especially for the African languages due to their relatively low visibility on the internet. Web-based material usually form only a small segment of African language corpora. Furthermore, users of African language corpora, such as lexicographers and terminologists, rarely have the necessary computational skills and knowledge to (a) evaluate and (b) apply these procedures to the African languages. Utilizing these strategies for text cleaning requires a high level of computational knowledge and skills – resources which are often in short supply when it comes to the African languages.

2.2 OCR scanning and cleaning of text-based material

It would seem that cleaning hard copy texts requires less specialized computational knowledge than cleaning web-sourced material. We argue below that (a) the type of noise in text-based material is different from that in web-sourced material, and (b) text-based material can be cleaned by utilizing little more than basic computational skills.

Different types of errors occur in the corpus material for Afrikaans and African languages. The three major types of errors are (a) text duplication, (b) orthographical, spelling and word division errors in the original texts, and (c) basic scanning errors. Each of these is discussed in the paragraphs below, with specific attention paid to scanning errors, which seem to have the biggest influence on the quality of the eventual corpus.

2.2.1 Duplication

Many instances of text duplication occur, varying from (a) basic duplication errors, e.g. the same text added to the corpus more than once, to (b) instances of text repeated in sister newspapers or on different dates in a particular newspaper – especially in the case of media corpora – and (c) boilerplate repetitions, specifically regarding web-sourced data (cf. Baroni and Kilgarriff 2006). Boilerplate repetitions pose a bigger challenge for Afrikaans than for the African languages, whereas text repetition is an issue for an Afrikaans corpus if the corpus contains media-based components. So, for example, almost a third of the articles published in the newspaper *Beeld* on a specific day was also published in *Die Burger*. A small experiment carried out on 18 April 2020 at 05:00 indicated that sixteen out of the 49 reports in *Die Burger* were also published in *Beeld* This simply means that word frequency counts will be heavily inflated and seriously skewed as a result of the duplication.

2.2.2 Spelling and orthographical errors

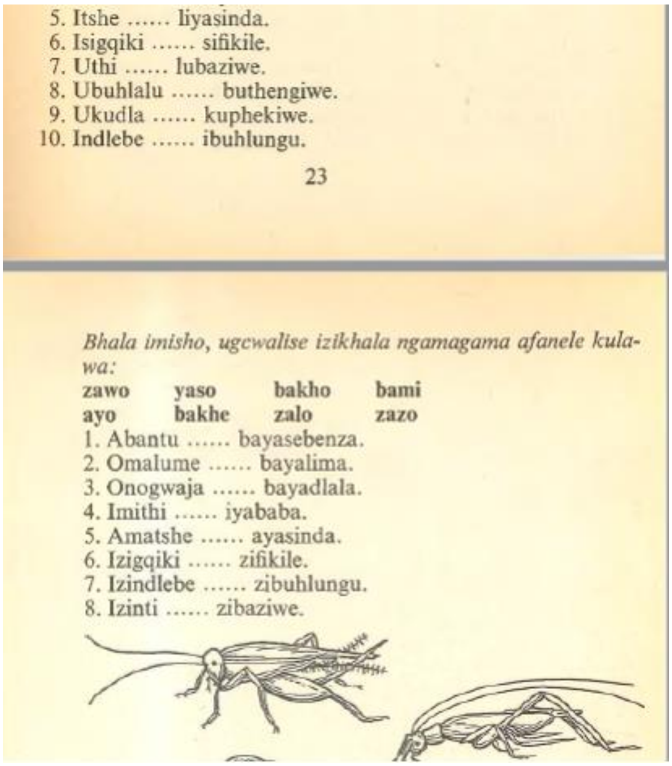
A second problematic dimension of text cleaning for African languages is incorrect spelling, old spelling, word division problems, grammatical errors and incorrect capitalization in the texts used for corpus compilation. Krstev and Stanković (2019:63) refer to these types of errors as cognitive errors, caused by a user's (mis)understanding of the relevant orthographical rules of a particular language. These types of errors can produce either valid (but not intended) or invalid words. For the African languages especially, there is an additional dimension to the orthographical issue, namely that the orthographies for these languages have only relatively recently been standardized. The latest guides containing the standardized spelling and orthographical rules for the African languages were only published in 2008 (PanSALB, 2008a-h). Texts published prior to this date will therefore contain words which are – in terms of the latest standardization principles – incorrectly spelled, incorrectly divided (one word instead of two, and vice versa) and incorrectly capitalized. To give a few examples: prior to the 1988 version of the Sepedi *Terminology and Orthography No. 4* (*T and O*) (Department of Education and Training), *bjalo ka* 'as, like' was written as one word, i.e. *bjaloka*; *gonabjale* 'immediately' was written as two words, i.e. *gona bjale*. In older isiZulu texts, the demonstrative was written conjunctively to the noun, e.g. *lelikhaya* 'this home/homestead', but in the *isiZulu Terminology and Orthography No. 4* (Department of Education and Training 1993: xii) this was officially changed to being written disjunctively, i.e. *leli khaya* 'this home/homestead' (Bosch 2020:15). Standardization, furthermore, does not only refer to spelling and word division, but also pertains to grammar. In Sepedi for example, the use of the so-called shortened demonstrative i.e. (a) *mosadi o* 'this woman' instead of *mosadi yo*, (b) *monwana o* 'this finger' instead of *monwana wo* and (c) *ngaka e* 'this doctor' instead of *ngaka ye* are prevalent in older texts, but were declared to be unacceptable in the *T and O* of 1988. Cases of incorrect capitalization e.g. *SePedi* instead of *Sepedi* and *SeZulu* instead of *Sezulu* also occur. These incorrect or non-standard forms pose a potential problem for cleaning of potential corpus material. The fact that these forms appear in texts, implies that dictionary users may come across these forms, and may therefore look them up in a dictionary. Should these errors be corrected in the texts making up a corpus which is to be used for lexicographic purposes, they will not come to the attention of the lexicographer, and will therefore not be treated in a dictionary. Having access to the metadata, e.g. date of publication of the texts contained in the corpus, is necessary to alert users to the fact that forms which are regarded as non-standard or incorrect in terms of current spelling and/or orthographic rules, were indeed correct in terms of the rules as they were during the time of publication of the source text.

2.2.3 Scanning errors

A major factor that corpus compilers of the African languages have to contend with is basic scanning errors, as the majority of texts are scanned from paper-based sources by means of Optical Character Recognition (OCR) scanning. As Krstev and Stanković (2019: 63) point out, a text that fully corresponds to the original is rarely obtained since OCR is prone to errors. They identify a number of factors that have an impact on the quality of the OCRed text, i.e. the software used, the quality of the paper and print quality of the original text, and the alphabet of the language. OCR scanning of old books poses yet another challenge – older fonts do not OCR well, there may be deterioration of the paper and in the case of books taken from a library, handwritten notes and comments may also be present. These challenges all pertain to Afrikaans and the African languages. The discussion that follows reports on a text cleaning effort as part of a bigger digitization project. In section 3 we give a general overview of typical scanning errors encountered to give a sense of the extent of scanning errors. This discussion is followed by a formal evaluation of three OCR scanning packages, i.e. *ABBYY*, *Omnipage* and *CTexTools*.

3. Overview of typical scanning errors

The original Afrikaans and African language texts were scanned using the software *ABBYY Finereader 14* (hereafter referred to as *ABBYY*), and different versions of *Omnipage*. Once a text had been scanned and the OCR process completed, the text was saved in UTF-8 format. The .txt file was then converted to Word format and opened in Unicode (UTF-8). The spellchecker for a given language was then activated and the text was cleaned to resemble the original source document. Once the Word document had been cleaned, it was then again saved in UTF-8 format. When using *ABBYY*, the scanned text cannot be directly saved in Word format as the software attempts to replicate the orginal scanned document. One would also encounter scanning errors which would most likely not appear in the .txt file. Refer in this regard to Figure 1A-C.



5. Itshe ...... liyasinda.
6. Isigqiki ...... sifikile.
7. Uthi ...... lubaziwe.
8. Ubuhlalu ...... buthengiwe.
9. Ukudla ...... kuphekiwe.
10. Indlebe ...... ibuhlungu.

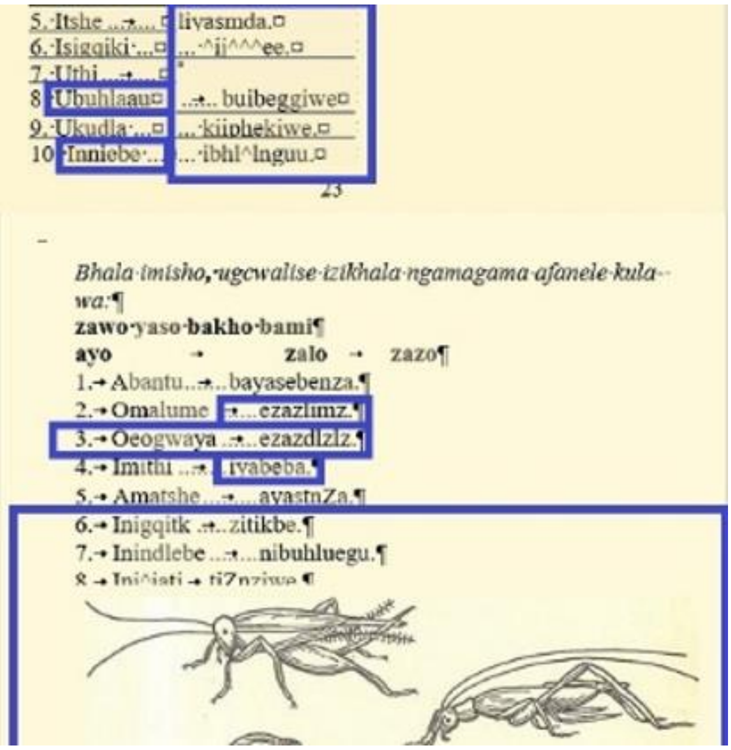
23

Bhala imisho, ugcwalise izikhala ngamagama afanele kulawa:

zawo	**yaso**	**bakho**	**bami**
ayo	**bakhe**	**zalo**	**zazo**

1. Abantu ...... bayasebenza.
2. Omalume ...... bayalima.
3. Onogwaja ...... bayadlala.
4. Imithi ...... iyababa.
5. Amatshe ...... ayasinda.
6. Izigqiki ...... zifikile.
7. Izindlebe ...... zibuhlungu.
8. Izinti ...... zibaziwe.

Figure 1A. An image from *Indlela yolimi lwesiZulu ibanga lesi-2*

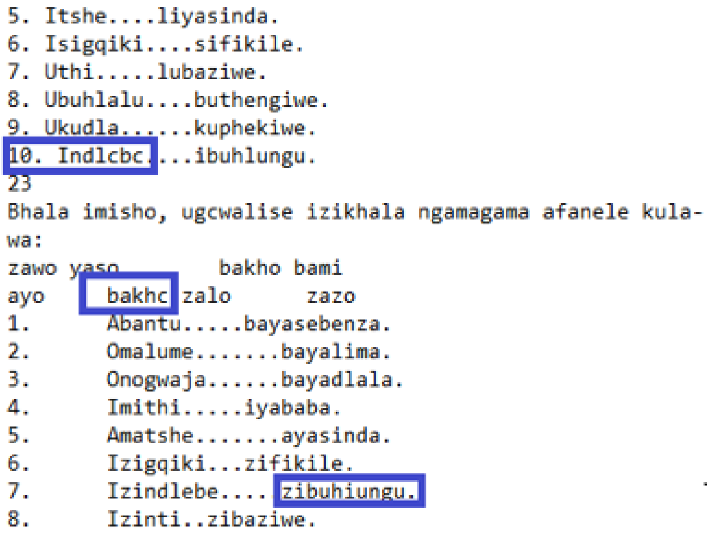


5.·Itshe ...→...¤ liyasmda.¤
6.·Isiggiki·...¤ ...·^ji^^^ee.¤
7.·Uthi ...→...¤
8·Ubuhlaau¤ ..→.. buibeggiwe¤
9.·Ukudla·...¤ ...·kiiphekiwe.¤
10·Inniebe·... ...·ibhl^lnguu.¤

23

Bhala·imisho,·ugcwalise·izikhala·ngamagama·afanele·kula-·wa:¶
zawo·yaso·bakho·bami¶
ayo → zalo → zazo¶
1.→Abantu...→...bayasebenza.¶
2.→Omalume →...ezazlimz.¶
3.→Oeogwaya ..→...ezazdlzlz.¶
4.→Imithi ...→ ivabeba.¶
5.→Amatshe ...→...ayastnZa.¶
6.→Inigqitk ..→..zitikbe.¶
7.→Inindlebe ...→...nibuhluegu.¶
8.→Ini^iati→ tiZnziwe.¶

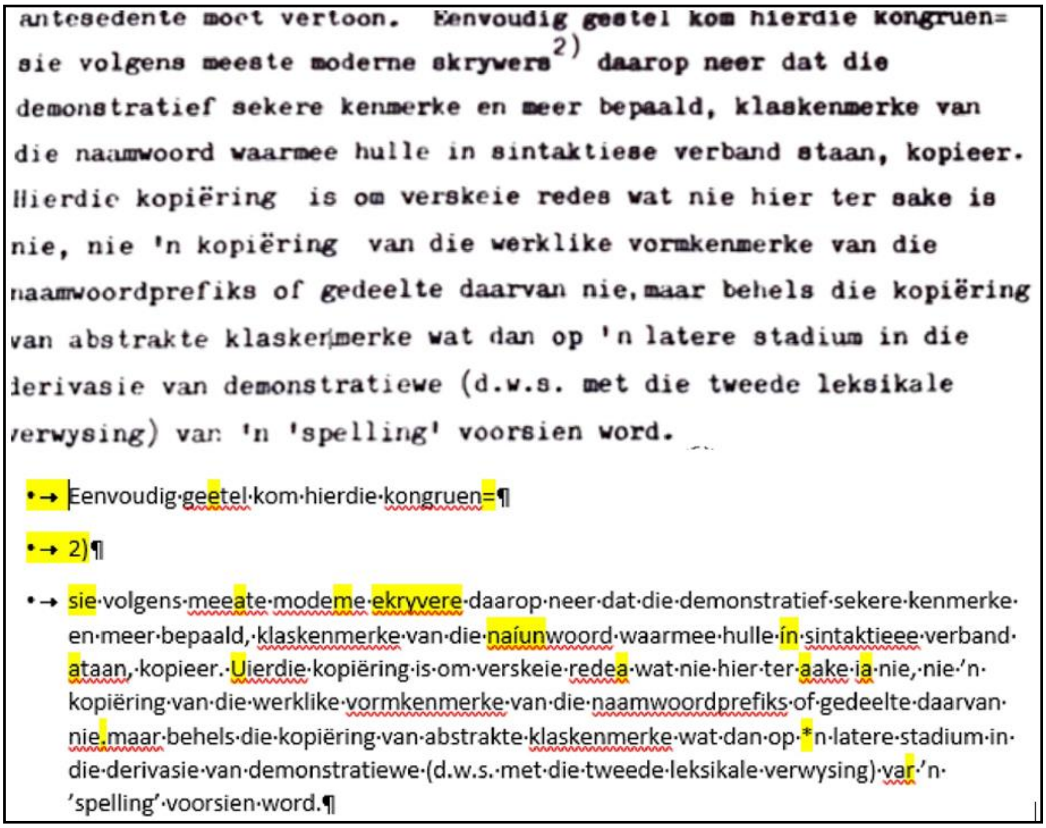
Figure 1B. An image from *Indlela yolimi lwesiZulu ibanga lesi-2*



```
5. Itshe....liyasinda.
6. Isigqiki....sifikile.
7. Uthi.....lubaziwe.
8. Ubuhlalu....buthengiwe.
9. Ukudla......kuphekiwe.
10. Indlcbc....ibuhlungu.
23
Bhala imisho, ugcwalise izikhala ngamagama afanele kula-
wa:
zawo yaso         bakho bami
ayo     bakhc zalo       zazo
1.       Abantu.....bayasebenza.
2.       Omalume.......bayalima.
3.       Onogwaja......bayadlala.
4.       Imithi.....iyababa.
5.       Amatshe.......ayasinda.
6.       Izigqiki...zifikile.
7.       Izindlebe.... zibuhiungu.
8.       Izinti..zibaziwe.
```

Figure 1C. An image from *Indlela yolimi lwesiZulu ibanga lesi-2*

The image in Figure 1A is the original scanned version, i.e. exactly as it appears in the source, the image in Figure 1B was saved directly in Word format, whereas the one in Figure 1C was saved in UTF-8 format. When a scanned text is saved directly in Word format, one encounters issues such as words being cut off, misspelling of words, unwanted characters such as ^^ and images which are not relevant for e.g. lexicographic or terminological use. Scanning errors are the result of the failure of the software to recognise spaces between words, or to distinguish between characters with and without diacritics, and between characters which appear similar to the software. Krstev and Stanković (2019:63) point out that OCR errors tend to be repeated in one text. Typical characters that tend to be misinterpreted by the software are e.g. *k* scanned as *lc*, *e* as *c*, *I* as *1*, *tl* (*t* + the letter *l*) as *t1* (*t* + the number *1*), *e* as *a*, *y* as *v* and *š* as *s* or *s^* or vice versa in each case. Certain font types are more problematic than others, e.g. Times New Roman where the the letter l and the number 1 can only be distinguished with difficulty, even by the human eye. Consider Figure 2 as a typical example of poor scanning quality:



antesedente moct vertoon. Eenvoudig gestel kom hierdie kongruen=
sie volgens meeste moderne skrywers 2) daarop neer dat die
demonstratief sekere kenmerke en meer bepaald, klaskenmerke van
die naamwoord waarmee hulle in sintaktiese verband staan, kopieer.
Hierdic kopiëring is om verskeie redes wat nie hier ter sake is
nie, nie 'n kopiëring van die werklike vormkenmerke van die
naamwoordprefiks of gedeelte daarvan nie, maar behels die kopiëring
van abstrakte klaskenmerke wat dan op 'n latere stadium in die
lerivasie van demonstratiewe (d.w.s. met die tweede leksikale
verwysing) van 'n 'spelling' voorsien word.

•→ Eenvoudig·geetel·kom·hierdie·kongruen=¶

•→ 2)¶

•→ sie·volgens·meeate·modeme·ekryvere·daarop·neer·dat·die·demonstratief·sekere·kenmerke·en·meer·bepaald,·klaskenmerke·van·die·naíunwoord·waarmee·hulle·ín·sintaktieee·verband·ataan,·kopieer.·Uierdie·kopiëring·is·om·verskeie·redea·wat·nie·hier·ter·aake·ia·nie,·nie·’n·kopiëring·van·die·werklike·vormkenmerke·van·die·naamwoordprefiks·of·gedeelte·daarvan·nie.maar·behels·die·kopiëring·van·abstrakte·klaskenmerke·wat·dan·op·*n·latere·stadium·in·die·derivasie·van·demonstratiewe·(d.w.s.·met·die·tweede·leksikale·verwysing)·var·’n·’spelling’·voorsien·word.¶

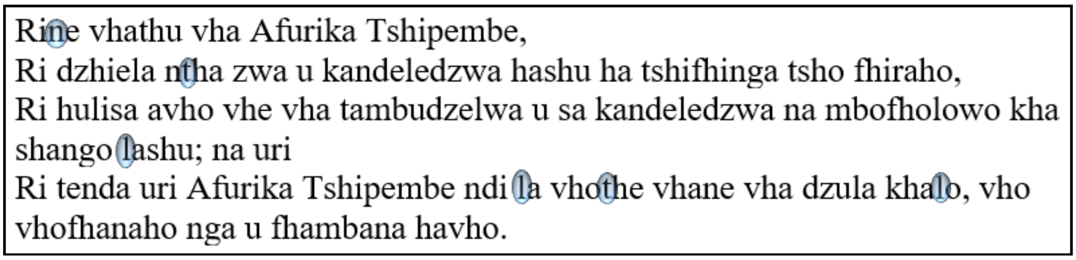
Figure 2. Scanned extract from an Afrikaans text

It is clear that the OCR software did not do well in terms of character recognition, e.g. (*eenvoudig*) *gestel* ‘(simply) put’ as “geetel”, *meeste moderne skrywers* ‘most modern authors’ as “meeate modeme ekryvere”, *naamwoord* ‘noun’ as “naiunwoord” etc. Diacritics are added where they should not be, e.g. ‘ín’ instead of ‘in’, or misscanned, *n instead of ’n.

Misscanning of diacritics is a major problem in languages such as Tshivenḓa, which utilizes seven orthographic symbols containing diacritics. Consider the following extract from The Constitution of the Republic of South Africa in Tshivenḓa (1996):

> *Riṋe vhathu vha Afurika Tshipembe,*
> *Ri dzhiela nṱha zwa u kandeledzwa hashu ha tshifhinga tsho fhiraho,*
> *Ri hulisa avho vhe vha tambudzelwa u sa kandeledzwa na mbofholowo kha shango ḽashu; na uri*
> *Ri tenda uri Afurika Tshipembe ndi ḽa vhoṱhe vhane vha dzula khaḽo, vho vhofhanaho nga u fhambana havho*.

Although the scanning quality is quite good, all diacritics are lost for the encircled consonants *n*, *t* and *l* in Figure 3.



Rine vhathu vha Afurika Tshipembe,
Ri dzhiela ntha zwa u kandeledzwa hashu ha tshifhinga tsho fhiraho,
Ri hulisa avho vhe vha tambudzelwa u sa kandeledzwa na mbofholowo kha shango lashu; na uri
Ri tenda uri Afurika Tshipembe ndi la vhothe vhane vha dzula khalo, vho vhofhanaho nga u fhambana havho.

Figure 3. Misscannings of diacritic signs in Tshivenḓa

In light of the abovementioned examples, it is clear that commercially available software is reasonably sufficient for the building of a "quick and dirty" corpus, though such a corpus may not be appropriate for all possible applications. As mentioned before, the level of clean-up required is directly related to the purpose of the corpus of which the scanned texts will form part.

4. Evaluation of three (commercially available) OCR tools

As stated earlier, software utilized for scanning is one of the three major factors that directly impact the quality of the OCRed texts. Hocking and Puttkammer (2016) indicate that, although commercial and open source OCR engines are available for several languages, not all languages are supported by these engines. Furthermore, the authors are of the opinion that language specific training of OCR software is necessary to maximize accuracy of scanning output. A small experiment was consequently carried out, in which the error and accuracy rates of selected OCR software packages were determined.

In our evaluation we compare two commercially available software packages, i.e. *ABBYY FineReader 14*, *Omnipage Professional 18*, and one locally developed scanning package, *CTexTools*. The OCR quality of these three tools was tested on good quality printouts from President Cyril Ramaphosa's 2020 State of the Nation (SONA) address.

The number of words in English and in the 10 different translations of this address in Afrikaans and the nine official South African languages are given in Table 1.

Table 1: The number of words in English and translations

English	**Afrikaans**	**Sepedi**	**Setswana**	**Sesotho**	**isiZulu**	**isiXhosa**	**Siswati**	**isiNdebele**	**Tshivenḓa**	**Xitsonga**
7530	7683	10521	12691	9714	5393	5959	5702	4823	9817	9630

From Table 1 it is clear that the Nguni languages (isiZulu, isiXhosa, Siswati and isiNdebele), being conjunctively written languages, have fewer orthographic words than the Sotho languages (Sepedi, Setswana and Sesotho), Tshivenḓa and Xitsonga, which all follow a disjunctive orthography. In a disjunctive orthography, word stems and morphemes are written as separate orthographic words, as in example 2, whereas they are all clustered together as single words in a conjunctive orthography as in example 1.

(1) **isiZulu**
Ngiyabasiza
(*ngi*: subject concord 1st person singular + *ya*: present tense marker + *ba*: object concord class 2 + *siza*: verb stem)
I [pres] them help
‘I help them’

(2) **Sepedi**
Ke a ba thuša
(*ke*: subject concord 1st person singular + *a*: present tense marker + *ba*: object concord class 2 + *thuša*: verb stem)
I [pres] them help
‘I help them’

As can be seen in examples (1) and (2), the concept ‘I help them’ is rendered as a single orthographic word in isiZulu, whereas in Sepedi it is written as four separate words. For the purposes of this OCR experiment, any mistake(s) in an orthographic word are counted as an error.

For the present experiment testing OCR quality, isiZulu was selected as representative of the conjunctively written Nguni languages, whereas the disjunctively written Sotho languages were represented by Sepedi. The other languages included in the experiment were Afrikaans and Tshivenḓa. Special attention was given to Tshivenḓa, Afrikaans and Sepedi as they use diacritics which require special attention in OCR scanning. Examples of diacritics include š (Sepedi), ë and ê (Afrikaans) and ṱ (Tshivenḓa).[1] These diacritics frequently occur in texts and are problematic for OCR software available on the commercial market. The respective texts were run through three OCR programmes, namely *ABBYY*, *Omnipage*, and *CTexTools*. The OCR software *ABBYY* was selected because of its known high overall scanning quality and its ability to recognize the frequently occurring character *š* of Sepedi. Even though Sepedi is not one of the languages supported by *ABBYY*, misrecognition of diactritics can be successfully circumvented by selecting a language such as Slovenian, which utilizes similar diacritic signs. *Omnipage* is generally recognized as a pioneer scanning package; it is currently in its 19th version and is highly rated for its built-in training facility. *CTexTools* was selected for its dedicated features for OCR of African languages, especially those employing diacritic signs.

The percentage of scanning errors per language are given in Table 2, graphically illustrated in Figure 3, and the resulting overall accuracy rate in each case in Table 3, graphically illustrated in Figure 5.

[1] š: 1,729 occurences in the Sepedi text of 10,481 words in the SONA, ë and ê: 79 and 26 occurences respectively in the Afrikaans text of 7,657 words, and ṱ: 314 ocurences in the Tshivenḓa text of 9,817 words.

Table 2: Percentage of scanning errors

	ABBYY	Omnipage	*CTexTools*
Afrikaans	0,4	0,9	0,9
Sepedi	0,3	3,7	0,5
isiZulu	0,4	4,8	3,2
Tshivenḓa*	4,4	4,5	1,7

*Statistics for Tshivenḓa were calculated on a subsection of 2,504 words.

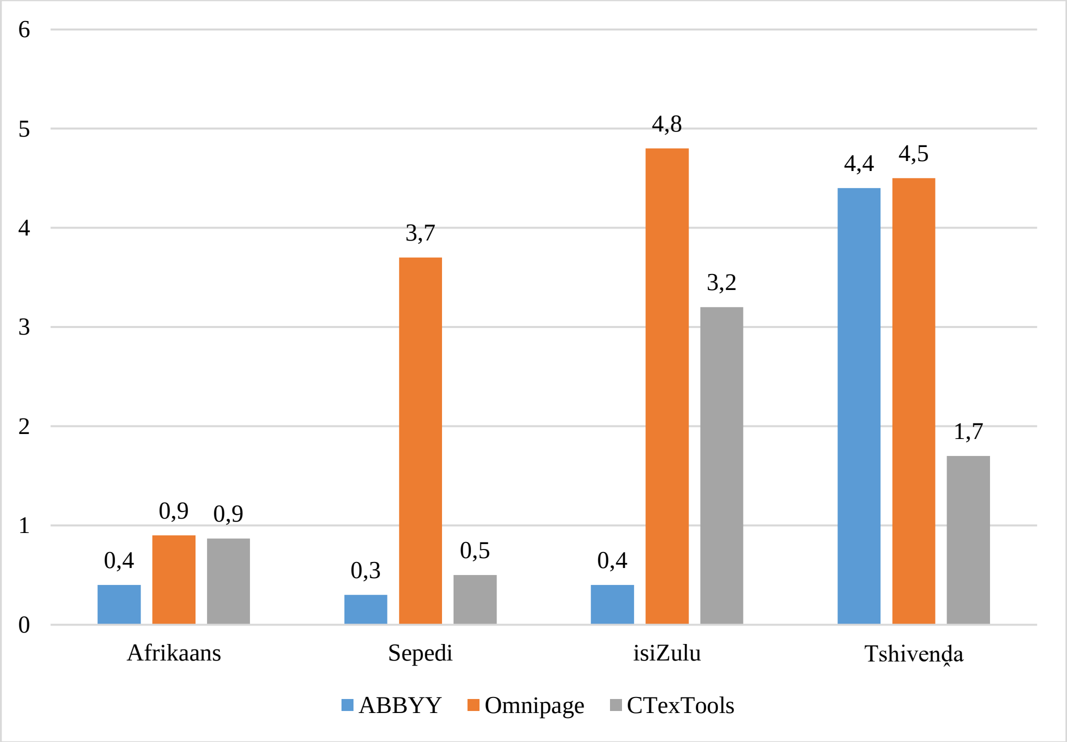


Figure 4. Percentage of scanning errors per language

Table 3: Percentage of accuracy rate

	ABBYY	Omnipage	*CTexTools*
Afrikaans	99,64	99,14	99,1
Sepedi	99,72	96,3	99,52
isiZulu	99,55	95,23	96,81
Tshivenḓa	95,61	95,5	98,3

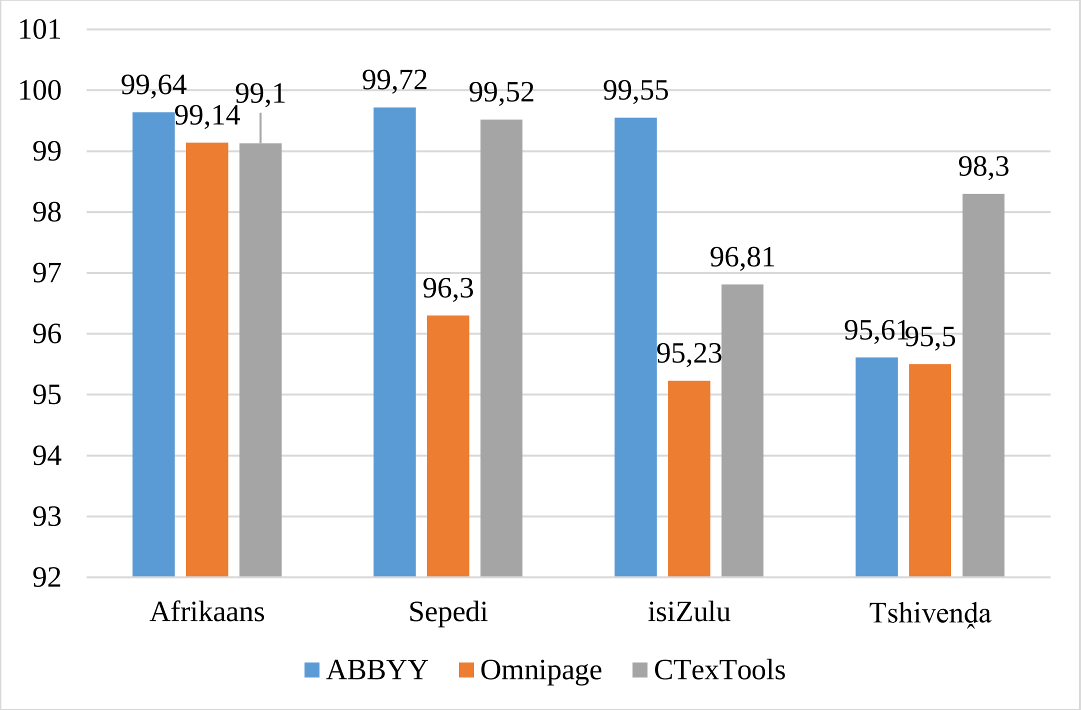


Figure 5. Percentage of OCR accuracy per language

Generally speaking, all three software packages erred by reading

- the number *1* or the letter *l* as the letter *I* e.g. *lebaka* as *Iebaka*;
- *m* as *rn*, e.g. *hom* as *horn* and *ikonomi* as *ikonorni*;
- *w* as *vv* e.g. *Zimbabwe* as *Zimbabvve*;
- *hl* as *hi*, e.g. *mohlagase* as *mohiagase*;
- *e* as *c* or vice versa, e.g. *terselfdertyd* as *tcrsclfdertyd*, *Nedlac* as *Nedlae* and *ukuthembeka* as *ukuthembcka;*
- *o* as *a*, e.g. *eziqinileyo* as *eziqinileya*;
- *o* as *e*, e.g. *yeSabelomali* as *yeSabelemali*; and
- *l* as *t*: *lale* as *tale.*

With regard to Afrikaans, it is clear from Tables 2 and 3 that - based on percentage of scanning errors and accuracy - all three packages performed well, with *ABBYY* being the best performing package, followed by *Omnipage* and *CTexTools*. OCR errors mainly occurred with the vowels *o* and *e* where diacritics were involved, e.g. *teëspoed* scanned as *teespoed*, *díe* as *die*, *belê* as *bele*, *môre* as *more*, etc. In some instances, there was confusion with diacritics, e.g. *gekoördineer* as *gekoórdineer* and *môre* as *more*, and some instances of random capitalisation, e.g. *ekonomie* as *ekonOmie* and *ons* as *Ons* also occurred.

For Sepedi, *ABBYY* performed the best, closely followed by *CTexTools*, with four times more OCR errors in *Omnipage*. Sepedi has only one diacritic, *š* or capital *Š*, which is problematic for OCR, but it is frequently used, i.e. 1,729 times in SONA. Furthermore, it stands in opposition to *s* as e.g. *seba* 'whisper' versus *šeba* 'eat on the side, flavour' and therefore, distinguishing correctly between *š* and *s* is crucial. Many scanning errors in respect of these two characters occurred, e.g. *sego* as *šego*, *šoma* as *soma*, etc. In other cases, š was scanned as another symbol, e.g. *mešomo* as me§*omo*. Finally, word division mistakes were common, e.g. *go tšwa* as *gotšwa*.

ABBYY also performed best for isiZulu, followed by *CTexTools*. Scanning with *Omnipage* however, resulted in up to 10% more OCR errors. Typical errors were omission of characters, e.g. *nokubuyisela* as *nokubuyi ela*, *entando* as *en ando*, and confusion between *i* and *l*, e.g. *asosizini* scanned as *asoslzini*, and *yokululama* as *yokuiulama*. There was also confusion between digits and letters, e.g. the number *1* was scanned as *i,* e.g. *81* scanned as *8i*, and some capitalisations were missed: *sokuZimisela* as *sokuzimisela.*

For Tshivenḓa, *CTexTools* outperformed the other two packages with *ABBYY* and *Omnipage* making twice as many OCR errors. *ABBYY* simply does not cater for the diacritics found in Tshivenḓa and even *Omnipage's* training function did not result in high OCR quality. *CTexTools'*s dedicated catering for diacritics was effective, although not faultless. Typical misscannings were *Nnḓu* as *Nndu*, *Muhaṱuli* as *Muhatuli*, *nṋe* as *nne*, *ḽala* as *lala*, etc.

Overall, the results indicate that *ABBYY* would be the preferred OCR tool for languages not utilizing more than a minimum of diacritic signs, even though those languages may not be specifically supported by the software. For languages such as Tshivenḓa which make use of substantial diacritic marking, *CTexTools* would be the tool of choice.

5. Granularity of cleanness and possible corpus applications

According to Uwe Quasthoff (Institute of Computer Science, Leipzig, Germany, personal communication) different granularity levels of cleanness of corpora enable different application possibilities. More in-depth research is required to determine the exact corpus cleaning strategies and degree of cleanness required, since applications vary in their requirement of cleanness of texts comprising a corpus. The applications of a dirty corpus are limited and might not be usable for applications that require a high(er) degree of e.g. correct spelling and grammatical accuracy. An exact indication of the degree of cleanness required for different applications of corpora will not be attempted in this article –Table 4 merely suggests a broad outline of the degree of cleanness and the possible applications of corpora.

Table 4: Layers of cleanness and their possible applications

Application of corpus	**Status/quality/condition of corpus**	**Required correction methods**
Frequency lists	**Dirty corpus**	No correction necessary, use as is
Authentic (corpus) examples	**Dirty corpus**	No correction necessary, use as is
Concordance lines (keyword-in-context)	**Dirty corpus**	No correction necessary, use as is
Text verification - part of speech (POS) matches	**Clean corpus**	Semi-automatic spelling checking or semi-automatic search and replace operations
Mark-up: e.g. POS, morphological analysis, lemmatisation	**Clean corpus**	Semi-automatic spelling checking or semi-automatic search and replace operations
Rare occurrences of words	**Relatively clean corpus**	Correction of typical OCR errors by semi-automatic search and replace operations

Spelling checkers, Grammar checkers, Text verification - exact matches	**100% clean corpus**	Correction of all OCR errors by proofreading of the text or at least semi-automatic spelling checking or semi-automatic search and replace operations

The three topmost rows of Table 4 refer to "raw corpora", i.e. corpora consisting of texts without any correction of scanning errors. These corpora are suitable for e.g. lexicographic applications, such as the generation and study of keywords in context, finding collocations and word clusters, selecting authentic examples of use for the treatment of lemmas and the compilation of frequency lists. The latter can, for example, serve as the point of departure for the compilation of lemma lists for dictionaries. Processing of texts for lexicographic purposes therefore tolerates a certain amount of noise, provided that it is offset by large volumes of usable data:

> As a general rule, lexicographers prefer size to granularity. That is, if the choice is between high volumes of data with the occasional bit of noise, or very 'clean', carefully annotated data in much smaller quantities, they will always go for the former.
>
> (Atkins and Rundell 2008:93)

The inaccuracy factors involved in the abovementioned lexicographic applications do not have a substantial impact on the tasks at hand. In the case of frequency lists, inaccuracy lies in a reduced count of the frequency of a word. Searching for *mošemane* 'boy' in a 10 million word Sepedi corpus results in 764 hits, which puts *mošemane* in the top 1000 frequencies for Sepedi. For frequently occurring words such as *mošemane*, an inaccurate frequency count does not really prevent the lexicographer from concluding that *mošemane* is a top frequency word and therefore a strong candidate for inclusion in the dictionary. The reduced frequency count will also not influence the star-rated frequency category indication, e.g. **mošemane**** in the *Oxford Bilingual School Dictionary: Northern Sotho and English* (ONSD) (De Schryver 2007).[2] For low frequency words, however, inaccurate frequency counts due to *inter alia* scanning errors in the texts of a corpus can lead to exclusion of such words from the dictionary.

An additional challenge for the lexicographic application of OCR software lies in finding authentic examples for dictionary use. Inaccurate scanning results in a reduced number of possible usage examples that can potentially be included in the dictionary as is demonstrated for Sepedi in Table 5.

[2] In the ONSD, the 500 most frequent words are labelled with three stars (***), the next 500 with two stars (**) and the 500 following that with one (*) star.

Table 5: Concordance lines with misscanned keywords in context for *mošemane* 'boy'

Context left	**Keyword**	Context right
Madisha-a-Leolo e be e le	**mogemane**	yo motelele bjalo ka thutlwa.
gwa se tsebe motho gore naa ke	**mo~emane**	goba mosetsana. O kgonne go lemoga
sengwe seo se kgahlilego Pebetse ka	**mo(emane**	yo. Ka nako ye nngwe pelo ya gagwe e
Mphahlele 0 ile Amerika. Fanyane,	**moiemane**	yo mokoto 0 feditse dijo. (e) Maina a ka
polao. Go bolawa motho yo mofsa,	**molemane**	goba ngwanenyana; eupša gagolo go
tIakaleng. Ke ile ka lala ke robetse le	**mosemane**	yola wa go mpule1a e le go Peter. Peter

Examples such as **mogemane*, **mo~emane*, **mo(emane*, **moiemane*, **molemane* and **mosemane* in Table 5 will not be offered in the OCR software as a misspelling of *mošemane* due to a scanning error, and thus will typically not be included in the search results However, a reduced number of concordance lines carries with it the possibility that evidence of a specific sense of a given word could be lost as a result of the misspelling/misscanning of the search word in the corpus. In the case of languages with substantial diacrtic marking, this is especially challenging. For Sepedi words containing the inverted circumflex on the letter "s", i.e. "š", the inaccuracy factor could be substantial for texts scanned with older technology OCR software. For example, in the case of *tšweletša* 'produce, continue', this word was erroneously recognised by the software as **tsweletsa*, **tsweletša* or **tšweletsa* in 62% (1,396 out of 2,250) of cases and could lead to the loss of valuable information. Reduced frequency counts are especially problematic when it comes to the study of rarely used words. If there is only a single or a small number of examples in the corpus, misspelled/misscanned words could lead to incorrect or insufficient conclusions regarding frequency, senses and examples of use.

For the next two levels in Table 4 related to part of speech (POS) mark-up, a margin of error can still be tolerated. If, for example, in a morpho-syntactic study on adjectives a small number of adjectives are not detected as a result of incorrect part of speech labels (for e.g. demonstratives, class prefixes and adjective stems), a sufficient number of correctly tagged adjective phrases might still be on offer and may be sufficient to draw valid syntactic conclusions. However, in spellchecker, grammar checker and text verification applications (cf. Prinsloo and De Schryver 2003, 2004; De Schryver and Prinsloo 2004; and Prinsloo 2019), exact matches are essential. For such applications, the corpus should be clean since the frequency of such matches in the corpus can be limited to a single occurrence, and any mistake in the tag(s) could return false negatives (i.e. correctly spelled words flagged as incorrect by a spelling checker), or false positive counts (i.e. incorrect words not flagged by a spelling checker).

6. Strategies for text cleaning

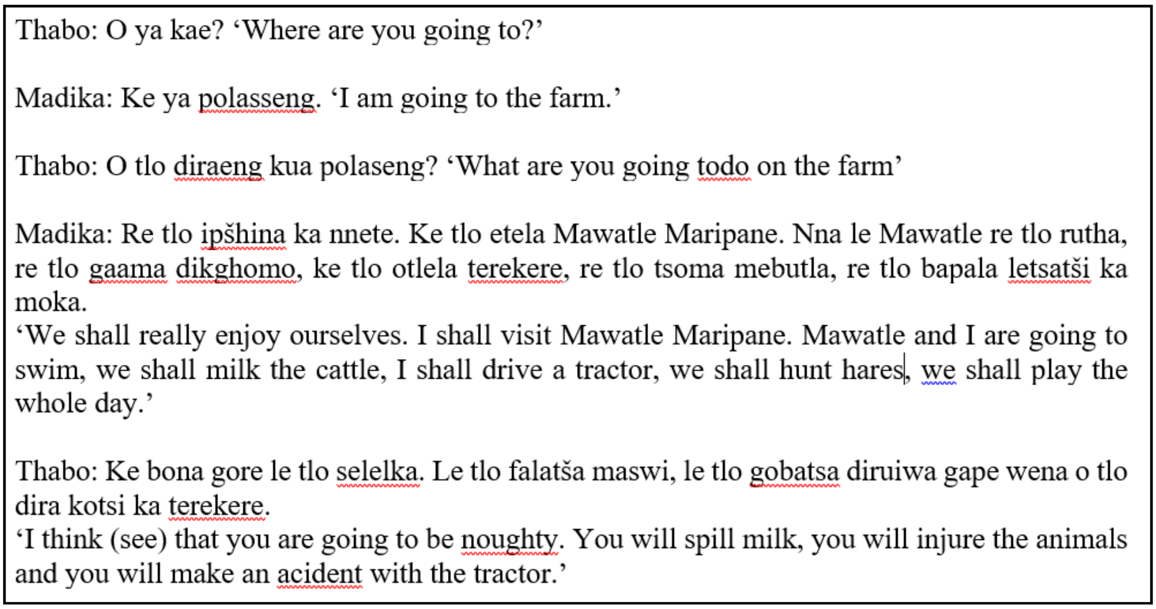
The present discussion has the linguist in focus and therefore favours strategies for text cleaning that can be performed by linguists themselves, i.e. actions that do not have to be performed by skilled computer programmers. More sophisticated procedures such as the use of N-grams require a high level of computational skill, which makes these procedures less ideal within the context of lesser-resourced languages. A detailed discussion on the use of N-grams will therefore not be attempted here.

Simple text cleaning strategies include:

- manual correction;
- spellchecker support;
- automatic search and replace individual items;
- automatic search and replace with basic macros;
- detecting and cleaning duplications through concordance line repetitions; and
- anonymizing texts containing sensitive information.

Manual correction of texts refers to a 100% human read through. Corrections are made by means of retyping words or incorrectly scanned letters in words, deleting incorrectly inserted letters, numbers, figures, punctuation marks, etc. and removing superfluous spaces between words and/or punctuation marks. This approach usually renders accurate and clean texts but is time consuming and not commercially feasible for large text collections. Moreover, this strategy is also still prone to human error.

Spellchecker support entails engaging a main spellchecker, e.g. Microsoft main dictionary for British English, which can be supported by custom dictionaries (wordlists) which are generated from existing corpora or gradually built up by the user. Simultaneous checking of multiple languages is possible, e.g. a main English dictionary supported by custom dictionaries for African languages, such as Sepedi, isiZulu and Xitsonga. Consider, for instance, the example presented in Figure 6 of a random Sepedi text riddled with mistakes, spellchecked by a Sepedi custom dictionary running parallel to an English main dictionary.



Thabo: O ya kae? ‘Where are you going to?’

Madika: Ke ya polasseng. ‘I am going to the farm.’

Thabo: O tlo diraeng kua polaseng? ‘What are you going todo on the farm’

Madika: Re tlo ipšhina ka nnete. Ke tlo etela Mawatle Maripane. Nna le Mawatle re tlo rutha, re tlo gaama dikghomo, ke tlo otlela terekere, re tlo tsoma mebutla, re tlo bapala letsatši ka moka.
‘We shall really enjoy ourselves. I shall visit Mawatle Maripane. Mawatle and I are going to swim, we shall milk the cattle, I shall drive a tractor, we shall hunt hares, we shall play the whole day.’

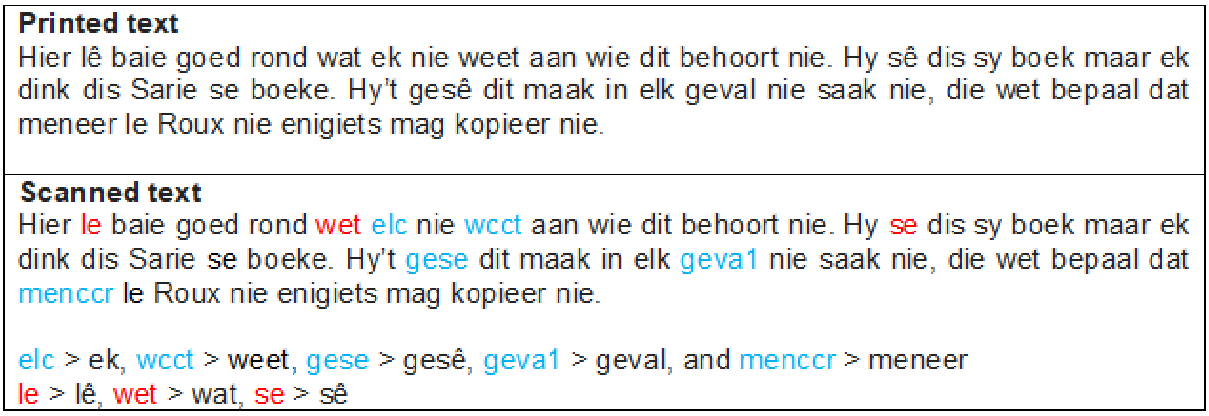
Thabo: Ke bona gore le tlo selelka. Le tlo falatša maswi, le tlo gobatsa diruiwa gape wena o tlo dira kotsi ka terekere.
‘I think (see) that you are going to be noughty. You will spill milk, you will injure the animals and you will make an acident with the tractor.’

Figure 6. Sepedi text containing spelling errors flagged by spellcheckers

All spelling errors in both Sepedi and English were detected simultaneously. The incorrectly spelled English words **todo*, **noughty* and **acident* were detected by the English main dictionary with suggested corrections *to do*, *naughty* and *accident* respectively. The Sepedi errors **polasseng (polaseng)*, **diraeng (dira eng)*, **ipšhina (ispshina)*, etc. were detected by the Sepedi custom dictionary.

The user can choose between a “replace all” automatic option or to accept/skip the flagged words one by one. The automatic option is somewhat risky as correctly used instances of a word could also be changed simply because its existence was not foreseen.

Automatic and semi-automatic search and replace of individual items refers to the “search & replace” functions available in a typical word processor. This function consists of progressing through the document with the “next” button and correction of incorrect words. Automatic replacement can be done by simply selecting “replace all” or the creation and use of basic macros. The user can create macros by simply recording a series of search and replace actions for known typical errors. Again, auto-replacement by means of the “find and replace” function should be used with caution for the same reasons as automatic changes by means of spelling checkers mentioned above. Consider a randomly selected Afrikaans text in Figure 7 in which many scanning errors occur, especially in regard to the circumflex “^” and the letter “e”. The correct original forms are given following the symbol “>”.



Printed text

Hier lê baie goed rond wat ek nie weet aan wie dit behoort nie. Hy sê dis sy boek maar ek dink dis Sarie se boeke. Hy't gesê dit maak in elk geval nie saak nie, die wet bepaal dat meneer le Roux nie enigiets mag kopieer nie.

Scanned text

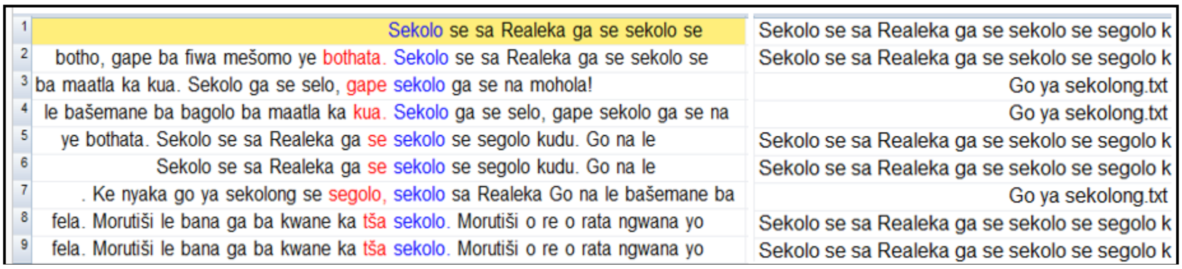
Hier le baie goed rond wet elc nie wcct aan wie dit behoort nie. Hy se dis sy boek maar ek dink dis Sarie se boeke. Hy't gese dit maak in elk geva1 nie saak nie, die wet bepaal dat menccr le Roux nie enigiets mag kopieer nie.

elc > ek, wcct > weet, gese > gesê, geva1 > geval, and menccr > meneer
le > lê, wet > wat, se > sê

Figure 7. Automatic and semi-automatic error correction

Errors such as *gese*, *elc*, *geva1*, *wcct*, and *menccr* indicated in blue can, with relatively low risk, be corrected by automatic search and replacement with their respective correct forms. For *wet*, *se* and *le* marked in red, however a semi-automatic process is required since, *wet* could be correct meaning ‘law’, or a misscanning of *wat* ‘what’, and the correct originals for *se* could be *se* ‘of’ or *sê* ‘say’ and *le* could be the correct form appearing in some surnames. It can therefore be concluded that a combination of automatic and semi-automatic error correction are useful strategies when applied with the necessary caution.

Various strategies can be used for *detecting and removing duplications*. One option is to generate concordance lines from the text to see duplications, as shown in Figure 8 for Sepedi.

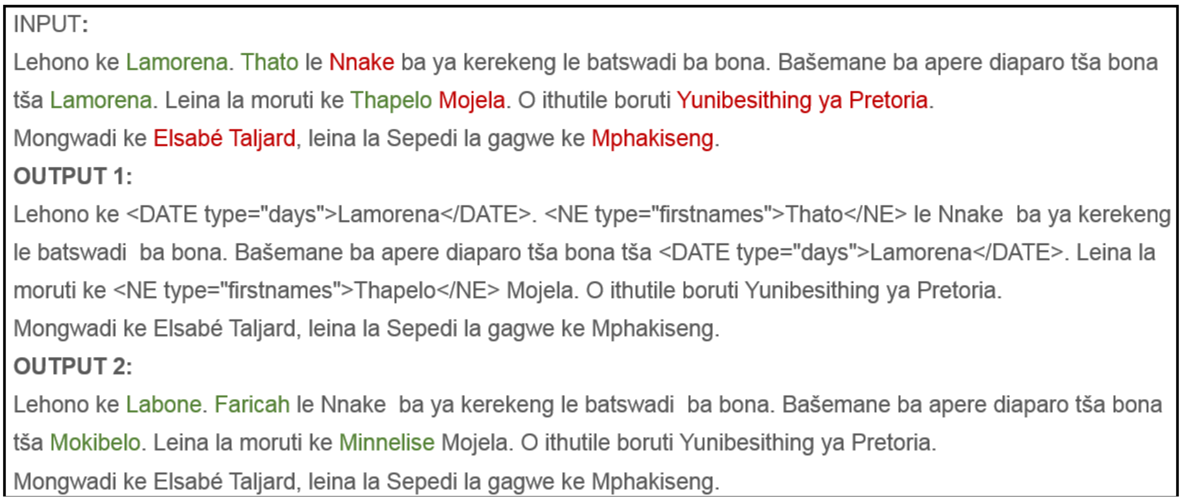


1	Sekolo se sa Realeka ga se sekolo se	Sekolo se sa Realeka ga se sekolo se segolo k
2	botho, gape ba fiwa mešomo ye bothata. Sekolo se sa Realeka ga se sekolo se	Sekolo se sa Realeka ga se sekolo se segolo k
3	ba maatla ka kua. Sekolo ga se selo, gape sekolo ga se na mohola!	Go ya sekolong.txt
4	le bašemane ba bagolo ba maatla ka kua. Sekolo ga se selo, gape sekolo ga se na	Go ya sekolong.txt
5	ye bothata. Sekolo se sa Realeka ga se sekolo se segolo kudu. Go na le	Sekolo se sa Realeka ga se sekolo se segolo k
6	Sekolo se sa Realeka ga se sekolo se segolo kudu. Go na le	Sekolo se sa Realeka ga se sekolo se segolo k
7	. Ke nyaka go ya sekolong se segolo, sekolo sa Realeka Go na le bašemane ba	Go ya sekolong.txt
8	fela. Morutiši le bana ga ba kwane ka tša sekolo. Morutiši o re o rata ngwana yo	Sekolo se sa Realeka ga se sekolo se segolo k
9	fela. Morutiši le bana ga ba kwane ka tša sekolo. Morutiši o re o rata ngwana yo	Sekolo se sa Realeka ga se sekolo se segolo k

Figure 8. Repetitions of concordance lines for Sepedi

In Figure 8, duplication of text can be seen in lines one and two, five and six, and eight and nine. Since the source of the concordance line is provided, duplicate text files can simply be removed from the corpus. A simple corpus query tool such as WordSmith Tools has a function for automatically identifying and deleting duplications.

A final issue related to the cleaning of texts is *anonymising texts*. Some data providers may request the anonymising of texts which could possibly contain sensitive information. Software for carrying out this process is available, i.e. the Autshumato Text Anonymizer. According to its self-description, the "Autshumato Text Anonymizer [is] a tool for the anonymisation of text corpora which entails the identification of entities that may convey confidential information and replacing those entities with randomly selected entities of the same type". This tool is available for all 11 official languages of South Africa. The anonimisation of texts is often executed by means of a (semi-)automatic search and replace action. Consider the Sepedi example in Figure 9 in which *Lamorena* 'Sunday', *Yunibesithing* ya *Pretoria* 'at the University of Pretoria' and the proper names *Thato*, *Nnake*, *Thapelo*, *Mojela*, *Elsabé*, *Taljard* and *Mphakiseng* are targets for anonymization in the Sepedi texts.



INPUT:

Lehono ke Lamorena. Thato le Nnake ba ya kerekeng le batswadi ba bona. Bašemane ba apere diaparo tša bona tša Lamorena. Leina la moruti ke Thapelo Mojela. O ithutile boruti Yunibesithing ya Pretoria.

Mongwadi ke Elsabé Taljard, leina la Sepedi la gagwe ke Mphakiseng.

OUTPUT 1:

Lehono ke <DATE type="days">Lamorena</DATE>. <NE type="firstnames">Thato</NE> le Nnake ba ya kerekeng le batswadi ba bona. Bašemane ba apere diaparo tša bona tša <DATE type="days">Lamorena</DATE>. Leina la moruti ke <NE type="firstnames">Thapelo</NE> Mojela. O ithutile boruti Yunibesithing ya Pretoria.

Mongwadi ke Elsabé Taljard, leina la Sepedi la gagwe ke Mphakiseng.

OUTPUT 2:

Lehono ke Labone. Faricah le Nnake ba ya kerekeng le batswadi ba bona. Bašemane ba apere diaparo tša bona tša Mokibelo. Leina la moruti ke Minnelise Mojela. O ithutile boruti Yunibesithing ya Pretoria.

Mongwadi ke Elsabé Taljard, leina la Sepedi la gagwe ke Mphakiseng.

Figure 9. Anonymising Sepedi texts

In this example the software succeeded in replacing *Sunday* with another day of the week, as well as the names *Thato* and *Thapelo* with *Faricah* and *Minnelise* respectively but failed to do so with *Nnake*, *Mojela*, *Elsabé*, *Taljard*, *Mphakiseng* and *Yunibesithing ya Pretoria.*

7. Conclusion

Cleaning texts which will eventually constitute a corpus is much more problematic than meets the eye. Errors in scanned texts include spelling mistakes, grammatical errors, misscannings, etc. The biggest challenges are posed by OCR-scanned texts, specifically for those languages that utilize a significant number of diacritics. One of the most important factors impacting the quality of scanned texts is the choice of software that is used for OCR. Three OCR engines were compared in the present study for error and accuracy rates. In contrast to Hocking and Puttkammer's (2016:1) remark that "using an OCR engine designed for another language could have a negative impact on the accuracy of the resulting text", the results from the present study indicate that *ABBYY* is the preferred tool for languages in which a minimum of diacritics is used, even though those languages may not be officially supported by the software. Conversely, for languages with considerable diacritic marking, such as Tshivenḓa, *CTexTools* consistently outperforms the other two engines. It is furthermore argued that different levels of cleanness are required for different tasks/applications. Linguists and researchers should select the cleaning strategy and level of cleanness which most closely alligns with the ultimate intended use of the corpus.

Acknowledgements

This research is supported by (a) the South African Centre for Digital Language Resources (SADiLaR) and (b) the National Research Foundation of South Africa (Grant specific unique references 145996 (E Taljard) and 85763 (D J Prinsloo)). The Grantholders acknowledge that opinions, findings and conclusions or recommendations expressed in any publication generated by the NRF supported research are those of the authors, and that the sponsors accept no liability whatsoever in this regard.

References

ABBYY Finereader 14. Available online: https://www.abbyy.com/en-eu/finereader/ (Accessed 14 April 2020).

Atkins, B.T. Sue and M. Rundell. 2008. *The Oxford Guide to Practical Lexicography*. Oxford, New York: Oxford University Press. https://doi.org/10.1093/ijl/ecn036

Autshumato Text Anonymizer. Available online: https://autshumato.sourceforge.net/projects/autshumatota (Accessed 19 April 2020).

Baroni, M. and A. Kilgarriff. 2006. Large linguistically processed Web corpora for multiple languages. *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations*: 87-90. Trento: Italy. https://doi.org/10.3115/1608974.1608976

Baroni, M., C. Francis, A. Kilgarriff, and S. Sharoff. 2008. CleanEval: a competition for cleaning webpages. *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008*. Marrakech, Morocco. 26 May-1 June 2008.

Beeld. Available online: https://www.netwerk24.com/za/beeld (Accessed 18 April 2020).

Bosch, S. 2020. Computational morphology systems for Zulu – a comparison. *Nordic Journal of African Studies*, 29(3): 1-28.

Bosch, S. and L. Pretorius. 2011. Towards Zulu corpus clean-up, lexicon development and corpus annotation by means of computational morphological analysis. *SA Journal of African Languages*, 31(1): 138-158. https://doi.org/10.1080/02572117.2019.12063275

De Schryver, G.-M. (ed.). 2007. *Oxford Bilingual School Dictionary: Northern Sotho and English*. (First edition.) Cape Town: OUP Southern Africa.

De Schryver, G.-M. and D.J. Prinsloo. 2004. Spellcheckers for the South African languages, Part 1: The status quo and options for improvement. *SA Journal of African Languages,* 24(1): 57-82. https://doi.org/10.1080/02572117.2004.10587226

Die Burger. Available online: https://www.netwerk24.com/za/die-burger (Accessed 18 April 2020).

Evert, S. 2008. A lightweight and efficient tool for cleaning Web pages. Available online: http://www.lrec-conf.org/proceedings/lrec2008/pdf/885_paper.pdf (Accessed 30 March 2020).

Fairon, C., H. Naets, A. Kilgarriff and G.-M. De Schryver, (eds.) 2007. Building and Exploring Web Corpora. *Proceedings of the 3rd Web as Corpus Workshop, Incorporating Cleaneval*. De Louvain: UCL Presses Universitaires.

Généreux, M., I. Hendrickx and A. Mendes. 2012. A Large Portuguese Corpus On-Line: Cleaning and Preprocessing. In *International Conference on Computational Processing of the Portuguese Language:*113-120. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-28885-2_13

Goldhahn, D., T. Eckart, and U. Quasthoff. 2012. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation* (LREC'12), 2012. https://doi.org/10.1007/978-3-319-12655-5_1

Graën, J., D. Batinic, and M. Volk. 2014. Cleaning the Europarl corpus for linguistic applications. *Proceedings of the 12th Edition of the KONVENS Conference* (1): 222-227. Hildesheim: Universitätsverlag Hildesheim.

Hocking, J. and M. Puttkammer. 2016. Optical character recognition for South African Languages. *Proceedings of the Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference,* 1-5. Stellenbosch: South Africa. https://doi.org/10.1109/robomech.2016.7813139

Hofmann, K. and W. Weerkamp. 2007. Web corpus cleaning using Content and Structure. *Proceedings of the 3rd Web as Corpus Workshop, Incorporating Cleaneval*. De Louvain: UCL Presses Universitaires.

Kantner, C., A. Kutter., A. Hildebrandt and M. Püttcher. 2011. How to get rid of the noise in the corpus: cleaning large samples of digital newspaper texts. *International Relations Online Working Paper.*

Knoblock C., D. Lopresti, R. Shourya and L.V. Subramaniam. 2007. Introduction to special issue on noisy text analytics. *International Journal on Document Analysis and Recognition,* 10: 127-128. https://doi.org/10.1007/s10032-007-0058-9

Kohlschütter, C., P. Frankhauser and W. Nedjl. 2010. Boilerplate Detection using Shallow Text Features. *Proceedings of the third ACM international conference on Web search and data mining*: 441–450. New York. https://doi.org/10.1145/1718487.1718542

Krstev, C. and Stanković, R. 2019. Old or new, we repair, adjust and alter (texts). *Infotheca – Journal for Digital Humanities*, 19(2): 61-80. https://doi.org/10.18485/infotheca.2019.19.2.3

Kuhn, T.Z., P. Dekker, B. Šandrih, R. Zviel-Gershin, S.A. Holdt and T. Schoonheim. 2019. Crowdsourcing corpus cleaning for language learning resource development. *enetCollect 3rd annual meeting*. Lisbon: 14-15 March 2019.

Molefe, A.I. (n.d.) *Indlela yolimi lwesiZulu ibanga lesi-2.* Cape Town: Via Afrika Limited.

NCLEANER. Available online: http://webascorpus.sf.net (Accessed 05 May 2020).

Omnipage: Available online: https://softfamous.com/omnipage-professional/ (Accessed 05 May 2020).

PanSALB. 2008a. *Imithetho yokubhala nobhalomagama LwesiZulu*. Pretoria: PanSALB.

PanSALB. 2008b. *Imithetho yokutlola nokupeleda isiNdebele*. Pretoria: PanSALB.

PanSALB. 2008c. *Imitsetfo yekupela nelubhalomagama LweSiswati*. Pretoria. PanSALB.

PanSALB. 2008d. *Melao ya mongwalo le mopeleto ya Sesotho sa Leboa*. Pretoria: PanSALB.

PanSALB. 2008e. *Melawana ya mokwalo le mopeleto Setswana*. Pretoria: PanSALB.

PanSALB. 2008f. *Melawana ya mopeleto le karohanyo ya mantswe mongolong wa Sesotho*. Pretoria: PanSALB.

PanSALB. 2008g. *Milawu ya mapeleleto na matsalelo ya Xitsonga*. Pretoria: PanSALB.

PanSALB. 2008h. *Milayo ya kupelet ele na kunwalelle kwa Tshivenḓa*. Pretoria: PanSALB.

President Cyril Ramaphosa: 2020 State of the Nation Address. Available online: https://www.gov.za/ve/speeches/president-cyril-ramaphosa-2020-state-nation-address-13-feb-2020-0000 (Accessed 08 March 2021). (Accessed 08 March 2021). https://doi.org/10.1093/ww/9780199540884.013.u31823

Prinsloo, D.J. 2009. Cleaning text corpora of Afrikaans and African languages for lexicographic purposes. *AFRILEX 2009*. University of the Western Cape: 6-8 July 2009.

Prinsloo, D.J. 2019. Detection and lexicographic treatment of salient features in e-dictionaries for African languages. *International Journal of lexicography*: 1-19. https://doi.org/10.1093/ijl/ecz031

Prinsloo, D.J. and G.-M. de Schryver. 2003. Non-word error detection in current South African spellcheckers. *Southern African Linguistics and Applied Language Studies,* 21(4): 307-326. https://doi.org/10.2989/16073610309486351

Prinsloo, D.J. and G.-M. de Schryver. 2004. Spellcheckers for the South African languages, Part 2: The utilisation of clusters of circumfixes. *SA Journal of African Languages*, 24: 83-94. https://doi.org/10.1080/02572117.2004.10587227

Prinsloo, D.J. and E. Taljard. 2019. Corpus cleaning strategies for African Language texts. *2nd International Conference of the Digital Humanities Association of Southern Africa (DHASA)*. University of Pretoria: 25-29 March 2019.

Schäfer, R. 2016. Accurate and efficient general-purpose boilerplate detection for crawled web corpora. *Language Resources and Evaluation*, 51: 873–889. https://doi.org/10.1007/s10579-016-9359-2

Terminology and Orthography No. 4. 1993. *isiZulu Terminology and Orthography*. Department of Education and Training. Pretoria: Government Printer.

Terminology and Orthography No. 4. 1988. *Northern Sotho Terminology and Orthography*. Department of Education and Training. Pretoria: Government Printer.

The Constitution in Tshivenḓa. 1996. Available online: https://www.polity.org.za/article/the-constitution-in-tshivenda-1996-2017-02-27 (Accessed 13 December 2020).

WordSmith Tools. Available online: https://www.lexically.net/wordsmith/ (Accessed 13 December 2020).