The Effects of a Corpus on isiZulu Spellcheckers based on N-grams

Balone NDABA[1], Hussein SULEMAN[1], C. Maria KEET[1], Langa KHUMALO[2]
[1]*Department of Computer Science, University of Cape Town, University Avenue 18, Cape Town, 7701, South Africa*
Tel: +27 21 6502667, Fax: + 27 21 6503551, Email: {bndaba,hussein,mkeet}@cs.uct.ac.za
[2]*University language Planning and Development Office, University of KwaZulu-Natal, Howard College Campus, Durban, 4041, South Africa*
Tel: +27 31 2603589, Fax: +27 31 2603589, Email: khumalol@ukzn.ac.za

Abstract: Correct spelling contributes to good content accessibility and readability for textual documents. However, there are few spellcheckers for Bantu languages such as isiZulu, the major language in South Africa. The objective of this research is to investigate development of spellcheckers for isiZulu and, more generally, an approach that can be reused across Bantu languages. To fill this gap in an extensible way, we used data-driven statistical language models with trigrams and quadrigrams. The models were trained on three different isiZulu corpora, being Ukwabelana, a selection of the isiZulu National Corpus, and a small corpus of news items. The system performed better with trigrams than with quadrigrams, and performance depended on the training and testing corpora. When the system was trained with old text (bible in isiZulu), it did not perform well when tested with the two corpora that contain more recent texts, such as the constitution and news items. The highest accuracy obtained was 89%. Given that data-driven statistical language models constitute a language-independent approach, we conclude that data-driven spellcheckers for all Bantu languages are indeed feasible. They are, however, sensitive to the training and testing data. This is less resource-intensive compared to manual specification of rules, and therefore the potential impact on realising spellcheckers for Bantu languages is now practically within reach. The potential societal impact of spellchecker-supported tools and apps is incalculable.

Keywords: spellchecker, n-grams, corpora, isiZulu.

1. Introduction

Spellcheckers are used in word processing programs, search engines, email applications, cell phones, blogs, forums and several other computer applications. They are also useful tools for language learning and to improve a Web page and increase its ranking. Current ICTs support languages other than English, yet there is still limited support for African languages, including isiZulu, which is the most widely spoken language in South Africa by first/home language speakers (23%, or about 11 million people). The only existing partial isiZulu spellchecker was developed by translate.org.za as a free plugin to OpenOffice[1]. This has not been updated since 2009 and does not function with OpenOffice 4.x anymore. Two other proof-of-concept spellcheckers have been evaluated [4,14,16], which claim to achieve about 95% lexical recall with a word-based approach and rules; however, they are not available for use.

There are various ways to create spellcheckers, principally using a data-driven or rule-based approach. We focus on the former approach, which uses a dictionary or a text corpus.

[1] http://extensions.openoffice.org/en/project/zulu-spell-checker

Whole words or extracted n-gram statistics are used to drive the spellchecker [7][15]. Because no freely accessible isiZulu dictionary is available, any data-driven statistical language model must be derived from text corpora. It is very well possible that the quality of the spellchecker then depends on the quality of the corpus. To examine this, we developed an isiZulu spellchecker, using trigrams or quadrigrams, and evaluated it with three corpora: Ukwabelana [18]; a sample corpus of the IsiZulu National Corpus (Sample INC) [10]; and an in-house small corpus of *Isolezwe* and *isiZulu.news24* news articles of the past few months. While the latter two corpora contain contemporary texts, Ukwabelana contains older text and different topics; e.g., there is no office of the public protector (news items) in the bible stories (Ukwabelana).

The remainder of the paper contains general background on spellcheckers in Section 2. Section 3 describes the system's design. Sections 4 and 5 cover the experimental set-up and the results. We discuss the results in Section 6 and conclude in Section 7.

2. Preliminaries and Related Work

We describe existing spellcheckers developed using a data-driven statistical language model, and the types of errors and the way these spellcheckers detect and correct errors.

There are several spelling errors [6], of which the two main types of errors most spellcheckers aim at finding are non-word errors and real-word errors. Non-word errors are spelling errors resulting from words that do not appear in the reference dictionary and real-word errors are words that are in the reference dictionary but are actually erroneous spellings of some other words [19]. A spellchecker will detect a misspelled word and, depending on the level error, provide a set of suggestions [13]. Non-word errors are relatively easier to detect and eradicate. Real word errors are more intricate, especially in an agglutinative language like isiZulu, because such an error usually affects the syntax and semantics of the whole sentence, which in some cases requires human involvement for detection [1]. Only a few spellcheckers attempt to detect real-word errors [8]. The use of spelling correctors should be handled with care. This is because some errors are attributed to auto-correctors [9]: a person may input a word they are not sure of and an auto-corrector can give completely wrong feedback.

A spellchecker detector module is responsible for determining whether a word is considered misspelled or not, for which an n-gram analysis can be used. An n-gram is an n letter subsequence of a string, where n usually is 1, 2, or 3 [1], referred to as unigrams, bigrams, and trigrams, respectively; e.g., the bigrams of *yebo* ('yes') are *ye*, *eb,* and *bo*. In general, n-gram analysis techniques check each n-gram in an input and compare it against an existing table of n-gram statistics that are computed from a text corpus [21], which are the frequency counts or probabilities of occurrence of n-grams. Provided the text corpus is good enough, it can be used without a dictionary, yet find the position of the error in the misspelled word. This is in some cases achieved by employing character-based n-gram language models [22] but it can also use word-based n-gram language models. N-gram analysis is a statistical approach, so it is mainly influenced by the corpus size, cleanliness, and lexical diversity. The frequency statistics for each word is compared with the system threshold, which differs in different spellcheckers [3]. Normally, if the frequency is below the threshold, the word is identified as wrongly spelled. This means n-grams that do not occur or occur infrequently are considered to be misspellings.

An error corrector module is responsible for providing a set of possible corrections for a misspelled word. Most misspellings involve at most one character change from the intended word [11], because of transposition of two letters (transposition error), adding an extra letter (insertion error), omitting one letter (deletion error) or mistyping one letter [5]. N-grams can also be used for error correction by assuming certain n-grams within a word are correctly spelled and to fix the remaining n-grams [9]. A list of words is established as

suggestions (e.g. [13]), based on a ranking system that may use the minimum edit (Levenshtein) distance that computes the shortest distances to suggested words [20] and the word with the shortest distance will be considered as the best suggestion. Error correction was not included in the isiZulu spellchecker.

A corpus is a large collection of texts [14], which can be built from collected documents in public archives, libraries, by consulting language experts, and using already existing dictionaries [20]. Language evolves, so so should the corpora to yield expected results. As a data-driven spellchecker is as good as its corpus, one needs to be careful in selecting and using corpora. There is currently only one open source isiZulu corpus—Ukwabelana [18]—but it is possible to create new corpora by crawling the Web or explicitly assembling the written word into a usable corpus.

3. System design and implementation

This section briefly describes the design and implementation of the data-driven spellchecker system, whose components are shown in Figure 1.

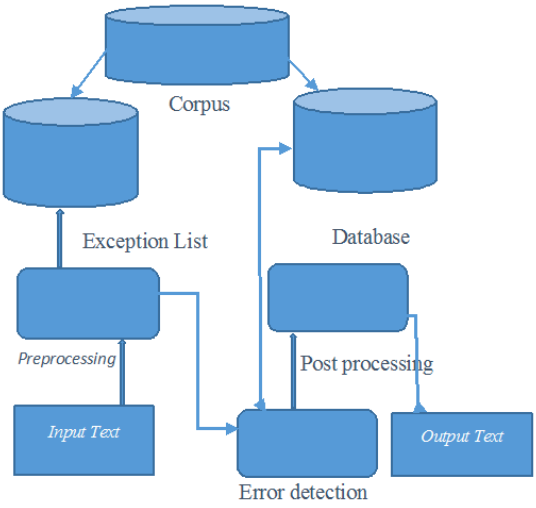


Figure 1. System architecture; the software is implemented in Java and a MySQL database.

The preprocessing algorithm (bottom-left of Figure 1) takes in the input text, tokenizes it, and searches for words in the exception list. The exception list contains known correctly spelled words that should not be spellchecked, so that only words not in the exception list are taken to the error detection algorithm. It is implemented using hashing [2], which is known to be fast.

For the error detection algorithm, we use a statistical method based on a language model that is a combination of the character trigrams or character quadrigrams and the probability of having each trigram or quadrigram in a particular training dataset. A character trigram model is a probabilistic model. A trigram is a set of 3 consecutive characters taken from a string (a word). The next step is to obtain frequencies for each 3-character sequence from the database, which the model then considers as a word. The algorithm then finds the probability P of each word w_i out of the total number of words in the training dataset N, given as,

$$P\left(w_i\right) = \frac{w_i frequency}{N\ frequency}$$

We set the threshold t. Any $P<t$ means that the word based on statistics is a wrong word form. A character quadrigram model used sequences of 4 characters and the process is as described for the trigrams. Note that these algorithms are language-independent.

The implementation of the spellchecker has 2 modules: a training module and a spellchecking module. The training module ingests a training text corpus, calculating the n-gram statistics and storing this in a database table indexed on n-gram. The spellchecking module checks if a query word or sentence (presumably from a test corpus) is correctly spelled. For each word, this module first consults a hash table (built from the training corpus) to determine if the query word is a known word. If this fails, the database is consulted to obtain a vector of n-gram probabilities for each n-gram contained in the query word. The threshold, as discussed above, is then applied to each n-gram probability to decide if the word is correctly spelled or not.

4. Experiment design and execution

The aim of the experiment is to determine which combination of variables or design choices yields the best spellchecker: either trigrams or quadrigrams; which threshold value; and which corpus: either Ukwabelana, a sample of the isiZulu national corpus, or the news items corpus.

4.1 The corpora

A corpus is "a collection of pieces of language text in electronic form, selected according to external criteria to represent, as far as possible, a language or language variety as a source of data for linguistic research." [17]. A corpus is thus useful for the development of Human Language Technologies such as spellcheckers. Three corpora were used in this experiment, which functioned both as training dataset and testing dataset.

The Ukwabelana Corpus (henceforth UC) [18] is an annotated isiZulu corpus, which comprises around 100 000 common isiZulu word types, and 30 000 isiZulu sentences compiled from isiZulu fictional works and the bible. The UC has a total of 288 106 running words and 87033 unique words. The second corpus is a sample of the isiZulu National Corpus (henceforth INC), which is being compiled at the University of KwaZulu-Natal [10]. The INC is a monitor corpus. This means it is constantly (daily or weekly) supplemented with fresh material and keeps increasing in size [12]. To illustrate the point, the INC was piloted in November 2014 at a modest size of 1.1 million words. By the end of November 2015 it had since grown to 8.3 million words. The sample corpus for this experiment was thus taken from the INC (henceforth Sample INC) and comprises of selected isiZulu grammar texts. The Sample INC has a modest set of 538 732 running words and 33020 unique ones. The last corpus used in the experiment is the news item corpus (henceforth NIC). The NIC is the smallest of the three corpora, with 21250 words, of which 9587 are unique ones. The NIC was compiled by copying the top 10 front-page articles from the online news sources *Isolezwe* and *isiZulu.news24*. This was done over a period of six different days between August and September 2015. The corpora have a type-to-token ratio of 0.30, 0.06, and 0.45 respectively.

4.2 Experimental design

We used 10-fold cross validation to partition each corpus into a training dataset and a testing dataset. We randomly split the corpus into ten sets (folds) of equal size. Then we carried out 10 experiments where we used 9 folds for training and unique words from the remaining one for testing. Each data split is used for testing once. Assuming that the corpus contains zero incorrect words, we intentionally introduced 46 known wrongly spelled words to each testing dataset in each experiment. The spellchecker was fed with the same wrongly spelled words as testing rotates across the 10 folds. The performance of the system on each test was evaluated using a confusion matrix and the average was computed from the

measures of all 10 tests to find the accuracy rate. In the confusion matrix for the classification of results, depicted in Table 1, true positives (TP) are those words where the spellchecker predicted a correct spelling and the word is correctly spelled, true negatives (TN) are the correctly identified misspelled words, false positives (FP) are words where the spellchecker predicted correct but the words are misspelled, and false negatives (FN) are correctly spelled words predicts as wrongly spelled. From this, we compute the accuracy of the spellchecker, with *N* the Total number of words in the test dataset, as (TP+TN)/N, the misclassification as ((FP + FN)/N), the true positive rate ((TP)/(TP+FN)) and true negative rate ((TN)/(FP+TN).

Table 1. Confusion matrix for the classification of results

	Correctly spelled	**Wrongly spelled**
Correct spelling predicted	True Positives (TP)	False Positives (FP)
Wrong spelling predicted	False Negatives (FN)	True Negatives (TN)

The second phase of our experiment was to test with the entire unique word set for each corpus using spellcheckers trained with the other 2 corpora.

All tests were run using different thresholds. The threshold is the probability at which we still consider character compositions within the word to be correct. These tests were run using both trigrams and quadrigrams to see where the spellchecker performs the best.

5. Results

We first summarise the results for trigrams versus quadrigrams, and subsequently report on the cross-corpora training and usage.

5.1 Trigrams versus quadrigrams

Table 2 and Table 3 present the average confusion matrices for trigrams and quadrigrams. In both graphs, it appears that smaller thresholds result in better outcomes for UC. This was because the proportion of correctly spelled words in the testing dataset was much higher than that of wrongly spelled words. What the results showed before averages were computed was that smaller thresholds result in more wrongly spelled words being flagged as correct. This means that the threshold should not be too high or too low. Despite the unbalanced quantities of correctly spelled words and incorrectly spelled words, the performance of the spellchecker tested with UC was still acceptable, with accuracy rate above 50% on all thresholds. This is because UC contains many high frequency words. In the graph for the Sample INC, the spellchecker performs best at the threshold of 0.003 and had 67% accuracy rate. This is a relatively big corpus and could perform better but it

Table 2. 10-fold cross-validation results for the three corpora, using trigrams.

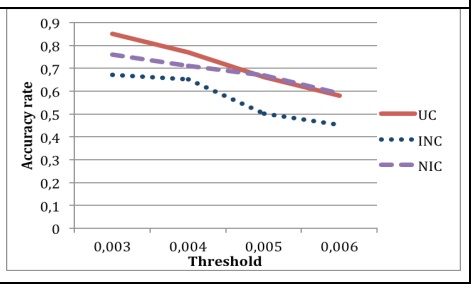
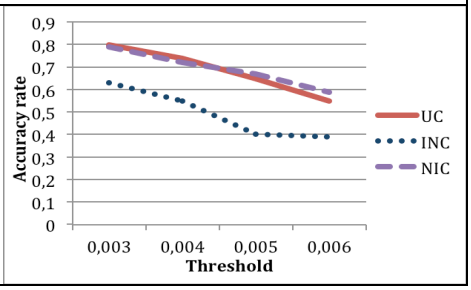
Threshold	UC corpus		Sample INC		NIC corpus		Performance accuracy rate
0.003	84.07 15.49	0.01 0.03	64.96 34.08	0.03 0.12	82.80 14.01	0.05 0.27	
0.004	75.84 22.73	0.01 0.03	58.30 41.16	0.03 0.12	76.50 19.92	0.05 0.27	
0.005	65.72 33.85	0.01 0.04	48.14 51.33	0.02 0.13	70.98 25.82	0.05 0.28	
0.006	57.53 42.03	0.01 0.04	43.44 56.02	0.02 0.13	61.17 33.50	0.05 0.28	

Table 3. 10-fold cross-validation results for the three corpora, using quadrigrams.

Threshold	UC corpus		Sample INC		NIC corpus		Performance accuracy rate
0.003	79.14 20.44	0.03 0.04	62.15 36.40	0.02 0.13	76.80 19.96	0.04 0.28	
0.004	73.94 25.59	0.03 0.03	54.66 43.69	0.02 0.13	69.50 26.78	0.04 0.28	
0.005	64.12 36.52	0.03 0.03	47.15 52.51	0.02 0.13	63.51 33.29	0.03 0.29	
0.006	56.23 44.99	0.02 0.04	40.61 57.89	0.02 0.14	56.18 40.63	0.03 0.30	

mostly contains infrequent words and this has affected performance. For trigrams, the accuracy approximately levels out at 0.004 but this does not happen with quadrigrams. The latter encodes a higher level of information content so smaller probabilities are more indicative of correctness than lower-level n-grams. For the NIC, the spellchecker performed best at the threshold of 0.003 and had 76% accuracy rate.

5.2 Training with one corpus, testing on another

Cross-corpus training and test results are summarised in Tables 4-6. Due to space limitations, the main performance results are shown for trigrams only, as they yielded the better results.

UC is arguably out-dated and was tested mostly with new words from the other corpora that are being introduced as language evolves; for instance, *yibuxakalala* and *adumbe* in the UC and *iFacebook* in the NIC, and multiple verbs miss a final vowel. The performance of the spellchecker for the given training dataset was therefore low, with accuracy rate of 54% at the threshold of 0.003 for the Sample INC. For the NIC, the performance of the spellchecker is acceptable: it stays above 50% for all tested thresholds and performed best at the threshold of 0.003 with 70% accuracy. Testing with this dataset gave good results but, because of the small size of the training dataset, the performance decreases with increase in thresholds. The spellchecker based on the Sample INC is accurate for the given training dataset and testing dataset, yielding above 80% accuracy for all thresholds. The other factor is possibly the lexical content of the testing corpus. Most words are short. This reduces the chance of finding a rejected sequence using character processing. In addition, the content of both the training dataset and the testing dataset is new evolving content in isiZulu.

Table 4. Results of Ukwabelana tested with IsiZulu National Corpus, and news item corpora.

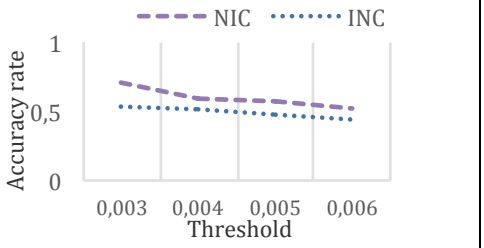
Threshold	Training dataset: UC				Performance accuracy rate
	Testing dataset: Sample INC		Testing dataset: NIC		
0.003	53.97 45.89	0.03 0.11	70.01 29.51	0.10 0.37	
0.004	51.05 48.82	0.03 0.11	58.14 41.38	0.10 0.37	
0.005	46.78 53.08	0.03 0.11	56.76 42.76	0.09 0.38	
0.006	44.12 55.75	0.03 0.11	51.45 48.07	0.09 0.38	

Table 5. Results of IsiZulu National Corpus tested with Ukwabelana Corpus, and news item corpora.

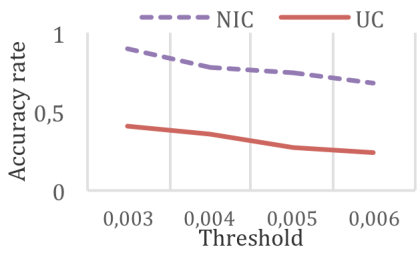
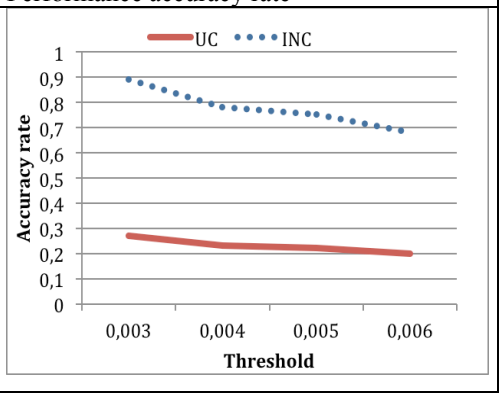
Threshold	Training dataset: Sample INC		Performance accuracy rate
	Testing dataset: NIC	Testing dataset: UC	
0.003	88.93 0.10 10.59 0.37	41.43 0.01 58.52 0.05	
0.004	77.40 0.09 22.12 0.38	35.07 0.01 64.87 0.04	
0.005	74.17 0.08 25.35 0.39	28.05 0.01 71.90 0.04	
0.006	68.06 0.08 31.47 0.40	24.01 0.01 75.93 0.04	

Table 6. Results of News item Corpus tested with Ukwabelana Corpus, and isiZulu national Corpus.

Threshold	Training dataset: NIC		Performance accuracy rate
	Testing dataset: Sample INC	Testing dataset: UC	
0.003	88.56 0.02 11.30 0.12	26.94 0.01 73.01 0.04	
0.004	85.25 0.02 14.61 0.12	22.95 0.01 79.99 0.04	
0.005	84.45 0.02 15.41 0.12	21.58 0.01 78.37 0.05	
0.006	83.31 0.02 16.56 0.12	20.17 0.01 79.78 0.05	

5.3 Recall: comparison with related works

While accuracy may be a better measure, there are two related works [4,14], which report on recall of the spellchecker. Our calculation of recall is to divide the number of true positives by the total number of true positives plus false negatives, which we assume [4,14] used as well. Bosch and Eiselen (BE05) [4] used an unspecified corpus of 225000 words and a set of 88 regular expressions on an unspecified text [4]. Prinsloo and De Schryver (PS04) use a lexicon of the top-600000 isiZulu words (provenance unknown) together with a set of rules that are "clusters of circumfixes", which they tested against the document "What is the African National Congress?" [14]. Our results and the comparison are shown in Table 7. Thus, our highest percentage of the n-gram based spellchecker seems to be as good as BE05 and PS04's lexicon-only results.

Table 7. Lexical recall with n-grams compared to related works; t = test; RE = regular expression; BE05: Bosch & Eiselen, 2005 [4]; PS04: Prinsloo & De Schryver, 2004 [14].

n-grams									BE05	PS04
	10-fold		Train UC		Train NIC		Train Sample INC		Lexicon: 89	Lexicon: 90
	3-grams	4-grams	t. NIC	t. INC	t. INC	t. UC	t. UC	t. NIC		
UC	85	80				30	41		Lexicon + REs: 95	Lexicon + circumfixes: 97
S. INC	66	63		54	89					
NIC	86	79	70					89		

6. Discussion

The results obtained with trigrams are encouraging, and constitutes, to the best of our knowledge, the first systematic attempt to create an n-gram based isiZulu spellchecker. De Schryver and Prinsloo [16] explored n-grams, noting that an improvement of at least 10%

lexical recall should be achievable with n-grams, over the 58% they obtained with wordlist-only technology in their experiment testing against a random *Bona* magazine article. With up-to-date training data of substantial size using n-grams, this can turn out much higher or slightly lower, depending on the corpus the spellchecker is trained with and on the corpus it is tested with. For recall, our results with the n-grams based spellcheckers demonstrate that success correlates with type-to-token ratio and content of the corpus, rather than size: 1) both UC and NIC perform better than the Sample INC in the 10-fold cross validation accuracy and recall, and 2) both the Sample INC and NIC have more compatible content cf. the older texts in UC, hence better cross-corpus results.

At first glance, our results do not appear to compare favourably with PS04 and BE05, where their lexicon-only recall is slightly better than our best lexical recall. PS04's wordlist—assuming to be types—is larger, which may explain the 1% difference there, but not for SE05's lexicon. SE05's lexicon is manually curated, whereas this is not the case for our corpora: if there is an error in the corpus, then this propagates to the spellchecker; e.g., UC's *wuluwulu* is archaic and not used in contemporary everyday language, *efilidi* strikingly rare, *ulueunda* is, perhaps, an OCR error (definitely incorrect with the successive vowels), and there are multiple verbs that miss the final vowel. Further, they mostly did not inject non-Zulu words to verify and lack cross-fold averaging to compare with. That is, we examined the more important discriminatory power cf. just TP+FN lexical recall. Unfortunately, the PS04 and BE05 corpora are not available to test with. Other than that, there are options to investigate further the effects of the data on such a data-driven approach, such as using the whole INC and constructing a wordlist of, say, 400000 and of >600000 words and rerun the experiments.

6. Conclusions

The spellchecker is accurate in detecting words that do not occur in the training corpus. All three corpora used in n-fold cross-validation performed well when tested with their fragments. For trigrams, at a threshold of 0.003, the Ukwebalana corpus (UC) had an accuracy rate of 84%, the subset of the isiZulu national corpus (INC) had an accuracy rate of 65%; and the news items corpus (NIC) had an accuracy rate of 83%. Testing them with each other demonstrated issues with reliance on a single corpus: an out-dated corpus can lead to poor performance. Notably, testing both corpora with UC resulted in an accuracy rate below 50%. Conversely, the UC had a 54% accuracy rate when tested with the Sample INC and 70% when tested with the NIC, whereas the NIC had 89% accuracy rate when tested with the Sample INC and vv. 89% as well. The most updated corpora are preferable, with a threshold of 0.003. The spellchecker performed slightly better with trigrams than with quadrigrams. The probability of finding quadrigrams of a word was lower than the probability of trigrams. This increased the number of false negatives for each.

The main lessons learnt from this investigation are twofold. First, a data-driven approach toward spellchecking is indeed feasible for at least isiZulu, and, by extension of the approach that is essentially language-independent, all Bantu languages. Second, the accuracy of such a spellchecker depends on the text corpus used for training the model as well as on the text document or corpus with which it is tested.

Next steps may involve extending the corpora to possibly obtain better results. It is a potential future addition to the spellchecker to improve its capabilities, such as providing suggestions (though before everything else, error detection should be working fully). A further improvement could be to augment the n-grams approach with a theory-driven linguistic model, which will help check whether infrequent words follow language rules. Another option is to build an extension to the model to detect unlikely combinations of words.

Acknowledgements. This work is based on the research supported in part by the National Research Foundation of South Africa (CMK: Grant Number 93397; HS: Grant Number 88209) and the UKZN Corpus Development Grant (LK).

References

[1] I. Anthony, A.P.C. Charibeth, C. Chan., V.J. Querol. SpellCheF: Spelling Checker and Corrector for Filipino. Journal of research in science, computing, and engineering. Vol. 4 No. 3 December 2007.

[2] S. Basheer, L. Sindhu. Survey of Spell Checking Techniques for Malayalam: NLP. International Journal of Computer Trends and Technology (IJCTT) – volume 17 Number 4 Nov 2014.

[3] B. Bidyut, A. Chaudhuri. A simple real-word error detection and correction using local word bigram and trigram. Proceedings of the Twenty-Fifth Conference on Computational Linguistics and Speech Processing (ROCLING 2013).

[4] S.E. Bosch, R. Eiselen. The effectiveness of morphological rules for an isiZulu spelling checker. S.A.J.Afr.Lang, 2005, 1: 25-36.

[5] F.J. Damerau. A technique for computer detection and correction of spelling errors, Comm. ACM 7(3):171-176, 1964.

[6] D. Jurafsky, J. Martin. Speech and Language Processing, Pearson. 1992.

[7] R. Morris, L.L. Cherry. Computer detection of typographical errors, IEEE Trans Professional Communication, vol. PC-18, no. 1, pp. 54-64, March 1975.

[8] H. Heidorn, K. Jensen, L.A. Miller, R.J. Byrd, M. Chodorow. 1982. The EPISTLE text-critiquing system. IBM Systems Journal, 21:305–326.

[9] G. Hirst, A. Budanitsky. Correcting real-word spelling errors by restoring lexical cohesion. Natural Language Engineering, 11(1):87–111, March 2005.

[10] L. Khumalo. Advances in Developing corpora in African languages. Kuwala, 2015, 1(2): 21-30.

[11] K. Kukich. Techniques for Automatically Correcting Words in Text. ACM Computing Surveys. Volume 24, Number 4. pp. 377-439. December 1992.

[12] T. McEnery, R. Xiao, Y. Tono. Corpus-based Language Studies: An advanced resource book. London: Routledge. 2006.

[13] T. Pirinen, S. Hardwick. Effect of language and error models on efficiency of finite-state spell-checking. In Proceedings of FSMNLP, Donostia–San Sebastían, Spain, 2012.

[14] D.J. Prinsloo, G.-M. de Schryver. Spellcheckers for the South African languages, Part 2: the utilisation of clusters of circumfixes. S.Af.J.Af.Lang. 2004, 1: 83-94.

[15] E.M. Riseman, A.R. Hanson. A contextual post-processing system for error correction using binary n-grams, IEEE Trans Computers, vol. C-23, no. 5, pp. 480-493, May 1974.

[16] G.-M. de Schryver, D.J. Prinsloo. Spellcheckers for the South African languages, Part 1: the status quo and options for improvement. S.Af.J.Af.Lang. 2004, 1: 57-82.

[17] J. Sinclair. Corpus and Text: Basic Principles. Wynne, M. (Ed.). Developing Linguistic Corpora: A Guide to Good Practice: 1-16. Oxford: Oxbow Books. 2005.

[18] S. Spiegler, A. van der Spuy, P. Flach. Ukwabelana - An Open-source Morphological Zulu Corpus. Proceedings of the 23rd International Conference on Computational Linguistics (COLING, 2010), 1020–1028

[19] A. Tavast, M. Koit, K. Muischnek. 2012. Human Language Technologies: The Baltic Perspective. Proceedings of the Fifth International Conference Baltic HLT 2012. IOS Pres BV. Netherlands.

[20] O.A. Vale, A. Candido Jr, M.C.M. Muniz, C.G. Bengtson, L.A. Cucatto, G.M.B. Almeida, A. Batista, M.C. Parreira, M.T. Biderman, S.M. Aluísio. Building a large dictionary of abbreviations for named entity recognition in Portuguese historical corpora. In LATECH2008. Paris: ELRA, v. 1. p. 1-10, 2008.

[21] R.A. Wagner, M.J. Fischer. The string-to-string correction problem, Journal of the Association for Computing Machinery, 21, 168-173, 1974.

[22] A. Wasala, R. Weerasinghe, R. Pushpananda,, C. Liyanage, E. Jayalatharachchi. A Data-Driven Approach to Checking and Correcting Spelling Errors in Sinhala. The International Journal on Advances in ICT for Emerging Regions. 2010 3 (1): 11 – 24.