Spellchecker for isiZulu and isiXhosa

Nthabiseng Mashiane
Department of Computer Science
University of Cape Town
Western Cape
mshnth009@myuct.ac.za

Frida Mjaria
Department of Computer Science
University of Cape Town
Western Cape
mjrfri001@myuct.ac.za

Siseko Neti
Department of Computer Science
University of Cape Town
Western Cape
ntxsis001@myuct.ac.za

ABSTRACT

With Nguni languages increasing their presence in the digital spectrum, there is an increasing need for spellchecking tools for Nguni languages. Very few spellcheckers exist for Nguni languages. In addition, these spellcheckers are limited in terms of scope, accuracy and functionality. According to the 2011 census, approximately 8.1 million people speak isiXhosa which accounts for about 16% of the population of SA. This is second to 23% of isiZulu speaking people in South Africa. With the increase in digitization of African languages, spellchecking tools are becoming more and more paramount in ensuring that documents written in these languages are free from typographical errors. Pretorius and Bosch [2003] state that people are generally comfortable and prefer to interact with technology in their native language.

In this proposal, we describe how we can improve on the number of spellcheckers that exist for Nguni languages by developing an error detector for isiXhosa using two approaches. The first approach will be using a statistical approach and n-grams and the second approach will be using morphological rules of the language and Finite State Automata (FSM). These two approaches will be compared to gage their feasibility and efficiency in developing an isiXhosa error detector. We also describe an approach to developing an error corrector using a statistical approach for isiZulu that could be integrated with the existing isiZulu spellchecker developed by Ndaba et al. [2016].

Keywords

Spellcheckers; Error correction; Error detection; Nguni languages; Natural Language Processor; Morphological analyser; Statistical approach; Bayesian approach

1. PROJECT DESCRIPTION

With the increase in text-based communication (email, social media messaging, voice recognition, etc.), spelling error detection and correction of words has become increasingly important [Farra et al. 2014; Kukich 1992; Samanta and Chaudhuri 2013]. Although many advances have been made in spellcheckers for languages such as English, very few spellcheckers exist for Nguni languages.

Currently, there are two spellcheckers that exist for Nguni languages developed by Ndaba et al.[1] [2016] and spellchecker.net. [2] Ndaba et al.'s spellchecker is a standalone isiZulu spellchecker that can only perform error detection of misspelled words. Spellchecker.net has a web-based spellchecker that offers error detection and correction of misspelled words for various languages, including all Nguni languages spoken in South Africa. Ndaba et al. [2016] uses data-driven statistical models and n-grams and can only detect misspelled words. The spellchecker has an accuracy rate of 89%.

We aim create an error corrector that will be integrated with the existing Ndaba et al.[3] [2016] error detector module to provide error correction functionality. We also aim to create an isiXhosa spellchecker which provides error detection only using two approaches, the statistical approach and the rule-based approach.

This proposal will discuss error detection and error correction techniques, the aim of the proposal, our methodologies and the overall project plan.

2. RELATED WORK

There has been quite a number of projects and/or research done in the field of spellcheckers for agglutinative languages. We have reviewed some of these projects for a better understanding and knowledge about the development of spellcheckers for such languages.

2.1. ERROR DETECTION TECHNIQUES

2.1.1. Knowledge- based approach

In an attempt to create a spellchecker for Quenchua which is a strongly agglutinative, suffix language from South America. a morphological analyser which used the XFST tool was used and was reported to be slow [Rios 2011]. Morphology refers to the study of the internal structure of words, and the systematic form-to-meaning correspondence of words, which deals with ways in which words are formed [Booij 2012]. Since the

[1] https://keet.wordpress.com/2016/11/11/launch-of-the-isizulu-spellchecker/

[2] https://www.spellchecker.net/africa_zulu_spell_checker.html

[3] https://keet.wordpress.com/2016/11/11/launch-of-the-isizulu-spellchecker/

spellchecker was slow another attempt based on the error metric of Levenshtein distance.

Katushemererwe [2010] describes the implementation of Finite State methods using the Xerox Finite State Tools (XFST) in analysing the morphology of the Runyakitara language which is an agglutinative Bantu languages from Uganda. Katushemererwe [2010] reported the tool to be 80% accurate thus showing that using this approach and XFST will be beneficial. Theron and Cloete [1997] show that isiXhosa should not be treated as a simple non-agglutinative language as the tool created based on this assumption will only work for some parts of the language. To show this Theron and Cloete [1997] analysed rules pertaining noun-locative pairs.

[Clark and Araki 2011] highlights the problem faced in the informal/casual writing of English which most often does not conform with the rules of spelling, punctuation and grammar. As a solution to this problem, automated tokenization, word matching and replacement techniques were used in combination with a high- quality, large scale, manually compiled database. In this program, firstly, this technique works by taking the user input and tokenizing it using regular grammar rules defined in PyParsing (another approach for defining grammars other than the lexc/yacc for defining regular expressions) and then they check the tokenized word against their database of defined words. This works well for English but may not work for isiXhosa due to the huge grammar difference between isiXhosa and English.

2.1.2. Statistical approach

Spellcheckers in South Africa have mainly been focused on non-word error detection. Non-word spelling error detection consists of two techniques namely, dictionary-lookup and n-gram analysis [Liang 2009]. A non-word is “a continuous string of letters of an alphabet which does not appear in a given word list or dictionary or that is an invalid word form” [Liang 2009].

Binary n-grams have been used successfully for Optical Character Recognition (OCR) applications and for spellchecking. probabilistic n-grams, where n is greater than two. These n-grams were used as they are richer in information [De Schryver et al. 2004].

Bell and Bird [2000] devised a language model which analyzes the orthographic structure of Bantu languages. Bosch et al. [2007] made some improvements to this model altering the way in which the breakdown of a word is captured in such a way that the efficiency of spellcheckers improved. In addition, the improved version of the model grouped nouns and verbs, looked at the significance of reflexive forms and inflections in a language.

Another tool, ‘Umqageli’ meaning ‘diviner’ in isiZulu is a spellchecker which works for SeSotho sa Leboa, SeSotho, Setswana and TshiVenda created by Maniacky [2003]. Maniacky [2003] reported that the tool performed poorly for the Nguni language group i.e. isiZulu, isiNdebele, isiXhosa and siSwati. Ndaba et al. [2016] successfully created a spellchecker detector model for isiZulu which uses trigrams to analyse a word and check for correctness. Ndaba et al. [2016] noticed that the quality of the corpora affects the quality of the spellchecker. It is therefore imperative that we use documents which are well-structured and from a reliable source corpora so that we obtain accurate results. Sources of language corpora are now widely available online and to ensure that these corpora are of good quality, one could use a morphological analyser as a preprocessor for the corpora [De Schryver 2002]. There is an online tool from webcorp.org[4] which allows you to create a wordlist from any online source you find. This tool can be used to supplement the corpus we are to receive from the African Languages department of the University of Cape Town. The issue with creating a corpus from a web page is that the corpus will have to be pre-processed as it might contain words from other languages besides isiXhosa.

2.2. ERROR CORRECTION TECHNIQUES

2.2.1. Statistical approach

The noisy channel model for error correction was first proposed by Mays et al. [1991] from IBM and Kernighan et al. [1990] from AT&T Bell Labs. Kernighan et al. [1990] developed a spellchecker, called *Correct*, which takes misspelled words detected by the Unix spell program and finds candidate corrections using a noisy channel model. They assume that a typist is aware of the word that they wanted to spell, but some noise was added to the word via the keyboard while typing this word causing it to become distorted. *Correct* achieved an accuracy rate of 87%. Similarly, Whitelaw et al. [2009] used the same approach in building confidence classifiers to determine thresholds for error detection and auto correction,

Gupta and Sharma [2015] implemented a Bayesian approach in their spellchecker. The spellchecker calculates the probability of context words surrounding a misspelled word as well as the probability of the context word occurring together with the misspelled word. These probabilities were used in determining candidate corrections for the misspelled words.

3. PROBLEM STATEMENT

Very few spellcheckers exist that can detect and correct spelling errors in Nguni languages. With 2 Nguni languages, isiZulu and isiXhosa, being the most widely spoken languages in South Africa and with the increase in digital text-based communication, the

[4] http://www.webcorp.org.uk

development of spellcheckers for Nguni languages is crucial. There are currently no standalone spellcheckers for isiXhosa. For isiZulu, the current spellchecker developed by Ndaba et al. [2016] does not perform error correction.

4. AIMS AND RESEARCH QUESTION

4.1 Aims

The aim of this project is to provide the African Language department at the University of Cape Town with an isiXhosa spellchecker which performs error detection which currently does not exist. We also aim to investigate whether an error detector implemented using a statistical based approach is more accurate compared to a rule-based approach. In addition to this, the project aims to investigate whether a statistical approach can be utilized in successfully implementing an error corrector for the existing isiZulu spellchecker developed by Ndaba et al. [2016].

4.2 Research Questions

These are the main research questions which need to be answered in order to realise the project objectives.

- Can a statistical model based error detector for an isiXhosa spellchecker achieve an accuracy of 85% or more?
- Is the rule-based approach more accurate at detecting misspelled words compared to the statistical approach?
- Can the error corrector correct more than 85% of the misspelled words in an input text using an error model based on Bayes' rule

The core requirements of the isiXhosa spellchecker using a statistical approach are as follows:

- The spellchecker must be able to identify and highlight incorrect words in isiXhosa.
- The solution must have some sort of interface for the end user.

5. PROCEDURES & METHODS

This section discusses the procedures and methods we intend to use to answer the abovementioned research questions in Section 4. We will state, explain and justify our selected approaches and highlight anticipated problems when implementing our methods. In addition, we will examine the way the project will be evaluated to ensure that it addresses the research questions.

5.1 Development procedures, methods and practices

Our project is both a software engineering and a research project. The software engineering section of our project is developing an isiXhosa error detector using a statistical based approach. The research section of our project is made of 2 parts: error detection and error correction. An error detector for isiXhosa will be implemented utilizing the morphological rules of isiXhosa. The performance will then be compared to the statistical approach to determine whether a rule-based error detector is as efficient as a statistical approach. The second part of the research section is used to investigate if an error corrector can be developed using a statistical approach for isiZulu. The error corrector will then be utilized with the existing isiZulu spellchecker to determine whether the performance of this spellchecker can be improved. We will measure performance improvement according to the percentage of misspelled words that the error corrector is able to provide the target word in the candidate corrections. This percentage should be 85% and above.

The rule-based approach involves a part where we first have to understand the language morphological rules and then from there we can actually create or determine the finite state rules that will be used by the error detector. Due to the above reason [Pettigrew 2000] proposes that this be a qualitative research using the grounded theory approach. We first started by doing a literature review and then from the studied literature we were able to tell that we will use finite state networks to represent the isiXhosa language rule. For these finite state networks [Karttunen 2005] states that every path encodes a string or an ordered pair of strings where the path is represented as a combination of states and arcs (arrows representing transition from one state to the other). For an agglutinative language like isiXhosa, there can be more than ten thousand states and arcs but using a transducer we can combine arcs with the same origin and destination states into a single multi-labelled arc. We then will use a regular expression to describe the language and then compile this regular expression into a network [Karttunen 2005].

The statistical approach will follow the development of the isiZulu error detector created by Ndaba et al, [2016]. An error detector module will be created wherein an n-gram tree as well as frequency statistics will be created and stored from the text corpus. The corpus will be compiled from documents provided by Dr Motinyane-Masoko from the African languages department at the University of Cape Town. The error detection will detect both correct and incorrect words, flagging the incorrect words and leaving correct words as is. A threshold calculated using the word frequency in the text and the frequency of the same word in the corpus. This threshold will be used to determine the likelihood of a word being correct or incorrect. The development process will follow an agile methodology.

For the error corrector, trigrams will be constructed from corpora from the University of KwaZulu Natal's linguistics department. These trigrams as well as the frequency at which each trigram occurs in the corpora will be stored in a database. Trigrams with a frequency below a given threshold will not be included in the database. The threshold is not known at this point in time and will be determined empirically. Each word in the dictionary will have a vector of trigrams associated with it and will make up the vocabulary of the error corrector. In order to locate misspelled words in a text, each word in the text will be broken down into its constituent trigrams. The trigrams will then be compared to the database of trigrams. Trigrams that are not found among those located in the database will cause the word to be flagged as a misspelled word. In order to correct this word, a noisy channel model, based on Bayes' rule, will be implemented. A list of words from the dictionary that contain most of the trigrams in the misspelled word will be populated. These words will be used with the noisy channel model to determine candidate corrections for the misspelled word. The noisy channel model assumes that a misspelled word was originally a correctly spelled word that became distorted by passing through a noisy channel that made the word difficult to recognize. The "noise" is from typographical errors such as insertion, deletion, substitution and/or transposition errors [Jurafsky and Martin 2016]. The model aims at finding a correct word w such that $P(w|x) = argmax\, P(x|w)\, P(w)$, where w is a word in the vocabulary and x is a misspelled word from the input text. Words that are determined to be candidate corrections for a misspelled word will be ordered according to their probabilities.

The development method that we intend on utilizing is an iterative method that includes weekly meetings with our supervisor, Maria Keet. We also intend on frequently meeting and regularly communicating with Dr. Mantoa Motinyane-Masoko from the African languages section of the University of Cape Town as well as Dr. Langa Khumalo from the linguistics department of the University of kwaZulu Natal. Dr. Mantoa Motinyane-Masoko is our client for the isiXhosa spellchecker and subject matter expert in isiXhosa, and Dr. Langa Khumalo is the subject matter expert in isiZulu.

5.2 Evaluation measures and Acceptance Testing

Our evaluation methods for error detection and error correction based on the accuracy rate achieved by each component.

For the software development component of our project, the accuracy of the error detector will be measured through using an automated tests and the interface of the spellchecker will be evaluated through user testing. The rule-based approach isiXhosa error detector as well as the isiZulu error corrector will use an automated system to test their accuracy.
We aim to achieve 85% accuracy for both isiXhosa error detectors and for the isiZulu error corrector as well. We will then use the much stricter confusion matrix approach in checking for the accuracy of the error detector and error corrector. This approach involves using an input text with known misspelled words where the error detector's output will be compared with the amount of known errors in order to determine its accuracy.

6. ETHICAL, PROFESSIONAL & LEGAL ISSUES

Any experiments conducted in this project will mostly not involve external users. The software engineering section will require usability testing to evaluate whether the interface is user friendly. This needs ethical clearance which has to be formally applied for and approved by the university before the usability testing commences. In addition, the documents received from the African Languages section of the University of Cape Town will be used to build a corpus and to test the functionality of the spellchecker and will not be published. The isiZulu corpora obtained from Dr Khumalo from the UKZN linguistic department will also not be published. Overall, we do not have any ethical issues.

The outcomes of this work will be made open-source and thus anyone can improve or advance this work without any legal issues involved.

7. ANTICIPATED OUTCOME

7.1 Expected Outcome

We hope to develop a tool which will assist the isiXhosa division in the African Languages department of the University of Cape Town and in the broader sense, a tool to aid anyone writing text in isiXhosa. In addition, we hope to see how the statistical approach of creating a spellchecker for isiXhosa compares to the rule-based approach.

We expect the statistical based approach isiXhosa spellchecker and the rule-based approach to have 85% detection accuracy.

7.2 Key success factors

In order for our project to be considered a success, the following conditions need to be met:

- isiXhosa spellchecker using the rule-based approach detects at least 85% of the misspelled words.
- isiXhosa spellchecker using the statistical based approach detects at least 85% of misspelled words correctly.
- Error corrector improves the performance of the isiZulu spellchecker by providing the target word in candidate corrections for at least 85% of misspelled words in an input text.

7.3 Impact

The African languages department at UCT, particularly the isiXhosa division, will be able to enhance the learning experience of their students through the use of the isiXhosa error detector. In addition, the isiZulu spellchecker will be improved with the addition of the error corrector thus offering a more complete spellchecker.

Both the isiXhosa and isiZulu spellcheckers will be open-source thus, anyone writing in either of the languages can use this tool to enhance their writing process.

The isiXhosa and isiZulu spellcheckers will also aid in the preservation of the languages as the language evolves over time.

8. PROJECT PLAN AND WORK ALLOCATION

8.1 Risk and Management Strategies

This section discusses our anticipated issues and difficulties in the project and presents a risk matrix (See Appendix E) which indicates more risks as well as our mitigation and monitoring strategies.

We use both a rule-based and statistical approach for our project and for each approach, we anticipate a few difficulties in our attempt to realise our goals. For the rule-based approach it might be difficult to detect words which are borrowed from another language and do not follow the orthographic rules of isiXhosa or isiZulu. In addition,the punctuation and capitalization of words might affect the accuracy of the spellchecker as correct words may be flagged as incorrect and vice versa.

Similarly, for the statistical approach, we anticipate the capitalization of words might affect the performance of the spellchecker negatively. Furthermore, the recency and reliability of the corpus, the size of the corpus as well as the variety of the lexica in the corpus may affect the robustness of the spellchecker [Ndaba et al. 2016]. Lastly, the threshold which will be used to compare the frequency of words needs to be selected carefully as it will affect the accuracy of the spellchecker.

8.2 Timeline

We utilize a Gantt chart (See Appendix A) to keep track of all milestones and deliverables. The Gantt chart is updated regularly to track the progress made by group members. Each group member has an individual timeline for their project component, which is updated regularly as well (See Appendices B-D).

8.3 Resources required

The rule-based error detector will require the morphological rules (e.g the noun classification classes) for the isiXhosa language to derive the rules that will be used in the finite state automaton. In addition, the Xerox finite state tools will be used. This is subject to change as we see fit during the project.

For the statistical error detector, we will need isiXhosa texts from the African Languages department of the University of Cape Town.

For the error corrector, the isiZulu National Corpus (INC) and isiZulu texts from Dr. Khumalo will be needed for training and testing the error corrector.

8.4 Deliverables

There are three main deliverables for our project:

- An isiXhosa error detector using a rule-based approach and non-deterministic finite state automaton.
- An isiXhosa spellchecker that offers error detection of misspelled words using a statistical model with an interface for the end users.
- An error corrector for isiZulu using a data driven approach. The error corrector will be integrated into the isiZulu spellchecker developed by Ndaba et al. (2016).

Other deliverables for our project include:

- Literature review
- Project proposal presentations
- Internal software feasibility presentation
- Final complete draft of paper
- Project paper final submission
- Project code final submission
- Final project presentation
- Poster
- Web page
- Reflection paper

The deliverables listed above have been highlighted in our Gantt chart (Appendix A).

8.5 Milestones

Our milestones include our Honours project deliverables as well as milestones that we have decided on, which are included in the Gantt chart (Appendix A).

8.6 Work Allocation

The aims of this project will be split equally among three people. Nthabiseng Mashiane will be creating an isiXhosa error detector, Siseko Neti will be investigating the feasibility of using a rule-based approach to create and isiXhosa spellchecker and Frida Mjaria will be creating an isiZulu error corrector. Our Supervisor is Dr Maria Keet who will guide us through the project. Professor Sonia Berman will assess our project deliverables.

All of the supervisor meetings as well as progress meetings must be attended by each group member. In addition, each group member will contribute equally to all written deliverables, presentations and iterative demonstrations.

9. REFERENCES

[1] Clark, E. and Araki, K. 2011. Text normalization in social media: progress, problems and applications for a pre-processing system of casual English. *Procedia-Social and Behavioral Sciences*. 27, 2-11.

[2] Farra, N., Tomeh, N., Rozovskaya, A. and Habash, N. 2014. Generalized Character-Level Spelling Error Correction. ACL. 2, 161-167.

[3] Gilles-Maurice de Schryver, D J Prinsloo. 2004. spellcheckers for the South African languages, Part 1: The status quo and options for improvement. *South African Journal of African Languages*. 24, 1, 57-82. http://dx.doi.org/10.2989/16073610309486351

[4] Gupta, S. A., Sharma S. 2015. Correction Model for Real-word Errors. Procedia Computer Science. 70, 99-106.

[5] Jurafsky, D. and Martin, J. H. 2016. Spelling Correction and the Noisy Channel (draft, 3ed). Speech and Language Processing. DOI=10.1145/146370.146380.

[6] Karttunen, L. 2003. Finite-state technology. *The Oxford Handbook of Computational Linguistics. Oxford University Press: Oxford*. 339-357.

[7] Katushemererwe, F. and Hanneforth, T. 2010. fsm2 and the morphological analysis of Bantu nouns–first experiences from Runyakitara. *International Journal of Computing and ICT research*. 4, 1, 58-69.

[8] Kernighan, M. D., Church, K. W. and Gale, W. A. 1990. A spelling correction program based on a noisy channel model. In Proceedings of the 13th conference on Computational linguistics-Volume 2. Association for Computational Linguistics. 205-210.

[9] Kukich, K. 1992. Techniques for automatically correcting words in text. ACM Computing Surveys (CSUR). 24, 4, 377-439.

[10] Laurette Pretorius and Sonja E. Bosch. 2003. Enabling computer interaction in the indigenous languages of South Africa: the central role of computational morphology. *interactions.* 10, 2 (March 2003), 56-63. DOI=http://dx.doi.org/10.1145/637848.637863.

[11] Mays, E., Damerau, F. J. and Mercer, R. L. 1991. Context based spelling correction. Information Processing & Management. 27, 5, 517-522.

[12] Maniacky,J.2003. Umqageli (Automatic Identification Of Bantu languages).http://www.bantu-languages. corn/en/tools/identification.php (Accessed: 20 May 2017).

[13] Ndaba,B., Suleman, H.,Keet,C.M.and Khumalo,L. 2016. The Effects of a Corpus on isiZulu Spellcheckers based on N-grams in *IST-Africa Week Conference*. 1-10. IEEE. DOI: 10.1109/ISTAFRICA.2016.7530643.

[14] Pettigrew, S.F. 2000. Ethnography and grounded theory: a happy marriage?. *NA-Advances in Consumer Research Volume 27*.

[15] Pieter Theron and Ian Cloete. 1997. Automatic acquisition of two-level morphological rules. In *Proceedings of the fifth conference on Applied natural language processing*. Association for Computational Linguistics. 103–110.

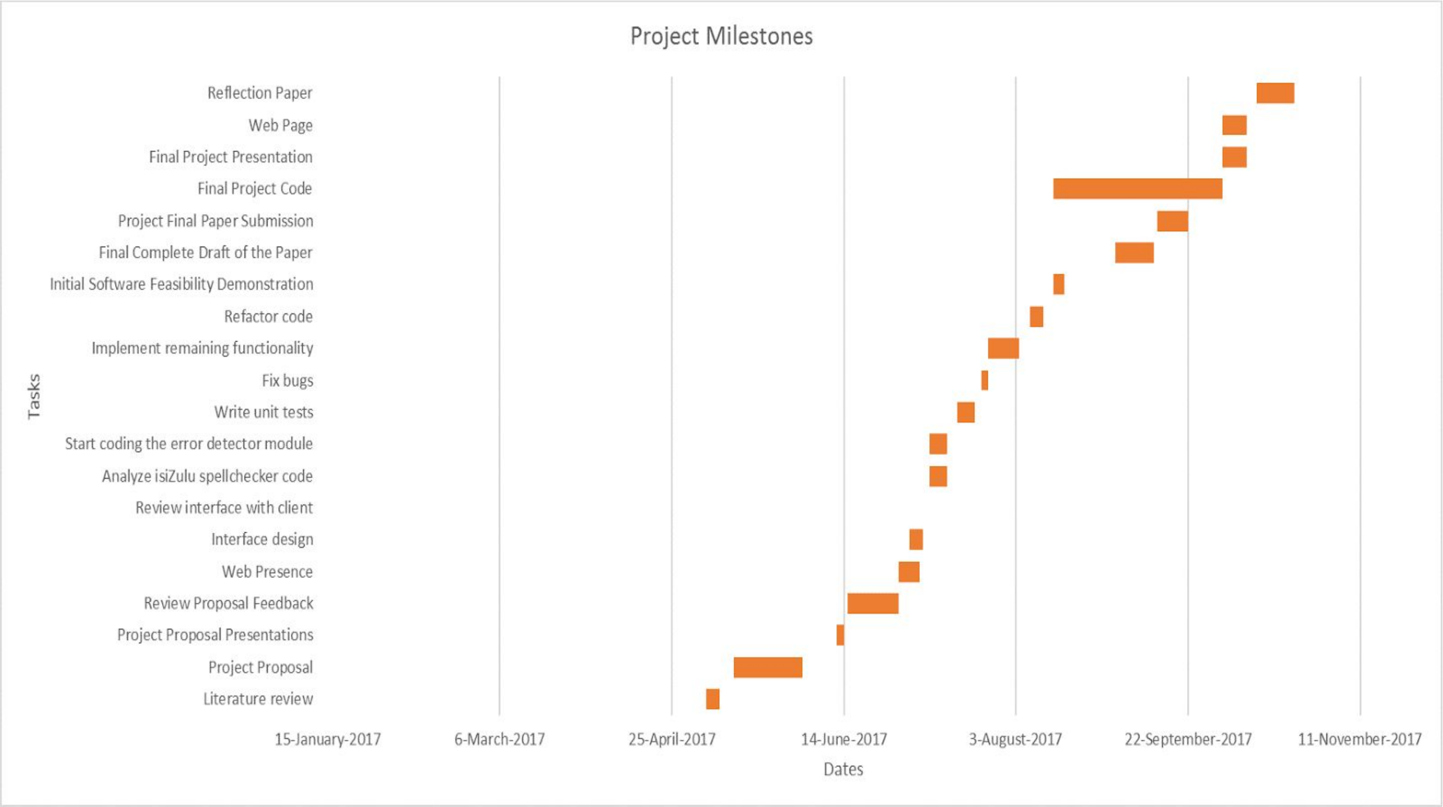
[16] Rios, A. 2011. Spell checking an agglutinative language: Quechua. In *5th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*. 51-55.

[17] Samanta, P. and Chaudhuri, B. B. 2013. A simple real-word error detection and correction using local word bigram and trigram. In *ROCLING*.

[18] Whitelaw, C., Hutchinson, B., Chung, G. Y. and Ellis, G. 2009. Using the web for language independent spellchecking and autocorrection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*. Association for Computational Linguistics. 890-899.

Appendix A

Gantt Chart



Appendix B

Error Corrector Timeline:

Deliverable	Start - End Date	Duration	Milestone	Comments
Corpus Clean-up	6 - 12 July	1 week		*Corpus cleaned. Find someone from the language department to check corpus.*
Language Model construction	13 - 19 July	1 week		
	16 July		***Trigrams constructed***	
	19 July		***Prior probabilities computed***	
Noisy Channel Model (Error Model) construction	20 July - 9 August	3 weeks		*Look at calculating frequency of letters in isiZulu compared to English for extrapolating English confusion matrix to formulate isiZulu confusion matrix..*
	26 July		***Confusion matrix for insertion and deletion error probabilities (EP) constructed and tested***	
	2 August		***Confusion matrix for substitution EP constructed and tested***	
	9 August		***Confusion matrix for transposition EP constructed***	

			and tested	
Feasibility Demo Testing/Code clean-up	10 - 13 August	4 days		
	14 -18 August		***Feasibility Demo complete***	
Empirical testing for threshold	14-19 August	5 days		
	19 August		***Threshold testing complete***	
Final complete draft report write-up	19 August - 11 September	3 weeks and 4 days		
	25 August		***First draft report hand-in***	
	31 August		***Second draft report hand-in***	
	8 September		***Third draft report hand-in***	
	12 September		***Final complete draft report hand-in***	
Final project paper write-up	13 - 21 September	9 days		
	22 September		***Final project paper submitted***	
Final code optimisation and clean-up	23 - 1 October	9 days		
	2 October		***Final Code submitted***	

Appendix C

IsiXhosa Error detector using Statistical approach timeline

Task	**Start date**	**Duration**	**Comment**

Feasibility study [check if we can create a Google chrome plugin]	14/06/2017	2 days	In progress
Design interface	16/06/2017	1 day	Not started
Create interface prototype	16/06/2017	1 day	Not started
Evaluate interface	[Depends on Dr Mothinyane’s availability]	1 day	Not started
Refine interface	[Follows from interface evaluation]	1 day	Not started
Evaluate improved interface	[Follows from refinement of interface]	1 day	Not started
Create error detection module for back-end	19/06/2017	3 weeks	Not started
Create automatic tests	28/06/2017	2 days	Not started
Rest	01/08/2017	3 days	Not started
Create end user usability study	07/08/2017	3 days	Not started
Conduct usability study	[Depends on ethical clearance and correspondence with client]	1 day	Not started
Improve code [refactor, more functionality?]	[Follows from usability study]	4 days	Not started
Documentation of interface design process and evaluation.	17/06/2017 [Intend to start documentation once interface has been created, I don’t have a reason for this, just preference. The documentation will then run parallel with the other tasks]	6 weeks	Not started

Appendix D

IsiXhosa Error Detector Using the Rule Based Approach

Start Date	End Date or Expected Finish Date	Task	Comments/Status
5-Jul-17		Familiarising with Finite State by reading chapter 1 and 2 of the book called Finite State Language Processing by Emmanuel Roche and Yves Schabes	Ch1 Done
6-Jul-17		Finishing with CH2. Playing with JFlap tool	ch2 done. Experimented with JFlap
7-Jul-17		Reading information about the rules of the language. Deciding which ones to focus on.	Reading isiXhosa books
8-Jul-17			
9-Jul-17			
10-Jul-17		Playing with Foma and OpenFST and some other tools that might be useful.	Played with HFST. Foma and OpenFST can be used as backend for the HFST
11-Jul-17		Work on noun rules	Reading books on noun rules
12-Jul-17		Work on noun rules	Confirming the noun rules with and start writing regular expressions

13-Jul-17		Work on Noun Rules	regular expression encoding.
14-Jul-17		Updating the final project proposal.	
15-Jul-17			
16-Jul-17			
17-Jul-17		encoding the rules bit by bit with testing.	
18-Jul-17		Continue with encoding the rules bit by bit with testing.	
19-Jul-17		Reading and Looking at Pronouns (This might change according to what has been found in the on the 7th July Task).	
20-Jul-17		Pronouns rules	
21-Jul-17		writing the pronoun rules as regular expression.	
22-Jul-17			
23-Jul-17			
24-Jul-17		encoding with a bit by bit with testing.	
25-Jul-17		Continue with encoding the rules bit by bit with testing.	
26-Jul-17		Verbs and Irregular Verbs	
27-Jul-17		Verbs and Irregular Verbs	
28-Jul-17		Verbs and Irregular Verbs	
29-Jul-17	13 Aug 17	Vacation	

14 August 2017	18 Aug 17	Initial Software Feasibility Demonstration to the supervisor and second reader	
14-Aug-17	18 Aug 17	Background/Theory Section based on literature review	
15-Aug-17			
16-Aug-17			
21-Aug-17	24 Aug 17	Paper Plan/Scafold , Final Experiment and Writeup	
26-Aug-17			
27-Aug-17			
24-Aug-17	29 Aug 17	Final Testing	
2-Sep-17			
3-Sep-17			
29-Aug-17	5 Sep 17	Outline of complete project report/paper with missing sections explained in 2/3 lines	
9-Sep-17			
10-Sep-17			
5-Sep-17	12 Sep 17	Final complete draft of the project report/paper	
12-Sep-17	22 Sep 17	Final project report/paper submission	
25 September 2017	29 Sep 17	Vacation	
1-Oct-17	2 Oct 17	Project Code to be finalised and submitted	
2-Oct-17	9 Oct 17	Project Demonstration as a group but each member demonstrating its own part.	
2 October 2017	9 Oct 17	Project poster for the group	
18-Aug-17	12 Oct 17	Web page for the project	
16-Oct-17	22 Oct 17	Reflection paper for this part of the project	

Appendix E

Risk Matrix

Risk	**Probability**	**Impact**	**Consequence**	**Mitigation**	**Monitoring**	**Management**
Unable to meet with the language department	**Medium**	**High**	Assumptions will have to be made in terms of requirements of the project, which may be very incorrect causing the project to be a failure as the software will not be used	Discuss and formulate possible requirements from the linguistic department with supervisor	Consistently contact the language department requesting for meetings	Meet and discuss with supervisor to find a way forward.
Unavailable dataset	**Low**	**Medium**	We would have to resort to the use of smaller resources which might deteriorate the quality of the spellchecker	Start compiling a language corpora for isiXhosa from other sources such as the internet	Consistently try to contact the linguistics department.	Look for alternative datasets to use and discuss viability of dataset with supervisor.
Scope creep	**Low**	**High**	Failure to complete the project on time.	Finalize scope and use a gantt chart to plan for the implementation of the scope	Monitor and update progress of project milestones and deliverables in Gantt chart	Reassess and adjust distribution of workload. Consult with supervisor about scope.
Unable to meet deadlines	**Low**	**Medium**	Production of incomplete and incompetent work.	Plan for days where work will not be done (leeway time)	Monitor Gantt chart and communicate regularly with team members	Set Draft hand-in dates with supervisor to ensure specific milestones are reached and progress is being made
Team member drops out	Medium	Medium	One of our three part deliverable will not be realized	Keep the group motivated	Weekly reviews	Distribute workload among other team members and reduce scope of project
Inability to implement a	**Low**	**High**	Project becomes a failure and no	Consult with supervisor	Weekly reviews	Find a different

solution			valuable contribution is made towards spellcheckers for Nguni languages.	on feasibility and progress of project on a regular basis		methodology to complete the project or reduce scope. Consult with supervisor.