Project proposal for Afrispel: Spell checkers for the isiZulu language

Victor Kabine
Balone Ndaba
Ditshego Phogole

1. Introduction

The methodology of spell checker is perhaps one of the oldest and most researched areas of language technology. The first research on spell checkers dates back to the mid20th century. Research is still being conducted on how the spell checker can be improved. The research however is focused on developed languages, languages such as English. There is not much focus on a lot of less-resourced languages, in particular African languages. In South Africa there are 11 official languages and almost all the languages in the country are under-resourced, especially languages belonging to the Bantu language family that is comprised of 400 languages all over Africa [L. Pretorius, 2003].
The language we will be focusing on is the isiZulu language which is spoken by roughly 11 million people in South Africa and the country's total population is roughly 52 million [Spiegler S.2010]. Making use of the three principle methodologies that are involved in building a spell checker, the theory-driven linguistic model, the data driven statistical model and the crowd-sourced dynamic user driven model. We aim to build a project contributes to the initiative of having multilingual support for these languages in office suites, search engines etc.

2. Problem statement

Spell checkers have been researched for so long however the very first fully developed morphological analyser for a language in the Bantu language family was implemented in 1992 (Hurskainen, 1992). It was for the Swahili language. This indicates the lack of resources that these languages have. Tools such as morphological analyzers are scarce for these kind of languages. There should be more resources like these available for the less resourced languages and these languages have to be researched as well. Mistakes in linguistics can sometimes lead to the writer being misunderstood by the reader, tools such as the morphological analyzers (spell checker) can also be used to educate the user on the language and thus more continuous research has to be done to address the problem at hand and try to find a helpful solution.
Unfortunately because research on this area has remained stagnant for a few years, there hasn't been any major findings on how the problem can be addressed. Over the course of time, languages tend to change in their phonetic, semantic, morphological or syntactic features. This is referred to as language change. The changes can be very slight nevertheless the changes can lead to the morphological analyzers failing to fully represent the language. The cause of the language change can vary from cultural environment to the medium that is used for communication, this applies for all of the languages. There has to be continuous research done on the languages in order for the morphological analysers developed from the research to stay relevant to the language in its current

(modern) form.

Most of the research and material on less resourced languages follows the focus of one principle or philosophy to create the morphological analyser either the data-driven statistical approach, the theory-driven linguistic approach or the crowd-sourced dynamic user-driven approach however there hasn't been much research that studied comparison between the three philosophies. It is important to know which of the three philosophies for developing spelling checkers truly works best when dealing with the less developed languages.

3. Research questions

The research questions are generated by looking at the problem statement, from the problem we were able to generate key aspects. The research questions are

a. Can we implement three fully functional spell checkers with two focusing on accuracy and another based on user testing?
b. How accurate is a spell checker?
c. Does crowdsourcing work to build a specification?

4. Objectives

The goal of this research project is to develop a set of spell checkers and evaluate the spell checkers based on accuracy and user experience. The evaluations will vary based on which methodology is used to build the spell checker. There will be three checkers following different methodologies, the theory based linguistic model. The spell checker designed by this model will make use of finite state automata. Finite state automata makes use of language processing and can be used in natural language processing. There are three different properties of a finite state networks, there is the epsilonfree, deterministic, minimal. Which of the properties our finite state spell checker has holds little value and is not that important. We are going to construct a finite state spell checker that makes use of the rules of the language and tests the user's input with the rules in the finite state machine and if the finite state machine does not recognize the input that is given it will flag and indicate that the word specified is incorrect. The finite state automata must also be able to generate suggest possible correct words based on the rules that the machine has on the language. The output that is generated by the machine has to be displayed for the user however the need for a user interface is of little importance and this output can be generated using the command line function.

Other methodologies that will be used in this section include data-driven statistical model and with this methodology, corpus-driven spell checkers will be created. Crowd-sourced dynamic user-driven model will make use of crowdsourcing techniques to build the spell checker. It is the only methodology that will make use of user experience as a form of evaluation and the other two methodologies will be evaluated in terms of accuracy. The crowd-sourced dynamic user-driven model will also make use of the current word list to provide suggestions on the correctness and incorrectness of words in a different way to the other two models.

5. Related works

There has been quite a lot of research that been done in this field. This research will serve as a basis

from which we will conduct our research project. There have been spell checkers (morphological analyzers) creates using the three different philosophies for different languages. We will look at one paper that focuses on each of the three philosophies.
[Wasala, 2010].In this article, the research project aimed at developing a data-driven spell checker based on n-gram statistics. The spell checker was developed for the Sinhala language. The
Spell checker had an accuracy rate of 82%. So this indicates that the accuracy for the spell checker was good.
(Spiegler S, 2010) focuses on the first ever corpus that was built for the isiZulu language, it also discusses the isiZulu morphology and the rules involved in creating the corpus for the language.
(Brabham, 2008) discusses crowdsourcing as a distributed problem solving and production model which has been successfully implemented by different companies. Such as Thread less, iStockphoto and InnoCentive, to name a few. It is a way of exploiting crowd wisdom.
De Clercq and others investigated the use of crowdsourcing for improving readability of text as an alternative to using expert annotators. A comprehensive crowdsourcing tool was developed and tested with non-expert users. They found out that crowdsourcing was just as consistent, although it used non-experts.

6. Methodology

The research methods that have been taken for this project are the literature review and the conceptual modeling. The study will first review the work that has been done on the field and identify the research questions with regard to our project. The study will first review the work that has been done on the field and identify the research questions with regard to our project. Spell checking methods have two main functions. The first one is to check possible misspellings that a user may have committed. The second function of spell checkers is to suggest the users' intended spelling of a misspelled word or at least to suggest a list of candidates in which the target word appears.

6.1. Tools and languages

For the theory-based linguistic model, we are going to make use of the finite state tools for the design and implementation of the spell checker. The tools that are chosen for this project at this point are not final and they are subjected to change if better languages arise. The Xerox finite state tools, they are robust, well documented and they are also freely documented for research however they are not open source [L. Karttunen, 2003]. The other tools that we might look at for the finite state spell checker are the hfst tools and the reason being that they are open source with little to no restrictions however the problem with the hfst tools is that they are still quite new. Both of the compilers run the same files so we will look into which of the tools we end up using. The Xerox compilers are lexc and xfst [L. Karttunen, 2003]. Lexc is used for the morphology, it is also known as a scripting language that is used to define sequences of morphemes as cascades of morpheme lexicons. The resulting script is then compiled into the finite state network that will map between the analysis language to which strings belong and an intermediate language. Xfst is for compiling the final state transducer. It is where morphophonological alteration rules are formulated [L. Karttunen, 2003]. We might also consider using other Xerox compilers such as lookup (for analysis and generation). Other finite state tools that might be used at are the Helsinki finite state transducer technology.

For the data-driven statistical language model, we are going to use a data driven algorithm based on n-gram statistics. The algorithm first tokenizes the input text stream to build a word list. The word list should contains unique words found in the text stream. The words are searched from a pre-compiled list of homophones and valid spelling variants. If a word matches, it is left off for further processing. A word that comprises of more than three syllables is broken into a three syllable sequence at this stage. The second algorithm then checks the n-gram statistics for suggestion evaluation. The words are processed independently. Ngram statistic are pre-compiled and stored in a database such that it is easily accessible to the algorithms. There will be an algorithm used to calculate syllable unigram, syllable bigram and syllable trigram frequencies. 0 frequencies shows non-existence of a syllable in the corpus. The algorithm will then calculate frequencies and check the most frequent from unigram, trigram or bigram models. The hope is that the most frequent suggestion is the closest suggestion. If a syllable is not in the corpus, a different algorithm checks for insertions, deletions and transpositions. All algorithms can be implemented using Java programming language and databases built using Oracle.

For the crowdsource-driven model, we will make use of a crowdsourcing system. Many existing crowdsourcing platforms were explored, from the Mechanical Turk to Stack Exchange and even AskBot. These platforms have relevant features for this context, such as upward and downward voting, dynamic reputation of the users overtime (based on the number of upward and downward votes) and they are well documented. However, they are not open source. Therefore we decided to build our own crowdsourcing sourcing, using open source platforms. We are still exploring these open source platforms to build from, but so far we looking at using Python Django platform. This is because it was used in many of the open source sites we looked at and we are familiar with the language.

7. Project plan

7.1. Project management approach

This project will be supervised by Prof. Hussein Suleman and Dr. Maria Keet and they will provide guidance on the overall structure of the project and how it is run. The project will have three separate parts and there are three project team members in this project. There will be three different spell checkers built from three different philosophies: the theory-based linguistic model, the data-driven statistical model and the crowd-sourced dynamic user-driven model. This will ensure that the work allocation will be equal among the project team members. Each project team member will design their own individual spell checker each making use of a different methodology, the design of these spell checkers will not be interdependent at any stage.

7.1.1 Project roles

Role	Person responsible	Responsibilities
Supervisor	Prof. Hussein Suleman	Give guidance on the nature of research and the standard expected
Co-Supervisor	Dr. Maria Keet	Give guidance on the nature of research and the standard that is expected
Second Reader	Gary Stewart	Provide independent assessment of the project prior to the final assessment
Theory-based linguistic model designer	Victor Kabine	Implementation of a rule based spelling checker that makes use of finite state automata
Data-driven statistical model designer	Balone Ndaba	Implementation of a corpus driven spelling checker
Crowd-sourced dynamic user-driven model designer	Ditshego Phogole	Implementation of a user-driven spelling checker that makes use of crowdsourcing techniques

7.1.2. Communication management plan

Communication will be key for the success of the project. We will have meetings regularly with our project supervisors, on a weekly basis, according to our schedules. This will ensure that they are informed of our progress and if there are any problems they can be rectified in time and schedules revised. Communication between the team members will be facilitated through email regularly, at least once a week. This will enable us to share information and help each other with are individual deliverables.

7.2. Project scope

The scope of the AfriSpel project will include planning, design, development and testing of three spell checkers. The project will meet the requirements that are set out by the supervisors and proposer of the project. The scope of this project might also include mobile/web development.

7.2.1 Deliverables and milestones

These are the deliverable that will be done by the application and they will span the planning, design and the testing of the spell checker.

Deliverables
Literature Review
Project proposal and project plan
Project Web Presence
First and Final Prototype
Final Project Report
Poster
Web Page
Reflection
Final Code

With the deliverables listed, it is important to know how much time will allocated to each of the tasks or deliverables.

Milestone	**Due date**
Literature Review	28-04-2015
Project Proposal and Project Plan	19-05-2015
Proposal Feedback Review	28-05-2015
Revised Proposal Finalized	12-06-2015
Project Web Presence	12-06-2015
Initial Feasibility Demonstration	20-07-2015
Background of Final paper	24-07-2015
Design Section	21-08-2015
First Implementation and Write-up	11-09-2015
Student Final Prototype and Write-Up	21-09-2015
Student Code Completion with Implementation	25-09-2015
Testing and Testing Write-Up	28-09-2015
Outline of Complete Report	02-10-2015
Complete Draft of Report	16-10-2015
Final Report Hand-in	26-10-2015
Poster Hand-in	02-11-2015
Web Page Upload	09-11-2015
Student Reflection Paper	13-11-2015

7.2.2 Requirements

The skills that are required from us both individually and as a group include knowledge on computational linguistics. We will make use of information on natural language processing morphology as well as techniques from information retrieval. There will also be web/mobile development and human computer interaction (HCI) skills that

will be required for the third part of the project.
We need to collect texts and store them electronically. This text can be collected from documents in public archives, libraries, consulting language experts and using already existing dictionaries. The Web can be of great use as a collection of data and the World Wide Web constitutes the largest existing source of texts. There are already existing text resources.

7.2.3 Key success features

We will measure the success of the project by the design of the spell checkers and the implementation of the spell checkers. The success of the project however is not dependent on there being three spell checkers to evaluate and thus if there is any hindrance or unforeseen incident that occurs to one project team member, the project is still able to continue without the other spell checker based on one of the philosophies. The spell checker makes use of standard isiZulu language with no emphasis on the dialect.
The spell checkers are able to flag for an incorrect word. The spell checkers are able to generate a list of possible suggestion as to what the possible correct answer is. The command line function is able to give output to the user in terms of options. A functional, well designed user interface will be created for the crowdsource dynamic user-driven spell checker. For the other two spell checkers it is a possible feature that might not be implemented because it is of less priority.

7.2.4 Project risks

There are some risks that will be involved in this project and the table below discusses ways in which we are going to mitigate the risks and how likely the risks that will affect the schedule of the project. The impact section depicts how harmful the problem will be to our project.

Risk	Probability 1-10	Impact 1-10	Avoidance Strategy	Mitigation strategy
Scope creep	5	7	Focus on the core features of the project. Expand on the key features after the completion of the core features	Have a good understanding with the supervisor and a good work ethic
Task estimation issues	4	7	Focus on the most important features of the project and prioritize them first	Give the tasks the maximum time it is allowed to complete them
Communication break-down	4	7	Gain good communication skills and holding regular meetings with one another	Ask for external help and hold meetings to resolve the problem
Skills required not learned in time	5	8	Have a good work schedule and have regular meetings to discuss the deliverables	Seek guidance from the supervisors.
No access to required resources/tools	5	8	Research where and how the tools are used and where they can be located	Seek guidance of the supervisors
Project team leader leaves or is not contributing	8	5	Keep a good communication with the team leader	The team leaders have independent roles in the project. We can remove the team member’s part

Bibliography

DE CLERCQ, O., HOSTE, V., DESMET, B., VAN OOSTEN, P., DE COCK, M., & MACKEN, L. (2014). Using the crowd for readability prediction. Natural Language Engineering, 20(3), 293-325. Doi: 10.1017/S1351324912000344

Hurskainen A. 1992: A two level formalism for the analysis of Bantu language: an application to Swahili, Nordic Journal of African studies 1 87-122

Spiegler S, Spuy A. van der and Flach P.A, 2010: Proceeding COLING '10 proceedings of the 23rd international conference on computational linguistics pp 1020-1028

Pretorius L and Bosch S.E. 2003. Finite state computational morphology: an analyser prototype for Zulu: Machine translation.

Karttunen, L. 2003: Finite state technology: the oxford handbook of computational linguistics

Appendix A (GANTT Chart)