A Rule Based Spelling Error Detector for IsiXhosa

Siseko Neti
Department of Computer Science
University of Cape Town
Rondebosch, Cape Town, 7700
ntxsis001@myuct.ac.za

ABSTRACT

Spellchecking is an important tool for the writing of text and with the increase in text based communication spellcheckers have become increasingly important. With the few advances that have been made in the field of spellchecking for Nguni languages, there are currently two spellcheckers available for the Nguni language isiXhosa, which both have limitations in terms of their functionality, scope and accuracy. In this report a rule based approach for isiXhosa spelling error detection was investigated. As a result we have implemented a system and wrote some of the grammar rules for the nouns, verbs, adjectives, pronouns and possessives which had effect on the spelling of words.

The system was implemented using a tool called SFST and Java, with Github used as a version control system. The system was tested in two iterations where we first tested the rules individually and then combined them as one system which was also tested on its own. From these tests the accuracy rate for POS tagging is 88.26% for the noun rules, 94.58% for verb rules, 97.91% for adjective rules, 98.12% for pronoun rules and 100% for the possessives rules. The spelling error detection of the overall system received an accuracy of 80.8%.

Keywords

Morphological Analyser, Error Detector, IsiXhosa, SpellChecker, Natural Language Processing, POS Tagging.

1. INTRODUCTION

Spellchecking is the act of detecting whether a word form is correct or not [Rios 2011] and it is used in many text based computer application. With the increase in the use of text based communication (social media messaging, email, e.t.c) spelling error detection has become increasingly important since incorrectly spelled words can cause confusion and misunderstanding in the communication.

Even though there is a number of projects that have been done in the field of spellchecking, only a few exist for Nguni languages and from these, only Spellchecker.net and an OpenOffice plugin are available for isiXhosa. Spellchecker.net is a web-based spellchecker that offers error detection and correction of misspelled words for various languages including isiXhosa. Spellchecker.net has no documentation for the spellchecker. The OpenOffice plugin is outdated and thus could not be used anymore and also no documentation was found about it.

On top of the isiZulu spellchecker developed by Ndaba et.al [2016], another spellchecker for one of the Nguni languages might be advantageous to the residents of this country since people are generally more comfortable and prefer to interact with technology in their native language [Pretorius and Bosch 2003]. IsiXhosa is chosen amongst all the other languages since it is closely related to all the other Nguni languages and we hope that any work done on this language can easily be bootstrapped to any of the Nguni languages [Mzamo et.al 2015].

Since the 11 official South African languages belong to the lesser studied and computer resourced languages in the world the government of South Africa has decided to have an open source repository for data/information about all these languages [Pretorius and Bosch 2009]. The repository is called RMA (https://rma.nwu.ac.za/) and currently it has 32 files for isiXhosa. Considering the fact that isiXhosa is an agglutinative language, using existing files to detect incorrectly spelled words may not produce a good spelling error detection accuracy rate since these files may not have most words of the language.

The most common and widely used spellchecking approach is the statistical based approach, but as stated above due to the agglutinative nature and the scarcity of computer resources for isiXhosa this approach might not perform at its best. Thus this project mainly focuses on investigating the feasibility of an error detector for isiXhosa using the rule based approach. Based on this investigation we then aim at comparing the accuracy of the rule based approach (presented in this paper) with the statistical based approach (presented on Nthabiseng Moshiane's paper).

Furthermore, the project's objective was the development of a spelling error detector for the isiXhosa language using the rule based approach. This is a theory based approach which focuses on studying and analysing the morphological structure of isiXhosa. We use these morphological structures and then write them as regular expression which will then be encoded as finite state transducers.

These kind of transducers give us a system that we call a morphological analyser and from this morphological analyser we will be able to feed in words which will then be checked whether they satisfy the rules or not. If they do then they will be regarded as correctly spelled, otherwise not.

This report first introduces a number of related papers and projects. Then it focuses on the design and implementation of the system. Thereafter it provides an in-depth analysis of how the system will be evaluated and then present results on the performance of the system in this evaluation. Finally, the report presents the conclusions that were reached as well as areas for future works.

2. BACKGROUND AND RELATED WORK

2.1 Background

An agglutinating language is one whose words are formed by a combination of different morphemes where these morphemes are not changed prior their use in any word [Prószéky and Kis 1999].

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.

There are currently two approaches that can be followed for developing an isiXhosa spelling error detector. The two approaches are the rule-based approach (presented in this paper) and the statistical based approach which uses dictionary lookup and/or n-gram analysis for spelling error detection (presented in Nthabiseng Mashiane's paper).

As mentioned in the section above, amongst these two approaches, the most popular approach in spellchecking is the use of a text corpus to flag incorrectly spelled words. Bosch and Eiselen [2005] highlight that since isiXhosa is a highly agglutinative language such text would be extremely huge and thus would consume a large amount of physical memory, which is not ideal for spellcheckers.

The rule based approach has been followed and has worked well in a number of projects that were based on agglutinative languages. These projects are discussed in the following subsection.

2.2 Related Work

Morphological analysis has been used in the spellchecking of most/many agglutinative languages, this is evident by projects such as the ones stated below. This morphological analyser is basically a finite state transducer that is built from the morphology of the language.

In an attempt to create a spellchecker for Quechua which is a strongly agglutinative and suffixing language from South America, a morphological analyser which used the XFST tool was produced [Rios 2011].

Katushemererwe [2010] describes the implementation of a morphological analyser using the XFST tool in analysing the morphology of the Runyakitara language which is an agglutinative Bantu language from Uganda. This analyser was reported to have an accuracy of 80% accuracy rate on Runyakitara nouns.

Kessikbayeva and Cicekli [2016] describe the implementation of a morphological analyser for Kazach which is an agglutinative language spoken in countries such as China, Russia, and more. This morphological analyser is reported to have an accuracy rate of 87%.

Kumar et.al [2012] developed a morphological analyser for Hindi using the SFST tool and the analyser was reported to have an accuracy rate of 97%. The morphological analyser developed was used in a Part Of Speech (POS) Tagger based on Stanford POS Tagger.

[Lipps] describes the implementation of a morphological analyser for an East African Bantu language called Swahili. The morphological analyser was implemented using the XFST tool and it was based on some of the noun, verb, nominal and adjectival morphology of the language.

Computational morphological analysers for Nguni languages has been reported to be feasible by Pretorius and Bosch [2002] where they produced a functional noun prototype of a morphological analyser for isiZulu. They further extended their work to other POS categories such as verbs and from this work a morphological analyser named ZuluMorph for isiZulu was produced. Furthermore, [Bosch et.al 2008, Pretorius and Bosch 2009] describe the process of bootstrapping the already existing morphological analyser for isiZulu to develop one for isiXhosa. Both of these papers reported that the process used the XFST tool and that using ZuluMorph as an already existing prototype took less development time for isiXhosa than it was when ZuluMorph was first developed. Bosch et.al [2008] also reported the morphological analyser to have correctly analysed 71.10% of isiXhosa words.

Agglutinative languages such as Runyakitara, Quechua, Runyakitara, Hindi and Kazach which have been investigated for error detection have used and shown that morphological analysis gives higher accuracy in spelling error detection of such languages. Therefore in the following section we discuss the design and implementation of a morphological analyser for isiXhosa.

3. SYSTEM DESIGN AND IMPLEMENTATION

3.1 Tools and Technologies

We have looked at a number of finite state tools such as the Xerox Finite State Tool (XFST), Helsinki Finite-State Tool (HFST), OpenFST, Foma and Jflap but since we could not obtain the functionality and results that we wanted in some of these tools, we then decided to use the Stuttgart Finite-State Transducer Tool (SFST) which is freely available under the GNU Public License. SFST is a linux based tool which through its robust programming language (SFST-PL) supports many different formats of regular expressions such as the ones used in grep, sed or Perl [Schmid 2005]. Below are some of the reasons and motives as to why all the other tools were not chosen.

3.1.1 XFST

XFST is a commercial software/tool, thus since we want the spellchecker to be freely available to everyone we decided not to use this tool.

3.1.2 JFlap

JFlap is a package of graphical tools that can be used for experimenting with topics in the computer science area of formal language and automata theory [Rodger et.al 2006]. The tool did not allow us to define multiple regular expressions which could be used simultaneously. Also the regular expressions defined in the tool cannot be used simultaneously with some set of correctly spelled words.

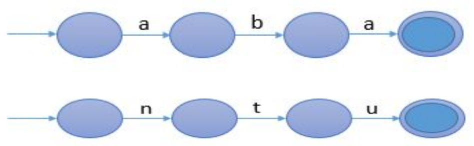
3.1.3 HFST
Helsinki Finite-State Transducer Technology (HFST) is a software intended for the implementation of morphological analysers and other tools which are based on weighted and unweighted finite state transducer technology. Regular expression written using this tool can only be edited using this tool, and since the tool supports weighted transducers it was a bit complicated and confusing working with it since the tool forced us to specify weights in some of our rules.

3.1.4 OpenFST
OpenFST is a library for weighted finite state transducers. Since we did not want to include/use weights in our rules this tool could not be used as it only support transducers that are weighted.

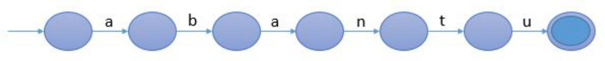
3.1.5 Foma
Foma which is a compiler, programming language, and C library for constructing finite-state automata and transducers [Hulden 2009]. Foma support most functionalities that are supported by the XFST tool and it is the one other tool that we could have worked with since it offers all the functionality that we needed but we were more comfortable in working with SFST than Foma.

3.2 Design Process
In developing this morphological analyser we first had to decide on which aspects of the morphology to focus on. As a result we have read about the morphology of the language which was found in books [Dowling and Wise 1998, Pahl et.al 1997, Satyo 1999, McLaren and Welsh 1936, Kosch 2011, Kotze 2003, Du Plessis and Visser 1995]. After reading about the morphology and consultating from Dr M Keet and Dr M Motinyane-Masoko we then decided to firstly focus on nouns and verbs.

We then followed a similar approach as in [Kessikbayeva and Cicekli 2016] where we created two different sets of two-level rules for each part-of-speech (POS) category which were later combined. We first created the orthographic rules which detect the spelling of the morphemes in the language and then we created the morphotactic rules which were specifying the allowed combination of these morphemes. For example if we want to detect the noun 'abantu' (belonging to noun class 2 and meaning: people), then since the word is formed by two morphemes: aba- and -ntu, our orthographic rules would detect whether these morphemes are correctly spelled such that we don't have cases like -nut instead of -ntu. The transducer rules would be as follows:



Then the morphotactic rules would allow us to make the correct combination which is 'abantu' instead of 'ntuaba', and the transducer for this would be:



These two set of rules for each POS category were then combined to form a transducer for each category that we decide to look at. These transducers for all these categories were then combined to form the integrated system which is our morphological analyser.

As part of our rule design process, all of the rules which will be described in subsections 3.4.1 - 3.4.5 have been confirmed by some of the students from the African Language section at the University of Cape Town. These rules were developed in an iterative approach, where in each iteration we wrote the rules for each POS category and then tested how each rule affected the overall accuracy of the system.

3.3 Architecture
The architecture of the system is very simple, there is a backend and a frontend. The backend interacts with the SFST program using Java. A more detailed overview of the system is shown in the system diagram below.

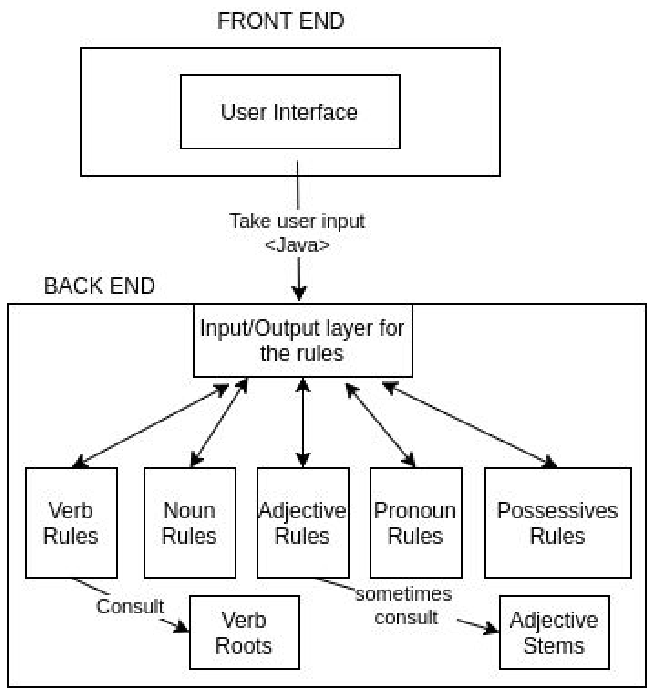


Figure 1: System Architecture

The input/output layer in the backend is simply what takes the user input and then split it into single words which are then passed through the morphological analyser. On the frontend we simply run a java program and then read a text file or enter words to be spellchecked manually one by one. The user interface was developed in Java Swing with the aim of allowing us to test the accuracy of our error detector. Even though we have tried to make an easy to use tool, user satisfaction was not the primary goal of building the interface and thus no user evaluations have been done for this interface.

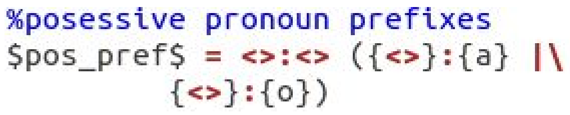
3.4 Rule Design
We briefly discuss major parts of our rule design process in the following subsections on a very high level definition, all files/codes are commented nicely with more explanations. The source code is available in the project website, on Github [Neti 2017] and also in Appendix A below.

In defining our rules we have used the following SFST syntax as defined and stated by Schmid [2005]:

1. Variables are defined by using two dollar signs and putting the variable name inside them e.g $mapp$ = a:e defines the variable name mapp as a mapping of an 'e'

to ‘a’. When calling these variable names we also use the name $mapper$ as it is.
2. An ’or’ condition is represented by | and |\ defines the ’or’ condition between two consecutive lines.
3. The symbol & is used as an ‘and’ condition between two statements.
4. % symbols are used for single line comments.

As an example, the snapshot below (figure 2) first starts with a comment (as the first line starts with a percentage sign) saying ‘possessive pronoun prefixes’ and then it defines a variable called ‘pos_pref’ (as this is between two dollar signs) which is the mapping of ‘a’ to an empty string or ‘o’ to an empty string. Note that the ‘or’ operator used here is the one that is between two consecutive lines.



```
%posessive pronoun prefixes
$pos_pref$ = <>:<> ({<>}:{a} |\
          {<>}:{o})
```

Figure 2: Snapshot from the possessive rules

For more information about the SFST syntax please refer to the manual in Schmid [2005].

Bosch et.al [2008] reported that the greater morphological analysis complexity of Nguni languages lies with the nominal and verbal morphology. As a result we first looked at the noun and verb rules, and thereafter looked at adjectives, pronouns and possessives.

Karttunen [1996] suggest that in order to avoid time and space problems transducers for different sections be separate and combined at the end instead of having the entire system as a single transducer. Thus, we have separated our transducers with regards to different morphological groups and each of the following subsection represents a separate transducer.

3.4.1 Nouns

We have first looked at the noun classes and have separated the nouns according to their classes. The 15 noun classes for isiXhosa are separated/differentiated with their prefixes so we have used these prefixes together with a set of rules that determine the orthography for the stems of the noun which were found in [Satyo 1999, Pahl et.al 1997]. Since the orthography of noun stems is quite complex, we only accept noun stems that do not have more than one consecutive vowel and has no more than three consecutive consonants (see line 7 in Appendix A1). Also no consonant is allowed to follow itself in the stem (see line 13 in Appendix A1) and then finally the stem has to end with a vowel (see line 37 in Appendix A1). These stems not only cater for noun stems but also cater for any other stems in the language as a whole, so we expect that the rules will detect more words as correctly spelled than should be. [Satyo 1999, Pahl et.al 1997, Dowling and Wise 1998, McLaren and Welsh 1936, Du Plessis and Visser 1995]

We have also looked at compound nouns but figuring out that the noun stems can recognise any words these compound nouns were also recognised by the rules of the regular nouns. Compound nouns are basically formed by combining two words, where one of them should be a noun [Satyo 1999, Pahl et.al 1997, Dowling and Wise 1998]. So based on that we have catered for cases where a noun is combined with another noun, an absolute pronoun, or combined with an adjective. In terms of morphotactic rules we have catered only for the following cases:

1. A noun is combined with another noun. In this case our rules output a final word defined by : pref1+root1+suffix1+root2+suffix2, where pref1 is the prefix of the first noun, root1 is the root of the first noun, suffix1 is the suffix of the first noun, root2 is the root of the second noun and suffix2 is the suffix of the second noun.
2. A noun is combined with an absolute pronoun. The output of our rules is pref1+root1+suffix1+root2, where pref1 is the noun prefix, root1 is the root of the noun, suffix1 is the suffix of the noun and root2 is the root of the pronoun.
3. A noun is combined with an adjective and the output is pref1+root1+suffix1+adjective_stem, where pref1 is the noun prefix, root1 is the root of the noun, suffix1 is the suffix of the noun and adjective_stem is the stem of the adjective.

Where a prefixes could either be a subject concord, a copulative concord, an object concord or a combination of 2 or all these concords in some cases. Some compound nouns with complex grammar were not covered in this work due to time constraints.

3.4.2 Verbs

We have managed to find a file with verb roots from the [Rma: 2012] website. The file had 4354 verb roots with 121 of the roots beginning with vowels. So now the main objective was on determining the orthographic rules of the prefixes and the allowed combination of prefixes and suffixes for these roots.

The prefixes are categorised into different combinations and they can be able to recognise only the verbs from whose roots are available in the file mentioned above.

We have managed to find the orthographic rules and the allowed combinations of prefixes and suffixes for verbs in the remote past tense, present tense, future tense, past subjunctive mood, izixando from the books. [Satyo 1999, Pahl et.al 1997, Dowling and Wise 1998, McLaren and Welsh 1936, Du Plessis and Visser 1995]. These rules have been written as regular expressions (see Appendix A2 below).

3.4.3 Adjective

Adjectives were the next POS category that we have looked at after verbs. Adjectives are subgrouped in the isiXhosa language and in this work we have looked at all of these subgroups. The prefixes for the adjectives are are called “adjective concords” and each group has its own set of adjective concords [Satyo 1999, Pahl et.al 1997, Dowling and Wise 1998, McLaren and Welsh 1936, Du Plessis and Visser 1995]. The first group of adjectives named “Iziphawuli” has 22 stems which are combined with these concords (see lines 1-21 and line 86 in Appendix A3). The second group of adjectives named “Izibaluli” has a number of stems which can be combined with its set of concords and we only managed to find 39 of these stems (see lines 21-40 and 87-89 in Appendix A3). The third group named “izimnini” has stems which are formed from different aspects of the grammar such as absolute pronouns, “iziphawuli” and nouns (see lines 42-65 and 90-94 in Appendix A3). The other subgroup named “Izichazi Zokukumbi” and the last subgroup that we looked at is named “Izichazi Zoquko” which are formed with the two stems “dwa” and “nke” (see lines 73-84 and 92-94 in Appendix A3).

All of the rules described above were found in books [Satyo 1999, Pahl et.al 1997, Dowling and Wise 1998, McLaren and Welsh 1936, Du Plessis and Visser 1995].

3.4.4 Pronouns

Pronouns in isiXhosa are called 'izimelabizo' and they are also subgrouped into different subsections. We have first looked at absolute pronouns, then demonstrative pronouns and then after that looked at Izimelabizo zoquko and izimelabizo zochazo. Demonstrative pronouns were mostly morphemes that were not combined with any other morphemes in most cases (see lines 11-19 in Appendix A4). Absolute pronouns include words such as yena (meaning 'him'/'her' in English translation) which can only be combined with prefixes 'a', 'e', or 'o' (see lines 1-9 and line 32 in Appendix A4). After these, we have then looked at izimelabizo zoquko which are pronouns used in grouping things/people together and we have not looked at those that involve numbers due to time constraints. Lastly we looked at izimelabizo zochazo which are used for purposes of informing us about the place/location of the noun, for both of these subgroups see lines 21-30 and line 34 in Appendix A4. All of these rules were found in books [Satyo 1999, Pahl et.al 1997, Dowling and Wise 1998, McLaren and Welsh 1936, Du Plessis and Visser 1995].

3.4.5 Possessives

Dowling and Wise [1998] stated that possessives occur quite more often in isiXhosa sentences and text. Thus for the better accuracy of our spelling error detector we have decided to also look at them in order to be able to recognise a huge chunk of these which would then boost the overall accuracy of our error detector. Possessives are also grouped according to the language characteristics they are based on, e.g we have possessives for nouns, verbs, e.t.c. In this project we have mainly focused on possessives for pronouns and then looked at some noun possessives aswell, but due to time constraints we could not cover much of these noun possessives. For a greater detail of this work on possessive please look at Appendix A5 below. [Satyo 1999, Pahl et.al 1997]

3.5 Use Case

The system can be used by anyone who is familiar with the language and would like to write something for the language. It could also be used by someone who is trying to learn the language as it would tell them when their words are incorrectly spelled. A usecase for this system is not included due to the fact that the system has a very basic functionality.

3.6 Implementation

We have wrote the rules in SFST-PL (which is a programming language that SFST uses) and with regards to testing the accuracy of our spellchecker we used Java as a programming language, where we developed a java swing GUI that allow users to just input words (either manually or by uploading a text file) and then run it through our error detector which then flags/highlight incorrectly spelled words. This interface was implemented in the testing stage of the error detector development since we wanted a system which was more simpler to use compared to the Linux terminal which was used prior.

Github was used as a version control system which then allowed us to be able to make changes to the code whenever we wanted to revert back to a code that we had modified if some rules did not work or perform as expected.

4. EVALUATION

4.1 Hypothesis

Test 1: The noun rules will have an accuracy rate of 60% or below since it is expected that the rules will regard some incorrectly spelled words as correct.

Test 2: The verb rules will achieve a very high accuracy rate of 90% or above since they will only accept valid isiXhosa verbs.

Test 3: The adjective rules will have an accuracy rate of 50% or below since we did not manage to find a very large portion of the adjective roots.

Test 4: The pronouns will have an accuracy rate of 90% or above.

Test 5: The possessions will have a high accuracy of 80% or above.

Test 6: The overall integrated system will achieve a very high accuracy rate of 85% or above.

4.2 Experiment Design

The aim of the entire experiment is to determine the accuracy of the transducer/rules that have been implemented. The system will be tested in two iterations.

The first iteration is for testing the rules with words that belong to the grammar aspect (the POS category) in which the rules are written for. The first experiment is for the individual transducer and it involves extracting a number of words from the text which belong to the POS category of the rules and then test them with the rules. The extracted words will then be confirmed by another native speaker for surety.

In the first iteration we also test the transducer using words from the other POS categories considered in this report. Note that in this case we don't extract new words over and over again but rather we store each set of extracted words in a text file for each POS category and then just use those throughout the experiment. This iteration will allow us to compute the accuracy of our POS category tagging for all our rules.

The second iteration involves testing the spelling error detection capabilities of each grammar aspect. This will also involve testing the integrated system with the whole corpus. This iteration is the one that will allow us to compute the accuracy rate of our spelling error detection. The results in both iterations will then be analysed using a confusion matrix which is clearly explained in subsection 4.4. Since Karttunen [1996] motivated us to avoid time problems, in both iterations we also timed our experiments using the timer which is available in Java in order to see how long do our computations take.

4.3 Experiment Documents

We will use 7 documents which were collected from the African Language section as our corpus for the entire experiment. These documents were a mixture of medical and academic documents with the majority being medical documents. These documents were all correctly spelled, with just few incorrectly spelled words. The table below show some characteristics of these documents.

Table 1: Characteristics of the text documents

	Total Words	Unique Words

Document 1	2203	1234
Document 2	3215	1316
Document 3	2731	1146
Document 4	2759	1399
Document 5	4704	2113
Document 6	2186	992
Document 7	4054	2400

Our corpus has 21852 words, therefore it should be a reasonable text for our testing purposes.

4.4 Evaluation matrix

To evaluate the system we have used a confusion matrix. A confusion matrix basically classifies the results of the text into four categories: true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) as depicted in the following table taken from Ndaba et.al [2016].

Table 2: Confusion Matrix

	Correctly Spelled	Incorrectly Spelled
Correctly Spelled	TP	FP
Incorrectly Spelled	FN	TN

The accuracy rate of the error detector is (TP+TN)/Total number of words in the test dataset. This will be calculated for all the text documents and then a final accuracy rate will be the summation of all. In order to get a feeling of how each rules affect the overall accuracy of the system these test will first be run on each set of rules e.g noun rules only, verb rules only, adjective rules only and then they will also be run on the entire system which is the combination of all these rules.

5. RESULTS AND FINDINGS

We have managed to find a list of nouns from the isiXhosa lemmatizer which we downloaded in the RMA [2012] website. The file had 20826 nouns and we used it throughout the experiment for nouns.

5.1 Results for rules by POS

In this section we provide the results for our first iteration of the evaluation. In the first iteration we have used the rules for one POS category and tested them with words from different POS categories in order to be able to tell how good the rules are in terms of detecting whether a certain word is for that specific POS category or not. For example: we used the noun transducers and then tested on it using only noun words, then used only verbs, then only adjectives, then only pronouns and then only possessives (see Appendix B1 for results). The results from these are then used to calculate the accuracy of the transducer by averaging all the rates obtained when testing single POS categories. This then is the POS tagging accuracy rate for the nouns.

This process was done for all the other POS categories aswell (see results in Appendix B2-B5). The table below is just a summary of how our POS tagging performed for each POS category.

Table 3: POS Category results

	Average Rate (%)	Average Time (ms)
Nouns	88.26	661.4
Verbs	94.58	490.2
Adjectives	97.91	490.8
Pronouns	98.12	444
Possessives	100	683

Overall these results show that our rules are quite good in terms of POS tagging.

5.2 Results for SpellChecking

In the second iteration we have then used each set of rules to test each document so that we may know how many correctly spelled words for each POS category are there in each of these documents. This also allows us to be able to tell the spelling error detection rate for each set of rules. The results are as follows:

Table 4: Number of unrecognised words per rules

	Rejected Nouns	Rejected Verbs	Rejected Adjectives	Rejected Pronouns
Document 1	1432	1728	2119	2164
Document 2	2103	2371	3030	3150
Document 3	1732	2089	2617	2690
Document 4	1804	2148	2643	2714
Document 5	3149	3843	4528	4636
Document 6	1470	1637	2049	2152
Document 7	2859	3387	3934	3998

From Table 4 above: The noun rules have detected 33.42% of the words as noun, verb rules have detected that 21.27% of the corpus is made up of verbs, adjective rules have detected 4.3% as correct adjectives, pronouns rules detected 2% of the corpus as pronouns and possessives rules detect 1.34% of the words in the corpus as correctly spelled possessives.

5.3 Results for the overall system

In terms of testing the overall system we have used all the 7 documents, but note that these documents were not modified so there are many words that are not recognised by the spellchecker as we did not have rules to cater for them.

Table 5: Words rejected by the error detector

	Rejected Words	Percentage (%)
Document 1	1213	55.06
Document 2	2103	65.41
Document 3	1732	63.42
Document 4	1804	65.39
Document 5	3149	66.94
Document 6	1470	67.25
Document 7	2859	70.52

Therefore, only 6522 words were recognised as correct by the spellchecker. Since the spellchecker did not implement all POS categories of the language, the overall accuracy below is measured by only considering the categories that we have rules for. As a results words which are incorrectly spelled according to the error detector are neither nouns, verbs, adjectives, pronouns nor possessives are regarded as true negatives.

Table 6: Results in a Confusion Matrix

	Correctly Spelled	Incorrectly Spelled
Correctly Spelled	5899	4120
Incorrectly Spelled	76	11757

Therefore, the accuracy of these rules is (11757+5899)/21852 = 80.8%

5.4 Results analysis

With regards to the incorrectly spelled words when testing using the noun transducer we found that nouns that began with letters w, y, z were regarded as misspelled. Also some of the nouns that were two words separated by white spaces were regarded as misspelled because we treat words separated by spaces as two different words.

When we were looking for possessives in the documents we only found a very small number of these, which in our case depicts that isiXhosa text does not have much possessives rather it mainly has nouns and verbs, so the argument that motivated us to look at this aspect was thus not true.

Considering the fact that we have tested the nouns with a very big text file, we are certainly sure that our rules can perform better than most systems. Both the pronouns and possessives achieved very high accuracy since there were not too many words found for these aspects, and thus we cannot enforce their accuracy as we are uncertain about how they would perform when tested with a very large text.

Some of the words which are regarded as incorrect by the integrated morphological analyser are words which are: not nouns, verbs, adjectives, pronouns or possessives. These words are then not counted as words which should be recognised as correctly spelled words since there are no rules in our system to recognise them. We have obtained these by looking through the spelled and misspelled words produced by the system.

The time which our computations take is reasonable enough for any spellchecking purposes, therefore separating the transducers into multiple files for each POS category has worked well. Our hypotheses have been falsified as shown by the results from the experiment.

6. CONCLUSION

The project primarily aimed at investigating the feasibility of developing a morphological analyser for isiXhosa. The project has shown that it is feasible to develop an error detector for isiXhosa using the rule-based approach and this project resulted in the successful implementation of the morphological analyser with noun, verb, adjectives, pronouns and possessives rules.

From these evaluation conducted in this project the accuracy rate for POS tagging is 88.26% for the noun rules, 94.58% for verb rules, 97.91% for adjective rules, 98.12% for pronoun rules and 100% for the possessives rules. The overall spellchecker received an accuracy of 80.8%.

The statistical based approach (presented on Nthabiseng Moshiane's paper) received a lower accuracy when tested with more nouns that were not in the dictionary that it used. Therefore, since the rule-based approach has been able to successfully detect more nouns than the statistical approach in the near future it might be very good in terms of accuracy to combine the best features from the two approaches.

7. FUTURE WORK

With no prior spelling checker for isiXhosa, a future project might look into providing a user interface for this morphological analyser so that end users can be able to use the spellchecker without any complicated steps to follow. Due to time constraints, the work provided here has not looked at all the morphology or POS category of the language and thus it is unable to function as a full spellchecker for the language, so work can be done on the aspects that were left out in this project. Thus, another future project might concentrate and focus in the rest of the grammar and also provide spelling error corrections for the incorrectly spelled words.

8. ACKNOWLEDGMENTS

I would like to thank Dr Maria for the support and supervision of the project. I would also like to thank Dr Motinyane-Masoko from the African Language section of the University of Cape Town for providing the books that were needed in getting the language rules and the documents that were required for testing purposes.Also like to thank the African Language section honours students for looking and confirming my rules.

9. REFERENCES

[1] Bosch, S.E. and Eiselen, R., 2005. The effectiveness of morphological rules for an isiZulu spelling checker. South African Journal of African Languages, 25(1), pp.25-36.

[2] Bosch, S.E., Pretorius, L. and Fleisch, A., 2008. Experimental bootstrapping of morphological analysers for Nguni languages.

[3] 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, pp.2-11.

[4] Dowling, T and Wise, P. 1998. Speak Xhosa with Us: Beginner to Advanced. Mother Tongues Multimedia Development. ISBN 0-620-22192-5

[5] Du Plessis, J.A and Visser, M. 1995. Xhosa Syntax. Second Edition. Via Afrika Limited. ISBN 0 7994 1326 7. 1-276

[6] Hulden, M., 2009, April. Foma: a finite-state compiler and library. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session (pp. 29-32). Association for Computational Linguistics.

[7] Karttunen, L., Chanod, J.P., Grefenstette, G. and Schille, A., 1996. Regular expressions for language engineering. Natural Language Engineering, 2(4), pp.305-328.

[8] 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.

[9] Kessikbayeva, G. and Cicekli, I., 2016. A rule based morphological analyzer and a morphological disambiguator for kazakh language. Linguistics and Literature Studies, 4(1), pp.96-104.

[10] Kosch, I. 2011. South African Journal of African Languages. University of South Africa. Volume 1 Number 1. 138-158

[11] Kotze, A. 2003. South African Journal of African Languages. University of South Africa. Volume 23 Number 1.

[12] Kumar, D., Singh, M. and Shukla, S., 2012. Fst based morphological analyzer for Hindi language. arXiv preprint arXiv:1207.5409.

[13] Lipps, J., xsma: A Finite-state Morphological Analyzer for Swahili.

[14] McLaren, J and Welsh, G.H. 1936. A Xhosa Grammar. Longmans, Green and Co.

[15] Mzamo, L., Helberg, A. and Bosch, S., 2015, November. Introducing XGL-a lexicalised probabilistic graphical lemmatiser for isiXhosa. In Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), 2015 (pp. 142-147). IEEE.

[16] 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.

[17] Neti, S. 2017. IsiXhosa Morphological Analyser. Github repository https://github.com/sisekoreggie/Morphological-Analyser

[18] Pahl, H.W, Ntusi, D.M and Burns-Ncamishe, S.M. 1997. IsiXhosa. Third Edition. APB. ISBN 0 7980 0062 7.

[19] Pretorius, L. and Bosch, S.E., 2002. Finite-state computational morphology-treatment of the zulu noun. South African computer journal, 2002(28), pp.30-38.

[20] Pretorius, L. and Bosch, S.E., 2003. Enabling computer interaction in the indigenous languages of South Africa: The central role of computational morphology. interactions, 10(2), pp.56-63.

[21] Pretorius, L. and Bosch, S., 2009, March. Exploiting cross-linguistic similarities in Zulu and Xhosa computational morphology. In Proceedings of the First Workshop on Language Technologies for African Languages (pp. 96-103). Association for Computational Linguistics.

[22] Prószéky, G. and Kis, B., 1999, June. A unification-based approach to morpho-syntactic parsing of agglutinative and other (highly) inflectional languages. In Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (pp. 261-268). Association for Computational Linguistics.

[23] 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.

[24] Rma: 2012. https://rma.nwu.ac.za/. Accessed: 2017-04-03

[25] Rodger, S.H. and Finley, T.W., 2006. JFLAP: an interactive formal languages and automata package. Jones & Bartlett Learning.

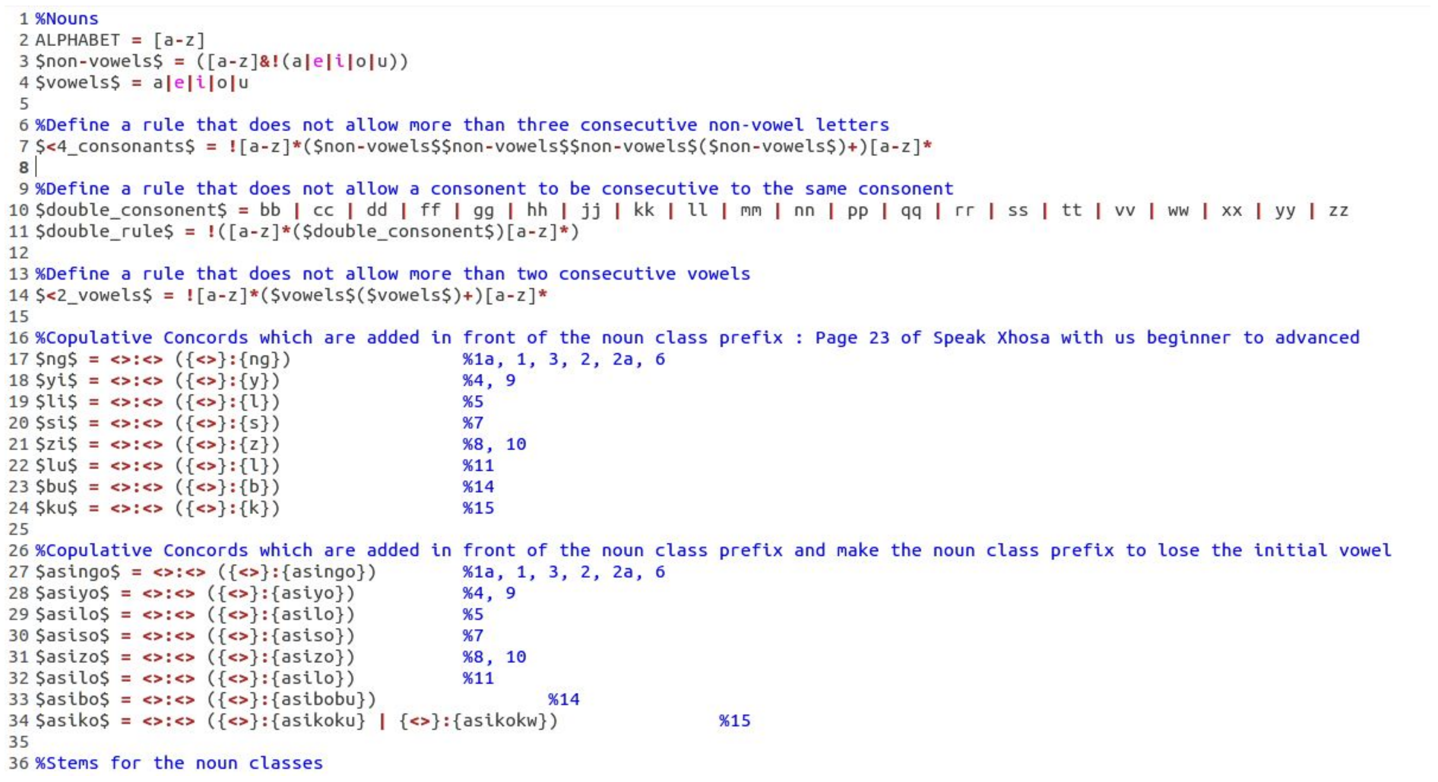
[26] Satyo, S.C. 1999. Igrama Noncwadi LwesiXhosa Ibanga 10. FIrst Edition. Via Africa. ISBN 0-7994-1105-1.

[27] Schmid, H., 2005, September. A programming language for finite state transducers. In FSMNLP (Vol. 4002, pp. 308-309).

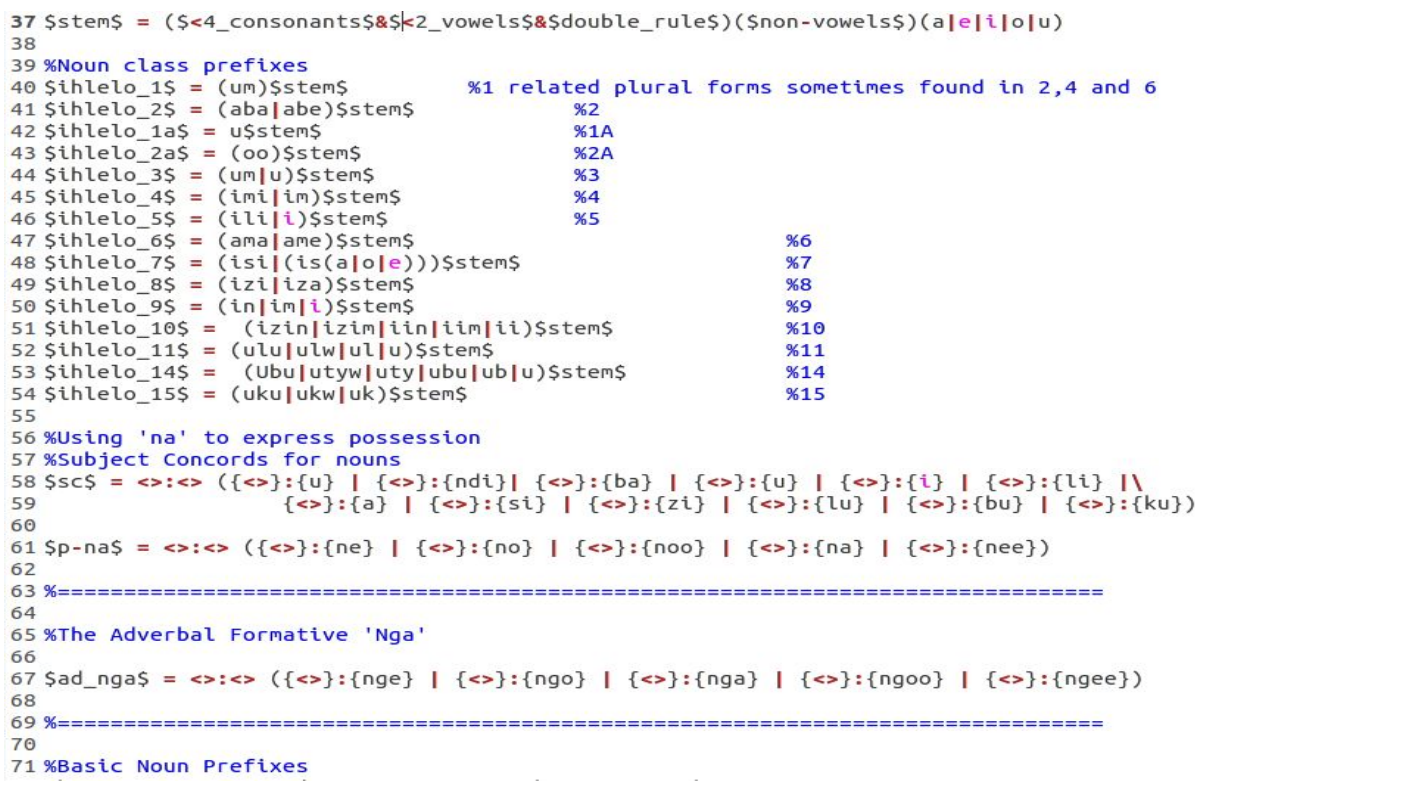
[28] Theron, P and Cloete, I. 1997. Automatic acquisition of two-level morphological rules. In *Proceedings of the fifth conference on Applied natural language processing*. Association for Computational Linguistics. 103–110.

Appendix A: IsiXhosa rules implemented using SFST

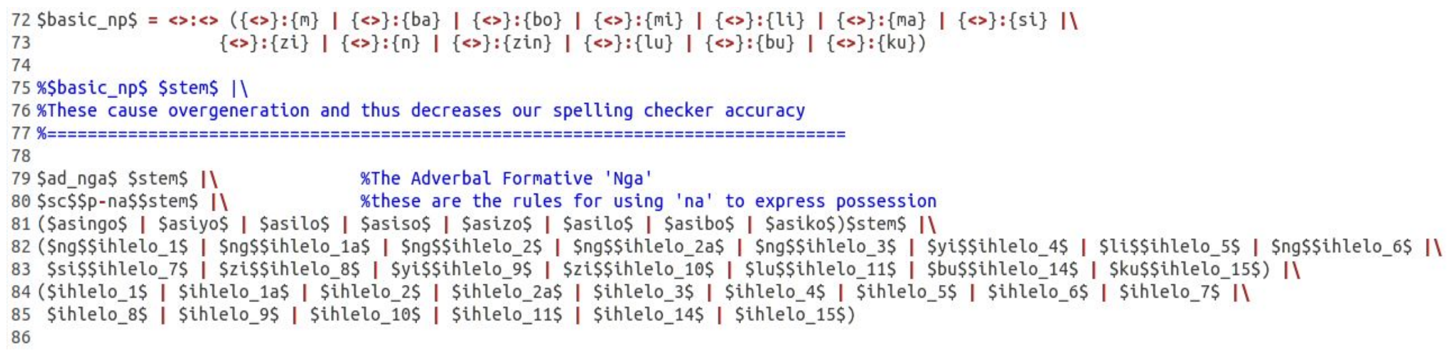
A1: Nouns



```
%Nouns
ALPHABET = [a-z]
$non-vowels$ = ([a-z]&!(a|e|i|o|u))
$vowels$ = a|e|i|o|u

%Define a rule that does not allow more than three consecutive non-vowel letters
$<4_consonants$ = ![a-z]*($non-vowels$$non-vowels$$non-vowels$($non-vowels$)+)[a-z]*

%Define a rule that does not allow a consonent to be consecutive to the same consonent
$double_consonent$ = bb | cc | dd | ff | gg | hh | jj | kk | ll | mm | nn | pp | qq | rr | ss | tt | vv | ww | xx | yy | zz
$double_rule$ = !([a-z]*($double_consonent$)[a-z]*)

%Define a rule that does not allow more than two consecutive vowels
$<2_vowels$ = ![a-z]*($vowels$($vowels$)+)[a-z]*

%Copulative Concords which are added in front of the noun class prefix : Page 23 of Speak Xhosa with us beginner to advanced
$ng$ = <>:<> ({<>}:{ng})              %1a, 1, 3, 2, 2a, 6
$yi$ = <>:<> ({<>}:{y})               %4, 9
$li$ = <>:<> ({<>}:{l})               %5
$si$ = <>:<> ({<>}:{s})               %7
$zi$ = <>:<> ({<>}:{z})               %8, 10
$lu$ = <>:<> ({<>}:{l})               %11
$bu$ = <>:<> ({<>}:{b})               %14
$ku$ = <>:<> ({<>}:{k})               %15

%Copulative Concords which are added in front of the noun class prefix and make the noun class prefix to lose the initial vowel
$asingo$ = <>:<> ({<>}:{asingo})      %1a, 1, 3, 2, 2a, 6
$asiyo$ = <>:<> ({<>}:{asiyo})        %4, 9
$asilo$ = <>:<> ({<>}:{asilo})        %5
$asiso$ = <>:<> ({<>}:{asiso})        %7
$asizo$ = <>:<> ({<>}:{asizo})        %8, 10
$asilo$ = <>:<> ({<>}:{asilo})        %11
$asibo$ = <>:<> ({<>}:{asibobu})              %14
$asiko$ = <>:<> ({<>}:{asikoku} | {<>}:{asikokw})             %15

%Stems for the noun classes
```

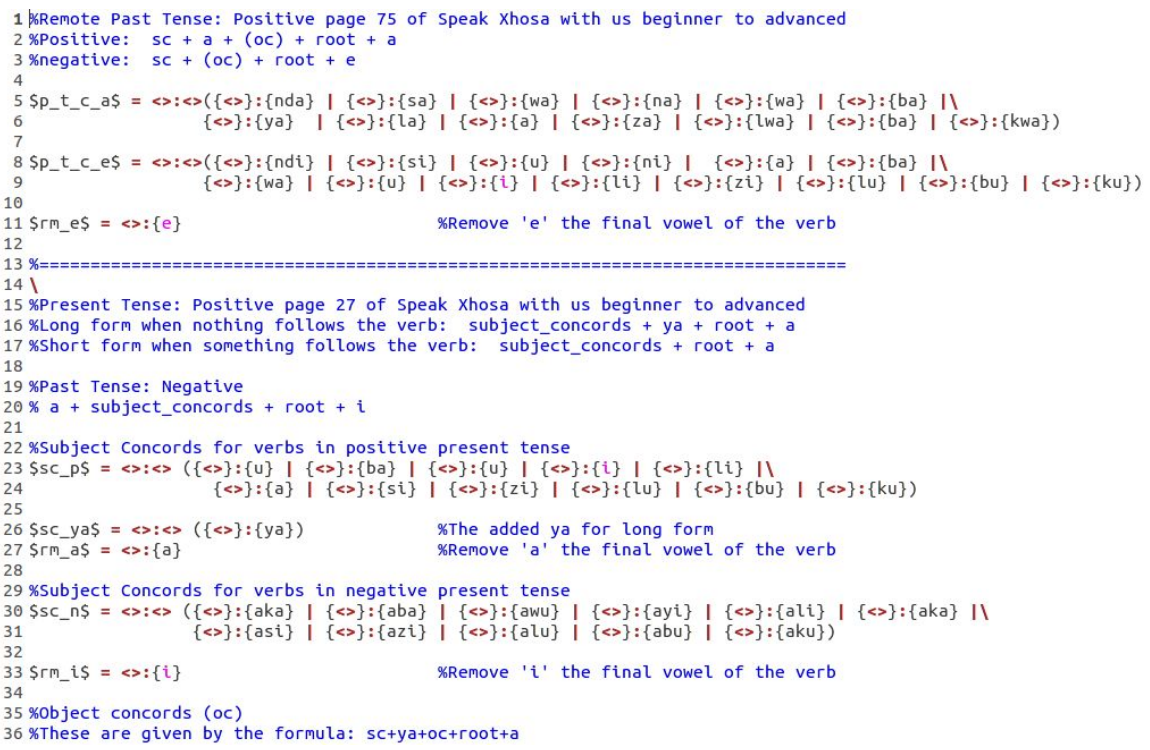


```
$stem$ = ($<4_consonants$&$<2_vowels$&$double_rule$)($non-vowels$)(a|e|i|o|u)

%Noun class prefixes
$ihlelo_1$ = (um)$stem$          %1 related plural forms sometimes found in 2,4 and 6
$ihlelo_2$ = (aba|abe)$stem$             %2
$ihlelo_1a$ = u$stem$                    %1A
$ihlelo_2a$ = (oo)$stem$                 %2A
$ihlelo_3$ = (um|u)$stem$                %3
$ihlelo_4$ = (imi|im)$stem$              %4
$ihlelo_5$ = (ili|i)$stem$               %5
$ihlelo_6$ = (ama|ame)$stem$                             %6
$ihlelo_7$ = (isi|(is(a|o|e)))$stem$                     %7
$ihlelo_8$ = (izi|iza)$stem$                             %8
$ihlelo_9$ = (in|im|i)$stem$                             %9
$ihlelo_10$ =  (izin|izim|iin|iim|ii)$stem$              %10
$ihlelo_11$ = (ulu|ulw|ul|u)$stem$                       %11
$ihlelo_14$ =  (Ubu|utyw|uty|ubu|ub|u)$stem$             %14
$ihlelo_15$ = (uku|ukw|uk)$stem$                         %15

%Using 'na' to express possession
%Subject Concords for nouns
$sc$ = <>:<> ({<>}:{u} | {<>}:{ndi}| {<>}:{ba} | {<>}:{u} | {<>}:{i} | {<>}:{li} |\
                {<>}:{a} | {<>}:{si} | {<>}:{zi} | {<>}:{lu} | {<>}:{bu} | {<>}:{ku})

$p-na$ = <>:<> ({<>}:{ne} | {<>}:{no} | {<>}:{noo} | {<>}:{na} | {<>}:{nee})

%================================================================================

%The Adverbal Formative 'Nga'

$ad_nga$ = <>:<> ({<>}:{nge} | {<>}:{ngo} | {<>}:{nga} | {<>}:{ngoo} | {<>}:{ngee})

%================================================================================

%Basic Noun Prefixes
```

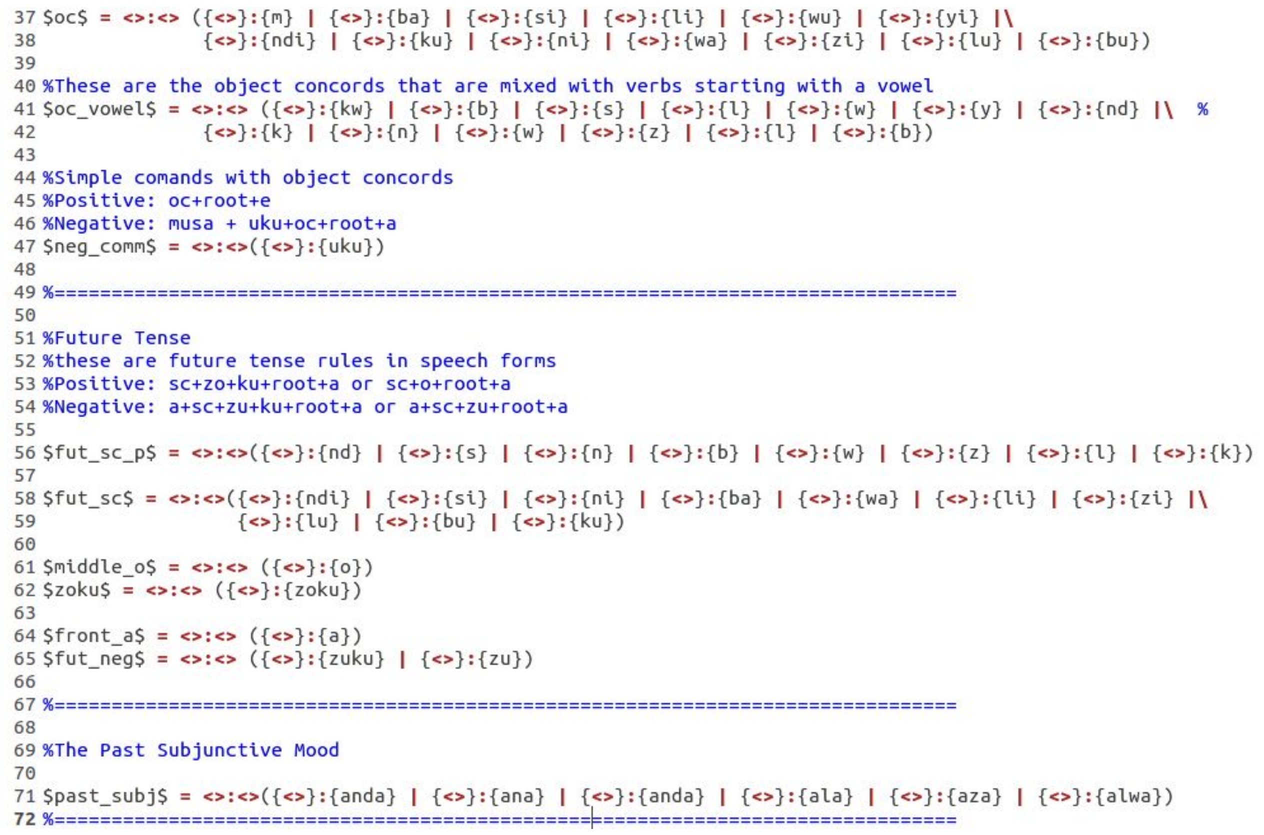


```
$basic_np$ = <>:<> ({<>}:{m} | {<>}:{ba} | {<>}:{bo} | {<>}:{mi} | {<>}:{li} | {<>}:{ma} | {<>}:{si} |\
                   {<>}:{zi} | {<>}:{n} | {<>}:{zin} | {<>}:{lu} | {<>}:{bu} | {<>}:{ku})

%$basic_np$ $stem$ |\
%These cause overgeneration and thus decreases our spelling checker accuracy
%=============================================================================

$ad_nga$ $stem$ |\              %The Adverbal Formative 'Nga'
$sc$$p-na$$stem$ |\             %these are the rules for using 'na' to express possession
($asingo$ | $asiyo$ | $asilo$ | $asiso$ | $asizo$ | $asilo$ | $asibo$ | $asiko$)$stem$ |\
($ng$$ihlelo_1$ | $ng$$ihlelo_1a$ | $ng$$ihlelo_2$ | $ng$$ihlelo_2a$ | $ng$$ihlelo_3$ | $yi$$ihlelo_4$ | $li$$ihlelo_5$ | $ng$$ihlelo_6$ |\
 $si$$ihlelo_7$ | $zi$$ihlelo_8$ | $yi$$ihlelo_9$ | $zi$$ihlelo_10$ | $lu$$ihlelo_11$ | $bu$$ihlelo_14$ | $ku$$ihlelo_15$) |\
($ihlelo_1$ | $ihlelo_1a$ | $ihlelo_2$ | $ihlelo_2a$ | $ihlelo_3$ | $ihlelo_4$ | $ihlelo_5$ | $ihlelo_6$ | $ihlelo_7$ |\
 $ihlelo_8$ | $ihlelo_9$ | $ihlelo_10$ | $ihlelo_11$ | $ihlelo_14$ | $ihlelo_15$)

```

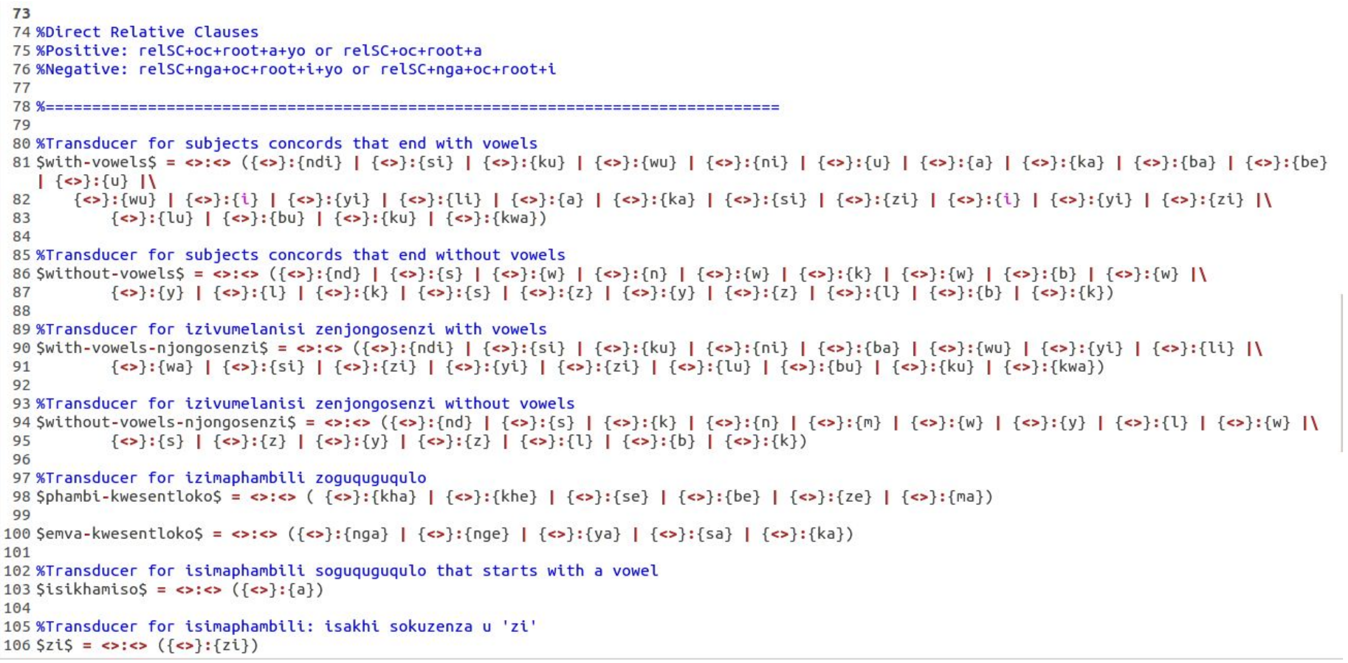
A2: Verbs



```
%Remote Past Tense: Positive page 75 of Speak Xhosa with us beginner to advanced
%Positive:  sc + a + (oc) + root + a
%negative:  sc + (oc) + root + e

$p_t_c_a$ = <>:<>({<>}:{nda} | {<>}:{sa} | {<>}:{wa} | {<>}:{na} | {<>}:{wa} | {<>}:{ba} |\
                 {<>}:{ya}  | {<>}:{la} | {<>}:{a} | {<>}:{za} | {<>}:{lwa} | {<>}:{ba} | {<>}:{kwa})

$p_t_c_e$ = <>:<>({<>}:{ndi} | {<>}:{si} | {<>}:{u} | {<>}:{ni} |  {<>}:{a} | {<>}:{ba} |\
                 {<>}:{wa} | {<>}:{u} | {<>}:{i} | {<>}:{li} | {<>}:{zi} | {<>}:{lu} | {<>}:{bu} | {<>}:{ku})

$rm_e$ = <>:{e}                         %Remove 'e' the final vowel of the verb

%=============================================================================
\
%Present Tense: Positive page 27 of Speak Xhosa with us beginner to advanced
%Long form when nothing follows the verb:  subject_concords + ya + root + a
%Short form when something follows the verb:  subject_concords + root + a

%Past Tense: Negative
% a + subject_concords + root + i

%Subject Concords for verbs in positive present tense
$sc_p$ = <>:<> ({<>}:{u} | {<>}:{ba} | {<>}:{u} | {<>}:{i} | {<>}:{li} |\
                  {<>}:{a} | {<>}:{si} | {<>}:{zi} | {<>}:{lu} | {<>}:{bu} | {<>}:{ku})

$sc_ya$ = <>:<> ({<>}:{ya})             %The added ya for long form
$rm_a$ = <>:{a}                         %Remove 'a' the final vowel of the verb

%Subject Concords for verbs in negative present tense
$sc_n$ = <>:<> ({<>}:{aka} | {<>}:{aba} | {<>}:{awu} | {<>}:{ayi} | {<>}:{ali} | {<>}:{aka} |\
                {<>}:{asi} | {<>}:{azi} | {<>}:{alu} | {<>}:{abu} | {<>}:{aku})

$rm_i$ = <>:{i}                         %Remove 'i' the final vowel of the verb

%Object concords (oc)
%These are given by the formula: sc+ya+oc+root+a
```



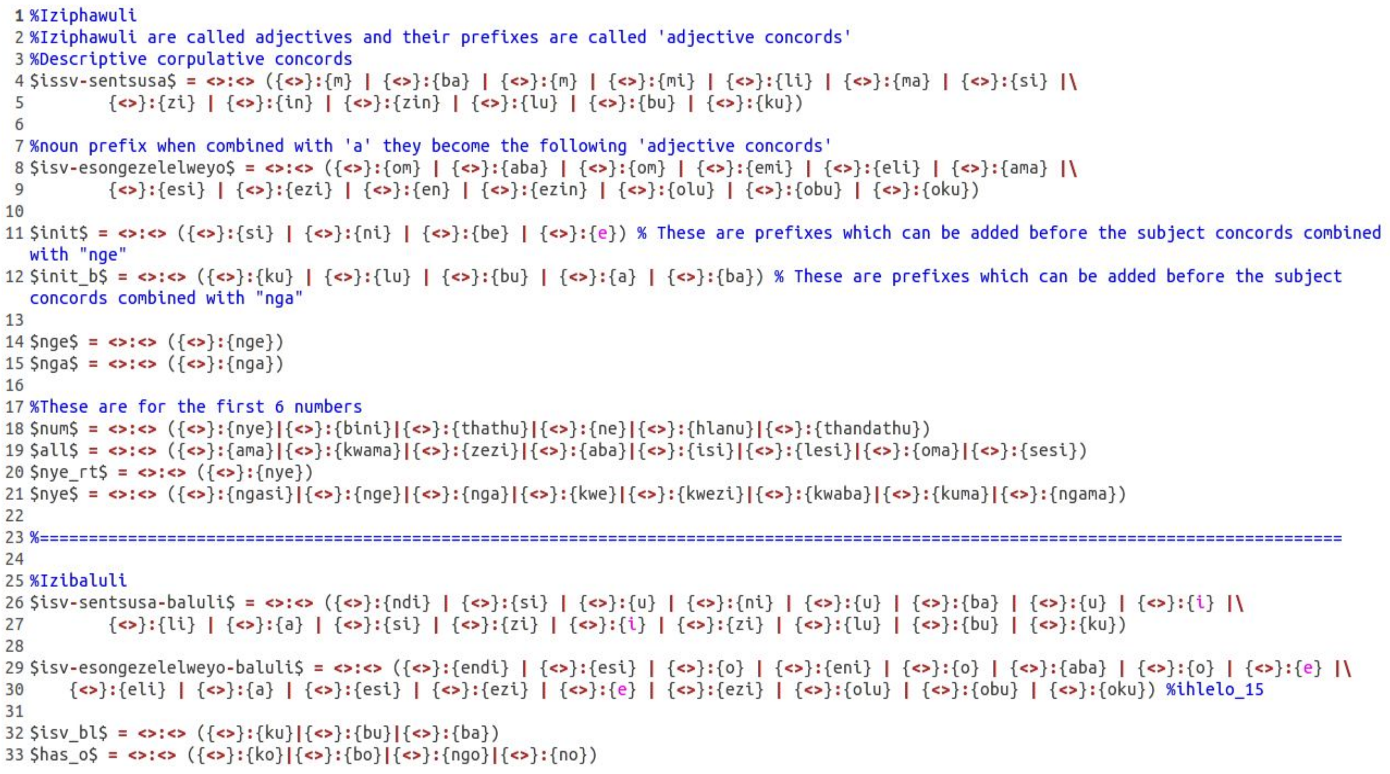
```
$oc$ = <>:<> ({<>}:{m} | {<>}:{ba} | {<>}:{si} | {<>}:{li} | {<>}:{wu} | {<>}:{yi} |\
             {<>}:{ndi} | {<>}:{ku} | {<>}:{ni} | {<>}:{wa} | {<>}:{zi} | {<>}:{lu} | {<>}:{bu})

%These are the object concords that are mixed with verbs starting with a vowel
$oc_vowel$ = <>:<> ({<>}:{kw} | {<>}:{b} | {<>}:{s} | {<>}:{l} | {<>}:{w} | {<>}:{y} | {<>}:{nd} |\  %
             {<>}:{k} | {<>}:{n} | {<>}:{w} | {<>}:{z} | {<>}:{l} | {<>}:{b})

%Simple comands with object concords
%Positive: oc+root+e
%Negative: musa + uku+oc+root+a
$neg_comm$ = <>:<>({<>}:{uku})

%=============================================================================

%Future Tense
%these are future tense rules in speech forms
%Positive: sc+zo+ku+root+a or sc+o+root+a
%Negative: a+sc+zu+ku+root+a or a+sc+zu+root+a

$fut_sc_p$ = <>:<>({<>}:{nd} | {<>}:{s} | {<>}:{n} | {<>}:{b} | {<>}:{w} | {<>}:{z} | {<>}:{l} | {<>}:{k})

$fut_sc$ = <>:<>({<>}:{ndi} | {<>}:{si} | {<>}:{ni} | {<>}:{ba} | {<>}:{wa} | {<>}:{li} | {<>}:{zi} |\
                {<>}:{lu} | {<>}:{bu} | {<>}:{ku})

$middle_o$ = <>:<> ({<>}:{o})
$zoku$ = <>:<> ({<>}:{zoku})

$front_a$ = <>:<> ({<>}:{a})
$fut_neg$ = <>:<> ({<>}:{zuku} | {<>}:{zu})

%=============================================================================

%The Past Subjunctive Mood

$past_subj$ = <>:<>({<>}:{anda} | {<>}:{ana} | {<>}:{anda} | {<>}:{ala} | {<>}:{aza} | {<>}:{alwa})
%=============================================================================
```



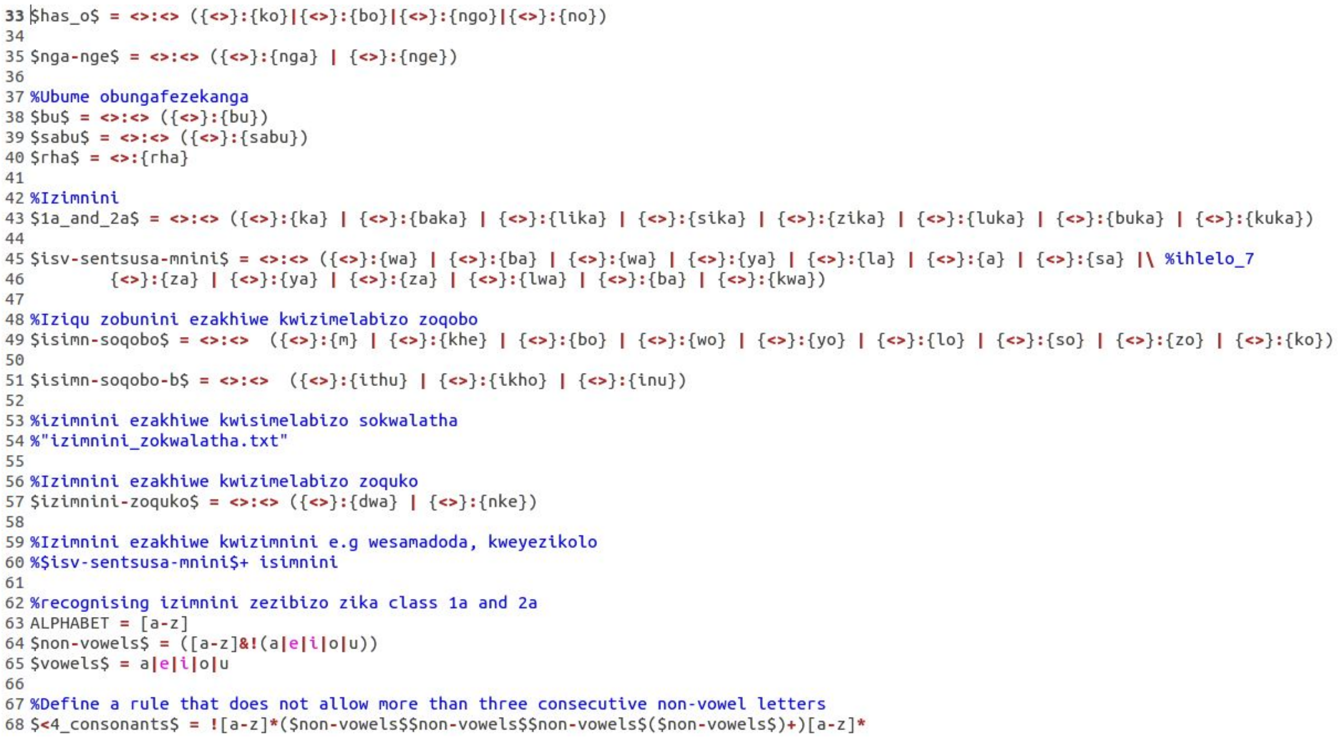
```

%Direct Relative Clauses
%Positive: relSC+oc+root+a+yo or relSC+oc+root+a
%Negative: relSC+nga+oc+root+i+yo or relSC+nga+oc+root+i

%=============================================================================

%Transducer for subjects concords that end with vowels
$with-vowels$ = <>:<> ({<>}:{ndi} | {<>}:{si} | {<>}:{ku} | {<>}:{wu} | {<>}:{ni} | {<>}:{u} | {<>}:{a} | {<>}:{ka} | {<>}:{ba} | {<>}:{be} | {<>}:{u} |\
   {<>}:{wu} | {<>}:{i} | {<>}:{yi} | {<>}:{li} | {<>}:{a} | {<>}:{ka} | {<>}:{si} | {<>}:{zi} | {<>}:{i} | {<>}:{yi} | {<>}:{zi} |\
       {<>}:{lu} | {<>}:{bu} | {<>}:{ku} | {<>}:{kwa})

%Transducer for subjects concords that end without vowels
$without-vowels$ = <>:<> ({<>}:{nd} | {<>}:{s} | {<>}:{w} | {<>}:{n} | {<>}:{w} | {<>}:{k} | {<>}:{w} | {<>}:{b} | {<>}:{w} |\
       {<>}:{y} | {<>}:{l} | {<>}:{k} | {<>}:{s} | {<>}:{z} | {<>}:{y} | {<>}:{z} | {<>}:{l} | {<>}:{b} | {<>}:{k})

%Transducer for izivumelanisi zenjongosenzi with vowels
$with-vowels-njongosenzi$ = <>:<> ({<>}:{ndi} | {<>}:{si} | {<>}:{ku} | {<>}:{ni} | {<>}:{ba} | {<>}:{wu} | {<>}:{yi} | {<>}:{li} |\
       {<>}:{wa} | {<>}:{si} | {<>}:{zi} | {<>}:{yi} | {<>}:{zi} | {<>}:{lu} | {<>}:{bu} | {<>}:{ku} | {<>}:{kwa})

%Transducer for izivumelanisi zenjongosenzi without vowels
$without-vowels-njongosenzi$ = <>:<> ({<>}:{nd} | {<>}:{s} | {<>}:{k} | {<>}:{n} | {<>}:{m} | {<>}:{w} | {<>}:{y} | {<>}:{l} | {<>}:{w} |\
       {<>}:{s} | {<>}:{z} | {<>}:{y} | {<>}:{z} | {<>}:{l} | {<>}:{b} | {<>}:{k})

%Transducer for izimaphambili zoguquguqulo
$phambi-kwesentloko$ = <>:<> ( {<>}:{kha} | {<>}:{khe} | {<>}:{se} | {<>}:{be} | {<>}:{ze} | {<>}:{ma})

$emva-kwesentloko$ = <>:<> ({<>}:{nga} | {<>}:{nge} | {<>}:{ya} | {<>}:{sa} | {<>}:{ka})

%Transducer for isimaphambili soguquguqulo that starts with a vowel
$isikhamiso$ = <>:<> ({<>}:{a})

%Transducer for isimaphambili: isakhi sokuzenza u 'zi'
$zi$ = <>:<> ({<>}:{zi})
```

```

%Rule for removing the suffixes for izixando
$remove$ = <>:{wa} | <>:{we} | <>:{iwe} | <>:{aneka} | <>:{iseka} | <>:{eleka} | <>:{ekekeka} | <>:{ekeka} | <>:{akala} | <>:{eka} |\
<>:{isana} | <>:{elana} | <>:{elelana} | <>:{ana} | <>:{olo} | <>:{ela} | <>:{ele} | <>:{la} | <>:{ya} | <>:{za} | <>:{sa} | <>:{esa} | <>:
{isa}

%Repeating suffixes for izixando
$repeated$ = ((<>:{el})+<>:{a}) | ((<>:{el})+<>:{e}) | ((<>:{is})+<>:{a}) | ((<>:{is})+<>:{e})

$remove$ = $remove$ | $repeated$

%| <>:{ene} | <>:{anwa} | <>:{enwe}
%Rule for removing suffixes zexesha elidlulileyo kwizivumo nakwizilandulo
$elidlulileyo$ = <>:{ile} | <>:{anga} | <>:{a}

%Isimamva sesininzi kwisiyaleli
$ni$ = <>:{ni} | <>:{ani} | <>:{eni}

%Isimamva sesibaluli
$yo$ = <>:{yo}

%Isigqibelo sesilanduli
$remove_i$ = <>:{i}

%Ezinye izimamva
$final$ = <>:{ula} | <>:{ulula} | <>:{uluka} | <>:{ama} | <>:{ma} | <>:{atha} | <>:{tha} | <>:{ala} | <>:{alala} | <>:{nga} | <>:{zela} |\
 <>:{ba} | <>:{pha} | <>:{leza} | <>:{ka} | <>:{da} | <>:{ta} | <>:{sha} | <>:{za} | <>:{bala}

$intloko-enesikhamiso$ = ($phambi-kwesentloko$$with-vowels$ | $with-vowels$$emva-kwesentloko$ | $with-vowels$)
$intloko-engenasikhamiso$ = $without-vowels$$isikhamiso$ | $isikhamiso$

$injongosenzi$ = $with-vowels-njongosenzi$$zi$ | $with-vowels-njongosenzi$

$vowel-starting-verbs$ = $intloko-engenasikhamiso$$without-vowels-njongosenzi$ | $intloko-enesikhamiso$$without-vowels-njongosenzi$ |\
 $without-vowels$ | $without-vowels-njongosenzi$



($past_subj$ "verbs.txt" $rm_a$) |\             %The Past Subjunctive Mood
($fut_sc$ $zoku$ "verbs.txt" $rm_a$) | ($fut_sc_p$ $middle_o$ "verbs.txt" $rm_a$) |\  %Positive Future Tense
($front_a$ $fut_sc$ $fut_neg$ "verbs.txt" $rm_a$) |\   %Negative Future Tense
($neg_comm$$oc$ "verbs.txt" $rm_a$) | ($oc$ "verbs.txt" $rm_e$) |\  %Simple comands with object concords
(($p_t_c_a$ "verbs.txt" $rm_a$) | ($p_t_c_e$ "verbs.txt" $rm_e$)) |\
($sc_n$ ($oc_vowel$)? "verbs.txt" $rm_i$) | ($sc_p$ ($sc_ya$|$sc_ya$$oc$)? "verbs.txt" $rm_a$) |\ %the following are the ones which come
from the verbs2 file
(($intloko-enesikhamiso$ | $intloko-engenasikhamiso$)? ($injongosenzi$)? "verbs.txt" ($remove$ | $elidlulileyo$ |\
 $ni$ | $yo$ | $remove_i$ | $final$)) | ($vowel-starting-verbs$ "verbs-vowel.txt" ($remove$ | $elidlulileyo$ | $ni$ | $yo$ | $remove_i$ |
$final$))
```

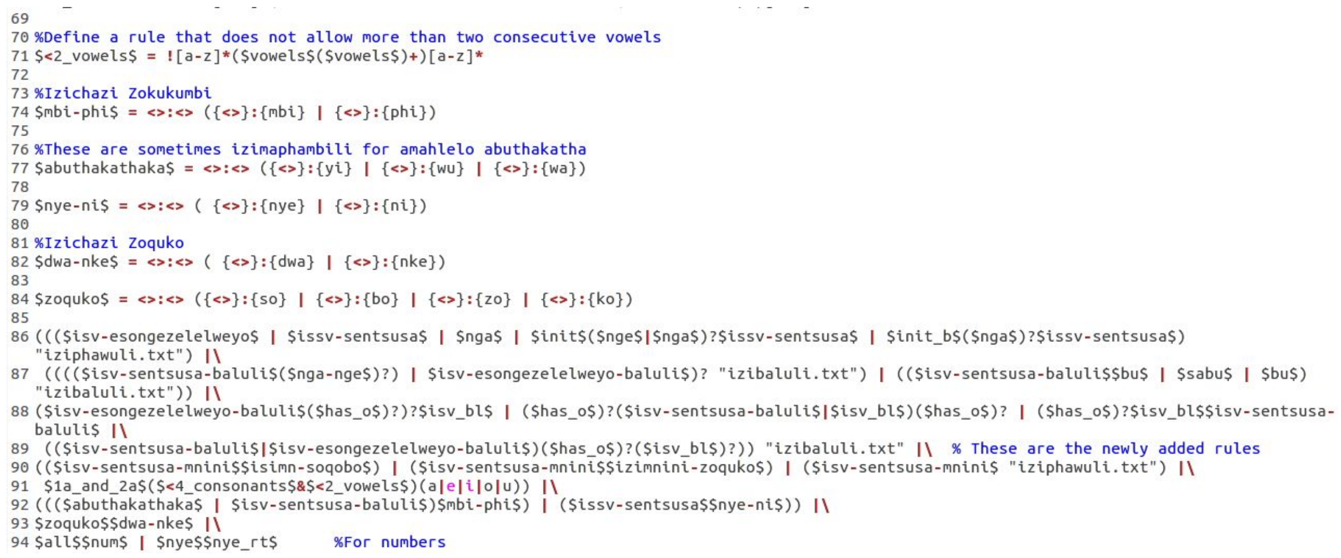
A3: Adjectives



```
%Iziphawuli
%Iziphawuli are called adjectives and their prefixes are called 'adjective concords'
%Descriptive corpulative concords
$issv-sentsusa$ = <>:<> ({<>}:{m} | {<>}:{ba} | {<>}:{m} | {<>}:{mi} | {<>}:{li} | {<>}:{ma} | {<>}:{si} |\
        {<>}:{zi} | {<>}:{in} | {<>}:{zin} | {<>}:{lu} | {<>}:{bu} | {<>}:{ku})

%noun prefix when combined with 'a' they become the following 'adjective concords'
$isv-esongezelelweyo$ = <>:<> ({<>}:{om} | {<>}:{aba} | {<>}:{om} | {<>}:{emi} | {<>}:{eli} | {<>}:{ama} |\
        {<>}:{esi} | {<>}:{ezi} | {<>}:{en} | {<>}:{ezin} | {<>}:{olu} | {<>}:{obu} | {<>}:{oku})

$init$ = <>:<> ({<>}:{si} | {<>}:{ni} | {<>}:{be} | {<>}:{e}) % These are prefixes which can be added before the subject concords combined with "nge"
$init_b$ = <>:<> ({<>}:{ku} | {<>}:{lu} | {<>}:{bu} | {<>}:{a} | {<>}:{ba}) % These are prefixes which can be added before the subject concords combined with "nga"

$nge$ = <>:<> ({<>}:{nge})
$nga$ = <>:<> ({<>}:{nga})

%These are for the first 6 numbers
$num$ = <>:<> ({<>}:{nye}|{<>}:{bini}|{<>}:{thathu}|{<>}:{ne}|{<>}:{hlanu}|{<>}:{thandathu})
$all$ = <>:<> ({<>}:{ama}|{<>}:{kwama}|{<>}:{zezi}|{<>}:{aba}|{<>}:{isi}|{<>}:{lesi}|{<>}:{oma}|{<>}:{sesi})
$nye_rt$ = <>:<> ({<>}:{nye})
$nye$ = <>:<> ({<>}:{ngasi}|{<>}:{nge}|{<>}:{nga}|{<>}:{kwe}|{<>}:{kwezi}|{<>}:{kwaba}|{<>}:{kuma}|{<>}:{ngama})

%===========================================================================================================================

%Izibaluli
$isv-sentsusa-baluli$ = <>:<> ({<>}:{ndi} | {<>}:{si} | {<>}:{u} | {<>}:{ni} | {<>}:{u} | {<>}:{ba} | {<>}:{u} | {<>}:{i} |\
        {<>}:{li} | {<>}:{a} | {<>}:{si} | {<>}:{zi} | {<>}:{i} | {<>}:{zi} | {<>}:{lu} | {<>}:{bu} | {<>}:{ku})

$isv-esongezelelweyo-baluli$ = <>:<> ({<>}:{endi} | {<>}:{esi} | {<>}:{o} | {<>}:{eni} | {<>}:{o} | {<>}:{aba} | {<>}:{o} | {<>}:{e} |\
    {<>}:{eli} | {<>}:{a} | {<>}:{esi} | {<>}:{ezi} | {<>}:{e} | {<>}:{ezi} | {<>}:{olu} | {<>}:{obu} | {<>}:{oku}) %ihlelo_15

$isv_bl$ = <>:<> ({<>}:{ku}|{<>}:{bu}|{<>}:{ba})
$has_o$ = <>:<> ({<>}:{ko}|{<>}:{bo}|{<>}:{ngo}|{<>}:{no})
```

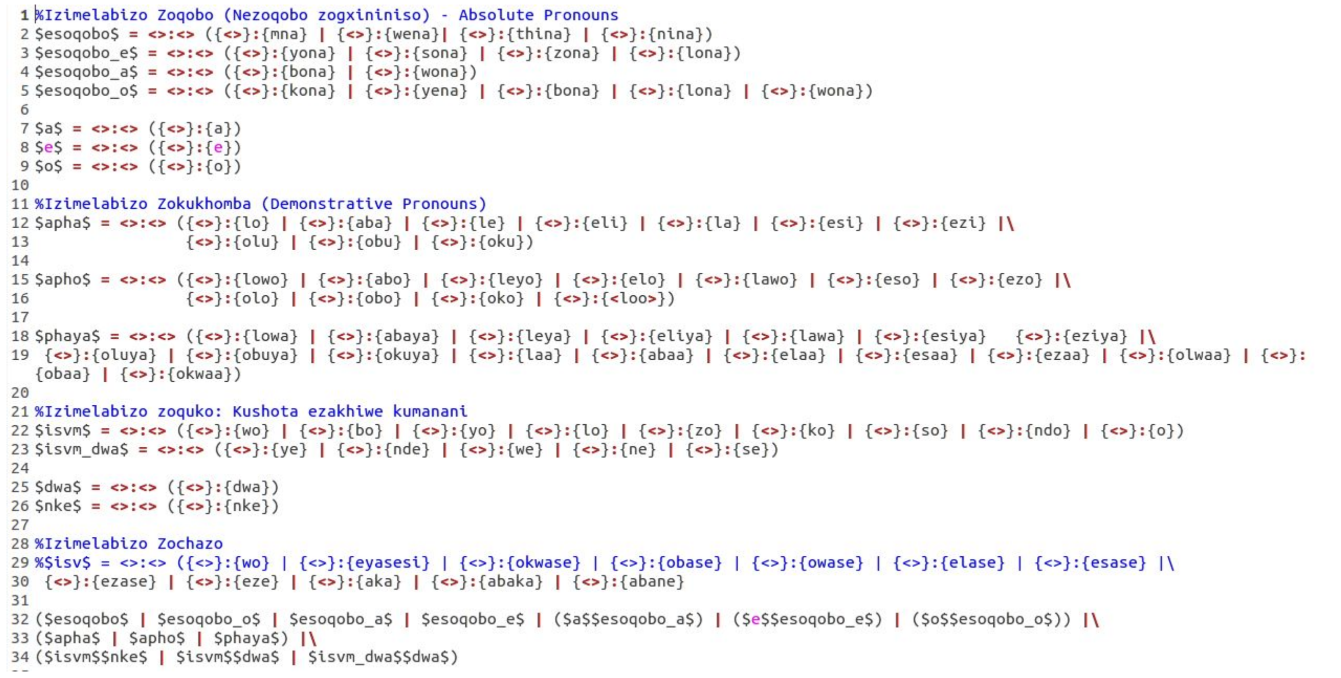


```
$has_o$ = <>:<> ({<>}:{ko}|{<>}:{bo}|{<>}:{ngo}|{<>}:{no})

$nga-nge$ = <>:<> ({<>}:{nga} | {<>}:{nge})

%Ubume obungafezekanga
$bu$ = <>:<> ({<>}:{bu})
$sabu$ = <>:<> ({<>}:{sabu})
$rha$ = <>:{rha}

%Izimnini
$1a_and_2a$ = <>:<> ({<>}:{ka} | {<>}:{baka} | {<>}:{lika} | {<>}:{sika} | {<>}:{zika} | {<>}:{luka} | {<>}:{buka} | {<>}:{kuka})

$isv-sentsusa-mnini$ = <>:<> ({<>}:{wa} | {<>}:{ba} | {<>}:{wa} | {<>}:{ya} | {<>}:{la} | {<>}:{a} | {<>}:{sa} |\ %ihlelo_7
        {<>}:{za} | {<>}:{ya} | {<>}:{za} | {<>}:{lwa} | {<>}:{ba} | {<>}:{kwa})

%Iziqu zobunini ezakhiwe kwizimelabizo zoqobo
$isimn-soqobo$ = <>:<>  ({<>}:{m} | {<>}:{khe} | {<>}:{bo} | {<>}:{wo} | {<>}:{yo} | {<>}:{lo} | {<>}:{so} | {<>}:{zo} | {<>}:{ko})

$isimn-soqobo-b$ = <>:<>  ({<>}:{ithu} | {<>}:{ikho} | {<>}:{inu})

%izimnini ezakhiwe kwisimelabizo sokwalatha
%"izimnini_zokwalatha.txt"

%Izimnini ezakhiwe kwizimelabizo zoquko
$izimnini-zoquko$ = <>:<> ({<>}:{dwa} | {<>}:{nke})

%Izimnini ezakhiwe kwizimnini e.g wesamadoda, kweyezikolo
%$isv-sentsusa-mnini$+ isimnini

%recognising izimnini zezibizo zika class 1a and 2a
ALPHABET = [a-z]
$non-vowels$ = ([a-z]&!(a|e|i|o|u))
$vowels$ = a|e|i|o|u

%Define a rule that does not allow more than three consecutive non-vowel letters
$<4_consonants$ = ![a-z]*($non-vowels$$non-vowels$$non-vowels$($non-vowels$)+)[a-z]*
```



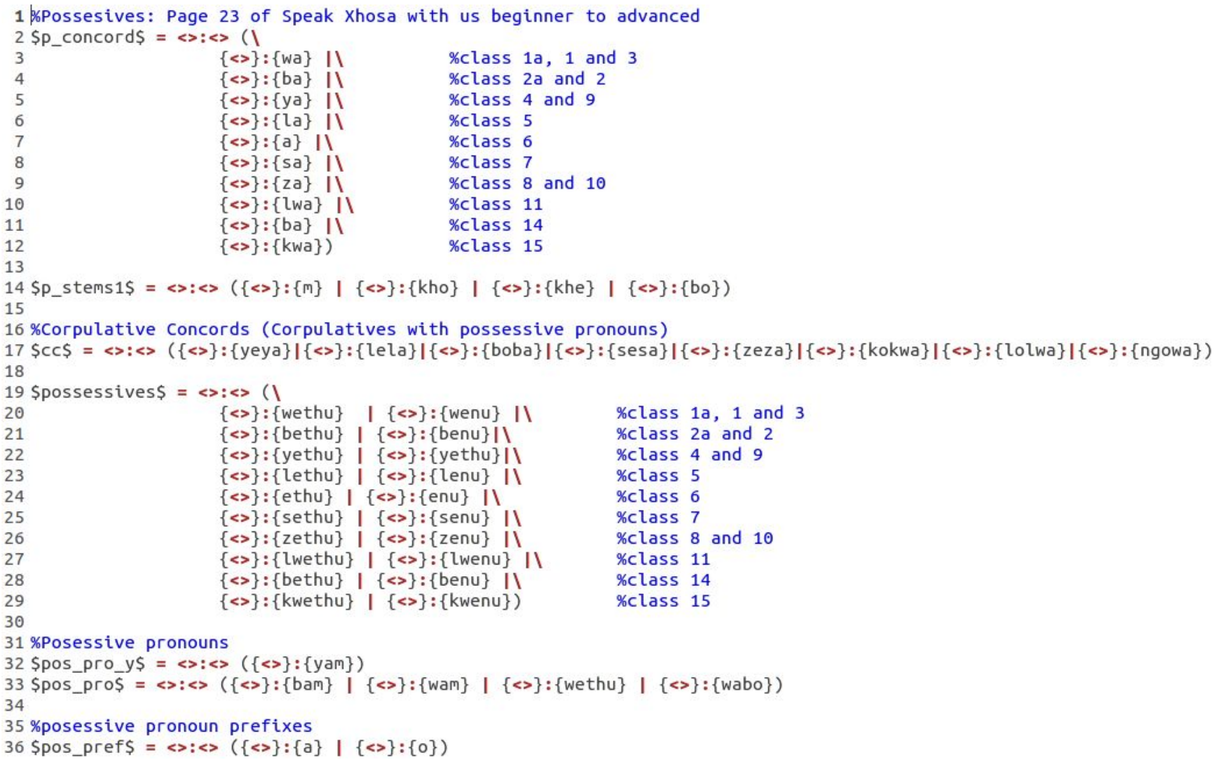
```

%Define a rule that does not allow more than two consecutive vowels
$<2_vowels$ = ![a-z]*($vowels$($vowels$)+)[a-z]*

%Izichazi Zokukumbi
$mbi-phi$ = <>:<> ({<>}:{mbi} | {<>}:{phi})

%These are sometimes izimaphambili for amahlelo abuthakatha
$abuthakathaka$ = <>:<> ({<>}:{yi} | {<>}:{wu} | {<>}:{wa})

$nye-ni$ = <>:<> ( {<>}:{nye} | {<>}:{ni})

%Izichazi Zoquko
$dwa-nke$ = <>:<> ( {<>}:{dwa} | {<>}:{nke})

$zoquko$ = <>:<> ({<>}:{so} | {<>}:{bo} | {<>}:{zo} | {<>}:{ko})

((($isv-esongezelelweyo$ | $issv-sentsusa$ | $nga$ | $init$($nge$|$nga$)?$issv-sentsusa$ | $init_b$($nga$)?$issv-sentsusa$) "iziphawuli.txt") |\
 (((($isv-sentsusa-baluli$($nga-nge$)?) | $isv-esongezelelweyo-baluli$)? "izibaluli.txt") | (($isv-sentsusa-baluli$$bu$ | $sabu$ | $bu$) "izibaluli.txt")) |\
($isv-esongezelelweyo-baluli$($has_o$)?)?$isv_bl$ | ($has_o$)?($isv-sentsusa-baluli$|$isv_bl$)($has_o$)? | ($has_o$)?$isv_bl$$isv-sentsusa-baluli$ |\
 (($isv-sentsusa-baluli$|$isv-esongezelelweyo-baluli$)($has_o$)?($isv_bl$)?)) "izibaluli.txt" |\  % These are the newly added rules
(($isv-sentsusa-mnini$$isimn-soqobo$) | ($isv-sentsusa-mnini$$izimnini-zoquko$) | ($isv-sentsusa-mnini$ "iziphawuli.txt") |\
 $1a_and_2a$($<4_consonants$&$<2_vowels$)(a|e|i|o|u)) |\
((($abuthakathaka$ | $isv-sentsusa-baluli$)$mbi-phi$) | ($issv-sentsusa$$nye-ni$)) |\
$zoquko$$dwa-nke$ |\
$all$$num$ | $nye$$nye_rt$      %For numbers
```

A4: Pronouns

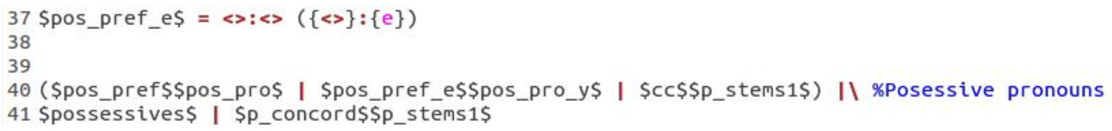


```
%Izimelabizo Zoqobo (Nezoqobo zogxininiso) - Absolute Pronouns
$esoqobo$ = <>:<> ({<>}:{mna} | {<>}:{wena}| {<>}:{thina} | {<>}:{nina})
$esoqobo_e$ = <>:<> ({<>}:{yona} | {<>}:{sona} | {<>}:{zona} | {<>}:{lona})
$esoqobo_a$ = <>:<> ({<>}:{bona} | {<>}:{wona})
$esoqobo_o$ = <>:<> ({<>}:{kona} | {<>}:{yena} | {<>}:{bona} | {<>}:{lona} | {<>}:{wona})

$a$ = <>:<> ({<>}:{a})
$e$ = <>:<> ({<>}:{e})
$o$ = <>:<> ({<>}:{o})

%Izimelabizo Zokukhomba (Demonstrative Pronouns)
$apha$ = <>:<> ({<>}:{lo} | {<>}:{aba} | {<>}:{le} | {<>}:{eli} | {<>}:{la} | {<>}:{esi} | {<>}:{ezi} |\
                {<>}:{olu} | {<>}:{obu} | {<>}:{oku})

$apho$ = <>:<> ({<>}:{lowo} | {<>}:{abo} | {<>}:{leyo} | {<>}:{elo} | {<>}:{lawo} | {<>}:{eso} | {<>}:{ezo} |\
                {<>}:{olo} | {<>}:{obo} | {<>}:{oko} | {<>}:{<loo>})

$phaya$ = <>:<> ({<>}:{lowa} | {<>}:{abaya} | {<>}:{leya} | {<>}:{eliya} | {<>}:{lawa} | {<>}:{esiya}   {<>}:{eziya} |\
 {<>}:{oluya} | {<>}:{obuya} | {<>}:{okuya} | {<>}:{laa} | {<>}:{abaa} | {<>}:{elaa} | {<>}:{esaa} | {<>}:{ezaa} | {<>}:{olwaa} | {<>}:{obaa} | {<>}:{okwaa})

%Izimelabizo zoquko: Kushota ezakhiwe kumanani
$isvm$ = <>:<> ({<>}:{wo} | {<>}:{bo} | {<>}:{yo} | {<>}:{lo} | {<>}:{zo} | {<>}:{ko} | {<>}:{so} | {<>}:{ndo} | {<>}:{o})
$isvm_dwa$ = <>:<> ({<>}:{ye} | {<>}:{nde} | {<>}:{we} | {<>}:{ne} | {<>}:{se})

$dwa$ = <>:<> ({<>}:{dwa})
$nke$ = <>:<> ({<>}:{nke})

%Izimelabizo Zochazo
%$isv$ = <>:<> ({<>}:{wo} | {<>}:{eyasesi} | {<>}:{okwase} | {<>}:{obase} | {<>}:{owase} | {<>}:{elase} | {<>}:{esase} |\
 {<>}:{ezase} | {<>}:{eze} | {<>}:{aka} | {<>}:{abaka} | {<>}:{abane}

($esoqobo$ | $esoqobo_o$ | $esoqobo_a$ | $esoqobo_e$ | ($a$$esoqobo_a$) | ($e$$esoqobo_e$) | ($o$$esoqobo_o$)) |\
($apha$ | $apho$ | $phaya$) |\
($isvm$$nke$ | $isvm$$dwa$ | $isvm_dwa$$dwa$)
```

A5: Possessives



```
%Possesives: Page 23 of Speak Xhosa with us beginner to advanced
$p_concord$ = <>:<> (\
                  {<>}:{wa} |\          %class 1a, 1 and 3
                  {<>}:{ba} |\          %class 2a and 2
                  {<>}:{ya} |\          %class 4 and 9
                  {<>}:{la} |\          %class 5
                  {<>}:{a} |\           %class 6
                  {<>}:{sa} |\          %class 7
                  {<>}:{za} |\          %class 8 and 10
                  {<>}:{lwa} |\         %class 11
                  {<>}:{ba} |\          %class 14
                  {<>}:{kwa})           %class 15

$p_stems1$ = <>:<> ({<>}:{m} | {<>}:{kho} | {<>}:{khe} | {<>}:{bo})

%Corpulative Concords (Corpulatives with possessive pronouns)
$cc$ = <>:<> ({<>}:{yeya}|{<>}:{lela}|{<>}:{boba}|{<>}:{sesa}|{<>}:{zeza}|{<>}:{kokwa}|{<>}:{lolwa}|{<>}:{ngowa})

$possessives$ = <>:<> (\
                  {<>}:{wethu}  | {<>}:{wenu} |\        %class 1a, 1 and 3
                  {<>}:{bethu} | {<>}:{benu}|\          %class 2a and 2
                  {<>}:{yethu} | {<>}:{yethu}|\         %class 4 and 9
                  {<>}:{lethu} | {<>}:{lenu} |\         %class 5
                  {<>}:{ethu} | {<>}:{enu} |\           %class 6
                  {<>}:{sethu} | {<>}:{senu} |\         %class 7
                  {<>}:{zethu} | {<>}:{zenu} |\         %class 8 and 10
                  {<>}:{lwethu} | {<>}:{lwenu} |\       %class 11
                  {<>}:{bethu} | {<>}:{benu} |\         %class 14
                  {<>}:{kwethu} | {<>}:{kwenu})         %class 15

%Posessive pronouns
$pos_pro_y$ = <>:<> ({<>}:{yam})
$pos_pro$ = <>:<> ({<>}:{bam} | {<>}:{wam} | {<>}:{wethu} | {<>}:{wabo})

%posessive pronoun prefixes
$pos_pref$ = <>:<> ({<>}:{a} | {<>}:{o})
```



```
$pos_pref_e$ = <>:<> ({<>}:{e})


($pos_pref$$pos_pro$ | $pos_pref_e$$pos_pro_y$ | $cc$$p_stems1$) |\ %Posessive pronouns
$possessives$ | $p_concord$$p_stems1$
```

Appendix B: Results for POS Category evaluation

B1: Nouns

	Test 1	Test 2	Test 3	Test 4	Test 5
POS category	Nouns	Verbs	Adjectives	Pronouns	Possessives
Number of words	20841	171	183	64	46
Inserted incorrect words	15	22	43	10	8
TP	18359	39	32	1	0
FP	2477	0	0	0	0
FN	0	33	30	3	3
TN	15	99	121	60	43
Time	2787ms	150ms	163ms	107ms	100ms

Accuracy	88.16%	80.70	83.6	95.37	93.47

B2: Verbs

	Test 1	Test 2	Test 3	Test 4	Test 5
POS category	Verbs	Adjectives	Pronouns	Possessives	Nouns
Number of words	171	183	64	46	20841
Inserted incorrect words	22	43	10	8	79
TP	133	6	15	0	2141
FP	27	0	0	0	0
FN	0	15	2	0	3
TN	11	162	47	46	18700
Time	156ms	138ms	109ms	106ms	1942ms
Accuracy	84.21	91.8	83.6	100	100

B3: Adjectives

	Test 1	Test 2	Test 3	Test 4	Test 5
POS category	Adjectives	Pronouns	Possessives	Verbs	Nouns
Number of words	183	64	46	171	20841
Inserted incorrect words	43	10	8	22	15
TP	125	6	5	0	71
FP	15	0	0	0	0
FN	0	0	1	0	8
TN	43	58	40	171	20762
Time	145ms	101ms	122ms	142ms	1944ms
Accuracy	91.8	100	97.8	100	99.96

B4: Pronouns

	Test 1	Test 2	Test 3	Test 4	Test 5
POS category	Pronouns	Possessives	Adjectives	Verbs	Nouns
Number of words	64	46	183	171	20841
Inserted incorrect words	10	8	43	22	15
TP	41	0	0	0	1
FP	6	0	0	0	0
FN	0	0	0	0	0
TN	17	46	183	183	20480
Time	104ms	98ms	137ms	171ms	1710ms
Accuracy	90.6	100	100	100	100

B5: Possessives

	Test 1	Test 2	Test 3	Test 4	Test 5
POS category	Possessives	Pronouns	Adjectives	Verbs	Nouns
Number of words	46	64	183	171	20841
Inserted incorrect words	8	10	43	22	15
TP	38	0	0	0	1
FP	0	0	0	0	0
FN	0	0	0	0	0
TN	8	64	183	171	20480
Time	464ms	880ms	143ms	134ms	1794ms
Accuracy	100	100	100	100	100