Context Based Afaan Oromo Language Spell Checker For Handheld Device

A Thesis Presented

By

Henok Dawit Daniel

To

The Faculty of Informatics

Of

St. Mary's University

In Partial Fulfillment of the Requirements for the Degree of Master of Science

in

Computer Science

June, 2022

ACCEPTANCE

Context Based Afaan Oromo Language Spell Checker for Handheld Device

By

Henok Dawit Daniel

Accepted by the Faculty of Informatics, St. Mary's University, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Thesis Examination Committee:

_____

Internal Examiner
Michael Melese (PhD)

_____

External Examiner
Minale Ashagrie (PhD)

_____

Dean, Faculty of Informatics
Alembante Mulu (PhD)

Tuesday, June 21, 2022

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Henok Dawit Daniel
Full Name of Student

_____
Signature

Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as advisor.

Alembante Mulu (PhD)
Full Name of Advisor

_____
Signature

Addis Ababa

Ethiopia

June, 2022

Acknowledgment

In the beginning, I am grateful thanks to heavenly father my GOD who helped me to succeed in all my life long learns. All of my Efforts would have gone for naught if it had not been for his importunate help. Then I offer my respected thanks to my advisor Dr. Alembante Mulu, who has supported me throughout my thesis project with his patience and knowledge whilst for directing my research, reviewing my ideas, and providing feedback on my work , allowing me the room to work in my own way and encouragement.

I would also like to give special thanks to my family for their unconditional love and support. My family members are always there to Support me in every situation. On top all, my lovely wife Etalem Tesfaye, My daughter Anna Henok and My Son Akiya Henok deserve very grateful thanks because without them I would not be who I am today.

Though it is difficult to mention the name of persons who gave me their hand while doing this Thesis, it is necessary to mention those who gave me their precious time to read the thesis Document, to share ideas, and gave me moral and material support Miss Tenanye Danye, Miss Mamite Mangesha ,Miss Sinidu Tasfaye ,Mr. Abenazer Girma and Mr. Ababaw Amare and Mr. Diriba Kiso, are few of them. I am very grateful to thank them for what they did.

Last but not least, I would like to sincerely thank all of my friends, colleagues, and classmates for their assistance, encouragement, and inspiration during this research.

Contents

List of Acronyms

CBAOSCHD: Context Based Afan Oromo language spelling correction for hand held device

FN: False Negative

FP: False Positive

HASCH: High Performance Automatic Spell Checker

IBM: International Business Machine Corporation

MED: The minimum edit distance

MT: Machine Translation

NLG: Natural language generation

NLP: Natural Language Processing

OCR: Optical Character Recognition

POST: Part-Of-Speech Tagging

QA: Question and Answering

SR: Speech Recognition

TN: True Negative

TP: True Positive

List of Figures

List of Tables

Abstract

Spellchecking is a spelling check app that will carefully go through your text to scan it for any spelling errors and correct them by providing possible ranked suggestion for user to select from list and fix misspelled words. This thesis describes the design architecture, implementation and testing of a model that have been developed by a programing language Python. This spellchecker came with an integrated user friendly graphical user interface, where users can input their text, detect misspelled words and choose from a list of five candidate correction words to correct them. Users can even add words to a pre-built dictionary. Error detection is based on the dictionary look up method, bigram and trigram analysis. The data collected from the different scientifically and error free as well as trusted sources and prepare the dictionary, bigram and trigram model for error detection and correction. Two types of error happened in spelling check system to detect and correct both context aware/ real word and non-word error types. The main focus of this study is to design context based spell checker for Afan Oromo language hand held devices depends on the spelling error patterns of language based on the sequence of words in the input sentences contextually.

The first types of spelling error that is non-word error candidate generation is based on dictionary lookup techniques, similarity is measured using the Levenshtein edit distance by considering Insertion, deletion, substitution and transposition of character of user input to the dictionary token and ranking top 5 probable suggestions accordingly. The second types of errors occur during spell cheek that is the real word error, for this types of error the bigram and trigram model created from the corpus and Stord based on statically/probabilistic analysis techniques was used to identify the misspelled word based on context to correct bad word according to context misspelled. To conduct experiment 1500 words were used to learn and test the model respectively. Experiment result shows that, the accuracy of 85% for spelling errors. According to gated result the accuracy of the system is 85%, this shows that the model is convenient and efficiency in order to correct misspelling Afan Oromo words both real word and nor word types of spell error occurred while user type texts to communicate.

Key words: *Context-Based Spellchecker, Real-word Error, N-gram, Levenshtein edit distance and natural language process (NLP).*

Chapter One

1. Introduction

1.1 Background

Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software. It also described as a subfield of computer science, linguistics, and artificial intelligence it deals with the interactions between human being language and computers, in other word how to program computers to process and analyze big amounts of natural language data. The ultimate goal of NLP is a computer ability to understand the contents of documents, including the contextual nuances of the human language within them. The technology can then accurately retrieve information and insights contained in the documents as well as categorize and organize the documents themselves **[48]**.

Natural language processing has many applications areas some of them are Speech recognition, Text-to-speech, segmentation of Speech, text summarization /Automatic summarization, Dialogue management ,Speech Recognition (SR) , Book generation , Machine Translation (MT), Part-Of-Speech (POS) Tagging and from natural language process application Spell checker is also an application area **[25]** and started early mid of 20th century by Lee Earnest at Stanford University, USA; but the first application was created by Ralph Groin, who is Lee's student. He uses a dictionary of 10,000 English words and design and develop an application of spell checker based on rule based method for correction system.

What is spell checking?

Spelling checking is a set of program written and tool for correcting misspelled spelling of a word. It's available in programs like word processing, text editor tools, email programs, cell phones, and a variety of other applications, such as social blogs and forums. Spell check lets you know when words are misspelled, corrects misspelled words as you type, and allows you to search a whole document for misspelled words **[13]**.

Date back to 1980s, a spell checker is more like a verifier. It has no corresponding suggestions to the spelling error detected. As many of the readers are Using word processor nowadays, a spell checker will first mark a word as mistaken (Detection) and give a list of replacement of word

(Suggestion). Therefore the definition of spell checking involve more than only checking, it is the process of detecting misspelled words in a document/sentence and suggest with a suitable word in the context **[14]**.

Therefore, to construct a spell checker, it needs to have the following features:

1. Spelling Detection is program that have to capable of detect a word error

2. Spelling Suggestion or correction is a program that have the ability to suggest a suitable word to users which matches their need in context

The Spell-checking in general defined as the process of detecting and suggesting incorrect spelled words in a contextual text. Spell checking system first detects the misspell words and from a candidate it suggests correct answers. Spell checking system is set of standard rules of the languages for which spell checking system is to be developed and a dictionary that contains the correct spellings of various words. Better rules and a large dictionary of words help to improve the rate of error detection otherwise all the errors cannot be detected. After wrong or misspelled words, the various suggestions are given. There are many systems available for detecting and correcting the text. The system is made to check the spellings from the list of words in a text file.

My motivation comes from first I have been one of the language user individual and as working language in and looking many error in SMS, social media, vacancy announcement, news, banner and letters written in Afan Oromo language mainly when using mobile phone keyboard is to small and experienced and literate also making mistake du to keyboard smallness. The second case is morphological and nature of the language some of the feature of the language was described as follows: Afaan Oromo language is an Afro-asiatic language that belongs to the Cushitic branch. It is native to the Ethiopian state of Oromia and spoken predominantly by the Oromo people and neighboring ethnic groups in the Horn of Africa. With 33.8% Oromo speakers, followed by 29.3% Amharic speakers, Afaan Oromo is the most widely spoken language in Ethiopia [7]. It is also the most widely spoken Cushitic language and the fourth-most widely spoken language of Africa, after Arabic, Hausa and Swahili. Forms of Oromo are spoken as a first language by more than 35 million Oromo people in Ethiopia and by an additional half-million in parts of northern and eastern Kenya It is also spoken by smaller numbers of emigrants in other African countries such as South Africa, Libya, Egypt and Sudan. Afaan Oromo is also

the working language of several of the states within the Ethiopian federal system including Oromia, Harari and Dire Dawa regional states and of the Oromia Zone in the Amhara Region. It is a language of primary education in Oromia, Harari, Dire Dawa, Benishangul-Gumuz and Addis Ababa and of the Oromia Zone in the Amhara Region [7].

1.2 Statement of the Problem

Know a day the electronic computing technology is very attached with human being and part of our day-to-day life. The paradigm of computing technology is shifting towards hand held mobile phone devices **[45]**. Currently, these handheld devices are becoming widely used around the world including our country. While this device used everywhere around the world Introducing texts to word processing tools in the handheld electronics phone devices may have result in spelling errors. Hence, various text processing programed tools has spellcheckers capability in communication platform. So that integrating technology of spellchecker into mobile phone devices is very important to increase the quality of information exchange and efficiency.

The applications of spelling check and correction for different devices are mostly likely with foreign languages and for resourceful languages around the globe **[15]**. If these handheld devices can provide their services in the local languages Afaan Oromo in our country and beyond, they will gain wide acceptance among the users and motivate other researcher to develop more application using the local language Afaan Oromo. Fast and error free context based spelling checker method is important thing for Afaan Oromo language user on handheld device like mobile phone. However, these tools are not available for the Afaan Oromo language. To improve the quality of life for the users specially by creating a mobile application that will help them communication effectively. As Many Afaan Oromo language speakers uses mobile phones, Why those Afan Oromo users cannot make their language part of the technology‘s language? And developing such application for low resourced language fosters the advancement of the Afan Oromo language **[16]**. In addition to that, the language can serve as an alternative text entry method for mobile phone like SMS. Therefore, this study proposed a system modeling for Afaan Oromo language spellchecker, detecting and correction for hand held device.

1.3 Research questions

1. How system accuracy and a suitable suggestion were given to a user for each type of spelling error?
2. What is the performance of the spell checker?
3. Is that efficient and convenient when errors are detected and corrected accurately?

1.4 Objectives

1.4.1 General objective

The main objective is to develop context based Afaan Oromo language Spell checker for handheld device.

1.4.2 Specific objectives

- To prepare Afaan Oromo corpus for training and testing the proposed model;
- To collect Afaan Oromo word/Dictionary;
- To study about the rule and writing of the language;
- To design hand held device spell checker and correction model for Afaan Oromo language phone user;
- To evaluate the performance of prototype spell checker;

1.5 Significance of the study

- To improve quality of Communication between users of this language
- To develop Afaan Oromo language Spell checker mobile application that have the ability to detect a word error
- To suggest a suitable word for correction to users which matches their need in context
- To identifies words that are valid in Afaan Oromo language, as well as the misspelled words in the language to be suggested for Correction.
- To Minimize an error while exchanging text between user of the language
- To fosters the advancement of the Afan Oromo language.

1.6 Scope and limitations

This research was scoped to develop a context based spell checking for hand held devices for Afan Oromo language based on statistical frequency of words by using bigram and trigram statically/probabilistic techniques of word occurrence preceding to another one and dictionary look up method. The model focuses on the spelling errors as a result of typographical errors which may happen due to deletion, insertion, substitution, or translation of character as well as, real word error. Obviously, there are two types of error in spell checking. The first one is non-word error, this error was an error that occurs when the word didn't found in the list of the dictionary stored or prepare from corpus.

On other hand, other type of spelling error is real-word error, an error happens when the word is correctly written since it matches with the list of bigram and trigram model but it appears in wrong position in the sentence. Generally, in this study both non-word and real-word errors are considered. However, this study did not cover all Afan Oromo words for corpus preparation. As a result, the model didn't provide the candidate for all Afan Oromo misspelled words. As well as, the model didn't automatically correct the misspelled words even when one candidate list is provided, it need the user interaction to fill the candidate for the misspelled word. Also compounds and abbreviation didn't covered by the model, the system accept compound word as different words. For instance, 'Mana kitaabaa' which means library, the model accepted as 'mana' and 'kitaabaa'.

1.7 Organization of the Thesis

This thesis is organized in six chapters. **The first chapter**, presents out the introduction to spelling checker, definitions of spell check, motivation to conduct the thesis, statement of the problem, and the general and specific objectives of the study together with scope, limitations and significance of the study are included.

Chapter two: Is literature review and it involves the following main tasks: History of spell checking, types of spelling error, Spelling Error Detection Techniques, Spelling Error Corrections Techniques, and Performance of spellchecker Evaluation method and work related to spell check and correction system and conceptual review. Conceptual review is review on basic

norms of context based spell check, concepts of spell check and correction, process of spell check and related topics. Related work involves work done so far on the research topic mainly related to the title of this thesis. Finally in chapter two of this thesis research gap is identified.

Chapter three: Discusses about nature of Afan Oromo language and the methodology used in this research as well as the main task of spelling cheek and correction system for Afan Oromo language context aware and non-word spelling errors.

Chapter four: This chapter describes method applied in this research, model architecture, techniques used and algorithm selected for context based spell checker for Afan Oromo language for hand held device.

Chapter five: In this chapter, the tools used to implement and algorithm that are describe in previous section to design a model and the experiment was conducted to demonstrate the spelling error detection and correction accuracy. The result of the experiment would be interpreted in this section and the performance of the spelling error detection and correction could be evaluated using evaluation method. Precision and recall were used to evaluate the accuracy, effectiveness and validity of detecting and correcting spelling errors based on the training and testing texts that have been used in this experiment. Findings of the study and issues in implementations are discussed in detail.

Finally in the **chapter six** major findings including faced challenges are written as a conclusion and works identified as future work and needs to get attention of other researchers are listed in recommendation section. So this chapter presents our conclusive remarks and recommendations for future work list down and discussed, beside this at last of the thesis list of reference, appendix of sample code, training data included.

Chapter Two

Literature Review and Related Works

Chapter two of this research deals with the state of the art relating to context based spell checking and correcting system for Afan Oromo language with its spelling error types in the written texts and techniques to identify the misspelled words and correcting mechanism.

2. Literature review

2.1 Introduction

Human being living around the world is needed to communicate in every day to day life for the interaction there is unique gift that is Language for humankind. It conveys collections of many data using rules and standards with some amount of informality. Adaptation to the informality and develop new rules to make sense from the informality is learned by human being quick **[47]**. However, understanding the informality in human language has been a challenging task for computer set of rule and standards written for program or software. Nowadays, software of many of language processing tools and architectures are still have some hole of vulnerability against all rank of informal use of the language. Output of a sophisticated natural language processing for tasks that enjoy very small error rates for simple spelling error can greatly impact the nlp tools we can use **[48]**.

Most of models for NLP currently used are not satisfy all users need in manner of enough while processing row input data collected from many sources However, errors such as grammatical mistakes, incorrect usage of words, Misspelled word and spelling errors are happened through human generated texts while collected many corpus from different sources. Subsequent activity of language during preprocessing text many of time affected when the corpus collected by human have many of mistakes collectively can came up many types of noise included in compiled text. In other case, reducing non normalization in the collected corpus by correcting suddenly by unknowingly created human mistakes used as an important step in increasing the performance models that model is said to be of machine learning model **[19]**. This paper focuses on contextual based error correction of misspelled words one of the most common type of errors

in natural language texts. In context of NLP errors of spelling can be mainly divided into two types, real-word error spelling error and non-word spell error **[49].**

Around the world a number of spelling checkers and correction system have been developed so far for many more languages. Among the most notable spelling checkers are those developed over the past few years for resourceful languages such as English; Spell checker in CET Designer [30], Spell Cheekers and Correctors [31], etc.

From Among the most notable grammar and spelling checkers system are those developed over the past few years for languages such as Amharic, Afaan Oromo And Tigrigna language: A rule-based Afaan Oromo Grammar Checker [1], system that developed for Amharic language checking the grammar by using morphological features of words and ngram based probabilistic methods [2] , context sensitive checking of spelling for the Afaan Oromo language writing system [9] and rule based Tigrigna language spellchecker and corrector system [10], etc.

A lot of work has gone into developing sophisticated systems that have gone into widespread use, such as grammar checking, speech to text convert, automatic translators and spell checkers from this developed applications most such software program developed are strictly for commercial purpose beside this there is no documentation of the algorithms and rules used available everywhere so that, For languages such as Afaan Oromo there is lacking of advanced tools developed and the language are technological resource lacking and still in the early stages of development. In day to day activity user use many tools from this tool one of the most widely used grammar checkers for English tools, like Microsoft Office Suite grammar checker and corrector, is also not above controversy. It shows that work on Spell checker and correction applications in real time is not as such a very easy task so that imitating the implementation task for language of Afan Oromo is a best feat and to the best of my knowledge, there is no Afaan Oromo Spell checker or published article that presents spell checking for Afaan Oromo language in our mobile phone devices. This thesis presents Context Based Afaan Oromo language Spell checker for handheld device.

2.2 History of Spellchecker

Spellchecker is an application program that identify the misspelled words from a given text **[26]** It may be stand-alone, or integrated with the other application, such as a, search engine, email

client, electronic dictionary, or word processor. Application of spellchecking is not new, it start 20th century, the first spell checker application was developed for mainframe computers in 1970s. It support group six languages for the IBM (International Business Machine Corporation) Company. The main function of this application is that it only shows the error instead of the correct word for replacing the misspelled one. Moreover, for personal computer spell checkers first introduced on 1981 from the IBM Company. By the mid-1980s many popular word processing applications (such as WordStar and WordPerfect) had integrated spell checkers

The main tasks of a spellcheckers are preprocessor (tokenization), error detection and correction, and ranking the suggestions. Error detection step focus on to identify and display misspelled words from the text. While, error correction module provide candidate suggestion for misspelled words to represent error words with correct words and ranking candidate suggestion.

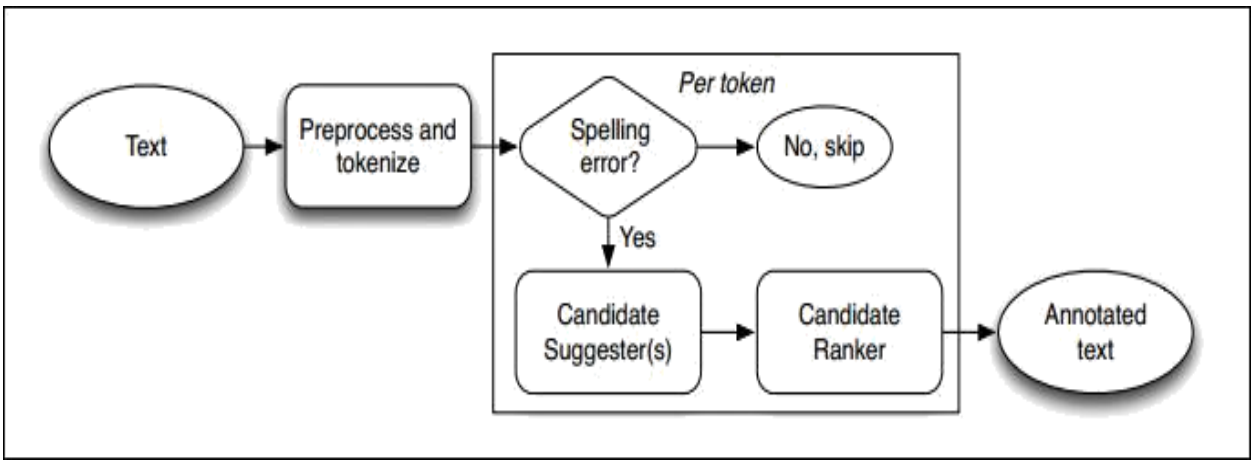

Figure 1.1: Spelling check procedure

2.3 Text Error

A word can be mistaken in two ways: the first is by incorrect spelling a word due to lack of enough information about the word spell or intentionally mistaking symbol(s) within the word, this type of error is known as non-word errors where the word cannot found in the language lexicon. The second type of mistake is using correctly spelled word in wrong position in the sentence or unsuitable context. These errors are known as real-word errors where the incorrect word is accepted in the language lexicon **[38]**.

2.3.1 Non-Word Errors

Non-words errors are spelling errors that not found in the list of words in the dictionary. In non-word error, a word may incorrectly type because there is extra space, extra character, misspelled word, or other possibilities. These errors are easier to detect, because just comparing the words in a text with the entries in a dictionary will filter out the erroneous words 80% of misspelled words that are non-word errors are the result of a single insertion, deletion, substitution or transposition of letters **[9]**:

- **Insertion.** Adding an extra letter, e.g., '*laekki'* instead of '*lakki'* which means no. An important special case is a repeated letter, e.g. '*deemmi'* instead of '*deemi'* relate with go in English context.
- **Deletion.** Missing a letter, e.g., *'tle' instead of 'tole'*. An important special case is missing a repeated letter, e.g., *'eyee instead of 'eyyee''*.
- **Substitution.** Substituting one letter for another, e.g., *'ejjenni' instead of 'ejjenno'*. The most common substitutions are incorrect vowels.
- **Transposition.** Swapping consecutive letters, e.g. '*jiaarchuu'* instead of '*jiraachuu'*.

Non-word spell checker relies on the prepared dictionary to detect the error as well as, to correct the misspelled words by providing the candidates; non-word error detection is used by the dictionary lookup methods. Dictionary is list of the word prepared from the collected corpus and accepted as the corrected or free non-error words. To correct the misspelled words most models used Levenshtein distance to generate candidate by measuring the similarity between misspelled and corrected word.

2.3.2 Real Word Error

These errors occur through mistaking an intended word by another on that is dictionary accepted. It is correctly spelled word in wrong position in the sentence or unsuitable context. This error is error where the incorrect word is accepted in the language lexicon and can be resulted from Cognitive errors.

The idea centers on generating candidate spellings for every misspelled word by applying simple edit operations such as insertion, deletion, and substitution, and then using *n*-gram statistics derived from a corpus by computing the probability of word. The spell checker of the recently released Microsoft Word 2007 is able to detect and correct some real-word mistakes. Concerning the correction techniques, context sensitive error correction deals with real-word error and mostly solved by using the n-gram analysis. Generally, correcting real word errors is context based in that it needs to check the surrounding words and sentences before suggesting candidates **[38]**.

2.4 Spelling Error Detection Techniques

2.4.1 Dictionary Lookup Technique

This technique is the first technique and work according to the definition given below to identify misspell identification. Dictionary lookup technique is used which checks every word of input text for its presence in dictionary. If that word present in the dictionary, then it is a correct word. Otherwise it is put into the list of error words. The most common technique for gaining fast access to a dictionary is the use of a Hash Table. To look up an input string, one simply computes its hash addresses and retrieves the word stored at that address in the pre- constructed hash table. If the word stored at the hash address is different from the input string or is null, a misspelling is indicated [33].

2.4.2 N-Gram Analysis

In this N- Gram analysis misspell identification based on as defined here that means N-gram are n-letter sub sequences of words or strings where n usually is one, two or three. One letter ngrams are referred to as unigrams or monograms; two letter n-grams are referred to as bi-grams and three letter n-grams as trigrams. In general, n-gram detection technique work by examining each n-gram is an input string and looking it up in a precompiled table of n-gram statistics to ascertain either its existence or its frequency of words or strings that are found to contain nonexistence or highly infrequent n-grams are identified as either misspellings.

2.5 Error Correction Techniques

Error correction Techniques is the method which the spell checker find out the candidate list for the misspelled words. Numerous approaches/ Techniques are proposed to correct spelling errors, such as Minimum Edit Distance Technique, Rule based technique, Similarity Keys, Probabilistic techniques.

2.5.1 A Minimum Edit Distance Technique

The minimum edit distance is the minimum number of operations (insertions, deletions and substitutions) required to transform one text string into another. In its original form, minimum edit distance algorithms require m comparisons between misspelled string and the dictionary of m words. After comparison, the words with minimum edit distance are chosen as correct alternatives. Minimum edit distance has different algorithms are Levenshtein algorithm, Hamming, Longest Common Subsequence.

2.5.1.1 The Levenshtein algorithm

This algorithm is a weighting approach to appoint a cost of 1 to every edit operations (Insertion, deletion and substitution). For instance, the Levenshtein edit distance between “dog” and “cat” is 3 (substituting d by c, o by a, g by t). 2).

2.5.1.2 The Hamming algorithm

This algorithm is measure the distance between two strings of equal length. For instance, the hamming distance between “sing” and “song” is 1 (changing i to o). 3).

2.5.1.3 The Longest Common Subsequence algorithm

This algorithm is a popular technique to find out the difference between two words. The longest common subsequence of two strings is the mutual subsequence. For instance, if i= 6750ABT4K9 and j= 0069TYA5L9 then LCS= 650AT9.

2.5.2 Similarity key technique

In this, Similarity key technique is to map every string into a key such that similarly spelled strings will have similar keys. Thus when key is computed for a misspelled string it will provide

a pointer to all similarly spelled words in the lexicon. A very early, often cited similarity key technique, the SOUNDEX system.

2.5.2.1 Soundex Algorithm

This algorithm is used for indexing words based on their phonetic sound. Words with similar pronunciation but different meaning are coded similarly so that they can be matched regardless of trivial differences in their spelling.

2.5.2.2 The SPEEDCOP System

It is a way of automatically correcting spelling errors predominantly typing errors in a very large database of scientific abstracts. A key was computed for each word in the dictionary. This consisted of the first letter, followed by the consonants letters of the word, in the order of their occurrence in the word, followed by the vowel letters, also in the order of their occurrence, with each letter recorded only once.

The Soundex code and SPEEDCOP key are ways of reducing to a manageable size the portion of the dictionary that has to be considered [33].

2.5.3 Rule Based Technique

Rule Based Techniques are algorithms that attempt to represent knowledge of common spelling errors patterns in the form of rules for transforming misspellings into valid words. The candidate generation process consists of applying all applicable rules to a misspelled string and retaining every valid dictionary word those results.

2.5.4 Probabilistic Techniques

In this, two types of Probabilistic technique have been exploited.

2.5.4.1 Transition Probabilities

They represent that a given letter will be followed by another given letter. These are dependent. They can be estimated by collecting n-gram frequency statistic on a large corpus of text from the discourse.

2.5.4.2 Confusion Probabilities

They are estimates of how often a given letter is mistaken or substituted for another given letter. Confusion probabilities are source dependent because different OCR devices use different techniques and features to recognize characters, each device will have a unique confusion probability distribution.

2.5.5 N-gram Based Techniques

Letter n-grams, including tri-grams, bi-grams and unigrams have been used in a variety of ways in text recognition and spelling correction techniques. They have been used by OCR correctors to capture the lexical syntax of a dictionary and to suggest legal corrections.

2.5.6 Neural Net Techniques

Neural nets are likely candidates for spelling correctors because of their inherent ability to do associative recall based on incomplete or noisy input.

2.5.6.1 Back Propagation Algorithm

This algorithm is the most widely used algorithm for training a neural net. A typical back propagation net consists of three layers of node: input layer, an intermediate layer, an output layer. Each node in the input layer is connected by a weighted link to every node in the hidden layer. Similarly each node in the hidden layer is denoted by a weighted link to every node in the output layer. Input and output information is represented by on-off patterns of activity on the input and output nodes of the net. A 1 indicates that a node is turned on and 0 indicates that a node is turned off [33].

2.6 Context based Spelling Checking Techniques

When we say Context based spelling error correction is the task of detecting and correcting errors that result in user type a correct spelled word when another intended to use a language for writing purpose. Because of the existence of real-word errors, we need to use a different approach in order to identify errors. When using the context for error detection, we do not only look at each individual word, but take the words surrounding it into account when deciding if some word is incorrect or not. We now take the two words, and try to produce similar words.

Context-sensitive spelling error correction tries to detect and correct such real-word errors by inspecting their structural contexts. For those words we calculate a probability that indicates how likely the words is occur with each other

2.6.1 Methods Based on Semantic Information

This approach was based on the observation that the words that a writer intends to use are semantically related to their surrounding words whereas some types of real-word errors. “It is my sincere (hope) that you will recover swiftly.” Such errors will result in a worry of the consistency and coherence of the text. According to this method use semantic distance measures in WordNet to detect words that are potentially anomalous in context that is, semantically distant from nearby words; if a variation in spelling results in a word that is semantically closer to the context, it is hypothesized that the original word is an error and the closer word is its correction.

2.6.2 Methods Based on Machine Learning

The machine learning method is regarded as a lexical disambiguation task and confusion set are used to model the ambiguity between words. The machine learning and statistical approaches are often based on pre-defined confusion sets which are sets of commonly confounded words, such as (their, there) and (principle, principal). These methods learn the characteristics of a typical context for each member of the set and detect situations in which one member occurs in context that is more typical of another. Such methods are limited to a set of common and predefined errors, but such errors can include both content and meaning words. Given an occurrence of one of its confusion set members, the spellchecker’s job is to predict which member of that confusion set is the most appropriate in the context.

2.6.3 Method Based on Probability Information

Moreover, according to Inkpen Mays proposed a statistical method using trigram (a sequence of three words) probabilities for detecting and correcting real-word errors without the need of requiring predefined confusion sets. In this method, if the trigram deduced that the probability of an observed sentence is lower than the sentence obtained by replacing one of the words with a spelling variation, then the original is a real-word error and the variation is what the user intended to use.

For current study, the probability information approach has been chosen, because as research has shown the accuracy of this method approach is relatively high according to this research Inkpen, as mentioned in the previous chapter, this study will focus Context-Based spell checking using bigrams probability. N-grams probabilistic was the appropriate algorithm to detect and correct real word error in spelling checker application. This technique detects errors by examining each n-gram from the give string looking it with a pre compiled n-gram statics table. N-gram techniques usually require either dictionary look up techniques or a large corpus of text in order to pre-compile an n-gram table. The major advantage of n-grams techniques are language independent which means didn't require knowing language knowledge to develop the spellchecking application. Additionally, n-gram help by providing probability information that estimate a given letter followed by another one to find a valid solution for real-word error [9].

2.7 Spellchecker Performance Evaluation method.

The performance measurement criteria of the context sensitive spelling error detection and correction experiment are divided into precision and recall, respectively, as shown in below Equation (1, 2, 3 and 4). Precision and recall are not different from the denominator values in the conventional equation; however, the value of the numerators is different. The numerators in the detection equation apply all cases where they are replaced by other candidates through probability value comparison, and the numerators in the correction equation apply the correct answer by selecting it from the values obtained from the detection [34].

2.7.1 Recall

Recall is calculated as the number of true positives divided by the total number of true positives and false negatives.

Recall = True Positives / (True Positives + False Negatives) ---------- (1)

2.7.2 Precision

Precision is calculated as the number of true positives divided by the total number of true positives and false positives.

Precision = True Positives / (True Positives + False Positives) ---------- **(2)**

F-measure is one of a performance measure that combines Recall and precision into a single measure of performance, this is just by taking into account the product of Precision and Recall divided by their sum.

The F-measure (or F1-score) can be used to represent more simply the previously obtained equations. The F-measure is also called a harmonious mean because it overcomes the imbalance of data and processes values with balanced data to calculate and adjust the same case in all cases. Because the precision and recall obtained under different conditions are unbalanced, the harmonic mean, which gives uniformity to the performance value, is highly reliable. The F-measure is expressed by Equation

F-Measure = (Precision*Recall)/ (Precision + Recall) ---------- (3)

2.7.3 Accuracy

Accuracy is a metric that generally describes how the model performs across all classes. It is useful when all classes are of equal importance. It is calculated as the ratio between the numbers of correct predictions to the total number of predictions.

$$Accuracy = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}}$$ ---------- (4)

Accuracy calculated from both precision and recall which measures the general quality of the spell checker.

2.8 Related work

In this title there are many different works reviewed from various sources such as Thesis, dissertations, web pages, electronics book, articles and journals. In related works cover mainly the spell checking and correction procedures, methodologies and techniques they used. What the researchers are developed and their contribution as well its achievement in spelling correction are reviewed as follows.

This study is about spelling and grammar checking based on morphology. According to this research, in other languages distance between words is not dependent on the family and order of characters except that of Amharic language. So that, using these approaches for Amharic language which is having complex morphology hence will not give the anticipated result. Because of this the analyzers of Amharic morphology that conducted before was to work with reasonable accuracy for valid words, their output for misspelled words to be corrected is not give possible suggestions. The model come up with the possibility of using morphology oriented approach to develop and design system for error detection and correction for non-word errors in the writing of Amharic language [29].

In the current research conducted developing a context aware spell checking and correcting method that is able to trace for misspell and suggest any kinds of human made real word errors types. Initial point for the context aware based spell checker and correction system is a lexicon based application to spelling correction based on error that detects and corrects non-word errors types. The contextual aware spelling checker and corrector can be combined with this lexicon based spelling correction application in order to create an application that is able to correct non-word errors as well as real-word errors both types of error occur in spelling checker applications. The method for this application for error detection and correction that is used three-word sequences i.e. trigrams rather than single token word based on this for the misspelling of a word often resulted in an unlikely sequence of three words sequentially occurred [24].

According to this study automatically to correct spell error According to this paper dataset can be developed synthetically for any non-resourceful language error types related to real world errors types. In sequence to sequence model the spelling check and correction system used deep learning techniques complete tasks of spell check. The studies develop a system of spelling

correction problem languages of Indic with the techniques of Deep learning. This developed model can be implemented for any low resource language. They finally test and evaluate their system Sequence-to-sequence text Correction Model for Indic Languages which trains end-to-end and perform well [23].

This study focused automatically to correct spell error According to this studies to Automatic check and correct spelling issues they used revised ngram spelling error correction tool in improving the effectiveness of retrieval a language independent spell checker and corrector presented this is based up on increasing the performance of the ngrams model they developed. On the spell check and correction process selecting the most promising candidates from a ranked list of correction candidates from many candidate suggestions based on lexical resources and ngrams probability of occurrence based on statistics to work properly the developed application. The proposed study shows that evaluation of the system developed showed that it perform well from other methods they compare during testing phase of the system life cycle [22].

According to the study about spelling check related dissertation, According to this dissertation a unified treatment of various spell checkers and correctors. The problems related to spell briefed in mathematics case to understand the spelling error and correction system to provide a better understanding about the issue concerned misspelled word and giving suggesting relevant candidate from list. An approach in which denotational semantics used to describe programming languages is adopted in developing the system. At the end, a mathematical investigations of many more spell checking and correcting packages take place and the researcher suggests a classification of these model in terms of their strategies of implementation take place ,functionalities of the system and performance of the system ,finally the system developed and tested working well as scope they bound [31].

This study spelling check and correction model, according to the research architecture design, implementation and testing of the proposed model that have been developed to detect and correct both types of spelling error occur while using language and write that types of error was non-word and real word types of error in every language. Focus area of proposed model was that design Context based spell checker for Afan Oromo writing for computer user depends on the spelling error patterns of language based on the sequence of words in the input sentences according to context. As technique the system used unsupervised statistical approach and

unsupervised statistical approach also used and helps to prepare manually tagged data sets to help under resource like Afan Oromo language from gathered corpus. According to this study there is three phases while spelling correction is undertaken, major phases are error detection, candidate suggestion and ranking candidate suggestion. Finally the testing is performing well and prominent score is gained to correct misspelled word in a context for writer of Afan Oromo language in computer machine [9].

This study is about automatically correcting the spelling error at the year 2020 According to this study The Spell checking is the process of detecting and suggesting incorrect spelled words in a paragraph. This developed Spell checking system initially find and detects if there is incorrect word in the written paragraph and then suggests correct answers if the system detect incorrect or misspelled word there. This system is a combination of many set language rules for which spell checking system is to be created and a dictionary that contains clear and error free words. The developed system was recommended that using better rules and a large and also pure dictionary of words help to improve the rate of error detection else all the errors not be detected and give relevant output in suggesting candidate for correction of word. After wrong or misspelled words, the various suggestions are given. There are many systems available for detecting and correcting the text. The system is made to check the spellings from the list of words in a text file [18].

According to this paper takes method known as Yarowsky's method as initial point as well hypothesizes that further improvements can be obtained by taking into account not only the single strongest piece of evidence, but all the available evidence. Yarowsky's method is presented for doing spelling check and correction system, this method is developed based on Bayesian classiers. The work reported here was applied not to accent restoration, but to a related lexical disambiguation task for context aware spelling detection and suggest for misspelled word according to context. The task is to correct spelling errors that happen in valid words considered at the label of the lexicon finally the system tested and perform well in correcting spelling based on context sensitive task [20].

According to this study errors that occur in grammar in order to detect and suggest most probable correction by taking into account words adjacently kept in the sentence or even the over whole sentence are having much more bottleneck and difficult tasks for computational linguists fields and asset of program coder or software developers than just checking orthography of Afan

Oromo language. Many times Error of Grammar are those violating, for example, the syntactic laws or the laws related to the structure of a sentence and syntax of a language. Based up on this one of these laws is the agreement between a noun and an adjective in gender and grammatical number in writer and user of the Afan Oromo language. The researcher used three methods are widely used for grammar checking in a language; syntax-based checking, statistics-based checking and rule-based checking [1].

According to this research the spellchecker and corrector system for Tigrigna language by using rule based approach only for mobile phone devices they collect corpus from newspaper, magazine, book and etc. this language is not resourceful so developing such system in develop the language it have its own role or contributions, this low-resourced Tigrigna language in this work they have proposed a systems modeling for Tigrigna language spellchecker, in detecting and correcting word label spelling mistakes. To evaluate the performance of a system a corpus of 430,379 Tigrigna words has been used collected and Stord for reference in dictionary format. By developing a prototype and designing an algorithm in order to indication the validity of the model and perform impressing result in the output and finally this paper proposed dictionary based spell cheek for Tigrigna language [10].

According to this paper the design and development of statistical grammar checker for Amharic by treating its morphological features. The morphologies of individual words that constructing the sentence are taking in to consideration to analyze and then The method they used was n-gram based on the probabilistic method and its morphology of each word in sentence are used to check grammatical errors in the sentence by looking a given Amharic sentence to check and correct if any grammatical issues happened. The system is tested well and with a test corpus and experimental results are reported promising results in correcting Amharic language grammatical error while user of the language write by using system developed [2].

According to this paper an advanced natural language processing technique is used for the Tamil language text to detect wrongly spelled words, and to provide possible correct word suggestions and the probability of frequency of occurrence of each word in the corpus is evaluated and implemented well. The (MED) minimum edit distance algorithm is used for by proposed model they recommend correct suggestions for the misspelled words and also customized for the Tamil vocabulary. Between the misspelled word and all possible permutations of the word distance

matrix is created to increase the performance of the system. To calculate the least possible change needed to fix the misspelled words and for suggesting most of appropriate words from vocabulary of Tamil language they use Dynamic programming for correcting easily the prototype model [25].

In accordance with this study, by using programming language of visual basic is developed and then implemented model for spell checker. For the development of prototype, the researcher use basic algorithms were applied in different lifecycle of the development stage. To select most probable suggested word for misspell word while user typing error spell the study use edit distance algorithm in other algorithm the researcher used was said to be The Metaphone algorithm he applied for spelling error detection and is applied to select most probable correct word for the miss spelled word. This implemented system is embedded in the Microsoft Office word 2010 word processing system as an add-in to check the how the developed system works and to check better usability of the model. The prototype was tested by using test data of 500 words and from these words, 100 of them are miss-spelled deliberately and a dictionary of 125,000 words is used. In the final stage of development life cycle that is testing phase, the model has shown excellent result in its error detection as well as error suggestions [26].

In accordance with this study the rise of the Web 2.0 caused a real democratization in the context of data generation. These data are mostly provided in the form of texts, ranging from the reports provided by news portals, using a formal language, to comments in blog and micro-blogging applications that abuse the use of an informal language. Address this heterogeneity is an essential preprocessing so that these data can be used by tools that aim to infer accurate information based on such data. Thus, this work presents the High Performance Automatic Spell Checker, whose objective is to correct spelling in Portuguese texts collected from the Web. Being a tool that aims to handle a large volume of data, HASCH is completely parallelized in shared memory. In their evaluation, they found that the HASCH was extremely effective in the correction of very large texts from different Web sources, with an almost superliner speedup [46].

2.9 Research gap

In our country, Ethiopia, the numbers of rule based or Dictionary based spelling cheek and correction system developed including Afaan Oromo language On the other hand, exchange of writing text by using electronic device is the very nature of human beings, so peoples need to correct real word or non-word error during exchange of written text between user of the language. ***This gap will raise need to have some system which can understand context***. When user writes their messages the system check the whole document for misspelled word and to suggest if there is any mistake. By using dictionary look up approach for non-word error correction, Bigram Analysis and trigram analysis by understanding the overall context to correct real word errors.

In the above related work reviews we have looking many system developed, method they use, the gap, future work, evaluation and performance technique and etc. then we identified the gap based on this we Generally came up with this Thiess the aim of this study was to develop Context based Afaan Oromo language Spell checker for handheld device, To transfer the knowledge correct spelling has great role in order to facilitate knowledge sharing. In Afan Oromo the structure of the language and the letter arrangement in Afan Oromo is different from the other languages. Context based spell checker for Afan Oromo specifically needed to correct the misspelled words in the hand held device system.

Chapter 3

3. Methodology

This has given us a clear idea on how to model the Afaan Oromo language spellchecker and correction system for mobile phone devices, design appropriate algorithms for Afaan Oromo spellchecker and correction system and instrument needed.

3.1. Nature of Afan Oromo Language

The Oromo people establish the single largest ethnic group in Ethiopia. Oromo People are the largest single ethno-nation in Eastern Africa, constituting at least 40% of the Ethiopian population **[32]**.The Oromo people speak Afan Oromo language, which belongs to the Eastern Cushitic family of Afro-Asiatic phylum. Outside Ethiopia, the language is spoken by thousands of other Oromo tribes in Kenya **[7]**. Besides being the widely used language in Africa, Afan Oromo has been included among the essential languages in the world. Justifying this, the report by the U.S Government and its Education Department conducted in the year 1985 has revealed that Afan Oromo has been considered as one of the 169 critical languages of the world. Currently, from 11 regional states of Ethiopia nations Afan Oromo language is the working language of Oromia regional state which is the largest one state among the current Federal states in Ethiopia. Being the official language, beyond working language of Oromia reginal state it has also been used as a medium of instruction for primary and secondary schools of the region [9].

Number	Capital	Short ,Long	Type
1	A	a, aa	Vowel
2	B	B	Consonant
3	C	C	Consonant
4	D	D	Consonant
5	E	e,ee	Vowel
6	F	F	Consonant
7	G	G	Consonant
8	H	H	Consonant
9	I	i,ii	Vowel
10	J	J	Consonant
11	K	K	Consonant
12	L	L	Consonant
13	M	M	Consonant

14	N	N	Consonant
15	O	o,oo	Vowel
16	P	p	Consonant
17	Q	Q	Consonant
18	R	R	Consonant
19	S	S	Consonant
20	T	T	Consonant
21	U	u,uu	Vowel
22	V	v	Consonant
23	W	W	Consonant
24	X	X	Consonant
25	Y	Y	Consonant
26	Z	Z	Consonant
27	CH	Ch	Double Consonant
28	DH	Dh	Double Consonant
29	NY	Ny	Double Consonant
30	PH	Ph	Double Consonant
31	SH	Sh	Double Consonant

Table 1.1 List of Afan Oromo letters, consonant and vowels [43]

3.1.1 Description of Afan Oromo Alphabets and Sound Systems

Afan Oromo uses Latin character consonants and vowels it use 26 letters of Latin character and 5 double consonants. However, later on a new letters was included in the alphabet as there are words which require the letter.

The writing system of Afan Oromo relies on is Latin Script according to different scholars identified that; the sounds of the language and the alphabets and are modifications of a system that is Latin writing. Thus, the language of Afan Oromo shares a lot of features with English writing system by the some modifications, and the writing alphabet of the language is known as ‘Qubee Afaan Oromoo’ which is designed based on the Latin script. Thus, letters in English or Latin Alphabets are also found in Afan Oromo except the ways they are combined in phonetic alphabets and the styles in which they are uttered.

Since Afan Oromo writing system is a modification to Latin writing system, it shares a lot of features of English writing with some modification. Thus, letters in English language is also in Oromo; however, the language structure is completely different. In Afan Oromo the construction of sentences is Subject-Object-Verb. But in English Subject-Verb-Object is the arrangement of

the sentences. For instance, “Inni Ameerkaa irraa dhufee” which means “He come from America”, Inni, represent subject, Amerika, represent object and dhufee, represent verb.

3.1.2 Vowels -Dubbachiiftuu

Afan oromo languge vowels has 5 in number and these are ‘a’, ‘e’, ‘o’, ‘u’ and ‘i’ . They are similar to that of English, but they are uttered differently. Each vowel is pronounced in a similar way throughout its usage in every Afan Oromo literature. In Afan Oromo words are constructed from the consonant and vowels. Vowels are sound makers and are sound by themselves. Vowels in Afan Oromo are characterized as short and long vowels ;‘aa’,’ee’,’oo’,’uu’ and ‘ii’ to change the meaning of words. For example, while “*seenaa*” means history, “*senaa*” is enter to house. In Afan Oromo Vowel shortest and longest can make spelling error. Additionally, consonants characterized as double and single. Spelling error can occur by the adding or doubling and make single consonant. For example, “*gubbaa*” is represented as over and “*gubaa*” is represented as hot in English context. Hence misspelled word in Afan Oromo formed because of the shorten and longest of vowels, making single of consonant and doubling constant and interchange the position of character, insertion of additional character and removal of character lead us misspell word in the language of Afan Oromo.

3.1.3 Afan Oromo Word Class

Current scholars have stated that Afan Oromo has five word classes those are verb, noun, ad-position, conjunction and adverb. Each of these classes again can be divided into other sub-classes. For instance, noun class is categorized as proper noun, common noun, pronoun, Preposition and postpositions are sub classes of ad- positions. The subclasses in turn can be divided into subclasses and the subdivision process may continue iteratively depending on the level and aim of the investigation.

3.1.3.1 Verb -Xumura

In Afan Oromo verbs are words that are used to indicate some action or event occurrence within time boundaries [9]. It can be transitive, intransitive, modals and auxiliary verbs. Transitive verbs are those verbs which transfer message to complement or object whereas, intransitive verbs do not transfer message to complement and hence, do not have complement or object. The

following examples illustrate this fact. Tolaan Konkolataa bite. This means ‘Tola bought a Car. Since the action of buying was transferred to object Konkolataa ‘Car, bite is transitive verb.

An Oromo verb consists minimally of a stem, representing the lexical meaning of the verb, and a suffix, representing tense or aspect and subject agreement. For example, in dhufne 'we came', dhuf- is the stem ('come') and -ne indicates that the tense is past and that the subject of the verb is first person plural.

As in many other Afro-asiatic languages, Oromo makes a basic two-way distinction in its verb system between the two tensed forms, past (or "perfect") and present (or "imperfect" or "non-past"). Each of these has its own set of tense/agreement suffixes. There is a third conjugation based on the present which has three functions: it is used in place of the present in subordinate clauses, for the jussive ('let me/us/him, etc. V', together with the particle haa), and for the negative of the present (together with the particle hin). For example, deemne 'we went', deemna 'we go', akka deemnu 'that we go', haa deemnu 'let's go', hin deemnu 'we don't go'. There is also a separate imperative form: deemi 'go (sg.)!’[32]

3.1.3.2 Adverbs -Ibsa Xumura

Adverbs are any words that explain or modify verbs [9]. These can be adverbs of frequency, place, time, manner and etc. Adverbs precede verbs they modify in Afan Oromo. For example Abdiin dafe dhufe. This means ‘Abdi came quickly’. Dafe means quickly is an adverb. caaltuun bor deemti. This means ‘Caltu will go tomorrow’. Bor means tomorrow in English is an adverb. Kananiisaan yeroo hunda ni mo’ata. This means ‘kananisaan wins every time. Yeroo hunda means every time in english is an adverb.

3.1.3.3 Noun -Maqaa

Nouns are any word that can be used to name or identify place, ideas or object has plural number and singular, but nouns that refer to multiple entities are not obligatorily plural. Nama means man, namoota means people, nama shan means five men, namoota shan means five people. Another way of looking at this is to treat the singular form as unspecified for number. When it is important to make the plurality of a referent clear, the plural form of a noun is used. Noun plurals are formed through the addition of suffixes [32].

English language uses two types of articles known as definite article (the) and indefinite article (a, an, some, any). In case of Afan Oromo, there are no articles that will be inserted before nouns. However, the suffix – (t) icha can be used in the same context as English language 'the' in masculine and – (t) ittii for feminine nouns. Ending vowels of nouns are dropped before these suffixes are added to the noun. For example: mucaa meaning 'boy', *mucicha* 'the boy',*mana* 'house' ,*manicha* 'the house' ,*durba* 'girl' *durbitti* 'the girl'.

Afaan Oromo Nouns are any word that can be used to name or identify place, object or ideas. Two types of grammatical genders exist in Afan Oromo nouns. These are masculine and feminine, and the entire nouns of the language belong to one of these gender categories. Similarly, there are two numbers (singular and plural) which can be identified by the morpheme it adds. Plural form of a given noun can be formed by adding suffix to the root noun. Various types of suffixes can be added to transform a singular noun to its plural form. All of these suffixes change singular noun to plural without variation in meaning. The last vowel of the singular noun is dropped before the suffix is added. The suffixes which include -oota;- [w]wan,-een, -eelee, -iin,-[a]an,-oolee, eewwan,-iilee etc are used to form plural.

3.1.3.4 Pronoun –Maqdhaala

Pronouns are words that can be used in place of nouns. Similar to that of nouns, pronouns have number and gender. For example, *ishee*/*isii* which means 'she' is feminine (singular) whereas 'isa' which means 'he' is masculine (singular) and '*isaan'* which means 'they' is plural can be masculine or feminine. Pronouns can also be categorized based on their functions and meanings in the sentence. These are personal pronoun, possessive pronoun, demonstrative pronoun, relative pronoun or reciprocal pronoun.

3.1.4 Afan Oromo Punctuation Marks

In Afan Oromo language Punctuation is placed in text to make reading easier and meaning clear. Analysis of Afan Oromo texts reveals that different punctuation marks follow the same punctuation pattern used in English and other languages that follow Latin Writing System. Like English, the following are some of the most usually used punctuation marks in Afan Oromo language [9]. For example, '*qooduu'* comma (,) is used to separate listing of ideas, concepts, names, items, etc and the full '*tuqaa'* stop (.) in statement indicate end of asentense like that of

english langeg, the '*mallattoo gaaffii'* question mark (?) in interrogative and the '*mallattoo raajeffannoo'* exclamation mark (!) in command and exclamatory sentences mark the end of a sentence finaly no uniqe punctuation and usage in this language.

3.1.5 Afan Oromo Morphology

Like in a number of other African and Ethiopian languages, Afan Oromo has a very complex and rich morphology [9]. It has the basic features of agglutinative languages involving very extensive inflectional and derivational morphological processes. In agglutinative languages like Afan Oromo, most of the grammatical information is conveyed through affixes, such as, prefixes, infix and suffixes attached to the root or stem of words. Although Afan Oromo words have some prefixes and infixes, suffixes are the predominant morphological features in the language.

Almost all Afan Oromo nouns in a given text have person, number, gender and possession markers, which are concatenated and affixed to a stem or singular noun form. In addition, Afan Oromo noun plural markers or forms can have several alternatives. For instance, in comparison to the English noun plural marker, *s (-es)*, there are more than ten major and very common plural markers in Afan Oromo including: *-oota, -oolii, -wwan, -lee, an, een, -eeyyii, -oo,* etc.). As an example, the Afan Oromo singular noun *mana* (house) can take the following different plural forms: *manoota (mana+oota), manneen (mana + een), manawwan (mana + wwan)*. The construction and usages of such alternative affixes and attachments are governed by the morphological and syntactic rules of the language [32].

Afan Oromo nouns have also a number of different cases and gender suffixes depending on the grammatical level and classification system used to analyze them. Frequent gender markers in Afan Oromo include *-eessa/-eettii, -a/-ttii* or *–aa/tuu.*

Consider the following example.

Afan Oromo	**Construction**	**Gender**	**English**
Sangoota	*Sangaa +oota*	Male	Ox
Jaarsolii	*Jaarsa +olii*	Male	Elder
Obboleessa	*Obol +essa*	Male	Brother
Beektuu	*Beek +tuu*	Female	Knowledgeable

Table 1.2 Gender and suffix construction in Afan Oromo letters.

3.1.6 Numerals-Lakkoofsa

Numerals include words that refer to number or quantity of something [9]. It can be cardinals such as tokko(one), lama(two) or it can be ordinals like tokkoffaa(first) lammaffaa(second). As discussed in [16], numerals in a sentence follow the category they describe their quantity or amount. Ordinals in Afan Oromo are formed by adding suffix –ffaa to the cardinal numerals. Consider the following sentences:

Abdiin hoolaa sadii bite: This means 'Abdi bought two sheep' .The word lama is cardinal numeral in the sentence. It describes the quantity of hoolaa "oxen".

Abduun daree isaatti tokkoffaa baaheee: This is to mean 'Abduu stood first from her classes'. In this case, the word tokkoffaa is ordinal which is to mean 'first'. It is formed from cardinal tokkoo 'one' by adding affixes –ffaa.

Afan Oromo spellchecker uses the collected Afan Oromo corpus to detect and correct the misspelled words. However in Afan Oromo the absence of standard collected corpus is big challenge to design context based spellchecker for Afan Oromo. Furthermore, the other difficulty for the Afan Oromo language is the different letters repeated that describe a single word. In this case the similar word can be written more than once with different letters like 'eessa' and 'eecha'. In data preparation, we tried to cover and include repeated letters and morphological complex.

3.2 Research Design

This study follows experimental research. According to [9], this type of research method is an approach that uses empirical evidence. It is a way of gaining knowledge by means of direct and indirect participation in the experience. As a result, in this study the researcher was used experimental method for model building and prototype development and evaluates the performance of a system. Generally for prototype system development the following procedures were applied in the study. These include literature review, corpus collection and preparation, system design and development, and finally evaluation of the system performance.

3.2.1 Literature Review

Many more Related works was reviewed to get a deeper understanding about Afan Oromo language spelling structure, preprocesses of corpus and user words as well as, spelling check development and fundamental concepts of related to this work. A review on different approaches of spell checking systems also made to identify the best approach.

3.2.2 Data Processing

To achieve a task of spellchecker and correction model, we collect Afaan Oromo word corpuses with 100,000 Afaan Oromo word, having a good collection of words and large size dictionary in the corpus database helps to design and develop better spellchecker and correction model. Afaan Oromo word corpuses prepared from different sources Construction of the text corpus is very helpful for the detection and correction of misspelled words in the spell checking and suggestion systems [3]. In this work, Afan Oromo corpus of text is created manually to apply in Afan Oromo Spell Checker and corrector system. Because, in Afan Oromo there is no standard corpus developed, therefore we collected free text corpus from Oromia Broadcasting Network (OBN), Oromia Culture and Tourism Office and Fana Broadcasting Corporation(FBC) afaan Oromo service different website like Voice of America (VOA) Afan Oromo section. To beastly meet our target a collected and prepared corpus contains different variety of contents of disciplines, from this some of them are listed as follows, social, economic, healthy , cultural, political and sports and in order to avoid data scarcity and to prepare rich dictionary as well as test the model. Dictionary prepared from the collected corpus was preprocessed by applying tokenization and normalization of the collected data. Stored words are saved in the forms of text and help us for cross check the user inserted words.

3.2.3 Architecture of the system

In the processing of spellchecker and corrector modelling is divides into three different stages

1. Detecting of errors to be corrected
2. Ranking a lists of suggestion for detecting misspelled word
3. List out ranking suggestions and User Select form the ranked list.

3.2.4 Development of System Procedure

The model foot paths unsupervised statistical approach. Since unsupervised approach allow manual preparation of tagged and annotated text which is ideal for under resourced languages like Afan Oromo. The Current study use collected corpus prepared to identify and correct both real-word and non-word error types of spell correction system. N-gram statistical methods help to detect and correct the spelling errors contextually depending on the neighboring words. In n-gram model to correct context based spell cheek and correction system of a word in a sentence is approximated by its probability of occurrence within a neighbor words. Based on that this study use Sequences of two words which means that bigrams are used and their probabilities are estimated from a large corpus collected, normalized and clean text prepared. These probabilities are combined to estimate the a priori probability of alternative acoustic interpretations of the utterance in order to select the most probable interpretation for types of real-word error generated.

Additionally, the model uses dictionary lookup method to detect non-word error form the given text and flagged error full word by highlighting and display candidate suggestion by computing distance measurements between misspelled and dictionary words. As well as, the model uses ngrams analysis method to detect and correct the real-word error from the given text and flagged error by highlighting.

To develop the context based spell checker for Afan Oromo python programing language environment was used to implementing the algorithm and prototype of context based spell checker for handheld device and for user to input text interface is designed.

3.2.5 Implementation tool

To develop spelling check applications for Afan Oromo language, Python 3.9 programming language were used for window and Ubuntu, Fedora and other Linux environment. Python is dynamic programming language that is used in a wide variety of application domains. For mobile applications Kivy is a cross-platform Python framework created to assist in rapid app development. It supports various user interfaces, including multi-touch screens and various platforms, including iOS, Android, and Windows.

Kivy has its own custom UI toolkit, which will look consistent and work exactly the same between Android, iOS, Linux, and Raspberry Pi, but it won't use any native features of any of those platforms. This can be a downside or an upside, depending on what kind of app you're planning to develop. On the one hand, users tend to favor native look in most apps, but UI design that stands out can be a powerful design choice that lets users work in your app on various platforms seamlessly.

3.2.6 Development Technical Approaches and Methodologies

The Context aware spell checking application for handheld device shall implement software development life cycle approach in considerations of building mobile applications, including:

Process – The process of software development is called the Software Development Lifecycle (SDLC). We'll examine all phases of the SDLC with respect to mobile application development, including: Inception, Design, Development, Stabilization, Deployment, and Maintenance.

Considerations – There are a number of considerations when building mobile applications, especially in contrast to traditional web or desktop applications. We'll examine these considerations and how they affect mobile development [43].

3.2.6.1 Mobile development software lifecycle

The lifecycle of mobile development is largely no different than the SDLC for web or desktop applications. As with those, there are usually 5 major portions of the process:

1) **Inception** – All apps start with an idea. That idea is usually refined into a solid basis for an application.
2) **Design** – The design phase consists of defining the app's User Experience (UX) such as what the general layout is, how it works, etc., as well as turning that UX into a proper User Interface (UI) design, usually with the help of a graphic designer.
3) **Development** – Usually the most resource intensive phase, this is the actual building of the application.
4) **Stabilization** – When development is far enough along, QA usually begins to test the application and bugs are fixed. Often times an application will go into a limited beta phase

in which a wider user audience is given a chance to use it and provide feedback and inform changes.

5) **Deployment**

3.2.6.2 Development considerations while Developing Mobile Applications

While developing mobile applications isn't fundamentally different that traditional web/desktop development in terms of process or architecture, there are some considerations to be aware of [43].

3.2.6.3 Common considerations

3.2.6.3.1 Multitasking

There are two significant challenges to multitasking (having multiple applications running at once) on a mobile device. First, given the limited screen real estate, it is difficult to display multiple applications simultaneously. Therefore, on mobile devices only one app can be in the foreground at one time. Second, having multiple applications open and performing tasks can quickly use up battery power. Each platform handles multitasking differently, which we'll explore in a bit.

3.2.6.3.2 Form factor

Mobile devices generally fall into two categories, phones and tablets, with a few crossover devices in between. Developing for these form factors is generally very similar; however, designing applications for them can be very different. Phones have very limited screen space, and tablets, while bigger, are still mobile devices with less screen space than even most laptops. Because of this, mobile platform UI controls have been designed specifically to be effective on smaller form factors.

3.2.6.3.3 Device and operating system fragmentation

It's important to take into account different devices throughout the entire software development lifecycle:

1. Conceptualization and Planning – Keep in mind that hardware and features will vary from device to device, an application that relies on certain features may not work properly on certain devices. For example, not all devices have cameras, so if you're building a video messaging application, some devices may be able to play videos, but not take them.
2. Design – When designing an application's User Experience (UX), pay attention to the different screen ratios and sizes across devices. Additionally, when designing an application's User Interface (UI), different screen resolutions should be considered.
3. Development – When using a feature from code, the presence of that feature should always be tested first. For example, before using a device feature, such as a camera, always query the OS for the presence of that feature first. Then, when initializing the feature/device, make sure to request currently supported from the OS about that device and then use those configuration settings.
4. Testing – It's incredibly important to test the application early and often on actual devices. Even devices with the same hardware specs can vary widely in their behavior.

3.2.6.3.4 Limited resources

Mobile devices get more and more powerful all the time, but they are still mobile devices that have limited capabilities in comparison to desktop or notebook computers. For instance, desktop developers generally don't worry about memory capacities; they're used to having both physical and virtual memory in copious quantities, whereas on mobile devices you can quickly consume all available memory just by loading a handful of high-quality pictures.

Additionally, processor-intensive applications such as games or text recognition can really tax the mobile CPU and adversely affect device performance.

Because of considerations like these, it's important to code smartly and to deploy early and often to actual devices to validate responsiveness.

It introduced general considerations for building mobile applications and examined a number of platform-specific considerations including design, testing, and deployment. [43]

3.2.6.4 A methodology used Agile Software Development Life Cycle.

Agile SDLC methodology is based on collaborative decision making between requirements and solutions teams, and a cyclical, iterative progression of producing working software. Work is done in regularly iterated cycles, known as sprints, that usually last two to four weeks. In Agile, you often don't design for needs that could come up in the future, even if they seem obvious. This is a point where development teams and security teams tend to struggle. Security teams aim to anticipate attacks, attackers, and risks. As needs emerge and are refined over time, security requirements can emerge that weren't anticipated at the beginning of the process. This is normal and natural in Agile, but it can be disorienting to security people who aren't able to secure against various likely attacks. A key takeaway from a security perspective is that Agile is all about the sprint. If a security requirement isn't in the backlog, it won't be scheduled for delivery in a sprint. If it isn't scheduled in a sprint, it won't get done. When security needs are articulated in the backlog, they're prioritized alongside everything else [44].

3.2.6.4.1 The Agile SDLC train

Agile SDLC works a lot like a train. Each rotation of the train wheels represents a sprint. During each sprint rotation, new needs are coming in from the backlog, rolling through the planning, implementation, testing, evaluation, and deployment phases of the Agile software development life cycle (SDLC).

Each Agile phase within each sprint rotation meets the software security tracks through a series of security activities tailored to each phase. There's no need to stop the train to think about security. If vulnerability is identified, treat it like any other bug and resolve it along the way [44].

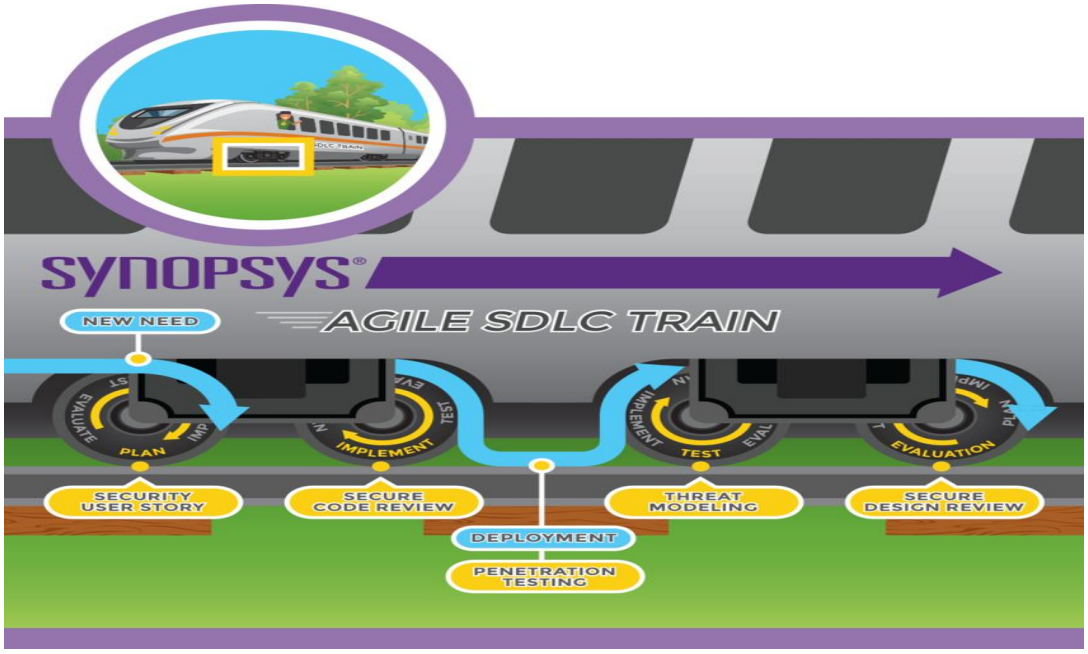

Figure 1.2: The Agile SDLC train Source [44].

3.2.6.5 Scope out and prioritize projects

During the first step of the software development life cycle, the application imitation scopes out and prioritize the model to be done and run first. Then extend to work on other model at the same time depending on the requirement gathered. For each concept, you should define the business opportunity and determine the time and work it will take to complete the project. Based on this information, you can assess technical and economic feasibility and decide which projects or models are worth pursuing.

3.2.6.6 Diagram Requirements for the Initial sprint

Once you have identified the project or models, work to determine requirements. You might want to use user flow diagrams or high-level UML diagrams to demonstrate how the new feature should function along with existing system. From there, work on the project and allocate resources. Create a timeline or a swim lane process map in Lucid chart to delineate responsibilities and clearly show when certain work needs to be completed for the duration of the sprint.

3.2.6.7 Construction/iteration

Once a developer defined requirements for the initial sprint based on requirements, the work begins. UX designers and developers begin work on their first iteration of the project, with the goal of having a working product to launch at the end of the sprint. Remember, the applications will undergo various rounds of revisions, so this first iteration might only include the bare minimum functionality and will have additional sprints to expand upon the overall context aware application.

3.2.6.8 Release the iteration into production

You are nearly ready to release your product into the world. Finish this software iteration with the following steps:

- Test the system. Your quality assurance (QA) should test functionality, detect bugs, and record wins and losses.
- Address any defects.
- Finalize system and user documentation. Lucid chart can help you visualize your code through UML diagrams or demonstrate user flows so everyone understands how the system functions and how they can build upon it further.
- Release the iteration into production.

3.2.6.9 Production and ongoing support for the software release

This phase involves ongoing support for the software release. In other words, you should keep the system running smoothly and show users how to use it. The production phase ends when support has ended or when the release is planned for retirement.

3.2.7 Evaluation Procedure

The designed prototype is evaluated in order to measure the accuracy of the model. Evaluation metric was one measurement to test the accuracy of context spell checker. Recall and precision evaluation metrics help to measure that actual performance of spell checkers for both non-word and real word spelling errors Afan Oromo Spell checker also used Recall and Precision evaluation metrics to measure the performance of the model [33].

3.2.8 The main task of spelling checking process

- Data collection
- Data extraction
- Important contributions to data cleaning and preprocessing
- Implementing backend spellchecking code and suggesting correction words
- Integrating backend code with GUI frontend written
- Adding text highlighting and right-click functionality,

Chapter four

4. Model of Context based Afaan Oromo language Spell checker for handheld device

4.1 Introduction

In the previous chapters we describe some of the related works on spell error detecting and correcting mechanisms for different languages and basic features of the Afan Oromo language to be taken into consideration before designing the model of the Afan Oromo spell checker and corrector are discussed. This chapter describes method applied in this research, model architecture, techniques used and algorithm selected for context based spell checker for Afan Oromo language for hand held device.

4.2 Spell checking Model

Context based spellchecker for Afan Oromo language for hand held device (CBSCAOLHD) model was developed to find any of word error and correct non-word and real-word error. CBSCAOLHD designing followed three steps process which involves: (a) Error detection for identifying misspelled words from user words, (b) Error correction for finding candidate for correcting the misspelled word and (c) Ranking candidates for selecting the best result with high similarity score and maximum bigram and trigram probability from the candidate suggestions. Generally, the Figure 4.1 describes the architecture flow to design Context based spellcheck for Afan Oromo writing system. As shown in figure 4.1, first the model detects the misspelled words from the preprocessed user words. From user word non-word error detected by using dictionary lookup method from prepared dictionary. Whereas, real word error detected by using bigram sequence analysis from bigram model. On other hand, the second step for the CBSCAOLHD model is error correction module try to provided candidate for misspelt words. The model applied Levenshtein algorithm to correct non-word error by computing distance between user word (misspelt word) and dictionary words (correct words). While, bigram and trigram statistical /probabilities applied to correct real word errors, by computing the statistical information of each individual word.

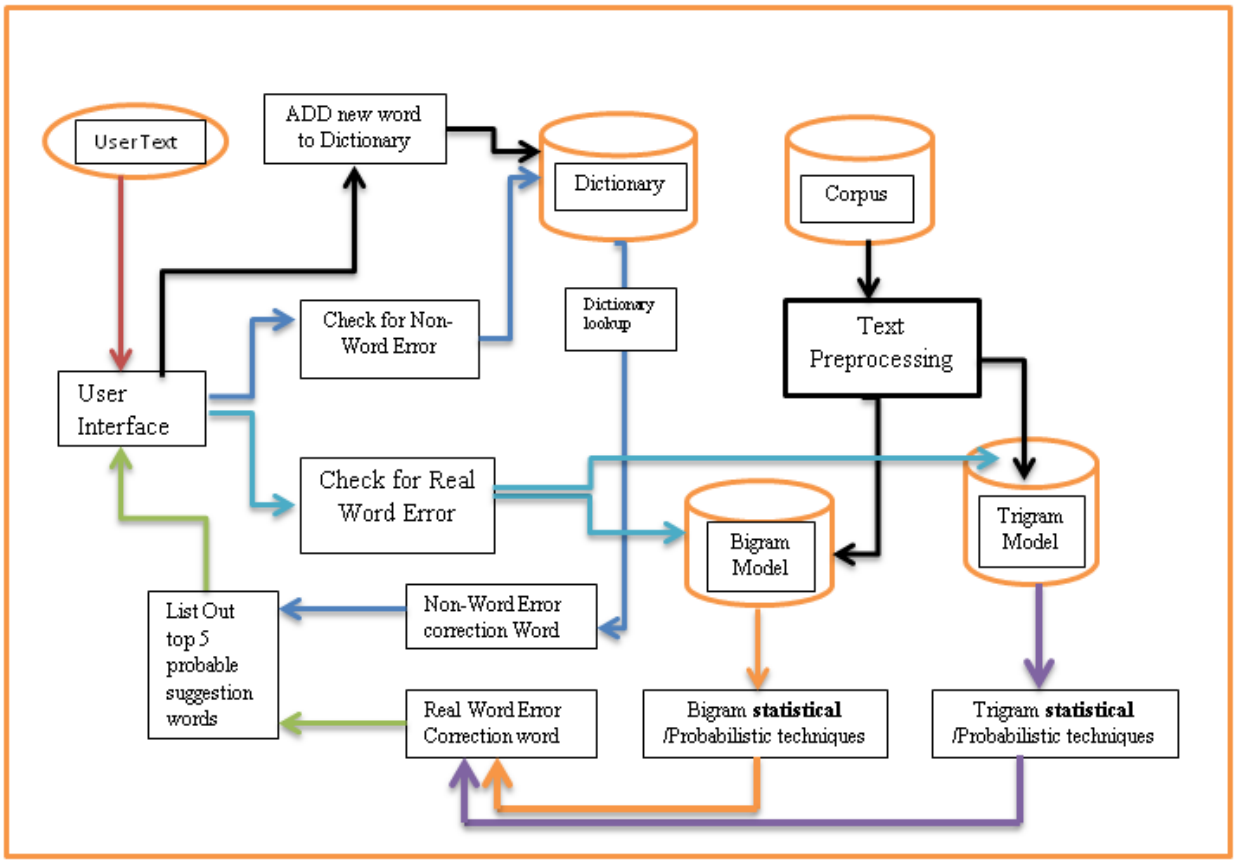

Figure 1.3: Spell Checker Architecture for Afan Oromo Language

4.3 Architecture and how it works

Collected corpus from scientifically error free and trusted sources like students text books organized in text format. This corpse first preprocessed Aside from removing the newline symbols, numbers and special characters, lowercasing all words, and dealing with extraneous spaces, an important text preprocessing step is to remove punctuation. Punctuation is not only present as commas or full stops, but are also used in contractions, such as "what's" or "they'll" or "student's". The problem is that we want to remove punctuation but not the ones present in the contracted words. Initially, the approach was to manually collect all the common contractions and process them to divide each contraction into two words.

Most Python spellcheckers require you to input your text on the command line interface and run it to return misspelled words or suggest corrections to them. However, we built a Python spellchecker that came with an integrated user-friendly GUI, where users can input their text, detect misspelled words and choose from a list of five candidate correction words to correct them. Users can even add words to a pre-built dictionary.

How it works: Cut-and-paste or type your text into the provided textbox in user interface provided to user of the applications. Then click CHECK action buttons from the interface that displayed when the application run and user text inputted in text box.

This will check for non-word errors (misspelled words that are not in the dictionary). These errors are highlighted in red. Select the error word only, excluding the punctuation near them, and right-click on it. Choose one correction from among 5, or add to dictionary. Then click CHECK a second time, which will check for real-word errors (words that are used in bad context). These will also be highlighted in red, and can be corrected by right-clicking on them and choosing a correction word.

4.4 Dictionary model Construction

Corpus was collected and prepared from different sources this Collected corpus was passed through natural language processing techniques text preprocessing that is tokenized, normalized and cleaned and then manually cleared from any kind of unnecessary errors and each word in the corpus were free from spelling errors which is valid to represent Afan Oromo words.
Dictionary is list of corrected words help to detect and correct misspelled words for non-word error. Dictionary size affect the response time, since it check user word against all dictionary words.

4.5 Bigram model Construction

In addition to dictionary the collected corpus from different source preprocessed to construct this word bigram model stored with the sequence of two words and accepted as contextually goes with each other in Afan Oromo language context based spelling checker for handheld system. Well prepared words were stored in the form of the dictionary and accepted as corrected words

and also the corpus used for constructing bigram model to detect and correct real word error from the user prepossessed words.

4.6 Trigram model Construction

The collected corpus from different source preprocessed to construct this word trigram model stored with the sequence of three words and accepted as contextually goes with each other in Afan Oromo language context based spelling checker for handheld system. Well prepared words were stored in the form of the dictionary and accepted as corrected words and also the corpus used for constructing trigram model to detect and correct real word error or contextual aware from the user prepossessed words.

4.7 Text Preprocessing of natural language processing for Afan Oromo language

4.7.1 Tokenization and Normalization

To detect user words and to prepare dictionary as well as compute bigram, it is needed to split the sentence into small parts called tokens this token is single word from sentences. The first task of the system is to break up the sentences into tokens or words and identifies the typographical errors using dictionary next look at bigram or sequence of two word. Understanding word boundary is crucial in the tokenization process. Afan Oromo language has its own word boundary and nearly a word boundary of sentence in Afan Oromo is similar with the English language. Afan Oromo uses the white space to indicate the end of one word and also, exclamation point (!), period (.), question mark (?), brackets (), quotes (‘’) are being used to show a word boundary.

Afan Oromo sentence was split by using tokenize algorithm done by the String Tokenizer of python programing split (‘’) clasess separated by white space. For instance if we have the sentence ‘*sirni gadaa malatoo uumata keenyaatii’* in our corpus, the tokenized into set of words, like ‘*sirni’ , ‘gadaa’, ‘ malatoo’, ‘uumata’, and ‘keenyaatii’*

1. Get a large enough corpus.
2. Define a desired sub word vocabulary size.
3. Optimize the probability of word occurrence by giving a word sequence.
4. Compute the probability of each sub word from corpus.
5. Sort or rank according to probability. To avoid out of vocabulary instances, character level is recommended to be included as a subset of sub words.
6. Repeat step 3–5 until reaching the sub word vocabulary size (defined in step 2) or there are no changes (step 5)

Figure 1.4: Tokenization Algorism

4.7.2 Normalization Algorithm to Clean Text

Normalizer component mainly deals with parsing words of a document into constituent words, usually by considering white spaces and punctuation marks this punctuation mark usage in Afan Oromo is similar to that of English. Beside, text normalized help to the capital letter-small letter problems from the given words. To identify the misspelled words automatically the words are converted to small letters. For instance, if the users enter "baGa nAgaaNa geeSsaN", the models automatically normalize to the "*baga nagaan geessan*". And also it will normalize non-standard words like number, "*guyyaa*11", into "guyyaa". This aspect was handled using phyton number removal and lowering all spelling by lower python class.

Text preprocessing is an important part of Natural Language Processing (NLP), and *normalization* of text is one step of preprocessing.

The goal of normalizing text is to *group related tokens together*, where tokens are usually the words in the text.

Depending on the text you are working with and the type of analysis you are doing, you might not need all of the normalization techniques in this post.

4.7.2.1 Normalization Techniques

In this study we will go over some of the common ways to normalize text.

- Tokenization
- Removing stop words
- Handling whitespace
- Converting text to lowercase
- Expanding contractions (don't -> do not)
- Handling unicode characters - accented letters and some punctuation
- Number words -> numeric
- Stemming and/or Lemmatization

Let Wao is Input afaan Oromo words from corpus. Wao is processed as follows.

1) Wao is loaded. Let's load the text data from collected and prepared Corpus so that we can work with it
2) Split by Whitespace: This means converting the raw text into a list of words and saving it again
3) Remove Punctuation as well as numbers: then use string translation to replace all punctuation with nothing (e.g. remove it) and number also removed.
4) Normalizing Case: It is common to convert all words to one case. Converting text to lowercase

Repeat all stapes to get clean and normalized text by looping the overall corpus
End

Figure 1.5: Normalization Algorism

4.8 Error Detection Algorithm

Error detection process usually consists of checking if an input string is a valid dictionary word or not valid for non-word and check if bigram words available or not based on context. For those not valid words or not available, the model accept as the incorrect words and flagged as misspelled words for the user. Dictionary lookup method was used to identify non-words error whereas, bigram analysis identify real words error from the user words.

4.8.1 Dictionary Lookup Algorithm

Non-word error types corrected by Dictionary look up method and used to identify the word that do not found in lexicon entries from the user word. Preprocessed user words detected by using dictionary lookup method in order to identify non-word error. The dictionary lookup compare each token in a user words against to dictionary words which contains correct spelled words. The tokens that match elements of the dictionary are considered as correct words, otherwise the tokens that didn't match the list of words in the dictionary are flagged as misspelled words.

To implement dictionary lookup *correct* class was used in this study. This suggests functions are preferable in order to minimize response time of dictionary lookup method. As a result for this study dictionary lookup method that implemented by python suggest function was selected.

4.8.2 Bigram Analysis Algorithm

Real-word errors types detection are aim at identifying the errors that occur when a user mistakenly types a correctly spelled word after another was intended to obtain the real word errors types detected in the sentence by considering the preceding and following of words. For instance, "*He come form America"* instead of "*He come from America".* 'Form' flagged as error based on context but form found in the English dictionary words as well correctly written, when bigram analysis calculated contextually form is misspelled word and need the user select suggested word based on frequency probability.

To implement the spelling check and correction system the study use Bigram statistics frequency count which help to detect real word errors by counting how many times the occurrence of the word bigram in other words frequency of occurrence of bigram words with each other to decide misspelling words from the given bigram sequence of words. If the words didn't occur the model flagged as real-word error.

(1 2 3) = (W1W2) (W2W3) ---------- (7)

For example "*gageessaan ummata isaatiif arsaa of godhe".* frequency(*gageessaan ummata*) = 1, frequency(*ummata isaatiif*) = 1, frequency(*isaatiif arsaa*) = 1, frequency(*arsaa of*) = 1, frequency(*of godhe*) = 1 from the given sentence. If the frequency of the bigram is zero, the given word is real word error.

4.9 Error Correction Algorithm

Error correction is the procedure of correcting an error once it has been detected. Once a string has been detected as an error, an error correction algorithm aims at finding candidate corrections for the erroneous word. Generally, the correction module used for this study was Levenshtein edit distance to correct non-word and Bigram probability to correct real word.

4.9.1 Levenshtein Edit Distance

Levenshtein edit distance used to correct non-word error. For this study, Levenshtein algorithm was implemented to find the minimum operation which includes insertion, deletion and substitution to find the candidate correction for the misspelled one. Insertion occurs when a letter needs to be inserted a misspelled word resulting in a correctly spelled word. But deletion occurs when a letter needs to be deleted from a misspelled word in order to result in a correctly spelled word. Substitution indicates to the replacement of a letter in the erroneous word by a correct letter, thus the resulting in the correctly spelled word.

The minimum edit distance between two strings, s1 and s2 is the minimum number of basic operations to convert s1 to s2. Basically, in the model, there are two string; given word /misspelled word and corrected word /from dictionary. Here in the model the algorithm needed to convert given word to dictionary word. For example, s1=”*rottu*”, represents misspelled word from the user and need to be correct. And s2=”*kottu*”, holds the correct word from the dictionary [35].

```
def levenshtein_Edit_distance(self, s1, s2):
                        d = {}
                lenstr1 = len(s1)
                lenstr2 = len(s2)
                for i in range(-1,lenstr1+1):
                        d[(i,-1)] = i+1
                for j in range(-1,lenstr2+1):
                        d[(-1,j)] = j+1
                for i in range(lenstr1):
                        for j in range(lenstr2):
                                if s1[i] == s2[j]:
                                        cost = 0
                                else:
                                        cost = 1
```

```
                    d[(i,j)] = min(
                            d[(i-1,j)] + 1, # deletion
                            d[(i,j-1)] + 1, # insertion
                            d[(i-1,j-1)] + cost, # substitution
                                )
                    if i and j and s1[i]==s2[j-1] and s1[i-1] == s2[j]:
                            d[(i,j)] = min (d[(i,j)], d[i-2,j-2] + cost) # transposition
    return d[lenstr1-1,lenstr2-1]
```

Figure 1.6: Levenshtein Algorithm source code.

4.9.2 Bigram Probability Algorithm

Real word spelling errors types could be corrected by using bigram language model. In the bigram word probability which computed from the collected corpus was applied to correct real word errors.

Here, count (s1) is the frequency of occurrences of s1 in the corpus, and count (s1; s2) is the number of times s2 immediately follows s1 [36]:

(s2 / s1)= (s1, s2) / (s1) ---------- (8)

As example consider the following phrase “Guddinni AFAAN saba tokkoo guddina hawaasa isaa waliin deema jachun bal’inniifi dagaaginni afaan tokkoo guddina qabeenyaafi AADAA saba sanaa waliin walqabata” This is lower cased to " Guddinni afaan saba tokkoo guddina hawaasa isaa waliin deema jachun bal’inniifi dagaaginni afaan tokkoo guddina qabeenyaafi aadaa saba sanaa waliin walqabata " and step 2) gives rise to the bigrams , indicated number shows how many times this bigram happened means frequency of occurrence of bigram.

[(('tokkoo', 'guddina'), 2), (('Guddinni', 'afaan'), 1), (('afaan', 'saba'), 1), (('saba', 'tokkoo'), 1), (('guddina', 'hawaasa'), 1), (('hawaasa', 'isaa'), 1), (('isaa', 'waliin'), 1), (('waliin', 'deema'), 1), (('deema', 'jachun'), 1), (('jachun', 'bal’inniifi'), 1), (('bal’inniifi', 'dagaaginni'), 1), (('dagaaginni', 'afaan'), 1), (('afaan', 'tokkoo'), 1), (('guddina', 'qabeenyaafi'), 1), (('qabeenyaafi', 'aadaa'), 1), (('aadaa', 'saba'), 1), (('saba', 'sanaa'), 1), (('sanaa', 'waliin'), 1), (('waliin', 'walqabata'), 1)] Step 3) then adds the attributes as the below bigram algorithm showes.

Let STR be the words that represent a phrase. STR is processed as follows.

1) STR is lower cased.

2) STR is broken into terms at spaces and these individual terms are used to produce bigrams. Words of length k+2 produce k+1 overlapping bigrams, while any words of length 2 or shorter is taken as the only bigram produced (for simplicity we shall refer to it as a bigram even if it has only one or two words). All such bigrams are attributes of STR.

3) The first bigram produced from each term derived from STR is marked at the right end by the addition of the symbol `!' and the result is included as an attribute with a local count of 1. Also the first word of the term is marked by adding the character `#' to the right and included as an attribute. Finally between any two adjacent terms in the phrase the bigram which consists of the first word separated by a space is added as an attribute.

Figure: 1.7 Bigram Probability Algorithms

4.9.3 Trigram Probability Algorithm

Let STR be the words that represent a phrase. STR is processed as follows.

1) STR is lower cased.

2) STR is broken into terms at spaces and these individual terms are used to produce trigrams. Strings of length k+3 produce k+1 overlapping trigrams, while any string of length 3 or shorter is taken as the only trigram produced (for simplicity we shall refer to it as a trigram even if it has only one or two words). All such trigrams are attributes of STR.

3) The first trigram produced from each term derived from STR is marked at the right end by the addition of the symbol `!' and the result is included as an attribute with a local count of 2. Also the first letter of the term is marked by adding the character `#' to the right and included as an attribute. Finally between any two adjacent terms in the phrase the trigram which consists of the first words separated by a space is added as an attribute.

Figure 1.8 Trigram Probability Algorithms

As an example consider the phrase "MOOTUMMAA fi qondaaltonni haaraan yaa’ii caffee sanaa irratti SEERAA haaraa waggaa saddeettan DHUFAN ittiin biyya bulchan tumu". This is lower

cased to "mootummaa fi qondaaltonni haaraan yaa'ii caffee sanaa irratti seeraa haaraa waggaa saddeettan dhufan ittiin biyya bulchan tumu " and step 2) gives rise to the trigrams [('mootummaaniifi', 'qondaaltonni', 'haaraan'), ('qondaaltonni', 'haaraan', 'yaa'ii'), ('haaraan', 'yaa'ii', 'caffee'), ('yaa'ii', 'caffee', 'sanaa'), ('caffee', 'sanaa', 'irratti'), ('sanaa', 'irratti', 'seeraa'), ('irratti', 'seeraa', 'haaraa'), ('seeraa', 'haaraa', 'waggaa'), ('haaraa', 'waggaa', 'saddeettan'), ('waggaa', 'saddeettan', 'dhufan'), ('saddeettan', 'dhufan', 'ittiin'), ('dhufan', 'ittiin', 'biyya'), ('ittiin', 'biyya', 'bulchan'), ('biyya', 'bulchan', 'tumu'),] Step 3) then adds the attributes.

4.10 Candidate Ranking

Candidates are those tokens with high similarity to the incorrect word. In the case of non-word errors, candidate list gated from the stored dictionary by calculating the similarity between the misspelled and corrected list of words. On other hand, real-word errors the candidate token is the one that is more likely to be intended. Candidate list generated from the bigram model competed from the collected corpus, by calculating the probability of occurrence of preceding words after we write first words [37].

Chapter Five

5. Experimentations and Evaluations

5.1 Introduction

In the previous chapter, an attempt was made to discuss the design of the model of the Context based spell checker for Afan Oromo writing. In this chapter, we implement the tools and algorithms that are describe in previous section to design a model and the experiment was conducted to demonstrate the spelling error detection and correction accuracy. The result of the experiment would be interpreted in this section and the performance of the spelling error detection and correction could be evaluated using evaluation method. **Precision and recall** were used to evaluate the accuracy, effectiveness and validity of detecting and correcting spelling errors based on the training and testing texts that have been used in this experiment.

5.2 Trained Data

The corpus were collected and prepared with linguistic expert depending on spelling features. Similarly, for Afan Oromo we prepare error free corpus to maximize the accuracy and performance of the model by making well understandable and pure words of Afan Oromo vocabulary with linguistic expert on spelling feature. Since collected corpus is contains unnecessary characters and words text preprocessing is very important to clean and smooth noise from corpus.

Moreover, collected corpus used to prepare dictionary that contains the 56,500 list of Afan Oromo words arranged alphabetically as well as, 25,000 Trigram Words 25,000 bigram words computed from the collected corpus. The bigrams are generated at word level rather than character level which are used to detect and correct the real word errors.

The twenty most popular bigrams in this Afan Oromo corpus

Word bigram	Frequency
hariiroo hawaasaa	161
akka hin	116
heera mootummaa	101
otoo hin	100

adda addaa	93
mana murtii	92
hin taane	83
akka ta’e	81
yoo ta’u	80
mirga namoomaa	76
tokko tokko	70
manneen murtii	69
kan hin	68
osoo hin	64
keenya keessatti	64
amala sirreessaa	64
sirna seeraa	62
waan taeef	58
akka qabu	55
sirna haqaa	55

Table 1.3: The twenty most frequently occurring bigram from corpus.

The twenty most popular Unigrams words in this Afan Oromo corpus

Word unigram	**Frequency**
Akka	1851
Kan	1842
Fi	1697
Hin	872
keessatti	667
Yoo	605
Waan	593
Tokko	589
Itti	581
Isaa	542
kana	526
irratti	498
yeroo	472
Ni	410
mana	402
sirna	399
isaanii	383
Kun	374
seeraa	372
seera	372

Table 1.4: The twenty most frequently occurring unigram from corpus.

The twenty most popular trigrams words in this Afan Oromo corpus

Word Trigram	Frequency
otoo hin taane	49
haqaa hariiroo hawaasaa	48
bulchiinsa haqaa hariiroo	30
qofa otoo hin	26
mana amala sirreessaa	25
seera hariiroo hawaasaa	22
hariiroo hawaasaa keenya	22
osoo hin ta'in	22
haa ta’u malee	21
bulchiinsa hariiroo hawaasaa	21
manneen amala sirreessaa	21
akka tae ni	18
hariiroo hawaasaa keessatti	18
biyya keenya keessatti	17
yeroo tokko tokko	17
seerri hariiroo hawaasaa	16
Sirni bulchiinsa haqaa	15
sirna seeraa siviilii	15
haa ta'u malee	14
heera mootummaa irratti	14

Table 1.5: The twenty most frequently occurring trigram from corpus

5.3 Experiment and Result

The experiment was conducted in this work to determine the accuracy of the Context based spell checker for handheld device Afan Oromo language. At experiment and result sections how data has been collected for test purpose and the after experiment has been conducted final outcome found. The interface of context based Afan Oromo language for hand held device contains three different components:

1. Input Area by using a keyboard user to write the desired words using an input method.
2. Button to check whether the inserted words is correct or not
3. Mouse pointer, to provide the candidate for the misspelled words when the user clicks on the error words.

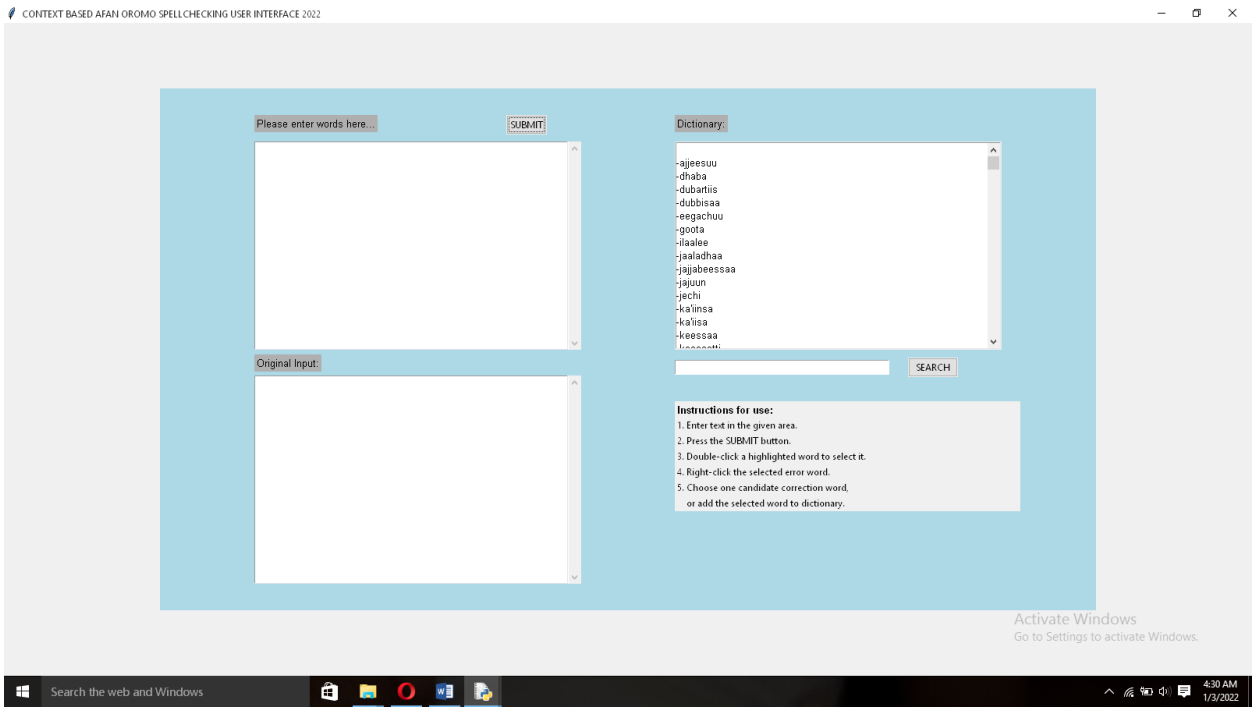

Figure 1.9 User Interface for context based spell check

Context based Afan Oromo language spell check for handheld device has text area which takes input from the users and input text is directly typed in the text area. Since, the model is interactive it wait for button “check” click in order to detect the error. The error detection module responsible to preprocessing and compare the inserted words with dictionary and bigram model. For those didn’t found in the list the model accept as the misspelled words, but for others the model leave as it is. Error detection was executed by using two ways. The first one is dictionary lookup for non-word and the second is of bigram analysis for real words.

5.3.1 Error Detection

5.3.1.1 Non-word spelling errors

The models detect error at word level to identify non-word spelling errors that not found on the list of dictionary by dictionary lookup method. In the user words the word that does not exist in the dictionary is detected as misspelled words change to red color, otherwise the model accept as corrected words.

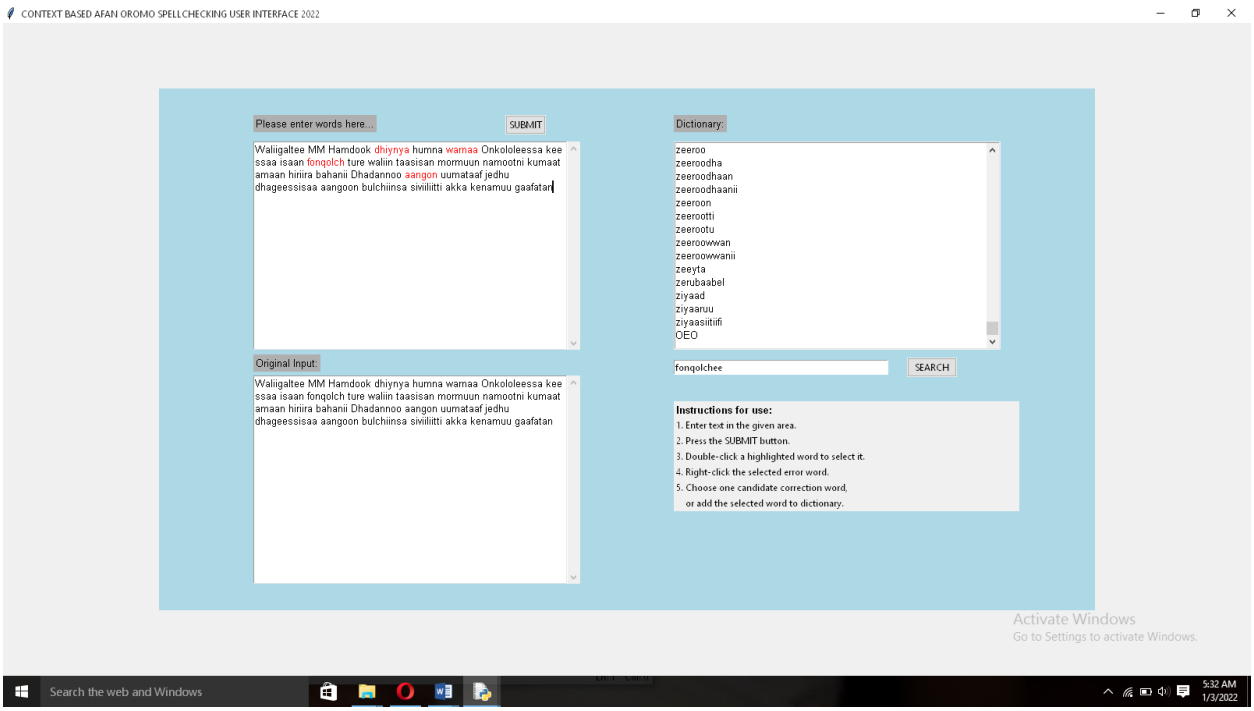

Figure 2.0: Non-word error detection.

5.3.1.2 Real word errors spelling errors

The real word errors were detected under the consideration of bigram words sequences that comes together and sequence of bigram words does not exist in the bigram list, it's detected as real word errors. The input sentences were breakdown into bigram and bigram words were generated along with its probability information which is used to rank the candidate suggestion to correct the errors. If the bigram words found in the bigram model, there is no error which is considered as valid word but if one of the word does not exist in the bigram word list it is considered as misspelled word and the model flagged by highlight with the red color and display misspelled words.

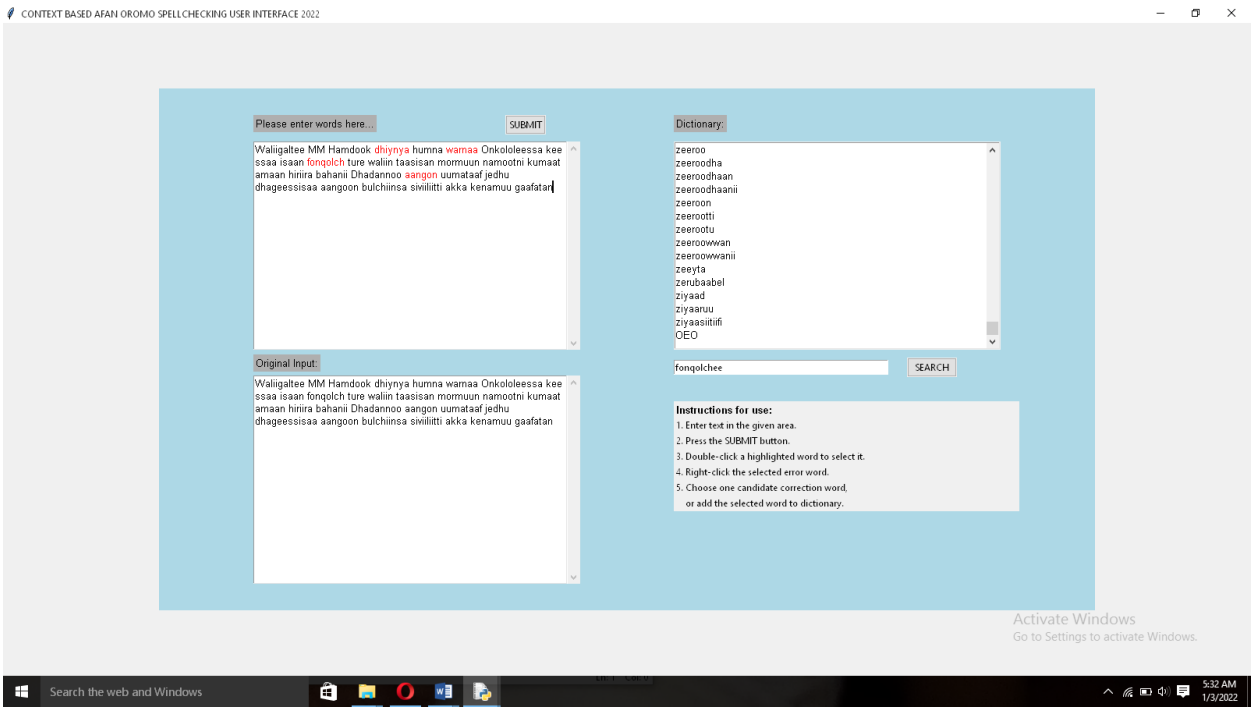

Figure 2.1: Real-word error detection

5.3.2 Error Correction

To correct the misspelled words the corrector algorithm is provide a set of possible candidate corrections. After a word is flagged as wrongly spelled, possible a set of suggestion is provided for the user. For non-word error Levenshtein edit distance responsible to generate the suggestion for misspelled words and take minimum values and rank them accordingly.

5.3.2.1 Non-word error correction

Correction module uses dictionary list to provide a spelling suggestions for each word errors flagged as misspelled in the given inputs of words. The errors were corrected and modified through the suggested words that displayed in the popup menu. In the context based spell check for Afan Oromo language handheld device the model provided at most five candidate list displayed depending on the value they return by computing the distance between them. Additionally, if the user aware of the misspelled words, the model give chance to include the misspelled words as corrected words in to the dictionary.

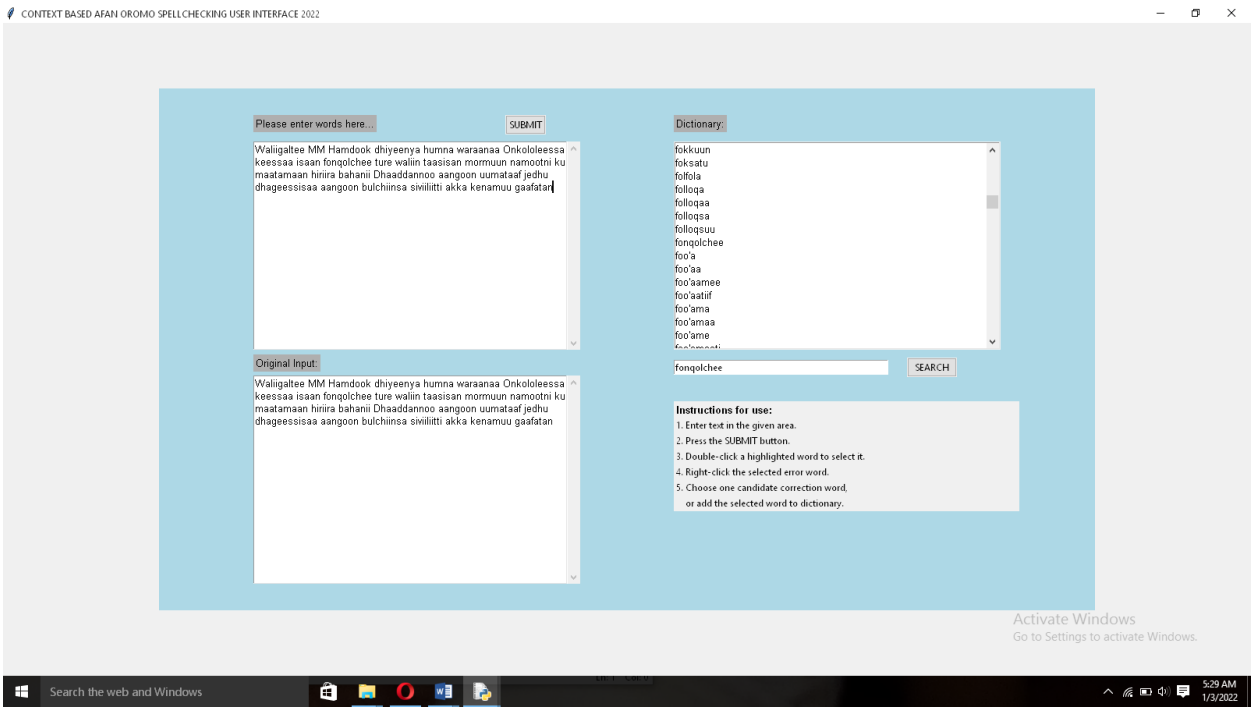

Figure 2.2: Non-word error correction

5.3.2.2 Real word errors correction

Real word errors were corrected using bigram probabilistic information. After misspelled words was flagged the correction module provides candidate list for real-word error from the corpus collected by computing the probability of the occurrence of words after another one. Then replace the invalid words by clicking each error words at any position and search the alternatives from the popup menu.

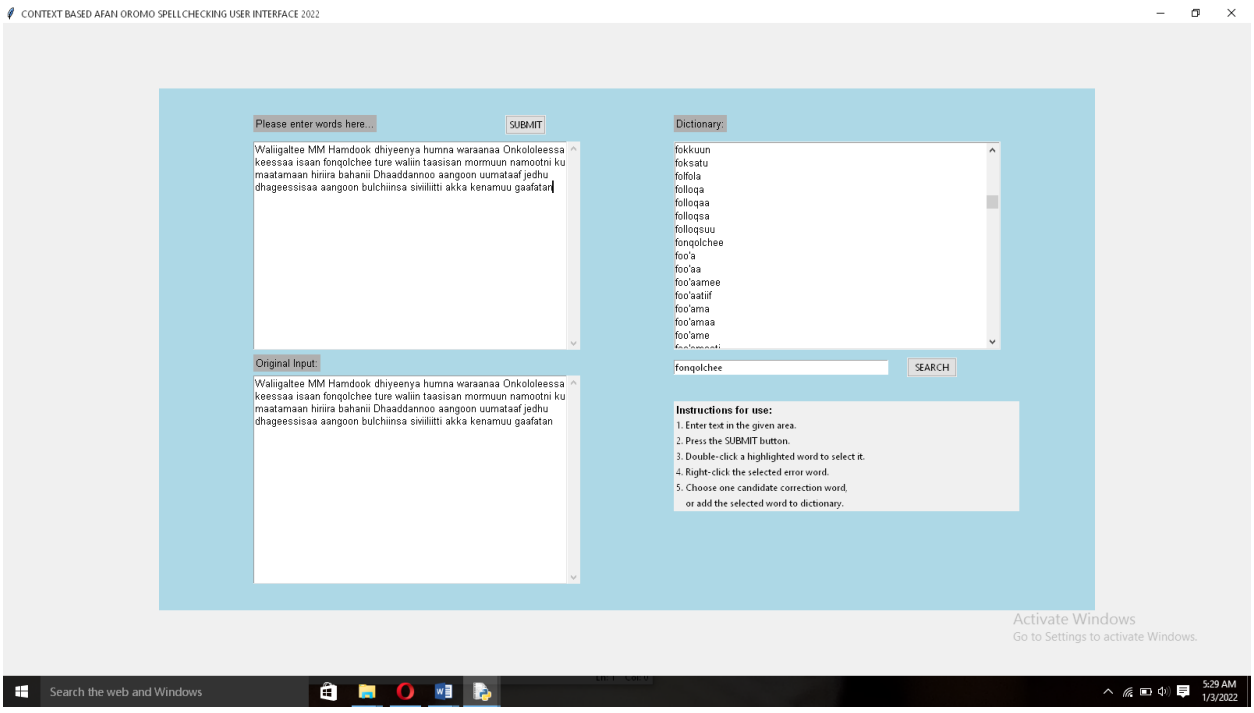

Figure 2.3: Real-word error correction

5.4 Evaluation of Context based spell check for Afan Oromo language

To evaluate the system the most frequent and basic statistical measures which are widely used to measure the effectiveness of spell checking is Precision, Recall and F-measurement. In the evaluation part the performance of the spell checking system of the application, which was designed in this study is evaluated. The final results of the developed system can be divided into two parts: the performance of the dictionary based spell checker and the performance of the context aware spell checker.

The evaluation technique has four categories: true positive (TP), false positive (FP), false negative (FN) and true negative (TN). TP indicate valid words recognized by the spelling checker, resulting correct non-flags. TN invalid words recognized by the spelling checker, resulting correct flags. FN produced when valid words not recognized by the spelling checker, resulting incorrect flags. FP invalid words not recognized by the spelling checker, resulting in incorrect non- flags. The performance of the spell checker have been developed were evaluated using evaluation metrics for both non-word and real word spelling errors in the written texts [39].

Moreover, Error precision(EP) also another metrics to measure the effectiveness of spell checker to test valid words which are not recognized by the spell checker system since the words do not exist in the training set but available in test set. To test the performance of the system researcher prepared testing data to measure the model. We collected text from the different sources and generated artificial spelling errors in the test set and marked these errors to evaluate the efficiency of our system.

We prepared that, the data set consists of 1500 correctly spelled words and 150 misspelled words. In this sample, 1140 were accepted as a valid word; 180 words were flagged as misspelled words by the system due to the nonappearance of words in the dictionary. On other hand, for the real word errors we prepare 1490 words and out of these 1125 are correctly spelled words and 149 misspelled real words. From the sample test data 1125 accepted as corrected words, whereas 199 not detected by system. Generally, the summary of the test result is presented in table below.

Recall for Non word error

Recall = TruePositives / (TruePositives + FalseNegatives)

Recall =1140/(1140+180)=1140/1320==0.863=0.86

Recall for Real Word error

Recall = TruePositives / (TruePositives + FalseNegatives)

Recall =1125/(1125+199)=1125/1324=0.8496=0.85

Precision

Precision for Non-Word errors

Precision = TruePositives / (TruePositives + FalsePositives)

Precision=1140/(1140+30)=1140/1170=0.97

Precision for Real-Word errors

Precision = TruePositives / (TruePositives + FalsePositives)

Precision=1125/(1125+16)=1125/1141=0.98

Accuracy is a metric that generally describes how the model performs across all classes. It is useful when all classes are of equal importance. It is calculated as the ratio between the numbers of correct predictions to the total number of predictions.

$$Accuracy = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}}$$

Accuracy for Non-word error types

Accuracy= (1140+150)/ (1140+150+180+30) =1290/1500=0.86

Accuracy for Real-word error types

Accuracy= (1125+149)/ (1125+149+199+16) =1274/1489=0.85

Error Types	TP	TN	FN	FP	R	P	Accuracy
Non-word	1140	150	180	30	.86	0.97	0.86
Real-word	1125	149	199	16	.85	0.98	0.856
Average	-	-	-	-	0.855	0.975	.85

Table 1.6: Experimental results of Afan Oromo spell checker.

5.5 Discussion

Context based Afan Oromo spell check for handheld device has done a good job of detecting, correcting and suggesting alternatives to the non-word spelling errors and real word errors.

According to the experimental results, the test prediction of the system is 1140 and 1500 as corrected words for both errors and 150 as the misspelling non words and 149 real word errors; accordingly the system achieved 85% performance accuracy for correction of both non-word and real word error. The result indicates that the models effectively and efficiently suggest and correct both types of Afan Oromo spelling check non-word errors and real word error.

Chapter Six

6. Conclusions and Future Works

This thesis has presented the development and experimental results of Context based for Afan Oromo language context based spell cheek and correction system for hand head device. In the previous chapters of this document, the theoretical basis of spell checking and suggestions system have been reviewed and as required part of contracting the developed system, the preparation of the Afan Oromo text corpus is showed. Moreover, the preparations of the dictionary and building of bigram and trigram model from the corpus also shown. The main achievements of this study have been the experimental results of our work, as shown in chapter 4. This chapter presents our conclusive remarks and recommendations for future work.

6.1 Conclusion

The ultimate goal of research on Natural Language Processing is to parse and understand language, which is not fully achieved yet. For this reason, research in NLP has focused on intermediate tasks that make sense of some of the structure inherent in a language without requiring complete understanding. One such task is spelling correction.

Spell checking and correcting have become a part of everyday life for today's generation. Whether it be working with text editing tools, such as MS Word, or typing text messages on one's cellular/mobile phone, spell checking and correcting are an inevitable part of the process. In this study, Context based spell checker for Afan Oromo language for handheld device has been designed, developed and tested on both non-word and real-world errors. Three approaches of spelling checking are used in this research: dictionary lookup approach, bigram and trigram approach. Dictionary lookup approach was implemented to detect and correct non-words error and it requires dictionary in order to cross check each user words. On the other hand, N-gram approach bigram and trigram analysis approach is designed and implemented for real word errors. The N-gram Statistical/probabilistic approach requires a good quality training data set. And also help us to check real word errors of a given words. These approaches are capable to handle both non-word and real word errors from the context based Afan Oromo spell check system.

As the performance of the context based spell checker for Afan Oromo language for handheld devices shows, the system registered 85 % accuracy. It needs to resolve enhance the performance of the context based spell checker.

Afan Oromo Spell checking model, did not recognize compound words as the single words. But, it detects as the different word by split each word. Moreover, abbreviations with capital letters and word with number are did not taken as correct word as the Microsoft Word (MS) does. In MS every capital letters abbreviation and word with number are accepted as corrected words. For instance in MS; 'frm1' is taken as correct word, even if the word did not found in English dictionary. However for context based Afan Oromo spell check model, 'kottu1' is taken as misspelled words, because in this model every numbers are removed before checked.

6.2 Recommendation

The results of this research have resolve the problem of misspelling when the user write Afan Oromo words in hand held mobile phone device. However, this work could mainly benefit from integration of different areas of researches. This section lists a brief list of areas of improvements for this research work.

Based on the finding of this study the following research directions are suggested as future works:

The results of this research have resolved the problem of misspelling when the users write Afan Oromo words. However, this work could mainly benefit from integration of different areas of researches. This section lists a brief list of areas of improvements for this research work. Based on the finding of this study the following research directions are suggested as future works:

- Future work focus on Conceptualization and Planning – Keep in mind that hardware and features will vary from device to device, an application that relies on certain features may not work properly on certain devices by study each device hardware re modification is needed according to requirement.
- Re Design and integrating to many communication channel user may use to exchange message SMS, Telegram ,Social Media ,email exchange and etc. need further work–

When designing an application's User Experience (UX), pay attention to the different screen ratios and sizes across devices. Additionally, when designing an application's User Interface (UI), different screen resolutions should be considered as well as how can I integrate too many communication platform is to be considered for the future.

- Re Development by adding many future based on different operating system have there– When using a feature from code, the presence of that feature should always be tested first because the OS about that device and then use those configuration settings because of the above reason for the future many hand held device have using different types of operating system have their so also need further work.
- For windows on any computer running many operating system this context aware application need for future to develop and available for the user.
- Need further work to correct especially real word error happened with noun, place and gender.
- In enhance the performance of the spelling checker to provide best candidate suggestions for spelling errors especially for real word errors gathering of large amount of corpus. Thus, preparing adequate and better size corpus must be one task in the future and having a standard dictionary with maximum word size is very important to increase the accuracy of spell checker.
- For better Performance could perhaps be using word probability information greater than trigram, quadrigram. To gate correct real word error and high possible suggestion

Reference

[1]T. Debela, “A rule-based Afaan Oromo Grammar Checker,” International Journal of Advanced Computer Science and Applications, Vol. 2, No. 8, CA ,pp. 126–130 , 2011.

[2]T. Aynadis and A. Yaregal, “Development of Amharic Grammar Checker Using Morphological Features of Words and N-Gram Based Probabilistic Methods,” MSc. Thesis, Dept. Comp. Sci. Addis Ababa Univ., Addis Ababa, Ethiopia, 2016.

[3]Y. Nigusu, “Context based Spell Cheeker for Amharic,” MSc. Thesis, Dept. Comp. Sci., Jimma Univ., Jimma, Ethiopia, 2016.

[4]G. Ganfure and D. Midekso, “Design and Implementation of Morphology Based Spell Checker,” vol. 3, no. 12, CA, pp. 118–125, 2014.

[5]T. Workineh , “Investigating Afaan Oromo Language Structure and Developing Effective File Editing Tool as Plug-in into Ms. Word to Support Text Entry and Input Methods,” MSc. Thesis, Dept. info. tec., Medawolabu Univ., Robe, Ethiopia, 2015.

[6]T. Workineh and T. Duresa, “Enhancing the Text Production and Assisting Disable Users in Developing Word Prediction and Completion in Afaan Oromo,” Journal from Inform Tech Software Engineering, 7:2 DOI: 10.4172/2165-7866.1000200, 2017.

[7]Wikipedia. (2021, May, 23).About afan oromo language [online]. Available: https://en.wikipedia.org/wiki/Oromo_language , May 13, 2021.

[8]S. Singh , “Artificial Intelligence Review,” Vol. 53, No. 8, CA ,pp. 110-115 , 2021.

[9]K. Tirate, “Context Based Spelling Checking for Afaan Oromo writing,” MSc. Thesis, Dept. Info. Sci., Jimma Univ., Jimma, Ethiopia, 2018.

[10]B. Atakilti and U. Petros, “Tigrigna language spellchecker and correction system,” Info. Tec., Aksum Univ., Aksum, Ethiopia, Vol 11, No 3, Oct 25, 2020.

[11]R. Hedine, “Spell checker in CET Designer,” Linköping Univ., Data Teknik, 16 ECTS, LIU-IDA/LITH-EX-G--16/069--SE, 2016.

[12]H. Lorraine, “Spell Checkers and Correctors,” MSc. Thesis, Dept. comp. Sci., Univ. of Pretoria, Pretoria, South Africa, November, 2008.

[13]Stady.com (2021, Oct, 04).Spell check definition [Online]. Available: https://study.com/academy/lesson/what-is-spell-check-definition-use-quiz.html

[14]Wikipedia.(2021, Oct ,04). About spellcheck [Onlinel]. Available: https://en.wikipedia.org/wiki/Spell_checker

[15]D. Megersa, "An Automatic Sentence Parser for Oromo Language Using Supervised Learning Technique,” MSc. Thesis, Dept. comp. Sci., AA Univ., Addis Ababa ,Ethiopia, 2002.

[16]D. Tesfaye, "Designing a Stemmer for Afan Oromo Text: A hybrid approach," Master’s thesis, School of graduate studies, Addis Ababa University, Ethiopia, 2010.

[17]T. Gamta, “Seera Afaan Oromoo", Finfinnee, Boolee Press.

[18]C. Patil, R. Rodrigues and R. Ron, "Auto-Spelling Checker using Natural Language Processing," Chinmay Patil Xavier Institute of Engineering, Mumbai, Maharashtra, India ,Volume: 07 No 08, Aug, 2020.

[19]Wikipedia. (2021, Dec, 09). About natural language processing [online]. Available: https://en.wikipedia.org/wiki/Natural_language_processing
[20]R. Andrew, “A Bayesian hybrid method for context sensitive spelling correction,” 1995
[21] M. Ann, B. Santorini, M. Marcus, “Building a Large Annotated Corpus of English: The Penn Treebank,” Univ. of Pennsylvania, October, 1993
[22]F. Ahmed, E. William, and A. Nürnberger, “Revised N–Gram based Automatic Spelling Correction Tool to Improve Retrieval Effectiveness,” Institute for knw and Lang Engi., Univ. of Magdeburg, Otto–von–Guericke ,Germany,2020
[23]P. Etoori , M. Chinnakotla and R. Mamidi, “Automatic Spelling Correction for Resource-Scarce Languages using Deep Learning,” Melbourne, Australia, July 20, 2018
[24]S. Verberne, “Context-sensitive spell checking based on word trigram probabilities,” MSc thesis, Taal, Dept. Spraak & Info. Univ. of Nijmegen, January 2002.
[25]P. Kumar, A. Kannan and N. Goel, “Design and Implementation of NLP-Based Spell Checker for the Tamil Language,” Noida 201309, India, 10, Nov, 2020.
[26]A. Getnet, “Automatic Amharic Spelling Error Detection and Correction using hybrid approach,” Dept. Comp. Sci. Punjabi Univ., Patiala, India , Volume 5, Issue 6, ISSN-2349-5162 June, 2018.
[27]A. Pal and A. Mustafi, “Vartani Spellcheck – Automatic Context-Sensitive Spelling Correction of OCR-generated Hindi Text Using BERT and Levenshtein Distance,” Sci. and Engi. Birla Inst. of Tecn, Dept. Comp., Mesra Ranchi, India, 2017
[28]Z. Abdulameer, “Analysis of spelling performance in English among students whose first language is Arabic,” MSc. Thesis, Texas A&M Univ., USA, 2009.
[29]S. Mariawit, “Amharic Spelling Error Detection and Correction System: Morphology-based Approach,” MSc Thesis, AA. Univ., Dept. Info. Scie. , Addis Ababa, Ethiopia, Sept, 2020.
[30]R. Hedin, “Spell checker in CET Designer,” Bachelor thesis, Linköping Univ., Dept. of Comp. Sci., 16 ECTS, Datateknik, LIU-IDA/LITH-EX-G--16/069—SE, 2016.
[31]H. Lorraine, “Spell Cheekers and correctors a unified treatment,” Master thesis, facu. Of engi. Dept. IT., Univ. of Pretoria, Pretoria, South Africa, November 2008
[32]Wikipedia.(2021,July,2).Oromo language [online]. Available: https://en.wikipedia.org/wiki/Oromo_language
[33]R. Mishra and N. Kaur, “A Survey of Spelling Error Detection and Correction Techniques,” International Journal of Computer Trends and Technology ,vol 4, Issue 3 , Dept. Comp. Sci. and Engi., Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab, India ,2013.
[34]J.-H. Lee, M. Kim, and H.-C. Kwon, “Deep Learning-Based Context-Sensitive Spelling Typing Error Correction,” Dept. of Electrical and Computer Eng., Pusan National University, Pusan 17579, South Korea, Digital Object Identifier 10.1109/ACCESS.2017.Doi Number
[35]Ideserve. (2021, Dec, 25). Minimum edit distance between given two strings [online]. Available: https://www.ideserve.co.in/learn/edit-distance-dynamic-programming

[36]S. sharmaa, S. Guptab , “A correction model for real-word errors” 4th International Conference on Eco-friendly Computing and Communication Systems ,Dept. Comp. Scie, Chandigarh Univ, Gharuan Mohali , Procedia Computer Science, pp 99 – 106 , Dec ,2015
[37]D. Jurafsky & J. Martin, “N-gram Language Models,” Speech and Language Processing Chapter 3 of books, September 21, 2021.
[38]k.Tirate, M.Million, and T.Workineh, “A Context Sensitive Text Writing Correction and Error Detection for Afaan Oromoo Words,” Gadaa Journal/Barruulee Gadaa , Vol. 3 No. 1 ,2020
[39]vitalflux. (2021, Oct, 01). Accuracy, Precision, Recall & F1-Score [online].Available: https://vitalflux.com/accuracy-precision-recall-f1-score-python-example
[40]towardsdatascience. (2018, Aug, 12). Tokenization algorithms [online]. Available: https://towardsdatascience.com/overview-of-nlp-tokenization-algorithms-c41a7d5ec4f9
[41]Lvngd. (2020, Mar, 22). Text Normalization for Natural Language Processing [online]. Available:https://lvngd.com/blog/text-normalization-natural-language-processing-python
[42]G. Abraham, “Improving Brill’s Tagger Lexical and Transformational Rule for Afaan Oromo Language,” MA.. Thesis, Dept. GIS, Univ. Hawassa, SNNPR, Ethiopia, 2020.
[43]Microsoft.(2021 ,Aug ,21).Mobile software development lifecycle[online]. Available: https://docs.microsoft.com/en-us/xamarin/cross-platform/get-started/introduction-to-mobile-sdlc
[44] Synopsys. (2021, Dec,7).Agile Software Development Life Cycle [online]. Available:https://www.synopsys.com/glossary/what-is-agile-sdlc.html.
[45]H. Oskar, “A quantitative research paper on mobile phone application offloading by cloud computing utilization,” MSc. Thesis, Dept. informatics, UMEA, 2021.
[46] G. Andradea, F. Teixeiraa, C. Xaviera,b, R..Oliveiraa, L. Rochaa, and A. Evsukoffb ,“High Performance Automatic Spell Checker for Portuguese Texts from the Web,” International journal on Computational Science ,Vol. 1, No. 2, CA ,pp. 1877-0509, 2012.
[47] S. Jyonica and L. Chaman, “Role of Language In Human Life,” International journal of English Language ,Literature and humanities ,Vol. 3, No. 7, CA ,pp. 2321-7065,2015
[48] techtarget.(2021,March,6) natural language processing[online]. Available: https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP
[49] H. Daniel, S. Jan and P. Matus, “Electronics Review Survey of Automatic Spelling Correction,” Vol. 9, No.2, CA, Pp. 1670, Oct, 2020. www.mdpi.com/journal/electronics

Appendix

I. Sample code

```
# Reading in the corpus and dictionary
                with io.open("AFO_Corpous_test.txt", "r", encoding = "utf-8") as f:
                        self.corpus = f.read()
                with io.open("AFOdictionary_test.txt", "r", encoding = "utf-8") as f:
                        dict_text = f.read()

                # create unigram model
                self.unigrams = self.corpus.split(' ')
                N_u = len(self.unigrams)
                self.counts_u = dict(Counter(self.unigrams))

                model_u = {}
                for (key,value) in zip(self.counts_u.keys(), self.counts_u.values()):
                        model_u[key] = value/N_u
                self.model_u = model_u

                # create dictionary list
                self.dictionary = sorted(set(dict_text.split('\n')))
                self.non_words = [] #empty list of non-words

                # create left bigrams (the usual bigram), right bigrams and trigrams
                self.bigramsl = list(ngrams(self.unigrams, 2))
                self.bigramsr = [(w1,w2) for (w1,w2) in zip(self.unigrams[1:],

        self.unigrams[:-1])]
                self.counts_bl = dict(Counter(self.bigramsl))
                self.counts_br = dict(Counter(self.bigramsr))
                N_b = len(self.bigramsl)

                self.trigrams = list(ngrams(self.unigrams, 3))
                self.counts_t = dict(Counter(self.trigrams))

                self.create_layout()
                print("\nStatus: Ready\n")

def make_bigram_model(self):
        model_bl = {}
        for key, value in zip(self.counts_bl.keys(), self.counts_bl.values()):
        model_bl[key] = value / self.counts_u[key[0]]
        model_br = {}
        for key, value in zip(self.counts_br.keys(), self.counts_br.values()):
        model_br[key] = value / self.counts_u[key[0]]
        return model_bl, model_br

def make_trigram_model(self):

        model_t = {}
        for key, value in zip(self.counts_t.keys(), self.counts_t.values()):
        model_t[key] = value / ((self.counts_bl[key[:2]] + self.counts_br[key[-1:-3:-1]]) / 2)
        return model_t
```

II. Sample Training Data

Onismoos Nasiib naannawa bara 1856 dhiheenya, Godina Iluu Abbaabooraa magaalaa Hurrumuutti dhalate. Maqaan isaa kan dhalootaa Hiikaa Awwaajii jedhama. Abbaan isaa waggaa afuritt iirraa du'e.Weerartoonni/garboomsitoonni saba biraarraa dhufan bara 1869 Hiikaa haadha isaa jalaa hatuun, maqaa haarawa Nasiib jedhu moggaasaniifi akka garbaatti gurguratan. Sanaan boodas yeroo heddu gurgurameera. Walumaagalatti Nasiib yeroo saddeet garbummaaf gurgurame. Dhumarratti, itti aanaa itti gaafatamaa Qoontsilaa Faransaayi kan ture, namni Weerner Munziinger jedhamu, Magaalaa Mitsiwwaa, qarqara Galaana Diimaatti isa argatee akka inni sanaan booda garbummaan hingurguramne bilisa isa baase. Ergamtoonni lallaba Kitaaba Qulqulluu Siwiidiin mana barnootaa ijoolleen dhiirri qofti itti baratan iddoo Imkulluu jedhamu waan qabaniif, Nasiib achi galee akka baratu taasisan. Nasiibis yeroo gabaabaatti barataa cimaa dandeettii addaa qabu akka ta'e mirkaneesse. Dhalatee waggaa 16tti gaafa Hiikaa/faasikaa Bitootessa 31 bara 1872 cuuphamee maqaan kiristaanummaa Onismoos jedhamu mogga'eef. Afaan Girikitiin 'Onesmos' jechuun faayidaa qabeessa jechuudha. Barnoota isaas waggoota shanitti xumure. Itti aansuun, dhaabbata barnootaa amantaa 'Johaaneluundis' jedhamu, kan magaalaa Biromaa, biyya Siwiidinitti argamutti ergamee waggoota shaniif barnoota amantaa ol'aanaa barate. Erga achii Mitsiwwaatti deebi'ee, shamara waggaa kudha sagalii kan Mihirat Hayiluu jedhamtu fuudhe. Onismoos Nasiib uummata isaa barsiisuuf fedhii cimaan wanta isa keessatti uummameef, haadha warraa isaa, abbaa ishiifi namoota biraa sadi waliin karaa Sudaaniin gara Wallaggaa seenuuf adeemsa eegale. Haa ta'u malee, loltoonni mooticha Minilik adeemsa isaaniitti gufuu wanta ta'aniif, Asoosaa darbuu wanta dadhabaniif gara magaalaa daangaa Itoophiyaafi Sudaanirra jirtu, kan Faamkaa jedhamtutti deebi'uuf dirqaman. Onismoosis achitti busaadhaan qabamee daran dhibame. Onismoosiifi miiltowwan isaa gara Kaartuumitti deebi'uuf waan dirqamaniif, Ebla 12, bara 1882 Kaartuum gahan. Onismoos achitti dhibee isaarraa fayyee gara Imkulluutti deebi'ee, hojii isaa wangeela barsiisuu itti fufe. Yeroo sanii kaasee, barreeffama addaddaa gara Afaan Oromootti hiikuu eegale. Erga yaaliin inni bara 1886 gara Wallaggaa deemuuf taasise yeroo lammaffaaf gufatee booda, Kitaaba Qulqulluu guutumaa-guutuutti gara Afaan Oromootti hiikuu eegale.Haa ta'u malee, ijoollummaa isaarraa eegalee uummataafi aadaa isaa keessatti wanta hinguddatiiniif, hanqina jechootaafi jechamoota Afaan Oromoo wanta qabuuf, gargaarsa barbaaduuf dirqame. Akka carraa ta'ee, shamara Asteer Gannoo jedhamtu, kan Iluu Abbaabooraatii garbummaan gara Yemenii fudhatamaa osoo jirtuu loltoota galaanarraa Xaaliyaanitiin bilisa baate, Imkulluutti argate. Ishiinis hojii isaa barreeffamoota gara Afaan Oromootti hiikuurratti gargaarsa olaanaa taasisteef. Gargaarsa ishiin gooteefiin Kakuu Moofaa gara Afaan Oromootti hiikee Waxabajjii bara 1897 xumure. Bara 1904 gara Wallaggaa deemee, uummata isaatti makamuu danda'e. Yeroo san bulchaa Wallaggaa kan ture, 'Dajjaazmaach' Gabra-Igizaabeer, fuula ifaan isa simate. Yeroo duraatiif uummata isaaf Afaan Oromootiin Kitaaba Qulqulluu lallabuu eegale. Haa ta'u malee, qeesonni amantaa Ortodooksii naannawa san turan afaanicha wanta hindhageenyeef, jibbiinsa cimaa irratti horatan. Jaalalaafi kabajni ol'aanaan inni uummata Oromoo biratti horate, wanta isaan rifachiiseef, sababa "Maaramii kabaja dhoorke" jedhuun isa yakkanii, Paatiriyaarkii Orotodooksii kan ture, Abuna Maatiwoos biratti akka dhiyaatu ta'e. Abunichis himannaa qeesotichi dhiheessanirratti hundaa'uun akka biyyaa bahu itti murteesse. Haa ta'u malee, Minilik, murtii Abunichaa haquun, Onismoos gara Naqamteetti akka deebi'u, garuu sanaan booda gonkuma akka wangeela hinlallabne itti murteesse. Sanaan booda sochiin inni ummata bal'aa keessatti taasisu waan daanga'eef, mana barnootaa Naqamteetti bane keessatti barsiisaa ture. Yeroo sanas balaan biyyaa ari'amuu isaa marsa ture. Lij-Iyyaasuun bara 1916 aangoo fudhannaan amantaa isaa akka barsiisu eeyyameef. Lij-Iyyaasuun, waggaa tokkoon booda aangoorraa fonqolchamus, murtiin isaa akkuma jirutti wanta itti fufeef, Onismoos hanga gaafa du'uutti barreeffamoota addaddaa gara Afaan Oromootti hiikee raabsuufi Kitaaba Qulqulluu barsiisuu kan itti fufe, yoo ta'u, barreeffamoota inni Afaan Oromootiin qopheesse keessaa kan armaan gadii eeruun nidanda'ama.

Faayidaa Bosonaafi Bineensotaa

Bosonni wabii jireenyaa, ilma namaa dabalatee, beeyiladootaafi lubbuqabeeyyii hundaati. Yemmuu faayidaa bosonaa dubbannu,waa'ee lubbuun jiraachuufi jiraachuu dhabuuti dubbanna jechuudha. Kallattiinis ta'ee, kallattiin ala lubbuufi jiruun ilma namaa bosonarratti hundaa'a. Faayidaalee gurguddaa bosonaafi miidhaalee bosona mancaasuun fidu adda adda baasnee tokko tokkoon haa ilaallu.Rakkina geeddarumsi qilleensaa addunyaatti fidaa jiru qolachuu ykn hir'isuu keessatti gaheen bosonaa ol'aanaadha. Akkuma beekkamu geeddarumsi qilleensaa rakkoolee jaarraa kana keessatti daran yaaddessoo ta'an keessaa isa tokko. Sababa geeddarumsa qilleensaatiin oo'inni addunyaa yeroodhaa gara yerootti dabalaa jira. Sababa oo'a dabaleetiin cabbiin qarqara lafaa jiru dhangala'aa waan jiruuf, olka'iinsi qaamolee bishaanii dabaluun biyyoota heddu kan lafa gadi-aanaarratti argaman liqimsuu danda'a sodaan jedhu dabalaa jira. Kana malees, rakkoolee akka wayitiin roobaa gegeeddaramuu, roobni xiqqaatee gogiinsi uummamuu, iddoo tokkotti ammoo roobni baay'achuun lolaan cimaan akka 'Sunaamii' faa uummamuu, dhukkuboonni addaddaa uumamuufi kkf, sababa jijjiirama qilleensaatiin kan uumamani. Bosonni, kaarboonii ardii tanaa harka guddaa hammatee jira. Kanaafuu, yeroo ammaa kana bosonni saffisaan manca'aa

jiraachuun isaa gadi lakkifamuu kaarboondaayi'oksaayidiif hanga 20% qooda qaba jedhama. Haalli kun ammoo akkuma jirutti yoo itti fufe, rakkinni hagamii uumamuu akka danda'u tilmaamuun nama hinrakkisu.Yoo bosonni manca'e kaarboondaayi'oksaayidiin qilleensatti gadi lakkifamuun jijjiirraa qilleensaa fida; yoo bosonni kunuunfame garuu, kaarboondaayi'oksaayidiin qilleensa keessa jiru nixuuxxama. Kanaafuu, bosonniifi geeddarumsi qilleensaa walitti dhufeenya gar-lamee qabu. Geeddarumsi qilleensaa bosonaarratti balaa fida; bosona kunuunsuun ammoo dhiibbaa geeddarumsa qilleensaa nisalphisa. Dhimma kanaan walitti qabatee, bosonni akka madda galiittis nifayyada. Biyyoonni guddatan, kan warshaalee guguddaarraa kaarboondaayi'oksaayidii hedduu burqisiisaan, biyyootii warshaalee gurguddaa hinqabneefi bosona heddu qabaniif maallaqa akka kanfalan taasifamaa jira. Bosonni gadi lakkifama kaarbondaayi'oksaayidii dhorkuun, akkasumas kan qilleensa keessa jiru xuuxuun qulqullina qilleensaa yoo uume, itti fayyadamtoonni uummata biyyichaa qofa osoo hintaane, uummata addunyaa mara. Kanaafuu, faayidaa waliinii kanaaf maallaqa kanfaluun ammoo sirruma.Kunis daldala kaarboonii jedhama. Gama biraatiin bosonni qabeenya bishaan lafa jalaa gabbisuufi rooba harkisuun; akkasumas manca'a biyyoo lolaafi bubbeen uummamu hanbisuun oomishtummaa dabala. Yoo bosonni manca'e, burqaaleen nigogu, biyyoo gabbataan lolaan dhiqama, bubbeedhanis nihaxaawwama. Kun ammo oomishtummaa hiri'isuufi gogiinsa uumuun beelli akka uummamu taasisa. Biyyoon lolaan dhiqaamu, lafa qullaatti hambisuun ala miidhaa biraas qaba. Biyyoota akka Itoophiyaa keessatti, hidhoota laggeen humna ibsaa burqisiisaniitti nam'uun, akka harataan guuttaman gochuun umrii tajaajila hidhoota kanneenii gabaabsa. Bosonni madda nyaataa, qorichaafi oomishaalee mukaa addaddaa ta'uunis nitajaajila. Firiin, baalli, hiddi (hundee)fi qaamni biqiltoota addaddaa, ilma namaa dabalatee, bineensotaafi beeyladoota hedduuf nyaata ta'ee tajaajila. Ilmi namaa beeyladoota bosonaa addaddaa kan akka kuruphee, booyyee, karkarroo, tarraaca(Bosonuu)fi kkf, foon isaanii wanta nyaatuuf madda nyaataa dabalaan argata jechuudha. Qorichoonni aadaas ta'anii, ammayyaa hedduun biqiltootarraa oomishamu. Kuni ammoo fayyaa namaa eeguu bira darbee madda galiis ta'uun faayidaa guddaa laata. Kana malees, meeshaaleen mukarraa hojjetaman kan mana keessatti tajaajilan, kan akka siree, minjaalaa, barcumaafi kkf; akkasumas kan waajjiraalee keessatti tajaajilaniifi kan ijaarsi addaddaa ittiin raawwatamu bosonarraa argamu. Kun ammoo madda galii guddaadha. Kanneen malees, bosonni iddoo jireenyaa bineensootaafi beeyladootaa addaddaati. Bineensoonnifi beeyilladoonni bosonaa ammoo akkuma armaan olitti caqasne, madda nyaataa ta'uun ala, madda tuuriizimii ta'uun tajaajilu. Daawwattoonni biyya alaafi biyya keessaa, bineensota bosonaa, kan haalaan qalbii namaa hawwatan, daawwatanii bashannanuuf maallaqa guddaa kanfalu. Keessattuu daawwattoota biyya alaarraa sharafni alaa waan argamuuf, misooma biyyaaf qooda guddaa gumaacha. Gama kanaan biyyoonni hedduun kan akka Keeniyaafaa faayidaa guddaa argachaa jiru. Yeroo ammaa kana galii biyyoota alaarraa Keeniyaan argattu keessaa harki guddaan Tuurizimiirraa argama. Fakkeenyaaf, bara 2006 waggaa tokkotti Keeniyaan Tuurizimiirraa qofa Doolaara Ameerikaa miiliyoona 803 argatte. Galiin kun tarii amma dachaan dabale ta'a. Itoophiyaanis bosonaafi bineensota bosonaa ishii haalaan yoo kunuunsite, kutaa diinagdee Tuurizimiirraa galii sharafa alaa guddaa argachuu akka dandeessu amanamaadha. Tuurizimiin alattis, bineensonni bosonaa karaa seera-qabeessa ta'een adamsamuun, gogaa, ilkaan, gaafa, summiifi dafqa isaaniirraa galiin guddaan ni'argama. Fakkeenyaaf, gogaan naachaa, qeerransaa, hardiidaafi weennii uffata, faaya, afata manaafi biiroof wanta oolaniif, gatii guddaa baasu. Ilkaan arbaa faayaafi wantoota biroos tolchuuf oola. Dafqi moor'ee shittoof, summiin bofaa ammoo qoricha oomishuuf oola. Jaldeessi, hantuunnifi illeentiifaan ammoo yaalii saayinsiif oolu. Haa ta'u malee, faayidaalee faaran murteessoo ta'an kanneen hubachuu dhabuun ykn qaamota fedhii dhuunfaa isaanii qofa ari'aniin bosonni addunyaa kanaa saffisa nama yaachisuun manca'aa jira. Kun ammoo gammoojjummaa babal'isuun miidhaa guddaa qaqqabsiisaa jira. Rakkoon gammoojjummaa ammoo dachee kanarra iddoo heddu jiraatuus, Aafrikaa keessatti biyyoota gammoojjii Sahaaraa gaditti argamanitti, Itoophiyaa dabalatee, daran hammaata. Itoophiyaa qofa akka fakkeenyaatti yoo fudhanne, hongee baroota 1973-1975tti uummamee tureen qofa namoonni 3000, loon 80%, hoolonni 50%, gaalonniifi ro'oonni 30% caalan dhumanii ture. Rakkinni hongee kun eegasiiyis turee deddeebi'uun miidhaa qaqqabsiisaa jira. Hongeefi beela hamaa bara 2003tti biyyoota gaanfa Afrikaatti uumamee lubbuu uummata Somaalee heddu galafateefi, uummattoota gammoojjiiwwan kibbaafi baha Itoophiyaa 4000,000 ol gargaarsaaf saaxile akka fakkeenyaatti fudhachuun nidanda'ama. Akkuma waliigalaatti bosonni uffata dacheeti. Akkuma namni uffata malee qullaa hinmiidhagne, dacheenis bosona malee hinmiidhagdu. Bosonni wabii lubbuutis. Ilmaan namaa misoomuuf qofa osoo hintaane, lubbuun jiraachuuf bosonaafi qabeenya bosonni hammatee jiru badiirraa eeguufi kunuunsuu