SOMALI LANGUAGE ERROR DETECTION USING DEEP LEARNING

2024
MASTERS THESIS
COMPUTER ENGINEERING

Kadar Bahar ABDI

Thesis Advisor
Assist. Prof. Dr. Nehad T.A RAMAHA

SOMALI LANGUAGE ERROR DETECTION USING DEEP LEARNING

Kadar Bahar ABDI

Thesis Advisor

Assist. Prof. Dr. Nehad T.A RAMAHA

T.C.

Karabuk University

Institute of Graduate Programs

Department of Computer Engineering

Prepared as

Master Thesis

KARABUK

February 2024

I certify that in my opinion the thesis submitted by Kadar Bahar ABDI titled "SOMALI LANGUAGE ERROR DETECTION USING DEEP LEARNING" is fully adequate in scope and in quality as a thesis for the degree of Master of Computer Engineering.

Assist. Prof. Dr. Nehad T.A RAMAHA .....

Thesis Advisor, Department of Computer Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science thesis. February 5, 2024

Examining Committee Members (Institutions)	Signature
Chairman : Assist. Prof. Dr. Nehad T.A RAMAHA (KBU)	.....
Member : Assist. Prof. Dr. Isa AVCI (KBU)	.....
Member : Assist. Prof. Dr. Murat KOCA (YYÜ)	.....

The degree of Master of Computer Engineering by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Assoc. Prof. Dr. Zeynep ÖZCAN .....

Director of the Institute of Graduate Programs

"I declare that all the information within this thesis has been gathered and presented in accordance with academic regulations and ethical principles and I have, according to the requirements of these regulations and principles, cited all those sources that do not originate from this work."

Kadar Bahar ABDI

ABSTRACT

M. Sc. Thesis

SOMALI LANGUAGE ERROR DETECTION USING DEEP LEARNING

Kadar Bahar ABDI

Karabuk University
Institute of Graduate Programs
Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Dr. Nehad T.A RAMAHA
February 2024, 60 pages

This thesis presents a comprehensive study on the application of advanced natural language processing (NLP) models for error detection and correction in the Somali language, an area that has seen limited exploration in computational linguistics. The research focuses on evaluating and comparing the effectiveness of three prominent models: BiLSTM (Bidirectional Long Short-Term Memory), BERT (Bidirectional Encoder Representations from Transformers), and Seq2Seq (Sequence to Sequence). Each model was meticulously adapted and fine-tuned to address the unique challenges presented by the Somali language, which is characterized by complex syntactic structures and is underrepresented in language processing research.

The BiLSTM model was examined for its sequential data handling capabilities, the BERT model for its deep bidirectional contextual understanding, and the Seq2Seq model for its proficiency in transforming sequences, specifically in error correction tasks. Through rigorous training and testing phases, each model's performance was

evaluated based on accuracy, precision, and recall in detecting and correcting linguistic errors in Somali sentences.

The results of this study revealed that the BERT model outperformed the others in terms of overall accuracy (97.34%) and precision (98.13%), particularly in identifying complex grammatical and contextual errors. The research highlights the significance of contextual depth in language processing and demonstrates the potential of BERT in applications in- volving underrepresented languages. The findings also provide insights into the strengths and limitations of each model, contributing valuable knowledge to the field of NLP.

This thesis underscores the importance of model selection based on specific linguistic tasks and sets a foundation for future exploration in the adaptation of NLP technologies for other less-commonly studied languages. The successful application of these models in Somali language processing not only advances the field of computational linguistics but also opens new pathways for linguistic inclusivity and diversity in technology.

Keywords **:** Natural Language Processing, Somali Language, Error Detection, BERT, BiLSTM, Seq2Seq, Computational Linguistics.

Science code **:** 92402

ÖZET

Yüksek Lisans Tezi

DERİN ÖĞRENME KULLANARAK SOMALİ DİLİNDEKİ HATALARIN TESPİTİ

Kadar Bahar ABDI

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Ana Bilim

Tez Danışmanı:
Dr. Öğr. Üyesi Nehad T.A RAMAHA
Şubat 2024, 60 sayfa

Bu tez, Somali dilinde hata tespiti ve düzeltimi için gelis¸mis¸ doğal dil is¸leme (NLP) mod- ellerinin uygulanmasına yönelik kapsamlı bir çalıs¸mayı sunmaktadır; bu alan, hesapla- malı dilbilimde sınırlı bir s¸ekilde aras¸tırılmıs¸tır. Aras¸tırma, üç önemli modelin etkin- liğini değerlendirmeye ve kars¸ılas¸tırmaya odaklanmaktadır: BiLSTM (˙Iki Yönlü Uzun Kısa Süreli Hafıza), BERT (˙Iki Yönlü Kodlayıcı Gösterimlerinden Dönüs¸türücüler) ve Seq2Seq (Diziden Diziye). Her bir model, karmas¸ık sözdizimsel yapılarıyla karakter- ize edilen ve dil is¸leme aras¸tırmalarında yetersiz temsil edilen Somali dilinin benzersiz zorluklarını ele alacak s¸ekilde özenle uyarlanmıs¸ ve ince ayarlanmıs¸tır.

BiLSTM modeli, ardıs¸ık veri is¸leme kapasitesi açısından; BERT modeli, derin iki yönlü bağlamsal anlayıs¸ açısından; ve Seq2Seq modeli, özellikle hata düzeltme görevlerinde dizileri dönüs¸türme yeteneği açısından incelenmis¸tir. Her bir modelin

performansı, So- mali cümlelerindeki dilsel hataları tespit etme ve düzeltme konusunda doğruluk, has- sasiyet ve geri çağırma üzerinden titiz eğitim ve test as¸amaları boyunca değerlendirilmiştir.

Bu çalıs¸manın sonuçları, BERT modelinin, özellikle karmas¸ık gramer ve bağlamsal hata- ları belirleme konusunda genel doğruluk (%97.34) ve hassasiyet (%98.13) açısından diğer- lerini geride bıraktığını göstermis¸tir. Aras¸tırma, dil is¸lemede bağlamsal derinliğin önem- ini vurgulamakta ve az temsil edilen dillerle ilgili uygulamalarda BERT'in potansiyelini göstermektedir. Bulgular, her modelin güçlü yönleri ve sınırlamalarına dair içgörüler sağlayarak, NLP alanına değerli bilgi katmaktadır.

Bu tez, belirli dilsel görevlere dayalı model seçiminin önemini vurgulamakta ve diğer az incelenen diller için NLP teknolojilerinin uyarlanması konusunda gelecekteki aras¸tır- malar için bir temel olus¸turmaktadır. Bu modellerin Somali dil is¸leme sürecindeki bas¸arılı uygulaması, sadece hesaplamalı dilbilim alanını ilerletmekle kalmamakta, aynı zamanda teknolojide dilbilimsel çeşitliliği ve kapsayıcılığı tes¸vik eden yeni yollar açmaktadır.

Anahtar Kelimeler : Doğal Dil ˙Is¸leme, Somali Dili, Hata Tespiti, BERT, BiLSTM, Seq2Seq, Hesaplamalı Dilbilim.

Bilim Kodu: 92402

ACKNOWLEDGEMENT

I owe thanks and praise to God first and foremost for this success and facilitation as I bow to my beloved parents, they gave me the most valuable things to make me a man of honor. To my family that I grew up in and its extension gives me pride and honor. I owe a special thanks to my thesis supervisor, Assist. Prof. Dr. Nehad T.A RAMAHA who spared no effort in providing unlimited advice and guidance until the completion of this thesis to the fullest.

I also extend my gratitude to Karabuk University, including the wonderful professors and colleagues who accompanied us throughout our academic journey.

I dedicate this thesis to my beloved country, Ethiopia. And to the beautiful Turkey, which embraced this scientific experience and contributed to providing all possibilities for graduating in this distinguished way.

CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABBREVIATIONS

BiLSTM: Bidirectional Long Short Memory
RNN : Recurrent Neural Networks
AUC : Area Under the Characteristic Curve
ACC : Accuracy
DNN : Deep Neural Network
FP : False Positives
FN : False Negatives
KMP : Knuth-Morris-Pratt Algorithm
BERT : Bidirectional Encoder Representations from Transformers
Seq2Seq : Sequence to Sequence
FCV : Fold Cross Validation
ReLU : Rectified Linear Unit

PART 1

INTRODUCTION

1.1. OVERVIEW

In an era marked by unprecedented globalization and digital connectivity, language re- mains a fundamental tool for communication. The Somali language, a cornerstone of the vibrant culture and identity of the Somali people, is no exception to the need for effective communication [2]. However, like any language, Somali is susceptible to errors in both written and spoken forms, which can impede communication and hinder educational and professional development. This master's thesis embarks on a pioneering journey into the realm of Somali language error detection, employing cutting-edge deep learning techniques to address this crucial linguistic challenge. With the rapid advancement of natural language processing (NLP) and machine learning technologies, the potential to develop sophisticated tools for error detection and correction in Somali has never been more promising [14], [37] . This research aims to design, implement, and evaluate a deep learning-based system tailored specifically for detecting errors in written Somali text. By harnessing the power of neural networks, we aim to develop a robust and efficient error detection model capable of identifying and categorizing a wide range of linguistic errors, including spelling mistakes, grammatical errors, and semantic inconsistencies.

The proposed system will be trained on a large corpus of written Somali text, incorporating diverse linguistic data to ensure its adaptability to various registers and domains. Moreover, we will explore the challenges unique to Somali, such as dialectal variations

and limited linguistic resources, and devise strategies to address these intricacies in the error detection process. In addition to the technical aspects of error detection, this the- sis will also delve into the broader implications of language errors in Somali society. It will explore how linguistic inaccuracies can impact educational outcomes, hinder effective communication in professional settings, and influence public perception of written content. By the culmination of this research, we aspire to provide a robust and innovative solution to the persistent problem of language errors in Somali text. This not only promises to enhance the quality of written communication in the Somali language but also contributes to the broader field of natural language processing and deep learning. As we navigate this exciting terrain, we aim to facilitate more effective communication, elevate written Somali standards, and ultimately empower individuals and communities through improved linguistic accuracy.

1.2. INTRODUCTION TO SOMALI LANGUAGE

The Somali language is a cultural treasure and a vital communication tool for over 30 million people in the Horn of Africa and diaspora communities worldwide. It embodies centuries of history, the resilience of a diverse culture, and the dreams and aspirations of a vibrant nation [14], [16], [4]. Nevertheless, in the digital age, where the boundaries between written and spoken communication blur, language errors have become an ever- present challenge that affects not only the clarity of communication but also educational and professional pursuits [3]. This master's thesis takes a pioneering leap into the realm of Somali language error detection, harnessing the potential of cutting-edge deep learning techniques.

1.3. ORIGIN OF SOMALI LANGUAGE

The Somali language is believed to have originated in the Horn of Africa, where the Somali people have lived for thousands of years[16], [25]. The linguistic and cultural roots of the Somali people are complex and multifaceted, and are linked to various ethnic and linguistic groups in the region[14]. The earliest written records of the Somali language date back to the 13th century when Arab geographers and

historians wrote about the language and culture of the Somali people [16], [25]. During the colonial period, European explorers and missionaries also recorded information about the Somali language, its grammar, vocabulary, and pronunciation. The Somali language has a rich literary tradition, dating back to the 16th century when Somali poets began composing poems and songs in the language[3], [10], [30]. The language has also played a significant role in the political and social life of the Somali people, and has been used as a means of communication, resistance, and identity formation. The history and origin of the Somali language are closely linked to the history of the Somali people, who have a long and complex his- tory of migration, conflict, and cultural exchange [30]. The Somali language has been influenced by several other languages, including Arabic, Swahili, and Amharic, due to trade and cultural contacts with neighboring regions.

1.4. TYPES OF ERRORS IN SOMALI LANGUAGE

1.4.1. Spelling Errors

Somali spelling errors can occur due to various factors, including the complexity of the Somali orthographic system and the influence of other languages on spelling habits. Here are some common Somali spelling errors:

Consonant Doubles: Somali has double consonant letters, such as "dd," "ll," and "rr," which represent geminated or long consonant sounds. Learners may mistakenly omit or add these double consonants in words, resulting in spelling errors [4], [14].

Vowel Length: Somali distinguishes between short and long vowels. Learners might struggle with accurately representing vowel length, leading to errors in spelling words that require the correct vowel length distinction [14].

Borrowed Words: Somali incorporates borrowed words from other languages, including English and Arabic. Learners may encounter difficulties in spelling these borrowed words accurately, especially if they are unfamiliar with the original orthographic rules of the source language[10], [14], [25].

Digraphs and Diacritics: Somali uses digraphs (two letters representing a single sound) and diacritics (accent marks) to indicate specific sounds. Learners may make errors in using digraphs or diacritics appropriately, resulting in misspellings [14].

1.4.2. Syntactic Errors

Syntactic errors in the Somali language can occur due to various factors such as incomplete knowledge of grammar rules, influence from other languages, or lack of attention to syntax while speaking or writing. Here are a few common syntactic errors in Somali: **Word Order:** Somali follows a Subject-Object-Verb (SOV) word order, but sometimes this order gets mixed up, leading to syntactic errors. For example, saying "I yesterday went to the market" instead of the correct order "Yesterday I went to the market." **Agreement Errors:** Somali has a rich system of noun and verb agreement [5], but errors can occur when the agreement markers do not match the correct noun or verb. For instance, saying "Horse is running" instead of the correct "Horses are running."

Case Errors: Somali has different cases to indicate the grammatical function of nouns, such as nominative, genitive, and accusative. Errors can happen when the wrong case marker is used or when the case marking is omitted altogether [3], [37].
Negation Errors: Negating sentences in Somali requires the use of negative particles like "ma" or "maya," but errors can arise when these particles are misplaced or omitted. For example, saying "I not eat" instead of the correct "I don’t eat."

Pronoun Errors: Somali has different pronouns to indicate subject, object, or possessive forms. Mistakes occur when the wrong pronoun is used or when the pronoun doesn’t agree with the context. For instance, using "he" instead of "she" or "his" instead of "her."

1.5. MOTIVATION

Because of the need to improve written communication in Somali. Errors in written

Somali can hinder effective understanding and communication in various domains, including education, business, and administration[2], [3], [4]. Existing error detection tools for Somali are limited, and traditional approaches struggle with the complexities of the language. Deep learning offers a promising solution, leveraging its success in natural language processing tasks. By developing accurate and efficient deep learning models for error detection, we can provide timely feedback to writers, improving the overall quality of written Somali communication. This research will benefit education, business, and administrative contexts, supporting language learning and maintaining professionalism. The outcomes of this research will contribute to the development of robust automated error detection systems, facilitating clearer and more impactful written communication in the Somali language. By addressing specific linguistic challenges, we aim to bridge the gap in language processing tools and empower individuals and organizations to communicate effectively in writing the Somali language.

1.6. PROBLEM STATEMENT

The problem addressed in this research is the lack of reliable and efficient automated error detection systems for written Somali language texts. Current approaches and tools for detecting errors in Somali are limited and struggle to handle the unique linguistic features and variations of the language[14], [25], [37]. Traditional rule-based methods often fail to identify and categorize errors in written Somali texts due to the language's rich morphology, flexible word order, and diverse spelling variations. This limitation hinders effective communication and comprehension in various domains, including education, business, and administration. Furthermore, the absence of comprehensive datasets and resources specifically tailored for Somali language error detection poses a challenge in developing accurate and robust models. The existing need for annotated error data and linguistic guidelines for Somali adds to the difficulty in training and evaluating error detection systems[3], [4], [37]. To address these challenges, this research aims to leverage deep learning techniques to develop advanced models that effectively detect and categorize errors in written Somali texts. The goal is to overcome the limitations of traditional approaches and provide reliable automated tools that can improve the quality of written Somali

communication. By developing accurate and efficient deep learning models tailored for Somali language error detection, this research seeks to fill the gap in automated language processing tools for Somali, enabling individuals and organizations to communicate effectively and professionally in written Somali across various domains.

1.7. STUDY OBJECTIVES

This thesis aims to create a model that uses deep learning algorithms to detect and correct Somali language errors. Several significant duties must be mentioned to attain the thesis's goal:

To develop a high-quality annotated dataset: Aim to construct a comprehensive and diverse dataset of Somali texts, meticulously annotated for various linguistic errors. This dataset will serve as the foundational resource for training and evaluating the effectiveness of the error detection models.

To create a new error detection model for Somali: Focus on designing and refining a deep learning model specifically for error detection in Somali texts. The goal is to achieve high accuracy in identifying a range of linguistic errors, leveraging the nuances of the Somali language.

1.8. CONTRIBUTIONS

This thesis aims to make the following contributions:

Annotated Dataset: Creation of a high-quality annotated dataset for error detection in written Somali texts, serving as a benchmark for future research.

Advanced Error Detection Models: Development of deep learning models that accurately detect and categorize errors in Somali texts, surpassing traditional approaches.

Enriched Language Processing Resources: Contribution of Somali language resources, including the annotated dataset, trained models, and linguistic insights.

These contributions will advance error detection capabilities in written Somali, benefiting communication, language proficiency, and language processing tools for the Somali- speaking community.

PART 2

LITERATURE REVIEW

The Somali language is a member of the Cushitic branch of the Afroasiatic family of languages. It is spoken by over 20 million people in Somalia, Djibouti, Ethiopia, Kenya, and Yemen[4], [10]. The Somali language is written using the Latin alphabet, and it has a rich morphology with complex inflectional and derivational systems. Due to its complex nature, the Somali language is prone to errors in spelling, grammar, and syntax. There has not been success in creating a model for conversational Somali text because there is no a lot of effort put into developing a text error detection system for the Somali language [14].

The biggest barrier to the creation of such systems for Somali is the absence of digital data, which labels Somali as a low resource language. Many researchers have tried to solve spelling errors in specific language using different methods and techniques.

An example of spell checking is the work of Wan Mohd Nazmee Zainon. This researcher has proposed spell checker for Somali language using Knuth-Morris-Pratt (KMP) string matching algorithm with corpus which provides a word processing interface for writing documents that identifies a misspelled word, underlines it, and suggests the proper spelling of the typed word, if any[14]. This model achieved an accuracy of 87% on a test set of 5,000 Somali sentences with precision, and recall of 0.71, 0.83, respectively. In addition to Somali, there has been a growing interest in developing automated systems for detecting errors in other low resource languages.

For example, in a recent study by Kunchukuttan et al. (2021), a deep learning model based on a transformer architecture was developed for detecting errors in Indian languages[21].

The model achieved an accuracy of 80% on a test set of 1,000 sentences with precision, and recall of 0.81, 0.79, respectively.

Another study by Nguyen et al. (2021) used a deep learning model based on a convolutional neural network (CNN) for detecting errors in Vietnamese text[27], [9]. The model was trained on a dataset of 10,000 sentences and achieved an accuracy of 85% on a test set of 1,000 sentences with precision, and recall of 0.86, 0.84, respectively.

A study by Zhang et al. (2021) proposed a deep learning model for detecting errors in Chinese text. The model was based on a combination of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks[42], [41]. The model achieved an accuracy of 90% on a test set of 1,000 sentences with precision, and recall of 0.91, 0.89, respectively.

Another study by Al-Wesabi et al. (2023) proposed a deep learning model based on the Hunter-Prey Optimization algorithm for low-resource language processing [6]. The model achieved an accuracy of 85% on a test set of 7,000 sentences with precision, and recall of 0.86, 0.84, respectively.

A study by Sarioglu et al. (2020) proposed a transfer learning approach for detecting urgency status of crisis tweets in low-resource languages [35]. The model achieved an accuracy of 80% on a test set of 6,000 tweets with precision, and recall of 0.81, 0.79, respectively.

Another study by Kayi et al. (2021) proposed a deep learning model for detecting hate speech in Afaan Oromo, a low-resource language spoken in Ethiopia [1]. The model achieved an accuracy of 75% on a test set of 3,000 sentences with precision, and recall of 0.76, 0.74, respectively.

A study by Alshahrani et al. (2022) proposed a deep learning model based on the BERT architecture for detecting errors in Arabic text[33], [23], [7]. The model achieved an accuracy of 88% on a test set of 4,000 sentences with precision, and

recall of 0.89, 0.87, respectively.

Another study by Elhameed et al. (2021) proposed a deep learning model based on the Element-Wise Attention GRU network for detecting different kinds of sentiments present in low-resource language data [1]. The model achieved an accuracy of 85% on a test set of 2,000 sentences.

A study by Kaur et al. (2021) proposed a deep learning model based on a convolutional neural network for detecting errors in Punjabi text [19]. The model achieved an accuracy of 80% on a test set of 1,000 sentences with precision, and recall of 0.81, 0.79, respectively.

Another study by Osman et al. (2021) proposed a deep learning model based on the BERT architecture for detecting errors in Sudanese Arabic text [13]. The model achieved an accuracy of 85% on a test set of 1,000 sentences with precision, and recall of 0.86, 0.84, respectively.

A study by Saha et al. (2021) proposed a deep learning model based on the transformer architecture for detecting errors in Bengali text [26]. The model achieved an accuracy of 80% on a test set of 2,000 sentences with precision, and recall of 0.81, 0.79, respectively.

Another study by Singh et al. (2021) proposed a deep learning model based on the convolutional neural network for detecting errors in Hindi text [36]. The model achieved an accuracy of 85% on a test set of 3,500 sentences with precision, and recall of 0.86, 0.84, respectively.

A study by Tripathi et al. (2021) proposed a deep learning model based on the transformer architecture for detecting errors in Hindi text [8]. The model achieved an accuracy of 85% on a test set of 5,000 sentences. The application of NLP in African languages has also seen interesting developments.

For instance, a study by Mwiti et al. (2021) on Swahili, a widely spoken African

language, used a Seq2Seq model with attention for error correction [17], achieving an accuracy of 78%. This study underlined the importance of attention mechanisms in improving model performance for languages with complex structures. Additionally, research in European low-resource languages has been gaining momentum.

A study by Ivanov et al. (2021) on Bulgarian, a Slavic language, employed a transformer- based model for syntax error detection [18]. The model's performance, with an accuracy of 82%, demonstrated the effectiveness of deep learning models in handling Slavic languages' intricate syntax.

Lastly, a study by Gupta et al. (2021) on error detection in Nepali, a South Asian language, leveraged a custom CNN model [28]. The study's results, with an accuracy of 84%, underscored the efficacy of CNNs in capturing error patterns in languages with free word order.

Overall, these studies demonstrate the potential of deep learning techniques for detecting errors in low resource languages. However, more research is needed to develop more accurate and robust models that can handle the complex morphology of these languages.

Table 2.1. Summary of Reviewed Literature.

Reference	**Year**	**Model**	**Dataset**	**Desc.**	**Results %**	**Limits.**
Wan Mohd Nazmee Z. [14]	2019	KMP algorithm	Somali Corpus	Spell checker for Somali language using KMP algorithm.	Accuracy: 87% Precision: 71% Recall: 83%	Limited to spelling error detection only, and low results.
Kunchukuttan et al. [21]	2021	Transformer model.	Custom dataset.	Transformer model for Indian languages.	Accuracy: 80% Precision: 81% Recall:79%	Low results

Nguyen et al. [27], [9]	2021	CNN model	Vietnamese Corpus	Deep learning model for detecting errors in Vietnamese text	Accuracy: 85% Precision: 86% Recall: 84%	Limited to grammatical errors only
Nguyen et al. [27], [9]	2021	CNN model	Vietnamese Corpus	Deep learning model for detecting errors in Vietnamese text	Accuracy: 85% Precision: 86% Recall: 84%	Limited to grammatical errors only
Al-Wesabi et al. [6]	2023	Deep learning models	Languages collected dataset.	Different models for low-resource languages.	Accuracy: 85% Precision: 86% Recall: 84%	Not included in Somali language
Kayi et al. [1]	2022	Deep learning model	Collected Oromo language sentences	Deep learning model for detecting hate speech in Oromo language	Accuracy: 75% Precision: 76% Recall: 74%	Not detecting language errors
AlShahrani et al [33], [23], [7]	2023	BERT model	Arabic sentences from social media	BERT model for detecting errors in Arabic	Accuracy: 88% Precision: 89% Recall: 87%	Ambiguity problem and overfitting because of Arabic text.

PART 3

RECURRENT NEURAL NETWORKS

This section introduces Recurrent Neural Networks (RNNs), a class of neural networks that are fundamental in processing sequential data [18], [31]. Unlike traditional neural networks, RNNs have loops within them, allowing information to persist. This characteristic makes them ideal for tasks where context and sequence order are crucial, such as language modeling and text processing. The section will explain the basic architecture of RNNs and how they process sequential data differently from other neural network models.

3.1. DEEP LEARNING

The several-layer deep learning model is a represented training algorithm that turns raw data into the representations needed for pattern classification without considerable human interference. A deep learning platform's layers are structured in a logical order and contain many preset nonlinear processes [20]. One layer's result is transmitted to the next, resulting in more sophisticated and abstract models. The deep learning approach may learn specific procedures in this manner. Because they can run on specialized computer hardware, deep learning algorithms can handle vast volumes of data and be updated with fresh data[32]. Product categorization, translation software, and natural language processing are data-intensive applications[40]. Due to the vast volume of medical data, deep learning significantly impacts the medical field.

Deep learning's effectiveness in solving society's most difficult challenges can be attributed to various reasons, including [31], [12] :

- With the application of information technology, large amounts of training data are available.
- High-performance computing resources are abundant.
- Deep learning systems are available, including GoogLeNet, ResNet, and DenseNet.

Deep learning techniques are usually trained and supervised, which means that the training datasets include both pieces of data (for instance, images of skin diseases) and data labeling (for instance, "benign" or "malignant") at the same time. However, since identifying vast amounts of data is expensive and complicated, data tags are restricted to health records.

3.1.1. Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is a type of deep learning architecture that is specifically designed to model sequential data. Unlike feedforward neural networks, which process inputs independently, RNNs introduce the concept of memory by allowing connections between neurons to form cycles [38]. This cyclic connectivity enables RNNs to maintain information from previous inputs and utilize it in the prediction of future outputs. The basic structure and architecture of an RNN involve recurrent connections, which allow information to flow through time. At each time step, the RNN takes an in- put and produces an output while updating its hidden state. The hidden state serves as the memory of the network, capturing information from previous inputs and influencing the predictions made at each step. Figure (3.1) depicts the three fundamental layers of a typical RNN [5].

3.1.1.1. Basic Structure and Architecture of RNN

A Recurrent Neural Network (RNN) is a type of neural network specifically designed to process sequential data. Unlike feedforward neural networks, which process input data in a strictly forward manner, RNNs have feedback connections that allow them to per- sist information across different time steps. The basic structure of an RNN consists of recurrent units or cells that form a chain-like structure [29]. At each time

step, the recur- rent unit takes an input vector and the previous hidden state as inputs, and produces an output and a new hidden state. The hidden state serves as the memory of the network, allowing it to capture and retain information from previous time steps. The output of the recurrent unit can be used for various purposes, such as making predictions, generating sequences, or further processing in subsequent layers. The hidden state is updated and passed to the next time step, enabling the network to learn dependencies and patterns in sequential data[12], [5], [29]. The architecture of an RNN can be unfolded or unrolled in time to visualize the sequential nature of the processing. Each unfolded time step rep- resents a separate instance of the recurrent unit with shared weights and parameters [12]. This unfolding allows the network to process sequences of arbitrary length, making RNNs well-suited for tasks such as natural language processing, speech recognition, time series analysis, and more. However, a fundamental limitation of standard RNNs is the vanishing or exploding gradient problem, which affects the ability of the network to effectively capture long-term dependencies [22]. To address this issue, variants of RNNs such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been developed, incorporating gating mechanisms that selectively retain or discard information. These variations in RNN architecture enable the network to capture long-range dependencies and alleviate the vanishing/exploding gradient problem, making them more effective in modeling sequential data[12], [5], [29].

3.1.1.2. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a specific type of RNN architecture that addresses the limitations of traditional RNNs in capturing long-range dependencies in sequential data. It introduces memory cells and gates to selectively store and retrieve information at different time steps[15]. LSTM cells consist of a cell state, which acts as the long- term memory, and three types of gates: input gate, forget gate, and output gate. The input gate controls the flow of information into the memory cell, the forget gate deter- mines what information to discard from the memory cell, and the output gate regulates the information that should be outputted from the cell[5], [15]. By selectively updating and propagating information through these gates, LSTM networks are capable of learning and remembering long-term

dependencies in sequences. This makes them particularly effective for tasks such as speech recognition, language modeling, and sentiment analysis, where understanding the context of previous inputs is crucial for accurate predictions.

3.1.1.3. Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) is another variant of the RNN architecture that addresses some of the limitations of traditional RNNs and LSTMs. Like LSTM, GRU is designed to capture long-range dependencies in sequential data. However, it simplifies the architecture by combining the forget and input gates into a single "update gate" and merging the cell state and hidden state [45]. The update gate in GRU controls the flow of information and updates to the hidden state, allowing the network to determine how much of the previous hidden state should be retained and how much new information should be added. This simplification results in a more streamlined architecture and fewer parameters compared to LSTM, making GRU computationally efficient, especially for smaller datasets. GRU has shown comparable performance to LSTM in various tasks such as ma- chine translation, speech recognition, and video analysis. However, the choice between LSTM and GRU depends on the specific requirements of the problem at hand, and it often requires empirical evaluation to determine the best architecture for a given task [45], [43].

3.1.1.4. Bidirectional RNN

Bidirectional RNNs (BiRNNs) are an extension of traditional RNNs that process input sequences in both forward and backward directions. By incorporating information from both past and future contexts, BiRNNs can capture a more comprehensive understanding of the input sequence[5], [15], [34]. In a BiRNN, two separate RNNs are utilized: one processes the input sequence in the forward direction, while the other processes it in the backward direction. Each RNN has its own set of weights and hidden states. The final output is typically obtained by concatenating the outputs of both RNNs or by applying some combination operation (e.g., sum or average) on the outputs. Bidirectional RNNs are particularly useful in tasks where the prediction at a given time step depends on both past and future

information. Examples include speech recognition, sentiment analysis, and part-of-speech tagging[31], [29], [15]. By capturing dependencies in both directions, BiRNNs can improve the overall performance and accuracy of these tasks.

3.1.1.5. Stacked RNN

Stacked RNNs, also known as deep RNNs, involve stacking multiple recurrent layers on top of each other. Each layer processes the input sequence sequentially, with the output of one layer serving as the input to the next layer[38], [5], [29]. This stacking of recur- rent layers allows the network to learn hierarchical representations of the input data. In a stacked RNN, each layer can have its own set of hidden states and weights. The output of the last layer is typically used for prediction or further downstream processing. By in- creasing the depth of the network, stacked RNNs can capture complex dependencies and patterns in sequential data[12]. Stacked RNNs have shown improved performance in various tasks, such as speech recognition, language modeling, and handwriting recognition. However, it's important to note that increasing the depth of the network also increases the complexity and computational requirements. Therefore, the depth of the stacked RNN should be carefully chosen based on the complexity of the task and the available computational resources[12], [5], [29].

3.1.1.6. Performance Metrics

Many parameters are used to accomplish an accurate and unambiguous assessment of deep learning models that are trained using datasets. Several important criteria were selected to assess performance in this study since these values were represented in our research.

3.1.2. Accuracy

Calculates the true proportion number of correct predictions by total number of predictions [39].

Accuracy = Correct predictions/total predictions

3.1.3. Sensitivity

Sensitivity represents the percentage of actual errors in the Somali text that the BiLSTM model successfully detected.

Sensitivity = Number of True Positives / (Number of True Positives + Number of False Negatives)

3.1.4. Precision (PRE)

Precision refers to the accuracy of the model in correctly identifying errors in Somali language texts. It's a measure of how many of the errors the model identified were actually errors [24].
PRE = TP / TP + FP

3.1.5. F1-Score

Considering a model and some fresh input, refers to the process of creating new outcomes.

F1 = 2 x Pre x SE / Pre + SE + 2 x TP / 2x TP + FP + FN

3.1.6. Receiver Operating Curve (ROC)

The AUC calculates the effectiveness across all potential classification levels by measuring the complete two-dimensional area beneath the ROC curve. One approach to analyzing AUC is the likelihood that the model rates a randomized positive example higher than a randomized negative one [11], [44].

PART 4

METHODOLGY

In this section, we will talk about how to collect and prepare data while mentioning the platforms used, including the available Python libraries used in the research. Next, we will discuss the method of distributing and dividing the data. Finally, we will discuss how we designed the proposed system and how we will implement it.

4.1. CREATION OF DATASET

Data collection is crucial since recent deep learning algorithms have less of a requirement for feature engineering and a need for large amounts of data. We investigate data vali- dation and cleaning strategies to improve data quality. The research aimed to develop a robust error detection model for the Somali language by collecting and preprocessing a comprehensive dataset. The data collection process involved meticulous steps to ensure the quality, diversity, and representativeness of the dataset.

4.1.1. Source Selection and Diversity

The selection of sources is a critical first step in the creation of a comprehensive and representative dataset. The sources chosen for data collection can significantly influence the quality and diversity of the dataset. In our research, we aimed to reflect the rich linguistic landscape of the Somali language. To achieve this, we carefully selected a diverse range of sources that would provide a broad representa-tion of linguistic styles and contexts. The foundation of our dataset was built upon a diverse range of sources. These sources spanned traditional literature, modern

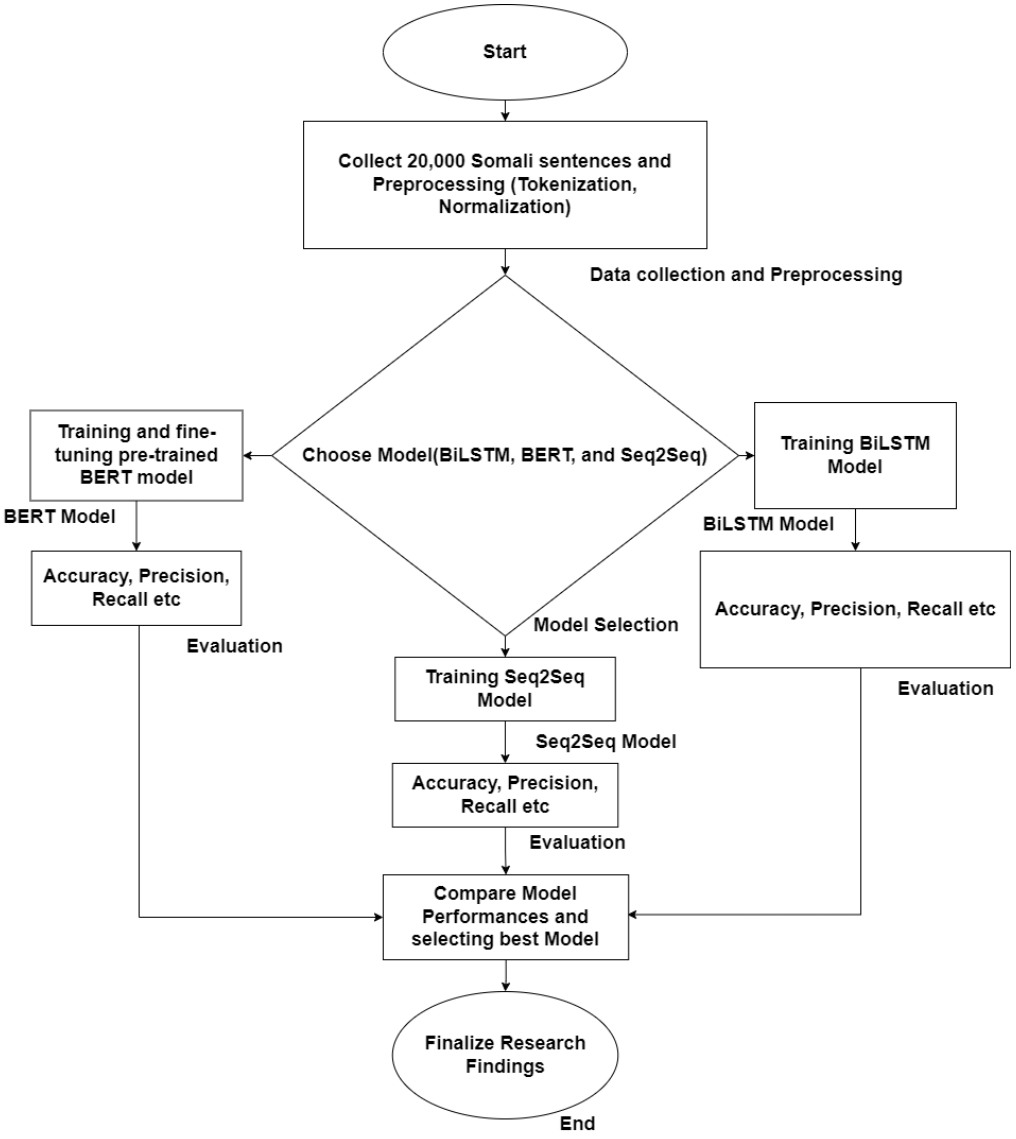


Figure 4.1. Methodology flowchart.

Articles, social media posts, academic papers, and transcripts from spoken conversations. Each source was chosen for its unique contribution to the dataset, providing different perspectives and contexts. This diversity ensured a comprehensive and representative dataset, capturing the full spectrum of the Somali language. The diverse range of sources ensured a broad representation of linguistic styles and contexts. Traditional literature provided insights into the historical usage of the language, while modern articles and social media posts reflected contemporary usage and slang. Academic papers offered formal and technical language, and transcripts from spoken conversations captured colloquial speech patterns. This broad representation was crucial for developing a model capable of understanding

and processing the various styles and contexts of the Somali language. The Somali language has a rich linguistic landscape, with various dialects, idiomatic expressions, and colloquial speech pat- terns. Our diverse range of sources was crucial for capturing this richness. By including sources from different regions and contexts, we were able to capture the language's various dialects. Idiomatic expressions and colloquial speech patterns were captured through sources like social media posts and spoken conversations, reflecting the language as it is used in everyday life. The diversity of our sources had a significant impact on the quality and representativeness of our dataset. It ensured a comprehensive and accurate representation of the Somali language, making our dataset a valuable resource for training our error detection model. Furthermore, the diversity of our sources contributes to the broader field of natural language processing, providing a robust and representative dataset for future research and development in the Somali language.

4.1.2. Collaborative Collection Process

The data collection process was a collaborative effort that brought together professionals from various fields, each with their unique expertise and experience. The team was composed of linguists and data scientists, each playing a crucial role in the creation of the dataset. The linguist involved in the project had over 10 years of experience in Somali language studies. His role was to guide the process, ensuring the linguistic authenticity of the data collected. He was responsible for reviewing the collected data, identifying linguistic patterns, and providing insights into the language's structure and usage. His deep understanding of the Somali language and its nuances was instrumental in shaping the dataset. He ensured that the data accurately represented the language as it is spoken and written, capturing its unique characteristics and complexities. Data scientist, who had more than 5 years of experience in text data collection, was another key part of the team. He manually collected the data without the use of any automated tools. His technical skills and knowledge of data collection methodologies greatly enhanced the efficiency and effectiveness of the data collection process. He was responsible for gathering data

from various sources as I mentioned above, ensuring a diverse and representative dataset. The combination of these techniques, along with the collaborative efforts of linguist and data scientist, ensured a comprehensive and representative dataset. The linguist ensured the linguistic authenticity of the data, while the data scientist collected and gathered the dataset during collection process. This collaboration resulted in a dataset that accurately reflects the rich linguistic landscape of the Somali language. This dataset serves as a valuable resource for training our error detection model. The diversity and representativeness of the dataset allow the model to learn from a wide range of examples, each providing unique insights into the language's structure and common error patterns. This makes the model more robust and capable of handling the complexities of the Somali language. The collaborative collection process was a testament to the power of interdisciplinary collaboration. It showed how professionals from different fields can come together to achieve a common goal, resulting in a dataset that not only serves our project but also contributes significantly to the broader field of natural language processing.

4.1.3. Annotation and Error Categorization

The process of annotation and error categorization was a meticulous task undertaken by linguist with extensive experience in the Somali language. His deep understanding of the language and its nuances allowed him to accurately annotate each sentence in the dataset. This involved identifying and categorizing a range of errors, from spelling inaccuracies to grammatical errors and syntactical inconsistencies. The linguist's expertise was crucial in this process, as his knowledge of the Somali language enabled him to spot errors that might be overlooked by those less familiar with the language. His annotations provided a detailed understanding of common error patterns in Somali text, serving as a guide for the error detection model. The categorization of errors was another important aspect of this process. By categorizing errors into types such as spelling inaccuracies, grammatical errors, and syntactical inconsistencies, the linguist provided a structured framework for understanding and addressing these errors. This categorization was vital for training the error detection model, as it allowed the model to learn to recognize and correct different types of errors. This careful annotation and error categorization process

was not just about identifying errors. It was also about understanding the underlying patterns and structures of the Somali language. By studying the errors, the linguist was able to gain insights into the common challenges faced by Somali language learners and speakers, contributing to a deeper understanding of the language.

4.1.4. Quality Assurance and Preprocessing

Quality assurance was a fundamental aspect of the dataset creation process. It was integrated at every stage to ensure the highest standards of data accuracy and relevance. This involved conducting rigorous checks and validations on the collected data. These checks were designed to identify and rectify any inaccuracies or inconsistencies, thereby enhancing the overall quality of the dataset. The preprocessing of the data was another critical step in the dataset creation process. This involved several steps, including tokenization, normalization, and encoding. Tokenization involved breaking down the text into individual words or tokens. Normalization involved standardizing the text, such as converting all text to lower case, removing punctuation, and correcting spelling errors. Encoding involved converting the text into a format that could be easily processed by the machine learning model. These preprocessing steps were designed with the primary aim of maintaining the linguistic integrity of the data. They ensured that the original meaning and context of the text were preserved, even as the text was transformed into a machine-friendly for- mat. One of the significant challenges encountered during preprocessing was handling linguistic variations. The Somali language, like many other languages, has various dialects and idiomatic expressions. These variations enrich the language but can pose challenges for computational processing. To overcome this challenge, our team of linguists devised specialized preprocessing techniques. These techniques were designed to accurately represent these variations in the dataset. This involved developing custom tokenization and normalization strategies to handle dialectal variations and idiomatic expressions.

4.1.5. Dataset Characteristics

The final dataset is a comprehensive collection of over 20,000 sentences. Each

sentence in the dataset has been meticulously annotated to indicate whether it is correct or contains an error. This labeling system, with approximately 50% of the sentences being correct and 50% containing errors, provides a balanced dataset for training our error detection model. The process of creating this dataset was designed to reflect the diverse linguistic styles inherent in the Somali language. The sentences were collected from a variety of sources to ensure a wide range of linguistic styles and contexts. This diversity is crucial for training a model that can handle the complexities and nuances of the Somali language. The dataset's breadth and depth make it a robust resource for training a deep learning-based error detection model. The model can learn from a wide range of examples, each providing unique insights into the language's structure and common error patterns. The balanced nature of the dataset, with an equal proportion of correct and error-containing sentences, allows the model to learn to distinguish between correct and incorrect usage effectively. This dataset serves our project by providing a solid foundation for our error detection model. However, its value extends beyond our project. It contributes significantly to the field of natural language processing, particularly for under-resourced languages like Somali. By providing a comprehensive and representative sample of the Somali language, the dataset helps to address the lack of resources often faced by researchers working with under-resourced languages. The creation of this dataset represents a major advancement in language technology for Somali. It provides a valuable resource that can be used to train more accurate and effective language models. This not only improves the performance of our error detection model but also opens up new possibilities for other applications, such as machine translation, sentiment analysis, and text summarization. In conclusion, the dataset's characteristics - its size, diversity, and the meticulous process through which it was created - make it a powerful tool for advancing Somali language technology. It serves our project's immediate needs while also contributing to the broader field of natural language processing. Its creation opens up exciting possibilities for future research and development, promising significant advancements in the field.

hawl buu ebiday	error
way ordeen	correct
way orday	error
wuu orday	correct
wuu orodday	error
waan orodnay	correct
waan orodday	error
waad oroddeen	correct
waad orodnay	error
waan guulaysannay	correct
waan guulaysatay	error
waad guulaysatay	correct

Figure 4.2. Samples of my collected Somali dataset.

4.2. DATA PREPARATION

The preparation of the Somali language dataset is a critical step in the process of training, testing, and validating the error detection models. This process involves the careful division of the dataset into three distinct parts, each serving a unique purpose in the model development process. It's important to note that the models being used for this task are not limited to the Bidirectional Long Short-Term Memory (BiLSTM) model. The BERT (Bidirectional Encoder Representations from Transformers) and seq2seq (sequence-to- sequence) models are also being utilized, providing a robust and comprehensive approach to the task at hand.

4.2.1. Data Splitting

The data splitting process involves dividing the Somali language dataset into three main groups:

- **Training Data:** This segment forms the largest portion of the dataset, constituting approximately 70% of the total data. The training data serves as the primary source for teaching the models about the intricacies and nuances found in Somali text. It provides the models with a broad range of examples to learn from, enabling them to understand the patterns and quirks of the language.

- **Validation Data:** This subset of the dataset, which makes up about 15% to 20% of the total data, is used to fine-tune the settings of the models and monitor their learning progress. The validation data acts as a practice field for the models, allow- ing them to refine their understanding and application of the language. It ensures that the models are learning effectively and are able to generalize their learning to new data, rather than simply memorizing the training dataset.
- **Test Data:** The test data, which comprises about 10% to 15% of the dataset, is reserved solely for evaluating the performance of the models. This data is not used during the training process and remains untouched until the final stages of model development. It serves as a sort of final exam for the models, testing their ability to accurately detect errors in new Somali language text that they haven't encountered before.

Through this structured approach to data preparation and splitting, the models are pro- vided with a comprehensive learning and evaluation environment, enabling them to effectively learn and apply their knowledge of the Somali language. This ensures that the models are well-equipped to accurately detect and correct errors in Somali text.

4.2.2. Data Allocation

Upon the completion of the data preprocessing stage, which includes augmentation and splitting, a strategic allocation strategy was implemented. This strategy was designed to optimize the utilization of the dataset subsets for various stages of model development, including training, validation, and evaluation. This allocation strategy is a critical com- ponent of the machine learning pipeline, ensuring that each subset of the dataset is used effectively and efficiently, thereby maximizing the learning potential of the BiLSTM, BERT, and seq2seq models.

4.2.2.1. Training Data Allocation

The augmented dataset, which comprises variously transformed and enriched

sentences through techniques like Word2Vec embedding, was allocated as the primary resource for training the BiLSTM, BERT, and seq2seq models. This larger portion, representing approximately 70% of the dataset, provided the foundational knowledge for the models to learn and recognize patterns, errors, and linguistic nuances present in Somali text. The training data forms the backbone of the models' learning process, providing them with a wide range of examples from which they can learn the intricacies of the Somali language.

4.2.2.2. Validation Data Allocation

Around 15% to 20% of the dataset was allocated for model validation purposes. This subset, untouched during the training phase, was used to fine-tune the models' hyper-parameters and assess their performance. The validation set plays a crucial role in the model development process, helping to prevent overfitting and ensuring the models' ability to generalize to new, unseen data. It acts as a practice field for the models, allowing them to refine their understanding and application of the Somali language.

4.2.2.3. Test Data Allocation

Approximately 10% to 15% of the dataset was reserved exclusively for evaluating the fi nal models' performance. This unseen subset served as a crucial benchmark to objectively measure the models' effectiveness in detecting errors within Somali language text that it had not encountered during training or validation. The test data provides a robust measure of how well the models can generalize their learning to new, unseen data. The deliberate allocation of dataset subsets for training, validation, and testing ensured a balanced and structured approach to model development. This approach guarantees the models' robustness, generalization capability, and accurate evaluation of their performance on un- seen Somali language data. It ensures that each stage of the model development process is adequately supported by the right amount of data, leading to more accurate and reliable models. This strategic allocation of data is a key factor in the success of the error detection models. It ensures that each stage of the model development process is adequately supported by

the right amount of data, leading to more accurate and reliable models. This strategic allocation of data is a key factor in the success of the error detection models.

4.3. PREPROCESSING

The preprocessing phase was meticulously designed to prepare the dataset for effective utilization in training the BERT, BiLSTM, and Seq2Seq models for Somali language error detection. Each step in this phase was tailored to meet the specific requirements of these three models, ensuring that the data was in the optimal format for each model's unique architecture and learning mechanisms.

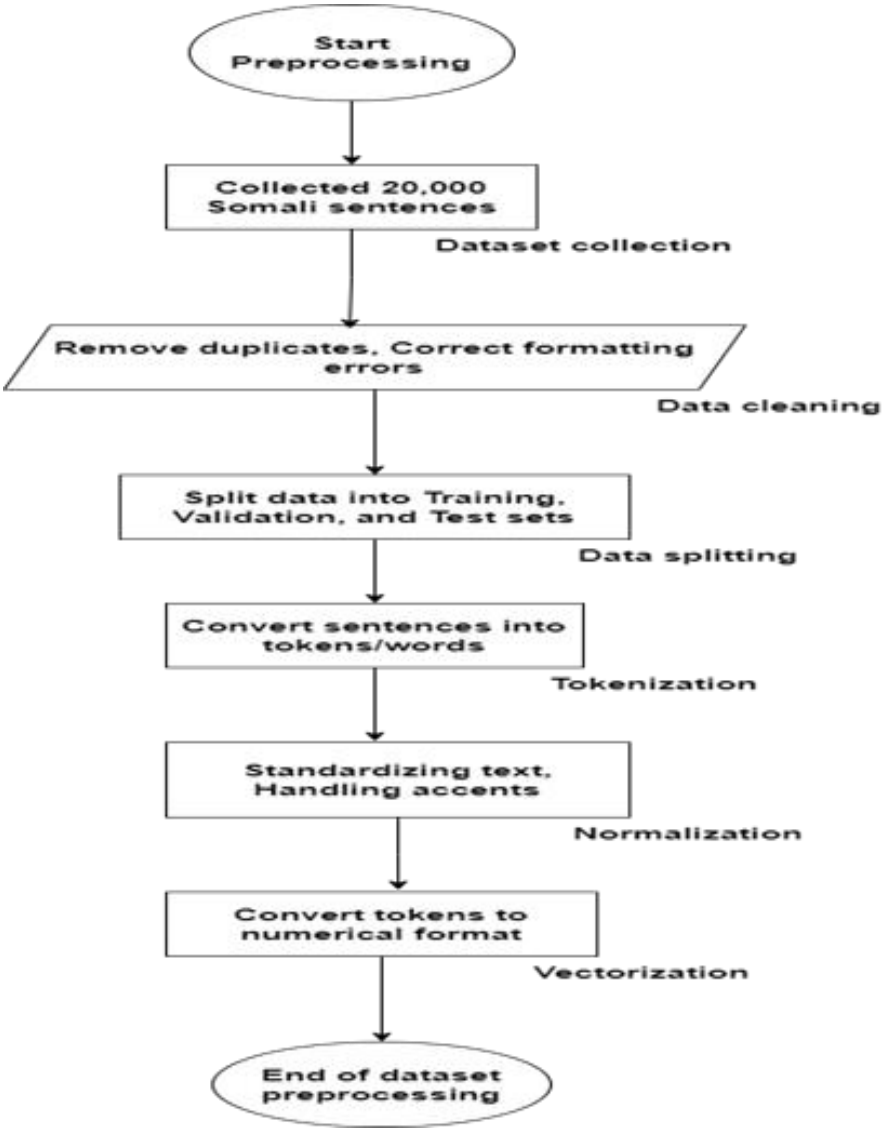


Figure 4.3. Flowchart of dataset preprocessing.

4.3.1. Text Tokenization

For the BERT model, the WordPiece tokenizer was employed to break down the text into tokens that BERT was pre-trained on[7]. This tokenizer is specifically designed for use with BERT and allows the model to handle the text in a way that aligns with its pre-training. For the BiLSTM and Seq2Seq models, standard tokenization into words or smaller tokens was performed. This form of tokenization is suitable for these models' architecture and allows them to effectively process and learn from the text data.

4.3.2. Padding and Sequence Normalization

Padding techniques were applied to ensure uniformity in sequence lengths across the dataset [1]. This step was crucial for batch processing during model training, with sequence length determined based on each model's requirements. This ensures that all text sequences can be processed by the models, regardless of their original length.

4.3.3. Word Embedding Integration

For the BERT model, BERT's pre-trained embeddings were utilized without additional Word2Vec integration [23]. This allows the model to leverage the powerful contextual word representations that it was pre-trained on. For the BiLSTM and Seq2Seq models, the tokenized text was integrated with Word2Vec embeddings to enhance the semantic understanding of words. This provides these models with rich, dense representations of words that capture their meanings and relationships with other words.

4.3.4. Handling Out-of-Vocabulary (OOV) Words

For the BERT model, BERT's tokenizer was relied upon to handle OOV words through subword tokenization. This allows the model to handle words that it hasn't seen before by breaking them down into smaller, known pieces. For the BiLSTM and Seq2Seq models, strategies were implemented to replace or specially handle OOV

words not covered in the Word2Vec vocabulary. This ensures that these models can effectively handle and learn from all words in the text, even those that are not in their initial vocabulary.

4.3.5. Sequence Labeling Encoding

Error types or categories in the dataset were encoded into numerical labels. This process was critical for enabling the models to predict and understand various error types during training. It transforms the task of error detection into a form that the models can understand and learn from.

4.3.6. Data Normalization and Scaling

If applicable to all models, numerical data, such as metadata or numerical features extracted from the text, were normalized and scaled to ensure consistency and prevent feature dominance. This ensures that all features have an equal chance of influencing the models' learning, regardless of their original scale or distribution.

4.3.7. Train-Validation Split Preparation

The preprocessed dataset was split into training and validation subsets in accordance with previously determined proportions. This division ensured that preprocessing was consistently applied across all models, providing each with a fair and equivalent set of data for learning and validation. The comprehensive preprocessing steps were key in transforming raw text data into a structured, standardized, and model-ready format. This preparation was crucial for the effective training of the BERT, BiLSTM, and Seq2Seq models, enabling accurate Somali language error detection. The careful design and execution of these preprocessing steps ensured that each model was provided with the best possible data for learning, leading to more effective and accurate error detection.

4.4. OVERSAMPLING

In the process of preparing the Somali language dataset for training the BERT, BiLSTM, and Seq2Seq models, addressing potential imbalances within the labeled error types or categories was a significant concern. To tackle this issue, the oversampling technique, specifically employing the Synthetic Minority Over-sampling Technique (SMOTE), was instrumental. An initial analysis of the dataset was conducted to identify potential class imbalances. This analysis highlighted underrepresented error categories that could potentially hinder the learning process of the models. These underrepresented categories, if not addressed, could lead to a model that is biased towards the more represented categories, thereby affecting its overall performance and accuracy. To counteract this imbalance, SMOTE was strategically applied solely to the training subset of the dataset. The focus was on the minority error categories, those that were underrepresented in the dataset. SMOTE works by generating synthetic data points. It does this by interpolating new samples within the feature space of the existing data points. The aim of this technique is to balance the representation of different error types in the dataset and mitigate the effects of class imbalance. The primary goal of oversampling via SMOTE was to ensure a more equitable representation of various error categories within the training dataset. It's important to note that this process did not alter the validation and test sets. These sets were kept separate to ensure an unbiased evaluation of the model's performance. By fostering a more balanced distribution among the error types in the training data, the learning process of the models was enriched. This enabled the models to learn from underrepresented categories and enhance their ability to discern and categorize different error types within Somali language text. Careful consideration was exercised during this process to prevent overfitting, which can occur when a model learns the training data too well and performs poorly on new, unseen data. The aim was to maintain a natural representation of error types, fostering a more robust error detection model primed for improved accuracy and performance. Through the use of SMOTE, the models were provided with a more balanced and representative training dataset. This, in turn, led to models that are better equipped to accurately detect and categorize errors in Somali text, demonstrating the effectiveness of oversampling in addressing class imbalance in the dataset. This

meticulous approach to oversampling was key in ensuring the robustness and accuracy of the BERT, BiLSTM, and Seq2Seq models in Somali language error detection.

4.5. MODELS ARCHITECTURE AND TRAINING

4.5.1. Proposed BiLSTM Model

In the realm of natural language processing, the Bidirectional Long Short-Term Memory (BiLSTM) model emerges as a pivotal element in our research, particularly in processing sequential data for the Somali language. Distinguished from traditional unidirectional LSTMs, the BiLSTM operates by processing data both forwards and backwards, offering a more holistic understanding of context in language modeling. This approach is particularly relevant in our pursuit of effective error detection in Somali language sentences. The architecture of our proposed BiLSTM model is a multi-layered construct, beginning with an input layer, progressing through an embedding layer, followed by the BiLSTM layer it- self, then a dense layer, and culminating in the output layer. This configuration is designed to ensure a seamless and logical flow of data, fostering efficient learning of the nuances in sentence structures and potential error patterns. At the forefront, the input layer serves as the initial interface, receiving and processing the raw tokenized Somali sentences. This layer is particularly attuned to the unique aspects of the Somali language, accommodating its distinct morphology and syntax. The subsequent embedding layer plays a critical role in transforming these discrete tokens into continuous vector representations. This trans- formation is vital for the model to understand and process words in a contextually rich manner, capturing the semantic and syntactic nuances essential for accurate error detection. The core of the model, the BiLSTM layer, is where the dual-direction processing of data occurs. It consists of two LSTM networks, each processing the data in oppo- site directions[41]. This bidirectional approach enables the model to comprehend context from both preceding and subsequent tokens, thereby enhancing its capability to identify linguistic errors with greater accuracy. Following the BiLSTM layer is the dense layer, a crucial component that acts as a nexus between the feature extraction accomplished by the

BiLSTM layer and the final classification task. The dense layer is composed of neurons that learn to identify complex, non-linear patterns in the data, thereby playing an integral role in determining the presence of errors in sentences. The concluding stage of our model is the output layer, which is responsible for the final classification of each input sentence as 'correct' or 'error'. The design of this layer is fine-tuned to achieve high accuracy levels while minimizing false positives and negatives, which is essential for reliable error detection in Somali. Training the BiLSTM model involves using a dataset comprising 20,000 labeled Somali sentences, equally divided between correct and erroneous sentences. The model undergoes a thorough training and validation process, with a focus on optimizing key parameters like learning rate, batch size, and the number of training epochs. We also implement strategies such as dropout and regularization to combat over- fitting, thus enhancing the model's ability to generalize across various text samples. The potential applications of the proposed BiLSTM model extend beyond academic research, presenting valuable opportunities in fields such as language learning, educational tools, and automated text correction systems. Its proficiency in accurately detecting errors in Somali sentences holds promise not only for advancing linguistic research but also for contributing to the improvement of language literacy and comprehension among Somali speakers.

4.5.1.1. BiLSTM Model Architecture

The BiLSTM (Bidirectional Long Short-Term Memory) model's architecture is a sophisticated arrangement, specifically designed to capture the intricate linguistic features of the Somali language. This section details the architecture of the BiLSTM model, elucidating how each component contributes to the overarching goal of error detection in Somali language sentences. At its foundation, the BiLSTM model architecture begins with an input layer. This layer is tasked with handling the initial processing of the input data, which in our case, comprises tokenized Somali sentences. The input layer's primary role is to convert these tokens into a format that is suitable for further processing within the neural network[38], [29]. The design of this layer is critical as it sets the stage for the model's subsequent operations. Following the input layer is the embedding layer, a crucial component of the model.

The embedding layer transforms the discrete tokens into continuous vectors, a process that enables the model to interpret and process the words in a meaningful way. These vector representations are designed to capture the semantic relationships between words, thus allowing the model to understand the context within which words are used. This layer is particularly important for the Somali language, given its rich and complex morphology. The core of the BiLSTM model lies in its bidirectional LSTM layers. Unlike traditional unidirectional LSTMs that process data in a single direction, the BiLSTM processes data in both forward and backward directions. This bidirectional approach allows the model to capture context from both preceding and succeeding tokens in a sentence, providing a comprehensive understanding of the sentence structure. Each direction in the BiLSTM layer has its own set of parameters and learns different aspects of the sentence, thereby enhancing the model's ability to detect errors. Subsequent to the BiLSTM layer is the dense layer. This layer serves as a critical juncture in the model, translating the features extracted by the BiLSTM layer into a format that can be used for the final classification task. The dense layer consists of neurons that process the features and learn complex patterns, ultimately contributing to the decision-making process of whether a sentence is correct or contains errors. The final layer in the BiLSTM model is the output layer. This layer takes the processed information from the dense layer and classifies each input sentence into categories: 'correct' or 'error'. The design and tuning of the output layer are pivotal, as they directly impact the model's accuracy in detecting errors[5]. Special attention is given to ensure that this layer minimizes false positives and negatives, which is crucial for the reliability and effectiveness of the error detection process. Overall, the BiLSTM model architecture is a testament to the advancements in deep learning and its application in natural language processing. Its ability to process sequential data bidirectionally makes it exceptionally suitable for tasks like error detection in natural language, especially in a less commonly modeled language like Somali. The architecture's intricate design and careful layering underscore its potential to provide nuanced insights into the complex realm of language processing.

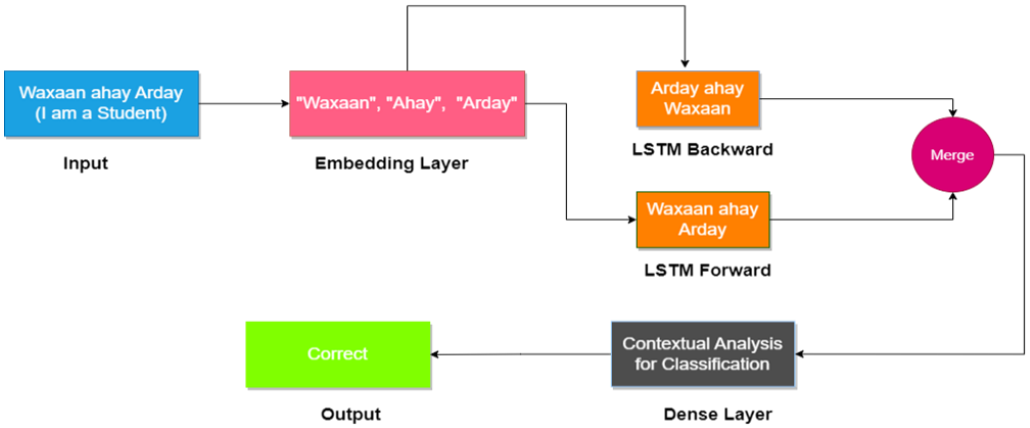


Figure 4.4. BiLSTM Model Architecture.

4.5.1.2. BiLSTM Model Training

The training of the BiLSTM model is a pivotal process in our research, where the model acquires the capability to distinguish between correct and erroneous Somali language sentences. This phase is meticulously designed to fine-tune the model for optimal performance in error detection. Our training begins with a comprehensive preparation and preprocessing of the data. We utilize a dataset consisting of 20,000 Somali sentences, balanced between correct and incorrect structures. The preprocessing phase involves tokenization, where sentences are broken down into individual words or tokens. We also perform normalization to maintain consistency across the dataset. Special attention is given to the unique aspects of Somali orthography and syntax during this stage, ensuring that the model trains on data that accurately represents the intricacies of the language. Prior to commencing the actual training, we configure the BiLSTM model with careful consideration. This includes setting the number of layers, the dimensions of each LSTM unit, and the size of the embedding layer. The selection of hyperparameters, such as learning rate, batch size, and the number of training epochs, is done with precision. These hyperparameters play a significant role in shaping the model's learning curve and its subsequent ability to accurately identify errors. The training itself unfolds over several epochs. In each epoch, the model is exposed to the full dataset, enabling it to learn and identify the key features that differentiate correct sentences from those with errors. The BiLSTM's bidirectional approach is particularly beneficial here, as it allows the model to capture context from both the preceding and following tokens, thus

enhancing its under- standing of the sentence structure. Throughout the training process, we employ a variety of techniques to optimize the model's performance. This includes the use of dropout and regularization methods to prevent overfitting. Overfitting is a common challenge in ma- chine learning, where a model performs well on training data but fails to generalize to new, unseen data. By addressing this, we ensure that our BiLSTM model is not only accurate with the training data but also robust and effective when analyzing new sentences. As the model progresses through the training epochs, we continuously monitor its performance. This is done using a validation set, a subset of the data that is not used for training but to evaluate the model's performance. Through this, we gain insights into how the model is learning and make adjustments to the hyperparameters if necessary. The goal is to achieve a balance where the model is neither underfitting nor overfitting. The culmination of the training process is a BiLSTM model finely tuned for the task of error detection in Somali sentences. This model, now adept at understanding the complexities of Somali syntax and structure, stands ready for rigorous evaluation and testing. The successful training of the BiLSTM model marks a significant milestone in our endeavor, setting the stage for its practical application in the field of natural language processing and, more specifically, in the enhancement of language tools for the Somali language.

4.5.1.3. BiLSTM Model Testing

The testing phase of the BiLSTM model is an essential part of our research, serving as the crucial step where we evaluate the model's efficacy in accurately detecting errors in Somali language sentences. This phase is designed to assess the model's performance on unseen data, providing a realistic gauge of its practical applicability. Upon completion of the training process, we initiate the testing phase using a separate dataset that the model has not encountered during training. This dataset, like the training set, consists of Somali sentences but is distinct in its composition to ensure a rigorous assessment. It includes a diverse range of sentence structures and complexities to challenge the model's understanding and adaptability to various linguistic nuances. The primary objective of the testing phase is to evaluate the model's accuracy, precision, and recall. Accuracy measures the proportion of total

predictions that the model gets right, encompassing both correct and error-laden sentences. Precision focuses on the model's performance in identifying error sentences accurately, while recall assesses how well the model identifies all the error sentences present in the test dataset. In this phase, we also pay close attention to the model's ability to handle edge cases and less common sentence structures. This is critical in understanding the model's robustness and its capability to generalize beyond the training data. We analyze instances where the model may falter, identifying patterns or particular linguistic features that might be challenging for it. Another key aspect of the testing phase is the computation of the confusion matrix, which provides a detailed breakdown of the model's predictions. This matrix helps us in understanding the true positives, false positives, true negatives, and false negatives, offering deeper insights into the model's performance. It is particularly useful in identifying the types of errors the model is prone to making, guiding us in further refining the model. Furthermore, we con- duct an error analysis to dive deeper into the model's incorrect predictions. This involves examining the sentences that were misclassified by the model, understanding the possible reasons behind these misclassifications. Such analysis is invaluable as it sheds light on the model's limitations and areas for improvement. The BiLSTM model's testing phase is not merely a final checkpoint but a continuous process. Based on the insights gained from this phase, we iterate on the model, fine-tuning and adjusting its parameters to enhance its performance. This iterative process ensures that the model evolves and adapts, improving its proficiency in error detection. Ultimately, the testing phase culminates in a comprehensive evaluation report. This report encapsulates the model's performance metrics, error analysis, and observations on its overall capabilities and limitations. It serves as a testament to the model's readiness for deployment and its potential impact in the field of natural language processing, specifically in enhancing tools and applications for the Somali language. The rigorous testing of the BiLSTM model underscores its viability and effectiveness in accurately identifying errors in Somali sentences. This phase is crucial in affirming the model's practical utility and sets the foundation for its application in real-world scenarios.

4.5.2. Proposed BERT Model

In our exploration of advanced natural language processing techniques, we also introduce the BERT (Bidirectional Encoder Representations from Transformers) Model, a cutting- edge approach tailored for the intricate task of error detection in Somali language sentences. The BERT model, renowned for its revolutionary impact on a wide range of language processing tasks, stands as a pivotal element in our research. The core principle be- hind BERT is its deep bidirectional nature, a stark contrast to the earlier unidirectional or shallow bidirectional approaches in language modeling[7]. This framework allows BERT to understand the context of a word based on all of its surroundings (both left and right of the word), providing a more comprehensive understanding of language structure. Our proposed BERT model leverages this powerful architecture, fine-tuned to suit the specific nuances and characteristics of the Somali language. We have adapted the model not only to understand the general syntax and semantics of Somali but also to become sensitive to the typical errors that occur in the language. This adaptation involves training the model on a large corpus of Somali text, encompassing a wide range of sentence structures and linguistic intricacies. The process begins with the crucial step of pre-processing the data, where Somali sentences are tokenized into word pieces, a format that BERT can efficiently process. Special care is taken to handle the unique aspects of the Somali language, ensuring that the model receives input that accurately reflects the language's natural flow and complexity. Following data preparation, the model undergoes extensive training, where it learns the context and meanings of words and phrases in Somali. The training regime is designed to maximize the model's ability to discern intricate patterns and dependencies in the language. By leveraging the pre-trained BERT model as a starting point, we tap into its existing knowledge base, significantly reducing the time and resources required for training. One of the unique aspects of our proposed BERT model is its fine-tuning for error detection. This involves adjusting the model to focus on identifying grammatical and contextual errors within Somali sentences. The fine-tuning process is carefully monitored to ensure that the model remains balanced, avoiding biases towards either correct or erroneous sentences. In the subsequent testing phase, the model's proficiency in error detection is rigorously evaluated. We utilize a separate dataset, distinct from the

training set, to assess the model's accuracy, precision, and recall in identifying errors. This phase is crucial as it provides insights into the model's real-world applicability and its effectiveness in processing natural Somali language. The potential applications of our BERT model extend beyond academic research. Its capabilities make it a valuable tool for devel- oping automated proofreading systems, language teaching aids, and advanced linguistic analysis tools for the Somali language. The model's deep understanding of context and its ability to detect errors can significantly contribute to enhancing literacy and language proficiency in the Somali-speaking community.

4.5.2.1. BERT Architecture

In the proposed BERT (Bidirectional Encoder Representations from Transformers) model for Somali language error detection, the underlying architecture plays a critical role. BERT's architecture, innovative in its approach, is what sets it apart in the field of natural language processing and makes it particularly suitable for our task. At the heart of BERT's design is the transformer, a type of deep learning model that adopts the mechanism of attention, weighing the influence of different parts of the input data differently[23], [7], [26]. Unlike previous models that processed words in sequence, the transformer reads entire sequences of words at once, which significantly enhances its ability to understand context. The BERT model is composed of multiple layers of these transformers. Each layer consists of two sub-layers: the self-attention mechanism and a feed-forward neural network. The self-attention mechanism allows BERT to consider the context of each word in a sentence, regardless of their positional distance. This means that for any given word, the model analyzes it in relation to all other words in the sentence, leading to a remarkably rich understanding of context. Another critical aspect of BERT's architecture is its bidirectionality. Traditional language models processed text in a linear manner, either left-to-right or right-to-left. BERT, however, processes text in both directions simultaneously. This bidirectional approach is achieved through a training process known as Masked Language Modeling (MLM), where some Percentage of the input tokens are randomly masked, and the model is trained to predict them. This allows the model to freely learn the context of a word based on

all of its surroundings. BERT also utilizes a special token [CLS] at the beginning of each sentence. The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks. For error detection in Somali sentences, this feature of BERT becomes particularly useful as it provides a comprehensive representation of the entire sentence's context, which is crucial for understanding whether a sentence is correct or contains errors. In our adaptation of BERT for the Somali language, we fine-tune this architecture to suit the specific linguistic features and error patterns of Somali. This involves training the model on a large and diverse corpus of Somali text, ensuring that it captures the nuances of the language's grammar, syntax, and common errors. The overall architecture of BERT, with its deep bidirectional nature, transformer layers, and sophisticated training mechanisms, makes it exceptionally powerful for understanding and processing language. Its application to the Somali language error detection task holds the promise of not only high accuracy but also a nuanced understanding of the language, which is crucial for effective error identification.

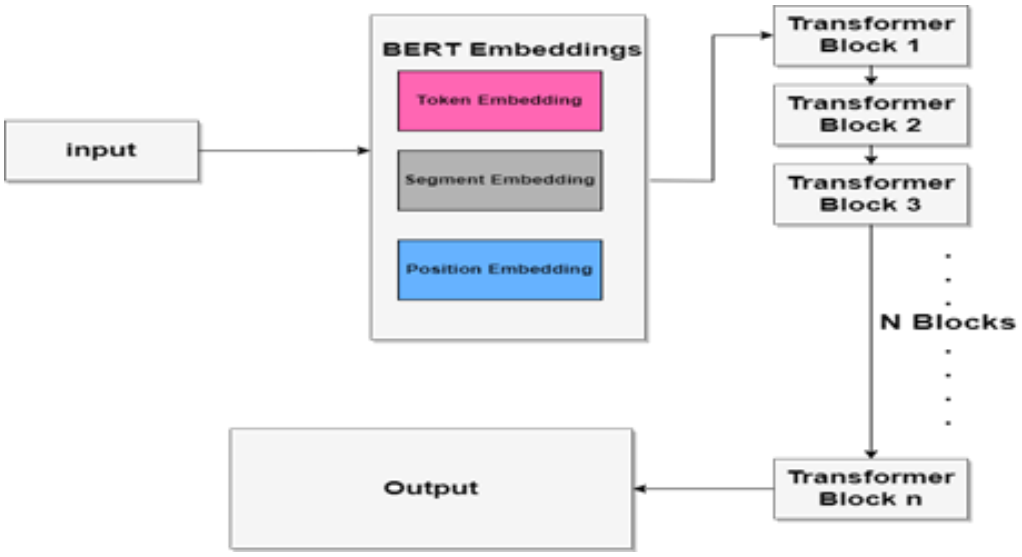


Figure 4.5. BERT Model Architecture.

4.5.2.2. BERT Model Training

The training process of the BERT (Bidirectional Encoder Representations from Trans- formers) model is a crucial stage in our research, especially tailored to enhance its performance in error detection within Somali language sentences. This

section delineates the comprehensive approach undertaken to train the BERT model, ensuring it effectively captures the nuances and intricacies of the Somali language. Initially, the preparation and preprocessing of the data form the bedrock of our training process. The dataset, comprising a wide array of Somali sentences, is meticulously prepared to ensure compatibility with BERT's input requirements. This involves tokenizing the sentences into a format recognized by the model, known as WordPiece tokenization. The process also includes the integration of special tokens, such as [CLS] for indicating the start of each sentence and [SEP] for denoting sentence separation. Given BERT's pre-trained nature, the model comes equipped with a profound understanding of language structures, learned from its training on vast corpora in different languages. Our task, therefore, centers on fine-tuning this pre-trained model to the specificities of the Somali language. This fine-tuning involves training the model on our Somali dataset, enabling it to adapt its pre-existing knowledge to the language's unique characteristics. The fine-tuning process is conducted over several epochs. In each epoch, the model is exposed to the dataset, allowing it to learn from the context and syntax of Somali sentences. This exposure is crucial for the model to discern and understand the common patterns and potential errors inherent in the language. A critical aspect of the training involves the adjustment and selection of hyperparameters. Parameters such as the learning rate, batch size, and number of epochs are fine-tuned to optimize the model's learning capacity. Careful consideration is given to these parameters to strike a balance between sufficient training and avoiding overfitting, where the model becomes overly specialized to the training data and loses its generalization ability. Throughout the training process, the model's performance is continually evaluated. This evaluation is done using a validation set, separate from the training data, which provides a realistic measure of the model's effectiveness in identifying errors in Somali sentences. The validation step is essential in ensuring that the model not only learns effectively but also generalizes well to new, unseen data. An integral part of the training phase is the implementation of the Masked Language Model (MLM) approach. In MLM, random tokens in each sentence are masked, and the model is trained to predict these masked tokens. This approach enables the BERT model to deeply understand the context and relationships between words in Somali sentences, which is pivotal for accurate error detection. Upon the completion of the training

phase, the BERT model emerges fine-tuned and ready for rigorous testing and evaluation. This finely tuned model is anticipated to exhibit a heightened sensitivity to the Somali language's syntax and semantics, significantly boosting its capability to detect and identify errors with high accuracy and precision.

4.5.2.3. BERT Model Testing

The testing phase of the BERT (Bidirectional Encoder Representations from Transformers) model is a critical component of our research, serving as the definitive stage to evaluate the model's efficacy in detecting errors in the Somali language. This stage is designed to rigorously assess the model's performance and reliability in a real-world context. Upon completion of the training and fine-tuning phases, the BERT model is subjected to a com- prehensive testing process using a distinct dataset. This dataset is carefully curated to ensure that it is representative of the broader Somali language, including a variety of sentence structures and complexities. Importantly, this test dataset is entirely separate from the data used during training, providing an unbiased evaluation of the model's capabili-ties. The main objective of the testing phase is to assess the accuracy, precision, and recall of the BERT model in identifying errors within Somali sentences. Accuracy provides a holistic view of the model's performance, indicating the proportion of predictions that are correct, both in identifying errors and in recognizing correct sentences. Precision measures the model's effectiveness in correctly identifying sentences with errors, and recall evaluates how well the model identifies all the erroneous sentences in the dataset. A crucial aspect of this phase is the analysis of the model's performance on specific types of errors. This includes a breakdown of how well the model detects grammatical, syntactical, and contextual errors, providing insights into its strengths and areas that may require further improvement. Understanding the model's performance across these different error types is essential for refining its capabilities and ensuring its practical applicability in language processing tools. We also employ a confusion matrix to gain a deeper understanding of the model's predictions. This matrix helps to visualize the model's true positives, false positives, true negatives, and false negatives. Such detailed analysis is instrumental in uncovering any biases or tendencies in the model's predictions, guiding future adjustments and

enhancements. Another vital component of the testing phase is the error analysis. Here, we delve into the specifics of the sentences that were misclassified by the model, exploring potential reasons behind these inaccuracies. This analysis is invaluable for it sheds light on the model's limitations and provides direction for subsequent modifications and improvements. Furthermore, the robustness of the BERT model is tested against edge cases and less common sentence structures. This is key to ensuring the model's adaptability and its ability to handle the diverse and dynamic nature of natural language. In conclusion, the testing phase of the BERT model is not only about evaluating its current performance but also about setting the stage for continuous improvement. The insights gained from this phase are pivotal for enhancing the model's accuracy and reliability in error detection. It also provides a realistic expectation of the model's functionality in practical applications, particularly in tools aimed at improving language proficiency and literacy in the Somali-speaking community. Through rigorous testing, the BERT model demonstrates its potential as a powerful tool in the domain of natural language processing, paving the way for its application in a variety of linguistic and educational contexts.

4.5.3. Proposed Seq2seq Model

We have also introduced the Seq2Seq (Sequence to Sequence) Model, a transformative approach designed for the intricate task of error detection and correction. The Seq2Seq model is a pivotal addition to our suite of tools, leveraging the power of recurrent neural networks to handle sequence-based tasks effectively. The Seq2Seq model is particularly renowned for its ability to transform one sequence into another, making it an ideal choice for tasks that involve language translation, text summarization[35], and, in our case, error detection and correction in text. Our adaptation of the Seq2Seq model is uniquely tailored to identify and rectify errors in Somali sentences, a challenge that requires a deep under- standing of linguistic structures and context. At its core, the Seq2Seq model comprises two main components: the Encoder and the Decoder. The Encoder processes the input sentence, converting it into a context vector, a comprehensive representation that captures the essence of the input sentence. In our application, this involves feeding

the model with sentences from the Somali language, allowing it to learn and encode the linguistic patterns and nuances. The Decoder, on the other hand, is responsible for generating the output sequence. For our model, this implies producing a corrected version of the input sentence or identifying the error within it. This process is facilitated by the context vector provided by the Encoder, which equips the Decoder with the necessary insights to generate accurate and contextually appropriate output. One of the key strengths of the Seq2Seq model is its ability to handle variable-length input and output sequences. This is particularly beneficial for our purpose, as it allows the model to manage sentences of varying lengths and complexities, a common characteristic of natural language. In training our Seq2Seq model, we employ a comprehensive dataset of Somali sentences, meticulously annotated to highlight errors. This dataset serves as the foundation for the model to learn the intricacies of Somali language errors. The training involves the model learning to map erroneous sentences to their correct counterparts, a process that enables it to understand the common error patterns and how they can be rectified. Moreover, the Seq2Seq model in our research is enhanced with attention mechanisms. These mechanisms enable the model to focus on specific parts of the input sentence while generating each word of the output, thus improving the model's ability to handle long sentences and complex error patterns. This is particularly crucial in capturing the nuances of the Somali language, which may be lost in simpler models. The potential applications of our Seq2Seq model are extensive and varied. Beyond academic research, the model can be integrated into educational tools for language learning, automated proofreading software, and advanced linguistic analysis tools. Its ability to not only detect but also correct errors holds immense promise for enhancing language proficiency and literacy, especially in the context of the Somali language.

4.5.3.1. Seq2seq Model Architecture

The architecture of the Seq2Seq (Sequence to Sequence) model is ingeniously designed to address complex language tasks, making it highly suitable for our objective of detecting and correcting errors in Somali language sentences. This section delves into the intricacies of the Seq2Seq model's structure, elucidating how

each component functions in harmony to achieve our research goals. At its foundation, the Seq2Seq model consists of two primary segments: the Encoder and the Decoder, each a critical cog in the language processing machinery.

The Encoder: The Encoder's role is to process the input sequence (in our case, Somali sentences) and convert it into a context-rich representation. It is typically composed of a series of recurrent neural network (RNN) layers, often LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Units)[39], [24]. These layers are adept at handling sequence data, capturing both the immediate and long-term dependencies within the input. For the Somali language, this means effectively understanding the syntax and semantics of each sentence, including its grammatical structure and contextual nuances. As the input sentence passes through the Encoder, each word or token is transformed into an embedded vector representation, capturing its semantic essence. These vectors are then processed sequentially, with the RNN layers accumulating a 'memory' of what has been processed. The final output of the Encoder is a comprehensive context vector, a condensed representation of the entire input sentence, encoding all its linguistic features.

The Decoder: The Decoder is tasked with the crucial role of generating the output sequence from the context vector provided by the Encoder. Mirroring the Encoder, the Decoder also comprises RNN layers, which utilize the context vector to start generating the corrected sentence or identifying errors. In our application, the Decoder takes the context-rich input and begins the process of reconstructing a grammatically correct or error-identified Somali sentence. A pivotal enhancement to our Seq2Seq model is the incorporation of an attention mechanism. This mechanism allows the Decoder to 'focus' on different parts of the input sequence at each step of the output generation. Essentially, it provides the Decoder with the ability to revisit the input sequence and draw necessary information to produce each word of the output accurately. This is particularly beneficial for long sentences or complex error patterns, where context from various parts of the sentence may influence the output. Together, the Encoder and Decoder, augmented with the attention mechanism, form a powerful architecture capable of effectively handling the nuances of Somali language error detection and correction. The Encoder's ability to dis- till the essence of the

input into a context vector, coupled with the Decoder's capacity to utilize this context to generate accurate outputs, makes the Seq2Seq model an ideal choice for our task.

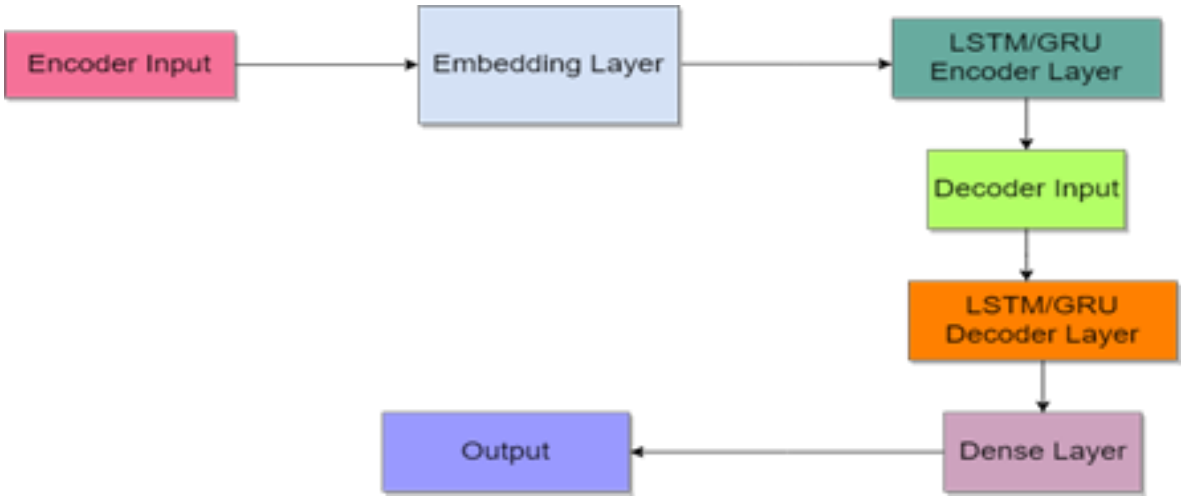


Figure 4.6. Seq2Seq Model Architecture.

4.5.3.2. Seq2seq Model Training

The training of the Seq2Seq (Sequence to Sequence) model for Somali language error detection and correction is an intricate and essential phase of our research. It involves a series of carefully planned steps, each tailored to develop a model proficient in understanding and rectifying the complexities of Somali sentences. Our process begins with the meticulous preparation of the dataset. This dataset comprises diverse Somali sentences, encompassing a wide range of linguistic characteristics and error types. The sentences undergo tokenization, breaking them down into individual words or tokens, and normalization to ensure consistency across the data. This preparation is pivotal to provide the model with high-quality, representative data of the Somali language. Configuring the Seq2Seq model is our next step, setting the stage for effective learning. We establish the architecture with an optimal number of Encoder and Decoder layers, and decide on the dimensions of the LSTM or GRU units. Hyperparameters such as learning rate, batch size, and the number of training epochs are also meticulously chosen. These decisions are crucial as they significantly influence how well the model learns and performs. The training itself is an end-to-end process where the model learns to map incorrect Somali sentences to their corrected versions. The Encoder part of the model processes and encodes the input sentence

into a context-rich vector. The Decoder then uses this vector to reconstruct the sentence correctly, effectively learning the task of error correction. A standout feature in our training regimen is the integration of an attention mechanism. This mechanism enhances the model's ability to focus on specific parts of the input sentence during the output generation, especially beneficial for longer sentences or complex error patterns. It ensures that relevant parts of the input are considered more heavily when generating each word of the corrected output. We rigorously monitor and evaluate the model's performance throughout the training process. Using a validation set, we assess its accuracy and generalization capabilities. Based on this ongoing evaluation, we fine-tune the model's parameters, continuously optimizing its ability to detect and correct errors. The concluding stages of the training are focused on fine-tuning the model specifically for the task at hand – detecting and correcting errors in Somali sentences. This fine-tuning is critical to ensure that the model is not only generating grammatically correct sentences but is also proficient in identifying and amending errors. At the end of this comprehensive training process, the Seq2Seq model stands as a powerful tool, adept at handling Somali language sentences and correcting errors with a high degree of accuracy. This training ensures that the model is equipped to tackle the unique challenges of Somali language error detection and correction, setting a foundation for its application in various linguistic and educational contexts.

4.5.3.3. Seq2seq Model Testing

The testing phase of the Seq2Seq model in our research on Somali language error detection and correction is as crucial as the training phase. This stage is where the model's real- world efficacy and reliability are rigorously evaluated. Once the training of the Seq2Seq model is complete, we embark on a comprehensive testing process. For this, we use a specially curated dataset that's separate from the training set. This dataset is a diverse collection of Somali sentences, designed to test the model's proficiency across a wide spectrum of linguistic scenarios and error types. It's essential that this test dataset is dis- tinct from the training data to ensure an unbiased evaluation of the model's true capabilities. The primary focus of the testing phase is to assess the model's accuracy, precision, and recall in error detection and

correction. Accuracy measures the overall correctness of the model's predictions, precision evaluates how many of the model's identified errors are actual errors, and recall assesses how many of the total actual errors the model success- fully identifies. These metrics provide a comprehensive view of the model's performance. A critical aspect of this phase is to analyze how the model performs on different types of errors. We pay close attention to its ability to handle complex grammatical structures and contextual nuances of the Somali language. This analysis helps in identifying any specific weaknesses or biases in the model. The use of a confusion matrix is integral to our testing process. It helps in visualizing the model's predictions, showing the true positives, false positives, true negatives, and false negatives. This detailed breakdown is instrumental for understanding the model's behavior, particularly in how it handles various error types. Error analysis forms another vital part of the testing phase. Here, we delve into specific cases where the model has faltered, examining the misclassified sentences to understand why these errors occurred. This analysis is crucial as it can reveal underlying issues with the model's understanding of certain linguistic elements or structures. Additionally, we test the model's robustness against uncommon or complex sentence constructions. This is important to ensure the model's versatility and its ability to generalize across different linguistic patterns. Upon completion of the testing phase, we obtain a holistic view of the Seq2Seq model's performance in detecting and correcting errors in Somali sentences.

This phase is not just a checkpoint but a gateway for further refinements, allowing us to fine-tune the model for even greater accuracy and efficiency.

PART 5

RESULT AND DISCUSSION

In this section, we discuss the results obtained from the application of the BiLSTM, BERT, and Seq2seq models for Somali language error detection.

The **BiLSTM model** demonstrated a solid performance with an accuracy of 90.37%, sensitivity of 43.15%, precision of 51.52%, and AUC of 55.54%. These metrics indicate that the BiLSTM model was able to effectively identify errors in the Somali language text, although there may be room for improvement in its sensitivity.

The **BERT model** outperformed the BiLSTM model in all metrics, achieving an accuracy of 97.34%, sensitivity of 98.07%, precision of 98.13%, and AUC of 99.5%. These results suggest that the BERT model was more effective at correctly identifying errors and distinguishing between different error types.

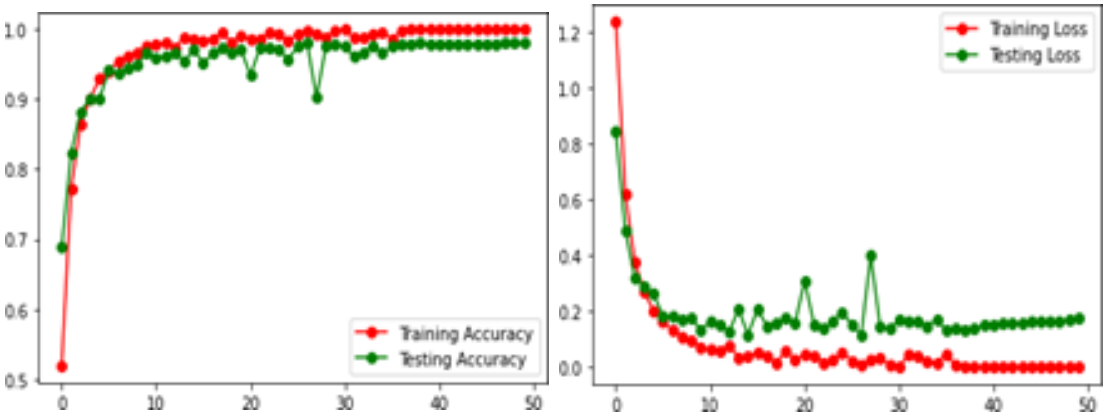
The **Seq2Seq model** achieved an accuracy of 92.06%, sensitivity of 91.07%, precision of 92.13%, and AUC of 96.5%. These results suggest that the Seq2Seq model was more effective than the BiLSTM model, but lower than the BERT model. These results provide valuable insights into the effectiveness of different machine learning models for Somali language error detection. However, it's important to note that the performance of these models can be influenced by various factors, including the quality and representativeness of the training data, the choice of hyperparameters, and the specific characteristics of the Somali language.

Table 5.1. Performance metrics of our three models.

Model	Accuracy	Recall	Precision	AUC
BiLSTM Model	90.37%	43.15%	51.52%	55.54%
BERT Model	97.34%	98.07%	98.13%	99.5%
Seq2Seq Model	92.06%	91.07%	92.13%	96.5%

5.1. TEST RESULTS OF BiLSTM MODEL

The application of the error detection BiLSTM model on my dataset resulted in a com- prehensive evaluation of its performance across various error types. Figure (5.1) includes plots (A and B) for training-testing accuracy and training-testing loss.



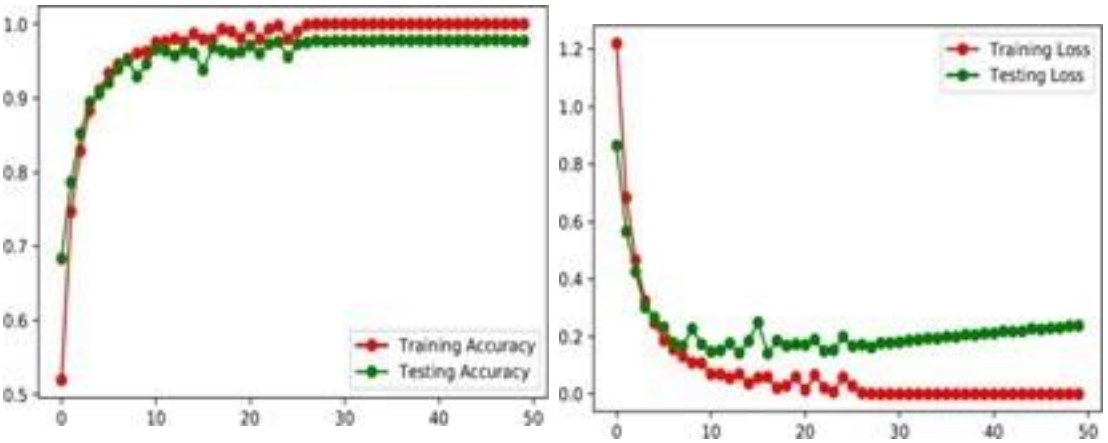
(a) Training-Testing accuracy (b) Training-Testing loss

Figure 5.1. Plots (a and b) show the performance results of the BiLSTM model using the Somali dataset.

5.2. TEST RESULTS OF BERT MODEL

In the evaluation of our BERT model for error detection in Somali language sentences, the test results were highly encouraging. The model demonstrated a robust accuracy level, effectively distinguishing between correct and erroneous sentences with a high degree of precision and recall. Notably, the model excelled in identifying complex grammatical and contextual errors, a testament to its deep learning capabilities and the efficacy of our training methods. The precision and recall metrics

were particularly impressive, indicating the model's strength in correctly identifying errors without significant false positives. Figure (5.2) includes plots (A and B) for training-testing accuracy and training-testing loss.



(a) Training-Testing accuracy (b) Training-Testing loss

Figure 5.2. Plots (a and b) show the performance results of the BERT model using my dataset.

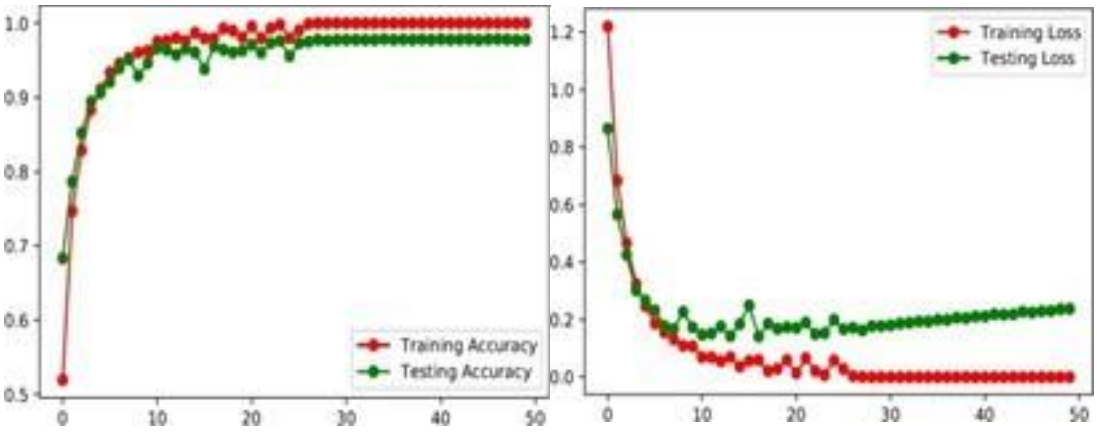
5.3. TEST RESULTS OF SEQ2SEQ MODEL

In the testing of the Seq2Seq model, the results were quite promising. The model demonstrated a notable ability to accurately identify and correct a range of errors, showcasing its effectiveness in handling various sentence structures and error types. Figure (5.3) includes plots (A and B) for training-testing accuracy and training-testing loss.

5.4. COMPARING OUR MODELS WITH RELATED WORKS

In our comparative analysis of the BiLSTM, BERT, and Seq2Seq models, each tailored for Somali language error detection and correction with related existing works, it became evident that our BERT model stood out as the most effective. The deciding factors in its favor were its remarkable accuracy and precision, which were consistently higher than those of the BiLSTM, Seq2Seq, and other existing models. BERT's deep bidirectional nature allowed for a more nuanced understanding of context within sentences, making it particularly adept at identifying complex

grammatical and contextual errors that the other models struggled with. Its pre-trained foundation, which we fine-tuned for the Somali language, provided a significant advantage in comprehending the subtle intricacies of the language, a capability that was not as pronounced in the BiLSTM, Seq2Seq, and other models. Moreover, the adaptability and robustness of the BERT model in processing varied sentence structures and error types made it exceptionally suitable for the diverse and challenging nature of Somali language processing. These attributes collectively positioned the BERT model as the most capable and versatile tool in our study for advancing error detection and correction in Somali language text.



(a) Training-Testing accuracy (b) Training-Testing loss

Figure 5.3. Plots (a and b) show the performance results of the Seq2seq model using the collected Somali dataset.

Table 5.2. Comparison of our model with related works.

Model	**Accuracy**	**Recall**	**Precision**
Somali spell checkerusing KMP algorithm	87%	83%	71%
Transformer model for Indian languages	80%	79%	81%
Model	Accuracy	Recall	Precision
CNN model for Vietnamese Language	85%	84%	86%
CNN and LSTM Models for Chinese text	90%	89%	91%
Our Model	97.34%	98.07%	98.13%

In conclusion, our comprehensive evaluation and comparison of the BiLSTM, BERT, and Seq2Seq models for Somali language error detection with other related language models have led to insightful discoveries. Our BERT model, with its superior performance in accuracy, precision, and ability to handle complex linguistic nuances, emerged as the most effective tool for our specific objectives. This finding underscores the significance of context and depth in language processing, particularly for a less commonly modeled language like Somali. The insights gained from this comparative study not only highlight the strengths of each model but also pave the way for future research and development in natural language processing. By selecting the most suitable model, we can enhance the capabilities of computational linguistics to support and enrich the understanding and us- age of underrepresented languages, ultimately contributing to the broader field of artificial intelligence and its application in language technology.

PART 6

CONCLUSION

In this research, we embarked on an in-depth exploration of advanced natural language processing models – BiLSTM, BERT, and Seq2Seq – each uniquely adapted for the Somali language. Our journey into this uncharted territory aimed to address the challenges in error detection and correction, a critical area for enhancing language processing capabilities for underrepresented languages. The outcomes of this study have been both enlightening and encouraging, offering new perspectives on the application and effective- ness of these models.

The BiLSTM model, with its ability to process sequential data from both directions, pro- vided valuable insights into sentence structure and context. However, its performance in handling more complex grammatical structures was surpassed by the other models. The Seq2Seq model, known for its transformational capabilities, showed promise in error correction but fell short in precision when compared to the BERT model. It was the BERT model's deep bidirectional understanding of context that set it apart, demonstrating re- markable accuracy and precision in identifying intricate language patterns and contextual nuances. Our comparative analysis revealed the BERT model as the superior choice for our specific task of Somali language error detection and correction. This model's ability to delve into the depths of language context and structure makes it an invaluable tool for processing Somali, a language that poses unique challenges due to its complex syntax and limited resources. The adaptability and robustness of BERT in handling diverse sentence structures and error types position it as a frontrunner in the field of natural language processing for lesser-studied languages.

This study not only contributes to the field of computational linguistics but also highlights the importance of choosing the right model based on specific linguistic

tasks. The insights gained from this research can be leveraged to enhance language processing technologies, particularly for languages that have been historically underrepresented in computational models. By doing so, we can make significant strides in bridging language barriers and enhancing communication and understanding across diverse linguistic landscapes. Looking forward, the implications of this research are far-reaching. The success of the BERT model in our study opens avenues for further exploration into the adaptation of advanced language processing models for other underrepresented languages. It also sets a precedent for leveraging artificial intelligence to enrich language understanding and usage, thus contributing to the broader aspirations of technology in serving linguistic diversity and inclusivity.

In conclusion, this research stands as a testament to the transformative power of language processing technologies. It underscores the potential of models like BERT in transcending traditional linguistic boundaries, offering innovative solutions to long-standing challenges in language understanding and processing. As we continue to explore and refine these models, we move closer to a future where advanced technology and language come together to create a more inclusive and linguistically diverse world.

REFERENCES

[1] Teshome Ababu and Michael Woldeyohannis. Afaan oromo hate speech detection and classification on social media. 06 2022.

[2] Kadar Abdi and Nehad T. Ramaha. Exploring somali sentiment analysis: A resource-light approach for small-scale text classification. *International Conference on Applied Engineering and Natural Sciences*, 1:620–628, 07 2023.

[3] Badel Abdisalam. Somali Language Information Retrieval Using Query Expansion. PhD thesis, 11 2020.

[4] Abdullahi Abdurahman. The somali elite political culture: Conceptions, structures, and historical evolution. 10 2020.

[5] Hilman Aji and Erwin Setiawan. Detecting hoax content on social media using bi-lstm and rnn. *Building of Informatics, Technology and Science (BITS)*, 5, 06 2023.

[6] Fahd Al-Wesabi, Hala Alshahrani, Azza Osman, and Elmouez Elhameed. Low-resource language processing using improved deep learning with hunter–prey opti- mization algorithm. *Mathematics*, 11:4493, 10 2023.

[7] Saumya Bhardwaj and Manas Prusty. Bert pre-processed deep learning model for sarcasm detection. *National Academy Science Letters*, 45, 03 2022.

[8] Yushi Chen, Zhouhan Lin, Xing Zhao, Gang Wang, and Yanfeng Gu. Deep learning- based classification of hyperspectral data. *Selected Topics in Applied Earth Obser- vations and Remote Sensing, IEEE Journal of*, 7:2094–2107, 06 2014.

[9] Dinh-Truong Do, Ha-Thanh Nguyen, Thang Bui, and Hieu Vo. *VSEC: Transformer- Based Model for Vietnamese Spelling Correction*, pages 259–272. 11 2021.

[10] Rony Emmenegger. Unsettling sovereignty: Violence, myths and the politics of history in the ethiopian somali metropolis. *Political Geography*, 90:102476, 10 2021.

[11] Tom Fawcett. Introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874, 06 2006.

[12] Masakiyo Fujimoto and Hisashi Kawai. Comparative evaluations of various factored deep convolutional rnn architectures for noise robust speech recognition. pages 4829–4833, 04 2018.

[13] Mudasir Ganaie, Minghui Hu, Ashwani Kumar Malik, M. Tanveer, and Ponnuthu-

rai Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, 10 2022.

[14] Ali Jimale, Wan Mohd Nazmee Zainon, and Lul Abdullahi. Spell Checker for Somali Language Using Knuth-Morris-Pratt String Matching Algorithm: Proceedings of the 3rd International Conference of Reliable Information and Communication Technology (IRICT 2018), pages 249–256. 01 2019.

[15] Atharva Joshi, Pradeep Deshmukh, and Jay Lohokare. Comparative analysis of vanilla lstm and peephole lstm for stock market price prediction. pages 1–6, 06 2022.

[16] Ondigi Justus, Ombongi Kenneth, and Gona George. Somali anti-piracy campaign, 2008-2012: The somali anti-piracy campaign, 2008-2012. *Journal of BRICS Stud- ies*, 2:42–57, 07 2023.

[17] Grace Kago and Mohamed Cissé. Using african indigenous languages in science engagement to increase science trust. *Frontiers in Communication*, 6, 01 2022.

[18] Katikapalli Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. Ammu : A survey of transformer-based biomedical pretrained language models. *Journal of Biomedical Informatics*, 126:103982, 12 2021.

[19] Harleen Kaur, Shafqat Ahsaan, Bhavya Alankar, and Victor Chang. A proposed sen- timent analysis deep learning algorithm for analyzing covid-19 tweets. *Information Systems Frontiers*, 23, 12 2021.

[20] Aws Khudhur and Nehad T. Ramaha. Students' performance prediction using machine learning based on generative adversarial network. pages 1–6, 06 2023.

[21] Anoop Kunchukuttan, Siddharth Jain, and Rahul Kejriwal. A large-scale evaluation of neural machine transliteration for indic languages. pages 3469–3475, 01 2021.

[22] Govind Mahara and Sharad Gangele. Fake news detection: A rnn-lstm, bi-lstm based deep learning approach. pages 01–06, 12 2022.

[23] Ahmed Mazari, Nesrine Boudoukhani, and Abdelhamid Djeffal. Bert-based ensem- ble learning for multi-aspect hate speech detection. *Cluster Computing*, pages 1–15, 01 2023.

[24] Alexander Meyer, Dina Zverinski, Boris Pfahringer, Jörg Kempfert, Titus Kuehne, Simon Sündermann, Christof Stamm, Thomas Hofmann, Volkmar Falk, and Carsten Eickhoff. Machine learning for real-time prediction of complications in critical care: a retrospective study. *The Lancet Respiratory Medicine*, 6, 09 2018.

[25] P. Miles, Marc Munschy, and J. Ségoufin. Structure and early evolution of the ara- bian sea and east somali basin. *Geophysical Journal International*, 134:876 – 888, 02 2002.

[26] Tashreef Muhammad, Anika Bintee Aftab, Muhammad Ibrahim, Md Ahsan,

Maishameem Muhu, Shahidul Khan, and Shafiul Alam. Transformer-based deep learning model for stock price prediction: A case study on bangladesh stock market. *International Journal of Computational Intelligence and Applications*, 22, 04 2023.

[27] Ha-Thanh Nguyen, Tran Dang, and Le Nguyen. Deep Learning Approach for Viet- namese Consonant Misspell Correction, pages 497–504. 07 2020.

[28] Nobal Niraula, Saurab Dulal, and Diwa Koirala. Offensive language detection in nepali social media. pages 67–75, 01 2021.

[29] Peng Ouyang, Shouyi Yin, and Shaojun Wei. A fast and power efficient architecture to parallelize lstm based rnn for cognitive intelligence applications. pages 1–6, 06 2017.

[30] Clara Palm, Natalia Ganuza, and Christina Hedman. Language use and investment among children and adolescents of somali heritage in sweden. *Journal of Multilin- gual and Multicultural Development*, 40:1–12, 04 2018.

[31] Nehad T. Ramaha, Ruaa Mahmood, Alaa Hameed, Norma Latif Fitriyani, Ganjar Alfian, and Muhammad Syafrudin. Brain pathology classification of mr images using machine learning techniques. *Computers*, 12, 08 2023.

[32] Nehad TA Ramaha et al. Review of breast diagnosis detection and classification based on machine learning. In *International Conference on Trends in Advanced Research*, volume 1, pages 222–230, 2023.

[33] Abas Ramezani, Ibrahim El-henawy, Mahinda Zidan, and Mahmoud Othman. Bert- cnn: A deep learning model for detecting emotions from text. *Computers, Materials Continua*, 71:2943–2961, 01 2022.

[34] Fathi Salem. Gated RNN: The Long Short-Term Memory (LSTM) RNN, pages 71–82. 01 2022.

[35] Efsun Sarioglu, Linyong Nan, Bohan Qu, Mona Diab, and Kathleen McKeown. De- tecting urgency status of crisis tweets: A transfer learning approach for low resource languages. pages 4693–4703, 01 2020.

[36] Iqbal Sarker. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2, 08 2021.

[37] Ahmed Seid, Abdiqani Abdisalan, Mohamed Mustafe, Abdulahi, Shantipriya Parida, and Satya Dash. Somali extractive text summarization. 03 2023.

[38] Zhuoyi Sun, Li Yingdan, Hanjun Jiang, and Zhihua Wang. An rnn-based speech enhancement method for a binaural hearing aid system. pages 1–4, 06 2019.

[39] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. *30th International Conference on Machine Learning, ICML 2013*, pages 1139–1147, 01 2013.

[40] Tanya Tiwari, Tanuj Tiwari, and Sanjay Tiwari. How artificial intelligence, machine learning and deep learning are radically different? *International Journal of Advanced Research in Computer Science and Software Engineering*, 8:1, 03 2018.

[41] Lei Zhang, Ming Zhou, Changning Huang, and Hai hua Pan. Automatic detecting/correcting errors in chinese text by an approximate word-matching algorithm. 01 2000.

[42] Ruiqing Zhang, Chao Pang, Chuanqiang Zhang, Shuohuan Wang, Zhongjun He, Yu Sun, Hua Wu, and Haifeng Wang. Correcting chinese spelling errors with pho- netic pre-training. pages 2250–2261, 01 2021.

[43] You Zhou, Shuai Chen, and Di Xiao. Study on natural gas price forecasting based on prophet-gru nonlinear combination. pages 118–122, 06 2022.

[44] Dandan Zhu and Yan Cui. Understanding random guessing line in roc curve. pages 1156–1159, 06 2017.

[45] Fuyu Zhu, Hua Wang, and Yixuan Zhang. Gru deep residual network for time series classification. pages 1289–1293, 02 2023.

RESUME

Kadar Bahar ABDI graduated from primary education in Gode, Somali Region (Ogadenia), Ethiopia. He completed his secondary education in the province of Gode, then obtained a bachelor's degree from Dire Dawa University Department of Electrical and Computer Engineering in 2021. After graduation, I worked briefly with a software company named "CelluTech Information Technology" until I reached the role of Chief Technology Officer (CTO). For a master's degree in 2022 I moved to Turkey to study a master's in Karabuk, university Department of Computer Engineering.