Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services

By

Fekadu Eshetu Hunde

A Thesis Submitted to

The Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia

July, 2021

Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services

By

Fekadu Eshetu Hunde

Advisor

Biruk Yirga (Ph.D.)

A Thesis Submitted to

The Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia

July, 2021

DECLARATION

I declare that this thesis's work entitled “Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services” has been carried out by me in the computer science and engineering department. Any sources of materials used in this thesis have been acknowledged, and a list of references provided. This thesis work has not been presented for another degree and diploma in any other universities.

Fekadu Eshetu Hunde	_____	_____
Name	Signature	Date

RECOMMENDATION

I the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services" prepared under my guidance by Fekadu Eshetu Hunde submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Computer Science and Engineering.

Therefore, I recommend the submission of revised version of the thesis to the department following the applicable procedures.

Biruk Yirga (Ph.D.)	_____	_____
Adviser Name	Signature	Date

APPROVAL PAGE

I/we, the advisors of the thesis entitled "Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services" and developed by Fekadu Eshetu Hunde, here by certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Biruk Yirga (Ph.D.)	_____	_____
Advisor Name	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Fekadu Eshetu Hunde have read and evaluated the thesis entitled Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

_____	_____	_____
Chairperson	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Biruk Yirga (Ph.D.)	_____	_____
Department Head	Signature	Date
_____	_____	_____
School Dean	Signature	Date
_____	_____	_____
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

I have received much support and feedback from many peoples in completing this study. But, first of all, I should thank St. Virgin Maryam and her son Jesus Christ, who has been so gracious in keeping me well and helping me complete this study.

Next, I owe my deepest gratitude to my advisor Dr. Biruk Yirga for his time, patience, and undeniably helping comments through this study. He was an inspiration for me to proceed whenever I face difficulties and he is easily approachable.

Then, I would like to express my profound gratitude to Dr. Mesfin Abebe, whose expertise was invaluable in giving the right direction, useful comments, and engagement from the beginning until the end. He continuously encourages me and always has been ready and enthusiastic to assist in any way he could throughout the process.

Next, I would like to acknowledge Mr. Gebrelibanos Teame who was the contact center director of Ethio telecom for his commitment and giving several pieces of information related to this study.

Last but not the least, I would like to thanks my best family specially my mother Serkalem Tegegn for her support. She was my everything. I would like to say to her “I love you mom”. “እማ ወድሻለሁ”, and “Harmee koo siin jaaladha”.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ALGORITHMS

LIST OF SAMPLE SOURCE CODES

LIST OF ACRONYMS AND ABBREVIATIONS

A

A.L.I.C.E · *Artificial Linguistic Internet Computer Entity*
AI · *Artificial Intelligence*
ANN · *Artificial Neural Network*
API · *Application Programming Interface*

B

BERT · *Bidirectional Encoder Representations from Transformers*
BOW · *Bag of Word*
Bi-LSTM. *Bidirectional Long Short-Term Memory*

C

CNN · *Convolutional Neural Network*
CSS · *Cascading Style sheet*

D

DNN · *Deep Neural Network*

G

GUI · Graphical User Interface

H

HTML · *Hyper Text Markup Language*
HTTP · *Hyper Text Transfer Protocol*

I

IPA · *International Phonetic Alphabet*

J

JSON · *Javascript Object Notation*

K

KNN · *K-Nearest Neighbor*

M

ML · *Machine Learning*

N

NLG · *Natural Language Generation*
NLP · *Natural Language Processing*
NLU · *Natural Language Understanding*

R

RNN · *Recurrent Neural Network*

S

SNS · *social network service*
SVM · *Support Vector Machine*

T

TF-IDF · *Frequency–Inverse Document Frequency*

X

XML · *eXtensible Markup Language*

ABSTRACT

A chatbot is an application that is used to make conversation with a computer by using natural language. This study aims to develop a bilingual chatbot that supports both Amharic and Afaan Oromo languages for assisting Ethio telecom customers on customer services. Ethio-telecom is an Ethiopian telecommunication company serving as the major internet and telephone service provider. In this study, the dataset was prepared by using both Amharic and Afaan Oromo languages in the form of JSON by filtering the frequently asked questions from the organization website. For the model design purpose, the tflearn DNN and Bi-LSTM model has applied. For the proposed system development purpose, we compared those models by using accuracy metrics to select the best. Based on this, the accuracy of the DNN model was 83.52% and Bi-LSTM model was 85.23% *before applying the regularization technique. To overcome the model overfitting challenge L2 regularization technique has applied. After regularization, the model has evaluated by using Accuracy, Precision, Recall, and F1-score and got 82.6, 85.7, 82.6, and 87.7 respectively. In addition to metrics evaluation, the model was evaluated by using human evaluation. This evaluation was used to identify two criteria; that is system performance and user acceptance. The model performance was evaluated based on the number of queries they provide for the proposed system and the number of responses that returned from the system. The second evaluation was users' acceptability of the proposed system. This evaluation was evaluated based on five parameters those are attractiveness, response time, user-friendly, efficiency, and system feasibility.*

Key Words: *DNN, Bi-LSTM, L2 regularization, JSON, Bilingual chatbot, Model overfitting, Afaan Oromo language, Amharic language*

CHAPTER ONE

1. INTRODUCTION

1.1 Background of the Study

Emperor Menelik II introduced a telecommunications service in Ethiopia in 1894 when the construction of the telephone line from Harar to the capital city, Addis Ababa, was commenced(Ethiotelecom, n.d.). Ethio-telecom is an Ethiopian telecommunication company serving as the major internet and telephone service provider. It is owned by the Ethiopian government and maintains a monopoly over all telecommunication services in Ethiopia(Abdu, 2018). Currently, the organization provides different services related to the internet and telephone for the customers.

Sometimes the customer needs assistant regarding service operations. To facilitate this, the organization has set up 994 to provide free telephone service to communicate with customers in their favorite languages. However, this 994-call service is having many problems providing as much as it needs to because the number of customers that call to 994 is high. This makes it difficult for the customers to contact the call center staff because the line is busy. To solve this problem, this study aimed to develop a bilingual chatbot. A bilingual chatbot is an application that is used to make conversation by using two languages.

Language is media that is used for communication purposes between two or more people. In our world, there are many languages useful for communication purposes. The existing system of the organization supports five languages such as Amharic, Afaan Oromo, English, Tigrigna, and Sumali. The proposed system of this study aimed to assist the organization's customers by using Afaan Oromo and Amharic. Those language was selected based on the number of calls that have declined after calling on the 994-call center.

Currently, the role of information technology is increasing from time to time. This technology also makes a platform to understand computer-human language through natural language processing. Natural language processing (NL) is an AI method of communicating with intelligent systems using a natural language(TutorialsPoint, 2020). Processing of natural language is required when an intelligent system like a robot to perform different activities based on our natural language instructions. NLP researchers deal with analyzing, understanding, and generating the languages that humans use naturally to interface with computers in both written and spoken contexts using human languages instead of computer languages. Currently, AI builds a technology that allows us to interact with machines as in normal human-to-human conversation. For this, the role of natural language

processing is great. NLP was applicable in different business areas like Sentiment Analysis, Chatbots, Customer Service, Managing the Advertisement Funnel, Market Intelligence.

The chatbot is a software application that is used for voice or text-based conversations between humans and computers. According to(Chung & Park, 2019), a chatbot is an intelligent interactive platform that can interact with users through a chatting interface. Since it can connect with the major social network service (SNS) messengers, general users can access the platform easily and receive a variety of services. In the current era, the Chatbot plays a great role in Healthcare, Insurance, Banking, Financial service, etc. Based on the development approach there are two types of chatbot. Those are rule-based and AI-based chatbots.

According to(Siangchin & Samanchuen, 2019) Rule-Based approach is a developmental approach of the chatbot to understand the users' queries it follows the rule that the developers are building. The development of this chatbot needs to cover all related conditions that have related to the users' queries. If not, the chatbot will not understand the conversation of the users'. That means Rule-based chatbots are upfront. They have been provided with a database of responses and given a set of rules that help them match out an appropriate response from the provided database. These types of chatbots cannot generate their answers; however, with an extensive database of answers and smartly designed rules, they can be very productive and useful. These bots are extremely limited and can only respond to queries if they are an exact match with the inputs defined in their database.

With the rise in the use of machine learning in recent years, a new approach to building chatbots has emerged. By applying different machine learning techniques, it has become to create extremely intuitive and precise chatbots tailored to specific purposes. Unlike rule-based, AI-based chatbots have developed based on machine learning model that enable them to self-learn. An AI-Based approach is a developmental approach to chatting that obtained intent and context to facilitate the chatbot(Siangchin & Samanchuen, 2019). It requires NLP to analyze the words for a more clear understanding of what the user exactly wants to communicate.

Many researchers have reported work research on Chatbot for different purposes and in different natural languages. According to (Khare, 2019) many industry analysts predict that chatbot are an emergent trend, which will offer real-time solutions for simple customer service assistance problems. They are useful in providing standard solutions to common problems in different areas. Our aim also to design and develop a Bilingual (Amharic and Afaan Oromo) Contextual AI-based Chatbot for

assisting Ethio-telecom customers regarding customer services by using machine-learning algorithms.

1.2 Motivation of the Study

The main motivations that make us study this thesis have discussed as follows

The first initiative is while we are reading various articles we understood the benefits of the chatbot in terms of its role in solving real-world customer assistance problems. There are several papers regarding the development of Chatbots for different purposes as well as evaluation or comparison of specific bots that have developed by different languages and for different purposes. After performing the initial survey and the study of related articles, we formulated the research gap and we initiate to fulfill those gaps. In addition to this, currently, multiple scholars have developed a chatbot that supports foreign Multilanguage(Vanjani et al., 2019)(Bisht et al., 2020). From those papers, we analyzed the feasibility of multilingual chatbot has more advantages than single language chatbot. However, as far as our knowledge in our country Ethiopia, there is no research has been conducted before on development of a chatbot that supports more than on language. This makes us study chatbot development by using more than one languages.

The second initiative is Ethio Telecom's problem with current customer assisting techniques regarding customer service. The organization is giving a lot of service to its customers and the customers are getting different services from this organization. However, sometimes the customer needs some help regarding service operations. To facilitate this, the company has set up 994 to provide free telephone service. Nevertheless, this service is having many problems to provide as much as it needs to, and the number of customers is one of them and it is very difficult to contact the call center staff. This initiates us to develop a bilingual chatbot for this organization.

1.3 Statement of the Problem

Ethio-telecom is an Ethiopian telecommunication company serving as the major internet and telephone service provider. Ethio telecom is owned by the Ethiopian government and maintains a monopoly over all telecommunication services in Ethiopia(Abdu, 2018). The organization is giving a lot of service to its customers and the customers are getting different services from this organization. However, sometimes the customer needs some help regarding service operations. To facilitate this, the company has set up 994 to provide free telephone service. Currently, on this free telephone service, the organization has 1205 advisers or customer assistants.

The 994-telephone service gives consulting programs in five languages such us; Amharic, Afaan Oromo, English, Tigrigna, and Somalia. According to the report that given to us shows the

Organization was handled 40%- 45% per day only. The other call around 55%-60 % calls is dropped. For this problem, the main reasons are the number of customers makes it difficult for the customers to contact the call center staff because the line is busy. The other problem is even if the call center is not busy the Ethio-telecom staff members that provide help for the customer through the free dial number are tired because they may help or talks to many people to assist. Those problems produce complaints between the customer and the organizations. This means the customer may not satisfy with the service that gives through the 994-dial number. For example, the following graph shows statistical success calls to get assistant on five consecutive dates from 03/10/2021 to 03/11/2021. This report data is prepared by the system of the organization.

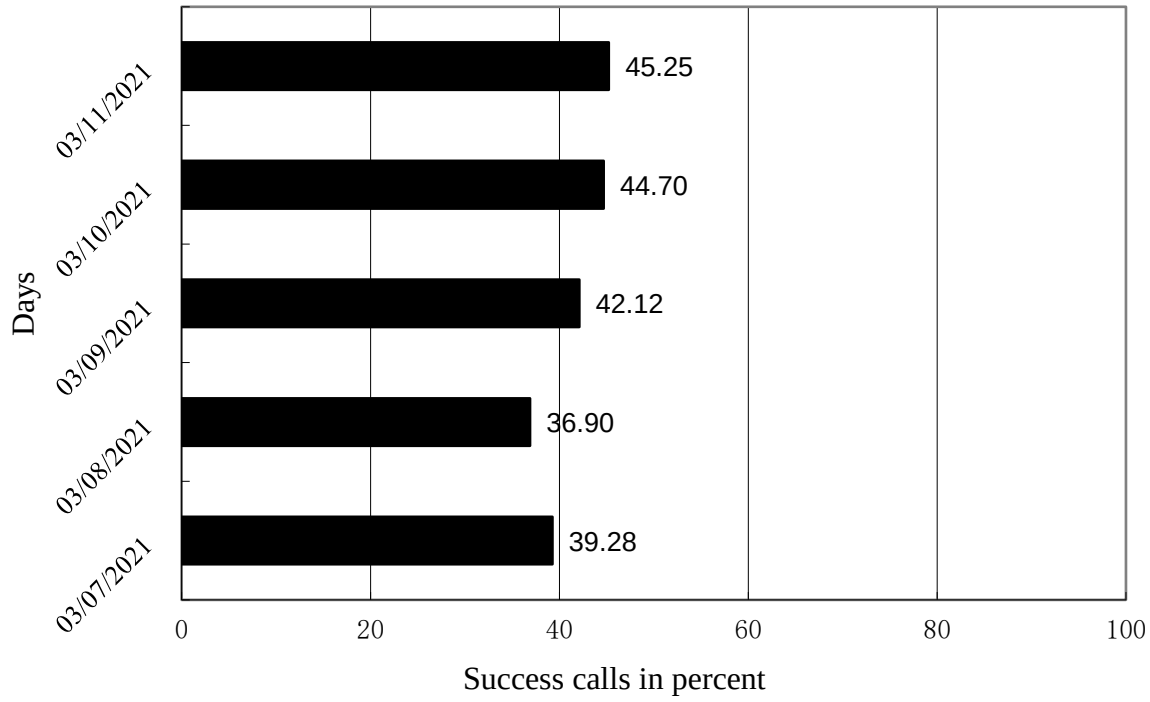


Figure 1. 1 Statistical report on 994 call center

From the above graph, the highest success calls are 45.25%; this indicates that the other calls have declined because of the above reasons. Therefore, this study aims to provide a way to assist customers regarding Ethio-telecom service in addition to free dial numbers by using a bilingual chatbot to advise or consult Ethio telecom customers. To solve this problem this study propose the chatbot for the customer consultancy purpose by using Amharic and Afaan Oromo text. Those languages are selected based on the statistical number of customers who are calling 994 call centers

many times per day. The following report was prepared from 03/07/2021 to 03/12/2021 by Ethio-telecom.

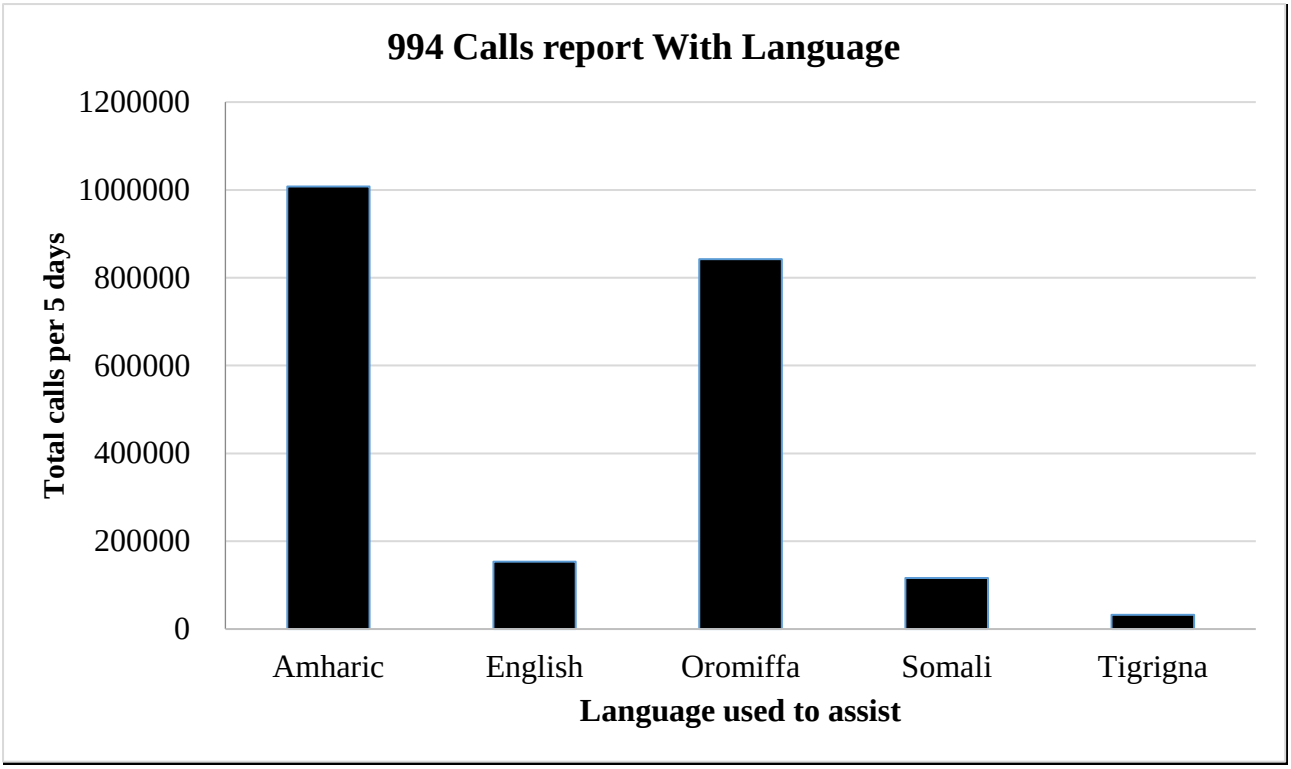


Figure 1. 2 Statistical reports on 994-call center based the language

Based on the above graph, the number of customers that call on 994-call center per day by using Amharic and Afaan Oromo language is high when we compare with the other three languages. As the number of customers calls to the 994, call center increase the line comes busy. Therefore, to minimize this load this study aims to develop a chatbot that supports both the Amharic and Afaan Oromo languages.

In addition to this different scholars have been trying to develop a chatbot for assisting the users by using deep learning techniques, specifically for assisting farmers in the case of maize productivity by using Afaan Oromo language(Bedasa, 2021) and assisting peoples in the case of Ethiopian constitutional law by using the Amharic language(Habitamu Asimare, 2020). However, as far as our knowledge, the study of developing a bilingual chatbot is not conducted yet, indeed which is a reason to have no dataset that related Ethio telecom service to work with it.

1.4 Research Questions

This research will set the following research question to examine the problem in developing the Bilingual chatbot.

- **RQ1:** How do we design the model that understands the user's dialogue efficiently and produces an appropriate answer (response) for the users by using Amharic and Afaan Oromo languages?
- **RQ2:** How we develop a user-friendly text-to-text chat interface to engage the deigned model with the Ethio telecom customers?
- **RQ3:** How we test and evaluate the proposed model?

1.5 Objectives

1.5.1 General Objective

The general objective of this research is to design and develop a Bilingual Chatbot to assist Ethio-Telecom Customers on Customer Service by Using Deep Learning Techniques.

1.5.2 Specific Objectives

To achieve the general objective, the following specific objectives have been performed.

- To design the architecture and model for developing a bilingual chatbot for assisting Ethio telecom customers.
- To tune the parameters of the designed model.
- To train the model by using the prepared data.
- To develop a prototype of the system to integrate with the designed model.
- To test and evaluate the proposed model.

1.6 Scope and Limitations

a. Scope

The scope of this research is developing the chatbot for assisting Ethio-telecom customers regarding customer service. The proposed system must handle the customer text-based Amharic and Afaan Oromo conversation and give a response to the user based on the user's request context.

b. Limitation

Due to some reasons like getting enough time to prepare more datasets and getting permission to access the Ethio-telecom database, this study has a limitation. The first limitation is this chatbot is only performed text-to-text conversations. The second limitation of this study is the chatbot has not considered a conversation that has not related to Ethio-telecom customers' services. The third limitation of the study is the chatbot does not access the Ethio-telecom database to check different information for the customer but it tells what the customer must have to do based on their request or this proposed system was assist the customer without checking up the customers' information.

1.7 Significance of the Study

This study is applicable in assisting Ethio-telecom customers by using the chatbot. From this study, there are many beneficiaries. From those beneficiaries, the Ethio-telecom organization is the first beneficiary of this chatbot by reducing the number of customers that dial to free teleservice number 994 and help their customers to assist them. This makes their customers satisfied with the organization's service. The second beneficiary of this study is the Ethio-telecom customers. When the Ethio-telecom customers need help in the case of telecom service, they used the chatbot and get help regarding customer services. The third beneficiary of this study is the Ethio-telecom service staff. Those staff members are a person that gives help the Ethio-telecom customer through the 994-call center. So if the users are used the chatbot may minimize the load of those employees.

1.8 Organization of the Thesis

This study has a different section that has been classified by chapter. In this phase, we try to explain the content of each chapter under this study report.

The first chapter presents the background of the study, motivation for this study, objectives of the study, statement of the problem, questions that were raised under this study, scope, and limitations of this study, and application of the study.

The second chapter discusses literature reviews and related works. Under this section, we tried to address an overview of a chatbot, history of a chatbot, types of a chatbot, application area of a chatbot, frameworks that used to develop a chatbot, overview of machine learning in chatbot development, an overview of a multilingual chatbot, Amharic language, Afaan Oromo language, and related works.

Chapter three deals with the research methodology and tools used for developing the proposed system. This part has a different section. Those are data collection methodology, dataset preparation

methodology, data preprocessing techniques, feature extraction techniques, model selection techniques, system evaluation techniques, techniques used to design the user interface and simulation tools.

The fourth chapter also contains different sections. Those are proposed system prototype architecture, implementation flow chart, proposed system model architecture, data preprocessing techniques implementation, feature extraction implementation, model implementation, and response selection implementation

In the fifth chapter, we discussed the result that we get from the proposed system development. It has a different phase. From those, the first phase deals with the evaluation result that we got from the proposed model. These evaluation results have based on our dataset and based on human evaluation. The second phase is the discussion of about how the proposed system answers the research question and discusses what makes it different from the previous work on chatbot development.

The last chapter has discussed on three different points. The first point is the conclusion of the general roles that have been performed in this study and the result that got from this study. The second point is the recommendation for different parts based on this study. The last point of this chapter is the plan of the researcher that will have added for the future of this study.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORK

2.1 Chapter Overview

This thesis aims to design and develop a bilingual chatbot for assisting Ethio-telecom customers. We organized this chapter in different parts as the following. The first phase was an overview of the chatbot and AI on solving the problem. The second phase deals with the history of the chatbot and the currently available framework that has been used for developing the chatbot. The third phase deals with research that has been done before on developing chatbots and their results.

2.2 An Overview of Chatbot Technology

In the current era, the application of AI is increasing from time to time. AI was applied to solve small to large problems. From those problems, one is assisting and adverting organization products for customers. It means that when one organization deploys a new service or product the customers may require help regarding the new service. This may need human labor to assist the customers manually. However, by using AI technology was solve this problem through the chatbot.

Chatbots is a computer program that is used to simulate conversation with human language by using either text or sound 24/7 365(24 hours per day, 7 days per week, and 365 days per year). It is an AI application that has been used for human-computer interaction through natural language. When users interact with a computer, the users was communicate by using natural languages like Amharic, Afaan Oromo, English, etc. However, the computer cannot understand natural language. For this purpose, the role of natural language processing is essential in chatbot development.

NLP has played a great role in a chatbot for the case of human-computer interaction. It is a subfield of AI, which is used for enabling the computer to and process human language. For this purpose, this study was applied different techniques like tokenization the process of breaking up a stream of characters into words, punctuation marks, numbers, and other discrete items(Kibble, 2013). In tokenization, the sentence **“ደንበኞቻችንን ማርካት የመጀመሪያ አላማችን ነው።”** was tokenized as ‘ደንበኞቻችንን’, ‘ማርካት’, ‘የመጀመሪያ’, ‘አላማችን’, ‘ነው’, ‘።’. In Afaan Oromo this sentence has the meaning “**Kaayoon keenya maamiltoota keenya gammachiisuudha**” also tokenized as ‘Kaayoon’, ‘keenya’, ‘maamiltoota’, ‘keenya’, ‘gammachiisuudha’, ‘.’. After splitting the sentence, the other preprocessing techniques like removing different punctuation marks, stemming, removing stop words. These all things make the computer understand the natural language easily.

In the chatbot application, Communication via a natural language requires two fundamental skills, producing text and understanding it. That has known as natural language understanding and natural language generation. Natural language understanding is the process of making computers read the text and understand the intent or the meaning of the text like a human(Navigli, 2018). After understanding the text sent from the user the computer generates the response for the user by understanding the context of the users' text this process is known as natural language generation(McDonald, 2010).

2.3 History of Chatbot

Recently the application of chatbots in different sectors is increasing. In the public sector, the chatbot was used for political purposes to inspire public opinion and intervene in any discussion in social media about politics(Almansor & Hussain, 2020). In addition, the chatbot was used for organization product and service advertisement and assisting customers regarding services. In this section of our thesis, this is the survey some of the existing chatbot and their applications.

To start with, the history of the chatbot, MIT professor Joseph Weizenbaum ever developed the first chatbot **ELIZA** in the 1960s. The program has been designed in a way that mimics human conversation. The Chatbot ELIZA worked bypassing the words that users entered into a computer and then pairing them to a list of possible scripted responses(Weizenbaum, 1966). It uses a script that simulated a psychotherapist. After that, the application was come advanced to simplify the life of humans. The ELIZA was currently available on the link https://web.njit.edu/~ronkowit/eliza.html. The following image shows the user conversation with ELIZA.

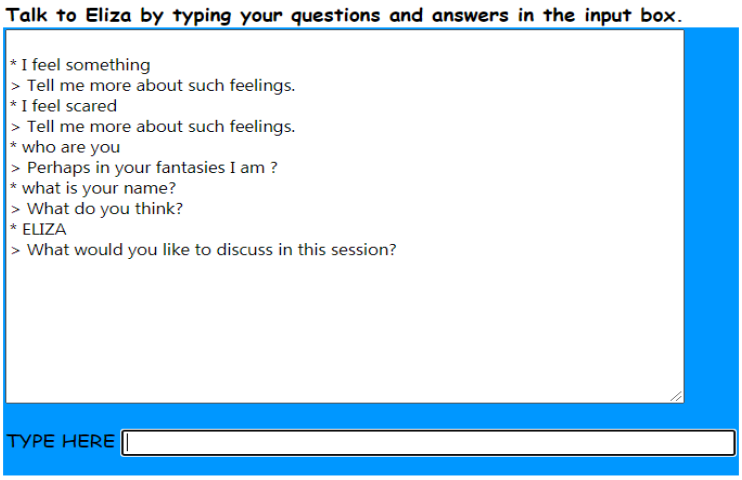


Figure 2. 1 ELIZA chatbot history

After ELIZA in Artificial Linguistic Internet Computer Entity(**A.L.I.C.E**), originated by Wallace in 1995(Shawar & Atwell, 2015). A.L.I.C.E utilizes the AIML, an XML language designed for creating

a stimulus-response chat robot. The main difference between ELIZA and A.L.I.C.E is, the A.L.I.C.E bot has at present more than 40,000 categories of knowledge, whereas the original ELIZA had only 200(Wallace, 2009). The following minimalist snapshot shows the AIML implementation for chatbot development.

<category> <pattern> እንዴት ነህ? </pattern> <template> ደህና ነኝ አንተ እንዴት ነህ </template> </category>	<category> <pattern> Akkam Jiirta? </pattern> <template> Jira Nagaa koodha. </template> </category>

In the above snapshot code

If user inputs: እንዴት ነህ?

Bot: ደህና ነኝ አንተ እንዴት ነህ

If user inputs: Akkam Jiirta?

Bot: Jira Nagaa koodha

In 2006, Thomas J.Watson develop the **IBM Watson** chatbot. Watson is a question-answering computer system capable of answering questions posed in natural language(Consulting, 2017). Watson has created for question answering to apply advanced natural language processing, information retrieval, knowledge representation, automated reasoning, and machine learning technologies to the field of open domain question answering.

In 2010 Apple company developed an artificial personal assistant **SIRI** that uses natural language(Gruber, 2009). Siri is a faster, easier way to do all kinds of useful things like Set alarms, timers, and reminders, Get directions, Preview calendars. Not only that, Siri can get the song for the user who wants to listen. Apple Music and Siri work together beautifully which means the users can ask Siri to find a new song that the users like, Put on a favorite album, or even answer questions about bands users love. Siri can also suggest their favorite playlist when the users arrive at the gym or start our drive home.

In 2014, Amazon Company developed an intelligent personal assistant **Alexa**. According to(Hard, 2016), the Alexa bot knows 50,000 words and able to apply them to a few hundred circumstances.

Alexa can now deliver the answer, in response to a query about a Whirlpool product for example, via software that mind touch has written that goes and fetches content based on the sentence the customer used(Hard, 2016).

The other chatbot that has developed for assisting customers is **Cortana**. Cortana is a virtual assistant developed by Microsoft which uses the Bing search engine to perform tasks such as setting reminders and give answers to Microsoft customers(Almansor & Hussain, 2020). The development of Cortana has started in 2009 by the Microsoft Speech products team with general manager Zig Serafin and Chief Scientist Larry He. Cortana recognizes natural voice without the requirement for keyboard input and answers questions using information from the Bing search engine.

2.4 Types of Chatbot

The main goal of the chatbot was to receive the users' request and understand the intent of the request after that it responds to the users'. To develop the chatbot there are two approaches(Thorat & Jadhav, 2020). Those are Rule-based Chatbots and Self-learning Chatbots.

In a Rule-based Chatbot, the system was handling the users' requests base on certain rules. However, if the users' input pattern does not match with any predefined rule then this Chatbot system is inefficient to answer the question(Thorat & Jadhav, 2020). To develop this type of chatbot the developer writes the rules by using AIML. This rule-based Chatbot system can deal with straightforward questions but if the request does not match the pattern this type of chatbot may return an incorrect response.

The second type of chatbot is Self-learning Chatbot. In this type of chatbot, the machine learning techniques make it learning things. There are two approaches to self-learning chatbot development(Thorat & Jadhav, 2020). That is a retrieval-based chatbot and a generative-based chatbot. In a generative-based chatbot, the chatbot generates new dialogue based on large amounts of conversational training data. It uses a combination of supervised learning, unsupervised learning, reinforcement learning, and adversarial learning for multi-step training. The other type of self-learning chatbot is a retrieval chatbot. A retrieval-based chatbot has trained to provide the best possible response from a database of predefined responses. It uses different techniques like keyword matching, machine learning, or deep learning to identify the most appropriate response(Fainchtein, 2020). The following figure shows types of chatbots based on the developmental approach.

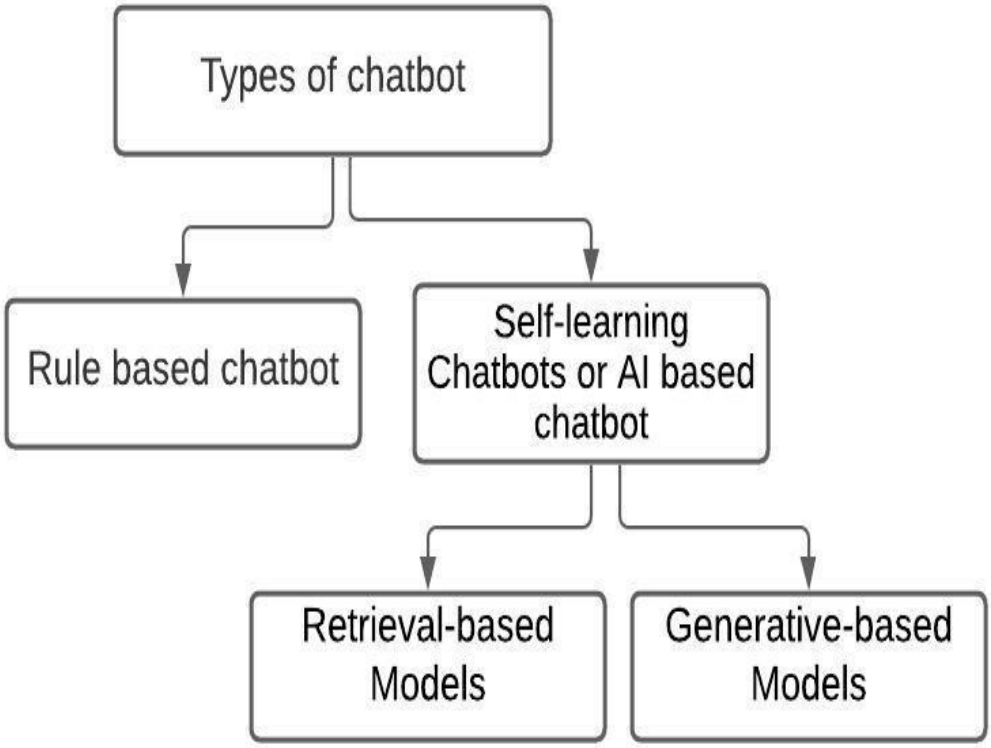


Figure 2. 2 Classification of chatbot

The types of chatbots that discussed on the above section, the proposed system have categorized under retrieval-based chatbot. Because, to develop the chatbot the corpus was prepared with the question and answer with both Amharic and Afaan Oromo languages. The chatbot will return the response for the users from the corpus that prepared based on the users' query intent.

2.5 Application Area of Chatbot

Currently, the chatbot applicable in a different area. This section was discussed about some application areas of chatbot and research work done before on different application areas like health, education, and customer service. Those areas was selected because of there are many published papers before.

2.5.1 Application of Chatbot in Health

Health is one of the necessary things for human beings. To keep our health we may follow different activities like types of food that we take, activities that we did for keeping our health well and checking our health status by discussing with a doctor or other health expert. However, keeping our health by discussing with a health expert is not much simple. Because the world's population is

growing rapidly. This may have a different effect like a shortage of getting an expert easily. For this problem, the application of the chatbot is great for diagnosis patient health and for consulting people on their health status. This section was reviewed some research done before on the application of chatbots on health.

In(Shangrapawar et al., 2020), the researcher was studying an application of an artificial intelligence-based healthcare chatbot system. In this study, the researcher was trying to connect the android phone with the robot through the Bluetooth HC-05 with a Smartphone app to control the movement of the robot. The robot has received sound through Bluetooth and diagnosis the patient like the doctor. The Bluetooth connection on the audio board receives text from the android app as characters and stores them as a string to the assigned string. The system took the disease from the database based on the patient's symptoms. To do this the researcher has connected the system with a google server.

The other study of an application of chatbot on health is Pharmabot: A Pediatric Generic Medicine Consultant Chatbot(Comendador et al., 2015). In this study, the research aim is to develop a chatbot to suggest and give information on generic medicines for children. The advantage of this application is consulting the patient or family of children, which confused about generic medicine. The researchers have developed the system by using Visual C# as its front end and MS Access as its back end.

2.5.2 Application of Chatbot in Education

The chatbot was also applicable for educational purposes. In the educational system, the student may need support from their lecturer. However, supporting a student individually is not easy. In recent, many researchers studied the application of chatbots in the education sector to solve such problems. This section was tried to review the researches that have related to the application of chatbots in education.

The chatbot has used for supporting the students in different courses in the teaching and learning process. In the learning and teaching process, the student may have a question that did not clear for them. Some student may have the problem to ask the question that did not clear for them because of different reasons. From this problem, one is afraid of their teacher to ask a question. For this purpose, the chatbot may play a great role. The student asks the question and the bot was answering the student's question. According to(Colace et al., 2018), the researcher tries to develop a chatbot for assisting the students in different courses. In this chatbot system, they focused on the course of Fundamentals of Computer Science and Computer Networks. The student asked the question on those courses and the chatbot has answered the question. To develop this chatbot the researchers

used AIML. The chatbot provides easy navigation and interaction within the content to assist the student.

The other research that focused on applying the chatbot in education is studied on Chatbot for education system(Hiremath et al., 2020). This study aims to develop an automated system that replies to the student question efficiently. The system assisted the student on academic and on registration or university calendar. To develop the system the researcher was used multi logic adapter. The logic adapter has used to generate responses according to the users' requests. In a multi-logic adapter, the system returned multiple responses with the highest confidence. The system compares the occurrence of the response and selects the highest one.

2.5.3 Application of Chatbot in Customer Service

Customer service plays an important role in for one organization to generate income and revenue(Cui et al., 2017). The organization may give different services to their customer. The customer may need assistant regarding the organization's service. To assist the customer manually may need human power, budget, and time. Currently, different organizations have used a chatbot for assisting their customers. This section tried to review different thesis papers done before for assisting a customer on customer service.

Different organizations may develop new services for their customer. From that one is banking system is one of the organizations that contains multiple services and customers. The customers may need some help or explanation to know about the services. To give such an explanation they may use different techniques. From that, the first technique is developing a chatbot that gives help or explanation to the customers. In(Zhang et al., 2020), the researchers aimed to develop personalized digital customer services for consumer banking call centers using neural networks. The researcher has developed the chatbot by using neural networks for assisting the customer with customer service services. The dataset has retrieved from the bank dialogue transcript between the customers and Customer Service Officers (CSOs) to identify the customers' reasons for calling.

According to (Chaitrali et al., 2017), Developing a chatbot will provide a smart solution to solve these queries, provide information as and when required, improve service, and increase the number of customers. The aim of this study is customer can interact with mentioning their queries in plain English and then the chatbot retrieves the users' queries and provide the response. On this chatbot, the researcher has considered if the bot does not answer the users' queries or the users are not satisfied with the answer the user can touch the dislike button. This makes the system train again and answers the queries correctly for the next time.

The other chatbot developed to assisting the bank customer regarding customer service is studied in(Chaitrali et al., 2017). This paper aims to assist the bank customer with customer service. The researcher used AIML for developing the chatbot application and JavaScript to integrate with the web. They also developed the android app for the user to interact with the system. The system also supports voice input.

2.6 Frameworks Used for Developing Chatbot Application

The framework is a platform or a library that used for developing a system. It may contain code libraries, a compiler, and other programs used in the software development process. The framework is used an application programming interface(API), which we used by importing into our project development(*Framework Definition*, n.d.). It may contain different classes and methods that have been used for performing different tasks. Currently, there are different types of the framework are available for chatbot development. Some are IBM Watson, Microsoft Bot Framework, Dialog flow, Chatterbot, etc. This section was tried to discuss some of the frameworks that are available for chatbot development.

Microsoft company is introduced its bot framework in early 2016(Machiraju & Modi, 2018). This framework contains SDK, which provides the resources a developer needs to build an intelligent conversational chatbot that interacts naturally. This framework supported 30 different languages. This makes it more feasible for developing a chatbot by using the Microsoft bot framework. This framework documentation was available on https://cloud.google.com/dialogflow/docs/ link.

Dialogflow is one of the frameworks used for chatbot development. Dialogflow was Google's chatbot platform has used for creating conversational AI systems that work on natural language processing. Dialog flow has Machine Learning as its backbone implementation(*A Step-by-Step Guide to Building a Basic Dialogflow Chatbot | PDF.Co*, n.d.). This framework is used for developing a chatbot for a different purpose. This framework contains chatbots, and voice bots, trained in 14 different languages. The Dialog flow framework is available on https://dialogflow.com/ link.

The other framework that is used to develop chatbots easily is Chatterbot. Chatterbot is a Python library that makes it easy to generate automated responses to user input. Chatterbot uses a selection of machine learning algorithms to produce different types of responses(Cox, 2019). Before using this library, installing the library by using pip install chatterbot commands was necessary. This library contains different classes and methods that are uses for performing different tasks. To use

this library we import those classes and methods in our project and we develop chatbot easily. We used this library like the following sample code.

```
# Import ListTrainer
from chatterbot.trainers import ListTrainer
trainer = ListTrainer(bot)
trainer.train([
'ሰላም ነው?',
'አለሁ እንዴት ኖት',
'እርዳታ ፈልጌ ነበር',
'ምን ልርዳዎት',
'Akkam Jiirta?',
'Nagaadha maal barbaadan?',
'Gargaarsa barbaaden ture?',
'Maal sii gargaaru?',
:
:
])
```

Code 2. 1 Chatbot implementation using chatterbot library

After the train, the users and bot chat as the following

User: ሰላም ነው?

Bot: አለሁ እንዴት ኖት

User: Akkam Jiirta?

Bot: Nagaadha maal barbaadan?

2.7 Overview of Machine Learning in Chatbot Development

Machine learning is a technique that is used to build a model to solve some problem based on a dataset known as train data. Currently, machine-learning algorithms have been applied in different areas such as NLP, computer vision, etc. There are different types of techniques in machine learning that used to solve a different existing problems. From the existing problem, assisting an organization customer or assisting people in a different area is one. A chatbot is one machine learning application that is used to solve this problem.

The chatbot is one NLP area that is used to make a computer understand human language and produce the response based on the users' queries by using different machine learning techniques. From those techniques, some are DNN, RNN, and ANN are some examples of deep learning techniques that have been used to develop a chatbot. Deep learning is more efficient than machine learning techniques in chatbot development because chatbot development has multiclass classification. The advantage of deep learning in natural language understanding is it makes the computer could be easily understood the conversation in a natural language and provide a more

accurate response. When we develop the chatbot the target class of our prediction is the intention of the user's or the idea of the user queries taken as the target classes. The intents or class prediction has based on the service that the system assists the customers. Therefore, it may contain more than 100 intents, which means it contains more than 100 target classes. Achieving this prediction by machine language is somewhat it is difficult. For this reason, many researchers have applied the deep learning algorithm for chatbot development.

When we used a deep learning algorithm, we have to convert the text data into equivalent numeric values. This type of preprocessing technique has called word embedding. Word embedding is a technique that is used to convert the text into an equivalent vector [32]. We convert the data from text to vector because when we use deep learning or machine learning it understand or work with a numeric value. There are different techniques used for converting text data into numeric such as TF-IDF, Bag of words (BOW), word2vec, etc. After we convert the text to an equivalent vector, we design our model by using a deep learning algorithm.

Deep learning is a subset of machine learning that perform a complex operation at different layer to predict the output. It contains three layers like input layer, hidden layer, and output layer. In the proposed model design, we specified each layer based on the dataset that we prepared and the result that we get from the model.

The input layers have used to provides information from the outside world and feed the hidden layers. When we use deep learning the data must have converted from text data into numeric data. In NLP, there are different types of techniques used to convert text data into numeric data. The converted numeric data of the corpus is used as an input layer for deep learning. The conversion of text data into an equivalent numeric vector known as word embedding. In the proposed system, we specified the number of neurons at the input layer based on the length of the vector that we converted the text data to an equivalent vector.

The hidden layer is used to take an input from the input layer, perform some operations, and transfer it to an output layer. When we used deep learning, we may have one or more hidden layers. There is no way to specify the number of hidden layers for our model. To specify the final number of the hidden layer based on the best result that we get from the model.

After the hidden layer, the last layer is an output layer, which is used to compute the result of the output. The main purpose of the output layer is to take in the inputs which are passed in from the layers before it, perform the calculations via its neurons, and then the output is computed. In chatbot

development, the number of neurons in the output layer is equivalent to the target class of the user's intents.

2.8 Overview of Multilingual Chatbot

The main advantage of using chatbot technology is making computers make conversation with a human through natural language. That natural language may Amharic Afaan Oromo, English, or others. The chatbot that supports more than one language is known as a multilingual chatbot. A multi-lingual chatbot has many advantages over a single language chatbot. From those advantages, some are; make Customers communicate with their native language, support a wider audience, increase the feasibility of the Chabot, save money and time. In this section, we tried to review some research papers that work on developing a multi-lingual chatbot.

In M. Vanjani et al(Vanjani et al., 2019), the researchers have developed chatbots for multilingual conversations. In this study, they developed a chatbot that supports one hundred three languages. To develop the chatbot the researcher was used google translation. That means first they prepared the corpus for the English language and train it by using machine-learning techniques. After they train the model the chatbot was retrieved the user's request and prepare a response for the users. To do this the system has retrieved requests by 103 languages and first it check either the language is English or not. If the user's request is English it gives the response by predicting according to the corpus. However, if the language is not English first it translates it into English and predicts the response by English, and again translates it into the user's request language.

The other study that is related to a multilingual chatbot is A. Bisht et al (Bisht et al., 2020) developed a multilingual chatbot with human conversational ability. The researchers aim to build a chatbot that will be like a virtual assistant. This chatbot makes conversation with humans to assist them. In this study, the researcher has developed a multilingual chatbot by using google translation. That means they prepared only an English corpus that has the conversation and trains the model by using this corpus. After they trained the model, the chatbot received the request from the user and respond to them. The chatbot was responding in two ways. The first one is if the user request is English the chatbot was giving a response directly another wise first translate the user request into English and prepare the response in the English language and finally translate it again into the user's request language and gives the response.

Our study aims to develop a bilingual chatbot for assisting Ethio telecom customers by using the Amharic and Afaan Oromo languages. In this study, we developed the chatbot that retrieves user’s requests by using Afaan Oromo or Amharic language and gives the responses based on the user’s

requests. I.e. if the users send the request by using the Amharic language, the proposed system will return the response by using Amharic language and if the user's request will be sent by Afaan Oromo the chatbot gives the response by Afaan Oromo language. As we discussed above the researcher was used google translator to develop a multilingual chatbot. However, we are not used google translator to develop the proposed system because google translator cannot support the Afaan Oromo language. Because of this, to develop the proposed system we prepared the corpus that contains both Afaan Oromo and Amharic language conversations.

2.9 Amharic Language

The proposed system can support the Amharic language. In this section, we tried to address the language and review some papers that have been done by using this language related to our research area. Amharic (also known as Abyssinian, Amarinya, and Amarigna) is the national language of Ethiopia. Since the 13th century has been the language of the court and dominant population in Highland Ethiopia. According to the 1994 census Amharic is the most commonly spoken language in Ethiopia(Wimsatt & Wynn, 2011). The upper and middle-class Ethiopians predominately speak the Amharic language.

2.9.1 Amharic Writing System

Amharic is the Semitic Language of the Afro-Asiatic language group that is related to Hebrew, Arabic, and Syrian(Mindaye & Atnafu, 2009). From the Semitic language group, Amharic is the second most spoken language next to Arabic in the world(Kelemework, 2013). This language has its writing system and has thirty-three (33) different symbols with seven (7) variations. It has also different symbols that contain the same sound. The following tables show three core Amharic characters with their six orders.

Table 2. 1 Amharic character with their seven orders

1st Order	2nd Order	3rd Order	4th Order	5th Order	6th Order	7th Order
ሀ Hä	ሁ Hu	ሂ Hi	ሃ Ha	ሄ He	ህ H	ሆ Ho
ለ Lä	ሉ Lu	ሊ Li	ላ La	ሌ Le	ል L	ሎ Lo
ሐ Hä	ሑ Hu	ሒ Hi	ሓ Ha	ሔ He	ሕ H	ሖ Ho
:	:	:	:	:	:	:

2.9.2 Challenges of Amharic Language in NLP

There are some problems associated with the Amharic language to apply natural language processing(Kelemework, 2013). From those problems, the first one as the Amharic language uses different symbols that have the same sound. This has challenges in the process of feature preparation

for classifier learning since the same word has represented in different forms. For example, "ስልኬ" which means my phone also written as "ሥልኬ". The second challenge is, In the Amharic language the users may express their idea by using different expressions. For example, to ask the greeting the users may use "ሰላም","ጤና ይስጥልኝ". In chatbot development, understanding such types of messages have challenges. The third challenge in chatbot development by using Amharic language Amharic is a technologically under-resourced language (Kelemework, 2013). To develop this proposed system it asks some resources like dataset related to our study, and stop words corpus that has prepared by us. The other challenge of Amharic in NLP is a variation of the same word. For example, the word computer may be written as ኮምፒዩተር or ኮምፒውተር.

To handle those challenges we will use a different methodology that has been written in chapter three of this document.

2.10 Afaan Oromo Language

Afaan Oromo is an indigenous Afro-Asiatic language spoken in many parts of Ethiopia and neighboring countries like Kenya, Djibouti, and Somalia, which have a horn of Africa coverage(Abera Hordofa & Dechasa Degefa, 2021). It is one of the Afroasiatic languages belonging to the Cushitic branch. It is native to the Ethiopian state of Oromia and spoken predominantly by the Oromo people and neighboring ethnic groups in the Horn of Africa [3]. The native speakers of the language are the Oromo people, they are the largest ethnic group in Ethiopia, and however, the exact percentage is controversial. It is an official working language of the Oromia regional state and is used as a learning medium in elementary schools and high schools.

2.10.1 Afaan Oromo Writing System

In the 19th century, scholars began writing in the Oromo language using Latin script(Wikipedia, 2020). The writing system of the language is straightforward, which has designed based on the Latin script. Unlike English or other Latin-based languages, there are no skipped or unpronounced sounds /alphabets in the languages. It has 33 which is consists of 26 single characters and 7 compounds, those letters can be separated as five vowels (a, e, i, o,u) and the rest are considered as a consonant, each character representing distinct sounds and it has both capital and small letters. Like, in English, Afaan Oromo has vowels and consonants. Afaan Oromo vowels are represented by five basic letters like a, e, i,o,u. Besides, it has the typical eastern Cushitic set of five short and five long vowels by doubling the five letters: aa, ee, ii, oo, uu. The following table shows Afaan Oromo letters within their International Phonetic Alphabet (IPA).

Table 2. 2 Afaan Oromo letter within their IPA

Qubee(single characters)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	a	b	c	d	e	f	g	h	i	J	k	l	m	n	o
IPA	/a/	/b/	/č'/	/d/	/e/	/f/	/g/	/h/	/i/	/ğ/	/k/	/l/	/m/	/n/	/o/
Qubee(single character)	P	Q	R	S	T	U	V	W	X	Y	Z				
	p	q	r	s	t	u	v	w	x	Y	z				
IPA	/p/	/k'/	/r/	/s/	/t/	/u/	/v/	/w/	/t'/	/y/	/z/				
Qubee(compound character)	CH	DH	NY	PH	SH	TS	ZH								
	ch	dh	ny	ph	sh	ts	zh								
IPA	/č/	/đ/	/ń/	/p'/	/š/	/s'/	/ž/								

2.10.2 Challenges of Afaan Oromo in NLP

However, several researchers studied NLP for the Afaan Oromo language still there are different challenges to develop a chatbot by using the Afaan Oromo language. In this section, we discussed the challenges in NLP by using the Afaan Oromo language. The first challenge is according to(Abera Hordofa & Dechasa Degefa, 2021), Many pieces of research in natural language processing have been done or going on Afaan Oromo, but applications do not available publicly. This makes it under resources to study on NLP, the second challenge is the users have expressed their ideas by using a different form or the users express their idea by using different sentences. In chatbot development, we have to handle the intention of the users according to their expression.

To overcome those challenges we have taken some activities. For the first challenge, we prepared different resources like a list of stop words, the corpus that related to this study by using a different methodology that we discussed in chapter three. To overcome the second challenge we prepared the corpus by considering it.

2.11 Related Works

In recent times, several new frameworks and approaches have been proposed to evaluate the usability of dialogue systems and the motivation behind their use(McAllister et al., 2020). Many researchers have done many things on the chatbot. In this section, we try to review different papers related to chatbot development.

In(Srivastava & Singh, 2020), An automatized medical chatbot is a system with human interaction using natural language diagnosis to provide medical aid. This medical chatbot has developed by using Artificial Intelligence Markup Language (AIML). AIML was developed by the Alicebot free software community and Dr. Richard S. Wallace during 1995- 2000(*Introduction to AIML*, 2018). The author has prepared the chatbot conversation by using a pattern and template tag. The pattern

was the users' input and the template was the response of the bot. The researchers were to get the symptoms of the diseases from users' input and predict the disease. This system is qualified to identify symptoms from user inputs with a standard precision of 65%. After identifying the user's symptoms, the system predicted the diseases by using K-nearest neighbor (KNN), SVM, Naive methods with an accuracy of 0.946, 0.886, and 0.8 respectively. However, even if the system achieves those things, it has limitations on understanding users' input. This means that the machine has understood if the users' input matches with the pattern.

In(Athota et al., 2020), the aim of this paper is chatbots are conversational virtual assistants that automate interactions with the users. This chatbot was developed for assisting users with minor health information. Initially when the users visit the website first register themselves and later can ask the bot, queries. The chatbot stores the information in the database to identify the keywords from the sentences and make a decision for the query and answers the question. In this paper, the machine was giving a response by finding the text similarity. For this purpose, the researcher has used n-gram, TF-IDF, and cosine similarity to find the text similarity between the users' input text and the dataset that the researcher was prepared. The problem in this paper is the machine was compared or calculate the similarity between users' input and each dataset. This makes the time complexity of the system high.

In (Rahman et al., 2019), the research study is focused on developing Machine Learning-Based Bangla Healthcare Chatbot. This chatbot was developed to make conversation with the user through the Bangladesh language. This study aimed to collect the symptoms of the user through Bangladesh text and predict the disease based on those symptoms. To compare the symptoms stored in the knowledgebase and the user input symptoms the authors were used cosine similarity and TF-IDF techniques. After the users' symptoms have been identified, the system will predict the disease based on the dataset. For this purpose, the researcher is used SVM models for disease prediction. However, the system will predict the disease accurately if the users are inserted at least three symptoms otherwise; it is not accurately predicted disease.

There are different types of frameworks are used for developing the chatbot. In(Gonda et al., n.d.), the researcher developed the chatbot for graduates assisting in teaching material. This chatbot is used in a teaching assistant training course run by the Center for Education Innovation at the Hong Kong University of Science and Technology. To do this the authors used a simple rule-based technique with the Google Dialog Flow framework. To develop this chat the researcher was applied rule-based techniques. The rule-based technique is used the if-then condition that means if the user request is

matched with the if condition the bot will return the response. The bot will return an accurate response if the knowledge engineer designed a good knowledge base. The drawback of this rule-based technique is the bot handles the user request if matches the condition. Therefore, this makes the bot not understand the language according to the context.

In (Suhel et al., 2020), The researcher has used the application of the chatbot to change the conversation of a bank customer and bank customer service officer in case of bank service through machine learning techniques. The authors have used the artificial intelligence markup language (AIML). According to(*Introduction to AIML*, 2018), these types of chatbots used pattern tag techniques the pattern is the user's query and the tag is the response to the users' request. By using this technique, the authors used to develop the chatbot by using Indian language text for helping bank customers. The advantage of this technique is easy to take the user's intent from the conversation but if the user is writing their idea according to the pattern that, the developer is used. That means the users' input must have the same with the dataset pattern.

In our country, Ethiopia different researchers study the development of chatbot applications for different purposes. Segni Bedasa was a study on developing Afaan Oromo text-based chatbot for enhancing maize productivity(Bedasa, 2021). This study aims to assist the farmer in enhancing maize productivity by using the Afaan Oromo language. To develop this chatbot they used deep learning techniques specifically RNN and DNN. For feature extraction, purposes also used TF-IDF techniques. Finally, they get the accuracy for RNN and DNN 80% and 84% respectively. However, they get this accuracy this study may have a limitation on the number of target classes or the intents of the request is small i.e they used only 25 target classes, and also the researcher doesn’t apply overfitting handling techniques or model regularization techniques.

The other researcher study on designing and implementing an adaptive bot model to consult Ethiopian published laws using ensemble architecture with rules integrated(Habitamu Asimare, 2020). This paper aims to develop a chatbot for consulting people regarding Ethiopian published laws. For this purpose, the researcher has used the Amharic language. The researcher used the RNN model for the development of the chatbot and they got an accuracy of 73.12%. However, the researcher this result this study has a limitation on handling model overfitting. They used overfitting handling techniques only by changing the number of the data-splitting ratio. This is not enough to handle model overfitting.

Table 2. 3 Related work summary

Titles of articles(reference)	**Methodology/ies**	**Gap/ s**
Automatized Medical Chatbot (Medibot)(Srivastava & Singh, 2020)	☞ AIML for taking symptoms For disease prediction ☞ KNN, SVM	☞ User inputs should be the same as with the pattern prepared by AIML.
Chatbot for healthcare system using Artificial Intelligence (Athota et al., 2020)	☞ Cosine similarity and TF-IDF to find text similarity between dataset and user query.	☞ Take high time complexity
Disha: An Implementation of Machine Learning based Bangla Healthcare Chatbot (Rahman et al., 2019)	☞ Cosine similarity and TF-IDF to identify the symptoms. ☞ SVM to predict disease.	☞ To predict the disease accurately if the user insert three or more symptoms
Creating conversational bots as a supplement for teaching assistant training course(Gonda et al., n.d.)	☞ Dialog flow framework with rule-based techniques	☞ The bot understands the query if the query is related to the rule.
Conversation to Automation in Banking Through Chatbot Using Artificial Machine Intelligence(Suhel et al., 2020)	☞ AIML	☞ The user's inputs should be the same with the pattern prepared by AIML.
Designing and Implementing Adaptive Bot Model to Consult Ethiopian Published Laws Using Ensemble Architecture with Rules Integrated(Habitamu Asimare, 2020)	☞ RNN	☞ Handling model overfitting by changing the number of data splitting ratios.
Developing Afaan Oromoo Text-based Chatbot for Enhancing Maize Productivity (Bedasa, 2021)	☞ DNN ☞ RNN	☞ Used only 25 intents and Not model overfit handling techniques applied

CHAPTER THREE

3. RESEARCH METHODOLOGY AND TOOLS

3.1. Chapter Overview

This thesis aims to develop a Bilingual chatbot for assisting Ethio telecom customers with customer services. To achieve the goals we use different techniques and tools. In this chapter, we try to explain the methodologies and tools used for developing the proposed system. The following techniques and tools have been used to achieve the objective of the study.

3.2 Research Design Procedures

To perform this study we pass through different procedures. The following procedures explain an action that takes place in this study.

- ☞ The first step is the identification of the research area. As explained in the motivation section we identify this study area by reading different research papers.
- ☞ The second step is to review different papers that have been done before on chatbots and identify the gap or limitations of those studies.
- ☞ After we identify the gap or limitation of the related study, we formulate the question that our research will be answered at the end of the study.
- ☞ The fourth step is collecting the data that have been used for developing the proposed system.
- ☞ The fifth step is designing the proposed system as it answers the research question.
- ☞ After we design the proposed system how it will answer the research question, the next step is we implement the proposed solution to solve the existing problems.
- ☞ After we implement the proposed model, we will evaluate the performance of the proposed model by using the evaluation techniques set under the evaluation techniques section.
- ☞ The last step is after we evaluate the proposed system we explain the result that gets from this study.

In short, the following diagram shows the procedures that performed to accomplish this study.

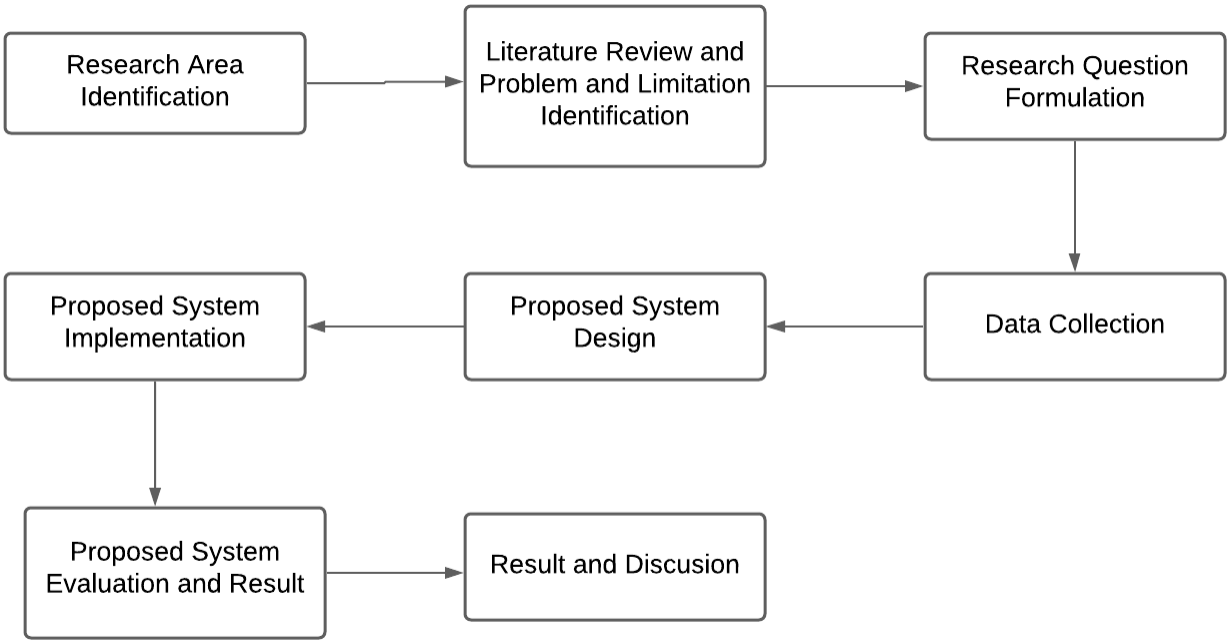


Figure 3. 1 Research procedures diagram

3.3 Data Collection Methodology

To develop the Bilingual chatbot the primary thing is preparing the dataset. For proposed system development, we gathered the data based on two sampling techniques. The first sampling technique is we gathered the data from the organization's different websites because it contains frequently asked questions about the organization's services. Therefore, by considering this, if we used those questions for the proposed system we achieve the objective of the study. The second technique is we also select the question that does not request the database checkup. Based on those techniques we prepared the data for the proposed system development. The data sources that are used for data collection are Ethio-telecom's official website, Ethio telecom portal system, Ethio telecom Facebook page, Ethio telecom telegram channel, and documents that the organizations are prepared on customer services. Besides, we collect from the website of Ethio-telecom website, Telegram channel, Ethio telecom portal system, and Ethio telecom telebirr website.

3.4 The Dataset Preparation

Once we have collected our data, the next step is to build the dataset to train the machine. The preparation of the dataset has some challenges. The first challenge is the dataset that we get from the website is prepared in the English language but we need the Amharic and Afaan Oromo datasets. To solve this problem first we translate the whole website from English to Amharic by using google translation. After we translate it into Amharic, the next step is to translate the Amharic data into Afaan Oromo manually. After the translation is completed we prepare the dataset by using JSON

format, we prepared a semi-structured dataset. JSON stands for JavaScript Object Notation. When we prepare the dataset by JSON, it contains three things such as tag, patterns, and responses.

- ☞ **Tag**: - is the intent or the idea of the users' query, which is used as a target class while we train the model.
- ☞ **Patterns**: -The set of queries request or the frequently asked question that are related to the intents.
- ☞ **Responses**: the set of possible responses that are related to the users' queries. This is used when the users.

The following are shows how we prepare the dataset. For example, the users want to ask greeting with Afaan Oromo or Amharic Languages. In Afaan Oromo they may asked "Akkam Jiirta?", "Akkam?", "Jirtaa?", "Nagaa qabdaa?", "Akkam buulte?", "Akkam olte?", "Fayaadhaa?","nagaa qabdaa?". In Amharic also users' may be asked "እንዴት ነህ?", "እንዴት ነሽ?", "ሰላም ነው?", "ሀይ", "ሰላም", "እንዴት ነሽ?". The response for those queris may "ታዲያስ ", "ሰላም ነው", "ደህና ነኝ", "አለሁ ምን ልርዳዎት?". From this, the intent or the tag of our dataset is greeting. Here we have to prepare two greeting tags that are for Amharic and also for Afaan Oromo. Therefore the converted JSON dataset was as the following.

```
{
"tag": "greetinga",
"patterns": ["እንዴት ነህ?", "እንዴት ነሽ", "ሰላም ነው", "ሀይ", "ሰላም", "እንዴት ነሽ?", "ሰላም ዋልክ?", "ሰላም ነሽ?", "ኣለህ", "ኣለሽ", "ሁሉ ሰላም", "አማን ነው"],
"responses": ["ታዲያስ ", "ሰላም ነው", "ደህና ነኝ", "አለሁ ምን ልርዳዎት?"]
    }, {
"tag": "greetingo",
"patterns": ["Akkam Jiirta?", "Akkam?", "Jirtaa?", "Haay", "Nagaa qabdaa?", "Akkam buulte?", "Akkam olte?", "Fayaadhaa?", "nagaa qabdaa?","nagaadhaa?","nageenyi badhaadhaa","akam jiirta?","ati akkam","oltiin akkami?","olmaan akkam?","fayyaa qabdaa?", "asham", "jirtuu?", "oltanii", "bultanii", "fayaa kessanii", "olmaan akkami ture", "nageeyni kessan akkami", "nageeyni kee akkami"],
"responses": ["Haay, nagaa koodha", "jira nagaa koodha"]
    },
```

From the above snapshot, we prepared two tags for a greeting that are **greetinga** for the Amharic language and **greetingo** Afaan Oromo languages. Based on this dataset we train the model and

finally, it predicts the tag and selects from the response randomly, and gives the appropriate responses.

In data preparation techniques explained the data was prepared in the form of JSON. This Json file contains a tag, patterns, and responses. The following table shows the dataset labeling result of the proposed system

Table 3. 1 Dataset labeling result

Dataset contents	Language	Result
Tags	Amharic	46
	Afaan Oromo	46
Patterns	Amharic	1000
	Afaan Oromo	1000
Responses	Amharic	100
	Afaan Oromo	100

3.5 Data Preprocessing Techniques

Data preprocessing is one of the most common tasks in many ML applications. It is traditionally an important step for natural language processing (NLP) tasks. It transforms text into a more digestible form so that machine-learning algorithms can perform better. The following preprocessing techniques will apply to develop a Bilingual Chatbot to assist Ethio-Telecom customers on customer service.

3.5.1 Text Normalization and Cleaning

Text normalization is used for a series of related tasks meant to place all text on a level playing field. The normalization of the Amharic and Afaan Oromo text is based on the writing system of the language. To normalize Afaan Oromo text convert all text to an equivalent case upper to lower was applied. For example, the tokenized word "**Ergaa"** has normalized as **"ergaa"** this makes the computer understand the word **"Ergaa"** was equivalent meaning with **"ergaa",** and **"ERGAA"** etc. In the case of the Amharic language, replace the letter with the letter that has the same sound. For example, in the word **"ዐጭር"** the letter **"ዐ"** was replaced with **"አ"** because those letters have contains the same sound this also helps the computer to understand the word **"አጭር"** has a similar meaning with **"ዐጭር".**

The other preprocess was clearing the text by removing different symbols and numbers from the text to make the computer understand the natural language easily. The Amharic language has its numbering system like [፩, ፪, ፫, ፬, ፭, ፮, ፯, ፰, ፱…] and it may use number [0, 9] the same as with Afaan Oromo language. Therefore, the first activity of this study was check if those numbers are exits or not. If they exist, remove them from the text. The other symbols are when the users are used chat they may use different emoji symbols like [😀 😃 😄 😁 😆 😅 😂 🤣 🤚 ✋ 🖐 👌 ♡....] to express their idea. In addition to this, removing those symbols has necessary for clearing the text. For example if Afaan Oromo users' type **"AKKAM jiirta✋✋✋"** and Amharic users write **"ዕንዴት ነሽ✋✋✋?"**

Finally, removing punctuation marks for both languages are another activity. For the Amharic language, there are different punctuation marks such as [፡ word separator, ። full stop (period), ፣ comma, ፤ semicolon, ፥ colon] in addition to this they used the question mark, exclamation mark. Afaan Oromo also used punctuation marks that are the same as with the English language. Therefore, we have to remove those punctuation marks from the text. The following table shows how this study was normalized or cleared the above two examples.

Table 3. 2 Data normalization and cleaning example

Before normalization and cleaning	**After Normalization and Cleaning**
AKKAM jiirta✋✋✋	akkam jiirta
ዕንዴት ነሽ✋✋✋?	**እንዴት ነሽ**

Generally, this study used text normalization for the process of putting all words on equal footing and allows processing to proceed uniformly[9]. Therefore in-text normalization, this study applied two things. The first is text clearing and replacing the letter with the same representation. For this purpose, the following algorithms have applied in the proposed system.

3.5.1.1 Text Cleaning Algorithm

Algorithm 3. 1 Text cleaning algorithm

INPUT: Uncleaned data

OUTPUT: cleaned dataset

START:

1: Read the dataset

2: WHILE(it is not the end of file):

IF data contain punctuations [',', '.', '"', ':', ')', '(', '-', '!', '?', '።','፣', '፤'] THEN

Remove punctuations

IF data contain number [፩ ፪ ፫ ፬ ፭ ፮ ፯ ፰ ፱ ፲ ፳ ፴ ፵ ፶ ፷ ፸ ፹ ፺ ፻ ፼ ፽] THEN

Remove Number

IF data contain symbols [😀 😃 😄 😁 😆 😅 😂 🤣 🤚 🖐 ✋ 🖖 👌 ❤….] THEN

Remove Symbols

Return cleaned data

3**. HALT**

3.5.1.2 Text Normalization Algorithm

Text normalization is the process of replacing the symbol with a different symbol that has the same sounds. As we developed a bilingual chatbot, we applied these techniques for both Amharic and Afaan Oromo Language.

For the Amharic language, this study applied tokenization by replacing the letter with another letter that has the same sound. The following algorithm shows how the proposed system performed normalization on Amharic text.

Algorithm 3. 2 Amharic text normalization algorithm

INPUT: Un normalized data

OUTPUT: Normalized data

START:

1: Read the dataset

2: WHILE (it is not the end of file):

IF data contains [ኻሐ]' THEN

replace with 'ሀ'

IF text contains [ሐኑ] THEN

replace with 'ሁ'

IF text contains [ኺሒ]

THEN replace with 'ሂ'

IF text contains[ኼሔ] THEN

replace with 'ሄ'

IF text contains [ሕኻ] THEN

```
            replace with 'ህ'
          IF text contains [ኖሖ] THEN
            replace with 'ሆ'
          IF text contains[ሠ ሡ ሢ ሣ ሤ ሥ ሦ] THEN
            replace with corresponding [ሰ ሱ ሲ ሳ ሴ ስ]
          IF text contains [ዐ ዒ ዓ ዔ ዕ ዖ ኣ] THEN
            replace with corresponding [ኡ ኢ አ ኤ እ ኦ አ]
          IF text contains [ጸ ጹ ጺ ጻ ጼ ጽ ጾ] THEN
            replace with corresponding [ፀ ፁ ፂ ፃ ፄ ፅ ፆ]
  Return normalized text
3 END
```

The purpose of normalizing Amharic text is to make the computer understand easily the same word that is written with a different symbol or letters. For example, the word **Renaissance** is written as **"ህዳሴ", "ሕዳሴ","ኅዳሴ", "ህዳሤ", "ሕዳሤ"** by using the above algorithm those words are converted as **"ህዳሴ"**. Therefore, if the users write the **"ሕዳሤ" or "ኅዳሤ" or …** the computer understands it as **"ህዳሴ".**

Afaan Oromo uses the letter [A-Z] as capital letters and [a-z] as small letters. In this study to Afaan Oromo text has normalized by converting capital letters into small letters. For this purpose, this study was applied the following algorithm.

Algorithm 3. 3 Afaan Oromo Text Normalization Algorithm

```
INPUT: Un normalized data
OUTPUT: Normalized data
START:
1: Read the dataset
2: WHILE(it is not the end of file):
      Convert data into small letter
  Return normalized data
3. END
```

The purpose of this algorithm on the proposed system is to make the computer understand the same words that are written with different letters. For example, the word **"message"** is written as **"ERGAA", "Ergaa", "ERgaa"…** but if the users write as above example after normalization it converted as **"ergaa".**

3.5.2 Tokenization

Tokenization is a step, which splits longer strings of text into smaller pieces, or tokens. Larger chunks of text have often been tokenized into sentences; sentences have often been tokenized into words, etc. Further processing is usually performed after a bit of text has been appropriately tokenized. This study applied tokenization to the Amharic and Afaan Oromo text by using spaces or punctuation marks. For example, Afaan Oromo text **"Ergaa gabaabaa qarshii meqaan ergina?"** and Amharic text **"የአጭር መልዕክት ታሪፍ ስንት ነው?"** is meaning "what is the tariff for sent message?" has tokenized as**:**

Table 3. 3 Tokenization example

Ergaa	gabaabaa	qarshii	meqaan	Ergina	**?**
የአጭር	መልዕክት	ታሪፍ	ስንት	ነው	**?**

3.5.3 Stop Word Removing

Words such as articles and some verbs are usually considered stop words because they don't help us to find the context or the true meaning of a sentence(*Why Is Removing Stop Words Not Always a Good Idea | by Lima Vallantin | Medium*, n.d.). This helps to reduce the size of the dataset without any doubt and makes increases the performance of the model. For this study, Afaan Oromo stop words and Amharic stop words have collected from different articles done before. Some of Amharic stop words are[ስለ የሚሆኑ ለዚያው, ከሆነው, ስለዚሁ……](Kassie, 2009). For afaan oromo some of stop words are like[amma, inni, kanaaf, kanaaf, sitti, ammo, irra, Kanaafuu, sun…](Tesfaye, 2010).

Algorithm 3. 4 Algorithm for removing stop words

INPUT: list of words in the dataset

OUTPUT: list of words does not have a stop word

Variable: ListOfStopwords[], ListOfRemovedStopWord[]

1: Read the dataset

2: WHILE(list of words in the dataset):

IF list of words in the dataset not in ListOfStopwords: THEN

Append the word in ListOfRemovedStopWord

Return ListOfRemovedStopWord

3: Halt

3.6 Feature Extraction Techniques

After cleared the text by removing the unnecessary symbols, punctuation, and text normalization the next step is feature extraction. When used a machine-learning algorithm and work with textual data, machine-learning algorithms cannot work on the raw text directly. Therefore, by using feature extraction techniques to convert text into a matrix (or vector) of features. Feature extraction is one of the issues of text classification is irrelevant and repeatedly occurred features affect the accuracy and performance of the classifiers(Gurmessa, 2020). It is a technique that is used to translate the raw data into a particular input machine-learning algorithm requires. There are different types of techniques for feature extraction.

For the proposed work, we used word2vec techniques for converting the text data into equivalent numeric value. The word2vec algorithm used in neural network model to learn word association from large dataset of text. We select Word2vec techniques because it is popular embedding method very fast to train.

3.7 Model Selection Techniques

A machine-learning model is a file that have trained to recognize certain types of patterns. To train the model the prepared data called training data has used. There are four types of machine learning algorithms (*A Guide to the Types of Machine Learning Algorithms | SAS UK*, n.d.). Those are supervised learning, semi-supervised, unsupervised learning and semi-supervised or reinforcement learning. In this work, this study used supervised because the prepared data is labeled or the target class is known.

For the proposed work, this study select the model based on the prepared dataset and giving accurate responses for the customer by considering customer requests. As we explained on data preparation methodology, we prepare our dataset by using JSON. This dataset is categorized based on its intents. The intents are the contextual meaning that the users request or query that have. Therefore, those intents are used as a class for prediction while we train the model. Therefore, we have many intents or target classes in our dataset. This makes the model selection criteria focus on multiclass classification.

When we compare machine learning algorithms with a deep learning algorithm, deep learning algorithms are more accurate than machine learning algorithms in multiclass classification. According to(Miškuf & Zolotová, 2016), the author tried to compare a machine learning algorithm with a deep learning algorithm and they get deep learning is more accurate in multiclass classification. Therefore, the proposed work is used deep learning techniques.

Deep learning is an artificial intelligence function that imitates the human brain on pattern creating and process data to make different predictions. In deep learning there are different techniques is used for different purposes. Those are Artificial Neural Network (ANN), Recurrent Neural Network (RNN), Deep Neural Network, Convolutional Neural Network (CNN), etc. From those techniques, the proposed system has been implemented by using the Deep Neural network (DNN) model and Bi-LSTM model.

In deep learning, three layers are the input layer, hidden layer, and output layer. A Deep Neural network is a feedforward network that observes the flow of data from the input to the hidden layer and then the output layer. In the proposed system works first to take the input text and preprocessed the text, then the preprocessed text was converted into the vector by using word2vec technique, after that the vector data was used in the input layer for initializes the model, finally, the model performed the calculation on the hidden layer and produces the output. The following architecture shows how the DNN has applied intent classification in the proposed system. Assume that if the user request "Siimiin koo najalaa bade" it means, "I lost my sim card" the system was predicted as the following.

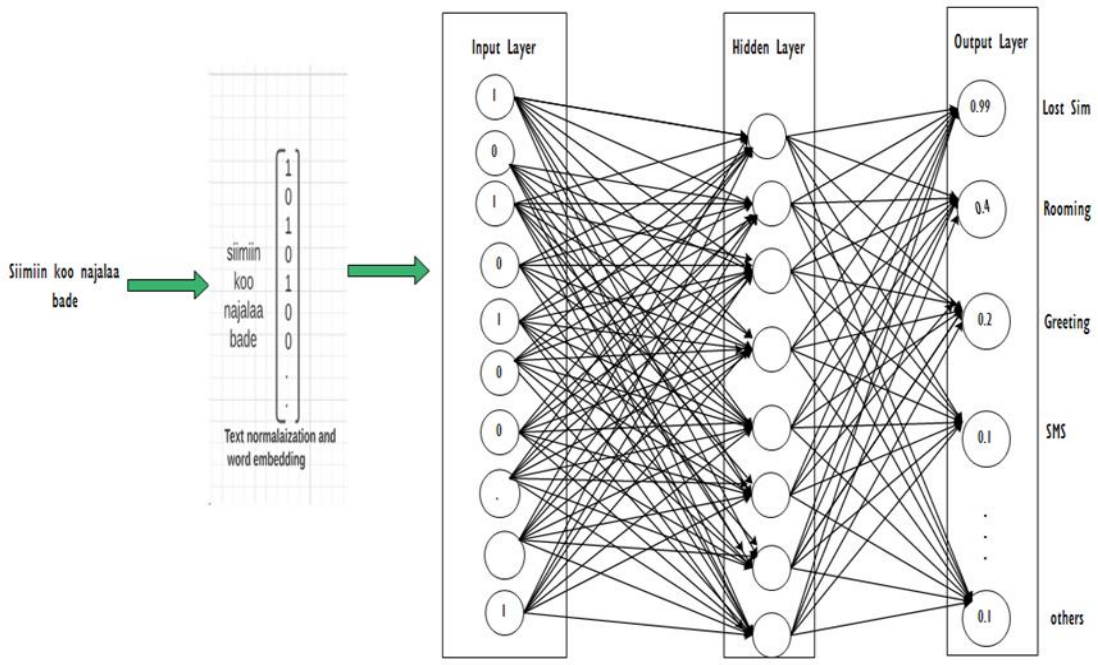


Figure 3. 2 DNN model intent classification

The above figure shows that how the DNN predicted the intent of the users' requests. If we see the output layer there is the value calculated by the model. This value is used to classify the query into the correct target class. This means the highest value indicates that the intent of the users is in that class.

From the above figure, the user request has been classified into lost sim class because it has the highest value. Therefore, the system gives the response by selecting randomly from the lost sim class.

3.7.1 Activation Function Techniques

An activation function is a mathematical equation used to determine the output of the neural network. In DNN the function is attached to each neuron in the network, and determines whether it should be activated ("fired") or not, based on whether each neuron's input is relevant for the model's prediction(*7 Types of Activation Functions in Neural Networks: How to Choose?*, n.d.). In deep learning, there are different types of activation functions. Those are Sigmoid (Logistic) function, TanH / Hyperbolic Tangent, ReLU (Rectified Linear Unit), Leaky ReLU, Softmax, and Swish. From those activation functions for the proposed system, we used at the input layer Relu activation function. We select this activation function by using different experiments and we got a good result when we compare with other activation function and we used the Softmax activation function at the output layer. We select this activation function because according to(*7 Types of Activation Functions in Neural Networks: How to Choose?*, n.d.), the softmax function has an advantage in handling multiple classes prediction.

Softmax activation function has been used for multiclass classification. When we used a deep neural network, there is a mathematical computation between each layer. This mathematical calculation is used to produce the confidence of the class prediction. The softmax activation function used in multiclass classification normalizes these values by dividing by the sum of all the exponentials. Doing so ensures the sum of all exponential values adds up to one. After we used the softmax activation function, we got the confidence values between zero and one. Based on those values the target class of the users' query has been identified. According to Wikipedia(Gao & Pavel, 2017), the softmax formula is the following one.

$$\delta(z)i = \frac{e^{zi}}{\sum_{j=1}^{k} e^{zj}} \; for \; i = 1, \dots, k \; and \; z = (z1, \dots, zk)$$

Equation 3. 1 Softmax Activation function formula

3.7.2 Model Overfitting Handling Techniques

One of the common problems in machine learning is model overfitting. Overfitting refers to a model that learns the training data too well but fails to generalize to new data. It occurs when a model learns the detail and noise in the training data to the extent that it negatively affects the performance of the model on new data. Different techniques are there that used to reduce model overfitting(Ying, 2019).

Those are Cross-validation, Regularization, Early stopping, Punning, Dropout, etc. From those techniques for the proposed work, we used Regularization techniques for handling model overfitting.

Regularization is a collection of strategies that enable machine-learning algorithms to generalize on the new input data and it is used to reduce the training time of the model. It is used to reduce generalization errors but not training errors. There different types of regularization techniques such as the L1 regularization technique is called Lasso Regression and the model which uses L2 is called Ridge Regression[58].

We select the L2 regularization because it adds squared magnitude as a penalty term to the loss function and it is used much more often than L1 according to (*Machine Learning Model Regularization in Practice: An Example with Keras and TensorFlow 2.0 | by B. Chen | Towards Data Science*, n.d.). L2 regularization is used to reduce the chance of model overfitting by keeping the magnitudes of the weight values small.

3.8 Model Evaluation Techniques

After doing everything, we have to evaluate our model to know whether the goals have been achieved or not. We evaluate the model by using two techniques. The first way is evaluating our models by using our data set. That means we will split our data into train datasets and test datasets. The training dataset has been used to train our model and the test dataset has been used to test the model. The second way is we evaluate the model by make conversation after the model has trained. This makes to know whether the chatbot gives the correct response for the user or not.

We evaluate the model, based on the dataset by using different evaluation metrics. From those model evaluation metrics, we used accuracy, recall, precision, and F1 score.

- **Accuracy**: - is the total correctly predicted data out of all training data. Is used to measure how the model was classified the data into their corresponding classes. Accuracy is calculated by using the following formula.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Equation 3. 2 Accuracy calculation formula

Where,

TP: True Positive: Predicted values correctly predicted as actual positive

FP: False Positive: Negative values predicted as positive

FN: False Negative: Positive values predicted as negative

TN: True Negative: Predicted values correctly predicted as an actual negative

- **Precision: -**The nearness of two or more capacities to each other is known as the precision of a material. In machine learning metrics evaluation, precision is the positive classes that are correctly identified by the model.

$$Precision = \frac{TP}{TP + FP}$$

Equation 3. 3 Precision calculation formula

- **Recall: -** is one of the evaluation metrics that used to measure the negative classes that are correctly identified by the model.

$$Recall = \frac{TP}{TP + FN}$$

Equation 3. 4 Accuracy calculation formula

- **F1 score:** - is an evaluation metric that is used to find the harmonic mean of the precision and recall of the model classification.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Equation 3. 5 F1 score calculation formula

The second technique that we used for system evaluation we test by using the human evaluation technique. We evaluated the proposed system by the user that has no idea about the proposed system dataset. They used either by Amharic language or Afaan Oromo language or both and then judge the system according to the form that was prepared for evaluation forms that we attached in appendix B1 and B2. The forms are prepared to evaluate the system's performance and the users' acceptance of the system.

3.9 User Interface Design Techniques

The proposed system was mainly developed to make computers communicate with users by using two languages. For this communication purpose, we have to develop a graphical user interface. To develop this graphical user interface we see with two perspectives. The first is how we develop the graphical user interface that supports multiple users at the time, and the second perspective is how we develop the system that is compatible with users' devices. To answers those questions we used different techniques. The first technique is we develop a web application that is integrated with the model. To do this we used the flask framework. The second technique is we used bootstrap to make the web easily compatible with different window size devices.

3.9.1 Flask Frameworks

Flask is a python web framework that is used to developing web applications. The proposed system was developed by using this framework. Flask has many advantages. From those advantages, some are flexibility and simplicity. For this reason, we used the Flask framework.

When we used the flask framework, we used two directories, which is known as a template and static. In the static folder, we put the files that are used to make attractive the system. those files are an image, CSS files, JavaScript files. In the template files, we put the HTML files that were used to develop a graphical user interface. After we create those, all files the next step is loading the trained model file that is saved in our hard disk and make it to predict the users' request and finally give an appropriate response.

3.9.2 Bootstrap

Bootstrap is a Cascading Style sheet (CSS) framework that is used for developing responsive web applications. It is an open-source framework used at the front end of application development. It contains different HTML templates that are used to develop a graphical user interface. Bootstrap has many advantages. From those the first advantage is to make the app compatible with the device that has different window sizes, the second advantage is easy to developing web application, the third advantage is it is integrated with JavaScript and makes to use different symbols easily. We used this framework to add all bootstrap features to the proposed system.

3.10 Simulation Tools

The proposed system has been developed by using numerous tools. Those tools are used for designing the proposed system and implementing the proposed system. In this phase, we explained those tools within their purposes.

3.10.1 Proposed System Design Tools

To develop the proposed system we design different tools to explain the proposed system, how it works, and different actions that are performed while we developing the system. For this purpose, we used two tools. Such as Edraw max 7.7 and online lucid. app.

Lucid.app is used to provide online diagramming to make it easy to draw flowcharts, different charts, wireframes, UML, etc. [54]. We used it to design different architecture that we have used in the development of the proposed system. This web application has supports different symbols, lines, architectures that are used for designing different diagrams. We select this web for design it is easy to

use and it is used for attractively designing the architecture and we easily export the diagram in a different way like png, jpg, Visio, and others and we used it for documentation and presentation.

The Edraw Max is a desktop application that is installed on the computer and used for designing different UML and other architecture. We also used this software for designing different architecture that relates to the proposed study. We used this software because it is easy to use and it is easy to integrate the diagram with Microsoft office.

3.10.2 Proposed System Implementation Tools

3.10.2.1 Programming Language

To develop the proposed system we used python 3.7.4 programming language. Python is a powerful programming language that has efficient high-level data structures and a simple. It is an effective programming language that uses an Object-oriented approach(*The Python Tutorial — Python 3.9.2 Documentation*, n.d.).

3.10.2.2 Integrated Development Environment (IDE) Tools

- **Spyder [1]3.3.6**

Spyder is a free open-source scientific development environment that use to write a python program designed by and for scientists, engineers, and data analysts. We used this IDE to write and run the python code. Specifically, when we designed the user interface by using a flask we used this IDE.

- **Anaconda Navigator [2]4.9.2**

Anaconda Navigator is a desktop application that used to navigate, manages conda package and environments without using the command-line command. We used this navigator to launch the jupyter notebook, to manage frameworks that are necessary for developing the proposed system.

- **Jupyter Notebook[3] 6.0.1**

Jupyter notebook is a web application that used to perform different simulations and data visualization. The advantage of using this application is it is easy to write the code, run code, and manage the file,

[1] *https://www.spyder-ide.org/*

[2] *https://docs.anaconda.com/anaconda/navigator*

[3] *https://jupyter.org/*

and data processing and visualization. We used this application to write, run a python program and for data processing and visualization while we developed the proposed system.

- **Colab[4]**

Colab also known as collaboratory is a free jupyter notebook that runs on the cloud and stores data on google drive. It allows the users to run deep learning code that takes a long time within a short time by allowing users to use GPU and TPU. We used this web application for the visualization of data, processing data, writing the code, and run the code.

3.10.2.3 Frameworks Used in Proposed System Implementation

- **nltk[5] 3.4.5**

Natural language toolkit (nltk) is a python library that is used to work in natural language processing. This library contains different APIs that are used to implement WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning. We used this library for applying data preprocessing techniques that applied to the development of the proposed system.

- **sklearn[6] 0.21.3**

sklearn is a python library used for applying machine learning algorithms. It contains different algorithms that use for classification, regression, clustering dimensional reduction. We used this library in the proposed system development for splitting the dataset into train data and test data.

- **tensorflow[7] 1.15.0**

Tensorflow is an open-source library developed by Google it uses in deep learning application development for classification and prediction. We used in proposed system development to applying deep learning algorithms for intent classification.

- **numpy[8] 1.16.5**

---

[4] https://colab.research.google.com/

5 https://www.nltk.org/
6 https://scikit-learn.org/
7 https://www.tensorflow.org/
8 https://numpy.org/

Numpy is an open-source numerical library that is used for the visualization of array and matrix data. We used this library while we converted the data into the vector and reshaping the vector to fit the vector with the proposed model.

- **random**[9]

Random is built a python module that is used to make random numbers. We used in the development of the proposed system for the query that has more than one response to select one response from those randomly.

- **json**[10]

json is a built-in library, which can be used to work with JSON data. In the proposed system, we prepared our dataset by using JSON format. To read and manipulate the prepared dataset we used this built-in library in the proposed system development.

- **matplotlib**[11]

matplotlib is a python library that uses for creating static, animated, and interactive visualizations in Python. In the development of the proposed system, we used it for visualizing data and results by using a graph.

- **flask**[12]

Flask is an API that is used for developing web applications by using a python programming language. In the proposed work, we used this library to develop a web that connects the trained model with customer requests.

3.10.2.4 Hardware Tools

The tools and the frameworks discussed in the above portion have been deployed on the personal computer that contains the following properties.

- Processor: - Intel(R) Core(TM) i5-5300U CPU @ 2.30GHz
- Installed memory (RAM): - 8 Gigabytes
- Hard disk: - 931.51 Gigabyte
- Operating system: - windows 10 pro
- System type: - 64-bit operating system, x64-based processor

9 https://docs.python.org/3/library/random.html

10 https://docs.python.org/3/library/json.html

11 https://matplotlib.org/

12 https://flask.palletsprojects.com/en/1.1.x/

CHAPTER FOUR

4. DESIGN AND IMPLEMENTATION OF THE PROPOSED SYSTEM

4.1. Chapter Overview

This chapter discusses the design and implementation of a Bilingual chatbot for assisting Ethio telecom customers with customer services. The first section deals with the design of the proposed model. The second section talks about the architecture of the proposed system, the third section is discus about flowchart for the implementation of the proposed system, and the fourth section deals with the implementation of the proposed model.

4.2 Proposed Model Architecture

We classified the process proposed model into the training process and classification process. The training process has different phases or packages. The first phase is the data preprocessing techniques that we discussed in the methodology section was applied in this phase and we converted the sentences into a list of words. The second package is converting the list of words that we get from the first phase into numeric vectors. After we convert the text into a numeric vector, the next phase is the model selection phase. In the model selection phase, we applied the Bi-LSTM model and the tflearn DNN model.

After we train our model, we saved the model in file format. The second process is the classification process. In the classification process, we applied data preprocessing and feature extraction techniques. Then we load the model and we make it to predict the intent of the text. The model was returning two values. The first value is the intent or the target class of the query and the second was the confidence value calculated by the model.

The confidence value is helpful to know how the model has predicted the users' queries into the target class i.e. the class that contains high confidence is selected as users' intent and the model was select responses randomly from that class and return for the users. If the confidence value is small the model did not understand the user's request so rather than selecting the response from the class that has a small confidence value it gives the response for small confidence by considering if the users' request is requested by Amharic or Afaan Oromo language. If the users' request is requested by using the Amharic language the model was returning "ምን ማለት እንደፈለጉ አልገባኝም" it means I don't understand what you mean message but if the user request is Afaan Oromo language it returns the message "Maal akka jechuu barbaadan naaf hin galle" response. The following figure shows the proposed model architecture.

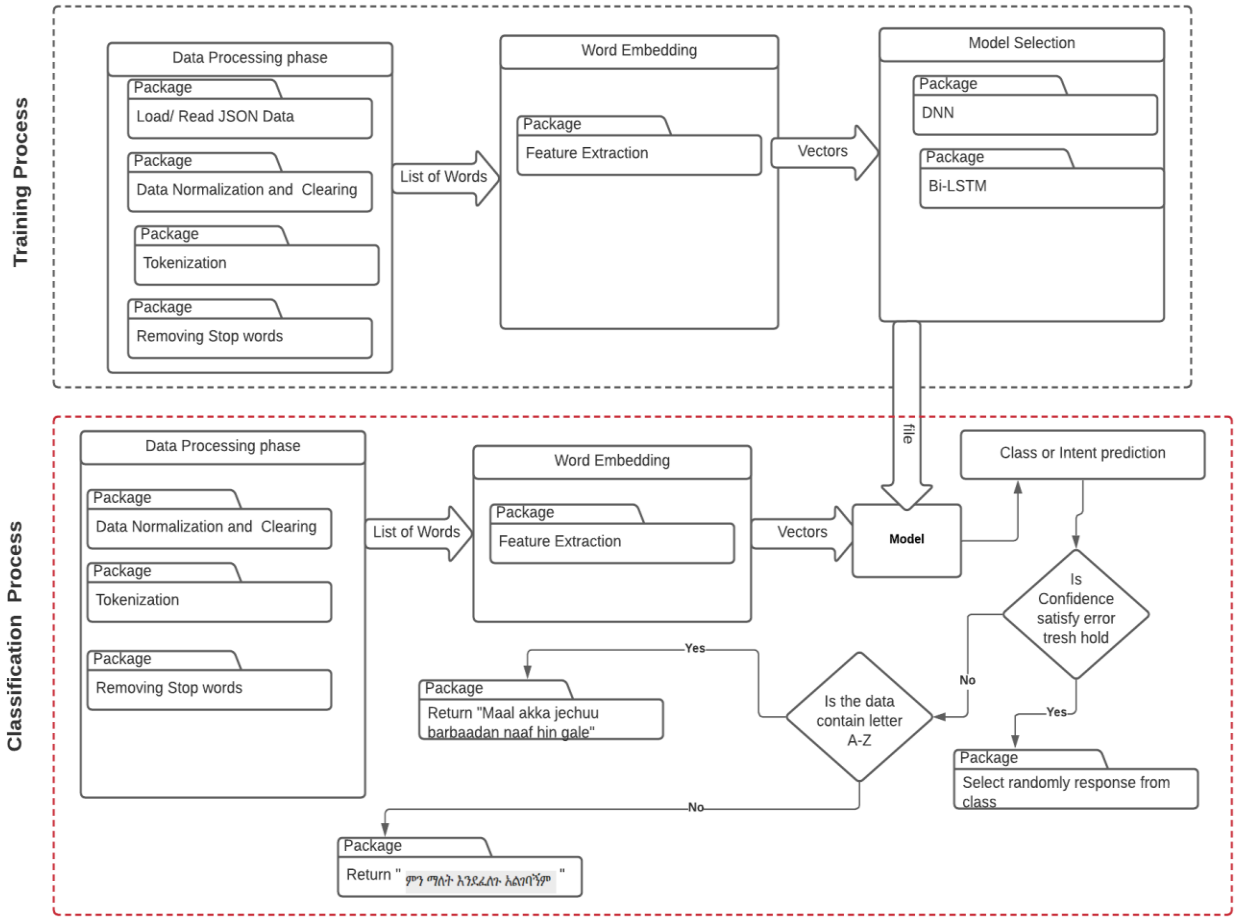


Figure 4. 1 Proposed model architecture

4.3 Proposed System Prototype Architecture

In this phase, we try to design the prototype for a proposed system prototype. As we discussed in the methodology phase we use different tools to develop the system. Here, we designed how all tools have integrated or communicated to each other after we designed and developed the model. From the following architecture, the customer will send the request through HTTP. The flask server will receive the HTTP request from the user and provide the request for the model. After the flask provides the request for the model, the model will understand the intent of the request i.e. natural language. The model will be select the response randomly that related to the user's intent. Then the model will respond to the flask server and the flask server will send an HTTP response for the users.

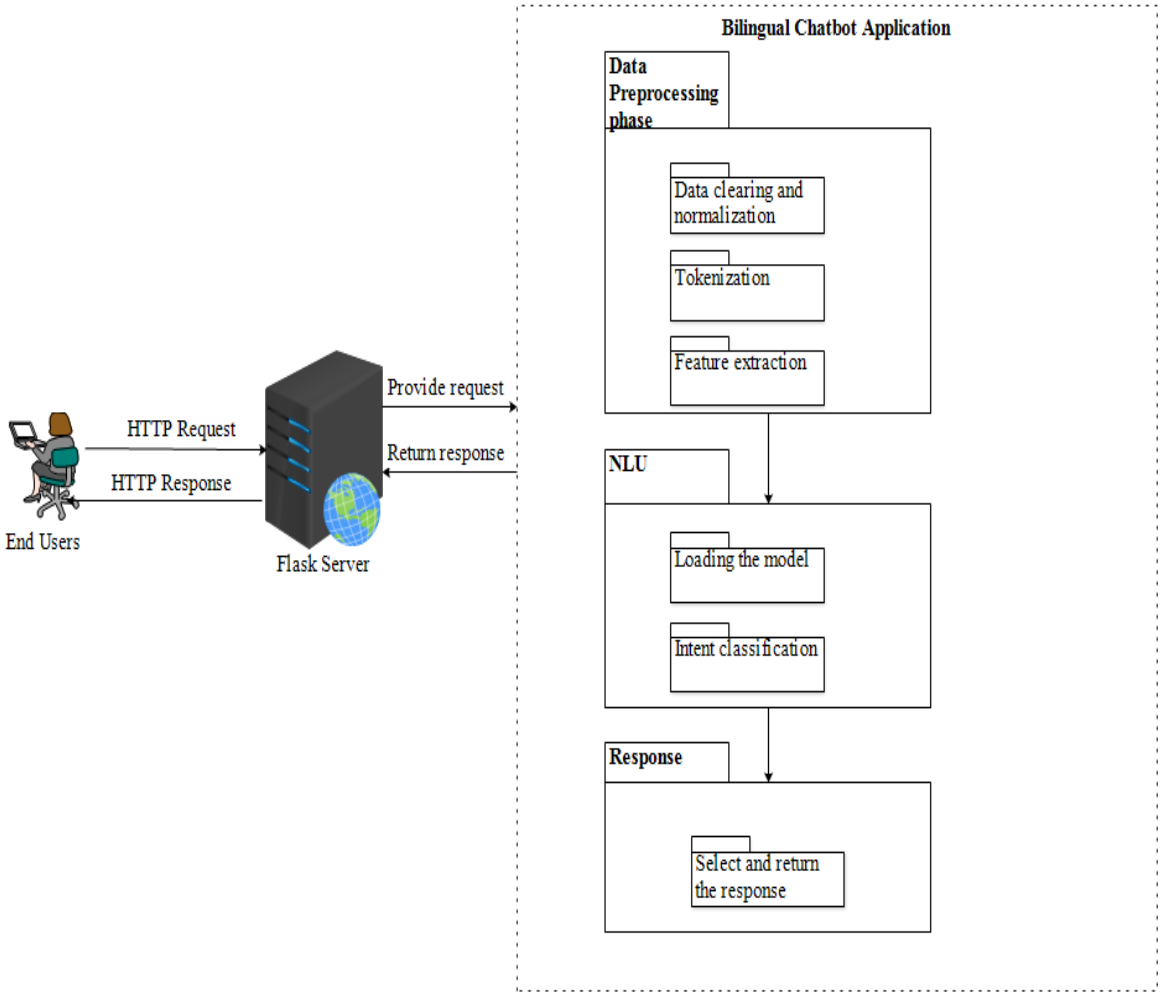


Figure 4. 2 Proposed system prototype architecture

4.4 Implementation Flowchart

In this section, we discussed how we implement the proposed system from reading the data to system evaluation. The implementation of the proposed system follows different steps such as loading or reading dataset, data preprocessing, text vectorization, model selection, train and save the model, loading the saved model, and finally evaluate the model.

- **Loading JSON data**: - this step was used while us reading our dataset that was prepared by using JSON. To read the files we import the Json library first and it has a load () method that is used for reading the data. By using this method or function, we read our dataset.
- **Data Preprocessing**: - before using the loaded data directly, the first thing is to preprocess the data. At this stage, we normalize, clean, and tokenize the data was performed.

- **Text Vectorization**: - in the proposed system implementation, we applied deep learning techniques so; to train the model we have to convert the text into the vector. In this study, we applied a word2vec to convert the text into a vector.
- **Model selection**: - for the proposed system, we used different deep learning techniques. In this step, we implement those models for the proposed system implementation.
- **Train and save model**: - after we select the model, we train the model by using our dataset. Therefore, when we give the input to predict the new data it was a response for us. Once we trained the model, we have to save it to use it repeatedly.
- **Load trained model:** after we train the model, we have to test the model by giving new input. For this purpose, we load the saved trained model for prediction purposes. In the proposed system implementation, we load the trained model by using the pickle library.
- **Evaluate model:** - in this stage, we evaluate the performance of the model that we are trained by using our dataset. We evaluated the model by using human evaluation i.e. by giving the input text is predict correctly or not and the second is by splitting the data into train and test data we evaluate the proposed model.

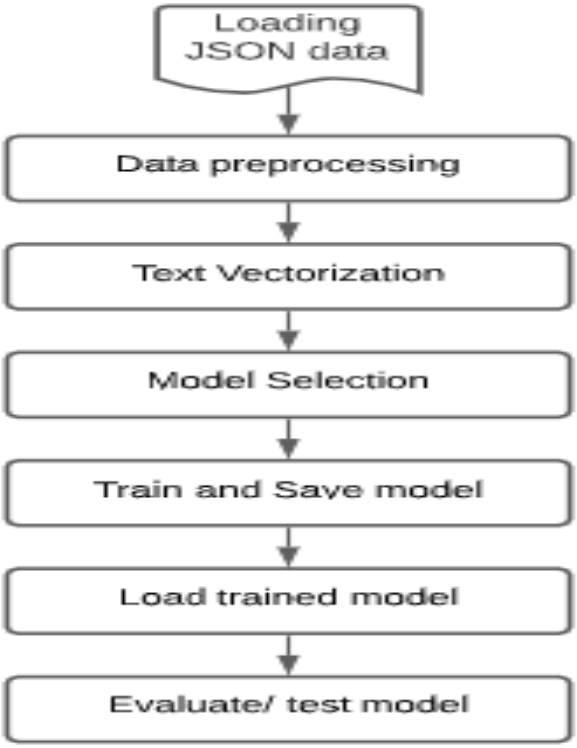


Figure 4. 3 Implementation architecture

4.5 Loading dataset

After we prepared the data, we used it to train and evaluate the model. To apply those things we have to read the data. As we discussed the data preparation methodology, we prepared the dataset by using JSON we read or load that data set by using the Json library like the following snapshot code.

Code 4. 1 Data loading implementation

```
#importing our intent file used for training the model.
with open("TelecomFAQ.json",encoding="utf8") as json_data:
intents = json.load(json_data)
# Loading data from TelecomFAQ.json file to variable intents
```

4.6 Proposed System Data Preprocessing Implementation

After data preparation, the next step is data preprocessing. The purpose of data preprocessing is to make the computer easily understand the natural language. As we discussed in the methodology section we apply different data preprocessing techniques to develop the proposed system. In this section, we try to show how we implement those methodologies that we used to develop the proposed system.

4.6.1 Tokenization Implementation

Tokenization is the process of splitting sentences into a list of words. The nltk library has contains the word_tokenize function that is used for performing tokenization. By using this module we implemented the tokenization for the proposed system. The following snapshot shows the implementation tokenization.

Code 4. 2 Tokenization Implementation

```
words=[]
# Starting a loop through each intent in intents["patterns"]
for intent in intents["intents"]:
   for pattern in intent["patterns"]:

      # tokenizing each word in the sentence by using word tokenizer and storing in w
      w = nltk.word_tokenize(pattern)

      # g tokenized words to words empty list that we created
      words.extend(w)
```

4.6.2 Text Cleaning and Normalization Algorithm

Text cleaning is the process of removing unnecessary symbols or characters from the data. We removed those punctuation marks, Amharic numbers, symbols, or characters to make our model

prediction accurate. To implement the proposed system we cleared the symbols like the following snapshot code.

Code 4. 3 Text cleaning implementation

```
def remove_extra(word):
   cleared=word
   char =['?', '፣','።','፤','!','%','፩', '፪', '፫', '፬', '፭', '፮', '፯', '፰', '፱ ','❤','≡', '❤','❥','❣' ❦', '❧','ღ', 'ღ','ᔕ', 'ღ','ლ','❤’]
   for letter in word:
      if letter in char:
         cleared=cleared.replace(letter, "")
  return cleared
```

The other technique that is applied in our proposed system implementation is text normalization. We applied this technique for both Amharic and Afaan Oromo text. In Amharic text, we applied by replacing the characters with other characters that have the same sound and we applied for Afaan Oromo language by converting all characters into small letters. The following snapshot code shows the implementation of text normalization applied to the proposed system development. For Amharic text normalization:

Code 4. 4 Implementation of Amharic text normalization

```
def datas(words):
   replacements =  {"ሐ":"ሀ","ኅ":"ሀ","ሑ":"ሁ","ኁ":"ሁ","ሒ":"ሂ","ኂ":"ሂ","ሕ":"ህ","ኅ": "ህ",
"ሖ":"ሆ","ኆ":"ሆ","ሠ":"ሰ","ሡ":"ሱ","ሢ":"ሲ","ሣ":"ሳ",
                "ሤ": "ሴ","ሥ": "ስ","ሦ": "ሶ","ዐ": "አ","ዑ": "ኡ","ዒ": "ኢ","ዓ": "ኣ","ዖ": "ኦ","ዕ": "እ","ዔ": "ኤ","ጸ": "ፀ","ጹ": "ፁ","ጺ": "ፂ","ጻ": "ፃ","ጼ": "ፄ","ጽ": "ፅ"
                ,"ጾ": "ፆ"}
   return "".join([replacements.get(c, c) for c in words])
```

For Afaan Oromo text normalization, we applied the following snapshot implementation.

Code 4. 5 Implementation of Afaan Oromo text normalization

```
words=[]
w = nltk.word_tokenize(pattern)
       # g tokenized words to words empty list that we created
       words.extend(w)
for w in words:
  w.lower()
  words=w
```

4.6.3 Stop Word Removing Algorithm

Removing the stop word is one of preprocessing techniques applied to the proposed system implementation. Before applying the stop word removing technique we prepared the corpora that contain Amharic and Afaan Oromo stop words and we put those corpora in nltk data folders for simplicity. After that, we applied the stop word removing techniques. The following snapshot indicates the implementation of stop word removal.

Code 4. 6 stop word removing implementation

```
from nltk.corpus import stopwords
AfaanOromoSTOPWORDS = stopwords.words('oromiffa')
AmharicSTOPWORDS = stopwords.words('amharic')
non_stop_words=[]
for w in words if w not in AfaanOromoSTOPWORDS and if w not in AmharicSTOPWORDS:
  non_stop_words.extend(w)
```

4.7 Proposed Feature Extraction Implementation

When we used a machine-learning algorithm or deep learning algorithm, we have to convert it into a numeric vector that equivalent to the text data. For this purpose, we used word2vec technique as we discussed in the methodology sections. The following implementation was indicating the implementation of the word2vec techniques for the development of the proposed system.

Code 4. 7 Loading the pre-trained word2vec model

```
from gensim.models import Word2Vec
models = Word2Vec.load('EthiotelecomFAQ.model')
word2idx = {"_PAD": 0}
vocab_list = [(k, model.wv[k]) for k, v in model.wv.vocab.items()]
embeddings_matrix = np.zeros((len(model.wv.vocab.items()) + 1, model.vector_size))
for i in range(len(vocab_list)):
   word = vocab_list[i][0]
   word2idx[word] = i + 1
   embeddings_matrix[i + 1] = vocab_list[i][1]
```

4.8 Dataset Splitting Implementation

As we discussed under evaluation methods we evaluate the proposed system by using two techniques the first technique is we split our dataset into train data and validation data. Based on those data we evaluate the proposed system by using those data. We split the dataset by 90% for training and 10% for validation or test. The following snapshot code implies how we implemented the data splitting techniques

Code 4. 8 Dataset splitting implementation

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(train_x, train_y, test_size=0.1, random_state=42)
```

4.9 Model Implementation

For the proposed system implementation, we applied different types of deep learning models. By using Keras API we developed a Bi-LSTM model and by using tflearn API we developed the DNN model. Under this section, we try to show how we implement the model in the proposed system. Before the implementation of the model in the following section we tried to explain the component that artificial neural network was used and applied in the proposed system implementation.

4.9.1 Layers of Deep Learning

As the human mind contains computing units that are known as neurons working in parallel to exchanging information through their connectors synapses the artificial neural network also has different neurons that are interconnected to each other to perform different tasks(Zayegh & Al Bassam, 2018). When we used a deep learning algorithm first of all we set the architecture of the network. In the deep learning implementation, there are three layers. Those are the input layer, the hidden layer, and the output layer. In the following section, we tried to explain the deep learning architecture that is applied in the proposed system.

4.9.1.1 The Input Layer

In the development of proposed the neural network is composed of input neuron and provide initial data to the system for further processing by the next layer of the system. The input layer of neurons is specified by the embedding data or the text data that converted into equivalent vectors. We specified the number of input layers neurons based on the length of the train data that converted into vectors. That means we have used the number of neurons at the input layer equals to the length of the vector.

4.9.1.2 The Hidden Layer

The hidden layer is found between the input layers and the output layer. This layer contains a function that applies weights to the inputs and directs them through an activation function as the output. At this layer, two operations were performed(TIROR, 2021). The first operation is a multiplication of weights and attributes and summation of all. The second operation is the application of the activation function to generate the expected output. We specify the number of neurons at this layer doing experiments by changing the numbers of neurons until we got a good result.

4.9.1.3 The Output Layer

The purpose of this layer is to generate the output of the process and checks the actual and desired output. If the difference between actual and desired output is very high output layer returns the process backward and makes it continuous (TIROR, 2021).

4.9.2 Deep Neural Network Implementation

The proposed system was implemented by using Keras and tflearn frameworks or libraries. The following snapshot code indicating the development proposed model by using tflearn frameworks DNN model.

Code 4. 9 DNN implementation by using tflearn framework

```
tf.reset_default_graph() #Reset Underlying Graph data
#Building our own Neural Network
net = tflearn.input_data(shape=[None, len(train_x[0])])
net = tflearn.fully_connected(net, 64)
net = tflearn.fully_connected(net, 64)
net = tflearn.fully_connected(net, len(train_y[0]), activation='softmax')
net = tflearn.regression(net)
#Defining Model and setting up tensorboard
model = tflearn.DNN(net, tensorboard_dir="tflearn_logs")
#Now we have setup model, now we need to train that model by fitting data into it by model.fit()
#n_epoch is the number of times that model will se our data during training
model.fit(train_x, train_y, n_epoch=100, batch_size=8, show_metric=True)
```

4.9.3 Bi-LSTM Model Implementation

By using Keras framework, we implement Bi-LSTM model to develop the proposed system. The following snapshot code the implementation of a Bi-LSTM model by using the Keras library.

Code 4. 10 Implementation of Bi-LSTM model

```
model2 = Sequential()

model2.add(layers.Embedding(max_words, 0, input_length=max_len))

model2.add(layers.Bidirectional(layers.LSTM(64)))

model2.add(layers.Dense(64))
```

```
model2.add(layers.Dense(92,activation='softmax'))

model2.compile(optimizer='adam',loss='categorical_crossentropy', metrics=['accuracy',
keras.metrics.Precision(), keras.metrics.Recall()])

#Implementing model checkpoins to save the best metric and do not lose it on training.

checkpoint2 = ModelCheckpoint("best_model2.hdf5", monitor='val_accuracy',
verbose=1,save_best_only=True, mode='auto', period=1,save_weights_only=False)

history = model2.fit(X_train, Y_train, epochs=60,batch_size=8, validation_data=(X_test,
Y_test),callbacks=[checkpoint2]) model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy', keras.metrics.Precision(),
keras.metrics.Recall(),keras.metrics.TopKCategoricalAccuracy(k=5)])
cp_callback = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path,save_best_only=True,
save_weights_only=True, verbose=0)
history = model.fit(X_train, y_train, epochs=100, batch_size=8, validation_data=(X_test,
y_test),callbacks=[cp_callback],verbose=1)
scores = model.evaluate(X_test, y_test, verbose=1)
model.save("model.h5")
```

4.9.4 Saving and Loading Model Implementation

After we train the model, we saved it to using the model for the next time without train again. The following source code shows how we save the model to use it.

```
import pickle
#Dumping training data by using dump() and writing it into training_data in binary mode
pickle.dump({"words":words, "classes":classes, "train_x":train_x,
"train_y":train_y,"X_train":X_train, "X_test":X_test, "y_train":y_train, "y_test":y_test},
open("training_data1", "wb"))
model.save(model.h5')
```

After we saved the model, we load the model to evaluate and to use it again. The following source code shows how we load saved model.

```
data = pickle.load(open("training_data1","rb"))
```

```
words = data['words']
classes = data['classes']
train_x = data['train_x']
train_y = data['train_y']
X_train = data['X_train']
X_test = data['X_test']
y_train = data['y_train']
y_test = data['y_test']
model=tf.keras.models.load_model("Chat.h5")
```

4.9.5 Intent Classification Implementation

After we load the trained model, we used it to classify the users' request intent and provide the response from the specific intent. The following code shows how we implement the proposed system to classify the users' queries and provide the response for the users'.

Code 4. 11 Intent classification implementation

```
ERROR_THRESHOLD=0.5
results=model.predict(input_datas)[0]
   #Filter out predictions below a threshold
   results = [[i,r] for i,r in enumerate(results) if r>ERROR_THRESHOLD]
   #Sorting by strength of probability
   results.sort(key=lambda x: x[1], reverse=True)
   return_list = []
   for r in results:
      return_list.append((classes[r[0]], r[1]))
if results==[]:
      if re.search('[a-zA-Z]', sentence):
         return "Maal akka jechuu barbaadan naaf hin gale"
      else:
         return " ምን ማለት እንደፈለጉ  አልገባኝም"
   #If we have a classification then find the matching intent tag
   if results:
      #Loop as long as there are matches to process
      while results:
```

```
    for i in intents['intents']:
        #Find a tag matching the first result
        if i['tag'] == results[0][0]:
            #A random response from the intent
            return random.choice(i['responses'])
```

CHAPTER FIVE

5. RESULT AND DISCUSSION

5.1 Chapter Overview

In this chapter, we discussed the overall results that the model achieved based on performance metrics quantitatively and the performance for a realistic environment is discussed. In addition to this in this chapter, we explain the designed model overfitting are applied and the result that we get after we applied model overfitting by using train and test loss curve graph with the discussion of the effect of the overfitting techniques on the result. Finally, we discussed on how this study was answered the research questions and what makes this study differ when we compare it with the research done before.

5.2 Model Evaluation Result

As we discussed in model selection techniques (section 3.7), we implement the proposed system by using DNN and Bi-LSTM model. After we implement the model, we select the model that gives the best accuracy. To do this we evaluate the model by using accuracy first, we split the dataset into training data and evaluation data. By using the evaluation data, we evaluate the proposed model as the following.

5.2.1 DNN Model Evaluation Result

To implement the DNN model we used the tflearn library that contains different classes and methods. We used the hyperparameter of the model as we explained in the implementation section. The proposed has an input layer, which contains the number of neurons that equals the length of the input data vector, has two hidden layers that contain 64 neurons for each hidden layer, and it contains the output layer that contains the number of neurons equal to the number of tag or intents. After designing the model like this by using the tflearn framework, we evaluate the model to get the accuracy of the model on the given data. In this framework, we can evaluate the model by using an accuracy metric. Accuracy is the degree of closeness of measurements of a quantity to that quantity's true value. That means finding the data that has been predicted correctly into their appropriate class by the trained model based on the training data set. To get this accuracy we used the following code

Code 5. 1 DNN model calculation implementation

```
import numpy
predictions = model.predict(X_test)
accuracy = 0
for prediction, actual in zip(predictions, y_test):
   predicted_class = numpy.argmax(prediction)
```

```
    actual_class = numpy.argmax(actual)
    if(predicted_class == actual_class):
        accuracy+=1
accuracy = accuracy / len(y_test)
```

The above code calculated the data that correctly predicted or classified into their target classes. The X_test and y_test are the data that we split our dataset as validation data. By using this calculation, we get the accuracy of the DNN model. When we evaluate the model by using this metric, we got 83.52% accuracy.

5.2.2 Bi-LSTM Model Evaluation Result

The Bi- LSTM model is implemented by using the Keras API library. This API has been used to design the model of the layer sequentially. Keras API has different functions that are used to evaluate the model accuracy like Keras.metrics.Precision() used for getting the model precision metric value, and keras.metrics.Recall() function is also used to get the recall metrics of the model.

To use those evaluation metrics we pass those functions in keras.model.compile method as an argument for the metric parameter. We also pass accuracy as the metric argument in model.compile function. The following minimalist snapshot code shows how we pass those arguments to evaluate the model metrics in a Bi-LSTM model.

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy', keras.metrics.Precision(), keras.metrics.Recall()])
```

On the above code, the compile() method takes different parameters. The first parameter is an optimizer that used for optimizing the model value or used for updating the model weight after some change has performed, the second parameter is loss, this also used to find the best parameter or weight for data, the third parameter is the metrics parameter that used for evaluating the performance of the designed model.

For the metrics parameter, we passed three values as an argument. After we pass those arguments when we evaluate the model, we got four values. The first value is the number of losses, the second value is the accuracy of the model, the third value is the precision value of the model and the last value is the recall value of the model. After we get precision and recall, we will get the value of the F1- score by using equation 3.5.

After we train the model, we get the following result for each model metric.

Table 5. 1 BI LSTM model evaluation result

Metrics Name	Result in percent
Accuracy	85.23
F1-score	91.23
Recall	85.25
Precision	98.11

The other one is if we used deep learning, the big challenge is it asks to prepare more data otherwise, the model overfitting problem will appear. Overfitting is when the model fits more data than its needs, its starts catching the noisy data and inaccurate values in the data. As a result, the efficiency and accuracy of the model decrease(*(885) Overfitting in Machine Learning | Python Tutorial | Machine Learning Tutorial | Edureka - YouTube*, n.d.).

To check the model overfitting of the model we used a matplot graph that shows the training loss for each epoch of the model and testing loss. The graph has generated based on the value of loss that we got in each epoch while we train the model. The following graph shows training loss and test loss across an epoch of the model generated based on the history that we get at the training time.

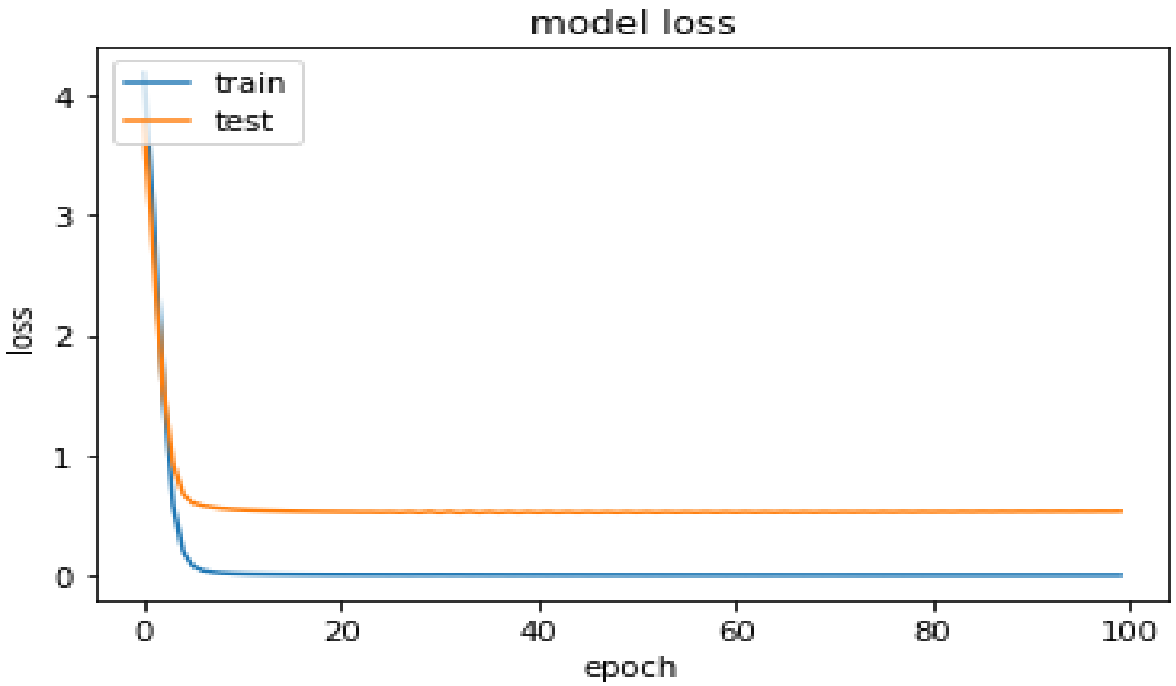


Figure 5. 1 Model training and validation loss across epochs before regularization

From the above graph, we see the training graph history the training loss and test loss curve shows that the model has over fitted. To overcome this problem we used L2 regularization techniques as the following snapshot code.

```
model2 = Sequential()

model2.add(layers.Embedding(max_words, 40, input_length=max_len))

model2.add(layers.Bidirectional(layers.LSTM(64)))

model2.add(layers.Dense((32)))

model2.add(layers.Dense(92,activation='softmax',kernel_regularizer=l2(0.5)))model2.compile(optim
izer='adam',loss='categorical_crossentropy', metrics=['accuracy', keras.metrics.Precision(),
keras.metrics.Recall()])checkpoint2 = ModelCheckpoint("best_model2.hdf5",
monitor='val_accuracy', verbose=1,save_best_only=True, mode='auto',
period=1,save_weights_only=False)

history = model2.fit(X_train, Y_train, epochs=40,batch_size=8, validation_data=(X_test,
Y_test),callbacks=[checkpoint2])

model2.save("chat.h5")
```

5.2.3 Bi-LSTM Model Evaluation Result after Regularization

Applying the regularization technique was affect the result that we got from the model that does not regularize. After we applied the regularization technique, we evaluated the model and we got the following result.

Table 5. 2 Bi-LSTM model evaluation result after regularization

Metrics Name	Result in percent
Accuracy	82.6
F1-score	87.7
Recall	82.6
Precision	85.7

The training loss and test loss of the model also changed after we applied L2 regularization techniques. The following graph shows that the result of train loss and test loss of the model.

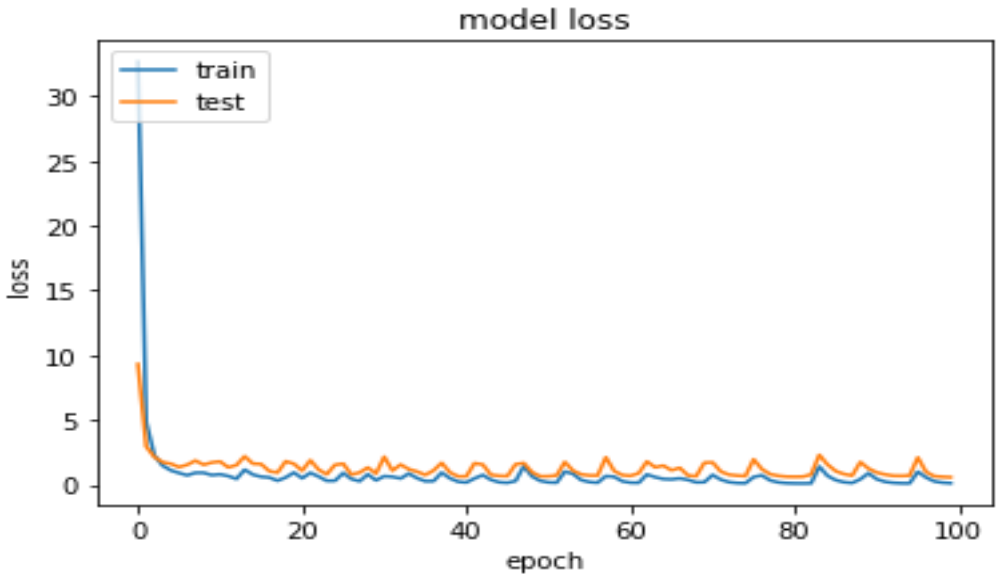


Figure 5. 2 Model training and validation loss across epochs after regularization

As we have seen on the above graph, the training loss and test loss has fitted well. This indicates that the proposed model has designed well based on our dataset.

5.2.3 Tuning Hyperparameters

When we used deep learning techniques the input layers, hidden layers, and output layers process used different parameters during model training. In the proposed study, we used different parameters. Those are Activation function, Batch size, Epoch, Loss, Number of neurons in each layer, Optimizer. However, there is automatic hyperparameter tuning mechanism is there we tune those parameters manually through trials of experiments.

- **Activation function:** is used to make an artificial neuron receives inputs (real numbers) from other neurons, calculate a weighted sum of it, and apply a non-linear activation function to the result, producing its output(Wang et al., 2020). As we discussed in section 3.6.1 we used the softmax activation function for the proposed system implementation.
- **Batch size: -** is a hyperparameter that is used to specify the number of samples to work through before updating the internal model parameters. Batch size is the size of data taken by the model during training per epoch(TIROR, 2021).
- **Epoch: -** a hyperparameter that defines the number of times that the learning model will work through the entire training dataset.
- **Loss: -** is used to optimize the hyperparameter values in the neural network. This loss function is used to evaluate how the model accomplishes the task that the network is intended to

do(TIROR, 2021). In the proposed system implementation, we used the Categorical cross-entropy function.

- ☞ **Optimizer: -** is hyperparameters that are used to change the weights and learning rate to reduce the losses. For the proposed system implementation, we used Adam optimizer. we select this optimizer because it simplifies the learning rate and has fast convergence properties(TIROR, 2021).

The following table indicate the Bi-LSTM model parameter tuning trial experiments.

Table 5. 3 Hyperparameter tuning trials

# Trials	Epoch	Batch Size	Activation function	Optimizer	Number of hidden layers
1	30	64	softmax	Adam	1
2	40	64	softmax	Adam	1
3	60	32	softmax	Adam	1
4	60	32	softmax	Adam	2
5	80	8	softmax	Adam	2
6	100	32	softmax	Adam	1
7	100	8	softmax	Adam	1
8	100	8	softmax	Adam	2
9	120	8	softmax	Adam	2

While we are doing those trials, we got different results of F1 score and accuracy. The following graph indicate the results that we got at each trial.

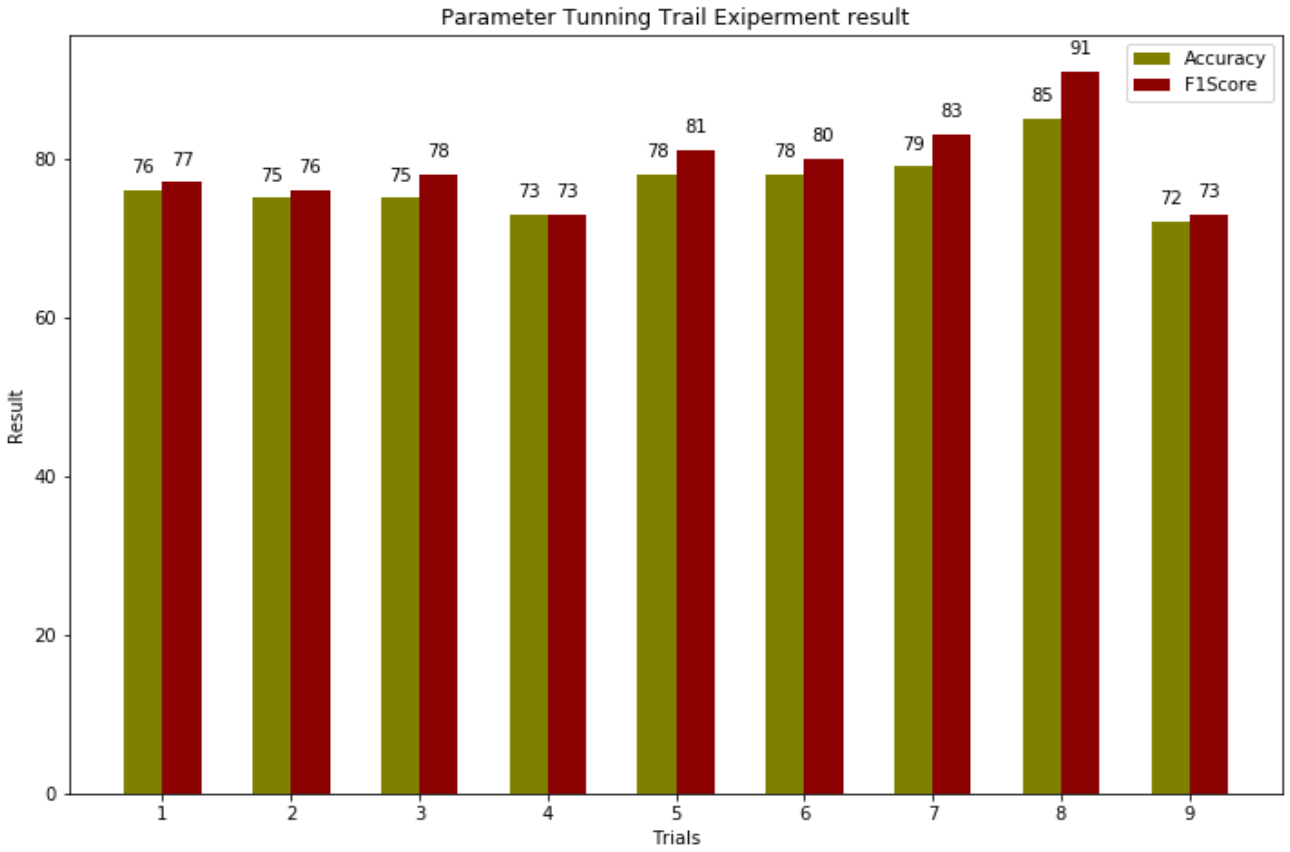


Figure 5. 3 Parameter tuning evaluation result at different trials

5.3 Human Evaluation

As we discussed the evaluation techniques in section 3.8, the proposed system is also evaluated by users. To perform this, we select Ethio telecom customers to use the proposed system and evaluate the system based on the model responses.

To evaluate the model through human evaluation we integrate the model to the web and telegram bot. By using this system, the users did text-to-text chat. The following figure indicates the text-to-text chat after we integrate the model with flask and telegram bot.

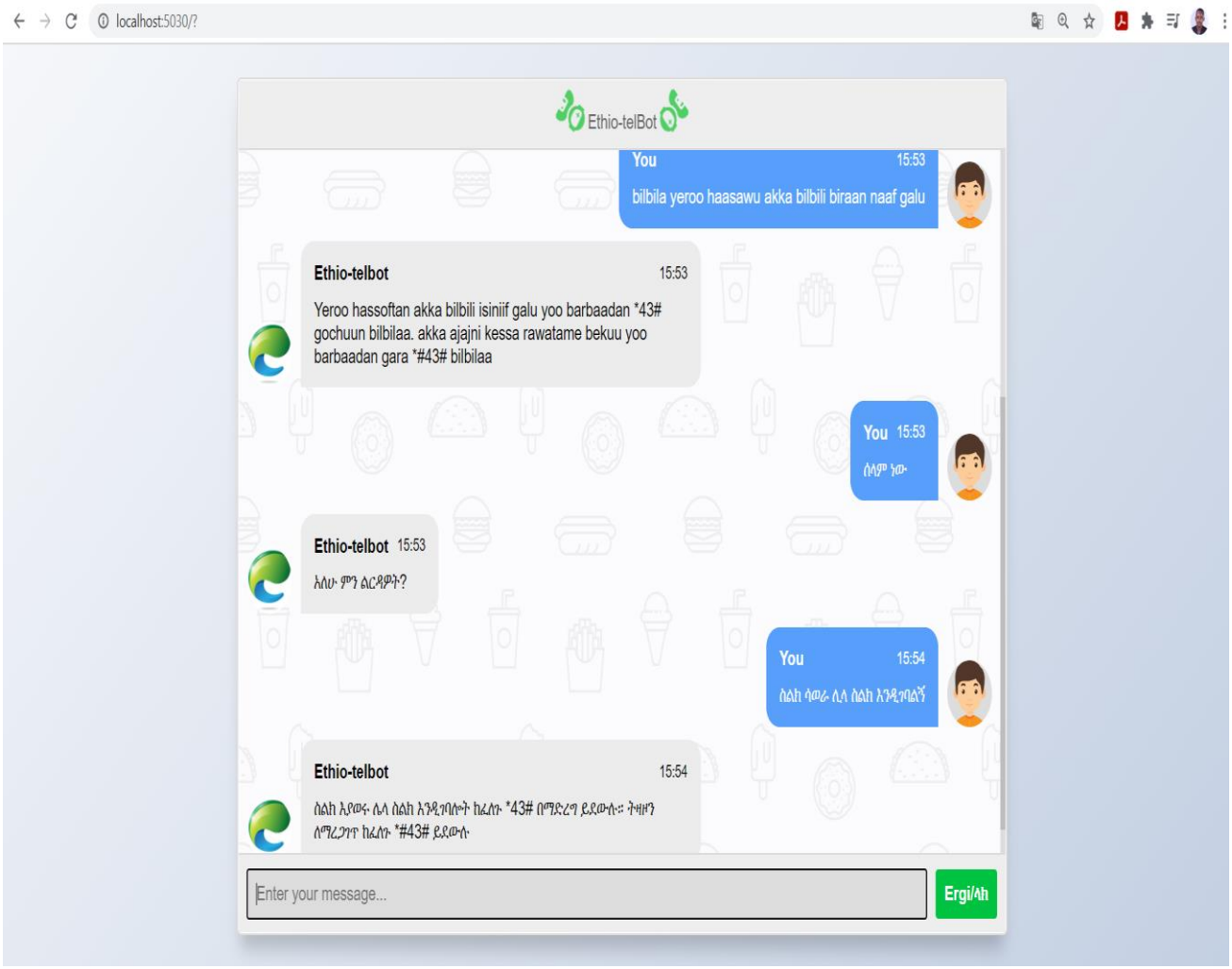


Figure 5. 4 Flask Web GUI Bilingual chatbot

The above figure indicates the text-to-text conversation between the user and the bot. As we have seen on the above figure, the system gave the response to the user by using both Amharic and Afaan Oromo text. This graphical user interface has designed by using the Flask framework.

We also integrated the model with a telegram bot. This makes it simple to evaluate the model because the users accessing the system from anywhere and they evaluate the model for us easily. The following picture shows the bilingual chat history performed by using a telegram bot.

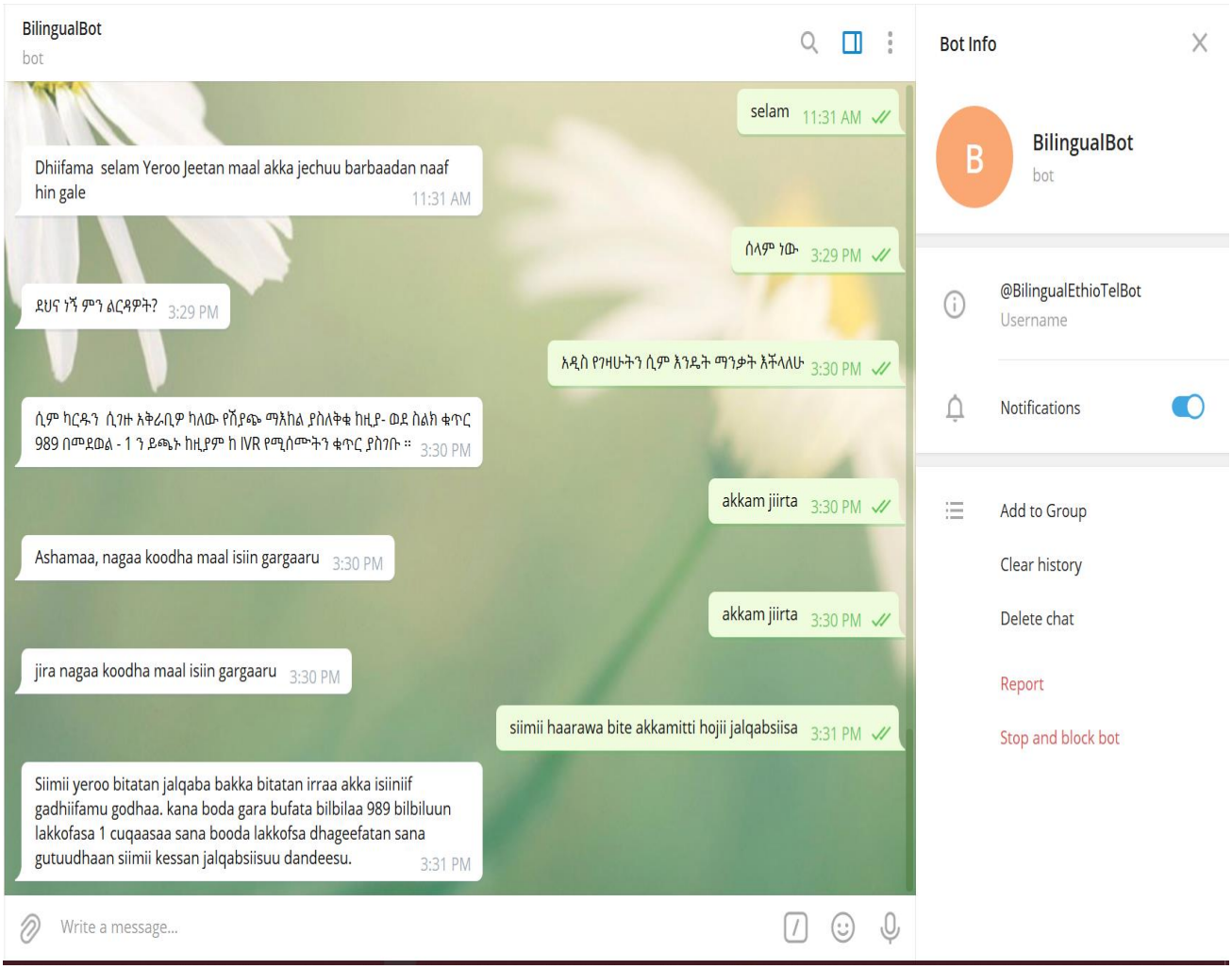


Figure 5. 5 Bilingual chatbot on telegram

After that, the user was evaluated the model by using the evaluation paper that we prepared in appendix B.1 and B.2. The users evaluated the proposed system for two criteria. The first evaluation is the performance of the model. This criterion has based on the number of the users' queries that were answered correctly by the proposed system. The second human evaluation criterion is the acceptance of the user. This criterion is based on the users' satisfaction with the proposed system. To do this we take the average of thirty persons based on the language that was used to test the system. Those persons have selected randomly based on the language that they understood. Out of those people, the first ten persons evaluate the model by using Afaan Oromo language only, the second ten persons evaluated by using Amharic language, and the last ten persons evaluated the system by using both Amharic and Afaan Oromo language. The following table indicates the result of the human evaluation of the system performance.

Table 5. 4 The human evaluation results on system performance

Number of persons	Language used	Total number of asked questions	Total number of correctly answered questions	Correctness in %
10	Afaan Oromo	35	22	62.86
10	Amharic	37	26	70.27
10	Both language	53	40	75.47
Total		125	88	70.4

As we have seen in the above table, the performance of the system has been evaluated by the user based on the number of questions that users are provided for the system and the total number of questions that the system is answered correctly. To calculate the correctness we used the following formula.

$$Correctness = \frac{total\ number\ of\ question\ answered\ correctly}{total\ number\ of\ question\ requested\ by\ users} * 100$$

Equation 5. 1 Human performance evaluation result calculation

Based on the result that we gathered from the users, the system was given the response to the users' requests more correctly. This indicated that the system understood the users' queries well.

After we evaluated the model performance based on correctly answered questions, we also evaluate the proposed system user's acceptability. To do this we evaluate the model based on five parameters. Those are -

- **Response time**: - this is based on the time interval between the users' questions for the system and the provided response by the system.
- **Efficiency**: - this is based on the system's correctness to give the efficient answer to provided questions.
- **Attractiveness**: - This criterion is based on the system attractiveness when the user has used the system
- **User-friendly**: - this parameter is used to evaluate the simplicity of the proposed system while the users' used it.
- **Feasibility**: - This parameter is used to identify the feasibility of the proposed system if the system was deployed for the users.

Based on those parameters, the users have put the value Excellent (4), very good (3), good (2), low (1), and very low (0). After we collect those results from the users, we take the average of each

parameter to identify the system acceptability by the users. The following result shows the system acceptance of users' results based on the above parameters.

Table 5. 5 The users' acceptance evaluation result

Parameters	30 person system evaluation result					
	Excellent	Very good	Good	Low	Very low	Total in percent
Response time	26	4				96.7
Efficiency	3	19	5		3	65.8
Attractiveness	3	20	7			71.7
User friendly	2	19	8	1		68.3
Feasibility	18	11	1			89.2
Total average						78.34

As we have seen in the above table, we evaluate the proposed model based on five parameters by users. We select those parameters based on(Habitamu Asimare, 2020). Each parameter was evaluated by 30 users. The users filled the evaluation for each parameter out of four. Therefore, each parameter has been evaluated totally out of 120, because we evaluate each parameter by 30 users. We calculate user acceptance of the proposed system based on those parameters based on the following formula.

$$Acceptance = \frac{\sum_{n=1}^{5} number\ of\ users\ select\ value * criterea\ value}{120} * 100$$

Generally, we got a total user acceptance of 78.34% based on those five parameters. This shows that the users are satisfied with the proposed system or the model has good acceptance.

5.4 Comparison with Existing Models

As we discussed in section 2.12 different researchers have done different research on chatbot development. In this section, we tried to compare our study with the researches that have more related to our work based on the language that we used for this study and the language that used their studies.

The first research that we select to compare with our work done by using the Amharic language. In (Habitamu Asimare, 2020), the aim of the researcher developing an Amharic chatbot for assisting the users on Ethiopian law. To do this, the researcher was used the RNN model. In addition, for model overfitting handling techniques the researcher has applied to change the dataset splitting ratio technique only. When we compare this study with ours, in our study, we applied DNN and Bi-LSTM model, and to handle overfitting we applied the L2 regularization technique. This technique is better than changing the dataset splitting ratio technique. The other one is the researcher has developed a

chatbot that supports only the Amharic language but our study has done on both Amharic and Afaan Oromo languages. This makes our study better than the (Habitamu Asimare, 2020) study based on those points.

The other study that related to our work is the study on developing Afaan Oromo text-based chatbot for enhancing maize productivity(Bedasa, 2021). The researcher has used the Afaan Oromo language to developing a chatbot. To develop the system the researcher was used the DNN model. When we compare our study with (Bedasa, 2021), our study was better on three things. The first one is our proposed system supported both the Amharic and Afaan Oromo languages. The second is on (Bedasa, 2021) study the researcher doesn't apply any techniques to handle overfitting but we used the L2 regularization technique. The third point is the researcher does not develop the prototype for their proposed work but by using flask we developed a web application to integrate the model and also we integrate the trained model with a telegram bot to make it easy to use. The fourth one is our study is used more dataset than (Bedasa, 2021).

5.5 Discussion

As we discussed in section 1.4.1 the objective of this study developing a bilingual chatbot to assist the Ethio telecom customers concerning customer services. In addition to achieving this objective, this research was able to answer the research question listed in section 1.4. In this section, we tried to discuss how this study was answered those research questions. To answer those questions we used different techniques. Let us see the techniques that have been applied to answer the research questions and how the proposed system answered those questions one by one.

The first research question of this study is "How do we design the model that understands the user's dialogue efficiently and produces an appropriate answer (response) for the users by using Amharic and Afaan Oromo languages?" to answer this question we applied different techniques. From those one building the chatbot dataset by using both Amharic and Afaan Oromo language. After we build our dataset, we study the challenges related to those languages. We applied different techniques that have been discussed in chapter three to overcome those challenges. Finally, we able to develop a chatbot that understands both Amharic and Afaan Oromo languages.

The second research question of this study is "How we develop a user-friendly text-to-text chat interface to engage the deigned model with the Ethio telecom customers?" to answer this question we also used different techniques. The first technique is to train the model by using the dataset that we build on Ethio telecom customer services. After we train the model, we saved it as binary data. After

that, we load that trained model and integrate it to the web and telegram bot. This makes the user access the system by using either a web or a telegram bot to make text-to-text conversations.

The last research question of this study is “how we test the proposed model?” this question deals after we developed the proposed model, we have to evaluate the model as it was given the correct response or not. The other evaluation is done after we developed the proposed system to identify is the system was acceptable by the user or not. To answer this question we applied two techniques. The first technique is applying the evaluation by splitting the dataset into training data and validation data. After we split those data, we train the model by using train data and we evaluate by validation data. We evaluate the model by using F1-score, precision, recall, and accuracy.

The other technique that we applied to the proposed system evaluation is to evaluate the system by users. After we design the web and telegram bot, the users have accessed the system by using either of two options and then evaluate the system. The users' evaluations have been used to measure two things. The first thing is the performance of the system and it is based on the number the request that the users are provided for the system and the number respond that are returned correctly by the system. The second thing is we evaluated the system, by the user to identify the users' acceptance of the system based on different five parameters.

CHAPTER SIX

6. CONCLUSION, RECOMMENDATION, AND FUTURE WORK

6.1 Chapter Overview

This is the last chapter of our study. This chapter discussed three different points. The first point is the conclusion part, which deals with the summarization of our work. In this part, we discussed the overall action that taken to accomplish this work and the result of the study. The second point has discussed the recommendations for different parts. The third part of this chapter deals with the plan that we need to add some features for the future to this work for making it more effective.

6.2 Conclusion

To start from the objective of this study, the general objective of this study is to design and develop a bilingual chatbot for assisting Ethio telecom customers with customer services. To do this we passed through a different procedure. In this section, we try to discuss the overall process that has taken while our study and the final result that we got from this study.

A bilingual chatbot means a chatbot that supports two languages. The outcome of this study is developing a chatbot that understands the Amharic and Afaan Oromo languages. To develop this chatbot we used different techniques. The first technique is collecting and building the dataset that has been used to train the model. To collect that dataset we collect from a different source and we build the dataset in the form of JSON. After we build our dataset, we implement the proposed system.

We implement the proposed system by using tflearn DNN and Bi-LSTM models. After we implement those models, we evaluate them by using accuracy metrics. The Bi-LSTM model achieved good accuracy than the tflearn DNN model. Because of this, we select this model for the proposed system development purpose. After selecting this model, we integrate the model to the web by using the flask framework and we integrate it into the telegram bot.

We integrate the model to web and telegram bot for two purposes. The first purpose is to make it easy for the users to access it. The second purpose is in addition to evaluating the models by a metric we also evaluate the model by users. The users evaluate the proposed system for two things identification. That is system performance and system acceptability.

The performance of the model means the ability that the system was answered the question correctly. To evaluate this we prepare the google form questionnaires that ask the users the number of the question that the users have filled. The human evaluation has collected from thirty persons. Those persons have been selected based on the language that is used to make conversation with the system.

Therefore, we select ten users that make conversation by using Amharic, ten users that make conversation by using Afaan Oromo language, and ten persons that make conversation by using both Amharic and Afaan Oromo language. After we gather that information, we find the final average result of human evaluation on the performance of the proposed system. Finally, we got 70.4% of the system was answered the users' questions correctly.

The second human evaluation has been used to identify the users' acceptance of the proposed system. The users' acceptance is evaluating based on five parameters. The first parameter is response time. It is the time interval between the user's request sent and the response returned. The second parameter is efficiency. This has based on the system response correctness. The third parameter is attractiveness. It is based on the satisfaction of users from the developed system. The fourth parameter is user-friendly which is used to identify how the system was easy to use. The last parameter is the feasibility of the proposed system. It is used to identify the users' intention if this system will be deployed. Based on those parameters the users gave different values for each parameter based on their judgment. Those values are Excellent (4), very good (3), good (2), low (1), and very low (0). After we collect the user's evaluation result, we analyze that our proposed system has good acceptance from the users.

6.3 Recommendation

To accomplish this study we pass through different stages. When we pass those stages, we analyzed different ideas that we have to recommend for two different parts. The first part that we need to recommend is the researcher that studies NLP by using Amharic and Afaan Oromo languages. The second part is the Ethio telecom organization.

- ☞ **For the researcher**s: - as we discussed in chapter two, one of the challenges to applying NLP on Afaan Oromo and Amharic language both languages are under-resourced. Therefore, to overcome this, the researcher was able to make their resource such as source code, dataset, papers, etc. that prepared for their study public.
- ☞ **For the organization**: - as we have seen in the model evaluation result, this study got good acceptance from the users. If we change this study into the project, it will minimize the current problems of the organization. We suggest the Ethio telecom organization see this study and convert it into a project.

6.4 Future Work

This study was concerned about the designing and developing of bilingual chatbots to assist Ethio telecom customers in customer services. To do this we applied different techniques for different purposes. However, we applied those techniques and we implemented the model by using two algorithms for the future we need to add some features. For this, we planned to focus on the following tasks.

- ☞ To develop the proposed system we used the tflearn DNN model and Bi-LSTM model. For the future, we need to develop by using other models like CNN, and BERT model.
- ☞ The third plan that we will add for the future is making the system dynamic. That means the current system that we are developed was only focused on the service that currently available services. However, in the future, the organization may announce a new service for the customers. So if the users' needs help on this service this system not handle such request. To overcome this, for the future, we will try to connect the system to the database and if new services will be launched, we simply add a dataset about that service through GUI, and then by click only on the button, we make the system know about new services.
- ☞ However, we applied different data preprocessing techniques; there are also different techniques that we are not applied. For the future, we need to add lemmatization and stemming data preprocessing techniques.

==========

SPECIAL ACKNOWLEDGMENT

This research work was funded by Adama Science and Technology university under grant number:

ASTU/SM-R/231/21

Adama , Ethiopia

References

(885) Overfitting in Machine Learning | Python Tutorial | Machine Learning Tutorial | Edureka - YouTube. (n.d.). Retrieved April 23, 2021, from https://www.youtube.com/watch?v=Om1SeVpBiiQ

7 Types of Activation Functions in Neural Networks: How to Choose? (n.d.). Retrieved March 13, 2021, from https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/

A guide to the types of machine learning algorithms | SAS UK. (n.d.). Retrieved March 15, 2021, from https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html

A Step-by-Step Guide to Building a Basic Dialogflow Chatbot | PDF.co. (n.d.). Retrieved March 4, 2021, from https://pdf.co/blog/building-a-basic-dialogflow-chatbot

Abdu, B. (2018). *Ethio-telecom service*. Reporter. https://www.thereporterethiopia.com/article/telecom-ceo-replaced-his-deputy

Abera Hordofa, B., & Dechasa Degefa, S. (2021). A Review of Natural Language Processing Techniques: Application to Afan Oromo. *International Journal of Computer Applications Technology and Research*, *10*(03), 051–054. https://doi.org/10.7753/ijcatr1003.1001

Almansor, E. H., & Hussain, F. K. (2020). Survey on Intelligent Chatbots: State-of-the-Art and Future Research Directions. *Advances in Intelligent Systems and Computing*, *993*(January), 534–543. https://doi.org/10.1007/978-3-030-22354-0_47

Athota, L., Shukla, V. K., Pandey, N., & Rana, A. (2020). Chatbot for Healthcare System Using Artificial Intelligence. *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, *5*(3), 619–622. https://doi.org/10.1109/ICRITO48877.2020.9197833

Bedasa, S. (2021). *Developing Afaan Oromoo Text based Chatbot for Enhancing Maize Productivity*. Adama Science and Technology University.

Bisht, A., Doshi, G., Arora, B., & Pansambal, S. (2020). *Multilingual Chatbot With Human Conversational Ability*. *13*(1), 138–146.

Chaitrali, K., Amruta, B., Savita, P., & Prof. Satish, K. (2017). BANK CHAT BOT – An Intelligent Assistant System Using NLP and Machine Learning. *International Research Journal of Engineering and Technology (IRJET)*, *04*(05), 4.

Chung, K., & Park, R. C. (2019). Chatbot-based heathcare service with a knowledge base for cloud computing. *Cluster Computing*, *22*, 1925–1937. https://doi.org/10.1007/s10586-018-2334-5

Colace, F., De Santo, M., Lombardi, M., Pascale, F., Pietrosanto, A., & Lemma, S. (2018). Chatbot for e-learning: A case of study. *International Journal of Mechanical Engineering and Robotics Research*, *7*(5), 528–533. https://doi.org/10.18178/ijmerr.7.5.528-533

Comendador, B. E. V., Francisco, B. M. B., Medenilla, J. S., Nacion, S. M. T., & Serac, T. B. E. (2015). Pharmabot: A Pediatric Generic Medicine Consultant Chatbot. *Journal of Automation and Control Engineering*, *3*(2), 137–140. https://doi.org/10.12720/joace.3.2.137-140

Consulting, V. (2017). *Chatbots Chatbots 101 by V-Soft Consulting*. 18. https://cdn2.hubspot.net/hubfs/1629777/Chatbots 101.pdf?t=1526138378633

Cox, G. (2019). *ChatterBot Documentation*.

Cui, L., Huang, S., Wei, F., Tan, C., Duan, C., & Zhou, M. (2017). Superagent: A customer service chatbot for E-commerce websites. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of System Demonstrations*, 97–102. https://doi.org/10.18653/v1/P17-4017

Ethiotelecom. (n.d.). *History of Ethio-telecom*. Retrieved December 28, 2020, from https://www.ethiotelecom.et/history/

Fainchtein, L. (2020). *No Title*. https://blog.cloudboost.io/generative-vs-retrieval-based-chatbots-a-quick-guide-8d19edb1d645

Fita, C. (2016). *Afaan Oromo List, Definition and Description Question Answering System*.

Framework Definition. (n.d.). Retrieved March 3, 2021, from https://techterms.com/definition/framework

Gao, B., & Pavel, L. (2017). On the properties of the softmax function with application in game theory and reinforcement learning. In *arXiv*. arXiv.

Gonda, D. E., Kong, H., Chu, B., & Kong, H. (n.d.). Creating conversational bots as a supplement for teaching assistant training course. *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD*.

Gruber, T. (2009). Siri a virtual Personal assistant: Bringing Intelligence to the Interface. *Semantic Technologies 2009*. http://www.macrumors.com/2010/04/28/siri-acquisition-brings-apple-

much-closer-to-the-knowledge-navigator-concept/

Gurmessa, D. K. (2020). *Afaan Oromo Text Content-Based Fake News Detection using Multinomial Naive Bayes*. *01*, 26–37.

Habitamu Asimare. (2020). *Designing and Implementing Adaptive Bot Model to Consult Ethiopian Published Laws Using Ensemble Architecture with Rules Integrated*. BAHIR DAR UNIVERSITY.

Hard, M. M. (2016). From Eliza to Alexa: Chatbots and the Future of Customer Experience (Whitepaper). *MindTouch*. https://mindtouch.com/info/wp-content/uploads/sites/5/2016/09/mtwhitepaper_chatbots.pdf

Hiremath, G., Aishwarya, H., Priyanka, B., & Nanaware, R. (2020). Chatbot for education system. *International Journal of Advance Research, Ideas and Innovations in Technology*, *4*(3), 37–43.

Introduction to AIML. (2018). https://www.tutorialspoint.com/aiml/aiml_introduction.htm

Kassie, T. (2009). *WORD SENSE DISAMBIGUATION FOR AMHARIC TEXT RETRIEVAL : A CASE STUDY FOR LEGAL*. Angeles, L., Advocacy, S., Location, O. (2002).

Kelemework, W. (2013). Automatic Amharic text news classification: Aneural networks approach. *Ethiopian Journal of Science and Technology*, *6*(2), 127–137.

Khare, J. (2019). *applications of natural language processing*. https://www.upgrad.com/blog/5-applications-of-natural-language-processing-for-businesses/

Kibble, R. (2013). Introduction to natural language processing Undergraduate study in Computing and related programmes. *Roeper Review*, *1*(2), 26.

Machine Learning Model Regularization in Practice: an example with Keras and TensorFlow 2.0 | by B. Chen | Towards Data Science. (n.d.). Retrieved May 17, 2021, from https://towardsdatascience.com/machine-learning-model-regularization-in-practice-an-example-with-keras-and-tensorflow-2-0-52a96746123e

Machiraju, S., & Modi, R. (2018). Developing Bots with Microsoft Bots Framework. *Developing Bots with Microsoft Bots Framework*. https://doi.org/10.1007/978-1-4842-3312-2

McAllister, P., Kerr, J., McTear, M., Mulvenna, M., Bond, R., Kirby, K., Morning, J., & Glover, D. (2020). Towards Chatbots to Support Bibliotherapy Preparation and Delivery. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture*

Notes in Bioinformatics): Vol. 11970 LNCS. https://doi.org/10.1007/978-3-030-39540-7_9

McDonald, D. D. (2010). Natural language generation. *Handbook of Natural Language Processing, Second Edition*, *March*, 121–144. https://doi.org/10.2307/414915

Mindaye, T., & Atnafu, S. (2009). Design and implementation of Amharic search engine. *Proceedings - 5th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2009*, *January*, 318–325. https://doi.org/10.1109/SITIS.2009.58

Miškuf, M., & Zolotová, I. (2016). Comparison between multi-class classifiers and deep learning with focus on industry 4.0. *2016 Cybernetics and Informatics, K and I 2016 - Proceedings of the 28th International Conference*. https://doi.org/10.1109/CYBERI.2016.7438633

Navigli, R. (2018). Natural language understanding: Instructions for (present and future) use. *IJCAI International Joint Conference on Artificial Intelligence*, *2018-July*, 5697–5702. https://doi.org/10.24963/ijcai.2018/812

Rahman, M. M., Amin, R., Khan Liton, M. N., & Hossain, N. (2019). Disha: An implementation of machine learning based bangla healthcare chatbot. *2019 22nd International Conference on Computer and Information Technology, ICCIT 2019*, 1–6. https://doi.org/10.1109/ICCIT48885.2019.9038579

Shangrapawar, A., Ravekar, A., Kale, S., Kumari, N., & Shende, A. (2020). *Artificial Intelligence based Healthcare Chatbot System*. 2019–2021.

Shawar, B. A., & Atwell, E. (2015). ALICE chatbot: Trials and outputs. *Computacion y Sistemas*, *19*(4), 625–632. https://doi.org/10.13053/CyS-19-4-2326

Siangchin, N., & Samanchuen, T. (2019). Chatbot Implementation for ICD-10 Recommendation System. *2019 International Conference on Engineering, Science, and Industrial Applications, ICESI 2019*, 1–6. https://doi.org/10.1109/ICESI.2019.8863009

Srivastava, P., & Singh, N. (2020). Automatized Medical Chatbot (Medibot). *2020 International Conference on Power Electronics and IoT Applications in Renewable Energy and Its Control, PARC 2020*, 351–354. https://doi.org/10.1109/PARC49193.2020.236624

Suhel, S. F., Shukla, V. K., Vyas, S., & Mishra, V. P. (2020). Conversation to Automation in Banking through Chatbot Using Artificial Machine Intelligence Language. *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, 611–618. https://doi.org/10.1109/ICRITO48877.2020.9197825

Tesfaye, D. (2010). *Designing a Stemmer for Afaan Oromo Text : A Hybrid Approach*.

The Python Tutorial — Python 3.9.2 documentation. (n.d.). Retrieved March 17, 2021, from https://docs.python.org/3/tutorial/index.html

Thorat, S. A., & Jadhav, V. (2020). A Review on Implementation Issues of Rule-based Chatbot Systems. *SSRN Electronic Journal*, *Icicc*, 1–6. https://doi.org/10.2139/ssrn.3567047

TIROR, A. S. (2021). *CHICKPEA GENOTYPE RECOMMENDATION AND YIELD PREDICTION USING MACHINE LEARNING*. Adama Science and Technology University.

TutorialsPoint. (2020). *No Title*. https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_natural_language_processing.htm

Vanjani, M., Aiken, M., & Park, M. (2019). Chatbots for Multilingual Conversations. *Journal of Management Science and Business Intelligence*, *9264*, 19–24. https://doi.org/10.5281/zenodo.3264011

Wallace, R. S. (2009). El programa Artificial Linguistic Internet Computer Entity (A.L.I.C.E.). *Parsing the Turing Test*, 181–210. https://freeshell.de/~chali/programowanie/Anatomy_of_ALICE.pdf

Wang, Y., Li, Y., Song, Y., & Rong, X. (2020). The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences (Switzerland)*, *10*(5). https://doi.org/10.3390/app10051897

Weizenbaum, J. (1966). ELIZA-A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36–45. https://doi.org/10.1145/365153.365168

Why is removing stop words not always a good idea | by Lima Vallantin | Medium. (n.d.). Retrieved March 11, 2021, from https://medium.com/@limavallantin/why-is-removing-stop-words-not-always-a-good-idea-c8d35bd77214

Wikipedia. (2020). *Afaan Oromo language*. https://en.wikipedia.org/wiki/Oromo_language

Wimsatt, A., & Wynn, R. (2011). *Amharic Language and Culture Manual*.

Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, *1168*(2). https://doi.org/10.1088/1742-6596/1168/2/022022

Zayegh, A., & Al Bassam, N. (2018). Neural Network Principles and Applications. *Digital Systems*. https://doi.org/10.5772/intechopen.80416

Zhang, X., Agarwal, S., Choy, R., Wong, K. J., Lim, L., Lee, Y. Y., & Lu, J. J. (2020). Personalized Digital Customer Services for Consumer Banking Call Centre using Neural Networks. *Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN48605.2020.9206709

APPENDIX

Appendix A: List of stop words

Appendix A.1: List of Afaan Oromo stop words(Fita, 2016)

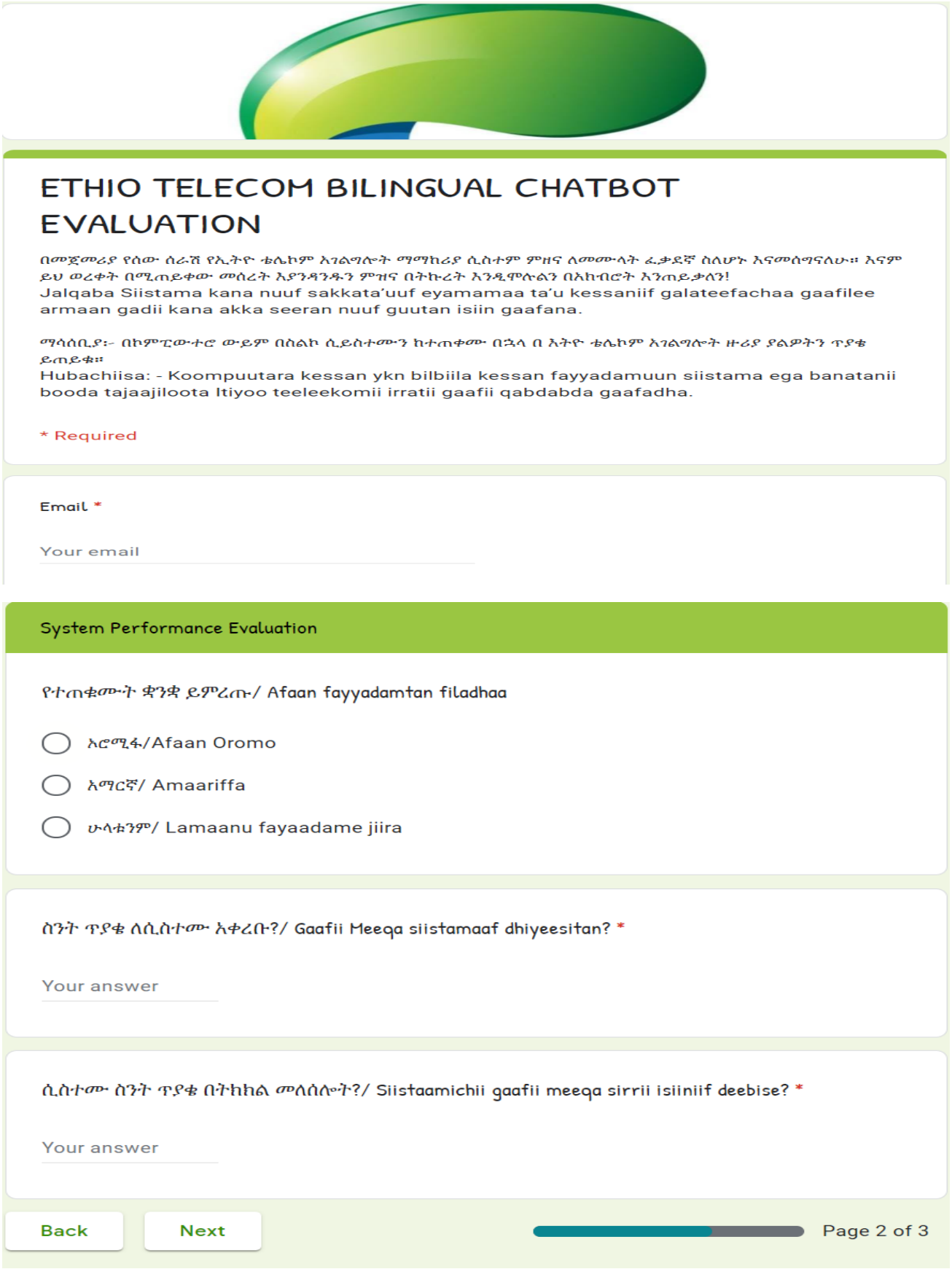
aanee	tanaafuu
gidduu	ati
itti	isaanirraa
narraa	keessa
akka	taʻullee
gubbaa	bira
ittuu	isatti
natti	keessan
hanga	teenya
jala	booda
nu	tun
akkasumas	keenya
henna	utuu
jara	booddee
nurraa	iseen
akkuma	keessatti
hogguu	waan
sana	dura
nuti	ishii
ala	kiyya
illee	warra
kan	duuba
siin	ishiif
alatti	koo
immoo	yeroo
kana	eega
silaa	ishiirraa
amma	kun
inni	yommuu
kanaafi	eegasii
kanaaf	isii
sitti	malee
ammo	yoo
irra	fi
kanaafuu	isin
sun	na
an	yookaan
isaa	gama
kee	isiin
tanaaf	naaf
ani	yoom
isaaf	kun
	keenna

Appendix A .2: List of Amharic Stop words

እኔ	በ	እያለ	በኩላ	ሆነ	ወይዘሮ
የእኔ	ለ	ሕዝብ	በአንድ	ለሆነው	ተብሎ
እኔ ራሴ	ጋር	ሀሙስ	የሆኑ	ሰላት	ሳይሆን
እኛ	ስለ	ለመሆኑ	ከአስራ	ብሎ	እንደሆነና
የእኛ	ላይ	ለምንድን	የሆነውን	ከሰላሳ	ከብር
እኛ ራሳችን	መካከል	ሌሎች	መሆኑ	የሚሆኑ	ሆኖም
አንቺ	ወደ	መጽሀፍ	ሌላውን	ላይም	የሌላ
ነህ	በኩል	ማክሰኞ	ከሰባት	የሆናል	ያላቸው
አላችሁ	ወቅት	ምን	ለሌላ	ከነዚህ	ይህንኑ
እርስዎ	ከዚህ በፊት	ሰኞ	አለበት	ያህል	ሆነው
ትፈልጋለህ	በኋላ	ሰው	ሲል	ከሆነና	በስተቀር
ያንተ	ከላይ	ሲሆን	ይሆናሉ	ለሆኑት	ስም
ራስህን	ከታች	ስንት	በሙሉ	እነዚሁ	እንደገና
እራሳችሁ	ከ	ረቡእ	አስራ	እንደሆኑ	የማያንስ
እሱ	ወደ ላይ	ቅዳሜ	ቢሆንም	ስለማናቸውም	እጅግ
የእሱ	ታች	በዚህ	አንዱ	ስለዚሁ	እንዲሆን
ራሱ	ውስጥ	ብላ	የሌላውን	ከአንዳንድ	እንኳ
እሷ	ውጭ	ነገር	ከሁለት	በእነዚሁ	ከሀያ
እሷናት	በላይ	አለ	የሆኑትን	በአምስት	ከሀምሳ
የእሷ	እንደገና	አርብ	በሆኑ	የሆኑበታል	ይኸው
እራሷ	ተጨማሪ	አንተ	ጀምሮ	ለነዚህ	ለአንድ
ነው	ከዚያ	እሁድ	በመሆን	ለማንኛውም	የሚችለውን
እነሱ	አንድ ጊዜ	እናንተ	ባለ	አንደኛ	በሚገባ
እነሱን	እዚህ	እንኳን	ይህንን	ይኸኛው	ይህም
የእነሱ	እዚያ	እግር	እንዲቆይ	ከርሱ	እንዲሆኑ
ራሳቸው	መቼ	ከመሆን	ሌላው	መሆኑን	ከሌላ
ምንድን	የት	ወይንም	የሚሆነው	ለዚያው	ለሆነ
የትኛው	ለምን	ዋና	በአንዱ	ለዚሁ	በሌሎች
ማን	እንዴት	ዘንድ	ሲባል	ለእነርሱም	አንደሆነ
ይህ	ሁሉም	የሚከተለው	ሳለ	እዚሁ	እንዲህ
የሚል ነው	ማንኛውም	ያኔ	የሆነው	ሐ	በነዚሁ
ያ	ሁለቱም	ይኼው	መሆናቸው	ረ	በእንደዚህ
እነዚህ	እያንዳንዳቸው	ገጽ	በዋና	ሸ	ስምንት
እነዚያ	ጥቂቶች	እነርሱ	በማቀድ	አምስት	ሲሆንና
ነኝ	በጣም	ን	ጊዜና	ከሶስት	ምንጊዜም
ናቸው	ሌላ	ና	ለዚህ	በተለይም	ለማናቸውም
ነበር	አንዳንድ	ዎች	ሶስተኛ	በሌላ	የአንድ
ነበሩ	እንደዚህ	ይጠበቃል	የነገሩ	ሺህ	እነዚህኑ
ሁን	ብቻ	ብለዋል	ስድስት	ማናቸውንም	ሲሆኑ
ቆይቷል	የራሱ	ሆ	በሆነው	ከአስር	በሁለቱም
መሆን	ተመሳሳይ	ሁሉ	ይሁን	የማይበልጥ	እንደነዚህ
አላቸው	ስለዚህ	አንቀጽ	ከዚሁ	እንዲሁም	የሆኑት
አለው	ይልቅ	እንደሆነ	በእነዚህ	ይህን	የማናቸውም
ነበረው	እንዲሁ	በማይበልጥ	ከማናቸውም	የዚህ	ይህንንም
ያለው	ት	መሰረት	ከነበረው	ማናቸውም	የአንድን
መስራት	ይችላል	ሁኔታ	በአንዳንድ	ከስድስት	በሙሉም
ያደርጋል	ይገባል	ይሆናል	በእያንዳንዱ	መቶ	በነዚህ
አደረገ	ይገባኛል	ሆኖ	ጊዜም	ያለ	የዚሁ
ማድረግ	አሁን	ከአንድ	አስከ	አንድን	ለእያንዳንዱ
ሀ	መ	በማናቸውም	የሌሎች	ያላቸውን	ስለሆነ
አንድ	ም	ወር	የሚሆኑት	ሊሆን	መሆናቸውን
የ	አ	ከአምስት	ከሆነው	ሶስት	ማንኛውንም
እና	ዳግም	በሆነ	የነበረውን	ካልሆነ	ሁለቱ
ከሆነ	መሆን	ከዚህ	ያሉ	ቢያንስ	እንጂ
ወይም	ሁለ	የሆነ	ከሌሎች	ቢሆን	ከስምንት
ምክንያቱም	ሁለም		አንዲት	እነዚህን	ሁለቱንም
እንደ			ለሌሎች	አንዱን	በሁለት
እስከ					

Appendix B: Human Evaluation form

Appendix B. 1 System performance evaluation form



ETHIO TELECOM BILINGUAL CHATBOT EVALUATION

በመጀመሪያ የሰው ሰራሽ የኢትዮ ቴሌኮም አገልግሎት ማማከሪያ ሲስተም ምዘና ለመሙላት ፈቃደኛ ስለሆኑ እናመሰግናለሁ። እናም ይህ ወረቀት በሚጠይቀው መሰረት እያንዳንዱን ምዝና በትኩረት እንዲሞሉልን በአክብሮት እንጠይቃለን!
Jalqaba Siistama kana nuuf sakkata'uuf eyamamaa ta'u kessaniif galateefachaa gaafilee armaan gadii kana akka seeran nuuf guutan isiin gaafana.

ማሳሰቢያ፡- በኮምፒውተሮ ውይም በስልክ ሲይስተሙን ከተጠቀሙ በኋላ በ አትዮ ቴሌኮም አገልግሎት ዙሪያ ያልዎትን ጥያቄ ይጠይቁ።
Hubachiisa: - Koompuutara kessan ykn bilbiila kessan fayyadamuun siistama ega banatanii booda tajaajiloota ltiyoo teeleekomii irratii gaafii qabdabda gaafadha.

* Required

Email *

Your email

System Performance Evaluation

የተጠቁሙት ቋንቋ ይምረጡ/ Afaan fayyadamtan filadhaa

- አሮሚፋ/Afaan Oromo
- አማርኛ/ Amaariffa
- ሁላቱንም/ Lamaanu fayaadame jiira

ስንት ጥያቄ ለሲስተሙ አቀረቡ?/ Gaafii Meeqa siistamaaf dhiyeesitan? *

Your answer

ሲስተሙ ስንት ጥያቄ በትክክል መለሰሎት?/ Siistaamichii gaafii meeqa sirrii isiiniif deebise? *

Your answer

Back Next Page 2 of 3

Appendix B.2 System user acceptance evaluation form

User Acceptance Evaluation

ሲስተሙን ሲጠቀሙ ምላሹን የሚመልስበት ሰዓት/ Siistama kana yeroo fayyadamtan gaafii kessan debisuuf yeroo itti fudhatu *

- እጅግ በጣም ጥሩ ነው/ Baay'ee bay'ee bareeda
- በጣም ጥሩ ነው/ Bay'ee bareeda
- ጥሩ ነው/ Bareeda
- ዝቅተኛ ነው/ Gadi bu'aadha
- በጣም ዝቅተኛ ነው/ Baay'ee bu'aadha

ሲስተሙን ሲጠቀሙ የሚሰጡ ምላሾችን ትክክለኛነት እንዴት አገኙት?/ Siistama kana yeroo fayyadamtan deebiin isiiniif keename sirri ta'u isaa akkamitti ilaaltan? *

- እጅግ በጣም ጥሩ ነው/ Baay'ee bay'ee bareeda
- በጣም ጥሩ ነው/ Bay'ee bareeda
- ጥሩ ነው/ Bareeda
- ዝቅተኛ ነው/ Gadi bu'aadha
- በጣም ዝቅተኛ ነው/ Baay'ee bu'aadha

ሲስተሙን ሲጠቀሙ የሲስተሙን ማራኪነት እንዴት አዩት?/ Siistama kana yeroo fayyadamtan hangam nama haawata? *

- እጅግ በጣም ጥሩ ነው/ Baay'ee bay'ee bareeda
- በጣም ጥሩ ነው/ Bay'ee bareeda
- ጥሩ ነው/ Bareeda
- ዝቅተኛ ነው/ Gadi bu'aadha
- በጣም ዝቅተኛ ነው/ Baay'ee bu'aadha

ሲስተሙን ሲጠቀሙ የሲስተሙን ለአጠቃቀም ቀላልነት እንዴት ያዩታል?/ Siistama kana yeroo fayyadamtan itti fayyadamuudhaaf hangam salphaadha? *

- እጅግ በጣም ጥሩ ነው/ Baay'ee bay'ee bareeda
- በጣም ጥሩ ነው/ Bay'ee bareeda
- ጥሩ ነው? Bareeda
- ዝቅተኛ ነው/ Gadi bu'aadha
- በጣም ዝቅተኛ ነው/ Baay'ee bu'aadha

ይህ ሲስተም አገልግሎት ላይ ቢውል እንዴት ያዩታል?/Siistama kana osoo hoojjii irra ole akkamitti ilaaltu? *

- እጅግ በጣም ጥሩ ነው/ Baay'ee bay'ee bareeda
- በጣም ጥሩ ነው/ Bay'ee bareeda
- ጥሩ ነው? Bareeda
- ትንሽ ይጠቅማል/ Xiqqo ni fayyada
- ምንም አይጠቅምም/ Hoomaa hin fayyadu

አስተያየት ካልዎት/ Yaada yoo qabaatan

Your answer

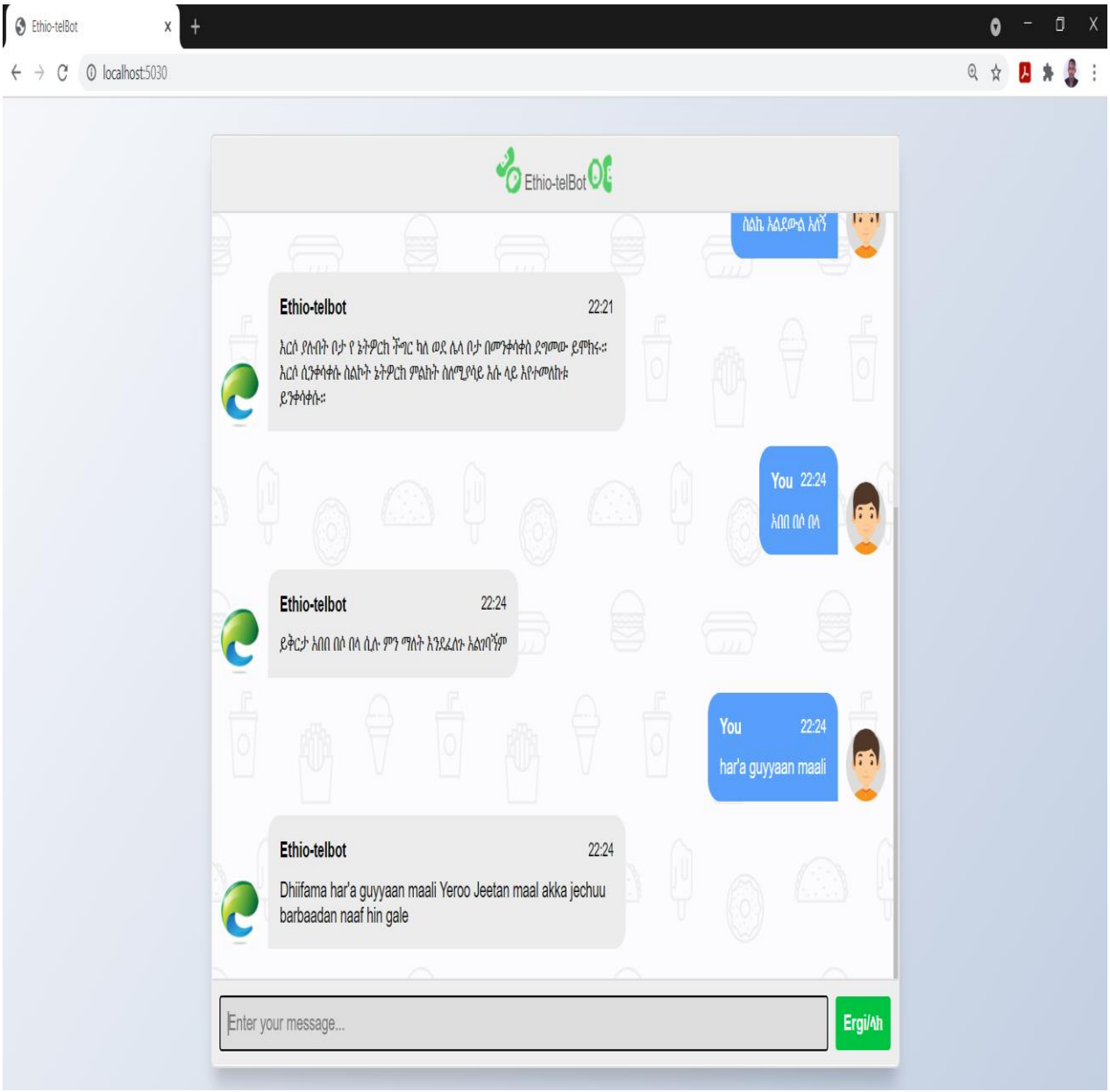
Back Submit Page 3 of 3

Appendix C: Dataset sample

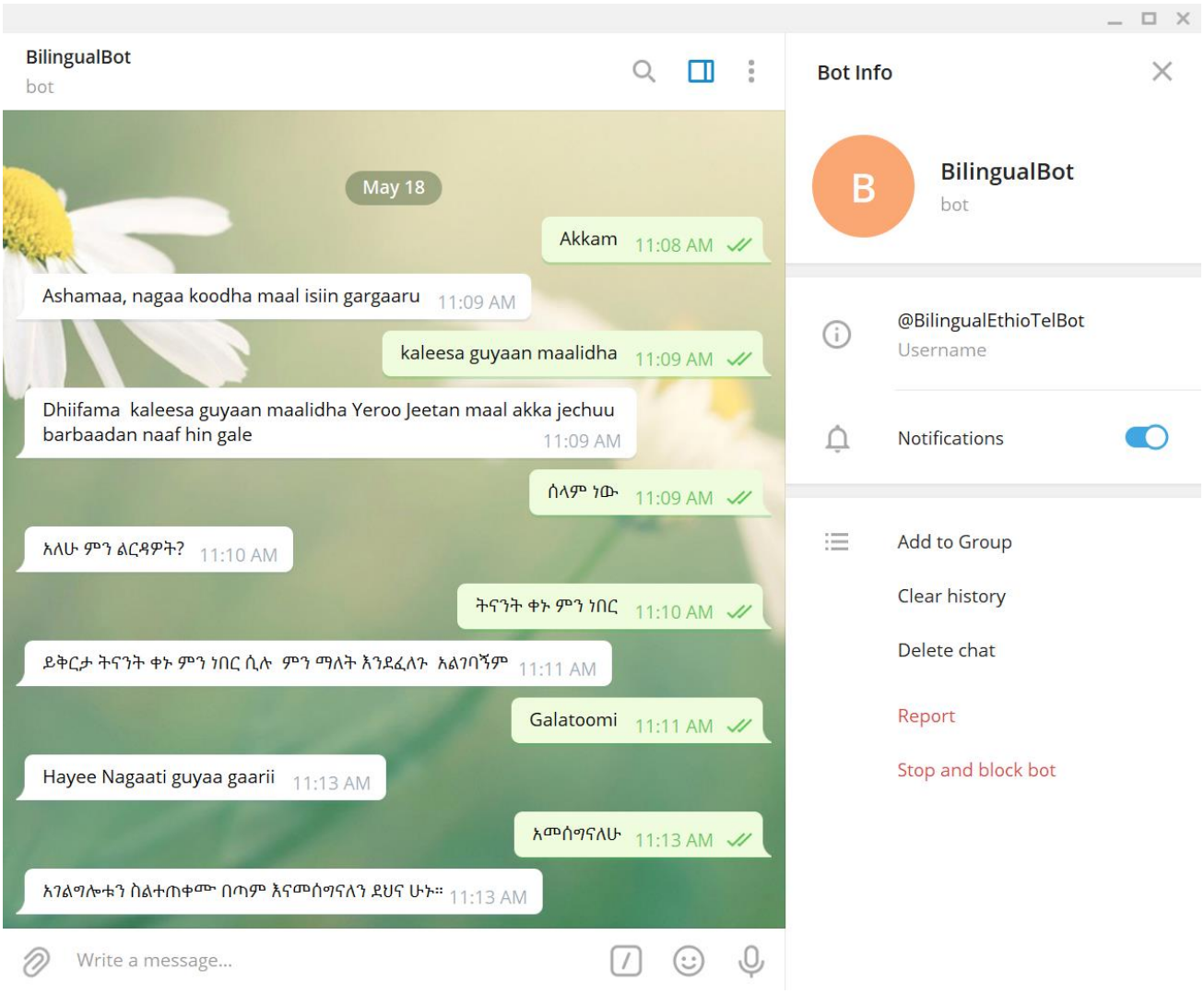

```
{
    "tag": "LostPhonea",
    "patterns": [
     "ሲም ካርዴን ከጣልኩ ምን ማድረግ አለብኝ?",
     "ሲም ካርዴ ከተሰረቀብኝ ምን ማድረግ አለብኝ?",
     "ሲም ካርዴ ከጠፋ ምን ማድረግ አለብኝ?",
     "ሲሜን ጣልኩ ምን ማድረግ አለብኝ?",
     "ሲም ካርዴን ከጣልኩ ምን ማድረግ አለብኝ?",
     "ሲሜ ጠፋብኝ ምን ማድረግ አለብኝ?",
     "ሲም ካርዴ ከጠፋ ምን ማድረግ አለብኝ?",
     "ስልኬ ከጠፋ ምን ማድረግ አለብኝ?",
     "ሞባይል ስልኬ ወይም ሲም ካርድ ከጠፋ ምን ማድረግ አለብኝ?",
     "ሲሜ ጠፋብኝ ምን ላድርግ?",
     "ስልኬ ጠፋብኝ?",
     "ስልክ ጣልኩ",
     "ስልኬ ተሰረቀብኝ",
     "ሲም ካርድ ጠፈብኝ",
     "ሲም ካርድ ጣልኩኝ",
     "ሲሜን ጣልኩት",
     "ሲሜ ጠፋብኝ"
    ],
    "responses": [
     "ሲም ካርድዎን ሌላ ሰው እንዳይጠቀምበት ወዲያውኑ የኢትዮ ቴሌኮም የጥሪ ማዕከልን በ 994 ያነጋግሩ ወይም *994# በማድረግ ትዕዛዙን በመከተል ማዘጋት ወይም ወደ 8994 አጭር መልዕክት በመላክ ማዘጋት ይችላሉ።።"
    ]
   },
   {
    "tag": "LostPhoneo",
    "patterns": [
     "bilbila gate",
     "Siim kaardiin najalaa haatame maal gochuu qaba?",
     "Siilkiin najalaa haatame maal gochuu qaba?",
     "Siimii najalaa haatame maal godhu?",
     "Siimiin koo yoo bade maal gochuu qaba?",
     "Siimii yoon gate maal gochuu qaba?",
     "Siimii koo bakka buuse dhabe?",
     "Siimiin najalaa bade?",
     "Siimii gate?",
     "Siim kaardii kiyya yoon gate maal gochuu qaba?",
     "Siimiin koo yoo najalaa bade maal goodhu?",
     "Biilbili koo yoo najalaa bade maal goodha?",
     "Siimiin koo najalaa bade maal goodhu?",
     "Biilbili koo bade",
     "Biilbila koo gate",
     "Biilbili koo najalaa haatame",
     "Siimiin koo najalaa bade",
     "Siimiin gate",
     "silkii gate",
     "silkii dhabe",
     "bilbiila bakka godhe dhabe"
    ],
    "responses": [
     "Gara buufata bilbiila 994 bibiilun ykn *994# gochuudhaan bilbiilaa ajaajoota hordofuun cufaa ykn ergaa gabaabaa gara 8994 erguudhaan siimiiin kessan akka namni biraa itti hin fayyadamne taasisaa"
    ]
   },
```

Appendix D: system response if not understand users request

Appendix D.1: System response if users request are not understood on Web



Appendix D.2: System response if users request are not understood on the telegram bot



Appendix E: Sample code

Appendix E.1: Sample code for Integrating the model to Web

```
def responseOfBot(sentence):
    results = classify(sentence)
    print(results)
    if results==[]:
        if re.search('[a-zA-Z]', sentence):
            return "Dhiifama "+sentence+ " Yeroo Jeetan maal akka jechuu barbaadan naaf hin gale"
        else:
            return "ይቅርታ "+sentence +" ሲሉ  ምን ማለት እንደፈለጉ  አልገባኝም "
    else:
    #If we have a classification then find the matching intent tag
        if results:
        #Loop as long as there are matches to process
            while results:
                for i in intents['intents']:
                #Find a tag matching the first result
                    if i['tag'] == results[0][0]:
                        #A random response from the intent
                        return random.choice(i['responses'])
            results.pop(0)
from flask import Flask, render_template, request
app = Flask(__name__)
app.static_folder = 'static'
@app.route("/")
def home():
    return render_template("index.html")

@app.route("/send")
def get_bot_response():
    userText = str(request.args.get('msg'))
    return responseOfBot(userText)

if __name__ == "__main__":
    app.run(host='localhost',port=5030)
```

Appendix E.2: Sample code for Integrating the model to Telegram bot

```
def responseOfBot(sentence):
    results = classify(sentence)
    if results==[]:
        if sentence=="/start":
            return  "Baga Giddu-gala Ethio-telbot nagayaan Dhuftan. □ እንኳን ወደ Ethio-telbot ማዕከል በደህና መጡ □"
        elif re.search('[a-zA-Z]', sentence):
            return "Dhiifama  "+sentence+ " Yeroo Jeetan maal akka jechuu barbaadan naaf hin gale"
        else:
            return "ይቅርታ "+sentence +" ሲሉ  ምን ማለት እንደፈለጉ  አልገባኝም "
    else:
   # print(results)
    #If we have a classification then find the matching intent tag
        if results:
        #Loop as long as there are matches to process
            while results:
                for i in intents['intents']:

                #Find a tag matching the first result
                    if i['tag'] == results[0][0]:

                        #A random response from the intent
                            return str(random.choice(i['responses']))
                results.pop(0)
print("ready...")
bot = telegram_chatbot("config.cfg")
def make_reply(msg):
    reply = None
    print(msg)
    if msg is not None:
        reply = str(responseOfBot(msg))
    return reply
update_id = None
while True:
    updates = bot.get_updates(offset=update_id)
    updates = updates["result"]
    if updates:
        for item in updates:
            update_id = item["update_id"]
            try:
                message = str(item["message"]["text"])
            except:
                message = None
            from_ = item["message"]["from"]["id"]
            reply = make_reply(message)
            bot.send_message(reply, from_)
```