Afaan Oromo Text-based Consultant Chatbot for Maternal Healthcare Using Deep Learning

Haile Guta Tolasa

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 Masters in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

June, 2022

Adama, Ethiopia

Afaan Oromo Text-based Consultant Chatbot for Maternal Healthcare Using Deep Learning

Haile Guta Tolasa

Advisor-Teklu Urgessa (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 Masters in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

June, 2022
Adama, Ethiopia

Declaration

I hereby declare that this Master Thesis entitled "**Afaan Oromo Text-based Consultant Chatbot for Maternal Healthcare Using Deep Learning**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Haile Guta Tolasa	________________	______________
Name of Student	Signature	Date

Recommendation

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Afaan Oromo Text-based Consultant Chatbot for Maternal Healthcare Using Deep Learning**" prepared under my/our guidance by Haile Guta Tolasa and submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science and Engineering. Therefore, I/we recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Dr. Teklu Urgessa	___________________	___________________
Major Advisor	Signature	Date

Approval Sheet of M.Sc. Thesis

I, the advisors of the thesis entitled "**Afaan Oromo Text-based Consultant Chatbot for Maternal Healthcare Using Deep Learning**" and developed by Haile Guta Tolasa; hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. Teklu Urgessa	____________________	____________________
Major Advisor	Signature	Date

Approval of Board of Reviewers

We, the undersigned, members of the Board of Examiners of the thesis by Haile Guta Tolasa have read and evaluated the thesis entitled “**Afaan Oromo Text-based Consultant Chatbot for Maternal Healthcare Using Deep Learning**” and examined the candidate during the 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
Reviewer 1	Signature	Date
Reviewer 2	Signature	Date

Final approval and acceptance of the thesis are 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).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENTS

First, my deepest thanks go to the Almighty God for his keeping with health within all my way. And then, I would like to thank my advisor Dr. Teklu Urgessa for providing me the great opportunity to work on this thesis and for providing continuous guidance throughout my study. I would also like to thank him for guiding me in the right direction, for timely feedback, and for his valuable suggestions to fulfill the thesis requirements.

And also, I would like to thank Dr. Mesfin Abebe for his willingness to help us whenever we want by giving us his time. Next, I would like to thank the entire faculty and staff of the Department of Computer Science and Engineering at Adama Science and Technology.

Next, I would like to thank my parents for their countless support in my journey, without them I wouldn't be here today. And also, I would like to express my gratitude to all my friends and classmates for their unconditional support throughout my journey.

❤️Finally, but not last, I would like to say “Immaa Koo Baay’een Si jaaladha” ❤️

TABLE OF CONTENTS

LIST OF TABLES i

LIST OF FIGURES ii

LIST OF EQUATION iv

LIST OF ABBREVIATION v

ABSTRACT vi

CHAPTER ONE 1

1. INTRODUCTION 1

1.1. Background of the Study 1

1.2. The Motivation of the Study 3

1.3. Statement of the Problem 4

1.4. Research Questions 5

1.5. Objectives of the Study 6

1.5.1. General Objective 6

1.5.2. Specific Objectives 6

1.6. Significance of the Study 6

1.7. Scope and Limitation of the Study 7

1.7.1. Scope of the Study 7

1.7.2. Limitation of the Study 7

1.8. Organization of the Thesis 8

CHAPTER TWO 9

2. LITERATURE REVIEW 9

2.1. Chapter Overview 9

2.2. Background and History of the Chatbot Technology 9
2.2.1. Background of a Chatbot 9
2.2.2. History of Chatbot Technology 10
2.3. Application Domain of Chatbot Technology....14
2.3.1. Chatbot in Healthcare 14
2.3.2. Chatbot in Marketing and Business for Customer Service....15
2.3.3. Chatbot in Education 15
2.3.4. Chatbot in Agriculture 15
2.4. Types of Chatbot....16
2.5. The Available Frameworks for Chatbot Technology 17
2.6. Approaches of Chatbot 19
2.6.1. Pattern Matching Approaches in Chatbot Development 19
2.6.2. Machine Learning Approaches Based Chatbot....21
2.7. An overview of Afaan Oromo Language....24
2.7.1. Afaan Oromo Word Formation 24
2.8. Related Works on Chatbot....25
2.8.1. Summary of Related Works 28
CHAPTER THREE....31
3. MATERIALS AND METHODS OF THE PROPOSED MODEL 31
3.1. An Overview of the Chapter....31
3.2. Research Design 31
3.3. Data Collection Method....33
3.4. Data preparation techniques....33

3.4. Data Preprocessing 35
3.5. Feature Extraction Methods 36
3.5.1. Bag of words 36
3.5.2. TF-IDF (Term Frequency-Inverse Document Frequency) 36
3.6. Word Embedding 37
3.7. Model Selection Techniques 37
3.8. Deep Learning Algorithms for the Chatbot Development 38
3.8.1. Recurrent Neural Network (RNN) 39
3.8.2. Long Short-Term Memory (LSTM) 40
3.8.3. BiLSTM (Bidirectional-LSTM) 41
3.8.4. Convolutional Neural Network (CNN) Model 42
3.8.5. CNN-LSTM Model 43
3.8.6. CNN-BiLSTM Model 44
3.9. Model Overfit Handling Techniques 44
3.10. Weight Initialization Techniques 45
3.11. Hyperparameter Tuning Method 45
3.12. Model Evaluation Techniques 46
3.12.1. Cross-Validation 46
3.12.2. Evaluation Metrics 47
3.12.3. Human Evaluation 47
3.13. Implementation Tools and Techniques 48
3.13.1. System Design Tools 48
3.13.2. Implementation Environments 48

3.13.3. Programming Language Tools 49
3.13.4. Important libraries 49
3.13.5. Data Preparation Tools 50
3.13.6. Deployment Tools 51
3.13.7. Hardware Tools 51
CHAPTER FOUR 52
4. DESIGN AND IMPLEMENTATION OF THE PROPOSED MODEL 52
4.1. Overview of the Chapter 52
4.2. The Architecture of the Proposed Model 52
4.3. Proposed Data Preprocessing 53
4.3.1. Data Cleaning 54
4.3.2. Stop Word removal 54
4.3.3. Text Normalization 55
4.3.4. Tokenization 55
4.4. Proposed Word Embedding 55
4.5. Proposed Deep Learning Model 56
4.6. Proposed Evaluation Techniques 57
4.7. Implementation of Proposed Data preprocessing 58
4.7.1. Loading Dataset 58
4.7.2. Text Normalization and Stop words Removal 58
4.7.3. Implementation of Data Cleaning 58
4.7.4. Implementation of Tokenization 59
4.8. Implementation of Proposed Word Embedding 59

4.9. Proposed Model Implementation 60
4.9.1. CNN Model Implementation 60
4.9.2. CNN-LSTM Model Implementation 62
4.9.3. BiLSTM Model Implementation 62
4.9.4. CNN-BiLSTM Model Implementation 63
4.9.5. Setting Threshold Value 64
4.10. Saving the Model for Future Use 64
CHAPTER FIVE 65
5. RESULT AND DISCUSSIONS 65
5.1. Overview of the Chapter 65
5.2. Result of the Proposed Model 65
5.2.1. Results of CNN model 66
5.2.2. Results of CNN-LSTM Model 66
5.2.3. Results of BiLSTM Models 67
5.2.4. Results of CNN-BiLSTM Models 67
5.2.5. The Two Types of Word2vec Model 68
5.2.6. Activation Function and Optimization Algorithms 68
5.3. Hyperparameter Tuning 70
5.4. Effects of Weight Initializers on the Proposed Model 71
5.5. Human Evaluation Result of the Model 71
5.6. Discussion 73
5.7. Chatbot Model Prototype Architecture 75
CONCLUSION AND RECOMMENDATION 77

Conclusion 77

Recommendation 78

Future work 78

REFERENCES 80

APPENDICES I

Appendix 1: Sample Code I

1.1. Text Normalization I

1.2. List of Stop words I

1.3. Sample Code For all Proposed Model II

1.3.1. CNN Model Sample Code II

1.3.2. CNN-LSTM Model Sample Code III

1.3.3. BiLSTM Model Sample Code III

1.3.4. CNN-BiLSTM Model Sample Code IV

1.3.5. Sample for Saving and Loading the Model V

1.3.6. Sample Code of user intent classification V

Appendix 2. Evaluation Form VI

2.1. Sample Evaluation Form of designed Chatbot Model VI

LIST OF TABLES

Table 2.1 Summary of Chatbot Overview from 1966 to 2016 12

Table 2.2 Afaan Oromo Letters with their IPA 25

Table 2.3 Summary of Related Work 29

Table 3.1 Description of Dataset 34

Table 5.1 Summary of the Proposed Model for Maternal Health Consultant Chatbot 68

Table 5.2 The comparison of Two Types of Word2vec Model 68

Table 5.3 The analysis of the Two Optimizers and Activation Functions 69

Table 5.4 Hyperparameter for all Proposed Models 70

Table 5.5 Correctness of the Chatbot Model through Human Evaluation 71

Table 5.6. Model Acceptance Evaluation Through Human Evaluation 72

LIST OF FIGURES

Fig.2.1 Type of Chatbots 16

Fig.2.2 Example of AIML for Chatbot Development 20

Fig.2.3 Examples of Chat Script Language 20

Fig.2.4 Example of Rivescript Language 21

Fig.2.5 The Neurons of Hidden Layers Operation(Patil et al., 2020) 22

Fig.2.6 How Neural Networks Operate Textual Data for Chatbot Development 23

Fig.3.1 Research Design 32

Fig.3.2 Sample Example of Data Preparation 34

Fig.3.3 Types of Word2Vec Model 37

Fig.3.4 The Architecture of the RNN model 39

Fig.3.5 The LSTM Model's architecture 41

Fig.3.6 Structure of the BiLSTM Model 42

Fig.3.7 The Architecture of the CNN Model(Yao et al., 2016) 43

Fig.3.8 The CNN-BiLSTM Model Architecture for Maternal Health Consultant Chatbot 44

Fig.4.1. The Proposed Model Architecture for Maternal Health Consultant Chatbot 53

Fig.4.2 Word2Vec Word Representation 56

Fig.4.3 Proposed Data Preparation, Preprocessing, and Word Embedding Techniques 56

Fig.4.4 Proposed Deep Learning Modeling 57

Fig.4.5 Implementation Procedures of the CNN Model 61

Fig.4.6 Implementation Procedures of the CNN-LSTM Model 62

Fig.4.7 Implementation Procedure of BiLSTM Model 63

Fig.4.8 Implementation Procedure of CNN-BiLSTM Model 64

Fig.5.1 CNN Model Accuracy and Loss through Epochs 66

Fig.5.2 Accuracy and Loss of the CNN-LSTM Model Across Epochs 66

Fig.5.3 BiLSTM Model Accuracy and Loss through Epochs 67

Fig.5.4 CNN-BiLSTM Model Accuracy and Loss through Epochs 67

Fig.5.5 The proposed DL Model comparison Chart..69

Fig.5.6 The architecture of the Chatbot Model Prototype 76

Fig.5.7 Sample Conversation with Developed Chatbot 76

LIST OF EQUATION

Eq.3.1 Equation for Computing Hidden and Output Layers value RNN model.......................39

Eq.3.2 How the Forget and Input gate value of LSTM is Computed.......................................40

Eq.3.3 Equation to Compute Output Layer value of BiLSTM..42

Eq.3.4 Equation for Measuring the Model's Accuracy..47

Eq.5.1 Human evaluation Equation...71

LIST OF ABBREVIATION

AI- -Artificial Intelligence

ANC- -Antenatal Care

ANN- -Artificial Neural Network

API - -Application Program Interface

BiLSTM- -Bi-directional LSTM

CUI- -Chatbot User Interface

CBOW- -Continuous Bag of Words

CNN- -Convolutional Neural Network

DL- -Deep Learning

DNN - -Deep Neural Networks

FAQ- -Frequently Asked Questions

FFFNN- -Feed Forward Neural Network

HEW- -Health Extension Worker

IBM- -International Business Machine

IPA- -International Phonetic Alphabet

TF/IDF- -Term Frequency/Inverse Document Frequency

JSON- -JavaScript Notation

LSTM- -Long Short-Term Memory

ML- -Machine Learning

NLP- -Natural Processing Language

OHB- -Oromia Health Bureau

OLF- -Oromo Liberation Front

RFE- -Recursive Feature Elimination

RNN- -Recurrent Neural Network

SDK- -Software Development Kit

SGD- -Stochastic Gradient Descent

ORHB- -Oromia Regional Health Bureau

UML- -Unified Modeling Language

ABSTRACT

Healthcare is very essential to leading a good life. In healthcare fields, maternal healthcare is a health service provided to pregnant women to retain their health safely. It is one of the essential strategies in decreasing maternal mortality and is provided by maternal health care providers. They offer a better understanding of pregnancy-related health problems. Every mother wants to get relevant advice every day, from prenatal up to postpartum. In developing countries, most expectant mothers do not get adequate prenatal care from health care providers. Because the ratio of demand and maternal health experts do not match; the health center faces a range of challenges to address the service for all users timely. Due to this reason, maternal and infant mortality remained high in developing countries. Currently, Ethiopia is identified with the high maternal mortality among other developing countries in the world due to poor maternal healthcare provision. Nowadays, technology has introduced the techniques of getting access to information simply and efficiently. One example of such technology is a chatbot model that can be considered the best way to assist a user in all industries. Because the chatbot can imitate humans and is available 24/7 hours to respond quickly to questions users ask. A chatbot solution can offer better information for pregnant women on known danger signs identification, nutritional tips, prenatal tips, family planning, noticeable symptoms, etc irrespective of time and place. This study introduced the Afaan Oromo text-based maternal health consultant chatbot using deep learning to tackle the challenges formerly mentioned. To accomplish the goal of this study, the dataset was collected from ORHB, online websites (https://fayyaa.net/page/1/), and some hospitals in Oromia to build the maternal health consultant chatbot model. The collected dataset was prepared and preprocessed using various NLP techniques. And then, the Word2Vec model is utilized to represent text data with vectors. In this work, CNN, CNN-LSTM, BiLSTM, and CNN-BiLSTM are presented and evaluated in terms of their performance to design a maternal health consultant chatbot. Based on the experiment performed, the CNN-BiLSTM model outperformed the other proposed models with 96.94% accuracy under Stratified 5-fold cross-validation. This study improves the performance of the existing chatbot model by merging the CNN model with the BiLSTM model to design a maternal health consultant chatbot model.

Keywords: *Chatbots, Deep learning, CNN, LSTM, Maternal Healthcare, NLP, Afaan Oromo*

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Healthcare performs an extensive role in our day-to-day lives. Health service is very crucial for protecting human life from disease and death. The health care services are provided by health administrators in the country with HEW, health centers, and general and primary hospitals in Ethiopia(Nefa et al., 2019). The Oromia Regional Health Bureau (ORHB) is one of the major sector bureaus that have a responsibility for managing and regulating all health services and health-related issues in the Oromia region[1]. The organization focuses on many different fields such as maternal healthcare, physical and mental healthcare, newborn healthcare, etc. This research work considers only maternal healthcare. Maternal Healthcare is the health service offered to all pregnant women to protect their lives from morbidity and death during pregnancy. It includes prenatal care and postnatal care. The main target of these services is to minimize maternal mortality and health problems. Prenatal care and post-natal care are the keys to maternal healthcare that can significantly reduce the maternal mortality rate(Shegaw et al., 2014). Prenatal care is a service that provides a better understanding for pregnant women during pregnancy and helps mothers and future babies stay healthy. Postpartum care is the time after delivery on how to care for her and new babies. These two services play an integral role in retaining the health of pregnant mothers and new births. Maternal healthcare is one of the essential strategies for decreasing newborn and maternal mortality(Atnafu et al., 2020). These health services are provided by maternal health care providers, they offer prenatal counseling services for all pregnant women. However, most expectant mothers do not get adequate healthcare services in developing countries. Due to this, maternal mortality has become high in developing countries.

According to a study by(Dash, 2017), more than 150 million women become pregnant, and 500,000 die of pregnancy-related causes in developing countries. This problem is due to the shortage of adequate maternal healthcare services. As described in the study (Shegaw Mulu

---

[1] *https://www.devex.com/organizations/oromia-regional-health-bureau-121410#section-1*

Tarekegn1*, 2014)(Atnafu et al., 2020), there is a high maternal mortality rate in Ethiopia, among other developing countries in the world. According to the cited studies, maternal mortality is related to poor maternal healthcare provision and a lack of awareness for pregnant women.

The healthcare center faces a range of challenges due to increasing service demand to address the services for all users. There is a shortage of trained maternal healthcare experts and healthcare centers to provide maternal health services for all pregnant women. As explained in the study(Dida, 2015) on maternal health in one zone of the Oromia region, there is a lack of maternal healthcare provision. The majority of maternal and infant deaths are preventable and can be avoidable by utilizing a variety of maternal healthcare service interventions(Kiross et al., 2021). Due to this fact, there is a need to use technological support that provides maternal health consultancy services efficiently.

Nowadays, technology has introduced many changes to the methods of getting access to information in many industries. Information has a crucial benefit in maternal healthcare, to give awareness about pregnancy-related health problems for pregnant women(Mugoye et al., 2019). Artificial intelligence and the latest computer solutions have made this information more efficient, simpler, and better user-friendly to use. With the improvement of AI, machine learning, and deep learning, machines have started to impersonate humans(Nuruzzaman & Hussain, 2018). Today, AI builds technology that enables users to communicate with machines similar to human-to-human communication. One excellent example of such technology is the chatbot implemented with the help of NLP. Combining machine learning, deep learning, and statistical models with NLP can enable computers to understand human communication with natural languages such as Amharic, Spanish, Chinese, Afaan Oromo, etc. Currently, a chatbot is used widely in many industries and businesses such as marketing, bank, customer service, healthcare, etc. A chatbot can also assist users with health-related issues by supporting them via text messages, applications, or instant messaging in healthcare(Vamsi et al., 2020).

This study aimed to design an Afaan Oromo text-based chatbot model that provides better understanding and advice on minor issues for pregnant women. Every mother wants to get essential information and consultancy every day, from prenatal to postpartum time. However, in Oromia, they do not get prompt and efficient consultancy and ANC services(Dida, 2015). The Oromia Regional Health Bureau (ORHB) is responsible for offering a full package of preventative, curative, and reconstructive health services to the community at large in Oromia regional state. In Oromia regional state, Afaan Oromo is a language widely spoken and a working language that began in 1991[2]. The ORHB is one of the major health sectors in Oromia, which works and provides services in the Afaan Oromo language. The sector manages and facilitates all hospitals and health centers in the region. The sector needs technical support to address the health services effectively and efficiently. In particular, today they are working on minimizing maternal health problems. Most health sectors and hospitals of ORHB serve their users in Afaan Oromo. More than 36 million people or 33.8% of the Ethiopian population speak the Afaan Oromo language(*Oromo Language - Wikipedia*, 2021).

In this work, the study intended to develop an Afaan Oromo text-based chatbot that provides better awareness of maternal health problems. A chatbot solution can provide efficiently better understanding and support for pregnant women in known danger signs identification, emotional states, nutritional tips, family planning, prenatal tips, noticeable symptoms, and counseling for additional pregnancy-related health problems. Most pregnant mothers want daily advice on the above issues. Therefore, one of the great benefits of developing a chatbot is that it is available 24/7 to respond quickly to questions users ask. This study presented an Afaan Oromo text-based maternal health consultant chatbot using a deep learning approach.

1.2. The Motivation of the Study

There are many reasons why we are inspired to conduct this study such as the poor provision of maternal healthcare services, high maternal mortality(Dida, 2015)(Atnafu et al., 2020), and the absence of modern chatbots in this area to contribute to reducing maternal health complications, etc. Currently, Ethiopia is recognized with the highest maternal mortality rate among the other countries in the world(Atnafu et al., 2020). As described in the cited study,

[2] *https://en.wikipedia.org/wiki/Oromo_language*

there is a problem in providing maternal healthcare services for pregnant mothers in Ethiopia. Today, feeding machine learning with data to perform some tasks is the best solution to many problems. Combining machine learning with natural language processing enables us to develop a chatbot model that provides better understanding and advice for pregnant women throughout their pregnancy time. Afaan Oromo is a language spoken by millions of people, as described in Section 2.7. However, no research has been conducted on chatbots for healthcare in the Afaan Oromo language. Using chatbot technology in healthcare will help the users to get a better understanding of their health status. In addition, it reduces the load on health centers and hospitals when there are many users to handle. One study(Fadhil, 2018) presented that most people, including the elderly, spend most of their time on chatting applications such as Telegram, WhatsApp, Facebook, etc. And also around 3.80 billion internet users are using social media(Kandpal et al., 2020). With these facts, we are inspired to develop a chatbot model that provides maternal health advice in the Afaan Oromo language. The motivation behind this presented study is to reduce maternal morbidity and mortality by providing better advice for pregnant women through the chatbot model.

1.3. Statement of the Problem

Ethiopia identified with high maternal mortality among other developing countries in the world. This issue arose as a result of inadequate maternal healthcare services. Maternal healthcare providers offer a better understanding of pregnancy-related health problems on issues like danger signs, family planning, nutritional tips, and counseling about pregnancy-related health concerns. Providing these services for all pregnant women is very challenging for maternal healthcare providers because the ratios of the available maternal health expert and demands do not match. The current maternal healthcare system requires an enormous effort by providers to offer effective interventions and consultancy support. In addition, for many pregnant women, visiting a maternity clinic for consultancy purposes every day can be difficult because of the distance from the health center. Some users prefer to use the search engine, but due to a large amount of information on Google, using a search engine will be so difficult to find relevant information. Because of all these problems, pregnant women do not have access to adequate maternal health information.

Different studies(Berri et al., 2020) (Dida, 2015) conducted in maternal healthcare fields have only identified the factors that increased maternal health problems in Oromia. Of these factors, one is the lack of providing enough information and advice for pregnant womens. The healthcare sector needs technical support to effectively and efficiently address better consultancy services for pregnant women. One example of such technology is a chatbot model that can be considered the best way to assist users in all industries, including healthcare than other solutions such as mobile apps, and instant messages(Mugoye et al., 2019). Because the chatbot can provide a promising way for people to get information whenever they need it, wherever they are. Therefore, this study presented an Afaan Oromo text-based chatbot for maternal healthcare using deep learning techniques. Most existing chatbot models for healthcare are designed to diagnose diseases and those presented for maternal healthcare are pattern-matching-based models. They are not advanced when compared to deep learning-based chatbots. Pattern matching-based chatbot responds if and only if the rule is satisfied, while the deep learning-based chatbot learns context from conversation or data. And also pattern matching-based chatbot requires writing the response manually by developers during system development. The modern chatbots model uses the concept of NLP and deep learning techniques. This study introduced a deep learning-based chatbot model that acts as a health consultant expert for maternal healthcare by Afaan Oromo.

1.4. Research Questions

This study focused on three research questions to tackle the problems stated in this study.

- **RQ1.** To what extent does the presented chatbot model contribute to providing relevant information on maternal health problems for pregnant womens?
- **RQ2.** Which deep learning algorithm is more suitable to design a chatbot model for maternal healthcare support?
- **RQ3.** How do we improve the overall performance of the proposed maternal health consultant chatbot model?

1.5. Objectives of the Study

1.5.1. General Objective

The general objective of this study is to develop an Afaan Oromo text-based maternal health consultant chatbot that provides relevant information on pregnancy-related health problems for pregnant women.

1.5.2. Specific Objectives

To achieve the general objective, the following specific objectives are set. To:

- Review the literature on the concepts of deep learning algorithms in modern chatbot development
- Collect and prepare the dataset associated with maternal health consultancy services
- Identify suitable deep learning algorithms for designing a chatbot model for maternal healthcare
- Build the model and search for the optimal model hyper parameters to improve the model performance
- Evaluate the performance of the proposed model for the development of a maternal health consultant chatbot using evaluation metrics
- Develop a prototype for the end-users (chatbot user interface)

1.6. Significance of the Study

The study aimed to develop a chatbot model that is used in healthcare sectors to assist and advise pregnant mothers. There are many beneficiaries from this study, including the Oromia Regional Health Bureau and all pregnant women. The health center has a shortage of qualified health experts to serve all users effectively. For this reason, it is hard to address a better understanding of pregnancy-related health problems and better counseling services for all individuals. Chatbot solutions can provide better awareness to support users with minor issues and suggest those who need medical intervention visit a doctor. It can provide basic details consultancy about pregnancy-related health problems before visiting a doctor. And then, the available health experts will focus on serving users who need medical intervention. That means it reduces the load on health centers and hospitals when there are many users to handle.

The second beneficiary of the study is all pregnant mothers receive a better understanding of their health status throughout their pregnancy time. Today, all mothers want a piece of advice on how to treat and protect themselves from diseases. A chatbot can be considered the best solution to offer awareness and advice to all users simply and efficiently. Generally, it will decrease healthcare costs for users and improve access to maternal healthcare services.

1.7. Scope and Limitation of the Study

1.7.1. Scope of the Study

The study focused on presenting Afaan Oromo text-based consultant chatbot for maternal healthcare. The study aimed to design a chatbot model that provides better understanding and advice for all pregnant women on how to preserve themselves from disease. To reach this aim, the primary dataset was collected from different sources, including ORHB, hospitals, health centers, individual midwives, and websites like https://fayyaa.net/page/1/. This work focused on developing a deep learning-based chatbot model for maternal healthcare support. To design a consultant chatbot model for maternal healthcare, various deep learning models were proposed. And then their performance was assessed using the model performance evaluation techniques. The study also merged two models to improve the performance of the proposed model for the development of a maternal health consultant chatbot.

1.7.2. Limitation of the Study

The presented chatbot model in this study can be suitable for any domain. However, due to time constraints in collecting and preparing the dataset, it is only limited to maternal health consultancy services. The presented study does not diagnose a disease or recommend medical treatment. It only offers consultancy support for pregnant women on disease prevention. Furthermore, the research is limited to monolingual and focuses solely on retrieval and closed domain-based chatbot model development, which is explained in section 2.4. The last limitation of the study is that it only looks at text-to-text conversation and not speech conversation.

1.8. Organization of the Thesis

In this section, the study describes the different topics covered in this thesis. The study is drafted into five chapters.

This chapter contains the thesis's introductory topics, which describe the background of the study, statement of the problem, objectives of the study, motivation, research question, significance of the study, and delimitation of the study.

The second chapter discusses various topics such as a literature review, an overview of the chatbot model and its history, the chatbot application domain, the framework used to design the chatbot model, the types of chatbot models, and an overview of available deep learning algorithms for effective development of the chatbot model, reviewed related works on the domain of chatbot and chatbot for maternal health care, and an overview of an Afaan Oromo language.

The chapter focuses on the methodology, tools, and approaches used to achieve the study's overall goal. This section of the thesis discusses various methodologies concepts such as data collection and preparation methods, data preprocessing, feature extractions and word embedding techniques, model selection methods, several evaluation techniques to evaluate the performance of the designed chatbot model, and tools used to create a friendly user interface.

Chapter four presents all about the design and implementation of the proposed model. The chapter contains different points such as the general architecture of the designed method, proposed data preprocessing, word embedding methods and their implementation, and implementation of the proposed model procedures.

Chapter five discusses the results and an evaluation of the designed models. The best model is chosen based on the results of the proposed model performance and discussed how the proposed model answers the research question and compared the newly designed model to the existing chatbot model and the architecture of the chatbot model prototype. Lastly, the end section of the study explains three different topics. The first is the conclusion of the study, the second is the recommendation, and the third is future work that we intend to improve or recommend to the next researcher.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. Chapter Overview

The chapter is all about the literature review and explains an overview of various chatbot model concepts. The study discusses different points such as an overview of the chatbot model and detailed history of a chatbot, application domain of a chatbot, framework used in chatbot development, types of a chatbot, approaches of chatbot model, and reviews the related works on the concepts of chatbot and chatbot for maternal healthcare, and an overview of Afaan Oromo language in this chapter.

2.2. Background and History of the Chatbot Technology

2.2.1. Background of a Chatbot

Currently, AI continues to grow and is used in many industrial platforms, including healthcare. This study reviews chatbot technology in healthcare. Healthcare plays a vital role in our daily lives to protect human beings from disease and death. Due to the increased number of users in healthcare sectors, there is a need for technological support such as AI-based technologies to serve users efficiently. One prominent area of AI is a chatbot technology that allows communication between humans and computers using natural language like Afaan Oromo, English, Amharic, etc. The idea of chatbot comes from the “imitation” game (Turing test) which was discovered by Alan Turing(Adamopoulou, 2020). The main aim of this game is to determine whether the machine could imitate human behavior like thinking, acting, reasoning, and perceiving. Today, there are various chatbot applications in every industry, such as customer service, business, marketing, agriculture, etc.

A chatbot is computer software that simulates human-like conversation using text or speech through the chatbot user interface. It is a software program capable of communicating with humans by NL. Chatbots have been widely used in different fields, such as customer service, education, marketing, healthcare, etc. Even though the chatbot model is widely used in many fields, it is not under proper implementation in the medical field. However, it is possible to use a chatbot system in healthcare as an assistant system to offer better awareness about health, as a disease diagnosis model, and perform tasks such as making appointments for the patient, etc. The main benefit of using chatbots for industries is that chatbots can answers users' questions anytime anywhere 24/7 service about the domains. We mean that the chatbot can be divided into two categories based on its knowledge domain, such as closed and open domains, as explained in section 2.4. A modern conversational chatbot is designed using deep learning approaches. However, the machine does not understand natural language, due to this reason developing an effective chatbot model is challenging as requires language understanding, contextual understanding, and language processing. Therefore, various forms of AI systems and NLP are required to develop an effective chatbot model.

The complexity of natural language needs AI scientists to provide a model that understands and process natural language with the help of NLP. NLP is a sub-field of AI that explores the capabilities of computers to understand and process human language like Afaan Oromo, English, Amharic, etc. NLP plays a vital role in chatbot development to facilitate human-computer interaction. In this work, various NLP techniques are used, such as tokenization which is used to break up sentences into words to remove unrelated items. For instance, in Afaan Oromo the sentence "What is maternal health?" is defined as "**Fayyaan haadholii maalidha?**" and can be tokenized as '**Fayyaan'**, '**haadholii'**, '**maalidha'**, '**?'**. After the sentence is tokenized into individual words, it is possible to apply different data preprocessing techniques such as stop words removal, stemming, removing of punctuation, and unrelated items from the text data. These all things are used to reduce the burden on the processing time and make learning algorithms fast, give better responses, and allow models to understand the natural language easily.

2.2.2. History of Chatbot Technology

In this section, the advancement of chatbot technology from the basic model to the advanced model was described. The term Chatbot was first conceptualized by the problem as an imitation game or now we called it as Turing Test which was coined first by Alan Turing in 1950(Hutapea, 2017). Based on the idea of the Turing Test, Joseph Weizenbaum created the first chatbot in 1966 called ELIZA at MIT (Adamopoulou & Moussiades, 2020). ELIZA identifies key phrases and pattern matches. Those key phrases must match a set of pre-programmed rules to generate a response for the interrogator. It had limited communication capabilities. However, it served as a source of inspiration for other future chatbots. The PARRY was created by Kenneth Colby at Stanford University in 1972, which impersonated a paranoid schizophrenic(Wei et al., 2018). The PARRY can personate the communication strategy and it was more advanced when compared to ELIZA(Adamopoulou & Moussiades, 2020). Artificial Linguistic Internet Computer Entity was the third chatbot invented in 1995 by Richard Wallace. It is a complicated bot that generates responses based on pattern matching inputs against the pattern and template pairs defined in documents(Adamopoulou & Moussiades, 2020). ALICE was developed using Artificial Intelligence Mark-Up Language (AIML) which is used to write pattern-template pairs documents. There is a big difference between ELIZA and ALICE, which ALICE is comprised of around 41,000 templates and patterns. ELIZA, on the other hand, had only 200 phrases and rules[3]. The other created chatbot is SIRI developed by Apple in 2010 that used as a personal assistant(Wei et al., 2018). SIRI is an application of Apples IOS created as an intelligent personal assistant and makes recommendations.

In recent years, different chatbot technology is proving to be fundamental tool and help in assisting customers. Microsoft Cortana, Amazon Alexa, IBM Watson, Google Now, and Microsoft Tay have taken over the world through the advancement in technology (Machine learning, deep learning, AI)(Wei et al., 2018)(Singh & Thakur, 2020). These all mentioned chatbots are the most common smart voice assistants. Microsoft Cortana is a personal assistant developed in 2014 and performs tasks such as sending emails, identification of time, and plays games, etc(Adamopoulou & Moussiades, 2020). Amazon Alexa is developed by Amazon and it is an intelligent personal assistant (Wei et al., 2018). IBM Watson was created by IBM in 2011; it could understand natural language well enough[4]. Google Now is an intelligent personal assistant created by Google in 2012 which is used to give information about the time of the day, location, and preferences to the user. Microsoft Tay was released by Microsoft Corporation in 2016 to control post inflammatory and offensive tweets(Wei et al., 2018).

[3] *https://www.sciencedirect.com/science/article/pii/S2666827020300062?via%3Dihub*
[4] *https://www.ibm.com/watson/about*

Table 2.1 Summary of Chatbot Overview from 1966 to 2016

Chatbot and year developed	***Approach***	***Architectural model***	***Application***	***Drawback***
ELIZA (1966)	NLP, Pattern matching	Retrieval based model	Tongue-in-cheek simulation of Rogerian psychotherapist	No logical reasoning capabilities, inappropriate responses
PARRY (1972)	Conceptual ontologies, Turing test	Retrieval based model	Mimic the behavior of a person with paranoid schizophrenia	Slower to respond and unable to learn from the dialog
jabberwocky (No w Clever-bot) (1988)	Contextual pattern matching	Generative model	To simulate natural human chat for entertainment	Cannot reply at high speed and work with a huge number of users
A.L.I.C.E. (1995)	NLP, AIML heuristic pattern matching	Retrieval based model	Pattern matching heuristics are used for user input for the human communication interface	Did not have intelligent features
Watson (2006)	NLP, IBMs Deep-QA software and Apache UIMA	Retrieval based model	QA system that won the 2011 Jeopardy competition	Does not process structured data. Supported on English
SIRI (2010)	Java, JS, Objective C, NLP, TTS, STT	Generative model	The computer software serves as an intelligent personal assistant	Difficult to hear the interlocutor

Google Now (2012)	RNN, neural network mechanism	Generative model cute	Google searches for mobile apps and takes steps by transferring queries to web services.	It has no personality Its question may violate the users' secrecy.
ALEXA (2015)	NLP, TTS, STT, Python, Java, node JS	Generative model	It is a personal assistant capable of home automation, entertainment, and setting alarms.	Privacy rights are signed away at purchase, always listening no privacy
Cortana (2015)	Python, Java, JS, NLP	Generative model	The intelligent assistant makes reminders, sends emails, etc.	It can run a program that will install malware.
Microsoft Tay (2016)	Python, Java, node JS	Generative model	It focused on learning from its surroundings through interaction with people.	Controversy on Twitter and caused social media for offensive Impact.

2.3. Application Domain of Chatbot Technology

A chatbot is a software program that can conduct a conversation with a human. It can respond to a question user asks. Initially, it was designed to improve customer service in businesses and industries. Currently, this computer program can be applied to many fields including healthcare, education, marketing, agriculture, etc. Some application areas of chatbot technology are mentioned and explained as follows.

2.3.1. Chatbot in Healthcare

Healthcare plays a vigorous role in our daily lives. The healthcare industries need AI-based technologies to provide service efficiently and effectively. One of such best technology is a chatbot model which is very crucial in healthcare to provide services and health information for the users. Today, chatbots are used in healthcare to impersonate health services such as health consultancy, disease diagnoses, health advisors, nutrition facts, etc. The main goal of designing a chatbot model for healthcare is to help patients self-diagnosing their health status and to understand how to prevent themselves from diseases. In this way, chatbots have a vital role in healthcare for the provision of prevention, diagnosis, and treatment services. In healthcare, there are several prominent chatbot applications[5].

Several previous studies presented on AI-based technology for healthcare industries are focused on disease prediction, disease diagnosis, etc.(Kidwai & Rk, 2020)(Kandpal et al., 2020), but the chatbot can also be used for consultancy and social and mental health support in the healthcare industries. Whenever users feel sick, they visit a doctor or nearby hospital to know the issues they are facing. Research by Google shows that one in every twenty Google searches is about health; this shows the need to receive treatment and health counseling digitally[6]. As result, chatbots come to use to serve patients 24/7 and reshape the service in the healthcare industry by providing preventive, predictive diagnoses and assisting patients.

The study aims to design Afaan Oromo text-based maternal health consultant chatbot using deep learning. The designed chatbot model is to offer better consultancy service for all

---

[5] *https://research.aimultiple.com/chatbot-healthcare/*
[6] *https://blog.vsoftconsulting.com/blog/application-use-cases-of-chatbots-in-healthcare*

pregnant women throughout their pregnancy time in disease prevention on common maternal health problems like danger sign identification, nutritional tips, known emotional signs, etc. This presented study is expected to minimize maternal mortality by supporting mothers in their health problems.

2.3.2. Chatbot in Marketing and Business for Customer Service

Chatbot technology is employed in marketing areas to automate interactions with users. It is primarily designed to answer questions and make conversation with website visitors. Chatbots can help businesses to advertise their products. Chatbots can assist users to automate marketing communication and provide consumers with immediate and timely replies. Thus, the chatbot has many use cases in marketing such as providing information, scheduling appointments, collecting customer reviews, etc. (Dilmegani, 2021).

2.3.3. Chatbot in Education

The chatbot has become a critical technology in e-learning in recent years. It is very important in the education sector to perform tasks like student support, online tutoring, generating results, teacher assistants, etc.[7] In this manner, the chatbot can alter the educational system. Several studies have shown that chatbot technology can be effectively applied in the educational sector. The chatbot concept may provide several benefits to the educational system, such as material integration, easy access, immediate support, engagement, etc(Okonkwo & Ade-ibijola, 2021)

2.3.4. Chatbot in Agriculture

A Chatbot in agriculture is designed to assist the farmers on how to enhance their productivity. Since agriculture is the key to human life, it is critical to support farmers and allow agriculture industries to increase production, customer service, etc., and then farmers will receive information on how to improve their productivity. The chatbot can offer information that helps farmers to improve their yield based on weather, rainfall, and soil type of the environment and suggest which crop is more suitable to grow (K et al., 2019). As discussed in the cited study,

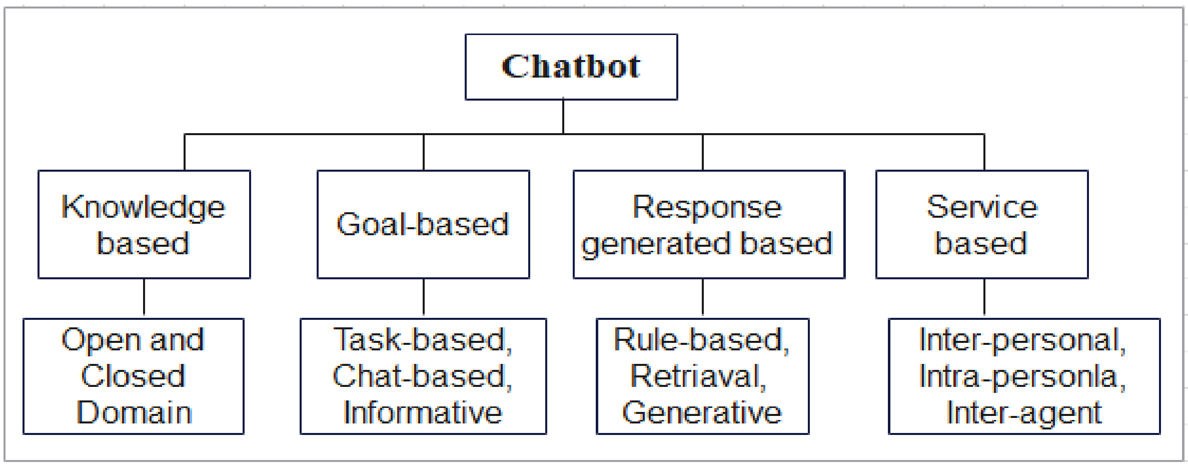
[7] *https://yellow.ai/resources/articles/applications-of-ai-chatbots-in-education-sector*

the chatbot can help users by answering their basic questions on farming, and they understand the type of crop to be grown based on different parameters.

2.4. Types of Chatbot

Chatbots can be categorized into different types based on different parameters, such as the knowledge domain, the service provided by the chatbot, the goals, and the design approach. This section will discuss several types of chatbots.

Fig.2.1 Type of Chatbots



A chatbot is categorized into two based on the knowledge domain that can access or data trained on, such as closed and open domain chatbots(Adamopoulou, 2020). The closed domain type of chatbot considers just a certain knowledge domain and may fail to answer all user questions, whereas the open domain type of chatbot discusses general topics. Chatbots that can respond to any question for the user from the general domain are called generic chatbots, Chorus can be considered a generic chatbot(Adamopoulou & Moussiades, 2020). It is hard to develop than closed domain chatbots. Chatbots that can operate more than one domain like Guardian, CRQA, and AskWiz are examples of open domain chatbots[8]. The closed-based chatbot is used in this study, to design a maternal health consultant chatbot.

Based on the main goal, chatbots are classified into three types: informative, chat-based, and task-based chatbots(Adamopoulou & Moussiades, 2020). Informative chatbots are created to

[8] *https://doi.org/10.1016/j.mlwa.2020.100006*

offer the user information that is accessible from fixed sources. These kinds of chatbots have been used when users need to get specific information from a fixed source. Chat-based chatbots look natural conservation with the user like another human being would do. They attempted to respond appropriately to the question raised for them. Task-based chatbots are referred to as task-oriented chatbots that are used to accomplish a certain task. Chatbots like room booking and FAQ are examples of task-oriented chatbots(Adamopoulou & Moussiades, 2020).

The third classification of chatbots is based on the model's design method such as rule-based, retrieval-based, and generative-based models(Hussain et al., 2019). Rule-based chatbots select an answer based on a set of rules(Adamopoulou, 2020). The first chatbots to use pattern matching were ELIZA and its successor ALICE. Generative chatbots are producing a response primarily based on current user messages and preceding messages. However, it is hard to build and train(Hien et al., 2018). They are effective for engaging users in casual open-domain dialogues. Generative models are no longer best for closed domain communications in which ruled-based chatbots and retrieval models are suitable. Retrieval based chatbots model provides much flexibility and it uses keyword patching, and machine learning to select the most appropriate response(Hien et al., 2018). In this work, a retrieval-based chatbot model has been used using deep learning algorithms. Finally, a chatbot can be classified as interpersonal, intrapersonal, or inter-agent based on the service provided by the chatbot(Adamopoulou, 2020).

2.5. The Available Frameworks for Chatbot Technology

As discussed in section 2.2.2, chatbots have been in existence started from 1966 and there have been several advancements in this area. Earlier chatbots have the exhaustion of using natural languages like text and speech. In the recent epoch, several frameworks are introduced by Google, Microsoft, and Amazon to enable any industry (business) to build a chatbot for their purposes. A chatbot framework helps to design and binding modules of the chatbot system. Modern chatbots make use of NLP and DL concepts(Kandpal et al., 2020). In this section different widely available frameworks for chatbot development are discussed.

As described in (Nuruzzaman & Hussain, 2018) dialog flow was developed by Google and lets developers, build chatbots and voice apps. It supports all fundamental messaging channels such as Skye, Facebook, Twitter, etc.[9] It can also analyze speech and text-based inputs(Kandpal et al., 2020). Dialog flow is built on Google Cloud and can quickly grow to service millions of users. The one drawback of this framework is handling context and intent simultaneously is very difficult(Borah et al., 2019).

Microsoft Bot Framework is a great chatbot framework with machine learning that allows developers to build and deploy chatbots on websites, apps, Microsoft Teams, etc. (Krishnan, 2020). The user input is semantically tested against previous information to generate relevant intent, and it also includes the entity idea to learn about the conversation. This framework offers services for developing intents, entities, and agents. Though, it does not offer a visual illustration of the dialog flow. Microsoft bot platform comprises two components such as channel connectors and Boatbuilder SDKs. The study (Almansor & Hussain, 2020) described the two drawbacks of the Microsoft bot framework, first, it does not allow developers to control context parameters. Second, it lacks a wide range of domains.

IBM Watson is a conservational interface provider and enables developers to develop assistant applications. Watson Assistant is provided by IBM cloud for building chat interfaces into any device or application(Kandpal et al., 2020). It uses machine learning to answer to NLP input and is developed on a neural network that consists of three main components namely, Entities, Intents, and Dialog.

Rasa Stack, often known as the Rasa platform, is an accessible machine learning-based chatbot development framework(Krishnan, 2020). Rasa Stack works on two open assets namely, Rasa NLU which is the NLP of the bot, and Rasa Core which works on the inputs based on intent and entities. The main benefit of using the Rasa Stack framework is chatbot can be installed on your server[10].

[9] *https://cloud.google.com/dialogflow/es/docs/integrations*
[10] *https://cedextech.com/chatbot-development-frameworks/*

2.6. Approaches of Chatbot

There are two methods of chatbot development based on the algorithms and techniques used: pattern matching and the machine learning approach. The two chatbot approaches are described as follows.

2.6.1. Pattern Matching Approaches in Chatbot Development

The pattern matching approach specifies the manually constructed rules in chatbot development. It matches user input to a condition and selects a pre-defined answer from a set of responses. In chatbot development, the pattern matching technique is used to identify user input as <pattern> and provide an appropriate answer stored in <template>. The <pattern> <template> pair used in pattern matching approach is constructed manually. The approach is used in both early and advanced chatbots. The sophistication of the procedures utilized in them varies. For example, the ELIZA chatbot used simple rules when compared to ALICE(Hussain et al., 2019), which uses more complex pattern matching rules. The approach has a scaling problem because all possible patterns are built manually. The pattern matching approach first was used by ELIZA, and ALICE chatbots, and PARRY, Chatterbot, and Jabberwacky also used this approach(Adamopoulou & Moussiades, 2020). There are three most common languages such as AIML (Artificial Intelligence Mark-up Language), Rivescript, and Chat script for the implementation of chatbots with pattern matching.

AIML is an open-source language developed from XML and used to build a conversation flow in chatbots(Adamopoulou & Moussiades, 2020). The AIML was created from 1995 to 2000, and the first chatbot that used AIML is ALICE. AIML is divided into three categories: chatbot rules, patterns to describe user input, and templates to describe the chatbot's answer. A simple example of AIML is given in Fig.2.2.

```
<Category>
<Pattern>What is a chatbot? *</pattern>
<template>Chatbot is software</template>
</category>
```

```
<Category>
<Pattern>Chatbootiin maali? *</pattern>
<template>Chatbootiin
softweeriidha</template>
</category>
```

Fig.2.2 Example of AIML for Chatbot Development

Chat script is a rule-based chatbot creation expert system that was invented in 2011. It is a successor of the AIML and uses concepts that are getting together of similar words considering the meaning and POS. Chatbots like Suzette, Rosette, Chip vivant, and Mistsuku are implemented using Chat script language(Adamopoulou & Moussiades, 2020). Fig.2.3. shows an example of a Chat script language.

```
Concept: ~ greeting [Akkam nagaadha]
Concept: ~ good_day [ Nagaan oolaa]
Topic: ~ chatbot [ ~greeting]
t: ~greeting
#! Hello
u:(* ~greeting *) [Hi][Hello], [maqaan kee eenyu? /What is your name?]
#! Maqaan Koo / My name is
u: (*name* is_) Bagan si baree_0! / Nice to meet you_0!
$username='_0
#! Guyyaa Gaariin siif hawwaa / I wish you a good_day
U:(<<!not* good day*>> good_day $usernames
```

Fig.2.3 Examples of Chat Script Language

The Rivescript was created in 2009, and it is an open-source and line-based scripting language(Adamopoulou & Moussiades, 2020). In this scripting language, the user input and response are represented by the symbol + and - respectively. It consists of a few simple rules that are combined in powerful ways to create a successful chatbot identity. Fig.2.4. shows how the Rivescript has been written to design a chatbot.

```
+Hello

-Hi! Maqaan kee eenyu? / What is your name?

+Maqaan koo* / My name is*

–<set name=<formal>>Baga wal baranne / Nice to meet you, <get name>!

+Good day/ Nagaan oolaa<get name>
```

Fig.2.4 Example of Rivescript Language

2.6.2. Machine Learning Approaches Based Chatbot

Today, it is the epoch of intelligent machines that are used widely to complete tasks. With the improvement of AI, machine learning, and deep learning, machines have started to impersonate humans. Changes in AI enhance the capabilities of chatbots to simulate human agents in dialogue. ML is a part of AI that uses algorithms and data to complete tasks and demonstrate its utility in our everyday activities. One field of ML is a conversational model initiated by NLP which is known as a chatbot. A chatbot model that used the Machine learning (ML) approach is modern chatbots that do not require a predefined rule as pattern matching approaches.

Machine learning approaches based on chatbots are called self-learning chatbots that can be categorized into two; retrieval models and generative models chatbots(Nuruzzaman & Hussain, 2018). These kinds of chatbots employ machine learning techniques to learn new things from data. From these algorithms, CNN, RNN, and DNN are some deep algorithms that have been used in modern chatbot development(Nuruzzaman & Hussain, 2018). Deep Learning is the one sub-field of ML which involves multiple data processing layers to allow the machine to learn from data through various levels of abstraction. It has gained more attention for NLP tasks from the ML and NLP research communities(Alshahrani & Kapetanios, 2016). Modern chatbots use the concepts of natural language understanding and apply deep learning algorithms(Kandpal et al., 2020).

Deep learning models use artificial neural networks to imitate how the human mind interprets and learns from data. It is one part of machine learning that uses a high level of abstraction to process data. Most DL method uses an ANN to build intelligent models and solve complex problems. ANN uses the approach of how the human brain computes information. A deep neural network is composed that has multiple hidden layers. Neural networks have three layers as depicted in Fig 2.6, and every layer consists of multiple nodes (neurons). Neurons are the building components of neural networks. Neurons in one layer are interconnected to neurons in two surrounding layers, which help to flow data between layers. Lines used to connect every neuron are called weights, denoted as wij.

- **Input layer*:*** the layer consists of the list of input features and often it is called a visible layer. It accepts input data and introduces it to the rest of the network. It has a responsibility to initialize the first weights.
- **Hidden layer*:*** hidden layers are the hidden pulp of the neural network that is responsible for the better performance and complexity of the model. Tuning the network with the correct hidden layer and neurons in each hidden layer needs more attention because hidden layers are where the operation is done in every single neuron, as shown in Fig.2.5. Where w denotes weight, x denotes input value, b denotes bias, and δ represents sigmoid activation function.

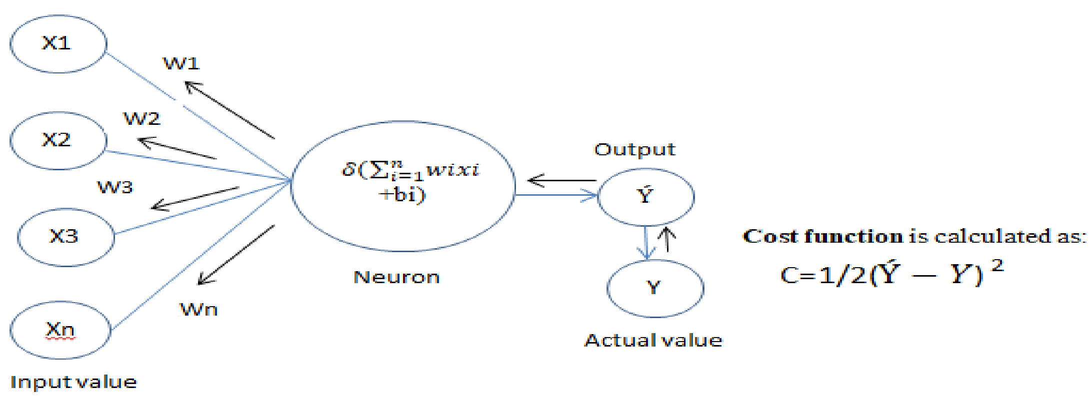


Fig.2.5 The Neurons of Hidden Layers Operation(Patil et al., 2020)

- **Output layer*:*** the last layer and where the data is generated out from the model. This layer has the same number of neurons as it does outputs. If the designed model is a regressor, the output layer has only one node.

In this study, the dataset was prepared with multiple patterns and responses that belong to different classes (tags). And then the patterns were preprocessed, embedded into vectors, and then trained with the proposed deep learning model. Then the model classifies them into appropriate classes to answer the response that belongs to that tag for the users. Assume that, if the user input is Fayyaan Haadholii maalidha naaf ibsuu dandeessaa? Then the model processes this input question through different steps to generate a response, as shown in Fig.2.6. Lastly, the model classifies the input pattern into the appropriate tag with the highest confidence value to fetch the random response that belongs to that tag. At the output layer, the SoftMax activation function is used because the problem at hand is multi-class classification.

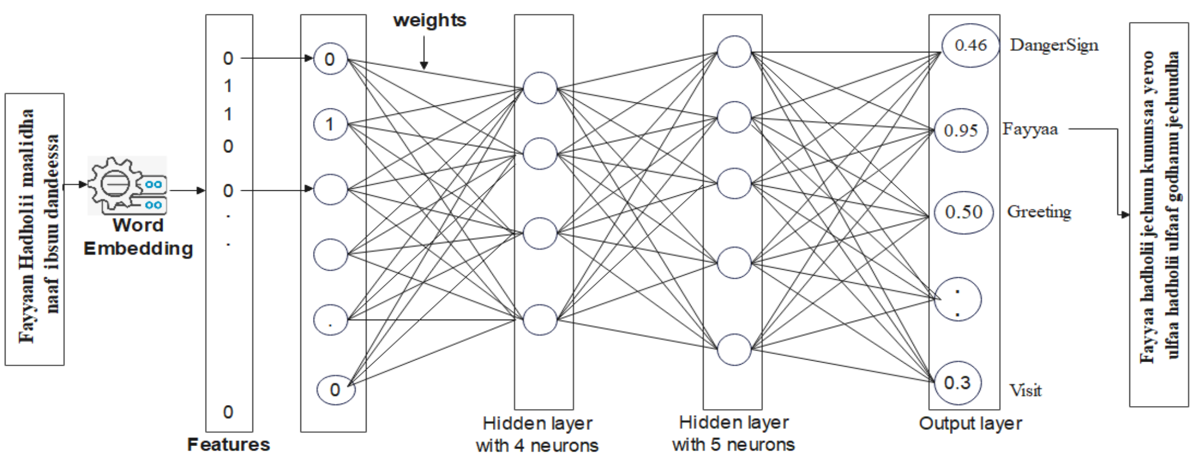


Fig.2.6 How Neural Networks Operate Textual Data for Chatbot Development

The input layer receives the vector-represented data, and all values are computed sequentially using the network layers. The output of one layer is utilized as input for the subsequent layer. The output values are computed by multiplying the input value and the weight for every unit in a current layer and adding them together. After that, a non-linear activation function like rectified linear unit (ReLU), sigmoid, etc., is applied to buffer the data before being sent to the succeeding layer. As the number of hidden layers rises, the output data obtained in the final layer becomes more abstract than the raw input data. As a result, setting the number of layers and neurons in the hidden layers needs more consideration to optimize model performance.

2.7. An overview of Afaan Oromo Language

Language is a method of communication between humans and allows people to interact with thoughts and knowledge and express themselves through signs, written symbols, and ordinary spoken language. The primary goal of the language is to facilitate communication (convey information) between persons. The Oromo people have their own language which is called Afaan Oromo or Oromiffa. It is an Afro-Asian language family branch that belongs to the Cushitic group and is Africa's third most broadly spoken language(Tolessa, 2013). More than 36 million people speak the language[11]. Afaan Oromo language is spoken in many areas of Ethiopia and bordering countries like Djibouti, Egypt, Kenya, and Somalia. The language is mostly spoken in Oromia by the Oromo peoples which are the largest ethnic group in Ethiopia. It is a language of primary education and working language in Oromia. Until the 1970s Afaan Oromo was written either in the Latin alphabet or geez. In early 1970, the Latin alphabet was chosen by OLF as the official alphabet to write Oromiffa[12].

2.7.1. Afaan Oromo Word Formation

Before 1991, the Afaan Oromo language lacked a script. In 1991, it was formally started written by a Latin alphabet called "Qubee"[13]. With the adoption of Qubee, more texts were written in the language. The Afaan Oromo contains 33 letters that represent different sounds and has both capital and small letters. Of these letters, 7 is a compound letter, as described in Table 2.2 The language has 5 short vowels such as a, i, e, o, and u, and five long vowels such as aa, ii, ee, oo, uu, and the remaining letters as consonants that are used to distinguish one word from another. The difference in length of vowels is to change the meaning of the word, for example, **hara** means lake and **haaraa** means new. The Afaan Oromo's alphabet letters and International Phonetic Alphabet is shown in Table 2.2.

[11] *https://en.wikipedia.org/wiki/Oromo_language*
[12] *https://omniglot.com/writing/oromo.htm*
[13] *https://en.wikipedia.org/wiki/Oromo_language*

Table 2.2 Afaan Oromo Letters with their IPA

Qubee		***IPA***	***Qubee***		***IPA***	***Qubee***		***IPA***
A	a	/a/	L	l	/l/	W	w	/w/
B	b	/b/	M	m	/m/	X	x	/t/
C	c	/č/	N	n	/n/	Y	y	/y/
D	d	/d/	O	o	/o/	Z	z	/z/
E	e	/e/	P	p	/p/	CH	ch	/č/
F	f	/f/	Q	q	/k/	DH	dh	/đ/
G	g	/g/	R	r	/r/	NY	ny	/ň/
H	h	/h/	S	s	/s/	PH	ph	/p/
I	i	/i/	T	t	/t/	SH	sh	/š/
J	j	/ğ/	U	u	/u/	TS	ts	/s/
K	k	/k/	V	v	/v/	ZH	zh	/ž/

2.8. Related Works on Chatbot

Many researchers have sought to enhance the performance of chatbots in many businesses and industries to assist users. Over the past years, several models have been introduced by different researchers. However, the deep learning models are still in their early phases in chatbot development(Kandpal et al., 2020). This research study is aimed to develop an Afaan Oromo text-based maternal health consultant chatbot model that offers consultancy services about pregnant-related health problems. As discussed in section 2.7 of this thesis, the Afaan Oromo language has millions of speakers in Ethiopia. With this fact, this study is presented the Afaan Oromo text-based chatbot, which offers a better comprehension of maternal health problems. In this section, the study reviewed different related studies on chatbots as follows:

Kandpal et al(Kandpal et al., 2020) presented “Contextual Chatbot for Healthcare Purposes (using Deep Learning)” to diagnose basic healthcare problems for some diseases. They have used FFNN to design a chatbot model that diagnoses what disease the patient might be facing. And they used the BoW technique to represent sentences with numerical data for the NN model. However, the researchers have not evaluated the model performance, and they do not specify the model accuracy. The prepared dataset is only to train the model, and they have not split the dataset to evaluate the model performance. The authors used the simple feature extraction technique called BoW.

Vamsi et al(Vamsi et al., 2020) work on “A Deep Neural Network Based Human to Machine Conversation Model”. They introduced a deep learning approach of creating a chatbot which is called DNN, with multiple hidden layers to learn and process the data. The simple dataset had been taken from the open-source Kaggle healthcare services competition with 14 different tags to train the deep neural network model. This dataset is preprocessed and converted into numbers using a one-hot encoding method and given as an input to the neural network. In this work, different optimizer algorithms are evaluated and select the best with the neural network model. The authors used train-test split ratios to train and test the model with a small dataset. However, due to the smaller size of the dataset, the researchers have not used the model overfitting handling method.

Kidwai & Rk(Kidwai & Rk, 2020) presented the development of a diagnostic chatbot system to diagnose user situations based on their symptoms. They have used a decision tree ML algorithm to diagnose disease for the users and obtained 75% accuracy. In this work, the authors trained the model on 50 different diseases and 150 symptoms labeled datasets. The developed chatbot collects the symptoms from users using speech or text in the form of doctor-to-patient conversion to diagnose the disease. However, even if the presented model is better for disease diagnosis, the response will depend on the users' response at each step. In this work, if the patient feels only a few symptoms, the model will continue to ask a question until the end of related symptoms. Due to this, users may not be satisfied because it is boring to use.

Reddy Karri & Santhosh Kumar(Reddy Karri & Santhosh Kumar, 2020) studied “Deep Learning Techniques for the Implementation of Chatbots”, and they discussed and compared the different technologies used in chatbots. In this study, a comparison has been made between several chatbots so far developed, like jabber wacky, Eliza, and Alice, in terms of speed, interoperability, Turing test, and scalability. The authors have used the Seq2Seq model with a bag of word features extraction method to overcome the limitations of those chatbots. However, the authors only feed the dataset to model and test the developed chatbot model through sample conversion. They have not evaluated the model using a test dataset. The authors used a big dataset size to train the model, but they haven't used the model overfitting handling technique.

Mugoye et al(Mugoye et al., 2019) studied “Design of Conversational Agents in Maternal Healthcare Support” that advises expectant mothers throughout their pregnancy time. The researchers have developed a conservational agent that collects different symptoms from users and responds to them on what types of disease they have and appropriate tips. In this study, the combination of dialog manager, API, reinforcement learning, and knowledgebase has been used to develop a multi-agent conversational system. However, since the system is a knowledgebase system, the model will return the response if the user request is matched with if condition.

Segni Bedasa(Bedasa, 2021) proposed “Developing Afaan Oromo Text-based Chatbot for Enhancing Maize Productivity”. The author has used DNN and Seq2Seq and achieved the best accuracy of 84.4% in DNN and developed a chatbot that provides vital information for farmers on how to enhance their maize productivity. The researcher used a small-size dataset with 25 tags and used TF-IDF feature extraction. However, even if the presented model can answer the question for the users, isn't meant that the chatbot is trained to answer any question that the user asks. The authors did not set the threshold probability for the question to which the system doesn't have answers. In addition, the researcher hasn't used model overfitting handling techniques.

Fekadu Eshetu(Eshetu, 2021) presented “Design of a Bilingual Chatbot that Assists Ethio-Telecom Customers on Customer Services”. The developed model assists the Ethio-Telecom customers with customer services in two languages Afaan Oromo and Amharic. The author used Bi-LSTM algorithms and found 82.6% accuracy with the L2 regulation technique, which is used to handle model overfitting. However, even if this presented model is better to assist Ethio-telecom customers with customer service, the researcher has not compared different optimizer algorithms to improve the performance of the model. There are several optimizer algorithms, and sometimes Stochastic gradient descent(SGD) outperforms when compared with Adam in the study by(Vamsi et al., 2020), and he didn’t use weight initializers.

2.8.1. Summary of Related Works

In general, many researchers have attempted to improve the performance of chatbots using the state-of-an-art approach to introduce a chatbot solution for all industries. The above-reviewed studies have been conducted on developing a chatbot for many different industries, including healthcare. However, the previously presented studies for maternal healthcare are a kind of knowledgebase system which is not advanced when compared to deep learning-based chatbots. Consequently, this study aimed to develop a maternal health consultant chatbot for maternal healthcare support using deep learning models. In addition to this, the proposed study intended to improve the performance of the existing chatbot model using different techniques such as merging two deep learning models, using weight initialization techniques, and searching for the right network hyperparameter for the proposed model using hyperparameter tuning, etc.

Table 2.3 Summary of Related Work

No	***Titles of the papers***	***Authors and publish year***	***Algorithms with accuracy***	***Gaps***
1	Contextual Chatbot for Healthcare Purposes (using Deep Learning)	Kandpal et al., 2020	FFNN Accuracy not specified	➢ The authors employed the BoW feature extraction technique, which does not consider the words contextually. ➢ No model overfit handling method is used
2	A Deep Neural Network Based Human to Machine Conversation Model	Vamsi et al., 2020	DNN Accuracy not specified	➢ Handling model overfitting by changing the number of datasets splitting ratios ➢ They haven't used a threshold value. Hence, the bot may answer inappropriate responses to unrelated users' questions
3	Deep Learning Techniques for Implementation of Chatbots	Reddy Karri & Santhosh Kumar, 2020	RNN Accuracy not specified	➢ Researchers have not used any model evaluation techniques, instead, they have checked whether the model responds to the user question or not. ➢ The authors used a big dataset size, but they haven't used the model overfit handling technique. ➢ The authors employed the basic feature extraction approach known as BoW.
4	Design and Development			➢ The model diagnoses the disease if the user

	of Diagnostic Chabot for supporting Primary Health Care Systems	Kidwai & Rk, 2020	Decision Tree (75% accuracy)	answers all questions asked by the bot. If the user feels a few symptoms, the model does not work.
5	Conversational Agent's Role in Maternal Healthcare Support	Mugoye et al., 2019	Dialog flow Knowledgebase system	➢ The system will not respond if the user question is not matched with the if-condition.
6	Developing Afaan Oromo Text-based Chatbot for Enhancing Maize Productivity	Segni Bedasa (2021)	DNN (84.4%) and Seq2Seq (80.0%)	➢ The author did not specify the threshold probability for candidate response. ➢ The researcher has not used the model overfit handling techniques. He used train test split ratios
7	Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services	Fekadu Eshetu (2021)	Bi-LSTM (82.6%)	➢ The author did not compare the algorithms of different optimizers to achieve better results. ➢ The author did not use weight initializer techniques, which are used to adjust the initial weights correctly to ensure the model with better accuracy

CHAPTER THREE

3. MATERIALS AND METHODS OF THE PROPOSED MODEL

3.1. An Overview of the Chapter

The methods, approaches, tools, and procedures used to accomplish the study's goal are all explained in this chapter. In this section, various methods such as data collection and preparation, data preprocessing, feature representation techniques, model selection methods, different evaluation techniques to evaluate the performance of the designed chatbot model, and tools used to develop a proposed model and graphical user interface, are discussed.

3.2. Research Design

The research design refers to the methods and techniques used to answer the study's research questions. Also, it includes the type of research to be designed and indicates the procedures used to conduct the study, as depicted in Fig.3.1. The research design designates the type of study carried out. In this study, we design an experimental research approach to select the best model for maternal health consultant chatbot development using newly collected data. The dataset was collected and prepared to address a research question empirically. The research design procedures we used to conduct this study are explained as follows.

- ***Research area identification:*** Machine learning has so many applications in various areas, including healthcare. The study identifies the healthcare fields, particularly maternal healthcare, to explore the capability of the chatbot model for consulting and supporting users.
- ***Literature review*:** This research now goes over what was presented in the previous study. Different previous studies, approaches, methods, and techniques will be reviewed to use them, identify the gap, and fill in this presented study.

- **_Gap identification:_** Identifying a gap is a way of identifying research gaps or gaps not covered in previous studies.
- **_Setting research question and objectives:_** At this step, the research question and research objective are formulated to fill the identified gap or to solve a research problem.
- **_Data collection and preparation:_** The dataset was collected, prepared, and preprocessed as suitable for the model to answer the research questions experimentally.
- **_Build and train the DL models:_** After the dataset is ready for the model, the next step will be to build and train the model with the prepared dataset.
- **_Model Evaluation:_** The performance of the proposed chatbot model is evaluated using various model evaluation techniques, and the best-performing model is identified.
- **_Developing chatbot UI:_** once the best model is known, the next step is developing a chatbot user interface through the end-user communicating with the model. Finally, use the chatbot model as a consultant system that simulates a maternal health expert to offer a better understanding of pregnant-related health complications. The presented model can be reachable to the end-users by deploying the model into the webserver, telegram, and other chatbot-supportive platforms. Fig.3.1, depicts the research design for this study.

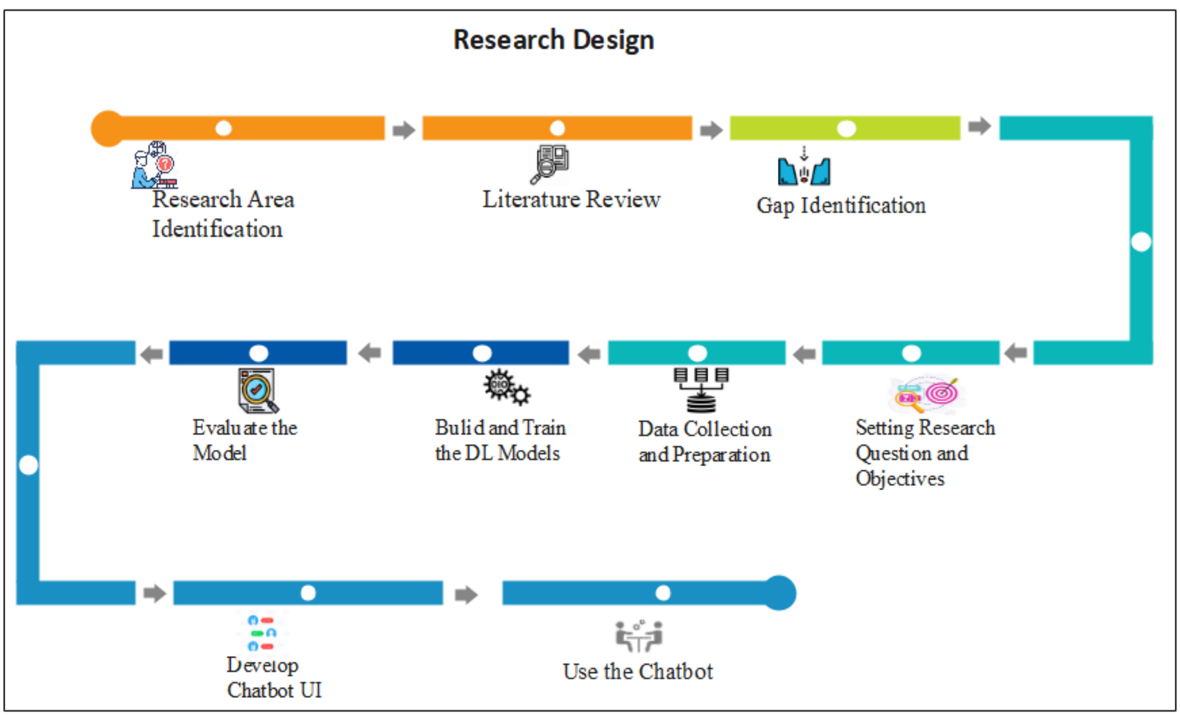


Fig.3.1 Research Design

3.3. Data Collection Method

The most common challenge of studying natural language processing in deep learning is a lack of sufficient data. The deep learning model accuracy depends on several factors, the main of which is the quality of the dataset and the size of the dataset. To properly train the proposed deep learning model, the dataset must be accurate, well-labeled, well enough, and meet the expected outcome. The primary dataset for this study was collected from the Oromia Regional Health Bureau which is responsible for offering a full package of preventative, curative, and reconstructive health services to the community at large in Oromia regional state(*Oromia Regional Health Bureau*, n.d.), individual midwives, health canter, online websites[14], and some hospitals in Oromia. Since there is no available dataset for this purpose previously.

The collected dataset must be reviewed or checked by a maternal health expert. And it was collected for only preventive purposes, and the designed model does not include any medical suggestions. Therefore, the collected dataset is associated with maternal health consultancy services on how to protect themselves from disease and morbidity. There is a guideline that is distributed to all health centers and hospitals from which the services are provided for pregnant women. This guideline serves as a data standard for data preparation. The prepared dataset for this study is associated with antenatal care (ANC), nutritional tips, family planning, danger signs identification, follow-up during pregnancy, etc.

3.4. Data preparation techniques

Data preparation refers to any form of data organizing done on the data for further processing. It is a very important step to prepare the data to make it more suitable for the learning model. The collected data must be filtered and stored in a format that is suitable for the learning models. Data preparation format depends on the purpose; it can be tabular, unstructured, or JSON. Dataset formats differ according to the use cases. In this study, the dataset was prepared using JavaScript Object notation (JSON) that can be easily created by machine and is completely language independent(Kandpal et al., 2020). The JSON file used for this study contains multiple classes. The classes are called tags, and each tag contains multiple patterns

[14] *https://fayyaa.net/page/1/*

(questions) and responses. Tags are keywords assigned to patterns and responses. Patterns are questions posed by the user to the chatbot. The sample dataset preparation format for this study is shown in Fig.3.2.

```
{'intents': [{'tag': 'Introduction',
   'patterns': ['Hey👋', 'Hello👋', 'Hi👋'],
   'responses': ['Hi, baga Nagaan dhufte.']},
  {'tag': 'Greeting',
   'patterns': ['Akkam?',
    'nagaadha👋?',
    'Akkam jirta?',
    'Nagaa keedha?',
    'Fayyumaa?',
    'Jirtaa?',
    'jirtuu?',
    'akkam jirtuu?',
    'Nagaa qabdaa?',
    'Nagaa jirtaa?',
    'Naguma keedha?',
    'Nagaa jirtuu?'],
   'responses': ['Ani nagaa koodha, Maal si gargaaruu?',
    'Nagaa akkam, Maal si gargaaruu?']},
```

Fig.3.2 Sample Example of Data Preparation

In the above sample example, there are two tags, Introduction and Greeting, and each of them contains patterns and responses. The first pattern contains three user inputs ‘Hey’, ‘Hello’, and ‘Hi’, and also two responses such as ‘Hi’ and ‘Baga nagaan dhuftan’. If the user asks one of these questions, the model randomly selects one response from the two responses. Additionally, if the user input does not match word by word with trained patterns, the model will give the response that belongs to the tag that has the highest probability. Table 3.1, describes all the datasets that the study used in this research.

Table 3.1 Description of Dataset

Contents	*Quantity*	*Description*
Tags	103	Tags are keywords assigned to patterns and responses to training chatbots.
Pattern	1962	Patterns are questions posed by the user to the chatbot.
Response	119	The response is the appropriate answer for the corresponding question.

3.4.Data Preprocessing

Data preprocessing is the main step in chatbot development as many researchers do that. It is the first and critical step while developing a learning model to make the data more suitable for the model. It contributes to improving the dataset's quality. The model cannot directly use data collected from the real world; because it contains noise and has an unorganized form which is not suitable for the model. So that to train the model effectively, data preprocessing must be applied to the datasets, and then the wholly cleaned data is ready for the model. Before developing the deep learning model, the data preprocessing techniques are employed. Various data preprocessing techniques were utilized in the study, including removing unrelated items and punctuation, eliminating stop words, normalization, and tokenization.

Data cleaning- Data cleaning is referred to as eliminating noise data from the dataset to make it more understandable by the model, such as removing punctuation marks, white spaces, removing extra symbols, etc. They do not show any significant contribution, rather than increasing the learning time of the model.

Stop words- all words found in the documents are not equally important for designing a chatbot model. Some words are irrelevant and they are not used to determine the users' questions into the appropriate tag. Those words must be detached from the dataset to reduce the learning time of the model. The Afaan Oromo stop words were collected from different previous studies(Khanna, 2021)(Eshetu, 2021) and prepared for this work. They are the most prevalent terms in any language like pronouns, prepositions, conjunctions, articles, and particles. Words like fi/and, kana/this, Akka/like, kanaaf/so, siif/for, irra/on, sun/that, isa/he, kee/yours, etc. are some commonly occurred in Afaan Oromo stop words.

Normalization- Users may write the same word differently, and the machine understands those words differently. This problem can be handled by text normalization, which transforms similar words written in different formats into one unique word. It helps the models to recognize the same word written in different ways. The most common and easy technique used in normalization is lowercasing. The model represents the word "Fayyaa", "FayyAA", and "FaYYaa" differently. So that, lowercasing is applied to the dataset to convert each pattern into a small letter. In addition to this, a regular expression is used to normalize words that have similar meanings but are written in different formats. For instance, the word haga/hanga, erga/ega, baayee/baayy'ee, osoo/otoo, mini/miti, etc., have identical meanings in the Afaan Oromo language, they have normalized into one single word.

Tokenization- When working with a textual dataset, tokenization is the most elementary thing performed on the text data. In-text preprocessing step, the patterns must be tokenized into tokens using spaces. For instance, "Hubannoon haadholii ulfaaf kennamu qabu maal fa'a?" is defined as "What is the awareness provided for pregnant mothers?" This sentence is tokenized as 'Hubannoon', 'haadholii', 'ulfaaf', 'kennamu', 'qabu', 'maal', 'fa'a', '?'. The fully cleaned data must be converted into numerical data using the text feature extraction or word embedding methods since the deep learning models do not directly work with text data.

3.5. Feature Extraction Methods

Text feature extraction is a way of taking the list of words and converting them into vectors and is also used for feature dimension reduction to develop the high-performance model. The two types of feature extraction techniques are discussed below.

3.5.1. Bag of words

This method considers only the lexicon of known words and the existence of those known words within the document. It doesn't consider the structure or sequence of the words[15]. The vector representation of documents is increasing with vocabulary size. This technique of feature extraction is not used in this study because of its limitation. Some shortcomings of this method are sparse representations, discarding word order, and ignoring the context(Jason Brownlee, 2017).

3.5.2. TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF is a well-known technique, and its performance is still even comparable with other novel techniques(Kadhim, 2019). TF-IDF solves the problem of BoW methods by considering the frequency of the words and how frequently the words occur in all documents. This technique is also not used in this study because it ignores the word order and semantic relation of words. It does not recognize compound words as a single word. These can affect the

[15] *https://www.mygreatlearning.com/blog/bag-of-words/*

accuracy of the learning model, and we lose extra information like the semantics, word order, and context around close words in every text.

3.6. Word Embedding

Word embedding is the dispersal representation of a word with a vector in many NLP tasks like automatic machine translation, chatbot, etc. Word embedding methods such as word2vec and GloVe are available(Rendel et al., 2016). The Word2Vec algorithm inserts semantically related words into the vector space nearby. The CBoW and skip-gram method are the two types employed in the word2vec model. The skip-gram method works as, given the target word and predicts the context neighbor words as output, as shown in Fig.3.3.b, and it works well for small datasets, the CBoW approach is the opposite of the skip-gram method, it predicts the target word with given the neighbor window size words, as shown in Fig.3.3.a.

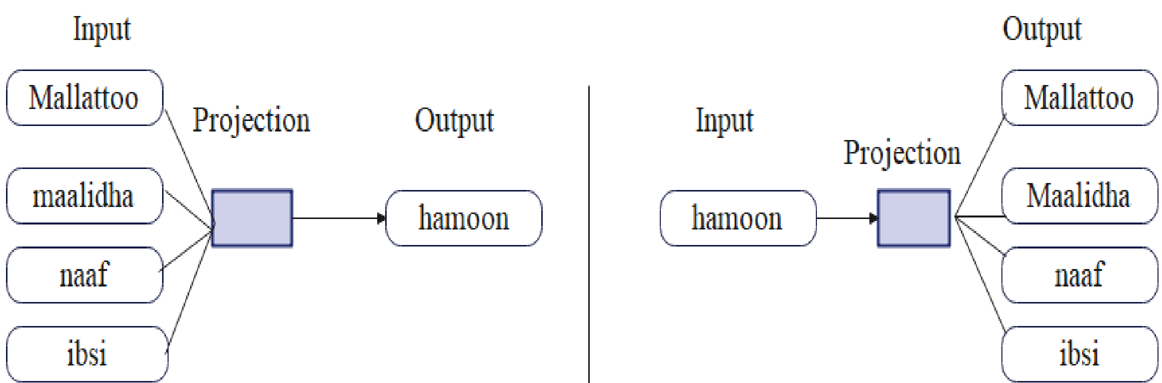


Fig.3.3.a. CBoW

Fig.3.3.b. Skip-gram

Fig.3.3 Types of Word2Vec Model

3.7. Model Selection Techniques

Advanced (modern) chatbot models make use of several deep learning algorithms. Due to the non-existence of perfect models to solve some particular problems, anyone cannot answer which model will perform the best before experimenting with them. This research established some criteria for choosing appropriate algorithms for the maternal health consultant chatbot model development.

The first criterion is the nature of the problem to be solved; is it classification, prediction, or regression? The proposed chatbot model for maternal healthcare support is a multi-class classification problem. The dataset was created as a JSON file with multiple tags. Every tag contains multiple patterns, which are users' input, and the output is the class to which the pattern belongs. Therefore, the study deals with multi-class classification problems. For multi-class classification tasks, the deep learning models are better than machine learning models(Eshetu, 2021).

The second criterion is based on machine learning types. Machine learning is classified into three types based on how the algorithms are trained, supervised, unsupervised, and reinforcement learning[16]. The supervised deep learning model was proposed for this study because we have a labeled dataset, and the target value is known. So that, it is possible to search for a suitable model that makes the most accurate classification to develop a maternal healthcare chatbot using a prepared dataset.

The third criterion is the most widely used and well-performed model in previous studies for processing NLP tasks. Various deep learning algorithms got success in NLP tasks, particularly for a conversational chatbot. These algorithms are DNN, RNNs-based models, CNN, etc. (Kumar & Ali, 2021)(Hussain et al., 2019)(Nuruzzaman & Hussain, 2018).

The last criterion is to make experimentation between the proposed deep learning models on the newly collected dataset to select the outperformed model. To our knowledge, one model isn't said to be better without experimenting with them. Once the best model is known, we can work with that model for further performance improvements.

3.8. Deep Learning Algorithms for the Chatbot Development

Deep learning models are utilized in a variety of NLP applications such as sentiment analysis, question answering, etc. (Liang et al., 2017). Advanced chatbot technology uses the concepts of understanding natural language and applying deep learning algorithms. A deep learning-based chatbot differs from pattern recognition methods, in which DL learns features from data.

[16] *https://www.potentiaco.com/what-is-machine-learning-definition-types-applications-and-examples/*

And we then select the RNNs-based deep learning models and some hybrid models RNN-based models with the CNN model for this work, which is explained in the next section.

3.8.1. Recurrent Neural Network (RNN)

RNN, as opposed to ANN, is formed of fully interconnected neurons. The recurrent neural network is an extension of the general feed-forward network that considers both the current input and past output while generating the output. It uses the concept of processing sequential data for tasks that involve sequential inputs. The models interpret the sentence as a sequence of words, which makes the algorithms better for capturing word dependencies. Recurrent neural network models have got successful in QA(Alshahrani & Kapetanios, 2016). The algorithm has two input values, the current input values, and values from the recent previous layer, as shown in Fig.3.4.

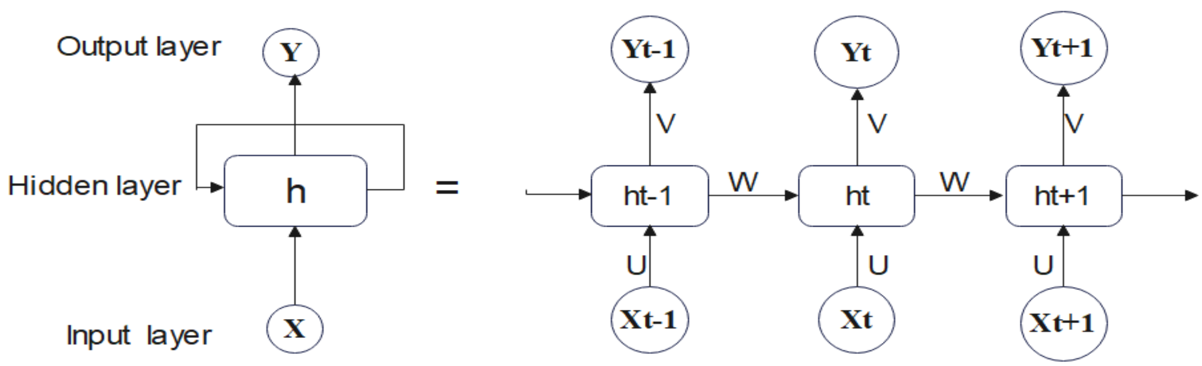


Fig.3.4 The Architecture of the RNN model

To compute the hidden and output layers at timestamp t, it uses the equation Eq.3.1:

$$ht = \delta(U * Xt + W * ht - 1 + bh)$$
$$yt = softmax(V * ht + by) \qquad Eq.3.1$$

Where *h(t) and* y_t *are* hidden layers and the output value time stamp t respectively, δ is the activation function, V is the weight vector of the output layer, The word vector's input value is X, the hidden layers weight vector is U, and the bias is b.

Despite the RNN being designed to work with sequential data, it has shown shortcomings in recalling input for long sequences. The models do not perform better and sometimes underperform the feed-forward neural networks for the sentence containing 20-40 words. Practically, an RNNs model presents limited achievement in long dependencies. Also, it has gradient exploding and vanishing problems. As a result, among various variants of RNNS, the LSTM is the best variant invented to handle the shortcoming of standard RNNs(Kandpal et al., 2020).

3.8.2. Long Short-Term Memory (LSTM)

LSTM is a more sophisticated recurrent neural network type with a better memory capacity. The model involves three gates; the input gate, forget gate, and the output gate. The sigmoid function generates these three gates. The forget gate Ft controls whether the previous timestamp information is flushed out or retained in the cell state. The value in the cell state is determined by the input gate layer whether it will be retained or updated. The forget gate Ft, input gate It, and output gate Ot layer values at hidden layer ht are computed by looking at ht-1 and Xt:

$$\begin{aligned} \mathrm{Ft} &= \delta(Xt * Uf + ht - 1 * Wf + bf) \\ \mathrm{It} &= \delta(Xt * Ui + ht - 1 * Wi + bi) \\ Ct &= \tanh(Xt * Uc + ht - 1 * Wc) \\ Nt &= Ft * Ct - 1 + It * Ct \end{aligned} \qquad \textit{Eq.3.2}$$

Where Xt is the current timestamps input, Uf is the input values weight, ht-1 is the previous timestamp of the hidden state, b is the bias, W is the hidden states weight matrix, Ct is a vector of new candidate values, and Nt is new information.

Finally, the output ot is calculated as - $Ot = \delta(Xt * Uo + ht - 1 * wo + bo)$, and then gets the final hidden state of timestamp t as - $Ht = Ot * tan\ h(Nt)$.

This shows how the three gates of the LSTM algorithm work together to produce h hidden state at time t. Fig.3.5, depicts the general architecture of the LSTM model.

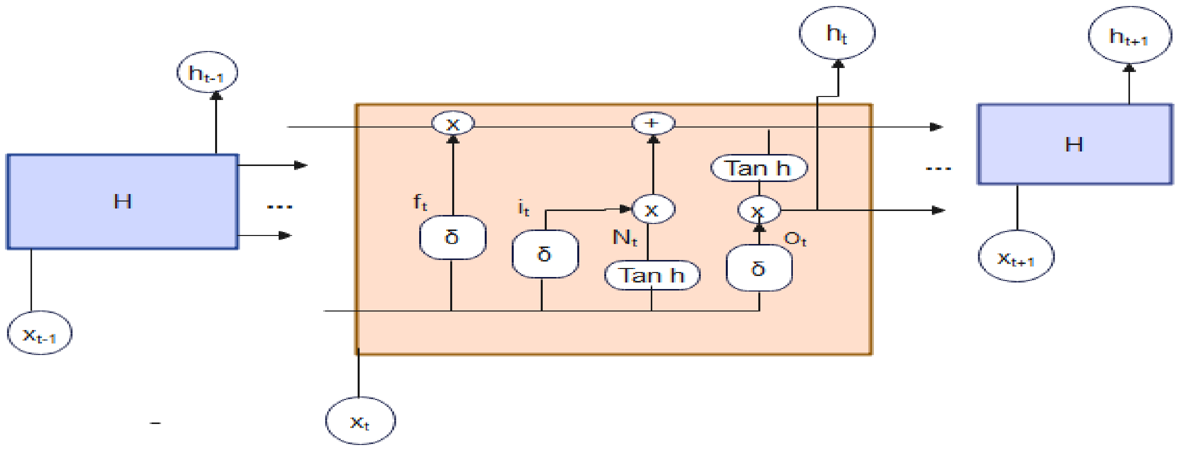


Fig.3.5 The LSTM Model's architecture

In this study, to develop Afaan Oromo text-based maternal health consultant chatbot model, the study proposed and assessed the advanced variants of recurrent neural networks such as BiLSTM, and CNN, and some hybrid models like CNN-LSTM and CNN-BiLSTM, instead of using the standard RNN and LSTM. The recurrent neural networks and even LSTM have no access to future information to learn sequence correlations. And then the network will be lost some extra information when processing sequential data. This can be improved using the BiLSTM model, which uses two LSTM models in opposite directions and is connected to one output.

3.8.3. BiLSTM (Bidirectional-LSTM)

To detail understand the structure of the BiLSTM model, first look at how the LSTM model works, which is explained in section 3.8.2. To improve the depletion of the standard RNN model and its extension LSTM, the BiLSTM model was introduced. The conventional RNNs and LSTMs do not have access to future information. They use merely the preceding output and current input in the current state. As illustrated in Fig.3.6, the connection goes from the input layer to the first hidden layer, then to the second hidden layer, and finally to the output layer. The notion of the BiLSTM model is taking two independent LSTM connected and sending a signal across their layer in opposite directions. This can helps the model to have both forward and backward information at each step(Wu et al., 2019). The last output layer is concatenated, and the final output Y is determined based on the Softmax activation function.

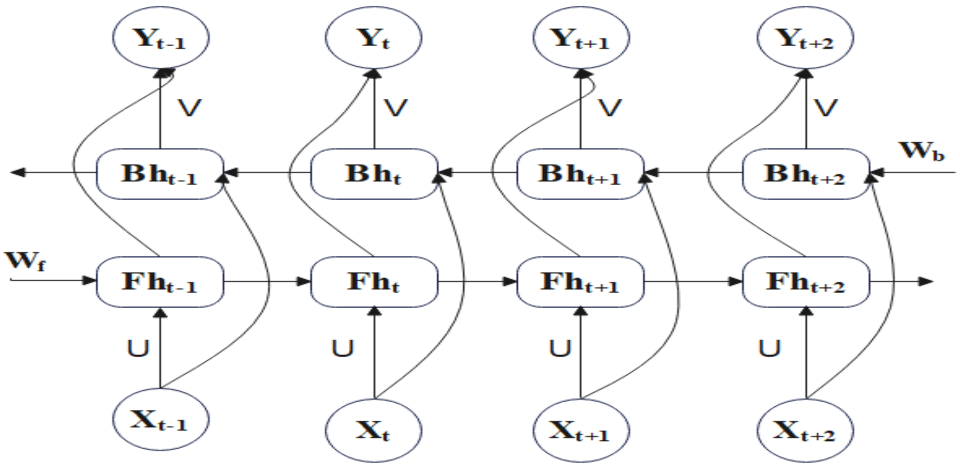


Fig.3.6 Structure of the BiLSTM Model

Fig.3.6. shows how BiLSTM it works and output Ot is computed using the equation Eq.3.3.

$$\begin{aligned} Fht &= F(WfFht-1+UfXt+bf) \\ Bht &= F(WbBht-1+UbXt+bb) \\ Ot &= VfFht+VbBht+bo \end{aligned} \qquad \textit{Eq.3.3}$$

Where W is weight to the hidden connection, V is hidden to an output connection, U is input to the hidden connection, F is the activation function, b is the bias, and Fht and Bht are the forward hidden unit and the backward hidden unit, respectively.

Even if the BiLSTM model can learn long sequences correlation by considering the past and future information in the current state, it cannot extract local features in parallel. For this reason, the study compares its performance with the proposed CNN-BiLSTM model which is explained in section 3.8.6.

3.8.4. Convolutional Neural Network (CNN) Model

A convolutional neural network is a deep learning model initially developed for image processing tasks and recently exhibits breakthrough results in NLP tasks, particularly in text classification and sentiment analysis(Yao et al., 2016). It's called ConvNets, and it requires numerical data to work with text; for this, the word embedding method is utilized. As shown in Fig.3.7, the word vector matrix, the one-dimension convolutional layer, MaxPooling, and the fully connected layer are required to work with text data when using the CNN model for text processing. MaxPooling carries out the network to hold only the maximum value in a feature vector which is the most useful and local feature[17]. The reason for a 1D convolutional layer is that the kernel is only moved in one dimension. When the CNN model process text data, it cannot learn sequence correlation. Consequently, as explained in section 3.8.5, the CNN model is combined with the LSTM model since the LSTM model may learn sequence correlation.

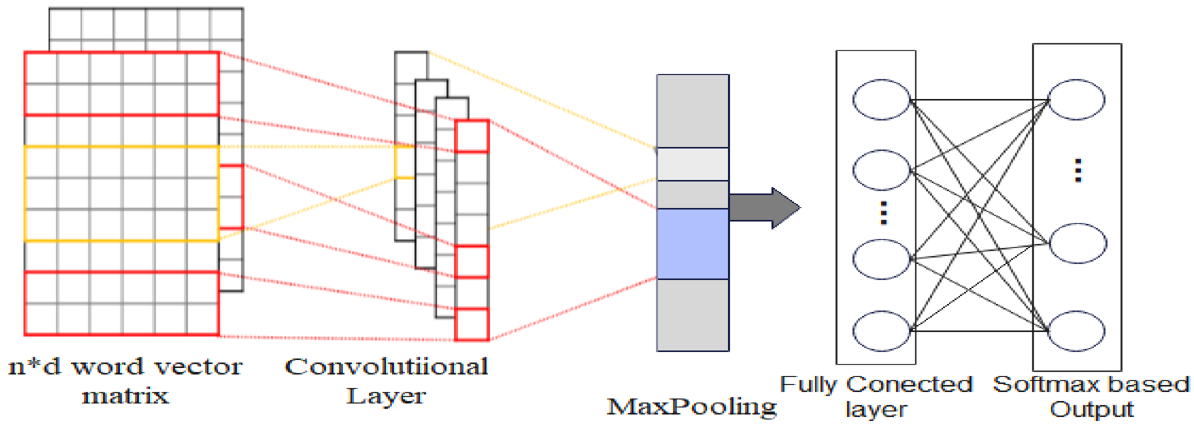


Fig.3.7 The Architecture of the CNN Model(Yao et al., 2016)

3.8.5. CNN-LSTM Model

As discussed in section 3.8.4, the CNN model is can process a text with a one-dimensional convolutional layer and word embedding. Even though the CNN model is better to extract local features from the word embedded data, it cannot effectively learn sequence correlations(Tam, 2021). To tackle such limitations of CNN, it is possible to use a hybrid method of CNN with LSTM, because the LSTM model can able to learn sequence correlations. We construct the convolutional layer on the top of the LSTM model and the data is passed through the layer of both algorithms. As a result, the local feature is extracted with the CNN model, and the fully connected LSTM model is used at the hidden layer to improve the capability of the model in local feature extraction and learning sequence correlation. The architecture of this model is also the same as Fig.3.7, but the difference is only that at the fully connected layer the model uses the LSTM layer and processes the information received from the convolutional layers.

[17] *https://cezannec.github.io/CNN_Text_Classification/*

3.8.6. CNN-BiLSTM Model

Instead of using single CNN and BiLSTM models separately for text processing, it is better to use the combined model of CNN with BiLSTM. Because, the CNN model cannot learn sequence correlations, and the BiLSTM model cannot extract local features in parallel. As a result, the proposed CNN-BiLSTM model is expected to resolve the drawback of a single model by constructing the convolutional layer on the top of the BiLSTM model. The general architecture of the proposed CNN-BiLSTM model is shown in Fig.3.8.

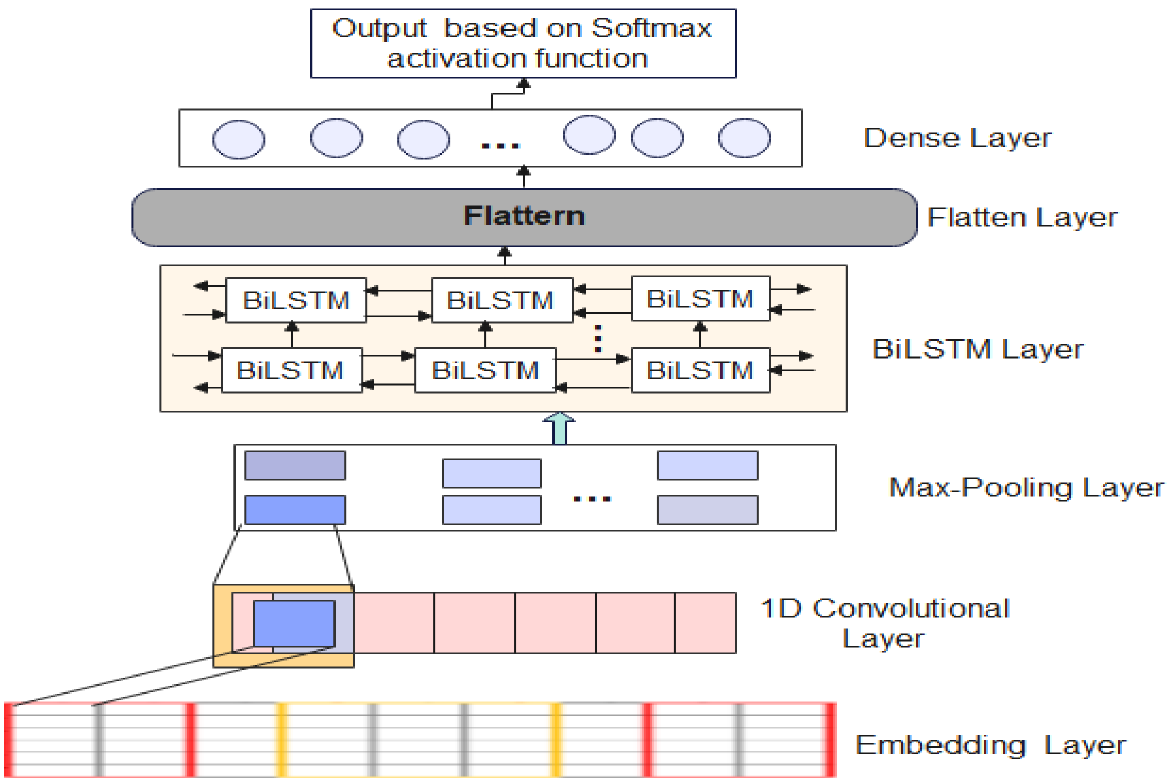


Fig.3.8 The CNN-BiLSTM Model Architecture for Maternal Health Consultant Chatbot

3.9. Model Overfit Handling Techniques

The most common problem in neural networks is model overfitting, which happens when the built model fails to generalize well for unknown data but is only well-optimized on the training set. In the other cases, model overfitting occurs when there is insufficient data or when the architecture becomes complex. There are numerous techniques for dealing with deep learning model overfitting. Cross-validation, the two most common regularization techniques[18], dropout, data augmentation, and early stopping are common in deep learning models to handle model overfitting[19]. In this study, the cross-validation method has been employed and the L2 regularization technique, since the proposed models are prone to overfitting. The reason why the L2 regularization technique is that it reduces training time, reduces loss of generalization error, and adds squared magnitude as a penalty to forcing the weights to a small value, but not zero(Kurtis, 2021).

3.10. Weight Initialization Techniques

While working on deep learning algorithms, it is vital to adjust the initial weights correctly to ensure the model with better accuracy. Before training the models on the training dataset, weight initialization assigns a small primary value to the neural network model parameters. Also, it may prevent the problem of RNN models vanishing or the explosion of the gradients during the backpropagation. There are many weight initialization techniques such as Zero Start, Randomly Start, Xavier (Glorot Start), etc.[20]. Among these techniques, we used Xavier (Glorot) weight initialization because it performs better in RNN-based models and Convolutional neural network-based models (Boulila et al., 2022).

3.11. Hyperparameter Tuning Method

When working with neural network-based models, the three layers of the models need to be tuned with the right hyper parameters. The variation in hyper-parameters in the same neural network forms the basis of distinction in terms of accuracy between models. Finding the best hyperparameter for the neural network model requires more attention to improve the performance of the model. Tuning the neural network with the correct hyperparameters the number of iterations (Epochs), learning rate, optimization algorithms, neurons in each layer, batch size, etc., is an arduous and experimental task. There are two hyperparameter tuning algorithms: grid search and randomized search, to find the optimal hyperparameter for the neural network(Nabi, 2019).

---

[18] *https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/*

[19] *https://www.v7labs.com/blog/overfitting*

[20] *https://www.geeksforgeeks.org/weight-initialization-techniques-for-deep-neural-networks/*

3.12. Model Evaluation Techniques

Model performance evaluation is a quantified representation of the learning process. This work evaluated the proposed model performance using several model evaluation techniques such as cross-validation and evaluation metrics like accuracy and loss. Selected the model with the highest accuracy on the training dataset does not enough to say the model is good, and the model must perform similarly in the unseen dataset. Thus, the following evaluation techniques are used to evaluate the performance of the proposed model.

3.12.1. Cross-Validation

Cross-validation is a method of measuring the model performance by separating the dataset into k equal partitions and used for training and testing iteratively. It is utilized to prevent model overfitting, in a case when the amount of data is limited and allows one to evaluate the model's generalization capability.

- ***K-fold Cross-Validation***

Using this evaluation technique, the dataset is randomly partitioned into K equal samples. And then, each fold is utilized for training and testing recursively; for example, 1 fold is employed for testing and K-1 folds are employed to train the model in the first iteration. As the K-fold cross-validation split the dataset randomly, there may be unbalanced folds that directly affect the performance of the model or lead the learning model training to be biased (Team, 2020). Because it shuffles the data into folds at random.

- ***Stratified K-fold Cross-Validation***

Stratified k-fold cross-validation is an extension of K-fold cross-validation, in which each fold has an equal number of samples from each class as the whole in a stratified manner. It places data from all classes across all folds. This technique is employed in this study, for it contributes to minimizing both bias and variance and as we do have an imbalanced dataset (Sharma, 2022). Thus, the stratified K-fold cross-validation ensures that each fold represents the complete data.

3.12.2. Evaluation Metrics

Accuracy: is a metric that commonly explains how the model performs through all classes. It is a vital metric in every industry to measure the performance of machine learning models. Describes what percentage of our test data is correctly classified. When all classes are equally significant, this metric is adequate. The model accuracy can be calculated as the number of correctly classified instances per the total number of classifications. It is computed using the mathematical formula shown in Eq.3.4.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad \textit{Eq.3.4}$$

Where TP is a true positive that is correctly predicted as actual, FP is a false positive that is a negative value predicted as positive, TN is a true negative that is correctly predicted as negative, and FN is a false negative that is a positive value predicted as negative.

Loss: - is an evaluation metric that is used to measure the training loss and validation loss of the model. The training loss measures how the model fits on the training dataset, whereas validation loss evaluates the model on the validation dataset[21]. Validation and training loss is computed from the summation of errors for each instance in the validation set and training set, respectively. This metric was used in this work to notify us whether the model requires additional tuning or not. Thus, we employ this metric to see how the model fits on the validation and training dataset, respectively by visualizing them on the graph.

3.12.3. Human Evaluation

The presented chatbot model should be evaluated to measure the performance and acceptance of the model. This can be done by developing the CUI for the model and give to the end-users. And based on the users' feedback the study evaluated the model accurateness on how much the model correctly answered a user question in the domain. Furthermore, the acceptance of the presented chatbot model is also assessed using human evaluation based on their satisfaction.

[21] *https://www.baeldung.com/cs/training-validation-loss-deep-learning*

3.13. Implementation Tools and Techniques

3.13.1. System Design Tools

- **E-draw Max**

E-draw max is a software diagramming tool that can help us to design any diagram in this study. This software is a very popular software to design UML diagrams, organizational charts, system design, workflow diagrams, etc. It appears to support over 280 different types of diagrams[22]. The software uses built-in editable symbols and templates based on the users' purpose. The study used E-draw max software to design different diagrams like research design, system architectural diagrams, etc.

3.13.2. Implementation Environments

- **Anaconda 4.10.3**

It is a Python and other programming language distribution environment for scientific computing such as machine learning, data science, data processing, etc. The primary goal of this environment is to simplify package and implementation. The anaconda distribution originated with 250 packages and 7,500 extra open-source packages[23]. It has different applications like Jupyter notebook, Spyder, JupyterLab, etc.

- **Jupyter Notebook 6.1.4**

Jupyter Notebook is a web-based interactive computational environment that allows you to create notebooks, code, and data. Jupyter notebook allows data scientists and machine learning developers to organize workflows.

---

[22] *https://www.g2.com/products/edraw-max/reviews*
[23] *https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)*

- **Colab**

Colab is a Google research product that allows anyone to write and run Python code, particularly machine learning, in the browser. It is a free Jupyter notebook environment that runs in the cloud to write and execute code in python language. One benefit of using Colab is it enables the different teams to edit the code simultaneously. Colab is a Jupyter notebook service that does not require any configuration and is free of resources such as the Tensor Processing Unit (TPU) and Graphical Processing Units (GPUs)[24]. We used it for our chatbot development because it supports deep learning libraries and provides free access to resources.

3.13.3. Programming Language Tools

- **Python 3.8.5**

Python is a high-level, object-oriented, well-structured, and readable programming language. The python programming language was created by Guido van Rossum from 1985- to 1990[25]. It is a general-purpose language designed to be used in various ranges of applications such as automation, data science, etc. We use the 3.8.5 version of the python programming language because of its compatibility with many packages. The language uses English keywords and has rarer syntactical constructions for this reason it is easy to use and understand by programmers.

3.13.4. Important libraries

- **NLTK**

NLTK is a widely known Python library for working with natural language. This library includes features such as tokenization, stemming, lemmatization, tagging, word count, etc(Bird et al., 2022). NLTK is built to support natural language processing and closely related areas like machine learning, AI, information retrieval, and linguistics. The NLTK library is used in this study to work and preprocess with Afaan Oromo datasets.

[24] *https://research.google.com/colaboratory/faq.html*
[25] *https://www.tutorialspoint.com/python/index.htm*

- **Keras 2.6.0**

Keras is a Python-based open-source high-level neural network API and runs on TensorFlow, CNTK, Theano, etc. Keras is designed to offer rapid experimentation with DL to allow for easy and fast prototyping and running seam on CPU and GPU(Choudhury, 2019). Unlike tensor flow, Keras only provides a high-level application programming interface, but tensor flow provides both high-level and low-level API.

- **TFlearn**

TFlearn is a transparent and flexible deep learning framework built on top tensor flow. Tensor flow is an end-to-end open-source deep learning framework created by Google in 2015[26]. TFlearn provides a higher-level application programming interface to tensor flow to speed up the implementation of deep neural networks(Kandpal et al., 2020). As explained in cited study TFlearn includes some features like optimizers, regularizes, and metrics. High-level APIs support the most recent deep learning models, such as CNN, BiLSTM, LSTM, etc.

- **Scikit-learn (Sklearn)**

It is a powerful open-source machine learning package for analyzing data in the Python programming language. The library has commonly been written in Python and works with NumPy, SciPy, and Matplotlib. NumPy is a Python library for manipulating large, multi-dimensional arrays and matrices.

3.13.5. Data Preparation Tools

- **Notepad ++**

Notepad ++ is a free source code editor that is used in this work to prepare an Afaan Oromo text dataset. It is a simple text editor to prepare datasets and save them with a JSON file extension. We prepared the dataset in the form of JSON format using notepad ++ code editor.

[26] *https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article*

3.13.6. Deployment Tools

Flask: after the proposed chatbot model is implemented and outperforming model is selected, it will deploy on a web server. Flask micro-framework is used in this study to deploy and use the designed chatbot model on the web. It does not need specific tools or libraries[27].

3.13.7. Hardware Tools

To install and work with the above tools, the personal computer that is well-found with an internal hard disk of 500GB, 6GB of physical memory, processor Intel(R) Celeron(R) CPU B830 @ 2.50GHz, Windows 10 pros, and system: 64-bit operating system, x64-based processor.

[27] *https://en.wikipedia.org/wiki/Flask_(web_framework)*

CHAPTER FOUR

4. DESIGN AND IMPLEMENTATION OF THE PROPOSED MODEL

4.1. Overview of the Chapter

The chapter describes the proposed model's design and implementation. This section will explain various points such as the general architecture of the designed model, proposed data preprocessing, proposed word embedding techniques, proposed deep learning model, implementation of data preprocessing, implementation of word embedding methods, and proposed model implementation.

4.2. The Architecture of the Proposed Model

The main objective of this study is to develop a chatbot model that provides a better understanding of pregnancy-related health problems. The dataset was collected and prepared from different sources to accomplish the objective of this study. For this study, it was collected manually from ORHB, health centers, individual midwives, hospitals, and online websites. The model does not make direct use of the collected dataset. It was filtered and prepared in a format suitable for the deep learning model, as explained in section 3.4. And then, the dataset is preprocessed using several NLTK libraries. Tokenization, stop word removal, text normalization, and the removal of unrelated content from the dataset are all text preprocessing techniques used in the study.

Once the dataset is clean up and preparation is complete, the next step is converting textual data into vectors. The reason is that the deep learning model does not work with textual data. Text data must be represented with vectors using word embedding methods which are discussed in section 3.6.

The feature vector data is ready for the models, and then we built the deep learning model with the correct hyperparameter and trained it with a training dataset. Building the model with the right network hyperparameter and training the model is the core task in this work. The grid search hyperparameter tuning algorithm, which is described in section 3.11, is used to determine the best hyperparameter for the proposed model. In this work, different Keras APIs models, such as CNN, CNN-LSTM, CNN-BiLSTM, and BiLSTM, have been proposed instead of using standard RNN, CNN, and LSTM. The RNN model fails to recall long sequences. And one variant of RNN known as the LSTM model improves these problems. The LSTM model, on the other hand, only considers the previous output and the current cell, not the future information. This may cause the network to lose extra information when learning sequence correlation. As a result, several models are presented to design the effective maternal health consultant chatbot model. Finally, the one that performs better with the highest performance has been saved and then used through CUI for the end-user, as depicted in Fig.4.1.

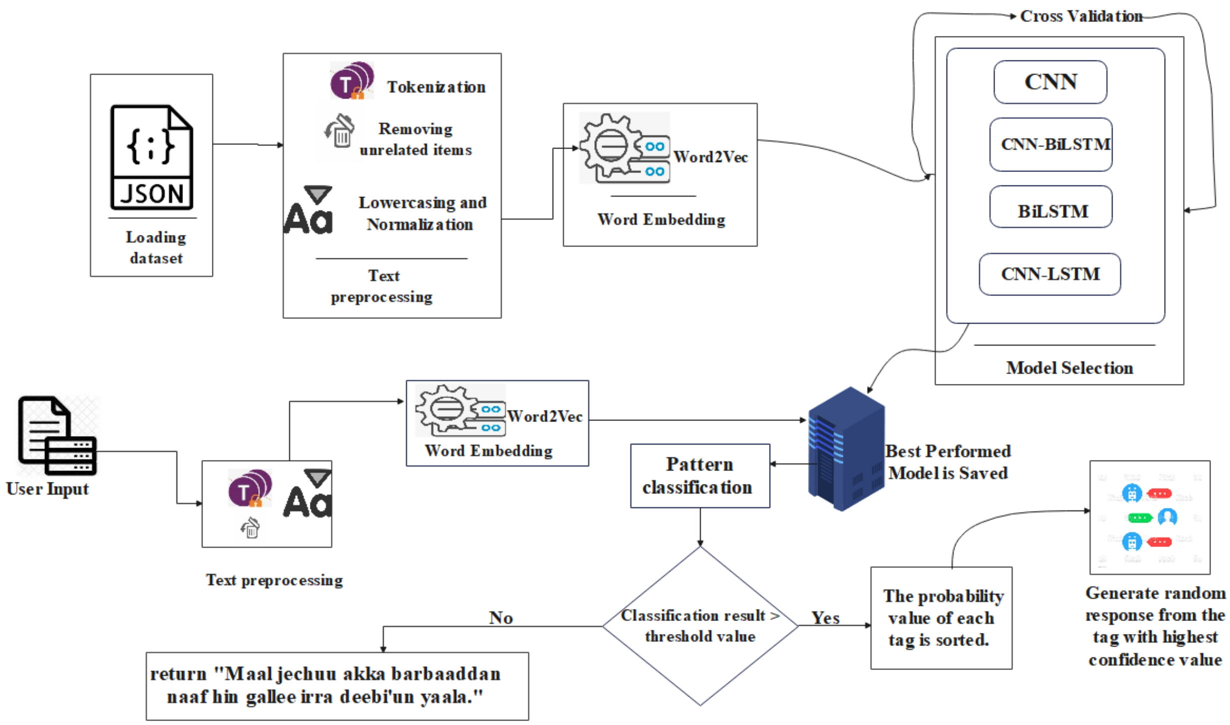


Fig.4.1. The Proposed Model Architecture for Maternal Health Consultant Chatbot

4.3. Proposed Data Preprocessing

When designing a deep learning model, do not always encounter clean, complete, and well-organized data from the real world. Therefore, it is mandatory to work with data and put it in an organized way according to study use cases. Various data preprocessing approaches were utilized in the study, including removing unrelated items and punctuation, eliminating stop words, text normalization, and tokenization.

4.3.1. Data Cleaning

The data cleaning is the method of removing unnecessary items, punctuation, white spaces, and extra symbols from the dataset to train the model effectively.

```
INPUT: Noise dataset
OUTPUT: cleaned dataset
INIT: Read the dataset
WHILE(it is not the end of the file):
        IF data contains special letters, emojs, puctuation and extra symbols ['@','+','#', '&',
'*','$','%','!',',','.','?',’|','a','aa','A','ka','b','B',’c','C','d','D','E','e','F','e','G','g','H','h','ph','PH','TS','ts
', 'I','i','j','J','K','k','L','l','M','m','n','N','o','O','P','p', 'ch','CH','NY','ny',
      'q','Q','r','R','s','S','t','T','I','U','u','v','W','V','w','y','Y','Z','z','J',
      '😍','👍',  '😍','🤲','🤜','😁','🧡','✋','🤣','1234567890','-','/','(',  ')'] THEN remove
characters and symbols, emojis, punctuation
        IF data contains [ tabs, extra white space] THEN remove
        Return cleaned data
        ENDIF
HALT
```

4.3.2. Stop Word removal

They are the most prevalent terms in any language like pronouns, prepositions, conjunctions, articles, and particles and they must be detached from the dataset. The Afaan Oromo stop word was collected, checked, and saved into NLTK stop word corpora with the “AOfStopwords” file name, and then loaded and removed from the dataset.

```
        INPUT: dataset
        OUTPUT:  dataset free from stop-words
        INIT: Read the dataset
        WHILE (dataset):
            IF dataset not in AOfStopwords:
            Return dataset
        ENDIF
HALT
```

4.3.3. Text Normalization

Text normalization helps the models to recognize the same word written in different ways. Users may write the same word differently, and the machine learning model also understands those words differently. Here is how text is normalized into unique words.

```
INPUT: unnormalized text
OUTPUT: clean text
INIT: Read the text
WHILE (it is not the end of file):
     IF text contains [DhugAA] THEN normalize into “dhugaa”
     IF text contains [dhakaa] THEN normalize into “dhagaa”
     IF text contains [otoo] THEN normalize into “osoo”
     Return normalized text
     ENDIF
HALT
```

4.3.4. Tokenization

Once the dataset is cleaned, the next step is tokenizing the long sentence into a small unit of words or tokens. Since the deep learning model does not work directly with text data, the text data T is tokenized into {t1, t2,….tn} and then represent with vectors using embedding methods to be the input for the model.

4.4. Proposed Word Embedding

After the dataset is cleaned and ready for the model, the next step will be converting the textual data into vectors. In this paper, the Word2Vec technique is employed because the model holds the semantics of the words in the sentence, and the model scales well and gives good results for both larger and smaller data(Sivakumar et al., 2020). The word2vec technique is trained with the data and represents them as vectors, which are later processed by the deep learning model. The model is first trained and saved in the file and then loaded and used to represent the text data with vectors. As shown in Fig.4.2, this type of word embedding is placed the corresponding word into a nearby vector. The output of this step will be fed into the deep learning model layer as an input.

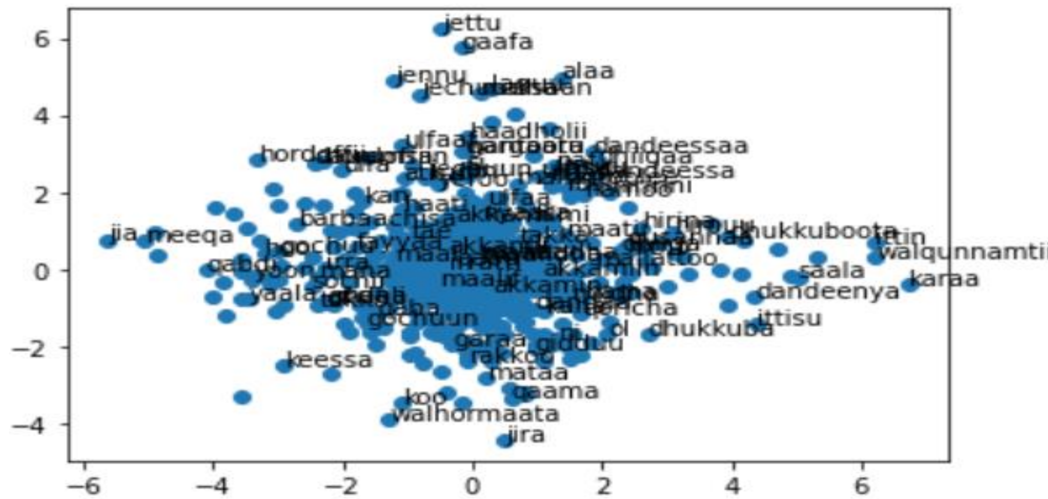


Fig.4.2 Word2Vec Word Representation

In general, the collected dataset passed through different steps to be ready for the proposed model. The dataset was collected and prepared in a format that meets the research goal, and various data preprocessing techniques are applied to clean the data. And then, the word representation techniques are used to represent text data with vectors. The word2vec model is used in the study to represent text data with vectors. Fig.4.3, illustrates the data preparation and preprocessing procedures for maternal health consultant chatbot development.

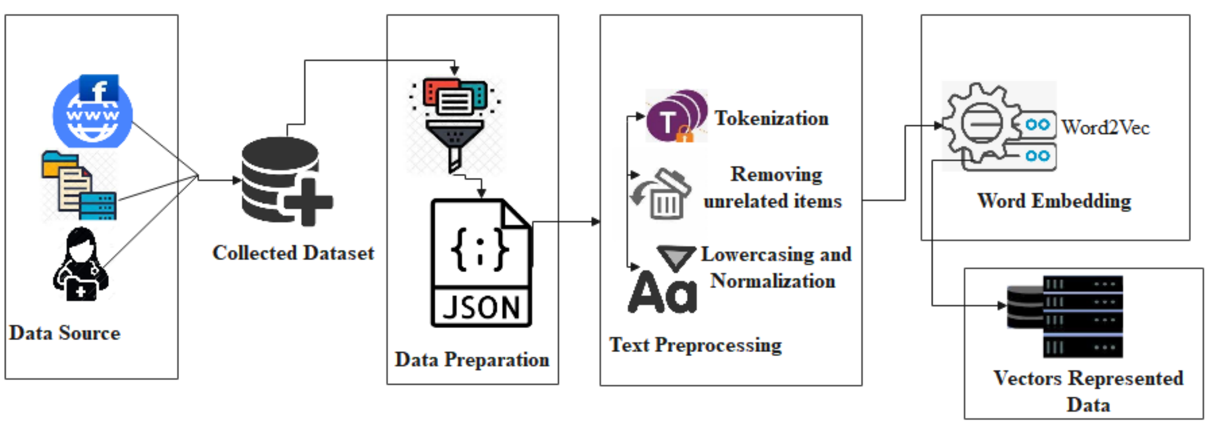


Fig.4.3 Proposed Data Preparation, Preprocessing, and Word Embedding Techniques

4.5. Proposed Deep Learning Model

To propose the deep learning model for this investigation, the study set different criteria, which are detailed in section 3.7. Based on those criteria, instead of using the standard RNN and LSTM, one variant of recurrent neural networks called BiLSTM and the CNN model and some hybrid models like CNN-LSTM and CNN-BiLSTM are proposed. The recurrent neural networks and even LSTM have no access to future information to learn sequence correlations. And then, the model will lose some extra information when processing sequential data. It has also proven that the RNN model has gradient exploding and vanishing problems. To overcome these drawbacks, the study proposed four deep learning models which are described in section 3.8, with their explicit capability and limitations in NLP tasks. The study experimented with the model using a newly collected dataset and evaluated them under stratified cross-validation to identify the most suitable model for designing a maternal health consultant chatbot, as shown in Fig.4.4.

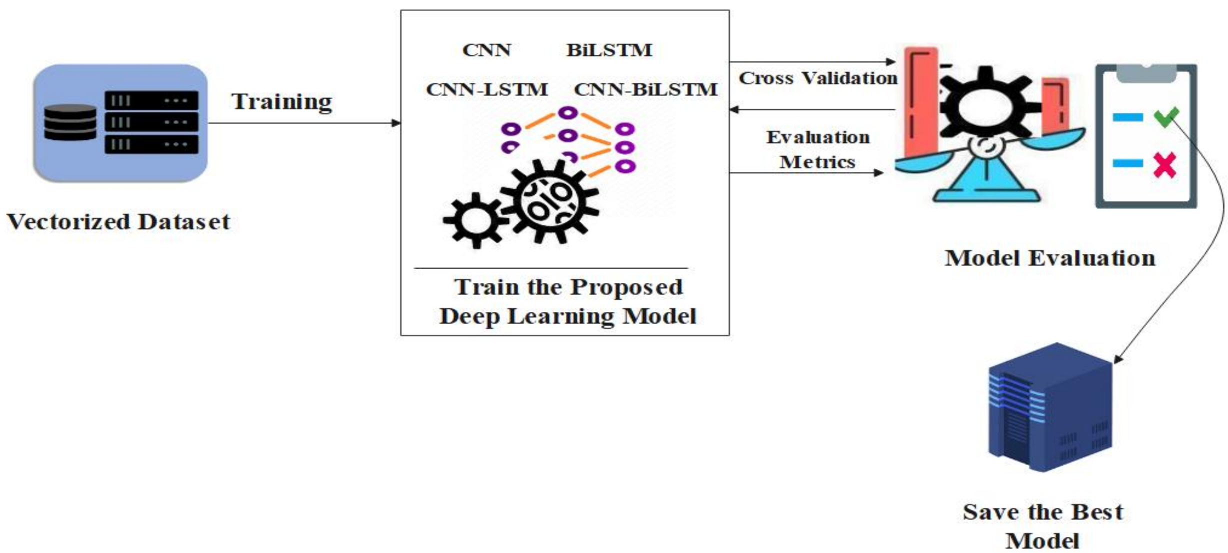


Fig.4.4 Proposed Deep Learning Modeling

4.6. Proposed Evaluation Techniques

When designing a machine learning model, it is necessary to check the generalization capability of the models. The model that well fits the training dataset with the highest accuracy does not have enough to say the model is good, and the model must perform similarly in the unseen dataset. There are several model evaluation techniques such as cross-validation and evaluation metrics, as explained in section 3.12. This study employs Stratified K-fold cross-validation, for it contributes to minimizing both bias and variance(Sharma, 2022). There is no formal rule for determining the K value; in most cases, the value is 5 or 10(Kant, 2021). A large k value is not recommended because it requires high running time and computing costs.

4.7. Implementation of Proposed Data preprocessing

4.7.1. Loading Dataset

Dataset is loaded into the python environment before any data is processed. To load the dataset into the python environment, the json.loads () method is used and parse the valid JSON file and transform it into a python dictionary. The json format dataset is loaded, as shown in the following sample code.

```
data = open (‘Path/dataset_name.json’, encoding= utf-8).read()
dataset = json.loads(data)
```

4.7.2. Text Normalization and Stop words Removal

After the patterns in the dataset are tokenized and cleaned, text normalization is another preprocesses technique to normalize similar words written differently, as shown in Appendix:1, 1.1. For instance, the word ‘Haga’ is normalized into ‘Hanga’, etc. And at this step, the Afaan Oromo stop words are also eliminated. The following is code for eliminating stop words from the dataset. As shown in the following sample code, the Afaan Oromo stop word was collected and saved with the “AfanOromoStpWords” file name into stopword corpora, and then imported and removed from the dataset.

```
from nltk.corpus import stopwords
AOstop_words = stopwords.words (AfanOromoStpWords)
pattern_tokens = [word for word in pattern_tokens if word not in AOstop_words]
```

4.7.3. Implementation of Data Cleaning

Various data cleaning process such as punctuation, numbers, letters, removing white space, etc., has been done on the dataset, as shown in the following sample code. Also, the study applies a typical example of text normalization which is called lowercasing.

```
#Here, different text preprocess is used to remove different symbols, punctuation, etc
punctuation = dict.fromkeys(p for p in range(sys.maxunicode)
                      if unicodedata.category(chr(p)).startswith('P'))
pattern_tokens = [string.translate (punctuation) for string in pattern_tokens] #remove any
puctuation mark
pattern_tokens = [string.strip () for string in pattern_tokens]#white space removal
remove = (['@','+','#', '&', '*','$','%','!',',','.','?',\
        '|', 'a', 'aa','A', 'b', 'B', 'c','C','d','D','E','e','F','e','G','g','H','h','ph','PH','TS','ts' \
        'I','i','j','J','K','k','L','l','M','m','n','N','o','O','P','p', 'ch','CH','NY','ny'\
        'q','Q','r','R','s','S','t','T','I','U','u','v','W','V','w','y','Y','Z','z','J'\
        '😍','👍', '😍','🤲','🤜','😁','🧡','🖐','🤣','1234567890','-','/','(', ')'])
pattern_tokens = [w.lower () for w in pattern_tokens if w not in remove]#to lowercasing and
to remove unrelated items
```

4.7.4. Implementation of Tokenization

After the dataset is cleaned, normalized, and the stop word is removed, the next steps will be tokenizing the patterns into small pieces called tokens, for further processing of the patterns data. Here's a sample example code of text tokenization.

```
patt=[]
for intent in dataset['intents']:
for pattern in intent['patterns']:
     w1 =  pattern.split(" ")
     w=patterns.append(pattern)
     patt.extend(w1)
     container.append((w1, intent['tag'
     if intent['tag'] not in classes:
          classes.append(intent['tag'
```

4.8. Implementation of Proposed Word Embedding

The following sample code indicates loading the trained word2vec model and preparing the data for the embedding layer of deep learning models. The word2vec was trained on the dataset with 300 embedding dimensions since each word was represented with a vector dimension. Then the model is saved and loaded whenever required.

```
from gensim.models import Word2Vec
W2V=Word2Vec(list_array,min_count=1,workers=4,size=emb_dim,sg=1, window=5)
W2V1=W2V.wv.save_word2vec_format('/Path/Word2vecModel.bin', binary=True)
PretrainedModel = gensim.models.Word2Vec.load('Path/Word2vecModel1.bin')
voc = list(PretrainedModel.wv.vocab)
voc_size=len(voc)
word_vec_dict={}
for word in voc:
  word_vec_dict[word]=PretrainedModel.wv.get_vector(word)
embed_matrix=np.zeros(shape=(voc_size,embed_dim))
for Word,i in tokenizer.word_index.items():
    embed_vector=word_vec_dict.get(Word)
    if embed_vector is not None:
      embed_matrix[i]=embed_vector
w2v_embedding=Embedding(input_dim=voc_size,output_dim=embed_dim,input_length=max_l
ength,embeddings_initializer=Constant(embed_matrix))
```

4.9. Proposed Model Implementation

After the dataset is preprocessed and represented as a vector, the next step is building the deep learning algorithms and training the model with the training dataset. As discussed in section 3.8, this study proposed various Keras APIs, including CNN, BiLSTM, CNN-LSTM, and CNN-BiLSTM. The source code for all of the proposed models is available in Appendix: 1.

4.9.1. CNN Model Implementation

The text data was represented in n*d dimensional vectors using the word2vec word embedding method. And we then used ‘n’ convolutional layers (Conv1D) with 128 filters, kernel size of 7, and relu activation function. A max-pooling layer (MaxPooling1D) with pool size 5 is added after the convolutional layer and converts each kernel size input into one output and condenses the dimensions of the output by retaining the most relevant information.

```
modelcn.add(w2v_embedding)
modelcn.add(Conv1D(filters=128, kernel_size=7, activation='relu'))
modelcn.add(MaxPooling1D(pool_size = 5))
modelcn.add(Dense(300, activation='relu'))
```

The number of filters, kernel size, and pool size are determined by grid search hyperparameter tuning for this work. The information is passed to a dense (fully connected) layer of 'n' units with a relu activation function. Lastly, a dense layer of 103 units with softmax activation determines the final outputs. Softmax activation is used here at the output layer since the problem at hand is multi-class classification. The activation function is applied to buffer the data before being sent to the succeeding layer. The general implementation procedure of the CNN model is given in Fig.4.5, and Appendix: 1, 1.2 shows the sample code for CNN models.

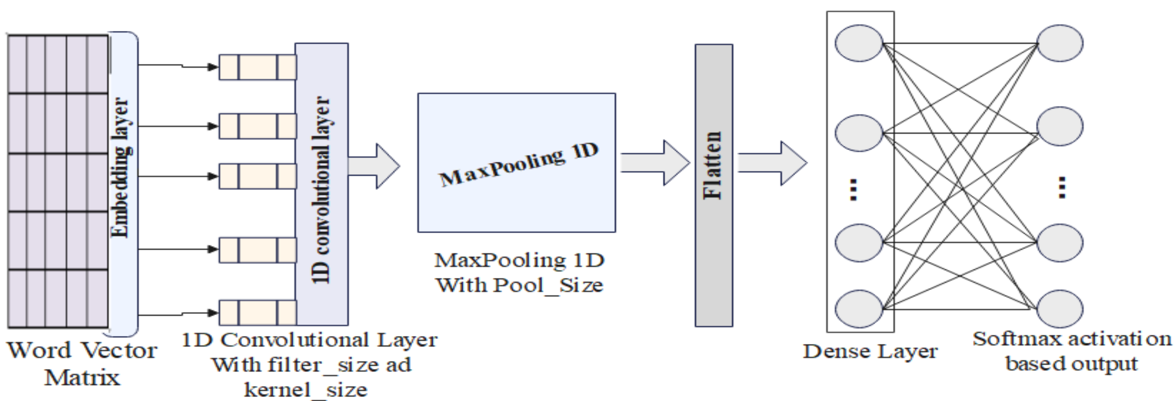


Fig.4.5 Implementation Procedures of the CNN Model

Embedding layer: is one of Keras layers mainly used in NLP tasks that involve neural networks, as shown in Fig.4.5. Voc_size indicates the total size of vocabulary, embedding_dim is a dimension of the word represented by the word2vec model, and input_length is equal to the maximum length sentence in the dataset.

Flatten: this layer concatenates the previous layer's output before passing it on to the dense layer. The output layer's confidence value is generated using the softmax activation function for each tag.

The dense layer: is a neural network layer that integrates data from all neuron layers preceding it. The neuron dense layer is adjusted to have an equal number of output target tags (classes).

Since the problem at hand involves multi-class classification, the softmax activation function is used to generate the confidence value for each tag.

4.9.2. CNN-LSTM Model Implementation

The CNN-LSTM model is similar to the CNN-based model, but the standard CNN uses a multi-layer neural network, and cannot learn sequence correlations. Instead of a multi-layer neural network, the LSTM layer was added to the network to learn sequence correlations. The Convolutional layer was utilized at the input layer, whereas the LSTM layer was employed at the fully linked layer, as illustrated in Fig.4.6.

```
model.add(Conv1D(filters=128,kernel_size=7,activation='relu'))
model.add(MaxPooling1D(pool_size = 5))
model.add(LSTM(500,return_sequences=True,activation='relu'))
```

The overall implementation procedures of the CNN-LSTM model are shown in Fig.4.6, and Appendix: 1, 1.3 shows the sample code for CNN-LSTM models.

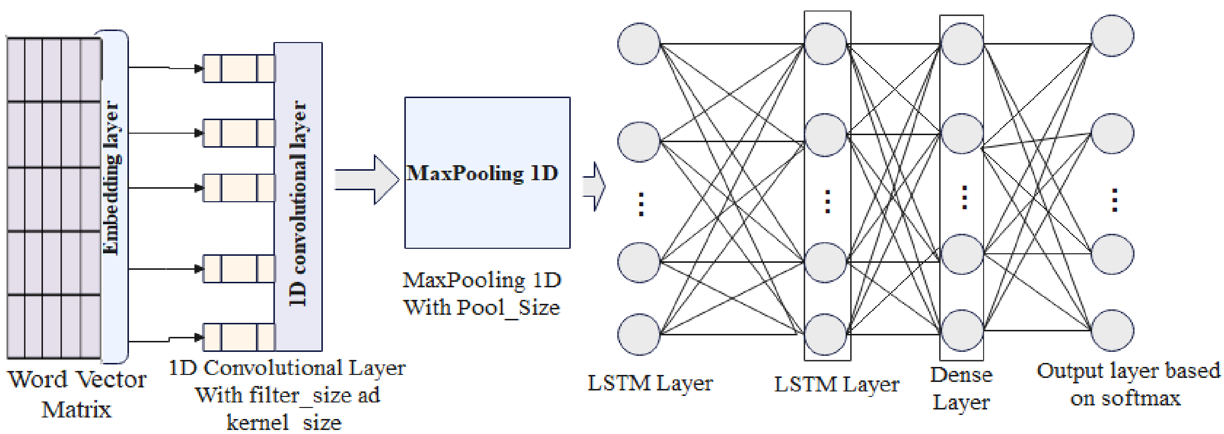


Fig.4.6 Implementation Procedures of the CNN-LSTM Model

4.9.3. BiLSTM Model Implementation

The BiLSTM Model connects two LSTM in the reverse direction to the one output. It allows the model to receive information from past and future situations, as depicted in chapter Fig.3.8.3. It has two LSTM hidden layers for both the previous input and future input in opposite directions. In this way, the model can understand the context, and improve the unidirectional LSTM by preserving information from both past and future cells.

```
modelbirnn.add(w2v_embedding)
modelbirnn.add(tf.Keras.layers.Bidirectional(LSTM(500,return_se
quences=True, activation ='relu')))
```

The flow of an implementation procedure of the BiLSTM model is shown in Fig.4.7, and Appendix: 1, 1.3 shows the sample code of BiLSTM models.

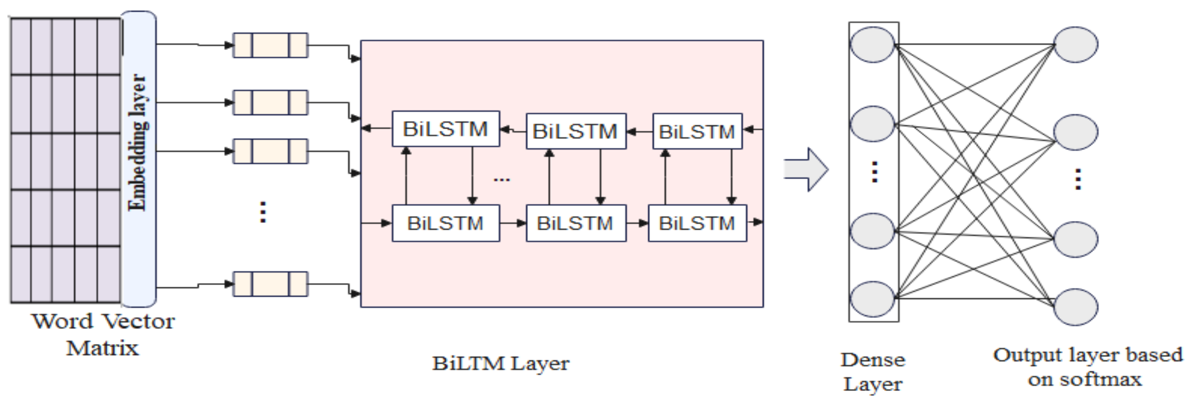


Fig.4.7 Implementation Procedure of BiLSTM Model

4.9.4. CNN-BiLSTM Model Implementation

Instead of using single CNN and BiLSTM models separately for text processing, it is very effective to use a combined model of CNN-BiLSTM. Because the CNN deep learning model cannot effectively learn sequence correlations or it does not capture the relationship between words. Similarly, the BiLSTM model cannot extract local features in parallel. However, it is expected that the proposed hybrid model will be able to resolve the shortcoming of the single model using the convolutional layer with BiLSTM. 1D convolutional layer is added, after embedding layer to the network with 128 filters, and kernel size 7 on the top of BiLSTM layers, as shown in the following sample code.

```
model.add(Conv1D(128,kernel_size=7,activation='relu')
model.add(MaxPooling1D(pool_size=5))
model.add(Bidirectional(LSTM(500,return_sequences=True,act
ivation='relu’))
```

Fig.4.8, shows how the CNN-LSTM model implementation procedures, and Appendix: 1, 1.3 shows the sample code for CNN-BiLSTM models.

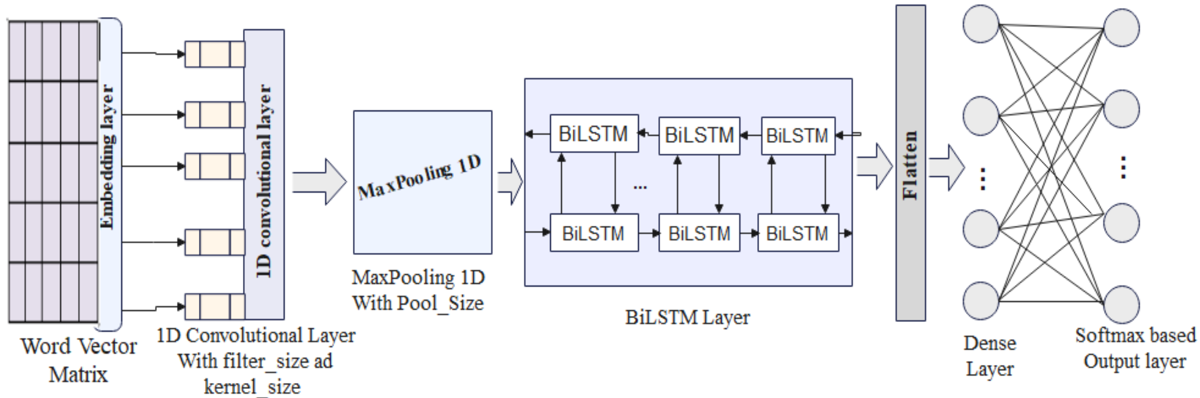


Fig.4.8 Implementation Procedure of CNN-BiLSTM Model

4.9.5. Setting Threshold Value

The chatbots may not answer any questions users ask. In this study, the closed domain chatbot model is designed only for a maternal health consultancy. So, there may be inappropriate responses if the user asks the questions out of the domain. Hence, the threshold value is set to a fixed probability value. If the probability value surpasses the threshold value, the response that belongs to the tag with the highest probability will be returned, if not the chatbot will return "Maal jechuu Akka barbaaddan naaf hin gallee irra deebi'uun yaala" which means "I'm sorry I didn't get it, please try again."

4.10. Saving the Model for Future Use

Once, the best model is known, no need to train the model again for future use. It is possible to save the model and make it as it responds to the users' questions quickly. To save python objects into a file pickle is used. Pickle is a Python function that converts a Python object into a byte stream that can be saved to a file. It allows us to serialize and deserialize a Python object type[28]. The following sample code shows how to save python objects like lists, dictionaries, etc., into the file and load them for future use. The sample code for saving and loading the model is available in Appendix: 1, 1.5.

[28] *https://www.synopsys.com/blogs/software-security/python-pickling/*

CHAPTER FIVE

5. RESULT AND DISCUSSIONS

5.1. Overview of the Chapter

In this chapter, the results of different proposed deep learning models for maternal health consultant chatbot is explained based on an experiment performed. In this section, based on the newly collected dataset, the study has summarized the results of various proposed deep learning models. The study also discusses the comparison between the presented study and other studies conducted on Afaan Oromo text-based chatbot model for various industries. Lastly, the chapter also focuses on the discussion of how the presented study answers the research questions and the designed model prototype.

The dataset contains 1962 instances; these instances are called patterns in which user intent is included. They are labeled with their appropriate tags; there are 103 tags in our dataset which is the number of classes. Each tag contains patterns and responses; the model classifies the pattern into the appropriate tag, to answer the response that belongs to the same class with patterns. Using this described dataset, the results of all proposed models are summarized as follows:

5.2. Result of the Proposed Model

One variant of RNNs-based models called BiLSTM and the CNN model also some hybrid models like CNN-LSTM and CNN-BiLSTM has been presented and evaluated to design a maternal health consultant chatbot in this study. In section 3.8, all proposed deep learning models are explained with a clear justification of their limitations and capability to work on NLP tasks. This paper summarized the results of each proposed deep learning model as follows:

5.2.1. Results of CNN model

The CNN model was built with the optimal network hyper parameters and then trained with the prepared dataset. With this model, the study achieved 95.21% under Stratified 5-fold cross-validation. And also, as shown in the following curve, the training accuracy and the validation accuracy are well fitted with the newly prepared dataset. Fig.5.1, exhibits the model's training, validation, and loss curves.

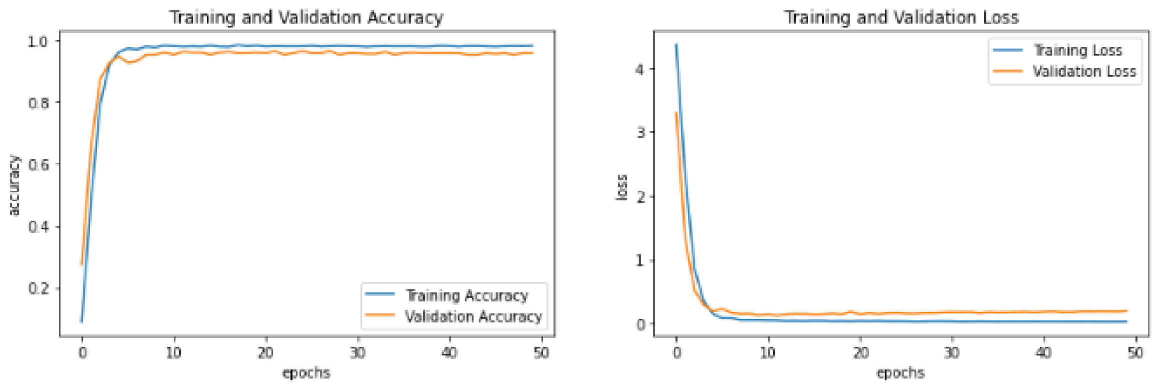


Fig.5.1 CNN Model Accuracy and Loss through Epochs

5.2.2. Results of CNN-LSTM Model

It is the combination of the CNN model with LSTM; this study has used a convolutional layer at the input layer of the model, following that, and incorporated the LSTM layer into the network. Word2vec model has been used as word embedding and then construct the convolutional layer on the top of LSMT and achieved 95.72% accuracy under Stratified 5-fold cross-validation. Fig.5.2 shows the model's training, validation, and loss function curves. As shown in the following curve this model fitted well with the prepared dataset.

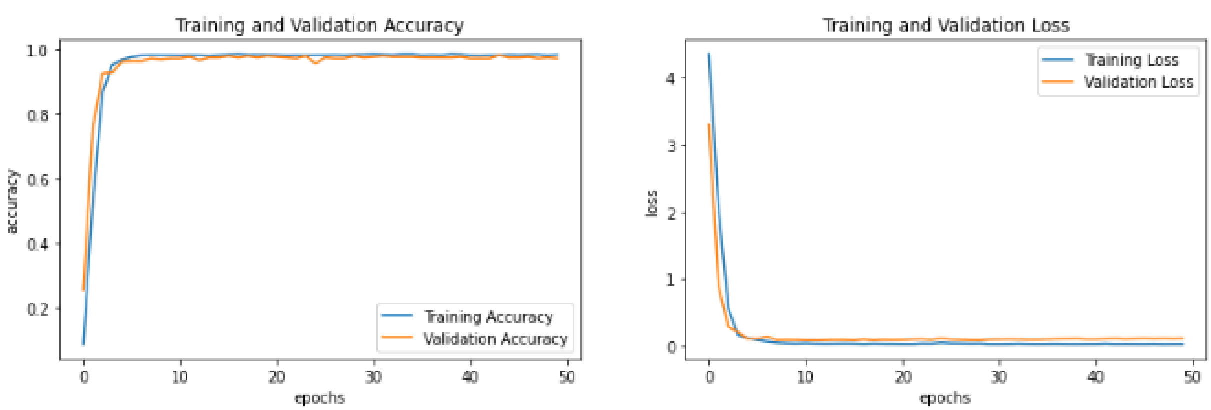


Fig.5.2 Accuracy and Loss of the CNN-LSTM Model Across Epochs

5.2.3. Results of BiLSTM Models

Instead of using standard unidirectional LSTM, it is better to use Keras API called BiLSTM which can utilize the information from both forward and backward directions. The output value is computed using both the forward and backward LSTM layers. Using this proposed BiLSTM model, the study achieved 95.11% accuracy under stratified 5-fold cross-validation. Fig.5.3, illustrates the model's training, validation, and loss curves.

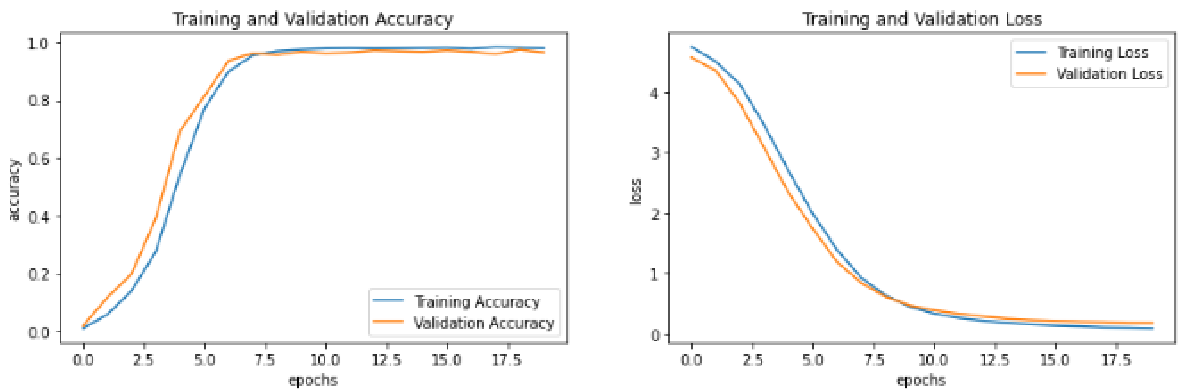


Fig.5.3 BiLSTM Model Accuracy and Loss through Epochs

5.2.4. Results of CNN-BiLSTM Models

The BiLSTM model can capture the long dependencies of the sequence and extends the LSTM model by using information from past and future cells. The hybrid proposed CNN-BiLSTM model can improve the limitation of both models at once such as local feature extraction and learn sequence correlations. The study constructs a convolutional layer on top of bi-directional LSTM and achieved the best result of 96.94% accuracy under stratified 5-fold cross-validation. Fig.5.4 depicts the model's training, validation, and loss curves.

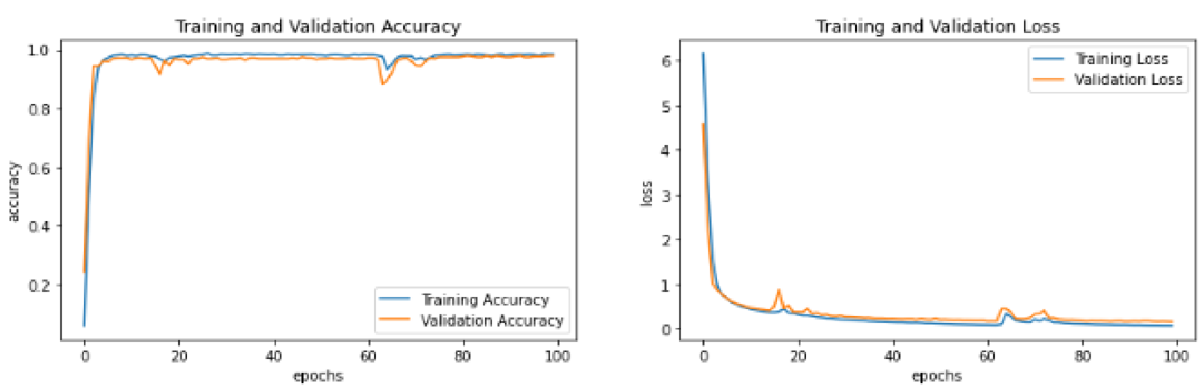


Fig.5.4 CNN-BiLSTM Model Accuracy and Loss through Epochs

Table 5.1 Summary of the Proposed Model for Maternal Health Consultant Chatbot

	CNN		CNN-LSTM		BiLSTM		CNN-BiLSTM	
Stratified	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
5-Fold CV	**95.21%**	0.1474	**95.72%**	0.1543	**95.11%**	0.1916	**96.94%**	0.1182
10-Fold CV	95.16%	0.2246	94.55%	0.2122	95.01%	0.1181	95.92%	0.1477

5.2.5. The Two Types of Word2vec Model

As explained in section 3.6, the word2vec model has two approaches, such as the Skip-gram approach and the Continuous bag of words. The Skip-gram approach predicts its neighbors based on the current words, whereas CBOW predicts the target word based on its neighbors. Table 5.2, exhibits the comparison of these two approaches when used in proposed deep learning models.

Table 5.2 The comparison of Two Types of Word2vec Model

	CNN		CNN-LSTM		BiLSTM		CNN-BiLSTM	
W2V Types	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
Skip Gram	**95.21%**	0.1358	**95.72%**	0.1569	**95.11%**	0.1570	**96.94%**	0.1783
CBOW	94.42%	0.1474	95.02%	0.1543	95.01%	0.1181	96.03%	0.1182

As shown in Table 5.2, the Skip-gram model outperforms the CBOW within all proposed deep learning models. In addition to the performance, we recommend the skip-gram model, for it is better to capture the semantic relationship between words, but the CBOW model considers only morphologically related words placed in nearby space.

5.2.6. Activation Function and Optimization Algorithms

The variation in activation functions has an impact on the model's performance. Relu and Sigmoid activation function is commonly used in neural networks and got success in many previous studies(Leonid, 2020)(Gupta, 2020)(Vamsi et al., 2020). Table 5.3, shows the performance of two activation functions when used in the proposed deep learning model. And we also compare the two optimization algorithms such as Adam and Stochastic gradient descent (SGD), in terms of the performance they have with the proposed deep learning model. Optimizers are algorithms that alter the weights and learning rates of hidden layer neurons to reduce network losses. Most of the time Adam optimizers achieve the best result, nonetheless sometimes SGD outperforms Adam's optimization algorithm(Vamsi et al., 2020). A comparison of the two methods has been carried out in this study, as shown in Table 5.3.

Table 5.3 The analysis of the Two Optimizers and Activation Functions

	CNN		**CNN-LSTM**		**BiLSTM**		**CNN-BiLSTM**	
Optimization Algorithms								
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
Adam	**95.21%**	0.1474	**95.72%**	0.1543	**95.11%**	0.1181	**96.94%**	0.1504
SGD	94.70%	0.2035	93.22%	0.2078	95.03%	0.1763	95.21%	0.1182
Activation Function								
Relu	**95.21%**	0.1474	**95.72%**	0.1543	**95.11%**	0.1181	**96.94%**	0.1504
Sigmoid	93.33%	0.2260	92.36%	0.247	95.06%	0.2566	95.59%	0.1182

The result of all proposed models is shown in Tables 5.1, 5.2, and 5.3, under stratified 5-fold cross-validation, the CNN-BiLSTM model outperformed all other proposed models for this study. As a Convolutional neural network requires the word embedding layer when it works with text data since the study utilized Skip gram word2vec word embedding and achieved 96.94% accuracy. This study applied L2 regularization to address the problem of model overfitting. The below chart showed the performance of all proposed deep learning models for designing a maternal health consultant chatbot.

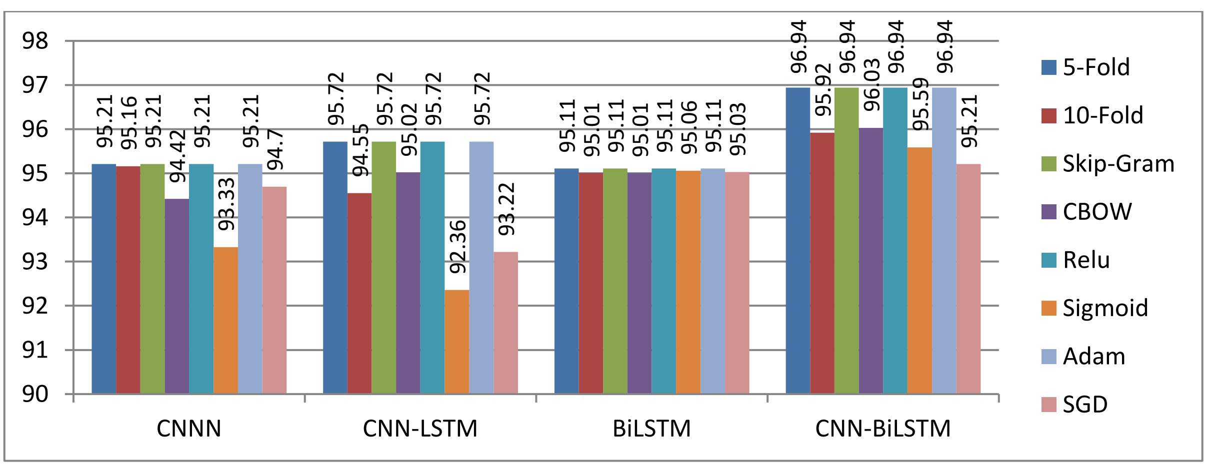


Fig.5.5 The proposed DL Model comparison Chart

5.3. Hyperparameter Tuning

This study searches for the optimal hyperparameter for the proposed neural network models using the grid search hyperparameter tuning method. Since it produces accurate results with small datasets, but with large datasets and high dimensions, it will greatly slow down computation and be very costly. Instead, a random search is preferable to use. With a small dataset size, a grid search will produce more accurate results than a randomized search, but with a big dataset, it will down computation time[29]. Thus, the grid search hyperparameter tuning method is searching the combination of possible optimal hyper parameters. The study has found the optimal hyperparameter for all proposed models, from Epochs(20,100,50), Batch size(32, 100, 64, 128), number of neurons(500, 256, 200, 300) Activation function(relu, sigmoid), Optimizers(Adam, Stochastic Gradient Descent), Learning rate(0.1, 0.001, 0.0001), Filters(128, 64, 256), Kernel size(5, 7, 9), Hidden Layers(1, 2, 3). The study employs Grid search hyperparameter tuning to find the best combination of hyperparameters. The optimal hyper parameters for all of the proposed models were then determined, as shown in Table 5.4.

Table 5.4 Hyperparameter for all Proposed Models

Models	***Epochs***	***Batch size***	***Activation***	***Optimizers***	***Learning rate***	***Filters***	***Kernel size***	***Hidden Layers***
CNN	50	64	Relu	Adam	0.001	128	7	2
CNN-LSTM	50	128	Relu	Adam	0.001	128	7	2
BiLSTM	20	64	Relu	Adam	0.001	-	-	2
CNN-BiLSTM	100	128	Relu	Adam	0.001	128	7	2

[29] *https://medium.com/@peterworcester_29377/a-comparison-of-grid-search-and-randomized-search-using-scikit-learn-29823179bc85*

5.4. Effects of Weight Initializers on the Proposed Model

The weight initializers technique is used to prevent the output of the layers from suddenly growing or dropping(Vamsi et al., 2020). Even if this technique has shown performance improvement in another study, such as the study by Vamsi, it only enhances the performance of the models slightly when used in our proposed model. However, it does not show the negative impact on the model's performance. This implies that the weight initializers approach made no significant difference in the experiment we performed with our dataset.

5.5. Human Evaluation Result of the Model

As explained in section 3.12.3, the study assessed the presented chatbot model through human evaluation to measure the accurateness and acceptability of the chatbot model. To reach this goal, this study deployed the model on a flask server and developed the chatbot user interface in which the model is reachable to the end-user. And then, we select some maternal healthcare experts and womens as they use the chatbot, and based on their feedback, the model is evaluated. The study evaluated the accurateness of the model based on how much the presented chatbot answered the users' questions correctly, this is detailed in table 5.5, and calculated as the following equation, Eq.5.1.

$$Correctness = \frac{Total\ number\ of\ correclty\ answered\ by\ the\ model}{Total\ number\ of\ question\ users\ ask} * 100 \quad \textit{Eq.5.1}$$

Table 5.5 Correctness of the Chatbot Model through Human Evaluation

Users	**No of Questions**	**Correctly responded**	**Wrongly responded**	**Correctness (100%)**
User_1(MHE)	35	30	5	85.71%
User_2(MHE)	25	22	7	88.00%
User_3(MHE)	40	34	6	85.00%
User_4(MHE)	30	24	8	80.00%
User_5(Moth)	30	27	5	90.00%
User_6(Moth)	35	29	8	82.86%
User_7(Moth)	36	28	8	77.78%
User_8(Moth)	40	34	2	85.00%
Average				**84.29%**

As explained in table 5.5, the accurateness (correctness) of the model based on the human evaluation is shown. This study selected four maternal health experts (MHE) and four mothers (Moth), those who can write and read the Afaan Oromo language. Those users ask the chatbot and count the total number of answers responded correctly by the model. In this way, the accurateness of the model is evaluated and 84.29% is obtained through human evaluation.

Additionally, the evaluation form is prepared as shown in (Appendix 2, 2.1) for those users to evaluate the acceptance of the model based on some chatbot evaluation parameters(Casas et al., 2020). Such as Efficiency which is the time interval between the users' questions for the model and the provided response by the model, user-friendly which is used to ensure the satisfaction (Attractiveness) of the presented method while the users use it, usability which is used to evaluate the simplicity of the model, and effectiveness which is referred to the accuracy and completeness with which specified users achieve their goals.

Table 5.6. Model Acceptance Evaluation Through Human Evaluation

	8 users					**Score(100%)**
Parameters	**Poor (0)**	**Not God(1)**	**Good(2)**	**Very good(3)**	**Excellent(4)**	
Response Time	0	0	0	1	7	96.8
User Friendly	0	0	1	2	5	81.3
Usability	0	0	0	2	6	93.6
Effectivenesss	0	0	2	1	5	71.8
Average						**85.93**

Based on the above parameters, all user fills the evaluation value for each parameter out of 4. Each parameter is evaluated in total out of 32 because the model is evaluated by 8 users. The acceptance of the presented chatbot model is calculated as the following equation through human evaluation.

$$Acceptance = \sum_{1}^{5} \frac{Number\ of\ users\ choose\ criteria\ value * Craiteria\ value}{32} * 100$$

Based on the above-mentioned parameters, the model was evaluated by end-users and the model got a total acceptance of 85.93%. As a result, we conclude from this as the presented chatbot model has a significant benefit in providing better information for pregnant women on pregnant-related health complications.

5.6. Discussion

To our knowledge, this study is the third study to explore Afaan Oromo Text-based Chatbot for different industries using deep learning approaches. This study undergoes several experiments with a newly collected dataset on four proposed deep learning models to improve the performance of the existing Afaan Oromo text-based chatbot. The CNN, CNN-LSTM, BiLSTM, and CNN-BiLSTM models have been used and evaluated in this study. Of the two types of word2vec model, the skip-gram model outperforms the CBOW model as experimented with them. The Grid search hyperparameter tuning algorithm is used in this work, to search for optimal hyper parameters for the network.

As explained in Tables 5.1, 5.2, and 5.3, the comparison has been made between all proposed models, and the CNN-BiLSTM model outperformed all other proposed models with 96.94% accuracy under stratified 5-fold cross-validation. Because the proposed CNN-BiLSTM model can extract local features using the convolution layer and learn sequence correlation with the help of the BiLSTM layer. And also, the Proposed CNN-BiLSTM model can use past and future information in the current state when working with sequential data. This is why the proposed CNN-BiLTM model performs better than other proposed deep learning models. The study ends up with an over-fitted model which is well-fitted only for the training set, even after we used cross-validation when illustrated in a loss curve graph, to handle these problems L2 regularization technique is employed. In addition, the two studies conducted on Afaan Oromo text-based Chabot previous are compared to this newly proposed method as follows.

The first study conducted on Afaan Oromo text-based Chatbot is “Developing Afaan Oromo Text-based Chatbot for Enhancing Maize Productivity” by Bedasa(Bedasa, 2021). The researcher has used DNN and Seq2Seq models and achieved 84.4% accuracy in the DNN model. When this study is compared to our proposed models, the proposed model is better in terms of accuracy in all models, as discussed in Table 5.1. The author used a small dataset with only 25 tags and used the test_split ratio since the author did not use the model overfit handling techniques. However, the proposed model employs cross-validation methods and the L2 regularization technique. Lastly, the researcher did not use the threshold value for the model to prevent the model from returning inappropriate responses.

The second study is presented on “Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers” by Fekadu(Eshetu, 2021). The researcher has used the BiLSTM model and achieved 82.6% accuracy with L2 regularization algorithms to handle model overfit. When Fikadu’s study is compared to the presented model in this study, our study is better in terms of accuracy in all models. This study used weight initialization algorithms, but Fikadu’s does not, this technique helps to prevent the recurrent neural network-based models from vanishing problems and preventing the output of the layers from suddenly growing or dropping. The author set the hyperparameter for the neural network manually, which is arduous to get the correct parameter to the network, however, our study uses the grid search hyperparameter tuning algorithm which is used to find the optimal hyperparameter for the models.

Finally, the research revealed an answer to the research questions posed in this study. As explained in section 1.4, the study poses three research questions and those questions are answered in this section.

RQ1. To what extent does the presented chatbot model contribute to providing the relevant information on maternal health problems?”. This research question deals with how the designed chatbot model contributes to helping pregnant mothers to minimize maternal health problems. As the chatbot model is used in many industries to assist users, it is very important to have this model in healthcare to offer better understanding and advice on pregnancy-related problems. Research by Google shows that one in every twenty Google searches is about health, this shows the need to receive treatment and health counseling digitally(Louisville, 2021). Many pregnant women want to know about their health status daily; the chatbot solution may assist them with pregnancy-related health concerns in ANC, such as danger sign recognition, dietary suggestions, family planning, emotional support, etc., 24/7. As many researchers suggest that traditional maternal health consulting has little significance when compared to chatbot model-based consultancy services(Mugoye et al., 2019). When comparing these all issues to the current way of offering counseling services for pregnant women, the presented chatbot method can provide a significant contribution to providing better awareness of maternal health problems. Generally, this research question is answered based on the acceptance and accurateness obtained through human evaluation.

RQ2. **“**Which deep learning algorithm is more suitable to design a chatbot model for maternal healthcare support?”. To answer this research question, the primary dataset was collected from different sources and trained the proposed deep learning models, as explained in section 3.3. The study set different criteria to determine the deep learning model typically employed in modern chatbot development, as mentioned in section 3.7. Based on those criteria, the study then presented four deep learning models such as CNN, CNN-LSTM, BiLSTM, and CNN-BiLSTM models. In an experiment we performed among these proposed models, the CNN-BiLSTM model outperformed the others. As a result, we chose the CNN-BiLSTM model as the best model for designing the maternal health consultant chatbot.

RQ3. **“**How do we improve the overall performance of the maternal health consultant chatbot model?”. To answer this research question, we undergo several steps like searching for the optimal hyperparameter for all proposed models using hyperparameter tuning, combining two models (hybrid models) such as CNN with BiLSTM, cross-validation, and weight initialization methods. We improved the performance of the proposed deep learning model for the maternal health consultant chatbot development by combining all of these factors.

5.7. Chatbot Model Prototype Architecture

Once the models are trained and outperform model is identified, no need to retrain the model again, it is then saved and deployed on the webserver. The best model is saved in .h5 file format and loaded on the flask web server and reachable to the users, as depicted in Fig.5.6. A user sends the question through an HTTP request, and the model processes the text question and responds with an appropriate response for the user.

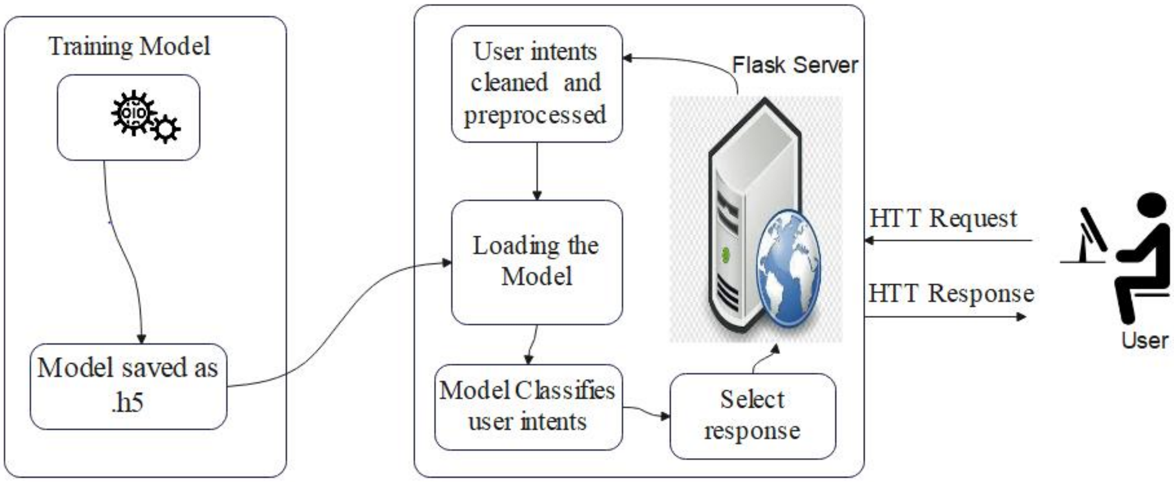


Fig.5.6 The architecture of the Chatbot Model Prototype

Here is some sample conversation of the designed chatbot for maternal healthcare support. The best-formed model is saved and deployed into the flask web server and used to respond to the users' questions, as depicted in Fig.5.7.

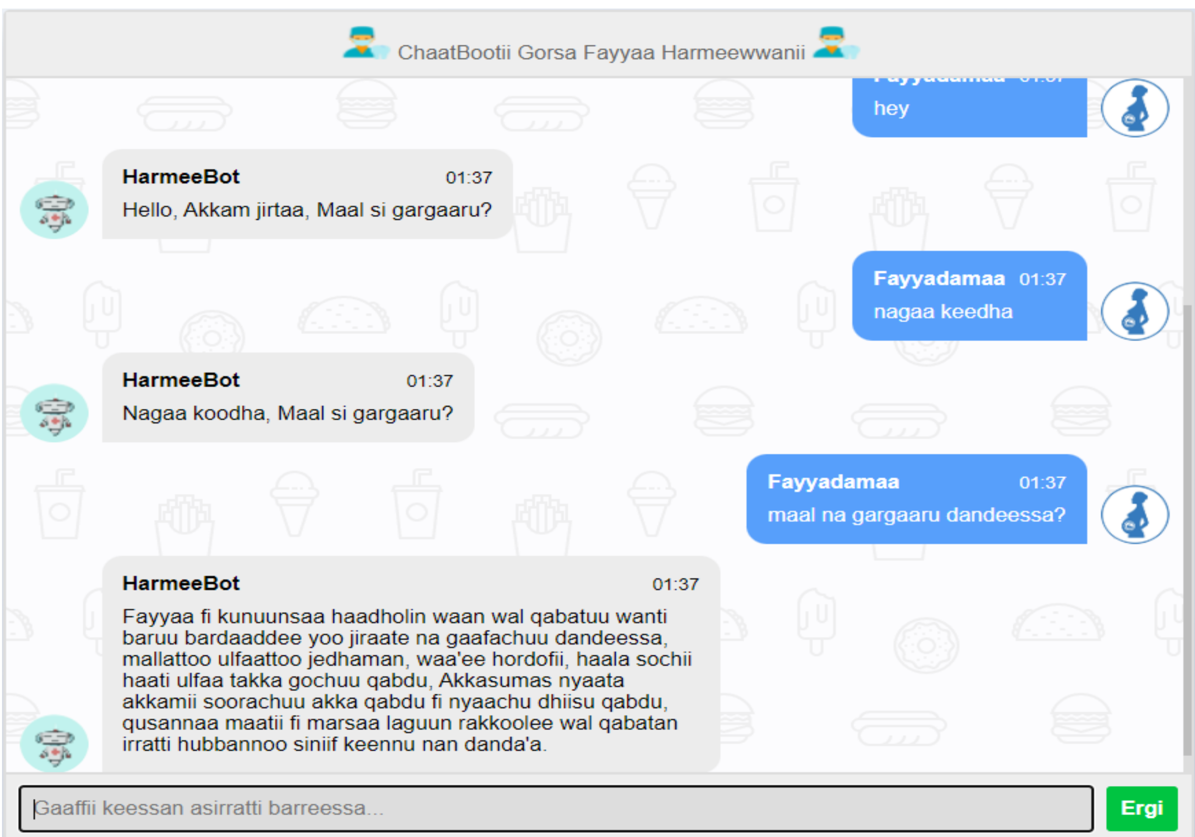


Fig.5.7 Sample Conversation with Developed Chatbot

CONCLUSION AND RECOMMENDATION

Conclusion

The study presented Afaan Oromo text-based maternal health consultant chatbot using deep learning approaches. Various approaches that have been used in modern chatbot design and related works are reviewed in this study. To achieve the objective of this study, the primary dataset that contains 1962 instances was collected from various sources such as ORHB, hospitals, health centers, websites, etc., and prepared; since there were previously no prepared datasets available for this purpose. The collected dataset was reviewed by two maternal health experts, to check the accurateness of the dataset. Several data preprocessing such as tokenization, stop word removal, removing unrelated items, etc., have been used to prepare and clean a dataset suitable for deep learning models. And, then we used word embedding techniques called Word2Vec models to represent textual data with vector representation. Of the two types of word2vec technique, the skip-gram outperforms the CBOW with the proposed models. The output of wor2vec word embedding is fed into deep learning models as input data.

We presented four deep learning models for this work such as CNN, CNN-LSTM, BiLSTM, and CNN-BiLSTM, and evaluated them under 5-fold cross-validation techniques. As we do have a smaller data size, the L2 regularization technique was employed to handle model overfitting. The performance of the proposed models was evaluated by tuning the right network hyperparameter using a grid search hyperparameter tuning algorithm for all models. The CNN model was the first model proposed in this study. It can extract local features to reduce dimensionality, but cannot learn sequence correlations. Rather, we looked into the CNN-LSTM model, which can extract local features and learn sequence correlations. The CNN-LSTM model is contrasted with the BiLSTM model, which was created to address the shortcomings of RNN and single LSTM models by incorporating past and future data in the current state. But the BiLSTM model cannot extract local features. Lastly, the CNN-BiLSTM model is presented, which can learn sequence correlation and captures local features at once. The model can extract local features and learn sequence correlations by considering past and future information. As the results of the model are discussed in section 5.2, the CNN-BiLSTM model outperforms other algorithms because its output layer obtains information from forward and backward directions in the current state to learn lean sequence correlation. This study generally has found that the hybrid CNN-BiLSTM model is the best performing model better than other proposed deep learning models with 96.94% accuracy.

Recommendation

This research aimed to address a gap in maternal healthcare. In this work, we are faced with sufficient data loss to develop a deep learning-based chatbot. Today, it is the epoch of AI. Most AI, for instance, ML and DL, use the collection of data to accomplish specific tasks and simplify tasks for humans in many industries. Though, there is no organized dataset in healthcare. Data is extremely important because it is the lifeblood of machine learning models. The machine learning model can be applied to healthcare fields to make services more efficient and effective if there are well-organized data such as patient history, known diseases with treatment, information about human health, etc. ML models can be used in the healthcare field to detect disease and health advisor, disease diagnosis, patient assistance, disease detection, etc., in this way it is possible to interconnect many industries to the health center as their customer diagnoses their health status anytime, anywhere. Therefore, we recommend the healthcare sector as they store information in a database for further use. We recommend next researchers as they use Hornmorpho which is used to analyze Amharic, Oromo, and Tigrinya words into their basic morphemes (meaningful parts) and produce words, given a root or stem and a representation of the word's grammatical structure

Future work

This work is aimed to develop an Afaan Oromo text-based maternal health consultant chatbot for maternal healthcare support. Today, many industries and companies want to have collaborated with hospitals through technology(Kandpal et al., 2020). And the users can then diagnose their health status irrespective of time and place. Thus, one example of such technology is AI-based models like the chatbot model. In the future, we plan to cover all health domains to reach customer satisfaction and use the chatbot model in all industries to diagnose patients anywhere, anytime. Such a service will assist the patient in identifying or understanding the problems they are facing.

In addition, the developed chatbot model uses text to communicate with people. Nevertheless, some users do not write the language with the correct language structure. Especially the Afaan Oromo language has a complex structure. For instance, the words ‘Gaara’ and ‘Garaa’ have different meanings, defined as ‘mountain’, and ‘abdomen’ respectively. Hence, to overcome such a problem the study planned to extend the study to the model that supports both text and speech conversation. And also there are some other NLPs text-preprocessing techniques such as lemmatization, steeming, inflectional morphology, number and date normalization, etc. Therefore, in the future, the study is planned to assess whether those techniques improve or affect the performance of the model. Furthermore, the research is open to any researcher to extend the study, using a multi-lingual dataset. He/she can repeat the same study with the same approaches and methods with a sufficient multi-lingual dataset.

***

Special Acknowledgement

This research was sponsored by Adama Science and Technology University with a grant number

ASTU/SM-R/-/22

Adama, Ethiopia

REFERENCES

Adamopoulou, E. (2020). An Overview of Chatbot Technology. Springer, 373–383. https://doi.org/10.1007/978-3-030-49186-4

Adamopoulou, E., & Moussiades, L. (2020). Chatbots: History, technology, and applications. Elsevier, 2(November), 100006. https://doi.org/10.1016/j.mlwa.2020.100006

Almansor, E. H., & Hussain, F. K. (2020). Survey on Intelligent Chatbots : State-of-the-Art and Future Research. 2, 534–543. https://doi.org/10.1007/978-3-030-22354-0

Alshahrani, S., & Kapetanios, E. (2016). Are deep learning approaches suitable for natural language processing? Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9612(June), 343–349. https://doi.org/10.1007/978-3-319-41754-7_33

Atnafu, A., Kebede, A., Misganaw, B., Teshome, D. F., Biks, G. A., Demissie, G. D., Wolde, H. F., Gelaye, K. A., Yitayal, M., Ayele, T. A., Azale, T., Derso, T., Gebremedhin, T., & Dellie, E. (2020). Determinants of the continuum of maternal healthcare services in northwest Ethiopia: Findings from the primary health care project. Journal of Pregnancy, 2020. https://doi.org/10.1155/2020/4318197

Bedasa, S. (2021). Developing Afaan Oromoo Text-based Chatbot for Enhancing Maize productivity (Issue January).

Berri, K. M., Adaba, Y. K., Tarefasa, T. G., Bededa, N. D., & Fekene, D. B. (2020). Maternal health service utilization from urban health extension professionals and associated factors among women who gave birth in the last one year in Ambo town, Oromia regional state, Ethiopia, 2018. BMC Public Health, 20(1), 1–10. https://doi.org/10.1186/s12889-020-08641-5

Bird, S., Klein, E., & Loper, E. (2022). NLTK - Natural Language Toolkit. In Http://www.nltk.org/ (pp. 1–11). https://www.nltk.org/%0Ahttp://www.nltk.org/book/ch00.html

Borah, B., Pathak, D., Sarmah, P., Som, B., & Nandi, S. (2019). Survey of Textbased Chatbot in Perspective of Recent Technologies. In Communications in Computer and Information Science (Vol. 1031). Springer Singapore. https://doi.org/10.1007/978-981-13-8581-0_7

Casas, J., Khaled, O. A., Mugellini, E., & Cudré-mauroux, P. (2020). Trends & Methods in Chatbot Evaluation Trends & Methods in Chatbot Evaluation. October. https://doi.org/10.1145/3395035.3425319

Choudhury, A. (2019). TensorFlow vs Keras: Which One Should You Choose. In Analytics India Magazine. https://analyticsindiamag.com/tensorflow-vs-keras-which-one-should-you-choose/

Dash, B. (2017). Maternal and Child Health Care. A Comprehensive Textbook of Community Health Nursing, January, 282–282. https://doi.org/10.5005/jp/books/12959_8

Dida, B. D., and N. (2015). Challenges of Maternal Health Services Utilization and Provision from Health Posts in Bale Zone, Oromiya Regional State, Southeast Ethiopia: Qualitative Study. Primary Health Care Open Access. https://doi.org/10.4172/2167-1079.1000189

Dilmegani, C. (2021). Top 5 Chatbot Use Cases & Applications in Marketing in 2022. https://research.aimultiple.com/chatbot-marketing/

Eshetu, F. (2021). Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services. http://etd.astu.edu.et/handle/123456789/20

Fadhil, A. (2018). Beyond Patient Monitoring: Conversational Agents Role in Telemedicine & Healthcare Support For Home-Living Elderly Individuals.
http://arxiv.org/abs/1803.06000

Gupta, D. (2020). Activation Functions _ Fundamentals Of Deep Learning. https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/

Hien, H. T., Cuong, P. N., Nam, L. N. H., Nhung, H. L. T. K., & Thang, L. D. (2018). Intelligent assistants in higher-education environments: The FIT-EBOt, a chatbot for administrative and learning support. ACM International Conference Proceeding Series, 69–76. https://doi.org/10.1145/3287921.3287937

Hussain, S., Ameri Sianaki, O., & Ababneh, N. (2019). A Survey on Conversational Agents/Chatbots Classification and Design Techniques. In Advances in Intelligent Systems and Computing (Vol. 927, Issue March). Springer International Publishing. https://doi.org/10.1007/978-3-030-15035-8_93

Hutapea, A. (2017). CHATBOT : Architecture, Design, & Development (tesis. no usar). Academia, 4.

Jason Brownlee. (2017). A Gentle Introduction to the Bag-of-Words Model. In A Gentle Introduction to the Bag-of-Words Model (p. 1). https://machinelearningmastery.com/gentle-introduction-bag-words-model/

Kadhim, A. I. (2019). Term Weighting for Feature Extraction on Twitter: A Comparison between BM25 and TF-IDF. 2019 International Conference on Advanced Science and Engineering, ICOASE 2019, 124–128. https://doi.org/10.1109/ICOASE.2019.8723825

Kandpal, P., Jasnani, K., Raut, R., & Bhorge, S. (2020). Contextual chatbot for healthcare purposes (using deep learning). Proceedings of the World Conference on Smart Trends in Systems, Security and Sustainability, WS4 2020, 625–634. https://doi.org/10.1109/WorldS450073.2020.9210351

Kant. (2021). How to perform Quality Assurance for Machine Learning models _ SennaLabs. https://sennalabs.com/en/blogs/how-to-perform-quality-assurance-for-machine-learning-models

Khanna, C. (2021). Text pre-processing: Stop word removal using different libraries. https://towardsdatascience.com/text-pre-processing-stop-words-removal-using-different-libraries-f20bac19929a

Kidwai, B., & Rk, N. (2020). Design and Development of Diagnostic Chabot for supporting Primary Health Care Systems. Procedia Computer Science, 167(2019), 75–84.

https://doi.org/10.1016/j.procs.2020.03.184

Kiross, G. T., Chojenta, C., Barker, D., & Loxton, D. (2021). Optimum maternal healthcare service utilization and infant mortality in Ethiopia. BMC Pregnancy and Childbirth, 21(1), 1–12. https://doi.org/10.1186/s12884-021-03860-z

Krishnan, K. (2020). 18 Chatbot Development Frameworks and Chatbot Platforms. https://cedextech.com/chatbot-development-frameworks/%0Ahttps://www.cedextech.com/blog/chatbot-development-frameworks/

Kumar, R., & Ali, M. M. (2021). A Review on Chatbot Design and Implementation Techniques A Review on Chatbot Design and Implementation Techniques. Research Gate, 7(January 2008). https://www.researchgate.net/publication/348569890%0AA

Kurtis, P. (2021). Fighting Overfitting With L1 or L2 Regularization_ Which One Is Better_ - Neptune. https://neptune.ai/blog/fighting-overfitting-with-l1-or-l2-regularization

Leonid, D. (2020). A Survey on Activation Functions and their relation with Xavier and He Normal Initialization. ArXiv. https://doi.org/https://doi.org/10.48550/arXiv.2004.06632

Liang, H., Sun, X., Sun, Y., & Gao, Y. (2017). Text feature extraction based on deep learning: a review. Eurasip Journal on Wireless Communications and Networking, 2017(1), 1–12. https://doi.org/10.1186/s13638-017-0993-1

Louisville. (2021). Application Use Cases of Chatbots in Healthcare. https://blog.vsoftconsulting.com/blog/application-use-cases-of-chatbots-in-healthcare

Mugoye, K., Okoyo, H., & McOyowo, S. (2019). Smart-bot Technology: Conversational Agents Role in Maternal Healthcare Support. 2019 IST-Africa Week Conference, IST-Africa 2019, 1–7. https://doi.org/10.23919/ISTAFRICA.2019.8764817

Nabi, J. (2019). Hyper-parameter Tuning Techniques in Deep Learning _ by Javaid Nabi _ Towards Data Science. https://towardsdatascience.com/hyper-parameter-tuning-techniques-in-deep-learning-4dad592c63c8

Nefa, S. A., Ibrahim, A. M., Mulushewa, S. S., Wonga, M. M., Lee, H., Siyum, G. A., Amentie, M. A., Chebo, A. F., Birhanu, B. G., Battle, J., Hiluf, M. K., Kebedom, A. G., Gobezayehu, A. G., Robins, A., & Bawah, A. A. (2019). Study design for the 2019 baseline survey of newly established longitudinal surveillance woredas as the field lab to serve child health improvement project of Benishangul- Gumuz region, Ethiopia. JGHS, 1(July), 1–14.

Nuruzzaman, M., & Hussain, O. K. (2018). A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks. Proceedings - 2018 IEEE 15th International Conference on e-Business Engineering, ICEBE 2018, 54–61. https://doi.org/10.1109/ICEBE.2018.00019

Okonkwo, C. W., & Ade-ibijola, A. (2021). Computers and Education : Artificial Intelligence Chatbots applications in education : A systematic review. Elsevier, 2, 100033. https://doi.org/10.1016/j.caeai.2021.100033

Oromia Regional Health Bureau. (n.d.). https://www.devex.com/organizations/oromia-regional-health-bureau-121410

Oromo language - Wikipedia. (2021). https://en.wikipedia.org/wiki/Oromo_language

Patil, S., Mudaliar, V. M., Kamat, P., & Gite, S. (2020). R ESEARCH A ARTICLE LSTM-based Ensemble Network to enhance the learning of long-term dependencies in the chatbot. 25.

Reddy Karri, S. P., & Santhosh Kumar, B. (2020). Deep learning techniques for implementation of chatbots. 2020 International Conference on Computer Communication and Informatics, ICCCI 2020, 20–24. https://doi.org/10.1109/ICCCI48352.2020.9104143

Rendel, A., Fernandez, R., Hoory, R., & Ramabhadran, B. (2016). Using continuous lexical embeddings to improve symbolic-prosody prediction in a text-to-speech front-end. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2016-May, 5655–5659. https://doi.org/10.1109/ICASSP.2016.7472760

Sharma, P. (2022). Different Types of Cross-Validations in Machine Learning - Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/02/different-types-of-cross-validations-in-machine-learning/

Shegaw Mulu Tarekegn1*, L. S. L. and V. G. (2014). Determinants of maternal health service utilization in Ethiopia: analysis of the 2011 Ethiopian Demographic and Health Survey. Paediatric and Perinatal Epidemiology, 15(s1), 1–42.

Singh, S., & Thakur, H. K. (2020). Survey of Various AI Chatbots Based on Technology Used. ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies, and Optimization (Trends and Future Directions), 1074–1079. https://doi.org/10.1109/ICRITO48877.2020.9197943

Sivakumar, S., Videla, L. S., Rajesh Kumar, T., Nagaraj, J., Itnal, S., & Haritha, D. (2020). Review on Word2Vec Word Embedding Neural Net. Proceedings - International Conference on Smart Electronics and Communication, ICO SEC 2020, Icosec, 282–290. https://doi.org/10.1109/ICOSEC49089.2020.9215319

Tam, S. (2021). A ConvBiLSTM Deep Learning Model-Based Approach for Twitter Sentiment Classification. 9, 41283–41293. https://doi.org/10.1109/ACCESS.2021.3064830

The team, G. L. (2020). What are Cross-Validation and its types in Machine learning_ Great Learning? https://www.mygreatlearning.com/blog/cross-validation/

Tolessa, T. (2013). Early History of Written Oromo Language up to 1900. Science, Technology and Arts Research Journal, 1(2), 76. https://doi.org/10.4314/star.v1i2.98787

Vamsi, G. K., Rasool, A., & Hajela, G. (2020). Chatbot: A Deep Neural Network Based Human to Machine Conversation Model. 2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020. https://doi.org/10.1109/ICCCNT49239.2020.9225395

Wei, C., Yu, Z., & Fong, S. (2018). How to Build a Chatbot: Chatbot framework and its capabilities. ACM International Conference Proceeding Series, 5, 369–373. https://doi.org/10.1145/3195106.3195169

Wu, J., Woo, H., Tamura, Y., Moro, A., Yamashita, A., & Asama, H. (2019). Pedestrian trajectory prediction using BiRNN encoder-decoder framework. 1864. https://doi.org/10.1080/01691864.2019.1635910

Yao, C., Qu, Y., Jin, B., Member, S., Guo, L., Li, C., Cui, W., & Feng, L. (2016). A Convolutional Neural Network Model for Online Medical Guidance. IEEE, January 2019. https://doi.org/10.1109/ACCESS.2016.2594839

APPENDICES

Appendix 1: Sample Code

1.1. Text Normalization

```
patterns= [string.replace("otoo","osoo")for string in patterns]
patterns= [string.replace("ugaa","dhugaa")for string in patterns]
patterns= [string.replace("dabree","darbee")for string in patterns]
patterns= [string.replace("mini","miti")for string in patterns]
patterns= [string.replace("baayyee","baay'ee")for string in patterns]
patterns= [string.replace("ykn","Yookiin")for string in patterns]
patterns= [string.replace("Waahee","Waa'ee") for string in patters]
patterns= [string.replace("eennu","eenyu") for string in patterns]
patterns= [string.replace("haga","hanga") for string in patterns]
patterns= [string.replace("ega","erga") for string in patterns]
patterns= [string.replace("jiha", "ji'a") for string in patterns]
patterns= [string.replace("bahe", "ba'e") for string in patterns]
patterns= [string.replace("yaahe", "yaa'e") for string in patterns]
```

1.2. List of Stop words

itti	kanaafuu	natti	akkasumas	sitti	isii	ofirratti
narraa	tanaaf	jala	na	kan	irratti	ofirraa
akka	immoo	gubbaa	nu	illee	wanta	jira
ati	hogguu	hanga	nuti	ala	asii	narratti
ammo	alatti	of	akkuma	sun	haa	keessaa
kana	siin	jara	nurraa	isaa	irraa	turuuf
an	ittuu	henna	sana	Kanaaf	kam	kootu
kanaafi	tanaafuu	duuba	silaa	eegasii	akkamii	kanneen
ani	amma	kee	kennaa	ishiif	sin	wanti
tun	waan	bira	siif	koo	baayeen	miti
booddee	warra	ishiirraa	eega	iseen	wal	sirraa
keessa	ta'ullee	teenya	yommuu	yoon	si	Miti
inni	aan	isatti	jedhaman	kana	jirutti	irraan
keessan	ishii	keenya	nan	yookiin	anan	naaf
gara	kanan	kami	gad	maalif	akkamiitu	isiin
isaan	baayee	akkamii	atoo	moo	takkaaf	Yookaan
naf	ni	kami	akkan	isa	akkamittin	Hoo

dura	isin	ishee	takkaa	ree	Kun	keen
kaa	takka	achi	gama	yoom	In	Mee
waa	eega	irra	keessatiif	utuu	isaanirraa	isaaf
booda	waan	keessatti	akkamiiti	kiyya	akkamiin	Akkamitti
malee	yoo	fi				

1.3. Sample Code For all Proposed Model

1.3.1. CNN Model Sample Code

```
kfoldcv = StratifiedKFold(n_splits=k, shuffle=True, random_state=42)
cvscores = []
for train, test in kfoldcv.split(Xw2v, Yw2v):
    modelcn = Sequential()
    modelcn.add(w2v_embedding)
    modelcn.add(Conv1D(filters=128, kernel_size=7, activation='relu'))
    modelcn.add(MaxPooling1D(pool_size = 5))
   #modelcn.add(Conv1D(filters=128, kernel_size=5,activation='relu'))
   #modelcn.add(MaxPooling1D(pool_size = 3))
    modelcn.add(Flatten())
    modelcn.add(Dense(300, activation='relu'))
    modelcn.add(Dense(200, activation='relu'))
    modelcn.add(Dense(103, activation='softmax'))
    sgd = SGD(learning_rate=0.001, decay=1e-
    6, momentum=0.9, nesterov=True)
    modelcn.compile(loss = 'sparse_categorical_crossentropy',optimizer=A
    dam(learning_rate=0.001),metrics =['accuracy'])
    historycn = modelcnwc.fit(Xw2v[train],Yw2v[train],epochs=50,verbose=
    1,validation_data=(Xw2v[test],Yw2v[test]), batch_size=64)
    score = modelcn.evaluate(Xw2v[test],Yw2v[test], verbose=1)
    print("%s: %.2f%%" % (modelcn.metrics_names[1], score[1]*100))
    cvscores.append(score[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (np.mean(cvscores), np.std(cvscores)))
```

1.3.2. CNN-LSTM Model Sample Code

```
kfold = StratifiedKFold(n_splits=k, shuffle=True, random_state=42)
cvscores = []
for train, test in kfold.split(Xw2v, Yw2v):
    modelcnlstm = Sequential()
    modelcnlstm.add(w2v_embedding)
    modelcnlstm.add(Conv1D(filters=128,kernel_size=7,activation='relu'))
    modelcnlstm.add(MaxPooling1D(pool_size = 5))
    modelcnlstm.add(LSTM(500,return_sequences=True,activation='relu'))
    modelcnlstmt.add(LSTM(200,return_sequences=False,activation='relu'))
   #modelcnlstm.add(Flatten())
    modelcnlstm.add(Dense(103, activation='softmax'))
    sgd = SGD(learning_rate=0.001, decay=1e-
    6, momentum=0.7, nesterov=True)
    modelcnlstm.compile(loss = 'sparse_categorical_crossentropy',optimiz
    er = Adam(learning_rate=0.001),metrics = ['accuracy'])
    historycnlstm = modelcnlstm.fit(Xw2v[train],Yw2v[train],epochs=50,ve
    rbose=1,validation_data=(Xw2v[test],Yw2v[test]), batch_size=64)
    score = modelcnlstm.evaluate(Xw2v[test],Yw2v[test],batch_size=64)
    print("%s: %.2f%%" %(modelcnlstm.metrics_names[1], score[1]*100))
    cvscores.append(score[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (np.mean(cvscores), np.std(cvscores)))
```

1.3.3. BiLSTM Model Sample Code

```
kfoldcv = StratifiedKFold(n_splits=k, shuffle=True, random_state=42)
cvscores = []
for train, test in kfoldcv.split(Xw2v, Yw2v):
    modelbirnn = keras.Sequential()
    modelbirnn.add(w2v_embedding)
    modelbirnn.add(tf.keras.layers.Bidirectional(LSTM(500, return_sequen
    ces=True, activation ='relu')))
    modelbirnn.add(tf.keras.layers.Bidirectional(LSTM(256, return_sequen
    ces=False, activation='relu')))
    modelbirnn.add(Dense(103, activation='softmax'))
    sgd = SGD(learning_rate=0.01, decay=1e-
    6, momentum=0.9, nesterov=True)
    modelbirnn.compile(loss='sparse_categorical_crossentropy', optimizer
    =Adam(learning_rate=0.001), metrics=['accuracy'])
    historybirnn=modelbirnn.fit(Xw2v[train],Yw2v[train], epochs=20,valid
    ation_data=(Xw2v[test],Yw2v[test]), verbose=1, batch_size=64)
    score = modelbirnn.evaluate(Xw2v[test],Yw2v[test], verbose=1)
    print("%s: %.2f%%" % (modelbirnn.metrics_names[1], score[1]*100))
```

```
    cvscores.append(score[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (np.mean(cvscores), np.std(cvscores)))
```

1.3.4. CNN-BiLSTM Model Sample Code

```
kfoldcv = StratifiedKFold(n_splits=k, shuffle=True, random_state=42)
cvscores = []
for train, test in kfoldcv.split(Xw2v, Yw2v):
    modelcnnBirnn= Sequential()
    modelcnnBirnn.add(w2v_embedding)
    modelcnnBirnn.add(Conv1D(filters=128,
    kernel_size=7, activation='relu'))
    modelcnnBirnn.add(MaxPooling1D(pool_size=5))
    modelcnnBirnn.add(Bidirectional(LSTM(500, return_sequences=True, act
    ivation='relu’))#kernel_regularizer=tf.keras.regularizers.l2(0.001)
    modelcnnBirnn.add(Bidirectional(LSTM(200,return_sequences=False,acti
    vation='relu')))#kernel_regularizer=tf.keras.regularizers.l2(0.001)
    modelcnnBirnn.add(Flatten())
    modelcnnBirnn.add(Dense(103, activation='softmax'))
    sgd=SGD(learning_rate=0.001, decay=1e-, momentum=0.9, nesterov=True)
    modelcnnBirnn.compile( loss='sparse_categorical_crossentropy',optimi
    zer=Adam(learning_rate=0.001), metrics=['accuracy'])
    from keras.callbacks import EarlyStopping, ModelCheckpoint
    e=EarlyStopping(monitor='val_loss',mode='min',verbose=1,patience=10)
    mc=ModelCheckpoint('./modelcnnBirnn.h5', monitor='val_accuracy', mod
    e='max', verbose=1, save_best_only=True)
    historyconvRnn=modelcnnBirnn.fit(Xw2v[train], Yw2v[train], epochs=50
    , verbose=1,validation_data=(Xw2v[test], Yw2v[test]),batch_size=128)
     score = modelcnnBirnn.evaluate(Xw2v[test], Yw2v[test], verbose=1)
    print("%s: %.2f%%" % (modelcnnBirnn.metrics_names[1], score[1]*100))
    cvscores.append(score[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (np.mean(cvscores), np.std(cvscores)))
```

1.3.5. Sample for Saving and Loading the Model

```
pickle.dump (pattern_words, open ('pattern_words.pkl', 'wb'))
pickle.dump (classes, open ('classes.pkl', 'wb'))
model.save (/Path/HarmeBot.h5') # save the best model
from keras.models import load_model
model = load_model(' HarmeBot.h5') #loading the saved model to use it.
pattern_words = pickle.load (open ('pattern_words.pkl','rb'))
classes = pickle.load(open('classes.pkl', 'rb'))
```

1.3.6. Sample Code of user intent classification

```
def predict_class(patterns, model):
    p = word_emb(patterns, voc,show_details=False)
    res = model.predict([p])[0]
    THRESHOLD_VALUE = 0.75#threshold Value
    results = [[i,r] for i,r in enumerate(res) if
r>THRESHOLD_VALUE]
    # sort by strength of probability
    results.sort(key=lambda o: o[1], reverse=True)
    return_sort = []
    for r in results:
        return_sort.append({"intent": classes[r[0]],
"probability": str(r[1])})
    return return_sort
def getResponse(ints, intents_json):
    try:
       tag = ints[0]['intent']
       list_of_intents = intents_json['intents']
       for i in list_of_intents:
            if(i['tag']==tag):
               return random.choice(i['responses'])
    except:
        return "Dhiifama maal Jechuu akka barbaaddan naaf hin
gallee irra deebi'un yaala!"
```

Appendix 2. Evaluation Form

2.1. Sample Evaluation Form of designed Chatbot Model

HarmeeBot Chatbot Evaluation Form

Gaaffiiwwan armaan gaditti dhiyaataniif akkaata HarmeeBot fayyadamtan irratti hunda'uun deebisa. Yoo kan baay'ee baay'ee sinitti tole ta'e haala waldura duuba jiruun tokko qofa filadha, Baay'ee baay'ee gaariidha(4), Baay'ee gaariidha(3), gaariidha(2), Gad bu'aadha(1), Baay'ee gad Bu'aadha(0)

haileguta08@gmail.com (not shared) Switch account

* Required

Maqaa Guutaa keessan barreessa! *

Your answer

1) Chatbot isin fayyadamtan kun deebii barbaaddan ariitin siniif deebisu irratti hangam sinitti tole? *

- ◯ Baay'ee Baay'ee Gaariidha
- ◯ Baay'ee Gaariidha
- ◯ Gaariidha
- ◯ Gad Bu'aadha
- ◯ Baay'ee Gad Bu'aadha

2) Chatbot isin fayyadamtan kun yeroo fayyadamta ammam hawwataadha yookiin fayyadamuuf hangam sinitti tolee? *

Chatbotin isin fayyadamtan kun itti fufee mamiltoota hundumaaf hojjirra akka ooluf karaa salphaa ta'een hojjachuun isaa hagam isin gammachisee? *

- ◯ Baay'ee Baay'ee Gaariidha
- ◯ Baay'ee Gaariidha
- ◯ Gaariidha
- ◯ Gad Bu'aadha
- ◯ Baay'ee Gad Bu'aadha

4) Chatbot isin fayyadamtan kun deebii sirrii ta'ee fi guutuu ta'e siniif deebisu irratti hagam sinitti tole? *

- ◯ Baay'ee Baay'ee Gaariidha
- ◯ Baay'ee Gaariidha
- ◯ Gaariidha
- ◯ Gad Bu'aadha
- ◯ Baay'ee Gad Bu'aadha

Yaada yoo qabattaan nuuf ka'a, Waan heyyamamaa taatanii nu gargartaniif galatooma🙏🙏🙏

Your answer

Submit

Clear form

***