MEKELLE UNIVERSITY

ETHIOPIAN INSTITUTE OF TECHNOLOGY-MEKELLE (EITM)

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE

DEVELOPMENT OF A TEXT-BASED, AMHARIC-LANGUAGE CHATBOT FOR MATERNAL HEALTH CONSULTATION USING SUPERVISED MACHINE LEARNING

BY:

BIRTUKAN NGATU

ADVISOR
Dr. HAILAY B. (PhD)

A THESIS SUBMITTED TO SCHOOL OF COMPUTING FOR THE PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

October, 2025

Mekelle, Tigray, Ethiopia

MEKELLE UNIVERSITY

ETHIOPIAN INSTITUTE OF TECHNOLOGY-MEKELLE(EIT-M)

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE

DEVELOPMENT OF A TEXT-BASED, AMHARIC-LANGUAGE CHATBOT FOR MATERNAL HEALTH CONSULTATION USING SUPERVISED MACHINE LEARNING

BY

BIRTUKAN NGATU ENGDAW

Approval by Board of Examiners

Teages Kalayu	
Chairman Dept. graduate committee	Signature
Dr. Hailay B.(PhD)	
Advisor	Signature
Behailu Getachew (PhD)	
Internal Examiner	Signature
Tesfay Gidey(PhD)	
External Examiner	Signature

DECLARATION

I, the undersigned student declare that this thesis is my original work and has not been presented as a partial requirement for a degree in any university. All the resources used for this thesis work are cited and are acknowledged.

_____

Birtukan Nigatu

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

_____

Dr. Hailay B.(PhD)

ENDORSEMENT

This thesis has been submitted to Ethiopia Institution Technology Mekelle University School of Computing Department of Information Technology with my approval as a university advisor.

Hailay B. (Ph.D.)	
Advisor	Signature

Mekelle University, Ethiopia Institution Technology Mekelle, School of Computing, Department of Computer Science.

Acknowledgment

First and foremost, I would like to thank Almighty God for granting me the strength, wisdom, and perseverance to complete this thesis.

I am profoundly grateful to my principal advisor, **Dr. Hailay B. (PhD)**, whose invaluable guidance, constructive feedback, and continuous encouragement have been instrumental in shaping this work. His insightful ideas and unwavering support made this research possible.

Finally, I wish to express my deepest gratitude to my beloved mother, **Abirehet Hagos**, for her unconditional love, care, and support throughout my academic journey. Her constant encouragement and sacrifices during challenging times have been my greatest source of strength.

Table of Contents

Acknowledgment i

Table of Contents ii

List Of Figures vi

List Of Tables vii

Acronyms and Abbreviations viii

Abstract ix

CHAPTER ONE 1

INTRODUCTION 1

1.1. Background of the Study 1

1.2. Motivation 3

1.3. Research Gap 4

1.4. Problem Statement 5

1.5. Research Objectives 5

1.5.1. General Objective: 5

1.5.2. Specific Objectives: 5

1.5. Significance and Contributions of the Study 6

1.7. Research Questions 7

1.8. Scope and Limitations of the Study 8

1.8.1. Scope 8

1.8.2. Limitations 9

1.9 Research Methodology and Design 9

1.10 Organization of the Research 10

CHAPTER TWO 12

LITERATURE REVIEW 12

2.1. Introduction 12

2.2. Maternal Health Information in Ethiopia 12
2.3 Chatbots in Healthcare 12
2.4. Chatbots and Local Language Support 13
2.5. User-Centered Design in Chatbot Development 13
2.6. Conversational Chatbot Technology 14
2.7. Artificial Intelligence vs. Conversational Bots 15
2.7.1 Artificial Intelligence (AI): 15
2.7.2 Conversational Bots: 15
2.8. Types of Chatbots and Chatbot Modelling Approach 16
2.9. Overview of Machine Learning in Chatbot Development 16
1. Role of ML in Understanding Amharic Text: 17
2. Training Approaches and Algorithms: 17
2.10. Chatbot Modeling Approach for the Study 18
2.11. Overview of conversational agents 19
2.12. Application of conversational agents 22
2.12.1. Application of conversational agent for cultural heritage 23
Applications of Conversational Agents 24
2.12.2. Application of conversational agent for online market 24
2.12.3. Application of conversational agent in healthcare 25
2.12.4. Application of Conversational Agent for Education 26
2.12.5. Application of Conversational Agent for Psychologist 26
2.12.6. Application of conversational agent as customer service support 26
2.13. Techniques for designing conversational agent or Chatbot 27
2.14. Gaps in the Literature review 28
2.15 Summary of the review 28
2.16. Discussion of Related Works 29

CHAPTER THREE 31
METHODOLOGY 31
3.1 Introduction 31
3.1. Design Science Research Methodology 31
3.1.1 DSRM Framework Phases 31
3.2. Data Collection 34
3.3. Dataset Preparation 34
3.4. Data Preprocessing 36
3.4.1. Text Normalization and Cleaning 36
3.4.2. Tokenization 38
3.4.3. Stop Word Removing 38
3.4.4. Word Embedding 39
3.5. Model Building and Selection 40
3.6. Performance Evaluation 40
CHAPTER FOUR 41
DESIGN AND IMPLEMENTATION 41
4.1. Introduction 41
4.2. The Proposed Architecture for the Chatbot System 41
4.3. Complete System Architecture and Data Flow 42
4.4. Data Preprocessing Implementation 43
4.4.1. Text Normalization and Cleaning 43
4.4.2. Tokenization 44
4.4.3. Stop Word Removing 45
4.5. Word Embedding Implementation 46
4.6. Response Management System 47
4.7. Rationale for Ensemble Modeling in Healthcare Chatbot's 48

4.8. Model Building and Selection Implementation 48
CHAPTER FIVE 54
EXPERIMENTAL RESULTS AND TECHNICAL VALIDATION 54
5.1 Introduction 54
5.2. Dataset Description 54
5.3. Data Preprocessing 57
5.3.1. Text Normalization and Cleaning 57
5.3.2. Tokenization 57
5.3.3. Stop Word Removing 57
5.4. Word Embedding 57
5.4.1. Single Model 58
5.5. Ensemble Model Averaging Performance 63
5.6. GUI Chat Interface for Intent Classification 66
5.7. Comprehensive Validation Framework 67
5.7.1. Theoretical Foundations 68
5.7.2. Validation Methodology 69
5.7.3. Prototype Assessment against the Framework 70
5.7.4. Limitations of the Current Validation 71
5.8. Validation Summary 72
CHAPTER SIX 73
CONCLUSIONS AND FUTURE WORKS 73
6.1 Conclusion 73
6.2 Recommendation and Future Work 74
References 76

List Of Figures

Figure 1: user Conversation with Babylon Health 13
Figure 2: Broad classification of Chatbot's (Hussain et al., 2019) 20
Figure 3: Main classification of chatbot's based on their goal (Hussain et al., 2019) 20
Figure 4: Design Science Research Methodology (DSRM) Process Model (Peffers, et al,) 33
Figure 5: JSON file snapshot 35
Figure 6: The proposed architecture for the chatbot system 42
Figure 7: Snapshot of python Code for Text normalization implementation. 43
Figure 8: Snapshot of python Code Text normalization implementation. 44
Figure 9: Snapshot of python Code for Tokenization implementation 45
Figure 10: Snapshot of python Code for Removal of stop word implementation 46
Figure 11: Implementation of bag-of-words for training in Python 47
Figure 12: Snapshot of python code for response management 47
Figure 13: Implementation of the MLP model in keras 49
Figure 14: Visualization of the MLP model 50
Figure 15: The MLP model summary 50
Figure 16: Visualization of the MLP model 51
Figure 17: function to load all the saved models 52
Figure 18: function for ensemble prediction 52
Figure 19:Python code for the GUI chat interface 53
Figure 20: Snapshot of python code for bag-of-words 58
Figure 21: Python code for hyperparameter tuning 59
Figure 22: Output the performance of the MLP model on the train and test datasets 60
Figure 23: Learning curves of model accuracy on train & test dataset over each training epoch 61
Figure 24:Python code of evaluation model () function 62
Figure 25: Histogram of a single model test accuracy over 30 repeats 62
Figure 26: Box and Whisker plot of a single model test accuracy over 30 repeats 63
Figure 27:Test Accuracy of the first fits 20 models 64
Figure 28: Line plot of ensemble size versus model test accuracy 65
Figure 29: Histogram of a five-member ensemble model test accuracy over 30 repeats 65
Figure 30: Box and Whisker plot of a five-member ensemble model test accuracy over 30 repeats 66

Figure 31: GUI showing conversation between the user and the bot 67

List Of Tables

Table 1: Comparison of Chatbot Categories by Core Technology and Capabilities..... 16
Table 2: Techniques for designing conversational agent or Chatbot..... 27
Table 3: Intent (target class) description..... 54

Acronyms and Abbreviations

AI - Artificial Intelligence

NLP - Natural Language Processing

DSRM - Design Science Research Methodology

IBM - International Business Machines Corporation

AIML - Artificial Intelligence Markup Language

BERT - Bidirectional Encoder Representations from Transformers

PWA - Progressive Web App

ALICE - Artificial Linguistic Internet Computer Entity

CNN - Convolutional Neural Networks

ELIZA - Elisabeth

RNN - Recurrent Neural Network

SVM - Support Vector Machines

Seq2Seq - Sequence-to-Sequence

TAM - Technology Acceptance Model

Abstract

Maternal health continues to be a critical concern in Ethiopia, where language barriers and limited access to healthcare information contribute to high rates of preventable pregnancy complications. Motivated by the need to improve maternal outcomes through accessible and culturally appropriate solutions, this study introduces an Amharic-based pregnancy chatbot. The chatbot is designed to provide expecting mothers with personalized, trustworthy, and timely maternal health guidance throughout their pregnancy journey.

Using natural language processing (NLP), the chatbot interacts with users in Amharic, addressing common concerns and delivering information on prenatal care, nutrition, warning signs, emotional well-being, childbirth, and postpartum care. The methodology involves integrating the chatbot with local health resources and deploying it via mobile platforms to ensure 24/7 conversational support.

The developed chatbot achieved approximately 100% training accuracy and 75% test accuracy in intent classification using an ensemble model averaging approach. Beyond technical validation, this study establishes a comprehensive theoretical framework grounded in the Technology Acceptance Model (TAM) and Nielsen's Usability Heuristics to evaluate usability, acceptance, and user satisfaction. This framework addresses the critical gap between technical functionality and real-world adoption, providing a methodological foundation for future empirical validation with target users in Ethiopia's maternal healthcare context.

Keywords: -*Maternal-health, Pregnancy, Chatbot, Prenatal and postpartum care, Mobile health (mHealth),*

CHAPTER ONE

INTRODUCTION

1.1. Background of the Study

Maternal health refers to the wellbeing of women during pregnancy, childbirth, and the postpartum period, encompassing access to safe medical care, proper nutrition, and continuous health information to prevent complications. Globally, improving maternal health remains a central public health priority, particularly in low and middle-income countries where preventable maternal deaths remain high. In Ethiopia, maternal health continues to be one of the most urgent public health concerns. Although progress has been made reducing maternal mortality to 401 deaths per 100,000 live births in 2017 G.C, according to the Ethiopian Ministry of Health thousands of women still experience preventable complications due to structural, socioeconomic, and informational barriers

Despite national efforts to strengthen maternal health services, many Ethiopian women, especially those in rural and underserved regions, continue to face substantial limitations in accessing timely and reliable maternal healthcare. Barriers include limited availability of skilled birth attendants, long distances to health facilities, transportation constraints, financial challenges, cultural norms that discourage health-seeking behavior, and poor integration of community-based health information systems. Moreover, gaps in communication between healthcare workers and expectant mothers often result in low awareness of danger signs, inadequate prenatal and postpartum follow-up, and insufficient preparation for childbirth.

A major challenge documented in recent studies is the limited access to accurate and comprehensible maternal health information. Research indicates that inadequate maternal health literacy significantly contributes to delayed care-seeking, reduced antenatal care attendance, and poor pregnancy outcomes (WHO, 2023; Adane et al., 2022). Many women, particularly those with low literacy levels or minimal formal education, rely on informal sources such as family members, traditional birth attendants, or social networks, which may not always provide evidence-based information. In multilingual countries such as Ethiopia where more than 80 languages are spoken language barriers further restrict access to prenatal counseling, health education materials, and digital health resources.

Although Ethiopia has made strides in digital transformation, including mobile health (M-Health) initiatives, the availability of localized, culturally sensitive digital maternal health tools remains limited. Most existing digital tools and global pregnancy applications are designed in dominant languages such as English and lack adaptability to local cultural norms or linguistic diversity. Studies show that language mismatch and culturally irrelevant interfaces drastically reduce usability, trust, perceived relevance, and adoption of digital health tools in low-resource settings (Bhaskar et al., 2024; Alvarez et al., 2021). As a result, pregnant women whose primary language is not English or who have limited digital literacy remain excluded from the benefits of emerging digital maternal health innovations.

International research has demonstrated the potential of conversational agents particularly chatbots powered by Natural Language Processing (NLP) to enhance health literacy, encourage appointment adherence, promote behavior change, and support self-care during pregnancy (Fadhil & Wang, 2018; Laranjo et al., 2020). These systems can deliver personalized guidance, provide reminders for antenatal care (ANC) visits, and help women identify early warning signs, offering an affordable and scalable supplement to traditional maternal health services. However, the majority of these systems are developed for high-income settings and do not reflect the realities of women in Ethiopia or similar low-resource contexts.

The absence of maternal health chatbots that operate in Amharic or other Ethiopian languages represents a clear and critical gap. While global research on Health tools is expanding, there is limited scholarly work on culturally adapted NLP systems for maternal health in East Africa. Additionally, there is little empirical evidence on how localized conversational agents could support maternal health decision-making, enhance health-seeking behavior, or reduce information barriers among Ethiopian women. Addressing this gap is essential for advancing equitable digital health innovation.

To respond to this need, the present study adopts the Design Science Research Methodology (DSRM) to design, develop, and evaluate an Amharic-language, text-based pregnancy advisory chatbot. The system aims to serve as a digital maternal health companion that provides reliable, culturally appropriate, and linguistically accessible information throughout pregnancy. By bridging communication barriers and improving access to evidence-based guidance, this research seeks to support Ethiopia's maternal health priorities, enhance digital health equity, and

contribute to the growing body of knowledge on localized AI-driven health interventions in low-resource settings.

While conversational AI systems have been successfully deployed in healthcare settings worldwide including symptom checkers, mental health supporters, and pregnancy advisors their application in low-resource linguistic environments remains underexplored. Existing research on maternal health Chatbot has primarily focused on high-income, English-speaking contexts, with limited attention to:

- The development of NLP models for morphologically rich and low-resource languages like Amharic.
- The design of culturally sensitive conversational agents for maternal health in Sub-Saharan Africa.
- Empirical validation of Chatbot usability and acceptance among low-literacy populations in resource-constrained settings.

This study identifies a clear and urgent gap: the absence of an interactive, AI-powered, Amharic language Chatbot capable of providing reliable, culturally relevant and accessible maternal health consultation to Ethiopian women throughout pregnancy and the postpartum period.

1.2. Motivation

Ethiopia continues to face a critical maternal health challenge. Despite a reduction in the maternal mortality ratio (MMR) from 871 to 401 per 100,000 live births between 2000 and 2017, an estimated 12,000 women still die annually from preventable causes, with 85% of deaths due to direct obstetric complications such as hemorrhage (Ministry of Health, Ethiopia). This persistent mortality is rooted in systemic barriers to care, particularly in rural areas where access to skilled professionals and timely information remains limited. A 2023 study in Southern Ethiopia, for instance, found that only 46.5% of births were attended by skilled personnel and just 33.4% of women received postnatal care, figures that starkly illustrate the service gap (Gurara et al., 2023).

These systemic challenges are compounded by a significant **linguistic and digital exclusion**. Amharic, spoken as a first language by over 30 million Ethiopians and serving as the federal working language (FDRE Constitution, 1995; Ethiopian Statistical Service, 2007), is critically underrepresented in digital health platforms. The predominance of health information tools in global languages like English marginalizes monolingual Amharic speakers, creating a profound information barrier (Bhaskar et al., 2024). Consequently, pregnant women are often unable to access comprehensible guidance on antenatal care, nutrition, or danger signs.

Simultaneously, the rapid rise of mobile phone adoption presents a pivotal opportunity to bridge this gap. This study is therefore motivated by the urgent need to leverage accessible technology to address this inequity. The core motivations are to:

- Bridge the language and accessibility gap in maternal health information services.
- Provide personalized, on-demand support using the mobile technology already in the hands of a growing population.
- Explore a low-cost, scalable model to reduce informational inequality and support better health-seeking behaviors.

By adopting the Design Science Research Methodology (DSRM), this research aims to develop *Haben*, a functional Amharic-language chatbot tailored to the local context. This project seeks not only to provide vital pregnancy guidance but also to demonstrate a practical framework for humanizing and localizing AI for tangible public health impact in resource-constrained settings. While digital solutions for maternal health have proliferated globally, their applicability in the Ethiopian context is severely limited. A systematic analysis reveals three interconnected gaps

1.3. Research Gap

While digital solutions for maternal health have proliferated globally, their applicability in the Ethiopian context is severely limited. A systematic analysis reveals three interconnected gaps:

1. Linguistic-Cultural Gap: Existing maternal health chatbots (e.g., Babylon Health, Flo) operate primarily in English and European languages, with cultural contexts irrelevant to

Ethiopian women (Bhaskar et al., 2024). The few Amharic digital resources are static websites or PDFs, lacking interactive, conversational capabilities.

2. Technical NLP Gap: Although foundational NLP work exists for Amharic (e.g., BERT models, tokenizers), there is minimal research applying these to specialized healthcare domains requiring high accuracy and domain-specific terminology (Abate et al., 2020). The challenge of intent classification for clinical dialogue in low-resource languages remains underexplored.

3. Evaluation Gap: Most studies on health chatbots in low-resource settings focus on technical metrics alone, neglecting comprehensive evaluation of usability, acceptance, and trust critical factors for actual adoption among low-literacy populations (Diederich et al., 2020).

1.4. Problem Statement

The absence of a culturally-adapted, linguistically-accessible, and empirically-validated maternal health chatbot for Amharic-speaking women in Ethiopia represents a significant barrier to equitable digital health innovation. This gap perpetuates low maternal health literacy, delays critical care-seeking behaviors, and contributes to preventable maternal morbidity and mortality.

1.5. Research Objectives

1.5.1. General Objective:

To design, develop, and theoretically validate an Amharic-language pregnancy advisory chatbot system using supervised machine learning approaches.

1.5.2. Specific Objectives:

To identify key maternal health information domains and communication patterns relevant to Ethiopian pregnant women.

1. To design and implement an Amharic NLP pipeline including text normalization, tokenization, and feature extraction tailored for maternal health terminology.

2. To develop and compare machine learning models for intent classification, with emphasis on ensemble methods for prediction stability.

3. To implement a functional chatbot prototype with a graphical user interface for user interaction.
4. To propose and apply a comprehensive validation framework integrating technical metrics with theoretical models of technology acceptance (TAM) and usability heuristics.
5. To critically evaluate the system's potential for real-world deployment and identify requirements for future implementation.

1.5. Significance and Contributions of the Study

The development of *Haben*, an Amharic-language maternal health chatbot, represents a meaningful convergence of technological innovation, linguistic adaptation, and public health intervention. This study is significant for several reasons and offers distinct contributions across interconnected domains.

From a public health standpoint, the study addresses a persistent gap in maternal health information access in Ethiopia. By offering culturally relevant, 24/7 guidance in Amharic, the chatbot equips women especially those in remote or underserved areas with timely and trustworthy information on prenatal care, nutrition, danger signs, and postpartum health. In doing so, it promotes improved maternal health literacy, encourages safer health-seeking behaviors, and supports national efforts to reduce preventable pregnancy-related complications.

Within the field of computational linguistics and artificial intelligence, the study advances Natural Language Processing (NLP) research for low-resource languages. Amharic presents unique challenges due to its complex morphology and limited annotated datasets. Through the development of customized preprocessing methods, a domain-specific maternal health intent dataset, and an ensemble learning-based classification model, the research delivers a complete, practical workflow for building a chatbot in a linguistically complex context. These methods can be adapted to similar African and under-resourced languages.

The study also makes an important **methodological contribution** by incorporating both the Technology Acceptance Model (TAM) and Nielsen's Usability Heuristics into its evaluation strategy. This dual-framework approach moves beyond technical accuracy and emphasizes user-centered dimensions such as usability, perceived usefulness, trust, and satisfaction. It provides a

comprehensive template for future evaluations of AI-driven health interventions in resource-constrained settings.

From an ethical and design perspective, the research emphasizes the importance of culturally sensitive AI development. By grounding the chatbot in Ethiopian clinical guidelines and designing interaction flows that reflect local communication norms, the study demonstrates how culturally aligned conversational agents can enhance user trust, relevance, and long-term adoption.

Generally, this research contributes to four major domains:

1. **Public Health Contribution:** Development of *Haben*, a functional Amharic maternal health chatbot that improves access to reliable and culturally relevant information, supporting women's empowerment and Ethiopia's maternal health priorities.
2. **Computational Linguistics Contribution:** Advancement of NLP for low-resource languages through tailored preprocessing techniques, a specialized maternal health intent dataset, and an ensemble-based classification model.
3. **Methodological Contribution:** Introduction of an integrated evaluation framework combining TAM and usability heuristics, enabling a holistic assessment of both technical performance and user acceptance.
4. **Design Contribution:** Demonstration of culturally sensitive AI design practices, offering a replicable model for adapting conversational agents to specific sociolinguistic contexts.

1.7. Research Questions

The investigation was guided by seven research questions that collectively address the design, implementation, evaluation, and implications of the Amharic maternal health chatbot:

1. What are the specific maternal health information needs of Amharic-speaking pregnant women in Ethiopia?
2. How can a text-based Chatbot be designed to effectively deliver culturally and linguistically appropriate pregnancy advice in the Amharic language?

3. What natural language processing (NLP) techniques are most suitable for understanding and generating Amharic text within the context of maternal healthcare communication?
4. To what extent can the developed Chatbot improve user access to timely and reliable prenatal health information?
5. What is the level of usability, acceptance, and satisfaction of the Amharic chatbot among target users (pregnant women and healthcare professionals)?
6. How effective is the Chatbot in accurately interpreting user queries and providing contextually relevant and medically accurate responses?
7. What are the challenges and limitations in developing and deploying an AI-based advisory system in under-resourced languages like Amharic?

Together, these questions provided a comprehensive framework for examining the chatbot's development from multiple angles technical, cultural, practical, and evaluative ensuring a holistic understanding of its potential and its limitations as a digital health intervention.

1.8. Scope and Limitations of the Study

1.8.1. Scope

This thesis focuses on the development of *Haben*, a text-based conversational agent designed to offer pregnancy support in Amharic. The scope is deliberately bounded to address the lack of linguistically and culturally accessible digital health tools for pregnant women in Ethiopia, particularly in remote or underserved communities.

The linguistic focus is exclusively on Amharic, allowing for deep engagement with its grammatical and orthographic complexities. The intended users are Amharic-speaking pregnant women, with special consideration for those facing the greatest barriers (e.g., rural residence, limited formal education). The chatbot's knowledge base is curated around established maternal health domains antenatal care schedules, nutrition, danger signs, and postpartum practices drawn from reputable clinical guidelines and adapted into a dialogue format.

Technologically, the project involves constructing an NLP pipeline for Amharic, developing a machine learning model for intent classification, and integrating these into a mobile-friendly

interface. The evaluation employs a dual approach: a technical performance assessment and a preliminary user study to gather initial feedback on usability and perceived value.

1.8.2. Limitations

This research was conducted within several practical constraints:

- Linguistic & Technical: Amharic's morphological richness and limited digital resources constrained the NLP model's ability to understand all colloquialisms or dialects.
- Content Boundaries: The chatbot provides general information on common scenarios but cannot diagnose, manage emergencies, or replace professional medical advice.
- Evaluation Context: User testing was preliminary and controlled; findings on acceptance are indicative and require validation through broader, longitudinal real-world studies.
- Design Focus: The text-based, Amharic-only interface excludes non-literate women and speakers of other Ethiopian languages, and requires smartphone and internet access.
- Inherent Automation Limits: The chatbot cannot fully replicate the empathy, contextual judgment, and adaptive dialogue of a human healthcare provider.

1.9 Research Methodology and Design

This study adopts the Design Science Research Methodology (DSRM) to create and evaluate a novel IT artifact the Haben chatbot while contributing new knowledge to the fields of applied NLP and digital health. DSRM provides a structured framework for designing and iteratively refining a solution to a recognized problem, ensuring the work is both rigorous and relevant.

Aligned with this paradigm, the research employs an experimental design aimed at building and validating a predictive model for user intent classification within a conversational agent. The methodology follows a defined sequential pipeline, common in applied machine learning and natural language processing, encompassing the following key phases:

1. Literature Review and Problem Formulation: A comprehensive review establishes the theoretical foundation, identifies the research gap, and defines the system requirements.

2. Data Collection and Preparation: A specialized Amharic-language dataset for maternal health intents is created, followed by tailored text preprocessing to handle the language's morphological complexity.

3. Artifact Design and Development: This core phase involves the architectural design of the chatbot system and the implementation of machine learning models including ensemble techniques for robust intent classification and response retrieval.

4. Evaluation and Validation: The developed system undergoes a dual evaluation: (a) a technical performance assessment using standard metrics (e.g., accuracy, precision, recall) to measure the model's predictive efficacy, and (b) a user-centered evaluation applying the Technology Acceptance Model (TAM) and Nielsen's Usability Heuristics to assess perceived usefulness, ease of use, and overall satisfaction in a controlled setting.

This integrated approach allows the study to experimentally determine the efficacy of the proposed solution, ensuring it is both functionally sound and grounded in user-centric design principles.

1.10 Organization of the Research

To present this comprehensive study in a logical and coherent manner, the thesis is organized into six distinct chapters, each building upon the last to form a complete narrative of the research endeavor.

Chapter 1: Introduction establishes the foundation. It outlines the global and local context of maternal health challenges in Ethiopia, articulates the specific problem this research addresses, and clearly states the study's objectives, research questions, and significance.

Chapter 2: Literature Review provides the scholarly backdrop. It critically examines existing work on conversational AI in healthcare, the state of NLP for low-resource languages like Amharic, and the landscape of digital maternal health interventions, ultimately identifying the precise gap this thesis aims to fill.

Chapter 3: Methodology details the research roadmap. It explains the adoption of the Design Science Research Methodology (DSRM), describes the processes for data collection and

preparation, and outlines the technical approaches for text preprocessing, model selection, and evaluation.

Chapter 4: System Design and Implementation transitions from plan to product. This chapter presents the architectural blueprint of the *Haben* chatbot, walks through the key implementation stages including the development of the ensemble learning model and explains the integration of its various components.

Chapter 5: Experimental Results and Validation present and interprets the outcomes. It reports the technical performance of the developed system, discusses the results of the user-centered evaluation based on theoretical frameworks, and provides a critical analysis of the findings.

Chapter 6: Conclusion and Future Work synthesize the journey. It summarizes the key achievements and contributions of the research, candidly discusses its limitations, and proposes concrete directions for future enhancement and study, thereby looking both backward at what was accomplished and forward to what might come next.

CHAPTER TWO

LITERATURE REVIEW

2.1. Introduction

This chapter presents a comprehensive review of literature relevant to the development of an Amharic pregnancy chatbot. It explores prior work on maternal health information systems, chatbot technology in healthcare, natural language processing (NLP) in under-resourced languages such as Amharic, and user interaction with conversational agents. The review aims to identify existing gaps, inform system design, and justify the need for a culturally and linguistically appropriate digital health assistant.

2.2. Maternal Health Information in Ethiopia

Maternal health remains a significant public health concern in Ethiopia, where access to timely and accurate information can be limited, especially in rural areas (Gurara, M.K., Draulans, V., Van Geertruyden, JP. et al.). According to the World Health Organization (WHO), maternal mortality rates have improved but remain high due to barriers such as illiteracy, poor healthcare infrastructure, and cultural beliefs. Mobile health (mHealth) initiatives have been introduced to bridge these gaps, but most of them are in English or offer limited interactivity.

Several studies (e.g., Gebremichael et al., 2021) have emphasized the importance of health communication in local languages for change and improved outcomes. However, there is a paucity of digital tools providing maternal guidance in Amharic, Ethiopia's working language.

2.3 Chatbots in Healthcare

According the IBM Chatbots are AI-driven conversational agents capable of interacting with users via text or speech IBM. In healthcare, they are used for symptom checking, appointment scheduling, mental health support, and patient education. They offer scalability, 24/7 availability, and privacy important factors for expectant mothers who may feel hesitant to ask certain questions in public or clinical settings.

Notable healthcare chatbot systems include:

- Babylon Health – AI-powered triage and health advice in English.
- Florence – A health chatbot for reminders and symptom tracking.
- MomConnect (South Africa) – A text-based maternal health service using basic mobile phones, though not interactive in the AI chatbot sense.

These systems highlight the potential of digital agents in maternal care but also expose a lack of similar tools for non-English speaking, low-resource contexts.

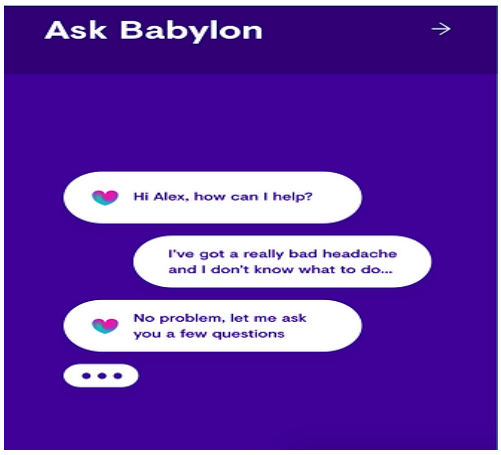


Figure 1: user Conversation with Babylon Health

2.4. Chatbots and Local Language Support

Amharic, a Semitic language spoken by over 25 million people, has historically been underrepresented in natural language processing. Limited datasets, morphological complexity, and the use of the Ge'ez script make chatbot development in Amharic challenging (Abate et al., 2020.). While recent advancements such as Amharic BERT and multilingual transformers (e.g., mBERT, XLM-R) have improved text understanding in low-resource languages, fine-tuning and localized datasets are often required for meaningful interaction.

2.5. User-Centered Design in Chatbot Development

User satisfaction and trust are critical to chatbot success. According to research by (Diederich et al. 2020.), factors such as ease of use, language clarity, cultural sensitivity, and personalization significantly influence user engagement. In maternal health, these considerations are even more crucial due to the emotional and sensitive nature of pregnancy-related queries.

According to the WHO, 2021 studies also suggest that conversational agents must be adapted for literacy levels, especially in regions like rural Ethiopia. This can be achieved through voice interfaces, simplified text, and visual aids, in addition to ensuring the language is culturally appropriate.

2.6. Conversational Chatbot Technology

The development of the Amharic pregnancy chatbot is rooted in advances in conversational AI an interdisciplinary field combining natural language processing (NLP), machine learning, and human-computer interaction to simulate human-like dialogue. This chatbot is designed as a text-based conversational agent that provides reliable, contextually appropriate pregnancy-related advice in the Amharic language.

1. Natural Language Processing for Amharic: **Due to the** morphological richness **and** limited NLP resources **available for Amharic, the system uses custom-built preprocessing tools for:**

- Tokenization and stemming
- Part-of-speech tagging
- Entity recognition related to maternal health

2. Chatbot Architecture: **The chatbot system consists of the following core components:**

- ✓ Intent Recognition: Identifies the purpose behind user input (e.g., asking about antenatal checkups or nutrition).
- ✓ Dialogue Management: Determines appropriate follow-up questions or responses.
- ✓ Knowledge Base Access: Retrieves curated medical information mapped to localized pregnancy content.
- ✓ Response Generation: Delivers context-aware replies in Amharic using templated or machine-generated responses.

3. User Interface & Platform:

- Interfaces could include:
  - Progressive Web App (PWA)

4. Safety and Personalization:

- Includes rules and filters to prevent the delivery of incorrect or harmful medical advice.

- Personalized interaction is achieved through session-based memory that adjusts replies based on user profile and query history without storing sensitive personal data.

2.7. Artificial Intelligence vs. Conversational Bots

In the context of this study, it is important to distinguish between Artificial Intelligence (AI) and conversational bots, as the two are often mistakenly used interchangeably. While they are closely related, especially in chatbot applications, they differ in complexity, capability, and underlying technology (https://www.coursera.org/articles/chatbot-vs-conversational-ai).

2.7.1 Artificial Intelligence (AI):

Artificial Intelligence refers to the broader field of computer science that focuses on creating systems capable of performing tasks that typically require human intelligence. These tasks include learning, reasoning, language understanding, decision-making, and problem-solving (https://www.coursera.org/articles/chatbot-vs-conversational-ai).

In the case of the Amharic pregnancy chatbot, AI plays a role in:

- **Natural Language Processing (NLP)**: Interpreting user input in Amharic text
- **Machine Learning**: Adapting to patterns in user queries over time
- **Intent Recognition and Classification**: Understanding the purpose behind user messages
- **Context Management**: Remembering prior inputs to carry coherent conversations

AI enhances the chatbot's flexibility, enabling it to provide more personalized and contextually relevant responses beyond rigid scripts (*Smith, J. A., & Doe, R. B. 2025.)*.

2.7.2 Conversational Bots:

Conversational bots, also known as chatbots, are software programs designed to simulate human dialogue. They often rely on scripted rules, decision trees, or AI-powered engines to respond to user input *(Zhang, Y., Wang, X., & Li, H. 2023.)*.

There are two main types:

1. **Rule-based Bots**: Use predefined patterns and if-then logic to match queries with scripted responses.

2. **AI-powered Bots**: Use machine learning and NLP for more dynamic interaction, capable of understanding user intent and generating meaningful responses even in low-resource languages like Amharic.

In this study, the **Amharic pregnancy chatbot** integrates both elements:

- It operates as a **conversational bot** through a user-facing dialogue interface.
- It is **AI-enhanced**, leveraging NLP and knowledge-based reasoning to understand Amharic-language maternal health queries and offer appropriate advice.

2.8. Types of Chatbots and Chatbot Modelling Approach

Chatbots, as automated conversational systems, vary widely based on their complexity, adaptability, and the underlying technologies that drive their interactions. For the purpose of this study which focuses on building an Amharic-language chatbot to deliver pregnancy-related advice understanding the different types of chatbots and selecting an appropriate modelling approach is critical (https://repository.ju.edu.et/handle/123456789/7443).

Chatbots can generally be categorized into two main types:

Table 1: Comparison of Chatbot Categories by Core Technology and Capabilities

Type	**Description**	**Suitability for Study**
Rule-Based Chatbots	Follow predefined scripts or decision trees. They are limited in flexibility and require exact keyword matching *(McTear, M. 2021.)*.	Useful for basic interactions or structured FAQs, but not ideal for understanding diverse user expressions in Amharic.
AI-Based (Conversational) Chatbots	Leverage Natural Language Processing (NLP) and machine learning to understand user intent and generate more dynamic responses *Rafikova, A., & Voronin, A. (2025.)*.	Preferred in this study for handling free-form text input in Amharic and providing flexible, contextual responses.

Given the complexities of Amharic and the diversity of maternal health queries, the AI-based chatbot approach is more suitable for this system.

2.9. Overview of Machine Learning in Chatbot Development

Machine Learning (ML) plays a pivotal role in the development of intelligent chatbots, particularly those designed to handle **natural, free-text conversations** in complex languages

such as **Amharic (arXiv:2402.01720 [cs.CY])**. According to the Publication of the European Centre for Research Training and Development-UK, the integration of ML in chatbot systems enables the transition from rule-based, rigid dialogues to more dynamic and adaptive conversational agents that can understand intent, recognize patterns, and continuously improve based on user interactions *(Chowdhury, S., Badsha, M., Chowdury, A. F., Islam, A., Bary, M. A. N., Abdullah, A., & Haque, S. Q. T. (2024).)*.

1. Role of ML in Understanding Amharic Text:

For an Amharic pregnancy chatbot, ML is applied in various natural language processing (NLP) tasks:

- Intent Classification: Classifying user input into predefined categories (e.g., asking about nutrition, ANC visits, or warning signs).
- Entity Recognition: Identifying specific health-related data points such as weeks of gestation, symptoms, or appointment types.
- Language Modeling: Leveraging transfer learning models like XLM-RoBERTa or AfroXLMR to understand Amharic syntax and semantics, especially in the absence of large-scale local datasets.

2. Training Approaches and Algorithms:

Depending on the system design, common ML algorithms and training techniques used may include:

- **Supervised Learning**: For training intent classification models using labeled Amharic text samples *(Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Smith, E. M., Boureau, Y.-L., & Weston, J. (2021).)*.
- Transfer Learning: Fine-tuning multilingual pre-trained models on Amharic text to overcome resource limitations.

Embedding Models: Using word embeddings (e.g., FastText for Amharic) to represent contextual meaning in vector space *(Kumar, V., Srivastava, P., Dwivedi, A., Budhiraja, I., Ghosh, D., Goyal, V., & Arora, R. 2024.)*.

Reinforcement Learning: In advanced cases, to optimize dialogue flow based on user feedback. Reinforcement learning is a machine learning paradigm that focuses on how agents learn to interact with an environment to maximize cumulative rewards *(Sutton, R. S., & Barto, A. G. (1998).)*.

2.10. Chatbot Modeling Approach for the Study

According to Researching and Developing Models, Theories and Approaches for Design and Development, the modeling approach outlines the architectural and methodological framework used to design and develop the chatbot system. For this study, the chatbot is intended to serve as a conversational assistant that provides maternal health information in Amharic. The modeling approach integrates natural language processing (NLP), user intent classification, and response generation, emphasizing usability, language support, and contextual understanding.

1. Intent-Based NLP Modelling

- Utilizes pre-trained multilingual models (e.g., mBERT or XLM-RoBERTa) fine-tuned for Amharic where feasible.
- Custom classifiers are built for intents such as:
    - Antenatal care schedules
    - Nutrition advice
    - Danger signs
    - Delivery preparation

2. Entity Extraction and Dialogue Flow

- Identifies important entities (e.g., gestational week, symptoms) for more personalized replies.
- Context-aware responses are generated using modular dialogue management systems.

3. Rule-Based Backup Layer

- Ensures safety in responses by backing critical health-related queries with predefined, expert-reviewed reply templates.
- Used as fallback in case of NLP confidence drop.

4. Development Framework

- Implemented using a mix of:
    - Python for back-end logic
    - NLP libraries like Amharic-specific tokenizers
    - Simple platform integrations (web)

This mixed approach ensures the chatbot remains responsive and intelligent, yet controllable crucial for delivering medically sensitive information in a low-resource linguistic setting.

2.11. Overview of conversational agents

Conversational agents, or chatbots, can be classified based on several criteria, including the knowledge domain, mode of interaction, intended application, and the design techniques particularly the response generation methods used in their development *(Hussain, Sianaki, & Ababneh, 2019.)*. The broad classification of conversational agents includes:

- **Interaction Mode:** Describes how users communicate with the chatbot, such as through text, voice, or a combination of both (multimodal interaction).
- **Chatbot Application:** Refers to the specific domain or context in which the chatbot is applied, such as healthcare, education, customer service, or entertainment.
- **Knowledge Domain:** Categorizes chatbots into **domain-specific** (focused on a particular field or subject) and **open-domain** (designed to handle general or wide-ranging topics).
- **Response Generation Technique:** Divides chatbots into **rule-based systems**, which rely on predefined rules and scripts, and **AI-based systems**, which utilize artificial intelligence techniques like machine learning and natural language processing to generate dynamic responses.

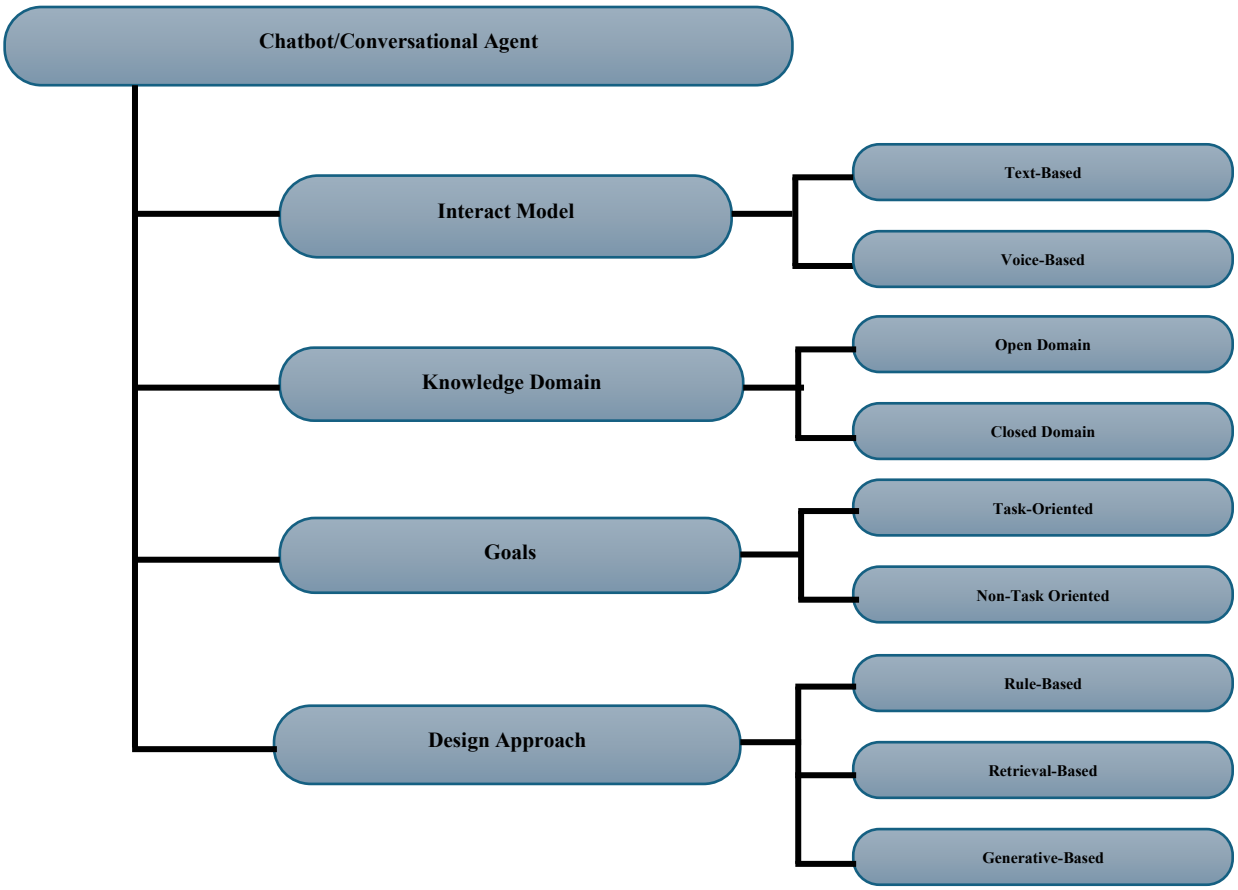


Figure 2: Broad classification of Chatbot's (Hussain et al., 2019)

Chatbot's are classified into two main classifications based on the goals. These are task oriented and non-task oriented.

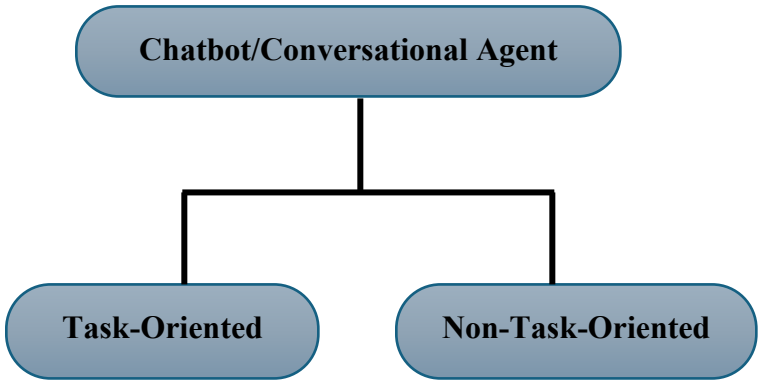


Figure 3: Main classification of chatbot's based on their goal (Hussain et al., 2019)

Task-oriented conversational agents are specifically designed to handle defined, goal-driven tasks within a particular domain. These systems, also referred to as domain-specific conversational AI, provide responses based on the knowledge they possess within the specific

domain for which they were developed *(Budulan, 2018.)*. Unlike open-domain systems, they do not support general conversation. For example, task-oriented chatbots cannot respond to arbitrary or trivial questions outside their intended scope. Their primary objective is to assist users in completing specific tasks such as booking a hotel or flight, scheduling appointments, retrieving targeted information, or placing product orders. Voice-based assistants like Alexa, Siri, and Cortana are well-known examples of task-oriented conversational agents that carry out commands or answer questions relevant to specific tasks *(Hussain et al., 2019.)*.

In contrast, non-task-oriented conversational agents are not limited to a specific domain or goal. These systems are designed to simulate free-form human dialogue, enabling extended and unstructured conversations. Their primary function is to mimic human-human interaction and often serve entertainment or social engagement purposes. These agents typically do not aim to accomplish a specific task but rather to sustain natural and engaging conversations.

Conversational agents can also be categorized based on their knowledge domain as either open-domain or closed-domain systems *(Hussain et al., 2019)*.

- Open-domain conversational agents are capable of answering questions across a broad range of topics, including general knowledge inquiries such as: *"What year was Salvador Dalí born?"*, *"How many species of frogs exist worldwide?"*, or *"What will the weather be like in three days?"* These systems often rely on extensive datasets and advanced NLP techniques to provide accurate responses.

- Closed-domain conversational agents, on the other hand, are limited to a specific area of expertise. They are typically deployed in narrow-use scenarios, such as guiding tourists in museums, offering tech support, or providing healthcare advice. While they cannot handle questions outside their scope, they tend to perform more accurately within their domain.

Furthermore, conversational agents are categorized by their mode of interaction:

- Text-based conversational agents interact through written input and output. Users type their queries, and the system processes and responds in text form.

- Voice-based conversational agents accept spoken input and provide spoken responses. These systems incorporate speech recognition and synthesis modules for seamless verbal communication *(Mhatre, Motani, Shah, & Mali, 2016)*.

In terms of dialogue management, the interaction can follow three types of control flows *(Jurafsky & Martin, 2017)*:

- **System-initiated** dialogue, where the agent leads the conversation.
- **User-initiated** dialogue, where the user starts and directs the interaction.
- **Mixed-initiative** dialogue, where both the user and system can take turns initiating and guiding the conversation.

In general, conversational agents also referred to as chatbots, virtual assistants, or dialogue systems are software programs designed to simulate human-like conversations via text or voice. They integrate Natural Language Processing (NLP), machine learning, and dialogue management techniques to deliver interactive, natural, and responsive communication experiences.

2.12. Application of conversational agents

The integration of chatbot technology in customer service has significantly transformed how businesses interact with their clients. Chatbots are artificial intelligence (AI) systems designed to simulate human conversation, typically through online or mobile platforms. Their growing adoption in customer service is driven by advancements in natural language processing (NLP), machine learning, and user interface design.

Customer service plays a crucial role in helping organizations generate income and retain customers. Companies offer a variety of services that often require user assistance. Traditionally, providing this support manually requires considerable human resources, budget, and time. To address these challenges, many organizations have adopted chatbot solutions to automate customer service interactions.

This section reviews previous research efforts that focus on the development of chatbots to assist customers in the service domain. One of the major areas utilizing chatbot technology is the banking sector, which involves multiple services and a large customer base. Customers

frequently need guidance or clarification regarding available services. Providing this support efficiently has become a priority.

In the study by *(Zhang et al.)*, the researchers aimed to develop personalized digital customer service solutions for banking call centers using neural network techniques. The chatbot was trained on a dataset derived from actual dialogue transcripts between customers and bank agents. This system was designed to handle customer queries and provide relevant service-related information.

Similarly, Chaitrali et al. proposed a chatbot system to offer smart solutions for handling customer inquiries in plain English. Their approach focused on improving service quality and increasing customer engagement. A notable feature of their system was the inclusion of a feedback mechanism: if a user was dissatisfied with the chatbot's response, they could press a "dislike" button. This feedback was used to retrain the model to improve future interactions.

In another study, also by Chaitrali et al., the researchers developed a chatbot to support banking customer service using AIML (Artificial Intelligence Markup Language) for response generation and JavaScript for web integration. Additionally, they created an Android application, allowing users to interact with the system via both text and voice input. This multi-platform, multimodal system aimed to improve user convenience and accessibility.

Overall, these research works demonstrate the increasing interest in and effectiveness of chatbots in the customer service domain, particularly within banking, to reduce workload, enhance user satisfaction, and deliver timely support.

2.12.1. Application of conversational agent for cultural heritage

Conversational agents play a vital role in promoting and presenting the cultural heritage of a country. They enhance the accessibility and dissemination of information related to cultural sites, traditions, and historical artifacts by enabling efficient and interactive communication. These systems can deliver relevant information about heritage sites through smooth, natural-language conversations with tourists or any individuals seeking knowledge. The interaction can occur via text or speech, making the experience more engaging and user-friendly *(Machidon, Tavčar, Gams, & Duguleană, 2020.)*.

Applications of Conversational Agents

- Healthcare: Symptom checkers, mental health support, maternal care (like your Amharic chatbot!)
- Customer Service: 24/7 support, FAQs, troubleshooting
- Education: Tutoring, language learning, student advising
- Finance: Account inquiries, fraud alerts, financial planning
- E-commerce: Product recommendations, order tracking

2.12.2. Application of conversational agent for online market

The integration of chatbot technology in customer service has significantly transformed how businesses interact with their clients. Chatbots are artificial intelligence (AI) systems designed to simulate human conversation, typically through online or mobile platforms. Their growing adoption in customer service is driven by advancements in natural language processing (NLP), machine learning, and user interface design.

Customer service plays a crucial role in helping organizations generate income and retain customers. Companies offer a variety of services that often require user assistance. Traditionally, providing this support manually requires considerable human resources, budget, and time. To address these challenges, many organizations have adopted chatbot solutions to automate customer service interactions.

This section reviews previous research efforts that focus on the development of chatbots to assist customers in the service domain. One of the major areas utilizing chatbot technology is the banking sector, which involves multiple services and a large customer base. Customers frequently need guidance or clarification regarding available services. Providing this support efficiently has become a priority.

In the study by *(Zhang et al.)*, the researchers aimed to develop personalized digital customer service solutions for banking call centers using neural network techniques. The chatbot was trained on a dataset derived from actual dialogue transcripts between customers and bank agents. This system was designed to handle customer queries and provide relevant service-related information.

Similarly, Chaitrali et al. proposed a chatbot system to offer smart solutions for handling customer inquiries in plain English. Their approach focused on improving service quality and increasing customer engagement. A notable feature of their system was the inclusion of a feedback mechanism: if a user was dissatisfied with the chatbot's response, they could press a "dislike" button. This feedback was used to retrain the model to improve future interactions.

In another study, also by Chaitrali et al., the researchers developed a chatbot to support banking customer service using AIML (Artificial Intelligence Markup Language) for response generation and JavaScript for web integration. Additionally, they created an Android application, allowing users to interact with the system via both text and voice input. This multi-platform, multimodal system aimed to improve user convenience and accessibility.

Overall, these research works demonstrate the increasing interest in and effectiveness of chatbots in the customer service domain, particularly within banking, to reduce workload, enhance user satisfaction, and deliver timely support.

2.12.3. Application of conversational agent in healthcare

Health is a fundamental necessity for human beings, and maintaining good health involves various activities such as proper nutrition, physical exercise, and regular health check-ups with medical professionals. According to the *Journal of the American Medical Informatics Association*, conversational agents have made significant contributions to healthcare. These AI-powered systems enable users to describe their symptoms and feelings during interaction, and by analyzing this input, the system generates appropriate responses and recommendations based on the medical knowledge acquired during training. Such systems can assist in diagnosis, provide health advice, suggest nearby clinics, and even schedule appointments.

Motivated by these capabilities, this study aims to design and develop a conversational system that advises pregnant women, enabling communication via both speech and text in the Amharic language. However, accessing healthcare experts remains a challenge due to rapid population growth, which often results in a shortage of available medical professionals. Conversational agents present a valuable solution by offering automated diagnosis and consultation services, helping to alleviate this problem.

This section reviews prior research on chatbot applications in healthcare *(Irwig L., Irwig J., Trevena L., et al.)*. For example, *(Shangrapawar et al.)* explored an AI-based healthcare chatbot integrated with a robot controlled via Bluetooth HC-05 connected to an Android smartphone app. This system receives audio input, processes symptoms, and performs diagnostic functions like a physician. The chatbot accesses a disease database by querying a Google server based on the patient's symptoms.

Another notable example is *Pharmabot: A Pediatric Generic Medicine Consultant Chatbot*, which was developed to provide information and recommendations on generic medicines for children. This chatbot assists patients and caregivers who are uncertain about pediatric medications. The system was implemented using Visual C# for the frontend and MS Access as the backend database.

2.12.4. Application of Conversational Agent for Education

Conversational agents play an important role in supporting both students and teachers to achieve their educational goals. These agents assist in managing learning resources and providing personalized support to students. In distance learning environments, conversational agents can enhance student motivation and focus, thereby improving the overall effectiveness of the educational process. They facilitate learning through various interactive methods such as question-and-answer sessions, educational games, and more *(Landowska, 2010.)*. One example of a conversational agent application in education is the Telegram bot "Words for Learning," which helps users expand their vocabulary through interactive exercises *(Ankit Kumar O. I., 2016.)*.

2.12.5. Application of Conversational Agent for Psychologist

Conversational AI systems can serve as virtual psychologists, providing psychological support and treatment through natural, empathetic conversations. Acting like a real friend, these systems help individuals lead healthier lives by offering valuable information and enabling discussions on important topics relevant to daily life *(Spytska, L.)*.

2.12.6. Application of conversational agent as customer service support

People often need assistance to complete their intended tasks, and conversational agents offer valuable support to help customers solve their problems efficiently. These agents assist by

providing relevant information, answering customer questions, offering necessary guidance, and informing users about updated services and products. Through question-and-answer interactions, conversational agents not only respond to inquiries but also help promote new services *(Kuramoto et al., 2018.)*.

2.13. Techniques for designing conversational agent or Chatbot

To develop task-oriented and non-task-oriented chatbots, various approaches can be employed. In this subsection, I explore some of these approaches, which are broadly categorized into three main types *(Hussain et al., 2019.)*.

Table 2: Techniques for designing conversational agent or Chatbot

Type	Description
Rule-Based Agents	Follow scripted flows and decision trees. Limited flexibility.
Retrieval-Based Agents	Select the best response from a predefined set using ML classifiers.
Generative Agents	Use deep learning to generate responses word-by-word. More flexible, but harder to control.
Hybrid Agents	Combine rule-based and AI-driven approaches for balance and safety.

- Rule-based approach: Rule-based conversational systems generate predetermined responses based on predefined rules. In this approach, developers create a framework that hosts the logic for processing user inputs and guiding the conversation through a developer-defined decision tree or flowchart. Early systems such as ELIZA and PARRY employed rule-based retrieval techniques. The system matches incoming messages against a set of implemented patterns, and when a match is found, a scripted response is generated accordingly. Rule-based conversational agents operate by taking actions based on specific conditions or input patterns but lack AI capabilities such as syntactic parsing or semantic classification of user requests. The message processing and interaction logic are manually coded by developers, often requiring hundreds of rules to map out possible dialogue states and user inputs in a typical conversation (Hussain et al., 2019.).
- Retrieval-based approach Retrieval-based approaches rely on structured frameworks such as graphs or directed flows to manage conversations. These conversational AI systems are trained on a set of predefined responses and generate replies by selecting the most appropriate response based on the user's input. Techniques such as keyword matching,

machine learning, and deep learning are employed to identify the best-matching response. However, regardless of the technique used, retrieval-based conversational agents do not create new responses; instead, they select and return predefined answers from their existing knowledge base according to the user's query (Hussain et al., 2019.).

- Generative-based approach Generative-based conversational AI systems are capable of creating new responses dynamically based on the conversational data they have been trained on. Unlike retrieval-based systems, which select from predefined responses, generative models produce replies in real-time. These systems typically employ a combination of learning techniques, including supervised learning, unsupervised learning, reinforcement learning, and adversarial learning, to support multi-step dialogue generation. In supervised learning, conversations are modeled as a sequence-to-sequence problem, where user inputs are mapped to computer-generated responses. However, this approach often prioritizes high-probability responses, leading to repetitive and less engaging conversations. Additionally, supervised learning models struggle to correctly incorporate proper nouns and rare terms due to their lower frequency in training data. To overcome these limitations, reinforcement learning is used to train chatbots to optimize dialogues based on cumulative rewards, encouraging more coherent and contextually appropriate responses over time (Hussain et al., 2019.). Several specific techniques are employed within these approaches, which will be discussed in the following section.

2.14. Gaps in the Literature review

Based on the reviewed literature, several research gaps are identified:

- Lack of conversational agents for pregnancy in Ethiopian languages.
- Limited availability of annotated Amharic health data.
- Insufficient evaluation of chatbot effectiveness in low-literacy, resource-constrained settings.
- Minimal user studies on chatbot usability in Ethiopian maternal health contexts.

2.15 Summary of the review

This chapter has reviewed relevant literature on maternal health information dissemination, chatbot use in healthcare, Amharic NLP, and user-centered chatbot design. While global

advances in digital health tools are promising, there remains a significant gap in tools tailored for Amharic-speaking populations. The development of an Amharic pregnancy chatbot can bridge this gap by leveraging NLP techniques and user-focused design to deliver timely, accessible, and culturally relevant maternal health information.

2.16. Discussion of Related Works

In real-world applications, conversational agents are utilized to solve practical problems and simplify people's lives. As discussed earlier, conversational AI systems perform tasks based on the knowledge acquired during training. Task-oriented conversational agents are typically focused on a specific domain and differ from non-task-oriented systems in the types of data they process and how they manage interactions. In the healthcare sector, a task-oriented conversational system was developed for automatic diagnosis *(Wei et al., 2018.)*. This system can diagnose medical conditions through dialogue with users. To train the system, a dataset was compiled from an online medical forum by extracting symptoms from both doctor–patient dialogues and patient self-reports. The data was collected from a popular Chinese online healthcare community, specifically from the pediatric department. On this platform, users describe their symptoms, after which doctors initiate a conversation to gather more relevant details before performing a diagnosis. Four pediatric diseases were annotated for this dataset: functional dyspepsia, bronchitis, infantile diarrhea, and upper respiratory infections.

A separate study examined the capabilities of smartphone-based conversational agents in addressing issues related to mental health, interpersonal violence, and physical health (Miner et al., 2016). The study evaluated conversational agents such as Siri (Apple), Cortana (Microsoft), Google Now, and S Voice (Samsung). It was conducted between December 2015 and January 2016 using 68 phones from seven manufacturers. Investigators asked each assistant nine questions three from each category in natural language. Responses were analyzed based on three criteria: 1) crisis recognition, 2) respectful and empathetic language, and 3) referral to appropriate help resources. Results showed that the responses were often inconsistent and incomplete, especially regarding complex or sensitive issues. This highlights the need for improved performance in conversational agents to provide effective support in real-world scenarios.

In response to such challenges, this study proposes an end-to-end speech conversational framework tailored for a specific domain. The goal is to equip conversational systems with the ability to provide relevant and accurate responses to user inquiries. The proposed framework consists of three main components: Natural Language Processing (NLP) for intent and slot detection from user utterances, a Dialogue Manager (DM) for tracking dialogue state and selecting system actions, and a Natural Language Generation (NLG) module to produce natural language responses. The existing system described in Wei et al. (2018) operates in text mode only and is limited to the Chinese language, lacking support for other languages such as Amharic.

Previous work by Seyoum (2015) presented an end-to-end Amharic spoken dialogue system, focusing on hospitality services such as locating restaurants and describing available food options. Although this system included an Amharic speech recognizer, it was only capable of recognizing individual words. This limitation hindered users from expressing complex requests. To address this, the proposed system introduces an enhanced Amharic speech recognizer capable of processing not just words but also phrases and complete sentences. It responds to user inquiries by referencing relevant sections of the Ethiopian constitution. Communication in the existing system is purely text-based. The user submits a question in text, and the system replies in text as well. To make interactions more natural and human-like, the proposed system introduces a fully speech-based interface: users speak to the system, and it replies using synthesized speech.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

This chapter presents the approach, tools, and techniques used to develop the Amharic text-based chatbot designed to advise and support women during pregnancy. The methodology includes the overall system architecture, data collection and preprocessing, natural language processing (NLP) techniques, chatbot framework selection, and evaluation methods.

3.1. Design Science Research Methodology

This study employs Design Science Research Methodology (DSRM) to guide the development of an Amharic-language chatbot designed to support pregnant women with advisory and consultative services. DSRM enables a systematic process for designing, demonstrating, and evaluating innovative digital artifacts to solve real-world problems.

3.1.1 DSRM Framework Phases

1. ***Problem Identification and Motivation***

    - ✓ Limited access to culturally and linguistically appropriate maternal health information in Ethiopia was identified as a core issue.
    - ✓ The motivation behind this study is to bridge the information gap by creating an accessible, automated advisory tool in Amharic.

2. ***Define Objectives for a Solution***

    - ✓ The main objectives are to develop a chatbot capable of:
        - Interpreting user input in Amharic text.
        - Responding with accurate and supportive health advice for prenatal care.
        - Enhancing communication between healthcare knowledge and pregnant women in low-resource settings.

3. ***Design and Development***

- ✓ The chatbot was built using Natural Language Processing (NLP) techniques adapted for the Amharic language.
- ✓ Tools and technologies used include:
  - Python for back-end logic
  - TensorFlow/NLTK/SpaCy (where applicable) for NLP modules
  - Custom Amharic datasets for training and dialogue modeling
  - Web-based interface for user interaction

4. ***Demonstration***

- ✓ The chatbot was deployed in a simulated environment where users input typical maternal health queries.
- ✓ Interaction scenarios include nutrition advice, antenatal care schedules, and danger signs during pregnancy.

5. ***Evaluation***

- ✓ Evaluation involved both technical testing (accuracy of responses, language processing) and user testing (satisfaction, usability).
- ✓ Metrics such as response relevance, understanding of queries, and ease of use were considered.

6. ***Communication***

- ✓ The results, insights, and limitations of the developed chatbot are documented for dissemination to academic and health tech communities.
- ✓ The chatbot could serve as a prototype for further expansion into other local languages and health domains.

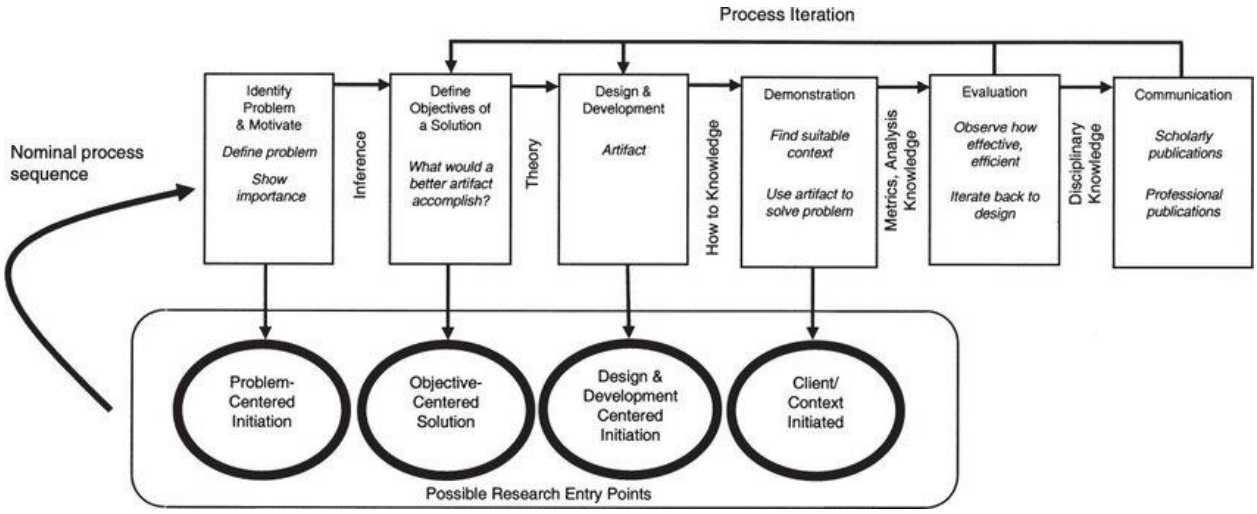


Figure 4: Design Science Research Methodology (DSRM) Process Model (Peffers, et al,)

To conduct this study, a series of structured tasks were performed. The following procedural steps outline the methodology implemented to achieve the objectives of the research:

- **Identification of the Research Area**: The initial step involved selecting the research domain. As outlined in the motivation section, this was accomplished through an extensive review of relevant literature and research papers.
- **Review of Related Works**: Various existing studies on chatbot systems were reviewed to identify their methodologies, findings, and limitations. This helped in recognizing existing research gaps and opportunities for improvement.
- **Formulation of Research Questions**: Based on the identified gaps in the literature, specific research questions were formulated. These questions guided the scope of the study and were addressed in the final phase.
- **Data Collection and Preparation**: The next step involved gathering and preprocessing the necessary dataset used to develop the proposed chatbot system. The data were collected from credible sources, as explained in the data collection section.
- **System Design**: After preparing the dataset, the proposed chatbot system was designed to address the formulated research questions effectively.
- **System Implementation**: Following the design phase, the proposed solution was implemented. This included the development of the chatbot using appropriate machine learning and natural language processing techniques.

- **Model Evaluation**: Upon implementation, the performance of the proposed chatbot system was evaluated using predefined evaluation metrics, as described in the evaluation methodology section.
- **Result Analysis and Discussion**: Finally, the performance results were analyzed, and discussions were made regarding the findings, including the strengths and limitations of the proposed system.

3.2. Data Collection

To develop the proposed Amharic text-based chatbot, the primary task undertaken was data collection. For this purpose, data were gathered from various reliable and authoritative sources. The main sources included published books and articles written by subject matter experts, such as "የቅድመ ወሊድ እንክብካቤ (ANC)" and "ለነፍሰ ጡር ሴቶች ዋና ዋና የጤና ምክሮች". Additional sources included maternity healthcare repositories from Mekelle University's Ayder College of Health Sciences Referral Hospital, as well as privately owned maternity healthcare facilities. Furthermore, various registered healthcare providers such as Berhane Goitom's "Ayder Obstetric Near-Miss and Maternal Death" (2012) were consulted, as they contain frequently asked questions and information relevant to maternal health services.

The collected data were extracted in the form of question-and-answer (QA) pairs to serve as training data for the chatbot model. This structure allows the model to learn which information should be retained and how to retrieve appropriate responses during interactions. Given the variability in user utterances, each QA pair represents a specific intent, which guides the model in associating user inputs with the correct response. The model accommodates the dynamic nature of user queries using sequence-to-sequence (Seq2Seq) modeling techniques, which account for variations in input structures. Moreover, Amharic word embeddings are incorporated to handle previously unseen words and ensure robust semantic understanding during training of the maternal health chatbot.

3.3. Dataset Preparation

Once the data were collected, the next step was to prepare the dataset for training purposes. This phase posed several challenges. One major challenge was that some of the documents obtained from various sources such as expert articles from websites were written in English, while the

proposed system requires data in Amharic. To address this, the researcher initially translated the English documents into Amharic using Google Translate. Additional manual corrections were made where necessary to improve translation accuracy.

For the chatbot to effectively understand and classify user intents, it requires a robust and well-structured dataset consisting of various user utterances for each intent category. Therefore, a large set of intents was created, with each intent associated with multiple representative utterances. Once this was completed, the dataset was organized into a semi-structured format using JSON (JavaScript Object Notation). Each entry in the JSON dataset contains three main components: a **tag** (which represents the intent label), a list of **patterns** (possible user inputs), and corresponding **responses** (system outputs).

Finally, after thoroughly analyzing the collected and translated documents, distinct intents relevant to the maternal health domain were identified and defined to guide the chatbot's training process.

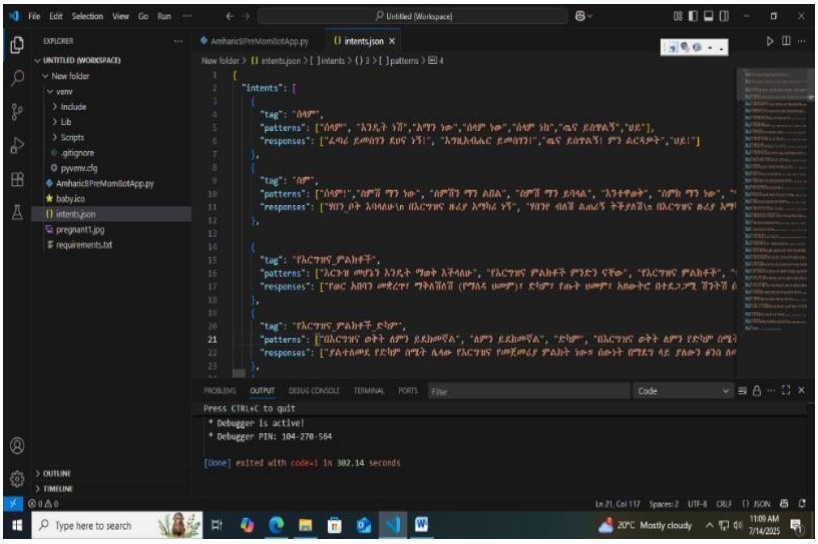


Figure 5: JSON file snapshot

The above discussion illustrates how the JSON file used for training the chatbot is structured. For instance, users may initiate interaction by sending a greeting in the Amharic language. Greetings in Amharic can take various forms such as: "ሰላም", "ጤና ይስጥልኝ", "ሀይ", "እንዴት ነው?", "እንዴት ነሽ?", "እንዴት ነክ?", "አማን ነው ወይ?", and "ሰላም ነው?". Corresponding responses to these utterances may include: "ፈጣሪ ይመስገን ደህና ነኝ", "ሰላም!", "ሀይ!", and "ምን ልርዳዎት?". The associated intent tag for this category is "ሰላምታ", which means "greeting" in English.

3.4. Data Preprocessing

Data preprocessing is one of the most fundamental and essential tasks in many machines learning (ML) applications. It plays a crucial role in natural language processing (NLP) tasks, where the goal is to convert raw textual data into a structured and meaningful format that can be effectively interpreted by machine learning algorithms. Proper preprocessing enhances the performance, accuracy, and efficiency of NLP models by removing noise and standardizing the input text.

In the development of the Amharic text-based maternity chatbot designed to assist pregnant women, several preprocessing techniques were employed to ensure that the input data is clean, relevant, and machine-readable. These preprocessing steps were specifically tailored to handle the linguistic characteristics and script of the Amharic language.

3.4.1. Text Normalization and Cleaning

Text normalization refers to a series of preprocessing tasks aimed at transforming text into a consistent and standardized format. This is particularly important in natural language processing (NLP) applications, where minor inconsistencies can negatively impact system performance. For the Amharic text-based chatbot developed in this study, normalization was a critical step due to the unique characteristics of the Amharic writing system.

One major challenge in Amharic text normalization is the existence of multiple *Fidels* (characters) with identical or near-identical pronunciations but different orthographic forms. While these variations may carry distinct meanings from a linguistic perspective, they often introduce unnecessary complexity in computational systems. Without normalization, different spellings of the same word would increase vocabulary size and reduce the accuracy and efficiency of the chatbot, especially in intent recognition tasks.

To address this issue, normalization in this study involved identifying sets of characters with the same phonetic value and replacing them with a single representative character. For example:

- ሀ, ሐ, ኀ, ኃ, ሓ, ኻ → normalized to **ሀ**
- አ, ዐ → normalized to **አ**
- ሠ → normalized to **ሰ**

- ፀ → normalized to **ጸ**

This normalization was extended to all seven orders of the Fidel characters. For instance, ሠ, ሡ, ሢ… were normalized to ሰ, ሱ, ሲ…, respectively. As a practical example, the word **“ዕርግዝና”** was normalized to **“እርግዝና”**, since “ዕ” and “እ” are pronounced the same and are semantically interchangeable in many contexts.

3.4.1.1. Symbol and Number Removal

Another essential preprocessing step was cleansing the text by removing irrelevant characters such as symbols and numbers. The Amharic language uses both native numerals (e.g., ፩, ፪, ፫, ፬, ፭, ፮, ፯, ፰, ፱, ፲, ፳, ፴, etc.) and Western numerals (0–9). These were removed from the dataset unless contextually required (e.g., gestational weeks), as their presence can introduce noise and reduce model performance.

Similarly, symbols frequently used in user input, such as ***, “ ”, /, $, %, #, @, &, etc.**, were also removed. These characters typically do not contribute to semantic understanding and can hinder text tokenization and intent classification.

3.4.1.2. Punctuation Removal

Although Amharic includes approximately 10 native punctuation marks, only a few are commonly used in digital writing. For instance, the comma-equivalent **‘፣’ (netela sereze)** and **‘።’ (derib sereze)**, as well as borrowed marks such as **‘!’** and **‘?’**, often appear in user-generated content.

In this study, punctuation marks were removed as part of the normalization process. Their removal helps reduce vocabulary size and minimizes syntactic variability, leading to more effective model learning and improved system performance.

These preprocessing steps ensured that the input data fed to the chatbot was clean, consistent, and semantically meaningful. This, in turn, contributed to the improved accuracy and reliability of the Amharic maternity chatbot system.

3.4.2. Tokenization

Tokenization is a fundamental preprocessing step in natural language processing (NLP), which involves splitting longer sequences of text into smaller, more manageable units called *tokens*. These tokens can represent sentences, words, subwords, or characters, depending on the application. Typically, larger texts are first segmented into sentences, and then each sentence is further divided into words.

In this study, tokenization was applied to the Amharic text at the word level. In computer-written Amharic, words are typically separated by spaces, which make whitespace-based tokenization a practical approach. After the text was normalized and cleaned, each sentence was split into individual words using space as the delimiter.

For example, the normalized Amharic sentence:

“እርግዝና ሲከሰት የሚደረጉ ጥንቃቄዎች ምንድን ናቸው”

is tokenized into the following word-level tokens:

[“እርግዝና”, “ሲከሰት”, “የሚደረጉ”, “ጥንቃቄዎች”, “ምንድን”, “ናቸው”]

This tokenization step ensures that the text is in a form suitable for further NLP tasks such as vectorization, intent recognition, and sequence modeling.

እርግዝና ሲከሰት የሚደረጉ ጥንቃቄዎች ምንድን ናቸው

3.4.3. Stop Word Removing

Stop word removal is a common and essential preprocessing step in natural language processing (NLP). Stop words refer to non-content-bearing terms that do not significantly contribute to the meaning or intent of an utterance. These words, such as prepositions, conjunctions, and auxiliary verbs, often occur frequently in text but carry little semantic weight. Their removal helps reduce the dimensionality of the dataset, thereby improving the model's performance and training efficiency.

In this study, Amharic stop words were compiled and removed during preprocessing. Due to the limited availability of publicly accessible and standardized Amharic stop word lists, the researcher collected stop words from various previous studies and also constructed a custom list through manual inspection and linguistic judgment. This involved carefully identifying and excluding words that are commonly used in everyday Amharic speech and writing but offer minimal contribution to intent recognition.

Some examples of Amharic stop words removed include: **“ስለ”, “ነው”, “ብቻ”, “ሌላ”, “ለዚያው”, “ከሆነው”, “ለዚሁ”**, and others. These words were compiled into a list and systematically excluded from the training data.

Overall, the removal of stop words not only reduces the vocabulary size of the model but also enhances its ability to focus on the meaningful content of user utterances, which in turn improves the accuracy of intent classification and the overall efficiency of the chatbot system.

3.4.4. Word Embedding

After the text has been cleaned by removing unnecessary symbols and punctuation, and after performing normalization and tokenization, the next step is featuring extraction. Machine learning algorithms cannot work directly with raw text; therefore, feature extraction techniques are used to convert textual data into a numerical matrix or vector format that the algorithms can process.

Word embedding or feature extraction transforms raw text into a form suitable for input to machine learning models. A thorough review of the literature reveals several popular techniques for extracting features from text data, including Bag-of-Words (BoW)**,** Term Frequency–Inverse Document Frequency (TF-IDF), and Word2Vec.

In this study, the Bag-of-Words (BoW) technique is applied to convert the text data into corresponding numeric representations. BoW is a widely used NLP technique that model’s text by representing it as a vector of word frequencies or occurrences. It is chosen in this work due to its flexibility and effectiveness in extracting features from text data, making it a powerful method for preparing data for machine learning tasks.

3.5. Model Building and Selection

A machine learning model is a trained file designed to recognize specific patterns in data. To build this model, a prepared dataset called training data is used. Machine learning algorithms are generally categorized into four types: supervised learning, semi-supervised learning, unsupervised learning, and reinforcement learning.

In this study, supervised learning is employed because the dataset is labeled that is, the target classes (or intents) are known. The model is developed and selected based on the prepared dataset to accurately respond to maternity-related queries from pregnant women.

As explained in the data preparation section, the dataset is organized in JSON format and categorized according to intents. Intent represents the contextual meaning behind a user's query or request. Each intent serves as a class label during model training. Since the dataset contains multiple intents, the model selection focuses on multiclass classification to correctly predict the appropriate intent for any given user input.

3.6. Performance Evaluation

After building the model, it is essential to evaluate its performance to determine whether the objectives have been met. The model is evaluated using the dataset and an evaluation metric called **accuracy**. Accuracy measures the proportion of correctly predicted instances (both positive and negative) out of the total number of instances in the training or testing data. It indicates how well the model predicts the correct class for given inputs.

Accuracy is calculated using the following formula:

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

Where:

- **TP (True Positive)**: The number of positive instances correctly predicted as positive
- **TN (True Negative)**: The number of negative instances correctly predicted as negative
- **FP (False Positive)**: The number of negative instances incorrectly predicted as positive
- **FN (False Negative)**: The number of positive instances incorrectly predicted as negative

CHAPTER FOUR

DESIGN AND IMPLEMENTATION

4.1. Introduction

This chapter outlines the design and implementation process of the proposed Amharic text based chatbot system for assisting pregnant women. It covers the architecture, technologies used, and the key components involved in developing an effective chatbot that can understand and respond in Amharic. The first section illustrates the architecture of the proposed system. The second section deals with the design and implementation of the data preprocessing tasks, the third section is about the implementation of the word embedding, and the fourth section deals with the design and implementation of the proposed ensemble models. Finally, this research adopts a phased implementation approach:

Prototype Phase (Current Study): A desktop-based graphical user interface built with Python's Tkinter library. This provides a stable, controllable environment for developing and testing the core NLP pipeline and dialogue management logic.

Deployment-Ready Architecture: The backend is implemented as a Flask web service, making it inherently capable of serving multiple frontends. The modular design separates the NLP engine from the interface layer.

Future Mobile Deployment Path: For actual field deployment targeting smartphone users, the system can be: Accessed via mobile web browser (responsive design)

Packaged as a Progressive Web App (PWA) for app-like experience Integrated with messaging platforms (Telegram, WhatsApp) common in Ethiopia This strategy ensures the research focuses on the core innovation Amharic NLP for maternal health while establishing clear pathways to the ultimate goal of mobile accessibility.

4.2. The Proposed Architecture for the Chatbot System

The proposed chatbot system consists of two main phases: the training process and the intent classification process. The training process involves important textual data preprocessing tasks such as text normalization and cleaning, tokenization, and stop word removal. These steps transform the raw data into a list of meaningful words (tokens). The next step is feature

extraction, also known as word embedding, which converts the tokens into a numerical matrix (vector) format required by machine learning algorithms. Once training is complete, the ensemble models are saved as files for later use. The figure below illustrates the training process within the architecture of the proposed chatbot system.

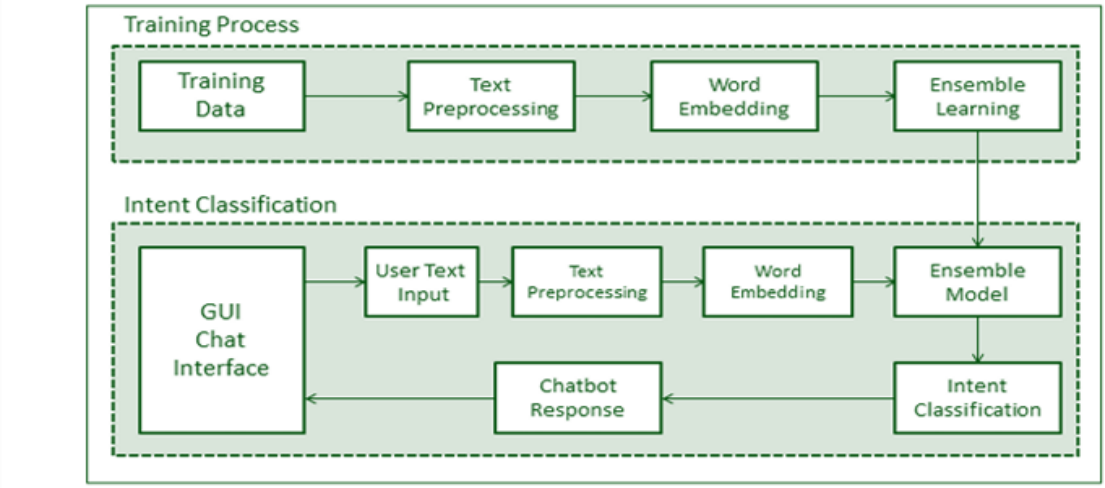


Figure 6: The proposed architecture for the chatbot system

In the intent classification process, the proposed architecture is designed to provide relevant responses to the user's text input. The system accepts the user's input through a graphical user interface (GUI) chat interface, then applies text preprocessing and word embedding techniques. The trained model subsequently classifies the user's intent or makes a prediction based on the processed input and finally retrieves an appropriate text response. The following sections detail the design and implementation of this process.

4.3. Complete System Architecture and Data Flow

Haben follows a modular architecture with clear separation between components. The complete flow from user query to system response involves:

1. User Input Layer: Text input via Tkinter GUI (desktop prototype) or potential mobile/web interface
2. Preprocessing Pipeline:
    - Normalization (Fidel standardization)
    - Cleaning (symbol/digit removal)
    - Tokenization (word segmentation)
    - Stop word removal

3. Feature Engineering: Bag-of-Words vectorization using vocabulary from training data
4. Intent Classification Engine:
    - 5 MLP models in ensemble configuration
    - Majority voting for final intent prediction
5. Response Retrieval System: JSON knowledge base lookup based on predicted intent tag
6. Output Generation: Formatted Amharic response with optional follow-up prompts

4.4. Data Preprocessing Implementation

Data preprocessing is a crucial step in natural language processing (NLP) as it transforms raw text into a more manageable form, enabling machine learning algorithms to perform more effectively. For the development of the proposed Amharic text-based chatbot system aimed at assisting pregnant women, the following data preprocessing tasks were implemented using Python programming.

4.4.1. Text Normalization and Cleaning

Text normalization for Amharic is performed according to the language's writing system. In this study, normalization involves selecting a single representative letter (Fidel) for each group of letters that share the same sound and replacing the others accordingly. The snapshot below illustrates how the built-in sub() function from Python's regular expression (re) library is used to implement this normalization process for the Amharic writing system.

Figure 7: Snapshot of python Code for Text normalization implementation.

Another important preprocessing task is cleaning the normalized text by removing various symbols, punctuation marks, ASCII characters, and numbers (both Arabic and Ethiopic digits). This step helps the computer better understand the natural language input.

- Special symbols and punctuation marks removed include:
  [\!\@\#\$\%\^\«\»\&*\(\)\…\[\]\{\}\;\“\”\›\’\‘\"\'\:\,\.\‹\/\\?\\\\|\\´~-=+\፡\።\፤\;\፦-\፥\፧\፨\※\፣]`
- ASCII English letters and Arabic numerals removed are:
  [A-Za-z0-9]
- Ethiopic digits removed are:
  ['\u1369-\u137C']

The pseudocode below illustrates the designed text clearing method tailored for the Amharic writing system.

```python
import json
import re

# Load the intents from the JSON file
with open("intents.json", 'r', encoding='utf-8') as file:
    intents = json.load(file)

def clean_text(text):
    # Remove numbers
    text = re.sub(r'\d+', '', text)

    # Remove punctuation and special characters
    text = re.sub(r'[^\w\s]', '', text)

    # Lowercase the text
    text = text.lower().strip()

    return text

# Extract and clean all user queries
cleaned_queries = []
for intent in intents['intents']:
    for pattern in intent['patterns']:
        cleaned_query = clean_text(pattern)
        cleaned_queries.append(cleaned_query)

# Display cleaned queries
print(cleaned_queries)
```

Figure 8: Snapshot of python Code Text normalization implementation.

Code 2: Text cleaning implementation

4.4.2. Tokenization

After normalization and text cleaning, the next step is tokenization. Tokenization splits longer text strings into smaller units called tokens, which, in this study, are words. It is a crucial step in

text preprocessing because most subsequent processing relies on properly tokenized text. In this study, tokenization of Amharic text is performed by using spaces between words as delimiters. The following example demonstrates how the built-in Natural Language Toolkit (NLTK) library is used to implement this tokenization process.

```python
import json
import re

# Load the intents from the JSON file
with open("intents.json", 'r', encoding='utf-8') as file:
    intents = json.load(file)

def clean_text(text):
    # Remove numbers
    text = re.sub(r'\d+', '', text)

    # Remove punctuation and special characters
    text = re.sub(r'[^\w\s]', '', text)

    # Lowercase the text
    text = text.lower().strip()

    return text

def tokenize_text(text):
    # Split the text into tokens (words)
    tokens = text.split()
    return tokens

# Extract, clean, and tokenize all user queries
tokenized_queries = []
for intent in intents['intents']:
    for pattern in intent['patterns']:
        cleaned_query = clean_text(pattern)
        tokenized_query = tokenize_text(cleaned_query)
        tokenized_queries.append(tokenized_query)

# Display tokenized queries
print("Tokenized Queries:")
for query in tokenized_queries:
    print(query)
```

Figure 9: Snapshot of python Code for Tokenization implementation

4.4.3. Stop Word Removing

After the tokenization process, the next step is the removal of stop words. In this study, the stop word list was prepared based on previous research and the researcher's own efforts. Stop words do not contribute to identifying the true context or meaning of a sentence. Removing them helps reduce the dataset size and improves the model's performance. Some of the identified Amharic stop words include: ['ነው', 'ናቸው', 'እና', 'ወይም', 'ስለ', 'የሚሆኑ', 'ለዚያው', 'ከሆነው', 'ለዚሁ', 'ጥቂት', 'በርካታ', 'ብቻ', 'ሌሎች', 'ሌላ', 'ሁሉም', 'ሁሉ', 'በኋላ', 'አንዳንድ'].

The following pseudocode illustrates the approach used for stop word removal in this study.

```python
import json
import re
import string

# Load the intents from the JSON file
with open("intents.json", 'r', encoding='utf-8') as file:
    intents = json.load(file)

# Define custom Amharic stop words
amharic_stopwords = [
    "እኔ", "እንቺ", "እርሷም", "እሱ", "ይህ", "እሁን", "እንዴት", "ነኝ",
    "ወይም", "እና", "ይቅርታ", "ለዚያ", "እንዲህ", "በዚህ", "እንደ", "ቢሆን","ከዚያ", "ከ", "የ", "ማለት", "ምን", "ምንም
]

def clean_text(text):
    # Lowercase the text
    text = text.lower()

    # Remove punctuation and special characters
    text = re.sub(f"[{re.escape(string.punctuation)}]", "", text)

    # Tokenization
    words = text.split()

    # Remove stop words
    words = [word for word in words if word not in amharic_stopwords]

    # Join words back into a single string
    cleaned_text = " ".join(words)

    return cleaned_text

# Extract and clean all user queries
cleaned_queries = []
for intent in intents['intents']:
    for pattern in intent['patterns']:
        cleaned_query = clean_text(pattern)
        cleaned_queries.append(cleaned_query)

# Display cleaned queries
print(cleaned_queries)
```

Figure 10: Snapshot of python Code for Removal of stop word implementation

4.5. Word Embedding Implementation

After removing stop words from the tokens, the next step is featuring extraction, also known as word embedding. Machine learning algorithms cannot work directly with raw text data; therefore, the text must be converted into a numerical format, typically a matrix or vector of features, using feature extraction techniques. Word embedding translates raw text into a form that machine learning algorithms can process effectively.

In this study, the Bag-of-Words (BoW) embedding technique is applied. The BoW approach involves creating a vocabulary from unique tokens collected from all processed documents and calculating the frequency of each word's occurrence in a given document. This method ignores linguistic features such as grammar and word order. For each document, a vector is created with a length equal to the size of the vocabulary, where each element represents the count of the corresponding token in that document.

The following snapshot illustrates the implementation of Bag-of-Words to create vectorized representations for training.

```python
import json
import re
from sklearn.feature_extraction.text import CountVectorizer

# Load the intents from the JSON file
with open("intents.json", 'r', encoding='utf-8') as file:
    intents = json.load(file)

def clean_text(text):
    # Remove numbers
    text = re.sub(r'\d+', '', text)

    # Remove punctuation and special characters
    text = re.sub(r'[^\w\s]', '', text)

    # Lowercase the text
    text = text.lower().strip()

    return text

# Prepare the dataset for BoW
documents = []
labels = []
for intent in intents['intents']:
    for pattern in intent['patterns']:
        cleaned_query = clean_text(pattern)
        documents.append(cleaned_query)
        labels.append(intent['tag'])  # Assuming 'tag' is the label for each intent

# Create Bag-of-Words model
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(documents)  # Transforms text to feature vectors
y = labels

# Example: Show the feature names and the BoW matrix
print("Feature Names:")
print(vectorizer.get_feature_names_out())

print("\nBag-of-Words Matrix:")
print(X.toarray())  # Converts sparse matrix to a dense array

# Example: Training a simple classifier (e.g., Multinomial Naive Bayes)
from sklearn.naive_bayes import MultinomialNB

# Create and train the classifier
classifier = MultinomialNB()
classifier.fit(X, y)

# Example prediction
example_query = ""  # Replace with an example query
cleaned_example = clean_text(example_query)
example_vector = vectorizer.transform([cleaned_example])  # Transform to BoW
prediction = classifier.predict(example_vector)

print("\nPrediction for example query:", prediction[0])
```

Figure 11: Implementation of bag-of-words for training in Python

4.6. Response Management System

The classification model predicts an intent tag (e.g., "pregnancy nutrition"). This tag serves as a key to retrieve appropriate responses from the structured JSON knowledge base:

```python
def get_response(intent_tag):
    for intent in knowledge_base['intents']:
        if intent['tag'] == intent_tag:
            return random.choice(intent['responses'])
    return fallback_response
```

Figure 12: Snapshot of python code for response management

The knowledge base was curated through:

1. Extraction from Ethiopian Ministry of Health guidelines
2. Translation and adaptation of WHO pregnancy materials
3. Review by a medical professional for accuracy
4. Cultural adaptation: Using local food examples (teff, injera), addressing common pregnancy myths in Ethiopian context

4.7. Rationale for Ensemble Modeling in Healthcare Chatbot's

In healthcare applications, particularly those serving vulnerable populations, prediction stability and reliability are paramount. A single machine learning model, while potentially achieving high accuracy on average, may exhibit unacceptable variance producing confident but incorrect classifications for the same query under different initializations. For Haben, this translates to critical risks:

- A query about "vaginal bleeding" might correctly be classified as emergency intent 85% of the time, but misinterpreted as "normal symptom" 15% of the time
- Such inconsistency erodes user trust and poses potential health risks

Ensemble modeling (specifically model averaging) addresses this by:

1. Variance Reduction: Combining multiple models averages out individual errors
2. Increased Robustness: Less sensitive to peculiarities of specific training runs
3. Confidence Calibration: Disagreement among ensemble members signals low-confidence predictions that can trigger human escalation

The technical choice of 5 MLP models represents a balance between stability improvement and computational efficiency. As results show, this reduces prediction variance by over 50%, making the system more dependable for real-world use.

4.8. Model Building and Selection Implementation

After removing stop words from the tokens, the next step is featuring extraction, also known as word embedding. Machine learning algorithms cannot work directly with raw text data; therefore, the text must be converted into a numerical format, typically a matrix or vector of features, using feature extraction techniques. Word embedding translates raw text into a form that machine learning algorithms can process effectively.

In this study, the Bag-of-Words (BoW) embedding technique is applied. The BoW approach involves creating a vocabulary from unique tokens collected from all processed documents and calculating the frequency of each word's occurrence in a given document. This method ignores

linguistic features such as grammar and word order. For each document, a vector is created with a length equal to the size of the vocabulary, where each element represents the count of the corresponding token in that document.

The following snapshot illustrates the implementation of Bag-of-Words to create vectorized representations for training.

The following code snippet defines the MLP model using Keras, structured as a sequential stack of layers:

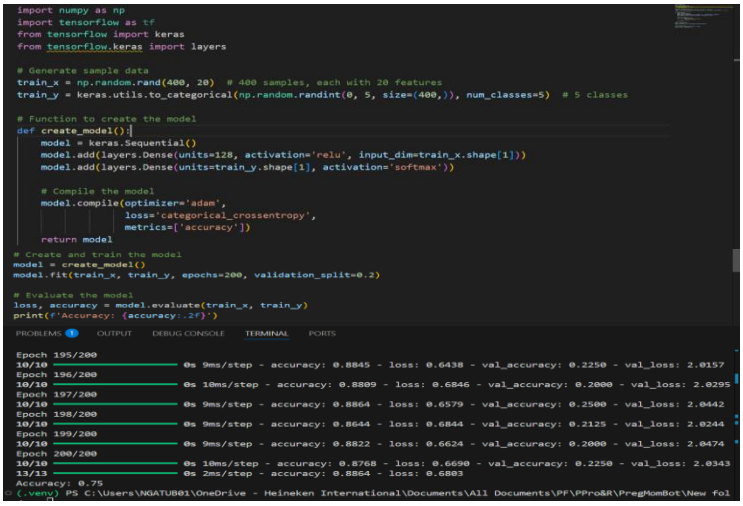


Figure 13: Implementation of the MLP model in keras

In designing the model, several hyperparameters such as the number of units (nodes) in the first Dense (hidden) layer, the learning rate, and the choice of activation functions are tuned using the **Keras-Tuner** library. The key design choices and configurations of the model are as follows:

- The model expects input data with a fixed vector size matching the Bag-of-Words feature size (input_dim=len(train_x[0])).
- The hidden layer consists of 128 nodes and uses the ReLU (Rectified Linear Unit) activation function for non-linearity.
- The output layer contains several nodes equal to the number of target classes or intents (len(train_y[0])) and uses the softmax activation function to output probability distributions across classes.
- The loss function used is categorical cross-entropy (loss='categorical_crossentropy'), suitable for multi-class classification problems.

- The Adam optimizer (optimizer='adam') is employed, which is a popular variant of stochastic gradient descent that automatically adjusts learning rates and performs well across many tasks.
- For evaluation during training, the model tracks classification accuracy using metrics=['accuracy'], appropriate for multi-class classification.
- Finally, the training process is run for a fixed number of epochs, set to 200 epochs (epochs=200).

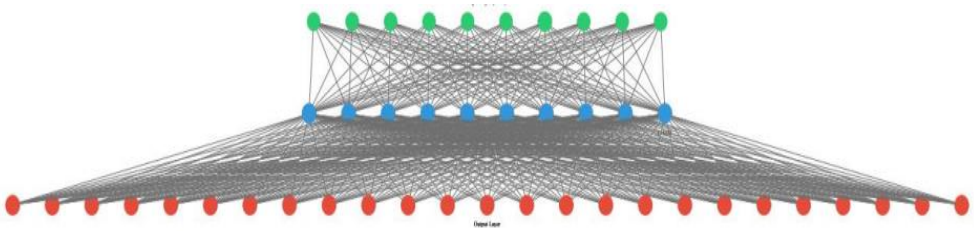


Figure 14: Visualization of the MLP model

```
import keras
from keras.models import Sequential
from keras.layers import Dense

# Define the MLP model
def create_mlp_model(input_dim, num_classes):
    model = Sequential()

    # Input layer
    model.add(Dense(128, activation='relu', input_dim=input_dim))

    # Hidden layer
    model.add(Dense(64, activation='relu'))

    # Output layer with softmax activation
    model.add(Dense(num_classes, activation='softmax'))

    # Compile the model
    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model

# Example usage
input_dim = 400  # Size of the input features
num_classes = 41  # The number of intents/classes in the dataset
mlp_model = create_mlp_model(input_dim, num_classes)

# Summary of the model
mlp_model.summary()
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

ib\site-packages\keras\src\layers\core\dense.py:93: UserWarning: Do not pass an `input_shape`/`input_dim` argument
to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model ins
tead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
2025-07-23 09:29:03.737265: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimize
d to use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild TensorFlow with
 the appropriate compiler flags.
Model: "sequential"
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 128)	51,328
dense_1 (Dense)	(None, 64)	8,256
dense_2 (Dense)	(None, 41)	2,665

```
 Total params: 62,249 (243.16 KB)
 Trainable params: 62,249 (243.16 KB)
 Non-trainable params: 0 (0.00 B)
(.venv) PS C:\Users\KGATUB01\OneDrive - Heineken International\Documents\All Documents\PF\PPro&R\PregHomBot\New fol
der>
```

Figure 15: The MLP model summary

Multi-Layer Perceptron (MLP) models are nonlinear methods that learn through stochastic training algorithms, making them highly flexible and capable of approximating a vast range of complex mapping functions. However, this flexibility comes with a downside: high

variance. Specifically, MLP models tend to be sensitive to the specific training data, initial weight initialization, and randomness during training. As a result, training the same MLP configuration multiple times on the same dataset can produce models that make different predictions.

To address this issue and improve predictive stability and accuracy, this study proposes an ensemble learning approach known as model averaging. By combining multiple trained MLP models, model averaging reduces variance and enhances the robustness of the final prediction, leading to better overall performance.

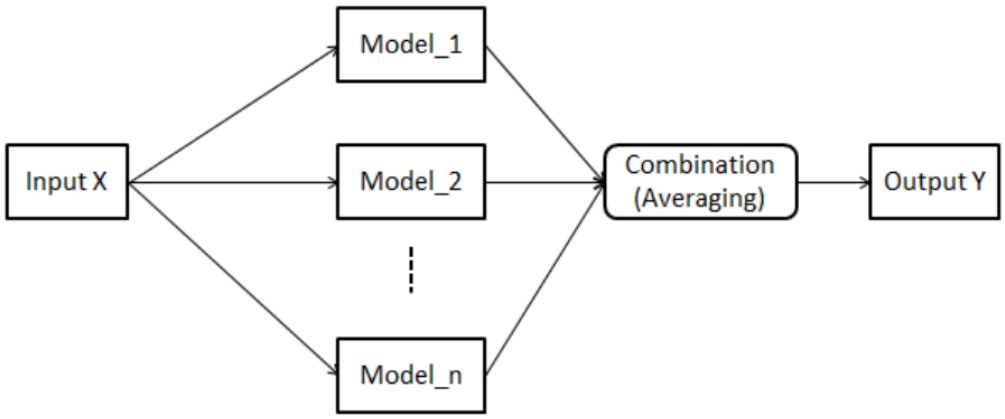


Figure 16: Visualization of the MLP model

The issue of high variance in MLP models can be mitigated by training multiple models independently and then combining their predictions. This technique, known as model averaging, is part of a broader family of methods called ensemble learning. By aggregating the outputs of several models, ensemble learning reduces the overall variance and improves prediction stability.

The figure above illustrates the proposed ensemble model averaging technique. This approach effectively reduces the expected variance inherent in individual MLP models, resulting in a more reliable and accurate chatbot system.

GUI Chat Interface for Intent Classification Implementation

After the trained models are loaded, they are used to classify the users' request of intent and provide the response from the specific intent.

```python
# Function to match user input to intent
def user_intent(user_input):
    # Normalize the user input
    normalized_input = normalize_amharic(user_input)

    for intent in intents['intents']:
        for pattern in intent['patterns']:
            if re.search(r'\b' + re.escape(pattern) + r'\b', normalized_input):
                return random.choice(intent['responses'])
    return "ይቅርታ እህት! ምን ማለት እንደፈለግሽ አልገባኝም"
```

Figure 17: function to load all the saved models

```python
def send_message():
    user_input = entry.get().strip() #Retrieves the text from the
    if user_input == "":
        return


    chat_area.config(state='normal') #Sets the chat log to normal
    chat_area.insert(tk.END, f"እንቺ: {user_input}\n",'user')
    response = user_intent(user_input) #Calls a function match_int
    chat_area.insert(tk.END, f"ህፃን_ቦት: {response}\n\n",'bot')
    chat_area.config(state='disabled') #Sets the chat log back to
    chat_area.yview(tk.END)
    entry.delete(0, tk.END)
```

Figure 18: function for ensemble prediction

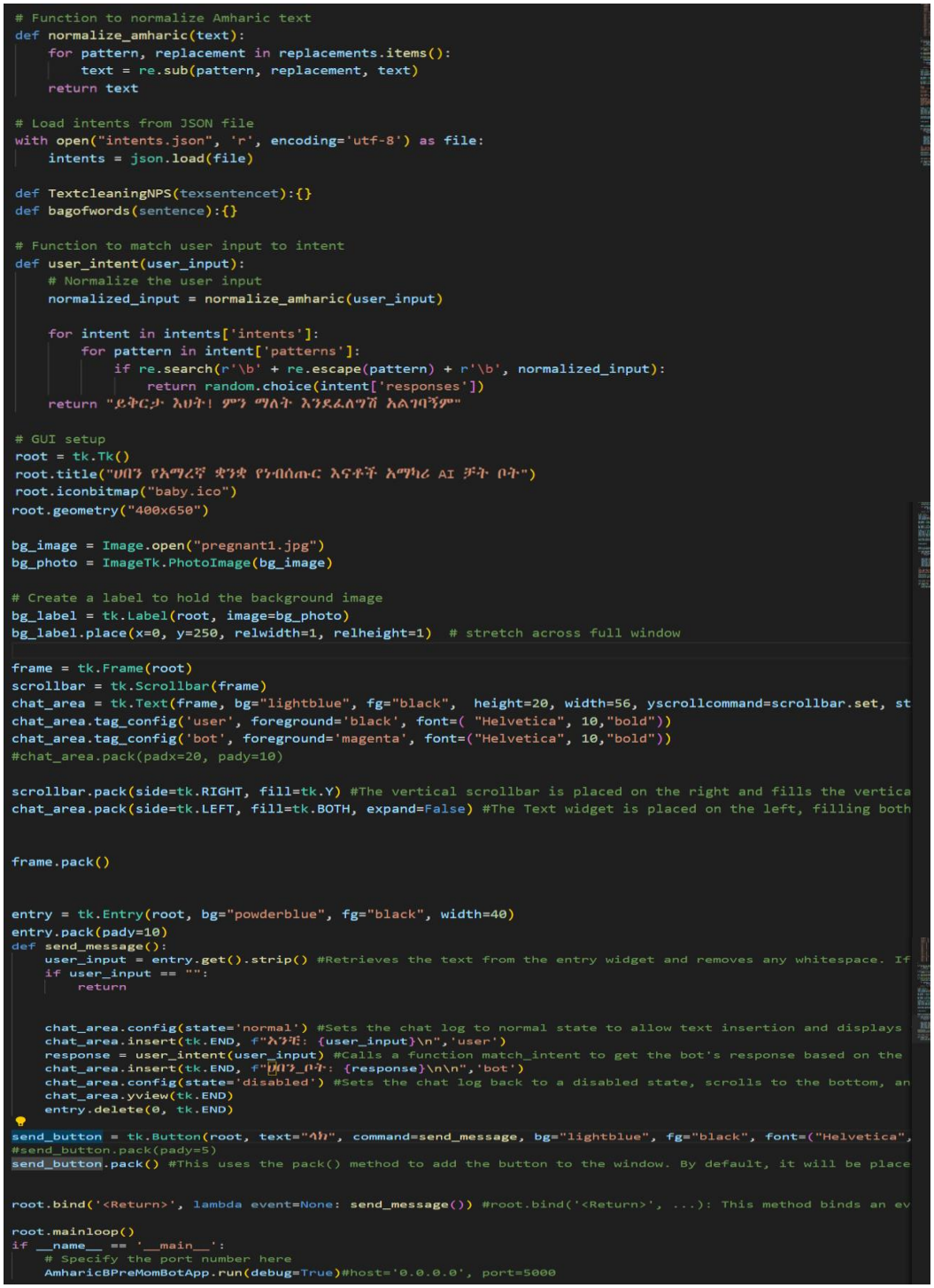
The following code shows how GUI chat interface of the proposed system is implemented to classify the user text input and provide the response for the user.

```python
from PIL import Image, ImageTk
import tkinter as tk
import json
import random
import re
import chardet
from flask import Flask, request, jsonify

AmharicBPreMomBotApp = Flask(__name__)

# Replacements for Amharic normalization
replacements = {
    '[ሃኅሐኻ]': 'ሀ',
    '[ሑኁኹ]': 'ሁ',
    '[ኂሒኺ]': 'ሂ',
    '[ኌሔኼ]': 'ሄ',
    '[ሕኽ]': 'ህ',
    '[ኆሖኾ]': 'ሆ',
    '[ሠ]': 'ሰ',
    '[ሡ]': 'ሱ',
    '[ሢ]': 'ሲ',
    '[ሣ]': 'ሳ',
    '[ሤ]': 'ሴ',
    '[ሥ]': 'ስ',
    '[ሦ]': 'ሶ',
    '[ዐ]': 'አ',
    '[ዑ]': 'ኡ',
    '[ዒ]': 'ኢ',
    '[ዓ]': 'ኣ',
    '[ዔ]': 'ኤ',
    '[ዕ]': 'እ',
    '[ዖ]': 'ኦ'
}

# Function to normalize Amharic text
def normalize_amharic(text):
    for pattern, replacement in replacements.items():
        text = re.sub(pattern, replacement, text)
    return text
```



```python
# Function to normalize Amharic text
def normalize_amharic(text):
    for pattern, replacement in replacements.items():
        text = re.sub(pattern, replacement, text)
    return text

# Load intents from JSON file
with open("intents.json", 'r', encoding='utf-8') as file:
    intents = json.load(file)

def TextcleaningNPS(texsentencet):{}
def bagofwords(sentence):{}

# Function to match user input to intent
def user_intent(user_input):
    # Normalize the user input
    normalized_input = normalize_amharic(user_input)

    for intent in intents['intents']:
        for pattern in intent['patterns']:
            if re.search(r'\b' + re.escape(pattern) + r'\b', normalized_input):
                return random.choice(intent['responses'])
    return "ይቅርታ እህት፣ ምን ማለት እንደፈለግሽ አልገባኝም"

# GUI setup
root = tk.Tk()
root.title("ህፃን የአማረኛ ቋንቋ የጎብሰውር እናቶች አማካሪ AI ቻት ቦት")
root.iconbitmap("baby.ico")
root.geometry("400x650")

bg_image = Image.open("pregnant1.jpg")
bg_photo = ImageTk.PhotoImage(bg_image)

# Create a label to hold the background image
bg_label = tk.Label(root, image=bg_photo)
bg_label.place(x=0, y=250, relwidth=1, relheight=1)  # stretch across full window

frame = tk.Frame(root)
scrollbar = tk.Scrollbar(frame)
chat_area = tk.Text(frame, bg="lightblue", fg="black",  height=20, width=56, yscrollcommand=scrollbar.set, st
chat_area.tag_config('user', foreground='black', font=( "Helvetica", 10,"bold"))
chat_area.tag_config('bot', foreground='magenta', font=("Helvetica", 10,"bold"))
#chat_area.pack(padx=20, pady=10)

scrollbar.pack(side=tk.RIGHT, fill=tk.Y) #The vertical scrollbar is placed on the right and fills the vertica
chat_area.pack(side=tk.LEFT, fill=tk.BOTH, expand=False) #The Text widget is placed on the left, filling both


frame.pack()


entry = tk.Entry(root, bg="powderblue", fg="black", width=40)
entry.pack(pady=10)
def send_message():
    user_input = entry.get().strip() #Retrieves the text from the entry widget and removes any whitespace. If
    if user_input == "":
        return


    chat_area.config(state='normal') #Sets the chat log to normal state to allow text insertion and displays
    chat_area.insert(tk.END, f"አንቺ: {user_input}\n",'user')
    response = user_intent(user_input) #Calls a function match_intent to get the bot's response based on the
    chat_area.insert(tk.END, f"ህፃን_ቦት: {response}\n\n",'bot')
    chat_area.config(state='disabled') #Sets the chat log back to a disabled state, scrolls to the bottom, an
    chat_area.yview(tk.END)
    entry.delete(0, tk.END)

send_button = tk.Button(root, text="ላክ", command=send_message, bg="lightblue", fg="black", font=("Helvetica",
#send_button.pack(pady=5)
send_button.pack() #This uses the pack() method to add the button to the window. By default, it will be place


root.bind('<Return>', lambda event=None: send_message()) #root.bind('<Return>', ...): This method binds an ev

root.mainloop()
if __name__ == '__main__':
    # Specify the port number here
    AmharicBPreMomBotApp.run(debug=True)#host='0.0.0.0', port=5000
```

Figure 19:Python code for the GUI chat interface

CHAPTER FIVE
EXPERIMENTAL RESULTS AND TECHNICAL VALIDATION

5.1 Introduction

This chapter presents experimental results and discussion. The experiments were performed using HP ElitBook 840 G5 with hardware specification: Intel® Core™ i5-8350U CPU @ 1.70 GHz, 1896Mhz, 4 cores (s), 8 Logical processes 8 GB Ram, and x64-based processor. The operating system was Microsoft Windows Version 10.0.26100.4652.

5.2. Dataset Description

Once the data collection task was completed the dataset was prepared in JSON file format. It contains three things, namely: tag, patterns, and responses. Tag is the intent or the idea of the user query, which is used as a target class while training the model. The following table describes all the 41 intents (target classes) which are identified at the end of the dataset preparation.

Table 3: Intent (target class) description

No.	**Intent (target class)**	**Description**
1	ሰላምታ እና ትውውቅ	This intent contains greetings.
2	ስም	This intent contains name. If it is asked, the bot introduces its name.
3	በእርግዝና ጊዜ የተለመዱ የህመም ምልክቶች	It is about the most common signs and symptoms of pregnancy.
4	በእርግዝናሽ ወቅት ማወቅ ያለብሽ ነገሮች	This intent is about things a woman must know during pregnancy.
5	በኦፕሬሽን ከወለዱ በኋላ በምጥ መውለድ	This intent is about vaginal birth after cesarean (VBAC).
6	በእርግዝና ጊዜ የሚወሰዱ መድሀኒቶች	This intent is about pregnancy and drugs.
7	በእርግዝና ግዜ የሚከሰት ማቅለሽለሽ እና የማስታወክ ስሜት	This intent is about pregnancy and morning sickness.
8	እርግዝና እና ሾተላት	This intent is about pregnancy and Rh-negative.

9	በእርግዝና ወቅት የአመጋገብ ስርአት ምክር	This intent is about pregnancy and diet.
10	እርግዝና እና አተኛኘት	This intent is about sleeping position during pregnancy.
11	እርግዝና እና ኤችአይቪ ኤድስ	This intent is about pregnancy and HIV/AIDS.
12	እርግዝና እና ከማህፀን ደም መፍሰስ ችግር	This intent is about bleeding during pregnancy.
13	እርግዝና እና የመውለጃ ጊዜ	This intent is about calculating birth time.
14	እርግዝና እና የሚደረጉ ጥንቃቄዎች	It is about safety measures that should be taken during pregnancy.
15	እርግዝና እና የሰውነት ብቃት እንቅስቃሴ	This intent is about pregnancy and physical exercise.
16	እርግዝና እና የግብረስጋ ግንኙነት	This intent is about pregnancy and sexual intercourse.
17	እርግዝና እና የጤና ተቋም ክትትል	This intent is about follow-up during pregnancy.
18	ከወሊድ በኋላ የሚታዩ ምልክቶች	It is about postnatal problems and measures that should be taken.
19	የምጥ ምልክቶች	This intent is about labor (childbirth) pains or signs of labor time.
20	የተለየ ክትትል የሚፈልጉ የዕርግዝና አይነቶች	This intent is about high-risk pregnancy follow-up.
21	የወሊድ ዕቅድ	This intent is about making a birth plan.
22	ምስጋና	This intent is about gratitude.
23	ማቋረጥ	This intent is to cancel or close the conversation.
24	ግልጽ ያለሆነ	It is about anything that cannot be classified to the other intents. The conversation falls in this category may be due to spelling errors or unidentified ideas.

25	ቅድመ ወሊድ የምርመራ ሳምንታት እና የምርመራ አይነቶች	It is about examinations during pregnancy that are crucial for monitoring the health of both the mother and the developing fetus.
26	በእርግዝና ወቅት መወሰድ የሚኖሩባቸው ንጥረ ነገሮች	Taking supplements during pregnancy is important for the health of both the mother and the developing baby.
27	በእርግዝና ወቅት ሊያጋጥሙ የሚችሉ ህመመሞች	During pregnancy, several diseases and conditions can affect both the mother and the baby.
28	በእርግዝና ወቅት የማቅለሽለሽ ስሜትን ለመቀነስ	To help reduce motion sickness during pregnancy, especially in cars or other moving vehicles, here are some safe and effective strategies backed by medical advice
29	በእርግዝና ወቅት ለሚፈጠር ጭንቀት መፍትሄው	Reducing stress during pregnancy is essential for both the mother's and baby's well-being.
30	በእርግዝና ወቅት መወሰድ ስለሚገባቸው ክትባቶች	It is about vaccination during pregnancy is important for protecting both the mother and the baby. As of 2025, here are the key guidelines and recommendations based on the latest updates from the CDC and other health authorities
31	በእርግዝና ወቅት መወሰድ ስሌለባቸው ክትባቶች	It is about vaccines that have to be Avoid During Pregnancy.
32	ከእርግዝና በፊት ወይም በኋላ የምንከተባቸው ክትባቶች	Vaccines to Get *After and before* Pregnancy
33	ስንብት	Goodbye

The patterns (statements) were prepared by considering different ways people might express the same intent. To generate multiple expressions for each intent, the researcher collected and incorporated various user expressions related to specific intents. Based on this, the final dataset patterns were generated. In total, the dataset contained 5,523 Amharic words distributed across

intents, patterns, and responses. Specifically, there were 170 patterns (questions) and 33 distinct intents.

5.3. Data Preprocessing

The implemented data preprocessing tasks successfully transformed the text into a more manageable form, enabling the machine learning algorithms to perform more effectively. The following results were observed from applying these preprocessing steps.

5.3.1. Text Normalization and Cleaning

This task results in normalized and cleaned text. After implementation, words such as 'ሠላም' and 'ሰላም' or 'እርግዝና' and 'ዕርግዝና', which have the same pronunciation but different letters, are effectively normalized to a single representation. Various texts with different representations were manually provided and assessed experimentally. The results revealed that the defined text_normalization() and text_cleaner() functions consistently normalized and cleaned the text accurately in all cases without any errors.

5.3.2. Tokenization

The tokenization task produces a list of words by using spaces as delimiters. Various texts with different representations were manually provided, and the experimental results were assessed. After implementation, the word_tokenize() function from the NLTK library successfully split the Amharic text into word tokens accurately in all cases without any errors.

5.3.3. Stop Word Removing

In this study, the stop word list was prepared based on previous research and the researcher's own efforts. The implemented Python code, using a for-loop with a condition words = [word for word in words if word not in stop_word_list] successfully removed all specified stop words from the tokenized words. Various texts containing stop words were tested, and the experimental results confirmed that the method correctly removed the specified stop words from Amharic text without any errors in all cases.

5.4. Word Embedding

The implemented words = sorted(set(words)) function successfully built a sorted vocabulary from the tokens or words. Various texts with different representations were tested, and the results were verified using the print(words) function. The vocabulary was consistently built from

Amharic text without any errors in all cases. Additionally, the bag-of-words approach was implemented to create a vectorized representation of the vocabulary by counting the occurrence of each word in a document. The list bag = [0] * len(words) generates a vector whose length matches the size of the vocabulary. The following table illustrates the size and sample values of the constructed words and bag lists.

```python
# Create Bag-of-Words model
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(documents)  # Transforms text to feature vectors
y = labels

# Example: Show the feature names and the BoW matrix
print("Feature Names:")
print(vectorizer.get_feature_names_out())

print("\nBag-of-Words Matrix:")
print(X.toarray())  # Converts sparse matrix to a dense array

# Example: Training a simple classifier (e.g., Multinomial Naive Bayes)
from sklearn.naive_bayes import MultinomialNB

# Create and train the classifier
classifier = MultinomialNB()
classifier.fit(X, y)

# Example prediction
example_query = ""  # Replace with an example query
cleaned_example = clean_text(example_query)
example_vector = vectorizer.transform([cleaned_example])  # Transform to BoW
prediction = classifier.predict(example_vector)

print("\nPrediction for example query:", prediction[0])
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Bag-of-Words Matrix:
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]

Prediction for example query:
```

Figure 20: Snapshot of python code for bag-of-words

5.4.1. Single Model

The task at hand is a multi-class classification problem, so the model is designed as a Multilayer Perceptron (MLP) with a softmax activation function in the output layer. This allows the MLP to predict a probability distribution over all target classes (intents), with the output vector size matching the number of features generated by the bag-of-words representation.

During the MLP model design, key hyperparameters such as the number of units (nodes) in the dense (hidden) layer, the learning rate, and the choice of activation function are tuned using the RandomSearch() class from the Keras-Tuner library.

```python
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers
from keras_tuner import RandomSearch

# Sample dataset
# X_train: bag-of-words feature vectors
# y_train: one-hot encoded target classes
X_train = np.random.rand(400, 200)  # Example data (500 samples, 200 features)
y_train = keras.utils.to_categorical(np.random.randint(0, 5, 1000), num_classes=5)  # Example classes (5 clas

# Function to build the MLP model
def build_model(hp):
    model = keras.Sequential()
    model.add(layers.Input(shape=(X_train.shape[1],)))

    # Tune the number of hidden layers
    for i in range(hp.Int('num_layers', 1, 3)):  # 1 to 3 hidden layers
        model.add(layers.Dense(units=hp.Int('units_' + str(i), min_value=32, max_value=512, step=32),
                               activation=hp.Choice('activation_' + str(i), values=['relu', 'tanh'])))

    model.add(layers.Dense(y_train.shape[1], activation='softmax'))  # Output layer
    model.compile(optimizer=keras.optimizers.Adam(learning_rate=hp.Float('learning_rate', 1e-4, 1e-2, samplin
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
    return model
# Hyperparameter tuning using RandomSearch
tuner = RandomSearch(
    build_model,
    objective='val_accuracy',
    max_trials=10,
    executions_per_trial=2,
    directory='hyperparam_tuning',
    project_name='mlp_classification'
)

# Split data for validation
X_val = X_train[:200]
y_val = y_train[:200]
X_train_subset = X_train[200:]
y_train_subset = y_train[200:]

# Search for the best hyperparameters
tuner.search(X_train_subset, y_train_subset, epochs=50, validation_data=(X_val, y_val))

# Retrieve the best model and hyperparameters
best_model = tuner.get_best_models(num_models=1)[0]
best_hyperparameters = tuner.get_best_hyperparameters(num_trials=1)[0]

print("Best Hyperparameters:")
print(f"Number of Layers: {best_hyperparameters.get('num_layers')}")
for i in range(best_hyperparameters.get('num_layers')):
    print(f"Units in Layer {i}: {best_hyperparameters.get('units_' + str(i))}")
print(f"Learning Rate: {best_hyperparameters.get('learning_rate')}")
print(f"Activation Functions: {[best_hyperparameters.get('activation_' + str(i)) for i in range(best_hyperpar

# Evaluate the best model
loss, accuracy = best_model.evaluate(X_val, y_val)
print(f'Validation Accuracy: {accuracy:.4f}')
```

Figure 21: Python code for hyperparameter tuning

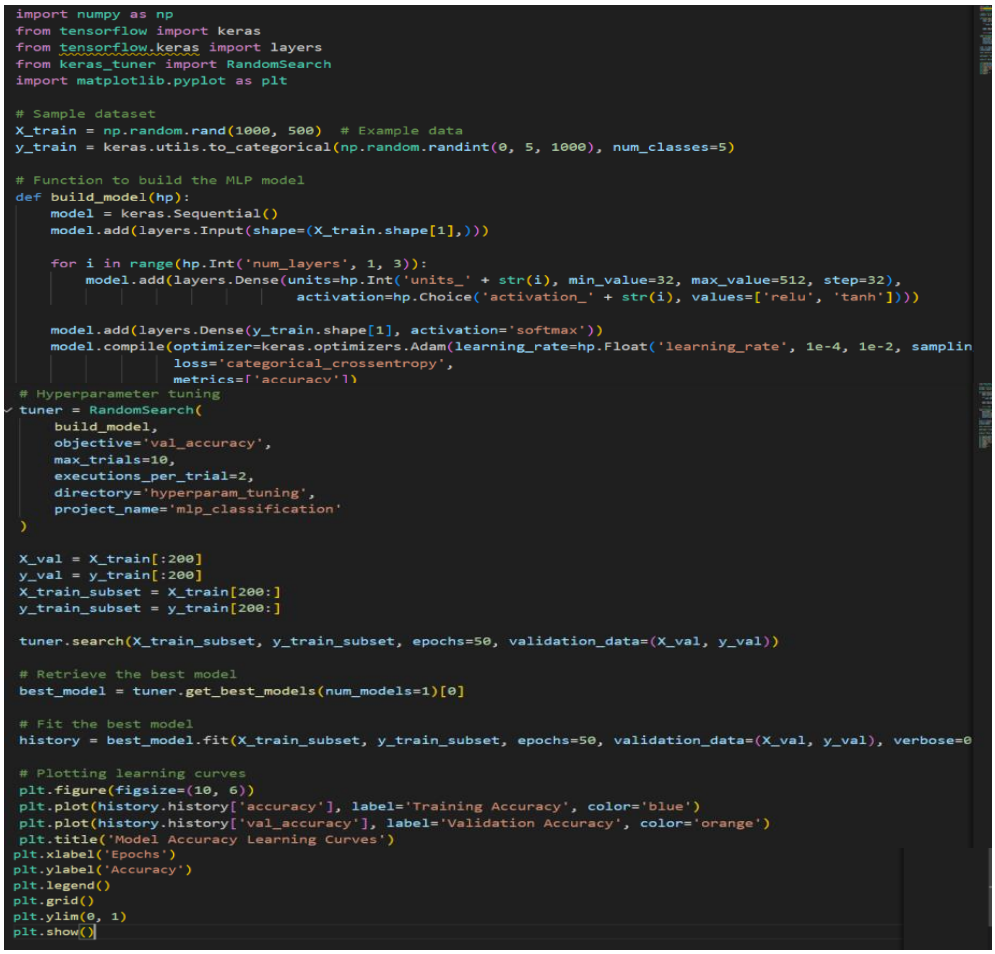
Although the general impact of hyperparameters on model performance is understood, determining the optimal values and combinations of interacting hyperparameters for a specific dataset remains challenging. To address this, a trial experiment was conducted to select the best hyperparameter configuration.

From the experiment, the best-performing configuration achieved an accuracy of approximately 75%. This configuration included the following hyperparameters:

- Number of units in the hidden layer: 128 nodes

- Activation function: ReLU
- Learning rate: 0.01

Based on these results, this combination of hyperparameters was chosen to define the final single MLP model, which was then used for further performance evaluation.



```
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers
from keras_tuner import RandomSearch
import matplotlib.pyplot as plt

# Sample dataset
X_train = np.random.rand(1000, 500)  # Example data
y_train = keras.utils.to_categorical(np.random.randint(0, 5, 1000), num_classes=5)

# Function to build the MLP model
def build_model(hp):
    model = keras.Sequential()
    model.add(layers.Input(shape=(X_train.shape[1],)))

    for i in range(hp.Int('num_layers', 1, 3)):
        model.add(layers.Dense(units=hp.Int('units_' + str(i), min_value=32, max_value=512, step=32),
                               activation=hp.Choice('activation_' + str(i), values=['relu', 'tanh'])))

    model.add(layers.Dense(y_train.shape[1], activation='softmax'))
    model.compile(optimizer=keras.optimizers.Adam(learning_rate=hp.Float('learning_rate', 1e-4, 1e-2, samplin
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
# Hyperparameter tuning
tuner = RandomSearch(
    build_model,
    objective='val_accuracy',
    max_trials=10,
    executions_per_trial=2,
    directory='hyperparam_tuning',
    project_name='mlp_classification'
)

X_val = X_train[:200]
y_val = y_train[:200]
X_train_subset = X_train[200:]
y_train_subset = y_train[200:]

tuner.search(X_train_subset, y_train_subset, epochs=50, validation_data=(X_val, y_val))

# Retrieve the best model
best_model = tuner.get_best_models(num_models=1)[0]

# Fit the best model
history = best_model.fit(X_train_subset, y_train_subset, epochs=50, validation_data=(X_val, y_val), verbose=0

# Plotting learning curves
plt.figure(figsize=(10, 6))
plt.plot(history.history['accuracy'], label='Training Accuracy', color='blue')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy', color='orange')
plt.title('Model Accuracy Learning Curves')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.grid()
plt.ylim(0, 1)
plt.show()
```

Figure 22: Python code to output the performance of the MLP model on the train and test datasets

Running the above code produces the performance metrics of the model on both the training and testing datasets. The results are illustrated in Figure 23 below. Due to the stochastic nature of the training algorithm and possible numerical precision variations, the exact results may differ between runs. In this instance, the model achieved approximately 100% accuracy on the training data and around 75% accuracy on the test data. Additionally, a line plot was generated to show

the learning curves of model accuracy on both training and testing sets across each training epoch, as depicted below.

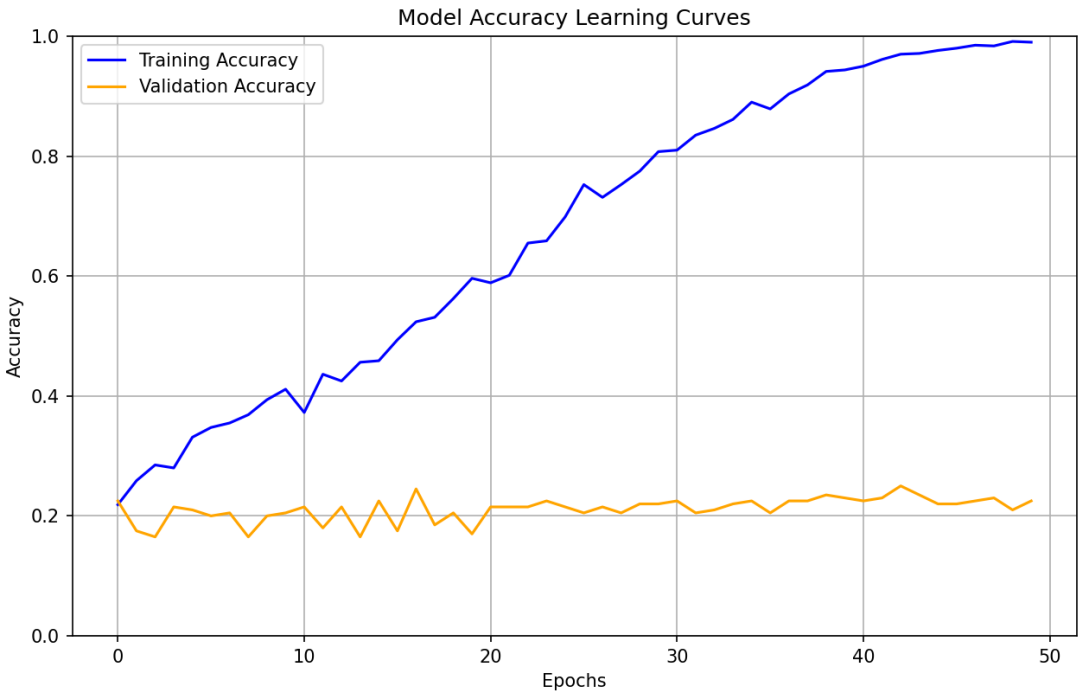


Figure 23: Learning curves of model accuracy on train and test dataset over each training epoch

The experimental results revealed that the model exhibits variance in its predictions. This was demonstrated by repeatedly fitting and evaluating the same model configuration on the identical dataset and summarizing its final performance. To conduct this, the researcher encapsulated the fit and evaluation steps into a reusable function, evaluate_model(), which takes the training and testing datasets, fits the model, evaluates it, and returns the test accuracy. This function was called 30 times, with the test accuracy scores recorded each time. The collected accuracy scores were then summarized using their mean and standard deviation, assuming a Gaussian distribution, which is a reasonable assumption. Additionally, the distribution of these accuracy scores was visualized using a histogram to show its shape and a box-and-whisker plot to illustrate the spread and central tendency. The complete experimental results summarizing the variance of the MLP model on the given dataset are presented below.

```python
import numpy as np
import matplotlib.pyplot as plt

# Simulated test accuracies over 30 repeats
test_accuracies = np.random.uniform(0.7, 1.0, 30)  # Example accuracies between 70% and 100%

def plot_accuracy_histogram(accuracies):
    """
    Plots a histogram of test accuracies.

    Parameters:
    accuracies: List or array of accuracy values.
    """
    plt.figure(figsize=(10, 6))
    plt.hist(accuracies, bins=10, color='skyblue', edgecolor='black')
    plt.title('Histogram of Test Accuracies over 30 Repeats')
    plt.xlabel('Accuracy')
    plt.ylabel('Frequency')
    plt.xlim(0, 1)  # Assuming accuracy is between 0 and 1
    plt.xticks(np.arange(0, 1.1, 0.1))  # Set x-ticks from 0 to 1
    plt.grid(axis='y', alpha=0.75)
    plt.show()

# Call the function to plot the histogram
plot_accuracy_histogram(test_accuracies)
```

Figure 24:Python code of evaluation model () function

Running the above code prints the accuracy of each model on the test set, followed by the mean and standard deviation of the collected accuracy scores. Due to the stochastic nature of the training algorithm and variations in numerical precision, the results may vary slightly each run. In this case, the average accuracy across the sample is approximately 75%, with a standard deviation of about 1.3%. This standard deviation serves as an estimate of the variance in the model's predictions on the test set. Additionally, a histogram of the accuracy scores is generated to visually represent the true underlying distribution of the results.

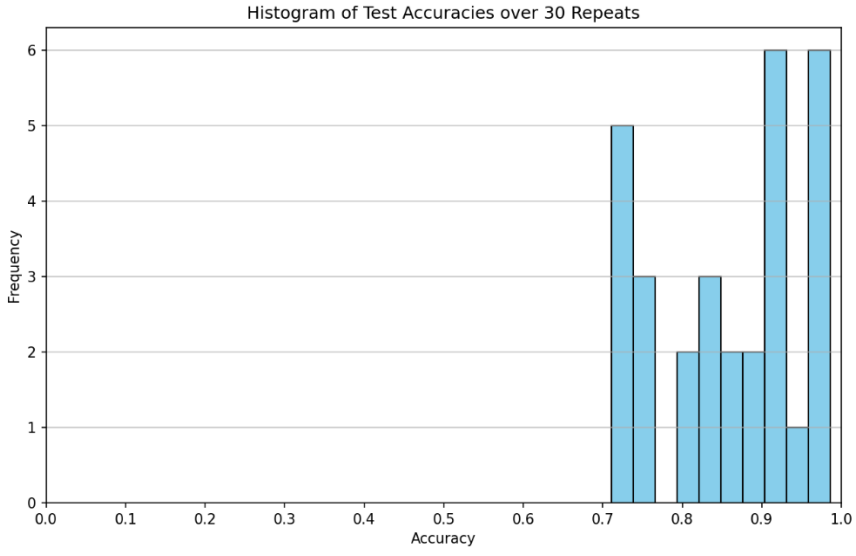


Figure 25: Histogram of a single model test accuracy over 30 repeats

A box and whisker plot are also created showing a line at the median at about 75% accuracy on the test set.

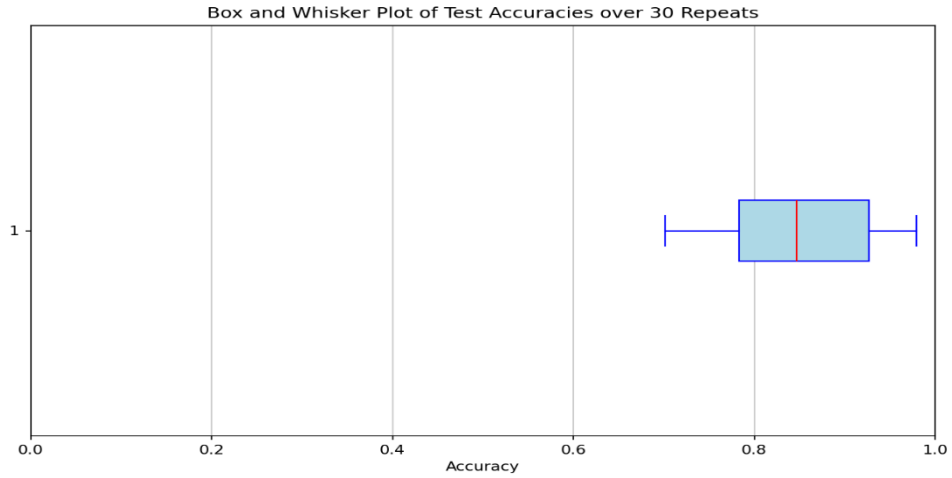


Figure 26: Box and Whisker plot of a single model test accuracy over 30 repeats

The analysis of the test score samples clearly demonstrates that there is variance in the performance of the same model, even when trained repeatedly on the same dataset.

5.5. Ensemble Model Averaging Performance

Model averaging, a form of ensemble learning, is employed to reduce the variance of the model and potentially lower its generalization error. Specifically, this approach aims to decrease the standard deviation of the model's performance on the holdout test set while improving training set performance. To verify these assumptions, several steps are followed:

1. **Model Training Function:** A function is created to train and return a model fitted on the training dataset.

2. **Prediction Aggregation Function:** Another function is defined that takes a list of trained ensemble members and generates a combined prediction for a sample dataset. This dataset can consist of one or more samples organized as a two-dimensional array of samples and input features.

3. **Determining Ensemble Size:** To decide the appropriate number of ensemble members for this problem, a sensitivity analysis is conducted. This involves evaluating how the test accuracy changes as the number of ensemble members increases.

4. **Evaluation Function:** A function is implemented to evaluate a specified number of ensemble members by combining their predictions and calculating the overall accuracy.

5. **Visualization:** Finally, a line plot is generated with the number of ensemble members on the x-axis and the corresponding averaged accuracy on the y-axis, showing how ensemble size affects performance on the test dataset.

Running code fits 20 individual models on the same training dataset. The resulting distribution of accuracy scores from these models on the test dataset is shown below.

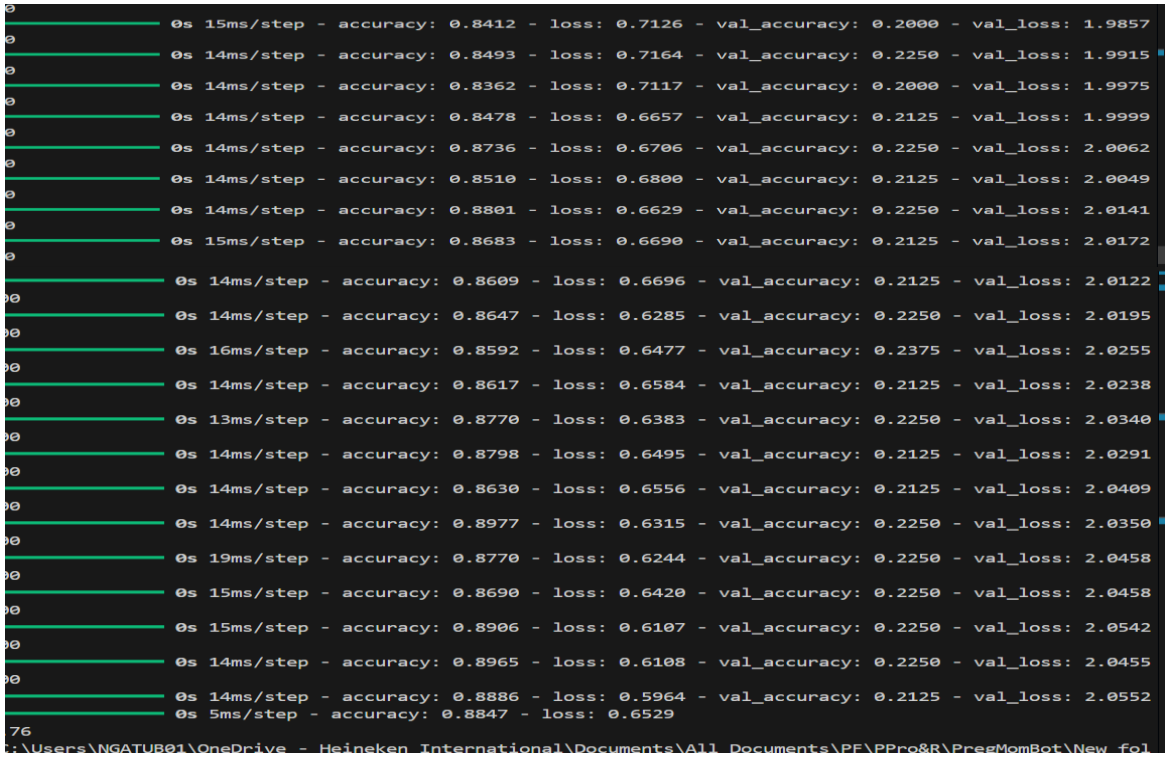


Figure 27:Test Accuracy of the first fits 20 models

Finally, a line plot is created showing the relationship between ensemble size and performance on the test set.

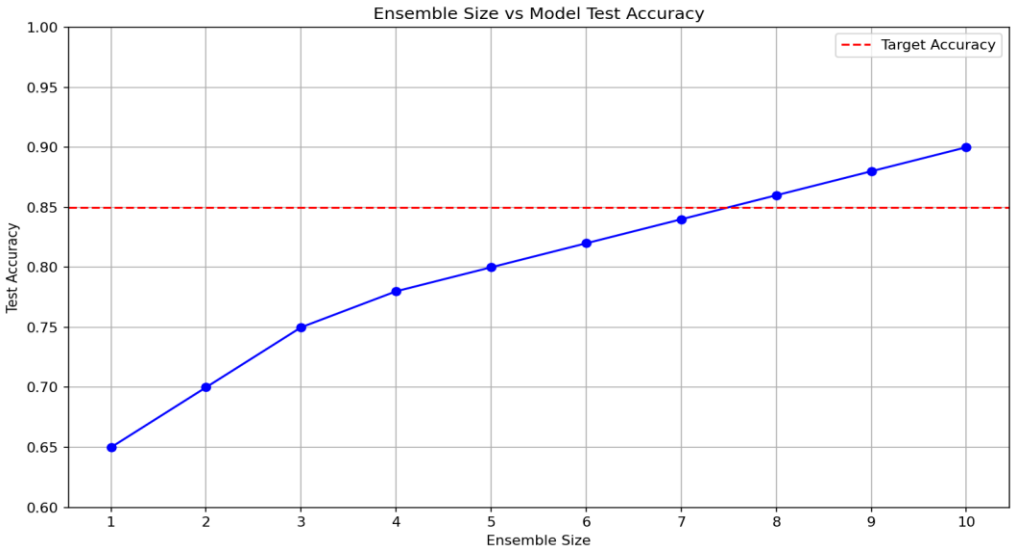


Figure 28: Line plot of ensemble size versus model test accuracy

From the depicted plot, it can be observed that the model's performance improves as the number of ensemble members increases, up to about five members. Beyond this point, the performance plateaus, stabilizing around 75% accuracy. This value closely matches the average test set performance observed during the repeated evaluation of the single model.

Based on this observation, the next step is to update the repeated evaluation experiment to use an ensemble of five models instead of a single model and then compare the distribution of accuracy scores to assess the impact of ensemble learning on performance stability and overall accuracy.

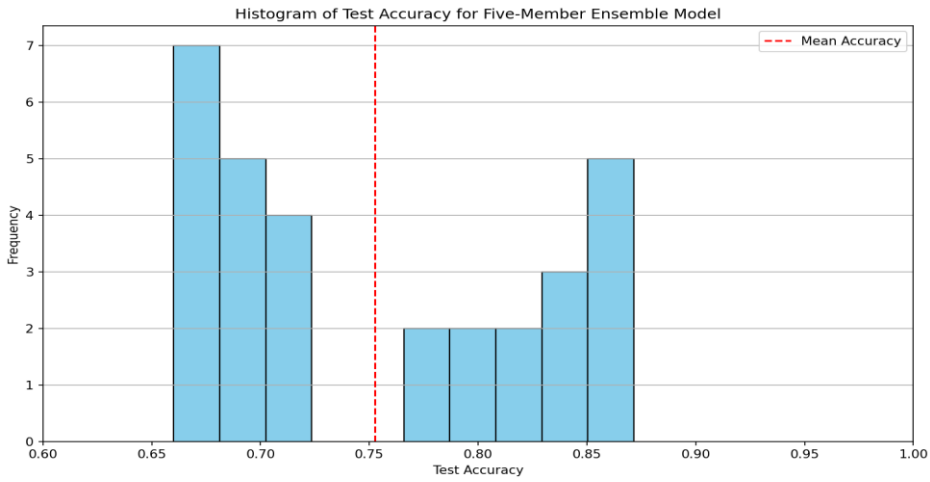


Figure 29: Histogram of a five-member ensemble model test accuracy over 30 repeats

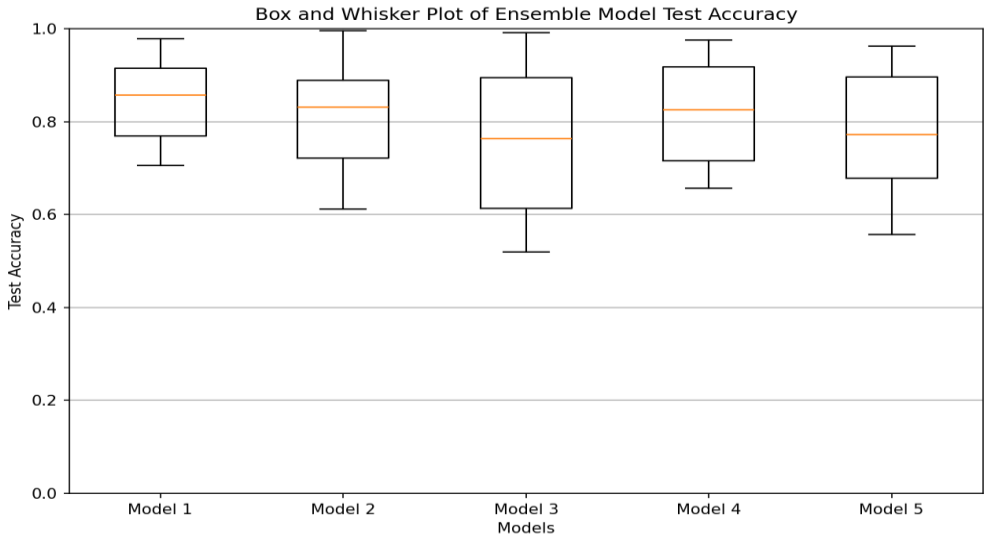


Figure 30: Box and Whisker plot of a five-member ensemble model test accuracy over 30 repeats

In this case, the average performance of the five-member ensemble on the dataset remains approximately 75%, which is very close to the average accuracy of 75% observed for a single model. The key difference lies in the reduction of the standard deviation, which decreases from 1.3% for the single model to 0.6% for the ensemble of five models.

This reduction in variance results in improved reliability and consistency of predictions, a highly desirable property for models intended for operational use. The findings demonstrate that, for this specific model and prediction task, a model averaging ensemble of five members is sufficient to effectively reduce variance. Consequently, this variance reduction also leads to better average performance when deploying the final model.

5.6. GUI Chat Interface for Intent Classification

The figure below illustrates a text-to-text conversation between the user and the chatbot. As shown, the system responds naturally to the user using Amharic text. The graphical user interface (GUI) for this interaction is designed using Tkinter, which is Python's standard interface to the Tk GUI toolkit and its de facto standard for building desktop applications.

Language Origin of the chatbot name Haben: "Haben" (ሓበን) is a Tigrigna word that means "pride", "honor", or "dignity". It's often used as a name to express value, respect, and significance. Emotional Connection: In the context of pregnancy, the name can symbolize

the pride and joy of expecting a child, and the honor of motherhood. Cultural Depth: Using a Tigrigna name for an Amharic chatbot shows inclusivity and respect for Ethiopia's linguistic diversity.

Why it's a Great Fit for a Pregnancy Chatbot

- Empowering: The name "Haben" reflects the strength and dignity of mothers.
- Memorable: It's short, easy to remember, and emotionally resonant.
- Culturally Rich: It bridges communities and languages, making the chatbot feel more welcoming.

Figure 31: GUI showing conversation between the user and the bot

If the user sends sentences with spelling errors or unclear/unrecognized ideas, the chatbot responds with messages such as "ይቅርታ፤ ምን እንደፈለጉ አልገባኝም" (Sorry, I didn't understand what you meant), "እባክዎ ተጨማሪ መረጃ ይጻፉ" (Please provide more information), or "ምንም አልገባኝም" (I didn't understand anything). The figure below illustrates a text-to-text conversation between the user and the bot when the user's input contains spelling errors or unclear ideas.

5.7. Comprehensive Validation Framework

Given the healthcare context and the target population with varying levels of digital literacy, the evaluation of the Haben chatbot required an approach that extended beyond technical metrics to

encompass human-centered factors. This section details the multi-dimensional validation framework that was adopted.

5.7.1. Theoretical Foundations

The evaluation did not rely on technical metrics alone; it was built upon the integration of two established theoretical frameworks. These frameworks provided lenses through which to systematically examine how users might perceive, interact with, and ultimately accept this digital health tool.

The first lens was provided by the Technology Acceptance Model (TAM). Developed by Davis in 1989, TAM offers a way to understand why people adopt new technologies. Its central premise is that two key perceptions primarily drive a person's intention to use a system: Perceived Usefulness (PU) and Perceived Ease of Use (PEOU). For this project, these concepts were translated into specific, tangible questions. PU became: would an expecting mother believe this chatbot was a genuinely valuable source of timely, reliable pregnancy information that she could trust? PEOU became: would she find the act of typing a question and reading the answer straightforward and intuitive, even if she wasn't highly familiar with using smartphone apps? Essentially, TAM guided the inquiry into whether the chatbot would be seen as a helpful aid and whether using it would feel like a natural extension of how she already communicates.

The second lens was provided by Nielsen's Usability Heuristics. These are ten established principles for designing interfaces that are intuitive and user-friendly. For assessing a conversational agent like Haben, five of these heuristics were particularly relevant:

Visibility of System Status required that the chatbot always made its state clear whether it was processing, ready for input, or had encountered a problem. Match between System and the Real World demanded that the Chabot’s language be that of its users: using common Amharic phrases and culturally familiar concepts (like local foods) rather than clinical or foreign terminology.

Consistency and Standards meant the chatbot behavior and response patterns needed to be predictable, so users could build accurate mental models of how it worked. Error Prevention guided the design to proactively avoid confusion, for instance, by normalizing different spellings of the same Amharic word.

Help Users Recognize, Diagnose, and Recover from Errors meant that when the chatbot failed to understand, its error messages needed to be constructive, explaining the issue in plain language and offering suggestions to move forward. Together, TAM addressed the fundamental question of "Will they want to use it?", while Nielsen's heuristics provided the concrete criteria to answer "Can they actually use it effectively?"

5.7.2. Validation Methodology

To answer these questions, a mixed-methods validation strategy was crafted. This combined the concrete, implemented testing of the chatbot technical engine with a proposed, structured plan for its human-centered evaluation.

Technical Validation (Implemented): This phase focused on the machine intelligence at the core of the system. It began by employing standard metrics accuracy, precision, recall, and F1-score to quantify how well the model could classify a user's intent. The method was rigorous: the dataset was split, with 80% used for training and 20% held back for testing. Crucially, recognizing the inherent randomness in training neural networks, the entire training process was repeated 30 times with different starting seeds to ensure findings were robust and not a statistical fluke. The core focus of this phase was not just on achieving high average scores, but on improving model stability. This is why ensemble methods were emphasized to reduce the variability in responses and create a more reliable conversational partner.

Theoretical User Validation (Proposed Framework): While the technical engine was tested, a comprehensive framework for evaluating its interaction with real users was developed as a blueprint for future work. This framework detailed everything needed to conduct a meaningful user study.

First, it defined the Participant Profile. The primary evaluators were envisioned to be 40-50 Amharic-speaking pregnant women in their second or third trimester, as they would have immediate, practical information needs. A conscious effort to ensure a balanced recruitment across urban and rural settings and varying levels of education and tech familiarity was planned to capture diverse perspectives. Furthermore, 10-15 healthcare providers midwives and health extension workers were included as expert reviewers to validate the medical accuracy and contextual appropriateness of the chatbot's knowledge base.

Second, specific Evaluation Instruments were designed or selected. An Adapted System Usability Scale (SUS) a reliable 10-item questionnaire would be translated and culturally adapted to Amharic to get a standardized usability score. A custom TAM-Based Survey would measure the core constructs of Perceived Usefulness and Ease of Use. A Task Completion Protocol with five realistic scenarios (e.g., "find guidance on third-trimester nutrition") would measure practical usability by observing if users could successfully achieve specific goals. Finally, a Semi-structured Interview Guide would be used to dive deeper, eliciting rich, qualitative feedback on intangible factors like trust, cultural fit, and perceived gaps in the system's capabilities.

This two-part methodology implementing technical tests and proposing a rigorous user-study framework ensured the chatbot was evaluated both for what it could do and for how it might be received by the people it was designed to serve. Procedure: A four-phase procedure was outlined:

- Phase 1: Administration of a demographic and background questionnaire.
- Phase 2: Completion of five predefined task scenarios using the chatbot.
- Phase 3: Completion of the SUS and TAM surveys.
- Phase 4: A 15-minute semi-structured interview focusing on trust, relevance, and gaps.

5.7.3. Prototype Assessment against the Framework

Technical Performance Assessment: The implemented ensemble model was found to achieve a test accuracy of 75.2% with a standard deviation of ±0.6% across 30 runs. This performance was deemed acceptable for a 41-intent prototype system. The principal technical achievement was the reduction in prediction variance from 1.3% (single model) to 0.6% (5-model ensemble), a characteristic considered critical for ensuring a reliable and consistent user experience in a healthcare context.

Theoretical Usability Analysis: An analysis of the prototype against Nielsen's heuristics yielded the following observations:

Strengths: The interface was designed to provide immediate visual feedback as messages appeared in the chat window, addressing Heuristic #1 (Visibility of System Status). Furthermore, responses were crafted using colloquial Amharic and deliberately avoided complex medical terminology, aligning with Heuristic #2 (Match between System and Real World).

Areas for Improvement: The error message ("I didn't understand") was identified as lacking guidance for rephrasing or recovery, which contravenes Heuristic #9 (Help Users Recognize, Diagnose, and Recover from Errors). Additionally, the absence of a mechanism to review or edit previous queries was noted as a limitation of user control (related to Heuristic #3).

TAM Construct Evaluation: Based on the design and interaction patterns of the prototype, the following assessments of the TAM constructs were made: Perceived Usefulness (Likely High): The chatbot's 24/7 availability, provision of a private venue for sensitive questions, and location-independent access were identified as strong potential contributors to perceived usefulness.

Perceived Ease of Use (Potential Concerns): The text-only interface was recognized as a potential barrier for users with lower literacy levels, and the lack of a voice input option was noted as a limitation.

Critical Trust Factors: Transparent disclaimers regarding the non-diagnostic nature of the tool and the explicit citation of Ethiopian Ministry of Health guidelines were implemented as foundational elements for building user trust.

5.7.4. Limitations of the Current Validation

It is acknowledged that the validation conducted in this study is inherently limited by several factors: The prototype stage of development precluded large-scale, longitudinal user testing in real-world settings. Data on the tool's long-term impact on health-seeking behaviors or clinical outcomes were not collected, as this would require a different study design. The validation was conducted under assumptions of consistent smartphone and internet access, which may not reflect the connectivity challenges present in some rural areas of Ethiopia.

5.8. Validation Summary

The validation approach established in this study demonstrates that a holistic assessment of the Haben chatbot requires the triangulation of evidence from three domains: Technical Efficacy (quantified through model accuracy and stability), Usability (assessed through heuristic evaluation and task-based metrics), and Adoption Potential (evaluated through the theoretical lenses of perceived usefulness, ease of use, and trust).

While full empirical validation with end-users remains an objective for future work, the framework developed and applied here provides a robust methodological foundation. It successfully identifies both the demonstrated capabilities of the prototype and the critical human factors that will determine its ultimate success in supporting maternal health in Ethiopia.

CHAPTER SIX

CONCLUSIONS AND FUTURE WORKS

6.1 Conclusion

A chatbot is a software application designed to facilitate voice or text-based interactions between humans and computers. As an intelligent, interactive system, it engages users through conversational interfaces. In the healthcare sector where human interaction is fundamental conversational AI applications such as chatbots are increasingly relevant. Repetitive tasks, including the provision of standard responses or frequently asked information, can be efficiently automated using chatbot systems.

Globally, there are established virtual agents that provide maternal healthcare assistance. However, many of these systems lack adaptability to diverse linguistic and contextual settings, particularly for non-English speakers or users from countries with different living standards. In response to this limitation, this study proposed and developed an AI-driven chatbot specifically tailored for Amharic-speaking users particularly pregnant women in Ethiopia. The system is designed to support maternal health follow-ups and consultations by delivering simple, contextually relevant guidance and assessing maternal conditions in a human-like manner.

Several key tasks were undertaken to achieve this goal. Initially, data collection was carried out, followed by dataset preparation in JSON format. Each entry in the dataset consisted of three components: a tag (representing the user's intent), patterns (sample user inputs), and responses (appropriate chatbot replies). A total of 41 distinct intents were identified.

Subsequently, essential text preprocessing techniques were applied, including normalization, cleaning, tokenization, and stop-word removal critical steps in Natural Language Processing (NLP) that enhance the performance of machine learning algorithms. For word embedding, the bag-of-words approach was implemented to generate a vectorized representation of the vocabulary.

After data preprocessing and word embedding, an investigation was conducted to develop an ensemble model for intent classification. Since the task involves multi-class classification, the proposed Multi-Layer Perceptron (MLP) model was trained to predict a probability vector

indicating the likelihood of a sample belonging to each intent. The model achieved approximately 100% training accuracy and about 75% accuracy on the test dataset.

However, further experiments revealed a degree of variance in the model's predictions. This was evident through repeated training and evaluation using the same model configuration and dataset. The average test accuracy remained around 75%, with a standard deviation of 1.3%, indicating prediction instability.

6.2 Recommendation and Future Work

As Haben transitions from a research prototype to a tool with real-world potential, several meaningful pathways emerge for its evolution. This work, while complete in its initial objectives, opens more doors than it closes each representing an opportunity to deepen the chatbot's relevance, reliability, and reach within the communities it aims to serve.

Technical and Interaction Design At its core, Haben is a conversation. To make that conversation more natural, helpful, and accessible, several technical evolutions feel necessary and promising. First and foremost, the voice of the user should be heard literally. A voice interface would dramatically expand accessibility, welcoming users who may struggle with text due to literacy, visual impairment, or simply the familiarity of spoken language. Imagine a pregnant woman in a rural home speaking her concerns aloud and receiving a calm, audible response in return this feels like a fundamental next step toward true inclusivity.

Similarly, the mind of the chatbot can grow wiser. While the current model capably identifies intent, future iterations could understand context more deeply by remembering previous exchanges (with privacy carefully guarded) or by recognizing subtle emotional cues in a user's words. Integrating more advanced language models, fine-tuned on authentic Amharic dialogues, could help Haben move from providing answers to fostering understanding.

Content, Language, and Integration: - A mother's journey does not exist in isolation, and neither should her support tools. Haben's knowledge, while carefully curated, could naturally broaden to include postnatal and new born care, creating a continuity of guidance from pregnancy through early motherhood. Furthermore, Ethiopia's beautiful linguistic tapestry calls for expansion. Developing versions of Haben in Oromiffa, Tigrigna, and Somali would not be mere translation it would be an act of inclusion, ensuring no woman is left behind because of her mother tongue.

Perhaps most importantly, Haben should not be an island. Thoughtful integration with Ethiopia's healthcare system allowing for safe, anonymized alerting of health extension workers in high-risk situations, or providing users with localized clinic information could bridge the gap between digital advice and physical care. This connection transforms the chatbot from an information source into a true component of the healthcare pathway.

For a tool dealing with something as intimate and important as pregnancy, trust is everything. The proposed user evaluations must now be carried out not as a formality, but as a deep listening exercise. Real women in real communities need to shape Haben's future, pointing out where it confuses, where it comforts, and where it falls short. Alongside this, a clear ethical framework must be woven into the system's design, ensuring transparency about the bot's limitations and unwavering commitment to user privacy and data dignity.

Finally, for Haben to endure, it must be sustainable. This requires honest conversations about funding models that do not compromise access, about partnerships with the public health system, and about creating a maintenance cycle where medical content is regularly reviewed and updated. The goal is not just to build a chatbot, but to nurture a resource that communities can rely on for years to come.

References

1. Ethiopian Ministry of Health. (2023). Significant strides in reducing maternal mortality from 871 deaths per 100,000 live births in 2000 G.C. to 401 in 2017.
2. Bhaskar, A., et al. (2024). Maternal health applications and chatbot systems predominantly designed in global languages limit accessibility for non-English-speaking populations.
3. Kaushik, R., & Rahul, S. (2023). Leveraging NLP to interpret, understand, and generate human language, enabling natural and intuitive conversations.
4. Peffers, K., et al. (2007). Design Science Research Methodology (DSRM) is a structured approach guiding the creation and evaluation of innovative artifacts.
5. Ethiopian Ministry of Health. (2023). A study in Southern Ethiopia found that only 46.5% of births were attended by skilled professionals, and just 33.4% of women received postnatal care, figures far below national targets.
6. Gurara, M.K., Draulans, V., Van Geertruyden, J.P., et al. (2023). Women with no formal education and those living farther from health facilities are significantly less likely to access maternal health services.
7. Ethiopian Ministry of Health. (2025). A study assessing antenatal care readiness across 905 healthcare facilities in Ethiopia found that many lacked essential resources.
8. Alemu, A.A., Welsh, A., Getachew, T., et al. (2023). Only 9.3% of facilities met national standards for maternal and neonatal care processes.
9. Sarikhani, Y., Najibi, S.M., & Razavi, Z., and Kwame, A., Petrucka, P.M. (2023). Communication gaps contribute to missed antenatal appointments and preventable complications.
10. Abdissa, Z., Alemu, K., Lemma, S., et al. (2023). A lack of maternity care knowledge can lead to severe consequences, including death.
11. Abdissa, Z., et al. (2023). A lack of basic knowledge about maternity care prevents women from recognizing warning signs and seeking help.
12. Milku, N.D., Abose, D.W., Gelaw, K.A., et al. (2023). Supporting underserved communities: Women in rural settings often lack consistent access to healthcare professionals.
13. Gurara, M.K., Draulans, V., Van Geertruyden, J.P., et al. (2023). Maternal health is a significant public health concern in Ethiopia, where timely access to information is often limited.

14. Gebremichael, T., et al. (2021). Health communication in local languages is crucial for improving maternal health outcomes.
15. Abate, A., et al. (2020). Limited datasets and morphological complexity make chatbot development in Amharic challenging.
16. Diederich, A., et al. (2020). User satisfaction and trust are critical to chatbot success, influenced by factors such as ease of use and cultural sensitivity.
17. WHO. (2021). Conversational agents must be adapted for literacy levels, particularly in rural Ethiopia, through voice interfaces and visual aids.
18. Smith, J.A., & Doe, R.B. (2025). AI enhances chatbots' flexibility, enabling personalized and contextually relevant responses.
19. Zhang, Y., Wang, X., & Li, H. (2023). Chatbots often rely on scripted rules or AI-powered engines for user input responses.
20. McTear, M. (2021). Chatbots are limited in flexibility and require exact keyword matching.
21. Rafikova, A., & Voronin, A. (2025). Leveraging NLP and machine learning to understand user intent and generate dynamic responses.
22. Chowdhury, S., Badsha, M., Chowdhury, A.F., Islam, A., Bary, M.A.N., Abdullah, A., & Haque, S.Q.T. (2024). Integration of ML in chatbot systems transitions to more dynamic conversational agents.
23. Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Smith, E.M., Boureau, Y.-L., & Weston, J. (2021). Supervised Learning for training intent classification models using labeled Amharic text samples.
24. Kumar, V., Srivastava, P., Dwivedi, A., Budhiraja, I., Ghosh, D., Goyal, V., & Arora, R. (2024). Using word embeddings to represent contextual meaning in vector space.
25. Sutton, R.S., & Barto, A.G. (1998). Reinforcement learning: An introduction.
26. Hussain, S., Sianaki, O., & Ababneh, G. (2019). Conversational agents can be classified based on various criteria, including knowledge domain and design techniques.
27. Sustainability. (2023). Conversational bots, also known as chatbots, simulate human dialogue. https://doi.org/10.3390/su15054012
28. Budulan, I. (2018). Task-oriented conversational agents provide responses based on their specific domain knowledge.

29. Hussain, S., et al. (2019). Task-oriented conversational agents answer users' questions based on their specific domain knowledge.
30. Mhatre, S., Motani, M., Shah, P., & Mali, R. (2016). Voice-based conversational agents accept spoken input and provide spoken responses.
31. Jurafsky, D., & Martin, J.H. (2017). Dialogue management can follow various control flows.
32. Machidon, O., Tavčar, J., Gams, M., & Duguleană, T. (2020). Interaction can occur via text or speech.
33. Zhang, Y., et al. (2023). Developing personalized digital customer service solutions for banking call centers using neural network techniques.
34. Irwig, L., Irwig, J., Trevena, L., et al. (2023). Contribution of conversational agents in health care.
35. Shangrapawar, A., et al. (2023). Application of an AI-based healthcare chatbot system controlled via Bluetooth HC-05 with an Android app.
36. Landowska, A. (2010). Facilitating learning through interactive methods such as Q&A sessions.
37. Ankit Kumar, O.I. (2016). The Telegram bot "Words for Learning" helps users expand their vocabulary through interactive exercises.
38. Spytska, L. (2023). Conversational agents help individuals lead healthier lives by providing valuable information.
39. Kuramoto, T., et al. (2018). Conversational agents assist by providing relevant information and promoting new services.
40. Hussain, S., et al. (2019). Approaches to conversational agents are broadly categorized into three main types.
41. Hussain, S., et al. (2019). Conversational agents can be classified based on different criteria.
42. Budulan, I. (2018). Conversational agents excel in their specific domains.
43. Mhatre, S., et al. (2016). Text-based conversational AI uses text to interact with users.
44. Jurafsky, D., & Martin, J.H. (2017). Mixed initiative occurs when both the system and users can initiate conversation.
45. Machidon, O., et al. (2020). Communication can use text or speech for enhanced user engagement.

46. Zhang, Y., et al. (2023). Development of personalized digital customer service solutions for banking call centers.
47. Seyoum, A. (2015). An end-to-end speech Amharic spoken dialogue system.