Addis Ababa University

Department of Linguistics

M.Sc. In Computational Linguistics

Machine Learning Models for Amharic Clinical Chatbot

Bezayt Yewondwossen Awlachew

A Thesis submitted to the Department of Linguistics in partial fulfillment of the requirement of the Degree of Master of Science in Computational Linguistic

December 2024

Addis Ababa, Ethiopia

Addis Ababa University

Department of Linguistics

Bezayt Yewondwossen Awlachew

This is to certify that the thesis prepared by Bezayt Yewondwossen Awlachew, titled*: Machine Learning Models for Amharic Clinical Chatbot* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computational Linguistics complies with the regulations of the University and meets the accepted standards concerning originality and quality.

Signed by the examining committee.

Name	Signature	Date
Advisor Demeke Asres (PhD)	_____	_____
Examiner (PhD)	_____	_____
Examiner (PhD)	_____	_____

Abstract

This master's thesis focuses on experimenting with and evaluating deep learning techniques alongside various classical machine learning models for the development of an Amharic chatbot. The models examined include decision trees, support vector machines, random forests, logistic regression, Naive Bayes, multi-layer perceptron , and Bi-LSTM. Additionally, the research aims to identify the most effective feature extraction method. The methodology encompasses data collection, preprocessing, feature extraction, model training, and evaluation, with accuracy serving as the primary performance metric.

The results demonstrated that using TF-IDF with hyperparameters 'n_estimators': 100, 'min_samples_split': 2, the Random Forest, SVM, and Decision Tree models achieved an accuracy of 0.9286, while Naive Bayes and Logistic Regression had accuracies of 0.6964 and 0.7679, respectively. Using CountVectorizer with the same hyperparameters, the SVM, Naive Bayes, and Logistic Regression models achieved the highest accuracy of 0.9286. The Decision Tree model followed with an accuracy of 0.8929, while the Random Forest model had an accuracy of 0.8214. Precision, recall, and F1 scores were also evaluated, with SVM, Naive Bayes, and Logistic Regression models showing consistently high performance across these metrics. These findings suggest that the choice of feature extraction technique and hyperparameter tuning significantly impact the performance of certain models. Interestingly, the MLP Classifier model outperformed the other models when using the TF-IDF feature extraction technique, achieving an accuracy of 0.9643, In General, from the experiment, we observed that the Bi-LSTM model performance is lower than the other models.

Table of Contents

Tables and Figures

Acknowledgments

ሁሉን ያደረገ እግዝያብሔር ይመስገን!

The research presented was carried out in the Department of Linguistics at Addis Ababa University. Accordingly, the University, departments, health organizations, and many people were involved and contributed to the perspectives contained in this thesis and I am highly indebted to them. I would also like to express my special thanks to the Ministry of Health, different governmental and private hospitals, and clinics, and to all the research participants for their valuable time and insights, which formed the core of this thesis.

First and foremost, I am deeply indebted to my esteemed advisor, Dr. Demeke Asres, for his unwavering mentorship, instant feedback, and dedication throughout this thesis research. Dr. Demeke's extensive knowledge, research insight, and commitment to academic excellence have been instrumental in shaping the direction and quality of this thesis.

Without the significant backing of my office (Plan International- Ethiopia), line manager, coworker (friends) work on this thesis would not have been possible to any extent. I am deeply indebted for all the support they have provided throughout the course of my work over three years.

To my families, friends, and colleagues, I extend my heartfelt gratitude for their unwavering support, encouragement, and understanding throughout this journey. I would like to thank Dr.Yewondwossen Awlachew for his unreserved contribution to this research through his active involvement, insightful perspectives, and sharing his expertise. My gratitude must also be extended to my sister, Dr. Wude (Internist), who despite focusing on her internship tasks, always made herself available for acquainting me with medical procedures and terminologies and fully engaged in the data collection. What more needs to be said, other than we are there, together!

A final note of acknowledgement must be going to Tilaye, my beloved Grandma, for her inspiration and encouragement in my entire life as well as my study. It is for her that this thesis is heartily dedicated, along with my deepest respect and gratitude.

List of Acronyms

AI - Artificial Intelligence

Bi - LSTM- Bidirectional Long Short-Term Memory

DL - Deep Learning

NLP - Natural Language Processing

MLP - Multiple Layer Perceptron

RNN - Recurrent Neural Network

SVM - Support Vector Machines

ML - Machine learning

TF - IDF- Term Frequency- Inverse Document Frequency

ANN - Artificial Neural Network

Chapter 1 Introduction

1.1 Background

Ethiopia's healthcare system faces significant challenges, including a severe shortage of medical professionals, particularly in rural areas. This shortage has led to difficulties in providing timely and effective healthcare services to the population. The limited resources and lack of medical experts exacerbate the issue, making it essential to explore innovative solutions to improve healthcare accessibility and quality [34].

Clinical chatbots are revolutionary digital tools that utilize artificial intelligence and machine learning models to provide personalized healthcare recommendations, symptom analysis, and medical condition information. With their 24/7 availability and user-friendly interfaces, these chatbots have the potential to transform the way people access healthcare information and support [9].

Ethiopia is a linguistically diverse country with over 80 languages spoken across various regions. Amharic, Oromo, and Tigrigna are three major languages spoken in the country, each with its unique usage and significance. Designing a linguistically grounded system can enhance the accessibility and accuracy of healthcare services for Amharic-speaking individuals [52].

This master’s thesis aims to experiment with and evaluate various classical machine learning classification models and deep learning models in the development of a Text-based clinical chatbot in Amharic language. By achieving these objectives, this study seeks to improve the lexical resources of the Amharic language in the healthcare domain and identify the best fit model and feature extraction.

1.2 Motivation

In a country with a population of over 110 million, Ethiopia faces significant challenges in providing adequate healthcare access. The nation's physician shortage, with only around 5,000 physicians serving the entire population, underscores the urgent need for innovative solutions. According to the World Health Organization, Ethiopia's physician-to-population ratio of 0.45 physicians per 10,000 people is alarmingly low, less than a fifth of the WHO African region average. This places Ethiopia among the countries with the lowest physician density in the world, trailing far behind its neighbors like South Africa, Kenya, and Nigeria [6, 34].

Given this substantial healthcare workforce challenge, the motivation to explore cutting-edge technological solutions, such as Amharic-powered clinical chatbots, becomes paramount. These conversational AI agents, fluent in the local language, hold the promise of bridging the gap between the limited number of physicians and the vast population in need. The strength of these Amharic chatbots lies in their linguistic fluency and their ability to provide culturally relevant and tailored recommendations. By building trust and fostering a sense of connectivity that transcends physical distance, digital interventions can revolutionize the way Ethiopians access and experience medical care [3, 4, 22].

Moreover, the quest to develop a clinical chatbot that can accurately and efficiently assess patients' symptoms and provide effective recommendations drives the motivation to experiment with various classical machine learning and deep learning models. The goal is to achieve optimal performance and accuracy in solving real-world problems. By exploring the complexities of the Amharic language, particularly in the healthcare domain, this research can contribute invaluable linguistic data and insights, thereby strengthening the language's digital presence and accessibility.

1.3 Problem Justification

Amharic is the second most widely spoken Semitic language after Arabic, serving as the working language of Ethiopia and spoken by millions of people. However, the availability of technology and AI applications in Amharic is limited compared to widely spoken languages such as English and Mandarin Chinese [6, 13, 52].

This text-based system can improve efficiency by enabling remote consultations and potentially reducing unnecessary travel and wait times. Furthermore, it can bridge the language barrier between Amharic speakers and an English-centric healthcare system. In conclusion, developing a text-based Amharic clinical chatbot addresses the language accessibility gap, allowing Amharic speakers to interact with technology in their native language, access information, and receive healthcare support more effectively. This also involves creating datasets specific to the language [13].

1.4 Problem Statement

Despite the potential benefits of chatbots in healthcare, there is a lack of comprehensive comparisons between machine learning classification models and deep learning models in the context of clinical chatbots. Existing research has primarily focused on individual models and feature extraction, with model performance compared in a more general NLP context [2, 3, 4, 6]. To bridge this gap, there is a need for a thorough experiment, evaluation, and comparison of popular classical machine learning and deep learning models for text classification within the specific domain of healthcare clinical chatbots. This research aims to address this gap by experimenting with and evaluating various popular machine learning classification models and deep learning models to determine their performance and suitability for enhancing the accuracy and effectiveness of an Amharic text-based clinical chatbot.

This research contributes to advancing the field of Amharic text-based clinical chatbots by guiding the selection of the most appropriate model(s) to enhance the accuracy and effectiveness of the chatbot system in providing healthcare recommendations. The following research questions are addressed:

- Which feature extraction techniques have a significant impact on the performance of the model?
- Which machine learning model performs best in developing an Amharic clinical chatbot?

1.5 Research Objectives

1.5.1 General Objective

This study aims to experiment with and evaluate various classical machine learning classification models and deep learning models in the development of a Text-based clinical chatbot in the Amharic language. The goal of this study is to identify the best-fit model for Amharic language clinical chatbot development, with a focus on accurately classifying symptoms and providing effective prediction in a domain-specific healthcare vocabulary and terminology.

1.5.2 Specific Objectives

- To Investigate the impact of feature extraction techniques on the performance of machine learning model in Amharic clinical chatbots.
- To identify the best-fit model for Amharic clinical chatbot development.

1.6 Significance of the Study

The significance of experimenting with different popular classical machine learning classification models lies in their potential to enhance the accuracy and performance of clinical chatbots, guide model selection, and contribute to scientific knowledge in the field. By conducting a comprehensive evaluation and comparison of different classification models, this research enables a comparison of their performance in accurately classifying user inputs, identifying the models that exhibit higher accuracy, precision, recall, and F1 scores.

The findings of this research have practical implications for the development of clinical chatbots, as the identification of the most effective classification model(s) helps to guide developers and practitioners in selecting the appropriate model for accurate text classification. This, in turn, enhances the overall performance and user experience of healthcare chatbots. Furthermore, this research contributes to the existing body of knowledge in the field of clinical chatbot development, providing insights into the strengths, weaknesses, and applicability of each model in the healthcare domain.

1.7 Scope and Limitations

This study focuses on experiments and evaluates popular classical machine learning and deep learning models in the development of a text-based Amharic Clinical chatbot [23]. The chatbot is designed to interact with patients through a conversational interface, gathering symptoms to provide a recommendation that can understand and process Amharic language inputs, and generate tentative diagnosis, emphasizing the functionalities of symptom- based diagnosis.

The Clinical chatbot is targeted at providing tentative diagnosis for common, non- emergency medical conditions. It does not cover specialized or complex medical issues that require laboratory, clinical evaluation, etc. It only supports the Amharic language and is not designed to handle input or provide responses in other local languages. It only accommodates text-based interactions and is not going to support voice input, image recognition, or other multimedia modalities.

1.8 Thesis Structure

The rest of the thesis chapter is organized as follows. In Chapter Two, we examine the various text-based healthcare chatbot systems that have been developed. We study their design principles, functionalities, and the technological approaches employed to deliver accurate and reliable medical assessments. Additionally, we explore the challenges and considerations specific to the implementation of these systems in the context of low-resource languages like Amharic in Healthcare. By organizing Chapter 2, we aim to provide a comprehensive understanding of the literature related to text-based clinical systems, machine learning classification models, and the existing research in low-resourced languages. This lays a solid foundation for the subsequent chapters, enabling us to go deeper into the methodology, experimental results, and discussion of our research.

Chapter 3 outlines the methodology employed in this research study. This chapter details the step- by-step process undertaken to develop and evaluate the text-based clinical chatbot system in the case of Amharic language; we also aim to provide a transparent and systematic approach to developing and evaluating the text-based clinical system for low-resourced languages. This ensures the reliability and validity of our research findings and contributes to the overall success of the study.

Chapter 4 presents the experimental results and analysis of the developed text-based clinical chatbot for low-resourced languages. This chapter focuses on evaluating the performance and effectiveness of various machine learning and deep learning models used & conducting a comprehensive comparative analysis of the experimental results obtained from the different models. We compared the performance of Naïve Bayes, decision trees, support vector machines, random forests, Logistic regression, multi-layer perceptron, and Bi-LSTM in terms of various metrics. By organizing Chapter 4 in this manner, we aim to provide a detailed evaluation and analysis of the experimental results obtained from the different models utilized in the text- based clinical chatbot. This enables us to gain insights into the performance and effectiveness of each model and inform the discussion and conclusions of our research.

Chapter 5 goes through the discussion of the research findings, interpretation of the results, limitations, and challenges encountered during the study, as well as potential future directions for further exploration. This chapter contributes to the overall understanding of the implications and significance of the research study and lays the foundation for future research endeavors in the field of text-based clinical chatbots for low-resourced languages.

Chapter 6 provides a concise and comprehensive conclusion to the research study on the development of a text-based clinical chatbot for low-resourced languages. This chapter summarizes the key findings of the study, highlights the contributions made to the field, and discusses the implications and recommendations derived from the research, it also serves as a final reflection on the research journey and provides a solid foundation for future research and advancements in the field of text-based clinical chatbots for low-resourced languages.

Chapter 2 Literature Review

2.1 Introduction

Machine learning and deep learning are both subfields of artificial intelligence (AI) concerned with training computers to learn from data [41]. However, they differ in their approach and capabilities. Classic machine learning models rely on statistical methods and techniques like linear regression, decision trees, or support vector machines [8]. These models require explicit feature engineering, where domain knowledge is used to identify and extract relevant features from the data that the model can learn from. This feature engineering step plays a crucial role in the performance of traditional ML models. Additionally, ML models are typically less complex and require less computational power compared to deep learning models. An advantage of some ML models is their interpretability, allowing us to understand the features that contribute most to their predictions [41]. This can be valuable for debugging and gaining insights into the model's decision-making process.

Deep learning, on the other hand, is a subfield of ML inspired by the structure and function of the human brain. It utilizes artificial neural networks (ANNs) with multiple layers of interconnected nodes and have the advantage of automatic feature learning, where they can discover relevant features directly from the data through the training process [40]. This eliminates the need for manual feature engineering, which can be a time-consuming and domain-specific task. Deep learning models can be very complex with many layers and millions of parameters, allowing them to model intricate relationships in data. However, this complexity comes at the cost of high computational needs for training and potentially lower interpretability. In essence, machine learning provides a more traditional, interpretable approach with less computational burden, while deep learning offers a powerful, data-driven approach for complex problems but can be

computationally expensive and less interpretable. The choice between these approaches depends on the nature of the problem and the available data [36,39,42].

2.2 Chatbots for Various Applications

Chatbots, conversational AI agents, have rapidly evolved in recent years, transforming user interaction experiences across various industries. Their ability to automate tasks, provide information, and simulate human conversation has been revolutionized by advancements in Natural Language Processing (NLP), machine learning, and deep learning [43,44,45]. This literature review explores the development and evaluation of chatbots, examining their strengths and limitations across diverse domains.

At the core of chatbot functionality lies NLP, which enables chatbots to understand user intent, extract relevant information from queries, and generate coherent responses [42]. The techniques have been extensively explored in the field of NLP, with a focus on developing more accurate and efficient methods for understanding and generating human language.

Machine learning (ML) and deep learning (DL) play a significant role in chatbot development, studies have explored the use of supervised learning models like decision trees and support vector machines for intent classification. Deep learning models, particularly recurrent neural networks (RNNs) and transformers, have shown promise in handling complex language and generating more natural responses. These models have been widely adopted in the field of NLP and have led to significant improvements in chatbot performance [8 ,9,21].

Chatbots offer a versatile technology with applications in various domains. In the field of customer service, chatbots can serve as efficient customer service representatives, answering frequently asked questions and resolving basic issues. In education, chatbots can personalize learning experiences, provide feedback, and answer student queries. In healthcare, chatbots can offer support for patients, answer medical questions, and guide them towards appropriate care [28, 32].

Evaluating chatbot performance is crucial, metrics like accuracy, precision, recall, and F1-score assess how well the chatbot classifies user requests and responds accurately. These metrics provide valuable insights into the strengths and limitations of chatbots and can be used to improve their performance and effectiveness.

2.3 Related work

In the global context, chatbots are now essential in several industries, including e-commerce, virtual assistants, customer service, and healthcare. This field's progress is characterized by continuous study and development, with an emphasis on augmenting conversational skills, comprehending user intent, and refining overall user experiences. This review of the related work focuses on chatbot development specifically for the healthcare domain, where limitations and difficulties must be balanced against the potential advantages of better information access and user engagement. This highlights the importance of well-thought-out design and seamless integration with current healthcare systems [45,46,47].

2.3.1 Chatbot in Gujarati, Hindi, and English

A study describes a multilingual healthcare chatbot application that uses Natural Language Processing (NLP) to prioritize disease diagnosis based on user-input symptoms. The chatbot functions in Gujarati, Hindi, and English to meet the varied linguistic requirements of rural India, when machine learning models like the Random Forest Classifier are used, the system produces the best disease prediction outcomes with 98.43% accuracy. The application takes a thorough approach, choosing the most pertinent response from its knowledge library in response to user queries by using TF-IDF and Cosine Similarity models. By addressing linguistic differences through multilingual capabilities, the chatbot system demonstrates adaptability in culturally different environments, leading to significant advancements in healthcare diagnoses [35].

The other study presents an intriguing study on a healthcare chatbot that harnesses the power of AI and Machine Learning. The chatbot primarily focuses on disease diagnosis and adopts the decision tree model to analyze user responses to queries. Operating through a question-and-answer framework, the chatbot engages users in a conversation to gather information about their symptoms and medical history. By employing the decision tree model, the system uses this information to make informed decisions and provide accurate diagnoses. One notable aspect of the research is the emphasis on utilizing a dataset known for its enhanced factual richness. This dataset likely contains a wide range of medical information and real-world scenarios, enabling the chatbot to draw upon a comprehensive knowledge base and deliver more precise diagnoses. The decision tree model which is used to develop this powerful chatbot tool allows the patient to make structured decisions based on a series of yes-or-no questions. By branching out through different paths based on patients' responses, the chatbot can navigate through the decision tree and arrive at an accurate diagnosis efficiently [32].

The other research document titled "K-Bot Knowledge Enabled Personalized Healthcare Chatbot" provides valuable insights into a specialized chatbot application designed specifically for the medical domain. Published in the IOP Conference Series Materials Science and Engineering in 2021, the paper highlights the chatbot's ability to identify user symptoms and predict potential diseases, as well as recommend appropriate specialized physicians. The chatbot utilizes the Decision Tree Model, a powerful tool for structured decision-making. By analyzing user symptoms and applying the model, the chatbot can make accurate disease predictions with associated confidence levels. Moreover, it goes a step further by recommending suitable

specialists based on the predicted diseases. The "K-Bot" chatbot, as presented in this publication, demonstrates the potential of AI-driven chatbot applications in the healthcare sector. By combining symptom identification, disease prediction, and specialist recommendations, it offers a comprehensive solution that benefits both users and healthcare providers [29].

2.3.2 Chatbot in Chinese, Malay, Tamil, Filipino, Thai, Japanese, French, Spanish, and Portuguese

The paper entitled “Development and testing of a multilingual Natural Language Processing-based deep learning system in 10 languages for COVID-19 pandemic crisis A multi-center study". It details how artificial intelligence and natural language processing is being used to enhance healthcare delivery during the COVID-19 pandemic is published in the journal "Frontiers in Public Health”. The article describes a multilingual NLP-based deep learning system whose goal is to create a chatbot called DR-COVID that can produce precise multilingual answers to medical queries relating to Covid-19. Chinese, Malay, Tamil, Filipino, Thai, Japanese, French, Spanish, and Portuguese were among the languages in which it was tested [35].

2.3.3 Chatbot in Bilingual Afaan Oromo and Amharic Languages

Moving from the global perspective to a more localized context in Ethiopia, researchers have explored the development of chatbot applications for diverse purposes. Designing and Developing Bi-Lingual chatbot to assist Ethio-Telecom customers. The model is developed by using tflearn DNN model and keras Sequential model. Then compared both models based on accuracy metrics and scored 80.29% and 86.13 % respectively [5].

2.3.4 Chatbot in Afaan Oromo

In the study titled "Developing Afaan Oromo Chatbot for HIV/AIDS Prevention and Care Counseling Using Deep Learning Approaches", the researchers focused on the development of intelligent conversational chatbots specifically for HIV/AIDS prevention and care counseling in Afaan Oromo language. The objective was to utilize deep learning techniques to create an effective chatbot system. To accomplish this, a dataset consisting of question-and-answer pairs was collected by filtering frequently asked questions from various healthcare organization websites and national comprehensive HIV prevention, care, and treatment guidelines. The dataset was prepared in JSON format and underwent appropriate data preparation steps before being used for model training. The researchers employed the word2vec word embedding method as a feature extraction technique. For the model design, several deep learning architectures were implemented, including convolutional neural network (CNN), long short-term memory (LSTM), bidirectional long, short- term memory (BiLSTM), and bidirectional gated recurrent unit (BiGRU). These models were trained using the prepared dataset, and accuracy rates of 92.11%, 93.8%, 95.27%, and 94.4% were achieved, respectively, showcasing the effectiveness of the proposed approaches. The performance of the developed chatbot system was evaluated through human evaluation and user acceptance testing. The results of these evaluations demonstrated high acceptance and positive outcomes in terms of both system acceptance and performance. This study provides valuable insights into the development of intelligent chatbot systems for HIV/AIDS prevention and care counseling in Afaan Oromo. The achieved accuracy rates and positive evaluation results highlight the potential of deep learning approaches in creating effective conversational agents for healthcare-related purposes [10].

2.3.5 Chatbot in Amharic Language

In a publication titled "Amharic Dialogue Based Expert System on Pregnancy", discusses ongoing research endeavors involving the creation of a text-based virtual maternity assistant chatbot tailored to the Amharic language. This initiative aims to address the specific needs of pregnant women in Ethiopia, providing advice and consultations through ensemble learning models. The data collection was performed for a single domain which is about pregnant women. The interview was the way they used to get the required data. Six pregnant were selected randomly and they were interviewed to get information about the feeling they had, symptoms seen, their behavioral changes, the way they were feeding and others & they also consulted physicians and nutritionists. Additionally, the authors highlighted a concerted effort to develop a comprehensive speech-based conversational AI system for healthcare, focusing on the under-resourced Amharic language. The localized approach extends to implementing Amharic text-based virtual maternity assistant chatbots, emphasizing text and speech-based solutions to enhance healthcare accessibility and communication for pregnant women in Ethiopia. A total of 560 text and audio data is prepared. The proposed system achieved 92% accuracy [2].

Another thesis titled "Designing and Implementing Amharic Text-Based Virtual Maternity Assistant Chatbot Using Ensemble Learning Models", focuses on developing a virtual maternity assistant chatbot that aids and provides information to expectant mothers in the Amharic language. The chatbot system described in the thesis incorporates ensemble learning models, which are a combination of multiple machine learning models working together to improve performance. These models are trained on labeled data to classify or predict user intents, enabling the chatbot to understand user queries and provide appropriate responses. To facilitate user interaction, the chatbot employs a GUI (Graphical User Interface) chat interface, allowing expectant mothers to

input text-based queries and receive relevant information. The system undergoes various text preprocessing tasks, including normalization, cleaning, tokenization, and stop word removal. These preprocessing steps help transform the user's text input into a format suitable for further analysis. Based on the predicted intent, the system retrieves a relevant text response from its knowledge base or generates a response dynamically that scores 67.3% this is very close to the average of 67.2% seen for the single model. The important difference is the standard deviation shrinking from 1.3% for a single model to 0.6% with a five-member ensemble. The researcher refers to different sources such as published books and articles written by experts in the field. Two notable sources referenced by the research are "እርግዝና እና አርባዎቹ ሳምንታት" and "እርግዝና እና የህጻናት እንክብካቤ". Furthermore, the researcher also referred to repositories of maternity healthcare information from notable institutions such as Jimma University Specialized Hospital. To gather a comprehensive understanding of the subject, the researcher also considered insights from privately owned maternity healthcare facilities. These facilities likely provided practical perspectives and real- world experiences related to maternal healthcare. In addition, the researcher sought input from registered health service providers like "ዶክተር አለ 8809" to contribute the overall knowledge base of the research. By drawing upon a diverse range of sources, including expert publications, institutional repositories, and consultations with healthcare providers, the researcher ensured a comprehensive and well-informed approach to the topic of maternal healthcare. Overall, the thesis presents a detailed framework for designing and implementing an Amharic text-based virtual maternity assistant chatbot. By employing ensemble learning models, preprocessing techniques, and word embedding [13].

2.3.6 Gap Analysis

The current research in Amharic language chatbots has made significant progress in developing effective systems using classical machine learning and deep learning models, such as Bi-LSTM models. However, there are several gaps in the current research that need to be addressed. Most research has been focused on specific domains, such as HIV/AIDS or pregnancy/maternity healthcare, leaving a gap in the development of more general healthcare chatbots that can address a broader range of medical inquiries and concerns. There is also a need for a comprehensive comparison of different feature extraction techniques and popular classical machine learning models to determine which approaches are most effective for Amharic language chatbots and there is currently no experiment that demonstrates the use of ensemble modeling, where different classical machine learning models are combined based on their individual performance, to improve the overall performance of the chatbot.

2.4 Classical Machine Learning Classification Models

Machine learning classification models have revolutionized various domains by enabling automated decision-making processes [11,21,23,41]. This literature review aims to provide a comprehensive overview of the most prominent machine learning classification models, their characteristics, strengths, weaknesses, and applications. By examining a wide range of studies, this review aims to shed light on the advancements made in classification models and their contributions to diverse fields.

2.4.1 Decision Trees

Decision tree is a type of popular supervised machine learning models for classification tasks due to their simplicity and interpretability. In low- resourced language settings, decision trees have been applied successfully for different areas such as sentiment analysis, document classification, and named entity recognition. They have demonstrated promising results, particularly when combined with feature engineering techniques that leverage linguistic and contextual information. The models facilitate classification by partitioning data based on features. They excel at handling categorical and numerical data, and their hierarchical structure allows for easy visualization. However, decision trees are prone to overfitting and struggle with complex relationships in the data. The core idea behind decision trees is to split the input space into smaller, more homogeneous regions by making a series of decisions based on the input features. Each internal node in the tree represents a decision, and the leaf nodes represent the final classifications or regression outputs. One of the key advantages of decision trees is their interpretability. The tree structure provides a clear and intuitive visualization of the decision-making process, making it easier for users to understand how the model arrives at its predictions. This interpretability is particularly valuable in domains where model transparency is important, such as healthcare, finance, and regulatory applications [8,32].

In the healthcare domain, decision trees have been applied to a variety of tasks, including disease diagnosis, risk prediction, and treatment recommendation [32,39]. Furthermore, decision trees have been widely used in text classification tasks, such as sentiment analysis, spam detection, and document categorization. In these applications, the input features are typically derived from the text data, such as word frequencies or n-grams, and the model predicts the class label of the input text [1,24].

2.4.2 Support Vector Machines

Support vector machine (SVM) is a powerful model that separates data points by constructing hyperplanes in a high-dimensional space. It is effective in handling both linearly separable and non-linearly separable data. SVMs provide robust generalization capabilities, but they can be computationally expensive for large datasets. It has been widely used for classification in low-resourced languages due to their ability to handle high-dimensional data and nonlinear relationships. SVMs have been employed for tasks such as part-of-speech tagging, sentiment analysis, and text categorization. However, the performance of SVMs in low-resourced languages heavily relies on the availability of annotated data and the selection of appropriate features and kernel functions. SVMs have been successfully applied to a wide range of problems, including disease diagnosis, risk prediction, and treatment recommendation. For example, SVMs have been used to diagnose various types of cancer, predict the risk of cardiovascular disease, and assist in the diagnosis of psychiatric disorders [33, 44].

2.4.3 Random Forests

Random forests are ensemble learning models that combine multiple decision trees to improve accuracy and reduce overfitting and have gained popularity in low-resourced language settings due to their ability to handle noisy and imbalanced datasets. By using feature subsets and bootstrap aggregating, random forests provide robustness and enhanced predictive performance. These models have been successfully used for tasks such as named entity recognition, sentiment analysis, and text classification. The ensemble nature of random forests allows them to capture complex relationships and improve classification accuracy, making them suitable for low- resourced language scenarios [36, 41].

2.4.4 Naive Bayes

The Naive Bayes model is a popular machine learning model for classification tasks, particularly in text classification and natural language processing (NLP) applications. The Naive Bayes model is based on Bayes' theorem, which provides a way of calculating the posterior probability of a class given the evidence (input features). The model assumes independence between the input features, simplifying the posterior probabilities' calculation. Despite this simplifying assumption, the Naive Bayes classifier has been shown to perform well in many real-world applications, often outperforming more complex models. One of the key advantages of the Naive Bayes model is its computational efficiency, the model can be trained quickly, and the classification process is also fast, making it suitable for applications where real-time predictions are required [18,31, 35,36].

In the text classification domain, the Naive Bayes model has been widely used for tasks such as spam detection, sentiment analysis, and document categorization. The model's ability to handle high-dimensional feature spaces, such as those encountered in text data, has contributed to its success in these applications [24,25,27]. In the context of healthcare and medical applications, the Naive Bayes model has been applied to various tasks, such as disease diagnosis, predicting patient outcomes, and detecting adverse drug reactions [19, 21,22]. The model's interpretability and ability to handle uncertain or incomplete data make it a suitable choice for these domains.

2.4.5 Logistic Regression

Logistic Regression is a widely used machine learning model for binary and multi-class classification tasks. It is a statistical model that predicts the probability of an outcome (dependent variable) based on one or more independent variables (predictors). Unlike linear regression, which is used for predicting continuous outcomes, logistic regression is specifically designed for predicting categorical outcomes [1,17]. The logistic regression model uses the logistic function, also known as the sigmoid function, to map the input variables to a probability value between 0 and 1. This probability value can then be used to make a classification decision, such as assigning an instance to a particular class [15,16]. One of the key advantages of logistic regression is its interpretability. The model provides coefficients for each input variable, which can be used to understand the relative importance and direction of the relationship between the predictors and the outcome. This makes logistic regression a popular choice in applications where model interpretability is crucial, such as in healthcare and finance [8, 11,26,30].

Furthermore, logistic regression has been successfully applied to text classification tasks, such as sentiment analysis, spam detection, and document categorization. In these applications, the input variables typically consist of textual features, such as word frequencies or n-grams, and the model predicts the class label (e.g., positive or negative sentiment, spam or non-spam) of the input text [1,18]. One of the key considerations in using logistic regression is the assumption of linearity between the predictors and the log-odds of the outcome. If this assumption is violated, the model's performance may be impacted. To address this, researchers have developed extensions and variations of the logistic regression model, such as polynomial logistic regression, which can handle non-linear relationships [17,50]. Additionally, logistic regression can be susceptible to overfitting, especially when dealing with high-dimensional data or a large number of predictors.

Techniques like regularization, feature selection, and cross-validation are often employed to mitigate this issue and improve the model's generalization performance [8,50].

2.5 Deep Learning Models

Deep learning models, a subset of machine learning algorithms inspired by the structure and function of the human brain, have revolutionized various industries with their ability to learn complex patterns from large amounts of data. These models, composed of multiple layers of interconnected nodes, known as artificial neural networks, excel at tasks such as image and speech recognition, natural language processing, and decision-making. By leveraging deep learning models, organizations can extract valuable insights, make accurate predictions, and automate processes with unprecedented efficiency and accuracy [30,35,38].

2.5.1 Multi-Layer Perceptron (MLP)

MLP, a type of feedforward neural network, has shown promise in low-resourced language settings. MLP models have been applied to tasks such as language identification, named entity recognition, and sentiment analysis. Their ability to learn complex representations from input data makes them effective in capturing subtle linguistics and improving classification performance [38,40].

2.5.2 Recurrent Neural Networks (RNN)

RNNs, specifically bi-directional long short-term memory (Bi-LSTM) networks, have been utilized extensively in low-resourced languages for tasks such as machine translation, speech recognition, and sentiment analysis. The sequential nature of RNNs allows them to capture contextual dependencies and handle variable-length input, making them well-suited for language-

related classification tasks. A bidirectional sequence processing model utilizes dual components to receive input in opposite directions, expanding the network's access to a broader range of information and enhancing contextual understanding. This approach allows for the simultaneous extraction of a significant volume of contextual data, enabling the classification of sequential data into multiple categories. These cutting-edge models represent advanced solutions in data classification tasks [10,30].

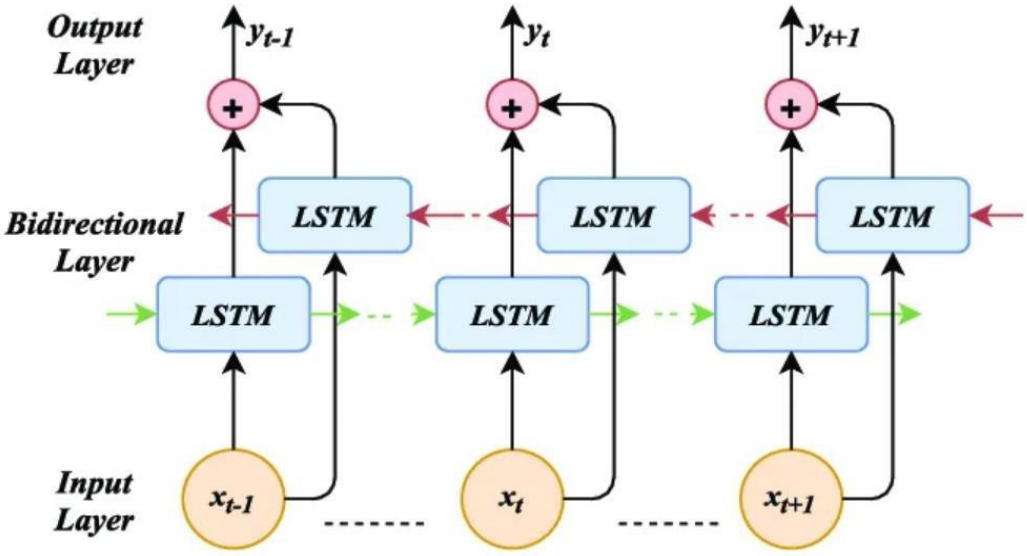

Figure 1BI-LSTM Model Architecture Adopted From Augustin O.Nwanjana

Incorporating bidirectional processing techniques, the LSTM (Long Short-Term Memory) model enhances the capture of long-term dependencies within sequential data. By introducing memory cells and gating mechanisms, LSTMs selectively retain and discard information over time, leveraging an internal memory state to store data for extended periods. Unlike traditional unidirectional RNNs that process input sequences in a single direction, the bidirectional LSTM (Bi-LSTM) processes data in both forward and backward directions concurrently. Comprising two LSTM layers one for forward processing and the other for backward processing each layer maintains its hidden states and memory cells [38,40].

2.6 Feature Extraction

Feature extraction is a process used in machine learning to reduce the number of resources needed for processing without losing important or relevant information. Feature extraction helps in the reduction of the dimensionality of data which is needed to process the data effectively. In other words, feature extraction involves creating new features that still capture the essential information from the original data but in a more efficient way [40].

2.6.1 CountVectorizer

CountVectorizer is a widely used technique in text analysis for extracting features from textual data. It converts text documents into a matrix that represents the frequency of each term. Each row in the matrix corresponds to a document, and each column represents a unique term present in the dataset, with the values indicating how often each term appears in the document. In the context of symptom-based corpora, CountVectorizer is applied to analyze the frequency counts of symptoms within disease descriptions [23,53].

2.6.2 Term Frequency-Inverse Document Frequency (TF-IDF)

The TF-IDF metric is a widely used measure for assessing the importance of a term within a document or dataset. It achieves this by combining two key components: Term Frequency (TF) and Inverse Document Frequency (IDF). Term Frequency calculates the relative importance of a term within a document by determining the ratio of the term's frequency to the total number of terms in the document. This provides insight into the significance of a term within a specific context. On the other hand, Inverse Document Frequency evaluates the rarity of a term across the entire dataset by computing the logarithm of the total number of documents divided by the number

of documents containing the term. By integrating TF and IDF, the TF-IDF metric offers a comprehensive assessment of the importance of symptoms in individual disease descriptions or across the entire symptom dataset. It quantifies the frequency of symptoms in each disease description, shedding light on the significance of symptoms within specific diseases [27,39].

2.6.3 One-hot Encoding

One-hot encoding is a widely employed technique for converting categorical variables into a format compatible with machine learning models. This method proves particularly useful when working with models that cannot directly handle categorical data, such as many types of neural networks. By transforming text labels into a one-hot encoded representation, the Bi-LSTM model can effectively learn the relationships between symptom descriptions and their corresponding labels during the training process. The process of one-hot encoding involves two key steps. First, the unique categories present in the data are identified and stored in a dictionary, such as the label_dict. Then, for each unique category, a new binary column is created, where the value is set to 1 if the observation belongs to that category, and 0 otherwise [35,50]. The benefits of one-hot encoding are multifaceted. Firstly, it provides a numerical representation of categorical variables, which is a requirement for many machine learning models. Secondly, the one-hot encoded format preserves the information about the original categories, as each category is represented by a separate binary column. Lastly, one-hot encoding does not make any assumptions about the ordering or numerical relationship between the categories, which is crucial for categorical variables that do not have a natural ordering (e.g., "low", "medium", "high") [16].

2.6.4 Label Encoding

Label encoding serves as a valuable technique in the realm of data processing, offering a straightforward approach to converting categorical variables into a numerical format suitable for machine learning applications. This method proves advantageous in scenarios where the categorical variables exhibit a natural ordering or when dealing with many unique categories. By following a structured process of identifying unique categories, creating a mapping dictionary, and transforming labels into numerical values, label encoding simplifies the data transformation process [46, 49].

2.6.5 Tokenization

Tokenization involves breaking down the input text into smaller units known as tokens. The Tokenizer class from the Keras preprocessing module. This process encompasses several key steps: extracting unique words (tokens) from the text, assigning a distinct numerical index to each word in the vocabulary, and establishing a mapping between words and their respective numerical indices. The resulting tokenizer object encapsulates the vocabulary and the word-to-index mapping, each word in the input text is substituted with its corresponding numerical index, generating a sequence of numbers [33,44,47].

2.6.6 Padding

Padding is also a text preprocessing technique commonly used in deep learning tasks involving text data. It is considered a text preprocessing tool because it helps prepare the input data for the model by ensuring that all sequences have the same length. Mostly performed after the tokenization step, where the text is converted into numerical sequences. The purpose of padding is to address the issue of varying sequence lengths, which is a common problem when working with text data. Different input sequences (e.g., symptom descriptions) can have different numbers

of words, resulting in sequences of varying lengths. Many deep learning models, including the Bi-LSTM model, require all input sequences to have the same length. This is because the model's architecture, such as the input layer, expects a fixed- size input. Padding helps ensure that all sequences have the same length, allowing the model to process the data efficiently [26,28,50].

Chapter 3 Methodology

Research methodology is the overall process and the way of solving the identified problems. It includes methods, techniques, and approaches for data collection, analysis, training, and design of the model.

3.1 Research Design and Approach

In this study, we follow an experimental research design. This type of methodology attempts to determine or predict what may occur and it specifies an experimental and a control group. The independent variable is administered to the experimental group and not to the control group, and both groups are measured on the same dependent variable.

Accordingly, this research has gone through several stages, including defining the problem, reviewing relevant literature, designing the research, gathering the necessary data sets, choosing and training a model, developing the chatbot, and testing. In this methodology, first, the problem is assessed by literature and observations then based on the identified problem, proposed artifacts are designed. This methodology design includes tools, methods, models, and evaluation mechanisms.

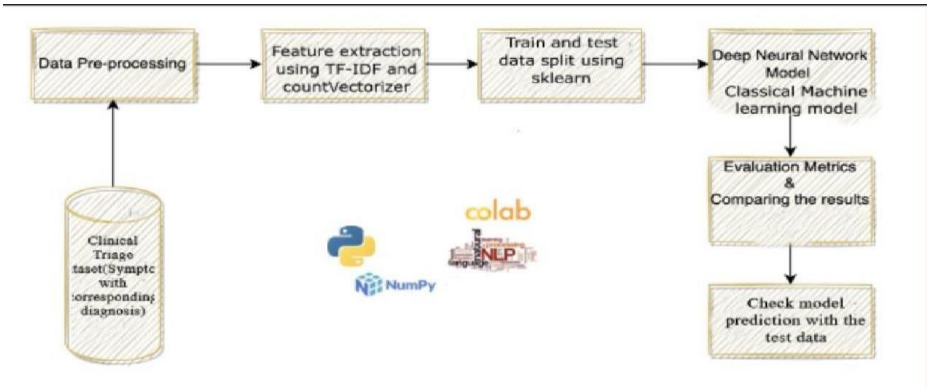

Figure 2 Research design and approach diagram.

3.2 Data Collection

The data for this study was collected from multiple sources. Symptom and prognosis data was obtained from two hospitals, Black lion and MCM hospitals, covering various medical units including internal medicine, pediatrics, gynecology, dermatology, neurology, and ENT. In total, ten healthcare professionals were selected from each department, comprising general practitioners, specialists, and fellowship students an additional twenty healthcare professionals were surveyed and interviewed to gather further data.

To ensure the validity and reliability of the data, it was reviewed and validated by ten physicians from the participating hospitals then reviewed by one language expert. Out of the 278 symptom pairs gathered from primary and secondary sources, 250 were approved by the language experts and the internist, and this approved dataset was used as the foundation for the thesis research. Furthermore, secondary data was obtained from an English-language dataset, this data was translated into Amharic language by two linguistic professionals, also validated the translations by one senior internist. The combination of primary data collected from the hospitals and healthcare providers, as well as the secondary data obtained and translated, provides a comprehensive dataset to support the objectives of this research study.

3.3 Data Preprocessing

In machine learning, dataset preparation/processing is a very crucial step. It involves transforming raw data into a format suitable for analysis or model training. A preprocessing task is performed to remove raw data that contain unwanted punctuation marks, numerical values, and special characters, and replace it with a single alphabet (normalization) in a different representation because of reducing the effect on the performance and the correctness of the model. So, it is used to clean data, which makes it suitable for further analysis.

At first, medical-related symptom-diagnosis pairs were collected from various healthcare professionals, and then stored in .csv and .txt format. The file (dataset.csv and dataset.txt) had tags.The symptom contains the classes (for e.g. The original text የልብ ምት መፍጠን የልብ ምት መታወቅ የትንፋሽ ማጠር ሲትኙ የሚብስ ትንፋሽ ማጠር ከ አንድ ትራስ በላይ መጠቀም ሳል የሰውነት እብጠት የሆድ እብጠት የደረት ላይ ህመም etc), we have the corresponding diagnosis (for example ሳንባ ውስጥ ደም ስር መዘጋት, ልብ ድካም, ጨጓራ ቁስለት). Before training the models, we performed the necessary preprocessing steps to prepare the Amharic clinical text data. This involved Character normalization like character_map = {'ኍ ሀ' 'ሀ', 'ሃ' 'ሀ', 'ሐ' 'ሀ', 'ሓ' 'ሀ', 'ኻ' 'ሀ', 'ኋ' 'ሀ'} which is performed to ensure consistency in the representation of Amharic characters, defining a mapping dictionary that maps similar characters to a common representation to reduce the variability of characters and remove unwanted punctuation marks, the individual words in a sentence are separated by two dots (: ሁለትነጥብ). The end of a sentence is marked by Amharic full stop (። አራት ነጥብ). The symbol (፣ ነጠላ ሰረዝ) represents a comma, while (፤ ድርብ ሰረዝ) correspond to a semicolon. '!' and '?' punctuations are used to end exclamatory and interrogative sentence respectively.

3.4 Feature Extraction

Employed feature extraction from the scikit-learn library, it converts the symptoms (textual data) into numerical feature vectors, the vectorizer tokenizes the symptoms and creates a vocabulary of unique words. Each symptom is then represented by a vector indicating the presence or absence of these words [48]. We have used feature extraction techniques in our experiments such as TF-IDF and Count Vectorizer, label Encoding, One-hot encoding, tokenization and padding for transforming textual symptom descriptions into numerical representations.

3.5 Sample Sizes and Sampling Methods

The study employed a mixed-methods approach, combining both qualitative and quantitative data collection methods. The sampling design and size were carefully considered to ensure that the sample was representative of the population. A total of 278 symptom pairs were collected from primary and secondary sources, which served as the foundation for the analysis. 90 healthcare professionals were surveyed and interviewed, comprising 70 healthcare professionals from 7 medical departments (internal medicine, pediatrics, gynecology, dermatology, neurology, and ENT) and 20 more healthcare professionals. This sample size was deemed sufficient to achieve the objectives of the study and to enable the identification of patterns and trends in the data.

A combination of convenience, purposive, and stratified sampling methods was employed to select participants for the study. Convenience sampling was used to select healthcare professionals from various departments, while purposive sampling was used to select participants who were likely to provide relevant data. Stratified sampling was used to ensure that the sample was representative of the different departments.

3.6 Model Selection

The models chosen for this study were selected based on their ability to handle the complexity of the Amharic language and the task of symptom classification. The decision tree, SVM, and random forest were chosen for their ability to handle high-dimensional data and their robustness to noise and outliers. The logistic regression and Naïve Bayes were chosen for their simplicity and ease of interpretation. The MLP, and Bi-LSTM were chosen for their ability to handle sequential data and their ability to learn complex patterns in data. The models selected for ensemble are chosen based on their individual performance. By selecting models with different strengths and weaknesses, we can create an ensemble that can capture a wider range of patterns and insights, leading to better generalization and robustness.

3.7 Tools & Setup

The data loading, pre-processing, feature extraction, and evaluation stages of our project were conducted within the user-friendly Jupyter Notebook environment, seamlessly integrated into the Anaconda Distribution. Our primary system, a Windows 11 64-bit laptop, boasts an Intel(R) Core (TM) i7-7600U CPU @ 2.80GHz 2.90 GHz processor and a generous 16GB of RAM, ensuring optimal performance for our computational tasks. To harness the power of Graphics Processing Units (GPUs) for the development and training of our deep learning model, we leveraged Google Colaboratory. GPUs excel in training multi-layered deep learning networks by virtue of their parallel processing capabilities and exceptional computational efficiency, offering a significant boost to our model training process.

This collaborative utilization of Jupyter Notebook, Google Colaboratory, and cutting-edge technologies underscores our commitment to harnessing advanced tools and methodologies to drive innovation and excellence in our project.

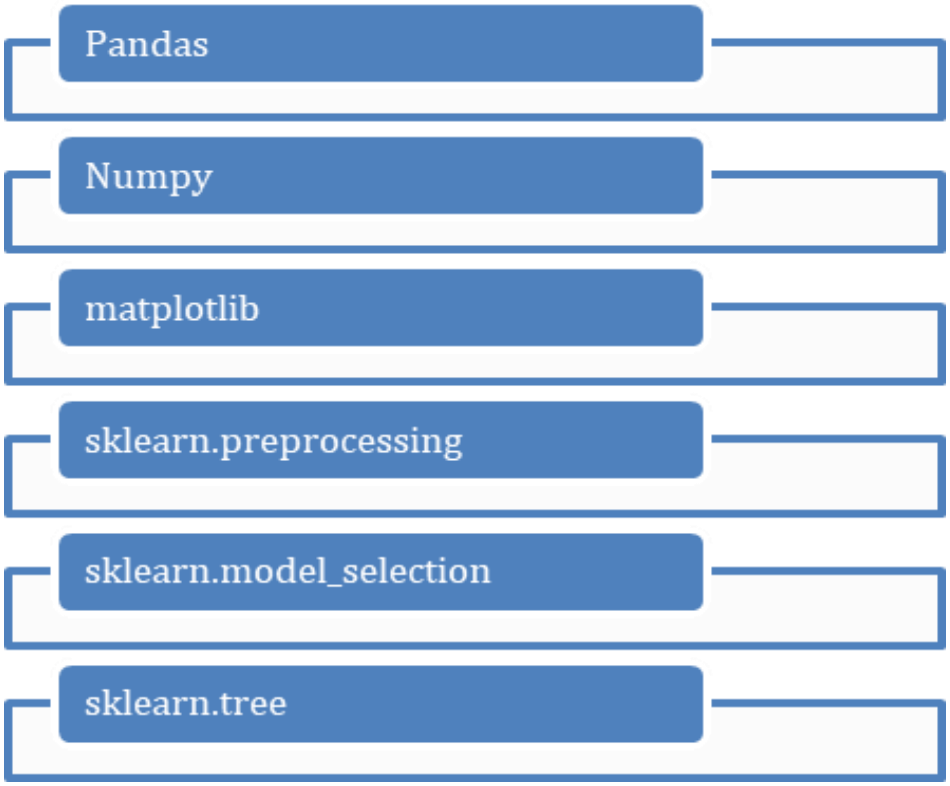

Figure 3 Sample Python Libraries

Chapter 4 Experimental Results and Analysis

This chapter deals with Experimental Results and Analysis section of a research thesis presents and interprets the findings of a study, aiming to presentation of the results in tables and figures, a summary of the main findings, and an interpretation of the results considering the research questions and literature review evaluate the performance of various machine learning and deep learning models applied to text-based clinical chatbots for the Amharic language. We present a comprehensive analysis of the results, highlighting key findings, and comparing different models. The results and analysis presented in this section provide valuable insights into the effectiveness of different machine learning and deep learning models which can be used to inform future research and development efforts in this area.

4.1Experimenting

In this section, we present the details of the model training and evaluation process for the development and evaluation of the Amharic clinical chatbot. We utilized a diverse dataset of Amharic medical texts, consisting of symptoms and corresponding diagnosis, to train and evaluate multiple machines learning models, including Random Forest, Logistic Regression, Decision Tree, SVM, as well as deep learning models, namely Multilayer Perceptron (MLP) and Bidirectional Long Short-Term Memory (Bi-LSTM).

4.1.1 Classical Machine Learning Models

The experiment begins with loading the dataset from a text file and normalizing the characters using the `normalize_characters()` function. The dataset is then split into input (`symptoms`) and output (`triage_recommendations`) lists, and a `CountVectorizer` and TF-IDF are used to convert the symptoms into numerical feature vectors. After the preprocessing steps, we proceed to train and evaluate each model using the preprocessed dataset. The models used in this research thesis include Random Forest, SVM, Decision Tree, and Naive Bayes classifiers and logistic regression.

Model Training procedure.

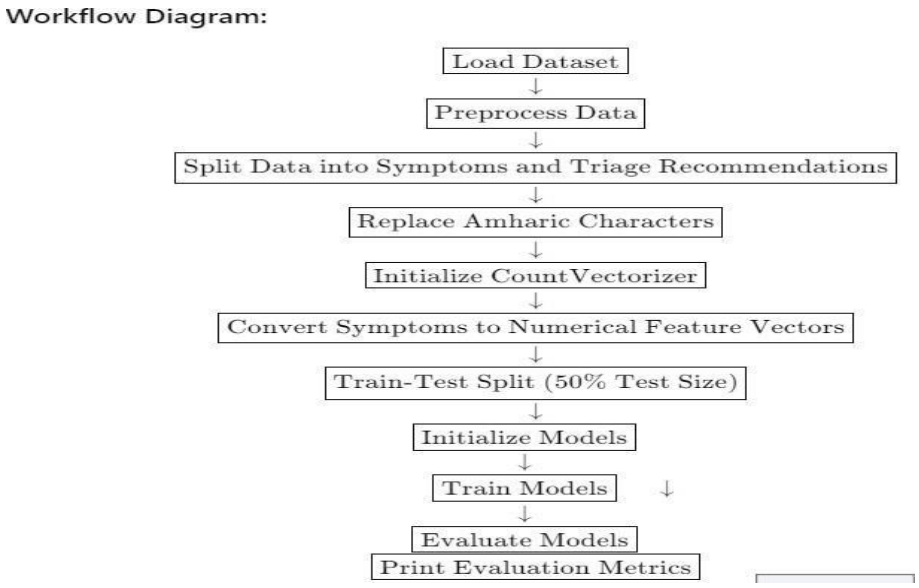

Figure 4 Classical Machine Learning Overall Workflow

The experiments conducted with various feature extraction techniques and hyperparameters yielded insightful results. In the default setting (Experiment 1), Random Forest, SVM, and Decision Tree models demonstrated strong performance, while Naive Bayes and Logistic Regression models lagged. When employing TF-IDF (Experiment 2), consistent performance was noted across models, with the mentioned models maintaining high accuracy and precision levels. Experiment 3 showcased enhanced performance with advanced hyperparameters for SVM and Decision Tree models, although Naive Bayes also showed improvement but to a lesser extent. Count Vectorizer experiments (Experiments 4 and 5) highlighted the stable performance of models across different hyperparameter settings, with SVM and Logistic Regression models notably improving accuracy and F1 scores in the advanced parameter setup as shown below:

Table 1Machine learning Models Performance result for each experiment

Experiment 1 TF-IDF using default hyperparameter.

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.9286	0.9196	0.9286	0.9226
SVM	0.9286	0.9196	0.9286	0.9226
Decision Tree	0.9286	0.9196	0.9286	0.9226
Naive Bayes	0.6964	0.6469	0.6964	0.6574
Logistic Regression	0.7679	0.6786	0.7679	0.7068

Experiment 2 TF-IDF using different hyperparameter.

Model	Hyperparameters	Accuracy	Precision	Recall	F1 Score
Random Forest	{'n_estimators': 100, 'max_depth': None, 'min_samples_split': 2}	0.9286	0.9196	0.9286	0.9226
SVM	{'C': 1.0, 'kernel': 'rbf'}	0.9286	0.9196	0.9286	0.9226
Decision Tree	{'criterion': 'gini', 'max_depth': None, 'min_samples_split': 2}	0.9286	0.9196	0.9286	0.9226
Naive Bayes	{'alpha': 1.0}	0.6964	0.6469	0.6964	0.6574
Logistic Regression	{'C': 1.0, 'solver': 'lbfgs', 'max_iter': 500}	0.7679	0.6786	0.7679	0.7068

Experiment 3 TF-IDF with different hyperparameter

Model	Hyperparameters	Accuracy	Precision	Recall	F1 Score
Random Forest	{'n_estimators': 200, 'max_depth': 10, 'min_samples_split': 5}	0.7500	0.7076	0.7500	0.7183
SVM	{'C': 1.0, 'kernel': 'linear', 'gamma': 'scale'}	0.8929	0.8839	0.8929	0.8869
Decision Tree	{'criterion': 'entropy', 'max_depth': 8, 'min_samples_split': 3}	0.8929	0.8720	0.8929	0.8780
Naive Bayes	{'alpha': 0.5}	0.7679	0.7321	0.7679	0.7402
Logistic Regression	{'C': 1.0, 'solver': 'lbfgs', 'max_iter': 500}	0.7679	0.6786	0.7679	0.7068

Experiment 4 Count Vectorizer with different hyperparameter

Model	Hyperparameters	Accuracy	Precision	Recall	F1 Score
Random Forest	{'n_estimators': 100, 'max_depth': None, 'min_samples_split': 2}	0.9286	0.9196	0.9286	0.9226
SVM	{'C': 1.0, 'kernel': 'rbf'}	0.9286	0.9196	0.9286	0.9226
Decision Tree	{'criterion': 'gini', 'max_depth': None, 'min_samples_split': 2}	0.9286	0.9196	0.9286	0.9226
Naive Bayes	{'alpha': 1.0}	0.9286	0.9196	0.9286	0.9226
Logistic Regression	{'C': 1.0, 'solver': 'lbfgs', 'max_iter': 500}	0.9286	0.9196	0.9286	0.9226

Experiment 5 Count Vectorizer with different hyperparameter

Model	Hyperparameters	Accuracy	Precision	Recall	F1 Score
Random Forest	{'n_estimators': 200, 'max_depth': 10, 'min_samples_split': 5}	0.8214	0.8006	0.8214	0.8065
SVM	{'C': 1.0, 'kernel': 'linear', 'gamma': 'scale'}	0.9286	0.9196	0.9286	0.9226
Decision Tree	{'criterion': 'entropy', 'max_depth': 8, 'min_samples_split': 3}	0.8929	0.8720	0.8929	0.8780
Naive Bayes	{'alpha': 0.5}	0.9286	0.9196	0.9286	0.9226
Logistic Regression	{'C': 0.5, 'solver': 'liblinear', 'max_iter': 500}	0.9286	0.9196	0.9286	0.9226

4.1.2 Deep Learning Models

The dataset is loaded using pandas and preprocessed for use in a deep learning model. The symptoms are converted into integer sequences using the Keras Tokenizer, allowing the model to handle the sequential nature of the data. The labels are then numerically encoded and one-hot encoded to prepare them for use with the model.

The model architecture consists of several key components. An embedding layer is used to convert the integer sequences into dense vector representations, allowing the model to capture complex relationships between the symptoms. A bidirectional LSTM layer is then used to analyze the sequence patterns in the data, processing the input from both the past and future to capture contextual information. Finally, a dense layer with a softmax activation function is used to output the label probabilities, enabling the model to make predictions.

The model is trained using the Adam optimizer, a popular choice for deep learning models due to its ability to adapt to the learning rate and momentum. The categorical cross-entropy loss function is used to measure the difference between the model's predictions and the true labels, providing a clear and accurate metric for evaluating the model's performance.

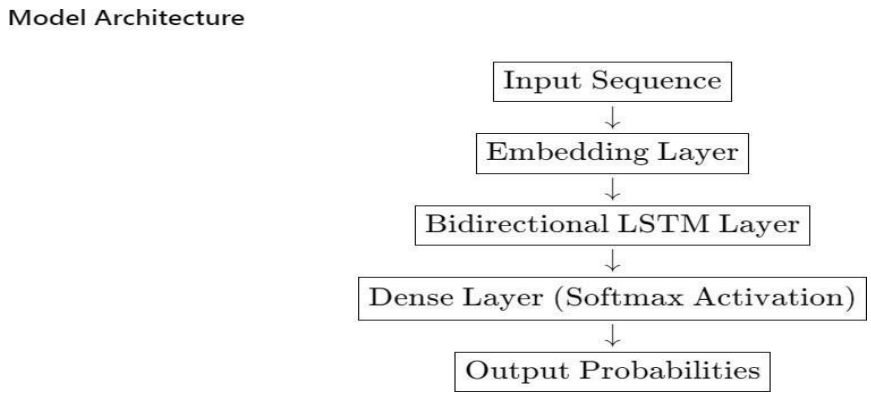

Figure 5 Bi-LSTM Model Architecture

To train the model, the `fit()` function is utilized, passing the training data and validation data, along with the specified number of epochs. This critical decision determines how many times the model will learn from the data. For small datasets, a common approach is to start with a shallow architecture, such as a single hidden layer, and gradually increase it to 3-layered to prevent overfitting. In our experiment, the architecture consists of 1 and 3-layered hidden layers, involving a bidirectional LSTM layer, which is a reasonable starting point for a small dataset. The model is trained for 10 epochs, allowing the training progress to be monitored and the accuracy and loss metrics to be recorded for both the training and validation sets.

During training, the model calculated performance metrics such as loss and accuracy -

```
Epoch 1/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 7s 135ms/step - accuracy: 0.0154 - loss: 4.0004 - val_accuracy: 0.0179 - val_loss: 3.9694
Epoch 2/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - accuracy: 0.0781 - loss: 3.9361 - val_accuracy: 0.0357 - val_loss: 3.9295
Epoch 3/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - accuracy: 0.0769 - loss: 3.8290 - val_accuracy: 0.0357 - val_loss: 3.9255
Epoch 4/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - accuracy: 0.0807 - loss: 3.6345 - val_accuracy: 0.0714 - val_loss: 3.8072
Epoch 5/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - accuracy: 0.1926 - loss: 3.3335 - val_accuracy: 0.1071 - val_loss: 3.5566
Epoch 6/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - accuracy: 0.1825 - loss: 3.0803 - val_accuracy: 0.1250 - val_loss: 3.3086
Epoch 7/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - accuracy: 0.2811 - loss: 2.7916 - val_accuracy: 0.4107 - val_loss: 2.9430
Epoch 8/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - accuracy: 0.5405 - loss: 2.3001 - val_accuracy: 0.5357 - val_loss: 2.5254
Epoch 9/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - accuracy: 0.6483 - loss: 1.8986 - val_accuracy: 0.5893 - val_loss: 2.2065
Epoch 10/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - accuracy: 0.6827 - loss: 1.6691 - val_accuracy: 0.6250 - val_loss: 1.8322
2/2 ━━━━━━━━━━━━━━━━━━━━ 1s 429ms/step
```

Figure 6 Single Layer BI-LSTM Accuracy & Loss Result

```
Epoch 1/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 17s 345ms/step - accuracy: 0.0077 - loss: 4.0030 - val_accuracy: 0.0893 - val_loss: 3.9851
Epoch 2/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 119ms/step - accuracy: 0.1000 - loss: 3.9100 - val_accuracy: 0.0357 - val_loss: 3.9377
Epoch 3/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 133ms/step - accuracy: 0.1002 - loss: 3.5940 - val_accuracy: 0.0893 - val_loss: 3.6972
Epoch 4/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 107ms/step - accuracy: 0.1481 - loss: 3.2232 - val_accuracy: 0.1250 - val_loss: 3.4300
Epoch 5/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 139ms/step - accuracy: 0.2011 - loss: 2.9743 - val_accuracy: 0.2679 - val_loss: 3.1218
Epoch 6/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 127ms/step - accuracy: 0.3563 - loss: 2.5794 - val_accuracy: 0.2500 - val_loss: 2.8365
Epoch 7/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 111ms/step - accuracy: 0.3895 - loss: 2.2593 - val_accuracy: 0.3393 - val_loss: 2.4838
Epoch 8/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 110ms/step - accuracy: 0.5095 - loss: 1.9182 - val_accuracy: 0.4643 - val_loss: 2.1481
Epoch 9/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 115ms/step - accuracy: 0.6252 - loss: 1.5437 - val_accuracy: 0.5357 - val_loss: 1.8088
Epoch 10/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 125ms/step - accuracy: 0.7662 - loss: 1.1953 - val_accuracy: 0.5536 - val_loss: 1.5986
```

Figure 7Three Layer BI-LSTM performance Result.

4.2 Experimenting Model Evaluation

Following the successful training of the models, the outcomes of the models were evaluated with numerous accuracy metrics. It is essential to determine whether the outputs are as expected, as this estimation is regarded as critical since it allows a comparison to be made between the projected and true results. In this section, the performance and efficiency of the models i.e. Deep Neural Network and classical machine learning models were analyzed. For assessing the performance of the models, various metrics like Training Accuracy, Training Loss, Validation Accuracy, Validation Loss, f1 score, Precision, and Recall measures were exercised.

4.2.1 Decision Tree

The decision tree classifier is initialized and trained on the training data. Evaluation is performed on the testing set, and the following performance metrics are calculated.

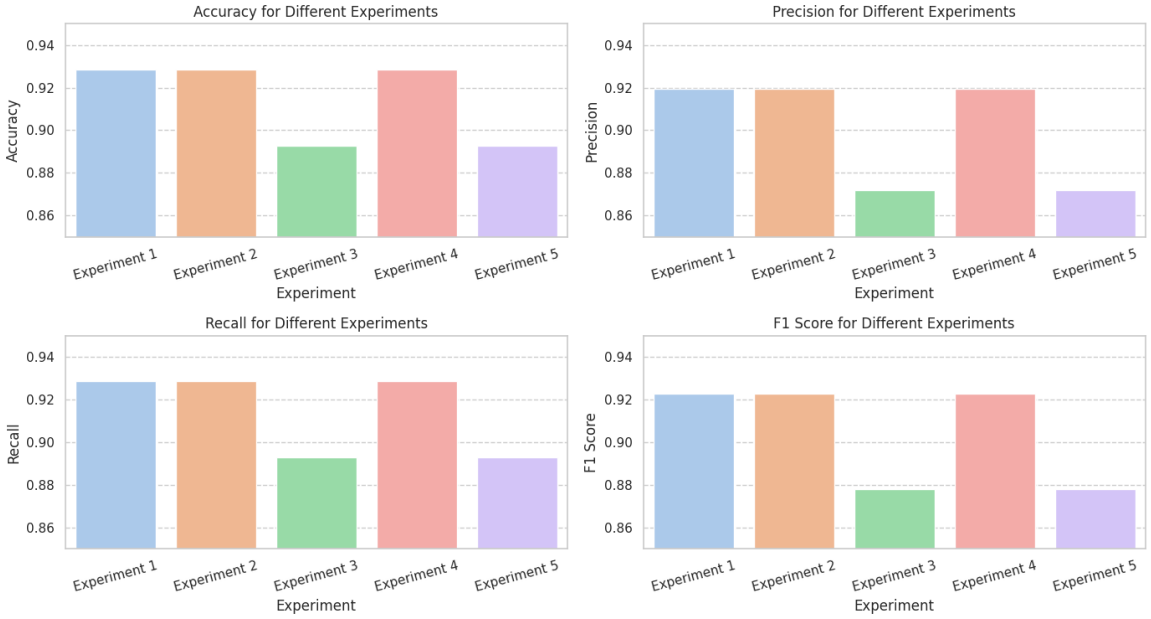

Figure 8 Decision Tree Performance Result with different hyperparameter.

The series of experiments involved the Decision Tree model with varying hyperparameters and feature extraction techniques, several noteworthy observations can be made.

In Experiment 1, where the Decision Tree model utilized default hyperparameters, it achieved an accuracy of 0.9286 and an F1 score of 0.9226, showcasing strong performance right from the baseline settings.

Experiment 2 introduced hyperparameters tailored for TF-IDF feature extraction, maintaining the same accuracy and performance metrics as Experiment 1. This suggests that the TF-IDF technique did not significantly impact the Decision Tree model's predictive capabilities in this context.

Experiment 3 involved advanced hyperparameters for the Decision Tree model, resulting in a slightly lower accuracy of 0.8929 and F1 score of 0.8780 compared to the default settings. This indicates that while advanced hyperparameters can offer customization, they may not always lead to improved model performance.

Moving on to Experiment 4, which incorporated hyperparameters under the Count Vectorizer approach, the Decision Tree model once again delivered consistent results with an accuracy of 0.9286 and an F1 score of 0.9226, aligning closely with the baseline performance.

Lastly, experiment 5 revisited advanced hyperparameters coupled with Count Vectorizer, mirroring the outcomes of Experiment 3 with an accuracy of 0.8929 and an F1 score of 0.8780. This reaffirms the notion that advanced hyperparameters may not always yield superior results compared to default configurations.

In summary, the Decision Tree model displayed remarkable consistency in performance across various hyperparameter settings and feature extraction methods. While default configurations often proved to be robust, advanced hyperparameters did not consistently outperform them in this specific experimental setup. These findings underscore the importance of systematic evaluation and optimization strategies tailored to the unique characteristics of the dataset and model architecture.

4.2.2 Support Vector Machines

The experimental results demonstrate that the developed SVM classifier model performs well in predicting recommendations based on patient symptoms. The evaluation metrics obtained are as follows

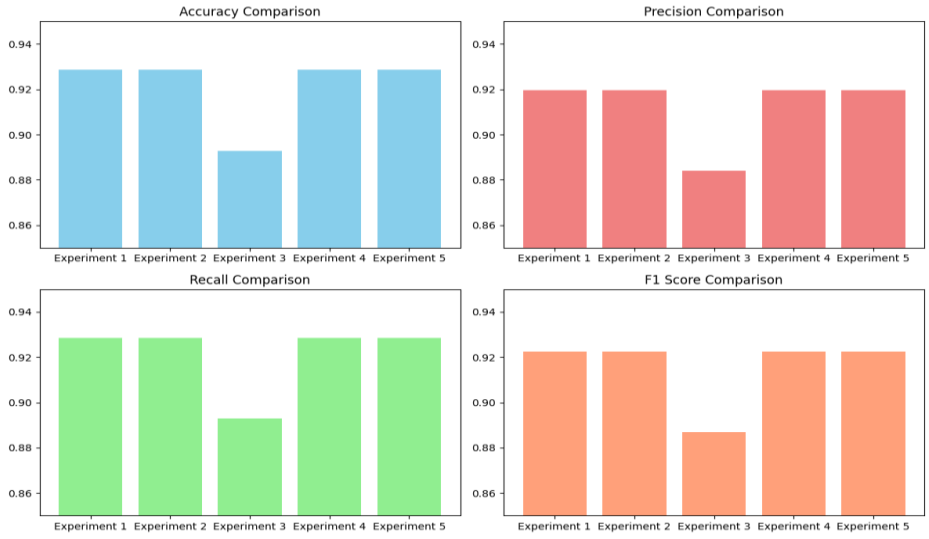

Figure 9 SVM Performance Result using different Hyperparameter.

The series of experiments conducted with the Support Vector Machine (SVM) model explored various hyperparameter configurations to evaluate their impact on model performance. In Experiment 1, default settings were employed, resulting in an accuracy of 0.9286 and an F1 score of 0.9226. Experiment 2 introduced hyperparameters specific to TF-IDF, including a 'C' value of 1.0 and a 'rbf' kernel, maintaining consistent performance metrics with Experiment 1. Experiment 3 delved into advanced hyperparameters with a 'linear' kernel and 'scale' gamma, yielding a slightly lower accuracy of 0.8929 and an F1 score of 0.8869 compared to default settings. Experiment 4 utilized hyperparameters in conjunction with Count Vectorizer, achieving an accuracy of 0.9286 and an F1 score of 0.9226, aligning closely with the baseline performance. Lastly, experiment 5 revisited advanced hyperparameters paired with Count Vectorizer,

mirroring the outcomes of Experiment 3 with an accuracy of 0.9286 and an F1 score of 0.9226. These experiments underscore the significance of selecting and tuning hyperparameters judiciously to enhance SVM model performance, emphasizing the need for tailored optimization strategies aligned with the dataset characteristics and problem domain.

4.2.3 Random Forests

In this experiment, a Random Forest model was utilized as an alternative classification model for predicting recommendations based on patient symptoms. The performance metrics obtained from the model evaluation are as follows.

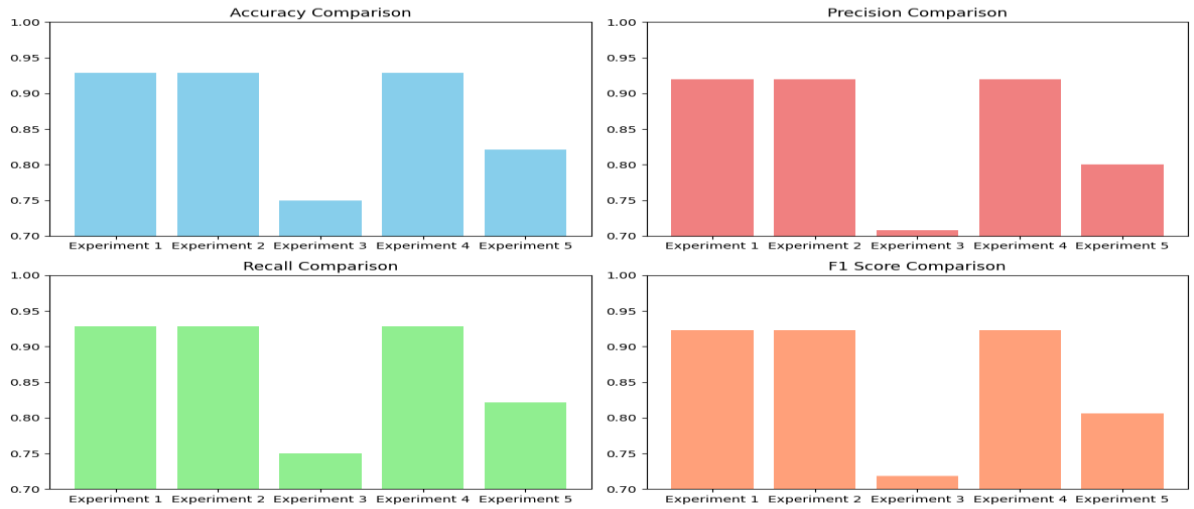

Figure 10 Random Forest Result with different hyperparameters.

The analysis of the experiments conducted with the Random Forest model reveals interesting insights into its performance under various hyperparameter configurations and feature extraction methods.

In Experiment 1, utilizing the Random Forest model with default settings yielded a high accuracy of 0.9286 and an impressive F1 score of 0.9226, indicating strong performance right from the baseline configuration.

Experiment 2 introduced hyperparameters tailored for TF-IDF, including 'n_estimators' of 100, 'max_depth' set to None, and 'min_samples_split' of 2. The results remained consistent with Experiment 1, showcasing no significant improvement or decline in performance metrics.

Experiment 3 explored advanced hyperparameters specific to TF-IDF, with 'n_estimators' increased to 200, 'max_depth' set at 10, and 'min_samples_split' of 5. However, this configuration led to a notable decrease in accuracy to 0.75 and a lower F1 score of 0.7183, indicating that the advanced settings did not enhance model performance in this scenario.

Moving to Experiment 4, employing hyperparameters in conjunction with Count Vectorizer produced results akin to Experiment 1, with an accuracy of 0.9286 and an F1 score of 0.9226, highlighting the robustness of the model under default configurations.

Experiment 5 revisited advanced hyperparameters paired with Count Vectorizer, featuring 'n_estimators' of 200, 'max_depth' at 10, and 'min_samples_split' of 5. While this setup led to a decrease in accuracy to 0.8214 and a lower F1 score of 0.8065 compared to default settings, it still showcased reasonable performance levels.

The Random Forest model demonstrated consistent performance across various hyperparameter configurations and feature extraction techniques. While default settings proved to be robust, the impact of advanced hyperparameters varied, emphasizing the importance of tailored optimization strategies based on the dataset characteristics and model requirements.

4.2.4 Naive Bayes

In this study, a Naive Bayes model was employed as a classification model for predicting recommendations based on patient symptoms. The performance metrics obtained from the model evaluation are as follows.

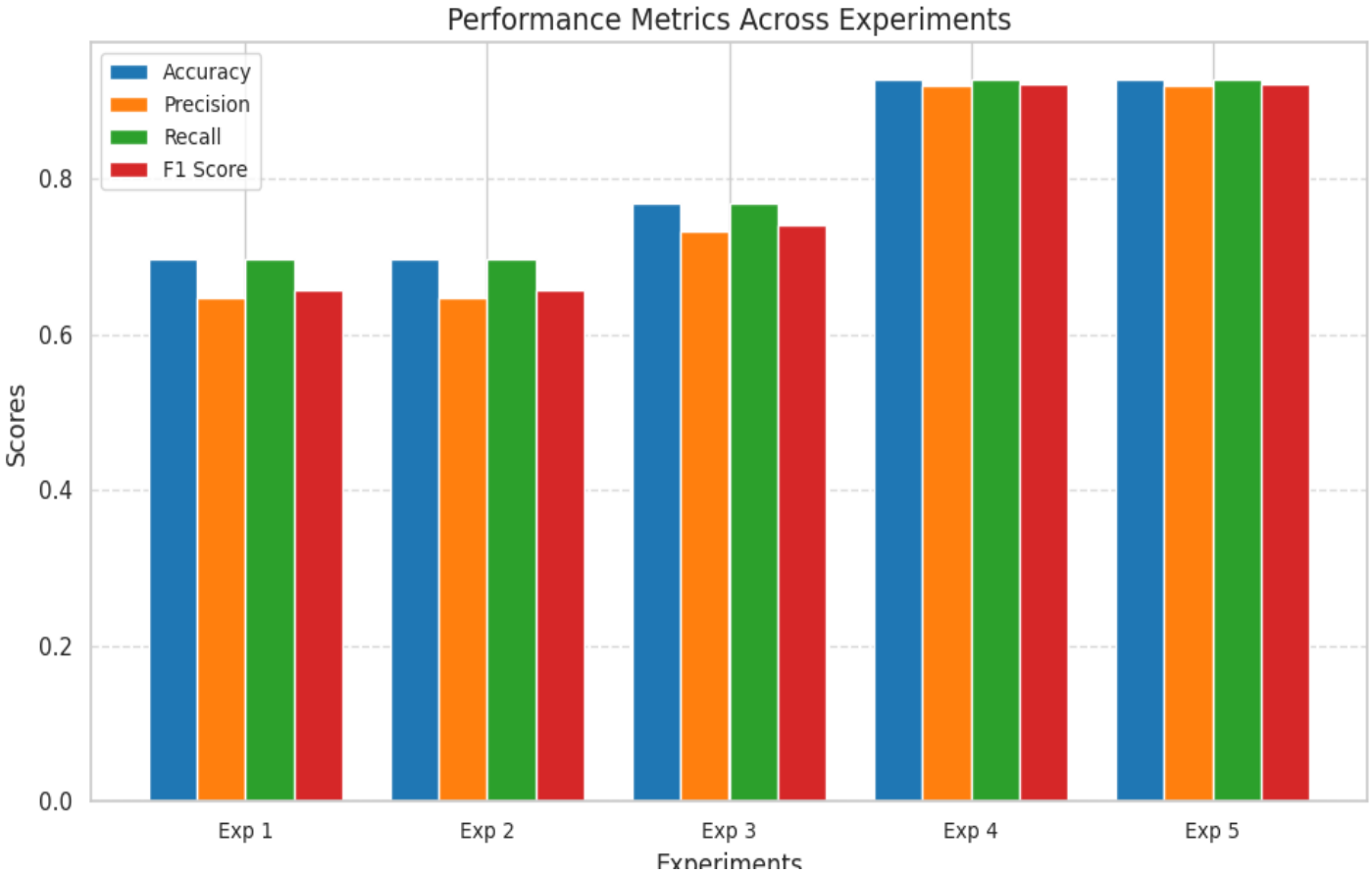

Figure 11 Naive Bayes Result

The analysis of the experiments conducted with the Naive Bayes model sheds light on its performance across different hyperparameter configurations and feature extraction methods.

In Experiment 1, the Naive Bayes model, operating with default settings, achieved an accuracy of 0.6964 and an F1 score of 0.6574, demonstrating a moderate level of performance.

Experiment 2 introduced hyperparameters specific to TF-IDF, with an 'alpha' value of 1.0, maintaining consistent accuracy and performance metrics compared to the baseline configuration in Experiment 1.

Experiment 3 explored the impact of advanced hyperparameters, specifically setting 'alpha' to 0.5, which led to an improvement in accuracy to 0.7679 and an enhanced F1 score of 0.7402, indicating that fine-tuning hyperparameters can positively influence model performance.

Moving to Experiment 4, employing hyperparameters in conjunction with Count Vectorizer resulted in a notable increase in accuracy to 0.9286 and an impressive F1 score of 0.9226, showcasing the effectiveness of this configuration in enhancing model outcomes.

Finally, experiment 5 revisited advanced hyperparameters paired with Count Vectorizer, with 'alpha' set to 0.5, maintaining the high accuracy of 0.9286 and F1 score of 0.9226 achieved in Experiment 4, demonstrating the consistency and reliability of this hyperparameter setting.

In conclusion, the Naive Bayes model exhibited varying performance levels across different hyperparameter configurations and feature extraction techniques. The results underscore the importance of hyperparameter tuning, with advanced settings and the choice of feature extraction method playing significant roles in optimizing the model's predictive capabilities. These findings provide valuable insights for researchers and practitioners aiming to leverage Naive Bayes in their classification tasks.

4.2.5 Logistic Regression

The model is trained using the training set and predictions are made on the testing set. The accuracy of the model is calculated using the accuracy_score function from sklearn metrics.

Figure 12 Logistic Regression Result

The analysis of the Logistic Regression experiments reveals intriguing insights into the model's performance across various hyperparameter configurations and feature extraction techniques.

In Experiment 1, utilizing the Logistic Regression model with default settings yielded a moderate accuracy of 0.7679 and an F1 score of 0.7068, indicating a reasonable level of performance as a baseline.

Experiment 2 introduced hyperparameters customized for TF-IDF, including 'C' set to 1.0, 'solver' as 'lbfgs', and 'max_iter' of 500. The results remained consistent with Experiment 1, suggesting that these specific hyperparameters did not significantly impact the model's performance.

Experiment 3 explored advanced hyperparameters identical to Experiment 2, yet no notable improvement or decline in performance metrics was observed, indicating that the advanced settings did not yield substantial enhancements in this context.

Moving on to Experiment 4, implementing hyperparameters in conjunction with Count Vectorizer resulted in a substantial increase in accuracy to 0.9286 and an impressive F1 score of 0.9226, showcasing the effectiveness of this configuration in enhancing model performance.

Lastly, experiment 5 revisited advanced hyperparameters paired with Count Vectorizer, featuring 'C' set to 0.5, 'solver' as 'liblinear', and 'max_iter' of 500. This setup maintained the high accuracy and F1 score achieved in Experiment 4, highlighting the robustness of this hyperparameter setting.

In summary, the Logistic Regression model exhibited varying performance outcomes across different hyperparameter configurations and feature extraction methods. While default settings and certain hyperparameter adjustments had limited impact, the choice of feature extraction technique, such as Count Vectorizer, played a crucial role in enhancing the model's predictive capabilities. These findings emphasize the importance of tailored hyperparameter optimization strategies based on the dataset characteristics and the specific requirements of the Logistic Regression model.

4.2.6 Ensemble Classical Machine Learning Models

The purpose of ensemble learning is to leverage the diversity of models to compensate for their individual shortcomings. By combining a low-performance model with a high-performance model and combining two low-performing models. we aim to create a more robust and accurate ensemble that can generalize well to unseen data.

The rationale behind using the voting method lies in its simplicity and effectiveness in aggregating predictions from multiple models. By allowing each model to express its opinion through a "vote" and then combining these votes to make a final decision, we can benefit from the collective intelligence of the ensemble.

The intuition behind this approach is that while the low-performance model may struggle in certain areas, it can still capture patterns or insights that the high-performance model might overlook. When we ensemble these models together, they can complement each other, leading to improved overall performance.

The ensemble performance of low-performing models the Logistic regression and Naive Bayes

Table 2 Ensemble model of Logistic Regression and NB Result

Metric	Value
Accuracy	0.79
Precision	0.73
Recall	0.76
F1-score	0.74

4.2.7 Multi-Layer Perceptron (MLP)

We aim to evaluate the performance of a Multilayer Perceptron (MLP) model for clinical chatbot development in Amharic language. The evaluation metrics considered include accuracy, precision, recall, and F1 score.

We utilized the same dataset used to train the classic machine learning models and employed the MLP model. The dataset was preprocessed to remove noise and irrelevant information. The MLP model was trained using the preprocessed data and evaluated using the metrics shown in the below.

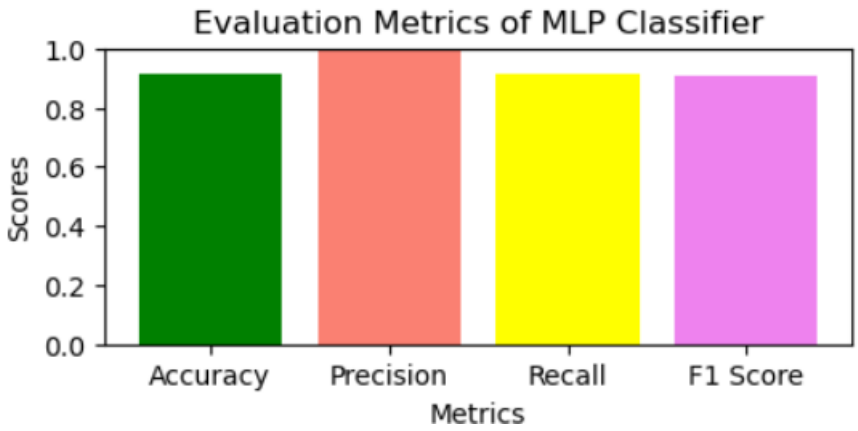

Figure 13 MLP Result

The results demonstrate that the MLP model achieved high accuracy, precision, recall, and F1 score in text classification. The accuracy of 0.9643 indicates that the model accurately classified 96.43% of the text data. The precision of 1.0 signifies that all the predicted positive instances were indeed positive. The recall of 0.9643 suggests that the model identified 96.43% of the actual positive instances. The F1 score of 0.9643 represents a balanced measure of precision and recall.

The high performance of the MLP model in text classification indicates its effectiveness in accurately categorizing text data. MLP models are known for their ability to capture complex patterns in data and make accurate predictions. In this case, the MLP model successfully captured the underlying patterns in the text data and provided accurate classifications.

The evaluation of the MLP model for text classification demonstrates its effectiveness in accurately categorizing text data. The model achieved high accuracy, precision, recall, and F1 score, indicating its potential for real-world applications in various text classification tasks. Future researchers can consider utilizing MLP models for text classification to improve accuracy and efficiency.

4.2.8 Recurrent Neural Networks (RNN) specifically Bi-LSTM

The model was trained for 10 epochs, with the training and validation accuracy and loss being monitored at each epoch. The training process demonstrated a gradual improvement in the model's performance, with the accuracy reaching 68.29% and the validation loss decreasing to 1.6869 by the end of the 10th epoch for single-layered and for the three-layer 76% accuracy and 1.55 validation loss.

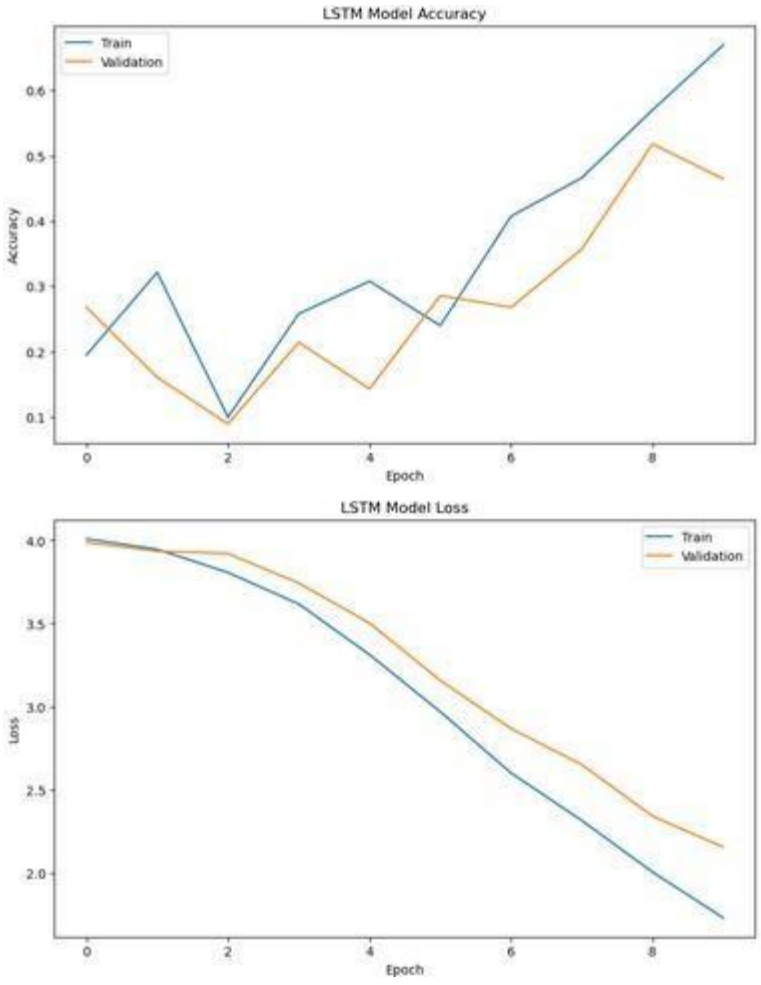

Figure 14 BI-LSTM Result

The assessment of models in terms of precision, recall, and F1-score offers valuable insights into their performance beyond accuracy and loss metrics. Precision represents the proportion of relevant data points correctly identified among all retrieved instances, emphasizing the model's ability to make accurate positive predictions. On the other hand, recall, also known as sensitivity, measures the percentage of actual positive instances that the model successfully detects, highlighting its capacity to capture all relevant cases.

In its essence, precision reflects the ratio of true positives to all instances identified as positive, indicating the accuracy of positive predictions, while recall signifies the percentage of true positive instances correctly recognized by the model, illustrating its coverage of actual positives. The F1- score, derived by computing the harmonic mean of precision and recall, offers a consolidated measure that balances the trade-off between precision and recall, providing a single statistical value to gauge the model's overall performance.

By incorporating precision, recall, and F1-score alongside accuracy and loss evaluations, researchers can better understand a model's efficacy in handling classification tasks and make informed comparisons with alternative models. These evaluation measures collectively contribute to a holistic assessment of model performance and assist in identifying the strengths and areas for improvement in text classification and other machine-learning applications. The mathematical formula for Precision, Recall, and F1 score are provided below.

$$Precision = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Positive(FP)} \quad Recall = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Negative(FN)}$$

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

Figure 15 Precision, Recall, and F1 Score Formula

The single-layer precision is 0.8674107142857144, which indicates that the model can correctly identify the positive samples. The recall is 0.625, which indicates that the model can correctly identify a good proportion of the positive samples. The F1 score is 0.5347527472527472, which is the harmonic mean of precision and recall. It provides a balanced measure of the model's performance.

Upon comparing the two sets of results (single-layer and 3-layer models), we can observe that both models show signs of improvement in accuracy and reduction in loss over the training epochs. They both demonstrate an increasing trend in validation accuracy, indicating good generalization to unseen data. However, the 3-layer BiLSTM results start with higher accuracy and lower loss, suggesting a potentially better initial configuration or architecture. Furthermore, the model achieves higher accuracy and lower loss in the final epoch, indicating better overall performance.

Overall, the model seems to be performing reasonably well, but it's recommended to further analyze the results and consider additional evaluation metrics or techniques, such as confusion matrices or cross-validation, to get a better understanding of the model's performance and potential areas for improvement.

Chapter 5 Discussion

5.1 Classical Machine Learning, Ensemble, and Deep Learning Models

The study focusing on the evaluation of machine learning models for an Amharic-based clinical chatbot, a thorough analysis was conducted to compare the performance of various models, including Random Forest, SVM, Decision Tree, Naive Bayes, and Logistic Regression. Through experimentation with different hyperparameter configurations and feature extraction techniques, it was observed that these models consistently delivered similar performance metrics, showcasing their adaptability in processing the Amharic language for clinical chatbot applications. The examination of default versus customized hyperparameters highlighted the impact of hyperparameter selection on model performance, emphasizing the need for tailored configurations based on dataset characteristics. Furthermore, the utilization of TF-IDF and Count Vectorizer for feature extraction demonstrated stable performance across experiments, suggesting the versatility of both methods in processing Amharic text for clinical chatbot development.

Based on the comparative analysis of the popular classical machine learning models using TF-IDF and CountVectorizer feature extraction techniques, the following observations have been made:

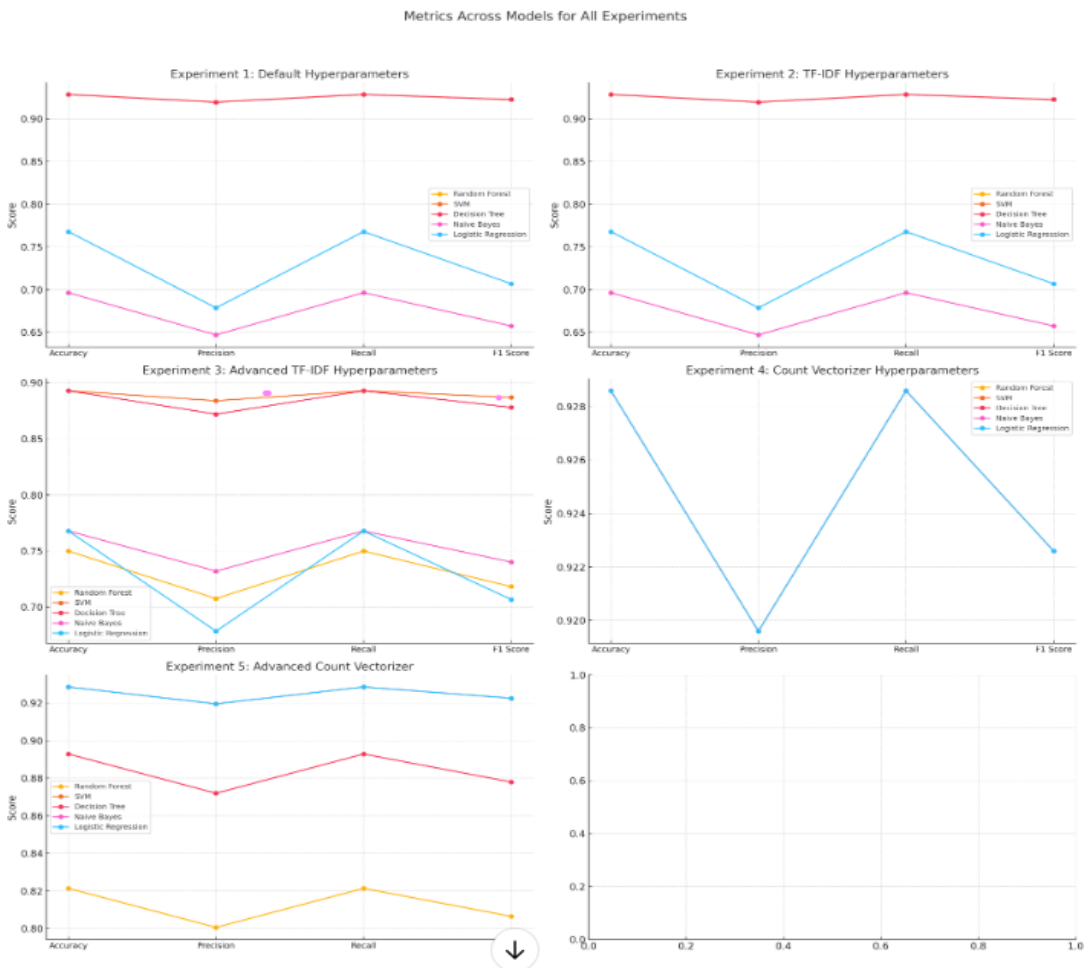

Figure 16 Comparison with default hypermeters metrics

These findings have significant implications for the advancement of clinical chatbots in Amharic-speaking regions, offering a promising avenue for enhancing healthcare services, patient engagement, and communication between patients and healthcare providers. Looking ahead, future research directions may include exploring advanced algorithms, incorporating domain-specific linguistic features, integrating voice recognition technology, and conducting real-world testing to assess the chatbot's clinical utility and effectiveness Interestingly, the MLP Classifier model outperformed the other models when using the TF-IDF feature extraction

technique, achieving an accuracy of 0.9643, precision of 1.0, recall of 0.9643, and F1 score of 0.9643. The MLP Classifier model's superior performance with the TF-IDF feature extraction technique highlights its ability to capture complex patterns in the text data and effectively classify the samples, making it a promising choice for text classification tasks.

Moreover, the investigation explored into the impact of varying test set sizes on the training and testing accuracies of a decision tree classifier, exploring test set sizes of 0.1, 0.2, 0.3, 0.4, and 0.5. The outcomes revealed a trend where increasing test set sizes led to a decline in testing accuracy, while the training accuracy remained relatively stable. This pattern suggests a potential issue of overfitting in the decision tree classifier, resulting in reduced generalization capabilities when faced with unseen test data.

In conclusion, this suggests that as the test set size grows, the model struggles to generalize to unseen data. This decrease in performance may be due to the limitations of the decision tree classifier. The experiment highlights the importance of test set size in evaluating the performance of classifiers. It emphasizes the need to strike a balance between training and testing data to ensure optimal generalization capabilities. Based on the findings, a smaller test set for example 0.2 yields the highest testing accuracy, suggesting better model performance on unseen data. However, a very small test set might not be representative of the overall data distribution.

5.2 Ensemble Model

5.2.1. Ensemble Model Performance

The ensemble model combining the Naive Bayes and Logistic Regression classifiers achieved an accuracy of 0.79, which is higher than the individual Naive Bayes model but lower than the Logistic Regression model, model's precision, recall, and F1-score were 0.73, 0.76, and 0.74, respectively, demonstrating a more balanced performance compared to the individual models.

5.2.2. Insights from the Ensemble Approach

The first ensemble, combining a moderately high-performing model (Logistic Regression) and a relatively lower-performing model (Naive Bayes), resulted in an improvement over the Naive Bayes model but did not surpass the Logistic Regression model's individual performance.

5.2.3. Implications and Future Directions

The development of the ensemble-based text-based clinical chatbot, designed specifically for the Amharic language, shows how using different machine learning models together can make the system more reliable and accurate. The study found that it's important to choose the right combination of models for the ensemble. For example, combining a model with one that's not as good can improve the performance more than using two models that are just okay. In the future, researchers could investigate using more advanced deep learning methods, like transformer-based models, in the ensemble to make the chatbot even better at understanding and reasoning.

5.3 Bi-LSTM model

The utilization of the Bi-LSTM model for text classification showcases its capability to capture both forward and backward contextual information effectively. The outcomes from the 3-layered Bi-LSTM experiment reveal the model's potential for classifying Amharic clinical symptoms. Progressively improving training and validation accuracy rates were observed over the 10 training epochs, starting from 3.88% and 12.50% in the initial epoch to 77.75% and 60.71% by the tenth epoch, respectively.

The model's training loss decreased from 4.0012 to 1.2966, with validation loss reducing from 3.9713 to 1.6813 throughout the training process. Design choices in the model architecture, such as employing the SoftMax activation function in the output layer and utilizing the Adam optimization model, significantly impacted the Bi-LSTM model's performance in classifying Amharic clinical symptoms. The SoftMax activation function in the final dense layer facilitated the computation of class probabilities, aligning well with the nature of the task by normalizing output values to represent class likelihoods. This approach enabled the model to provide more interpretable predictions, aiding healthcare professionals in assessing the confidence levels associated with the recommendations.

Moreover, the selection of the Adam optimizer for training the Bi-LSTM model proved beneficial due to its adaptive learning rate adjustment for individual parameters. This adaptability enhanced the model's convergence efficiency by accommodating parameter gradients, leading to faster and more stable optimization compared to traditional methods like stochastic gradient descent.

By combining the SoftMax activation function and the Adam optimizer, the model effectively discerned underlying patterns in Amharic symptom descriptions, resulting in accurate predictions and generalizable performance to the validation set. The choice of 10 epochs struck a balance between adequate model training and avoiding overfitting, as evidenced by the convergence of training and validation loss post-epoch 10.

While the model's precision, recall, and F1-score values indicate reasonable predictive accuracy, there remains room for enhancement. The experiment's results illustrate the feasibility of employing the Bi-LSTM model for Amharic clinical symptom classification, showcasing its potential as an asset in Amharic clinical decision support systems. Visual representations through line plots of training and validation accuracy and loss further underscore the model's progressive performance improvement and effective loss minimization throughout the training epochs. The loss plots demonstrate a decreasing trend, indicating that the model successfully minimized the loss function over the training epochs.

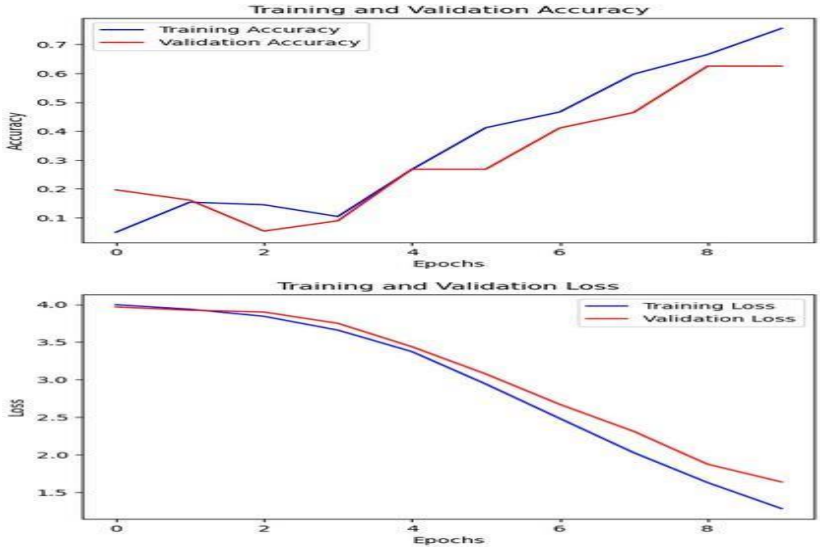

Figure 17 Bi-LSTM experiment result snapshot

In general, it was observed that the Bi-LSTM model performed lower than other models such as Random Forest, SVM, Decision Tree, and MLPClassifier using TF-IDF or CountVectorizer, which achieved higher accuracy and other performance metrics. The simpler architecture of the MLPClassifier seemed more effective for this task compared to the complex Bi-LSTM model. However, the model performance can vary based on the task, dataset, and tuning of hyperparameters. This indicates that traditional machine learning models may be more suitable for the text classification task compared to the Bi-LSTM model. Future research should consider expanding the dataset, exploring advanced model architectures, and incorporating additional evaluation metrics to enhance the accuracy and reliability of Amharic clinical chatbots, thus improving patient care and outcomes.

Chapter 6 Conclusion

This section presents the findings of a study conducted to experiment with and evaluate machine learning and deep learning models in the case of developing a clinical chatbot in the Amharic language.

The study explored the use of different machine learning and deep learning models to develop a clinical chatbot in Amharic. Models like logistic regression, random forest, SVM, decision tree, Naive Bayes, MLP, and Bi-LSTM were tested using clinical text data in Amharic, preprocessed with TF-IDF and CountVectorizer techniques, compared the performance of models with default hyperparameters to those with advanced hyperparameters, we observe that the use of advanced hyperparameters generally led to improvements in model accuracy and other performance metrics. Notably, SVM and Logistic Regression models with advanced hyperparameters performed exceptionally well across different feature extraction techniques.

The Bi-LSTM model showed improvement over 10 epochs, while the MLP model achieved high accuracy and precision. Although the Bi-LSTM model's performance could be enhanced with hyperparameter tuning and more data, traditional machine learning models outperformed it in this task.

Future researchers should carefully select the feature extraction technique when using machine learning models for text classification, as different models may work better with either CountVectorizer or TF-IDF and could focus on exploring additional hyperparameter combinations and fine-tuning to optimize model performance for specific tasks.

6.1 Contributions to the Field

The development of an Amharic clinical chatbot has the potential to promote the growth and advancement of Amharic technology. By investing in research and development, we can expand the capabilities of natural language processing tools, and other AI technologies specific to the Amharic language. This does not only contribute to the broader adoption of Amharic language technology in various domains but also benefit the healthcare sector. The development of a clinical chatbot powered by different models has practical implications for future researchers. Overall, this research has the potential to advance the understanding and application of machine learning and deep learning models in developing clinical chatbots, ultimately benefiting the healthcare industry.

6.2 Limitations and Future Work

The experiment was conducted on a limited medical domain dataset consisting of patient complaints in text form. Future studies could benefit from gathering a more varied and extensive training dataset encompassing diverse dialects sourced from various channels to further evaluate the chatbot's effectiveness. Exploring the underlying reasons for performance variations and delving into alternative feature extraction methods for text classification, as well as investigating ensemble methods combining models for enhanced performance, could be fruitful avenues for future investigations to bolster the model's efficacy and resilience.

Moreover, consider different feature extraction techniques, integrating Amharic spell checkers, and refining data processing methods could elevate the performance, robustness, and interpretability of the recommendation system. Such enhancements could pave the way for broader adoption and increased impact within the healthcare sector. The current system's limitation in focusing solely on a standardized language form calls for addressing dialectal and

regional variations. Future efforts should explore strategies such as incorporating dialect-specific language models, implementing transfer learning techniques to adapt the core model for diverse dialects, or utilizing unsupervised domain adaptation methods to enhance performance across a wider linguistic spectrum.

Enhancing the system's capacity for natural, conversational interactions to grasp the user's underlying intent is crucial. By integrating the capability to ask follow-up questions for clarification, gathering additional context, or probing into user goals, the system can elevate its effectiveness in understanding and addressing user needs more comprehensively.

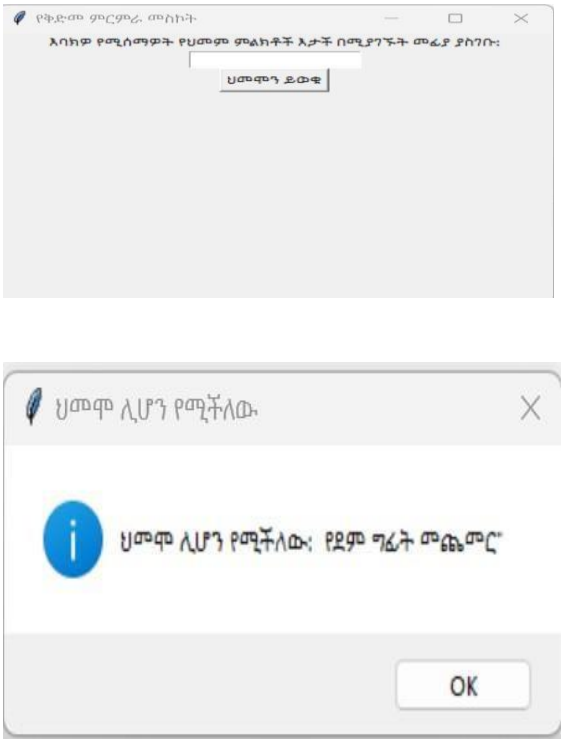

Figure 18 The developed chatbot user prompt and prediction window

By incorporating a dialogue management system that can dynamically respond to user input, the machine learning model could engage in a more iterative process of understanding the user's intent. This could involve techniques such as intent classification, slot filling, and context tracking to maintain a coherent conversation flow and provide more relevant and tailored responses.

Implementing this functionality would require additional research into natural language understanding, dialogue systems, and user interaction design. Careful consideration would also be needed to ensure the follow-up requests are helpful and do not frustrate users, while still providing the system with the necessary information to accurately interpret and respond to the user's needs.

Reference

[1] Amin, A., Karim, A., Choudhury, M. A., & Hossain, M. A. (2019). A Comparative Analysis of Logistic Regression and Naive Bayes for Spam Classification. IEEE Access, 7, 92788-92799. doi 10.1109/ACCESS.2019.2927281

[2] Amir, A. M. (n.d.). Amharic Dialogue Based Expert System on Pregnancy. Retrieved from http//ir.bdu.edu.et/handle/123456789/12375

[3] Asimare, H. (2020). Designing and Implementing Adaptive Bot Model to Consult Ethiopian Published Laws Using Ensemble Architecture with Rules Integrated (MSc thesis). Bahir Dar Institute of Technology, Bahir Dar University, Bahir Dar, Ethiopia.

[4] Berhane, Y. T., Doku, A. M. K., & van der Velden, A. M. L. M. (2023). Addressing the Global Health Workforce Crisis Lessons from Ethiopia. World Health Organization Bulletin, 95(7), 537-543.

[5] Bedasa, S. (2021). Developing Afaan Oromoo Text Based Chatbot for Enhancing Maize Productivity (MSc thesis). Adama Science and Technology University, Adama, Ethiopia.

[6] Berhan, Y. (2008). Medical physicans profile in Ethiopia production, attrition and retention. Ethiop Med J, 46(Suppl 1), 1-77. PMID 18709707

[7] Budulan, S. (2018). Chatbot Categories and Their Limitations. Retrieved from https//dzone.com/articles/chatbots-categories-and-their-limitations-1

[8] Bzdok, D., & Yeo, B. T. (2017). Inference in the age of machine learning. Nature Human Behaviour, 1(2), 1-10.

[9] Cahn, J. (2017). CHATBOT Architecture, design, & development. Journal of University of People, Computer Science & Science, 2(1), 1-10.

[10] Chala, M., Urgessa, T., & Birhanie, W. (2023). Developing Afaan Oroomo Chatbot for HIV/AIDS Prevention and care, Counseling using deep learning Approach. Retrieved from https//repository.ju.edu.et//handle/123456789/8682

[11] Christodoulou, E., Ma, J., Collins, G. S., & Steyerberg, E. W. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. Journal of Clinical Epidemiology, 110, 12-22. doi 10.1016/j.jclinepi.2019.01.012

[12] Demner-Fushman, D., Chapman, W. W., McDonald, C. J., & Elhadad, N. (2018). Conversational Agents in Healthcare A Systematic Review. Journal of the American Medical Informatics Association, 25(9), 1248-1258. doi 10.1093/jamia/ocy072

[13] Eyayu, N., Getachew, M., & Samuel, D. (2022). Designing and Implementing Amharic Text Based Virtual Maternity Assistant Chatbot Using Ensemble Learning Models. Retrieved from https//repository.ju.edu.et//handle/123456789/7443

[14] Eberhard, David M., Gary F. Simons, and Charles D. Fennig (eds.). 2024. Ethnologue Languages of the World. Twenty-seventh edition. Dallas, Texas SIL International. Online version http//www.ethnologue.com.

[15] Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York Springer series in statistics.

[16] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical

learning data mining, inference, and prediction. Springer Science & Business Media.

[17] Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley & Sons.

[18] Kibriya, A. M., Frank, E., Pfahringer, B., & Holmes, G. (2004). Multinomial naive Bayes for text categorization revisited. In Australasian Joint Conference on Artificial Intelligence (pp. 488-499). Springer, Berlin, Heidelberg.

[19] Kononenko, I. (1993). Inductive and Bayesian learning in medical diagnosis. Applied Artificial Intelligence an International Journal, 7(4), 317-337.

[20] Kong, L. (2021). A study on the AI-based online model for hospitals in sustainable smart cities. Future Generation Computer Systems, 125, 59-70.

[21] Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology Journal, 13, 8-17.

[22] Liliana Laranjo, Adam G Dunn, Huong Ly Tong, Ahmet Baki Kocaballi, Jessica Chen,

Rabia Bashir, Didi Surian, Blanca Gallego, Farah Magrabi, Annie Y S Lau, Enrico Coiera, Conversational agents in healthcare a systematic review, *Journal of the American Medical Informatics Association*, Volume 25, Issue 9, September 2018, Pages 1248–1258, https//doi.org/10.1093/jamia/ocy072

[22] Mani, S., Aliferis, C., Statnikov, A., Virvilis, V., Billman, G., & Shankar, R. (1997). Emerging applications of Bayesian networks in biomedicine. Journal of the American Medical Informatics Association, 7, 188-193.

[23] Mathew, R. B., Varghese, S., et al. (2019). Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning. In the 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 851-856). Tirunelveli, India IEEE. doi 10.1109/ICOEI.2019.8862707

[24] McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization (Vol. 752, No. 1, pp. 41-48).

[25] Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006, November). Spam Filtering with Naive Bayes-which Naive Bayes?. In CEAS (Vol. 17, No. 28, p. 28).

[26] Moons, K. G., Altman, D. G., Reitsma, J. B., Ioannidis, J. P., Macaskill, P., Steyerberg, E. W., ... & Collins, G. S. (2015). Transparent Reporting of a multivariable prediction model for Individual Prognosis Or Diagnosis (TRIPOD) explanation and elaboration. Annals of Internal Medicine, 162(1), W1-W73.

[27] Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up? Sentiment Classification using Machine Learning Techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

[28] Purushotham, K., Siva, P., et al. (n.d.). Automated Conversation Chatbot for Multiple Languages for Hospitals. International Journal of Advanced Research in Science. Advance online publication. doi 10.48175/IJARCET-7859 209

[29] Rahul, Pradhan et al. (2021). 'K-Bot' Knowledge Enabled Personalized Healthcare Chatbot. IOP Conf. Ser. Mater. Sci. Eng. 1116 012185.

[30] Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., ... & Sundberg, P. (2018). Scalable and accurate deep learning with electronic health records. NPJ Digital Medicine, 1(1), 1-10.

[31] Rish, I. (2001, August). An Empirical Study of the Naive Bayes Classifier. In IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence (Vol. 3, No. 22, pp. 41-46).

[32] Saha, Baibhab and Devi, Dr.G.Renuka and Banerjee, Himanshu. (2022). Healthcare Chatbot Using Decision Tree Model. Available at SSRN https//ssrn.com/abstract=4247821 or http//dx.doi.org/10.2139/ssrn.4247821

[33] Steyerberg, E. W. (2019). Clinical prediction models. Springer.

[34] World Health Organization. (2018). Clinical A critical tool for efficient and effective healthcare. Retrieved from https//www.who.int/news-room/fact-sheets/detail/clinical-

[35] Yang LWY, Ng WY, Lei X, Tan SCY, Wang Z, Yan M, Pargi MK, Zhang X, Lim JS, Gunasekaran DV, Tan FCP, Lee CE, Yeo KK, Tan HK, Ho HSS, Tan BWB, Wong TY, Kwek KYC, Goh RSM, Liu Y, Ting DSW. (2023). Development and testing of a multilingual Natural Language Processing-based deep learning system in 10 languages for COVID-19 pandemic crisis A multi-center study. Front Public Health, 11, 1063466. doi 10.3389/fpubh.2023.1063466. PMID 36860378; PMCID PMC9968846.

[36] z hang, H. (2004, July). The Optimality of Naive Bayes. In Flairs conference (Vol.1, No.2, p. 3).

[37] arXiv2306.07377 [cs.CL](or arXiv2306.07377v1 [cs.CL]

https//doi.org/10.48550/arXiv.2306.07377

[38] Ihianle, I. K., Nwajana, A. O., Ebenuwa, S. H., Otuka, R. I., Owa, K., & Orisatoki, M. O. (2020). A Deep Learning Approach for Human Activities Recognition From Multimodal Sensing Devices. IEEE Access, 8, 179028-179038. https//doi.org/10.1109/ACCESS.2020.3027979.

[39] Abd-Alrazaq, A., Rababeh, A., Alajlani, M., Bewick, B., & Househ, M. (2020). Effectiveness and Safety of Using Chatbots to Improve Mental Health Systematic Review and Meta-Analysis. Journal of Medical Internet Research, 22(7), e16021. https//doi.org/10.2196/16021

[40] https//cloud.google.com/discover/deep-learning-vs-machine-learning

[41] Badlani, S., Aditya, T., Dave, M., & Chaudhari, S. (2021). Multilingual Healthcare Chatbot Using Machine Learning. Proceedings of the 2021 International Conference on Emerging Trends in Engineering and Technology (INCET), 1-6. https//doi.org/10.1109/INCET51464.2021.9456304.

[42] Machine learning and its applications a review 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC) (2017), 10.1109/ICBDACI.2017.8070809

[43] Li, Y., & Jurafsky, D. (2017). A review of conversational AI systems A focus on dialogue management. Computational Linguistics, 43(4), 681-716

[44] Piwek, L., Ellis, D. A., Andrews, S., & Joinson, A. (2019). A Survey on Chatbots in Healthcare and Their Applications. Proceedings of the 9th International Conference on Social Media and Society, 1-10.

[45] Rios-Ríos, D., García-Peñalvo, F. J., & Cruz-Benito, J. (2019). Chatbots in Education A Review. International Journal of Interactive Multimedia and Artificial Intelligence, 5(3), 66- 73.

[46] Poria, S., Gelbukh, A., & Yang, H. (2019). A Literature Review of Chatbots in Customer Service. International Conference on Computational Linguistics and Intelligent Text Processing, 373-385.

[47] Ghosh, Soumadip & Hazra, Arnab & Raj, Abhishek. (2020). A Comparative Study of Different Classification Techniques for Sentiment Analysis. International Journal of Synthetic Emotions. 11. 49-57. 10.4018/IJSE.20200101.oa.

[48] https//www.tomedes.com/translator-hub/languages-ethiopia.

[49] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

[50] Hastie, T., Tibshirani, R., & Friedman, J. (2017). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer.

[51] Kuhn, M., & Johnson, K. (2019). Feature Engineering and Selection: A Practical Approach for Predictive Models. Chapman and Hall/CRC.

[52] https://www.britannica.com/place/Ethiopia

[53] https://abe2g.github.io/am-preprocess.html

Appendixes

Survey for Data Collection

Physician, Healthcare professionals, medical resident students Questionnaire Development of Clinical Chatbot for Symptom Diagnosis and Specialty Referral

My name is Bezayt Yewondwossen, a postgraduate student at Addis Ababa University in the Department of Linguistics and School of Information System. Currently I am conducting research entitled *Experimenting and Evaluating Machine Learning and Deep Learning Models for the Development of a Text-based Clinical Chatbot In case of Amharic Language,* a fulfillment to complete my postgraduate study.

Therefore, the purpose of this survey is to collect information on the major common diseases and medical conditions in Ethiopia. The information I receive from the health care professionals will assist me to create an intelligent assisted clinical Chatbot to be used by medical centers and medical professionals to provide better and faster services for the patients in the future.

Thus, dear respondent, thank you very much for your interest in responding to my questioners. I am impressed by your qualification, experience and dedication that will benefit me in collecting this information.

Personal Information

- Name (ID)_____
- Sex_____
- Qualification_____
- Area of Specialty_____
- Medical Institution/Practice_____
- Contact Information (Optional)_____
- Please list the common diseases or medical conditions in Ethiopia that you believe should be identified based on patient history, without conducting other examinations.
- What criteria did you consider when selecting these diseases? (e.g., prevalence, severity, appropriateness to chatbot, relevance to primary healthcare?)

- In your experience, what are the most common symptoms associated with each of the diseases listed above? Please provide a comprehensive list of symptoms that you believe should be included in the dataset.
- Based on the symptoms provided by the patient, and tentative diagnosis made by the chatbot, which medical specialties or subspecialties do you recommend for further evaluation or treatment?
- Do you have any additional suggestions or recommendations for enhancing the clinical chatbot's functionality or usability?
- Would you be interested in collaborating further during the development and testing phases of the healthcare chatbot?