Amharic Text Based Chatbot for Driving Trainee Using Machine Learning Approach

By
Adanu Amenu Yigezu

A Thesis Submitted to the Department of Computer Science and Engineering, School of Electrical Engineering and Computing Presented in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

September, 2024
Adama, Ethiopia

Amharic Text Based Chatbot for Driving Trainee Using Machine Learning Approach

By

Adanu Amenu Yigezu

Advisor

Bahiru Shifaw (Ph.D.)

A Thesis Submitted to the Department of Computer Science and Engineering, School of Electrical Engineering and Computing Presented in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2024

Adama, Ethiopia

DECLARATION

I declare that this thesis entitled **"Amharic Text Based Chatbot for Driving Trainee Using Machine Learning Approach"** is my own work and has not been submitted to any university for similar purpose. The references used in this proposal are duly recognized by proper citations.

Name of Student:	**Signature:**	**Date:**
Adanu Amenu Yigezu		September, 2024

RECOMMENDATION

I, the major advisor of this thesis research, hereby certify that I have closely advised the student while developing and read the draft thesis entitled **"Amharic Text Based Chatbot for Driving Trainee Using Machine Learning Approach"** prepared under my guidance by Adanu Amenu Yigezu. Therefore, I recommend the submission of the thesis to the department for further review and evaluation.

Name of Advisor: **Signature:** **Date:**

Bahiru Shifaw (Ph.D)

APPROVAL PAGE

I/we hereby certify that the recommendation and suggestion given by the thesis review committee are appropriately incorporated into the final thesis/dissertation entitled **"Amharic Text Based Chatbot for Driving Trainee Using Machine Learning Approach"** by Adanu Amenu Yigezu.

Name of Advisor: **Signature:** **Date:**

Bahiru Shifaw (Ph.D)

Approval of Board of Reviewers

We, the undersigned, members of the Board of Reviewers of the thesis open defense by Adanu Amenu Yigezu have read and evaluated the thesis entitled **"Amharic Text Based Chatbot for Driving Trainee Using Machine Learning Approach"** and assessed the understanding of the candidate about the proposed research. This is, therefore, to certify that the thesis proposal is accepted and we recommend the implementation of the proposal **"Amharic Text Based Chatbot for Driving Trainee Using Machine Learning Approach".**

Chairperson	Signature	Date
Reviewer 1	Signature	Date
Reviewer 2	Signature	Date

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

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

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to God, whose guidance and grace have been my constant source of strength throughout this journey. His wisdom and blessings have provided me with the perseverance and courage needed to complete this study, and for that, I am eternally grateful.

My heartfelt thanks go to my family, whose unwavering support and encouragement have been invaluable to me. My parents, whose love and sacrifices have paved the way for my academic and personal growth, deserve special recognition. Their belief in my abilities has been a driving force behind my success. I also want to extend my appreciation to my siblings and extended family for their constant motivation and understanding during the demanding phases of this research.

I am deeply indebted to my advisor Bahiru Shifaw, whose expertise, patience, and insightful guidance have been instrumental in shaping this study. His constructive feedback and encouragement have challenged me to think critically and strive for excellence. Without his mentorship, this research would not have reached its current level of depth and rigor.

Finally, I would like to acknowledge the support of my friends, colleagues, and all those who have contributed to this study in various ways. Whether through intellectual discussions, technical assistance, or moral support, their contributions have been crucial in helping me achieve this milestone. I am truly thankful to everyone who has played a role in this academic journey.

Table of Contents

ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
AIML	Artificial Intelligence Markup Language
ANN	Artificial Neural Network
Bi-LSTM	Bi-directional Long Short-Term Memory
DNN	Deep Neural Networks
JSON	JavaScript Object Notation
KNN	K-Nearest Neighbours
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Processing
RNN	Recurrent Neural Network
SVM	Support Vector Machines

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The increasing demand for effective and user-friendly Amharic chatbots, particularly in the context of assisting driver's license trainees, highlights the need for advanced natural language processing (NLP) systems tailored to the Amharic language. This study addresses the challenge of developing a chatbot capable of understanding and responding accurately to user queries in Amharic, a language with a complex writing system and limited resources for NLP research. The research aims to design and implement an Amharic chatbot that can provide accurate responses to driver's license-related queries, thereby improving the learning experience for trainees.

The study employs a comprehensive methodology, beginning with the collection and preprocessing of Amharic text data, followed by the application of various text normalization, cleaning, tokenization, and stop-word removal techniques. The vectorized textual data is then used to train a machine learning model using the Keras Sequential API, with an 80/20 split between training and validation datasets. The model architecture includes a deep neural network with three layers, optimized through a series of heuristic adjustments. The study also explores different text representation methods, with a focus on the bag-of-words (BoW) approach, to find the most suitable technique for the limited Amharic text corpus.

Experimental scenarios were conducted to evaluate the performance of the model, using accuracy and loss metrics. The chatbot achieved an accuracy of 87.9% on the test dataset, demonstrating its potential effectiveness in real-world applications. TensorBoard was employed to visualize the model's performance over 150 training epochs, revealing the model's strong generalization capabilities.

The key findings indicate that the proposed chatbot model is capable of providing accurate and contextually appropriate responses to user queries, making it a valuable tool for license trainees. The study's implications suggest that future research should focus on expanding the Amharic text corpus, exploring more advanced NLP techniques, and refining the model to enhance its conversational flexibility and accuracy. This research contributes to the growing field of Amharic NLP and sets the stage for further advancements in chatbot development for under-resourced languages.

Keywords: Amharic Chatbot, NLP, Machine learning, ANN, Driving Trainee

CHAPTER ONE
INTRODUCTION

1.1. Background of the Study

After the rise of the web and mobile apps, the role of information technology is increasing from time to time. Technology provides a platform to understand computer-human language through Natural Language Processing (NLP) techniques. NLP is an AI method of communicating with intelligent systems using a natural language processing and required when an intelligent system like robot to perform different activities based on our natural language instructions. NLP researchers deal with analyzing, understanding, and generating the languages of human instead of computer language. Currently, AI builds a technology that allows us to interact with machines as in normal human-to-human conversation. Hence, the role of natural language processing is great. NLP was applicable in different business areas like sentiment analysis, chatbots, customer service, managing the advertisement funnel, market intelligence.

When it comes to application of NLP in chatbots, it requires various data preprocessing as Chatbot is a software application that is used for voice or text-based conversations between humans and computers. According to (Chung & Park, 2019), a chatbot is an intelligent interactive platform that can interact with users through a chatting interface. Since it can connect with the major social network service (SNS) messengers, general users can access the platform easily and receive a variety of services. In the current era, Chatbot plays a great role in healthcare, insurance, banking, financial service, etc. Based on the development approach there are two types of chatbot. Those are rule-based (set of predefined questions and responses) and AI-based chatbots that is by learning questions and produce a replay on their own based on natural language processing technology.

Since its inception in the 1960s, chatbot systems have advanced significantly. Machine learning, natural language processing, and computer science's hardware and software all have made significant advancements. Chatbots have developed from machines that provide machine-like responses to agents that resemble people and are able to forge lasting connections with users as a result of the development of these emerging technologies. ELIZA and PARRY are two of the most well-known early chatbot implementations. Microsoft's XiaoIce, Apple's Siri, Amazon's Alexa, and other companies have modern chatbots as well (Xufei, 2021).

Currently, machine-learning algorithms have been applied in different areas such as NLP, computer vision, etc. There are different types of techniques in machine learning that used to solve different existing problems. From the existing problem, assisting an organization customer or assisting people using conversational agent is one of the commonly used applications in various domains. Therefore, chatbot is ideally one of machine learning application area that is used to solve such problem.

Therefore, in this study, we propose conversational AI Chatbot model, specifically for Amharic language, to support preparing for driving license test by interacting in a more human like way. The proposed Chatbot system aims at better accessibility, personalization, and efficiency of conversational agents that have the potential to improve the driver's competence based on task oriented conversational system. The system responds to various queries in real-time to consult and inform first time drivers during their preparation for driving license test. The system also responds to queries comes from novice drivers in order to enhance their driving skill. The proposed Chatbot will be based on machine learning model which is trained over documents about driving skills, road safety rules and regulations. The required documents are available and published in Amharic text.

1.2. Statement of the Problem

Road traffic crashes account for the deaths of an estimated 1.35 million people globally each year, with over 50 million suffering injuries. The brunt of these accidents is borne by low- and middle-income countries, where road traffic fatalities have remained stubbornly high, with no reductions in low-income nations since 2013 (United Nations, 2020). Ethiopia, with 65 fatalities per 10,000 vehicles, ranks among the highest in road traffic deaths per vehicle worldwide (UNIDO, 2021). According to Ethiopia's Ministry of Transport, a staggering 87% of road accidents in 2012 E.C. were linked to driver incompetence, which is largely attributed to insufficient driver education and inadequate traffic safety awareness. While government strategies have focused on uniform driving laws and punitive measures, the persistence of road accidents signals a fundamental problem in driving education and the acquisition of safe driving habits.

Despite efforts to improve road safety, one of the most critical barriers remains the limited accessibility and quality of driver education, particularly for Amharic-speaking communities. The scarcity of personalized, Amharic-based learning materials hampers the effectiveness of conventional training methods, leaving many learners without adequate resources to fully grasp the complexities of driving rules and road safety regulations. Existing instructional approaches are largely static and one-size-fits-all, often delivered in languages other than Amharic, creating a disconnect between the language of instruction and the learner's understanding. This problem is further compounded by the linguistic and cultural nuances unique to Amharic-speaking regions, making it difficult for trainees to engage effectively with the material. The lack of interactive, user-friendly, and linguistically appropriate educational tools results in poor retention of essential driving concepts and inadequate preparation for driving tests. These gaps underscore the need for an innovative, AI-driven solution that addresses the educational and linguistic challenges faced by Amharic-speaking license trainees.

By leveraging conversational AI, this study aims to develop a chatbot that enhances the accessibility and quality of driver education for Amharic speakers, providing tailored support to improve road safety outcomes in Ethiopia.

1.3. Research Questions

The study addresses the following research questions:

- How to prepare a dataset for designing Amharic chatbot that provide appropriate answers for license trainee?
- Which suitable vector representation of text to use on a limited Amharic text corpus or dataset?
- Which machine learning technique is appropriate in maintaining flexibility of conversation for text-oriented dataset?
- To what extent the proposed model performs?

1.4. Objective of the Study

1.4.1. General Objective

The main objective of this study is to develop Amharic text based Chatbot for driving trainee using machine learning techniques.

1.4.2. Specific Objectives

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

- To collect text data about driving skills, road safety rules and regulations so as to prepare dataset in appropriate format.
- To preprocess Amharic text using text normalization, cleaning, tokenization and stop word removal.
- To implement vector representation for Amharic text.
- To build machine learning model for intent prediction.
- To evaluate the proposed model.
- To develop prototype for the proposed chatbot system.

1.5. Scope and Limitations of the Study

The scope of this study is delimited to the development of a text-based Chatbot model specifically tailored for supporting individuals preparing for driving license tests in the Amharic language. The Chatbot's functionality is grounded on a limited Amharic text corpus, which may result in a constrained set of intents and responses. Emphasizing a text-based interface, the Chatbot is solely process written inputs and generate text-based responses, excluding voice conversation capabilities. It is imperative to underscore that the proposed Chatbot is conceived for research purposes exclusively, and as such, it is not positioned as a substitute for the instructor-led training provided by various training centers across the country. Some of the limitations of the study include:

1. Limited Amharic Text Corpus: Due to constraints in the availability of annotated Amharic text data, the Chatbot's comprehension of user intents may be circumscribed. This limitation could potentially hinder its ability to effectively address a comprehensive range of queries pertaining to driving training.

2. Text-Based Interface: The Chatbot's reliance on text-based communication may pose accessibility challenges for users who prefer voice-based interactions or possess limited literacy skills. Consequently, certain segments of the target population may be excluded from benefiting fully from the Chatbot's functionalities.

3. Research Purposes Only: The developed Chatbot is designated for research purposes and is not intended for immediate deployment in practical driving education settings. Further refinement and validation would be necessary before considering its suitability for real-world application.

4. Ethical Considerations: While efforts are made to address ethical considerations such as user privacy and data security, complete assurance of data confidentiality and security may not be attainable within the confines of this research. It is essential to acknowledge this limitation and strive to mitigate associated risks to the best extent possible.

1.6. Significance of the Study

The proposed study offers significant contributions in multiple areas, particularly in enhancing driving education, advancing conversational AI, and promoting innovation in underrepresented languages like Amharic. First, by developing a chatbot specifically tailored for Amharic-speaking individuals preparing for driving tests, the study addresses accessibility challenges that many face in traditional driving education settings. The chatbot enables users to improve their driving skills, learn traffic rules, and access critical information at any time, from any location, overcoming geographical or socioeconomic barriers. This is especially relevant in areas with limited resources for in-person driving education, making it a valuable tool for expanding educational reach.

Second, the study's focus on driving competence and awareness has the potential to contribute to reducing road traffic accidents in Ethiopia. By providing up-to-date, reliable information about road safety, driving regulations, and safe driving practices, the chatbot equips users with essential knowledge that can enhance their driving abilities and reduce accidents. Additionally, the study makes a pioneering effort in advancing conversational AI for linguistically diverse communities. By tackling the complexities of natural language processing for Amharic, the

research contributes to broader discussions on AI solutions for underserved languages. Lastly, the project provides a foundation for future research, offering methodologies and insights that can be used to develop similar chatbots for other languages and domains, promoting continued innovation in AI and education.

1.7. Ethical Considerations

The proposed research is conducted according to the ethical considerations stated by the Ministry of Science and Technology (2014). This research meets the eight criteria of this national guideline that includes; Ethical justification, Science and social value, Favorable risk-benefit ratio to research participants and their communities, Fair selection and enrollment of human subjects, Privacy, Informed consent, and Community engagement.

1.8. Organization of the Study

This document is meticulously structured into six comprehensive chapters, each serving a distinct purpose in advancing the study. The first chapter, **Introduction**, lays the foundation by providing a detailed overview of the study's background, clearly articulating the statement of the problem, and outlining both the general and specific objectives that the research aims to achieve. It also highlights the significance of the study and delineates its scope, setting the stage for the subsequent chapters.

The second chapter, **Literature Review**, delves into existing research and theoretical frameworks relevant to the study. It critically examines previous work in the field, identifies gaps in the current body of knowledge, and positions the proposed research within the broader academic context. This chapter serves as a crucial link between the study's objectives and the methodological approach employed.

In the third chapter, **Research Methodology**, the document outlines the methods and techniques utilized to conduct the research. This chapter provides a thorough explanation of the research design, data collection methods, and analysis techniques. It ensures transparency and rigor in the study, enabling replication and validation of the findings by future researchers.

The fourth chapter, **Design and Implementation**, presents a detailed account of the practical steps taken to develop and implement the proposed solution. This chapter is particularly focused

on the technical aspects of the study, including the design choices, implementation challenges, and the rationale behind the selected approaches.

The fifth chapter, **Results and Discussion**, is dedicated to analyzing the findings of the study. It presents the results in a clear and logical manner, followed by an in-depth discussion that interprets the findings in relation to the research questions and objectives. This chapter also addresses the implications of the results, offering insights into the study's contributions to the field.

Finally, the sixth chapter, **Conclusion and Future Works**, encapsulates the study by summarizing the key findings and their significance. It also reflects on the study's strengths and limitations, offering recommendations for future research and potential areas for further exploration. This final chapter provides closure to the document while also opening avenues for ongoing research in the field.

CHAPTER TWO
LITERATURE REVIEW

2.1. Overview

The literature review chapter serves as a critical foundation for this study, providing a comprehensive overview of the existing research and theoretical underpinnings relevant to the development of an Amharic text-based chatbot for driving trainees using machine learning techniques. The aim of this chapter is to contextualize the proposed research within the broader academic and practical landscape, highlighting both the advancements and the gaps that this study seeks to address. Given the rapid evolution of artificial intelligence (AI) and natural language processing (NLP), it is imperative to examine the diverse methodologies, applications, and outcomes associated with these technologies. This review delves into the historical development and current state of chatbots, with a particular focus on their use in educational and training environments.

Furthermore, it explores the specific challenges and opportunities associated with developing NLP models for underrepresented languages, such as Amharic. By synthesizing insights from previous studies and identifying areas for further exploration, this chapter aims to establish a solid conceptual framework that informs the design and implementation of the proposed chatbot. It also seeks to underscore the significance of leveraging machine learning to enhance driving training efficacy, thereby contributing to improved road safety outcomes in Ethiopia. Through a detailed examination of relevant literature, this chapter illuminates the intersection of technology, language, and education, setting the stage for the subsequent empirical investigation.

2.2. Chatbot Technology and Natural Language Processing (NLP)

Chatbot technology aims to facilitate communication between humans and computers using natural language. However, the complexity of human language has led scientists to provide models capable of understanding human language using NLP methods. NLP is the field of study that explores the computer's ability to understand human language (Haan, 2018). NLP has played a big role in chatbot in case of human-computer interaction. It is a subfield of AI, used to allow computers to process and process human language. In a chatbot application, natural language communication requires two basic skills: text generation and text comprehension. This is called natural language understanding and natural language generation. Natural language understanding

is the process of helping computers read text and understand the intent or meaning of the text just like humans. After understanding the text submitted by the user, the computer generates a response to the user by understanding the context of the user's text. This process is called natural language generation (Karlin and Alexander, 1962).

Conversational AI is one of the hottest areas of research where NLP and AI technology are applied. This is the set of technologies behind automated voice and messaging applications that provide a human-like interaction between computers and humans by recognizing intent to understand speech and text, by decoding different languages and respond in a way that mimics human conversation. A chatbot is “an artificial intelligence model that may be capable of handling conversation through auditory or text-based methods. Such shows are often designed to convincingly simulate human behavior as speakers. (Klopfenstein, Delpriori and Malatini, 2017).

Without a doubt, conversational AI or chatbots can be observed more and more often, in more and more everyday situations through various daily activities. Since we are in the age of artificial intelligence and the advancement of self-processing machines for many tasks, it is possible to implement a conversational chatbot. This is possible through machine learning algorithms with natural language processing capabilities, building models that make machines smarter to decide needs, analyze data, make recommendations. and provide automated services in different languages.

Chatbots are an artificial intelligence (AI) designed to simulate human conversation and interactive communication to handle specific tasks such as customer service, restaurant reservations, and inquiries. shopping, medical services, health consultation, etc. (Klopfenstein, Delpriori and Malatini, 2017). They can be used in various designed services and functions, such as reservations, entertainment, sales, tourism, public services and consulting, use in agriculture, sports, medical services by providing simple communication interfaces (Følstad, Skjuve and Brandtzaeg, 2019).

Hence, there are completely different conversational systems designed as mentioned below. One of the oldest and most well-known chatbots is a program called Eliza, created by the MIT Artificial Intelligence Laboratory, which operated from 1994 to 1996. The program is based on pattern matching of a certain keyword and based on these, Eliza answers an open-ended question

message response is complete. Eliza can be considered as the first conversational agent developed by Joseph Weizenbaum (Klopfenstein, Delpriori and Malatini, 2017). Conversations are monitored by the user's speech and the system tries to process some keywords and generate a keyword-based response between the system and the user's speech (Klopfenstein, Delpriori and Malatini, 2017).

2.3. Natural Language Processing in Amharic Language

Natural Language Processing (NLP) refers to AI method of communicating with an intelligent system using a natural language such as English. Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions, when you want to hear decision from a dialogue based clinical expert system, etc. The field of NLP involves making computers to perform useful tasks with the natural languages as humans use. The input and output of an NLP system can be speech or written text.

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

2.4. Machine Learning Techniques

Machine learning plays a significant part in the development of prediction models. It is defined as a branch of computer science that develops algorithms and approaches for handling automated

problems that are difficult to solve using traditional methods. Machine learning is a subset of artificial intelligence, which is a broader field (AI). One of AI's goals is to learn, make conclusions, and adapt to a changing environment, and its scope aids this purpose. Other definitions of machine learning characterize it as a set of tools that use algorithms to identify hidden patterns in data, giving them a previously unidentified meaning. The idea of machine learning assumes that natural, human learning methods should not be taken as the only possible cognitive pathways and its main goal is to explore alternative learning mechanisms. The basis of machine learning techniques is to supply a selected model with a set of training data, on the basis of which its parameters are adjusted by the selected algorithms. The process of adjusting model parameters is called 'Training' whereby the model is expected to respond to future input data based on the patterns detected in the dataset used to create it. The approach to the learning process determines the division of machine learning techniques into two main categories: supervised learning and unsupervised learning (Wach & Chomiak-Orsa, 2021).

2.4.1. Supervised machine learning

In a simple machine learning model, the learning process is divided into two steps: training and testing. In the training phase, samples from the training data are used as input, and the learning algorithm or learner learns the features and builds the learning model. The execution engine is used by the learning model to produce predictions for test or production data during the testing phase. The final prediction or categorized data is tagged data, which is the output of the learning model (Nasteski, 2017).

In supervised machine learning, labeled data has to be provided the algorithm with a dataset. This could be, for example, a set of one million criminal cases and their decisions. The decisions, in this example, would be the target that the algorithm attempts to predict. The more data, the better the algorithm can become. The algorithm can be used to achieve a variety of objectives: The majority of the time, its regression or classification. The method in regression takes the input and tries to guess a numerical value. In my example, it may try to predict criminal case outcomes based on a variety of variables. The closer the algorithm's evidence is to actual criminal case decisions, the better. Classification tries to put the example into a class. Features at this point, the computer has no idea how to deal with this information. It must be transformed into a set of attributes or a numerical representation of the data. Feature engineering is the term for this. It is a

difficult task that necessitates a significant amount of time and experience in the field. The evidence and testimonial, the time the crime occurs, the location, and the relationship with the victim are all relevant variables in my earlier example of predicting the decisions of a crime. Evaluation there has to be a way to evaluate the algorithm. This is typically used by the computer internally to determine how the algorithm it currently runs is performing; training once the computer has a better understanding of how it is currently performing, it makes minor adjustments to the algorithm to improve performance on the next attempt. Training is the term for this procedure. After training, the engineer frequently returns to modify the features or algorithm used to improve the model's performance.

Based on sample input-output pairs, supervised learning uses ML tasks to train a function that translates an input to an output. As a result, this learning process is based on comparing calculated and projected outputs, i.e., learning entails computing the error and modifying the error in order to get the desired output. Nave Bayes classification, linear and logistic regression, and Support Vector Machines are examples of such techniques (SVMs). Face recognition is useful as a security measure at an ATM, surveillance areas, closed circuit cameras, the criminal justice system, and image tagging in social networking sites like Facebook are examples of applied supervised learning. Another well-known example of supervised learning is epileptic seizure detection, which can be used to create patient-specific detectors capable of immediately identifying episode occurrences and avoiding the danger of bodily injury and death(Pugliese et al., 2021).

In this approach, supervised learning, a training set with acceptable objectives is provided. In supervised learning, there are two categories: classification and regression. The trained system assigns inputs into classes using classification methods in classification. The sources in regression are continuous rather than discrete. Regression predictions are evaluated using the root-mean-squared error, while classification predictions are evaluated using accuracy. The purpose of supervised learning is to predict a known output from a shared dataset. Most tasks that are accomplished by supervised learning can also be completed by a skilled person. Supervised learning focuses on classification, which is selecting the best subgroups that best describe a new instance of data, and prediction, which entails predicting an unknown parameter.

This is frequently used to evaluate and model risk while uncovering linkages that people cannot see (Jayatilake & Ganegoda, 2021).

2.4.2. Unsupervised machine learning

Unsupervised learning algorithms are employed when the training data lacks classification or labeling. These algorithms focus on exploring the underlying patterns and structures within the data without predefined labels or categories. Unlike supervised learning, where the goal is to predict an output based on input-output pairs, unsupervised learning aims to infer the hidden structure from the unlabeled data. This involves identifying inherent groupings, associations, or relationships in the data, enabling the system to draw meaningful inferences about the dataset's structure and characteristics (Kuiper et al., 2019).

Unsupervised learning encompasses a variety of techniques, each suited to different types of analysis. Clustering algorithms, such as k-means and hierarchical clustering, group similar data points together, revealing natural clusters within the dataset. Dimensionality reduction techniques, like principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE), reduce the number of variables under consideration, making the data easier to visualize and interpret while preserving its essential structure. Association rule learning, another unsupervised method, identifies interesting relationships and associations between variables in large datasets. These algorithms are particularly useful in exploratory data analysis, anomaly detection, and pattern recognition, providing valuable insights and revealing hidden patterns that would be difficult to discern through manual analysis alone.

2.4.3. Reinforcement machine learning

Reinforcement learning (RL) algorithms are a unique type of machine learning that operates through interaction with an environment, generating actions and receiving feedback in the form of rewards or penalties. Central to reinforcement learning is the concept of trial-and-error search and the optimization of delayed rewards. Unlike other learning paradigms, RL focuses on enabling machines and software agents to automatically determine the best behavior in a given situation to improve their performance and efficiency (N. Xu, 2019). This involves an agent learning to make decisions by taking actions in an environment and observing the consequences, with the ultimate goal of maximizing cumulative rewards over time.

Reinforcement learning is particularly well-suited for complex, dynamic environments where optimal behavior is not predefined but must be discovered through exploration and exploitation. This approach is driven by the reward-punishment mechanism, where agents learn to choose actions that maximize rewards and minimize penalties. RL has shown great potential in automating and optimizing sophisticated systems, such as robotics, autonomous driving, manufacturing, and supply chain logistics. By continually learning from interactions with their environment, RL agents can adapt to changing conditions and improve operational efficiency. However, it is important to note that RL is not suitable for solving simple or straightforward problems due to its inherent complexity and the need for significant computational resources (Sarker, 2021). Its power lies in its ability to handle intricate decision-making processes and optimize performance in challenging scenarios.

2.5. Supervised Machine Learning Algorithms

Supervised learning algorithms, a fundamental category within the machine learning domain, operate by learning patterns from labeled training data to make predictions or decisions without human intervention. These algorithms encompass a wide array of techniques, each originating from diverse research fields such as statistics, computer science, and neural networks. According to Bellazzi and Zupan (2008), these algorithms vary significantly in their modeling approaches and can be evaluated based on several criteria. These criteria include their ability to handle missing data and noise, their treatment of different types of attributes, the transparency of the resulting classification models, the computational cost associated with training, their decision-making explainability, and their generalization capability—i.e., how well they perform on unseen data.

Predictive machine learning algorithms come in various forms, each with its own strengths and weaknesses tailored to specific types of problems. For instance, decision trees are appreciated for their interpretability, allowing domain experts to understand and examine the decision-making process, whereas support vector machines (SVM) are known for their robustness in high-dimensional spaces. Neural networks, particularly deep learning models, have shown remarkable performance in handling complex data but often come with higher computational costs and less transparency. Additionally, ensemble methods like random forests and gradient boosting combine multiple models to improve prediction accuracy and generalization capabilities. The

choice of algorithm depends on the specific requirements of the application, such as the need for interpretability versus predictive power, the nature of the data, and the computational resources available. Subsequent sections delves deeper into these various predictive machine learning algorithms, discussing their unique characteristics and applications.

2.5.1. Naïve Bayesian classifier

The Naïve Bayes classifier is one of the simplest yet highly effective machine learning algorithms. It operates on the principle of conditional probability, assuming that all variables in the dataset are "naïve," meaning they are considered independent and not correlated with each other. Despite this strong and often unrealistic assumption, the Naïve Bayes classifier often performs comparably to more sophisticated algorithms. Its simplicity and speed in training make it a popular choice as a baseline algorithm in comparative studies. When other, more complex algorithms outperform Naïve Bayes, it often indicates the presence of non-linear interactions between the attributes in the dataset (Bellazzi & Zupan, 2008).

One of the primary advantages of the Naïve Bayes classifier is its efficiency. It is quick to predict the class of the test dataset and handles multiclass prediction problems effectively. When the independence assumption holds true, Naïve Bayes can outperform models like logistic regression and requires less training data. Additionally, it performs well with categorical input variables. However, for numerical variables, the algorithm assumes a normal distribution (bell curve), which can be a limiting factor. Despite these constraints, its ease of implementation and speed make it a valuable tool in many practical applications.

However, the Naïve Bayes classifier has several notable limitations. If a categorical variable in the test dataset was not observed during training, the model assigns a zero probability, making it unable to make a prediction, a problem known as zero frequency. This can be mitigated using smoothing techniques. Additionally, Naïve Bayes is known to be a poor estimator, meaning the probability outputs should not be taken too seriously. The most significant limitation is the assumption of independence among predictors, which is rarely true in real-world scenarios. This assumption can lead to less accurate predictions when the features are correlated, underscoring the importance of understanding the context and characteristics of the data before applying Naïve Bayes.

2.5.2. Artificial Neural Networks

Artificial Neural Networks (ANNs) are a class of machine learning models inspired by the structure and function of the human brain. While there is no universally accepted definition of ANNs, they are generally described as networks composed of many simple processors, or "units," each potentially with a small amount of local memory. These units are interconnected by communication channels, or "connections," that usually carry numeric data. Each unit operates on its local data and inputs received through these connections, though this restriction may be relaxed during the training phase to enhance learning (Bishop, 1995).

The learning process in ANNs involves adjusting the weights of the connections based on the data provided. This is often referred to as the "training" rule. ANNs learn from examples, similar to how children learn to distinguish between different objects by observing examples. When trained effectively, ANNs can generalize beyond the training data, producing approximately correct results for new, unseen cases. One of the key strengths of ANNs is their potential for parallelism, as the computations of individual units are largely independent of each other. This characteristic allows for efficient processing of large datasets and complex computational tasks (Bishop, 1995).

However, ANNs come with several challenges and limitations. They are highly sensitive to the parameters of the method, including those that determine the network's architecture, which can significantly affect performance. The training process itself is computationally intensive, often requiring substantial resources and time. Furthermore, ANNs are often criticized for their "black-box" nature; the models they induce can be difficult to interpret, making it challenging for domain experts to extract and understand the underlying knowledge represented by the network's weights. Despite these drawbacks, ANNs remain a powerful, self-adaptive, and flexible computational tool capable of detecting complex, nonlinear relationships in data and handling large-scale datasets efficiently.

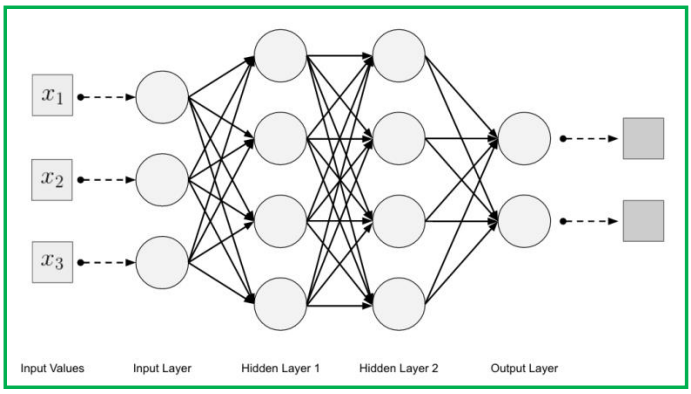


Figure 1: Typical Artificial neural network architecture

2.5.3. The k-nearest neighbours

The k-nearest neighbors (KNN) algorithm is inspired by the intuitive approach that domain experts often use to make decisions: evaluating new cases based on previously encountered similar instances. For a given data instance, the KNN classifier identifies the k most similar training instances and classifies the new instance based on the prevailing class among these neighbors. While this method mirrors human decision-making processes and can be very effective, it can also be computationally intensive. The search for the most similar instances requires accessing the entire training set during classification, which can slow down the process, particularly with large datasets (Bellazzi & Zupan, 2008).

One of the primary advantages of the KNN algorithm is that it has a zero-cost learning process. Since there is no explicit training phase, the model makes no prior assumptions about the data's characteristics, which allows it to learn complex concepts through local approximation with simple procedures. This makes KNN highly versatile and straightforward to implement. Additionally, KNN is robust in handling noisy training data and can be easily adapted for various types of problems, from classification to regression.

However, the KNN algorithm also has notable disadvantages. The lack of an explicit model means that the learned concepts cannot be easily interpreted or described, which can be a significant limitation in domains requiring model transparency. The algorithm's computational cost is another major drawback, as finding the k nearest neighbors becomes increasingly expensive with larger datasets. Moreover, KNN's performance can degrade with high-dimensional data, a phenomenon known as the "curse of dimensionality." As the number of

dimensions increases, the distance metrics used to find nearest neighbors become less reliable, impacting the algorithm's accuracy and efficiency. Despite these challenges, KNN remains a popular and effective method in many modern data mining applications due to its simplicity and effectiveness in various contexts.

2.5.4. Decision trees

Decision trees are a popular machine learning technique that utilizes recursive data partitioning to create models that are easy to understand and interpret. These models consist of a series of binary decisions leading to outcomes, structured in a tree-like format. This transparency allows for clear visualization and straightforward interpretation, making decision trees particularly useful in applications where model interpretability is crucial. However, one of the potential drawbacks is that the leaves of decision trees, which represent the final decisions, can sometimes contain too few instances, leading to unreliable predictions due to over-segmentation of the data (Bellazzi & Zupan, 2008).

The computational complexity of decision tree induction algorithms is relatively low, thanks to the use of powerful heuristics. This efficiency is one reason why decision trees are widely used in various data mining applications. Some of the most well-known decision tree algorithms include C4.5, its successor See5, and the CART (Classification and Regression Trees) algorithm. These algorithms are capable of handling both categorical and quantitative values, can manage missing data by filling in with the most probable values, and can solve both classification and regression problems. They are also suitable for handling multi-output problems, adding to their versatility and applicability in diverse scenarios.

Despite their advantages, decision trees have some notable disadvantages. A small change in the dataset can result in a significantly different tree, which can make them unstable. The heuristic nature of decision tree algorithms means they do not guarantee finding the globally optimal tree, which can limit their effectiveness in some situations. Furthermore, for categorical variables with many levels, decision trees can show a bias towards attributes with more levels, impacting the accuracy and reliability of the model. Despite these limitations, decision trees remain a valuable tool in the machine learning arsenal, particularly for their interpretability and ease of use.

2.5.5. Decision rules

Decision rules are a machine learning technique that expresses knowledge in the form of simple, interpretable rules. These rules are typically structured as "IF condition-based-on-attribute-values THEN outcome-value," making them highly intuitive and easy to understand. Decision rules can be constructed from induced decision trees, such as those generated by the C4.5rules algorithm, or derived directly from data using algorithms like AQ and CN2. While decision rules share many characteristics with decision trees, including their performance, they can sometimes be computationally more expensive due to the complexity of generating and managing numerous rules (Bellazzi & Zupan, 2008).

One of the key advantages of decision rules is their ease of transferability and understanding. They provide clear, actionable insights that can be easily communicated and applied to different types of problems. This practicality makes them a valuable tool for various applications, from medical diagnosis to financial forecasting. Additionally, decision rules are versatile and can solve both classification and regression problems effectively. Their straightforward nature also means that they can be more easily implemented and adapted compared to more complex models.

However, decision rules also have some significant disadvantages. One of the main challenges is scaling up to large datasets, as the rule induction process can become computationally intensive and less efficient with increasing data size. Additionally, when learning from noisy or corrupted data, decision rules can become overly complicated, producing a large number of trivial or irrelevant rules. This can lead to models that are difficult to manage and interpret, negating some of the advantages of their simplicity. Despite these challenges, decision rules remain a powerful and practical approach in many machine learning applications, particularly when clarity and interpretability are essential.

2.6. Related Works

Several research works have been conducted to develop Chatbot system to automate human conversation by employing various methods and techniques of artificial intelligence. Many researchers have done many things on the chatbot. In this section, we try to review different papers related to chatbot development.

Srivastava and Singh (2020) presented an automatized medical chatbot, Medibot, designed to assist in disease prediction by taking user inputs of symptoms. The methodology employed Artificial Intelligence Markup Language (AIML) for managing symptom inputs and utilized K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) for disease prediction. While this approach demonstrates the integration of machine learning techniques in healthcare chatbots, a significant limitation is that user inputs must match the predefined patterns in AIML, limiting the chatbot's flexibility in understanding varied user queries. This constraint highlights a gap in the adaptability of the chatbot, which may reduce its effectiveness in real-world scenarios where user input can vary widely.

Athota et al. (2020) developed a healthcare chatbot using artificial intelligence, focusing on finding text similarity between a user's query and a pre-existing dataset. The methodology incorporated Cosine similarity and TF/IDF (Term Frequency-Inverse Document Frequency) to match the user input with the relevant data. However, the implementation faced challenges related to high time complexity, making it less efficient for real-time applications. The time complexity issue poses a significant limitation, particularly in contexts where quick response times are crucial for user satisfaction and effective interaction.

Rahman et al. (2019) introduced a Bangla healthcare chatbot named Disha, which uses a machine learning-based approach for disease prediction. The chatbot applies Cosine similarity and TF/IDF to identify symptoms from user inputs and employs SVM for disease prediction. While the chatbot can predict diseases with some accuracy, it faces limitations when the user inputs three or more symptoms. This challenge in handling multiple symptoms simultaneously suggests a need for more robust techniques to improve accuracy and provide more reliable predictions in complex scenarios.

Gonda et al. (2019) explored the use of conversational bots as a supplement for teaching assistant training courses, utilizing the Dialogue Flow framework combined with rule-based techniques. The chatbot was designed to understand and respond to queries related to the rules predefined in the system. However, the chatbot's ability to understand and respond effectively was limited to queries that aligned with the established rules. This reliance on rule-based systems suggests a

gap in the chatbot's ability to handle more dynamic or unexpected queries, which could limit its usefulness in more varied educational settings.

Suhel et al. (2020) focused on automating banking conversations using a chatbot built with Artificial Machine Intelligence Language (AIML). While the chatbot was effective in automating banking interactions, it required that user inputs exactly match the patterns predefined in AIML. This limitation reduces the chatbot's flexibility and adaptability, making it less effective in handling diverse user queries or variations in input. The dependency on predefined patterns suggests a need for more sophisticated natural language processing techniques to enhance user interaction.

Habtamu (2020) addressed the challenge of designing an adaptive bot model for consulting Ethiopian published laws using an ensemble architecture integrated with rules. The methodology involved using Recurrent Neural Networks (RNNs) and focused on handling model overfitting by adjusting the data splitting ratio. Despite this innovative approach, the study's primary limitation was related to the overfitting problem, which could hinder the model's generalizability and performance in different contexts. Addressing overfitting through improved data management or more advanced techniques could enhance the model's applicability.

Segni (2021) developed an Afaan Oromoo text-based chatbot aimed at enhancing maize productivity, using Deep Neural Networks (DNN) and Recurrent Neural Networks (RNN). Although the chatbot was innovative in applying machine learning to agricultural productivity, it was limited by the use of only 25 intents and lacked techniques to handle model overfitting. This constraint suggests that the chatbot may struggle with generalization, particularly when faced with a broader range of user inputs or intents. The study points to the need for more extensive intent coverage and overfitting prevention strategies to improve the chatbot's performance.

Fekadu (2021) focused on designing and developing a bilingual chatbot for assisting Ethio-Telecom customers with customer service inquiries. The chatbot utilized Deep Neural Networks (DNN) and Bi-directional Long Short-Term Memory (Bi-LSTM) networks to process user queries. However, a significant gap in this study was the lack of hyperparameter tuning, which could potentially limit the model's performance. Proper tuning of hyperparameters is crucial for

optimizing the model's accuracy and efficiency, suggesting that further refinement is needed to fully realize the chatbot's potential in customer service applications.

Table 1: Summary of the related works

Author (year)	Title	Methodology	Gap
Srivastava and Singh (2020)	Automatized medical chatbot (medibot)	• AIML for taking symptoms • KNN and SVM for disease prediction	Users inputs should be the same as with the pattern prepared by AIML
Athota *et al.* (2020)	Chatbot for healthcare system using artificial intelligence	• Cosine similarity and TF/IDF to find text similarity between dataset and user query	Takes high time complexity
Rahman et al. (2019)	Disha: An implementation of machine learning based Bangla healthcare Chatbot	• Cosine similarity and TF/IDF to identify the symptoms • SVM to predict disease	To predict the disease accurately if the user insert three or more symptoms
Gonda et al. (2019)	Creating conversational bots as a supplement for teaching assistant training course	• Dialogue flow framework with rule-based technique	The bot understands the query if the query is related with the rule
Suhel et al. (2020)	Conversation to automation in banking through Chatbot using artificial machine intelligence language	AIML	The user's inputs must be the same with the pattern prepared by AIML
Habtamu (2020)	Designing and Implementing Adaptive Bot Model to Consult Ethiopian Published Laws Using Ensemble Architecture with Rules Integrated	RNN	Handling model overfitting problem by changing the number of data splitting ratio
Segni, B. (2021) [5]	Developing Afaan Oromoo Text based Chatbot for Enhancing Maize Productivity	• DNN • RNN	Used only 25 intents and not model overfitting handling techniques
Fekadu, E. (2021)	Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services	• DNN • Bi-LSTM	Hyper parameters are not tuned

2.7. Research Gaps

The proposed study is highly appropriate due to the critical research gaps it addresses, particularly in the context of language-specific challenges and domain-specific knowledge requirements in Amharic driving education. While the availability of driving-related datasets and basic natural language processing (NLP) techniques is established, there remains a significant gap in developing effective tools tailored for the Amharic language, which possesses unique linguistic features such as its rich morphology and script variations. Existing NLP tools, which are predominantly designed for widely spoken languages like English, are not readily adaptable to Amharic, especially in specialized domains like driving education. This makes the proposed study crucial, as it focuses on refining and applying NLP methods, such as text normalization, tokenization, and stop word removal, specifically for Amharic, ensuring that the language's intricacies are well-handled in the model. Addressing this gap ensures that Amharic speakers have access to learning tools that cater to their language's particular needs, fostering more effective and accessible driving education in Ethiopia.

Moreover, the study is timely and necessary as it aims to create a comprehensive chatbot model that fills the gap in providing accurate and contextually relevant responses to driving-related queries. While a general Amharic text corpus exists, there is a lack of specialized tools and applications designed to address domain-specific content, such as driving regulations, road safety practices, and driving test preparation. The existing resources are often static and do not offer interactive or personalized support, leaving learners with limited access to real-time assistance. The study proposes to bridge this gap by developing a chatbot that not only processes Amharic text effectively but also leverages tailored content to meet the specific learning needs of driving trainees. This focus on a practical, real-world application in a critical domain like road safety further underscores the study's appropriateness, as it seeks to mitigate the high rates of traffic accidents in Ethiopia by improving the quality of driving education through advanced technological solutions.

CHAPTER THREE

RESEARCH METHODOLOGY

The research methodology chapter is a crucial component of this study, outlining the systematic approach adopted to develop and evaluate an Amharic text-based chatbot designed to assist driving trainees using machine learning techniques. This chapter details the comprehensive plan and specific methods employed to achieve the research objectives, ensuring the study's reliability and validity. The development of a conversational AI model tailored for Amharic language users necessitates a meticulous selection of methodologies that address both technical and linguistic challenges. This chapter begins by delineating the overall research process, to provide a robust framework for chatbot development and assessment. It then elaborates on the data collection processes, including the acquisition and preprocessing of Amharic text corpora essential for training the chatbot.

Furthermore, this chapter describes the machine learning techniques employed, emphasizing the selection criteria for algorithms and the rationale behind choosing supervised learning methods over other approaches. The subsequent sections delve into the design and implementation phases, highlighting the iterative processes of model training, testing, and refinement. Evaluation metrics and validation strategies are also discussed in detail, ensuring that the chatbot's performance is rigorously assessed against predefined criteria. Finally, this chapter addresses potential ethical considerations and limitations of the study, acknowledging the constraints and providing a balanced perspective on the expected outcomes. By presenting a clear and detailed roadmap, the research methodology chapter aims to convey the rigor and coherence of the research process, paving the way for the development of an effective and reliable Amharic text-based chatbot for driving training.

3.1. Research Process

The research process undertaken in this study represents a systematic and iterative journey aimed at developing a text-based Chatbot model tailored specifically for driving trainees in the Amharic-speaking community. Grounded in the principles of machine learning and natural language processing, this process encompasses a series of interconnected steps, each designed to contribute to the overarching objective of enhancing driving education accessibility and

effectiveness. From data collection and preprocessing to model development and evaluation, each phase of the research process is meticulously orchestrated to ensure the development of a robust and user-centric Chatbot solution. By following this structured approach, the research endeavors to address the multifaceted challenges associated with driving education in linguistically diverse contexts, laying the groundwork for meaningful advancements in both technological innovation and educational empowerment. Throughout this journey, collaboration, innovation, and a commitment to excellence serve as guiding principles, driving the research forward towards its ultimate goal of fostering a culture of responsible driving behavior and road safety awareness in Ethiopia and beyond.

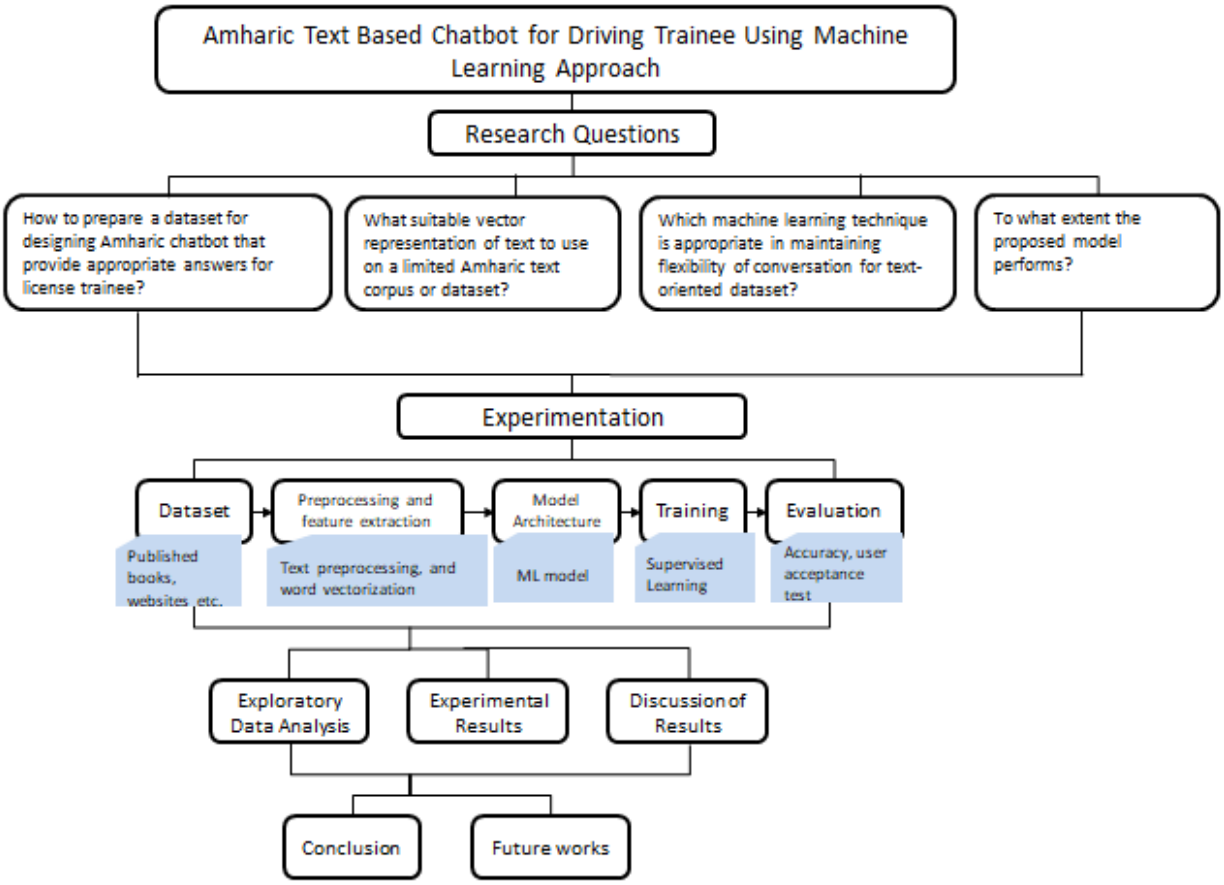


Figure 2: Research process of the study

3.2. Data Collection

The data collection process for this study encompasses a diverse range of sources, each carefully selected to provide comprehensive insights into Amharic driving training materials and learner

requirements. The primary data source comprises various published books such as “የመንጃ ፈቃድ መማሪያ” and “የመንጃ ፈቃድ 1000 ጥያቄዎችና መልሶቻቸው,” which serve as foundational texts for driving trainees in Ethiopia. These books offer a wealth of knowledge on driving rules, regulations, road signs, and safety practices, providing valuable content for the development of the chatbot's knowledge base. In addition to published materials, various websites and applications are explored as secondary data sources. These platforms host frequently asked questions, quizzes, and interactive tutorials related to driving lessons, making them invaluable resources for understanding common learner queries and misconceptions. By analyzing the content available on these platforms, the research can ensure that the chatbot addresses the specific needs and concerns of Amharic-speaking driving trainees. Furthermore, informal interviews and surveys with driving instructors and learners are conducted as supplementary data collection methods, allowing for direct input and feedback on the chatbot's design and functionality. This multi-faceted approach to data collection ensures a comprehensive understanding of the driving training landscape in Ethiopia, facilitating the development of a chatbot that is informative, user-friendly, and tailored to the needs of its target audience.

3.3. Dataset Preparation

The dataset preparation process involves structuring the data in a format suitable for training the chatbot model. In this study, the dataset is organized using JSON (JavaScript Object Notation), a lightweight data-interchange format that is both human-readable and machine-parseable. The dataset consists of three main components: tags, patterns, and responses. Tags represent the intents or ideas behind user queries, serving as the target classes during model training. These tags categorize the queries into distinct groups, enabling the chatbot to understand and respond appropriately to different types of user input. The patterns component comprises the set of queries or frequently asked questions related to each intent. These patterns serve as the training data for the chatbot, allowing it to learn the associations between user queries and their corresponding intents. Finally, the responses component contains the set of possible responses associated with each intent. These responses provide the chatbot with the knowledge and language it needs to effectively communicate with users and provide relevant information or assistance. By structuring the dataset in this JSON format, the research ensures that the data is

organized, accessible, and easily adaptable for training the chatbot model. The following shows sample JSON file.

```json
{
  "tag": "የትራፊክ ህጎች እና ደንቦች",
  "patterns": [
    "የትራፊክ ህጎች እና ደንቦች",
    "የትራፊክ ህጎች እና ደንቦች ምንድን ናቸው?",
    "ስለትራንስፖርት ሕግ እና ደንብ",
    "የሚያስጠነቅቁ የመንገድ ዳር ምልክቶች",
    "የሚቆጣጠሩ ዓለም አቀፍ የመንገድ ዳር ምልክቶች",
    "ቅድሚያ የሚያሰጡ",
    "የሚያስገድዱ የመንገድ ዳር ምልክቶች",
    "መረጃ ሰጪ የመንገድ ላይ ምልክቶች"
  ],
  "responses": [
    "የመንገድ ዳር ምልክቶች ስለትራንስፖርት ሕግ እና ደንብ፣ አደገኛ ስለሆኑ ሥፍራዎች፣ አገልግሎት መስጫ ተቋሟት በም
```

Figure 3: Sample JSON file

Utilizing the JSON dataset provided herein, a model is trained accordingly. Subsequently, the model undertakes predictive analysis to determine the appropriate tag and subsequently selects a response randomly from the dataset. The selected response is then presented as the model's output. Moreover, a substantial collection of utterance pairs has been meticulously curated to facilitate the training, validation, and testing phases of the proposed model.

3.4. Text Preprocessing

One of the most frequent jobs in numerous machine learning applications is data preprocessing. Among the different natural language processing (NLP) jobs, it is a crucial stage. It simplifies language so that machine learning algorithms can operate more effectively. The development of an Amharic text-based chatbot to aid in driver's license preparation uses the preprocessing techniques listed below.

3.4.1. Text Normalization

Text normalization is a crucial process encompassing a series of tasks aimed at standardizing text to ensure uniformity and consistency across various linguistic datasets. This is particularly important for processing Amharic text, given the specific characteristics of its writing system. In Amharic, normalization involves substituting characters with their phonetically equivalent counterparts to achieve consistency in representation. For instance, the characters "አ" and "ዐ" both produce the same sound. Consequently, in words such as "አመት" and "ዐመት", normalization necessitates replacing "ዐ" with "አ" to maintain a consistent representation of the phoneme. This step is essential for computational systems to recognize and process words with identical sounds but different spellings as semantically equivalent. By standardizing these characters, the text normalization process facilitates improved accuracy in natural language processing tasks, enabling the computer to correctly interpret and process words like "ዐመት" and "አመት" as having the same meaning.

3.4.2. Text Cleaning

Another essential preprocessing task in natural language processing is text cleaning, which involves removing various symbols and numbers from the text to enhance the computer's ability to comprehend natural language. The Amharic language has a unique numbering system that includes symbols such as [፩፡፳፡፴፡፶፡፷፡፸፡፹፡፺], in addition to the Arabic numerals [0-9] that are also used in English. During the text cleaning process, it is necessary to identify and remove these numerical symbols to prevent them from interfering with text analysis and processing. Additionally, users often incorporate various emoji symbols like [🤚🖐👍👎🤦☺✌👋] in text-based communication to convey their emotions or ideas. For example, a user might write [ሠላም👋👋👋!!] to express a greeting with emphasis. These emoji symbols must be removed to maintain textual clarity and consistency.

Furthermore, punctuation marks play a significant role in the structure of Amharic text, with common symbols including the word separator [፡], full stop (period) [።], comma [፣], semicolon [፤], and colon [፥], as well as question marks and exclamation marks. To achieve a clean and uniform text corpus, these punctuation marks must also be removed. Through the meticulous elimination of these extraneous elements, text cleaning ensures that the textual data is in its most simplified and analyzable form, thereby facilitating more effective natural language processing.

3.4.3. Tokenization

Tokenization is a fundamental step in natural language processing that involves dividing longer strings of text into smaller, more manageable pieces known as tokens. This process often starts by breaking down large text segments into sentences, and subsequently, sentences into words. Tokenization facilitates further text processing, allowing for more detailed analysis and manipulation. In this study, tokenization is applied to Amharic text using spaces or word separators as delimiters. For instance, consider the Amharic sentence "አደባባይ ላይ ቅድሚያ የሚሰጠው ተሽከርካሪ የቱ ነው?" which translates to "Which vehicle has priority on the square?" in English. To tokenize this sentence, each word and punctuation mark needs to be treated as a separate token. This ensures that the text is broken down into its constituent parts, allowing for more precise computational analysis and processing. Tokenization is a crucial step as it lays the groundwork for subsequent stages in text analysis, such as parsing, stemming, and lemmatization, by providing a structured representation of the text data. Proper tokenization is particularly important for Amharic, given its unique script and linguistic characteristics, to ensure that the text is accurately segmented and prepared for deeper linguistic processing.

3.4.4. Stop Word Removing

In natural language processing, it is a common practice to remove non-content bearing terms, known as stop words, which do not contribute to identifying the specific intent of utterances. Stop words, such as prepositions and conjunctions, are generally not useful for discerning the context or true meaning of a sentence. Their removal helps in reducing the size of the dataset, thereby enhancing the performance of the model. This study compiles a list of Amharic stop words based on previous research conducted by Himanot (2014), Hassen, Tessema, and Solomon (2009), and Tewodros (2003). Some of the Amharic stop words identified in these studies include “ስለ” (about), “ነው” (is), “ብቻ” (only), “ሌላ” (other), “ለዚያው” (for that), “ከሆነው” (if it is), and “ለዚሁ” (therefore). Removing these stop words is crucial as it reduces the vocabulary size of our model, allowing it to focus on the more significant words that convey the core meaning of the sentences. This step not only simplifies the text but also improves the efficiency and effectiveness of the subsequent text analysis processes, leading to more accurate and relevant outcomes in understanding and interpreting the Amharic language.

3.5. Feature Extraction

After completing the text preprocessing activities, including text normalization, text cleaning, tokenization, and stop word removal, the subsequent phase involves feature extraction. Machine learning algorithms are unable to process raw textual data directly; thus, feature extraction is employed to convert the text into a matrix (vector) of features. This transformation is crucial as it prepares the raw data into a format suitable for input into a machine learning system. A thorough review of the literature indicates that there are several established methods for extracting features from text data. Among these, the most widely used word embedding techniques in natural language processing are bag-of-words (BoW), term frequency-inverse document frequency (TF/IDF), and Word2Vec.

In this study, the bag-of-words (BoW) technique utilized to translate text data into corresponding numeric values. The BoW approach is a text modeling technique in natural language processing that extracts features by representing a text as a vector of integers. Each integer corresponds to the frequency or presence of a word within the text. The choice of the BoW method for this study is due to its versatility and efficacy in extracting features from textual data. By converting text into a numerical representation, BoW facilitates the application of various machine learning algorithms, thereby enhancing their ability to analyze and interpret the text effectively. This method's simplicity and effectiveness make it particularly suitable for handling large volumes of text data, ensuring that the essential features are captured and utilized for further processing in the machine learning pipeline. It was chosen due to its simplicity and efficiency in text representation, particularly for a chatbot that focuses on short, structured queries like those related to driving education. BoW treats text as a collection of words, allowing straightforward feature extraction based on word occurrence. While more advanced techniques like Word2Vec or TF-IDF capture deeper semantic relationships, BoW is computationally less intensive and works well for the task at hand, where understanding specific keywords (e.g., traffic rules or regulations) is more important than grasping the semantic nuances of long sentences.

3.6. Model Building and Selection

Model building and selection involve a series of rigorous steps, grounded in both theoretical understanding and empirical experimentation, to determine the most effective model for solving a given problem. This process is iterative and demands numerous trials and refinements of design concepts before achieving a satisfactory outcome. The ultimate goal is to enhance our understanding of which solutions are most effective and applicable to the problem at hand. In this specific research, experimental method is employed to evaluate the efficacy of different design ideas, guiding the development of the classification model.

For this study, the model is constructed and selected based on a carefully prepared dataset designed to yield accurate responses to user requests. As previously detailed in the dataset preparation section, the dataset is organized using a JSON file format, categorized by intents—the contextual meanings behind user queries. These intents are served as classes for prediction during model training, resulting in multiple intents or target classes within the dataset. Consequently, the model selection process focuses on multiclass classification.

To determine the most effective model for intent classification in the proposed chatbot, various machine learning algorithms commonly used in supervised learning tasks are evaluated. These algorithms include neural networks, decision trees, logistic regression, and support vector machines, among others. Each of these techniques are assessed for their ability to accurately classify user intents and respond appropriately, ensuring that the final model provides reliable and contextually relevant answers to user inquiries. This comprehensive approach aims to identify the best-performing model through a combination of theoretical insights and experimental validation, ultimately leading to the development of a robust and efficient chatbot.

3.7. Performance Evaluation

Once the model is constructed, it is imperative to evaluate its performance to determine if the predefined objectives have been met. This evaluation is conducted using the dataset and is primarily assessed through an evaluation metric known as accuracy. Accuracy represents the proportion of correctly predicted instances out of the total instances in the dataset, whether from the training or testing phases. This metric is crucial as it quantifies the model's effectiveness in correctly classifying the given data into their respective categories. Accuracy is computed using

a specific formula, which provides a clear measure of the model's predictive capability. By analyzing accuracy, we can gauge how well the model performs in real-world scenarios and make any necessary adjustments to improve its performance. This step is essential for validating the model's reliability and ensuring it meets the expected standards for practical application.

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+TN+FN+Fp}}$$

where,

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

FP: False Positive: Negative values predicted as positive

FN: False Negative: Positive values predicted as negative

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

3.8. User Acceptance Testing

Following the accuracy evaluation of the proposed prototype, user acceptance testing is conducted to assess the system's practical utility. Target users for this evaluation are selected through a purposive sampling technique to ensure a representative assessment of the system's usefulness. Additionally, domain experts are included in the evaluation process to provide an informed perspective on the prototype's performance.

The criteria for user acceptance testing are adapted from the framework established by Prat et al. (2014). These criteria are tailored to the specific context of this study to provide a comprehensive evaluation. The modified criteria focus on several key aspects: ease of use, time efficiency, operational efficiency, accuracy, completeness, utility, and the system's learning capability to support target users. This holistic approach ensures that the prototype system is not only accurate but also practical and beneficial for end-users, thereby validating its overall effectiveness and readiness for real-world application.

3.9. Tools

In this study, the following various libraries and tools are used.

Anaconda Navigator

It is a command-line command, users can navigate, manage conda packages, and manage environments using the desktop program Anaconda Navigator. It is used to start the various Python development environments required for creating the proposed Chatbot.

Spyder IDE

Spyder is a free open-source scientific python development environment that is used to write a python program designed by and for scientists, engineers, and data analysts. This IDE is used to write and run python code for data processing and creating the proposed Chatbot system.

nltk

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

scikit-learn

scikit-learn is a python library used for applying machine learning algorithms. It contains different algorithms that are used for classification, regression, clustering dimensional reduction. We used this library in the proposed study for splitting the dataset into train data and test data.

Tensorflow

Tensorflow is an open-source library developed by Google, and it is used in machine learning application development for classification and prediction. It is used in the proposed Chatbot development for applying machine learning algorithms for intent classification.

numpy

Numpy is an open-source numerical library that is used for the visualization of array and matrix data. It is used to convert the data into the vector and reshaping the vector to fit the vector with the proposed model.

json

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

matplotlib

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

tkinter

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard GNU/Linux, Miscrosoft Windows and macOS installs of Python. It is used for creating GUI chat interface for the proposed Chatbot.

CHAPTER FOUR
DESIGN AND IMPLEMENTATION

The Design and Implementation chapter represents the culmination of theoretical insights and practical endeavors, embodying the transition from abstract concepts to tangible solutions. It serves as the cornerstone of this research endeavor, encapsulating the systematic process through which theoretical frameworks were transformed into functional systems. In this chapter, we embark on a journey through the intricacies of system design, exploring the underlying principles, methodologies, and considerations that informed every aspect of the development process. From the initial conceptualization to the final realization, each step is meticulously documented, providing readers with a comprehensive understanding of the rationale behind design decisions, the architecture of the system, and the methodologies employed to ensure its efficacy. Through detailed exposition and empirical validation, this chapter illuminates the intricate interplay between theory and practice, underscoring the importance of a systematic approach to system design and implementation in advancing the boundaries of knowledge and innovation.

4.1. Architectural Framework of the Chatbot

The common chatbot system design and development has two phases of process, training process and intent classification process. In the training process, the textual data preprocessing tasks such as text normalization, cleaning, tokenization and stop word removing are applied. They refine the training data to list of words (tokens). Following it, word vectorization and model building tasks are performed. The word vectorization task is used to convert the text into a vector of features which is required by machine learning algorithms. After the training process is completed, the built classification model is saved. In the intent classification process, the proposed system is designed to interact with user by providing relevant answers to the user's inquiry. It is designed based on accepting user's text input via GUI chat interface and then text preprocessing and word vectorization tasks are applied. Then the pre-trained model makes intent classification or prediction, and finally retrieves a text (response).

The following diagram shows the training process and architectural framework of the proposed chatbot system.

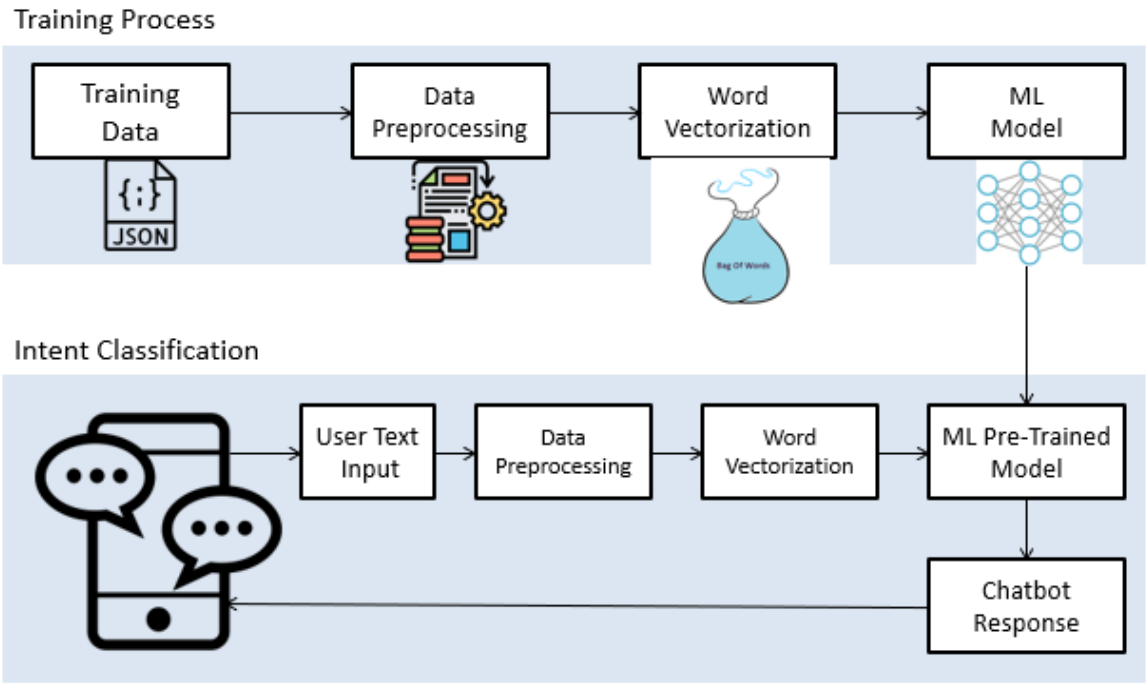


Figure 3: Architectural framework of the chatbot system

4.2. Text Preprocessing

The design and implementation of text preprocessing tasks represent a pivotal phase in the journey towards harnessing the power of natural language processing (NLP) techniques for meaningful analysis and understanding of textual data. Text preprocessing lays the groundwork for subsequent stages in NLP pipelines, serving as a critical preprocessing step to transform raw textual data into a structured format amenable to computational analysis. In this section, we delve into the intricate details of text preprocessing, elucidating the methodologies, techniques, and considerations underpinning each task. From text normalization and cleaning to tokenization and stop word removal, each preprocessing task is meticulously crafted to address the unique linguistic nuances and challenges inherent in the target language. Through a systematic exploration of design choices and implementation strategies, this section endeavors to shed light on the complexities of text preprocessing and its pivotal role in facilitating downstream NLP tasks.

4.2.1. Text Normalization

The process of text normalization in Amharic entails conforming textual data to standardized conventions inherent in the language's writing system. In this study, text normalization

procedures are meticulously crafted to align with the linguistic principles governing Amharic orthography. The methodology involves identifying groups of letters within the text that share similar phonetic representations and subsequently selecting a canonical representation to replace the redundant variants. The ensuing pseudocode provides a structured framework elucidating the intricacies of the designed text normalization approach tailored specifically for the Amharic writing system.

INPUT: raw text file
OUTPUT: normalized text
START:
1. Read the raw text
2. WHILE (it is not the end of file):
    IF raw text contains '[ሃኅኃሐሓኻ]' THEN replace with 'ሀ'
    IF raw text contains '[ሑኁኹ]' THEN replace with 'ሁ'
    IF raw text contains '[ኺኼኾ]' THEN replace with 'ሂ'
    IF raw text contains '[ኄኼዄ]' THEN replace with 'ሄ'
    IF raw text contains '[ሕኅ]' THEN replace with 'ህ'
    IF raw text contains '[ኍሖኾ]' THEN replace with 'ሆ'
    IF raw text contains '[ሠሡሢሣሤሥሦ]' THEN replace accordingly with 'ሰሱሲሳሴስሶ'
    IF raw text contains '[ዓኣዐ]' THEN replace with 'አ'
    IF raw text contains '[ዑዒዔዕዖ]' THEN replace accordingly with 'ኡኢኤእኦ'
    IF raw text contains '[ጸጹጺጼጽ]' THEN replace accordingly with 'ፀፁፂፄፅ'
3. Return normalized text;
END:

```python
#method to normalize character level missmatch

def text_normalization(self,input_text):
        rep1=re.sub('[ሃኅኃሐሓኻ]','ሀ',input_text)
        rep2=re.sub('[ሑኁኹ]','ሁ',rep1)
        rep3=re.sub('[ኺኼኾ]','ሂ',rep2)
        rep4=re.sub('[ኄኼዄ]','ሄ',rep3)
        rep5=re.sub('[ሕኅ]','ህ',rep4)
        rep6=re.sub('[ኍሖኾ]','ሆ',rep5)
        rep7=re.sub('[ሠ]','ሰ',rep6)
        rep8=re.sub('[ሡ]','ሱ',rep7)
        rep9=re.sub('[ሢ]','ሲ',rep8)
        rep10=re.sub('[ሣ]','ሳ',rep9)
        rep11=re.sub('[ሤ]','ሴ',rep10)
        rep12=re.sub('[ሥ]','ስ',rep11)
        rep13=re.sub('[ሦ]','ሶ',rep12)
        rep14=re.sub('[ዓኣዐ]','አ',rep13)
        rep15=re.sub('[ዑ]','ኡ',rep14)
        rep16=re.sub('[ዒ]','ኢ',rep15)
        rep17=re.sub('[ዓ]','ኣ',rep16)
        rep18=re.sub('[ዕ]','እ',rep17)
        rep19=re.sub('[ዖ]','ኦ',rep18)
        rep20=re.sub('[ጸ]','ፀ',rep19)
        rep21=re.sub('[ጹ]','ፁ',rep20)
        rep22=re.sub('[ጺ]','ፂ',rep21)
        rep23=re.sub('[ጻ]','ፃ',rep22)
        rep24=re.sub('[ጼ]','ፄ',rep23)
        rep25=re.sub('[ጽ]','ፅ',rep24)
        rep26=re.sub('[ፀ]','ፆ',rep25)

        return rep26
```

Figure 3: Amharic text normalization function

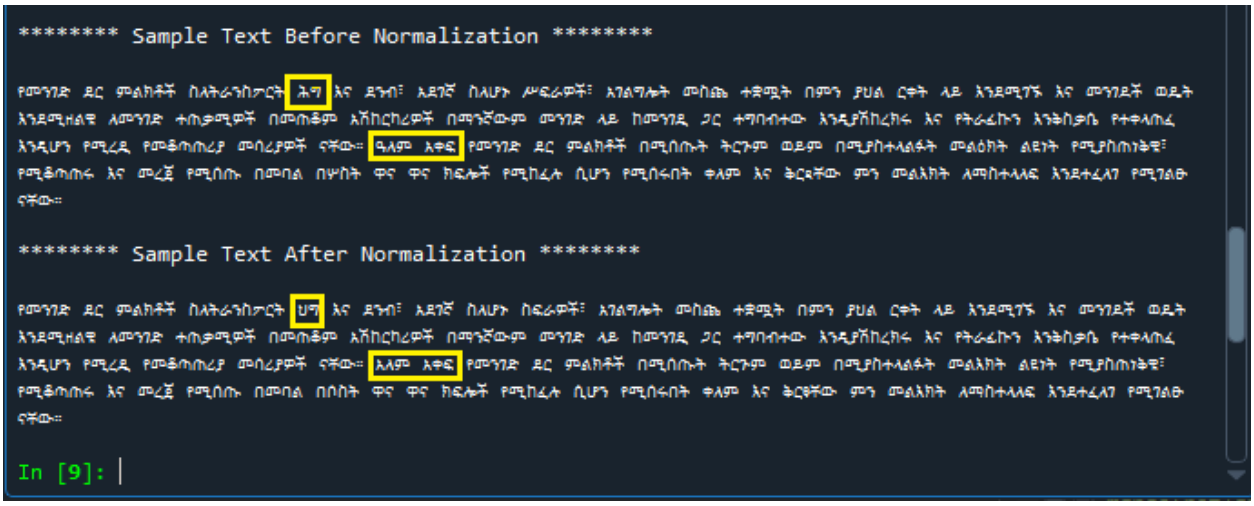


Figure 4: Sample text before and after normalization

4.1.1. Text Cleaning

In the design and implementation of the text cleaning subsection, the primary focus is on refining the normalized text by removing extraneous symbols, punctuation marks, ASCII characters, and numbers—both Arabic and Ethiopic digits—to facilitate the computer's comprehension of the natural language. This process involves eliminating various special symbols and punctuation marks, such as (\!\@\#\$\%\^\«\»\&*\(\)\…\[\]\{\}\;\“\”\›\’\‘\"\':\,\.\‹\/\<\>\?\\\\|\`\´\~\-\=\+\:\።\፤\;\፦-\፧\፨\※\፣), along with ASCII English characters (A-Za-z0-9) and Ethiopic digits ('\u1369-\u137C'). The following pseudocode provides a structured framework for the designed text clearing method tailored specifically for the Amharic writing system, ensuring the text is devoid of any non-essential elements that could hinder accurate natural language processing.

INPUT: uncleaned text
OUTPUT: cleaned text
START:
1. Read the uncleaned text
2. WHILE (it is not the end of file):
    IF text contains symbols & punctuations '[\!\@\#\$\?*\:\።\፤\;\፦-\፧\፨\※\፣]' THEN replace with ' '
    IF text contains ASCII characters and numbers '[A-Za-z0-9]' THEN replace with ' '
    IF text contains Ethiopic numbers [\'\u1369-\u137C\'] THEN replace with ' '
3. Return cleaned text;
END:

```
38      def text_cleaner(self,input_text):
39              #Removing special symbols and puctuations
40              symb_punc_removed = re.sub('[\!\@\#\$\%\^\«\»\&\*\(\)\…\[\]\{\}\;\"\”\›\
41              #Removing ASCII English characters (both upper and lower cases) and Arab
42              number_ascii_removed = re.sub('[A-Za-z0-9]','',symb_punc_removed)
43              #Removing Ethiopic digits
44              cleaned_text = re.sub('[\'\u1369-\u137C\']+','',number_ascii_removed)
45              return cleaned_text
46
47      sample_text = "የመንገድ ደር ምልክቶች ስለትራንስፖርት ሕግ እና ደንብ፣ አደገኛ ስለሆኑ ሥፍራዎች፣ አገልግሎት መስጫ ተቋሟት በምን
48
49      normalized_text = text_normalization(sample_text, sample_text)
50      clean_text = text_cleaner(normalized_text, normalized_text)
51
52      print("\n******** Sample Text Before Cleaning ********\n")
53      print(normalized_text)
54      print("\n******** Sample Text After Cleaning ********\n")
55      print(clean_text)
```

Figure 5: Amharic text cleaning function

```
******** Sample Text Before Cleaning ********

የመንገድ ደር ምልክቶች ስለትራንስፖርት ህግ እና ደንብ፣ አደገኛ ስለሆኑ ስፍራዎች፣ አገልግሎት መስጫ ተቋሟት በምን ያህል ርቀት ላይ እንደሚገኙ እና መንገደኞች ወዴት
እንደሚዘልቁ ለመንገድ ተጠቃሚዎች በመጠቆም አሽከርካሪዎች በማንኛውም መንገድ ላይ ከመንገዱ ጋር ተግባብተው እንዲያሽከረክሩ እና የትራፊኮን እንቅስቃሴ የተቀላጠፈ
እንዲሆን የሚረዳ የመቆጣጠሪያ መሰረያዎች ናቸው። አለም አቀፍ የመንገድ ደር ምልክቶች በሚሰጡት ትርጉም ወይም በሚያስተላልፉት መልእክት ልዩነት የሚያስጠነቅቁ፣
የሚቆጣጠሩ እና መረጃ የሚሰጡ በመባል በሶስት ዋና ዋና ክፍሎች የሚከፈሉ ሲሆን የሚሰሩበት ቀለም እና ቅርፃቸው ምን መልእክት ለማስተላለፍ እንደተፈለገ የሚገልፁ
ናቸው።

******** Sample Text After Cleaning ********

የመንገድ ደር ምልክቶች ስለትራንስፖርት ህግ እና ደንብ አደገኛ ስለሆኑ ስፍራዎች አገልግሎት መስጫ ተቋሟት በምን ያህል ርቀት ላይ እንደሚገኙ እና መንገደኞች ወዴት
እንደሚዘልቁ ለመንገድ ተጠቃሚዎች በመጠቆም አሽከርካሪዎች በማንኛውም መንገድ ላይ ከመንገዱ ጋር ተግባብተው እንዲያሽከረክሩ እና የትራፊኮን እንቅስቃሴ የተቀላጠፈ
እንዲሆን የሚረዳ የመቆጣጠሪያ መሰረያዎች ናቸው አለም አቀፍ የመንገድ ደር ምልክቶች በሚሰጡት ትርጉም ወይም በሚያስተላልፉት መልእክት ልዩነት የሚያስጠነቅቁ
የሚቆጣጠሩ እና መረጃ የሚሰጡ በመባል በሶስት ዋና ዋና ክፍሎች የሚከፈሉ ሲሆን የሚሰሩበት ቀለም እና ቅርፃቸው ምን መልእክት ለማስተላለፍ እንደተፈለገ የሚገልፁ
ናቸው

In [11]:
```

Figure 6: Sample text before and after text cleaning

4.1.2. Tokenization

Following the normalization and clearing of the text, the subsequent step in the text preprocessing pipeline is tokenization. Tokenization involves segmenting longer strings of text into smaller units, or tokens, which in this context are individual words. This step is crucial as it lays the foundation for further text processing tasks, ensuring that subsequent analyses are conducted on appropriately segmented text. In this study, tokenization of Amharic text is implemented using word spaces as delimiters. The process is facilitated by the built-in Natural Language Toolkit (nltk) library, which provides robust support for tokenizing text in various

languages, including Amharic. At this stage, it was needed to iterate through the patterns and tokenize them. The words (tokens) were needed to be stored in a list. Thus, we created a list to store words, and named “words”. Based on these words from the pattern, its “tag” is identified. Then, we created a list to store these tags, and named “classes”.

The Tokenization task results in a list of words by using word spaces as a delimiter. We have manually provided various texts with different representations and assessed the experimental results. After it is implemented, the built-in function from nltk library has performed splitting the Amharic text into tokens of words without any wrong doing in all cases. Once the text is tokenized, duplicates are removed from these lists, and finally, stored these lists in the pickled form.

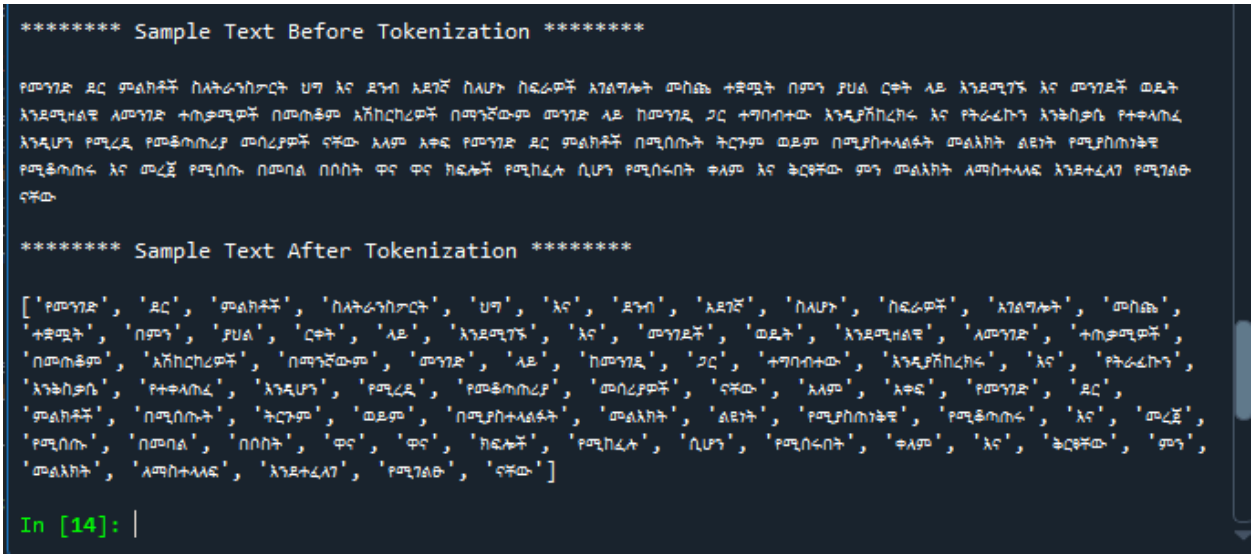


Figure 7: Sample text before and after text tokenization

4.1.3. Stop Word Removing

Following the tokenization process, the next critical step in text preprocessing is the removal of stop words. In this study, a comprehensive list of stop words has been compiled based on prior research and additional efforts. Stop words are terms that do not contribute significantly to the contextual or semantic understanding of a sentence and their removal helps in reducing the dataset size and enhancing model performance.

INPUT: list of words
OUTPUT: list of words without a stop word
VARIABL: stop_word_list = []
1. Read the list of words
2. WHILE (word in list of words):
    IF word NOT IN stop_word_list THEN append word to list of words
3. Return list of words;
END:

Examples of identified stop words include: ['ነው', 'ናቸው', 'እና', 'ወይም', 'ስለ', 'የሚሆኑ', 'ለዚያው', 'ከሆነው', 'ለዚሁ', 'ጥቂት', 'በርካታ', 'ብቻ', 'ሌሎች', 'ሌላ', 'ሁሉም', 'ሁሉ', 'በኋላ', 'አንዳንድ']. The above pseudocode illustrates the methodology designed for the removal of stop words in this study. In this study, stop word list is prepared based on previous studies and our own effort. The implemented stop word removal has successfully able to remove all the specified stop words from the tokens or words. We have manually provided various texts with stop words and assessed the experimental results. It has performed removing the specified stop words from the Amharic text without any wrong doing in all cases.

Figure 8: Sample text before and after stop word removal

4.3. Bag-of-Words (BoW) Word Vectorization

The code snippet below is designed to prepare a Bag of Words (BoW) representation for training a machine learning model, specifically for the chatbot application. This BoW model is a

fundamental technique in natural language processing that transforms text into a vector of binary values, indicating the presence or absence of specific words within the text. The process begins by initializing an empty list called `training` to store the feature vectors and corresponding target labels. An `output_empty` list is created, containing zeros, with its length matching the number of classes or intents identified for the chatbot.

For each document in the `documents` list, which contains pairs of tokenized word patterns and their associated intents, an empty list called `bag` is initialized. The code then iterates over each word in the global `words` list, appending a 1 to the `bag` list if the word is present in the current document's word pattern, and a 0 otherwise. This results in a binary vector where each position corresponds to a word in the vocabulary, and the value indicates whether that word appears in the document.

Next, the code creates a copy of the `output_empty` list and updates it to reflect the intent of the current document. The index corresponding to the document's intent is set to 1, creating a one-hot encoded vector representing the target label. Finally, the feature vector (`bag`) and the target vector (`output_row`) are combined into a single list and appended to the `training` list.

```
#preparing bag of words for training
training = []
output_empty = [0] * len(classes)

for document in documents:
    bag =[]
    word_patterns = document[0]
    for word in words:
        bag.append(1) if word in word_patterns else bag.append(0)

    output_row = list(output_empty)
    output_row[classes.index(document[1])] = 1
    training.append([bag, output_row])

random.shuffle(training)
training = np.array(training, dtype=object)
```

Figure 9: Preparing bag of words for training

After all documents have been processed, the `training` list is shuffled to ensure a random order of samples, which helps improve the generalization of the model during training. The list is then converted into a NumPy array, with `dtype=object` to accommodate the mixed data types of feature vectors and target labels, making it ready for the next steps in the machine learning pipeline.

4.4. Model Building and Selection

In this study artificial neural networks are employed for model building because the prepared data is labeled or the target class is known. Artificial neural networks allow programs to recognize patterns and solve common problems in artificial intelligence, machine learning and deep learning. Designing an optimal neural network architecture is not trivial and requires optimal values of hyperparameters. Keras sequential API is used to build a deep neural network that has hidden layers. Initial hyperparameters were set and adapted heuristically one at a time until we are satisfied with the network architecture. Finally, the following hyperparameters specification is applied.

Table 2: Hyperparameters specifications

Hyperparameters	Implemented Values
Layers	Input Layer: Expects rows of data with the same vector size as the bag-of-words. Hidden Layer_1 = 128 neurons Hidden Layer_2 = 64 neurons Output Layer: It has equal number of nodes of the target classes or intents. It uses the softmax activation function.
Activation function	Relu
Loss function	categorical_crossentropy
Optimizer	SDG
Epochs	150
Dropouts	0.2
Learning rate	0.001
Decay	1e-6

Momentum	0.9
Batch size	5

The decision to use a 3-layer architecture in the neural network is based on a balance between model complexity and computational efficiency. A 3-layer deep neural network allows the model to capture sufficient complexity and non-linear patterns in the data without overfitting, which can happen with too many layers, especially when data is limited. The input layer handles the raw data (word embeddings), the hidden layer captures feature interactions, and the output layer is responsible for classifying user intents or providing responses. This design ensures the model remains both effective and computationally feasible.

The values of the hyper-parameters in the study were not derived directly from standard research papers or pre-established norms but were determined heuristically through an iterative process of experimentation. By running multiple experiments, the optimal configuration was fine-tuned using insights gained from neural network architectures shared on platforms like GitHub and Kaggle. These platforms provided valuable references to existing architectures, allowing for the exploration of different combinations of hyper-parameters, such as learning rates, batch sizes, and the number of epochs, to suit the specific requirements of the Amharic chatbot model.

Through a trial-and-error approach, adjustments were made by observing model performance metrics such as accuracy and loss during training and validation phases. This iterative process helped in identifying the best set of hyper-parameters, ensuring the model was well-optimized to handle the nuances of Amharic text while balancing computational efficiency. This approach reflects a practical, hands-on methodology in tuning the model for real-world applications rather than adhering strictly to academic standards.

As we explained on data preparation, we prepared our dataset by using JSON. This dataset is categorized based on its intents. The intents are the contextual meaning that the users request or query that have. Therefore, those intents are used as a class for prediction while we train the model. Therefore, we have many intents or target classes in our dataset. This makes the model selection criteria focus on multiclass classification. Hence, Keras model is proposed using a softmax activation function on the output layer. This means that the Keras model will predict a

vector with the same size as the bag-of-words (number of elements) with the probability that the sample belongs to each of the target classes, i.e., the intents.

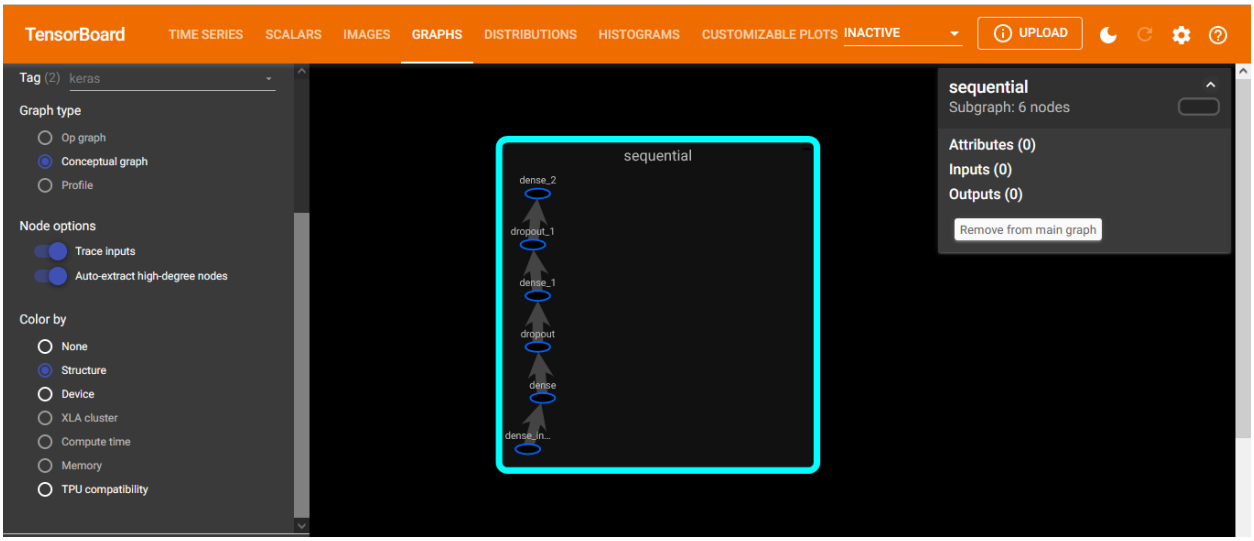


Figure 10: Visualization of the conceptual graph of the proposed Keras model

4.5. Graphical User Interface

The design and implementation of the Graphical User Interface (GUI) represent a critical component of the Amharic chatbot system, ensuring an intuitive and user-friendly interaction for the end-users. The GUI was developed using the Tkinter library, a standard Python interface to the Tk GUI toolkit. This choice was driven by Tkinter's simplicity and effectiveness in creating robust and interactive desktop applications. The primary objective of the GUI is to facilitate seamless communication between the user and the chatbot, allowing users to input their queries and receive responses in a coherent and natural manner. The interface was meticulously designed to accommodate Amharic text, ensuring that the linguistic nuances and script of the language were accurately represented. Special attention was given to the layout and functionality to ensure that users could easily navigate the application, enter their queries, and view the responses. The design process also incorporated feedback from initial user testing to refine the interface, making it more intuitive and responsive to user needs. The following sections detail the step-by-step implementation of the GUI, highlighting the technical decisions made and the functionalities incorporated to enhance user experience and system usability.

```
#Creating GUI with tkinter
import tkinter as tk
from tkinter import NORMAL, DISABLED, END

def send():
    message = EntryBox.get("1.0",'end-1c').strip()
    EntryBox.delete("0.0",END)
    if message != '':
        ChatLog.config(state=NORMAL)
        ChatLog.insert(END, "You: " + message + '\n\n')
        ChatLog.config(foreground="white", font=("Verdana", 12))
        ints = predict_class(message)
        res = get_response(ints, intents)
        ChatLog.insert(END, "Bot: " + res + '\n\n')
        ChatLog.config(state=DISABLED)
        ChatLog.yview(END)
base = tk.Tk()
base.title("በትራፊክ ደንብና መሪና የመንደት ክህሎት የሚያግዝ AI ቢስተም")

from PIL import Image, ImageTk

#this is the image file
path = "traffic-sign-icon.png"
load = Image.open(path)
render = ImageTk.PhotoImage(load)
base.iconphoto(False, render)
base.geometry("400x500")
base.resizable(width=False, height=False)

#Create Chat window
ChatLog = tk.Text(base, bd=0, bg="#00688B", height="8", width="100", font="Aria
ChatLog.config(state=DISABLED)
#Bind scrollbar to Chat window
```

Figure 11: A code snapshot for creating GUI with tkinter

CHAPTER FIVE
RESULTS AND DISCUSSION

The Results and Discussion chapter provides a comprehensive analysis of the outcomes derived from the implemented methodologies and their implications in the context of the research objectives. This chapter delves into the performance metrics of the proposed model, examining its accuracy, and other relevant evaluation criteria. Through detailed tables, graphs, and analyses, the effectiveness of the text preprocessing techniques such as normalization, cleaning, tokenization, and stop word removal is scrutinized. Additionally, the chapter discusses the significance of these results in relation to existing studies and theoretical frameworks, highlighting the contributions of this research to the field. The discussion also addresses any anomalies or unexpected findings, providing potential explanations and suggesting avenues for further investigation. By synthesizing the data and reflecting on the practical implications, this chapter aims to offer a thorough understanding of how the implemented approaches have advanced the state of knowledge and what future directions can be pursued to build on these findings.

5.1. Data Preparation

Once the data collection task was completed, the dataset was meticulously organized in a JSON file format to facilitate structured storage and ease of access. The dataset comprises three primary components: tags, patterns, and responses. The tag represents the intent or the underlying idea of the user query, serving as the target class during the model training phase. This structured approach ensures that each user query can be accurately classified into one of the identified intents. The comprehensive dataset includes 27 distinct intents or target classes, meticulously identified through thorough analysis during the data preparation phase. To effectively manage this data, an "intents.json" file was created, encapsulating the information in a systematic manner. Each entry in this file includes tags, which categorize the data into specific classes; patterns, which are combinations of words that users might use to query the chatbot; and responses, which are the expected outputs based on the given patterns. This structured data format not only aids in training the model but also ensures that the chatbot can deliver accurate and relevant responses to user queries. The following table provides a detailed description of all

27 intents or target classes identified in the dataset preparation stage, highlighting the thoroughness and precision of the data preparation process.

Table 3: Tag class description

No.	Tag	Description
1	ሰላምታ	This context of the intent is about greetings.
2	ስም	This context of the intents introduces the bot.
3	የትራፊክ_ህጎች_እና_ደንቦች	This context of the intent is about the most common traffic laws and regulations.
4	የማሽከርከር_ስነ_ባህሪ	This context of the intent is about driving behavior.
5	የትራፊክ_ማስተላለፊያ_መብራቶች	This context of the intent is about traffic lights.
6	የትራፊክ_አደጋ_መንስኤዎች	This context of the intent is about causes of traffic accidents.
7	አቅጣጫ_መቀየሪያ_ህጎች	This context of the intent is about diversion of traffic lanes.
8	የመንገድ_አጠቃቀም_ህጎች	This context of the intent is about rules of road use.
9	የእንስሳትን_የመንገድ_ላይ_እንቅስቃሴ_የሚወስኑ_ህጎች	This context of the intent is about Laws that determine the movement of animals on the road.
10	የተሽከርካሪ_ቴክኒክ_ብቃት_አፈታተሽ	This context of the intent is about troubleshooting technical issues of a vehicle.
11	የሞተር_ማቀዝቀዣ_ውሃ_መቀየር	This context of the intent is about changing water for cooling the engine.
12	የሞተር_ዘይት_መቀየር	This context of the intent is about changing oil engine.
13	የኃይል_አስተላላፊ_ክፍሎች_ሰርቪስ_ማድረግ	This context of the intent is about servicing power transmission units.
14	የአየር_ማጣሪያ	This context of the intent is about air filter.
15	የኤሌክትሪክ_መስመሮች_አፈታተሽ	This context of the intent is about checking electric power lines.
16	የፍሬን_እና_የመሪ_ሁኔታ_አፈታተሽ	This context of the intent is about checking brake and steering condition.
17	ተሽከርካሪ_በሚጸዳበት_ጊዜ_የሚወሰድ_ቅድመ_ጥንቃቄ	This context of the intent is about precautions to be taken when cleaning a vehicle.
18	የቆሻሻ_የራዲያተር_ውሃ_በሚቀየርበት_ወቅት_የሚከናውኑ_ተግባራት	This context of the intent is about activities performed during the change of waste radiator water.
19	ለተሽከርካሪ_ጎማ_መደረግ_ያለበት_ጥንቃቄ	This context of the intent is about precautions to be taken for vehicle tires.
20	የነዳጅ_ፊልትሮን_መቀየር	This context of the intent is about changing fuel filter.
21	ባ.መ.ዘ.ዉ.ን.ነ._BLOWAF _ምርመራ	This context of the intent is about the main important parts of a vehicle abbreviated as Battery, Light, Oil, Water, Air, and Fuel.
22	የጥፋት_እና_ቅጣት_ምድብ_የቅጣት_አፈፃፀም_እና_የሪከርድ_አያያዝ_ህጎች	This context of the intent is about penalty category, penalty execution and record keeping rules.
23	የተሽከርካሪ_ክብደት_ርዝመት_ስፋት_እና_የጭነት_መጠን	This context of the intent is about Vehicle weight, length, width and load capacity.
24	መረጃን_የመሰብሰብና_የመተርጎም_ሂደት	This context of the intent is about the process of collecting and interpreting information.
25	ምስጋና	This context of the intent is about gratitude.
26	ማቋረጥ	This context of the intent is about to cancel the conversation.
27	ግልጽ_ያልሆነ	This context of the intent is about anything that cannot be classified to the other intents. The conversation falls in this category may be due to spelling errors or unidentified ideas.

The provided code snippet demonstrates the initial steps of data preparation for a chatbot model using a JSON file containing user intents. It begins by importing the `json` module, which is essential for working with JSON data in Python. The code then opens the file `intents.json` located at the specified path using a context manager, ensuring the file is properly closed after reading. The content of the file is loaded into a Python dictionary named `data`. Following this, the code utilizes the `pandas` library, imported as `pd`, to convert the 'intents' section of the dictionary into a DataFrame. This DataFrame, stored in the variable `df`, provides a tabular representation of the dataset, facilitating easier data manipulation, analysis, and subsequent preprocessing steps. Displaying the DataFrame allows for a quick inspection of the dataset's structure, ensuring that tags, patterns, and responses are correctly loaded and ready for further processing in the model training pipeline.

In [2]:

```
import json

with open('/kaggle/input/chatbot-dataset/intents.json', 'r') as f:
    data = json.load(f)

df = pd.DataFrame(data['intents'])
df
```

Out[2]:

	tag	patterns	responses
0	[illegible]	[illegible]	[illegible]
1	[illegible]	[illegible]	[illegible]
2	[illegible]	[illegible]	[illegible]
3	[illegible]	[illegible]	[illegible]
4	[illegible]	[illegible]	[illegible]
5	[illegible]	[illegible]	[illegible]
6	[illegible]	[illegible]	[illegible]
7	[illegible]	[illegible]	[illegible]
8	[illegible]	[illegible]	[illegible]
9	[illegible] BLOWAF [illegible]	[illegible] BLOWAF [illegible]	[illegible]
10	[illegible]	[illegible]	[illegible]
11	[illegible]	[illegible]	[illegible]
12	[illegible]	[illegible]	[illegible]
13	[illegible]	[illegible]	[illegible]
14	[illegible]	[illegible]	[illegible]
15	[illegible]	[illegible]	[illegible]
16	[illegible]	[illegible]	[illegible]
17	[illegible]	[illegible]	[illegible]
18	[illegible]	[illegible]	[illegible]
19	[illegible]	[illegible]	[illegible]
20	[illegible]	[illegible]	[illegible]
21	[illegible]	[illegible]	[illegible]
22	[illegible]	[illegible]	[illegible]
23	[illegible]	[illegible]	[illegible]
24	[illegible]	[illegible]	[illegible]
25	[illegible]	[illegible]	[illegible]
26	[illegible]	[illegible], asddg, qwert]	[illegible]

Figure 12: Displaying the DataFrame

5.2. Explanatory Data Analysis

The Explanatory Data Analysis (EDA) subsection is a crucial phase in the data preparation process, aimed at gaining insights into the dataset's underlying structure and patterns. EDA involves using statistical tools and visualizations to summarize the main characteristics of the data, ensuring it is ready for model training. By performing EDA, we can identify any anomalies, missing values, or inconsistencies within the dataset, thereby informing necessary preprocessing steps. This thorough examination helps in understanding the distribution of different intents, the frequency of various patterns, and the overall data quality, which are essential for building a robust and effective chatbot model.

In [9]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 137 entries, 0 to 136
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   tag        137 non-null    object
 1   patterns   137 non-null    object
 2   responses  137 non-null    object
dtypes: object(3)
memory usage: 3.3+ KB
```

In [10]:
```
def num_classes(df, target_col, ds_name="df"):
    print(f"The {ds_name} dataset has {len(df[target_col].unique())} classes")

num_classes(df, 'Tag', "Chatbot")
```

```
The Chatbot dataset has 27 classes
```

Figure 13: Summary of the DataFrame

```
In [17]:
plt.figure(figsize=(20, 20))
wc = WordCloud(font_path='/kaggle/input/noto-sans-ethiopic/NotoSansEthiopic-VariableFont_wdth,wght.ttf', max_words=2000, width=1600,
height=800).generate(' '.join(df['Pattern']))
plt.imshow(wc, interpolation='bilinear')
plt.xticks([])
plt.yticks([])
plt.show()
```

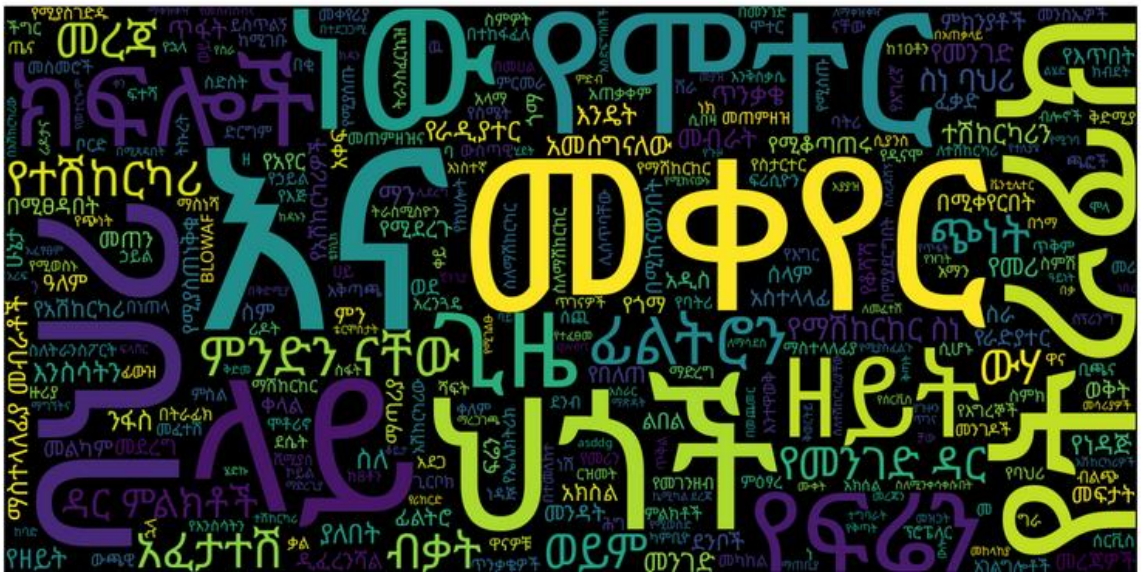


Figure 14: Visualizing the most frequently occurring words

To visualize the most frequently occurring words in the dataset, we utilized a Word Cloud, a popular text visualization technique. The provided code snippet achieves this by first setting up a figure of size 20x20 inches to accommodate the Word Cloud. Using the WordCloud library, the font path for Ethiopic script is specified to ensure proper rendering of Amharic characters. The `max_words` parameter is set to 2000, allowing a substantial number of words to be included, while the `width` and `height` parameters define the dimensions of the generated Word Cloud. The `generate` method is used to create the Word Cloud from a concatenated string of all patterns in the dataset, highlighting the most common terms. Finally, `plt.imshow` is employed to display the Word Cloud image, with `bilinear` interpolation for smoother visualization. Axes ticks are removed for a cleaner presentation, providing an insightful overview of the textual data's prominent features.

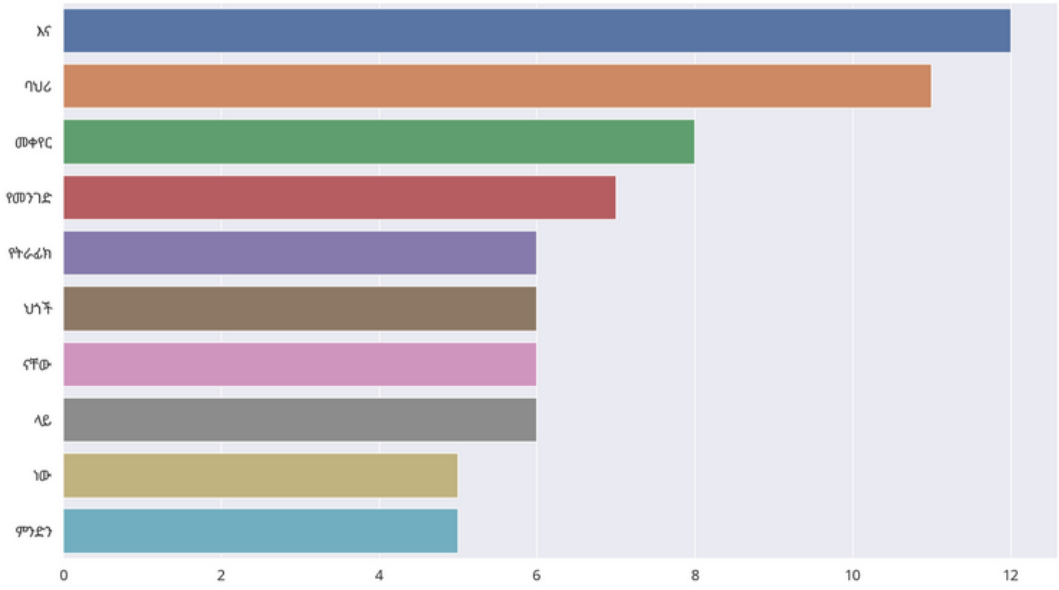


Figure 15: Common words in the data

5.3. Tag Pattern and Response Analysis

To analyze the distribution of tags, patterns, and responses within the dataset, a visual representation is created using Seaborn. The code begins by setting the font to "Noto Sans Ethiopic" to ensure proper rendering of Amharic characters and adjusting the font scale for readability. The order variable is set to arrange the tags based on their frequency in descending order. The count_plot function is then used to generate a bar plot that illustrates the distribution of different tags within the chatbot dataset.

```
In [14]:
sns.set(font_scale = 1.2, font="Noto Sans Ethiopic")
order = df['Tag'].value_counts().index
count_plot(df['Tag'], df, "Chatbot Tags distrbution", "Tags", "Frequency", 20,10, order=order,rotation=True, palette="summer")
```

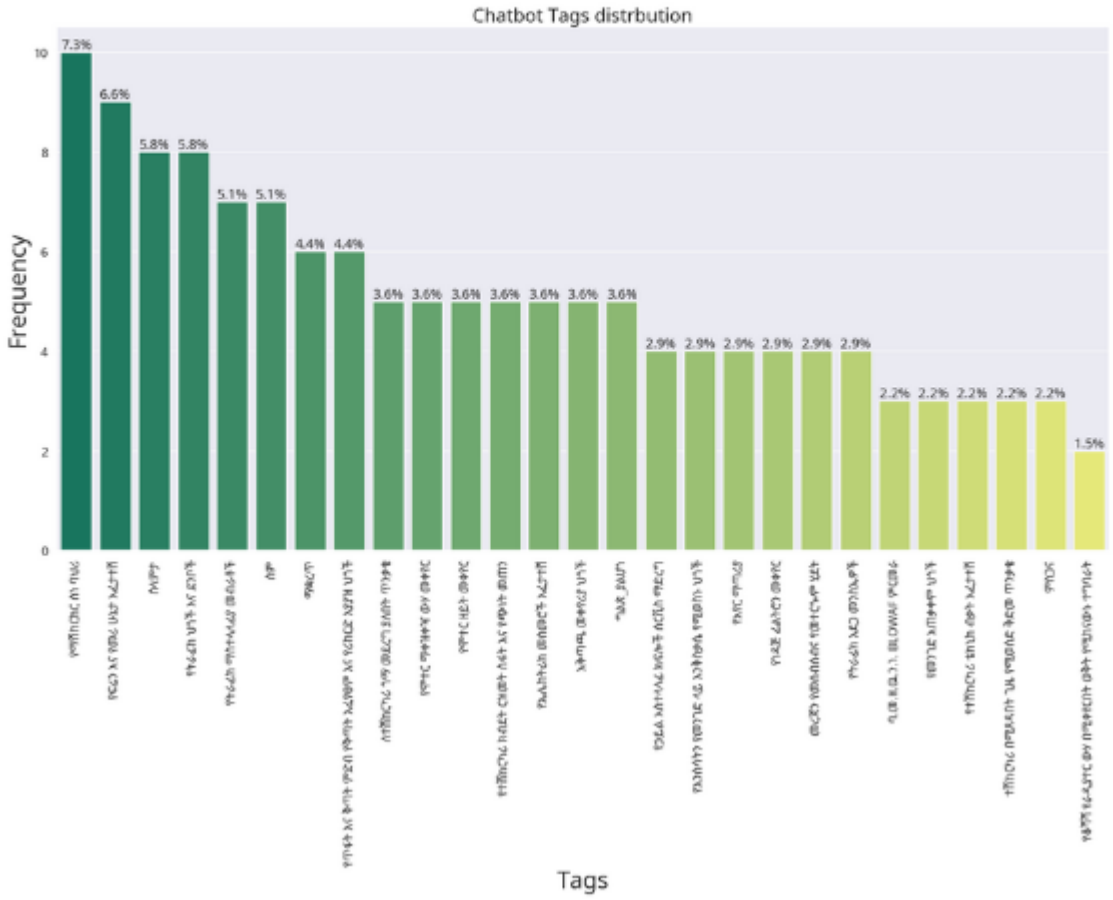


Figure 16: Tags distribution

This plot is customized with the title "Chatbot Tags Distribution" and labeled axes for clarity. The plot dimensions are set to 20x10 inches to ensure the plot is spacious and easy to interpret. Additionally, the tags are rotated for better readability, and the "summer" palette is used to provide a visually appealing color scheme. This analysis helps in understanding the frequency and distribution of various tags, which is crucial for evaluating the dataset's comprehensiveness and the chatbot's potential response coverage.

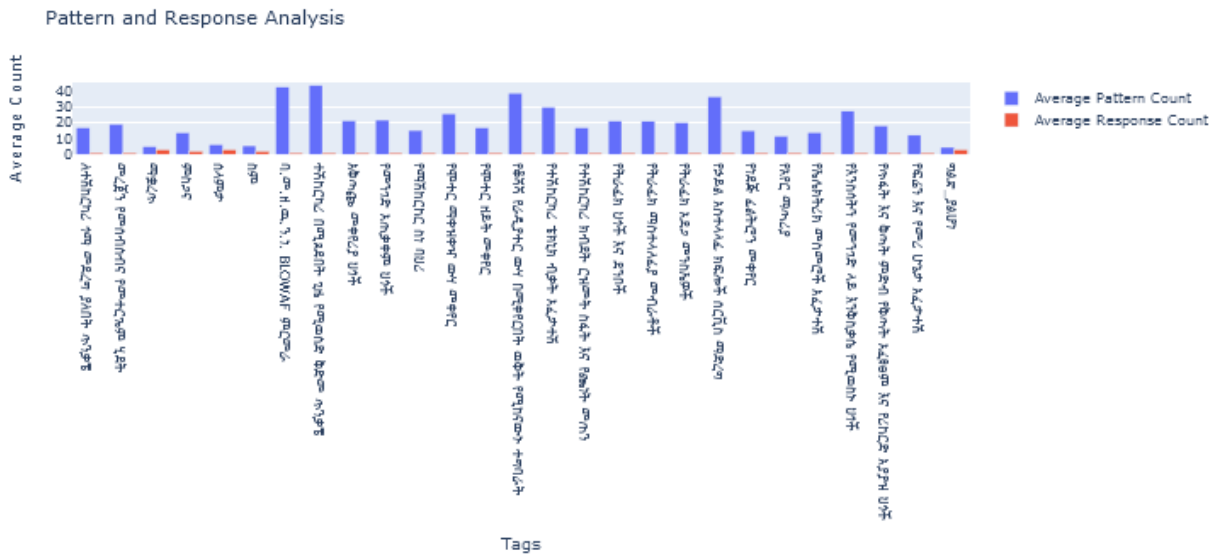


Figure 17: Pattern and response analysis

5.4. Performance Evaluation

Once the various text preprocessing tasks were completed, the vectorized textual data was prepared for training a machine learning model. The dataset was divided into an 80% training set and a 20% validation set to facilitate robust model training and evaluation. The Keras Sequential API was utilized to construct a deep neural network comprising three layers. Initial hyperparameters were set and iteratively adjusted based on heuristic methods until an optimal network architecture was achieved. The Keras model was compiled using the Stochastic Gradient Descent (SGD) optimizer and subsequently trained over 150 epochs. Upon satisfactory training and validation performance, the final model was saved in h5 format for deployment and further use. This meticulous approach ensured that the model was well-calibrated to handle the nuances of the Amharic language and effectively respond to user queries in the context of license training.

The results, illustrated in the figure below, include the loss and accuracy metrics for the intent classification task, revealing that the model achieved approximately 87.9% accuracy on the test set. Throughout the experimentation phase, TensorBoard was employed to provide essential measurements and visualizations within the machine learning workflow. Line plots were created to display the accuracy and loss for the model on both the training and test sets over each training epoch, as depicted in the figures below. From these plots, it is evident that the model demonstrated significant generalization skills, maintaining high performance on both the training

and test datasets. This indicates that the model not only learned the training data effectively but also retained the ability to generalize its learning to unseen data, thereby confirming its robustness and reliability in the context of Amharic intent classification.

```
val_loss: 1.0008 - val_accuracy: 0.8788
Epoch 142/150
16/16 [==============================] - 0s 27ms/step - loss: 1.1001e-04 - accuracy: 1.0000 -
val_loss: 1.0008 - val_accuracy: 0.8788
Epoch 143/150
16/16 [==============================] - 1s 37ms/step - loss: 2.5468e-04 - accuracy: 1.0000 -
val_loss: 1.0007 - val_accuracy: 0.8788
Epoch 144/150
16/16 [==============================] - 0s 28ms/step - loss: 2.2020e-04 - accuracy: 1.0000 -
val_loss: 1.0005 - val_accuracy: 0.8788
Epoch 145/150
16/16 [==============================] - 0s 31ms/step - loss: 2.7969e-04 - accuracy: 1.0000 -
val_loss: 1.0004 - val_accuracy: 0.8788
Epoch 146/150
16/16 [==============================] - 0s 29ms/step - loss: 6.8183e-04 - accuracy: 1.0000 -
val_loss: 1.0010 - val_accuracy: 0.8788
Epoch 147/150
16/16 [==============================] - 0s 31ms/step - loss: 8.9086e-04 - accuracy: 1.0000 -
val_loss: 1.0004 - val_accuracy: 0.8788
Epoch 148/150
16/16 [==============================] - 0s 28ms/step - loss: 8.8405e-05 - accuracy: 1.0000 -
val_loss: 1.0003 - val_accuracy: 0.8788
Epoch 149/150
16/16 [==============================] - 0s 26ms/step - loss: 2.3651e-04 - accuracy: 1.0000 -
val_loss: 1.0003 - val_accuracy: 0.8788
Epoch 150/150
16/16 [==============================] - 0s 26ms/step - loss: 5.0046e-04 - accuracy: 1.0000 -
val_loss: 1.0007 - val_accuracy: 0.8788
3/3 [==============================] - 0s 11ms/step - loss: 4.7993e-07 - accuracy: 1.0000
2/2 [==============================] - 0s 38ms/step - loss: 1.0007 - accuracy: 0.8788
Train: 1.000, Test: 0.879

Training session complete!


Serving TensorBoard on localhost; to expose to the network, use a proxy or pass --bind_all
TensorBoard 2.12.3 at http://localhost:6006/ (Press CTRL+C to quit)
```

Figure 18: Completion of training Keras model

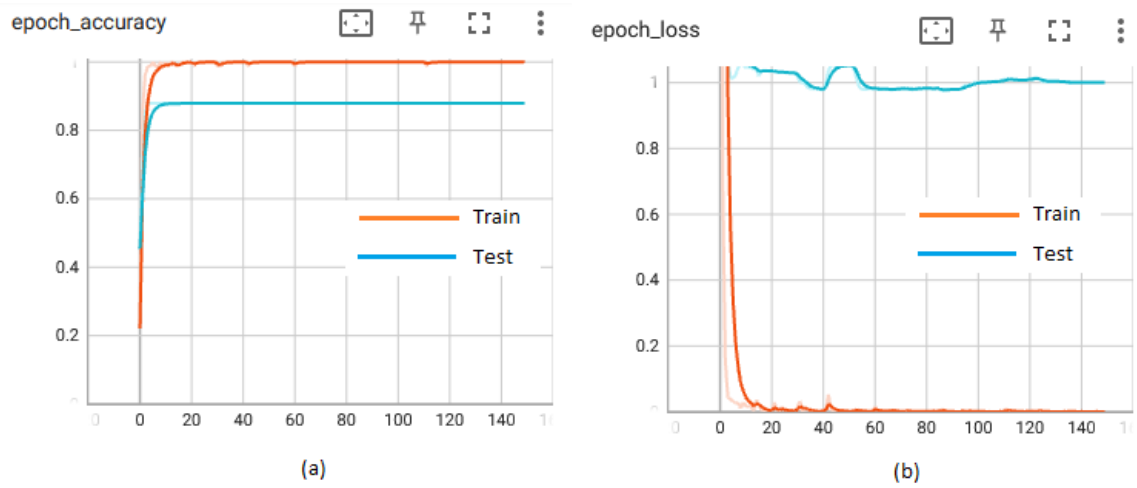


Figure 19: Line plots showing (a) Accuracy Progression Over Training Epochs on Training and Test Sets (b) Loss Reduction Over Training Epochs on Training and Test Sets

The above plot (a) shows how the accuracy of the model changes over time during the training process. The training accuracy generally increases with more epochs, indicating that the model is learning from the data. The test accuracy, which represents performance on unseen data, should ideally follow a similar upward trend without deviating significantly. In plot (b) the loss function measures how well the model is predicting the output compared to the actual output. This plot shows a decreasing trend for both training and test losses as the model becomes more accurate. A steady decrease in training loss indicates that the model is learning, while a similar reduction in test loss suggests the performance of generalization.

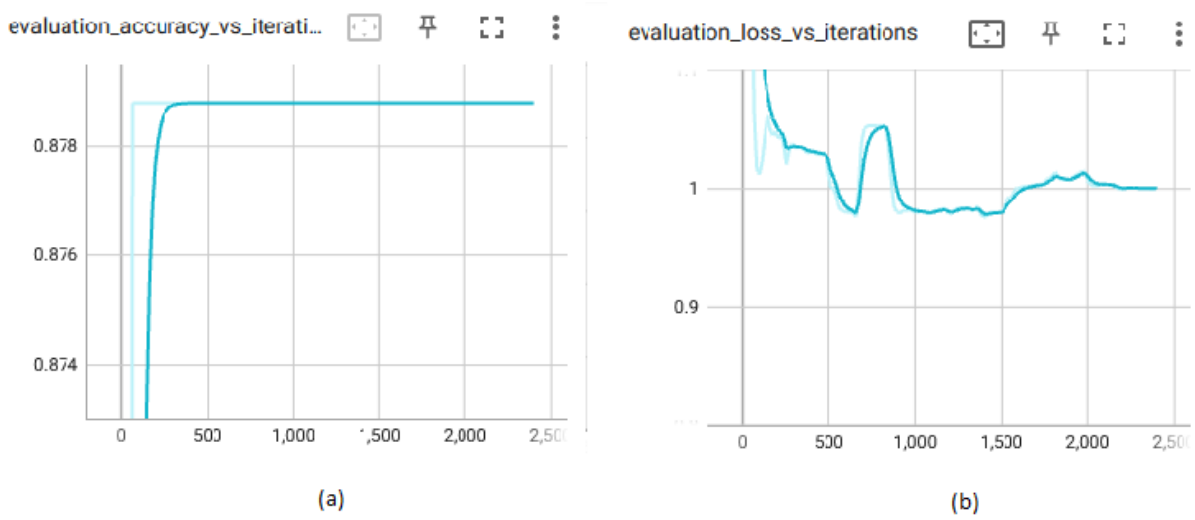


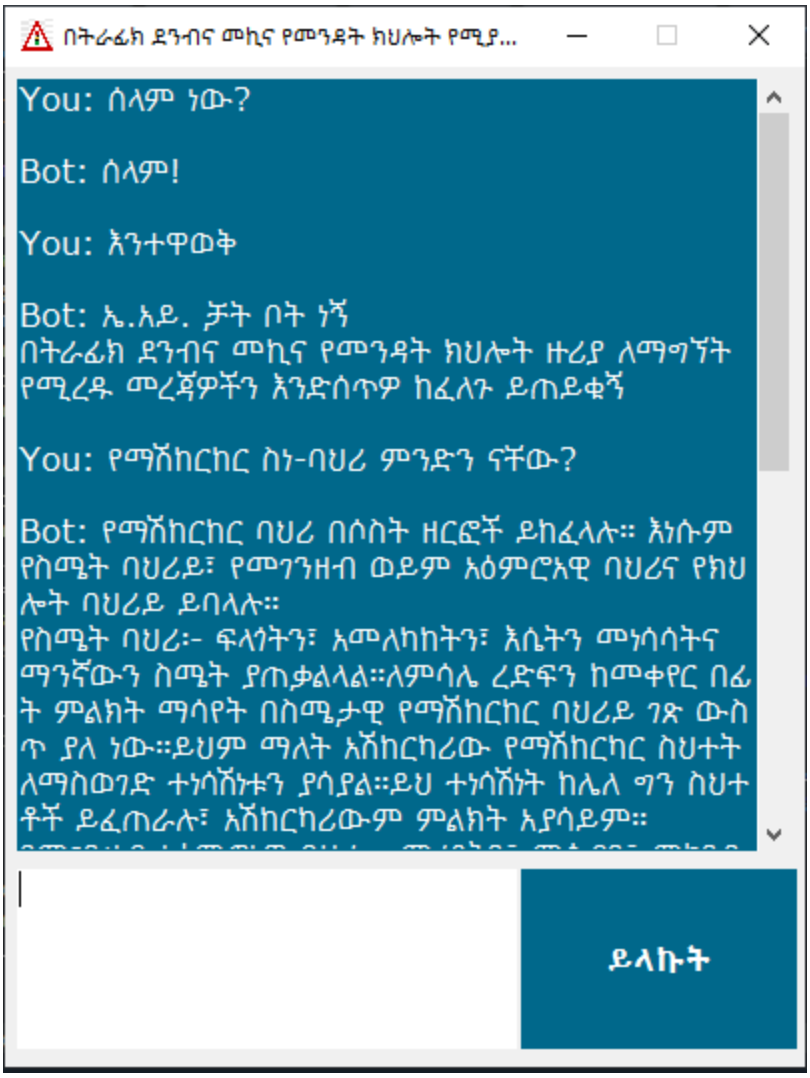
Figure 20: Line plots showing (a) Accuracy Evaluation at Various Iteration Points, and (b) Loss Evaluation at Various Iteration Points

The above plot (a) visualizes how the accuracy evolves as the model is evaluated at different iteration points. It helps assess how stable the model's performance is as training progresses. A consistent increase in accuracy with each iteration indicates that the model is improving steadily. Any fluctuations could suggest instability or issues with learning rate or model convergence. In plot (b) it tracks how the loss changes across iterations during evaluation. A smooth decrease in loss is a positive sign that the model is converging and improving its predictions. The significant

jumps or fails indicate issues with the model's learning process or the presence of noise in the data affecting the training.

5.5. Graphical User Interface (GUI)

To create an intuitive and user-friendly graphical user interface (GUI) for our chatbot, we utilized the Tkinter library in Python. Tkinter provides a robust framework for building GUI applications, enabling us to design a window where users can input their queries and receive responses from the chatbot. The interface facilitates a seamless text-to-text conversation between the user and the bot, displaying responses directly on the screen through functions we have implemented. The figure below illustrates an example interaction, showcasing how the chatbot responds naturally in Amharic text. This natural interaction enhances user experience by providing relevant and coherent replies to user queries. However, the effectiveness of the chatbot's responses may vary depending on how users interact with it. To address potential issues such as spelling errors or ambiguous queries, we have incorporated an intent labeled "ግልጽ_ያልሆነ," which translates to "unclear." This intent is designed to handle any input that cannot be classified under the predefined intents. When the chatbot encounters such input, it responds with messages like "ይቅርታ፤ ምን እንደፈለጉ አልገባኝም" ("Sorry, I did not understand what you meant"), "እባክዎ ተጨማሪ መረጃ ይጻፉ" ("Please provide more information"), or "ምንም አልገባኝም" ("I did not understand anything"). These responses help manage user expectations and maintain the flow of conversation, ensuring that the chatbot can still provide meaningful interactions even when faced with unclear or erroneous input.


በትራፊክ ደንብና መኪና የመንዳት ክህሎት የሚያ...
You: ሰላም ነው?
Bot: ሰላም!
You: እንተዋወቅ
Bot: ኤ.አይ. ቻት በት ነኝ
በትራፊክ ደንብና መኪና የመንዳት ክህሎት ዙሪያ ለማግኘት የሚረዱ መረጃዎችን እንድሰጥዎ ከፈለጉ ይጠይቁኝ
You: የማሽከርከር ስነ-ባህሪ ምንድን ናቸው?
Bot: የማሽከርከር ባህሪ በሶስት ዘርፎች ይከፈላሉ። እነሱም የስሜት ባህሪይ፣ የመገንዘብ ወይም አዕምሮአዊ ባህሪና የክህሎት ባህሪይ ይባላሉ።
የስሜት ባህሪ፡- ፍላጎትን፣ አመለካከትን፣ እሴትን መነሳሳትና ማንኛውን ስሜት ያጠቃልላል።ለምሳሌ ረድፍን ከመቀየር በፊት ምልክት ማሳየት በስሜታዊ የማሽከርከር ባህሪይ ገጽ ውስጥ ያለ ነው።ይህም ማለት አሽከርካሪው የማሽከርካር ስህተት ለማስወገድ ተነሳሽነቱን ያሳያል።ይህ ተነሳሽነት ከሌለ ግን ስህተቶች ይፈጠራሉ፣ አሽከርካሪውም ምልክት አያሳይም።
ይላኩት

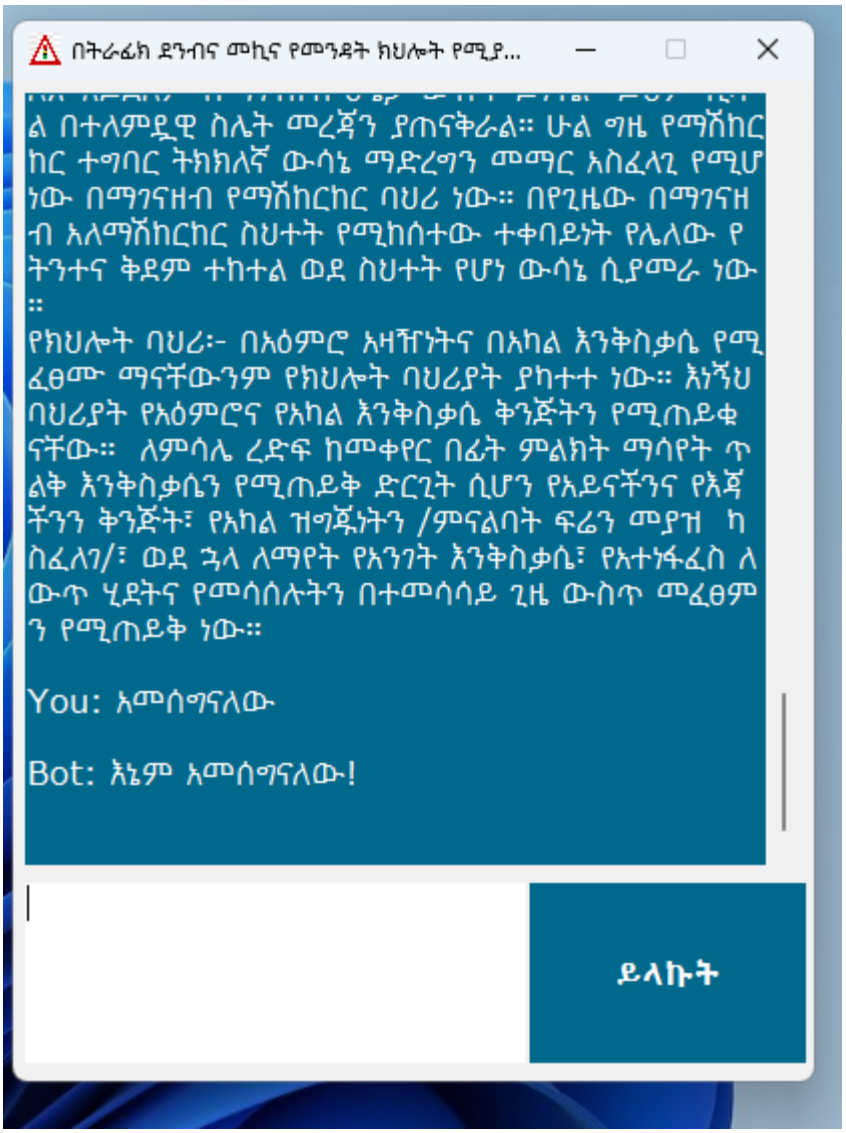


Figure 21: User-friendly graphical user interface (GUI) for the chatbot

5.6. User Acceptance Testing

To conduct a thorough user acceptance testing for the chatbot system, we adapted the criteria from Prat et al. (2014) to evaluate several key aspects. The evaluation involved 30 users and focused on ease of use, time efficiency, operational efficiency, accuracy, completeness, utility, and learning capability. Each criterion was rated on a scale from 1 to 5, with 1 being the lowest and 5 being the highest. The following tables summarize the results of the user acceptance testing:

Table 4: User Acceptance Testing Results

Criteria	Scale	Average Score	Standard Deviation	% of Users Rating Satisfactory or Above
Ease of Use	1-5	4.2	0.6	87%
Time Efficiency	1-5	3.9	0.8	80%
Operational Efficiency	1-5	4.1	0.7	85%
Accuracy	1-5	4.0	0.5	83%
Completeness	1-5	3.8	0.9	76%
Utility	1-5	4.3	0.6	90%
Learning Capability	1-5	3.7	0.8	75%

Explanation of Results:

Ease of Use (4.2/5): This indicates that users found the chatbot relatively easy to navigate, with 87% rating it as satisfactory or above. A small percentage of users may have found it slightly difficult, reflected in the standard deviation.

Time Efficiency (3.9/5): Most users (80%) felt that the system responded quickly and efficiently, although a few felt that it took longer than expected to process some queries.

Operational Efficiency (4.1/5): This score suggests that the chatbot operates smoothly, with minimal lag or crashes, contributing positively to the user experience.

Accuracy (4.0/5): Users found the system generally accurate in providing correct responses to driving-related queries, though there were minor concerns reflected by the lower percentage (83%) of users rating it satisfactory or above.

Completeness (3.8/5): While the chatbot generally covers most topics, some users (76%) felt that it could benefit from more comprehensive content, possibly addressing specific or nuanced queries.

Utility (4.3/5): This high score indicates that users found the chatbot useful in their preparation for driving tests, with 90% rating it satisfactory or above.

Learning Capability (3.7/5): The chatbot's ability to adapt to user input was rated slightly lower, indicating room for improvement in terms of personalized responses and learning from interactions.

Table 5: Detailed User Acceptance Results by Category

Category	Criteria	Average Score	Standard Deviation	% of Users Rating Satisfactory or Above
Usability	Ease of Use	4.2	0.6	87%
	Time Efficiency	3.9	0.8	80%
	Operational Efficiency	4.1	0.7	85%
Functionality	Accuracy	4.0	0.5	83%
	Completeness	3.8	0.9	76%
Utility & Learning	Utility	4.3	0.6	90%
	Learning Capability	3.7	0.8	75%

Summary of the Results by Category:

Strengths: High scores in ease of use, utility, and operational efficiency suggest the chatbot is well-designed and provides significant value to end-users in preparing for their driving tests.

Areas for Improvement: Lower scores in completeness and learning capability suggest that users would appreciate more comprehensive content and an improved ability of the system to learn from user inputs to provide more tailored responses.

This table provides a comprehensive assessment of the chatbot's performance based on user feedback, demonstrating its effectiveness and highlighting areas for further enhancement.

5.7. Discussion of the Findings

The discussion of findings section aims to address the research questions posed in this study, focusing on the design and implementation of an Amharic chatbot tailored for license trainees. In line with the research objectives, this section evaluates the methods employed in developing the chatbot, explores suitable vector representations for the limited Amharic corpus, examines the machine learning techniques utilized for maintaining conversational flexibility, and discusses the

metrics preferred for evaluating the model's performance. By systematically analyzing these research questions, we can gain insights into the effectiveness of the proposed approach in addressing the challenges of designing an interactive and efficient chatbot system for license trainees. Additionally, this discussion provides valuable implications for future research directions and practical applications in the domain of natural language processing and conversational AI.

Research Question 1: How to prepare a dataset for designing an Amharic chatbot that provides appropriate answers for license trainees?

The findings from the development and evaluation phases of the Amharic chatbot reveal key insights into the effective preparation of a dataset for such a system. One of the first considerations in preparing the dataset was identifying relevant content that aligns with the core needs of driving license trainees in Ethiopia. The dataset was compiled from authoritative sources such as books and driving manuals, including "የመንጃ ፈቃድ መጣሪያ" and "የመንጃ ፈቃድ 1000 ጥያቄዎችና መልሶቻቸው." These resources provided foundational content related to driving rules, road signs, and general safety regulations. To enrich the dataset, additional sources such as websites and informal interviews with driving instructors were utilized, ensuring that the chatbot's knowledge base encompassed both theoretical and practical aspects of driving education.

A crucial step in preparing the dataset involved preprocessing the Amharic text to ensure that the chatbot could effectively interpret user queries and provide accurate responses. This process involved text normalization, tokenization, and stop word removal. For instance, normalization helped standardize variations in the Amharic script (e.g., ensuring consistent representation of characters like "አ" and "ዐ") to avoid ambiguity in the chatbot's responses. Tokenization divided the text into manageable pieces, such as individual words or phrases, while stop word removal streamlined the data by eliminating non-essential words like "ነው" (is) or "ስለ" (about), reducing computational complexity and enhancing focus on relevant content. These preprocessing techniques contributed to transforming raw text into structured, analyzable data that could be easily vectorized for training the chatbot model. The overall dataset preparation strategy ensured that the Amharic chatbot was equipped with an accurate, comprehensive, and well-structured

knowledge base capable of providing appropriate and relevant answers to driving license trainees.

Research Question 2: Suitable Vector Representation of Text for Limited Amharic Corpus

In addressing the challenge of working with a limited Amharic text corpus, the study explored various vector representation techniques to capture the semantic meaning of text data. Despite the constraints posed by the size of the dataset, the Bag-of-Words (BoW) method emerged as a suitable approach for converting text into numeric vectors. By representing each word as a feature and constructing a vocabulary based on the corpus, the BoW technique proved effective in transforming raw text data into a format suitable for machine learning algorithms.

Research Question 3: Appropriate Machine Learning Techniques for Conversational Text Dataset

To maintain flexibility in conversation and accommodate the nuances of a text-oriented dataset, the study evaluated various machine learning techniques for intent classification and response generation. Supervised learning algorithms, in particular neural networks, were employed to classify user queries into predefined intents or categories. Through iterative experimentation and model refinement, the chatbot system demonstrated adaptability and responsiveness in interpreting user input and generating appropriate responses. The iterative nature of the model development process allowed for the identification of optimal hyperparameters and feature representations, leading to improved conversational capabilities.

Research Question 4: Evaluating the Proposed Model Performance

In evaluating the performance of the proposed chatbot model, several metrics were considered to assess its accuracy, efficiency, and effectiveness. While accuracy provided insights into the model's classification performance, in addition, user acceptance testing played a crucial role in assessing the chatbot's usability, utility, and user satisfaction. By soliciting feedback from target users and domain experts, the study was able to identify areas for improvement and refinement in the chatbot system. Overall, the combination of quantitative metrics and qualitative feedback provided a comprehensive evaluation of the proposed model's performance and usability in real-world scenarios.

CHAPTER SIX
CONCLUSION AND FUTURE WORKS

6.1. Conclusion

In summary, this study aimed to design and implement an effective chatbot system capable of understanding and responding to user queries in Amharic. By employing a comprehensive approach that included text normalization, text cleaning, tokenization, and stop word removal, we ensured that the Amharic text data was adequately preprocessed for machine learning tasks. Feature extraction using the Bag-of-Words technique and model training with various algorithms allowed us to build a robust classification system. The performance of the chatbot was rigorously evaluated using accuracy metrics, and further validation was achieved through user acceptance testing, which provided valuable feedback on the system's practical usability and effectiveness.

One of the strengths of this study is the thorough preprocessing of the Amharic text, which addressed the unique linguistic characteristics of the language. The use of a well-structured JSON dataset with clearly defined tags, patterns, and responses facilitated efficient model training and prediction. Additionally, the implementation of a graphical user interface using the Tkinter library provided an accessible and user-friendly platform for interacting with the chatbot. The study also benefited from the use of established natural language processing tools and techniques, ensuring a high standard of methodological rigor.

However, the study also faced several challenges. The complexity of the Amharic writing system, with its various characters and phonetic nuances, presented difficulties in text normalization and tokenization. The limited availability of comprehensive Amharic text datasets necessitated the creation of a custom dataset, which was time-consuming and required careful validation. Furthermore, while the user acceptance testing provided valuable insights, the relatively small sample size of 30 users may not fully capture the diverse range of user interactions and preferences. Future research should aim to address these challenges by expanding the dataset, refining the preprocessing techniques, and conducting broader user testing to enhance the chatbot's performance and applicability in real-world scenarios.

6.2. Future Works

Based on the findings and challenges encountered in this study, several recommendations are outlined to further enhance the chatbot system's applicability and performance. First, addressing the dataset's class imbalance problem should be a priority in future work. The current study did not account for data imbalance, which can affect the model's ability to generalize across various categories of user queries. Implementing methods such as oversampling, undersampling, or using more advanced techniques like Synthetic Minority Over-sampling Technique (SMOTE) can help mitigate this issue, leading to more balanced and reliable model performance across different classes of user inputs. Balancing the dataset will be key to improving the chatbot's accuracy and responsiveness.

Secondly, expanding the dataset is crucial for enhancing the chatbot's understanding of diverse user queries. Increasing the volume and variety of Amharic text data, including colloquial expressions and domain-specific terminologies, will significantly improve the chatbot's ability to engage with users across different contexts. Collaboration with native speakers and linguists will help gather more comprehensive datasets, ensuring the chatbot remains relevant and adaptable to various situations.

Thirdly, advancing the preprocessing techniques should be another focus of future research. While the current study achieved substantial progress in normalizing and tokenizing Amharic text, future work could explore more advanced methods such as deep learning-based language models like BERT (Bidirectional Encoder Representations from Transformers) or GPT (Generative Pre-trained Transformer). These models have proven effective in other languages and can better handle the complexities of Amharic. Enhancing the stop word removal process by developing a more extensive and context-sensitive stop word list can also contribute to refining text input and improving model accuracy.

Finally, future research should aim to expand the scope of user acceptance testing and system evaluation. A larger and more diverse user base would provide a broader range of feedback on the chatbot's performance across different demographics and use cases. Integrating feedback loops where users can directly report errors or suggest improvements via the chatbot interface will ensure continuous improvement. Additionally, implementing adaptive learning mechanisms

that allow the chatbot to evolve based on user interactions will make the system more dynamic and responsive. By addressing these considerations, future work can significantly build upon this study to create a more robust, accurate, and user-friendly Amharic chatbot system.

REFERENCES

A. Ram et al., . (2018). Conversational AI: The science behind the Alexa prize. *arXiv*, 1–18.

Acomb, K., Bloom, J., & Dayanidhi, K. (2007). Bridging the Gap: Academic and Industrial research in Dialog technologies. *Work. Assoc. Comput. Linguist.* http://dl.acm.org/citation.cfm?id=1556332.

Bellazzi, R., & Zupan, B. (2008). Predictive data mining in clinical medicine: Current issues and guidelines - Review. *International Journal of Medical Informatics*, 81–97.

Berndtsson, M., Hansson, J., Olsson, B., & Lundell, B. (2008). *Thesis Projects: A Guide for Students in Computer Science and Information Systems* (2nd Edition ed.). Springer.

Bishop, C. (1995). *Neural Networks for Pattern Recognition.* Oxford: Oxford University Press.

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

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

Claire, C., Natasha, M., & Jessica. (2017). *Machine learning:the power and promise of computers that learn by example.* London.

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

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

Dawit, B. (2003). *The Development and Dissemination of Ethiopic Standards and Software Localization for Ethiopia.* Addis Ababa: The ICT Capacity Building Programme of the Capacity Building Ministry of the FDRE and United Nations Economic Commission for Africa.

F. B. Politecnico et al. (2018, May). 'Hey Siri, do you understand me?': Virtual Assistants and Dysarthria. *The Dog gateway View project Ambient Intelligence, https://www.researchgate.net/publication/325466714*.

Feitelson, D. G. (2006, December). *Experimental Computer Science: The Need for a Cultural Change.* School of Computer Science and Engineering, The Hebrew University of Jerusalem, 91904 Jerusalem, Israel.

Fekadu, E. (2021). *Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services.* Msc. Thesis, Adama Science and Technology University, The Department of Computer Science and Engineering.

Følstad, A., Skjuve, M., & Brandtzaeg, P. (2019). *Different chatbots for different purposes: Towards a typology of chatbots to understand interaction design.* Lect. Notes Comput. Sci., doi: 10.1007/978-3-030-17705-8_13.

Fulio, Z., Torez, D., & Gusholo. (2018). *Voicebot and chatbot design Flexible conversational interfaces.* Packt.

George Tzanis, Ioannis Katakis, Ioannis Partalas, Ioannis Vlahavas. (2016). Modern Applications of Machine Learning. 1-10.

Haan, H. d. (2018). *Chatbot Personality and Customer Satisfaction.* Bachelor Thesis.

Habitamu, A. (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.

Hassen, R., Tessema, M., & Solomon, A. (2009). Search Engine for Amharic Web Content. *IEEE AFRICON*.

Hill, J., Randolph, W., & Farreras, I. (2015). Real conversations with artificial intelligence: A comparison between human-human online conversations and human-chatbot conversations. *Comput. Human Behav., 49*(doi: 10.1016/j.chb.2), 245–250.

Himanot, M. (2014). *Semantic compression for enhancing the efficiency of Amharic text retrieval.* MSc Thesis, University of Gondar, Gondar, Ethiopia.

Hoy, M. (2018). Alexa, Siri, Cortana, and More: An Introduction to Voice Assistant. *Med. Ref. Serv. Q., vol. 37, no. 1*, 81–88, doi: 10.1080/02763869.2018.1404391.

Karlin, J., & Alexander, S. (1962). Communication Between Man and Machine. *IRE, vol. 50, no. 5*, pp. 1124–1128, doi: 10.1109/JRPROC.1962.288017.

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

Klopfenstein, L., Delpriori, S., & Malatini, S. (2017). The rise of bots: A survey of conversational interfaces, patterns, and paradigms. *2017 ACM Conf. Des. Interact. Syst.* (pp. 555–565, doi: 10.1145/3064663.3064672). DIS.

L. T. Car et al. (2020). Conversational agents in health care: Scoping review and conceptual analysis. *Journal of Medical Internet Research, 22, no. 8, doi: 10.2196/17158.*

Nimavat, K., & Champaneria, T. (2017). Chatbots: An Overview Types, Architecture, Tools and Future Possibilities. *Int. J. Sci. Res. Dev., vol. 5, no. 7,* 1019–1026.

Sandbank, T., Shmueli-Scheuer, M., Herzig, J., & Kon, D. (2018). Detecting egregious conversations between customers and virtual agents. *NAACL HLT Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, (pp. 1802–1811, doi: 10.18653/v1/n18-1163).

Segni, B. (2021). *Developing Afaan Oromoo Text based Chatbot for Enhancing Maize Productivity.* MSc. thesis, Adama Science and Technology University, Adama, Ethiopia.

Shaik, D., Mohanty, V., & Selvanambi, R. (2020). Machine learning in healthcare. *Comput. Intell. Mach. Learn. Healthc. Informatics*, 278–307. doi: 10.1515/9783110648195-01.

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

Simeone, O. (2017). A Very Brief Introduction to Machine Learning:With Applications to Communication Systems. 1-20.

Tewodros, H. (2003). *Amharic Text Retrieval: An Experiment Using Latent Semantic Indexing Indexing (LSI) with Singular Value Decomposition (SVD).* MSc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia.

UNIDO. (2021). *TRAINING INSTITUTE FOR COMMERCIAL VEHICLE DRIVERS IN ETHIOPIA.* Austria, Vienna: United Nations Industrial Development Organization.

United Nations. (2020). *Road Safety Performance Review Ethiopia.* United Nations publication issued by the Economic Commission for Europe (ECE) available at http://creativecommons.org/licenses/by/3.0/igo.

Vaira, L. et al. (2018). MamaBot: a System based on ML and NLP for supporting Women and Families during Pregnancy. *the 22nd International Database Engineering & Applications Symposium*, pp. 273-277.

Verlag, H. (2003). Amharic Literature. In *Encyclopedia Aethiopica.* Wiesbaden, Germany.

Xufei, H. (2021). *CHATBOT: DESIGN, ARCHITECUTRE, AND APPLICATIONS.* Senior Capstone Thesis, University of Pennsylvania.

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

APPENDIX
User Acceptance Testing Questionnaire

Dear Respondents,

Thank you for participating in this user acceptance testing of the proposed Amharic chatbot system designed to assist driver's license trainees. Your feedback is crucial for evaluating and improving the system. Please answer the following questions based on your experience with the chatbot. Your responses will be kept confidential and used solely for research purposes.

Warm regards,

Adanu Amenu Yigezu

Instructions:

For each statement, please indicate your level of agreement by selecting one of the following options:

1. Strongly Disagree
2. Disagree
3. Neutral
4. Agree
5. Strongly Agree

Section 1: Ease of Use

1. The chatbot interface is user-friendly and easy to navigate.
2. It is easy to understand how to interact with the chatbot.
3. I did not require any help to use the chatbot effectively.
4. The chatbot’s responses are presented in a clear and understandable manner.

5. The instructions provided by the chatbot were easy to follow.

Section 2: Time Efficiency

6. The chatbot responded to my queries quickly.

7. The time it took to get a response from the chatbot was reasonable.

8. The chatbot helped me complete tasks faster than I expected.

9. I was able to find the information I needed without unnecessary delay.

10. The chatbot's interaction speed met my expectations.

Section 3: Operational Efficiency

11. The chatbot efficiently guided me through the steps necessary to complete tasks.

12. The system worked smoothly without any significant interruptions or errors.

13. The chatbot provided helpful suggestions during the interaction.

14. The chatbot's operational performance met my expectations.

15. I was satisfied with how the chatbot handled my requests.

Section 4: Accuracy

16. The chatbot provided accurate responses to my questions.

17. The chatbot correctly understood my inputs.

18. The chatbot’s responses were relevant to the context of my queries.

19. The accuracy of the chatbot's responses met my expectations.

20. I was confident in the information provided by the chatbot.

Section 5: Completeness

21. The chatbot provided complete answers to my questions.

22. The information given by the chatbot was comprehensive.

23. I did not need to ask follow-up questions to clarify the chatbot’s responses.

24. The chatbot covered all the necessary aspects of the topics I inquired about.

25. The responses from the chatbot were thorough and detailed.

Section 6: Utility

26. The chatbot is a useful tool for driver's license trainees.

27. The chatbot’s functionality is relevant to my needs as a trainee.

28. The information provided by the chatbot is valuable for my learning process.

29. I would use this chatbot regularly if it were available to me.

30. The chatbot adds value to my driver's license training experience.

Section 7: Learning Capability

31. The chatbot helped me learn new information about the driver's license process.

32. Interacting with the chatbot improved my understanding of the content.

33. The chatbot's responses were educational and informative.

34. I feel more prepared for the driver's license test after using the chatbot.

35. The chatbot's ability to teach me new concepts met my expectations.

Thank you for your participation!