APPRENTICE BOT MODEL DESIGN AND IMPLEMENTATION FOR PSYCHOLOGICAL CLIENTS' THERAPY

Kibrom Gidey Kidanemariam

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirements for the Degree of Master of science in Computer Science

Office of Graduate Studies

Adama Science and Technology University

September 2022

Adama, Ethiopia

APPRENTICE BOT MODEL DESIGN AND IMPLEMENTATION FOR PSYCHOLOGICAL CLIENTS` THERAPY

Kibrom Gidey Kidanemariam

Advisor- Hailay Beyene (Ph.D.)

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical and Computing

Presented in partial fulfillment of the Requirements for the Degree of Master of science in Computer Science

Office of Graduate Studies

Adama Science and Technology University

September 2022

Adama, Ethiopia

DECLARATION

I, the undersigned, declare that the thesis entitled "**Apprentice Bot Model Design and Implementation for Psychological Clients' Therapy**" is my original work. In compliance with internationally accepted practices, I have acknowledged and refereed all materials used in this work. I understand that non-adherence to the principles of academic honesty and integrity, misrepresentation, or fabrication of any idea/data/fact/source will constitute sufficient ground for disciplinary action by the University and can also evoke penal action from the sources which have not been properly cited or acknowledged.

Kibrom Gidey Kidanemariam	_____:	_____
Name	Signature:	Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Apprentice Bot Model Design and Implementation for Psychological Clients' Therapy**" prepared under my guidance by Kibrom Gidey Kidanemariam submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Hailay Beyene (Ph.D.)	_____	_____
Advisor	Signature	Date

APPROVAL PAGE

I, the advisors of the thesis entitled "**Apprentice Bot Model Design and Implementation for Psychological Clients' Therapy**" and developed by Kibrom Gidey Kidanemariam, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

_____	_____	_____
Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Kibrom Gidey have read and evaluated the thesis entitled "**Apprentice Bot Model Design and Implementation for Psychological Clients' Therapy"** and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in computer science.

_____	_____	_____
Chair Person	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date

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

_____	_____	_____
Chair Person	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	. Date

Dedicated

Towards everyone

Who has backed me up throughout my contests!

ACKNOWLEDGEMENT

First and foremost, I want to thank **G**od, called the Father, the Son, and the Holy Spirit, for providing me with the strength and patience I needed to overcome all of the difficulties and obstacles I encountered throughout these days. And I'd like to thank **S**aint **V**irgin **M**arry, **G**od's mother, who is always concerned about her children's well-being, especially mine.

Then after, I'd like to express my gratitude to my beloved family, as well as my love Malle, for their morals and motivations. I'd like to convey my heartfelt gratitude to my thesis advisor, Dr. Hailay Beyene, for his imperative advice, comments, and recommendations throughout the thesis.

Furthermore, I would like to express my gratitude to Mr. Kahsay (Department of Psychology, Mekelle University) for his professional assistance in the completion of this thesis, particularly for providing me with relevant psychology documents and working ideas, as well as a member of the department of psychology at Mekelle University.

Finally, I want to express my gratitude to my adored friends and coworkers, who were a huge help to me when I was working on this thesis and during my work-life confrontations.

TABLE OF CONTENTS

Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AGI	Artificial General Intelligence
AI	Artificial Intelligence
API	Application Interface
AIML	Artificial Intelligence Markup Language
ALICE	Artificial Linguistic Internet Computer Entity
ANN	Artificial Neural Network
BoW	Bag of Words
CAI	Conversational Artificial Intelligence
CV	Cross-validation
DICT	Dictionary
DS	Dialog Systems
DSM – 5	Diagnostic and Statistical Manual of Mental Disorders 5th edition
EEMeN	End-to-End Memory Networks
ELIZA	Elisabeth
EoS	End of Statement
FFNN	Feed Forward Neural Network
GRU	Gated Recurrent Unit
JSON	Java Script Object Notation
LSTM	Long Short-Term Memory
MemN2N	Memory Network to Network
MoH	Ministry of Health
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
NN	Neural Network
Psycho	Psychology
QA	Question Answer
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RQ	Research Question
WHO	World Health Organization
WORD2VEC	Word to Vector

ABSTRACT

*In terms of accessibility and reliability, conversational agents employed for advising play an essential role. Conversational bots are not straightforward since different domains may require different algorithms. Extracting information on cases of mental illness, such as consulting clients, the disorder causes treatment and recommendations is challenging in a contemporary mental disorder-related document. Currently, Ethiopia is suffering from a high prevalence of mental health among other developing countries in the world due to poor mental healthcare provision. An apprentice AI text-based chatbot that can consult commonly reported mental disorder cases in Ethiopia for users of the Amharic language has been developed as part of this study. For consulting purposes, this task-oriented bot recognizes features such as general knowledge queries, what to do if a mental disorder happens or is likely to happen, auto-generated consulting advice, and client stories. To construct the bot, the syntactic and semantic structure of data was investigated, as well as a user chat's memory network. To find the best-fitting neural network for the model, four neural networks were experimented with, and existing models were compared. The approach uses word embedding for the semantic extraction of Amharic words and a seq2seq model for suitable response generation. In this thesis, an ensemble architecture of both generative-based and retrieval-based approaches has been employed so that it creates new replies to handle and track user dialogue. The scruffy technique is used to generate word embedding, which is then used to extract semantically comparable terms, to construct unique Amharic word2vec from **DSM**-5 manual book, articles, and discussion forums obtained from websites. The model has also embedded custom rules for smoothness and error handling during the conversation. The performance and outcomes of the psychological chatbot were demonstrated using perplexity, accuracy measurement, f-1 score metrics, and human-evaluation methods. The model used different model optimization techniques to improve its accuracy. The model's accuracy was 79.62% likelihood of delivering relevant responses at the time of testing, and the model's accuracy was reported as a performance metric. The results demonstrate that for user utterances, the ensemble architecture based on seq2seq modeling with embedded custom rules and Amharic word vector implementation provides pertinent responses.*

Keywords: *Chatbot, Mental disorder, Word embedding, Sequence to sequence (Seq2Seq), Recurrent Neural Network*

CHAPTER ONE

INTRODUCTION

The fundamental appearance and background of distinct conversational bots, as well as the thematic nature of data beyond the topic presented in this thesis, are described in this chapter. Background of various but related works is discussed, as well as present challenges, reasons for this effort, a solution planned to solve the problem, and predicted results.

1.1 Conversational Systems their Background

In conversational systems, artificial intelligence (AI) plays a significant role. It replaces humans in conversation and attempts to retrieve the appropriate responses. Conversational Artificial intelligence is an intelligence system that supports communication in terms of chat using a natural language interface. Conversational AI is gaining popularity because it facilitates human-computer interaction (Csaky, 2019). A conversational Bot can work as a virtual assistant for people who need assistance, and the dialogue can be spoken or written. The spoken-based AI bot is based on a user's speech utterance, which is processed by the AI bot and returned in the form of voice, whereas the text-based AI bot is based on a chat interface that uses natural language. CAI can be used in a variety of business situations. To automate their complete work process, companies need to conduct a study on model-building strategies. Many businesses benefit from designing and deploying their own CAI to customers. As a result, since the AI era, the conversational interface has become a researchable area, and users' need for virtual support is considerable.

We can accomplish autonomous human-computer conversation by combining big data and deep learning approaches. Task-oriented and non-task-oriented conversation systems can be broadly classified based on their task implementation (Ilievski et al., 2018; Schlicht, 2016; Weston et al., 2015). A task-oriented conversational system is goal-oriented and centered on a certain intended action. Conversational artificial intelligence (CAI) knows only one domain and is uninterested in the rest of the knowledge. Goal-oriented dialog systems are ones in which many systems collaborate to achieve a certain goal by the end of the conversation. Open-domain conversational systems are non-task-centered conversational systems that attempt to understand world knowledge and respond to human utterances appropriately. It's difficult to train the system with all of the available knowledge, and require a lot of data.

Text-based conversational bots like; Telegram, Messenger, and Slack and voice-based virtual assistants like; Cortana, Siri, and Alexa are two important channels in conversational bots (Batish, 2018). These conversational bots automate conversations based on the data they have. Distinct implementation regions, which are merged with a single environment and a different environment, are also depicted. Conversational bots are implemented in a single context, whereas integrated bots are implemented in conjunction with other systems, such as Telegram bots (ZEMČÍK, 2019). Conversational systems can be developed with three design approaches types. Rule based, retrieval based and generative based. The responses of retrieval-based conversational bots come from prepared answers, which is why they are grammatically correct. According to (Jwala et al., 2019), the production strategy for most customers' task-oriented implementation of conversational agents falls into this category.

According to (Song et al., 2018), if the new information is not part of the dataset with predefined replies, the retrieval-based conversational agent will not be able to see it. Generated-based conversational agents, on the other hand, are built to learn by forecasting new knowledge. The state of generated-based CAI on its own, according to (Bhagwat, 2018), performs poorly in terms of the naturalness of responses. It can vary the reaction to input utterances repeatedly, and the replies are unrelated. This occurs as a result of the generative-based strategy's minimal data features.

This thesis is based on a task-oriented conversational system that gathers and uses the DSM-5 Diagnostic and Statistical Manual of Mental Disorders 5th edition, as well as other related and relevant documents that are publicly available on the web, and feeds the information to a model that processes questions about the data learned. The model takes knowledge from the DSM-5 and other relevant and related papers, and it is very task-oriented. Ultimately, the bot model uses it to retrieve pertinent information, and the communication channel is built based on the text. Using the prepared psychobot dataset, this study attempted to uncover the accuracy of the generative approach without assembling it with the retrieval approach to explore the results and demonstrate the contribution of the retrieval approach to the model. Using retrieval-based or generated-based methodologies to develop conversational bots could not be satisfactory because each has its own set of flaws, as previously indicated. The study has sought to absorb their merits and take the best solution to follow the optimization principle, which employs ensemble methods and combines the best aspects of both approaches (Sheikh et al., 2019).

Recurrent neural networks are used to keep track of data sequences (Bianchi et al., 2017). Its variants, LSTM and GRU, do away with the vanishing gradient problem, allowing them to follow input and output sequences (Salehinejad et al., 2017). The model is built using sequence-to-sequence modeling and an ensemble method that takes into account various types of models, such as retrieval-based models with a syntactic and semantic data structure. In this thesis, the study uses many strategies and methods to get the best answer. The AI psycho bot model is built using the DSM-5, as well as other related and relevant documents as a repository of knowledge, and the model is considered a specialist in the most commonly observed psychological mental disorder cases in Ethiopia.

Conversational bots are designed to mimic human-like dialogue and communication to complete certain activities such as restaurant reservations, shopping requests, health services, legal consultations, and health advisories among others (Klopfenstein et al., 2017). They can be utilized in a variety of activities and for a variety of objectives, such as booking, leisure, advertising, transportation and utilities, sports, healthcare, and legal assistance, and they provide an easy communication channel (Følstad et al., 2019).

As a result, various conversational systems have been developed, as detailed below. ELIZA is a slightly older natural language-based artificial intelligence chatter system that has been developed at the MIT artificial intelligent group since 1964 (ZEMČÍK, 2019). It is based on pattern matching of certain keywords and responds with an open-ended answer based upon these words. Joseph Weizenbaum's earliest conversational agent, Eliza, can be regarded as his earliest conversational agent. And the discussions are tracked by the user's speech, with the system attempting to work on certain keywords and generating a response depending on the keyword shared by the system and the user's speech (Weizenbaum, 1966). Another conversational system called ALICE uses a subset of Extensible Markup Language called AIML to store learning (XML). Richard Wallace developed ALICE, a robot or artificially intelligent machine. It is a robot or artificially intelligent device that can engage in discourse on open domain knowledge. (Bruno Marietto et al., 2013). ALICE employs natural language processing and, unlike ELIZA, can handle more complex interactions such as pattern building from various types of data and asking the user for an explanation of material that ELIZA does not understand. “jabberwocky” is a conversational AI system designed by Rollo Carpenter to mimic genuine talks using open knowledge base (Lim & Goh, 2016).

The Amazon Alexa Prize, worth approximately 2.5 million dollars for twenty minutes (Ram et al., 2018), is designed to open up conversation and connect with people on hot topics like

sporting events, political hot topics, people's enjoyment, fashion trends, and technology while the study examines the current conversational AI. Another virtual personal assistant available on many phones is Siri, which is made by Apple. (*[PDF] "Hey Siri, Do You Understand Me?": Virtual Assistants and Dysarthria | Semantic Scholar*, n.d.). Google Assistant (for Android phones), Dragon Mobile Assistant (it's tough to call it a conversational bot, although it can provide important and useful weather data) (Hoy, 2018), and a smart voice assistant that can control the Android interface with voice commands.

Conversational AI is commonly used to analyze conversations using natural language; however, conversational bots can be built using deep learning, statistical models, or a combination of the two (Nimavat & Champaneria, 2017) even though distinct conversational systems have their flaws, such as conversation monitoring and managing proper answers for user utterances, they are meant to generate a simple interface between a real person and an artificial agent to provide communication solutions on difficulties that the person actively has.

Because each conversational agent must be mapped with life experience or constrained task knowledge, conversational bots are categorized as task-oriented (Siri, Alexa, Cortana, etc.) or non-task-oriented (Microsoft Tay conversational bot). The utilization of natural language dialogue and simulating human interaction is a typical feature of chatty bots.

1.2 SWOT Analysis on Chatbots in Mental Healthcare

An analysis of the usage and limitations of chatbots for mental healthcare is undertaken in a studies conducted in (Sweeney et al., 2021) that briefly discusses the need for chatbots. Chatbots are thought to help remove some barriers to mental health care, such as the shame associated with seeking out psychological treatment and geographic remoteness, which can make it difficult for people to attend in-person therapy sessions (Miner et al., 2017). Accessing mental health services may be difficult for those who work unusual shifts. Given that chatbots are non-intrusive and simple for anyone with a mobile phone to use, they could be employed as a potential solution to these issues. Additionally, there is some evidence to support the use of chatbots in the treatment of those who have major depressive disorder (Vaidyam et al., 2019).

Chatbots can be used to improve mental health counseling since they can give users quick information (Cameron et al., 2017), are available around-the-clock, and will save users money on things like travel and phone costs (Kowatsch et al., 2017). Users also like the anonymity that chatbots provide since, as indicated by Lucas et al., some patients have been seen to give

sensitive material to a chatbot even though they wouldn't have done so with a human therapist (Lucas et al., 2017).

According to a recent study, patients are sometimes more ready to provide sensitive information to conversational assistants powered by artificial intelligence than to a real therapist (Miner et al., 2017). Many young people believe that their issues are too personal, or they worry that others might learn about this sensitive information (West, 1991). As reported in (Sweeney et al., 2021), 77% of professionals and experts in mental health believed that chatbots are either somewhat (66%) or very significant (11%) when asked about the perceived importance of chatbots. Conversely, 23% thought chatbots were either very unimportant (3%), or only marginally important (20%).

The conversational agents in the other scenario are restricted, making it impossible for them to communicate the core mental health issue by giving an inappropriate response or by responding in a way that is ambiguous or absurd. To avoid any misunderstanding, chatbots should periodically remind users that they are "artificial intelligences." Many people are terrified and appalled by the idea of using chatbots to prevent suicide, especially in the field of mental health. They claim that such a complicated issue cannot possibly be solved by a computer program (Vinet & Zhedanov, 2011). Additionally, chatbots fall short in providing mental health support for delicate subjects like suicide risk and abuse reporting (Skjuve & Brandtzæg, 2018). In fact, there is little scientific evidence to justify the use of chatbots in suicide prevention (Sweeney et al., 2021)

1.3 The Motivation of the Study

Working on computers that act like humans is fascinating, thus the age of artificial intelligence, particularly in conversational bots, can be admirable. Various tactics in developing conversational bots can be found by studying literature (Valério et al., 2017). The first discovery is that speaking with bots as if they were humans is the most engaging component of the conversational approach. Conversational bots can be employed as virtual assistants who can help with work by increasing automation (Mhatre et al., 2016). The next goal is to define the problems and solutions and to choose design principles. This work has been labeled as a mental health-based psychological conversational chatbot for clients' therapy.

Another motivator is the creation of natural dialogues based on task selection, as the primary goal of conversational bots is to mimic human discourse while also assisting humans with chores (Hill et al., 2015). Today, feeding machine learning with data to perform some tasks is

the best solution to many problems. Combining machine learning with natural language processing enables us to develop a chatbot model that provides a better understanding and consulting for psychological clients.

Amharic is the second most commonly spoken mother tongue in Ethiopia (after Oromo), but the most widely spoken in terms of total speakers. It is also the second-most commonly spoken Semitic language in the world after Arabic (Wikipedia, 2017). However, no research has been conducted on chatbots for psychological disorder consultancy in the Amharic language. The use of chatbot technology in mental health will aid clients in understanding their current state of health. Additionally, it reduces the workload placed on hospitals, mental disorder re-habitation facilities, and health hubs when there are numerous clients to serve. One study (Fadhil, 2018) showed that the majority of people, including elders, spend the majority of their time on messaging apps like Telegram, WhatsApp, Facebook, etc. These facts motivate us to create an Amharic chatbot model that offers mental health counseling services. The goal of this study is to minimize the problems associated with mental health by using a chatbot model to give clients with psychological disorders better advice and consulting methods.

1.4 Problem Statement

We live in a complex, fast-moving, and interconnected world in which the importance of physical, mental, and social wellbeing has not been well emphasized. In this interwoven situation of life, one could understand that mental health plays a vital role in the well-being of individuals, the community, countries, and the entire world.

Currently, mental health professionals like psychologists and psychiatrists, who are the main source of mental health services in Ethiopia, are centered and limited in the office room and active office time only. To obtain information and a diagnosis regarding their symptoms, clients should visit these mental health professionals in the office. In addition, due to privacy concerns and lack of knowledge, clients are reluctant to talk about how they are feeling even while they are being counseled (Yitbarek et al., 2021).

About 25 million Ethiopians are thought to suffer from mental health disorders, but less than 10% of them are believed to receive treatment of any kind, and less than 1% are believed to receive specialized care (Alem et al., 2009), Which is saddening because it has a severe shortage of mental health experts. There are only 63 psychiatrists in the country, which equates to 0.65 psychiatrists per million people. Meanwhile, Since the majority of psychiatrists are

located in major cities, there is a treatment gap because a large portion of Ethiopians, more than 80% of the country's population, live in rural areas and cannot access mental health services (Ayano, 2016). The study also noted that the mental health care system in the community and the unequal distribution of mental health resources, issues with accessing services in remote areas, affordability, and social acceptability concerning ignorance and belief systems are some of the barriers to mental health care services in our communities, The result of all these obstacles is an increase in the number of people living on the streets with mental health disorders, which is a serious social issue that requires immediate attention.

In Ethiopia, there are few mental health consultancy service providers[1] and among them, Amanuel Specialized mental health hospital is one of the governmental service providers. Amanuel mental specialized hospital provides treatment for mental illnesses and related problems and mental health rehabilitation services[2]. According to a study conducted in Amanuel specialized mental health hospital, the clients experience the following mental health disorder types with their percentages. substance-use disorder (28.9%), alcohol-use disorder 36.6%, and drug-use disorder 53.1%(Duko, 2016). According to the noted study; 9.9% of the population had a lifetime diagnosis of major depression, 19.9% had a drug-use disorder, 24.3% of this population had an alcohol-use disorder and/or major depressive disorder and 16.2% of whom had both major depressive and alcohol use disorders, 33.7% of people with a diagnosis of schizophrenia, 38% of persons with bipolar I disorder and 19% of those with bipolar II disorder.

According to other reports, 18% of adults and 15% of children in Ethiopia suffer from mental health disorders (Sathiyasusuman, 2011). Show some of the sub-population experience up to 29% depression(Bifftu et al., 2018), 28% anxiety(Mekuria et al., 2017), 40% psychological distress(Dachew et al., 2015) as well as the combination of one or more of these problems(Tegegne et al., 2015; Tesfaw et al., 2016; Tsegabrhan et al., 2014). People should therefore have access to mental health services nearby or at their fingertips to help reduce this situation and maintain daily activities, as well as to prevent the indirect costs associated with seeking specialized care in far-off places. However, due to travel time and service fees, clients now spend time and money looking for psychology consulting services. Even though some television shows and newspaper articles are devoted to providing information on some mental health issues, they cannot include every instance of mental disorder. Some users also prefer to

[1] *https://www.opencounseling.com/*
[2] *https://ethiopia.worldplaces.me/view-place/67492192-amanuel-mental-specialized-hospital.html*

use search engines, but with so much information available on google, it can be challenging to find what you are directly looking for.

The study observed that language barriers in healthcare cause miscommunication between medical professionals and patients, lowering both parties' satisfaction as well as the quality of healthcare delivery and patient safety (Al Shamsi et al., 2020). Thus, utilizing a native language increases healthcare quality and satisfaction among both medical providers and patients. Since the audience of the study are Amharic language users, the model learned and interact with the Amharic language.

To the best of the authors' knowledge, there is no conversational AI bot model that is reasoning user query on the most commonly observed psychological disorder cases in Ethiopia, neither task-oriented nor non-task-oriented.

Therefore, computing may be used to meet the need for healthcare in conversation with a conversational bot who can be consulted in the Amharic language and to keep track of one's mental health in Ethiopia.

1.5 Research Questions

The following research questions are addressed in this work.

RQ1. Which deep learning algorithm is more suitable to design an ensemble consultant chatbot model for psychological client therapy?

RQ2. Does the Amharic custom rule embedded in the proposed model affect the conversation flaw?

RQ3. Does the inclusion of the Amharic word2vec model change how accurate and pertinent the proposed model is?

1.6 Objective of the Study

1.6.1. General Objective

The general objective of this study is to model and implement an apprentice intelligent chatbot that has reasoning capability for the client's Amharic utterance.

1.6.2. Specific Objectives

To attain the general objective of the study, the following specific objectives has attempted. To:

- Identify the current approaches used for modern chatbot development and find their gaps.

- Create a word2vec model for Amharic semantic analysis.
- Prepare and embed Amharic custom rules to the proposed model.
- Design and develop the ensembled architectural model for the proposed model
- Compare and contrast the best fitting model with other developed existing models.
- Evaluate the performance of the proposed model for the development of a mental health consultant chatbot using evaluation metrics.

1.7 Scope and Limitation of the Study

The most frequent mental psychological disorder types observed in Ethiopia (Kassa & Abajobir, 2018) are covered in this thesis. Bi-polar disorder / የባይፖላር ዲስኦርደር ወይም ጥንድ ዋልታ መታወክ, (Trauma, anxiety and Stressor-Related Disorders / የጭንቀት መታወክ), (Dissociative disorders / የመገለል ስሜት መታወክ), (Sleep-Wake Disorders / የእንቅልፍ መዛባት መታወክ), (Conduct Disorders / የስነምግባር መታወክ), (Depressive Disorders / የድብርት መታወክ), (Substance-Related and Addictive Disorders / የአደንዛዥ ዕፅ አጠቃቀም መታወክ), and (Schizophrenia disorder / የመዘንጋት መታወክ). However, it excludes other types of mental disorders and knowledge found throughout the world. The working language of the model is Amharic and is focused on a deep learning-based chatbot model. Since the corpus preparation is done by hand, relevant patterns may be missed.

This model uses the DSM - 5, as well as other related and relevant documents as a source, and combines sequence-to-sequence modeling by applying custom rules to the data and the Amharic Word2Vec model to bridge the gap in conversation flexibility. As a backend, it employs RNN variants. Emotional dialogues, emojis, and Abbreviations aren't taken into account, though the model can sometimes extract certain abbreviations.

1.8 Contribution of the Study

- A new dataset prepared via hand crafted in Q and A format from the DSM-5 book set for users, researchers, and developers of NLP tasks.
- The study proposed a deep learning approach for developing an Amharic text-based Chatbot for mental health consultancy by using RNN based on the Seq2Seq model approach.
- New Amharic word2vecmodel data for mental health consulting has been prepared which is used for semantic analysis during the conversation.

1.9 Significance and Beneficiaries of the Study

It's very natural to be concerned about one's health. People have a lot of questions about health issues, especially mental health, daily, both large and trivial. It would be especially helpful if

they had access to a psychological specialist who could offer advice and therapy for their mental illness. As a result, creating a conversational chatbot would be extremely helpful. As a result, the study will have the following benefits to the clients, health institutions, and researchers.

Clients: Today, every person wants a piece of consult on how to treat and protect themselves from a mental disorder. A chatbot can be considered the best solution to offer consulting services, awareness, and consult, to all users simply and efficiently. It also provides first support through one-on-one conversations. Generally, it decreases mental healthcare costs and time for users and improves ease of access to mental healthcare services.

Health institutions: can use mental health consultant bots to develop their interfaces based on the properties of their clients, such as adding a psycho bot to their websites, social media sites, and more.

Researchers: get the other half of the pie. They can reduce the challenge of developing conversational bots, particularly in task-oriented text-based bot models, and modify the best model by looking into the study's limitations. Considering this work provides insight into new design strategies. It can also be a motivating task for researchers, particularly in Ethiopia, if they adapt it depending on their findings; for example, mental health researchers can determine the significance of conversational bots in mental health firms by examining the relevance of this model.

1.10 Thesis Organization

The rest of the thesis was organized in the following way. The essential building blocks of the conversational bots were dug out in **Chapter 2**. After all, in **Chapter 3**, the research design, data preprocessing, feature extraction, activation function, model overfit handling, hyperparameter tuning, and model evaluation techniques are discussed. In **Chapter 4** of the study, the design and experiment of the proposed model have been investigated. Each implementation flow of the experiment such as data preprocessing implementations, feature extraction implementations, selection of the RNN neural networks variants, and word2vec generation has conferred well. C**hapter 5**, is the result and discussion unit. Testing the result of the model through different evaluation metrics has been explored. Different activation functions and model optimization techniques have been employed. And after human evaluation, a comparison of the model with the different existing models has done. In the last chapter, **chapter 6,** proposed research findings described and concludes the study, and last gave bearings to conceivable future work.

CHAPTER TWO

LITERATURE REVIEW AND RELATED WORKS

This study's chapter explores and discusses the overview of various paperwork and finds research gaps, which aids in the identification of research agendas. It also simply describes particular subjects and their relevance, tools and methodologies, and associated topics.

2.1 Overview of Conversational Agents and their Technology

Chatbots are data-driven artificial intelligence models that can communicate with humans and carry on conversations in their place (Hill et al., 2015). Deep learning techniques are now the state of the art in the implementation of conversational bots (Bhagwat, 2018). They are based on dynamic memory networks, Sequence to Sequence modeling, Deep Convolutional neural networks, or Recurrent Neural Networks and can be retrieval-based or generative-based techniques (Dhyani & Kumar, 2019).

Conversational bots, particularly chatbots, were named as a new application because they were given a lot of tasks. In 2016, there was a big movement with conversational bots, which has faded for the time being but is still going on in the background (Brandtzaeg & Følstad, 2017).

Looking into the literature, the study finds that there are various sorts of conversational AI bots with various implementation methodologies (A. & John, 2015). Metrics like the BLeU score, Turing test, intent classifier accuracy, and perplexity are used to assess reasoning abilities (A. & John, 2015). The implementation of conversational bots differs depending on the task they are completing. For the successful generation of a response, the knowledge management approach is frequently coordinated with many nodes. In the following sections, we'll look at the concepts and features that underpin various conversational AI implementations.

Based on the domain, conversational AI can be split into two categories: open domain and closed domain (Adiwardana et al., 2020). Artificial general intelligence (AGI) models are open-domain conversational AI systems that can react to a wide range of topics, such as general knowledge (AGI). For example, a conversational AI for Facebook conversations might be open-domain, because the themes of these conversations might range from many different world knowledge that people discuss with one another. A mental health consultant conversational AI, on the other hand, is a closed domain because it only knows about mental health. While open-domain conversational AI can be described as non-task oriented, closed-domain conversational AI can be described as goal-driven, as Venkatesh (Adiwardana et al., 2020) points out.

Another conversational categorization is based on answer generation, which may be divided into two categories: retrieval-based and generative-based conversational models (Lim & Goh, 2016). The difference is in the manner the replies are organized. The CAI retrieves answers from a given dataset in retrieval-based models, and the technique is correct for grammatical structure and there is no doubt about it, as well as the answer is very accurate if the user utterance is matched in some cases, but it is difficult to generate new responses not available in the datasets (Song et al., 2018). On the other hand, the generative-based model CAI is a method for generating new responses from the hidden state, but the grammatical accuracy of the responses is smaller than that of the retrieval-based model (Bhagwat, 2018).

Conversational AI can be classed into two forms based on how communication to the system is created; spoken-based conversational systems and text-based conversational systems (Mhatre et al., 2016). The titles of these conversational systems reveal the differences between them. Conversational bots, in general, are designed to process questions and answers in an interactive manner using natural language (Bhagwat, 2018). Those responses are given by a bot that has learned from datasets such as document files, websites, or a conversation held on various social media forums. The purpose of a conversational system is to extract required and pertinent information from the user's utterance, rather than to respond to all pattern matching like search engines do (Gentsch, 2018).

The conversational model's interaction is based on the user's and system's desires. In a system initiative conversational model, the model asks the user for information so that the conversational bot can pose a question and the client can fill out the response. User-initiated conversational models, on the other hand, are designed to provide services to people by allowing users to ask questions about the model and receive responses from the conversational bot. And, to mimic real-life conversations while also improving the usability of AI bot models, a combination of projects is necessary. When a user requires information, he or she might request a response concurrently, in which the AI bot model requests information from the users and the users respond to the system's request, a process known as mixed initiatives (Klüwer, 2011).

A conversational approach for question answering issues is the Deep Question Answering (QA) framework. It generates responses by taking into account the SoftMax answer to the question and learning the available evidence from the training dataset. It consists of question analysis (chosen to represent and evaluate a user query in conversational AI form), request decomposition to a manageable level, hypothesis generation for queries based on an analysis of the sequence of questions, and soft filtering technique (representing the main affected words

and assigning a score), and final implementation (synthesis to the output form based on the co-occurrence) (Ferrucci et al., 2011). Figure 1 shows the general framework of deep QA.

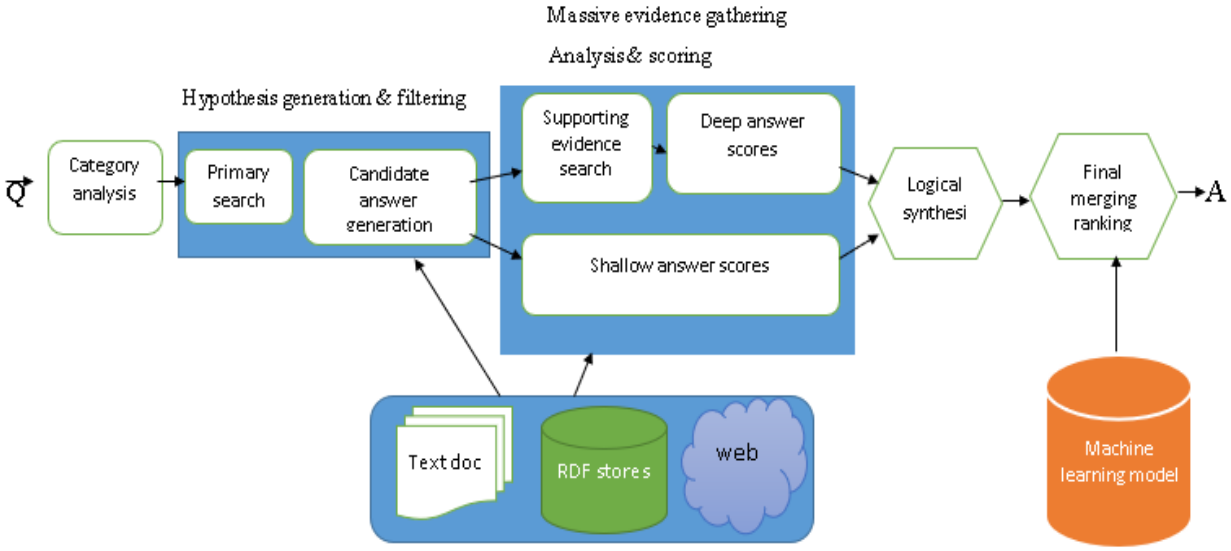


Figure 1 Summary of the DeepQA architecture

The other conversational bot can be built using recurrent neural networks (RNNs). In this approach, the data is used to train a sequence-to-sequence model (Weston et al., 2015). Two long short-term memory (LSTM) networks are used in the sequence-to-sequence paradigm, one for encoding and the other for decoding. As a result, it functions as an encoder and decoder, with the encoder being the method for processing a sequence of data in terms of tokens individually. It creates a hidden state that covers the sense of situational data clarity and is used to generate the answer. The model decoder's goal is to catch the encoder's context production and create an answer from it. To achieve this, the decoder RNN keeps a SoftMax layer along with the vocabulary set. This layer accepts and analyzes the hidden state of the decoder at each time stamp, generating a probability distribution over all words in the vocabulary it has produced.

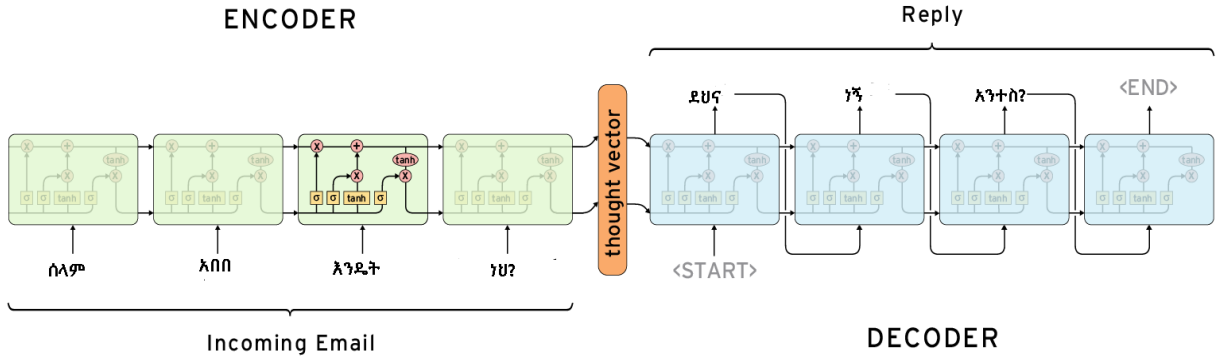


Figure 2 Seq2seq context for modeling dialogs

Memory networks hypothesize with inference segments operate with a long-term memory portion, and memory networks learn how to use these jointly to track the dialogue (Weston et al., 2015). Figure 3 depicts a simple data input function that can be generalized to some memory tracking functions. It works by encoding sentences with a weight function and building an output memory network to predict the outcome of future utterances.

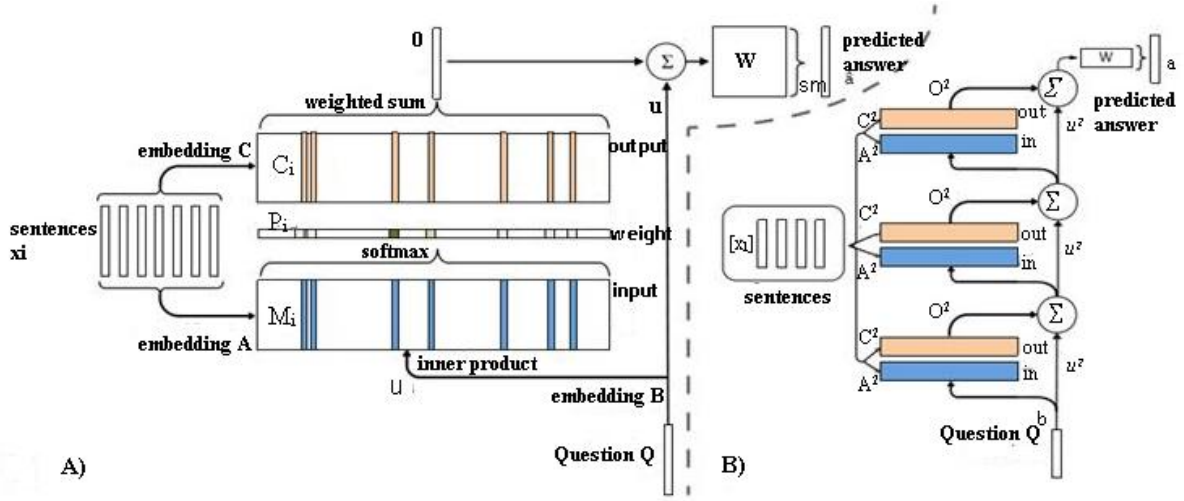


Figure 3 (a) Single-layer memory network model. (b): Three-layer memory network model

figures 3(a) and (b). Let's say given an input set of $x_1...x_n$ is in memory. The full set of xi is turned into memory vectors mi of dimension d, which are generated by embedding each x_i in a continuous space using an embedding matrix A (of size d*V) in the simplest case. To obtain an internal state u, the question q is moreover embedded (within the simplest example, though every other embedding matrix B with the identical dimensions as A). Then compute the match between u. and each memory m_i in the embedding space by taking the inner product followed by a SoftMax:

$$p_i = soft\max(U^T m_i) \tag{2.1}$$

Where $Softmax(pi) = e^{zj} / \sum_j e^{zj}$. Defined in this way, p is a vector inputs probability. And the second is Output memory representation: Each xi has a corresponding output vector ci (given in the simplest case by another embedding matrix C). Then the memory response vector o is then a sum of the transformed inputs ci, weighted by the vector of input probability:

$$o = \sum p_i c_i \tag{2.2}$$

The third part as expressed in figure 3(b) is generating the final prediction by which in the single-layer case, the sum of the output vector o and

$$\hat{a} = Soft\max(W_{(o+w)}) \tag{2.3}$$

2.2 Applications and Implementations of Chatbots

A conversational agent, in general, is a service that is automated using rules and, on rare occasions, artificial intelligence. Clients can use the chat interface to communicate with a bot, much like they would with a real human. Voice converse, text converse, or a mixture of both can be used to communicate with conversational bots. According to (Chatbot, 2020), 55% of customers are very glad to communicate with a company using messaging applications to fix a problem. Conversational AI bots could be used for a variety of purposes, ranging from daily duties to business emergencies. The major goal is to make life easier for communities and to make certain information more accessible. Conversational systems are capable of processing simple acts such as taxi requests as well as more complex activities such as automatically creating legal papers. Chatbots in general works as task coordinator, media outlets, schooling, delivery salesman, legal services agriculture, customer service, healthcare, and others. Let`s see some of them.

2.2.1. Chatbots for Health Care Service

Healthcare is extremely important in our daily lives. AI-based technologies are required in the healthcare industry to provide efficient and effective service. A chatbot model is one such best technology that is very important in healthcare to provide services and health information to users. Chatbots are now being used in healthcare to impersonate health services such as health consulting, disease diagnosis, health advisors, nutrition facts, and so on.

The primary goal of developing a healthcare chatbot model is to assist patients in self-diagnosing their health status and understanding how to prevent diseases. Chatbots can thus play an important role in healthcare by providing prevention, diagnosis, and treatment services. There are several well-known chatbot applications in healthcare.

Numerous preceding studies on AI-based technology for healthcare industries have been presented, with a focus on disease prediction, disease diagnosis, and so on.(Kandpal et al., 2020; Kidwai & Rk, 2020) However, the chatbot can also be used in the healthcare industry for consultation and social and mental health support. When users become ill, they go to a doctor or a nearby hospital to find out what is wrong. According to Google research, one out of every twenty Google searches is about health; this demonstrates the need for treatment and health counseling to be delivered digitally. As a result, chatbots are being used to serve patients around the clock and reshape the healthcare industry by providing preventive, predictive diagnoses and assisting patients.

The study aims to design and implement an ensemble architectural text-based bot model for mental health consultancy using deep learning techniques. The designed chatbot model is to offer better consultancy service for frequently observed mental disorder cases in Ethiopia, by replying with better answers including client history, treatment recommendations, mental health-based general knowledge, etc. This presented study is expected to minimize mental health disorder problems by supporting clients in their mental health problems.

2.2.2. Conversational Agents as Customer Service

Customer service is the most popular application for conversational bots. Conversational bots automate the process of establishing required communication with clients, allowing organizations to save money on customer service by shortening response times, freeing up workers for more complex tasks, and providing relevant answers to up to 80% of extremely specific questions (Xu et al., 2017).

The benefit of using customer service conversational bots for clients is that they get a quick response. It is not necessary to wait a long period; instead, you may speak with the company and make a decision right now. Conversational bot responses are instantaneous, ensuring that you never lose a lead (Dale, 2016). According to Tec InStore[3] , the German firm has always provided several processes requested by its clients. Customers were describing problems with their phones and asking customer service a variety of standard questions. It invests a lot of money and works hard to improve customer happiness and respond to their questions. Tec InStore was able to reduce the number of requests to a user utterance processing center by 50% by deploying a conversational bot to present users with relevant information and familiar inquiries.

2.3. Conversational Bot Development Techniques

In this section, we'll go over some of the fundamental deep learning approaches for NLP that are covered in most research articles. These techniques include recurrent neural networks, word embedding, LSTM, and GRU.

2.3.1. Word Embedding

Since the 1990s, vector space models have been employed in distributional semantics (Turney & Pantel, 2010). Since then, several models for estimating continuous representations of words have emerged, including Latent Dirichlet Allocation (LDA) (Campbell et al., 2015) and Latent Semantic Analysis (LSA) (Peters, 2018). The technique of converting words into vectors or

[3] *https://botscrew.com/*

other real numbers that can be preprocessed by natural language modeling is known as word vectoring (Borrelli, Mattia; Zappa, 2017). After all, word embedding is the final product. Word embedding (the result of vectorizing a word) can be used to store the meaning of a word in a low-dimensional vector (a representation of words that includes context), and it can be applied to a variety of NLP studies. The word embedding strategy involves storing statistical information on word co-occurrence with a specific sequence to determine phrases, words, or paraphrases.

Essentially, any feed-forward neural network that accepts words from a dictionary or word library as input and integrates them as vectors into a lower-dimensional space then adjusts them by back-propagation, creates word embedding, also known as the Embedding Layer (Bartusiak et al., 2019). A vector representation of text vocabulary can be thought of as word embedding. It can recognize the context of each word in a document, as well as semantic (meaning) and syntactic (structure) similarity, and integration with other words (Karani, 2018).

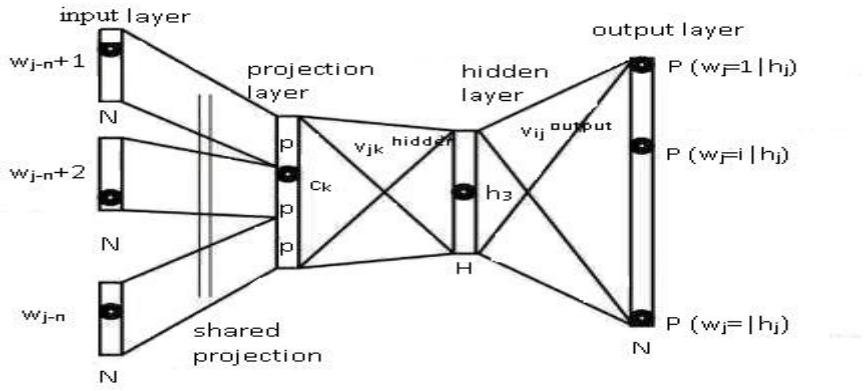


Figure 4 A neural language model

To understand how a language model works, consider the challenge of predicting the next word in the sequence "I want a drink of orange." The condition is implemented using a neural language word embedding in the following figure.

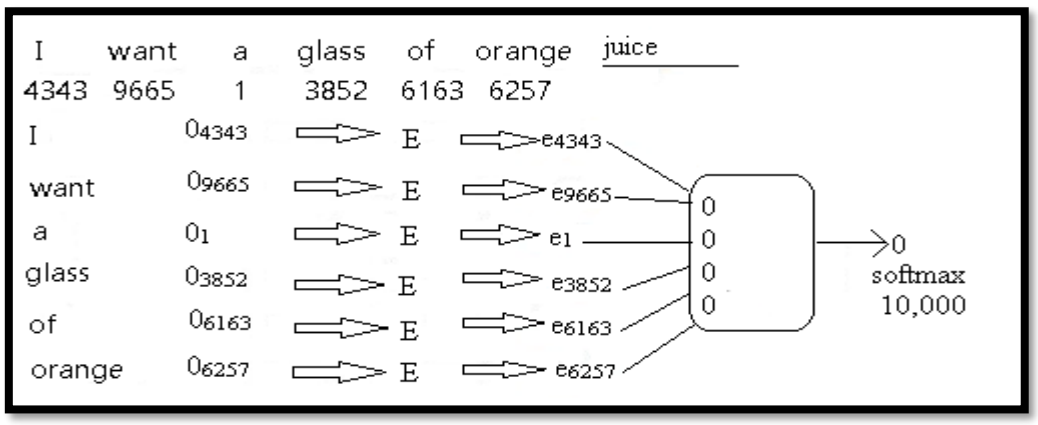


Figure 5 How neural language modeling works

You must create a neural network to predict the subsequent word in the series: For every word, there is a hot vector Ow that is multiplied by an embedding matrix E to produce an embedding vector e w. The dimension of each embedding vector is 300 dimensions (assuming E is 300x10000). Following that, each embedding vector is sent to a neural network, which receives input from a SoftMax layer and outputs probabilities for each word as the subsequent word. Since the aforementioned example uses 300-dimensional embedding vectors and contains 6 words, the neural network will get 1800-dimensional input. To forecast the next word, only a limited window of four words is usually used. In this situation, the ultimate number is reduced from 1800 to 1200.

The primary distinction between a neural language model network and Word2Vec is computationally intensive, which is why word embedding has become so popular in the NLP area. Its existence has undoubtedly been aided by today's rapid increase and high production of computer power. GloVe and Word2Vec's training strategies are the other comparisons for selecting effective word embedding, with both weeds on the generation of word embedding that records sequences of semantic relationships between words and are frequently employed in different deep learning applications. In contrast, stable neural networks extensively provide task-oriented embedding with restrictions on their ubiquitous use.

Bengio (Bengio et al., 2001) uses a real-valued word feature vector in R to demonstrate the basic principle of word embedding. Their model's foundation can still be used in a variety of downstream NLP applications. They formally established three key layers of their network, which may be accessed via a paper (Bengio et al., 2001).

- Embedding Layer: the function of this layer is to create an index vector with a word embedding array to generate word embedding.
- Intermediate Layer(s): this layer consists of one or more layers that provide an intermediate representation of the input, such as a fully connected layer that applies non-linearity to the transition of the previous n-word embedding.
- SoftMax Layer: this layer optimizes the model by attempting to generate a probability distribution in vector space V.

The process of turning words into vectors is known as word embedding. A machine learning system is then able to directly feed the features of this vector format. This procedure can be implemented in a variety of ways, ranging from a simple count vector to deep learning algorithms like a word to the vector (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). Finally, based on this viewpoint, there is particular interest in this study because it has proven to be effective in the domain of communication channels. The skip-gram model is shown in

particular because it is the strategy used for embedding layers in the Python library Keras. According to Mikolov (Mikolov et al., 2013), the skip-gram model's core premise is simple: train an oversimplified neural network with a single hidden layer of fixed size to forecast a given word context. In terms of design, Figure 6 shows a representation of the skip-gram and a CBOW model. If the window size is 5, fictitious training instances of the type (wt., [wt-2, wt-1, wt+1, wt+2]) are produced. The most important task here is to filter the inner layer and utilize It serves as the vector representation of the previously learned vocabulary. Only when the input words are one-hot encoded does it function as a lookup table of vector representations. In practice, this requires first transforming the corpus to index sequences, then training the skip-gram model on those sequences, and then transforming the indices sequences into vector sequences to learn the final algorithm. Word2Vec employs deep neural network techniques even though it is not a deep learning neural network. An embedding layer of this kind can be made using Word2Vec. There are two ways to make it: CBOW or skip-gram (both of which use Neural Networks as their training strategy) (Popov, 2018).

- **Skip Gram:** This method uses the context of each word as input and attempts to anticipate the word that corresponds to it. Skip-gram has been proven to display unusual words well and can handle a small amount of data well.

- **Common Bag of Words (CBOW):** It appears that the CBOW model for numerous contexts has been inverted. To a degree, this is correct. The target term is entered into the network. C probability distributions are generated by the model. Then receive C probability distributions of V probabilities for each context position, one for each word. In both cases, the network is trained using back-propagation methods. CBOW is speedier and offers better representations for terms that are used more frequently.

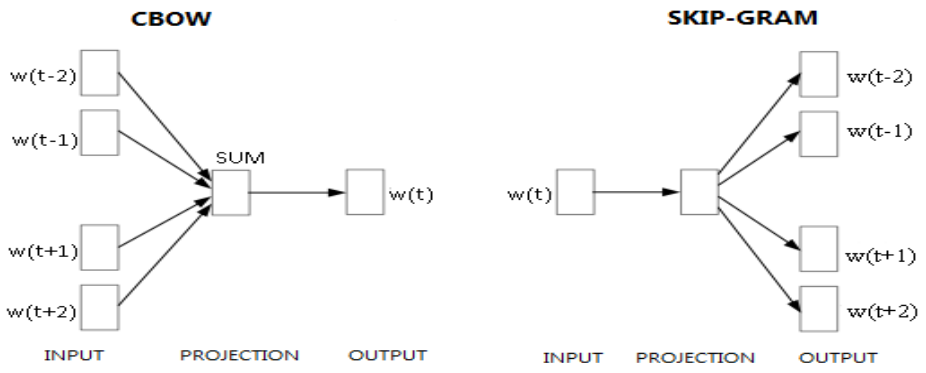


Figure 6 Skip-gram and CBOW Word2Vec method

2.3.2. Recurrent Neural Network (RNN) and its Variants

A recurrent neural network (RNN) current input is governed by the previous hidden state that was learned from the prior hidden state (Vinayakumar et al., 2017). RNNs can receive one or more input vectors and output one or more vectors, with the output(s) being influenced not just by the weights assigned to the inputs, as in a regular neural network, but also by a "hidden" state vector reflecting the context based on prior input(s)/output(s) (s). As a result, the same input could result in a different output depending on the inputs that came before it in the sequence.

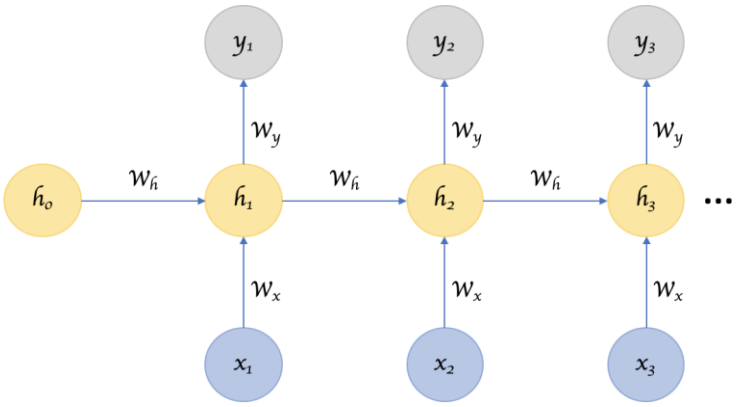


Figure 7 RNN and its information tracking

Each recurrent neuron contains two sets of weights: one for the inputs x(t) and the other for the outputs y from the prior time step (t-1). These weight vectors have the respective names of wx and wy. Put all the weight vectors into the two weight matrices, wx and wy, if we are studying the entire recurrent layer rather than just one recurrent neuron. Equation (b is the bias vector and () is the activation function, for example, ReLU) can then be used to determine the output vector of the entire recurrent layer.

$$Y_{(t-1)}\phi(W_{X^T}.X_{(t)}+W_{Y^T}.Y(t-1)+b \quad (2.4)$$

Notice that $\mathbf{Y}_{(t)}$ is a function of $\mathbf{X}_{(t)}$ and $\mathbf{Y}_{(t-1)}$, which is a function of $\mathbf{X}_{(t-1)}$ and $\mathbf{Y}_{(t-2)}$, which is a function of $\mathbf{X}_{(t-2)}$ and $\mathbf{Y}_{(t-3)}$, and so on. This makes $\mathbf{Y}(t)$ a function of all the inputs since time $\mathbf{t} = 0$ (that is, $\mathbf{X}_{(0)}$, $\mathbf{X}_{(1)}$ … $\mathbf{X}_{(t)}$). In this first stage t=0 there is no previous output, so they are usually assumed to be zero.

asserting that all of a recurrent neuron's inputs from past time steps are a function of its output at time step t, proving that it has memory. A memory cell is a type of neural network component that maintains a state throughout subsequent time steps (or simply a cell). Although a single recurrent neuron or a layer of recurrent neurons is a fairly basic sort of cell, we'll look at some more complex and potent cell types later in this chapter.

A cell's state at time step t is represented by the symbol h(t), where "h" stands for "hidden," and it depends on both its state at time step t-1 and particular inputs at that time step: h(t) = f(h(t-1), x. (t). The output at time step t, indicated by the symbol y, is also a function of the prior state and the current inputs (t). The output is usually equal to the state in the simpler cells that have been studied so far, but this isn't always the case in more complicated cells, as seen in figure 8.

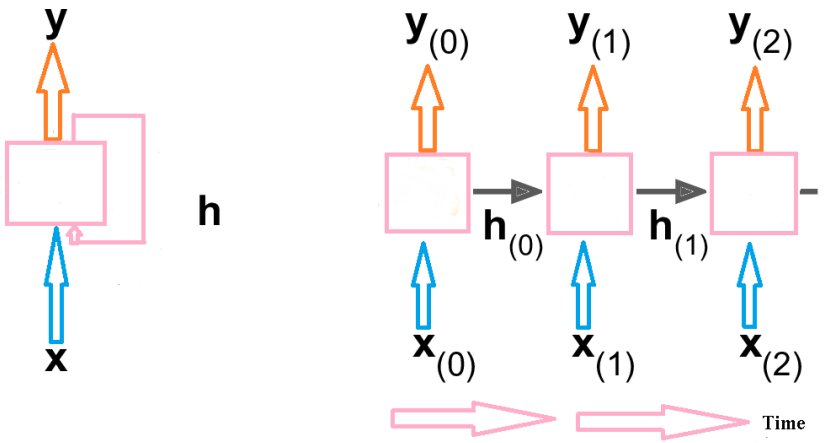


Figure 8 A cell's hidden state and its output

Encoder-Decoder RNNs, also known as Sequence-to-Sequence RNNs, are commonly used in translation services. The core notion is that there are two RNNs, one of which is an encoder that continuously updates its hidden state and outputs a single "Context" output. The context is then sent to the decoder, which converts it into a series of outputs. Another significant distinction in this configuration is that the lengths of the input and output sequences do not have to be identical. In section 2.3.3, it is thoroughly discussed.

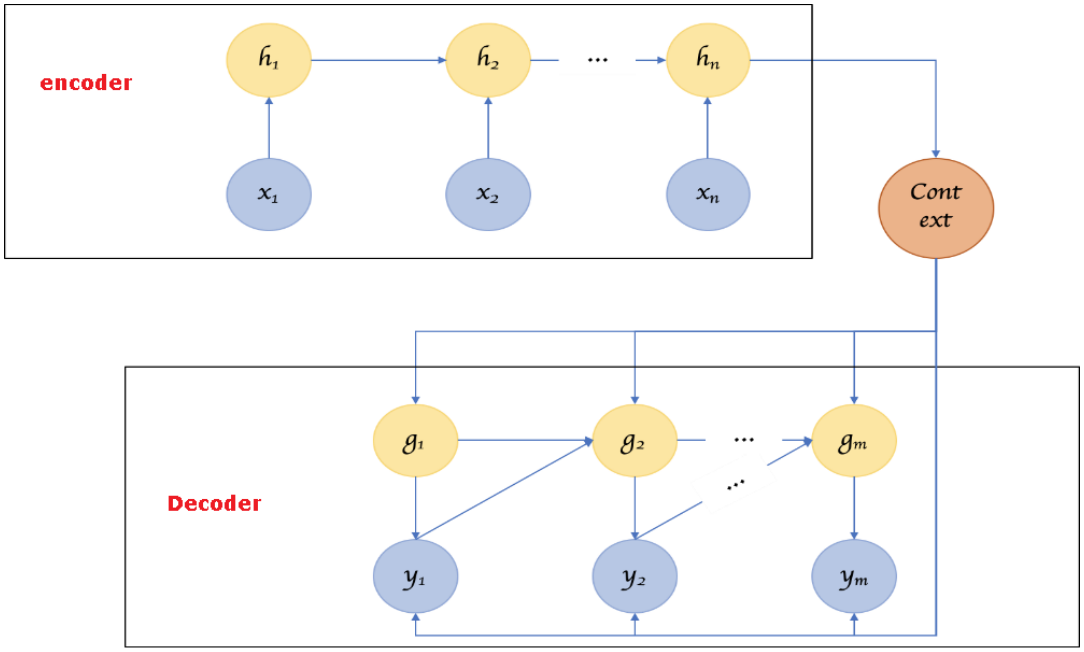


Figure 9 Sequential RNNs

An RNN can process several inputs and outputs at the same time (Vinayakumar et al., 2017). This type of network, for example, is useful for predicting time series such as stock prices: one can feed its prices from the previous N days, and it must output prices from N – 1 day ago to tomorrow. Can also give the network a series of inputs while ignoring all outputs save the last. Also, could, on the other hand, feed the network a single input at the first-time step (and zeros for all subsequent time steps) and have it produced a sequence. This is a network that transforms vectors into sequences.

The input may be a picture, and the output could be a caption for that picture. A vector-to-sequence network, also known as a decoder, may come after a sequence-to-vector network, also known as an encoder (see figure 9). This could be used, for instance, to translate a sentence between two languages. A sentence in one language would be supplied to the network, which would then encode it into a single vector representation before decoding it into a sentence in another language. The concluding words of a sentence can affect the starting words of the translation when using a single sequence-to-sequence RNN (like the one pictured on top left). As a result, you must wait until you have heard the complete sentence before translating it. This two-step encoder-decoder model performs significantly better. For each time stamp, handle a sequence of some vector X using the following recurrent formula:

$$h_t = f_w(h_{t-1}, h_t) \tag{2.5}$$

Where h_t is the network's new state, f_w is a function with w weights equal to h_{t-1}, the network's old state, and h_t is the vector input from a time stamp.

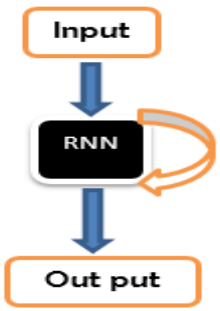


Figure 10 RNN and its recurrent function

When a vanilla RNN is visible, a single hidden layer that handles data mapping between the input and output is present in the state.

$$h_t = \tanh(W_{kk} h_{t-1} + W_{xh} X_t)$$

$$Y_t = W_{hy} h_t \tag{2.6}$$

h_t is the hidden state on the recurrent neural network with the activation function of tan_h, and Y_t is the new hidden state on the recurrent neural network, which is currently output and serves as an input for the RNN layer's next process; at this time, the weight is updated to map correctly between input and output.

2.3.2.1. Long Short-Term Memory (LSTM)

RNNs typically learn from data, rather than remembering the entire word sequence, and instead choose relevant aspects of the models (*Recurrent Neural Networks and Natural Language Processing. by Christopher Thomas BSc Hons. MIAP Towards Data Science*, n.d.). To communicate about food, we define the phrase using significant words that clarify its meaning, as well as explanations gleaned from the environment. RNNs work in a recursive fashion, reprocessing input and output until the network is optimized to a certain threshold value.

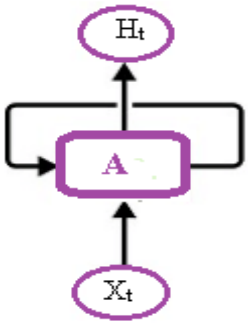


Figure 11RNN and its looping work

With the input from X_t and h_t, the looping network distributes information from one state to another, as shown in figure 11. Unrolled RNNs are created by the information traveling through each network, allowing RNNs to work in a data sequence.

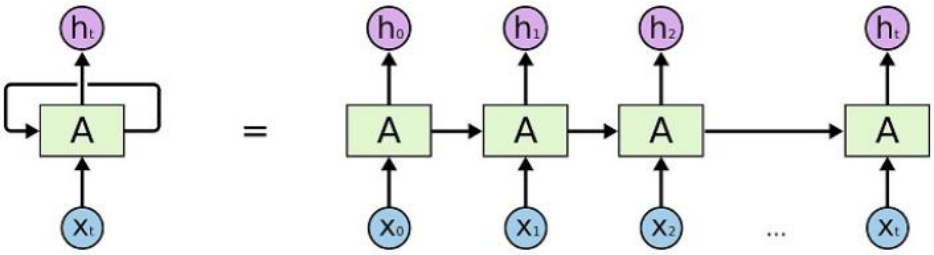


Figure 12 Unrolled RNNs

The RNN processes data in each state and passes its state to the next processing hidden network. In RNN, the output of one hidden layer is the input of another state. language rearrangement, translation (from one language to another language, or one input to another input, such as a given question to an answer), and captions are examples of RNN applications. LSTM long-short memory plays an important part in the design of these systems.

Consecutive data that requires long-term dependencies to select on the current data cannot be managed by RNNs alone because of the vanishing gradient problem. The LSTM version of the RNN can capture long-term dependency, and it was created by (Bordes et al., 2017) to solve the problem with RNNs.

In LSTM, there are three layers: forget gate, input gate, and output gate, as shown in figure 13. (Sutskever et al., 2014). Forget the gate that determines which data should be erased from the cell's state. The task of tossing information is carried out by the sigmoid layer. The activation function "Sigmoid" examines the previous hidden state (h_{t-1}) and the current input (X_t), producing an output that ranges from 0 to 1 for each cell specified (C_{t-1}).

$$f_t = \delta(W_f . [h_{t-1}, X_t] + b_f \quad (2.7)$$

The forgot cell of the LSTM is shown in Equation 2.4, which generates a new cell state f_t by sigmoid the previously hidden state h_{t-1} with the current input using weight functions W_f to capture relevant functions.

The input gate controls the state of the input cell, which is the LSTM's next state. In this layer, fresh information values are recorded and stored as Ct.

$$i_t = \delta(W_i . [h_{t-1}, X_t] + b_i) \quad (2.8)$$

$$\acute{C}_t = tan_h (W_c . [h_{t-1}, X_t] + b_c) \quad (2.9)$$

To prevent information loss, Ct refreshes the cell by processing the preceding sigmoid function with the tanh function, which allows viewing the information broadly. .

$$C_t = f_t * C_{t-1} + i_t * \acute{C}_t \quad (2.\ 10)$$

Equation 2.10 is the updated state C_t, which is obtained by multiplying the old state of cell C_{t-1} by the old state of network f_t, which returns the information on forget state, and then adds it to C_t to provide the updated state of the network.

The LSTM network's output layer is the following layer, which extracts relevant features from the data. The output data is shown in Equations 2.11 and 2.12.

$$o_t = \delta (W_o [h_{t-1}, x_t] + b_o) \quad (2.11)$$

$$h_t = O_t * tanh C_t) \quad (2.12)$$

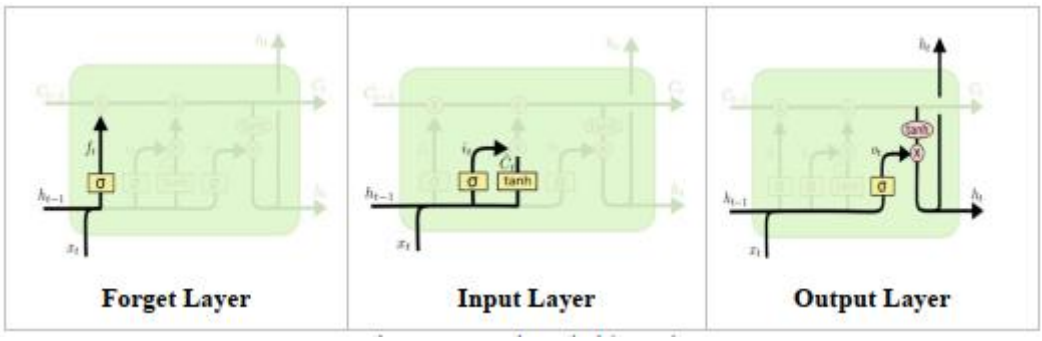


Figure 13 LSTM layers

2.3.2.2. GRU-Gated Recurrent Unit

Cho (*Understanding LSTM Networks -- Colah's Blog*, 2016) introduced GRU, which is a small variant of LSTM. It only has two cells: an update gate (which combines input, output, and forget gates) and an output gate that is computed on the same hidden layer as the cell state. Compared to LSTM models, the computational complexity is lower. The LSTM variant of the RNN is employed in a variety of memory-dependent applications.

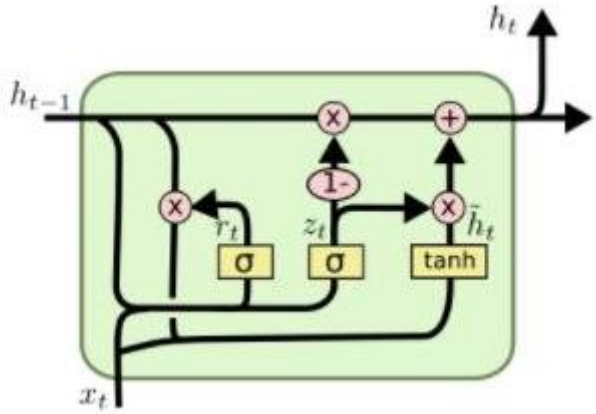


Figure 14 GRU cell state and layers

GRU and LSTMS are nearly identical, however, their layer utilization differs, with LSTM employing three layers and GRU employing only two, as shown in Figure 14. In GRU, equations 2.13, 2.14, 2.15, and 2.16 describe how each layer works. The sigmoid function is used by the update gate to throw and take information, while the $\tan_h$ function is used to process cell state and concealed states.

$$Z_t = \delta(W_z . [h_{t-1}, x_t]) \quad (2.13)$$

$$r_t = \delta\,(W_r . [h_{t-1}, x_t]) \quad (2.14)$$

$$\acute{h}_t = tan_h\,(W . [r_t * h_{t-1}, x_t]) \quad (2.15)$$

$$h_t = (1 - Z_t) * h_{t-1} + z_t] * \acute{h}_t) \quad (2.16)$$

Greff (R. K. Srivastava et al., 2015) looked at different LSTM and GRU network structures. Based on the results of the comparison, the researcher concludes that the two networks are nearly equal in quality (Jozefowicz et al., 2015), nonetheless, analyzed tens of thousands of

RNN structures and discovered that GRU recorded was in some ways superior to LSTMs since it can handle specific tasks. We may therefore conclude that LSTMs and GRUs are equally effective for handling tasks requiring memory dependencies.

2.3.3. Sequence to Sequence Modeling

Input-output vectorization of data (Sutskever et al., 2014), to create an acceptable mapping between the input (encoder) and output (decoder) processes, there is a hidden layer between input and output that can be built.

$$O_t = f(O_{t\text{-}1}) \qquad (2.17)$$

Equation 2.17 describes the function's output features by taking into account the priority of the sequence of information $O_{t\text{-}1}$, and the mapping task is determined by the function f between the previous and next sequences of information.

Sequence-to-sequence models are generative models. The model can anticipate new information once it has been trained. seq2seq models are suitable for models with a large amount of data. The training of a data series for a linguistic model is depicted in Figure 15.

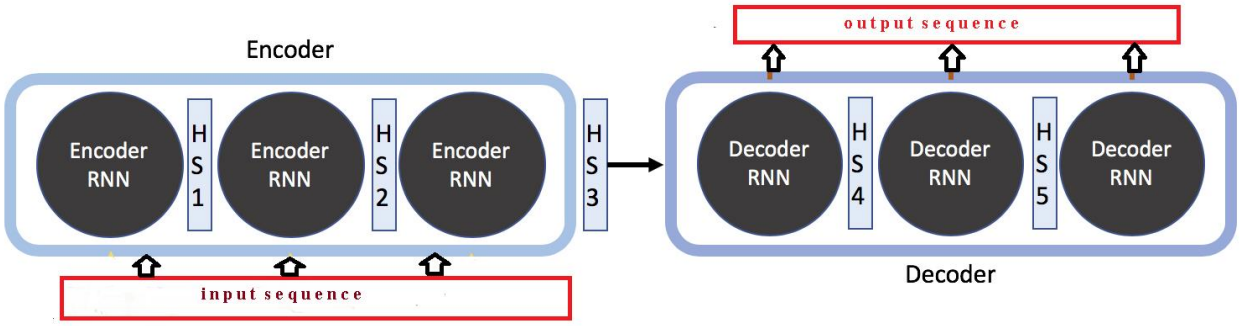


Figure 15 Input and output sequence processing using the Seq2Seq model

The seq2seq model is divided into two components, the first of which is encoding and the second of which is decoding. RNN cells are utilized to transmit the context to the decoder component during the decoding of the input sequence. As shown in Figure 15, the encoder part also uses RNN units to process and predict the next sequence.

2.3.4. Dynamic Memory Network for Simple Question Answering

Four modules make up a dynamic memory network (Kumar et al., 2016). The first module is an input module that accepts a list of facts in a specific order. The second module is the episodic module, which memorizes important phrases from a data stream. The third module is the question module, which handles simple inquiries and passes them to the layers of episodic and input modules. The answer module, which refines the response, is the final module. The DMN network is seen in Figure 16.

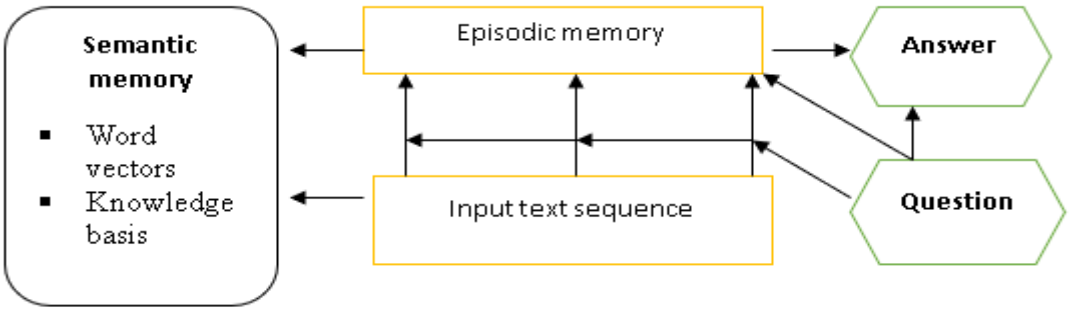


Figure 16 General DMN structure

2.4. Data Consumption Management and Evaluation Metrics

Conversational agents learn from their data, and their replies are dependent on the requirements of utterances (Shawar & Atwell, 2007). The data is not fed immediately; it requires information oversight and planning for learning procedures. The challenge with conversational bots is multi-classification, and data intent management is a subtask of constructing chatbots.

2.4.1. Intent and Response Management for Knowledge Extraction

The conversational agent must be able to identify and make clear the user's desire for the task at hand when receiving a new user utterance. This can be viewed as a multiple categorization problem, with labels denoting different user intentions. As a result, intent categorization serves as the conversational bots' primary engine. This can be considered "text input," which is categorized by a system function called a "classifier," which links the provided data to a particular "intent," and which, as a result, creates a thorough grasp of the token sequence for the system to understand. (Allamanis et al., 2018). This classification is created using artificial neural networks, pattern matching based on regular expressions, and machine learning techniques like SVM, which can be used as a classifier (ANNs currently applied in different fields including classification problems). In this situation, long-short-term memory networks can be used to activate long-term dependence (LSTM networks)(Meng & Huang, 2018).

Knowledge extraction is the process of identifying essential data and delivering it in a way that is appropriate for a given system (Huang et al., 2007). For the most part, artificial intelligence systems make decisions based on knowledge extraction. This is a key source of concern about allowing computers to collect and process knowledge, which has advanced dramatically since the 1980s under the guise of knowledge engineering (Studer et al., 1998). To handle or regulate information and develop new information through first and second-order reasoning, untimely skills frequently needed a deduction engine. It's a method of creating responses to unanswered queries, and it's usually simple to convert into API calls. In the case of conversational agents, for example, to respond to fundamental user inquiries for broad facts, knowledge engineering

is critical. To retrieve data from the web and other sources, Siri and Amazon Alexa use internal knowledge decision and extraction mechanisms. Knowledge management is currently primarily accomplished through API calls and efficient database requests. Although, on occasion, more foreign ways inspired by the graph-structured nature of beings are applied for knowledge (Chung & Park, 2019).

The user receives selective information during response generation. An important strategy when building conversational bots is response generation and management (Allamanis et al., 2018). The discussion is conducted in the manner in which consumers expect to be addressed, namely by maintaining a focus on context while conversing. When user utterances are fed into a conversational model, the hidden layer can often produce relevant and contextually similar information, after all, the qualifying of the messages has been processed depending on the model's final purpose. When considering this, two significant techniques came to mind: retrieval-based and generative-based methodologies. Retrieval-based strategies attempt to assess a discussion by relying on a huge database of nominated responses and matching them with user input messages to get the most important and reliable response.

According to the current state-of-the-art in conversation management, responses are assessed using activation functions derived from nominated responses that are either generative or rule-based. The benefits of retrieval-based systems include responses that are grammatically correct and can readily communicate with humans.

The generative-based technique, which is based on responding to new information that was not supplied during the model's training and testing, is another significant aspect of producing replies. Extraction and computation of novel patterns can easily yield previously unseen responses. The issue in While responding to human utterances, generative models are prone to grammatical errors and the omission of crucial information. Although generative model responses are beneficial, they should be seen as a backup method, meaning that if the data isn't collected, the model will resort to producing responses (Tammewar et al., 2017).

2.4.2. Performance Metrics for Conversational Chatbots

Due to the subjectivity of the working technique, conversational agents make judgments based on several criteria. Each performance metric has advantages and disadvantages, and different types of metrics are used to solve problems in each evaluation technique. The metrics workflow is further explained in the parts that follow.

2.4.2.1. Perplexity Measure

The following can be regarded as the average expectation that can be realized inside the field of information and data theory (Chen, 2016) and reflected in the allocation of the language model effectiveness:

Since entropy is the average number of bits required to encode the information contained in a random variable, and since perplexity is simply the exponentiation of entropy, the exponentiation of entropy must be the total amount of all possible information, or more specifically, the weighted average number of options for a random variable (*Exponential Distribution — Intuition, Derivation, and Applications | by Aerin Kim | Towards Data Science*, n.d.). The likelihood of a token depends on or, in most circumstances, is predicated on the token that has already been revealed. Every new sequence of tokens is given the average number of non-redundant words given by the information origin intrinsic in the data through the per-word split up H as follows:

$$W_{1\ldots}W_{i-1}\ \mathrm{H} = \lim_{n\to\infty}\frac{1}{m} \tag{2.18}$$

$$W_1\ldots,W_{i-1}H = \lim_{n\to\infty}\frac{1}{m}\sum W_1,W_2\ldots,W_m\,(p(W_1,W_2,W_{3,\ldots}W_m)\log_2\,p(W_1,W_2,W_{3,\ldots}W_m)) \tag{2.19}$$

This possible sequence of all words is expressed in summation, which is used to send successful information on the data preset; however, if the source cannot be obtained, the new summation of total possible word sequences can be ignored, and the transmitted information can be expressed as:

$$H = \lim_{n\to\infty}\frac{1}{m}\log_2\,p(W_1,W_2,W_{3,\ldots}W_m) \tag{2.20}$$

Since the total samples space of the information encoded based on the vocabulary set of the data is used to calculate the entropy of how much information is flowing well, the language model should decide whether or not those vocabulary sets are necessary unless doing so will make it more difficult to convey information. During a model's fluctuating property, the huge value of m, H can be re-approximated using the new optimized calculation:

$$\bar{H} = -\frac{1}{m}\log_2\,p(W_1,W_2,W_{3,\ldots}W_m) \tag{2.21}$$

Another predetermined method is to create a linguistic model for predictive evaluation. Starting from the language model, a language model that predicted words using all of the language features available would have an equivalent per-word entropy of H.

As a result, it makes sense to evaluate the real performance of a language model using an entropy-related metric. Perplexity, pp is an example of a commonly used metric, as follows:

$$pp = 2^{\acute{h}} \quad (2.22)$$

Substituting equation 2.21 into equation 2.22 and rearranging obtains:

$$pp = \acute{p}\,(w_1, w_2, w_3, \ldots w_m)^{-1/m} \quad (2.23)$$

Where, $\acute{p}\,(w1, w2, w3, \ldots wm)$ is the probability estimate assigned to the word sequence (w_1, w_2, w_3, … w_m) by a language model. Perplexity can be thought of as an indicator of the likelihood that there will, on average, be many words following a specific word. Although this only suggests that they "model language better," rather than performing better in voice recognition systems, lower perplexities represent better language models. A voice recognition system cannot recognize the importance of words that are acoustically similar or distinct, hence perplexity is only tangentially connected with performance.

Since a language model and a test text are needed to calculate perplexity, using the same test text and word vocabulary set in both scenarios is necessary for a proper comparison of two language models based on perplexity. It is obvious why vocabulary size is significant: as vocabulary cardinality decreases, the number of viable words given any history must monotonically drop, raising the average size of the probability and reducing confusion.

2.4.2.2. Other Metrics for Evaluating Chatbots

Different metrics are typically employed at different phases to evaluate a chatbot's performance. In the training process, a loss is frequently used to discover the "optimal" parameter values for a model (e.g., weights in neural networks). It's what happens when you try to improve your training by changing the weights. Accuracy is more of a practical consideration. The study uses these metrics to measure how accurate our model's prediction is compared to the genuine data once found the optimum parameters above. BLEU (Bilingual Evaluation Understudy), which was created for machine translation, may compare translations by using a reference. It can be used as an evaluation metric for text generated in NLP tasks. However, BLEU has a drawback when it comes to assessing meaning: sentence structure cannot be observed directly, and the evaluation method is not natural, meaning it does not meet human judgment (Post, 2018). In classification issues, F-1 score assessment metrics are used to evaluate intentions. For unbalanced data distribution between intents, the F-1 score works well, whereas accuracy measurements used for balanced data distribution can fail for unbalanced data distribution between intents (Chawla, 2009). There are no uniform criteria for evaluating a conversational agent's model; instead, different researchers apply their performance criteria (Shawar & Atwell, 2007).

The following table summarizes the literature review with their respective authors and year.

Table 1: Summary of the literature reviews

Chatbot name and year developed	Approach/ methods used	Architectural model	Application area	Downside
1966, ELIZA (Radziwill & Benton, 2017)	NLP, Pattern matching	Retrieval based model	Tongue-in-cheek simulation of Rogerian psychotherapist	No logical reasoning capabilities, inappropriate responses
1972, PARRY (Greenler et al., 1977)	Conceptual ontologies, Turing test	Retrieval based model	Mimic the behavior of a person with suspicious schizophrenia	Slower to respond and unable to learn from the dialogue
1988, JABBERWOCKY now named Clever bot) (Lim & Goh, 2016)	Contextual pattern matching	Generative model	To simulate a natural human chatbot for entertainment	Cannot reply at high speed and work with a huge number of users
1995, ALICE (Shawar & Atwell, 2015)	NLP, AIML heuristic pattern matching	Retrieval based model	Pattern matching heuristics are used for user input for the human communication interface	Did not have intelligent features
2006, Watson (Forloina, D., & Jufied, 2019)	NLP, IBMs DeepQA software and Apache UIMA	Retrieval based model	QA system that won the 2011 Jeopardy competition	Does not process structured data. Supported on English
2010, SIRI (*[PDF] "Hey Siri, Do You Understand Me?": Virtual Assistants and Dysarthria | Semantic Scholar*, n.d.)	Java, JS, Objective C, NLP, TTS, STT	Generative model	The computer software serves as an intelligent personal assistant	Difficult to hear the interlocutor
2012, Google Now (Reyes et al., 2019)	RNN, neural network mechanism	Generative model cute	Google searches for mobile apps and takes steps by transferring queries to web services.	It has no personality Its question may violate the users' secrecy.
2015, ALEXA (Ram et al., 2018)	NLP, TTS, STT, Python, Java, node JS	Generative model	It is a personal assistant capable of home automation, entertainment, and setting alarms.	Privacy rights are signed away at purchase, always listening no privacy
2015, Cortana (Paul, 2017)	Python, Java, JS, NLP	Generative model	The intelligent assistant makes reminders, sends emails, etc.	It can run a program that will install malware.

2016, Microsoft Tay (Wolf et al., 2017)	Python, Java, node JS	Generative model	It focused on learning from its surroundings through interaction with people.	Controversy on Twitter and caused social media for offensive Impact.

2.5. Related Works

To better serve users, many researchers have worked to improve the functionality of chatbots in various businesses and industries. Different researchers have introduced many models over the past few years. Nevertheless, the development of chatbots using deep learning models is still in its early stages (Kandpal et al., 2020). Task-oriented conversational systems work on a single task and have a distinct working technique from non-task-oriented conversational bots.

To diagnose basic medical issues for some diseases, Kandpal et al. (Kandpal et al., 2020) presented "Contextual Chatbot for Healthcare Purposes (using Deep Learning)". They created a chatbot model that uses FFNN to identify the disease that a patient may be dealing with. Additionally, they represented sentences with numerical data for the NN model using the BoW technique. The model accuracy is not specified, and the researchers have not assessed the model performance. The dataset has not been divided to assess the model's performance; it has only been prepared to train the model. The authors employed the straightforward feature extraction method known as BoW.

According to (Vinyals & Le, 2015), a neural conversational model produced a task-oriented model by using classic RNN, which is dealing with troubleshooting problems in the technology domain. However, this strategy does not solve the issue of long-term dependency. And then after a year, Bordes & Weston (Bordes et al., 2017) proposed memory networks in 2016 to solve the problem of long-term dependencies, and they use LSTMs to create memory network cells that deal with task-oriented applications, such as restaurant reservations. The authors use 929k, 73k, and 82k words for training, validation, and testing dispersed over a vocabulary of 10k words. This is a pre-processed version of the first 100M characters dumped from Wikipedia. Like a train, validation, and test sets, this is divided into 93.3M, 5.7M, and 1M characters. The <UNK> token is used to replace any terms that appear less than 5 times, resulting in a vocabulary size of 44k. Although this model is capable of memorizing facts, the amount of data it consumes is large, and this technique cannot be used with limited knowledge base techniques. It also needs a long memory to smoothly summarize a discourse.

“Design of Conversational Agents in Maternal Healthcare Support”, which guides expectant mothers throughout their pregnancy, was studied by Mugoye et al. (Mugoye et al., 2019). The researchers have developed a conservation agent that gathers various symptoms from users and

informs them of the diseases they may have and helpful advice. In this study, a multi-agent conversational system has been created using a dialog manager, API, reinforcement learning, and knowledgebase. The model will, however, only return the response if the user request matches the if condition because the system is a knowledgebase system.

“Deep learning methods for question-answering systems” is described by Yashvardhan S Gupta (Sharma & Gupta, 2018), who employs a generic dataset called babI, which was developed by Facebook AI. Essentially, they take into account supportive facts, and this activity involves memorizing data sequences. This explains happy work for simple inquiries like WH questions and yes/no questions that require a lot of information.

Reddy Karri and Santhosh Kumar (Reddy Karri & Santhosh Kumar, 2020) investigated "Deep Learning Techniques for the Implementation of Chatbots," and they compared and discussed the various chatbot technologies. This study compares some chatbots that have already been created, including jabberwocky, Eliza, and Alice, in terms of speed, interoperability, Turing test, and scalability. To get around those chatbots' limitations, the authors combined a bag of word feature extraction with a Seq2Seq model. However, the dataset is the only thing the authors use to model and test the created chatbot model. They haven't tested the model with a dataset. The authors used a big dataset size to train the model, but they haven't used the model overfitting handling technique. The model was trained using a large dataset, but the authors did not employ a model overfitting handling strategy.

A diagnostic chatbot system was developed by Kidwai and Rk (Kidwai & Rk, 2020) to identify user situations based on their symptoms. They achieved 75% accuracy when using a decision tree ML algorithm to diagnose diseases for the users. In this study, the authors used 150 datasets with labeled symptoms and 50 different diseases to train the model. The created chatbot converts from a doctor to a patient by collecting symptoms from users via text or speech to diagnose the illness. The response will depend on the users' responses at each step, even if the presented model is better for diagnosing diseases. In this study, the model will keep asking questions until all associated symptoms have subsided if the patient experiences only a few symptoms. Due to this, users may not be satisfied because it is tedious to use.

In(P. Srivastava & Singh, 2020), “An automated medical chatbot” is a system that uses natural language diagnosis and human interaction to provide medical assistance. This medical chatbot was created with the help of Artificial Intelligence Markup Language (AIML). The chatbot conversation was prepared by the author using a pattern and template tag. The pattern represented the users' input, and the template represented the bot's response. The researchers were tasked with predicting disease symptoms based on user input. With a standard precision

of 65%, this system is qualified to identify symptoms from user inputs. After identifying the user's symptoms, the system predicted the diseases with an accuracy of 0.946, 0.886, and 0.8 using the K-nearest neighbor (KNN), SVM, and Nave methods, respectively. Even if the system accomplishes these goals, it has limitations in understanding user input. This means that the machine has determined whether the users' input matches the pattern.

The other researcher investigated “designing and implementing an adaptive bot model with integrated rules to consult published Ethiopian laws” (Asimare, 2020). The purpose of this paper is to create a chatbot that can consult people about Ethiopian published laws. The Amharic language was used by the researcher for this purpose. The researcher created the chatbot using the RNN model and achieved an accuracy of 73.12%. However, the researcher discovered a limitation in dealing with model overfitting in this study. They only changed the number of data-splitting ratios to use overfitting handling techniques. This is insufficient for dealing with model overfitting.

Segni Bedasa(Dilbo, 2021) proposed "developing an Afaan Oromo Text-based Chatbot to Increase Maize Productivity." The author used DNN and Seq2Seq to achieve the best accuracy of 84.4% in DNN and to create a chatbot that provides vital information to farmers on how to increase maize productivity. The researcher used TF-IDF feature extraction on a small dataset with 25 tags. However, just because the presented model can answer the user's question does not mean that the chatbot has been trained to answer any question the user may ask. The authors did not specify a probability threshold for the question for which the system does not have an answer. Furthermore, the researcher did not employ model overfitting management techniques.

Fekadu Eshetu(Fekadu Eshetu Hunde, 2021) gave a presentation titled "Design of a Bilingual Chatbot to Assist Ethio-Telecom Customers with Customer Service." The developed model provides customer service in two languages, Afaan Oromo and Amharic, to Ethio-Telecom customers. The author used Bi-LSTM algorithms and obtained an accuracy of 82.6% using the L2 regulation technique, which is used to deal with model overfitting. Even if the presented model is superior for assisting Ethio-telecom customers with customer service, the researcher has not compared different optimizer algorithms to improve the model's performance. There are several optimizer algorithms, and sometimes Stochastic gradient descent (SGD) outperforms Adam in the study by (Vamsi et al., 2020), even though he did not use weight initializers.

2.5.1 Summary of Related Works

In summary, several scholars have attempted to improve chatbot functionality using cutting-edge methods to develop a chatbot solution for every industry. The development of chatbots for a variety of industries, including the healthcare sector, has been the subject of the studies mentioned above. However, studies that have been presented previously for various health domains only use either generative or retrieval-based approaches, which may not be as effective as an advanced approach, such as an overall architectural model of the two approaches, and did not apply necessary accuracy enhancement techniques such as hyperparameter tuning and model overfitting control techniques. Accordingly, the objective of this study was to create an ensemble chatbot consultant for the therapy of psychological clients with mental disorders. Moreover, various methods to improve the functionality of the current chatbot model will be used. Such as; using weight initialization techniques, searching for the right network hyperparameter for the proposed model optimization using hyperparameter tuning, etc. Generally, using the new architecture model and a limited corpus of the most commonly seen mental disorders in Ethiopia from the DSM-5, as well as other related and relevant documents, this effort aimed to develop the model and overcome the challenge of smooth conversation tracking.

Table 2 Table summary of related works.

No	Titles of the papers	Algorithms with accuracy	Gaps identified
1	Contextual Chatbot for Healthcare Purposes (using Deep Learning) (Kandpal et al., 2020)	FFNN Accuracy not specified	▪ There is no model evaluation technique used. ▪ The authors used the BoW feature extraction technique, which does not take into account the context of the words.
2	A neural conversational model (Vinyals & Le, 2015)	classic RNN	▪ A common failure mode of this model is inconsistency. ▪ The problem of long-term dependency
3	Deep Learning Techniques for Implementation of Chatbots (Reddy Karri & Santhosh Kumar, 2020)	RNN	▪ Researchers did not use any model evaluation techniques; instead, they checked whether or not the model responded to the user question. ▪ The authors used a large dataset size but did not employ the model overfit handling technique. ▪ The authors used the basic feature extraction method known as BoW.

4	Design and development of diagnostic Chabot for supporting Primary Health Care Systems (Kidwai & Rk, 2020)	Decision Tree (75% accuracy)	▪ If the user answers all of the bot's questions, the model will diagnose the disease. If the user experiences a few symptoms, the model fails.
5	Learning end-to-end goal-oriented dialog (Bordes et al., 2017)	RNN with LSTM memory networks	▪ cannot be used with techniques that have a limited knowledge base. ▪ It is also necessary to have a large memory.
6	Automatized Medical Chatbot (Medibot)	AIML for taking symptoms for disease prediction KNN, SVM	▪ User inputs should be the same as with the pattern prepared by AIML
7	Conversational Agent's Role in Maternal Healthcare Support (Mugoye et al., 2019)	Dialog flow Knowledgebase system	▪ If the user question does not match the if-condition, the system will not respond.
8	Developing Afaan Oromo Text-based Chatbot for Enhancing Maize Productivity (Dilbo, 2021)	DNN (84.4%) and Seq2Seq (80.0%)	▪ The threshold probability for candidate response was not specified by the author. ▪ The researcher did not employ model overfitting handling techniques. He made use of train test split ratios.
9	Designing and Implementing Adaptive Bot Model to Consult Ethiopian Published Laws Using Ensemble Architecture with Rules Integrated (Asimare, 2020)	RNN (73.12%)	▪ Limitation of handling model overfitting. i.e., it uses data splitting ratios only
10	Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services (Fekadu Eshetu Hunde, 2021)	Bi-LSTM (82.6%)	▪ To achieve better results, the author did not compare the algorithms of different optimizers. ▪ The author did not use weight initializer techniques, which are used to correctly adjust the initial weights to ensure the model's accuracy.

CHAPTER THREE

RESEARCH METHODOLOGY

This chapter briefly described the research design, including data collection methods, data analysis, preparation and utilization, tools and techniques, along with experimental procedures with assessment metrics.

3.1. Research Design Methods

To address the issue raised in chapter one's section 1.4, which looks at quantitative research techniques such as data collecting, cause-and-effect invariance, assessment metrics, and statistical analysis to support or contradict a theory, the study uses experimental research. To perform the study, data has been gathered and prepared model input, designed and implemented, and evaluated the experimental research design. The information in this paper was gathered from the DSM-5, as well as other related and relevant materials, client stories, online psychology advising sites, and general daily conversation data. The fitting networks between distinct neural networks are investigated, and descriptive ways to visualize the importance of the model are studied.

The model was developed with the intent of combining the most important elements of retrieval and generative models. The importance of obtaining relevant information from those models for the design of suggested architecture is significant. As mentioned above, the retrieval model is decent at retrieving grammatical information if the user utterance is identical under a certain threshold, implying that the retrieval model is only concerned with the structural similarity of the user utterance and not with the meaning of the utterance. In addition, the generative model has its own set of issues when it comes to responding; it frequently yields non-grammatical data, and the accuracy of the answer is limited; yet, it has a high quality when it comes to retrieving previously undiscovered qualities from the dataset.

The optimization principle of the two models with a relatively new design is employed in this case, which considers the nature and amount of data. The architecture appears to be as follows. The data core is the first thing that is obtained. The data core is grammatically correct, and relevant data is collected as a result; the task-oriented conversational AI should be as accurate as feasible. On top of the data core, a flexible user utterance manager is created, which is achieved by an agent correlated with the Amharic Word2Vec model and the Psycho bot data core, allowing for a smooth conversation process and good retrieval of important information. The study was conducted to evaluate the current state of conversational agents. This allows for the creation of a task-oriented conversational agent for the most commonly observed mental

disorders in Ethiopia, based on the DSM-5, as well as other related and relevant documents. The goal of this study is to design a psycho bot conversational agent with a proposed architecture that takes into account the state of the art in conversational agents. The conversation process is assessed using perplexity, accuracy, and f-1 scores, and the proposed model is subsequently reviewed using user-acceptance testing to check the naturalness of the conversation and relevant responses for the desired converse.

The general research design procedures used to conduct this study are summarized as follows.

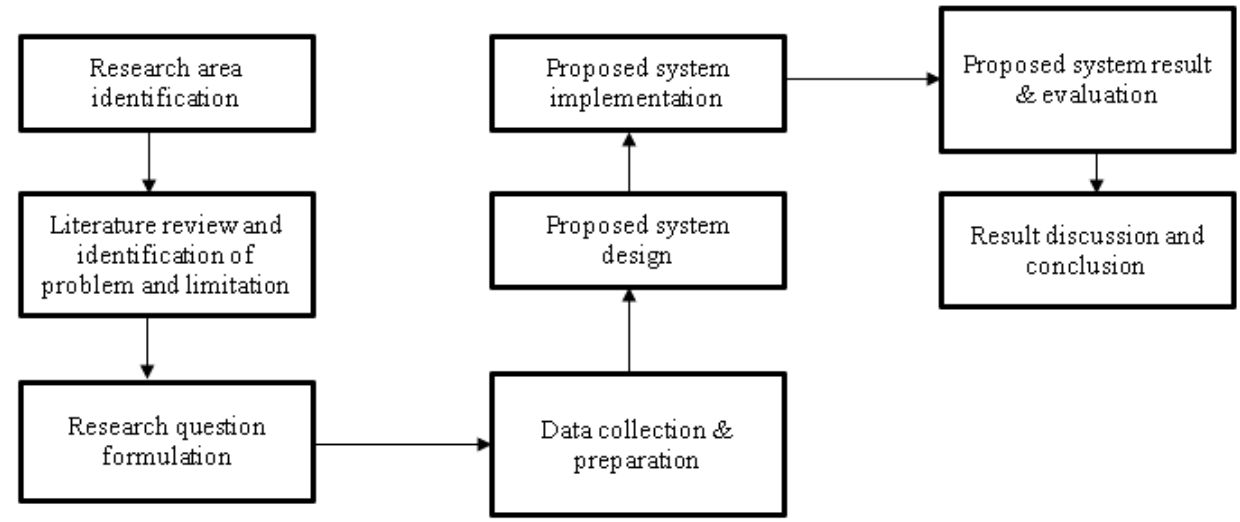


Figure 17 Research design procedure description diagram

3.1.1 Experimental Works

The experimental technique, which chooses a model network for training, testing, and validating purposes, is one of many methods used to assess the performance of the model. Four networks are assessed based on the proposed architecture: LSTM, single GRU, double GRU, and Transposed GRU. The network is selected based on its loss, accuracy, and training time characteristics. The results of these studies are presented in Appendix I. The model was also assessed based on perplexity (a total sample space to encode the information), which is displayed in Appendix I by summarizing experimental findings, training epoch investigation, and test accuracy of the model.

3.2. Data Collection Preparation and Utilization Methods

3.2.1. The Data Source and Collection Methods

What is DSM?

Health care practitioners in the United States and much of the rest of the world use the DSM as a reliable resource for consulting or diagnosing mental disorders and making treatment

recommendations. The DSM includes information on the causes, signs, and other diagnostic standards for mental disorders. It establishes valid diagnoses that can be utilized in research on mental diseases and their classification and gives mental health professionals a common language to discuss their clients (American Psychiatric Association, 2022). It also gives researchers a common language to examine the standards for upcoming changes, provide treatments and consultations, and develop other involvements.

The DSM-5 Diagnostic and Statistical Manual of Mental Disorders 5th edition(del Barrio, 2004) as a primary source, as well as other related and relevant documents available, are gathered from the internet (such as Aha psychological service [4], Abrhot Specialized Psychodiagnostics & Therapy Center[5], VIDA mental health consultancy[6] and Natanim counseling and training plc[7], Impact Ethiopia psychological service and public health consultancy[8], Melkam psychotherapy[9], St, Amanuel mental specialized hospital[10], Sitota center for mental health care[11], Born Again Mental Health Rehabilitation and Healing Charity Association[12] and are our site source. Data preparation has been done on the documents that have been collected. To build our task-oriented Amharic word vector, we used the DSM-5, as well as other related and relevant documents, 120,000 sentences were collected through the scrappy method from Ethiopian mental health consultancy websites and various online discussion forums. The study extracted semantically comparable words using the Amharic word vector approach. To appropriately handle the user conversation, it is mostly made up of mental health discussion forums and consultancy websites. This Amharic word vector model accepts several sentences, which are then converted into vectors, allowing for additional operations on those words.

The collected dataset must be reviewed or checked by a mental health expert. And it was collected for only preventive purposes, and the designed model does not include any medical prescriptions unless possible suggestive consultations. Therefore, the collected dataset is associated with mental health consultancy services such as general knowledge queries, what to do if a mental health disorder happens or is likely to happen, auto-generated consulting advice, and client stories for re-habitation aims. The DSM-5, which is the product of quite ten years of

[4] *https://www.ahaethiopia.com*
[5] *https://www.abrhot.com*
[6] *https://www.mentalhealthafrica.org*
[7] *https://www.natanimethiopia.org*
[8] *https //www.impactet.com*
[9] *https://www.melkam-psychotherapy.business.site*
[10] *https://www.amsh.gov.et/*
[11] *https://www.sitotapsy.com*
[12] *https://www.bornagainrehabilitation.org*

effort by hundreds of international consultants in all told aspects of mental health and is currently in its 5th edition(Catherine Pearson & Pearson, 2013) is a guideline for different types of mental health disorders. This guideline serves as a data standard for data preparation.

Google Translate, which is one of google machine translation technology nowadays which uses an AI-powered neural machine translation algorithm, has been used to translate the resources used for the model from the source language i.e., English to Amharic. According to a survey conducted in 2017 and an updated review in 2021 (Aiken, 2019), the Google translator has reached 85% of its total accuracy while translating world languages. So, the researchers used this worth from the technology and later reviewed it by the Amharic language professionals after translating directly by google Translate. The letter for language review from language professionals is displayed in Appendix VI.

3.2.1.1 Questionnaires

Two questioner types are conducted here, the first is used to generate utterances in terms of [Q]: [A] pairs (the generation of utterance from the Psycho bot document is based on 45 people's utterance generation), and the second is used to evaluate the model by end-users based on naturalness, response generation of the model, response time, and way of user query understanding and the system replying for 20 people. Appendix VII is where you'll find this statistical design method.

3.2.2 The Data Preparation and Utilization

The data from the collected documents are first extracted into question (Q) and answer (A) pairs, which are then fed into the model, which subsequently learns what to remember and how to extract information. The model's utterances are not the only ones; they may vary depending on the user, and the given question or intent is representative of announcing the fact to the model. The question's dynamic property is eventually tracked using sequence-to-sequence modeling techniques, which take into account the bot's input structure and Amharic word vector for unseen words during training.

The DSM-5, which is the product of quite ten years of effort by hundreds of international consultants in all told aspects of mental health and is currently in its fifth edition(Catherine Pearson & Pearson, 2013), as well as other related and relevant documents, mental health disorder cases, and discussion topics, mental disorder advice from psychologists and psychiatrists, client stories, and general mental health information, are the main data sources for this work. Section 3.2 explains how to acquire data using these methods. The model is fed with the hand-crafted data preparation as [Q]: [A] pairs.

The main data source for the model is the DSM-5, as well as other related and relevant documents for the most commonly reported mental disorders in Ethiopia available on the web and in discussion forums. Then simply used all information and tagged them appropriately with JavaScript Object notation (JSON) file format.

When preparing the dataset by JSON, it contains three things such as tags, patterns, and responses.

- **Tag:** is the user's query's intent or idea, which is used as a target class while training the model.
- **Patterns:** a collection of queries or frequently asked questions that are related to the intents.
- **Responses:** the collection of possible responses to the users' queries. This is employed when the users.

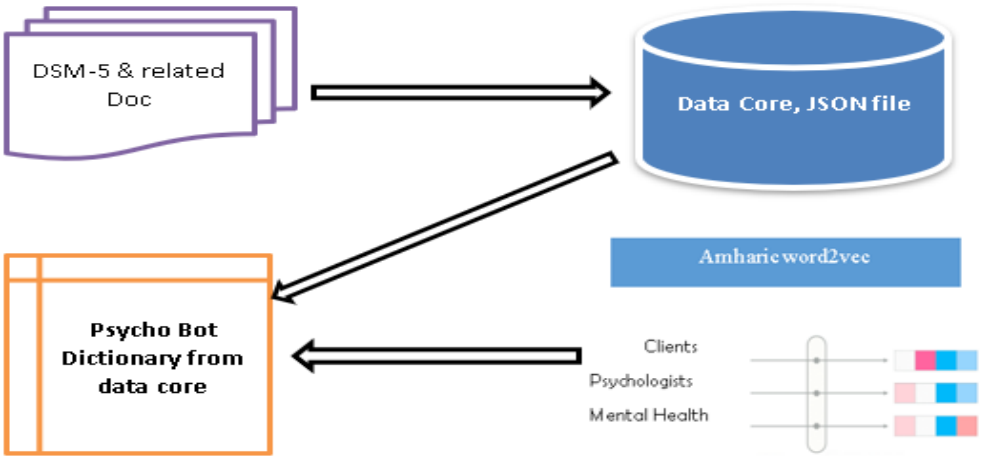
The following shows how the dataset is prepared. For example, the users want to ask greeting "እንዴት ነዎት?", "ጤና ይስጥልን?", "ሰላም?", "ሀይ", "ሰላም", "እንዴት ነሽ?". The response for those queries may "ጤና ይስጥልን ", "ሰላም ነው ምን ልረዳዎት እችላለው", "አለሁ ደህና ነኝ", "አለሁ ምን ልርዳዎት?". From this, the intent or the tag of our dataset is greeting. Here, have prepared two greeting tags. Therefore, the converted JSON dataset was as the following;

```
{
"tag": "greeting",
"patterns": ["ሰላም?", "እንዴት ነህ", "ሰላም ነው", "ሀይ", "ሰላም", "እንዴት ነሽ?", "ሰላም ዋልክ?",
"ሰላም
ነሽ?", "ኣለህ", ",ጤና ይስጥልን ", "ሁሉ ሰላም", "አማን ነው"],
"responses": ["ታዲያስ ", "ሰላም ነው", "ደህና ነኝ", "አለሁ ምን ልርዳዎት?"]
}
```

Based on this dataset then train the model and finally, it predicts the tag and selects from the response randomly, and gives the appropriate responses.

Each information tag has a distinct purpose and ID. Each purpose was eventually obtained from the user's utterance. The essential data core of the intended system is created by this data arrangement. The handcrafted tagging is done manually. For the sake of representative tagging, the researchers used a strategy in which clients communicate with mental health professionals and get information through diagnosis. The isolated model created by the Amharic word vectors derived from discussion forum sentences has an edge in retrieving context-comparable words before the user utterance is analyzed.

Figure 18 The data utilization description



From the data core, a dictionary with a set of vocabularies is generated, which is then used to obtain semantically related words based on the vocabularies, which is then utilized as an inference during the training and testing of the model. As a result of maximizing the use of those terms, or optimizing the consumption of those words, the language model can be quickly evaluated, and dynamic dialogue can be implemented while the user converses with the model. The utterances were written after considering 45 people considered different ways of expressing the same topic. Then accept the expression of 45 persons solely since did not get any additional expression of themes to construct multi-expression for a single topic. Around 35 of those persons are employing similar expressions for specific purposes. The researchers generate patterns for the dataset by generalizing them. And a generating model with 45k pairs of utterances is ready to train, validate, and test. The utterances are generated as follows based on thirty-five (35) people saying patterns and are depicted in table 4.

Table 4 Sample utterance creation process

Retrieved specified intent	**Generated utterances based on 35 people saying patterns**
ባይፖላር ወይም ጥንድ ዋልታ መታወክ በሰው ስሜት እና በኃይል ደረጃ ላይ ከፍተኛ ለውጥ የሚያመጣ የአእምሮ ጤና ሁኔታ ነው ። ሁሉም ሰው ውጣ ውረዶችን የሚያጋጥመው ቢሆንም እንደ ባይፖላር ወይም ጥንድ ዋልታ መታወክ ዓይነት በመመርኮዝ በሰው ሕይወት ላይ ከባድ ተጽዕኖ ሊያሳድሩ የሚችሉ የስሜት እና የባህሪ ለውጦች አሉ ። በተጨማሪም ባይፖላር መታወክ ያለበት ሰው በጣም ከፍ ያለ ወይም ብስጭት የተሞላበት ስሜት (ማኒክ ክፍሎች	ባይፖላር ወይም ጥንድ ዋልታ መታወክ ምንድ ነው? ስለ ባይፖላር ወይም ጥንድ ዋልታ መታወክ የሆነ ነገር ንገረኝ? ባይፖላር ወይም ጥንድ ዋልታ መታወክ ምን ማለት ነው? ባይፖላር ወይም ጥንድ ዋልታ መታወክ ምን አይነት ህመም ነው? ስለ ባይፖላር ወይም ጥንድ ዋልታ መታወክ ልታስረዳኝ ትችላለህ?

ይባላል) እንዲሁም የመንፈስ ጭንቀት ክፍሎች ሊያጋጥመው ይችላል ።	
After those patterns are realized JSON file format is put like this Patterns: ባይፖላር ወይም ጥንድ ዋልታ መታወክ ምንድ ነው", "ስለባይፖላርወይም ጥንድ ዋልታ መታወክ የሆነ ነገር ንገረኝ", ባይፖላር ወይም ጥንድ ዋልታ መታወክ ምን ማለት ነው", ባይፖላር ወይም ጥንድ ዋልታ መታወክ ምን አይነት ህመም ነው", "ስለባይፖላርወይም ጥንድ ዋልታ መታወክ ልታስረዳኝ ትችላለህ" Content: ባይፖላር ወይም ጥንድ ዋልታ መታወክ በሰው ስሜት እና በኃይል ደረጃ ላይ ከፍተኛ ለውጥ የሚያመጣ የአእምሮ ጤና ሁኔታ ነው ። ሁሉም ሰው ውጣ ውረዶችን የሚያጋጥመው ቢሆንም እንደ ባይፖላር ወይም ጥንድ ዋልታ መታወክ ዓይነት በመመርኮዝ በሰው ሕይወት ላይ ከባድ ተጽዕኖ ሊያሳድሩ የሚችሉ የስሜት እና የባህሪ ለውጦች አሉ ። ባይፖላር ወይም ጥንድ ዋልታ መታወክ ያለበት ሰው በጣም ከፍ ያለ ወይም ብስጭት የተሞላበት ስሜት (ማኒክ ክፍሎች ይባላል) እንዲሁም የመንፈስ ጭንቀት ክፍሎች ሊያጋጥመው ይችላል።	

The dataset contains a minimum of one and a maximum of ten utterances for the same subject. The proposed model's goal is to answer more queries from those patterns that need retrieving the same content. If there isn't anything to recover, it moves on to the next step, which is to manufacture. The next sections go over those procedures.

3.2.3 Preparing Custom Rules

Aside from the utterance prepared, several guidelines or templates are incorporated. By establishing guidelines for the dialogue, certain incorrect responses can be eliminated. In table 5, some of the regulations are listed.

Table 5 Sample for custom rules generated

Rule Templates	**Description of the rule**
U: _unk_ R: ይቅርታ አልገባኝም!	If the utterance has one unidentified token, and difficult to analyze the utterance then the model replies ይቅርታ አልገባኝም!
U: _unk_ _unk_ R: ይቅርታ ፤ አልገባኝም በደንብ ቢገልጹልኝ ልንግባባ እንችላለን።	If the utterance has two unidentified tokens, and difficult to analyze the utterance then the model replies ይቅርታ ፤ አልገባኝም በደንብ ቢገልጹልኝ ልንግባባ እንችላለን።
U: _unk_ _unk_ _unk_ R: ይቅርታ ፤ ምንም አለረዳሁትም እባክዎትን በደንብ ያስራዱኝ በደንብ መግባባት እንድንችል!	If the utterance has three unidentified tokens and is difficult to analyze utterance then the model replies ይቅርታ ፤ ምንም አለረዳሁትም እባክዎትን በደንብ ያስራዱኝ በደንብ መግባባት እንድንችል!
U: _unk_ _unk_ _unk_ _unk_ R: ይቅርታ ፤ ግን በደንብ ቢያብራሩልኝ በድግሜ ከይቅርታ ጋር?	If the utterance has four unidentified tokens, and difficult to analyze the utterance then the model replies ይቅርታ ፤ ግን በደንብ ቢያብራሩልኝ በድግሜ ከይቅርታ ጋር?

U: _unk_ _unk_ _unk_ _unk_ _unk_ R: ይቅርታ ፤ ግራ ተጋብቻለሁ ። በደንብ ቢገልጹልኝ ደስ ይለኛል?	If the utterance has five unidentified tokens and is difficult to analyze utterance then the model replies ይቅርታ ፤ ግራ ተጋብቻለሁ ። በደንብ ቢገልጹልኝ ደስ ይለኛል?
U: _unk_ _unk_ _unk_ _unk_ _unk_ _unk_ R: ይቅርታ ፤ በዚህ ጉዳይ ላይ ያወኩት ነገር የለም ይልቁንስ ስለ አእምሮ ጠናዎ ብናወራ ይመቸኛል ።	If the utterance has six unidentified tokens, and difficult to analyze the utterance then the model replies ይቅርታ ፤ በዚህ ጉዳይ ላይ ያወኩት ነገር የለም ይልቁንስ ስለ አእምሮ ጠናዎ ብናወራ ይመቸኛል ።
U: _unk_ ማነው? R: ይቅርታ አላውቀውም!	If the utterance has one unidentified token with ማነው? Token, and difficult to analyze the utterance, then the model replies ይቅርታ አላውቀውም!
U: _unk_ _unk_ ማነው? R: ይቅርታ ፤ ላውቀው አልቻልኩም!	If the utterance has two unidentified tokens with ማነው? Token, and difficult to analyze the utterance, then the model replies ይቅርታ ፤ ላውቀው አልቻልኩም!
U: _unk_ ታውቀዋለህ ማነው? R: ይቅርታ አላውቀውም!	If the utterance has one unidentified token with ታውቀዋለህ ማነው? Token, and difficult to analyze the utterance, then the model replies ይቅርታ አላውቀውም!
U: _unk_ _unk_ ማን እንደሆነ ልትነግረኝ ትችላለህ? R: ይቅርታ ፤ ላውቀው አልቻልኩም!	If the utterance has two unidentified tokens with _ ማን እንደሆነ ልትነግረኝ ትችላለህ? Token, and difficult to analyze the utterance, then the model replies ይቅርታ ፤ ላውቀው አልቻልኩም!
U: _unk_ ታውቀዋለህ ማነው? R: ይቅርታ አላውቀውም!	If the utterance has one unidentified token with _ ታውቀዋለህ ማነው? Tokens, and difficult to analyze the utterance, then the model replies ይቅርታ አላውቀውም!
U: _unk_ _unk_ ማን እንደሆነ ልትነግረኝ ትችላለህ? R: ይቅርታ ፤ ላውቀው አልቻልኩም!	If the utterance has two unidentified tokens with ማን እንደሆነ ልትነግረኝ ትችላለህ? Tokens, and difficult to analyze the utterance, then the model replies ይቅርታ ፤ ላውቀው አልቻልኩም!

3.3. Data Preprocessing Techniques

As many researchers do, data preprocessing is the most important step in chatbot development. It is the first and most important step in developing a learning model because it prepares the data for the model. It helps to improve the dataset's quality. The model cannot use data collected from the real world directly because it contains noise and is in an unorganized format that is unsuitable for the model. To effectively train the model, data preprocessing must be applied to

the datasets, and then the completely cleaned data is ready for the model. Data preprocessing techniques are used before developing the deep learning model. In the study, various data preprocessing techniques were used, such as removing unnecessary items and punctuation, removing stop words, normalization, and tokenization.

3.3.1 Data Cleaning

Data cleaning is the process of removing noise data from a dataset to make it more understandable by the model, such as removing punctuation marks. There are various punctuation marks in Amharic, such as [word separator, full stop (period), comma, semicolon, colon]. They also used the question mark and exclamation mark. They make no significant contribution to the model's learning time. Some other preprocess was text clearing, which involved removing various symbols and numbers from the text so that the computer could easily understand natural language. The Amharic language has a numbering system that looks like [፩, ፪, ፫, ፬, ፭, ፮, ፯, ፰, ፱…] and may use numbers [0-9]. As a result, the first activity of this study was to determine whether or not those numbers were exits. Remove them from the text if they exist. The other symbols are when users are using chat, they may use different emoji symbols like [😀 😁 😆 😂 🤣 🤚 🤚 ✋ 🖖 👌 🤲 🤝 🙏 ✍ 👮 👷 💂 🙅 🧑 🈯 🤦 ….]to convey their message. Furthermore, removing those symbols were required to clear the text. For example, users could type "ሰላም ✋✋✋?".

3.3.2 Removing Stop Words

For designing a chatbot model, not all words found in the documents are equally important. Some words are irrelevant, and they are not used to categorize the questions of users. To reduce the learning time of the model, those words must be removed from the dataset. The Amharic stop words were gathered from previous studies. The previously prepared by (Fekadu Eshetu Hunde, 2021) is used for this work. Pronouns, prepositions, conjunctions, articles, and particles are the most common terms in any language. The list of Amharic words used are attached at APPENDIX V. Words like “እኔ”, “እኔ”, “የእኔ” , “አንቺ”, “ያንተ”, “የአንተ”, “ምንድን”, “የትኛው”, “እኔ”, “ነው”, “ናቸው”, “ነበር”, “አንድ”, “የ”, “እና”, “ግን”, “ከሆነ”, “ወይም”, “በ”, “ለ”, “ጋር”, “ስለ”, “መካከል”, “ውስጥ”, “በኩል”, etc. are some commonly occurred stop words in Amharic language.

3.3.3 Data Normalization

Text normalization refers to a set of related tasks designed to level the playing field for all text. The writing system of the language is used to normalize the Amharic text. Replace the letter with the letter that has the same sound in Amharic. For example, in the word "ሳይኮሎጂ" the

letter "ሳ" was replaced with "ሣ" because those letters have the same sound. This also helps the computer understand that the word "ሳይኮሎጂ" has a similar meaning to "ሣይኮሎጂ".

In general, text normalization was used in this study to put all words on equal footing and allow processing to continue consistently (Dilbo, 2021).

3.3.4 Word Tokenization

Tokenization is the most basic operation performed on text data when working with it. The patterns must be tokenized into tokens using spaces during the text preprocessing step. For instance, "የአእምሮ ጤና ምንድ ነው?" which is mean "What is mental health?" This sentence is then tokenized as 'የአእምሮ', 'ጤና', 'ምንድ', 'ነው', '?'. As deep learning models do not work directly with text data, the fully cleaned data must be converted into numerical data using text feature extraction or word embedding methods.

3.4 Feature Extraction Techniques

The next step is feature extraction, which is performed after the text has been cleared by removing unnecessary symbols, punctuation, and text normalization. As discussed in section 2.3 of Chapter 2, when using a machine-learning algorithm and working with textual data, machine-learning algorithms cannot work directly on the raw text. As a result, feature extraction techniques should be used to convert text into a feature matrix (or vector).

The study used word2vec techniques to convert text data into numerical values for the proposed work. The entire corpus is scanned in the word2ve model, and the vector creation process is carried out by determining which words the target word occurs with the most frequently, and then the semantic closeness of the words to each other is revealed. (Mikolov et al., 2013). In a neural network model, the word2vec algorithm is used to learn word associations from a large dataset of text. The study chose Word2vec techniques because they are a popular embedding method that is also very fast to train.

3.5. Activation Function Techniques

Activation functions play a significant role in shaping the output of a neural network model in deep learning. The activation functions are powerful decision-makers when it comes to translating nominated inputs to certain output values, and they have a significant impact on the accuracy and efficiency of deep learning models. The activation function can prevent the training model from learning the models in some cases.

There are various types of activation functions in deep learning. Sigmoid (Logistic) function, TanH / Hyperbolic Tangent, ReLU (Rectified Linear Unit), Leaky ReLU, SoftMax, and Swish

are some examples. The study chose the ReLU activation function among those for the proposed system's input layer. The study chose this activation function through various experiments and achieved good results when compared to other activation functions, and used the SoftMax activation function at the output layer. According to (Kumawat, 2019), the SoftMax function has an advantage in handling multiple classes prediction.

ReLU activation function, which is half-rectified and maps inputs from 0 to infinity, is not dying and is not saturated, so the middle value is not impacted the same way tanh and sigmoid functions are. After that, a layer of neurons that employ the tanh activation function receives the requester's response. Binary cross-entropy was used as a loss function at this time. It is a unique example of the definite cross-entropy when m = 2. The loss function that is optimized during training, including the formalization of the ReLU layer, is:

$$L(\theta) \triangleq -\frac{1}{N}\sum_{j=1}^{N} y_j(\theta) + (1-y_j)\log(\theta)) + \lambda w^T w \qquad (2.23)$$

where N is the total number of samples displayed, y_j is sample j's actual label, y_j is the label the network forecasted for sample j, and w is the weight matrix of the ReLU layer. For multiclass classification, the SoftMax activation function was used. There is a mathematical computation between each layer in a deep neural network. This mathematical calculation is used to generate the class prediction's confidence level. In multiclass classification, the SoftMax activation function normalizes these values by dividing by the sum of all the exponentials. This ensures that the sum of all exponential values equals one. Then obtained confidence values ranging from zero to one after using the SoftMax activation function. The target class of the users' query has been identified based on those values. The SoftMax formula is as follows, according to (Gao & Pavel, 2017).

$$\delta(z)i = \frac{e^{zi}}{\sum_{j=1}^{k} e^{zj}} \; for i = 1,...,k and z = (z_1,...,z_k) \qquad (2.24)$$

3.6. Weight Initialization Techniques

While working on deep learning algorithms, it is vital to adjust the initial weights correctly to ensure the model with better accuracy. Before training the models on the training dataset, weight initialization assigns a small primary value to the neural network model parameters. Also, it may prevent the problem of RNN models vanishing or the explosion of the gradients during the backpropagation. There are many weight initialization techniques such as Zero

Start, Randomly Start, Xavier (Glorot Start), etc.[13]. Among these techniques, the study used Xavier (Glorot) weight initialization because it performs better in RNN-based models.

3.7. Model Overfit Handling Techniques

Model overfitting is the most common problem in neural networks, which occurs when the built model fails to generalize well for unknown data but is well-optimized on the training set. Model overfitting occurs in other cases when there is insufficient data or when the architecture becomes complex. There are a variety of methods for dealing with deep learning model overfitting. To deal with model overfitting, deep learning models commonly use cross-validation, the two most common regularization techniques[14], dropout, data augmentation, and early stopping. Because the proposed models are prone to overfitting, the cross-validation method and the L2 regularization technique were used in this study. The cost of L1 regularization is related to the absolute value of the parameters. Some of the weights will be equal to zero as a result, whereas L2 regularization will add a cost based on the squared value of the parameters. As a result, the weights are reduced. The L2 regularization technique is useful because it reduces training time, reduces loss of generalization error, and adds squared magnitude as a penalty for forcing the weights to a small, but not zero, value (Pykes, 2021).

3.8. Hyperparameter Tuning Method

When working with neural network-based models, the three layers of the models must be tuned with the appropriate hyperparameters. The difference in accuracy between models is based on differences in hyper-parameters within the same neural network. Finding the best hyperparameter for the neural network model necessitates more attention to improve the model's performance. Tuning the neural network with the correct hyperparameters, such as the number of iterations (Epochs), learning rate, optimization algorithms, neurons in each layer, batch size, and so on, is a time-consuming and experimental task. There are two tuning hyperparameters; grid search and randomized search (Nabi, 2019), to find the optimal hyperparameter for the neural network.

3.9. Model Evaluation Techniques

The study employs various evaluation processes during and after the model implementation, such as cross-validation (for model optimization), data organization, model training accuracy, loss, perplexity, and subsequently human evaluation study testing of the model. Even though

[13] *https://www.geeksforgeeks.org/weight-initialization-techniques-for-deep-neural-networks/*
[14] *https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/*

different researchers evaluate different conversational areas differently, there are no uniform standard evaluation criteria, as indicated in the paper (Simeon Kostadinov, 2017). We've decided to evaluate the model based on how much it learned (model evaluation in training based on perplexity and accuracy of the model), as well as user acceptability testing after the model has been developed and presented to the end-users. The data organizations are assessed depending on how the query is expressed. Since the model should have dynamically understood the context, patterns to convey intents are required.

Cross-validation is a method of measuring model performance that divides the dataset into k equal partitions and is used for iterative training and testing. It is used to prevent model overfitting when the amount of data is limited and to assess the model's generalization capability. There are different types of cross-validation. Among them are; K-fold and stratified K-fold cross-validation. The dataset is randomly partitioned into K equal samples in the K-fold evaluation technique, whereas stratified is an extension of K-fold cross-validation in which each fold has an equal number of samples from each class as a whole stratified manner. It distributes data from all classes to all folds (Pandian, 2022). Thus, the stratified K-fold cross-validation ensures that each fold represents the complete data.

The review is used to examine and recreate the semantic and structural expressions of the utterance preparations. During the training of neural networks, the correctness of the training model is also critical. The major evaluation is how much knowledge is lost and learned throughout the training. While training the model, the model's accuracy is recorded during the testing and training phases, and the model's perplexity is studied to see how to track the information during the testing and training stages. Then again, user acceptance testing will be used to evaluate the model. Information regarding user satisfaction points is employed in user acceptance testing. The naturalness of the model response, response relevance of the model, model usage, beauty of the user interface, and way of user query understanding and the system replying are all taken into account when collecting information from users.

3.10. Development Tools and Techniques Used

3.10.1 Designing Tools

These methods are used for developing, presenting, and interpreting design concepts. Microsoft Visio (*What Is Microsoft Visio® | Lucidchart*, n.d.) is a program that allows users to make various diagrams. This is used for designing this work. It's a simple yet effective visual design tool for making professional-looking flowcharts, network diagrams, and flowchart diagrams.

3.10.2. Hardware Development Tools

To implement this research, the following hardware tools are used. To install and work with the below tools, the personal computer that is well-found with an internal hard disk of 500GB, 4GB of physical memory, processor Intel(R) CPU, 2.50GHz, Windows 10 pros, and system: 64-bit operating system, x64-based processor.

3.10.3 Software Development Framework Tools

The study assesses the model's context as well as the implementation environment before deciding to develop the research. **Python** is our chosen programming language for designing, implementing, and testing our chatbot. There are numerous tools, but Python is interactive with its accessible packages, and it's not a tough language to learn. The Natural Language Toolkit (**NLTK**) is a well-known Python library for working with natural language. Tokenization, stemming, lemmatization, tagging, word count, and other features are available in this library. NLTK is designed to support natural language processing and closely related areas such as machine learning, artificial intelligence, information retrieval, and linguistics. In this study, the NLTK library is used to work with and preprocess the datasets.

The first is **Keras** (B Arnold, 2017), which allows for quick experimentation with the overall purpose of offering a standard and easy-to-use interface for a variety of deep learning frameworks such as **TensorFlow** (Pattanayak, 2017). **Theano** is a program that allows us to efficiently define data, optimize, and evaluate mathematical formulas involving multi-dimensional arrays, among other things. On top of the TensorFlow library, the conversational bot employs Keras. **Pandas** (Bernard, 2016) is another useful package, which efficiently provides data structures and operations for handling numerical tables and time series. It comes in handy once addressing monumental amounts of data. The **sci-kit-learn library** (Paper, 2020), which was intended to function alongside the Python programming language numerical and scientific libraries, NumPy and SciPy, is tiny (compared to this implementation) but important. This is used for automatic utilities like partitioning a dataset into train validation and test sets in this context. The proposed approach is implemented with the **Anaconda** application version 1.9.7 with 64-bit support, which is used to install the latest version of Python together with all of its numerous modules and IDEs. Additionally, the **Jupyter Notebook** is the used which is most widely used and convenient IDE for working with Python among AI and deep learning researchers. **Colab** also known as Collaboratory is a free Jupyter notebook that runs on the cloud and stores data on google drive. It allows the users to run deep learning code that takes a long time within a short time by allowing users to use GPU and TPU.

The study used this web application for the visualization of data, processing data, writing the code, and run the code. Finally, **Notepad ++** is a free source code editor that is used in this work to prepare an Amharic text dataset. It is a simple text editor to prepare datasets and save them with a JSON file extension. The dataset is prepared in the form of JSON format using notepad ++ code editor.

CHAPTER FOUR

DESIGN AND EXPERIMENT OF THE PROPOSED MODEL

The design of the proposed ensemble architecture model, the procedure to drive the intended model function, and basic techniques for constructing a conversational bot are examined in this chapter. The best fit processes are shown, and network strategy selection is explained via performance graphs that visualize the casual design techniques.

4.1 The Proposed Architecture of AI Psycho Bot

The architecture is meant to convert a specific input to a specific output utilizing a mapping sequence to a specific fixed-length value. This can produce appropriate replies beyond the user's flexible input sequence, which is a collection of all words from the inquiry. Each word is characterized by x_i, where i signifies the word's order. And the formula (Mishra & Gupta, 2016) is used to calculate the hidden states h_i:

$$h_t = f(W^{(hh)}h_{t-1} + W^{(h_x)}X_t) \qquad (1.25)$$

The model receives and retrieves texts; hence its overall behavior is a text-to-text conversation model, with Amharic as the conversation language. The model architecture of the psycho bot conversational agent is shown in Figure 19.

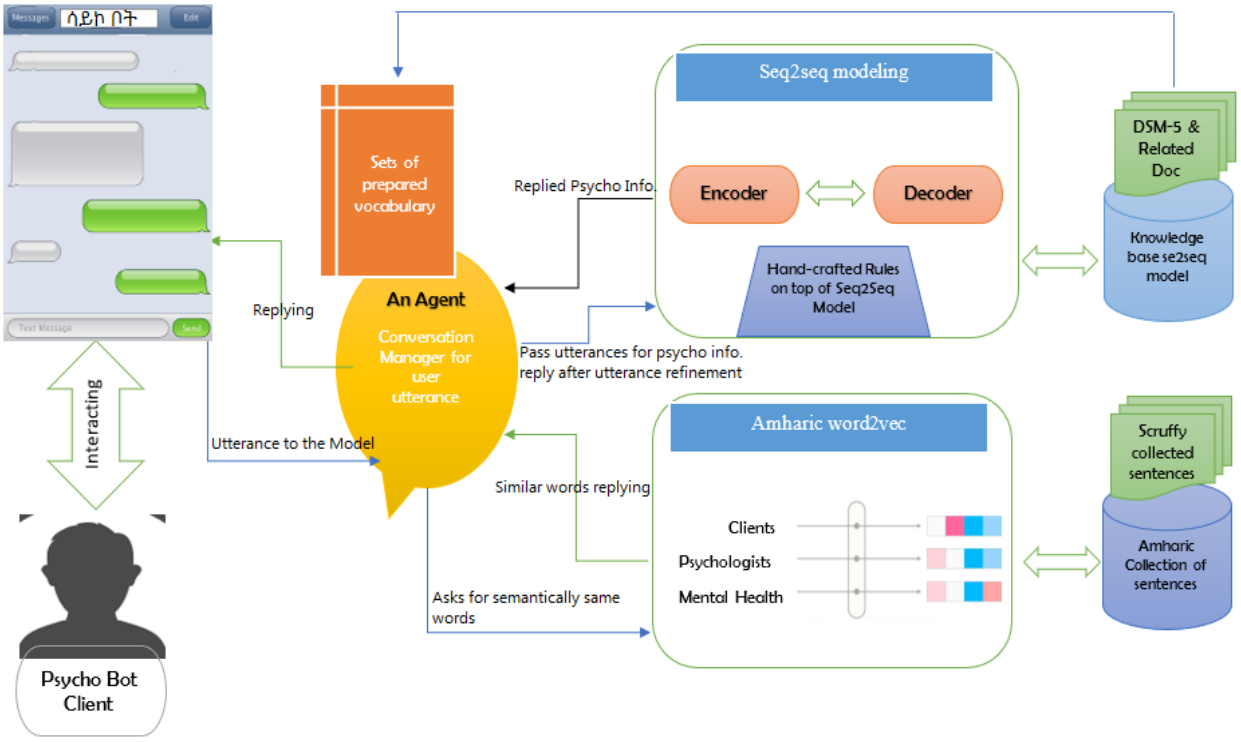


Figure 19 The proposed Architecture of psycho Bot model

Here, the model is built using **seq2seq** with custom rules on top to manage the user conversation, and Amharic **word2vec** is used to extract new semantically comparable terms

about the vocabulary sets. The architecture that is intended is based on taking text via chatting, and the agent then refines the user input by pulling pertinent data from the core model. The utterance is fine-tuned by scrubbing characters that aren't important to the conversation (#, @, $, %, etc.). So, cleaning means either removing extraneous marks or changing and eliminating known contextual terms, thru the Amharic Word2Vec model. The seq2seq modeling module, which pulls data from nominated replies using an activation function, receives the data before supplying it to the memory network layer. According to figure 19, the communication process is broken down into seven general processes for a single conversation, and this process is repeated n times for n interactions. The vocabulary set is used for perplexity language modeling evaluation as well as inference during training, testing, and conversation establishment. The **encoder-decoder** architecture uses an LSTM or GRU encoding-decoding layer for sequence-to-sequence modeling, with the training performance of the LSTMs or GRUs being chosen. Because sequence-to-sequence modeling is encoder-decoder based, it incorporates recurrent neural network principles. The decoder forecasts the response while the encoder converts the input to output. The data for the seq2seq model is hand-crafted using guidelines from the DSM-5, as well as other relevant documents. The role of the agent is depicted in Figure 20 during the dialogue. By combining the two end components of all conversational bot models, the user, and the machine, Agent has created a communication facilitator.

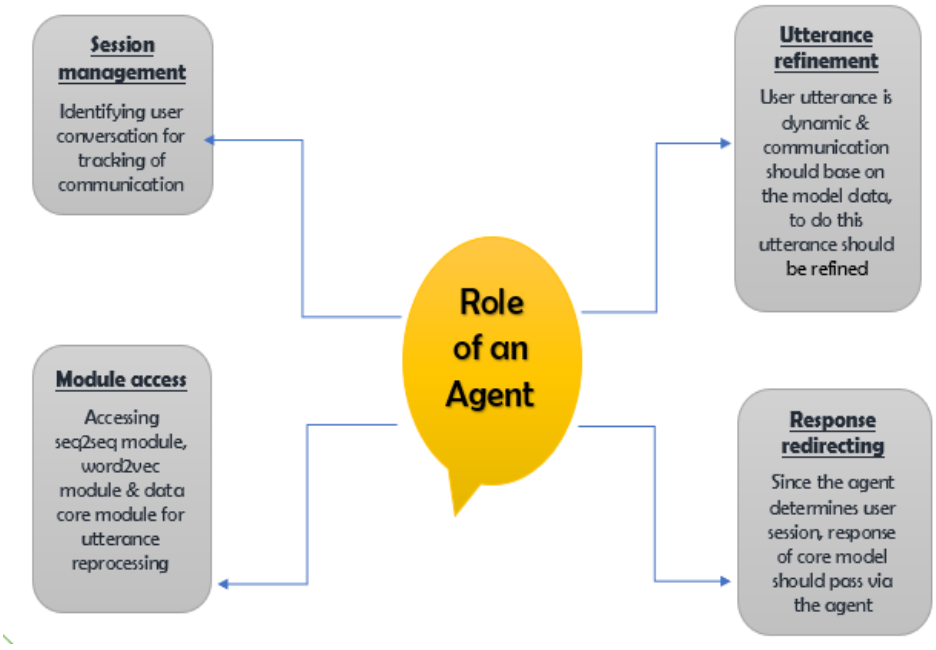


Figure 20 Agent environment model & the role of the agent

Some components must be incorporated in the agent environment model i.e., an intermediate of the Psycho Bot model for proper information flow. For instance, the user message is a dynamic and flexible utterance that the agent may at any time see on the system's interface.

They are composed of a character manipulation that represents the actual text that the user submitted as well as a useful representative metadata structure that includes additional information. For example, the structure contains a pointer that links the message to the structure that represents the specific conversation it belongs to as well as the time and platform at which the message was sent. The information held by the psycho bot agent environment can only be received; it cannot be changed. The fact that the agent has access to the psycho bot's backend, which contains additional user data as well as information about the database's current state, is another significant feature of the agent environment. The agent has access to and control over a few backend functions. The conversational bot can also reply to users to get more details from them or just to let them know that their request has been heard and properly handled. Essentially, the agent refines the conversation during the interaction, which is handled by the agent. It has four key functions: user input refinement, module access for user input refinement, chat session management, and response redirection from the seq2seq model.

4.2 Implementation Flow Description

This section covered all aspects of the proposed system's implementation, from data reading to system evaluation. The proposed system's implementation is broken down into several steps, including dataset loading or reading, data preprocessing, text vectorization, model selection, model training and saving, model loading, and model evaluation.

- **Reading our dataset:** which was prepared using JSON, required the step of "loading JSON data." first import the JSON library, which has a *load()* method used for reading the data from the files. We read our dataset using this method or function.
- **Data preprocessing:** The first step is to preprocess the data before using it directly. At this time, the data has been cleaned, normalized, and tokenized.
- **Text vectorization:** In the proposed system's implementation, deep learning techniques were used, so to train the model, the text had to be transformed into a vector. In this study, text was converted into a vector using a word2vec program.
- **Model selection:** various deep learning techniques are used for the proposed system. For the proposed system implementation, then put those models into practice in this step.
- **After choosing a model,** trained using data from our dataset before saving it. As a result, providing the input to forecast the new data was a response for us. The model must be saved after training to be used repeatedly.
- **Load trained model:** Following model training, it must put the model to the test by

providing fresh data. Then load the trained model that has been saved for this purpose to make predictions. The pickle library is used to load the trained model in the proposed system implementation.

- **Model evaluation:** In this step, finally the study assessed the efficacy of the model trained using our dataset. By asking whether the input text was predicted correctly or incorrect, and then by dividing the data into train and test data, the model was evaluated using human evaluation.

4.3 Proposed System Data Preprocessing Implementation

Data preprocessing comes next after data preparation. Data preprocessing aims to facilitate the computer's comprehension of natural language. To create the suggested system, various data preprocessing techniques have been employed, as covered in the methodology section. In this section, how we put the methodologies used to create the suggested system into practice has tried to demonstrate.

4.3.1 Tokenization Implementation

Tokenization is the process of splitting sentences into a list of words. The NLTK library contains the word tokenize function that is used for performing tokenization. By using this module, tokenization has been employed for the proposed system.

4.3.2 Data Cleaning and Normalization Algorithms

Eliminating unnecessary symbols or characters from the data is a process known as text cleaning. To increase the accuracy of our model's prediction, any punctuation, Amharic numbers, symbols, or characters has been removed. The symbols were cleared using the following algorithms (Abebawu Eshetu, n.d.) to put the suggested system into practice.

4.3.2.1 Data Cleaning Algorithm

INPUT: Uncleaned data

OUTPUT: cleaned dataset

START:

1: Read the dataset

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

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

Eliminate punctuations

```
def remove_punc_and_special_chars(text):
    normalized_text
```

```
=re.sub('[\!\@\#\$\%\^\«\»\&\*\(\)\…\[\]\{}\¤\¦\?\
    return normalized_text
```

IF data include number and English characters [𝟘 𝟙 𝟚 𝟛 𝟜 𝟝 𝟞 𝟟 𝟠 𝟡 𝟢 𝟣 𝟤 𝟥 𝟦 𝟧 𝟨 𝟩 𝟪 𝟫 A B C . . . a b c . . . 0 1 2 …] THEN

Eliminate Number

```
def remove_ascii_and_numbers(text_input):
    rm_num_and_ascii=re.sub('[A-Za-z0-9]','',text_input)
    return re.sub('[\'\u1369-
u137C\']+','',rm_num_and_ascii)
```

IF data include symbols [😀 😁 😆 😂 🤣 🤚 🤚 ✋ 🖖 👌 🤛 🤝 🙏 ✍ 👮 👷 💂 👩‍⚕️ 👦 🈯 👨‍⚕️....] THEN

Eliminate Symbols

```
from clean-text import clean
clean(text, no_emoji=true)
```

Return cleaned data

3. STOP

4.3.3.2 Data Normalization Algorithm

INPUT: Unnormalized data
OUTPUT: Normalized data
START:
1: Read the dataset
2: WHILE (it is not the end of the file):
IF data includes [ጎሐ]' THEN
replace with 'ሀ'
IF text includes [ሑኑ] THEN
replace with 'ሁ'
IF text includes [ኂሒ]
THEN replace with 'ሂ'
IF text includes[ኌሔ] THEN
replace with 'ሄ'
IF text includes [ሕጎ] THEN
	def normalize_char_level_missmatch (input_token):
	cha1=re.sub('[ሃጎኃሐሓኻ]','ሀ',inp ut_token)
	cha2=re.sub('[ሑኙዅ]','ሁ',cha1)
	cha3=re.sub('[ኂሒኺ]','ሂ',cha2)
	cha4=re.sub('[ኌሔኼ]','ሄ',cha3)
	cha5=re.sub('[ሕጎ]','ህ',cha4)
	cha6=re.sub('[ኖሖኾ]','ሆ',cha5)
	cha7=re.sub('[ሠ]','ሰ',cha6)
	cha8=re.sub('[ሡ]','ሱ',cha7)
	cha9=re.sub('[ሢ]','ሲ',cha8)
	cha10=re.sub('[ሣ]','ሳ',cha9)
	cha11=re.sub('[ሤ]','ሴ',cha10)

replace with 'ህ'
IF text includes [ኖሐ] THEN
replace with 'ሆ'
IF text includes [ሠ ሡ ሢ ሣ ሤ ሥ ሦ] THEN
replace with corresponding [ሰ ሱ ሲ ሳ ሴ ስ]
IF text includes [ዐ ዒ ዓ ዔ ዕ ዖ ኣ] THEN
replace with corresponding [ኡ ኢ አ ኤ እ ኦ ኣ]
IF text includes [ጸ ጹ ጺ ጻ ጼ ጽ ጾ] THEN
replace with corresponding [ፀ ፁ ፂ ፃ ፄ ፅ ፆ]
Return normalized text
3: **END**
	cha12=re.sub('[ሥ]','ስ',cha11)
	cha13=re.sub('[ሦ]','ሶ',cha12)
	cha14=re.sub('[ዓኣዐ]','አ',cha13)
	cha15=re.sub('[ዑ]','ኡ',cha14)
	cha16=re.sub('[ዒ]','ኢ',cha15)
	return cha16

4.3.2.3 Stop Word Removal Algorithm

INPUT: all list of words in the dataset **OUTPUT:** list of words does not have a stop word **Variable:** Stopwords [], RemovedStopWord [] 1: Read the dataset 2: WHILE (list of words in the dataset): IF the list of words in the dataset is not in Stopwords []: THEN *Append* the word in RemovedStopWord [] Return RemovedStopWord [] **3: Halt**

4.4 Proposed Feature Extraction Implementation

When using deep learning or a machine learning algorithm, input text data should be transformed into a numeric vector. the word2vec technique was employed for this purpose, as covered in the methodology sections. The word2vec techniques were used for the proposed system's development, as shown by the implementation that came after.

4.5. Amharic Word2Vec Generation

This aids the conversational bot's ability to extract contextual words for a successful bot response. For confined corpus, the Amharic word vector plays a big role in making user

conversations more vibrant. A custom-made Amharic Word2Vec is created here, with the majority of the sentences focused on mental health and derived from a variety of websites, books, and manuals. Finally, gathered 120,000 sentences from which we constructed a word vector. The posterior distribution of words, which is a polynomial distribution, is obtained by applying a SoftMax of a log-linear classification model.

$$p(w_j \mid w_1 = y_j = \frac{\exp(u_j)}{\sum_{j=1}^{u} \exp(u_j)} \qquad (4.1)$$

u_j is a score for each word in the vocabulary, and y_j is the output of the j-the unit in the output layer. It is the loss function (E). where $j * c$ is the value of the vocabulary word that belongs in the c-th output context. We use this to train the custom Amharic Word2Vec corpus for 1554 minutes (25hr and 9min.)

4.5.1. Procedures For Creating an Amharic Word2Vec Model

Preparing data: The Amharic text used to develop the Amharic word2vec is shown in Figure 21. This data comes from a variety of mental health discussion boards, manuals, websites, and other related sources.

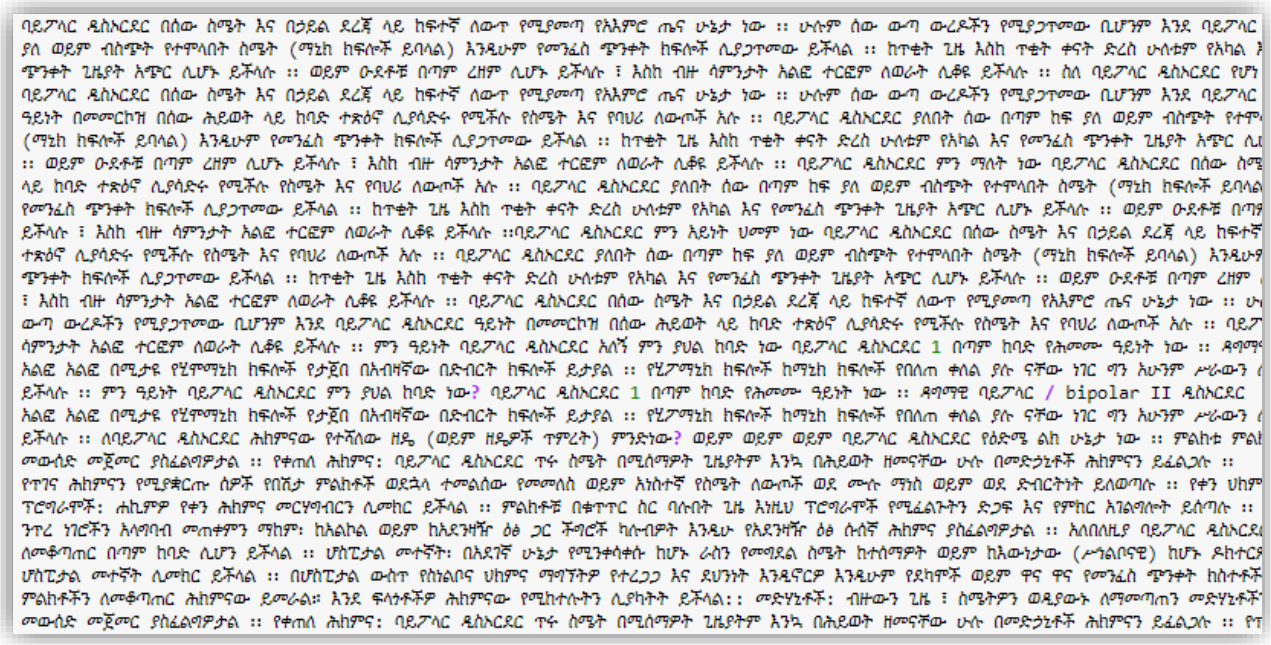


Figure 21 Snapshot of Amharic text corpora for constructing Word2Vec

Removing stop words from the corpus that was gathered: The primary stage after loading the Amharic corpus is to clean it to remove stop words. Except for stop words and special characters, all types of information are useful for generating Word2Vec. And any punctuation

that isn't needed for vector construction. Once the corpus sentences are separated by (: :) and each sentence is embedded, the sequence of words and their mapping is created. After eliminating stop words, data is separated with (: :) and passed to another code called *get sentence list(corpus)*, which removes irrelevant punctuations (! ⁝ ⁝ (,) , = ,¨, _, +, -, …) and special characters (#,@,%,&,*,~,^,…). The below shows a source code that was used to remove stop words.

```
#Cleaning the psychobot word2vec corpus before processed
def clean_text(text):
     delete_dict = {sp_character: ‘ ‘ for sp_character in
string.punctuation}
     delete_dict[‘ ‘]= ‘ ‘
     table = str.maketrans(delete_dict)
     text1=text.translate(table)
     #print(‘cleaned: ‘+ text1’)
     textArr=text.split()
     text2=’ ‘.join([w for w in textArr if (notw.isdigit()]
and (not w.isdigit()) and len(w)>2))])

     return text2.lower().split(‘ ‘)
```

Define Windows Size – The model uses neighbors 3 to extract context-relevant words. The size of the window impacts how context is captured. three neighbors for context capturing are employed here. As the window size is increased, more contextual comparable words may be obtained, but the model's performance suffers since top semantic comparable word extraction is required in our model. Furthermore, dropping to 2 loses information; hence, choose to use Windows Size 2.

```
[] word2int = {}
  For I, word in enumerate(words):
          Word2int[word] = i
     sentences = []
For sentence in corpus:
   sentence.append(sentence.split())
WINDOW_SIZE= 3
Data = []
```

```
For sentence in sentences:
     For idx, word in enumerate(sentence):
         For neighbor in sentence[max(idx – WINDOW_SIZE, 0):min
(idx + WINDOW_SIZE, len(sentence)) + 1] :
              If neighbor!=word:
                  data.append([word, neighbor])
```

Generating one-hot encoding: Integers are represented by a single hot encoding. Because vectors are integer mappings of data. Each resource value is identified by a single hot encoding, and mathematical procedures are carried out using just one hot encoding.

```
ONE_HOT_DIM = len(words)
# Function to convert numbers to one hot vectors
def to_one_hot_encoding(data_point_index):
     one_hot_encoding = np.zeros(ONE_HOT_DIM)
     one_hot_encoding[data_point_index] = 1
     return one_hot_encoding
  X=[]      # for the input word
  Y=[]      # for the target word
 For x, y in zip(df[ ‘input’], df[‘label’]):
     X.append(to_one_hot_encoding(word2int[x]))
     Y.append(to_one_hot_encoding(word2int[y]))
# To covert to numpy arrays
  X_train = np.asarray(X)
  Y_train = np.asarray(Y)
```

Generating vectors: Vectors are data representations used to perform mathematical operations. After 3,000 repetitions, the following snapshot source code prints information. Users can end the training when the minimum loss is reached, but the maximum iterations are greater than 300000, and the number is chosen at random.

```
sess = tf.session()
Init = tf.global_variables_initializer()
Sess.run(init)
Iteration = 300000
For i in range (iteration):
```

```
      Sess.run(train_op, feed_dict={x: X_train, y_train:
      Y_train})
      If i%3000==0:
Print('iteration ' + str(i)+' loss is : ', sess.run(loss,
feed_dict={x: X_train, y_train: Y_train}))
vectors =sess.run(w1 + b1)
#print(vectors)
W2v_df=pd.Dataframe(vectors, vectors, columns = ['x1', 'x2'])
W2v_df['wor'] = words
W2v_df= w2v_df[['word', 'x1', 'x2']]
W2v_df
```

Then acquires the final sequences from this stage, are suitable for training the neural networks. The next stage is to analyze our intent using network topologies as a result of this: For the intent categorization subtask, various network variants were examined. Each of them is briefly described in this section, along with their performance.

4.6. Structure of a Recurrent Neural Network

Each network's applicable network has the same fundamental structure, as seen in Figure 22. The tokenized utterances were delivered to the buried layer (input messages). At this point, the tokenize stage is executed, converting each word (w_1, w_2, w_3, etc.) into a series of vectors (x_1, x_2, x_3, etc.), which each uniquely defines the word's frequency and its state.

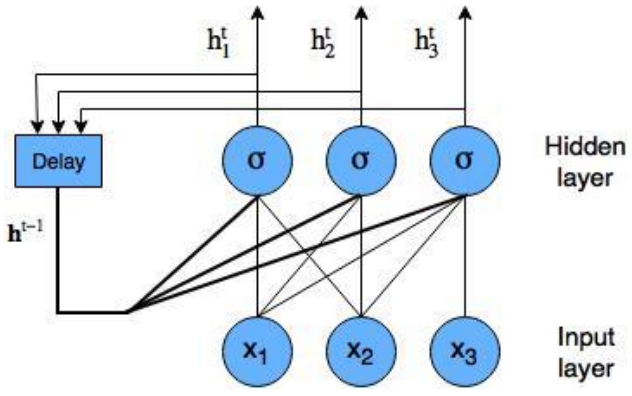


Figure 22 RNN network nodes diagram

Then, as shown in figure 23, an embedding layer transforms those sequences of integers once more into a sequence of real vectors of size 128.

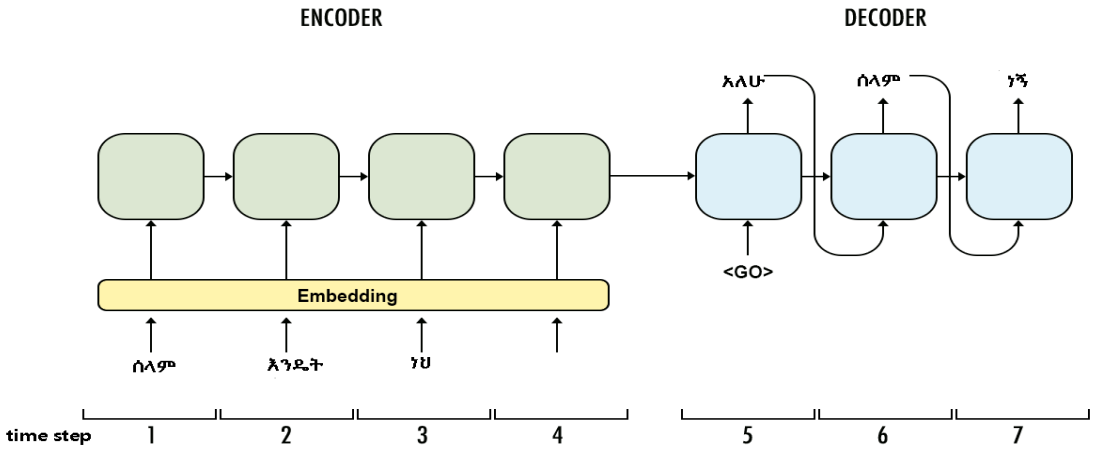


Figure 23 Seq2Seq Embedding inputs

After that, these integer sequences are passed to the RNN substructure. Lastly, using the usual SoftMax function, S (z), the RNN's output is directed via a SoftMax layer

$$S(z)_j \triangleq \frac{e^{z^{t}wj}}{\sum_{i=1}^{m} e^{z^{t}wj}} \tag{4.3}$$

where m is the output size of the SoftMax layer or the number of classes, and w is the weight vector for all connections between the SoftMax layer and the preceding layer. The SoftMax function's intriguing characteristic is that

$$\sum_{j=1}^{m} S(z)_{(j)} = 1 \tag{4.4}$$

And therefore,

$$S(z)(j) \approx P(y = j|z) \tag{4.5}$$

The intent classifier is supposed to output a vector of probability along with its prediction; thus, this is helpful in this situation. The categorical cross-entropy loss is the minimized loss function during training and is provided as:

$$L(\theta) = -\frac{1}{m}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij} \log(\dot{y}_{i,j}(\theta)) \tag{4.6}$$

$y_{i,j}$ 0,1 is 1 if the class of sample i is j and 0 otherwise, and $y_{i,j}$ [0,1] is the projected probability that sample i has label j. The accuracy of the model's predictions is another statistic that is created during training. To train the neural networks, the Adam optimizer was used because it has been proven to give good results in the field of deep learning. It's also computationally efficient, which comes in handy here because there are so many samples to optimize.

4.7. Selection and Implementation of Network Structures

Distinct network structures, which are different RNNs versions by Appling network variants, are shown in this section. In general, network selection for sequence-to-sequence modeling is required to determine which loss, training duration, accuracy, and perplexity were measured during training (see the appendix experiment section). Based on the findings of (Peters, 2018), four networks were tested to find the best fit. In this study, the perplexity and accuracy of the four networks were compared throughout the training period. All the models were built with the optimal network hyperparameters and then trained with the prepared dataset to get the optimal value from each variant.

4.7.1. Results of Single LSTM Network

The single-layer LSTM network structure is the first network structure. Figure 24 depicts the structure of a single LSTM cell neural network. To generate appropriate mathematical functions throughout the network cell structure, the basic method of neural networks in a sequence of data is to take the data, change it to the kind of words in the time stamp, and then modify those input sequences of words to the embedded method.

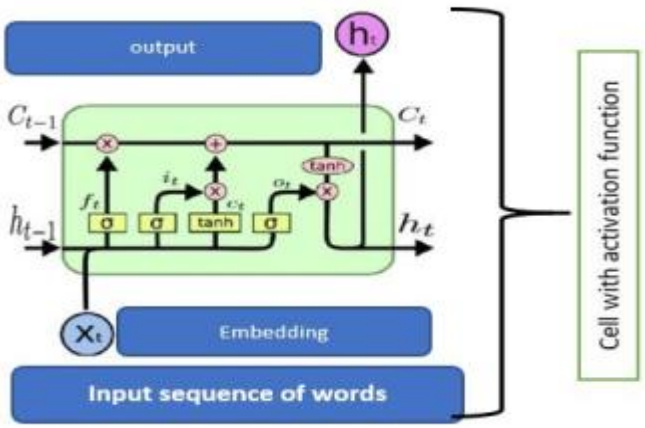


Figure 24 Single LSTM, cell workflow from accepting input to output

Figure 25 shows how the network accuracy develops as the number of training and validation epochs increases. The loss and accuracy of the network display significant variation during training, which makes its properties inconsistent. However, the network learns almost entirely to properly identify the whole training set on the validation set, with a final measured accuracy of 78.53%. The workout is completed in 1680 minutes.

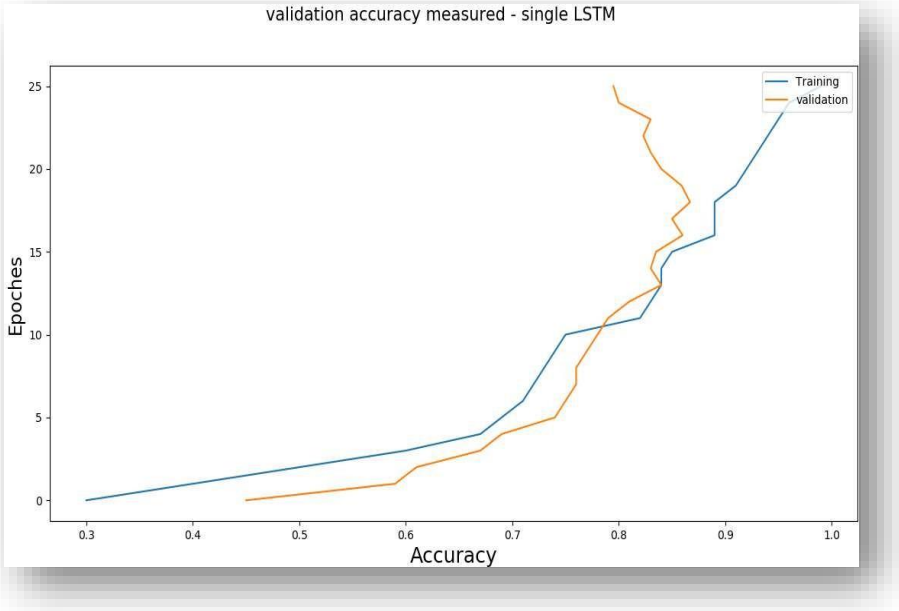


Figure 25 Single LSTM - network accuracy

Figure 25 shows the perplexity measured during single LSTM training. The perplexity has fallen, and the model is capable of strong language modeling. At epoch 25, the observed perplexity in a single LSTM is 0.411. As shown in figure 26, language modeling improves as the epoch increases, although perplexity alone does not indicate training efficacy.

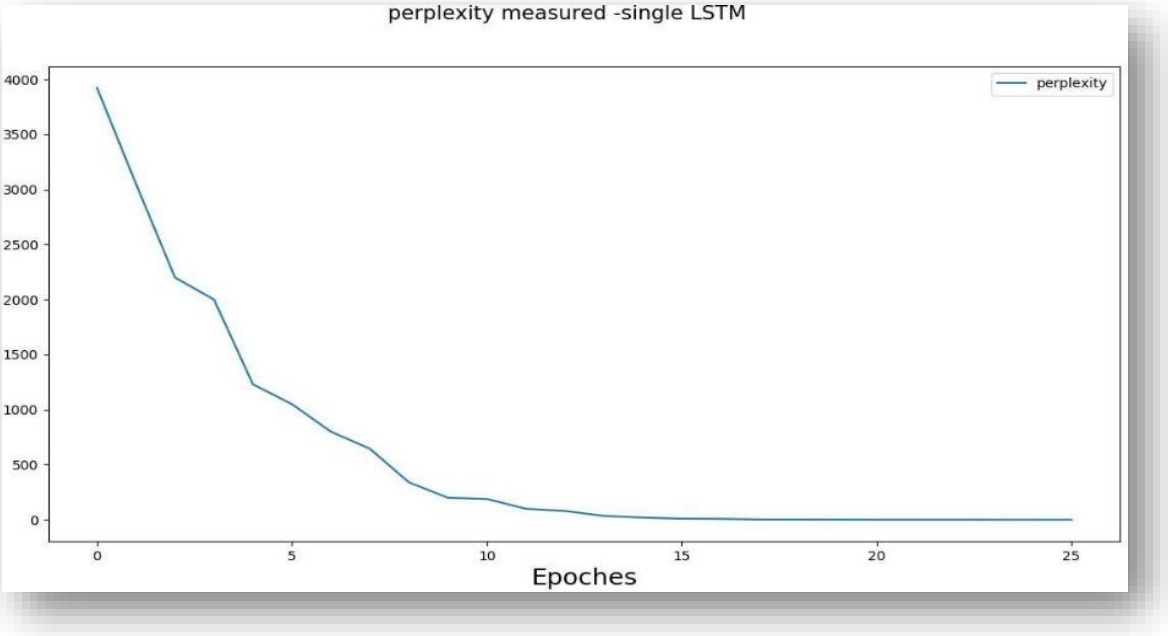


Figure 26 Single LSTM – perplexities measurement

4.7.2. Results of Single GRU Network

GRU-Gated Recurrent Unit is the other recurrent neural network technique. The goal of including GRUs in our experiment is to look at the similarities and differences between

employing LSTM and GRU layers in terms of network consideration. Figure 27 depicts the design of one GRU network. In this network training, a single GRU layer layout yield results that are remarkably similar to those of a single LSTM layer, as shown in figure 28. The network achieves roughly the same end accuracy of 79.05%, as seen in the graph while continuing to display the same unpredictable behavior throughout training. However, the total amount of time needed for complete training is roughly 1584 minutes.

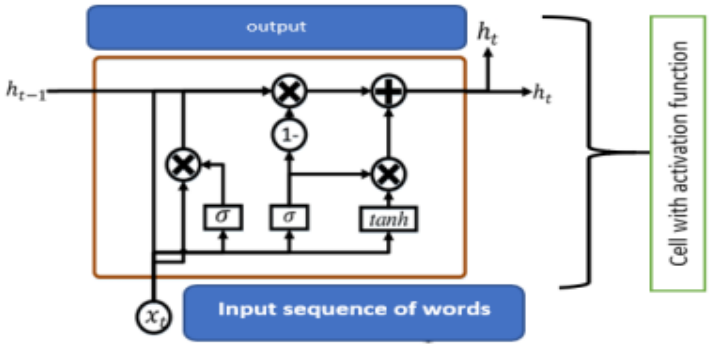


Figure 27 Single-cell GRU network structure [123]

The gated recurrent unit lowers training time in half while keeping almost identical performance. The studied perplexity of language modeling using a single GRU training for 25 epochs is shown in Figure 28. As can be seen, the perplexity of the researched perplexity is slightly higher (0.65), yet it provides results that are similar to those of the LSTM. Single LSTM perplexity outperforms single GRU, whereas single GRU training time and accuracy outperform single LSTM somewhat.

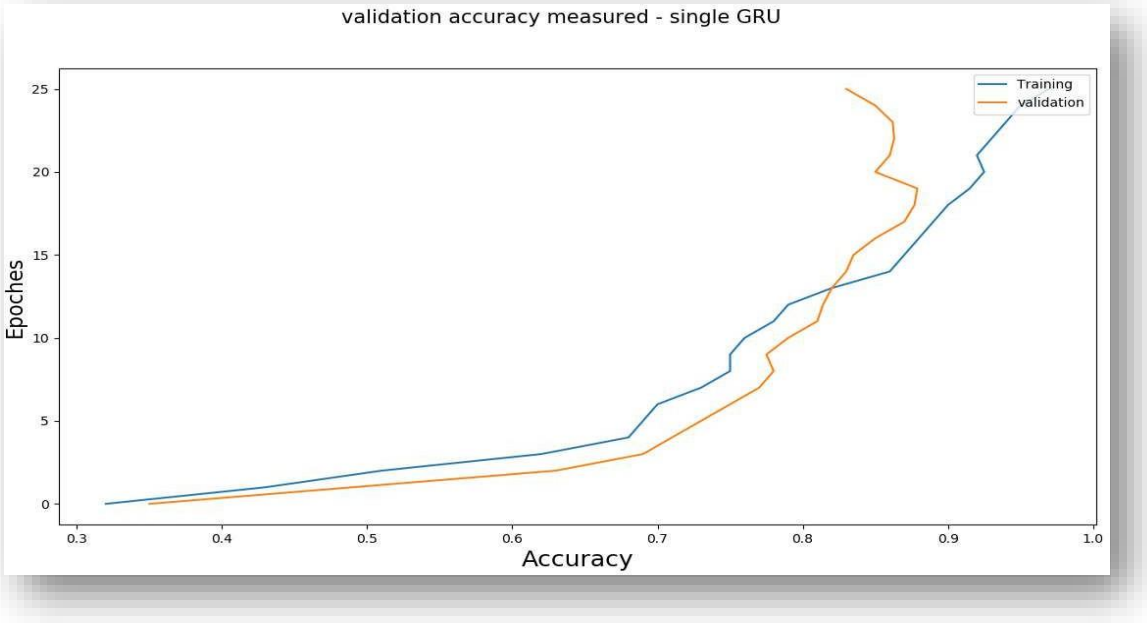


Figure 28 Single GRU training performance

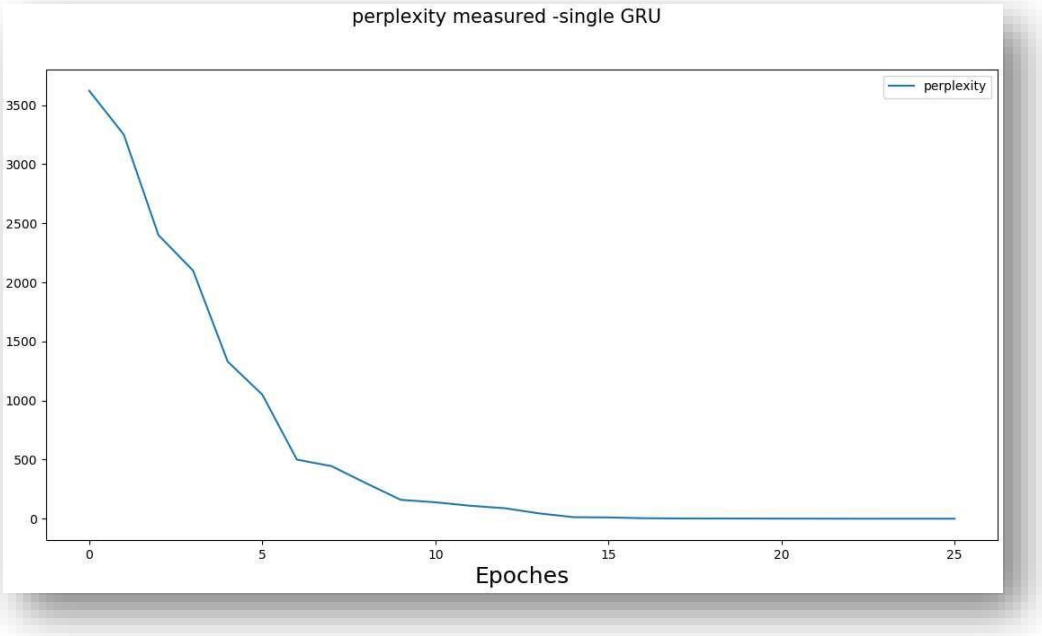


Figure 29 Single GRU – perplexity measures

4.7.3. Results of Transposed or Inverted GRU Network

Comparatively, the LSTM performs poorly, as seen in section 4.7.1, and neither inverting LSTM nor transposing GRU by altering the input sequence can improve training accuracy more than GRU. By modifying the structure (moving places) of the data acquisition, the architecture demonstrates a smaller difference from the one-layer GRU architecture. Since then, transposing the input order has proven to be a simple trial that might potentially alter the performance of certain networks, particularly in machine translation and its subfields (Sutskever et al., 2014).

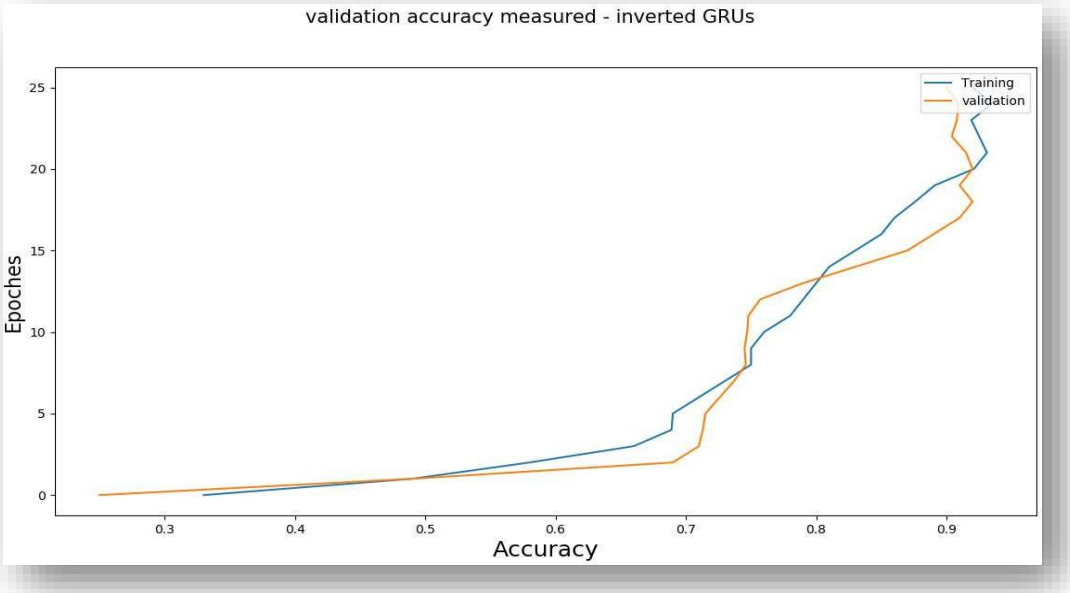


Figure 30 Transposed GRU performance

Figure 30 depicts the progression of its accuracy. Due to decreased jitter for loss and accuracy, this structure is noticeably more stable during training than the prior ones. Despite having the input sequence reversed, the training process only took 1434 minutes and resulted in a 79.86% higher accuracy score. Figure 31 depicts the perplexity of transposed/inverted gated recurrent units which is 0.221; the language model is still good on all three network variations, and perplexity changed insignificantly, even though fractions can cause model training distortion.

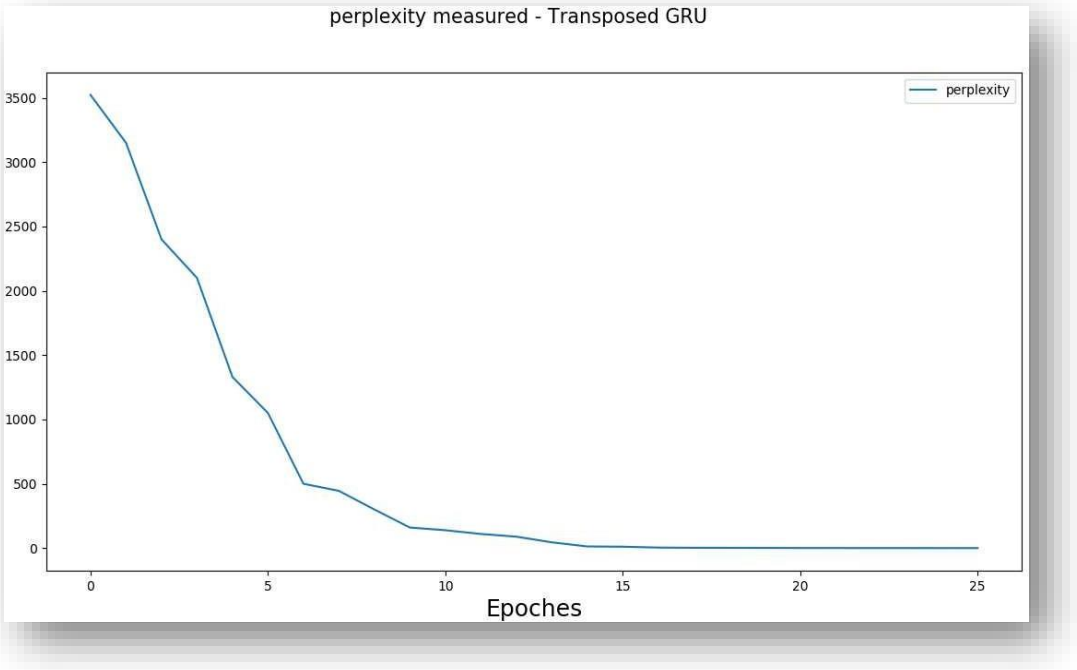


Figure 31 Transposed GRU - perplexity measure

4.7.4. Results of Double GRU Network

We can track important information from the input sequences by stacking two layers of GRUs on top of one another, which is another method for increasing accuracy. This type of network, as seen in Figure 32, does not outperform earlier network variants. When all is said and done, double GRU layers perform worse than single GRU layers. This could be because the cost of training an extra GRU layer outweighs the benefit it is meant to bring. Compared to a straightforward single GRU layer structure, the network's best accuracy is just 75.17%, and training takes 1738 minutes. Figure 33 depicts the perplexity of this network, which is better than the transposed and exhibits improvement when compared to inverted/transposed GRUs. This network has a 0.201 perplexity rating.

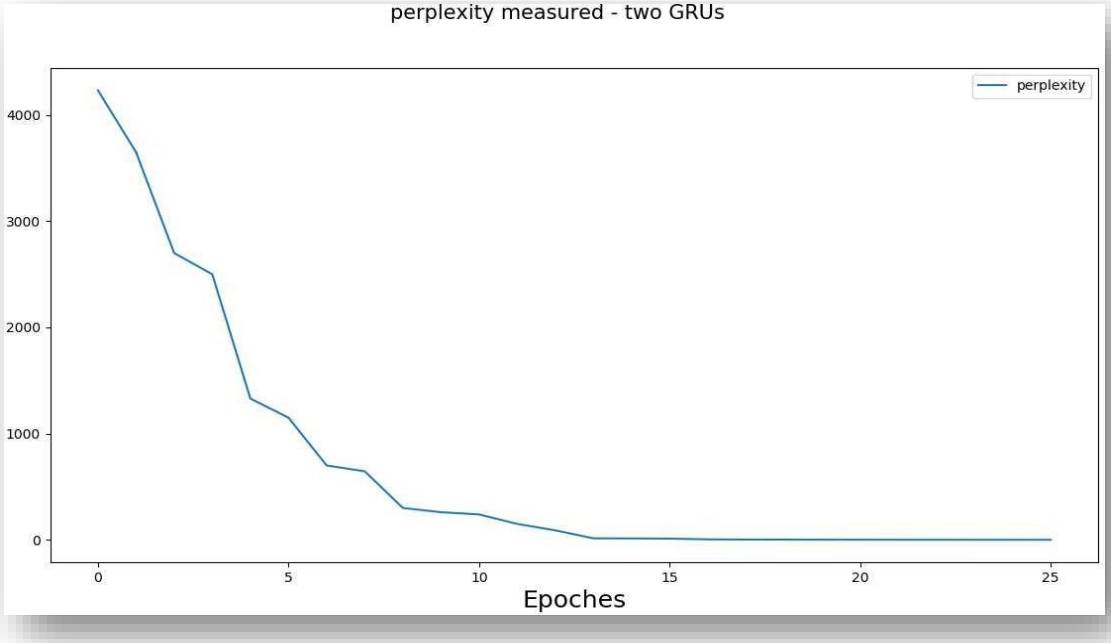


Figure 32 Performance of network - double GRUs

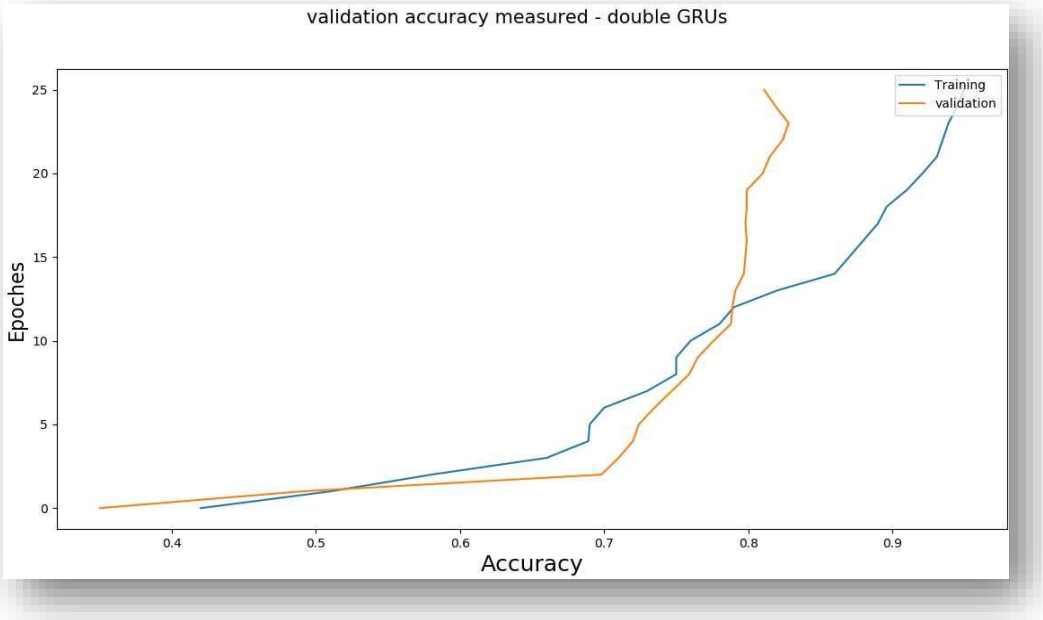


figure 33 Perplexities measured - 2 GRUs

4.7.5. Summary of the Network Variants

Starting with section 4.7.1, the four networks are investigated. Table 6 summarizes the findings while taking loss and training time perplexity into account.

Table 3 Summarizing the network variants

Network Architecture	Validation loss (lower is chosen)	Validation accuracy	Perplexity (lower is chosen)	Time used [min]

		(higher is chosen)		
Single LSTM	0.347	78.53%	0.151	1680
Single GRU	0.423	79.05%	0.191	1584
Transposed GRU	0.345	79.86%	0.221	1434
Two GRU layers	0.494	75.17%	0.201	1738

Viewing the properties of each network based on its visualization is critical to choosing the ideal network architecture. According to the validation loss and accuracy validation, a transposed GRU is preferred, a single LSTM is preferred due to perplexity, and the time required to fully train a transposed GRU is preferred. Three times, transposed GRU were chosen, however, perplexity may be reduced by increasing the training epoch and the training accuracy of the validation set, and time is important when choosing a network. As a result, just a **Transposed GRU network** is chosen. Because it has a high level of accuracy and a quick learning rate.

The effects of a growing number of epochs are now thoroughly explored and stated. For optimal epoch selection, the neural network is trained for 80 epochs to ensure that the most favorable training progress does not lie from the preceding experiment of 25 epochs. And the first 30 epochs are employed because, after that, the minor change in perplexity, learning rate, loss, and accuracy is explored.

4.8 Setting Threshold Value

The chatbots may not answer any questions users ask. In this study, the closed domain chatbot model is designed only for a mental health consultancy. So, there may be inappropriate responses if the user asks the questions out of the domain. Before the user utterance is provided, it should be clarified whether or not the converse is readily available in the actual model. The representative inquiries in the corpus can be expressed in several methods by various users, and for effective messages based on the psycho bot lexicon, the semantics of each user word will converge into a single word and a reply. This is done by employing a custom Amharic Word2Vec for user-inputted unknown terms. If Word2Vec is unable to locate a matching, related context word, sequence-to-sequence modeling is used to calculate the measurement based on the given set of rules. When a user requests unknown information, for instance, rules for unknown information are returned to the user. (i.e. እባክዎትን ግልጽ ያድርጉልኝ፣ አልገባኝም በጣም ይቅርታ ፣ ይቅርታ ልረዳዎት አልቻልኩም በደምብ ብያብራሩልኝ እኔም በደምብ እመልሳለው፣...) The results of the experiment are presented in the following chapter, along with a discussion of the findings.

4.9 Saving the Model for Future Use

No further training in the model is required once the best model has been identified. The model can be saved and modified so that it can answer users' questions quickly. Python objects are saved into files using pickle. Using the Pickle function in Python, a Python object can be transformed into a byte stream and saved to a file. We can serialize and deserialize a Python object type using it.

```
self.model.saver.saver.save(sess, os.path.join(result_dir,
“basic”),
global_step=train_epoch)
summary_writer.close()
pickle.dump(model, open('model.pkl', 'wb'))
```

CHAPTER FIVE
RESULTS AND DISCUSSION

This chapter describes the model's overall outcomes based on quantitative performance measures as well as its performance in a realistic context. The findings were used to evaluate the research questions.

5.1 Deep Learning Algorithms and Their Behavior

The amount of data that is provided to deep learning algorithms is crucial (Bhagwat, 2018). Deep learning algorithms use an approach that allows them to learn from a massive amount of data. Deep learning algorithms extract sequential features from those datasets using deep learning methods. This is because the amount of data required to develop Amharic task-oriented natural speech is ineffective. Deep learning algorithms, on the other hand, are inadequate at handling and tracking discussions. If someone asks about a kidney or heart disease that isn't covered in this thesis, the deep learning algorithm may respond with an irrelevant response. Again, the issue is that the researcher of this work did not use an Amharic model as a reference (the nature of language impacts the smoothness of dialogue, for example, to drive the rule we first envision the nature of the data pattern and then defining rules is simple to achieve). The challenge in this study is the inability to retrieve useful information for confined data sets, especially in task-oriented systems, according to a review of related works in (chapter 2, section 2.5). Unavailability of Amharic-language conversational bots Aside from the scientific issue of retrieving information, consulting concerning mental health is time and money-consuming. Conversational bots come in a variety of implementation styles, and most artificial bots are data-driven (the nature of information varies the conversation characteristics and architecture)

5.2 Testing Result of the Model

5.2.1 Using Stratified Cross-Validation

Different data processing is done while the system model is implemented, as stated in the previous chapter three. In chapter four, the overall accuracy of the training model is displayed. Using cross-validation as discussed in the methodology section, is applied and a stratified cross-validation type was selected for validation. After partitioning the data into training, validation, and testing sets, Table 7 displays, the accuracy captured during training and testing in five (5- fold) different data distributions.

Table 4 Data distribution description with its performance

Data Distribution Based on Training, Validation, and Testing)	Model Training Accuracy	Model Testing Accuracy	Description
(50%,5%,45%)	85.63%	57.22%	Over-fitting, and increasing the validation set can solve this problem
(60%,10%,40%)	74.94%	66.31%	Still, over-fitting is there
(60%,20%,30%)	73.05%	65.29%	Accuracy diminution
(70%,10%,20%)	81.88%	79.62%	Achieves well than others
(80%,10%,20%)	89.65%	73.77%	The model generalizes the training, and high over-fitting is recorded

When the 45k utterance data is split into 70%, 10%, and 20%, which is 31.5k, 4.5k, and 9k data for training, validation, and testing, the performance is better than other data distributions. Because the documented difference between modeling, training, accuracy, and testing accuracy is smaller in this training distribution than in others, there is less overfitting.

By incorporating a large amount of data, the model's performance can be improved. Only Amharic word vectors and a few rules incorporated on top of the sequence-to-sequence model can help this task-oriented model with data about mental health information. As a result, it necessitates the use of very advanced Amharic Word2Vec, which has a vast amount of Amharic knowledge. Even though the model's accuracy is less than 80%, the data preparation is done by hand, and it is not too open to everyone's opinion, because the way data is presented might affect the model's performance.

When compared to a model with custom-rule embedded, the model performance without embedded rules is 71.56%, which is 79.86%. This demonstrates how integrating rules can improve the model's performance. The below table describes the accuracy of each network variant with a custom rule embedded as it yields better accuracy.

Table 5 Summary of each network under stratified 5-fold cross-validation

LSTM			**GRU**		**Transposed GRU**		**Doubled GRU**	
Stratified CV	Accuracy	Lose	Accuracy	Lose	Accuracy	Lose	Accuracy	Lose
5 – Fold	78.53%	0.170	79.05%	0.178	79.86%	0.208	78.17%	0.218

The ensemble architecture, which is built on seq2seq modeling and uses Word2Vec to manage the conversation's contextual flow, has a 79.62% accuracy. The distribution of each intent in

the test data is not equal, therefore focusing solely on accuracy may not accurately reflect the model's performance.

5.2.2 Using the F-1 Score

The other validation technique is the F-1 score, the f-1 score (i.e., good for unbalanced data distribution) is then employed by evaluating the distribution of the test data for intentions. The f-1 score of each model is described below. Table 6 Summary of each network under F-1 score evaluation

	LSTM	**GRU**	**Transposed GRU**	**Doubled GRU**
F-1 score	0.7580	0.7856	0.8822	0.7517

For the proposed model, we get the same f- 1 score regardless of the data provided for different classes. The below table describes the confusion matrix of the psycho bot model with its F-1 score value. We calculated the value of the F-1 score as follows.

`Precision = True Positive / (True positive + False Positive)

Average precision of the model = 0.7995

Recall = True Positive / (True Positive + False Negative)

Average Recall of the model = 0.9841

F-1 score = 2 x (precision x recall) / (precision + recall)

F-1 score = 2 x (0. 7995x 0. 9841) / (0. 7995+ 0. 9841)

F-1 score = 2 (0.78679) / (1.7836)

F-1 score = 1.57358 / 1. 7836

F-1 score = 0.8822

Table 7 Confusion matrix of the proposed model

The model confusion is summarized in table 11. The confusion indicates an inability to correctly identify the given intent; according to table 12, the majority of the confusion occurs due to related mental disorders natures, and the model's response can be incorrectly responded to, however, some confusions arise due to non-related mental disorders, which is not studied

The confusion matrix and their value for each the disorder type								
ባይፖላር ወይም ጥንድ ዋልታ መታወክ	**582** **10.97%**	132 2.03%	104 1.68%	37 0.67%	11 0.20%	22 0.40%	60 1.03%	**948** **64.22%** **35.78%**
ስነ ምግባር መታወክ	66 1.10%	**822** **14.80%**	11 0.18%	26 0.43%	52 .87%	44 .73%	90 1.5%	**1111** **73.98%** **26.02%**
ጭንቀት መታወክ	27 0.45%	2 0.03%	**841** **14.12%**	21 0.35%	58 0.97%	14 0.23%	47 0.78%	**1010** **83.23%** **16.77%**
እንቅልፍ መዛባት መታወክ	9 0.15%	6 0.10%	117 1.95%	**537** **7.95%**	13 0.22%	26 0.43%	64 1.07%	**772** **70.56%** **39.44%**
የመገለል ስሜት መታወክ	6 0.30%	2 0.03%	13 0.22%	34 0.57%	**612** **12.30%**	13 0.22%	82 1.37%	**762** **81.38%** **18.62%**
የመዘንጋት መታወክ	38 0.63%	29 0.48%	31 0.52%	48 0.80%	27 0.45%	**575** **10.58%**	33 0.55%	**781** **76.82%** **23.18%**
ድብርት መታወክ	19 0.25%	20 0.40%	11 0.18%	44 0.72%	70 1.18%	39 0.65%	**408** **8.90%**	**611** **65.78%** **34.18%**
አደንዛዝ እጽ አጠቃቀም መታወክ	**753** **78.65%** **21.35%**	**1007** **84.63%** **15.37%**	**1125** **72.76%** **27.24%**	**749** **73.70%** **26.30%**	**845** **74.43%** **25.57%**	**735** **76.23%** **23.77%**	**786** **53.91%** **46.09%**	**6000** **79.62%** **20.38%**

in this thesis.

Table 8 Confusion matrix description of the psycho bot model

Intent-class	**Confusion recorded**
የባይፖላር መታወክ / Bi-polar disorder	Highly confused in የመገለል ስሜት መታወክ / Dissociative Disorder, because the intent classes have the similarity of symptoms and treatment options.
የስነምግባር መታወክ / Conduct Disorder	Confused with የአደንዛዥ ዕፅ አጠቃቀም መታወክ / Substance - Use Disorder due to shared treatment options recorded.
የጭንቀት መታወክ / Stress or Anxiety Disorder	Typically confused with የድብርት መታወክ/ Depression Disorder due to shared signs recorded.

የእንቅልፍ መዛባት መታወክ / Sleep-Awake Disorder	Confused about የጭንቀት መታወክ / Stress or anxiety disorder
የመገለል ስሜት መታወክ / Dissociative Disorder	Highly confused in የባይፖላር መታወክ / Bi-polar disorder, የመገለል ስሜት መታወክ / Dissociative Disorder, because of the similarity in symptoms recorded in each intent class.
የስኪዝኖፈርንያ መታወክ / Schizophrenia	Slightly confused about, የመገለል ስሜት መታወክ / Dissociative Disorder and የእንቅልፍ መዛባት መታወክ / Sleep-Awake Disorder
የድብርት መታወክ/ Depression Disorder	Confused about የጭንቀት መታወክ / Stress or anxiety disorder
የአደንዛዥ ዕፅ አጠቃቀም መታወክ / Substance -use Disorder	Typically Confused with የስነምግባር መታወክ / Conduct Disorder due to shared treatment options recorded.

The confusion arises as a result of the resemblance of the intentions. According to table 11, in the confusion matrix, confusion arises as a result of the requirement for a profound expression of such intents to properly convey the model. Because there are more related intents indicated in table 11, based on the confusion matrix, those intents data require more unique expressions for the model to learn and retrieve with better performance than previously reported performance.

5.2.3 Using the Two Types of Word2vec Model

The word2vec model has two approaches, as explained in section 2.3.1: the Skip-gram approach and the Continuous bag of words. The Skip-gram method predicts its neighbors using the current words, whereas CBOW predicts the target word using its neighbors. Table 12 compares these two approaches when applied to proposed deep learning models.

Table 9 Comparison of the 2 word2vec models

	LSTM		**GRU**		**Transposed GRU**		**Double GRU**	
Word2vec type	Accuracy	Lose	Accuracy	Lose	Accuracy	Lose	Accuracy	Lose
Skip gram	78.53%	0.151	79.05%	0.191	79.86%	0.221	75.17%	0.201
CBOW	77.69%	0.1685	79.01%	0.188	79.84%	0.219	75.14%	0.170

The Skip-gram model outperforms the CBOW among all proposed deep learning models, as shown in the table above. Aside from performance, the skip-gram model is recommended

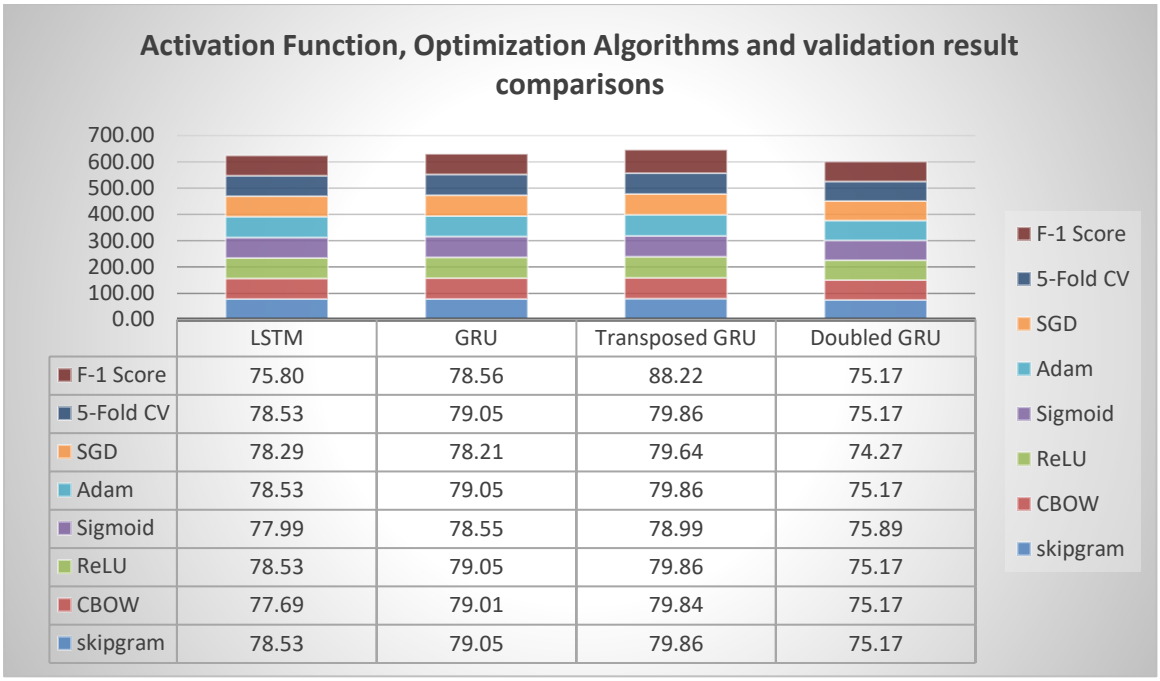
because it better captures the semantic relationship between words, whereas the CBOW model only considers morphologically related words placed in nearby space.

5.3 Activation Function and Optimization Algorithms

The model's performance is affected by the variation in activation functions. The ReLU and Sigmoid activation functions are widely used in neural networks and have demonstrated success in numerous previous studies (Gupta, 2020; Liang et al., 2017; Vamsi et al., 2020). The performance of two activation functions when used in the proposed deep learning model is shown in Table 13. In addition, the study compared the performance of two optimization algorithms, Adam and Stochastic gradient descent (SGD), with the proposed deep learning model. Optimizers are algorithms that change the weights and rates of learning of hidden layer neurons to reduce network losses. Although Adam optimizers achieve the best results most of the time, SGD occasionally outperforms Adam's optimization algorithm (Vamsi et al., 2020). A comparison of the two methods has been carried out in this study, as shown in below. Table 10 Summary on activation functions and optimizers

LSTM			**GRU**		**Transposed GRU**		**Doubled GRU**	
Optimization algorithms								
	Accuracy	Lose	Accuracy	Lose	Accuracy	Lose	Accuracy	Lose
Adam	78.53%	0.151	79.05%	0.191	79.86%	0.221	75.17%	0.201
SGD	78.29%	0.155	78.21%	0.179	79.64%	0.197	74.27%	0.198
Activation functions								
ReLU	78.53%	0.151	79.05%	0.191	79.86%	0.221	75.17%	0.201
sigmoid	77.99%	0.199	78.55%	0.188	78.99%	0.198	74.89%	0.211

Tables 12 and 13 show the outcomes of all proposed models. Because the study used Skip gram word2vec word embedding and achieved 79.86% accuracy, a Recurrent neural network requires the word embedding layer when working with text data. To address the issue of model overfitting, this study used L2 regularization. The chart below depicts the performance of all deep learning models proposed for designing a mental health consultant chatbot.



	LSTM	GRU	Transposed GRU	Doubled GRU
F-1 Score	75.80	78.56	88.22	75.17
5-Fold CV	78.53	79.05	79.86	75.17
SGD	78.29	78.21	79.64	74.27
Adam	78.53	79.05	79.86	75.17
Sigmoid	77.99	78.55	78.99	75.89
ReLU	78.53	79.05	79.86	75.17
CBOW	77.69	79.01	79.84	75.17
skipgram	78.53	79.05	79.86	75.17

Figure 34 The proposed models' summary comparison chart description

5.4. Hyperparameter Tunning

The grid search hyperparameter tuning method is used in this study to find the best hyperparameter for the proposed neural network models. Because it produces accurate results with small datasets, it greatly slows down computation and is very expensive with large datasets and high dimensions. A random search is preferable instead. A grid search will produce more accurate results than a randomized search with a small dataset, but it will reduce computation time with a large dataset. As a result, the grid search hyperparameter tuning method looks for the best possible combination of hyperparameters. The study discovered the optimal hyperparameter for all proposed models, including Epochs (20,50,30), Batch size (32, 100, 64, 128), number of neurons (512, 256, 200, 300), Activation function (ReLU, sigmoid), Optimizers (Adam, Stochastic Gradient Descent), Learning rate (0.1, 0.001, 0.0001), Hidden Layers (1, 2, 3). To find the best combination of hyperparameters, the study employs Grid search hyperparameter tuning. The optimal hyperparameters for all of the proposed models were then determined, as shown below.

Table 11 Summary of hyperparameter for all network variants

Models	**Epochs**	**Batch size**	**Activation**	**Optimizers**	**Learning rate**	**Hidden Layers**
LSTM	40	128	ReLU	Adam	0.001	1
Single GRU	45	256	ReLU	Adam	0.001	1
Transposed GRU	45	256	ReLU	Adam	0.001	2

Double GRU	50	256	ReLU	Adam	0.001	2

5.5 Weight Initializers' Effects on the Proposed Model

To avoid a sudden increase or decrease, i.e. not happening to vanish or exploding gradients, in the output of the layers, the weight initializers technique is employed (Vamsi et al., 2020). Even though this method has improved performance in other studies, such as the study mentioned above, it only marginally improves the performance of the models when applied to our suggested model. It does not, however, demonstrate the detrimental effect on the model's performance. This suggests that in the experiment the study conducted using our dataset, the weight initializers approach did not significantly alter the results.

5.6 Human Evaluation Testing and its Approach

As previously noted, when it comes to conversational bots, performance indicators do not reveal the whole story (Davis & Venkatesh, 2004). As a result, a manual method of testing is designed from the ground up to ensure that the chatbot exhibits the desired behavior. Like a direct messaging program, the goal is to be able to type messages straight into a terminal or other keyboard input method.

As explained in section 3.9, the study assessed the presented chatbot model through human evaluation to measure the accurateness and acceptability of the chatbot model. To reach this, the study deployed the model on 20 computers and assessed the chatbot via the google Colab environment in which the model is reachable to the end-users. And then, the study selects some mental healthcare experts as they use the chatbot, and based on the user's feedback, the model is evaluated. Based on this, the user fills out a questionnaire, which is displayed in Appendix VII. The naturalness of conversation, relevancy of responses, use of the model, response time, and way of user query understanding and the system replying based on the recommendation of (Klopfenstein et al., 2017) There are five possible values for this attribute: very-low (0), low (1), good (2), very good (3), and superb (4). Table 16 depicted the overall results recorded during user acceptance testing.

Table 12 Human evaluation testing table description

Attributes	**A score of 20 – persons recorded**						
	Very low	**Low**	**Good**	**Very good**	**Excellent**	**Average**	**Total in %**
Naturalness			1	2	17	3.8	76%
Responses relevance			1	3	17	3.8	76%

Way of user query understanding and the system replying			1	3	16	3.75	75%
Usage of the model			1	1	17	3.8	76%
Response time				2	18	3.9	78%
Average user acceptance testing score						3.81	76.2%

The average user acceptance testing score is 3.81 out of the five attributes provided, implying that 76.2% of users approve of the conversational bot naturalness created in this study. Relevant results are retrieved 76% of the time from the five testing attributes. The way of user queries understanding and the system replying is scored 75% of the time, and the model correctly identifies the naturalness of the Amharic language is scored 76% of the time. The model is used (usage of the model) 76% of the time, and the response time is appealing 73% of the time.

The evaluators indicate that the model helps give and supports users by giving mental health information to the clients when necessary and for general knowledge and that if the model works by voice commands, which is voice conversations (since working in it continues to support many more users than the present system), the evaluators also suggest that supportive answers be incorporated more than the present system.

5.7 Comparisons with Existing Models

Applying data from currently available relevant work models, such as the dynamic memory network model (43.31% accuracy), the question-answering model (34.27% accuracy), and the Keras sequence to sequence model (54 %). The model comparisons are summarized in Table 17. The application of intent management for data, use of the Word2Vec model, perplexity management of the model, and logical preparation of rules are the major reasons for the excellent accuracy of the ensemble architecture of the model when compared to current models.

Table 13 Comparisons of models, with their performance

Model	Recorded Accuracy Performance	Description
Ask Me anything: dynamic memory network model (Kumar et al., 2016)	43.31%	- Because the model is highly data-intensive, restricted data (Knowledge base) restricts the model's accuracy.
End-to-End question answering model (Meng & Huang, 2018)	34.27%	- The model is intended to answer simple queries and solve problems. - Intents are not taken into account.

Keras sequence to sequence model for task-oriented systems (Sutskever et al., 2014)	54.33%	- Dependent on a large data collection, feature learning requires several expressions for a single intent to comprehend what it means.
The proposed psychobot model (with seq2seq modeling, word2vec and rules embedded)	79.62%	- In contrast to pure seq2seq modeling, this model minimizes many expressions for a single intent.
The proposed psychobot model (built on seq2seq modeling, and Word2Vec) without embedded rules	69.56%	- By establishing rules in the discussion, open-ended responses can be prevented; accuracy rules should be put into practice based on the model.
The proposed psychobot model (built on seq2seq modeling, by removing Word2Vec and embedding rules)	59.81%	- This is based on sequence-to-sequence modeling employing rules, and Word2Vec allows for the discovery of contextual meanings during the conversation, however, the model accuracy is low.
The generative approach built on using seq2seq modeling (excluding retrieval approach based, Word2Vec and embedding rules)	57.89%	This is the generative modeling approach and on sequence-to-sequence modeling but without a retrieval approach based on employing custom rules, and Word2Vec allows for the discovery of contextual meanings during the conversation, however, the model accuracy is low.

This work aimed to develop a mental health assistant by incorporating the DSM-5, as well as other related and relevant documents for the most commonly observed mental disorders in Ethiopia, as well as Amharic word2vec and embedded rules on top of sequence-to-sequence modeling for improved performance. This study's results are superior to those of previous studies. In general, the study suggests looking at this work approach to creating successful task-oriented conversational bots with limited knowledge sources.

5.8. Research Question Discussion

The goal of this study is to develop and apply an apprentice bot model for psychological client therapy, as we covered in section 1.5. In addition to accomplishing this goal, the study was able to provide an answer to the inquiry posed in section 1.4. We attempted to discuss how these research questions were addressed in this part. We applied various methods to provide answers to those questions. Let's examine the methods used to respond to the study questions and how the suggested system handled each one individually.

RQ1. "Which deep learning algorithm is more suitable to design an ensemble consultant chatbot model for psychological client therapy?". To answer this research question, the primary dataset was collected from the DSM-5th edition manual book and different sources and trained the proposed deep learning models, as explained in section 4.3. The study set different criteria to determine the deep learning model typically employed in modern chatbot development, as mentioned in section 3.7. Based on those criteria, the study then presented four deep learning models such as LSTM, single GRU, inverted GRU, and doubled GRU models. In an experiment we performed among these proposed models, the inverted GRU model outperformed the others. As a result, we chose the inverted GRU model as the best model for designing the mental health consultant chatbot.

RQ2. "Does the Amharic custom rule embedded in the proposed model affect the conversation flaw?" To show the contribution of embedding custom-prepared rules, as discussed in section 3.2.3, several guidelines or templates are incorporated on top of the proposed model. By establishing guidelines for the dialogue, certain incorrect responses can be eliminated, and the flexibility during conversation flaws can be handled easily. The study examined the role of the custom-prepared rules during the experiment by including them and not including them in the proposed model. Hence, the accuracy while embedding into the proposed model was 79.62 % and 69.56 % respectively, as discussed in section 5.7 of this chapter.

RQ3. "Does the inclusion of the Amharic word2vec model change how accurate and pertinent the proposed model is?". Amharic word2vec word embedding techniques have been employed to support the semantic relations between words. i.e., to get comparable words in meaning. As users' utterances can vary, even if they have the same objective, the model should reply to each user's utterance with probably the same meaning response. About 120K sentences have been collected to construct the Amharic word2vec model. To show the influence of the word2vec model on the proposed model, the study examined both with and without the word2vec model. As discussed in section 5.7 of this chapter, the accuracy of the proposed model with the support of Amharic word2vec was recorded at 79.62%, and excluding the Amharic word2vec model, 59.81% accuracy has been noted. As a result, we concluded that using the word2vec model for semantic analysis of words provides greater accuracy than not using the word2vec word embedding technique.

CHAPTER SIX

CONCLUSION AND FUTURE WORK

This chapter regurgitates what has already been practiced by summarizing the content of all preceding sections. It provides a full analysis of the final statement as well as a signal for future study options.

6.1. Conclusion

This study utilizes an ensemble architecture to bridge the gap between deep learning data flexibility and hand-crafted data preparation. Since the same context but a different list of features is given to make neural network learning, and deeply learned models require a large amount of data to extract features for extracting knowledge and remembering the contextual framework, the ensemble architecture is used to employ the most significant characteristics of sequence-to-sequence modeling, word-to-vector conversion, and rule generation from a dataset and embedding it on top of the sequence. To make conversations and obtain suitable replies from the model, sequence-to-sequence modeling is used. To gain answers to the research questions, the model is tested using several metrics.

The study employs many measurements to validate the flexibility of dialog, such as preparing custom rules and the Amharic Word2Vec model, which offers users dynamic attributes of utterances. This method improves the model's ability to respond to specific utterances. Finally, the study assesses the model's correctness (with rules embedded, without rules embedded, including Amharic word vector, excluding Amharic word vector, the generative approach with the given data set, and by combining those qualities) as well as end-user acceptance testing. According to user acceptability testing, the model's naturalness is 76%, indicating that it can process conversations in the Amharic language, relevance is 76%, representing as the recorded responses are returned to the users; the model's usage is 76%. Finally, the model's response time is, which receives a score of 78%, demonstrating that interactions with the psycho bot are simple, and the way of user queries understanding and the system replying is scored 75%, which indicates there is smooth interaction in between.

If there are new vocabulary sets available, Word2Vec is a supporting module that allows the model to deliver semantically related words to the user's utterance. The significance of the Amharic Word2Vec model is assessed. Without using the Amharic word2vec, the accuracy is 59.81%. As a result, the implementation of word2vec plays a significant role in controlling the dynamic conversation. The window size has an impact on Word2Vec implementations, as it

influences the running duration and navigation of neighbors for context capture of words. It also necessitates a large amount of Amharic sentence data to drive the most contextually related words from the Word2Vec model. In our case, we use three window sizes to properly control the model by considering running time and context. Increasing the size of the window increases the model's running time and captures more neighbor's contextual data, so increasing the window size increases the model's running time and captures more neighbor's contextual data. The skip-gram word2vec model has best perform than CBOW during the implementation of the model.

Looking back on the study, it can be said that the solution developed is adequate and allows for simple deployment and development if alterations are needed. Almost any system that can support a Python environment can install it. Additionally, it offers a view of other jobs, help, and user problems that are connected to the issue in this work. Although it is not a perfect solution, given the limitations, it works well in practice.

Future system optimizers should be aware of this since there may be a few treatable situations when the conversational agents do not perform as well as they should. The user's conversation from conversation systems with huge amounts of data can be added to this work, and the model's significance can be raised by using multi-turn responses with massive amounts of information and speech conversation support, even though there is no absolute curative implementation. The first problem in building conversational bots is comprehending how data flows. The other obstacle is a lack of resources and selecting the appropriate method to handle the problem because data is the backbone of conversational bots' development and strategies are developed based on data formation. This work has a flaw that needs to be addressed because it only supports one best turn for a single phrase when it should include multi-turn with voice support communications. After 4 different deep learning models (LSTM, single GRU, inverted GRU, and double GRU) has investigated, different hyperparameter tuning mechanisms for model optimization and varieties of deep learning validation techniques (cross-validation, F-1 score, accuracy, loss, perplexity) have been employed, this study generally has found that the Transposed GRU model is the best performing model better than other proposed deep learning models with 79.86% accuracy value.

6.2. Future Work

The knowledge base is built using a hand-craft or manual approach to data set preparation, and a pattern for preparing utterance pairs may be forgotten during this time. Multiple answers that support the facts and communicate what users are saying are beyond the scope of this thesis. Conversational agents rely on good distributed features of datasets in this situation, and automatic question-generating efforts are advised to exist. The planned model is not included for user utterances that require two associated responses. The following tasks should be completed as future work to improve the model. *Standardized corpora preparation;* There are currently no standard corpora in Amharic, thus researchers must create unique corpora to meet their specific needs. However, if a common Amharic corpus is established and made public, AI researchers will be able to simply tag, tokenize, and clean their data. *Amharic question auto-generator;* receives numerous paragraphs and generates multiple questions for each paragraph or sentence, resulting in the creation of an auto-generated knowledge base. *In an improved and publicly available Amharic word vector model;* words are the fundamental building blocks of languages. As a result, the model may be used to extract semantic similarities between words, forecast words, and many other useful applications. Even if confined corpus word vectors exist, they are useless for other studies. As a result, a word vector model that includes all Amharic words should be implemented. *Multi-turn responses in an attention-based model;* multi-turn responses can be required for supporting facts, and building such a model improves the job of conversational agents. *Implementation of speech conversation (Voice-based);* Speech can generate appropriate discussions and serve multiple aspects of life. Conversations can be established regardless of whether or not someone reads or writes.

REFERENCES

[PDF] "Hey Siri, Do You Understand Me?": Virtual Assistants and Dysarthria | Semantic Scholar. (n.d.). Retrieved October 30, 2021, from https://www.semanticscholar.org/paper/%22Hey-Siri%2C-Do-You-Understand-Me%22%3A-Virtual-and-Ballati-Corno/b423cecb9022a8cda5320884f819e914a384be08

A., S., & John, D. (2015). Survey on Chatbot Design Techniques in Speech Conversation Systems. *International Journal of Advanced Computer Science and Applications*, *6*(7). https://doi.org/10.14569/ijacsa.2015.060712

Abebawu Eshetu. (n.d.). *Text Preprocessing for Amharic*. https://abe2g.github.io/am-preprocess.html

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., & Le, Q. V. (2020). *Towards a Human-like Open-Domain Chatbot*. http://arxiv.org/abs/2001.09977

Aiken, M. (2019). An Updated Evaluation of Google Translate Accuracy. *Studies in Linguistics and Literature*, *3*(3), p253. https://doi.org/10.22158/sll.v3n3p253

Al Shamsi, H., Almutairi, A. G., Al Mashrafi, S., & Al Kalbani, T. (2020). Implications of language barriers for healthcare: A systematic review. *Oman Medical Journal*, *35*(2), 1–7. https://doi.org/10.5001/OMJ.2020.40

Alem, A., Kebede, D., Fekadu, A., Shibre, T., Fekadu, D., Beyero, T., Medhin, G., Negash, A., & Kullgren, G. (2009). Clinical course and outcome of Schizophrenia in a predominantly treatment-naive cohort in rural ethiopia. *Schizophrenia Bulletin*, *35*(3), 646–654. https://doi.org/10.1093/schbul/sbn029

Allamanis, M., Barr, E. T., Devanbu, P., & Sutton, C. (2018). A survey of machine learning for big code and naturalness. *ACM Computing Surveys*, *51*(4). https://doi.org/10.1145/3212695

American Psychiatric Association. (2022). *Psychiatry.org - Frequently Asked Questions*. American Psychiatric Association. https://psychiatry.org/psychiatrists/practice/dsm/frequently-asked-questions

Asimare, H. (2020). *Designing and Implementing Adaptive Bot Model to Consult Ethiopian Published Laws Using Ensemble Architecture with Rules Integrated* [Bahirdar University]. http://ir.bdu.edu.et/handle/123456789/12718

Ayano, G. (2016). Primary Mental Health Care Services in Ethiopia: Experiences, Opportunities and Challenges from East African Country. *Journal of Neuropsychopharmacology & Mental Health*, *1*(4). https://doi.org/10.4172/2472-095x.1000113

B Arnold, T. (2017). kerasR: R Interface to the Keras Deep Learning Library. *The Journal of Open Source Software*, *2*(14), 296. https://doi.org/10.21105/joss.00296

Bartusiak, R., Augustyniak, Ł., Kajdanowicz, T., Kazienko, P., & Piasecki, M. (2019). WordNet2Vec:

Corpora agnostic word vectorization method. *Neurocomputing*, *326–327*, 141–150. https://doi.org/10.1016/j.neucom.2017.01.121

Batish, R. (2018). Chatbot and Voicebot Design: Flexible Conversational Interfaces with Amazon Alexa, Google Home, and Facebook Messenger. *Birmingham : Packt Publishing Ltd*, 1–296. https://books.google.com/books?hl=en&lr=&id=XSxxDwAAQBAJ&oi=fnd&pg=PP1&dq=chatbot&ots=Ra5soB5Xb_&sig=jHBZgkMiKh6QbdiFPcA81jPjjFs

Bengio, Y., Ducharme, R., & Vincent, P. (2001). A neural probabilistic language model. *Advances in Neural Information Processing Systems*, *3*, 1137–1155.

Bernard, J. (2016). Python Data Analysis with pandas. In *Python Recipes Handbook* (pp. 37–48). Apress. https://doi.org/10.1007/978-1-4842-0241-8_5

Bhagwat, V. A. (2018). Deep Learning for ChatBots [San Jose State University]. In *Scholarworks.Sjsu.Edu*. https://doi.org/10.31979/etd.9hrt-u93z

Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., & Jenssen, R. (2017). Recurrent neural network architectures. In *SpringerBriefs in Computer Science* (Vol. 0, Issue 9783319703374, pp. 23–29). https://doi.org/10.1007/978-3-319-70338-1_3

Bifftu, B. B., Takele, W. W., Guracho, Y. D., & Yehualashet, F. A. (2018). Depression and Its Help Seeking Behaviors: A Systematic Review and Meta-Analysis of Community Survey in Ethiopia. *Depression Research and Treatment*, *2018*, 1–11. https://doi.org/10.1155/2018/1592596

Bordes, A., Lan Boureau, Y., & Weston, J. (2017). Learning end-to-end goal-oriented dialog. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. http://arxiv.org/abs/1605.07683

Borrelli, Mattia; Zappa, D. (2017). *From unstructured data and word vectorization to meaning: text mining in insurance*. 1–6.

Brandtzaeg, P. B., & Følstad, A. (2017). Why people use chatbots. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 10673 LNCS* (pp. 377–392). https://doi.org/10.1007/978-3-319-70284-1_30

Bruno Marietto, M. das G., Aguiar, R. V., Barbosa, G. de O., Botelho, W. T., Pimentel, E., Franca, R. dos S., & da Silva, V. L. (2013). Artificial Intelligence Markup Language: A Brief Tutorial. *International Journal of Computer Science & Engineering Survey*, *4*(3), 1–20. https://doi.org/10.5121/ijcses.2013.4301

Cameron, G., Cameron, D., Megaw, G., Bond, R., Mulvenna, M., O'Neill, S., Armour, C., & McTear, M. (2017). Towards a chatbot for digital counselling. *HCI 2017: Digital Make Believe - Proceedings of the 31st International BCS Human Computer Interaction Conference, HCI 2017*, *2017-July*. https://doi.org/10.14236/ewic/HCI2017.24

Campbell, J. C., Hindle, A., & Stroulia, E. (2015). Latent Dirichlet Allocation: Extracting Topics from Software Engineering Data. *The Art and Science of Analyzing Software Data*, *3*(4–5), 139–

159. https://doi.org/10.1016/B978-0-12-411519-4.00006-9

Catherine Pearson, & Pearson, C. (2013). *DSM-5 Changes: What Parents Need To Know About The First Major Revision In Nearly 20 Years*. The Huffington Post. http://www.huffingtonpost.com/2013/05/20/dsm5-changes-what-parents-need-to-know_n_3294413.html

Chatbot. (2020). *ChatBot | Customer Service AI Chatbots for Your Website*. https://www.chatbot.com/

Chawla, N. V. (2009). Data Mining for Imbalanced Datasets: An Overview. *Data Mining and Knowledge Discovery Handbook*, 875–886. https://doi.org/10.1007/978-0-387-09823-4_45

Chen, R. X. F. (2016). *A Brief Introduction to Shannon's Information Theory*. http://arxiv.org/abs/1612.09316

Chung, K., & Park, R. C. (2019). Chatbot-based heathcare service with a knowledge base for cloud computing. *Cluster Computing*, *22*(S1), 1925–1937. https://doi.org/10.1007/s10586-018-2334-5

Csaky, R. (2019). *Proposal Towards a Personalized Knowledge-powered Self-play Based Ensemble Dialog System*. http://arxiv.org/abs/1909.05016

Dachew, B. A., Bisetegn, T. A., & Gebremariam, R. B. (2015). Prevalence of mental distress and associated factors among undergraduate students of University of Gondar, Northwest Ethiopia: A cross-sectional institutional based study. *PLoS ONE*, *10*(3), e0119464. https://doi.org/10.1371/journal.pone.0119464

Dale, R. (2016). The return of the chatbots. *Natural Language Engineering*, *22*(5), 811–817. https://doi.org/10.1017/S1351324916000243

Davis, F. D., & Venkatesh, V. (2004). Toward preprototype user acceptance testing of new information systems: Implications for software project management. *IEEE Transactions on Engineering Management*, *51*(1), 31–46. https://doi.org/10.1109/TEM.2003.822468

del Barrio, V. (2004). Diagnostic and Statistical Manual of Mental Disorders. In *Encyclopedia of Applied Psychology, Three-Volume Set*. https://doi.org/10.1016/B0-12-657410-3/00457-8

Dhyani, M., & Kumar, R. (2019). An intelligent Chatbot using deep learning with Bidirectional RNN and attention model. *Materials Today: Proceedings*, *34*, 817–824. https://doi.org/10.1016/j.matpr.2020.05.450

Dilbo, S. B. (2021). *Developing Afaan Oromoo Text based Chatbot for Enhancing Maize productivity* (Issue January). Adama Science and technology University.

Duko, B. (2016). Prevalence and Correlates of Co-occurring Substance Use Disorder among Patients with Severe Mental Disorder at Amanuel Mental Specialized Hospital, Addis Ababa, Ethiopia. *Journal of Neuropsychopharmacology & Mental Health*, *01*(01). https://doi.org/10.4172/2472-095x.1000101

Exponential Distribution — Intuition, Derivation, and Applications | by Aerin Kim | Towards Data Science. (n.d.). Retrieved October 29, 2021, from https://towardsdatascience.com/what-is-exponential-distribution-7bdd08590e2a

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

Fekadu Eshetu Hunde. (2021). *Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services,Adama Science and Technology University*. Adama Science and technology University.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., Schlaefer, N., & Welty, C. (2011). Building watson: An overview of the deepQA project. *AI Magazine*, *31*(3), 59–79. https://doi.org/10.1609/aimag.v31i3.2303

Følstad, A., Skjuve, M., & Brandtzaeg, P. B. (2019). Different chatbots for different purposes: Towards a typology of chatbots to understand interaction design. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 11551 LNCS* (pp. 145–156). https://doi.org/10.1007/978-3-030-17705-8_13

Forloina, D., & Jufied, C. (2019). *Watson Assisatant*. https://www.ibm.com/developerworks/community/files/form/anonymous/api/library/211ab62%0Aa-38a4-4232-b61a-12d9a43eba9a/document/89a235ce-69d3-4212-be6b-%0A88a8ad073612/media/Lab 03 Watson Conversation.pdf%0A

Gao, B., & Pavel, L. (2017). *On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning*. http://arxiv.org/abs/1704.00805

Gentsch, P. (2018). Conversational Commerce: Bots, Messaging, Algorithmen und Artificial Intelligence. In *Künstliche Intelligenz für Sales, Marketing und Service* (pp. 83–116). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-19147-4_6

Greenler, R. G., Mallmann, A. J., Mueller, J. R., & Romito, R. (1977). Form and origin of the Parry arcs. *Science*, *195*(4276), 360–367. https://doi.org/10.1126/science.195.4276.360

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

Hill, J., Randolph Ford, W., & Farreras, I. G. (2015). Real conversations with artificial intelligence: A comparison between human-human online conversations and human-chatbot conversations. *Computers in Human Behavior*, *49*, 245–250. https://doi.org/10.1016/j.chb.2015.02.026

Hoy, M. B. (2018). Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. *Medical Reference Services Quarterly*, *37*(1), 81–88. https://doi.org/10.1080/02763869.2018.1404391

Huang, J., Zhou, M., & Yang, D. (2007). Extracting chatbot knowledge from online discussion forums. *IJCAI International Joint Conference on Artificial Intelligence*, 423–428. http://dvl.dtic.mil/stop_list.html

Ilievski, V., Musat, C., Hossmann, A., & Baeriswyl, M. (2018). Goal-Oriented chatbot dialog management bootstrapping with transfer learning. *IJCAI International Joint Conference on*

Artificial Intelligence, *2018-July*, 4115–4121. https://doi.org/10.24963/ijcai.2018/572

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). An empirical exploration of Recurrent Network architectures. In *32nd International Conference on Machine Learning, ICML 2015* (Vol. 3, pp. 2332–2340). PMLR. https://proceedings.mlr.press/v37/jozefowicz15.html

Jwala, K., Sirisha, G. N. V. G., & Padma Raju, G. V. (2019). Developing a chatbot using machine learning. *International Journal of Recent Technology and Engineering*, *8*(1), 89–92.

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

Karani, D. (2018). *Introduction to Word Embedding and Word2Vec*. Towards Data Science. https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

Kassa, G. M., & Abajobir, A. A. (2018). Prevalence of common mental illnesses in Ethiopia: A systematic review and meta-analysis. In *Neurology Psychiatry and Brain Research* (Vol. 30, pp. 74–85). Elsevier GmbH. https://doi.org/10.1016/j.npbr.2018.06.001

Kidwai, B., & Rk, N. (2020). Design and Development of Diagnostic Chabot for supporting Primary Health Care Systems. *Procedia Computer Science*, *167*, 75–84. https://doi.org/10.1016/j.procs.2020.03.184

Klopfenstein, L. C., Delpriori, S., Malatini, S., & Bogliolo, A. (2017). The rise of bots: A survey of conversational interfaces, patterns, and paradigms. *DIS 2017 - Proceedings of the 2017 ACM Conference on Designing Interactive Systems*, 555–565. https://doi.org/10.1145/3064663.3064672

Klüwer, T. (2011). From chatbots to dialog systems. In *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices* (pp. 1–22). IGI Global. https://doi.org/10.4018/978-1-60960-617-6.ch001

Kowatsch, T., Volland, D., Shih, I., Rüegger, D., Künzler, F., Barata, F., Filler, A., Büchter, D., Brogle, B., Heldt, K., Gindrat, P., Farpour-Lambert, N., & L'Allemand, D. (2017). Design and evaluation of a mobile chat app for the open source behavioral health intervention platform mobilecoach. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 10243 LNCS* (pp. 485–489). https://doi.org/10.1007/978-3-319-59144-5_36

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., & Socher, R. (2016). Ask me anything: Dynamic memory networks for natural language processing. *33rd International Conference on Machine Learning, ICML 2016*, *3*, 2068–2078. http://arxiv.org/abs/1506.07285

Kumawat, D. (2019). *7 Types of Activation Functions in Neural Network | Analytics Steps*. https://www.analyticssteps.com/blogs/7-types-activation-functions-neural-network

Liang, H., Sun, X., Sun, Y., & Gao, Y. (2017). Text feature extraction based on deep learning: a review. *Eurasip Journal on Wireless Communications and Networking*, *2017*(1), 211. https://doi.org/10.1186/s13638-017-0993-1

Lim, S. L., & Goh, O. S. (2016). *Intelligent Conversational Bot for Massive Online Open Courses (MOOCs)*. http://arxiv.org/abs/1601.07065

Lucas, G. M., Rizzo, A., Gratch, J., Scherer, S., Stratou, G., Boberg, J., & Morency, L. P. (2017). Reporting mental health symptoms: Breaking down barriers to care with virtual human interviewers. *Frontiers Robotics AI*, *4*(OCT). https://doi.org/10.3389/frobt.2017.00051

Mekuria, K., Mulat, H., Derajew, H., Mekonen, T., Fekadu, W., Belete, A., Yimer, S., Legas, G., Menberu, M., Getnet, A., & Kibret, S. (2017). High Magnitude of Social Anxiety Disorder in School Adolescents. *Psychiatry Journal*, *2017*, 1–5. https://doi.org/10.1155/2017/5643136

Meng, L., & Huang, M. (2018). Dialogue intent classification with long short-term memory networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 10619 LNAI* (pp. 42–50). https://doi.org/10.1007/978-3-319-73618-1_4

Mhatre, N., Motani, K., Shah, M., & Mali, S. (2016). Donna Interactive Chat-bot acting as a Personal Assistant. *International Journal of Computer Applications*, *140*(10), 6–11. https://doi.org/10.5120/ijca2016909460

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*. http://arxiv.org/abs/1301.3781

Miner, A. S., Milstein, A., & Hancock, J. T. (2017). Talking to machines about personal mental health problems. *JAMA - Journal of the American Medical Association*, *318*(13), 1217–1218. https://doi.org/10.1001/jama.2017.14151

Mishra, C., & Gupta, D. L. (2016). Deep Machine Learning and Neural Networks: An Overview. *International Journal of Hybrid Information Technology*, *9*(11), 401–414. https://doi.org/10.14257/ijhit.2016.9.11.34

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

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

Nimavat, K., & Champaneria, T. (2017). *Chatbots: An Overview Types, Architecture, Tools and Future Possibilities*. International Journal of Scientific Research and Development. https://www.researchgate.net/publication/320307269_Chatbots_An_overview_Types_Architecture_Tools_and_Future_Possibilities

Pandian, S. (2022). *K-Fold Cross Validation Technique and its Essentials*. Data Science Blogathon. https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/

Paper, D. (2020). Scikit-Learn Regression Tuning. In *Hands-on Scikit-Learn for Machine Learning Applications* (pp. 189–213). Apress. https://doi.org/10.1007/978-1-4842-5373-1_7

Pattanayak, S. (2017). Introduction to Deep-Learning Concepts and TensorFlow. In *Pro Deep Learning with TensorFlow* (pp. 89–152). Apress. https://doi.org/10.1007/978-1-4842-3096-1_2

Paul, Z. M. (2017). Cortana-Intelligent Personal Digital Assistant: a Review. *International Journal of Advanced Research in Computer Science*, *8*(7), 55–57. https://doi.org/10.26483/ijarcs.v8i7.4225

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1532–1543. https://doi.org/10.3115/v1/d14-1162

Peters, F. (2018). Design and implementation of a chatbot in the context of customer support. *Liege University*, 1–77. http://matheo.uliege.be

Pollock, K., Armstrong, S., Coveney, C., & Moore, J. (2010). *An evaluation of Samaritans telephone and email emotional support service*. The NIHR Research Design Service for the East Midlands . http://wrap.warwick.ac.uk/46736/

Popov, A. (2018). Neural network models for word sense disambiguation: An overview. *Cybernetics and Information Technologies*, *18*(1), 139–151. https://doi.org/10.2478/cait-2018-0012

Post, M. (2018). A Call for Clarity in Reporting BLEU Scores. *WMT 2018 - 3rd Conference on Machine Translation, Proceedings of the Conference*, *1*, 186–191. https://doi.org/10.18653/v1/w18-6319

Pykes, K. (2021). *Fighting Overfitting With L1 or L2 Regularization: Which One Is Better? - neptune.ai*. https://neptune.ai/blog/fighting-overfitting-with-l1-or-l2-regularization

Radziwill, N. M., & Benton, M. C. (2017). *Evaluating Quality of Chatbots and Intelligent Conversational Agents*. http://arxiv.org/abs/1704.04579

Ram, A., Prasad, R., Khatri, C., Venkatesh, A., Gabriel, R., Liu, Q., Nunn, J., Hedayatnia, B., Cheng, M., Nagar, A., King, E., Bland, K., Wartick, A., Pan, Y., Song, H., Jayadevan, S., Hwang, G., & Pettigrue, A. (2018). *Conversational AI: The Science Behind the Alexa Prize*. http://arxiv.org/abs/1801.03604

Recurrent Neural Networks and Natural Language Processing. by Christopher Thomas BSc Hons. MIAP Towards Data Science. (n.d.). Retrieved October 29, 2021, from https://towardsdatascience.com/recurrent-neural-networks-and-natural-language-processing-73af640c2aa1

Reddy Karri, S. P., & Santhosh Kumar, B. (2020). Deep learning techniques for implementation of chatbots. *2020 International Conference on Computer Communication and Informatics, ICCCI 2020*, 1–5. https://doi.org/10.1109/ICCCI48352.2020.9104143

Reyes, R., Garza, D., Garrido, L., De la Cueva, V., & Ramirez, J. (2019). Methodology for the Implementation of Virtual Assistants for Education Using Google Dialogflow. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 11835 LNAI* (pp. 440–451). https://doi.org/10.1007/978-3-030-33749-0_35

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). *Recent Advances in Recurrent Neural Networks*. http://arxiv.org/abs/1801.01078

Sathiyasusuman, A. (2011). Mental health services in Ethiopia: Emerging public health issue. *Public Health*, *125*(10), 714–716. https://doi.org/10.1016/j.puhe.2011.06.014

Schlicht, M. (2016). *The Complete Beginner's Guide To Chatbots*. Chatbots Magazine. https://chatbotsmagazine.com/the-complete-beginner-s-guide-to-chatbots-8280b7b906ca#.7hn1va3sa

Sharma, Y., & Gupta, S. (2018). Deep Learning Approaches for Question Answering System. *Procedia Computer Science*, *132*, 785–794. https://doi.org/10.1016/j.procs.2018.05.090

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

Shawar, B. A., & Atwell, E. (2007). Different measurements metrics to evaluate a chatbot system. *Proceedings of the Workshop on Bridging the Gap Academic and Industrial Research in Dialog Technologies - NAACL-HLT '07*, 89–96. https://doi.org/10.3115/1556328.1556341

Sheikh, S. A., Tiwari, V., & Singhal, S. (2019). Generative model chatbot for Human Resource using Deep Learning. *2019 International Conference on Data Science and Engineering, ICDSE 2019*, 126–132. https://doi.org/10.1109/ICDSE47409.2019.8971795

Simeon Kostadinov. (2017). *Understanding GRU Networks. In this article, I will try to give a… | by Simeon Kostadinov | Towards Data Science*. Towardsdatascience. https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be

Skjuve, M., & Brandtzæg, P. B. (2018). Chatbots as a new user interface for providing health information to young people. *Youth and News in a Digital Media Environment-Nodric-Baltic Perspectives, May*. http://brage.bibsys.no/sintef

Song, Y., Li, C. Te, Nie, J. Y., Zhang, M., Zhao, D., & Yan, R. (2018). An ensemble of retrieval-based and generation-based human-computer conversation systems. *IJCAI International Joint Conference on Artificial Intelligence*, *2018-July*, 4382–4388. https://doi.org/10.24963/ijcai.2018/609

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

Srivastava, R. K., Greff, K., & Schmidhuber, J. (2015). Training very deep networks. *Advances in Neural Information Processing Systems*, *2015-Janua*, 2377–2385.

http://arxiv.org/abs/1507.06228

Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge Engineering: Principles and methods. *Data and Knowledge Engineering*, *25*(1–2), 161–197. https://doi.org/10.1016/S0169-023X(97)00056-6

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, *4*(January), 3104–3112. http://arxiv.org/abs/1409.3215

Sweeney, C., Potts, C., Ennis, E., Bond, R., Mulvenna, M. D., O'Neill, S., Malcolm, M., Kuosmanen, L., Kostenius, C., Vakaloudis, A., McConvey, G., Turkington, R., Hanna, D., Nieminen, H., Vartiainen, A. K., Robertson, A., & McTear, M. F. (2021). Can Chatbots Help Support a Person's Mental Health? Perceptions and Views from Mental Healthcare Professionals and Experts. *ACM Transactions on Computing for Healthcare*, *2*(3), 1–15. https://doi.org/10.1145/3453175

Tammewar, A., Pamecha, M., Jain, C., Nagvenkar, A., & Modi, K. (2017). *Production Ready Chatbots: Generate if not Retrieve*. http://arxiv.org/abs/1711.09684

Tegegne, M. T., Mossie, T. B., Awoke, A. A., Assaye, A. M., Gebrie, B. T., & Eshetu, D. A. (2015). Depression and anxiety disorder among epileptic people at Amanuel Specialized Mental Hospital, Addis Ababa, Ethiopia. *BMC Psychiatry*, *15*(1), 1–7. https://doi.org/10.1186/s12888-015-0589-4

Tesfaw, G., Ayano, G., Awoke, T., Assefa, D., Birhanu, Z., Miheretie, G., & Abebe, G. (2016). Prevalence and correlates of depression and anxiety among patients with HIV on-follow up at Alert Hospital, Addis Ababa, Ethiopia. *BMC Psychiatry*, *16*(1), 368. https://doi.org/10.1186/s12888-016-1037-9

Tsegabrhan, H., Negash, A., Tesfay, K., & Abera, M. (2014). Co-morbidity of depression and epilepsy in jimma university specialized hospital, southwest ethiopia. *Neurology India*, *62*(6), 649–655. https://doi.org/10.4103/0028-3886.149391

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, *37*, 141–188. https://doi.org/10.1613/jair.2934

Understanding LSTM Networks -- colah's blog. (2016). http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Vaidyam, A. N., Wisniewski, H., Halamka, J. D., Kashavan, M. S., & Torous, J. B. (2019). Chatbots and Conversational Agents in Mental Health: A Review of the Psychiatric Landscape. *Canadian Journal of Psychiatry*, *64*(7), 456–464. https://doi.org/10.1177/0706743719828977

Valério, F. A. M., Guimarães, T. G., Prates, R. O., & Candello, H. (2017). Here's what i can do: Chatbots' strategies to convey their features to users. *ACM International Conference Proceeding Series*, 1–10. https://doi.org/10.1145/3160504.3160544

Vamsi, G. K., Rasool, A., & Hajela, G. (2020). Chatbot: A Deep Neural Network Based Human to

Machine Conversation Model. *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020*, 1–7. https://doi.org/10.1109/ICCCNT49239.2020.9225395

Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017). Evaluation of recurrent neural network and its variants for intrusion detection system (IDS). *International Journal of Information System Modeling and Design*, *8*(3), 43–63. https://doi.org/10.4018/IJISMD.2017070103

Vinet, L., & Zhedanov, A. (2011). A "missing" family of classical orthogonal polynomials. *Journal of Physics A: Mathematical and Theoretical*, *44*(8), 343–354. https://doi.org/10.1088/1751-8113/44/8/085201

Vinyals, O., & Le, Q. (2015). *A Neural Conversational Model*. http://arxiv.org/abs/1506.05869

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

West, J. S. (1991). *Student Perceptions That Inhibit the Initiation of Counseling.* School Counselor. https://www.jstor.org/stable/23901606

Weston, J., Chopra, S., & Bordes, A. (2015). Memory networks. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://doi.org/10.1007/978-3-030-82184-5_11

What is Microsoft Visio® | Lucidchart. (n.d.). https://www.lucidchart.com/pages/what-is-microsoft-visio

Wikipedia. (2017). *Amharic - Wikipedia*. https://en.wikipedia.org/wiki/Amharic

Wolf, M. J., Miller, K. W., & Grodzinsky, F. S. (2017). Why We Should Have Seen That Coming. *The ORBIT Journal*, *1*(2), 1–12. https://doi.org/10.29297/orbit.v1i2.49

Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017). A new chatbot for customer service on social media. *Conference on Human Factors in Computing Systems - Proceedings*, *2017-May*, 3506–3510. https://doi.org/10.1145/3025453.3025496

Yitbarek, K., Birhanu, Z., Tucho, G. T., Anand, S., Agenagnew, L., Snr, G. A., Getnet, M., & Tesfaye, Y. (2021). Barriers and facilitators for implementing mental health services into the ethiopian health extension program: A qualitative study. *Risk Management and Healthcare Policy*, *14*, 1199–1210. https://doi.org/10.2147/RMHP.S298190

ZEMČÍK, M. T. (2019). A Brief History of Chatbots. *DEStech Transactions on Computer Science and Engineering*, *aicae*. https://doi.org/10.12783/dtcse/aicae2019/31439

APPENDIXES

APPENDIX I: cross entropy, learning rate, epoch, and perplexity diagrams

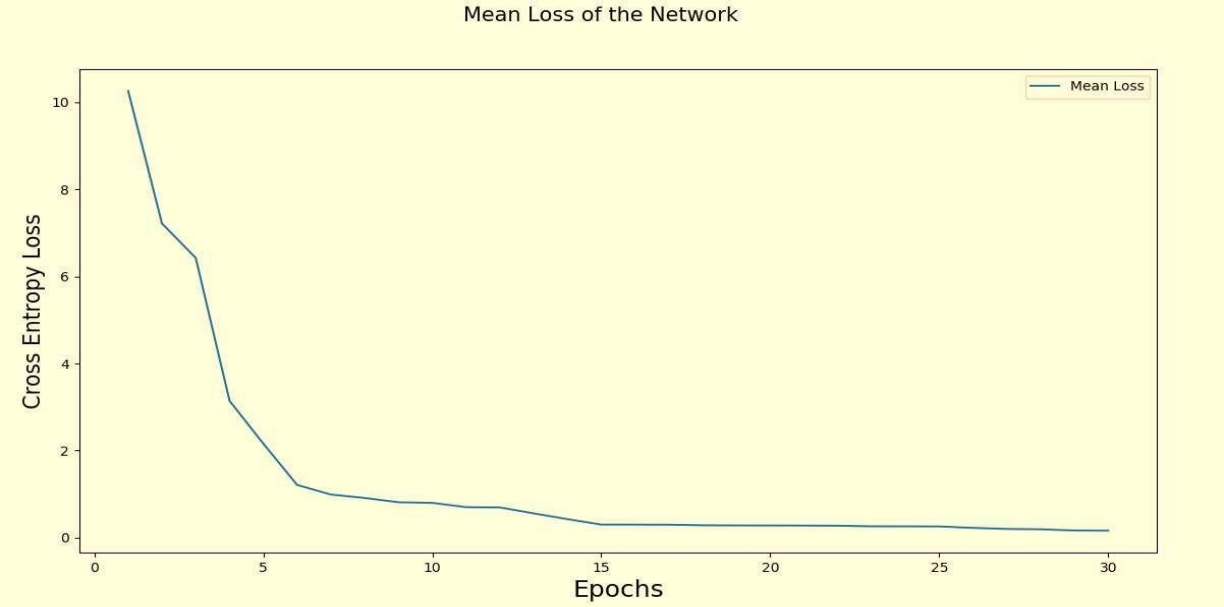


Figure 35 cross-entropy loss description

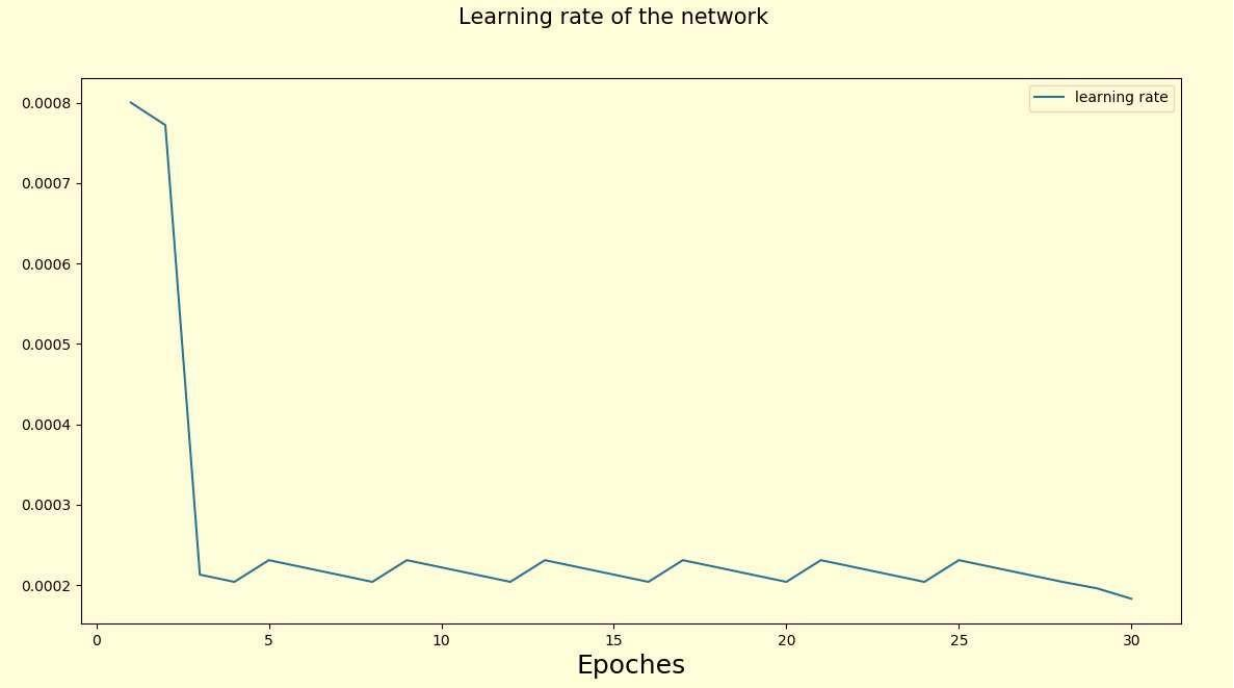


Figure 36 Perceived learning rate during training

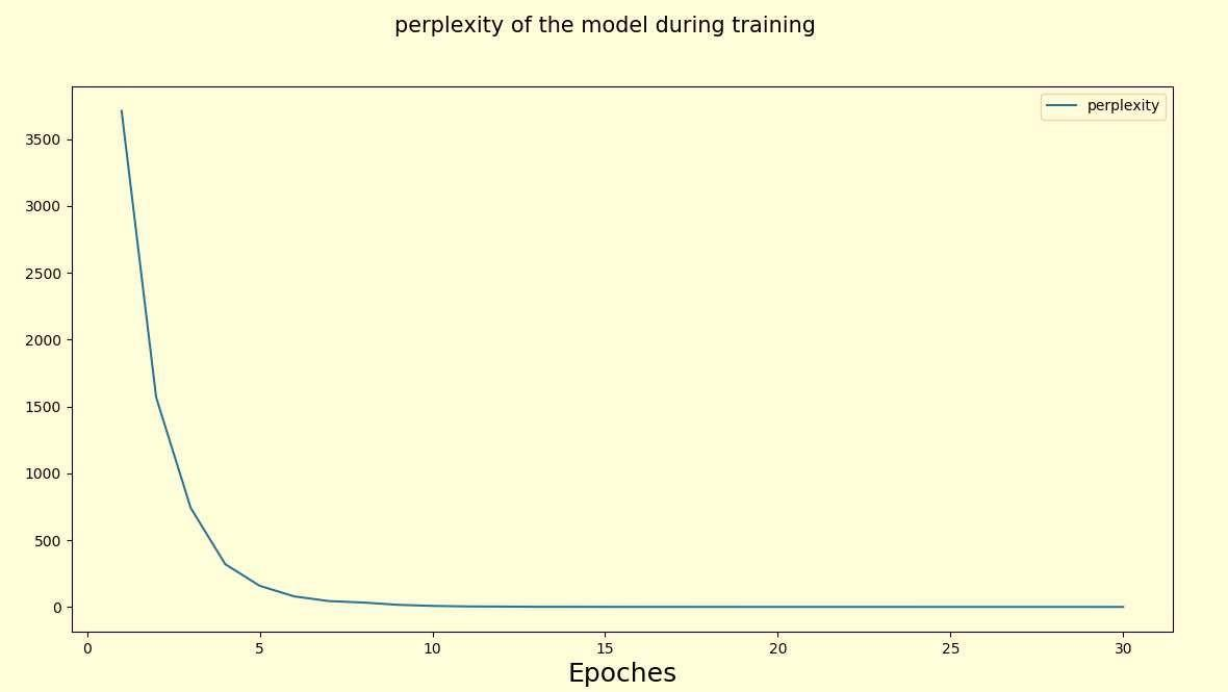


Figure 37 Perplexities of the model during training

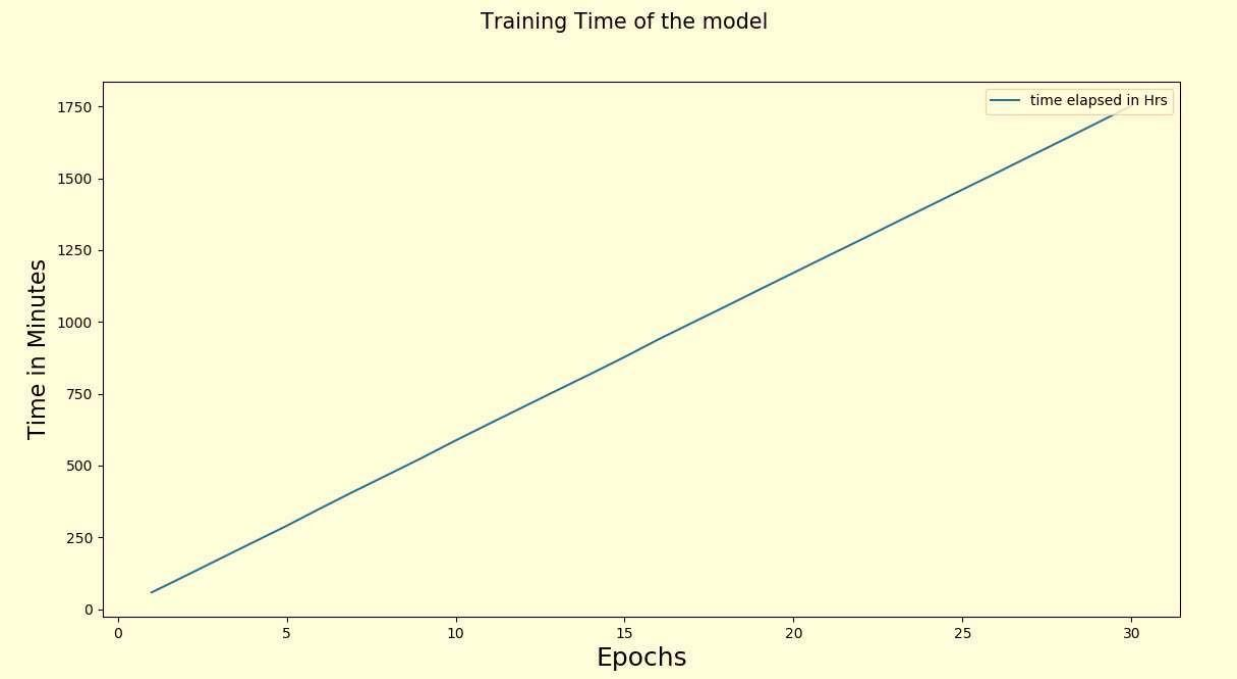


Figure 38 Elapsed time during training

APPENDIX II: sample JSON file preparation


```
{"intents": [
  {"tag": " ሰላምታ ",
  "patterns": ["ጤና ይስጥልኝ", "ሰላም", "እንዴት ሰነበርትክ" ],
  "Responses": ["ጤና ይስጥልኝ!", "ሰላም ጤና ይስጥልኝ ምን ልታዘዝ ?", "እንዴት ሰነበርትክ ", "ሰላም ምን ላግዝህ እችላሎህ ?"],
  "Context_set": ""
  },
  { "tag": "ቻው ደና ሁን አመሰግናለው!",
  "patterns": ["ቻው!", "ቻው", "ደና ሁን አመሰግናለው!", "ቻው ደና ሁን!", "ደና ሁን!", "ቻው" ],
  "Responses": ["እሺ ቻው ደና ሁን ", "እሺ ላንተም መልካም ግዜ", "መልካም ግዜ ይሁንል", "እኔም አመሰግለው"],
  "Context_set": ""
  },
  { "tag": "ባይፖላር ዲስኦርደር ምንድን ነው?",
  "patterns": ["ባይፖላር ዲስኦርደር ምንድን ነው? ", "ስለ ባይፖላር ዲስኦርደር የሆነ ነገር ንገረኝ?", "ባይፖላር ዲስኦርደር ምን ማለት ነው? ", "ባይፖላር ዲስኦርደር ምን አይነት
  "Responses": [" ባይፖላር ዲስኦርደር በሰው ስሜት እና በኃይል ደረጃ ላይ ከፍተኛ ለውጥ የሚያመጣ የአእምሮ ጤና ሁኔታ ነው ። ሁሉም ሰው ውጣ ውረዶችን የሚያጋጥመው ቢሆን
  "Context_set": ""
  },
  { "tag": "ምን ዓይነት ባይፖላር ዲስኦርደር ",
  "patterns": ["ምን ዓይነት ባይፖላር ዲስኦርደር አለኝ? ምን ያህል ከባድ ነው? የበሽታውን ችግር ሊያስረዱኝ ይችላሉ?", "ምን ዓይነት ባይፖላር ዲስኦርደር ምን ያህል ከባድ ነው?",
  "Responses": ["ባይፖላር 1 ዲስኦርደር በጣም ከባድ የሕመሙ ዓይነት ነው ። ዳግማዊ ባይፖላር / bipolar II ዲስኦርደር አልፎ አልፎ በሚታዩ የሂሞማኒክ ክፍሎች የታጀበ በአብ
  "Context_set": ""
  },

 { "tag": "ባይፖላር ዲስኦርደር በዕድሜ ይለወጣል? ",
 "patterns": ["ባይፖላር ዲስኦርደር በዕድሜ ይለወጣል? ", "ባይፖላር ዲስኦርደር በዕድሜ ለውጥ ይቀያየራል? "],
 "Responses": ["ብዙውን ጊዜ በልጅነት ዕድሜያቸው ከሚጀምሩ ምልክቶች ጋር ባይፖላር ዲስኦርደር በተለምዶ የዕድሜ ልክ መታወክ ተብሎ ይታሰባል ። አሁን ተመራማሪዎቹ ዕድሜያቸው ከ 18 እስከ 25 ዓመት ከሆኑት መ
 "Context_set": ""
 },
 { "tag": "ባይፖላር ዲስኦርደር በረጅም ጊዜ ላይ እንዴት",
 "patterns": ["ባይፖላር ዲስኦርደር በረጅም ጊዜ ላይ እንዴት ይጎዳኛል?", "ባይፖላር ዲስኦርደር ሁኔታ የረጅም ግዜ ውጤቶች ምንድ ናቸው?", "ባይፖላር ዲስኦርደር ሲቆይ ምን ሊጎዳኝ ይችላል?" ],
 "Responses": ["ለረዥም ጊዜ ተስፋ ቢስ ወይም አቅመ ቢስነት ስሜት ወይም በራስ መተማመን ዝቅተኛ መሆን። የተቀነሰ የኃይል መጠን። ትኩረት ለማድረግ ወይም ቀላል ውሳኔዎችን ለማድረግ አለመቻል ። እንደ መብላት
 "Context_set": ""
 },
 { "tag": "በእንክብካቤዬ ውስጥ ሊሳተፉባቸው የሚገቡ ሌሎች የሕክምና ዓይነቶች ወይም የአእምሮ ጤና ባለሙያዎች አሉ?",
 "patterns": ["በእንክብካቤዬ ውስጥ ሊሳተፉባቸው የሚገቡ ሌሎች የሕክምና ዓይነቶች ወይም የአእምሮ ጤና ባለሙያዎች አሉ?" ],
 "Responses": ["የአእምሮ ህክምና ባለሙያዎች የአእምሮ ጤና ጠበብት የሆኑ የህክምና ዶክተሮች ናቸው ። ባይፖላር ዲስኦርደር ያለባቸውን ሰዎች በመመርመር እና በማከም ረገድ ስፔሻሊስቶች ናቸው ። የአእምሮ ሐኪሞች የሕ
 "Context_set": ""
 },
  { "tag": " ለባይፖላር ዲስኦርደር ሆስፒታል መተኛት መቼ?",
 "patterns": ["ለባይፖላር ዲስኦርደር ሆስፒታል መተኛት ጠቃሚ ወይም አስፈላጊ የሚሆነው መቼ ነው?", "ባይፖላር ዲስኦርደር ሁኔታ ሆስፒታል መተኛት ያስፈልጋል?", "የባይፖላር ዲስኦርደር ህመም ሲሰማኝ ሆስፒታል መተ
 "Responses": ["ባይፖላር ዲስኦርደር ውስጥ ሆስፒታል መተኛት እንደ ድንገተኛ አማራጭ ተደርጎ ይወሰዳል ። መታወክ እንድ ሰው ለራሱ ወይም ለሌሎች አፋጣኝ ስጋት እንዲሆን በሚያደርግበት በጣም ከባድ በሆኑ ጉዳዮች አስ
 "Context_set": ""
 },
  { "tag": " ባይፖላር ዲስኦርደር በችግር ውስጥ እንደሆኑኩ ከተሰማኝ ",
 "patterns": ["በባይፖላር ዲስኦርደርበ በችግር ውስጥ እንደሆንኩ ከተሰማኝ ወይም የአስቸኳይ ጊዜ እርዳታ ከፈለግኩ ምን ማድረግ አለብኝ?", "በባይፖላር ዲስኦርደርበ አስቸኳይ ጊዜ እርዳታ ከፈለግኩ ምን ማድረግ አለብኝ?"
 "Responses": ["የቱንም ያህል ዝቅ ወይም ከቁጥጥር ውጭ ቢሆኑም ፣ ባይፖላር ዲስኦርደርን በተመለከተ ኃይል እንደሌለዎት ማስታወሱ አስፈላጊ ነው ። ከሐኪምዎ ወይም ከሕክምና ባለሙያው ከሚሰጡት ሕክምና ባሻገር ም
 "Context_set": ""
 },
```


Figure 39 Sample JSON data produced from [Q] [A] pairs

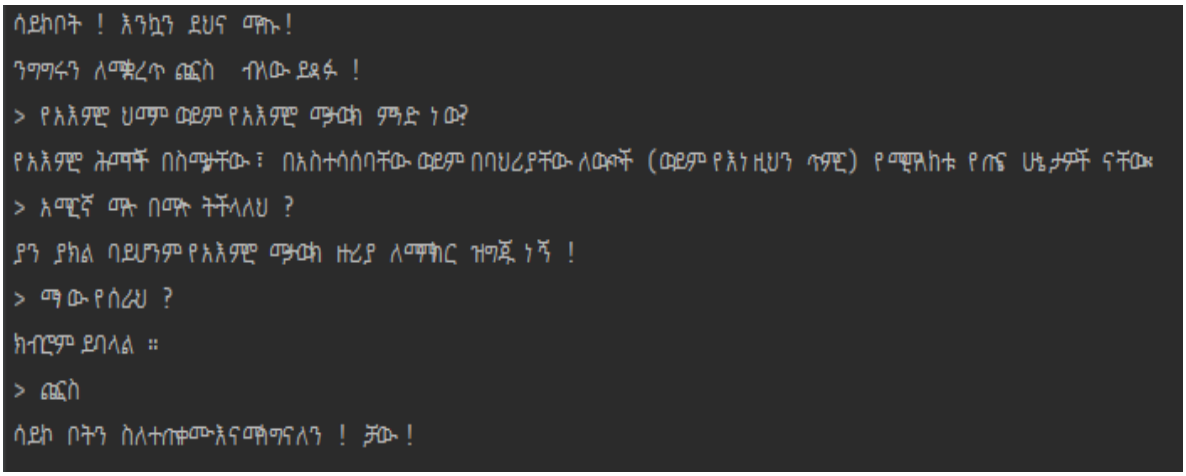
APPENDIX III: Snapshoot during training epochs

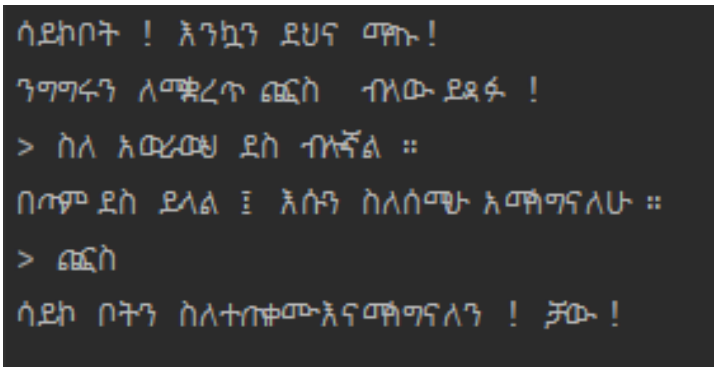
```
# Training loop started @ 2021-07-03 12:54:32
# Finished epoch  1 @ step    15 @ 2021-07-03 13:06:23. In the epoch, learning rate = 0.000800, mean loss = 10.1788, perplexity = 26339.8877, and 711.04 seconds elapsed.
# Finished epoch  2 @ step    30 @ 2021-07-03 13:14:04. In the epoch, learning rate = 0.000768, mean loss = 7.3908, perplexity = 1620.9866, and 460.68 seconds elapsed.
# Finished epoch  3 @ step    45 @ 2021-07-03 13:20:45. In the epoch, learning rate = 0.000737, mean loss = 6.8701, perplexity = 963.0592, and 399.99 seconds elapsed.
# Finished epoch  4 @ step    60 @ 2021-07-03 14:58:41. In the epoch, learning rate = 0.000708, mean loss = 6.2672, perplexity = 526.9920, and 5876.42 seconds elapsed.
# Finished epoch  5 @ step    75 @ 2021-07-03 15:19:37. In the epoch, learning rate = 0.000680, mean loss = 5.5424, perplexity = 255.2810, and 1254.89 seconds elapsed.
# Finished epoch  6 @ step    90 @ 2021-07-03 15:25:47. In the epoch, learning rate = 0.000653, mean loss = 5.1294, perplexity = 168.9151, and 369.87 seconds elapsed.
# Finished epoch  7 @ step   105 @ 2021-07-03 15:36:53. In the epoch, learning rate = 0.000627, mean loss = 4.9179, perplexity = 136.7187, and 665.09 seconds elapsed.
# Finished epoch  8 @ step   120 @ 2021-07-03 16:58:35. In the epoch, learning rate = 0.000602, mean loss = 4.4599, perplexity =  86.4747, and 4901.06 seconds elapsed.
# Finished epoch  9 @ step   135 @ 2021-07-03 17:04:36. In the epoch, learning rate = 0.000600, mean loss = 4.1171, perplexity =  61.3830, and 361.23 seconds elapsed.
# Finished epoch 10 @ step   150 @ 2021-07-03 17:10:41. In the epoch, learning rate = 0.000600, mean loss = 3.7245, perplexity =  41.4507, and 364.19 seconds elapsed.
# Finished epoch 11 @ step   165 @ 2021-07-03 17:16:50. In the epoch, learning rate = 0.000600, mean loss = 3.2744, perplexity =  26.4268, and 367.45 seconds elapsed.
```

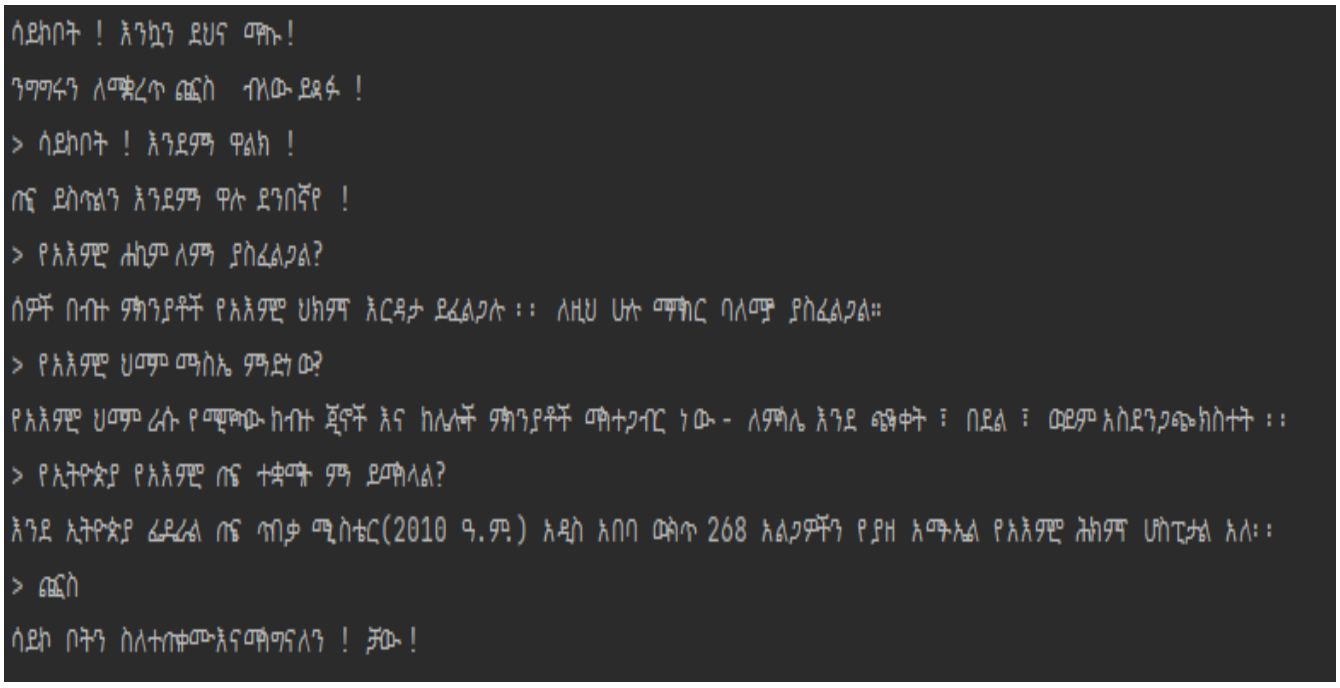
```
# Finished epoch 12 @ step   180 @ 2021-07-03 17:21:54. In the epoch, learning rate = 0.000576, mean loss = 3.1038, perplexity =  22.2815, and 304.05 seconds elapsed.
# Finished epoch 13 @ step   195 @ 2021-07-04 09:02:47. In the epoch, learning rate = 0.000553, mean loss = 3.0631, perplexity =  21.3942, and 56452.76 seconds elapsed.
# Finished epoch 14 @ step   210 @ 2021-07-04 09:12:45. In the epoch, learning rate = 0.000531, mean loss = 2.8578, perplexity =  17.4231, and 597.00 seconds elapsed.
# Finished epoch 15 @ step   225 @ 2021-07-04 09:18:58. In the epoch, learning rate = 0.000510, mean loss = 2.6460, perplexity =  14.0970, and 372.18 seconds elapsed.
# Finished epoch 16 @ step   240 @ 2021-07-04 09:25:09. In the epoch, learning rate = 0.000400, mean loss = 2.4362, perplexity =  11.4300, and 369.77 seconds elapsed.
# Finished epoch 17 @ step   255 @ 2021-07-04 09:32:18. In the epoch, learning rate = 0.000384, mean loss = 2.3676, perplexity =  10.6718, and 428.34 seconds elapsed.
# Finished epoch 18 @ step   270 @ 2021-07-04 09:41:08. In the epoch, learning rate = 0.000369, mean loss = 2.1596, perplexity =   8.6674, and 530.02 seconds elapsed.
# Finished epoch 19 @ step   285 @ 2021-07-04 09:47:16. In the epoch, learning rate = 0.000354, mean loss = 2.0316, perplexity =   7.6260, and 366.98 seconds elapsed.
# Finished epoch 20 @ step   300 @ 2021-07-04 09:53:11. In the epoch, learning rate = 0.000340, mean loss = 1.9741, perplexity =   7.2001, and 354.84 seconds elapsed.
# Finished epoch 21 @ step   315 @ 2021-07-04 09:59:21. In the epoch, learning rate = 0.000326, mean loss = 1.8222, perplexity =   6.1853, and 369.55 seconds elapsed.
# Finished epoch 22 @ step   330 @ 2021-07-04 10:04:38. In the epoch, learning rate = 0.000313, mean loss = 1.8026, perplexity =   6.0652, and 316.94 seconds elapsed.
# Finished epoch 23 @ step   345 @ 2021-07-04 10:09:38. In the epoch, learning rate = 0.000300, mean loss = 1.7461, perplexity =   5.7322, and 298.83 seconds elapsed.
# Finished epoch 24 @ step   360 @ 2021-07-04 10:18:44. In the epoch, learning rate = 0.000288, mean loss = 1.5329, perplexity =   4.6316, and 545.56 seconds elapsed.
# Finished epoch 25 @ step   375 @ 2021-07-04 10:25:02. In the epoch, learning rate = 0.000276, mean loss = 1.5659, perplexity =   4.7870, and 376.90 seconds elapsed.
# Finished epoch 26 @ step   390 @ 2021-07-04 10:29:40. In the epoch, learning rate = 0.000265, mean loss = 1.5950, perplexity =   4.9281, and 277.44 seconds elapsed.
# Finished epoch 27 @ step   405 @ 2021-07-04 10:35:30. In the epoch, learning rate = 0.000254, mean loss = 1.5044, perplexity =   4.5015, and 349.01 seconds elapsed.
# Finished epoch 28 @ step   420 @ 2021-07-04 10:40:31. In the epoch, learning rate = 0.000244, mean loss = 1.4848, perplexity =   4.4142, and 300.32 seconds elapsed.
# Finished epoch 29 @ step   435 @ 2021-07-04 10:45:43. In the epoch, learning rate = 0.000234, mean loss = 1.3475, perplexity =   3.8478, and 310.81 seconds elapsed.
# Finished epoch 30 @ step   450 @ 2021-07-04 10:51:59. In the epoch, learning rate = 0.000225, mean loss = 1.3315, perplexity =   3.7867, and 375.64 seconds elapsed.
# Finished epoch 31 @ step   465 @ 2021-07-04 10:57:15. In the epoch, learning rate = 0.000216, mean loss = 1.1774, perplexity =   3.2461, and 314.26 seconds elapsed.
```

Figure 40Snapshots for training epochs

APPENDIX IV: Snapshoot for sample console conversation







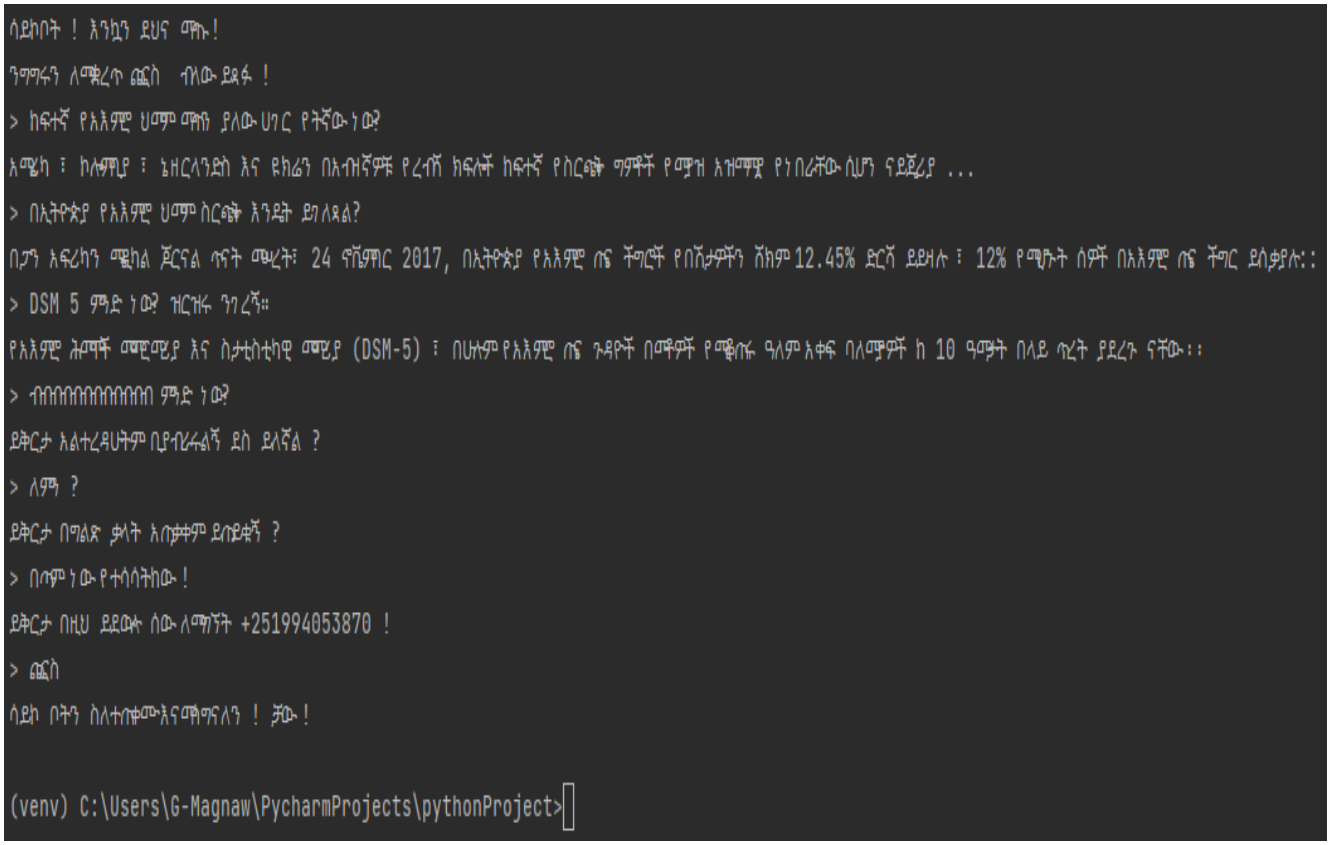


Figure 41 snapshots for sample console conversation

> is an input from the user, whereas the rest was replying from the psychobot model

APPENDIX V: list of Amharic stop words

እኔ	በለ	እያለ	በጉላ	ሆነ	ወይዘሮ
የእኔ	ጋር	ሕዝብ	በአንድ	ለሆነው	ተብሎ
እኔ ራሴ	ስለ	ሀሙስ	የሆኑ	ሰኣት	ሳይሆን
እኛ	ላይ	ለመሆኑ	ከአስራ	ብሎ	እንደሆነና
የእኛ	መካከል	ለምንድን	የሆነውን	ከሰላሳ	ከብር
እኛ ራሳችን	ወደ	ሌሎች	መሆኑ	የሚሆኑ	ሆኖም
አንቺ	በኩል	መጽሀፍ	ሌላውን	ላይም	የሌላ
ነህ	ወቅት	ማክሰኞ	ከሰባት	የሆናል	ያላቸው
አላችሁ	ከዚህ በፊት	ምን	ለሌላ	ከነዚህ	ይሆንኑ
እርስዎ	በኋላ	ሰኞ	አለበት	ያህል	ሆነው
ትፈልጋለህ	ከላይ	ሰው	ሲል	ከሆነና	በስተቀር
ያንተ	ከታች	ሲሆን	ይሆናሉ	ለሆኑት	ስም
ራስህን	ከ	ስንት	በሙሉ	እነዚሁ	እንደገና
እራሳችሁ	ወደ ላይ	ረቡእ	አስራ	እንደሆኑ	የማያንስ
እሱ	ታች	ቅዳሜ	ቢሆንም	ስለማናቸውም	እጅግ
የእሱ	ውስጥ	በዚህ	አንዱ	ስለዚሁ	እንዲሆን
ራሱ	ውጭ	ብላ	የሌላውን	ከአንዳንድ	እንኳ
እሷ	በላይ	ነገር	ከሁለት	በእነዚሁ	ከሀያ
እሷናት	እንደገና	አለ	የሆኑትን	በአምስት	ከሀምሳ
የእሷ	ተጨማሪ	አርብ	በሆኑ	የሆኑበታል	ይኸው
እራሷ	ከዚያ	አንተ	ጀምሮ	ለነዚህ	ለአንድ
ነው	አንድ ጊዜ	እሁድ	በመሆን	ለማንኛውም	የሚችለውን
እነሱ	እዚህ	እናንተ	ባለ	አንደኛ	በሚገባ
እነሱን	እዚያ	እንኳን	ይህንን	ይኸኛው	ይህም
የእነሱ	መቼ	እግር	እንዲቆይ	ከርሱ	እንዲሆኑ
ራሳቸው	የት	ከመሆን	ሌላው	መሆኑን	ከሌላ
ምንድን	ለምን	ወይንም	የሚሆነው	ለዚያው	ለሆነ
የትኛው	እንዴት	ዋና	በአንዱ	ለዚሁ	በሌሎች
ማን	ሁሉም	ዘንድ	ሲባል	ለእነርሱም	አንደሆነ
ይህ	ማንኛውም	የሚከተለው	ሳለ	እዚሁ	እንዲህ
የሚል ነው	ሁለቱም	ያኔ	የሆነው	ሐረሽ	በነዚሁ
ያ	እያንዳንዳቸው	ይኼው	መሆናቸው	አምስት	በእንደዚህ
እነዚህ	ጥቂቶች	ገጽ	በዋና	ከሶስት	ስምንት
እነዚያ	በጣም	እነርሱ	በማቀድ	በተለይም	ሲሆንና
ነኝ	ሌላ	ጉና	ጊዜና	በሌላ	ምንጊዜም
ናቸው	አንዳንድ	ዎች	ለዚህ	ሺህ	ለማናቸውም
ነበር	እንደዚህ	ይጠበቃል	ሶስተኛ	ማናቸውንም	የአንድ
ነበሩ	ብቻ	ብለዋል	የነገሩ	ከአስር	እነዚህኑ
ሁን	የራሱ	ሆ	ስድስት	የማይበልጥ	ሲሆኑ

ቆይቷል	ተመሳሳይ	ሁሉ	በሆነው	እንዲሁም	በሁለቱም
መሆን	ስለዚህ	አንቀጽ	ይሁን	ይህን	እንደነዚህ
አላቸው	ይልቅ	እንደሆነ	ከዚሁ	የዚህ	የሆኑት
አለው	እንዲሁ	በማይበልጥ	በእነዚህ	ማናቸውም	የማናቸውም
ነበረው	ት	መሰረት	ከማናቸውም	ከስድስት	ይህንንም
ያለው	ይችላል	ሁኔታ	ከነበረው	መቶ	የአንድን
መስራት	ይገባል	ይሆናል	በአንዳንድ	ያለ	በሙሉም
ያደርጋል	ይገባኛል	ሆኖ	በእያንዳንዱ	አንድን	በነዚህ
አደረገ	አሁን	ከአንድ	ጊዜም	ያላቸውን	የዚሁ
ማድረግ	መምአ	በማናቸውም	አስከ	ሊሆን	ለእያንዳንዱ
ሀ	ዳግም	ወር	የሌሎች	ሶስት	ስለሆነ
አንድ	መሆን	ከአምስት	የሚሆኑት	ካልሆነ	መሆናቸውን
የ	ሁለ	በሆነ	ከሆነው	ቢያንስ	ማንኛውንም
እና	ሁለም	ከዚህ	የነበረውን	ቢሆን	ሁለቱ
ከሆነ		የሆነ	ያሉ	እነዚህን	እንጂ
ወይም			ከሌሎች	አንዱን	ከስምንት
ምክንያቱም			አንዲት		ሁለቱንም
እንደ			ለሌሎች		በሁለት
እስከ					

APPENDIX VI: Language translation review confirmation letter

መአዛ የትርጉም ስገልግሎት
Meaza Translation Service
አድራሻ: አጥቢያም ደነ ህንፃ 1ኛ ፎቅ
Address: Around Stadium Yeha Building 1st floor
0911-456216/0911251575 E-Mail:- Meazatranslation@yahoo.com

ቀን ፡ 02/10/2014 ዓ.ም

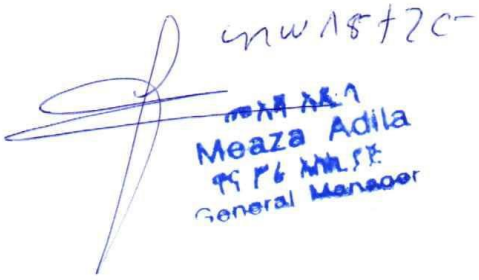
ቁጥር ፡ መ/ት123/2014

ለ ተማሪ ክብሮም ግደይ ኪዳነ ማርያም

ለሚመለከተው ሁሉ

ጉዳዩ ፡ ማረጋገጫ መስጠትን ይመለከታል

ተማሪ ክብሮም ግደይ ኪዳነ ማርያም የአዳማ ሳይንስና ቴክኖሎጂ ዩንቨርስቲ በኮምፒውተር ሳይንስና ኢንጂነሪንግ ትምህርት ክፍል ውስጥ የሁለተኛ ድግሪ ተማሪ ሲሆን በዩርቨርስቲው ውስጥ የሁለተኛ ድግሪ ማሟያ የጥናት ጹሁፍ " *Apprentice Bot Model Design And Implementation For Psychological Clients Therapy Using Ensemble Architecture And Embedded Custom Rules* " ለሚለው "*Diagnostic And Statistical Manual -5th Edition*" በሚል የተዘጋጀውን መጽሀፍ የአማርኛውን ትርጉም የአርትዖት ስራ እንነዲሰራላቸው በጠየቁት መሰረት የመጸሀፉን ስምንት ክፍል ትርጉም ትክክለኛነት ከእንግሊዘኛው ቅጂ ጋር በማስተተያየት ወደ አማርኛ ቋንቋ መተርጎሙን በአግባቡ ተመልክተን ያረጋገጥን መሆኑን እናሳውቃለን።



APPENDIX VII: human evaluation assessment evaluation paper.

Questionnaire: This questionnaire is accepting the user's satisfaction from the attributes prepared. These are; naturalness of conversation, response relevance, Way of user query understanding and the system replying, usage of the model, and response time.

የየሳይኮቦት ሲስተም ተጠቃሚዎች ተቀባይነት የሙከራ ግምገማ ወረቀት

Psycho bot user acceptance testing assessment evaluation paper

በመጀመሪያ የሰው ሰራሽ የአእምሮ ጤና ማማከሪያ ሲስተም ምዘና ለመሙላት ፈቃደኛ ስለሆኑ እናመሰግናለን። እናም ይህ ወረቀት በሚጠይቀው መሰረት እያንዳንዱን ምዘና በትኩረት እንዲሞሉልን በአክብሮት እንጠይቃለን!
በመሙላት ስለተባበሩንም በቅድምያ እናመሰግናለን።
ማሳሰቢያ፡- በኮምፒውተር የተከፈተው ሲስተም ጋር በመገናኘት በአእምሮ ጤናዎ እና ተያያዝ ነገሮች ዙሪያ የሚፈልጉትን ጥያቄ፣የአእምሮ ጤና ምክር ወይም ትምህርቶች በቀረበው የኮምፒውተር ሲስተም በመጻፍ ንግግርወትን ይጀምሩ። ቀጥሎ ከዚህ በታች ያለውን ጥያቄወች "X" ምልክት በማድረግ ይሙሉልን።

መጠይቆች	በጣም ዝቅተኛ(0)	ዝቅተኛ(1)	ጥሩ(2)	በጣም ጥሩ(3)	እጅግ በጣም ጥሩ(4)
1. ሲስተሙ የሚመልስልወት ቋንቋ ግልጽነት ?					
2. ሲስተሙ የሚመልሰውን መልስ እንዴት አገኙት ?					
3. የዚህ ሲስተም ጠቀሜታ በምን ይገልጹታል ?					
4. የተጠቃሚ ጢያቄን መረዳት እና ትክክለኛ መልስ መስጠት					
5. የምላሽ ግዜ					

አስተያየት ካለወት እዚህ ላይ ይጻፉልን!

_____
_____
_____
_____ **ስለ ትብብርዎ እናመሰግናለን።**