2022-12

Amharic chatbot on Ethiopian civil code law using a deep learning approach

Bizuayehu, Tadege

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING

POSTGRADUATE PROGRAM

Department of Information Technology

M.Sc. Thesis On:

Amharic chatbot on Ethiopian civil code law using a deep learning approach

By: Bizuayehu Tadege

Advisor: Dr. Tesfa Tegegne (Assoc Prof)

December 2022

Bahir Dar, Ethiopia

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTING

Amharic chatbot on Ethiopian civil code law using a deep learning approach

By

Bizuayehu Tadege

A Thesis submitted to the school of research and graduate studies of Bahir Dar Institute of Technology, BDU in partial fulfillment of the requirements for the degree of Masters of Science in the Information Technology program in the Faculty of Computing

Advisor: Dr. Tesfa Tegegne (Assoc Prof)

December 2022

Bahir Dar, Ethiopia

Declaration

This is to certify that the thesis entitled “Amharic chatbot on Ethiopian civil code law using deep learning approach”, submitted in partial fulfilment of the requirements for the degree of Master of Science in Information Technology under the faculty of computing, Bahir Dar Institute of Technology is a record of original work carried out by me and has never been submitted to this or any other institution to get any other degree or certificates. The assistance and help received during this investigation have been duly acknowledged.

Signature: -_______________

Bizuayehu Tadege

BAHIRDAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTING

Approval of thesis for defense result

I hereby confirm that the changes required by examiner have carried out and incorporated in the final thesis.

Name of Student: **Bizuayehu Tadege Tegegne** Signature____ Date 14/12/2022

As members of the board of examiners, we examined this thesis entitled "Amharic chatbot on Ethiopian civil code law using deep learning approach" by Bizuayehu Tadege Tegegne. We hereby certify the thesis is accepted for fulfilling the requirements for the Award of the degree of Masters of Science in "Information Technology".

Name and signature of Members of Board of Examiner for approval

1. Name of Advisor	**Signature**	**Date**
Dr. Tesfa Tegegne (Associate Professor)		20/12/2022
2. Name of External Examiner	**Signature**	**Date**
Dr. Million Meshesha		December 14, 2022
3. Name of Internal Examiner	**Signature**	**Date**
Dr. Abdulkerim Mohammed		23.12.2022
4. Name of Chair Person	**Signature**	**Date**
Mr. Belete Biazen (Assistance Professor)		20/12/2022
5. Name of Chair Holder	**Signature**	**Date**
Dr. Abdulkerim Mohammed		23.12.2022
6. Name of Faculty Dean	**Signature**	**Date**
Asegahegn Endalew		22-12-2022

Faculty Stamp

Acknowledgement

I would like to thank my advisors, Dr Tesfa Tegegne, for their constant guidance and support from title selection to now. He used his time to comment on my work, follow up on my study, and direct me in the proper way. Additionally, I want to express my gratitude to Amhara regional state supreme court employees for their deepest help in gathering civil code documents. I want to express my gratitude to my family for their love, tolerance, and support. I also admire the priceless families that have helped and inspired me throughout my life. I ended by expressing my gratitude to my friends for supporting and encouraging me throughout my career. A special thanks to Bahir dar university; Bahir dar institute of technology provides my master's degree opportunities.

Abstract

A chatbot is a computer program used to interact between humans and computer systems using natural language processing via speech or text. In this research, we developed Amharic chatbots that support lawyers and other users of chatbots in assisting with civil codes i.e., about people's rights and inheritance rights. The majority of the population either do not afford to hire a lawyer or do not know the civil code. As a result, this study aims to develop an Amharic chatbot that can support individuals, lawyers, and judges in making legal decisions. We used an experimental research methodology to develop a model for civil code laws in this research. In designing a model, we used RNN (LSTM and BiLSTM), and transformer encoder-decoder chatbot in training and testing a model. We used BLEU score metrics and user acceptance testing to measure the model's performance. The BLEU score measurement technique measures the similarity between the target civil code answer and the model-predicted civil code answers on the user's query. After experimenting with each model with different hyperparameters, we got a BLEU score of 9.88% in the transformer model. Compared with the LSTM and BiLSTM models, the transformer model achieves good performance on training time, memory usage, and prediction accuracy. Generally, the design model responds to the user's query based on the learned weight in training. Improving the performance of a model for the transformer model, the dataset size, and different user utterances involvement remains a challenge.

Keywords: -Amharic language, LSTM, BiLSTM, chatbot, conversational agent

Table of Contents

List of abbreviation

AI:	Artificial Intelligence
ALICE:	Artificial Linguistic Internet Computer Entity
API	Application Programming Interface
BERT:	Bidirectional Encoder Representations from Transformers
Bi-LSTM:	Bidirectional Long Short-Term Memory
CNN:	Convolutional Neural Network
DC:	District of Columbia
DL:	Deep Learning
DNN:	Deep Neural Network
GRU:	Gated Recurrent Unit
GUI	Graphical User Interface
IR:	Information Retrieval
LSTM:	Long Short-Term Memory
ML:	Machine learning
NLP:	Natural Language Processing
POST:	Part Of Speech Tagging
QA:	Question Answering
RLU:	Rectified Linear Unit
RNN:	Recurrent Neural Network
RQ:	Research Question
TF-IDF:	Term Frequency-Inverse Document Frequency

List of figures

List of tables

CHAPTER ONE: INTRODUCTION

1.1. Background of the study

Nowadays, the advancement of technology provides an easy and comfortable environment for human beings. With the introduction of linguistics as a science, language has been one means of communication between human beings(Hammarström, 2016). Around the world, 7151 languages are used in daily communication between humans and societies, and most of those languages have their writing and grammatical structures(SIL international, 2022). In Ethiopia, over 80 languages are used in day-to-day communication. Of those languages, Amharic is the dominant and official working language of the government of Ethiopia (Mindaye et al., 2010). Despite this, digitization of the Amharic language is limited and a resource-constrained language.

The introduction of artificial intelligence(AI) in 1950 played a significant role in natural language processing(NLP), computer vision, and model development in different application areas(Caldarini et al., 2022a; Shabbir & Anwer, 2018). Nowadays, AI systems can learn from data and analyze various algorithms to produce users' desirable output(Kaul et al., 2020; Shabbir & Anwer, 2018). In a digital world, we can use AI to design and develop an intelligent agent that can perform the creative function of humans, automatically find solutions to problems, and draw conclusions to make decisions based on data patterns learned via NLP. This capability of a model in AI is designed with the help of machine learning, deep learning, and predictive analysis to enhance agents' learning, reasoning, and thinking ability(Shabbir & Anwer, 2018).

Researchers use the terms chatbots, conversational agents, dialogue systems, advisory systems, and virtual assistance to describe software-based interaction. They use natural language to interact with users; they take input from users using natural languages such as voice or text and responds to users based on the provided data and pre-trained data patterns (Athota et al., 2020; Klein et al., 2017; Motger et al., 2021; Wahde & Virgolin, 2022). Or it is an AI software that can interact with users using natural language through websites, mobile apps, and other media remotely(Dhankhar & Delhi, 2018).

Chatbots are developed to offer services to different areas such as healthcare, agriculture, business, law, and other institutions(Allouch et al., 2021; Alnefaie et al., 2021).

The pattern-based, information retrieval-based and generative-based chatbots are the three primary kinds of chatbots based on the type of response they generate(Wahde & Virgolin, 2022). The pattern-based chatbot, first introduced in 1966, resembled a psychiatrist who gradually guides a patient through thoughtful dialogues, frequently changing and reflecting on the user's words. Eliza is an example of such a chatbot. The other chatbot is focused on information retrieval. This category selects a suitable sentence from a huge dialogue corpus, i.e., a database of stored talks, and it works based on the cosine similarity concept(Wahde & Virgolin, 2022).

While the two previous categories of chatbots rely on pre-existing utterances, either in pre-defined patterns or retrieved from a dialogue corpus, the generative chatbots generate their responses using statistical models known as generative models. When one intends to model and predict, each token of the input sentence is turned into an embedding and fed to the encoder (RNN). This recurrent neural network generates the output sentence. This information is passed to the decoder, another recurrent neural network that generates the output sentence(Wahde & Virgolin, 2022).

So, by considering such chatbot types, we can develop distinct chatbots in different languages for different purposes. Nevertheless, previously designed chatbots had some limitations in responding to users' queries. Hiring a lawyer in Ethiopia and developed countries is very expensive and unaffordable. Hence, people with low income and a gap in knowing the country's civil code are cumbersome to follow their cases and get Judgment promptly. Especially in Ethiopia, Due to the escalation of conflict, the number of cases is increasing from time to time. At the same time, the vast majority of Ethiopians are farmers, uneducated, and unaware of civil law, making it difficult to resolve disputes quickly.

In addition to that, hiring lawyers for a particular case is costly, and they cannot afford to hire a lawyer, so they waste their time and money without getting what they desire. This research aims to design an Amharic chatbot on Ethiopian civil code law using a deep learning approach that tackles the above problem and serves as a supportive tool for those who need to know civil codes. A lawyer provides fast responses about civil codes to help customers improve their benefits. It answers questions for users (plaintiffs and defendants) by analyzing user input to a civil code and providing a general solution or direction.

In this area, further research was conducted; in 2020, Asimare conducted an Amharic bot model that consulted Ethiopian published law. The researcher uses a machine learning approach with rule-integrated architecture(Asimare, 2020). In 2018 LAWBO: A Smart Lawyer Chatbot is developed for a judicial system that provides lawyer's services. The research is conducted on English conversations using deep learning and a dynamic memory network(Shubhashri et al., 2018).

All over the world, a crime is committed by someone knowingly or unknowingly. However, people are being abused without knowing whether they are guilty or not for an extended period. Jerry Hartfield was arrested for 35 years in the USA for a case of murder from 1977 up to 2012(Initiative, 2017). Robert DuBoise was also arrested for 37 years in rape and murder convictions(Press, 2021). Still, they are free after a long time with new evidence. Furthermore, chatbots can be used by a plaintiff or defendant without any worries or cost to know about their case and solution. The proposed model is necessary for lawyers, users, and researchers to conduct further research.

1.2. The motivation for the study

Artificial intelligence (AI) plays a significant role in supporting humans today by providing a platform for designing a model that mimics human behavior in various domains and delivers remote service. Nowadays, human tasks are mostly done using a computer application to facilitate work, reduce unwanted wastages, and get high benefits in different companies worldwide. One example of such an application is a chatbot, a computer program that interacts with humans and machines using natural language via text or voice.

In Ethiopia, most of the population is uneducated and requires support from experts to get clarity for some cases. However, most live in rural areas and are also low-income generators. While this happens, they try to know the issues by going long distances in person. So, the motive to do this work is why they waste their time and money by traveling a long distance to know their problem.

A chatbot plays a crucial role in such a case in assisting humans based on their case and the pre-trained data. Especially in our country, many problems arise in criminal and other civil code laws, so knowledge of the law is essential for people to address their issues in less time and cost. In general, the significant issues addressed in this research work are assisting every person on Ethiopian published law remotely and providing a direction for their problem. There are previous works in this area, but they are not addressed multiturn responses, different user utterances, and long-range dependency on the user's query.

1.3. Statement of the problem

In recent years, the development of conversational agents(chatbots) has been a hot research area. A chatbot is a computer program used to interact with humans and it fulfills their needs, and gives the response to the user query, and sometimes they are capable of executing tasks also(Jwala et al., 2019; Vijayaraghavan et al., 2020). However, the development of chatbots in a different language does not resolve the problems applied in the Ethiopian language.

In Ethiopia, more than 82 languages are spoken, of which Amharic is the most spoken and work language of Ethiopia(Mindaye et al., 2010). But, still a challenge to employ current chatbot technologies.

Moreover, different conflicts and disputes arise between families, friends, and groups in Ethiopia. As a result, knowledge of the law is vital. A person might be accused in different cases and be called to court; however, most of the population do not know legal issues like civil codes and other laws. To this end, people mainly depend on hiring a lawyer or prospectors to help the defendants. But the cost of a lawyer is unaffordable for the majority of the Ethiopian population; therefore, developing a chatbot that can be used as a virtual lawyer is essential.

The absence of such supported technology poses a problem for the justice system and individuals in Ethiopia and developed countries because individuals may not be aware of written civil codes. As a result, people spend money and time defending their claims and hiring lawyers. Especially in Ethiopia, it is difficult to get and hire a lawyer, or the knowledge of the law is very little, as most of our people are illiterate, semi-literate, and unaware of civil codes(Hiil, 2020). In addition, as the Amhara regional state annual report showed, more than 601525 cases are reported in civil code law than in other laws such as criminal code, commercial, and other laws. The developed chatbot is accessed through their mobile phone or web-based, and the interaction is text-based.

To tackle such problems developing a law chatbot has immense advantages. In this area, several conversational chatbots are designed for different purposes in different languages, such as English (Shubhashri et al., 2018), Amharic (Asimare, 2020; Hunde, 2021) etc. But the Amharic language has its grammatical structure, so chatbots developed in English cannot be directly applied. In addition, most of the work used rule-based and pattern matching in designing a model. Still, such chatbot models are unsuitable for a new user query because the response of a new user query is unrelated to the user query or does not provide a valuable response as they desire(Caldarini et al., 2022b). Also, most works do not support multi-turn responses with users because the researcher did not incorporate different user utterances in designing a model(Asimare, 2020). So, the main task of this work is to design a generative Amharic chatbot by including the above problems. To this end, this research attempts to answer the following questions.

RQ1: What features are extracted in the Amharic chatbot from collected data using the feature extraction technique?

RQ2: Which deep learning algorithm is suitable to develop a model that understands users' queries and responds appropriately in the Amharic language?

RQ3: To what extent a law chatbot predicts civil law in the Amharic language

1.4. Objective of the study

1.4.1. General objective

The main objective of the study is to design and develop an Amharic chatbot on Ethiopian published law using a machine learning approach.

1.4.2. Specific objectives

The following specific objectives are formulated to achieve the general objective.

- To review literature such as available algorithms and techniques in Amharic as well as other languages

- To collect and prepare dataset(corpus).
- To design a chatbot model for predicting law cases.
- To develop a prototype for the designed model.
- To test the performance of the Amharic chatbot.

1.5. Scope and limitation of the study

In this research work, the scope of the study is designing and developing an Amharic chatbot for Ethiopian published law using a deep learning approach. The study considers Ethiopian civil code laws because most cases are recorded on civil code laws, such as people's rights and inheritance rights, which are shown in detail. As the Amhara regional state annual report showed, more than 601525 cases are reported in civil code law than in other laws such as criminal code, which reported as 150213 cases in criminal code. In addition, the government may hire a lawyer to follow the defendant's case for criminal code laws; however, such a mechanism also rears rather than critical crimes. Thus, the focus of this study is on the Ethiopian civil code.

This work was not considered audio files, criminal codes, family codes, commercial codes, or other laws. In addition, the designed chatbot performs a conversation using text data but does not appropriately respond to queries unrelated to civil code laws; the research does not support voice-based interactions. The developed chatbot is a closed domain, which works only on civil code law, and the chatbot is also a task-based chatbot that performs some tasks based on the user's need on civil code law. The limitations of this work are the chatbot used textual data or did not support voice-based interaction. In addition, stemming and lemmatization are not performed in this research work.

The DL approach was followed to conduct the research because it is suitable for performing the generative chatbot model. In addition, the experimental research methodology was followed in this research work. It is used to identify the best DL algorithm through various experiments done by changing the parameters on each of the used algorithms.

1.6. Significance of the study

The investigation is crucial for researchers, lawyers, and users. The researcher provides an opportunity to conduct further research, like speech-based Amharic chatbots that offer a service for the disabled.

For lawyers, an Amharic chatbot is essential for saving time and money when serving customers from various areas who want to know whether their case will go forward or not. The lawyer can earn a large sum of money in a short period by submitting a query to it, which responds to a solution promptly.

The designed model is also fundamental to saving time and money. It provides consultancy to users before coming to court, and a chatbot saves them money. Because hiring a lawyer is very expensive in Ethiopia and developed countries, a chatbot provides clues about civil law based on the cases to users.

1.7. Thesis organization

The reminder of this thesis is organized as follows. The introduction, problem statement, study objective, the scope and limitations of the research, and the importance of the study were discussed in chapter one. The Amharic language, Natural Language Processing (NLP) methods, and literature reviews that are carried out for Amharic NLP research are addressed in Chapter two. The proposed Amharic chatbot model and techniques are presented in Chapter three. A general experimental investigation, an assessment of the suggested algorithms, and a comparison with selected machine learning algorithms for chatbot models for Amharic civil code were all included in chapter fours. Finally, the study's conclusion and future work were addressed in the last chapter based on the findings and outcome.

CHAPTER TWO: LITERATURE REVIEW

2.1. Overview of natural language processing

Human languages are born, grow, and in the process, they also die. Natural language processing introduces in 1975 to process languages using computer algorithms (Ayalew, 2021). NLP is a field study that processes human languages in various language ambiguity studies such as semantic, syntactic, grammatical detection, and part of speech tagging (POST). The study of such NLP allows interaction between computer and human languages(Jwala et al., 2019). The main goal of NLP is the manipulation of languages in a computer system using different algorithms. The advance of NLP provides several valuable tools and approaches for implementing natural language to interact with a computer system, and the interaction may apply in various ways.

2.1.1. Overview of Amharic language

The Amharic(አማርኛ) language is one of the second most spoken Semitic languages next to Arabic around the world and follows left to right writing system also. The language has its alphabet(Fidel), also called hohyat (ሆህያት), adopted from Geez, from Ha to pe(ከሀ እስከ ፐ), a total of 33 Fidel and each of the alphabets(Fidel) have seven letters (Gereme et al., 2021; Hagos & Asefa, 2020).

The Amharic language is the working language of the federal government of Ethiopia, primarily spoken in the Amhara regional state as a mother tongue. In addition, the language is also used as an additional working language in America, Washington DC, and most Ethiopian documents, news, and jobs are written in the Amharic language(Gereme et al., 2021; Wondwossen Mulugeta & Michael, 2014). However, while this is happening, the language is under-resourced for processing a computer-based application in different aspects.

The Amharic language is one of the complex morphological analysis languages to process the root of a term. Morphology is the internal structure of word-formation from different affixes of a word(Ayalew, 2021; Wondwossen Mulugeta & Michael, 2014). By using various affixes to generate derivational and inflectional morphemes, morphology analyzes the pattern of word formations such as inflection, derivation, or compound word formation(Ayalew, 2021). The smallest meaningful unit of a word is called a morpheme, which comprises a phoneme or group of phonemes(Ayalew, 2021). For instance, from the noun ልጅ "child," another noun, ልጅነት "childhood," from the adjective ደግ "generous", the noun ደግነት “generosity" can be derived. Generally, there are several different word classes in the Amharic language, including nouns, verbs, adverbs, pronouns, adjectives, etc.(Ayalew, 2021).

2.2. Meaning of chatbot

Now we live in the technology era, in which devices and applications do tasks without the involvement of humans; such devices or applications are called intelligent devices, which are designed and developed in artificial intelligence, or machine learning approach. Chatbots are software applications designed to perform tasks like humans by interacting with human languages such as text, voice, and sometimes an image. They support users as other persons(Jwala et al., 2019; Wahde & Virgolin, 2022).

The chatbot application operates in artificial intelligence via the website, mobile app, and other plugins to provide services in different domains. AI techniques are applicable for chatbot technology to identify user feelings, intent, utterance, and meaning to respond to queries formulated in natural language. It is important to do activities remotely, and users get a response to what they desire without any wastage of time and money. It also reduces the manual duplication of cases in the court system. In Ethiopia, published laws are stored as benchmarks for courts to follow users' cases based on the data they receive and make a decision. However, the user may not have such access to the published law, so they follow their case by going in person. So, the task of the chatbot in this work is to guide users based on their case regarding Ethiopian published law documents, mainly civil code law.

A civil code is a set of laws that deal with the most important aspects of private law, such as property, family, contracts, torts, unlawful enrichment, and business-related actions and practices(Biresaw, 2021). According to the Amhara regional state supreme court annual report, more than 601525 cases are recorded from woreda to regional state on civil code. Most cases are delayed because the people do not understand their problems when they come to court. This is because the problem is so complex, and the elements to be evaluated are so various that such judgments are only valid if the field of law in question (civil law or the code) is precisely understood the user's case and the user also easily communicate their case in considering the civil codes to courts(Biresaw, 2021).

2.2.1. Classification of chatbot

To design and develop a chatbot, first identify the classification chatbot technology. The knowledge domain, the service provided, the goals, the input processing, and the response generation method are some of the parameters used to classify chatbots(Adamopoulou & Moussiades, 2020a, 2020b).

The chatbots are classified into two, i.e., closed and open domains based on the knowledge domain. Open-domain chatbots respond to appropriate answers to any user's query. It is designed to answer users' questions by analyzing the query's intent with the chatbot's knowledge, also called a general-purpose chatbot. In contrast, closed-domain chatbots are designed to respond to a specific knowledge domain. It does not work for questions unrelated to the pre-defined data(Adamopoulou & Moussiades, 2020; Nimavat & Champaneria, 2017).

The technique of processing inputs and generating responses is considered when classifying based on input and response generation methods. Three approaches are used to produce the desired replies: rule-based, retrieval-based, generative models and a hybrid of them. The response is produced when the model rules match the users' query in the rule-based model. The retrieval-based models respond to the user's query based on the similarity of the query with the predefined data (trained data). The other model used for chatbot technology is generative-based; it responds to a query more appropriately than the two models. It answers questions by analyzing the current and previous conversations and the response and queries formulated via text, voice, and images. These human-like chatbots are designed using machine learning and a deep learning approach(Adamopoulou & Moussiades, 2020).

Chatbots are classified into informative, task-based, and conversational based on the goal of the dialogue or conversation(Adamopoulou & Moussiades, 2020b). The informative chatbot provides information to users stored in a fixed source (e.g., Facebook). The task-based chatbot, is designed to serve users as a virtual assistant for a specific task. E.g., booking in restaurants and booking a flight etc. The chat-based/conversational chatbots are a chatbot that interacts with humans as other people. This chatbot responds to users' queries by analyzing user intent in various ways (Adamopoulou & Moussiades, 2020; Nimavat & Champaneria, 2017). In general, the categories of chatbot are presented in Figure 2.1.

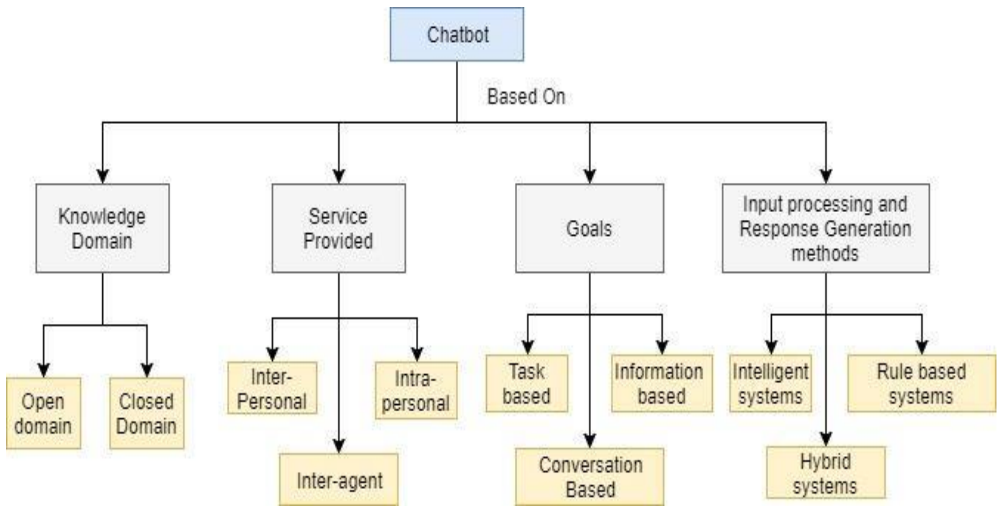

Figure 2. 1: Categories of chatbots(Nimavat & Champaneria, 2017)

2.2.2. Evaluation of existing chatbot

The advancement of AI technology has significantly changed chatbot development; several chatbots were developed after introducing the first chatbot application, i.e., EilzaBot, where different bots are developed to provide customer service (Nuruzzaman & Hussain, 2018). According to Nuruzzaman & Hussain, (2018), existing chatbot technologies are described as follows.

- **Eilzabot**

The first computer program made natural language communication possible on the computer system in 1966 Massachusetts institute of technology[M.I.T) lab(Nuruzzaman & Hussain, 2018; Shawar & Atwell, 2015). The bot works based on the keyword(pattern) matched with the query to training data, and it follows a rule-based technique that the user query does not contain keywords defined in the database; it does not respond to any result. Eliza aims to provide service to patients virtually. If the user query does not match any keyword in a knowledge base, Eliza provides a chance to improve on patient queries. The main drawback of Eliza is that it cannot learn patterns from user interaction to discover new patterns from user queries (Agumasie, 2012; Nuruzzaman & Hussain, 2018).

- **Mitsuku**

This chatbot technology was designed using a supervised machine learning approach, and it is a human-like chatbot developed using an Artificial intelligence machine learning framework. The bot works based on the heuristic pattern matching as Eliza works, but the special capability of this chatbot is that it provides direction to the data in the database and the users can communicate based on the directions. In addition to this, the bot module can learn from users' conversations and improve the response capability in processes. Nevertheless, it requires a huge dataset to operate professionally to users' conversations. It is integrated on Twitter and telegram to interact with users, and it can store users' personal information in communication; such ability makes it secure and better than the Eliza chatbot(Agumasie, 2012; Nuruzzaman & Hussain, 2018).

- **A.L.I.C.E bot**

Artificial Linguistic Internet Computer Entity (A.L.I.C.E) is an open-source computer program designed and originated by Wallace in 1995. This chatbot technology is based on the modified version of the ELIZA bot, the conversation between users and bots is performed with query pattern matching. It was the first chatbot with a Knowledge Base implemented in the A.I.M.L language, the data stored in the knowledge base. It generates responses based on the dialogue history and user utterance using a collection of artificial intelligence markup language (A.I.M.L) templates. In the beginning, A.I.M.L takes the user sentence as input and stores it in a category.

Each category has a response template and a set of conditions that give the template meaning referred to as context. The model then takes possession of it and compares it to nodes in the decision tree. When the chatbot recognizes the user's input, it responds or take action. The A.I.M.L templates use recursive approaches to repeat the user's input utterance, and the replies are not necessarily understandable. As a result, string-based rules must be used to determine whether the response produces an accurate or meaningful result(Agumasie, 2012; Nuruzzaman & Hussain, 2018; Shawar & Atwell, 2015).

- **Cleverbot**

Many chatbots are designed and developed; Cleverbot is the one popular entertainment chatbot that uses rule-based AI techniques to interact with humans. Through crowdsourcing, it collects vast data based on online conversations with people. Cleverbot's responses are not preprogrammed like those of other chatterbots. Rather, it mimics real dialogue by learning from user input and communicating with feedback.

Cleverbot examines all keywords or phrases that match the user's input. It responds to the input by looking up how a user responded to the input when asked in one of its recorded conversations. Cleverbot is unusual because it learns what users have said to it in past saved talks and then applies that information to ongoing discussions.

The bot has its human representation that expresses emotions to add reality to the interaction. Cleverbot's underlying technology processes not just spoken and written interactions but also facial emotions and gestures, resulting in a more proper discussion. Cleverbot's limitations include unpredictable responses and the bad idea of unexpectedly changing the subject and responding without context. It cannot hold a long conversation, is not very good at translating languages, and is not recommended for kids(Agumasie, 2012; Nuruzzaman & Hussain, 2018; Singh, 2020).

- **IBM Watson**

Watson is a rule-based AI chatbot produced by IBM's DeepQA project for information retrieval and question-answering. It uses natural language processing and a hierarchical machine-learning technique. It employs a variety of processes to identify and assign feature values to the generated response, such as names, dates, geographic locations, and other entities. The machine-learning algorithm then learns how to combine the values of these features to get a final score for each response, and based on the score, it ranks all possible replies, finally deciding on one as the best. Watson examines the phrase structure and grammar of the query using a range of technologies, including Hadoop and the Apache Unstructured Information Management Architecture (UIMA) framework, to better understand what is being asked(Agumasie, 2012; Nuruzzaman & Hussain, 2018). Watson's fundamental cognitive computing technology has immense applications. Because it can perform text mining and advanced analytics on massive amounts of unstructured data and handle massive amounts of information, and with additional input, the application will be able to identify enough patterns to generate reliable predictions. Aside from the benefits of Watson, there are some limitations, such as the reality that it does not process structured data directly, does not use relational databases, has a higher maintenance cost, is targeted at larger organizations, and takes more time and effort to train Watson to use its full potential(Agumasie, 2012; Nuruzzaman & Hussain, 2018; Singh, 2020).

- **Microsoft Luis**

Microsoft's Language Understanding Information Service (LUIS) is a domain-specific AI engine. It uses intents and a custom build domain entity model to process natural language and information. LUIS uses Natural Language Processing (NLP) against Big Data to find intentions from a sentence. It is made to recognize important information in discussions, evaluate user goals, and extract data. Active learning is also used to constantly increase the quality of natural language models. A model begins with a list of common user goals, such as booking a flight or contacting the help desk. After the intentions have been determined, the user provides utterances (sample words). Then mark the utterances with any specific information the user wants LUIS to extract from it. After the model has been created, trained, tested, and published, it is ready to receive and process utterances (Nuruzzaman & Hussain, 2018).

The utterance is received as an HTTP request by LUIS, responding with the extracted user intentions, and to design and deploy a solution more quickly, powerful development tools are integrated with customizable pre-built apps and entity dictionaries, such as Calendar, Music, and Devices. Dictionaries are compiled from the internet's common knowledge and contain billions of entries, assisting the model in identifying useful information from user discussions. The main disadvantage of LUIS was that it required Azure subscriptions. On the other hand, LUIS works perfectly with the Azure Bot Service, making it simple to build a sophisticated bot(Nuruzzaman & Hussain, 2018).

- **Google Dialog flow**

Virtual assistance plays a great role in an organization that provides service to their customer in various ways. Form that assistance, google Dialog flow is the one API.AI developed by Google, and part of the google cloud platform. It enables app developers to give their consumers speech and text interactions enabled by machine learning and natural language processing technology(Agumasie, 2012; Kanakanti & Sabitha, 2020; Nuruzzaman & Hussain, 2018). This allows them to concentrate on other important aspects of app development rather than understanding complex grammatical rules. Next, Dialog flow recognizes the user's purpose and context. Then, using entities, match user input to specified intents and retrieve relevant data from them. Finally, provide the conversational interface the ability to respond. The lack of a mobile device version, an interactive user interface, and poor documentation are all limitations of Dialog flow(Agumasie, 2012; Nuruzzaman & Hussain, 2018).

- **Amazon lex**

Amazon Lex is a tool created by Amazon that uses the same technology as their successful Alexa product. Developers can use the toolkit to construct text or speech bots(Kumar & Rajagopal, 2020; Mierzwa et al., 2019). Amazon launched it, and it offers deep learning functionality as well as the flexibility of natural language understanding (NLU) and automatic voice recognition (ASR) to create extremely interesting user experiences with lifelike, conversational interactions(Kumar & Rajagopal, 2020). Thanks to Amazon Lex's integration with AWS Lambda, users may quickly activate functions to execute back-end business logic for data retrieval and updates. The problem with Amazon Lex is that it is not bilingual and currently only supports English. Lex, unlike Watson, has a specific online integration procedure to follow. The dataset preparation is difficult, and the utterances and entity mapping are crucial(Agumasie, 2012; Nuruzzaman & Hussain, 2018).

2.3. Chatbot design techniques

Several designing approaches are proposed to develop a task-oriented or non-task-oriented chatbot(Adamopoulou & Moussiades, 2020a; Allouch et al., 2021). The three main approaches are discussed below.

2.3.1. Rule-based approach

Over the last few years, developing a rule-based chatbot was one of the techniques used before introducing an advanced approach to chatbot technology. This approach is the traditional method that uses handcrafted rules by the developers. Such complex hard-coded rules are not applicable for flexible environments to collect all the rules, and the data may vary over time. Also, it is tedious to develop such technology, but the bot has high accuracy to the provided rules in conversation.

The rule-based techniques are built on predetermined simple queries and possible responses. It does not require the use of machine learning, nor does it necessarily require language processing. They are designed for simple questions and may fail to answer more complex topics since they cannot generate responses. One-to-one input and responses are the characteristics of this type of chatbot. As a result, a bot would be programmed to follow the rules(Ayanouz et al., 2020; Wayesa, 2020).

2.3.2. Retrieval-based approach

With the advancement of digitization, the chatbot development approach also improved from rule base to retrieval-based. This design and development method is better than rule-based technology since it works based on some computation with the user query to the document corpus stored in development. Compared to the rule-based approach, this approach performs a similarity measure to identify the closest document to a given question and then replies to several documents based on high score rank. Therefore, the user query must be similar to the predetermined probable question to get responses unless the user may not get a solution for the questions(Akkineni et al., 2022).

2.3.3. Generative based approach

As their names suggest, generative-based models generate new responses word by word from the user's input. Thus, these models can generate entirely new sentences in response to user requests; but they must be trained to understand syntax and sentence structure, and the results may not always be of high quality or consistency.

Usually, a huge dataset of real-world conversational words is used to train generative models. The model learns vocabulary, syntax, and sentence structure through the data it has been fed. The algorithm should produce a suitable, linguistically appropriate response, depending on the input sentence, to achieve the overall goal. Typically, a Deep Learning algorithm based on an encoder-decoder neural network model is used in this method.(Caldarini et al., 2022b; Kapočiute-Dzikiene, 2020).

2.4. Introduction to artificial intelligence

Artificial intelligence (AI) simulates human intelligence in devices trained to think and act like humans. The name can also refer to any machine demonstrating human-like characteristics like learning and problem-solving. Introducing AI aims to design an AI model that works like humans in various industries, restaurants, commercials, and others(Chowdhury & Chakraborty, 2017). Those artificial machines use human interaction mechanisms in communicating with humans or other artificial machines like voice, text, and gesture.

2.4.1. Machine learning approach

Humans have utilized various tools to make certain tasks easier to complete. Several tools and devices have been developed based on the nature and capacity of the human brain. These devices made life easier for humanity by allowing them to address various demands, such as travel, industry, and computing. One among them is machine learning(Batta, 2020).

The term machine learning is usually connected with the name of Cornell University psychologist Frank Rosenblatt, who founded a group that designed a machine to recognize letters of the alphabet based on his theories about how the human nervous system works(Fradkov, 2020). According to Biresaw(2021), machine learning is the field of study that allows computers to learn without being explicitly programmed(Batta, 2020). It is a subfield of AI that the machine can learn things like a human in different aspects and perform tasks as humans do.

The ML approach is mainly categorized into several ways, but those are grouped into two, i.e., supervised and unsupervised, as shown in Figure 2.2, based on the data they used(Reddy & Babu, 2018).

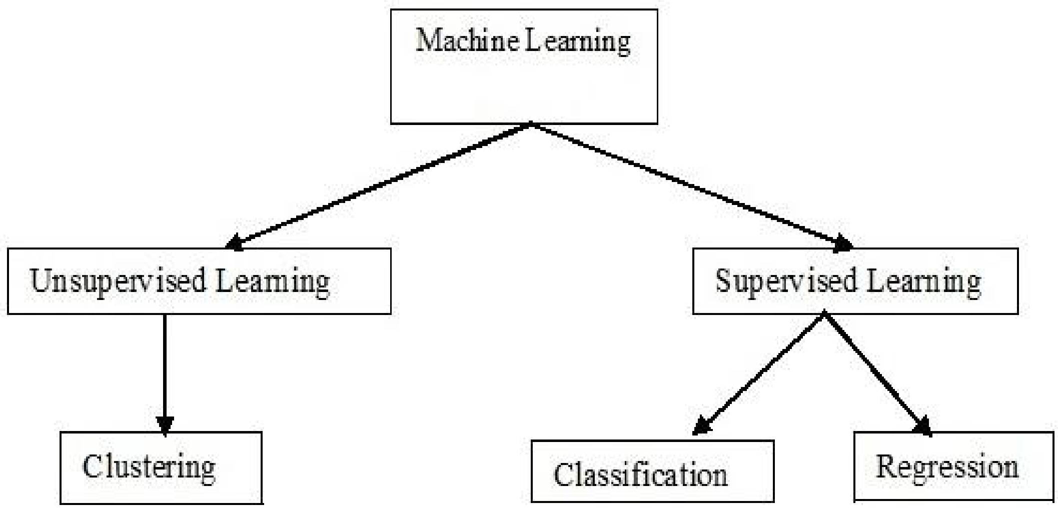

Figure 2. 2: Main types of machine learning(Reddy & Babu, 2018)

Supervised machine learning is an approach that uses label data for a given algorithm in developing a model as input and an output for the predetermined information. The task of supervised machine learning is to learn a feature that translates input to an output based on sample input-output pairs. It suggests a value from a set of labeled training instances. The supervised machine learning algorithms require external support, and the input dataset is split into two parts: train and test. An output variable in the training dataset must be predicted or categorized (Batta, 2020; Reddy & Babu, 2018). This machine learning approach is used for classification and regression, as described above. The SVM, DT, and Naive Bayes are some of the most popular supervised machine learning algorithms used in various application areas(Batta, 2020).

The other approach used in machine learning is unsupervised ML, which builds a model based on unlabeled data and uses it to estimate and characterize key properties of the data without any prior knowledge of the data(Batta, 2020; Reddy & Babu, 2018). This approach does not have predetermined input-output data; rather, the model learns from data. Unsupervised learning algorithms learn a specific feature from the data. When new information is introduced, it detects the data's class using previously known features. It's mostly employed for feature reduction and clustering(Batta, 2020). The K means algorithm is the most used algorithm for such an approach (Batta, 2020). Generally, the above-discussed algorithms are shallow learning algorithms. But shallow machine learning techniques are not applicable for complex NLP tasks unless they need to do feature engineering, which is time-consuming, expensive, and challenging to understand the syntax and semantic aspects of the input text data. So, deep learning approaches have recently been employed to generate competitive performance for several NLP tasks to develop such complex problems compared to classic machine learning approaches.

2.4.2. Deep learning approach

Deep learning(DL) is the process of learning a procedure for handling a problem, such as simple classification or flexible reasoning, using vast volumes of data utilizing deep neural networks(Torfi et al., 2020). DL applies to complex NLP problems because the model works as human neurons on the provided data to construct valuable features and predict accurate results (Torfi et al., 2020). Compared with the classical ML algorithm, the deep learning algorithm produces competitive performance in different NLP tasks. Several deep learning algorithms have been developed in time, such as RNN, LSTM, BiLSTM, and seq2seq, used in various studies. However, new essential features are added with the first introduced deep learning algorithms, i.e., RNN, which also draws back on using those algorithms. For example, most of the above algorithms are time consumers in training a huge amount of data and encoding a sentence. Therefore, they take on one feature at a time. So, to handle such a problem, transformer-based deep learning algorithms are introduced, which work as a parallel encoding of features.

RNN and its version

Researchers came up with the neural network concept to describe biological information processing systems in the 1940s(Digital, 2021). The feed-forward neural network, the multilayer perceptron model, is the simplest and has produced good results in many application tasks. However, training is more challenging due to the model's enormous computational complexity. Nevertheless, it is now possible to train large-scale and deep neural networks because of ongoing improvements in computer performance. In recent years, deep convolution networks have significantly advanced graphics and image processing, video and audio processing, and other domains. Simultaneously, recursive networks have performed well in sequence data like text and voice(See, 2019).

Initially, the recurrent neural network performed well in handwritten digit recognition and other sequenced text data processing. The name recurrent refers to repeating the same work for each sequence instance, with the outcome being dependent on prior computations and results. A fixed-size vector is formed by feeding tokens one by one to a recurrent unit to express a sequence. RNNs have a kind of memory of prior computations used in current processing. The RNN framework is suitable for several NLP applications, such as machine translation, human-machine conversation, and image captioning, and this capability has made popular in NLP applications in recent years (See, 2019; Tarwani & Edem, 2017; Young et al., 2018).

However, in practice, these simple RNN networks suffer from the classic vanishing gradient problem, which makes learning and modifying the parameters of the network's earlier layers extremely difficult. So, various networks, such as long short-term memory (LSTM), gated recurrent units (GRUs), and residual networks (ResNets), have been developed to address this issue, with the first two being the most commonly used RNN variations in NLP applications(Young et al., 2018).

The LSTM (Long Short-Term Memory) features more forget gates than a simple RNN. Just one method allows it to solve both vanishing and exploding gradient problems. Unlike a traditional RNN, the LSTM permits an error to propagate backward via an unlimited number of time steps(Young et al., 2018). It estimates the hidden state by combining these three gates: input, forget, and output.

The Gated Recurrent Units (GRU): A simpler gated RNN version known as GRU was developed with empirically similar performance to LSTM in most tasks. GRU is made up of two gates: a reset gate and an update gate, and it works in the same way as an LSTM but without the memory unit. As a result, it rapidly exposes all hidden content. GRU can be a more efficient RNN than LSTM because it is less complex(Young et al., 2018).

The general deep LSTM encoder-decoder framework was proposed that maps one sequence to another. The source sequence, which can be text in the original language (machine translation), the question to be answered (QA), or the message to be replied to, is encoded as a fixed-size vector using one LSTM (dialogue systems). The vector is utilized as the initial state of the decoder, which is another LSTM. The decoder generates tokens during judgment, updating its hidden state with the most recently generated token(Young et al., 2018).

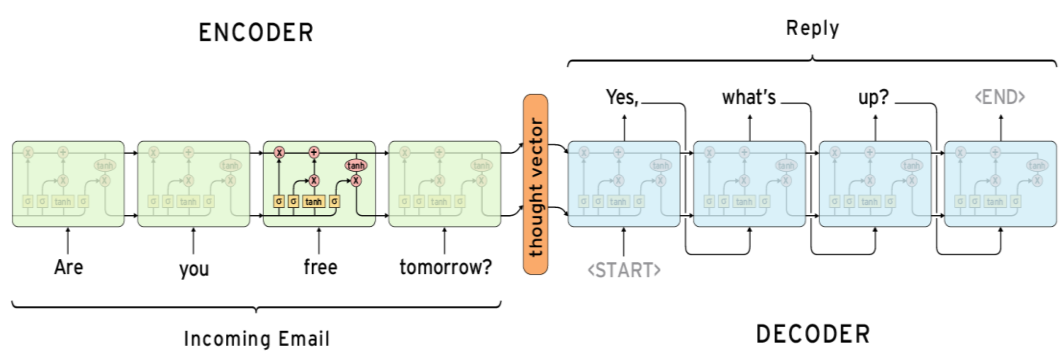

Figure 2. 3: LSTM Encoder decoder architecture(Sojasingarayar, 2020)

As shown in Figure 2.3, the LSTM-based encoder-decoder applied in sequence to sequence (Seq2Seq) model for language translation and conversational chatbot, the Sequence-to-Sequence approach, has become the best framework. The two RNNs, an encoder and a decoder, make up the system. Each time step, the encoder processes one symbol (word) from a sequence (sentence). Its goal is to turn a sequence of symbols into a fixed-size feature vector that encodes only the sequence's significant information while reducing the rest. Local information flow from one end of the sequence to the other can be shown as data flow in the encoder along the time axis(Sojasingarayar, 2020).

Each hidden state affects the following hidden state, and the last hidden state can be considered as the sequence's summing up. This state is known as the context or thought vector because it represents the sequence's intent. The decoder constructs a new sequence from the context, one symbol (word) at a time. The decoder is influenced by the context and previously created symbols at each time step(Sojasingarayar, 2020).

But there are a couple of drawbacks to employing this model. The model's inability to handle variable-length sequences is the most problematic. Unfortunately, it's also common since nearly all sequence-to-sequence applications use variable-length sequences. The vocabulary size is the next factor to consider. As the vocabulary size increases, encoding and decoding that vocabulary is a tedious task, slowing down the training process.

The attention mechanism plays a significant role in the vanishing problem and variable-length sequence, enabling the decoder to look at the input sequence selectively while decoding. Such a case saves the encoder the burden of encoding all helpful information from the input. Furthermore, instead of employing a fixed context (the encoder's last hidden state), a distinct context vector is utilized to generate words at each time step in the decoder. The attention mechanism with encoder-decoder architecture performs best in a conversational agent and machine translation(Vaswani et al., 2017).

In addition, a new deep learning approach is introduced, which is a transformer-based approach suitable for conversational systems with encoder-decoder architecture. The transformer-based encoder-decoder works with a parallel computation to encode a sentence and use a multi-head attention mechanism in a sentence's encoder and decoder process. This way of encoding and decoding solves a vanishing gradient problem and reduces the training time.

2.5. Techniques applied in chatbot

2.5.1. Word embedding

A computer system does not understand sentences or words directly, so it should convert them into numeric values to make a computer understand them. Also, textual data are unsuitable for understanding sentences' intent in deep learning algorithms because the machine can only acknowledge numeric data. So to perform any task using textual data, text must encode to a numeric value known as a vector representation of words or sentences using different methods such as word2vec, GloVe, BERT, FastText, ELMO, and TFIDF are word embedding techniques(Wang et al., 2020). It is categorized into two, i.e., context-dependent and context-independent word embedding, as shown in Figure 2.4. This encoding method is introduced to make textual data suitable for the deep learning algorithm by extracting feature vectors for words in a sentence.

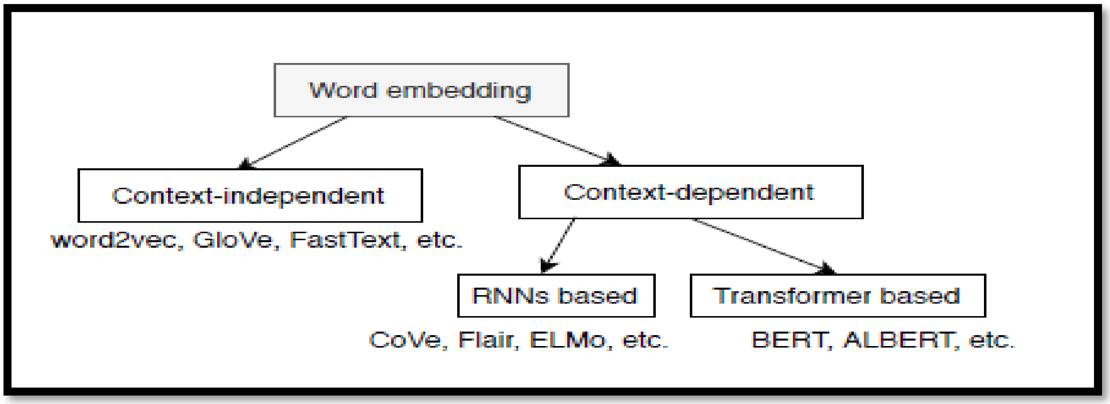

Figure 2. 4:A taxonomy of word embeddings(Wang et al., 2020)

The traditional approach of context-dependent embedding uses a static representation of a word in the vocabulary of the Amharic chatbot parallel corpora data set. Unfortunately, such encoding techniques are ineffective for word polysemy because they do not consider the context of a phrase when encoding words in multiple sentences(B. Wang & Kuo, 2020). Take, for example, the riverbank and the financial bank. In this case, the term bank has two distinct interpretations. However, the traditional method eliminates a word's meaning in the context of a sentence. There are a variety of algorithms for producing these embeddings, with word2vec, GloVe, and FastText being some of the most popular.

As opposed to traditional embedding, context-dependent embedding overcomes the limitations of constructing a word vector representation by assessing the context of a word in a sentence. For example, the bank has two meanings, i.e., bank-related and river-related context. BERT is pre-trained, the most popular and state-of-the-art technology to extract such vector representation, which is applicable for deep neural language models. It uses a deep neural network transformer encoder-decoder architecture to perform tasks(B. Wang & Kuo, 2020; C. Wang et al., 2020).

In encoder-decoder language modeling, the encoder converts word vectors (sequences of vectors) into sentence vectors. As a result, each row of the sentence vector encodes the meaning of each word in context, as shown in Figure 2.5. Then, feed the encoded to the context vector so the decoder network can use it. The decoder component receives the context from the encoder data via an attention mechanism or directly from the network's last layer of encoders. The decoder searches for a term that matches the word representation sequence's derived context.

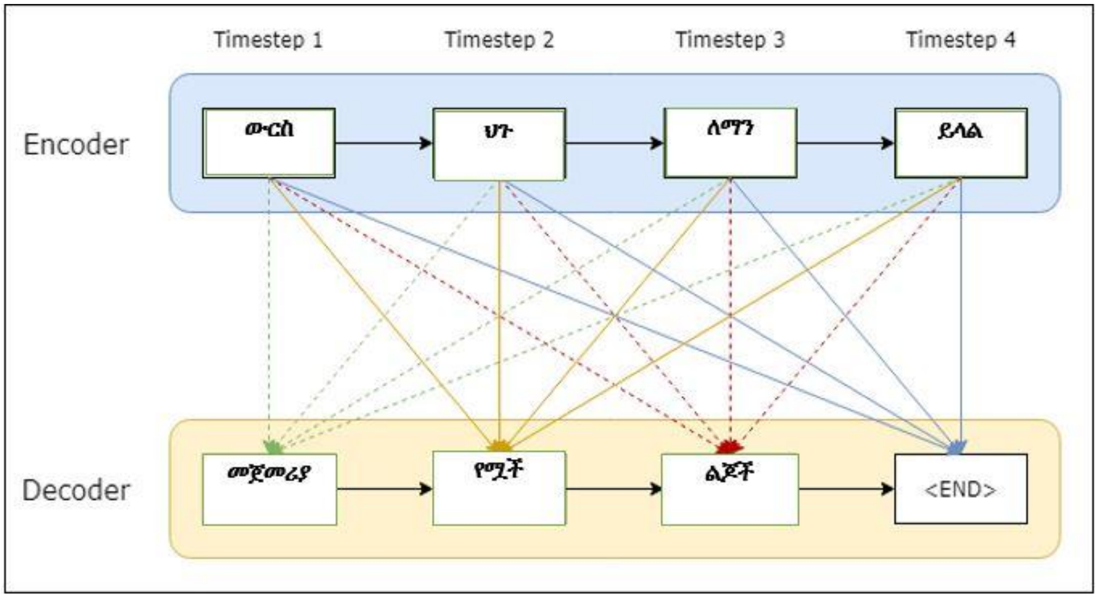

Figure 2. 5: Encoder decoder model workflow(Sojasingarayar, 2020)

2.5.2. Transformer neural network

With the complexity of NLP, researchers tried to simplify problems that arise in the NLP research area and computer vision. In processes, they found a suitable deep learning model, a transformer neural network; the model works based on the seq2seq model(Lin et al., 2021).

The principal aim of introducing this algorithm is the algorithm can encode sentences parallelly and also solve the problem of vanishing gradient, which arises in RNN encoding and decoding. So, the model reduces the training time and improves its performance because it is not vulnerable to vanishing gradient problems like RNN. Furthermore, the model can handle variable-length inputs to process user data and respond based on the inputs, as shown in Appendix V (Ghojogh et al., 2017; Lin et al., 2021).

2.6. Chatbot evaluation metrics

Evaluation is one task applied in chatbot development. For example, knowing whether the model identifies user intent and responds accordingly to the training data or not is measured using accuracy measures of BLEU score, precision, recall, and f1 score. So, after doing everything, the evaluation is performed in two ways: first, the dataset splits into two training and testing, and the model learns on the training dataset and then tests with the remaining dataset to measure whether the model understands the dataset or not. Second, the conversation is performed with a trained model and user query to show whether the model accurately predicts the user response (Vakili et al., 2020). Finally, evaluation metrics may be used to evaluate the model depending on the dataset, such as accuracy, BLEU score, recall, precision, and F1 score. But this work used BLEU score evaluation metrics.

BLEU score: An algorithm called (Bilingual Evaluation Understudy) was created primarily to assess how accurate machine-translated text was. Here, we apply the same methodology to evaluate the text response quality we get from the chatbot in response to user input. BLEU is one of the first evaluation metrics to demonstrate a strong correlation with human quality assessments. To compare the chatbot's answer text with the reference text in the ground truth test data, BLEU employs an n-grams modeling approach(Post, 2018). The BLEU score can be determined at the corpus or sentence level. In this case, we use the sentence-level BLEU score technique to calculate scores for each response text our chatbot generates. The BLEU score is used for various NLP benchmark activities, including text generation, conversational chatbots, question-answering, summarization, and machine translation(Dutta, 2019). The BLEU score metrics are based on n-gram word-by-word matching of the actual phrases in a civil code dataset with the predicted output(Blagec et al., 2022). In general, A BLEU score metric is used in this research work which measures the similarity predicted from the developed model with the actual civil code dataset.

2.7. Related works

Asimare and Mesha (2020) designed and implemented an Adaptive Bot Model to Consult Ethiopian Published Laws Using Ensemble Architecture with Rules Integrated. To conduct this research, they used the word to vector (Word2Vec) for semantic extraction of words, sequence to sequence model in generating responses and a rule integrated on the top to extract features from an Amharic law document. To evaluate the performance of a model, the researchers examine four neural networks to select the best-fitting neural network.

Finally, the researcher shows the performance and results of the law conversational bot by using perplexity, accuracy measurement, and f-1 score metrics. The model classifies the intent with 73.12% accuracy, and the user acceptance test was also employed. However, as they present, the model does not work for a multi-turn response or audio files, and the design model only works like the retrieval-based chatbot approach. So, this approach is not appropriate for new user queries, and the above gaps must be included to make the model full-fledged.

Hunde (2021) developed a bilingual chatbot that supports both Amharic and Afaan Oromo languages for assisting Ethio telecom customers on customer services. The researcher uses 2292 Amharic and Afaan Oromo texts from frequently asked questions on the telecom website. They used DNN and the Bi-LSTM model and evaluated using accuracy, precision, recall, and f1 score, and achieved a result of 82.6%, 85.7%, 82.6%, and 87.7%, respectively. Finally, the researcher evaluates the usability of the model on selected parameters such as attractiveness, response time, user friendly, efficiency, and system feasibility. However, the work does not show whether the design model supports a multi-turn response or not. In addition, the prepared dataset is minimal for chatbots, and the chatbot works on the data provided as a retrieval-based system.

Shubhashri et al. (2018) conducted research on a smart lawyer chatbot for the English language. This study supports lawyers who find case solutions from a huge document corpus. The researcher used a different technique to design a chatbot for English text, like a word mover distance dynamic memory network. They evaluated the design model within four testing accuracy measures on the judgment document and achieved 85% of test accuracy on yes/no questions. So, the work is exciting in articulating the paper, the techniques used for feature extraction, and which machine learning algorithm was used to evaluate the designed model. But the research does not compare the design model using deep learning algorithms like LSTM, GRU, Bi-LSTM, and transformer because that algorithm depends on sentence sequence and length in response to users. In addition, the dataset used to measure a model's performance was not addressed.

A medical chatbot is also developed by Gadge, (2021) using a deep learning approach. To investigate this research, they use a bag of words as a feature extraction technique and a deep learning approach to predict and evaluate the model of an AI-powered chatbot.

As we know, COVID-19 has been a very critical situation from 2019 up to now. The researchers try to fill this situation by designing an AI chatbot supporting societies living in rural areas. However, the dataset size and the model performance accuracy did not explicitly present.

Athota et al. (2020) designed a Chatbot for Healthcare System Using Artificial Intelligence. This research aims to create a medical chatbot using Artificial Intelligence to diagnose disease and provide basic details before consulting a doctor. In this work, ranking and sentence similarity measure calculations were performed using N-gram, TFIDF, and Cosine similarity. After this calculation, the designed chatbot responds to the user based on the ranking algorithm, which has a high value.

Mahendra N (2020) researched crop prediction using machine learning approaches. According to studies, 60% of India's population relies on agriculture and practices traditional farming methods. However, according to their findings, such farming does not enhance the yield of crops in India since farmers do not employ scientific input such as PH value, temperature, or soil type. As a result, the study's primary goal is to supply helpful information to farmers by studying the soil type, PH value, and temperature and selecting appropriate crops for land to improve crop productivity.

Researchers gather data from various sources, such as government websites, Agriculture College of VC Farm Mand, etc. To get an approximate dataset for the system, they consider the following attributes: soil PH, temperature, humidity, rainfall, and crop data, those parameters will be considered for crop prediction, and for the annual rainfall prediction, they collected the previous year's rainfall data. Then, they apply data pre-processing techniques to make the collected data suitable for machine learning models. Finally, they used two machine learning algorithms for prediction, i.e., the SVM algorithm for rainfall prediction and the Decision Tree algorithm for crop prediction.

The designed model is vital for farmers to farm with scientific input and improve productivity. However, the researcher mentions recommended crops in the evaluation section, not how much the model predicts based on rainfall data and land features. In addition, the collected dataset from various sources is not reliable for model accuracy today because weather conditions and rainfall vary over time.

Generally, this research is conducted to implement a civil code chatbot using a deep learning approach by including the gaps, such as multi-turn response between model and user, using a generative chatbot model.

- **Summary of related works**

Author's	Title	Methodology	Gap’s
(Shubhashri et al., 2018)	A smart lawyer chatbot	Word mover distance, dynamic memory network	Handling input complexity and splitting ratio, also used limited user query for a case.
(Asimare, 2020)	Designing and implementing the Adaptive Bot Model to Consult Ethiopian Published Laws Using Ensemble Architecture with Rules Integrated	word2vec, sequence to sequence, LSTM	Not work for a multi-turn response, the design model only works just like the retrieval-based chatbot approach, and management of model overfitting by changing data size, regularization technique, and splitting ratio
(Gadge, 2021)	Chatbots for medical purposes using deep Learning	The bag of word, and deep learning algorithm	It does just like the rule-based approach and predicts the disease accurately if the user inserts two or more symptoms

2.8. Chapter summary

Several studies have proposed different chatbots or conversational agents to solve real-time problems. Although some studies use different approaches, most chatbots are designed using rule-based and retrieval-based approaches. In this research, we have employed deep learning approach to design a chatbot for civil law documents. We choose an encoder-decoder deep learning strategy to develop a model based on the above literature and related work. It is the most advanced technology for conversational chatbots, benefiting from adopting an encoder-decoder language model to simplify complicated tasks. This study chose the LSTM, Bi-LSTM, and Transformer models.

CHAPTER THREE: DESIGNING AMHARIC CHATBOT MODEL FOR CIVIL CODE

3.1. Overview

This section discusses the method and techniques used in this study. The chapter presents the architecture of the study. In this study, an experimental research methodology is used. This chapter discussed the main components of the overall architecture of the proposed model. Section 3.2 presented the general overview of the proposed architecture. In Section 3.2.1, data preprocessing, section 3.2.2, feature extraction, 3.2.3, an encoder-decoder model overview is presented.

3.2. Proposed system architecture

In this section, we have discussed the overall architectural design of the proposed system, its main subcomponents, and their interactions. The proposed system architecture has four main components: pre-processing module, a word embedding module, a user input analysis module, and a possible answer generation module. The proposed system architecture is presented in Figure 3.1.

This section explains the system design, the input corpus from processing to delivery, and the activity is done at each phase. Pre-processing the input corpus, which includes tokenization and punctuation mark removal, is the first step that the architecture takes. The input corpus can be based on words or sentences. Following the pre-processing of the input corpus, the pre-processed corpus is converted to a numeric value, a one-hot vectorization vector. The dimensions of a one-hot vectorization are larger. As a result, we used Kera's embedding layer to transform it into a lower-dimension vector. Finally, we've provided our encoder-decoder model embedded corpus. The encoder model processes the input token and accepts the embedded token.

The decoder model accepts the encoder's final context vector as input. During training and validation, the decoder model performs slightly differently. The decoder model accepts the context vector from the last encoder hidden layer and the sequence of target embedding during training to predict the target output. However, during learning, we did not give the target token to the decoder model. As a result, the decoder model predicts the output based on the last encoder hidden layer's context vector and the prior predicted output.

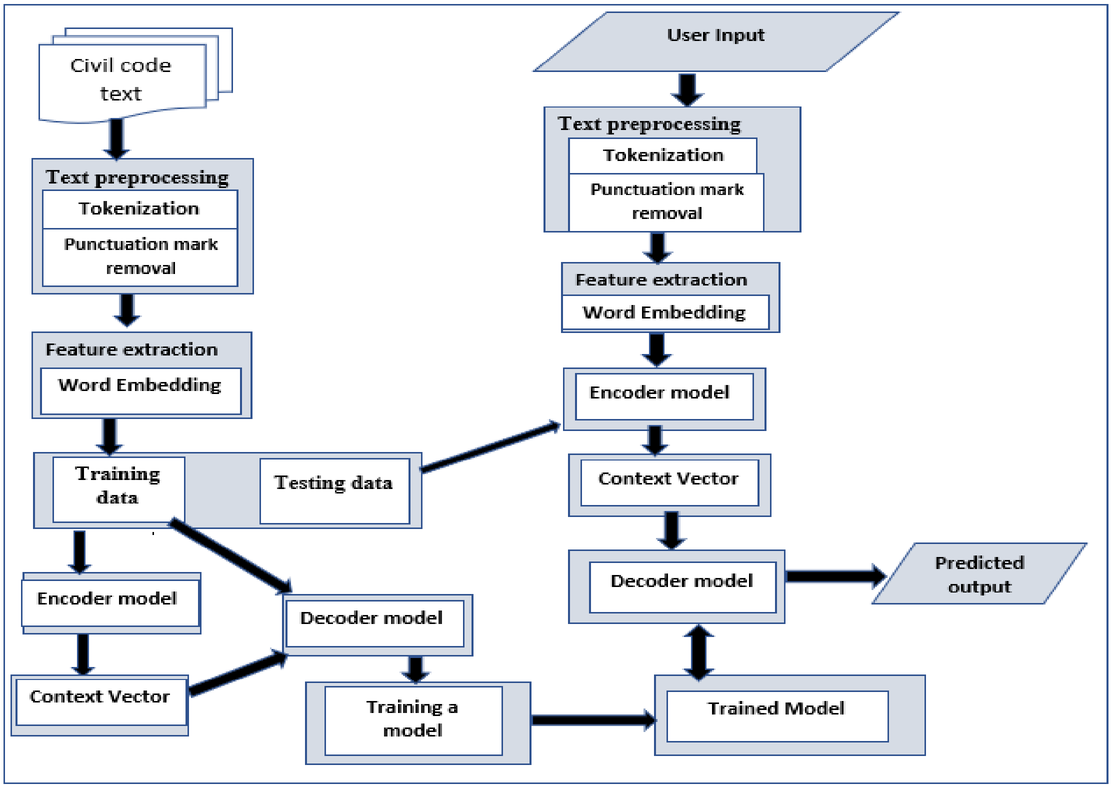

Figure 3. 1:Proposed model architecture

3.2.1. Data pre-processing

Data is a crucial resource used in different aspects, especially for AI-based systems. For example, data plays a significant role in training the models, making a model predict accurately and in reliable for information response in machine learning and deep learning. So, data preparation is the first task for machine learning and deep learning approaches in developing a conversational agent. Those data should be pre-processed and labeled for text generation and other NLP tasks because unlabeled and unprocessed data significantly impact the deep learning approach performance. In addition, the conversational agent requires a large corpus for training and testing to evaluate the system's accuracy.

There is no well-defined dataset for civil codes for the Amharic language. Moreover, it is challenging to conduct a study because the obtained dataset is encoded as an image file, so it is difficult to modify the articles and codes. The image data is converted manually, which is laborious and tedious. After digitizing the data, we pre-processed our dataset. In this study, the following preprocessing steps are carried out.

- **Tokenization**

Tokenization splits raw text or sentences into smaller segments or words called tokens. These tokens support the interpretation of the context or the development of the NLP model. To evaluate the text's meaning, first, the sequence of each word must be split into tokens using a tokenizer. Tokenization is a type of lexical analysis that divides sentences into tokens.

In the tokenization processes, we identified the extent of the vocabulary, the maximum length of sequences, and the representation of words with unique numbers when we tokenized. To represent each word by a unique number, we accessed and visited all of the parallel corpus data. As a result, once a particular word is viewed, assigning a unique number to that word begins, and that number represents the word. The most recently visited word is allocated the final number based on each language's lexicon length. By transforming the original dataset to integer number representation form, these events have made data preparation for training easier. Tokenized data is data that has been translated to an integer format. The row data is tokenized for data pre-processing, and the vocabulary size is set. Each vocabulary is represented by an integer that is proportional to its size. The index of 0 represents the initial vocabulary, and the index of vocabulary size-1 represents the last vocabulary.

We have 16884 vocabulary and 6233 unique vocabularies for the civil code corpus. However, we have used <start> to identify the beginning of a new sentence and <end> to indicate the end of sentences, for example, '<start>', 'አንደኛ', 'መጽሐፍ', 'ስለ', 'ሰዎች', 'አንቀጽ', '፮', 'ስለ', 'ሰዎች','<end>" to determine the boundary of end sentences.

To get this tokenizer, we utilize Kera's tokenizer. This Python programming package can change the tokenizer on an unstructured Amharic document and convert it to a list of sequences or tokens.

Example:

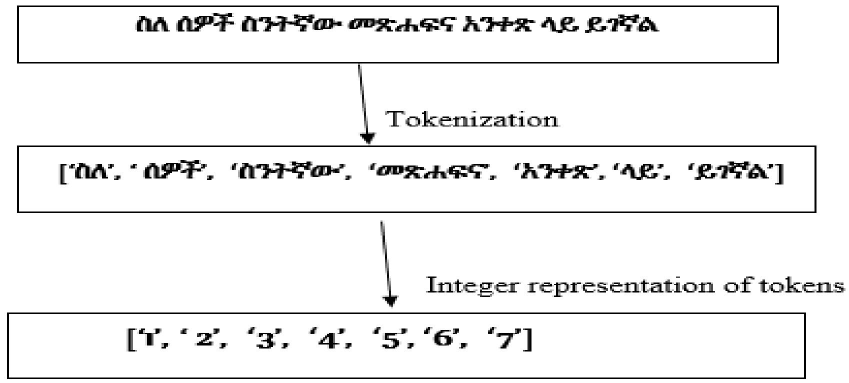

Figure 3. 2: Tokenization and integer representation

The pseudo-code used to tokenize and assign integer values for each token from a given dataset is presented in algorithm 1 below.

```
#Algorithm1: tokenization and integer representation
Input: untokanize dataset
Output: tokens and integer values
Start:
Read dataset
While input parallel sentences end with a new line:
        Split each sentence into two
         if the column is got
                Tokenize both sentences into words and assign an integer value
                Put both tokenized sentences in text_pars by appending
        End if
End loop
Return text_pars
Halt
```

Punctuation mark removal

In preparing a dataset, many punctuations exist, so before feeding to the model, they must be removed because it enhances the training time and storage in computer memory. It also does not have a discriminatory power in sentence representation to replay data from a trained model, so removing those from the dataset is essential for model performance. The punctuation marks used in Amharic are [“።”, “፥”, “፤”, “?”, “!”, “:”, “፡”, “፡-”, “'” “፣”,” ፣”,”;”].

Example:

```
[
 ('ስለ ህገ ውጥ ብርበራና ስለ ሰው ልጆች ነፃነት እንዴት ይሁን ይላል.',
    "ቁጥር፡ ፲፩ ። በነፃነት ፡ ላይ ፡ ስለሚደረግ መሰናክልና፡ ስለ፡ መበርበር ።በሕግ ፡ በተመለከተው ፡
    መሠረት ፡ ካልሆነ፡ በቀር ፡ ማንም፡ሰው፡የግል' ነፃነቱን ፡ ማጣትም ፡ ሆነ ፡ ያላገባብ ፡ መበርበር ፡
                        ሊደርስበት አይገባም ።")]
```

As shown in the above data, there are several punctuation marks to be removed using different techniques, and after cleaning the data being as shown below

```
[('ስለ ህገ ውጥ ብርበራና ስለ ሰው ልጆች ነፃነት እንዴት ይሁን ይላል',
 ቁጥር ፲፩ በነፃነት ላይ ስለሚደረግ መሰናክልና ስለ መበርበር በሕግ በተመለከተው መሠረት ካልሆነ በቀር
            ማንም ሰው የግል ነፃነቱን ማጣትም ሆነ ያላገባብ መበርበር ሊደርስበት አይገባም
")]
```

The pseudo-code used to remove punctuation marks from a given dataset is presented in algorithm 2 below.

```
#Algorithm2: punctuation mark removal
Input: unclean dataset, punctuation mark
Output: cleaned dataset
Start:
Read dataset
Read punctuation
While the dataset is not the end of the file:
        If data contains punctuation:
                Remove punctuation mark
        End if
End loop
Return dataset
Halt
```

- **Stop word removal**

The analysis of word distribution in the corpus is crucial in information retrieval systems. In light of this, the terms' frequency distribution is considered. Each sentence's word choice must be examined. Words or phrases are put together to produce sentences. These terms' frequency distributions in the corpus have varying values. There are high and low-frequency terms, in other words. Regarding information retrieval, content words perform better than function or stop words, which, despite having high frequencies, are less significant. Stop words were thus eliminated before document indexing for retrieval(Asimare, 2020; Kassie, 2009).

Example:

Some of stop words: [(“ነህ” , “ነሽ”, “ነበረ”, “ነበረች”, “ነበሩ”,” ነበር”, “ነች”, “ነን”, “ነኝ”, “ነዋ”, “ነው”, “ነይ”, “ነገር”, “ነገሮች”, “ነገሮችን”, “ናት”, “ናቸው”, “አሁን”, “አለ”, “አላውቅም”, “አልነበረም”)]

Document: [(ቁጥር ፰፻፵ (፪) አፈጻጸም (፩) ከዚህ በላይ ባለው ቍጥር ደንብ ምክንያት በውርስ ለአንድ ወራሽ ከአንድ ውርስ የመጣለትን ድርሻ ሊሰጠው የማይቻል በሆነ ጊዜ በርስትነት ሊሰጠው ከማይቻለሁ የማይንቀሳቀስ ንብረት ላይ የሪም መብት ብቻ ይሰጠዋል (፪) በዚህም የተነሣ ከመብቶቹ ለተቀነሱበት ወራሽ የሚከፈለው ምንም ኪሣራ የለም)]

As shown in the above document, there are several stop words to be removed using different techniques, and after cleaning the data being as shown below

Document: [(ቁጥር ፰፻፵ (፪) አፈጻጸም (፩) ከዚህ በላይ ባለው ቍጥር ደንብ ምክንያት በውርስ ለአንድ ወራሽ ከአንድ ውርስ የመጣለትን ድርሻ ሊሰጠው የማይቻል በሆነ ጊዜ በርስትነት ሊሰጠው ከማይቻለሁ የማይንቀሳቀስ ንብረት የሪም መብት ብቻ ይሰጠዋል (፪) በዚህም የተነሣ ከመብቶቹ ለተቀነሱበት ወራሽ የሚከፈለው ምንም ኪሣራ)]

The pseudo-code used to remove stop words from a given dataset is presented in algorithm 3 below.

```
#Algorithm 3: Stop word removal
Input: list of words, stop word list
Output: list of non-stop words
Start:
Read a list of words
Read stop word list
While the list of words is not ended:
    If list of words contains stop words
        Remove stop word
    End if
End loop
Return non-stop words
Halt
```

3.2.2. Feature extraction

Machine learning and deep learning algorithm cannot understand textual data unless converted into numerical values. Feature extraction is the process used to convert text data into machine-understandable formats. Different techniques used to convert text data to numerical values, such as Bag of word, TF IDF, word2Vec, one-hot encoding, and word embedding, are used in feature extraction, each of which has a limitation.

- **One hot vector representation**

After converting the dataset to integer/vector format, it must be converted to a two-dimensional vector, also known as a one-hot vector, to develop a conversational chatbot. So, this representation employs a unique vector representation for each word in sentences.

The neural network does not directly operate on the vocabulary represented by the integer. Using the neural network to handle the vocabulary must switch to a vector representation, known as one-hot vector representation. We assigned a value of 1 to the vocabulary that corresponded to its place and a value of 0 to the vocabulary that did not.

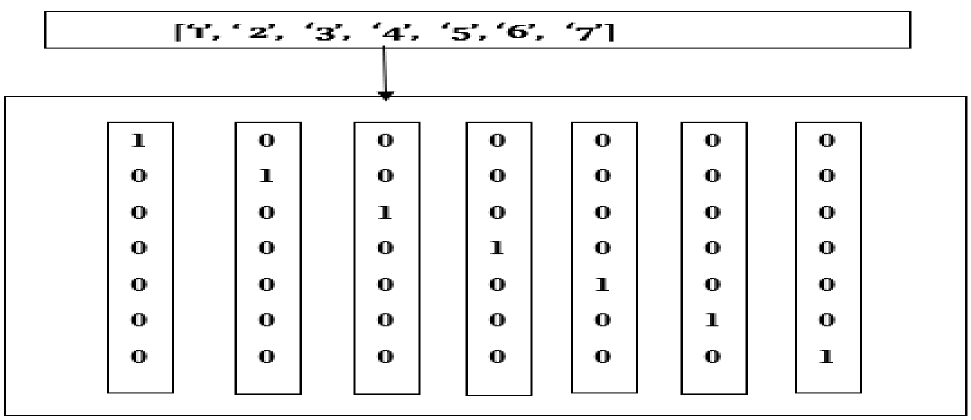

Figure 3. 3: one-hot vector representation

We applied zero to the right to set the length of the sentences fixed during one-hot vector representation, known as zero padding. The maximum size of the sequence in our situation is 200. As a result, we've utilized zero to the right for the remaining index, as shown in Figure 3.4. We have given such vectors to the neural network by embedding once the padding and standardized vector size has been completed.

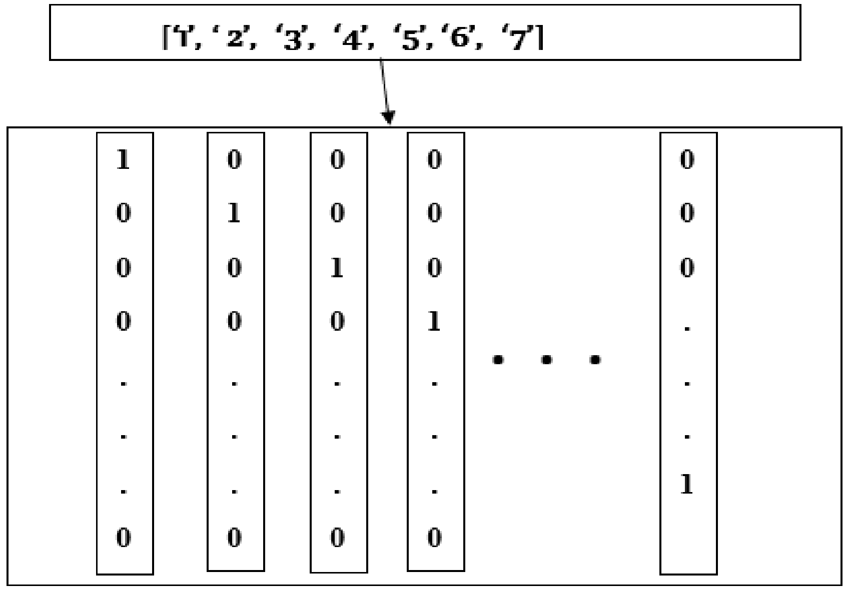

Figure 3. 4:One hot vector padding

- **Word embedding**

When one-hot vector data is supplied directly to the neural network, there may be a problem with the neural network training waste rather than learning when zero values are detected. So, we employed word embedding to tackle this problem, which converts a higher-dimensional one-hot vector to a lower-dimensional one-hot vector. Word embedding translates words into numeric representations ranging from -1 to 1. To represent the embedding vector space, we chose 128 embedding dimensions.

To construct word embedding, we used the Kera's embedding layer. The Embedding layer is supplied with random weights ranging from -1 to 1 and is tasked with learning an embedding for each word in the training dataset. And for each word, get the embedding vectors. The vectors are then utilized to represent the words in a phrase. It defines the input dimension (the size of the vocabulary in the dataset), the output dimension (the size of the vector space in which the words are embedded), and the input length (the number of words in the vector space).

As a result, our neural network used to embed to read one hot vector from our input. Two embedding vectors have been constructed. The first embedding vector is source embedding, which the encoder feeds, and the second embedding vector is target embedding, which the decoder accepts during training. However, this second-word embedding is not required during the interpretation phase because the target embedding vector is only fed to the decoder during the learning phase. We give the prior predicted output to the decoder model at predictable times. There are two forms of embedding for both the source and the target: token embedding and position embedding.

We used various models to conduct our experiments, including RNN (LSTM and Bi-LSTM) and transformer models. In contrast to RNN models, the Transformers model simultaneously processed all sequences. The model does not know the order of the tokens at this moment. As a result, we require an additional encoding vector known as the positional encoding vector. As a result, we only employed token embedding for the RNN models. However, we used both a token and a position embedding vector in the Transformer model. The encoder-decoder model is the next stage after embedding.

3.2.3. Encoder decoder model

We employed the training and testing encoder-decoder models individually, as shown the proposed architecture in Figure 3.1. The critical distinction is that while training an encoder-decoder model, both the encoder and decoder models must be compiled and executed. We stored the constructed encoder-decoder model after constructing and running it. However, we do not need to build and run the constructed model when testing. We just used the saved training weight to define the model. The other major distinction is that we sent the target embedding to the decoder during the training model, known as the teacher learning force. However, we did not feed the target embedding into the testing model. Because only the previous anticipated output is used in the testing model to predict the sequence, the prior expected output could be all previous or current predicted, depending on the models. For example, in RNN, only the current anticipated output is used. However, in the transformer, all prior predicted output is used. The encoder model was used to take the source embedding input, and the final context vector was then supplied directly to the decoder. The decoder was then used to predict the target output. Let's look at each model separately.

LSTM encoder-decoder model

Each internal state of the encoder LSTM model sequentially accepts the query embedding before, and we only use the most current internal state as a context vector at the end. We did not use a single hidden state for our prediction during the LSTM. The decoder LSTM model accepts the context vector and the current predicted output to predict the following output. For the first phase, we currently have no expected output. Consequently, we have used the beginning of the sentence in our situation, which is <sos>. From the beginning of the sequence until the end, the decoder model predicts the following output. In our case, we added a <eos> to the end of each sentence. The following pseudo-code demonstrates how the LSTM model's training and testing are carried out(Kinfe, 2022; H. Wang et al., 2022).

```
Algorithm 4: pseudo-code for training and testing of LSTM model
Xi=Input embedding sequence
SL=Sequence Length
// HS is Hidden State
//CS is Cell State
XXi=Encoder Output
While(i<SL)
        Encoder (HS_i+1, CS_i+1, XX_i+1)=X_i+[HS_i,  CS_i]
        i=i+1
end of while
CV=Encoder (HS_i, CS_i)
PO_j=Predicted Output
TES_j=Target Embedding Sequence
To=<sos>
Decoder (HS_j, CS_j) =To+CV
While (PO_j!=<eos>)
        Decoder (HS_j, CS_j, PO_j+1)=PO_j+Decoder(HS_j, CS_j)  //Only during testing
        Decoder(HS_j+1,CS_j+1, PO_j+1)=T_j+Decoder(HS_j, CS_j)  //Only during training
        Set PO_j //Current predicted output
        j=j+1
End of while
```

BiLSTM encoder-decoder model

The Bi-LSTM model follows the same workflow as the LSTM model. When the input sequence length is long, we simply add a backward internal state (backward hidden state and backward cell state) to check for and emphasize missing information. The following pseudo-code demonstrates how the Bi-LSTM model's training and testing are carried out(Kinfe, 2022; Shahin & Almotairi, 2021).

```
Algorithm 5: Pseudo-code for training and testing of Bi-LSTM model
Xi = Input embedding sequences
L = Sequence length
//FHS forward hidden state; BHS backward hidden state;
 //FCS forward cell state; BCS backward cell state
 FHS0, FCS0, BHS0, BCS0= 0
XXI = Encoder output
 While(i<L)
        Encoder (FHSi+1, FCSi+1, XXi+1) =Xi+[FHSi+FCSi]
        i=i+1
End of while
CVF=Encoder (FHSi,FCSi)
While(k<L)
        Encoder (BHSk+1, BCSk+1, XXk+1) =Xk+[BHSi+BCSi]
        k=k+1
End of while
CVb=Encoder (BHSi,BCSi)
H=Concat (FHSi,BHSi)
C=Concat (FCSi, BCSi)
CV=Encoder(H,C)
Po=Predicted Output
TES=Target Embedding Sequence
ST=<sos>
Decoder (Ho Co Po)=ST+CV
While (Po!=<eos>)
        Decoder (Ho+1,Co+1,Po+1)=Po+Decoder(Ho,Co) //only during testing
        Decoder (Ho+1,Co+1,Po+1)=ST+Decoder(Ho,Co) //only during Traning
        Set Po //current predicted output
        o=o+1
End of while
```

Transformer encoder-decoder model

There is no additional attention mechanism in our transformers model. We have given our transformer model both a token embedding and a position embedding vector. Because the transformer model accepts all inputs simultaneously, we used a single number of layers in our transformer model. Like the Bi-LSTM model, LSTM and transformer model used only the last encoder context vector to pass directly to the decoder without additional attention mechanisms since the transformer model has self-owned attention. Self-multi head attention and a feed-forward neural network are features of the encoder transformer model.

Multi-head attention and a feed-forward neural network are all masked in the decoder model. In both the encoder and decoder models, we used the RLU feed-forward network(Gad et al., 2022). The pseudo-code below illustrates how the transformer model's training and testing are carried out(Gad et al., 2022; Kinfe, 2022).

```
SL= sequence length
IES= input embedding sequence
PE= position embedding
i=0
While(i<SL)
      Encoder MA= [PE_i, IES]
      Encoder FFN=Encoder MA (PE_i, IES_i)
      i=i+1
End of while
CV=Encoder FFN (PE_i, IES_i)
TES=Target Embedding Sequence
TPE=Target Positional Embedding
st=<start>
TPE=0
Decoder MA= [st, TPE]
Decoder MA=decoder MA (st, TPE) +CV
Decoder FFN=decoder MA ([st, TPE] +CV)
Dense layer=decoder FFN ([st, TPE] +CV)
Pi, i=1… =predicted
Po=dense ([st, TPE] +CV)
While (Pi! =<END>)
      Set X= [TPE………i] +CV //only during testing
      Set X= [st………i, TPE…. st] +CV // only during train-
      ing
      Decoder MMA =X
      Decoder MA=Decoder MMA(X)
      Decoder FFN=Decoder MA(X)
      Dense layer = Decoder FFN(X)
      Pi+1=dense(X)
      Set Pi+1 //current predicted output
      i= i+1

End of while
```

3.3. Chapter Summary

The system architecture shows that an encoder and decoder model is used to build a model based on user queries and civil code datasets. Our chatbot operates in two stages. These are the training and testing steps (inference). Before starting the prediction, the machine can learn the structure of the dataset pairs repeatedly from the training dataset. Finally, the training model has been stored. We later used the saved model for inference to predict the target response. We began testing the algorithm on our test dataset after training. The testing phase is when the system starts predicting an unknown user inquiry based on the knowledge obtained during the training phase. Finally, we used the BLEU score to assess the system's quality, which compares the predicted answer to data from the civil code.

CHAPTER FOUR: EXPERIMENTAL RESULT AND DISCUSSION

4.1. Overview

This section discusses the experimental findings by demonstrating experimental setups and system testing outcomes using BLEU score metrics. In addition, we compared the results of the experimental systems in user queries on actual civil code and the predicted civil code by the developed model. The step is presented in the subsequent subsection.

4.2. Dataset

To carry out the experiment, we collected data from the Ethiopian civil code document. In this study, we have prepared a dataset that contains queries and answers for each civil code article. The dataset is composed of 3215 parallel sentences used as queries and answers. After data shuffling, we divided the dataset into training and testing. Of the dataset, 80 percent of the dataset was used for training, while 20 percent was used for testing. Because most related works classified their data using the Pareto Principle (80/20) techniques, which specifies that 80% of the total dataset is used for training and 20% is left over for system testing(Kinfe, 2022; Wogaso, 2020). As a result, the test dataset contains 643 parallel sentences, while the training dataset contains 2572 sentences. The dataset used to train a model has a look, as shown in Appendix I. The model performance was evaluated using both the training and testing queries.

Sample training dataset:

['አባትና እናት ስለማስተዋወቅ ክፍል ስንት ላይ ይገኛል.', '[start] ክፍል ፩ አባትንና እናትን ስለ ማወቅ [end]'], ['አንድ የማይንቀሳቀስ ንብረትን በከፊል እንዲለቀቅ ስለ ማድረግ.', '[start] ቁጥር ሺ፬፻፷፱ አንድ የማይንቀሳቀስ ንብረትን በከፊል እንዲለቀቅ ስለ ማድረግ (፩) አንድ ሕንጻ በከፊል እንዲለቀቅ የተደረገ እንደ ሆነ ባለሀብቱ ይህን ሕንጻ በሙሉ ኣድሚኒስትራሲዮኑ እንዲወስደው ለማስገደድ ይችላል (፪) እንዲሁም አንድ መሬት በከፊል እንዲለቀቅ ሲደረግ ቀሪው የማይጠቅም የሆነ እንደ ሆነ የመሬቱ ባለሀብት አድሚኒስትራሲዮኑ በሙሉ እንዲወስድ ለማስገደድ ይችላል (፫) ከዚህ በላይ የተባሉት መብቶች እንዲለቀቅ በተደረገው የማይንቀሳቀስ ንብረት ላይ ካላቸ መብቶች ሌላ ባለርስቱና የአላባ ጥቅም ተቀባዩ ከዚህ በላይ ባሉት ኀይለ ቃሎች የተጻፉት መብቶች ይኖራቸዋል [end]'], ['በሕግ የሰውነት መብት የተሰጠው ድርጅት አመዘጋገብ ሁኔታ.', '[start] ቁጥር ሺ፮፻፷ በሕግ የሰውነት መብት የተሰጠው

ድርጅት አመዘጋገብ (፩) የማይንቀሳቀሰው ንብረት ያንድ የንግድ ማኅበር ወይም ያንድ ማኅበር የሆነ እንደ ሆነ በቤተ ዘመድ ስም ተጠያቂው ስምና በአባቱስም ፈንታ የሰው መብት የተሰጠውን ድርጅት ስም ወይም የድርጅቱን የመጠሪያ ስም የሚገልጽ ይሆናል (፪) እንዲሁም በተመዘገበው ጽሑፍ የሚጠቀመው አንድ የንግድ ማኅበር አንድ ማኅበር ወይም ማናቸውም የሰውነት መብት የተሰጠው ድርጅት በሆነ ጊዜ ከዚህ በላይ ባለው ኀይለ ቃል የተጻፈው ቃል ይፈጸማል [end]'], ['የባልና ሚስት መሪ የሆኑ ሐሳቦች ምን ሊሆኑ ይችላል.', '[start] ቁጥር ፮፻፲፫ መሪ የሆኑ ሐሳቦች (፩) በመሠረቱ የዘመድ ዳኞች የሚወስኗቸወ ቅጣቶች ከዚህ በላይ በተጻፉት ቍጥሮች ውስጥ የተመለከቱትን ነው (፪) ስለ ሆነም በማናቸውም ጊዜ እነዚሁኑ ለመወሰን ወይም ላለመወሰን እንዲሁም ተፈጻሚ የሚሆኑበትን መጠን ለመወሰን ሙሉ ሥልጣን አላቸው (፫) ለጉዳዩ ውሳኔ በሚሰጡበዙ ጊዜ ለጉዳዩ ምክንያት የሆነውን ነገርበተለይም መፋታትን ያስከተለውን የተጋቢዎቹን ጥፋት ከባድነትና በሕሊናም በኩል (በሞራ) የፍቺው ጥያቄ የሚያስወቅስበትን መጠን መመርመር አለባቸው [end]'], ['የጋራ የሆነ የማይንቀሳቀስ ንብረትና የማይንቀሳቀሰው ንብረት የጋራ ባላህብቶች የኑ የብዙ ሰዎች ሆኖ ባለጉዳና አንዱ ባለሀብት ብቻ የሆነ እንደ ሆነ ምን መደረግ አለበት.', '[start] ቁጥር ሺ፮፻፲ የጋራ የሆነ የማይንቀሳቀስ ንብረት (፩) የማይንቀሳቀሰው ንብረት የጋራ ባላህብቶች የኑ የብዙ ሰዎች ሆኖ ባለጉዳና አንዱ ባለሀብት ብቻ የሆነ እንደ ሆነ የዚህኑ ባዕዳይ ድርሻዕብቻ በመጥቀስ በታተመው ጽሑፍ የ የጋራ ባለሀብት ስም ብቻ ይጠቀሳል (፪) ጽሑፉ የሚመለከተው የጋራ ባለሀብቶችን ሁሉ ወይም ከነሱ አብዛኛዎቹን የሆነ እንደ ሆነ በታተመው ጽሑፍ ውስጥ የጋራ ሀብት የሆነ የማይንቀሳቀስ ንብረት ተብሎ ተጨማሪ አስረጂ በሆኑት ጹሑፎች ውስጥ የማይንቀሳቀሰው ንብረት ሀብትነት ወደሚመለከተው ሌላ ሰነድ መምሪያ ይደረግባቸዋል (፫) በጽሑፉ ወስጥ ለተመለከቱትለያንዳንዱ የጋራ ባለሀብቶች በተመደበው ገጽ ላይ በባለሀብቶች መዝገብ ጽሑፉ የሚጠቀስ ይሆናል [end]'].

Sample test dataset:

['የሞግዚትነት ሥራ ሒሳቦች ንዑስ ክፍል ስንት ላይ ይገኛል.', '[start] ንኡስ ክፍል ፪ የሞግዚትነት ሥራ ሒሳቦች [end]'], ['ስለ ማቃናት ሥራ አደራረግና የቃሉ ማቃናት የሚፈጸመው በታተመው ጽሑፍ ወስጥ የሚያቃኑትን ቃሎች እንዴት ነው.', '[start] ቁጥር ሺ፮፻፸፬ (፬) ስለ ማቃናት ሥራ አደራረግ (፩) የቃሉ ማቃናት የሚፈጸመው በታተመው ጽሑፍ ወስጥ የሚያቃኑትን ቃሎች ልዩ ቀለም በመሰረዝ ነው (፪)ዐቃቤ መዝገቡ በታተመው ጽሑፍ ፊት ላይ የተቃናበትኝ ቀንና እንዲቃናም ዳኞች የሰጡትን ፍርድ ከሚጠቅስ ቃል ጋራ ተቃንቷል የሚል ቃል ይጽፍበታል (፫) ስለዚህ ጉዳይ የተሰጠውም ፍርድ ከአስረጂ ሰነዶች ጋራ በቤተ መዛግብት ይቀመጣል [end]'], ['የሕዝብ የክብር መዝገብ ጽሑፎች የጊዜ ዉሳኔ የተወሰነ ከአለፈ በኃላ የሚያትት ቁጥር ስንት ላይ ይገኛል.', '[start] *ሀ*) የከተማ ቀበሉ ቁጥር ፳፫ የጊዜው ውሳኔ አለመፈጸም (፩) የተወሰነው ሕጋዊ ጊዜ ካለፈ በኋላ የሚጻፉ የሕዝብ የክብር መዝገብ ጽሑፎች ዋጋቸው እንደ ተራ ማስታወቂያዎች አስረጂ ብቻ የሚቈጠሩ ናቸው (፪) ስለ ሆነም በመዝገቡ ላይ የተመዘገቡት በአንድ ፍርድ መሠረት እንደ ሆነ ከዚህ በላይ የተጻፈው አይጸናም (፫) ይህ ሲሆን ጽሑፉ ፍርዱን የሚጠቅስ ቃል በላዩ ላይ ይኖርበታል [end]'], ['የኅብረት ስለ ሆኑ የእርሻ መሬቶች ምዕራፍ ስንት ይናገራል.', '[start] ምዕራፍ ፪ የኅብረት ስለ ሆኑ የእርሻ መሬቶች [end]'], ['የቀዳሚነት መብት እንዲሁም ንብረቱን የአስተዳደሩ

ክፍል መልሶ ከሸጠበት ወይም ከሥራው አፈላእም ጋራ የማይስማማ በሆነ ግዜ.', '[start] ቁጥር ሺ፬፻፹፬ የቀዳሚነት መብት (፩) ከዚህ በላይ በተጻፈው ቊጥር ውስጥ የተነገረው የቀዳሚነት መብት በመዝገብ ተራ ቊጥር የያዘ ወይም የተመዘገበ ባይሆንም እንኳ ንብረቱን የአስተዳደሩ ክፍል መልሶ ከሸጠበት ወይም ከሥራው አፈላእም ጋራ የማይስማማ መብት በዚህ በማይንቀሳቀስ ንብረት ላይ ለሌላ ሦስተኛ ወገን ከሰጠበት ቀን አንሥቶ እስከ አንድዓመት ድረስ በሦስተኛ ሰዎች ላይ መቃወሚያ ሊሆን ይችላል (፪) ከዚህ በቀር የዚህ የተባለው መብት የሚሠራበት ሁኔታና የሚያስከትለው ወጤት በዚህ ሕግ በአንቀጽ ፳ ስለ ጋራ ሀብት ስለ ስለ ኣላባ ጥቅምና ስለ ሌሎች ግዙፍ መብቶች በተጻፈው ሕግ ውስጥ በተነገረው መሠረት ተወስኗል (፫) የቀዳሚነት መብት ያለው ሰው በዚሁ በመብቱ መሠረት የቀድሞ ንብረቱን የሚገዛው ቀድሞ ንብረቱን በማስለቀቅ በተወሰደበት ጊዜ ከአስተዳደር ክፍል በተቀበለው ዋጋ ልክ ነው [end]'],

4.3. Development environment

In this work, we used Python Programming Language as the primary programming development tool, from preprocessing to model building and evaluation. Python is useful in text processing and has many open-source NLP packages. Include TensorFlow with the deep learning library, Kera's, NumPy, and other necessary dependencies. In addition, we use 12 GB of RAM and a GPU for our research work from google Collaboratory, which is processed with a fast-training time. Experiments on the LSTM, BiLSTM and transformer models have been conducted to train and test each model's performance. Finally, we shall assess each other's effectiveness (time, memory usage, accuracy): then, we select the best model with the best prediction result, and for the selected model, a web-based and mobile-based application programming interface (API) to access the saved model. The web-based API flask framework is used, which is suitable for developing a web application written in python. In addition, the Django REST framework is also used to develop a mobile API for users, which is essential to access a model everywhere through their mobile phones.

TensorFlow: is an open-source library developed primarily for deep learning and machine learning applications. It is used for large numerical computations.

Kera's: is a well-known and straightforward deep learning model for defining, building, and evaluating model performance.

4.4. Hyperparameter selection

We carried out different experiments on various parameters to attain the desired result. We employed embedding and latent dimensions to choose the best optimum dimension based on our dataset. Obviously, choosing appropriate embedding dimensions are challenging but essential to create embeddings for accomplishing our tasks. First, we selected the embedding dimension and chose the embedding dimensions 128 and 256(Kinfe, 2022; Wogaso, 2020). Then, based on related works, we chose latent dimensions of 128, 256, 512, and 1024(Kinfe, 2022; Wogaso, 2020).

Here to choose the best, we have used a 64-batch size, a 0.001 learning rate, Adam optimizers, and a 0.2 dropout rate. The experiments were done using 50 epochs. Finally, as shown in Table 4.1, we have the following results. These outcomes are nearly identical, with minor differences. In each experiment below, the loss level decreases as it progresses from higher to lower.

Table 4. 1: Training time, training accuracy, and training loss for selecting embedding and latent dimension

Embedding dimension	Latent dimension	Training accuracy	Training loss	Time taken(second)
128	128	0.9601	0.0424	236
128	256	0.9613	0.0333	237
128	512	0.9623	0.0316	339
128	1024	0.9626	0.0311	241
256	256	0.9549	0.0442	440
256	512	0.9465	0.0736	445
256	1024	0.9512	0.0488	476

The best results from the above experiments are 128 embedding dimensions and 1024 latent dimensions. As a result, we've chosen 128 as the embedding dimension and 1024 as the latent dimension. Dimension 128 embedding takes the same time; the only difference is a small fraction of a second. Then we conducted experiments using the dropout rate of 0.2, 0.5, and 0.7, a learning rate of 0.001 and a batch size of 64 using 50 epochs based on the previous work (Kinfe, 2022; Tan et al., 2021; Wogaso, 2020). Thus, we achieved a minimum loss with a dropout rate of 0.2, as shown in Table 4.2.

Table 4. 2: Training and accuracy Loss for selecting Dropout rate

Embedding dimension	Latent dimension	Dropout rate	Training loss	Training accuracy
128	1024	0.5	0.0430	0.9648
128	1024	0.2	0.0312	0.9718
128	1024	0.7	0.1066	0.9279

The learning rate becomes a problem as the underlying system cannot make useful generalizations when very large training samples are provided. This can also happen if the network has too many neurons and the computation volume exceeds the dimensionality of input in the vector space. So, selecting the value of the learning rate is important to develop a good model. The range used for the learning rate in deep learning is between a small value of 0.0 to a large value of 1.0(JACOB WILSON, 2022; Peace et al., 2015).

So, we chose the rate of learning that begins with a huge value, such as 0.1, and then attempted exponentially lower weights, such as 0.01, 0.001, and 0.0001, with losses of training 2.3027, 2.2839, 0.0312 and 1.1259, respectively(Pavel Surmenok, 2017). Finally, we achieved the best outcome with a learning rate of 0.001. In Figure 4.1 (a-d) below, we also illustrated the loss level in each learning rate with embedding dimension 128, latent dimension 1024, batch size 64, dropout 0.2 and epoch 50.

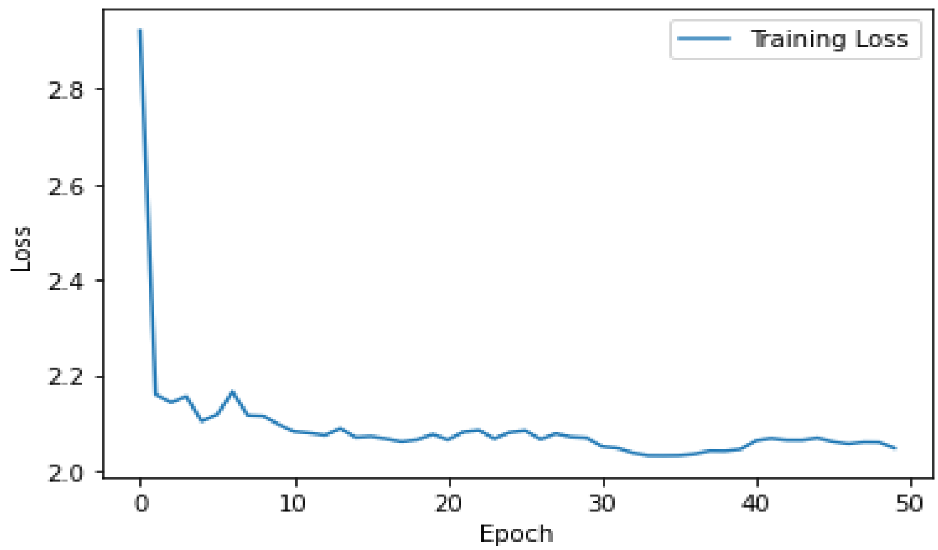

Figure 4. 1: Loss level with learning rate 0.1

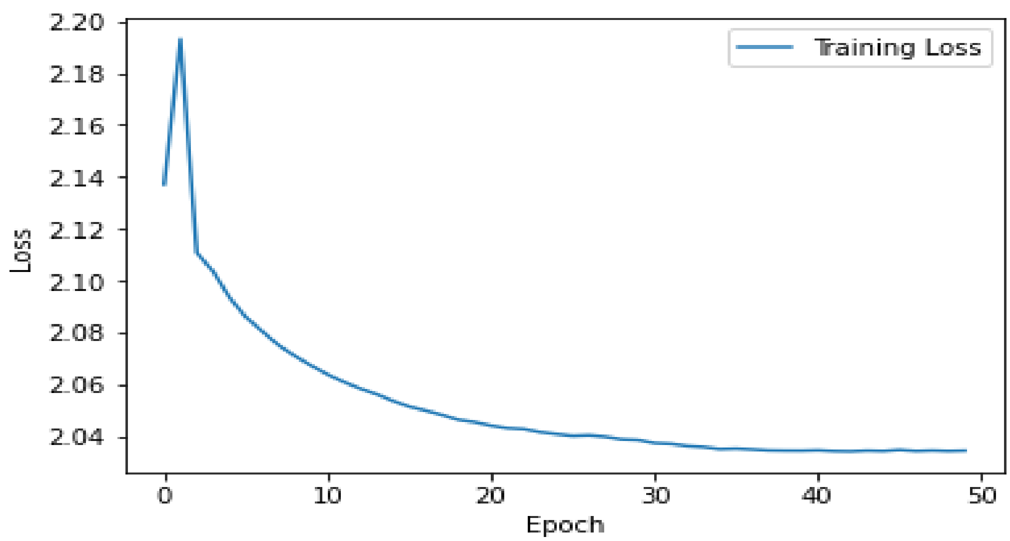

Figure 4. 2: Loss level with learning rate 0.01

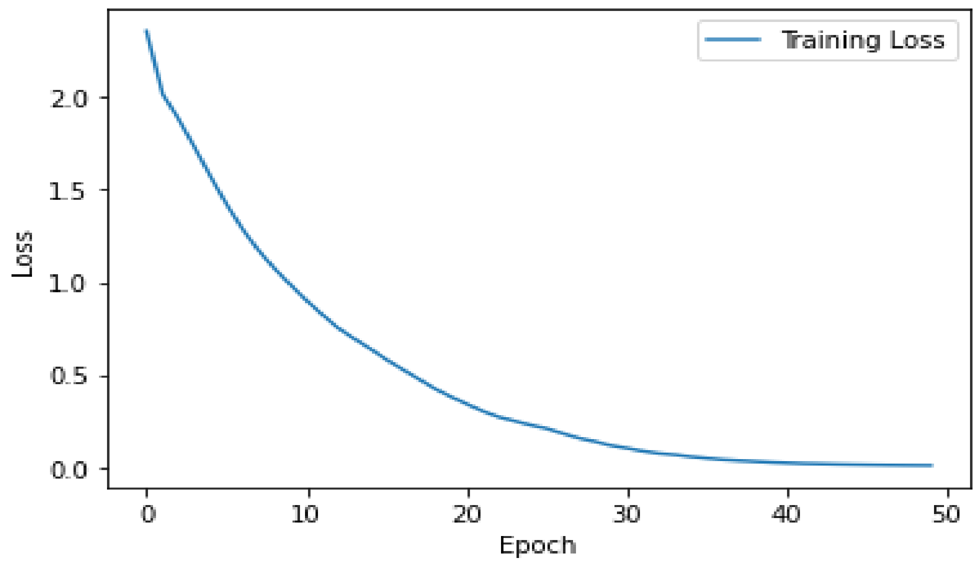

Figure 4. 3: loss level with learning rate 0.001

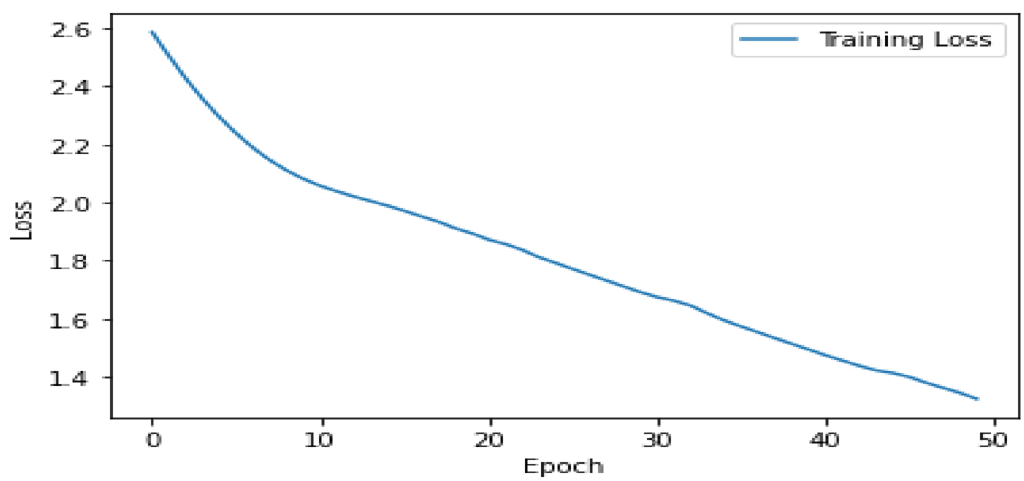

Figure 4. 4:loss level with learning rate 0.0001

As shown in Figure 4.1, with a learning rate of 0.001, as the number of epochs increases, the training loss decreases and obtains a minimum training loss compared to other learning rates. Because as the learning rate increases, the model may reduce training time and reach an optimal level. But on the optimum learning rate, the model is trained reliably even if a model takes a longer time. There is no well-defined used learning rate in deep learning, so in this thesis, we use a learning rate of 0.001 throughout our experiment.

All the parameters mentioned above were selected tests to utilize a civil code chatbot with the Transformers model. Based on the above best parameters of the transformer model, we conduct our experiment for LSTM and BiLSTM models to identify the best model for predicting the user's query response, the time taken to train, and the converging level of the models. In general, the hyperparameters used in this study are shown in Table 4.3.

Table 4. 3:Hyperparameter settings

Parameter	Value
Embedding dimension	128
Latent dimension	1024
Learning rate	0.001
Epoch	50
Dropout	0.2
Bach size	64
Activation	RLU
Optimizer	Adam

4.5. Experimental results

This study used generative approaches such as LSTM, BiLSTM, and transformer models. We similarly train this system to compare their performance based on our findings. We build the system in each model using the prepared civil code user inquiry and the articles' responses. Finally, we evaluated the model's performance based on training time memory usage and accuracy. And we chose and recommended the best model with the best response unit. The civil code dataset experiments employed a sample parallel corpus detailed in Appendix I. Lastly, we have shown the results of those models in Table 4.4 below.

Table 4. 4:Experimental results of a chatbot models

Experiments	Models	BLEU score in %	Time taken in seconds
Experiment 1	LSTM	4.78	259
Experiment 2	BiLSTM	6.82	345
Experiment 3	Transformer	9.88	322

We ran our first, second and third experiments using the prepared civil code dataset 3215 using the LSTM, BiLSTM and transformer model. We have observed parameters for training time, BLEU scores to assess performance over 50 epochs and a 0.0001 learning rate value. Training the LSTM model using the provided dataset took 259 seconds, while BiLSTM TOOK 345 seconds. The BiLSTM training time is higher than LSTM because BiLSTM learns the data in a forward and backward direction, and this learning technique improves the model performance for long-ranged sentences in our dataset. On the other hand, the transformer model took 322 seconds to train our model using the Transformers model using 50 epochs.

We used the BLEU score to compare the testing outcomes to the above models. LSTM obtained a BLEU score of 4.78. Compared to BiLSTM and transformer, it showed minimum value because the LSTM model learns the data forward only. So, as the sentence length in a dataset increases, the model loses terms at the beginning of a sentence. In addition, the model used too long to train because it encodes each term in a timestamp manner in a sentence. The learning of the BiLSTM model is used to hold the context of a word in the sentence. The experimental findings showed a BLEU score of 6.82 in anticipating a user query response. The transformer model achieved a BLEU score of 9.88 in predicting user inquiries' responses.

The Transformers model showed the best training time, memory utilization, and accuracy results. Our experiments' findings indicate that the civil code chatbot responds better using the Transformers model. Therefore, the transformer model is what we recommend using for the user's case. The sample of answers from our suggested deep-learning model for the given query is displayed below in Table 4.5.

Table 4. 5: Query response using transformer model

Queries	Model response
ተመላሽ የሆኑ ንብረቶች በተመለከተ ምን መሆን አለባቸዉ	ቁጥር ፫ሺ፱፻፴፩ (፫) ተመላሽ የሆኑ ንብረቶች (፩) ኮንሴሲዮን ሰጭ ኪሣራ ሳይከፍልባቸው የኮንሴሲዮኑን የማይንቀሳቀሱ ንብረቶች መልሶ ይረከባል (፪) እንደዚሁም ኮንሴሲዮን ሰጭ በግዴታዎች ደብተር ውስጥ ኪሣራ ሳይከፈልባቸው ተመላሽ ይሆናሉ ተብሎ የተመለከቱትንም ተንቀሳቃሽ ንብረቶች ኪሣራ ሳይከፈልባቸው ይረከባል
ወድቀው ስለሚገኙ ልጆች የጽሑፍ መደራጀት እንዴትና ምን ምን መረጃዎች ይደራጃሉ ብሎ ያሰቀምጣል	ቁጥር ፻፫ ወድቀው ስለሚገኙ ልጆች (፩) ወድቆ ለሚገኝ ማን መሆኑ ላልታወቀ ለአራስ ልጅ ሁሉ የመወለዱ ጽሑፍ መደራጀት አለበት (፪) ልጁ የተገኘበትን ቀንና ስፍራ ይኖረዋል የሚባለውን ዕድሜ ፆታውን የተሰጡትን የቤተ ዘመድ ስምና የግል ስሞች በዝርዝር የሚያሳይ ፕሮሴቬርባል ይደራጃል (፫) ስለ ልጁ መወለድ በተጻፈው ጽሑፍ ላይ ወደ ፕሮቴቪርባሉ የሚመራ ምልክት ይጠቀሳል
በሥራ አገልግሎት ላይ ያሉ ወታደሮች በሞቱ ግዜ መሞቱን ማሳወቅ ያለበት ማን ይሁን ይላል	ቁጥር ፻፱ በሥራ አገልግሎት ላይ ያሉ ወታደሮች ወታደሩ ከቤተ ሰቡ ጋራ የሚኖር ሆኖ ወይም ለዕረፍት በተፈቀደለት የስንብት ጊዜ ወይም ክፍለ ጦሩ ከሰፈረበት ውጭ የሞተ ካልሆነ በቀር በሥራ አገልግሎት ላይ ያሉ ወታደሮችን መሞት መግለጽ ያለበት ክፍሉን የሚያዘው ሹም ነው
የልጆቹ አባትና እናት በተፋቱ ግዜ ሞግዜት የመምረጥ መብት የማን ነው	ቁጥር ፪፻፺ የልጆቹ አባትና እናት መፋታት (፩) አባትና እናት በተፋቱ ጊዜ ለልጁ ሞግዚትና አሳዳሪ የሚመርጡ የቤተ ዘመድ የሽምግልና ዳኞች ናቸው (፪) ከተፋቱትም

	ባልና ሚስት አንደኛው የሞተ እንደ ሆነ በዚሁ ምክንያት ብቻ በሕይወቱ ያለው አባት ወይም እናት ይህን ሥልጣን ለመያዝ መብት አይኖረውም
ልደት ሞትና ጋብቻ ያጠራጠሩ እንደ ሆነ ፍትሐብሔር ህጉ በማን አስረጂነት ይጣራ ይላል	ቁጥር ፺፯ የአስረጅነቱ ዐይነት (፩) ልደት ሞትና ጋብቻ ያጠራጠሩ እንደ ሆነ ወይም ክርክር የተነሣባቸው እንደ ሆነ የሚረጋገጡት በክብር መዝገብ ጽሑፍ አስረጅነት ነው (፪) እንዲሁም ሕግ በሚፈቅድበት ጊዜ በታወቀ ሰነድ አስረጅ አማካይነት ወይም በሁኔታ መኖር ሊረጋገጡ ይችላሉ

4.6. User acceptance testing

The proposed transformer model is also reviewed by users, as described in section 4.5 when discussing the evaluation method. To do this, we choose novice, knowledgeable, and expert users to use the suggested transformer model. Then we test the model response based on the users' input on the developed prototype, as shown in Figures 4.5 and 4.6. And when it comes to conversational agents, model performance measures do not provide complete model accuracy for users. Thus, a manual testing method ensures that the conversation chatbot displays the proper model usage from different perspectives. The goal is to enable direct keyboard entry of messages into the interface and examine the model responses based on the queries.

We integrate the model with the web and mobile bots to evaluate the model using human evaluation. The users conducted text-to-text conversations utilizing this system. After using the model with Flask for the web bot, the text-to-text communication is shown in Figure 4.5. In addition, we used the model with Django for mobile text-to-text communication, as shown in Figure 4.6.

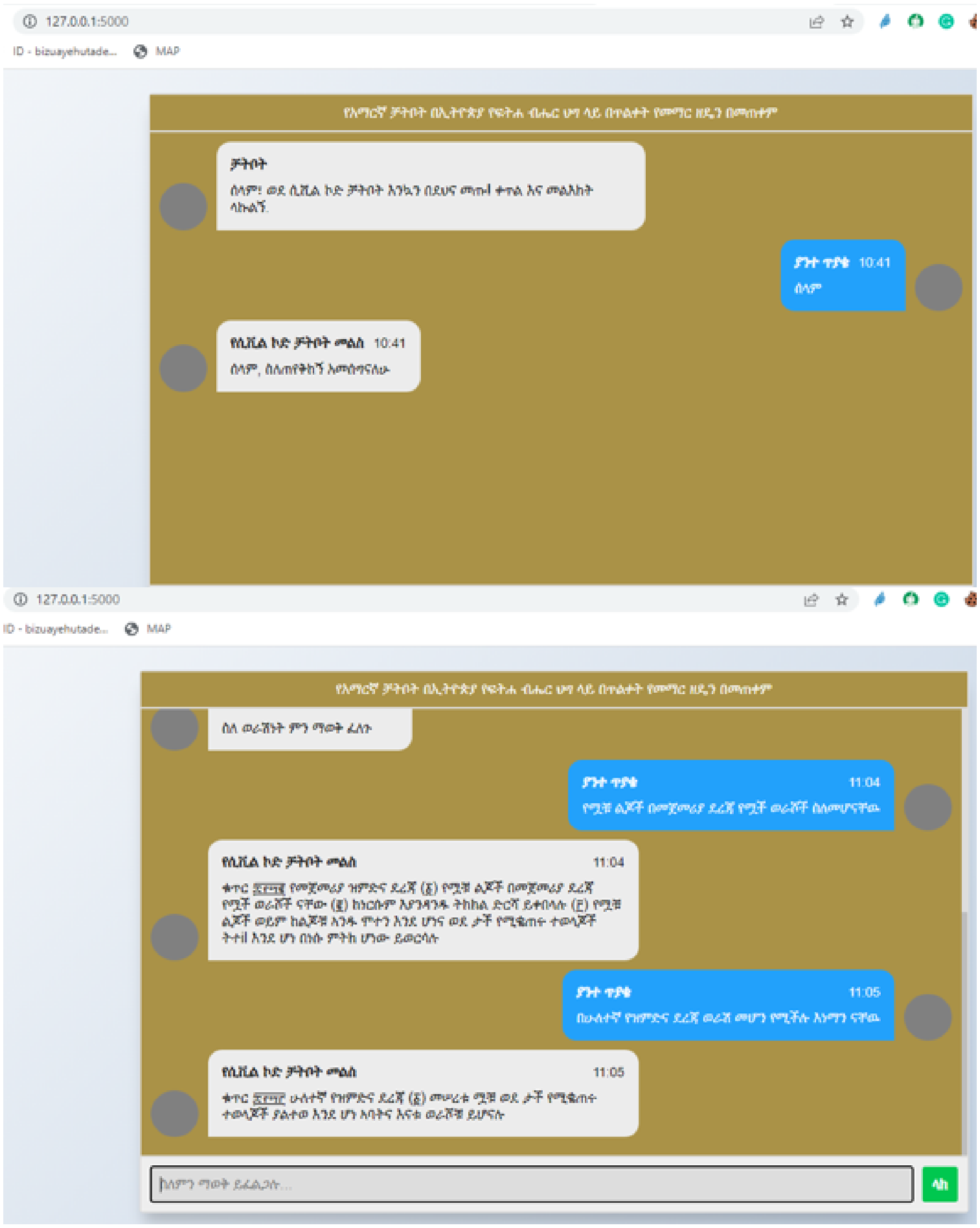

Figure 4. 5: Flask Web GUI civil code chatbot

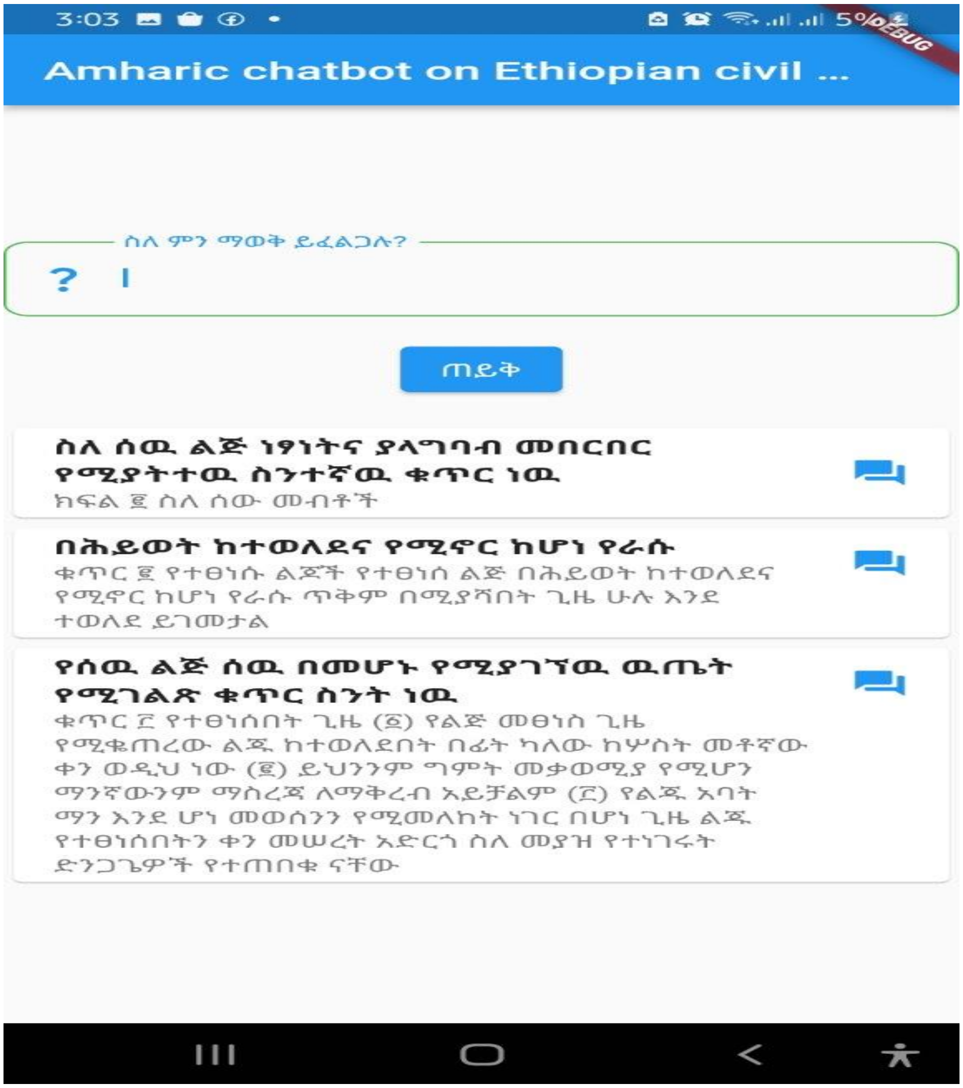

Figure 4. 6: Django mobile GUI civil code chatbot

The model was assessed by 30 users whose perspectives are used to generate their queries about civil codes. Of the 30 users chosen at random, 15 are knowledgeable users, 10 are novice users, and the remaining 5 are expert users. Ten pair conversations are held between each user and chatbot, as shown in Table 4.7 below.

Novice user: a person who is new or inexperienced in using a system and has no knowledge about civil code.

Knowledgeable users: are users who know about using a system but do not know about civil code.

Expert users: are users who have rich interaction knowledge about civil code, know how to use a system, and are skillful in formulating a query about civil code.

Table 4. 6: User evaluation results on model performance

User type	Number of users	Query asked	Correctly respond answer	Correctness in %
Novice	10	10	2	20
Knowledgeable	15	10	5	50
Expert	5	10	8	80

As we can see from Table 4.6, the users have evaluated the system's performance based on the number of queries given to it and the total number of queries it correctly responds. The accuracy was determined using the formula below.

$$Correctness = \frac{total\ number\ of\ question\ answered\ correctly}{total\ number\ of\ question\ requested\ by\ users} * 100$$

Figure 4. 7: Correctness performance evaluation result calculation(Hunde, 2021)

Based on this, the user filled out a questionnaire displayed in Appendix III. The criteria include the chatbot's responses' relevance, model usage, user satisfaction, and effectiveness based on the suggestion of an attribute. We used a rating scale of: excellent(4), very good(3), good(2), low(1), and very low(0)(Asimare, 2020; Hunde, 2021).

Once we have those data from the users, we average each metric to determine whether or not the users find the model acceptable. The following result demonstrates how the system accepts human results using the above criteria.

Table 4.7: User acceptance testing

Attribute	30 users' evaluation					
	Very low	Low	Good	Very good	Excellent	Average in %
Clarity of the language that the system responds to		1	6	14	9	75.83
Response relevance		2	5	12	11	76.67
Usage of the model		2	7	9	12	75.83
GUI attractiveness	2	3	6	7	12	70
Average						74.58

As we can see from Table 4.7, we based our evaluation of the suggested model on five user-provided attributes. We choose those criteria based on(Asimare, 2020; Hunde, 2021). Thirty users evaluated each parameter. Users completed assessment forms for all four parameters. As a result of our 30 user evaluations, each parameter has received a total score out of 120. We apply the following formula to determine the user acceptability of the suggested model based on those parameters.

$$Acceptance = \frac{\sum_{n=1}^{5} number\ of\ users\ select\ value * criterea\ value}{120} * 100$$

Figure 4. 8: Acceptance evaluation formula(Hunde, 2021)

The users suggest that the model gives and supports the people by giving them civil code information when needed and for general knowledge. They also suggest that the model works by voice commands, which are voice conversations. Especially for novice users, voice-based conversation is essential because they know what they want, but they can not formulate a query easily for the cases.

As shown in Table 4.7, correctly responded answers for novice users are minimal compared to knowledgeable users because they have some knowledge about query formulation and model usage, even though they do not know the civil codes. In addition, the correct reponed answer for an expert user is higher than the knowledgeable users because they know the civil codes and have an idea for query formulation for a given case.

Based on those four characteristics, we obtained a user acceptability score of 74.58%, demonstrating that the users are happy with the suggested model or it has a high level of acceptance.

4.7. Discussion

The primary goal of this study is to test machine learning techniques on an Amharic conversational chatbot that deals with civil law; incredibly, human and inheritance rights are discussed in this work. The prepared civil code dataset was used in a variety of research. Creating and implementing a conversational chatbot for the civil code was our primary focus to accomplish the thesis's objective. And by employing a transformer model, we achieved the best BLEU score of 9.88. Nevertheless, we have used three distinct models: LSTM and BiLSTM.

Additionally, we recommended the transformer model as the best model for anticipating answers to user queries. Since we have demonstrated the experimental findings, the transformer models are efficient in a minimal time and produce the best BLEU score result compared to others. The transformer model is the best when memory usage and training time are compared to LSTM and BiLSTM. Because the BiLSTM learns the data in both the forward and backward directions, it takes more time to train than LSTM, but the learning of the BiLSTM model maintains the context of a word in a sentence compared to LSTM. The transformer model facilitates long-range sentence dependency and parallel computation to encode a sentence. And this functionality makes the model outperform in predicting and training our dataset.

Our LSTM and Bi-LSTM models could not be run using 12 GB of RAM at their best performance. The advantage of the Transformers model is that, compared to other complex civil code articles, the interdependence of words at the beginning and end of long sentences becomes more significant. Because the interdependence of every word in a particular word order within a sentence determines the context, even though it is more context-dependent, it functions better for shorter sentences than longer ones(Kinfe, 2022). So far, our experiments have been with a civil code dataset and a user query. The training time and BLEU score metrics to evaluate performance using 50 epochs are shown here for the transformer model. The model took 341 seconds to train with 50 epochs, and we achieved a BLEU score of 9.88 for the prepared user query with parallel civil code article numbers.

However, the transformer model used a user query and the response for the given query in training a model, and we used 3215 parallel sentences to train and test the model. Compared to the dataset used for another conversational chatbot, it is small, so the model does not predict well. By increasing the dataset size, we can improve the model's performance. Generally, the transformer model is suitable for large dataset sizes and is the suggested model for the list of models used for civil code datasets.

4.8. Chapter Summary

In this chapter, we have carried out four experiments using deep learning models. In the experiment, we used 3215 parallel sentences that contain queries and answers. We employ uniform hyperparameters for the deep learning models (LSTM, BiLSTM, Transformer). Based on the conducted experiment, we achieved a BLEU score of 14.78 when comparing model predicting answer and civil code datasets, especially the human and inheritance rights with hyperparameters of dropout 0.2, learning rate 0.001, and embedding and latent dimensions of 128, 1024, respectively, on a 12GB RAM and GPU processor using the transformer model. The transformer chatbot model performs well for the user query and performs tasks in minimal time.

CHAPTER FIVE: CONCLUSION AND RECOMMENDATION

5.1. Overview

The study aims to develop a chatbot model using deep Learning to improve the conversation between a user query and an answer from a civil code corpus. We implemented Transformers, LSTM, BiLSTM, and transformer models. We use 12 GB RAM and GPU because the GPU version quickly processes and trains our models. As a result, we conducted our studies in as little time as possible.

5.1. CONCLUSION

The major objective of our thesis work is to develop and implement a conversational Amharic chatbot for civil code by preparing a pair of queries/answer datasets with an article number and utilizing a transformers model. Because of this, this thesis work is conducted on a civil code pair with a limited supply of resources using a data set of 3215 parallel corpora that were gathered from Amhara regional state supreme court office data sources. In this study, we followed an experimental research method to model and suggested the best model first carried out several experiments by manipulating hyperparameters. To conduct our experiment, we used preprocessing and feature extraction techniques as discussed in section 3.2. as embedding, we used the default karas embedding layer and default transformer model.

So, before selecting a transformer model, we tested the LSTM and Bi-LSTM models. Regarding time, memory, and BLEU score with the same parameters as the transformer model. Finally, the Transformers model is recommended as the best model from LSTM and BiLSTM.

Our experiments led to a better BLEU score of 9.88 for civil code user queries for our Transformers-based model. This indicates to the research question of to what extent the proposed civil code chatbot work does. The LSTM and Bi-LSTM models have consumed a lot of memory, and the response time is high too. The lack of data is a limitation of our research. Though conversational chatbots require enormous amounts of data for training and the creation of an ideal model that learns the various aspects of the queries and civil code articles, we have trained our model with a limited corpus due to the unavailable parallel user queries.

The unavailability of different user utterances for each civil code article makes it challenging to model a suited deep learning model; in this case, the performance of the transformer civil code answer is not as much as good for different user utterances in testing. Generally, the transformer model is recommended for our dataset because the model responds to a query compared to the LSTM and BiLSTM models. The transformer model understands the context of a sentence and supports long-ranged sentences.

5.2. Contribution of the study

Using deep learning, we proposed an Amharic chatbot for civil code. The contributions of this study are presented as follows:

- We proposed an Amharic civil code chatbot using a transformer chatbot model
- We have implemented various parameters to increase the model's accuracy and minimize loss.
- Dataset preparation for a designed model by collecting data from civil code document
- We compare a deep learning algorithm to suggest the best model for the prepared dataset

5.3. Future works

We suggest the following as a future research topic based on the analysis of the study's findings.

- As an initial investigation of the Amharic civil code conversational chatbot, there is no well-suited parallel corpus previously created for the civil code, so we gathered and prepared 3215 parallel corpora as a query with their response. A large dataset is required to increase the model's accuracy, so we recommend that future researchers collect more data on the civil code and other legal documents (criminal law, etc.).
- We advise future research to carry out a speech-based chatbot model because this study used only a text-based model chatbot
- We advise future researchers to check the syntax and grammatical structure of the user sentences to respond appropriately.
- Finally, we suggest implementing a conversational chatbot model that generates paragraph-based user suggestions for a given query.

Reference

Adamopoulou, E., & Moussiades, L. (2020a). An Overview of Chatbot Technology. In *IFIP Advances in Information and Communication Technology: Vol. 584 IFIP*. Springer International Publishing. https://doi.org/10.1007/978-3-030-49186-4_31

Adamopoulou, E., & Moussiades, L. (2020b). Chatbots: History, technology, and applications. *Machine Learning with Applications*, *2*(November), 100006. https://doi.org/10.1016/j.mlwa.2020.100006

Agumasie, N. (2012). *Chatbot based customer service model using Deep Learning: the case of Ethiopian Airlines*. *June*, 2–4.

Akkineni, H., Lakshmi, P. V. S., & Sarada, L. (2022). Design and Development of Retrieval-Based Chatbot Using Sentence Similarity. *Lecture Notes in Networks and Systems*, *244*(January), 477–487. https://doi.org/10.1007/978-981-16-2919-8_43

Allouch, M., Azaria, A., & Azoulay, R. (2021). Conversational Agents: Goals, Technologies, Vision and Challenges. *MDPI*, 1–48.

Alnefaie, A., Singh, S., Kocaballi, B., & Prasad, M. (2021). *An Overview of Conversational Agent: Applications, Challenges and Future Directions*. *Webist*, 388–396. https://doi.org/10.5220/0010708600003058

Asimare, H. (2020). *Designing and Implementing Adaptive Bot Model to Consult Ethiopian Published Laws Using Ensemble Architecture with Rules Integrated*.

Athota, L., Shukla, V. K., Pandey, N., & Rana, A. (2020). Chatbot for Healthcare System Using Artificial Intelligence. *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, 619–622. https://doi.org/10.1109/ICRITO48877.2020.9197833

Ayalew, A. (2021). *DESIGN EVENT EXTRACTION MODEL FROM AMHARIC TEXTS USING DEEP LEARNING APPROACH - PDF Free Download.pdf*.

Ayanouz, S., Abdelhakim, B. A., & Benhmed, M. (2020). A Smart Chatbot Architecture based NLP and Machine Learning for Health Care Assistance. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3386723.3387897

Batta, M. (2020). Machine Learning Algorithms - A Review . *International Journal of Science and Research (IJ*, *9*(1), 381-undefined. https://doi.org/10.21275/ART20203995

Biresaw, S. M. (2021). THE COMPARATIVE ANALYSIS OF THE CONTENTS AND STRUCTURES OF THE ETHIOPIAN CIVIL CODE VIS-À-VIS THE MAJOR EUROPEAN CODES COMPARATIVE ANALYSIS OF THE CONTENTS AND STRUCTURES OF THE ETHIOPIAN CIVIL CODE VIS-À À-VIS THE MAJOR EUROPEAN CODES * Samuel Maireg Bir. *International Journal of Current Research*, *December 2018*. https://doi.org/10.24941/ijcr.32015.11.2018

Blagec, K., Dorffner, G., Moradi, M., Ott, S., & Samwald, M. (2022). *A global analysis of metrics used for measuring performance in natural language processing*. *3*, 52–63. https://doi.org/10.18653/v1/2022.nlppower-1.6

Caldarini, G., Jaf, S., & McGarry, K. (2022a). A Literature Survey of Recent Advances in Chatbots. *Information (Switzerland)*, *13*(1). https://doi.org/10.3390/info13010041

Caldarini, G., Jaf, S., & McGarry, K. (2022b). A Literature Survey of Recent Advances in Chatbots. *Information (Switzerland)*, *13*(1), 1–24. https://doi.org/10.3390/info13010041

Chowdhury, S., & Chakraborty, P. pratim. (2017). Universal health coverage - There is more to it than meets the eye. *Journal of Family Medicine and Primary Care*, *6*(2), 169–170. https://doi.org/10.4103/jfmpc.jfmpc

Dhankhar, P., & Delhi, N. (2018). *RNN and LSTM based Chatbot using NLP*. *10*(2), 214–217.

Digital, D. (2021). *A Brief History of Neural Networks - DATAVERSITY*. https://www.dataversity.net/a-brief-history-of-neural-networks/#

Dutta, S. (2019). *Saarland University Department of Computational Linguistics Block Project : Neural Networks Software Project Evaluating a neural multi-turn chatbot using BLEU score Author : April*. https://doi.org/10.13140/RG.2.2.13374.84808

Fekadu Eshetu Hunde. (2021). *Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services*. https://doi.org/10.1088/1751-8113/44/8/085201

Fradkov, A. L. (2020). Early history of machine learning. *IFAC-PapersOnLine*, *53*(2), 1385–1390. https://doi.org/10.1016/j.ifacol.2020.12.1888

Gad, W., Alokla, A., Nazih, W., Aref, M., & Salem, A. B. (2022). DLBT: Deep learning-based transformer to generate pseudo-code from source code. *Computers, Materials* *and Continua*, *70*(2), 3117–3132. https://doi.org/10.32604/cmc.2022.019884

Gadge, H. (2021). *A Chatbot for Medical Purpose using Deep Learning*. *10*(05), 441–447.

Gereme, F., Zhu, W., Ayall, T., & Alemu, D. (2021). Combating fake news in “low-resource” languages: Amharic fake news detection accompanied by resource crafting. *Information (Switzerland)*, *12*(1), 1–9. https://doi.org/10.3390/info12010020

Ghojogh, B., Ghodsi, A. L. I., & Ca, U. (2017). *Attention Mechanism , Transformers , BERT , and GPT : Tutorial and Survey*.

Hagos, G., & Asefa, A. (2020). Linguistic Evolution of Ethiopic Language : A Comparative Discussion Nankai University. *International Journal of Interdisciplinary Research and Innovations*, *8*(1), 1–9.

Hammarström, H. (2016). Linguistic diversity and language evolution. *Journal of Language Evolution*, *1*(1), 19–29. https://doi.org/10.1093/jole/lzw002

Hiil. (2020). *Justice Needs and Satisfaction in Ethiopia*.

Initiative, E. justice. (2017). *Jerry Hartfield Released from Texas Prison After 35 Years Without Valid Conviction*. https://eji.org/news/texas-releases-jerry-hartfield-after-35-years-without-valid-conviction/

JACOB WILSON. (2022). *What is the effect of learning rate in neural network? – Technical-QA.com*. https://technical-qa.com/what-is-the-effect-of-learning-rate-in-neural-network/

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.

Kanakanti, C. R., & Sabitha, R. (2020). Ai and Ml Based Google Assistant for an Organization using Google Cloud Platform and Dialogflow. *International Journal of Recent Technology and Engineering*, *8*(5), 2722–2727. https://doi.org/10.35940/ijrte.e6354.018520

Kapočiute-Dzikiene, J. (2020). A domain-specific generative chatbot trained from little data. *Applied Sciences (Switzerland)*, *10*(7), 1–19. https://doi.org/10.3390/app10072221

Kaul, V., Enslin, S., & Gross, S. A. (2020). History of artificial intelligence in medicine. *Gastrointestinal Endoscopy*, *92*(4), 807–812. https://doi.org/10.1016/j.gie.2020.06.040

Klein, G., Kim, Y., Deng, Y., Crego, J., Senellart, J., & Rush, A. M. (2017). OpenNMT: Open-source toolkit for neural machine translation. *20th Annual Conference of the European Association for Machine Translation, EAMT 2017*, *December*, 22.

Kumar, S., & Rajagopal, P. (2020). Cloud Methodologies for A Seamless Integrated Chatbot. *International Journal of Engineering Research & Technology (IJERT)*, *9*(08), 44–48.

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2021). *A Survey of Transformers*. *1*(1). http://arxiv.org/abs/2106.04554

Mahendra N. (2020). Crop Prediction using Machine Learning Approaches. *International Journal of Engineering Research And*, *V9*(08), 23–26. https://doi.org/10.17577/ijertv9is080029

Mengistu Kinfe. (2022). *Amharic-Kistanigna Bi-directional Machine Translation using Deep Learning*. Bahir dar University.

Mierzwa, S. J., Souidi, S., Conroy, T., Abusyed, M., Watarai, H., & Allen, T. (2019). On the Potential, Feasibility, and Effectiveness of Chat Bots in Public Health Research Going Forward. *Online Journal of Public Health Informatics*, *11*(2). https://doi.org/10.5210/ojphi.v11i2.9998

Mindaye, T., Sahlemariam, M., & Kassie, T. (2010). The need for amharic WordNet. *Global Wordnet Conference, GWC 2010*.

Motger, Q., Franch, X., & Marco, J. (2021). *Conversational Agents in Software Engineering: Survey, Taxonomy and Challenges*. *1*(1). http://arxiv.org/abs/2106.10901

Nimavat, K., & Champaneria, T. (2017). Chatbots: An Overview Types, Architecture, Tools and Future Possibilities. *International Journal of Scientific Research and Development*, *5*(7), 1019–1026.

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

Pavel Surmenok. (2017). *Estimating an Optimal Learning Rate For a Deep Neural Network | by Pavel Surmenok | Towards Data Science*. https://towardsdatascience.com/estimating-optimal-learning-rate-for-a-deep-neural-network-ce32f2556ce0

Peace, I. C., Uzoma, A. O., & Ita, S. A. (2015). Effect of Learning Rate on Artificial Neural Network in Machine Learning. *International Journal of Engineering Research and Technology (IJERT)*, *4*(2), 359–363. www.ijert.org

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

Press, A. and floridapolitics. (2021). *Robert DuBoise, exonerated of 1983 murder, sues Tampa over imprisonment*. https://floridapolitics.com/archives/463567-robert-duboise-exonerated-of-1983-murder-sues-tampa-over-imprisonment/

Reddy, R. V. K., & Babu, U. R. (2018). A review on classification techniques in machine learning. *International Journal of Advance Research in Science and Engineering*, *7*(3), 40–47.

See, A. (2019). *Natural Language Processing with Deep Learning: Natural Language Generation*. *2022*, 1–39.

Shabbir, J., & Anwer, T. (2018). *Artificial Intelligence and its Role in Near Future*. *14*(8), 1–11. http://arxiv.org/abs/1804.01396

Shahin, A. I., & Almotairi, S. (2021). A deep learning BiLSTM encoding-decoding model for COVID-19 pandemic spread forecasting. *Fractal and Fractional*, *5*(4). https://doi.org/10.3390/fractalfract5040175

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

Shubhashri, G., Unnamalai, N., & Kamalika, G. (2018). LAWBO: A smart lawyer chatbot. *ACM International Conference Proceeding Series*, 348–351. https://doi.org/10.1145/3152494.3167988

SIL international. (2022). *How many languages are there in the world? | Ethnologue*. https://www.ethnologue.com/guides/how-many-languages

Singh, S. (2020). *A survey paper on chatbot*. *June*.

Sojasingarayar, A. (2020). *Seq2Seq AI Chatbot with Attention Mechanism*. *May*, 1–18. http://arxiv.org/abs/2006.02767

Tan, Z., Wang, S., Yang, Z., Chen, G., & Huang, X. (2021). Neural machine translation : A review of methods , resources , and tools. *AI Open*, *1*(October 2020), 5–21. https://doi.org/10.1016/j.aiopen.2020.11.001

Tarwani, K., & Edem, S. (2017). Survey on Recurrent Neural Network in Natural Language Processing. *International Journal of Engineering Trends and Technology*, *48*(6), 301–304. https://doi.org/10.14445/22315381/ijett-v48p253

Teshome Kassie. (2009). *Word Sense Disambiguation for Amharic Text Retrieval: A Case Study for Legal Documents* [ADDIS ABABA UNIVERSITY]. http://etd.aau.edu.et/bitstream/handle/123456789/3315/Teshome Kassie.pdf?sequence=1&isAllowed=y

Torfi, A., Shirvani, R. A., Keneshloo, Y., Tavaf, N., & Fox, E. A. (2020). *Natural Language Processing Advancements By Deep Learning: A Survey*. 1–23. http://arxiv.org/abs/2003.01200

Vakili, M., Ghamsari, M., & Rezaei, M. (2020). *Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification*. http://arxiv.org/abs/2001.09636

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 5999–6009.

Vijayaraghavan, V., Cooper, J. B., & Rian Leevinson, R. L. (2020). Algorithm Inspection for Chatbot Performance Evaluation. *Procedia Computer Science*, *171*(2019), 2267–2274. https://doi.org/10.1016/j.procs.2020.04.245

Vira, I., & Vasiļjevs, A. (2014). The Development of Conversational Agent Based Interface. *Frontiers in Artificial Intelligence and Applications*, *268*(April), 46–53. https://doi.org/10.3233/978-1-61499-442-8-46

Wahde, M., & Virgolin, M. (2022). *Conversational Agents: Theory and Applications*. https://doi.org/10.1142/9789811246050_0012

Wang, B., & Kuo, C. C. J. (2020). SBERT-WK: A Sentence Embedding Method by

Dissecting BERT-Based Word Models. *IEEE/ACM Transactions on Audio Speech and Language Processing*, *28*(8), 2146–2157. https://doi.org/10.1109/TASLP.2020.3008390

Wang, C., Nulty, P., & Lillis, D. (2020). A Comparative Study on Word Embeddings in Deep Learning for Text Classification. *ACM International Conference Proceeding Series*, 37–46. https://doi.org/10.1145/3443279.3443304

Wang, H., Wu, H., Zhu, H., Miao, Y., Wang, Q., Qiao, S., Zhao, H., Chen, C., & Zhang, J. (2022). A Residual LSTM and Seq2Seq Neural Network Based on GPT for Chinese Rice-Related Question and Answer System. *Agriculture*, *12*(6), 813. https://doi.org/10.3390/agriculture12060813

Wayesa, F. (2020). *Design and Implementation of a Rule Based Afaan Oromoo Conversational Chatbots*. *5*(8), 27–30.

Wondwossen Mulugeta, & Michael, G. (2014). Learning Morphological Rules for Amharic Verbs Using Inductive Logic Programming Learning Morphological Rules for Amharic Verbs Using Inductive Logic Programming. *Researchgate*, *May 2012*, 7–12. https://doi.org/10.13140/2.1.5171.2001

Workineh Wogaso. (2020). *Attention-based Amharic-to-Wolaita Neural Machine Translation* (Issue October). MSc Thesis, Addis Ababa University, Addis Ababa, Ethiopia.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing [Review Article]. *IEEE Computational Intelligence Magazine*, *13*(3), 55–75. https://doi.org/10.1109/MCI.2018.2840738

Appendixes

Appendix I: Civil code query with answer dataset preparation

User query	Civile code answer
ስለ ሰው ልጅ የመኖሪያ ቤት አለመደፈርና ስለ ደንነቱ ምን ብሎ ያስቀምጣል.	ቁጥር፡ ፲፫ ፡ የመኖሪያ ፡ ቤት ፡ አለመደፈር ።(፩) ያንድ ፡ ሰው ፡ ማናቸውም ፡ መደበኛ ፡ መኖሪያ ፡ ቤት ፡ የማይደፈር ፡ ነው ። (፪) በሕግ ፡ በተመለከተው ፡ መሠረት ፡ ካልሆነ ፡ በቀር ፡ ባለቤቱ ፡ ሲቃወም ፡ማንኛውም፡ ሰው ፡ ከሌላ ፡ ሰው ፡ መኖሪያ ፡ ቤት፡ ለመግባት፤ የሌላውን ፡ ሰው፡መኖሪያ ፡ ቤት ፡ ለመበርበር፡ ኣይችልም ።
አንድ ሰው የማሰብ ነፃነት አለው ማለት ምን ማለት ነው.	ቁጥር፡፲፬ ። የማሰብ ፡ ነፃነት ።(፩) ማንኛውም፡ ሰው ፡ የማሰብና ፡ አሳቡን ፡ የመግለጽ ፡ ነፃነት ፡ አለው ። (፪) ይህንንም ፡ ነጻነቱን ፡ ሊቀንሱበት ፡ የሚችሉት፡ የሌሎችን ፡ መብቶች ፡የማክበር ፡ ግዴታዎች ፤ መልካም ፡ ባህልንና ፡ ሕጎችን ፡ አክብሮ ፡ የመጠበቅ ፡ ግዴታዎች ፡ ናቸው ።
ስለ ፡ ማግባትና ፡ ስለ፡መፍታት እንዲሁም ስለ ውል ግዴታ የተደነገገው ምን ቁጥር ላይ ነው ምን ብሎስ አስቀምጠው.	ቁጥር፡ ፲፯፡ ስለ ፡ ማግባትና ፡ ስለ፡መፍታት ።(፩) ማንኛውም ፡ ሰው ፡ ሚስት ፡ እንዳያገባ ፡ ወይም ፡ እንደ ፡ ገና ፡ እንዳያገባ ያደረገው፡የውል ግዴታ ፡ ሁሉ ፡ በፍትሐ ብሔር ፡ ሕግ ፡ አስተያየት፡ረገድ፡ዋጋ፡ የሌለው፡ነው ።(፪) እንዲሁም ፡ ማንም ፡ ሰው ፡ እፈታለሁ ፡ ወይም ፡ አልፈታም ፡ ብሉ፡ የሚገባው፡ ግዴታ፡ ዋጋ ፡ የለውም ።
ልደት ፤ ሞትና፡ ጋብቻ ፡ ያጠራጠሩ፡ እንደ ፡ ሆነ ፍትሐብሔር ህጉ በማን አስረጂነት ይጣራ ይላል.	ቁጥር፡ ፵፯ ። የአስረጅነቱ ፡ ዐይነት ።(፩) ልደት ፤ ሞትና፡ ጋብቻ ፡ ያጠራጠሩ፡ እንደ ፡ ሆነ ፤ ወይም ፡ ክርክር ፡ የተነሣባቸው ፡ እንደ ፡ ሆነ ፡የሚረጋገጡት ፡ በክብር ፡ መዝገብ ፡ ጽሑፍ፡ አስረጅነት ፡ ነው ። (፪) እንዲሁም ፡ ሕግ ፡ በሚፈቅድበት ፡ ጊዜ ፡ በታወቀ ፡ ሰነድ ፡ አስረጅ ፡ አማካይነት ፡ ወይም ፡ በሁኔታ ፡ መኖር ፡ ሊረጋገጡ ፡ ይችላሉ ።

የመሞት ፡ ጽሑፍ ፡ የሚያመለክተው ምን ምን መረጃ ማካተት ሲችል ነው ብሎ ያስቀመጠ እንደሆነ ነው. ያስታወቀውን ፡ ሰው ፡ የቤተ ፡ ዘመድና ፡ስም ፡ የግል ፡ ስም ፤የተወለደበትን ፡ ቀንና ፡ ቦታ ፡ ነው ።	ቁጥር ፻፬ ። የጽሑፉ ፡ ቃል "የመሞት ፡ ጽሑፍ ፡ የሚያመለክተው (ሀ) የሞተበትን ፡ ቀን ፡ ወርና ፡ ዓመት ።(ለ) የሞተውን ፡ ሰው ፡ የቤተ፡ ዘመድ ፡ ስም ፡ የግል ፡ ስም ! የተወለደበትን ፡ቀንና ፡ ቦታ (ሐ) የአባቱን ፤ የእናቱን ፤ የቤተ ፡ ዘመድና፡የግል ፡ ስም ፡ የተወለደበትን ' አገርና ፡ ቀን ።(መ) በሕይወት፡ ያለ፡ እንደ ፡ ሆነ ፡ የሞተውን ፡ ሰው ፡ ባል ፡ ወይም ፡ ሚስት ፤የቤተ ፡ ዘመድ ፡ ስምና ፡ የግል ፡ ስም ፤ የተወለደበትን ፡ ቀንና ፡ ቦታ ጋብቻው ፡ የተደረገበትን ፡ ቀን ።(ሠ) ያለም ፡ እንደ ፡ ሆነ'የሞተውን ፡ ያስታወቀውን ፡ ሰው ፡ የቤተ ፡ ዘመድና ፡ስም ፡ የግል ፡ ስም ፤የተወለደበትን ፡ ቀንና ፡ ቦታ ፡ ነው ።
ሰፈር፡ ወይም፡የቀበሌ ፡ ክፍል ፡እንዴት ይከለላል እንዴትስ ይሰየማል.	ቁጥር ፵፱ ሰፈር፡ ወይም፡የቀበሌ ፡ ክፍል ፡(፩) አንድ፡ የከተማ ቀበሌ ፡ በብዙ ፡ ሰፈሮች ፡ የተከፈለ ፡ እንደ ፡ ሆነ ፡ ለያንዳንዱ ፡ ልዩ ፡ ሰፈር፡ አንዳንድ፡ የተለየ ፡ የክብር ፡ መዝገብ ፡ ምን ፡ ጠቅላይ ፡ ገዢው፡ ሊሾም፡ይችላል ።(፪) እንዲሁም'ያንድ ፡ የባላገር ፡ ቀበሌ ፡ ክፍል ፡ ለብቻው ፡ የሆነ ፡ ወይም ፡ ከዋናው፡ ቀበሌ፡ የተራራቀ ፡የሆነ ፡ እንደ ፡ ሆነ፡አገረ፡ ገዢው ፡ ለዚሁ፡የተለየ ፡የክብር ፡ መዝገብ ፡ ሹምን ፡ ለመሾም ፡ ይችላል ።(፫) እንዲህም ፡ ሆነ ፡ ጊዜ ፡ ለያንዳንዱ ፡ የክብር ፡ መዝገብ ፡ ሹም ፡ አንድ ፡ ወይም፡ብዙ ፡ ምትክ ፡ ይሾምላቸዋል ።
የመርከብ፡አዛዦች በመርከብ ላይ ሁነው ስለሚከናውን ማንኛውም ተግባር እንደምን ይሁኑ ይላል.	ቁጥር ፶፫ የመርከብ፡አዛዦች ።የኢትዮጵያ፡ መርከብ' አዛዦች፡በመርከባቸው፡ ውስጥ፡ በሚደርሰው፡ ልደት ፡ሞትና፡ጋብቻ፡ረገድ፡ የክብር፡ መዝገብ ሹሞች ናቸው

Appendix II: User acceptance testing questionnaire form

<u>የሰው ሰራሽ ፍትሕብሔር   ህግ አማካሪ ምዘና ወረቀት</u>

በመጀመሪያ የሰው ሰራሽ ህግ ማማከሪያ ሲስተም ምዘና ለመሙላት ፈቃደኛ ስለሆኑ እናመሰግናለሁ።

እናም ይህ ወረቀት በሚጠይቀው መሰረት እያንዳንዱን ምዝና በትኩረት እንዲሞሉልን በአክብሮት እንጠይቃለን!

ማሳሰቢያ: በኮምፒውተር የተከፈተው ሲስተም ጋር በመገናኘት በኢትዮጵያ ፍትሕብሔር ህጎች ዙሪያ የሚፈልጉትን ጥያቄ፣ ፍትሕብሔር ህግ ምክር ወይም ትምህርቶች በቀረበው የኮምፒውተር ሲስተም በመጻፍ ንግግርወትን ይጀምሩ። ቀጥሎ ከዚህ በታች ያለውን ጥያቄወች "√" ምልክት በማድረግ ይሙሉልን።

መጠይቆች	በጣም ዝቅተኛ	ዝቅተኛ	ጥሩ	በጣም ጥሩ	እጅግ በጣም ጥሩ
1. ሲስተሙ የሚመልስልወት ቋንቋ ግልጽነት ?					
2. ሲስተሙ የሚመልሰውን መልስ እንዴት አገኙት ?					
3. የዚህ ሲስተም ጠቀሜታ በምን ይገልጹታል ?					
4. የሲስተሙ አሰራር ከአጠቃቀም አንጻር፤ በቀላሉ ንግግር ከማድረግ አንጻር እንዴት ይመለከቱታል ?					

አስተያየት ካለወት እዚህ ላይ ይጻፉልን!

____
____
____
____
____
____

Appendix III: dataset questionnaire form

የአማራ ብሔራዊ ክልላዊ መንግስት

ጠቅላይ ፍርድ ቤት

The Amhara National Regional State

Supreme Court

ቁጥር አብክመ/ጠ/ፍ/ም/ፕ/93/7/2014

ቀን 1/7/2014 ዓ.ም

- ለሰበር ዳኝ/አገ/አሰ/ዳይሬክቶሬት
- ለአይሲቲ ዳይሬክቶሬት
- ለፍትሐብሔር ዳኝ/አገ/አሰ/ዳይሬክተር
- ለወንጀል ዳ/አገ/አሰ/ዳይሬክቶሬት
- ለዕቅድ በ/ዝግ/ክ/ግ/ዳይሬክቶሬት

ጠ/ፍ/ቤት

- ለ----------------------ዞን ከፍተኛ ፍርድ ቤት

በያሉበት

ጉዳዩ፡-ትብብርን ይመለከታል

የባህር ዳር ዩኒቨርስቲ ቴክኖሎጂ ኢንስቲትዩት በቁጥር FC/206/14 በቀን 01/07/2014 ዓ.ም በተፃፈ ደብዳቤ በኢንስቲትዩት የኮምፒዩቲንግ ፋካልቲ ባልደረባ እና የሁለተኛ ዲግሪ ተማሪ የሆኑት አቶ ብዙአየሁ ታደገ "Amharic Chatbot for Customers using Deep Learning Algorithm on Ethiopia Published Law" በሚል ርዕስ ጥናት ለመስራት ጉዳዩ ከሚመለከታቸው ከጠ/ፍ/ቤቱ እስከ ወረዳ ፍርድ ቤቶች ድረስ ካሉት የስራ ክፍሎች አስፈላጊውን መረጃ እንዲያገኝ ትብብር እንዲደረግላቸው ጠይቋል፡፡

ስለሆነም ከላይ የተጠቀሳችሁት የጠ/ፍ/ቤቱ ዳይሬክተሮች እና የ13ቱ ዞን ከፍተኛ ፍርድ ቤቶች፣ ተገቢውን ትብብር እንድታደርጉላቸው እናሳስባለን፡፡

« ከሰላምታ ጋር »

ግልባጭ

ለብዙአየሁ ታደገ

ባህር-ዳር

ስልክ :- (+251 -58) 2 20 09 15 / 2 20 10 97
ፋክስ :- (+251-58) 2 26 41 05 / 2 20 40 96
ፖስታ ሣጥን ቁጥር :- 901
E-mail Add. :- supremcourt@Yahoo.com
Web Add. : - www.amhara.gov.et

አድራሻ :-
ባህር ዳር
ኢትዮጵያ

Appendix IV: Transformer encoder-decoder architecture