HAWASSA UNIVERSITY

INSTITUTE OF TECHNOLOGY

FACULTY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

AMHARIC MULTI-HOP QUESTION ANSWERING IN HISTORICAL TEXTS: A DEEP LEARNING APPROACH

M.Sc. Thesis

BEREKET ENDALE

HAWASSA UNIVERSITY, HAWASSA, ETHIOPIA

NOVEMBER, 2024

AMHARIC MULTI-HOP QUESTION ANSWERING IN HISTORICAL TEXTS: A DEEP LEARNING APPROACH

BEREKET ENDALE

A THESIS SUBMITTED TO THE

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF INFORMATICS, SCHOOL

OF GRADUATE STUDIES

HAWASSA UNIVERSITY

HAWASSA, ETHIOPIA

IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE

DEGREE OF

MASTER OF SCIENCE IN COMPUTER SCIENCE

NOVEMBER, 2024

Declaration

I hereby declare that this thesis is my original work and has not been presented for a degree in any other university. All sources of material used for this thesis have been acknowledged appropriately.

Name: Bereket Endale Ararso

Signature: Andargachew Mekonnen (PhD)

Date: _____

SCHOOL OF GRADUATE STUDIES HAWASSA UNIVERSITY ADVISORS' APPROVAL SHEET

This is to certify that the thesis entitled "AMHARIC MULTI-HOP QUESTION ANSWERING IN HISTORICAL TEXTS: A DEEP LEARNING APPROACH" submitted in partial fulfilment of the requirements for the degree of Master of Science (MSc) with specialization in Computer science, the Graduate Program of the Department of computer science in faculty of informatics, and has been carried out by Bereket Endale, under my supervision.

Therefore, I recommend that the student has fulfilled the requirements and hence hereby can submit the thesis to the department.

Andargachew Meknonnen (PhD)		25/11/2024
Name of Advisor	Signature	Date

SCHOOL OF GRADUATE STUDIES HAWASSA UNIVERSITY EXAMINERS' APPROVAL SHEET

We, the undersigned, members of the Board of Examiners of the final open defence by **Bereket Endale,** have read and evaluated her thesis entitled "AMHARIC MULTI-HOP QUESTION ANSWERING IN HISTORICAL TEXTS: A DEEP LEARNING APPROACH", and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfilment of the requirements for the degree.

Andargachew Meknonnen (PhD)		25/11/2024
Name of Advisor	Signature	Date
Tesfaye Bayu (PhD)		
Name of Internal Examiner	Signature	Date
Teshager Kassa (M.Sc.)		
Name of Internal Examiner	Signature	Date
Michael Melese (PhD)		27/11/2024
Name of External Examiner	Signature	Date
SGS Approval	Signature	Date

Final approval and acceptance of the thesis is contingent upon the submission of the final copy of the thesis to the School of Graduate Studies (SGS) through the Department/School Graduate Committee (DGC/SGC) of the candidate's department.

Stamp of SGS Date:_____

Acknowledgment

First and foremost, I would like to express my deepest gratitude to **Almighty God** for providing me with the strength, guidance, and blessings to complete this work. Without His grace, this thesis would not have been possible.

I am also deeply indebted to my advisor, **ANDARGACHEW MEKONNEN (PhD)**, for his invaluable advice, insightful feedback, and unwavering support throughout the course of this research. His guidance has been instrumental in bringing this work to completion.

A special thanks goes to my family for their continuous support, advice, and patience. Their involvement and encouragement have been a cornerstone in completing this work.

Finally, I am grateful to my friends for their motivation and support, which helped me stay focused and finish this thesis. Once again, I conclude with heartfelt thanks to Almighty God. Words cannot fully express my gratitude for His infinite support.

ABSTRACT

In our daily lives, questioning is the most effective way to gain knowledge. However, manual extraction of answers is time-consuming and requires expertise in the field. As a result, implementing fully question answering could accelerate extraction times and reduce the requirement for human labour. Numerous studies have been done on question answering in full resource languages like English, and others using various recent techniques. However, unlike previous research, which concentrated exclusively on single-hop question answering, this thesis proposes the concept of multi-hop question answering in Amharic. Until yet, no studies have investigated multi-hop question answering in the context of the Amharic language, which includes reasoning over numerous pieces of evidence or documents to generate an answer. Furthermore, there is no existing question answering data set to address these issues; therefore, this study used deep learning for the Amharic multi-hop question answering problem, a neural network method. To do this, we preprocess our dataset using tokenization, normalization, stop word removal, and, padding before feeding it to a deep learning model such as CNN, LSTM, and Bi-LSTM to create question type classification based on the given input. Because there is no multi-hop Question answering training dataset in Amharic, training data must be created manually, which is time-consuming and tedious. It is around 1500 questions and contexts associated with five classes. The class depicts as ((0) for factoid_date, (1) for factoid_person, (2) for factoid_location, and (3) for factoid_organization. Accuracy, precision, the F-measure, and the confusion matrix are performance metrics used to evaluate the model's overall efficiency when applied to the provided dataset. According to performance measurements, the maximum achievable accuracy rates for this study's LSTM, CNN, and Bi-LSTM were 96%, 96.38%, and 97.04%, respectively. The findings indicated that the suggested Bi-LSTM outperformed the other two models in terms of Amharic multi-hop questions type classification.

Table of Contents

List of tables

List of Figures

List of Abbreviations

AI: Artificial Intelligence

AMHQAS: Amharic Multi Hop Question Answering System

Bi-LSTM: Bidirectional Long Short-Term Memory

CBOW: Continuous Bag of Words

CNN: Convolutional Neural Network

Glove: Gloss Vector

GRU: Gated Recurrent Unit

IR: Information Retrieval

LSTM: Long Short-Term Memory

ML: Machine Learning

NER: Named Entity Recognition

NLP: Natural Language Processing

POS: Part Of Speech

QA: Question Answering

QAS: Question Answering System

ReLU: Rectified Linear Units

RNN: Recurrent Neural Network

Word2vec: Word to Vector

CHAPTER ONE

1. INTRODUCTION

1.1. Background

A large amount of electronic information is generated by various individuals, organizations, and nations worldwide. Additionally, there is important quantity of electronic documents in Amharic. As we increasingly navigate large volumes of information, especially on the Internet, automatic knowledge discovery and information are becoming increasingly essential. Accessing relevant information creates a considerable challenge for users across all fields of knowledge. Specifically, users often lack the time to filter out through numerous available documents to find concise and accurate responses to their queries. As a result, Information Retrieval and Information Extraction are gaining importance for effectively searching and utilizing this information.

Information Retrieval is both an art and a science focused on extracting relevant information from a collection of documents based on a user's query [1]. In an Information Retrieval (IR) system, such as search engines, users can ask questions in natural language. The IR systems extract keywords from these questions and utilize sophisticated search methods across a document database, returning a ranked list of documents instead of a straightforward answer. However, users still need to filter out through these documents to find the answers they seek, which presents a challenge in obtaining relevant and brief information. In these instances, Question Answering systems serve as an effective solution to address this challenge [2].

Question answering is an automated method for retrieving brief, accurate responses in natural language to questions posed by humans [1]. In Natural Language Processing (NLP), the Question Answering (QA) system is a man-machine communication tool that gives users accurate, brief answers to their questions. When standard document retrieval systems and question answering systems are compared, it can be seen that QAS must provide a specific response to a Natural Language (NL) query, standard document retrieval systems only return relevant documents to user queries, meaning that

traditionally, IR focuses on finding entire documents while QAS attempts to provide only one or a small set of specific answers to an input question [1]. This is because, in comparison to most web search engines, question answering systems demand a far deeper comprehension and processing of text [3].

According to [4] question-answering systems fall into two types. Single-hop question answering deals with questions that may be answered with information from a single source or context. In contrast, multi-hop question answering requires gathering information from multiple sources or contexts in order to generate an answer.

Multi-hop question answering (QA) is a problem in natural language processing (NLP) where questions are answered by logically interpreting multiple contexts or bits of information. When using multi-hop, the system needs to gather and combine information from multiple sources [4], as opposed to single-hop, which just needs one paragraph or page to have the answer. Usually, this approach includes establishing links between concepts or details from one text passage to another, followed by the application of logic to determine the correct answer. To respond to fact-based, multi-paragraph (multi-document) questions that require more in-depth understanding and reasoning, good retrieval is necessary, it is commonly used in applications like information retrieval, reading comprehension, and dialogue systems [4].

Recent developments in deep learning have shown promise for QA with neural network models. These systems need a large amount of training even though they often include a smaller learning pipeline [5]. Deep learning is an essential method for research jobs including question answering. In recent years, deep learning techniques have been used for important question answering investigations. Deep learning algorithms that learn from examples are used to train machines.

Deep learning is widely applied in sectors like advertising, e-commerce, and entertainment. It uses artificial neural networks to carry out complex calculations on enormous amounts of data [5]. In a few simple steps, neural models allow for flexible

natural language creation and processing, eliminating the need for manually developed rules to extract context-specific information. Remarkable performance advantages over more traditional systems have been achieved recently with advances in neural models [6]. Modern QA models, particularly those based on deep learning [7] perform better as a result of the rapid advancement of computing technology.

In this study, we present AMHQA - Amharic Multi-Hop Question Answering which has the ability to handle the types of factoid date questions, factoid location questions, factoid Organization questions and factoid person questions that are commonly asked in the history domain. Neural network models are used in today's state-of-the-art question-answering systems. Thus, our objective is to design a system that is capable of improving the performance of question-answering system for history domain.

1.2. Statement of the problem

People need to grasp information in day-to-day activities. Because they want to solve problems, it develops the thinking potential and practice something and facilitate the learning environment, also people may need precise information to fill their background knowledge. But finding relevant information from a huge source of documents might not return relevant answer for a particular query. While searching on the web the user gets a relevant document that might hold answer for their query, which require reading each document to get the exact answer for the question, it is time-consuming and tedious. So, developing question answering can reduce the time that required to get the answer.

Amharic is the most widely spoken language in Ethiopia. It has its own alphabet and phonological and morphological features. Words are represented by phonemes/characters. Amharic has both constant and vowel characters where the base forms of consonants are modified by following the vowel [8]. The language serves as the official working language of the Ethiopian federal government, and is also the official or working language of several of Ethiopia's federal regions [9]. As a result, availability of Amharic language textual information is highly increasing from time to time.

There are pioneer research works in Amharic question answering that are Amharic Question Answering for Definitional, Biographical and Description Questions[10], Amharic Question Answering for Factoid and List Questions using Machine-learning Approaches [11]. And the other existing research focused on Designing an Amharic Question Answering model for healthcare using a deep learning approach [12] it combines both factoid and non-factoid questions, Existing methods, frequently fail to offer appropriate responses to context-specific or reasoning-dependent queries, resulting in decreased system performance when dealing with complicated or multi-hop issues. Furthermore, while great progress has been made in Amharic question answering for single-hop tasks, there has been little to no investigation into multi-hop question answering systems in the Amharic language, which are critical for answering questions that require reasoning across numerous texts.

There are other question types like why, list, and multi-hop factoid questions are another research direction. Including multi-hop factoid date questions such as" በኢትዮጵያ ታሪክ እና በአለም አቀፍ ዲፕሎማሲ ውስጥ ቁልፍ ሰው የነበሩት ንጉስ ኢትዮጵያን የገዙት በየትኞቹ አመታት ነው?" ("In what years did the king, who was a key figure in Ethiopian history and international diplomacy, rule Ethiopia?"), factoid person questions such as "የአፄ ፋሲለደስ አባት የዙፋን ስማቸው ማን ይባላል?" ("What is the throne name of Emperor Fasiledus' father?") factoid location questions "የአፄ ኃይለ ሥላሴ እናት የት ተወለዱ?" ("Where was Emperor Haile Selassie's mother born?") factoid organization questions such as "የኢትዮጵያ ንግስት እሌኒ በእንጀራ ልጃቸው ዘመነ መንግስት ስልጣን የያዙበት ከተማ ውስጥ ያለው ታሪካዊ ቦታ ምን ይባላል?" ("What is the name of the historical place in the city where Queen Eleni of Ethiopia took power during the reign of her stepson?") which require the integration of multiple information or needs reasoning to give answer for the question. The factoid questions mentioned above cannot be answered by existing Amharic factoid question answering and Amharic non-factoid question answering because their answers are not directly extract from single source. This barrier blocks the development of advanced Amharic question answering systems for real-world applications such as education and research, where reasoning across many contexts is critical. To close this gap, this study presents a unique deep learning-based strategy for

creating an Amharic multi-hop question answering system (AMHQA) capable of accurately understanding and answering complicated multi-hop questions.

1.3. Research Questions

This study attempts to answer the following research questions:

I. To what extent the Deep learning approach provide correct answer for User's multi-hop kind of query?
II. Which of the deep learning algorithm is more effective for Amharic multi-hop question answering?
III. What are the optimal values of hyperparameters that give best performance?

1.4. Objective

1.4.1. General Objective

The general objective of the proposed research is to design and develop Amharic Multi-Hop Question Answering by using a Neural Network Approach.

1.4.2. Specific Objective

In order to accomplish the previously mentioned overall goal, the following particular objectives are carried out: -

- To review related works written in various languages, including Amharic.
- To study the general features of Amharic multi-hop questions.
- To prepare training data (corpus) from Amharic articles.
- To conduct experiments to find optimal hyperparameter values which provides improved performance for the model.
- To build models that used to classify multi-hop questions.
- To train the model using the selected machine learning approaches.
- To evaluate the model.

1.5. Significance of the study

Designing Amharic multi-hop question answering by using deep learning approach is a contribution of this research. Sometimes, users may forget or be unaware of key information when asking questions, such as 'What is the throne name of Emperor Fasiledus' father?' In this case, the user does not know who Emperor Fasiledus' father is, which makes it necessary to consider multi-hop questions. Multi-hop questions are essential because they guide users through a step-by-step reasoning process, allowing them to retrieve and connect the required information to find the answer. The research improves the understanding of Amharic multi-hop question answering. AMHQA has a great contribution in different natural language and real-world applications such as entertainment, customer service and education etc.

1.6. Scope of the study

This research aims to develop Amharic multi-hop question answering from only Amharic text documents. It focuses on multi-hop question answering, where it performs extraction of answer from multiple contexts or sources to answer a question. Moreover, this research uses only Amharic textual documents from electronic newspapers and other Amharic written hardcopy documents without any table, figure and pictorial representations or image Since the study focuses on a particular field, it only covers issues related to history spanning from ancient to the present. We focus in this particular topic since the researcher thinks there are a lot of questions that can be asked in this field.

1.7. Limitation of the study

The magnitude of the dataset used in order to understand the pattern of semantic associations between objects, we used a deep learning approach called LSTM/BiLSTM and CNN in this study. This strategy requires a huge number of training datasets. However, it was challenging to find further Amharic multi-hop related data, and only 1500 question-type pair were used in this investigation. The thesis focuses on a specific domain (history), only used Amharic text data sets for the experimental purpose because we are familiar to the language. We feel that numerous questions can be raised and used in QA

in this domain. This research tries to answer only Amharic Multi-hop factoid question (date, organization, location and person).

1.8. Organization of the study

This chapter is one of five that make up the organization of the research. The second chapter included a survey of the literature on similar studies and various researchers approaches. The research system architecture is given in Chapter 3. The experiment carried out, the outcome, and a discussion of the findings were the main topics of Chapter 4. Finally, Chapter five discusses the study's conclusion and recommendations.

CHAPTER TWO

2. LITERATURE REVIEW

This chapter covers the question answering framework and context. Included are a definition, background information, and a selection of significant and helpful literature. The current state of the art in this field is also discussed, along with related works, methodologies, and strategies, as well as questions and answers.

2.1. Overview of question answering

"*Question answering (QA) is a field of natural language processing (NLP) and artificial intelligence (AI) that aims to develop systems that can understand and answer questions posed in natural language*." [13]. Question answering is tasked with understanding a given question and giving an appropriate answer based on the information it has been trained on. The input has a means of possibilities depending on the application, it can be image [14] the aim is to accurately respond in natural language to a question regarding an image given the image and the question, text [15],[16] and knowledge graph-based [17]. The main goal of question answering is to provide accurate, relevant, and understandable answers to questions posed by users.

Based on the dataset in which the question answering model trained on it can be classified into two classes: closed domain QA and open domain QA. If the model trained on some specific dataset eventually, we will get a closed domain QA which is capable of answering that typical specific dataset. In contrast if the model is trained on a wide range of topics, you will get a model which gives answer from a wide range of domains without any restrictions.

In terms of the type of question to be answered, question answering can be classified into factoid questions and non-factoid questions [18], [19]. Factoid questions are about general knowledge that have a specific answer which can be answered using named entities such as dates, locations, proper nouns, other short noun phrases, or short sentences [20]. These questions aim to obtain clear, short responses from a knowledge source that are

straightforward and unambiguous. Questions that require detailed explanations, descriptions, arguments, or procedural justifications are referred to as non-factoid questions [10]. A non-factoid questions cover a wide range of questions that often do not have a single answer where the expected answer may be a sentence, a paragraph, an essay, or require some long- explanation. Unlike factoid questions, which generally seek specific information or answer.

There may be a single document or passage (technically known as a 'context') that can provide solution. However, multi-hop questions cannot be answered in a single context. The task of answering such questions is known as multi-hop question answering (MHQA). The purpose of MHQA is to predict the proper answer to a question that needs numerous reasoning 'hops' across different contexts.

By integrating natural language processing (NLP) with deep learning techniques researchers can leverage the strength of both fields, deep learning has been successful in various areas of natural language processing (NLP) such as question answering, machine translation, sentiment analysis, natural language generation from image etc. This results in researchers to use deep learning for question answering. The development of deep learning algorithms, including different types of neural networks like Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Gate Recurrent Units (GRUs), Long Short-Term Memory (LSTM), and others, has contributed to the improvement in the performance of question answering models. Researchers are also combining these algorithms in various ways to enhance their effectiveness for specific tasks. Question answering is a highly significant NLP task which offers practical uses for our daily lives across education, research, and information retrieval.

2.2. QA General Architecture

A large range of methods are covered by QA systems, including question type category, external knowledge databases, heuristics for extracting specific sorts of answers, answer generation, answer justifications, inference rules, feedback loops, and machine learning. As such, it is not practical to include every variation in one architecture. Given the

commonalities among most systems, it is possible to provide a general architecture for a question answering system. As per reference [21], Figure 2.1 illustrates the principal constituents of a generic architecture and their relationships. Question processing, Document processing, and Answer processing are the three parts of the system.

This method is referred to as a question answering pipeline. It works as follows: natural language queries go into the first module, which handles question analysis; answers come out of the last module, which extracts and prepares the answer for the user. Modules are chained such that the output of an upstream module is connected directly to the input of the next adjacent downstream module.

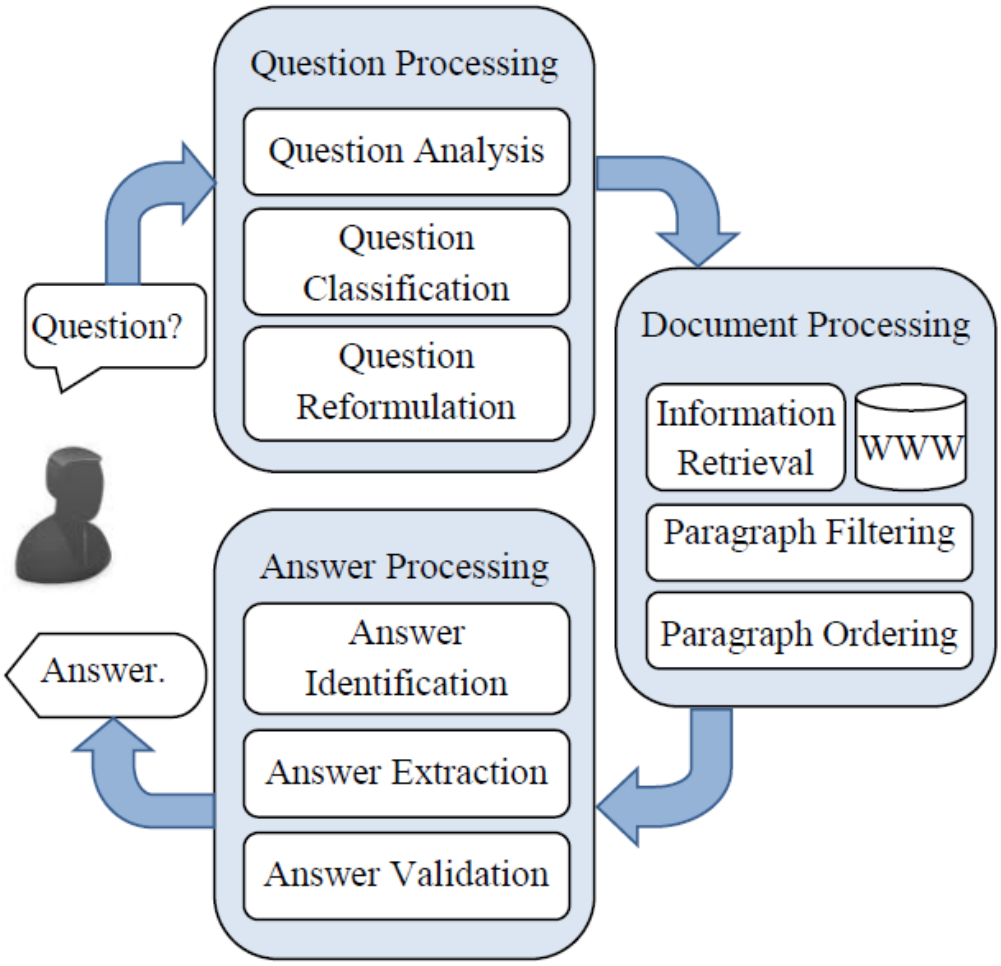

Figure 2. 1: The General Approach to QA[21]

Analyzing the query itself, which is posed by a user in plain language is the first stage. At the very least, a morphological analyzer or an analysis of the question may be part of the question analysis component. In order to identify what the question is requesting—for example, a date, a location, a person's name, a definition, a description, etc.—it is also classed. A retrieval query is created and sent to the document retrieval component based on the morphological analysis and the type of question. The response extraction component also receives some of this data, including the question class and a syntactic analysis of the question.

The document processing component finds a subset of documents from the entire document collection that include terms from a particular query. This is typically accomplished by using a traditional document retrieval system. By including those documents, the retrieval component provides a set or ranked list of documents that are thought to be the most likely to have a response to the query. After that, the answer processing component examines them in more detail.

Documents that are likely to include an answer to the initial query along with guidelines for what kinds of terms should be considered appropriate responses are fed into the answer extraction component. The question analysis component creates this specification. Many potential responses are extracted by the answer extraction component and forwarded to the answer validation component. The answer validation component uses the question analysis component's specification of the suitable type of phrase to identify the phrase that has the highest probability of being the correct answer among several phrases of that type. After being retrieved from the documents, the user is given a ranking of the candidate responses based on their likelihood of being right [22].

2.2.1. Question processing

Question Analysis

The most crucial aspect of answering questions is question analysis. This part interprets the query, determines the type of query, and generates a list of retrieval keywords. Tokenization, POS tagging, named entity recognition, natural language parsing, and other

pre-processing operations are all included in question processing. What kinds of answers are expected is also indicated by the type of question. Accurately determining the predicted answer type will aid in accurately recognizing answers at a later stage of answer extraction. The method finds sentences where answers are likely to be located with the use of definition terms and keyword identification [23].

The same question could be asked by users in several ways. Certain components of question analysis also attempt to identify the questions semantics depending on the retrieval and answer extraction strategies. By comparing the expected and probable answers, the system is assisted by the semantic context in selecting between many possible solutions. To process queries of this kind, a WordNet-based technology can be employed. When the analysis is unable to produce the predicted result, the semantic context also aids in approximating its type. Additionally, it enables the subclassification of large concepts for each specific query, and by comparing the expected and likely responses semantic contexts it aids the system in selecting the best possible response [23].

Question classification

Is the method of classifying the questions into various semantic groups the predetermined collection of possible classes includes only a few basic sets that vary based on the main question word, for example, "…ማን ነው? /ማን ናት? / ማናቸው?" The factiod_person of (who) an individual, "…መች ነው?" mean (When) factoid_date which seeks date [10]. For example, the search space of likely answers will be greatly reduced if the algorithm recognizes that the question "የፋሲለደስ አባት ስም ማን ነው?" (what is the name of Fasiledius father?) "Expects a person's name as an answer.

For the Question Answering system to function well overall question classification accuracy is crucial. Because of this the majority of systems turn to a more thorough examination of the question to find out further limitations on the answer entity. Some methods of doing this include locating keywords in the question that will be utilized to match candidate sentences that contain answers or determining syntactic or semantic relationships that should exist between a candidate answer entity and other entities or

events that are mentioned in the question. In an effort to classify the input question into the relevant category, several systems have constructed classification of question types based on the kinds of answers desired. A broad category definition is used by the majority of question-answering systems. Table 2.1 lists several question classes that correspond to several of the question types found in many Q&A systems available today.

Table 2. 1: List of question types [10]

የግለሰብ_ስም (PERSON)	የቦታ_ስም (PLACE)	ቀን (DATE)
ብዛት (NUMBER)	ርዝመት (LENGTH)	ትርጉም (DEFINITION)
ማብራሪያ (DESCRIPTION)	ዘዴ/ሂደት (METHOD/WAY)	አጎጽሮተ-ጽሑፍ (ABBREVIATION)
የገንዘብ መጠን (MONEY)	ምክንያት (REASON)	ጥቅም/ፋይዲ (PURPOSE)
መጠሪያ/ስም (KNOWNFOR)	ድርጅት (ORGANIZATION)	ሌላ ዓይነት ጥያቄ (OTHER)

Question classification can be applied in a number of ways. Using a set of rules to map question patterns into question types is the simplest approach. Regular expressions on the surface form are used to express the patterns. Analyzing the question's interrogative phrases (wh-terms) is often how the answer type is determined. For instance, considering the query “በኢትዮጵያ ታሪክ እና በአለም አቀፍ ዲፕሎማሲ ውስጥ ቁልፍ ሰው የነበሩት ንጉሱ ኢትዮጵያን የገዙት በየትኞቹ አመታት ነው?” ("In which years did the king, who was a key figure in Ethiopian history and international diplomacy, rule Ethiopia?") uses the phrase “በየትኞቹ አመታት ነው?” (when), indicating that a date is looked for. With minimal work, this method could

produce satisfactory results for broad question categories. Moreover, people can easily read the classification rules that have been established by humans. However, when the question categories are detailed or when extremely high precision is desired, the creation of such rules can be quite time-consuming. Also, when the domain of application changes, new classification rules must usually be created almost entirely from scratch [24].

Using machine learning techniques is an alternate method for classifying questions. This method treats question categorization as a standard classification task that may be performed using statistical classification programs. This approach is pretty simple after a set of features has been determined, provided that there is a corpus of questions accessible along with annotations showing the right class of each question.

In order to create approximation generalized models of language phenomena based on real examples of these phenomena provided by the text corpora without significantly adding linguistic or world information, statistical approaches frequently use huge text corpora and a variety of mathematical techniques [25]. The primary source of methods for statistical natural language processing (NLP) is machine learning, a scientific field that focuses on learning from data. In other words, to gather data, identify trends, forecast missing data using available data, or, in a broader sense, create probabilistic models from the data. Supervised and unsupervised machine learning approaches are the two machine learning strategies [24].

Predicting missing information from seen data is the primary goal of supervised learning. It uses statistical techniques to build a prediction rule from training data that has been labeled. supervised learning methods include naive Bayes, decision trees, neural networks, and support vector machines (SVMs). Unsupervised learning aims to create clusters from the data.

The classifier, or its scoring function, is taught using a set of labeled examples, often known as training data, in supervised learning. Test data is a different collection of labeled data used to assess the system's performance.

Query Reformulation

In addition to extracting the information included in the question, question-analysis also generates queries that an information retrieval system may use to extract the text that contains the answers from the complete collection of documents [24]. Answer-pattern generating and keyword selection techniques are frequently used to obtain these queries:

- The process of choosing keywords involves identifying question terms that, when found in a text suggest the presence of a potential answer in the surrounding text. Regarding the query " ከማራ ታክላ ሃይማኖት በኋላ በነገሠው በንጉሠ ነገሥቱ ዘመን የተጻፈው የታሪክ ዜና መዋዕል ስሙ ምን ይባላል?" (What is the name of the historical chronicle written during the era of the emperor who reigned after the Mara Takla religion?), the phrases "ከማራ ታክላ ሃይማኖት በኋላ በነገሠው በንጉሠ ነገሥቱ ዘመን የተጻፈው የታሪክ ዜና መዋዕል" (The Chronicles of History written during the reign of the emperor who reigned after the Mara Takla religion), "የታሪክ ዜና መዋዕል" (Chronicle of history), and “"ከማራ ታክላ ሃይማኖት” (From Mara Takla religion). Then, by utilizing synonyms and/or morphological variants, this list of keywords may be enlarged.
- In order to provide questions that express potential forms in which the answer could be discovered, answer pattern generation aims to create queries that comprise one or more combinations of question terms.

Query reformulation modifies the original query to more closely match relevant documents in an effort to increase search effectiveness. While previous research has explored more complex techniques, traditional approaches concentrate on producing related words and sentences. In order to capture significant query-level dependencies, [26] suggest modelling reformulation as a distribution of real queries.

2.2.2. Document processing

Information retrieval (Document retrieval)

Document retrieval is a feature of quality assurance systems that helps to focus the search for a solution. Its primary goal is to choose the initial collection of candidate documents

containing answers from a huge text corpus before forwarding them to a subsequent answer extraction module.

In order to obtain documents that contain the expected response to a user's question, question answering systems employ document retrieval [12]. This suggests that retrieval systems are the main source of dependency for question-answering systems. A retrieval subsystem is typically included in question answering systems to aid in the identification of documents or passages that may include answers to common questions. The next part of the question answering system, can use the huge number of ranked documents that the document retrieval component delivers. Certain Q&A systems employ the retrieval system to obtain relevant documents, which are then utilized by the passage retrieval subsystem.

Paragraph Filtering

The information retrieval system may return a very high number of documents, as previously described. The amount of candidate text in each document can be decreased as well as the total number of candidate documents by using paragraph filtering. Based on the idea that most relevant texts should have the question keywords concentrated in a few adjacent paragraphs rather than distributed across the text, paragraph filtering is a notion. Consequently, the document is rejected from further processing if none of the keywords are detected in any of the N consecutive paragraphs that make up that set of paragraphs [21].

The remaining paragraph are then ranked based on how likely they are to have a solution to the question. Consequently, the named entity or answer type categorization will be applied to the retrieved paragraph as the initial step. The type of response that the query yielded will tell us what kind of response we might expected. Next, a limited number of criteria are used to rate the remaining paragraph, including the amount of named entities of the correct kind in the passage, the quantity of query key words, and the rank of the document from which the passage is extracted [27].

Paragraph ordering

Sorting the paragraphs into order based on how likely it is that they contain the right answer is the goal of paragraph ordering. The radix sort algorithm is the standard method used for arranging paragraphs [21]. Three different scores are used in the radix sort to arrange paragraphs: Same word sequence score, Distance score, and Missing keyword score.

2.2.3. Answer Extraction

Answer Extraction is the last component of question answering, which is responsible for recognizing, extracting and validating responses from the collection of ordered paragraphs provided to it. The process of extracting answers involves examining the candidate passages, which contain a wealth of data such as named entity tags and point of sale information. The identified thing in the candidate passages will be selected as a potential answer if it matches an expected answer type. An answer ranking technique based on the heuristics method is used to determine the best candidate answer when there are several candidate answers [22]. In accordance with the response extraction patterns specific to each question type, the answer extraction component for non-factoid questions additionally extracts answers from a paragraph of retrieved documents [28]. This part looks for the language patterns associated with each question type which were previously discussed for every passage.

Next question answering systems are presented with a series of text fragments that could be accurate replies after classifying questions, retrieving documents, and extracting answers. These candidates often come from various passages and are frequently extracted through various techniques. Furthermore, some passages of documents might not always provide complete solutions. Answer extraction may yield the following set of answer candidates [29]: Whole response: a text section that completely meets the user's information needs. An answer that is too detailed a passage of text that provides less detail than what was expected and only partially or not at all satisfies the user's information demand. An excessively general response is a text section that is more detailed than the expected response. A text part that provides some of the information needed to partially

address the question is called a partial answer. To create a complete response, partial responses from several texts or paragraphs must be merged.

The work of creating accurate, comprehensive solutions with matching confidence scores from the result of answer extraction, or candidate answer components, is known as answer generation. In order to produce meaningful, accurate responses with high confidence scores, this task requires synthesizing information from multiple answer candidate components [29].

There are several approaches to choose or rank potential responses, much like the other components. Once a text unit with an expected answer type has been identified, more limitations are placed on the unit. Essentially, an answer section would be created around a candidate answer holding paragraph once an expression of the correct answer type was found in it. A weighted numerical heuristic could then be used to compute an overall score for the section based on various quantitative features like word overlap between the question and the answer section. In order to calculate an overall rating for all answer candidates, a score is obtained for the answer-section containing the answer candidate for each candidate answer holding paragraph that contains an expression of the correct response type [22]. Apart from word-based comparison methods, another criterion for answering choices is a candidate answer's frequency. The number of times a candidate answer was connected to the question is its frequency. Redundancy based answer selection is another term for choosing a response based on frequency. It is possible to limit the counting of these frequencies to the documents that were taken into account in the response extraction component, but it is also possible to extend it to a broader collection. Certain question answering systems calculate the number of times a candidate response appears alongside terms from the query by looking through the entire collection of documents. Some systems even go one step farther and obtain these frequencies via the World Wide Web, bypassing the actual document collection process [22].

2.3. Approaches to Question Answering

QA system approaches are categorized into groups according to the techniques employed. Certain QA systems fall within the category of basic information retrieval and natural language processing, whilst other QA systems rely on using natural language reasoning [30]. The previous one employs free text documents as a data source; the questions are primarily WH-type and use the current IR assessment techniques. It uses NLP techniques including syntax processing, NER, and information retrieval techniques to achieve the goals of a question answering system. The later approach, on the other hand, employs higher order reasoning procedures; knowledge bases are utilized as a source of data, mostly domain-oriented questions may go beyond the WH-type and assessment techniques are not incorporated.

2.3.1. Rule Based Approach

This method employs a broad range of natural language processing (NLP) techniques to maximize response retrieval accuracy. These systems use a semantic model to first provide training and test data. Certain systems of this kind produce rule for every kind of question, including questions of the who, when, where, and why semantic classes. "Who" rules search for names that belong to the noun class. "When" rules only include time expressions. Whereas most "where" rules search for locations, Amharic may have distinct requirements. These systems are mostly employed for reading comprehension questions and must learn rules from training data. Heuristic methods that search for lexical and sematic clues are employed by a system named Quarc to determine the question class [31].

2.3.2. Pattern Matching Approach

Rather than using the sophisticated processing found in other competing systems, this approach uses text patterns. Many question-answering systems do not use linguistic expertise or techniques like named entity instead, they automatically learn text patterns From passages. WordNet, ontologies, and so on for information retrieval. Take the query, " በማራ ታክላ ሃይማኖት የተተካው ንጉሠ ነገሥት ሚስት ማን ነበረች?" (Who was the wife of the emperor who was succeeded by the Mara Takla religion?) for instance it will search for the "who was of?" pattern. and the response would be something like " የትግሬ ተወላጅ የነበረችው የንጉሡ

ሚስት ንግሥት ብሌን ሳባ ነበረች።" (The king's wife was Queen Blain Saba, who was a native of Tigre) While some pattern matching quality assurance systems generate responses using templates, the majority of them use surface text patterns.

2.3.3. Surface Text Pattern

A template-based method uses questions with preformatted patterns. This method places more emphasis on illustration than on interpreting the questions and responses. Each member of the template set represents a variety of questions of a different kind, and the set is constructed to have the most templates possible while yet sufficiently covering the issue area. Templates contain entity slots, which are empty spaces pertaining to the question's notion that must be filled in order to create the query template and obtain the relevant database response. The response produced by query would be raw data, which is returned to the user [32]. Because human intelligence is used to identify the key terms for obtaining replies, template-matching precision of document retrieval is high [33].

2.3.4. Machine Learning Approach

One method of classifying questions in a question-answering system is machine learning. It has the ability to automatically create a highly effective question classification software that can be applied to more question features [34]. A learnt classification program's performance usually gets better with additional training data. Certain machine learning classification tasks, including question type identification, can be viewed as traditional classification tasks that can be resolved by various machine learning methods. The machine learning approach has the advantage of being more maintainable, flexible, and adaptable [35]. This research uses this method because it eliminates the need to manually create rules that must be maintained and adjusted to new developments on a regular basis. One example of a machine learning-based system is IBM Watson, which classified questions and answers using N-gram features and the Maximum Entropy model [32].

2.3.5. Deep Learning Approach

Deep learning can use neural networks to extract meaning from data, which sets it apart from machine learning. A neural network (NN) is often composed of several neurons,

which are simple, interconnected units that produce a sequence of real-valued activations. Input neurons are stimulated by sensors that pick-up information about their surroundings, while other neurons are stimulated by weighted connections from previously activated neurons. Certain neurons may be able to change their environment by initiating actions. Every input has a weight that represents its relative importance to the other inputs.

Recent research has demonstrated the effectiveness of deep learning models for a range of natural language processing tasks, such as text summarization, machine translation, and semantic analysis. Neural network topologies convert textual context into logical representations that are then used to predict responses. According to the research [36], deep learning techniques have a lot of advantages for natural language processing among them Deep learning techniques can take the place of current, successful natural language systems, Deep learning techniques automatically identify the features needed by the model using natural language or images, increasing the likelihood that new approaches to a range of natural language problems will be adopted. The deep learning model's performance is grounded in empirical research on natural language processing. Large end-to-end deep learning models can be utilized to increase performance in a range of natural language applications, and the improvement seems to be ongoing and even increasing.

2.4. Answering Factoid Questions

Factoid questions typically have brief responses that need for precise information like the place, name, and time. IBM Watson was the first to demonstrate that factual question answering could be accomplished using supervised machine learning. Factoids typically include some information regarding the nature of the answers they ask. Short text passages can be used to derive the one right response to a factual question. Compared to the other categories, factoid inquiries are significantly easier to answer. The response ranking is a crucial aspect of answering factual questions. A quality assurance system should prioritize the right response to a query at the top of the list of possible answers.

2.5. Neural network approaches

2.5.1. Recurrent Neural Network (RNN)

RNN is a neural network method for sequential data that values context because its output depends on both the previous and current inputs. Based on the hidden state, RNN determines what to do with one or more inputs and returns one or more outputs. The architecture of RNN includes a loop that functions as internal memory to help record and remember the previous output to act as input with the current state. As a result, different outputs may be produced from the same input because of the hidden state. RNNs are said to have two inputs: the recent past and the present. RNN is being used in various fields these days, including question answering, machine translation, and natural language processing (NLP). Thus, this achievement results in its application in study on question answering.

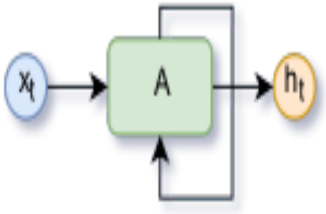

Figure 2. 2: Rolled-up RNN*[37]*

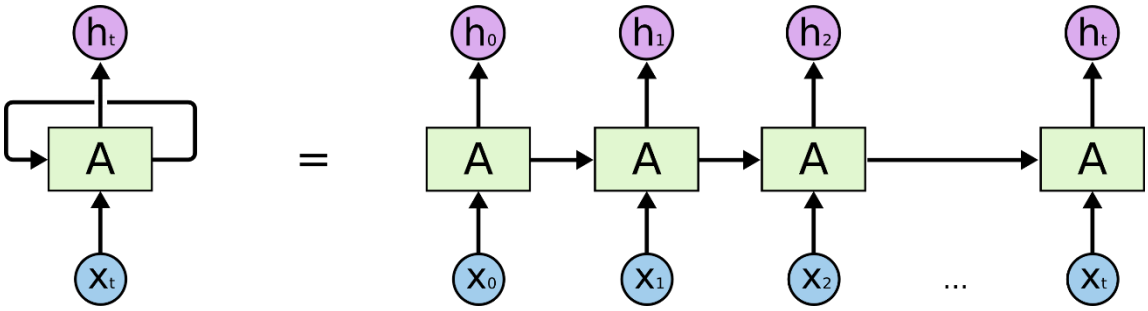

Figure 2. 3: An unrolled recurrent neural network.*[37]*

Figure 2.2 illustrates the use of RNN for a simple loop process that loops through A state (RNN cell) and transfers data from one state to another by accepting inputs xt and ht.

Recursive neural networks (RNNs) operate by processing input and output in successive steps until the network reaches its threshold value, as seen in Figure 2.2.

The structure of all recurrent neural networks is a series of successive neural network modules. This repeating module in conventional RNNs is made up of a single tanh layer, as seen in Figure 2.4 below.

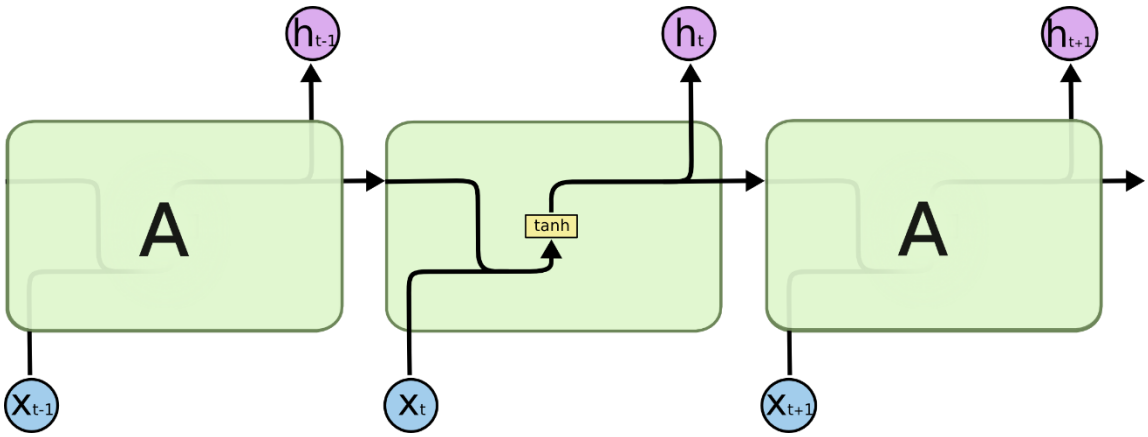

Figure 2. 4: The repeating module in a standard RNN contains a single layer*[37]*

2.5.2. Long Short-Term Memory (LSTM)

The word "LSTM" refers to Long Short-Term Memory networks, which are a unique type of RNN that can recognize long-term dependencies. The vanishing problem that RNN has makes it unable to learn long-term dependencies; LSTM is designed to address this issue. The structure of an LSTM is similar to that of an RNN; however, it differs in the repeating module. Four neural network layers, each interacting in a unique way, are present rather than a single one [37].

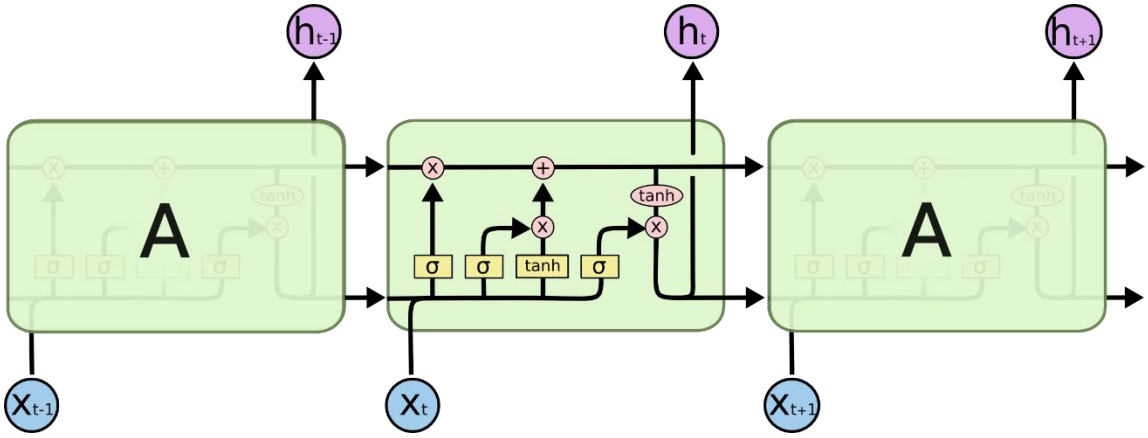

Figure 2. 5: The repeating module in an LSTM contains four interacting layers*[37]*

Forget gate

The LSTM network's first gate makes this decision on whether or not to retain information. Information from the current output (Xt) and information from the prior hidden state (ht-1) are so combined and pass through the first activation function before the forget gate (ft) is created. The value of those data lies between 0 and 1 when they pass through the SoftMax activation; values closer to 0 indicate that the information is irrelevant to form the next hidden state and is therefore forgotten, while values closer to 1 indicate that the information is relevant to form the next hidden state and is therefore retained.

Input gate

Using the previous hidden state and the current input beforehand, the next gate in an LSTM updates the cell state. The tanh activation function is then used to control value pass through the network by squishing the value between -1 and 1. Therefore, in order to regulate the network, the previous hidden state and the current input pass through the Tanh activation first, followed by the SoftMax activation where the information kept for updating is decided. Finally, the output of the Tanh activation is multiplied by the output

of the SoftMax activation. Which tanh function data must be updated and maintained is determined by the SoftMax output.

Cell state

The network's memory is where pertinent and durable data is kept. In order to provide the option of deleting certain values from the cell state in the event that it is most likely multiplied by zero, the prior cell state and the forget vector combine to construct a pointwise multiplication. The cell state is then updated to new values that the neural network deems significant via pointwise addition following the input gate's output.

Output gate

The final gate of the LSTM is responsible for determining the next hidden state. Predictions are made using the knowledge from the previous hidden state in the subsequent hidden state. As a result, the updated cell state passes via the Tanh activation function after the prior hidden state and the current output pass through the SoftMax activation. Subsequently, the SoftMax and tanh outputs undergo pointwise multiplication to determine the information contained in the new hidden state. A new hidden state and a new cell state are the output gate's ultimate products, and they are carried over to the following time step of the LSTM sequence.

2.5.3. Bidirectional Long Short Term Memory Networks (BiLSTM)

Sequential knowledge is the center of bidirectional long short-term memory, a kind of recurrent neural network. Long Short-Term Memory models employ information from neurons prior states [12]. The output of each LSTM is combined using their total, and the data processing proceeds back and forth at the same time. Additionally, bidirectional LSTM (Bi-LSTM) can extract more contextual data at once. Since these models classify sequential input into several groups and represent a state-of-the-art result, they are more successful in resolving the voice recognition problem. Given that our Amharic Question classification is also a multi-class classification, the Bi-LSTM model does good on that too. The entire architecture of the BiLSTM is shown in Figure 2.6.

The increasing and disappearing gradient issues that afflict conventional RNNs are resolved by the gated recurrent neural network architecture known as the LSTM architecture. Unlike feedforward neural networks, it has cyclic connections, which helps with sequence simulation. Two unidirectional LSTMs were reversed to generate a bidirectional LSTM (BilSTM). Bi-LSTM is capable of performing a multi-class classification procedure. Additionally, it has two LSTM forward and backward layers that improve memory capacity and make use of extra information.

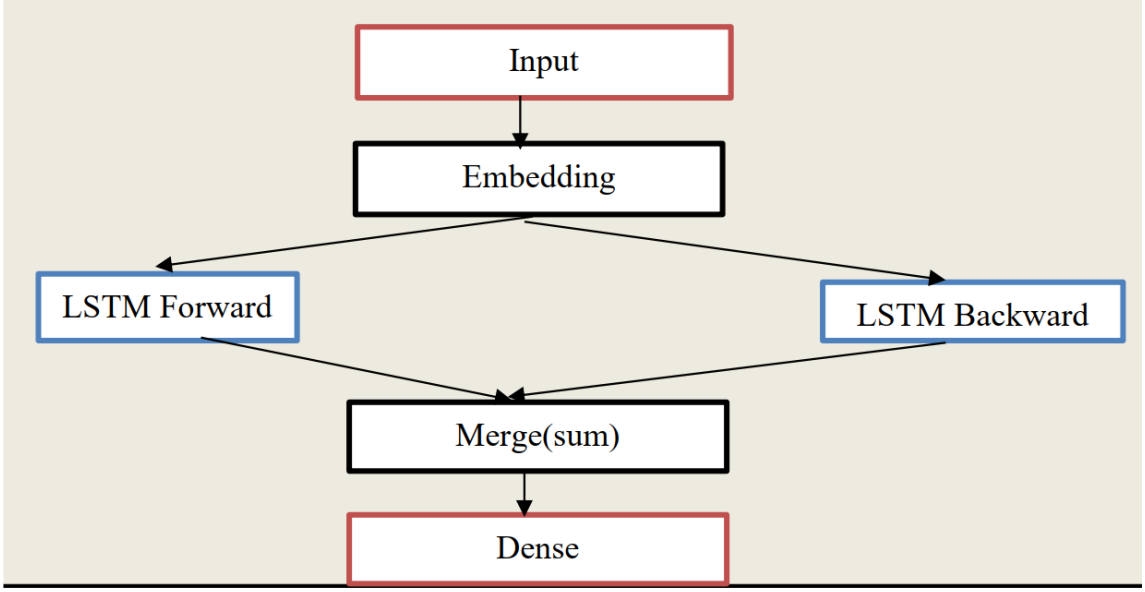

Figure 2. 6:Bi-LSTM Architecture[12]

2.5.4. Convolutional Neural Network

Convolutional neural networks, also known as multi-layered artificial neural networks, are capable of extracting features from text, image, and other complex data types. CNNs have shown useful in computer vision tasks like classifying, detecting objects in images, and segmenting images. On the other hand, CNNs have lately been applied to text-related problems.

A layer of neurons in convolutional neural networks (CNNs) executes the convolution process; CNNs are a sort of forwarding neural network [38]. Neurons use a convolutional kernel, sometimes referred to as a filter, of a particular scale in response to the input of

activations from neighboring neurons. Deep, feed forward artificial neural networks, or CNNs, are effectively used to analyze visual imagery. Recent research has demonstrated the effectiveness of CNN models in the following areas: semantic parsing, sentence modeling, part-of-speech tagging, name object recognition, sentence modeling, and search query retrieval [38].

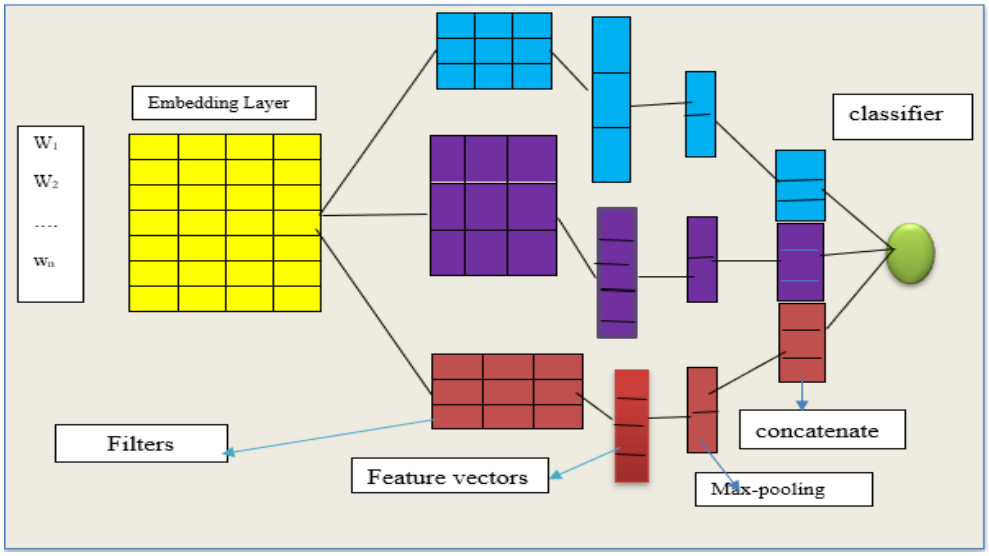

Figure 2. 7: Convolutional Neural Network[39]

Convolutional, pooling, and fully connected layers are the three types of layers that make up CNN, as seen in figure 2.7. Each convolutional layer contains several convolution kernels, which are used to construct feature maps from the preceding layer. Different feature representations of the initial inputs would have been learned by the convolution method and transmitted to further layers. A region of nearby neurons that is assigned to a single neuron on the following layer is known as the receptive field of a single neuron. A feature map is created by convolving the input through a trainable kernel and applying the output using an element wise nonlinear activation function. Pooling layers help prevent over-fitting, minimize the number of outputs, and lower computing complexity. Normally, the sampling layers are applied immediately behind the convolution layers since applying them creates redundant data. As they process each of the different inputs, the kernels alter [38].

2.6. Hyperparameter

Since deep learning produces state-of-the-art results, it has become popular in natural language processing and applied for numerous applications such as text classification, named entity recognition, and part of speech tagging. However, it is not a simple task. It requires careful hyperparameter selection and optimization [40]. The word embedding, learning rate, dropout rate, activation function, and optimizer are a few examples of hyperparameters.

2.6.1. Activation function

The network can employ the activation functions to suppress irrelevant data points and exploit the crucial information. In contrast, the activation function at the end chooses what should be fired to the subsequent neuron in a neuron-based model that is modeled like our brains. [39] Two types of network architectures and traditional activation can be used to categorize activation functions. Tanh and Sigmoid activation functions as well as network architectural ReLU, SELU, ELU, ISRLU, and GELU are included in the conventional approach [39].

Tanh functions were used to map the product of inputs and trainable parameters to a range from -1 to 1. It highlights a possible issue known as vanishing gradients, which arises when the gradient is either excessively small or large, or a saturated issue. The issue of diminishing gradient approaches to 0 and 1 affects Tanh Sigmoid in the same way. This issue prevents trainable parameters from being changed, which stops learning.

On the other hand, ReLU may encounter a dead state issue if it is stuck on the left side of zero. An issue that impacts a large number of nodes in a network reduces the output of the network. In [39], neurons with ReLU have the same gradient sign, meaning that all of the layer's weights either grow or drop. ELU was created to address dead states and disappearing gradients. In positive functions, the same as ReLU; in negative functions, different. ReLU responded to 0 in negative functions, but ELU generates negative values even in the absence of issues when weights are changed only in one direction. Before data is sent into the neural network, batch normalization is the purpose of SELU. Batch dropout

regularization was resolved by GELU instead of the activation function mentioned above. ELU was the location of the exponent function on the left side of the zero extremely cost computation problem. ISRLU was created as a solution to this issue and to expedite learning; however, it requires an alternative experimental configuration.

2.6.2. Optimizers

Optimizers are algorithms that are used to decrease the loss function and improve the model's accuracy in order to modify the parameters of the created model [41]. Hyperparameter optimization is the process of identifying all the exact model configuration arguments that lead to the best possible model performance on a certain dataset. One characteristic that is utilized to track and manage the learning process is the value of a hyperparameter. One of the most used optimization methods and the most used technique for neural network optimization is gradient descent [41]. It seeks to limit loss or expense while adjusting weight.

The second optimizer, called stochastic gradient descent, aims to run a training epoch for each dataset and updates the model's parameters more often. Its advantage is that it uses less memory because it does not need to keep the values of the loss functions. Its drawbacks were a high variance in the model parameters and the requirement to gradually lower the learning rate value in order to achieve the same convergence as gradient descent [41].

The third class of optimizers is called adaptive gradient descent. This algorithm for gradient-based optimization uses dynamic learning rates for each parameter instead of the fixed learning rate approach. Using parameter-specific learning rates, they are adjusted to the training frequency of a parameter. This suggests that bigger updates are made for parameters that are used infrequently and smaller changes are made for parameters that are used frequently. The term "Adaptive delta" (Adadelta) refers to the advancement from Adagrad. Rather than accumulating all previous gradients, this robust modification of adagrad modifies the learning rate based on a gradient update [41]. The difference between

the weights as of right now and those that have been modified is referred to as delta in Adadelta.

Adam and Nesterov are combined to create another sort of objective function called Nesterov Accelerated Adaptive Moment Estimation (Nadam) [41]. Using Nesterov, Nadam updates the gradient one step ahead of time by substituting the current ˆm for the prior ϕm of the Adam optimizer. By adding up the exponential decay of the moving averages for the prior and present gradients, the learning process is speeded up [41].

2.6.3. Dropout

When a network performs properly on trained data but fails to function on unknown data during testing, it is said to be over-fitting. The dropout strategy is an appropriate answer to the over-fitting issue in deep neural networks. During training, units and their connections are dropped at random to implement dropout. In order to enhance generalization error and lessen over-fitting, dropout provides an efficient regularization technique. However, one drawback is that it requires more iterations to converge. [42]

2.6.4. Word Embedding

TF, TF-IDF, and BOW are the most widely used weighted word feature extraction techniques. In these methods, each word is assigned a number that indicates how many times it appears in the overall document or corpus. Every document is converted into a vector that is the same length as the text and contains all weighted word approaches' word frequencies. Despite being intuitive, the approach is constrained by the possibility that particular phrases that are often used in the language could guide these representations [43].

Words with comparable meanings are represented similarly in a learned text representation called a word embedding. One of the major advances in deep learning for difficult natural language processing problems may be this method of expressing words and documents. The core of natural language processing is word embedding, which

captures syntactic information for words as well as semantics and semantic similarity between words by representing the words in a text in a R dimensional vector space [44]. Currently available word embedding algorithms that may turn words into meaningful vectors are Word2Vec, FastText, and Global Vectors (GloVe). Each dimension of the vectors generated from the model captures a distinct characteristic of words. These techniques are able to capture the semantic and syntactic relationships between words. Word embedding has become a regular method for QA task.

Static word embedding

Every word in static word embedding has a single, distinct vector representation without considering its context. For instance, the bank has the same vector representation for material relating to rivers as it does for financial matters. Glove, FastText, and Word2vec are a few static word embedding models.

a) Word2vec

Word2vec is a tool that uses a fixed length vector to represent each word and can convey the semantic relationship between words. Word2vec offers two methods: skip-gram and continuous bag of words (CBOW). While continuous bag of words (CBOW) functions in reverse to produce the center word from the given context, skip-grams generate the context word from the given center word [45].

Continuous skip-gram model

The continuous skip-gram model takes a word, Wi, and predicts the context words (Wi-2, Wi-1, Wi+1, and Wi+2) that surround it. It is not necessary for context terms to be immediate. Within a specific window size, some words can be skipped to look both forward and backward from the target word. The single hidden multilayer neural network of the Skip Gram model is depicted in Figure 2.8.

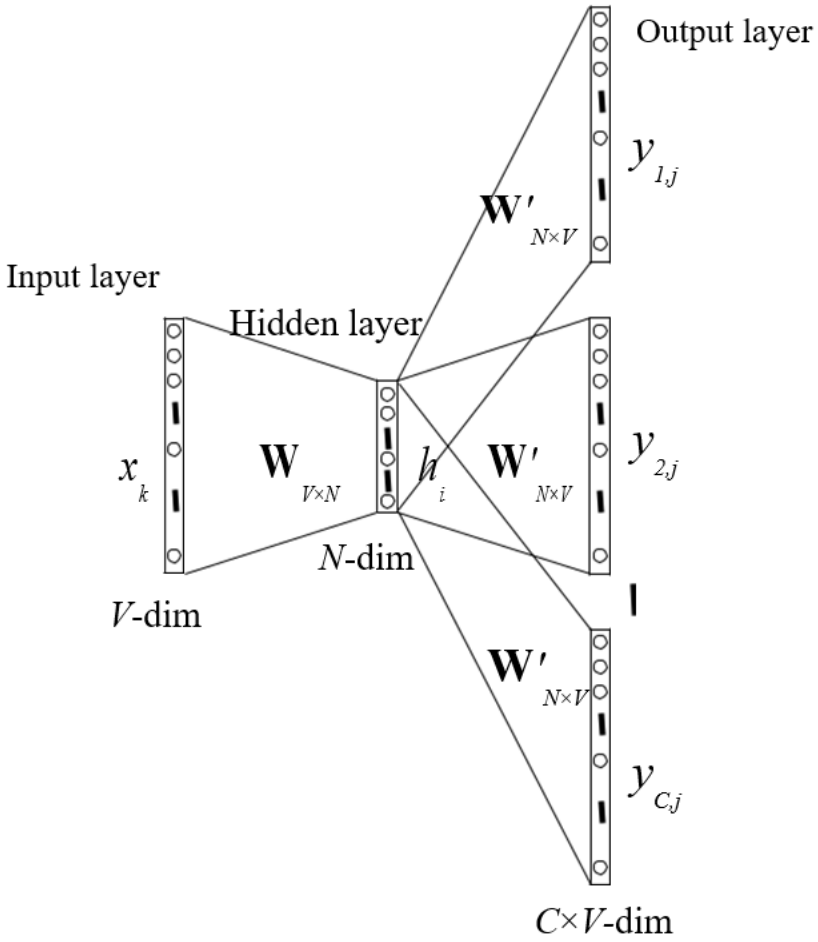

Figure 2. 8:The skip-gram model*[12]*

Continuous bag of words (CBOW)

is reverse of skip gram model. Given the context (Wi- 2, Wi-1, Wi+1, Wi+2) the task is to predict the word. The continuous bag of words (CBOW) is depicted in Figure 2.9.

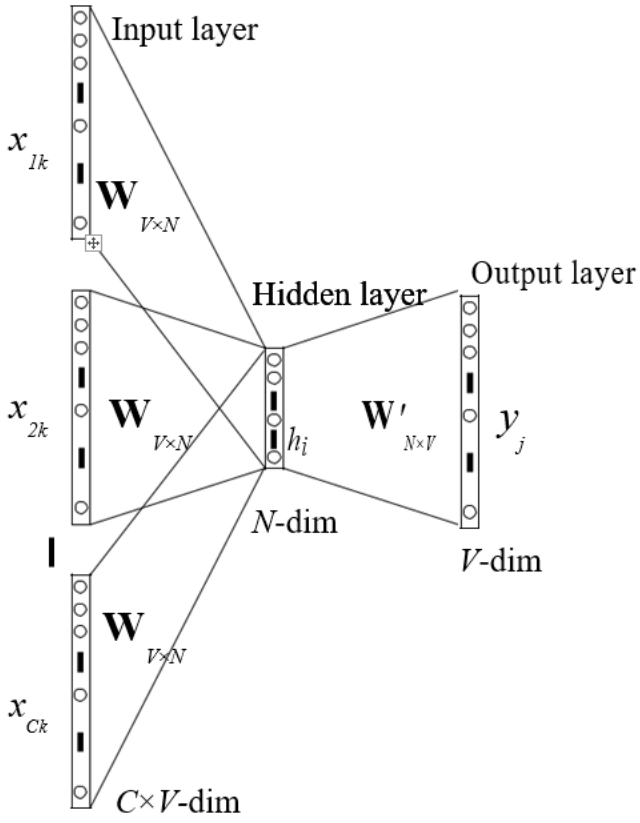

Figure 2. 9: Continuous bag-of-word model*[12]*

b) FastText

Rare and non-vocabulary words are difficult for Word2vec to handle; nevertheless, FastText can locate word embeddings for these types of words. Word2vec and FastText are comparable, however FastText uses characters to express word embeddings. By utilizing sub-word-based modeling, FastText is able to examine a word's underlying structure or morphological characteristics in order to express it as a vector. Thus, the total of the vector representations of the word's subwords constitutes the vector representation of the word itself.

c) Glove

Glove is useful for collecting language features globally by studying word co-occurrence across corpora, since it can learn the efficient representation of words using word to word co-occurrence data from a corpus [46].

Contextual word embedding

In contrast to statistical word embedding, contextual word embedding assigns a unique vector representation to each word according to its context. For instance, the vector representation of a bank varies depending on whether the phrase is related to rivers or finances.

a) **Embeddings from Language Models (ELMO)** A contextual word embedding technique called Embeddings from Language Models (ELMO) is used to represent words in vectors. There are two layers used. Two Bi-LSTM layers make up the other layer, which is the character encoding layer. The word embedding is formed by the character encoding layer, which encodes a word's character sequence into its word representation before passing it on to the following two Bi-LSTM layers [46].

b) **Bidirectional Encoder Representations from Transformers (BERT)** predicts words that are randomly masked in a sequence using a masked language model, which is a transformer-based language representation model. Furthermore, it is employed in sentence association and next sentence prediction [46].

2.7. Amharic Language

With over 100 million speakers within the nation, Amharic is the most frequently spoken Semitic language family globally, after Arabic. It is the official language of the government of the Federal Democratic Republic of Ethiopia. In contrast to Arabic and Hebrew, it is written from left to right in its witting script, Fidel, which is derived from the Geez alphabet and consists of 44 letters (275 characters) and 33 consonant letters with seven vowels for each consonant.

The word categories in Amharic are ስም (noun), ግስ (verb), ቅፅል (adjective), ተውሳከግስ (adverb), መስተዋድድ (preposition), and ተውላጠስም (pronoun) [20].

(ስም) Nouns: prefixes and suffixes, primarily suffixes, are used in Amharic to indicate gender, number, definiteness, case, and direct object status. Nouns in Amharic can be either masculine or feminine. To indicate the gender of a noun, suffixes are applied. While some nouns have only one gender, others might have both masculine and feminine genders. Both femininity and small things are denoted by the feminine gender. As an instance, ቤቱ ትንሽ ነች፡፡ Whether a word ends in a vowel or a consonant, the suffix ዎች or አች is added to make plurals.

ግስ (Verb): are words that have person, number, gender, mood, voice, and polarity inflected into them through roots and affixes. Adjectives agree with their subjects. Verb agreement is not required with objects. In Amharic, verbs are typically found at the end of sentences. for example, አበበ በሶ በላ፡፡

ተውሳከግስ (Adverbs): can be used to qualify verbs in sentences by providing additional meaning. There are only a few Amharic adverbs, such as ትላንት፣ ዛሬ፣ ገና፣ ቶል፣ እንደገና …

ቅፅል (Adjective): any word that follows an adverb or modifies one in order to become a noun ንበዝ ተማሪ፣ በጣም ንበዝ. Adjectives also have the unique characteristic of repeating the previous letter of the word when it is pluralized (e.g., ትንሽ፡ ትንንሽ).

መስተዋድድ (A preposition) ፡ is a word that functions as an adverbial operator before a noun, indicating place, time, cause, and so forth. It is never used to build a new word and is incapable of accepting any suffix or prefix. Included are ስለ፣ ወደ፣ እንደ፣ and ከ….

Pronoun: this category can also be separated into semantic specifiers, such as እሱ፣ እኔ፣ አንተ፣ አንቺ፣ እነሱ…; quantitative specifiers, such as ጥቂት፣ አንዳንድ፣ አንድ… and possession specifiers, such as የአንተ፣ የኔ፣ የሱ…..[11].

2.7.1. Amharic punctuation

Amharic punctuation marks are similar to those used in English but have distinct characteristics. Amharic contains twelve punctuations with distinct meanings within the language structure. Table 2.2 lists the key punctuation marks used in Amharic with their respective functions.

Table 2. 2 Amharic punctuations adopted from [47]

Punctuation mark	Symbol	Purpose	Punctuation mark	Symbol	Purpose
Four dots or double colon	::	Mark end of sentence	Semi-colon	፤	Used like semi-colon
Colon	:	Separate words in a sentence: not common	Three dots	…	For deliberate omission of words, phrases or sentences
White space		Separate words in a sentence: current practice	Quotation mark	<<>>	Used at the beginning and at the end of quoted words, phrase, etc.
Question mark	?	Placed at the end of question sentence	Parenthesis	()	To enclose elaboration
Exclamation mark	!	Used at the end of sentence to show exclamation	Stroke	/	Separate date, month, year, etc.
Comma	፣	Used like comma	Mocking mark	፧	Placed at the end of mocking sentence.

2.7.2. Amharic numbers

In Amharic, numbers can be expressed with Arabic symbols. It uses Ethiopic number representations. Numbers can also be expressed alphabetically in words. Table 2.3 displays the Arabic, Amharic, and alphanumeric representations of numerals.

Table 2. 3 Amharic numbers adopted from [20]

Arabic	Ethiopic	Amharic	Arabic	Ethiopic	Amharic
1	፩	አንድ	20	፳	ሃያ
2	፪	ሁለት	30	፴	ሰላሳ
3	፫	ሦስት	40	፵	አርባ
4	፬	አራት	50	፶	አምሳ / ሀምሳ
5	፭	አምስት	60	፷	ስልሳ / ስድሳ
6	፮	ስድስት	70	፸	ሰባ
7	፯	ሰባት	80	፹	ሰማኒያ
8	፰	ስምንት	90	፺	ዘጠና
9	፱	ዘጠኝ	100	፻	መቶ
10	፲	አስር	1000	፼	ሺ / ሽህ

In Amharic, fractions and ordinals are represented differently [20]. Table 2.4 displays fractional and ordinal representations in Amharic.

Table 2. 4 fractional and ordinal representations adopted from [20]

Fraction	Amharic Representation	Ordinal	Amharic Representation
½	ግማሽ	1st	አንደኛ / ቀዳማዊ
1/3	ሲሶ	2nd	ሁለተኛ / ዳግማዊ
¼	ሩብ / እርቦ	3rd	ሦስተኛ / ሳልስ
2/3	ሁለት ሲሶ / ሁለት ሦስተኛ	4th	አራተኛ / ራብዕ
¾	ሶስት-አራተኛ	9th	ዘጠንኛ / ዘጠነኛ
1/10	አስራት	10th	አስረኛ
2 ×	እጥፍ	--	--
2.x	ሁለት ነጥብ x	--	--

Dates in Amharic can be written in several ways. Dates can be written using Arabic numbers (e.g., 12/01/2001), Ethiopic numerals, or alphanumeric characters [20].

2.7.3. Amharic sentence structure

Amharic has a different basic sentence structure than English, which is Subject-Verb-Object (SVO). In Amharic, a sentence structure is either Subject-Object-Verb (SOV) “በረከት መኪና ገዛ። “/ “Bereket mekina geza”/ ” Bereket bought a car.” (where በረከት/Bereket is subject, መኪና/ Car is an object and ገዛ/ bought is verb) and its English form is “Bereket bought a car" (where “Bereket is subject, bought is a verb and Car is object). Or a straightforward Subject + Verb arrangement the subject and verb of the sentence must agree in terms of person, gender, and number for the Amharic sentence to be said to have subject-verb agreement. For instance, the sentence "ከበደ መኪና ገዛች።" is grammatically incorrect since the subject and verb do not agree on gender, with "ከበደ"/Kebed /subject/ being masculine and "ገዛች" /bought /verb/ being feminine.

Other OSV formats exist, but they are less popular for example, in the sentence “መኪናውን እኮ ከበደ ገዛው።”/ “Kebede bought the car” OSV word order is used and in this case the object is suffixed by object marker “ን/n”.

2.7.4. Amharic Interrogative Sentence Structure

A sentence that asked a question and concludes with a question mark is known as an interrogative sentence. Languages vary in how they form interrogative sentences. Regarding the Amharic language, we can inquire about the sentence's action, specifier, complement, and subject. Typical integrative words in integrative sentences in Amharic are: ስንት, ስንቱ, ከስንት, በስንት, ማን , ማን , ማነው, እነማን , ማን ማን , ማንን , ማና ማን , እነማንን , የማን , ከማንኛው, ምን ,ምንድን , የምን , ምኑን , ስለምን , ምን ,ማን ኛው, ማን ኛይቱ, ማን ኛዋ, ማን ኛቸው, በ ማን , ተናገር , ጥቀስ , ግለፅ , ዘርዝር , ጥራ, የት, የ ቱ, በየ ት, የ ቷ, የ ቲቱ, የ ቶቹ, ወዴት, ከየትኛው, የየት, የትኛው, የ ትኛዋ, ከየት, እስከየት, የትኛው,በምን , እስከምን , ከምን, ምን ጊዜ, ***and*** መቸ, በ መቼ, እስከመች.[39]

2.8. Related Works

TETEYEQ (ተጠየቅ): AMHARIC QUESTION ANSWERING SYSTEM FOR FACTOID QUESTIONS (By: Seid Muhie Yimam)

[20] attempted with the first factual question answering system in Amharic. The data was compiled from 15,000 news articles from various publications. To ensure that the format of the gathered documents was correct, some preprocessing was done. The main preprocessing methods utilized in the study were stemming, stop word removal, tokenization, character normalization, number normalization, and gazetteer preparations. After the documents were preprocessed, they were index using the Lucene indexer. The next phase, which contributes to the creation of a well-structured question, is processing the user inquiry. After receiving user-generated questions, the system uses a rule-based question type identification process to ascertain the nature of the inquiry and the anticipated response. The system was assessed using precision, recall, and MRR. Following preprocessing, the pages were indexed using the Lucene indexer. Processing the user inquiry is the next stage, which aids in producing a query that is well-structured. The system receives inquiries from users, and it uses a rule-based question type identification process to figure out what kind of inquiry the user is asking and what kind of response is anticipated. Precision, recall, and MRR were used by the researcher to assess the system. According to the rule-based question categorization module, around 89% of the questions are correctly classified. The researcher tried a number of answer selection techniques before determining that the gazetteer-based solution, which employed a paragraph response selection strategy, was the most accurate, answering 72% of the questions. The study claims that improving performance can benefit from the development of NER, POS tagger, stemmer, parser, and Amharic spell checker.

Amharic Question Answering for Definitional, Biographical and Description Questions (By: Tilahun Abedissa)

It has been suggested by [10] that Amharic be used to respond to inquiries pertaining to definition, biography, and description. Preprocessing of documents, question and document analysis, and response extraction are all included in the system. The questions are categorized using both machine learning and rule-based methods. The method used by

the document analysis component gathers pertinent documents and applies filtering patterns to definition and description questions; for biography questions, a retrieved document is retained only if it includes every phrase in the target in the same order as the query. The answer extraction part applies a type-by-type technique. Using response extraction patterns that have been manually created, the definition-description answer extractor extracts patterns. They did use SVM and rule-based approaches to question classification. Of the test questions, the SVM-based classifier properly identifies approximately 83.3%, whereas the rule-based correctly classifies about 98.3%.

Amharic Question Answering for Factoid and List Questions using Machine learning Approaches (**By: Medhanit Getachew**)

Machine-learning Approaches for Amharic Question Answering for Factoid and List Questions are proposed in article [11]. In this study, an attempt is made to build a NER-based response extraction system and a machine learning-based system for responding lists and factoids. Ethiopian history is the main topic of the research, which is a closed domain QA in Amharic. It consists of three components. Question extraction component: selects answers from the highest ranked sentences using a NER created especially for this study; passage retrieval: selects pertinent sentences using sentence-level retrieval; question classification: uses two algorithms: HMM and SVM; and answer extraction component: uses sentence-level retrieval to select relevant sentences. The study's F-measure for question classification using the SVM classifier was 73%, while the study's F-measure for question classification using the HMM classifier was 65%. Based on the findings they collected, they concluded that SVM has a better answer extraction capability than HMM for question categorization. Lastly, the researcher suggests deep learning to enhance the system's performance. A variety of machine learning techniques can be utilized to identify the type of answer and categorize questions.

Designing Amharic Question Answering model for Healthcare Using Deep Learning Approach (**By: Abreham, Alene Taye**)

[12] Within a closed domain, this study's Amharic Healthcare investigation Answering System (AHQAS), which is based on deep learning, was created to integrate a variety of

inquiry forms, including factoid and non-factoid categories including Definition, Description, Factoid_numeric, and List. Deep learning techniques like Word2Vec have the advantage of automatically extracting features from unstructured text data, in contrast to traditional methods that necessitate extensive manual feature engineering. By using the contextual similarity of words throughout the dataset, this method creates semantic word embeddings, which represent words as vectors. This study's 1800 question-answer pairs came from a collection of Amharic health information platforms, including Hellodoctorethiopia, Doctor Alle, Healthinfoamharic, Medicinenet, Dradugnaw, and Experts. The dataset was prepared by applying preprocessing tasks like stemming, tokenization, normalization, and stop-word removal. Word embeddings were created after preprocessing, and LSTM and BiLSTM models were trained for question type classification and answer extraction. In comparison to LSTM, which obtained 96.2% accuracy for question type categorization, the BiLSTM model performed better in the second experiment, with 98.3% accuracy. With an average precision, recall, and F1-score of 87.5% and an overall accuracy of 81.8%, the BiLSTM model fared better in response extraction as well. Because BiLSTM is bidirectional and enhances performance in Amharic QA tasks, the study found that it is the best model.

Author	Title	Problem	Method/Tool	Result
Abreham Alene Taye [14]	Designing Amharic Question Answering Model for Healthcare Using Deep Learning Approach	Amharic question and answer construction differs from English and other languages, requiring specific handling in QAS.	Deep learning with Bi-LSTM for question classification, pre-trained Word2Vec for word embeddings, and cosine similarity for answer retrieval.	98.1% question classification accuracy, and 87.5% F-score in answer retrieval.
Medhanit Getachew [12]	Amharic Question Answering for Factoid and List Questions using Machine-learning Approaches	The majority of earlier studies simply employed the SVM method to classify questions, and none of them employed NER to extract answers.	HMM and SVM are used for question categorization, while the NER tool was utilized for answer extraction.	73% of SVM classifiers and 65% of HMM classifiers. Answers for precision (75%), recall (72%), and F-score (73%) are all accurate.
Seid Muhie Yimam [13]	TETEYEQ (ተጠየቅ): Amharic Question Answering System for Factoid Questions	Amharic question and answer construction differs from English and other languages, requiring extra attention in the QAS.	Using the Eclipse Java Editor, algorithms are designed in the Java programming language. Gazetteer is utilized for preprocessing instead of named entity recognizer, and Lucene indexer is used for programming.	72% of user inquiries have accurate answers, while 89% of questions are accurately classified.
Tilahun Abedissa [11]	Amharic Question Answering for Definitional, Biographical, and Description Questions	It is not possible to respond to definitional, biographical, or descriptive questions in Amharic using Amharic factoid questions.	Lucene indexing API; rule-based and SVM-based question classification methods.	SVM classifier: 83.3%, Rule-based: 98.3%, Recall: 61%, Precision: 68%, F-score: 59%.

CHAPTER THREE

3. Design of Amharic Multi hop Question Answering System

3.1. overview

This chapter describes the architecture and implementation of the Multi-hop Amharic Question Answering System. The design and implementation of the proposed QA system consists of three primary phases: preprocessing the dataset, question analysis, and answer extraction. Examples of dataset preprocessing include character normalization and stop word removal. Question analysis includes question classification. The last step in the design and implementation process is answer extraction. This chapter offers a comprehensive description of the tools, methods, and algorithms used by the many components of the proposed QA system, along with information on how these modules were designed and put into operation. Because this research uses design science research methodology, it attempts to develop certain artifacts that identify the ideas and products through which the analysis, design, implementation, and use of the system being built can be effectively accomplished. Design science develops and assesses artifacts that aim to solve specified challenges. A design science research study produces an artifact that is produced and implemented to solve a specific problem. The design science method consists of six steps: problem identification, solution objective specification, design and development, demonstration, assessment, and communication [48]. This study takes the steps described above into account.

The first step in design science study is to identify the problem. It is the process of determining important aspects and factors to take into account when conducting research on a particular subject. We conducted an observation and a review of the relevant literature in order to recognize and comprehend the issue we are addressing. We tried to come up with a solution by focusing on the issues that were discovered through observation and asking individuals questions regarding the study and literature review.

The goals of the research should be clearly specified when the issues being addressed have been identified. Once we are aware of our goals, we can plan and carry out the research we are doing. Thus, following the identification of the issues, the study's goals are clearly stated, and we proceed with the design and development of the research while keeping these goals in mind.

3.2. System Architecture

Figure 3.1 below illustrates the overall outflow model of the Amharic Multi-hop question answering model, which starts with obtaining Amharic corpus from several sources to address the extraction problem. It responds to factoid_person, factoid_organization, factoid_location, and factoid_date type queries using deep learning and a pre-trained Amharic word2vec model. These include question type categorization, named entity recognition, word embedding, response extraction, data preparation, and LSTM/Bi-LSTM/CNN model development. After synthesizing word feature vectors and learning word representations from preprocessed data, we begin developing our deep learning model in a hyperparameter experimental configuration. Because of their enormous potential when working with sequential data, such text type data, we decided to look into LSTM and BiLSTM. CNNs were also chosen for our study because to their ability to recognize local patterns and features in data, which makes them highly useful for tasks like text categorization and sentence modeling. By using convolutional filters, CNNs can identify n-grams and other relevant sequences in text input. This is useful for understanding contextual links and hierarchical features in natural language processing applications. Because of their strong feature extraction capability, CNNs are a suitable choice for handling text-type data, especially when combined with techniques like pooling for dimensionality reduction.

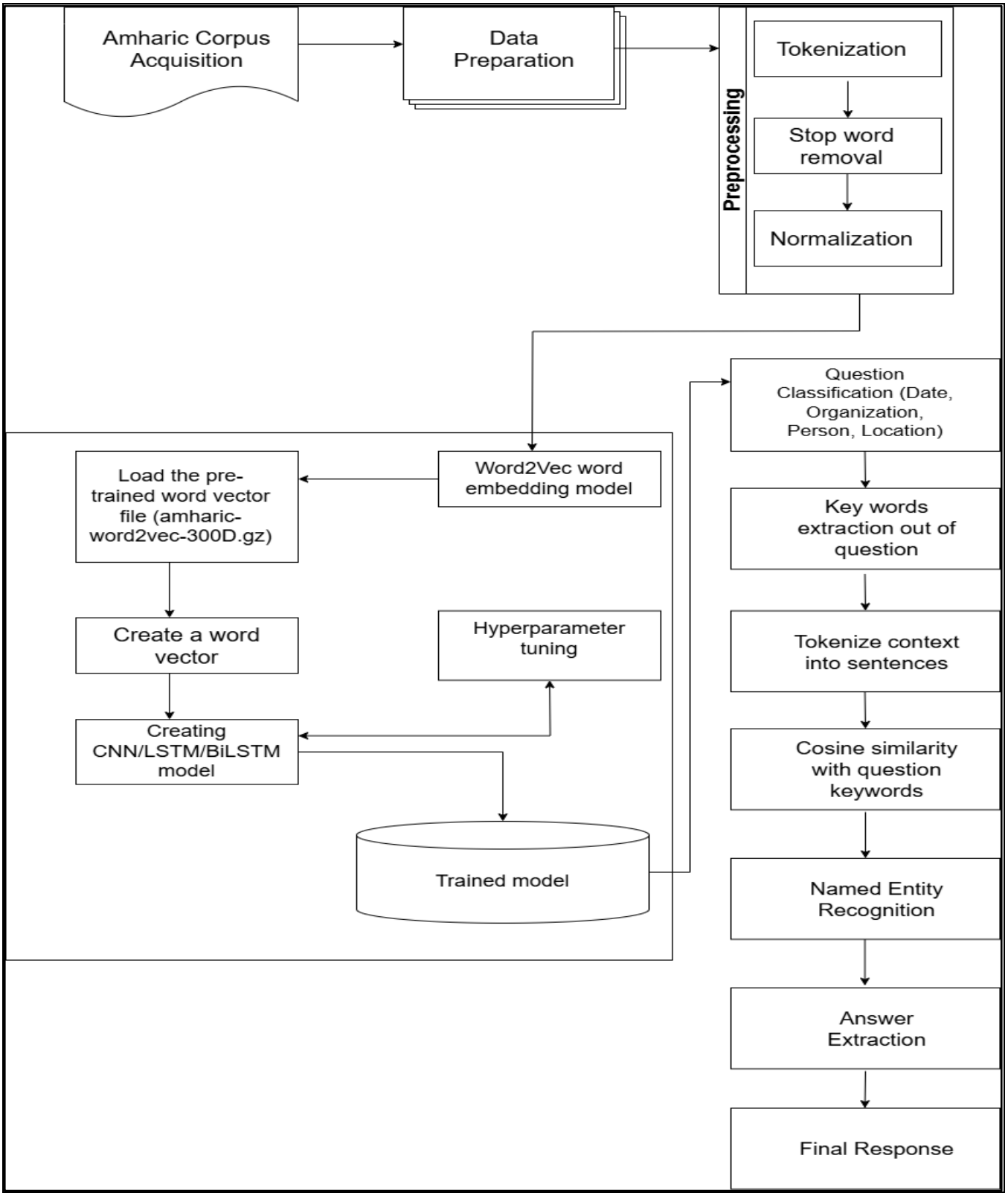

Figure 3. 1: The Architecture of AMHQA

3.3. Data Preprocessing

Data preprocessing is the process of getting raw data ready for a deep learning or machine learning model. Preprocessing is required before sending a raw data set to the learning model. A raw data set needs to be preprocessed before being supplied to the learning model since it could contain noise and irrelevant data, which lengthens the training period and lowers the model's performance. Such data must therefore be tokenized, padded, and normalized. Additionally, certain terms that add nothing to the dataset's substance must be removed.

3.3.1. Tokenization

The practice of tokenizing a corpus into word-level segments or breaking up data into small word-sized parts is known as tokenization. By taking the input text from the user or from the given corpus and breaking it up into a series of tokens, it turns a stream of text into words. Lastly, it offers a list of the terms that will be used in the next step of preprocessing. In most cases, white spaces and other punctuation like the question mark "?" are used as the closest representations of Latin word-to-word delimiters, which are boundary marks that separate word sequences. Like the other languages, Amharic has its unique punctuation marks that separate texts or phrases into a stream of words. Amharic punctuation marks include ‘ሁለት ነጥብ’ (፡)/two points/ ኮለን (,) /colon/ ‘አራት ነጥብ(።)/full stop/, ‘ነጠላ ሰረዝ’ (፣)/semicolon/, ‘ድርብ ሰረዝ’ (፤)/double comma/, ጥያቄ ምልክት?’/question mark/, ቃል አጋኖ (!)/exclamation mark/, ይዘት (.)/dot/, and ትምህርተ ስላቅ (¡)/Education sarcasm/are used as sentence delimiter or as white space.

3.3.2. Normalization

Tokenization separates the given text into words or paragraphs. Since the Amharic language has three distinct kinds of normalization issues, normalization in this study involves arranging a collection of characters into a single character. The first is a compound term that is thought of as two distinct words in the system, such as "ህገ" and "መግስት," but in Amharic, it is only one word. This kind of word is dealt with using a list of double words, and if the situation arises, we can use a substitute like ህገ_መግስት [10].

The second kind of normalization, which involves a slash (/) and a dot (.), extends and writes as ፍርድ_ቤት when it appears occasionally as a title like ፍ/ቤት. In conclusion, Amharic normalization results from consistent writing styles, such as አላማ ዐላማ ዓላማ substitute {አ፣ ዐ፣ዓ} by አ. However, some linguists and academics have voiced concerns regarding the use of these Amharic character normalization methods. They say that the issue is that the algorithm substitutes a sound for every character. Meaning and linguistic standardization are thus problematic. Examine the next two terms: The sole distinction between the two terms is the shape of the starting characters, ሰ and ሠ, which are ሰረቀ (stole) and ሠረቀ (stole), respectively. We investigated whether the aforementioned normalizing method had an impact on our model because the linguistic experiment yielded a range of findings based on norms for the Amharic Language. According to our research, the previously described normalization procedures have no effect on the Amharic question answering model.

Table 3. 1: Characters with their normalizing characters

Characters Used Interchangeably	**Normalized To**
ሀ, ሃ, ሐ, ሓ, ኀ, ኃ	ሀ
ሰ, ሠ, ሱ, ሡ, ..., ፣ ሶ, ፣ ሦ	ሰ, ሱ, ..., ፣ ሶ
ጸ, ፀ, ጹ, ፁ, ..., ፣ ጾ, ፣ ፆ	ጸ, ጹ, ..., ፣ ጾ
ቸ, ቼ	ቸ
ቻ, ቿ	ች
ሸ, ሼ	ሸ
የ, ዬ	የ
አ, ኣ, ዓ, ዐ	አ
ኝ, ኟ	ኝ
ሽ, ሿ	ሽ

3.3.3. Stop Word Removal

There has been discussion about the value of stop words common terms that are frequently eliminated from natural language processing tasks in a variety of settings. Despite being conventionally regarded as non-predictive [41]. While stop word removal has varying effects on different applications, it can potentially reduce corpus size by 35–45% and

increase text mining efficiency [41]. In many NLP applications, stop words are routinely eliminated since they change the meaning of important phrases in a document. Considering that during model training, it uses more memory and processing time and takes up most of the page. There are no universal or standard stop words in Amharic since they are context-dependent. The most often used words in any language's text are also the ones with the least information. Amharic writings frequently use conjunctions, articles, and prepositions, although they seldom ever offer any useful information. Thus, in order to improve accuracy, reduce memory use, and yield better results by focusing on the key terms that reduce noise and false positives, stop words were removed from our corpus before training the model. Those, terms like ወይም ፣ በዚህ ፣ እንደ ፣ ነገር ፣ እና ፣ ና ፣ … that were included in the dataset were eliminated by a list of stop words.

3.3.4. Padding

Similar in size and form inputs are needed by all neural networks. Not every phrase has the same length when we pre-process and use the texts as inputs for our models, like LSTM, CNN and BiLSTM, because some sequences are long and others are short. This is where the padding comes in: we need all of the inputs to be the same size. The process of lengthening a sentence to the appropriate length by appending padding tokens to its beginning or finish is known as padding. If needed to meet the length requirements, we can also omit a few words. Pad sequence is a useful tool for trimming sentences to a precise length or for adding padding to the beginning or conclusion of a sentence. To ensure that the maximum number of word documents is uniform, the zero (0) padded is applied during the padding process.

3.3.5. Word Embedding

One type of word representation that makes it possible to show words with comparable meanings is word embeddings. Deep learning achieves remarkable results on challenging natural language processing problems primarily due to its novel distributed representation for text. Word2Vec is a neural network with two layers that is utilized for predictive language modeling training of word representations. A vocabulary with vectors associated to each item (word) is its frequently used output. These vectors can be queried to find

correlations between words or fed into a deep learning network. In our thesis, we built data representation and semantic meaning extraction for the deep network using word2vec as an embedding schema. The size of the corpora affects the quality of the embedding models, so we are used pre-trained word embedding model [49].

3.4. Question Type Classification

Question classification is one of the sub-components of question processing, and it's critical to recognize the different types of questions in order to know what kind of answer to expect. To determine the type of question, we applied deep learning techniques. Several deep learning models have recently been employed to handle these problems, with impressive results in NLP. The adoption of Word2vec models increased the classification models performance greatly by learning the text's semantic and syntactic connections between words. Question type classification involves the transformation of question types into feature vectors. Word embedding generates word representations that can be fed into our model, which learns to identify text classes based on the input vector sequences. Current research on question classification is using Bidirectional Long Short-Term Memory (Bi-LSTM). Bi-LSTM had been used in question classification studies [50], because it is able to classify sentences without seeing any sentence patterns. In addition, Bi-LSTM can carry out a multi-class classification process. The results obtained in the study indicated that Bi-LSTM had better performance. In our study, the impact of using LSTM, CNN and Bi-LSTM recurrent neural network on AmharicMHQAS is investigated. SoftMax classifier is used to classify the kind of questions we receive. SoftMax classifier was employed in our thesis. The classifier is trained for Factoid date, Factoid person, Factoid location, and Factoid organization questions. The classification process is carried out on the trained model using test data that has been preprocessed and is embedded with the same dimension as the training and validation data. The classification report is organized once the model has been tested. As a result, the testing result's ultimate output is a set of questions, each with its own question type.

3.5. Named Entity Recognition

The MasakhaNER dataset, which contains named entity recognition (NER) data from news stories in ten distinct African languages, was utilized to fine-tune the transformer-based NER model that is employed in this question answering system. With a maximum sequence length of 200, batch size of 32, and learning rate of 5e-5, the fine-tuning procedure was carried out across 50 epochs. The training process employed five distinct random seeds, and the model with the highest aggregate F1 scores across the test set was determined to be the top performer. On November 20, 2021, this model which was improved by Michael Beukman as part of a project at the University of the Witwatersrand in Johannesburg was made available as version 1 under the terms of the Apache License, Version 2.0.

Luckily, the developers work has freed researchers from having to invest a lot of time and money in creating a NER system from the ground up, allowing them to concentrate more on the actual QA process. Since many fact-based answers contain entity names that may be effectively recognized and retrieved using NER, understanding entity names is essential for attaining high performance in QA. The accuracy and simplicity of response extraction are much improved by the integration of this technology. Despite being created as a stand-alone system, the NER model is crucial to increasing the overall effectiveness of the QA system since it helps to eliminate unnecessary content and discover important entities in the corpus. Researchers can concentrate on more intricate areas of the QA assignment as the NER model simplifies the extraction process by eliminating extraneous strings that don't include answers. This version highlights how the model frees researchers to concentrate on the actual QA task while expressing gratitude for the model's creation.
For this research we used a transformer-based model fine-tuned on the MasakhaNER dataset which recognizes the entities; person (PER), organization (ORG), date (DATE), location (LOC) and entity (ENTY) which appeared in our corpus.

3.6. Answer Selection

The final words of the question are the main focus of our approach's answer extraction phase since, in multi-hop question answering tasks, they frequently express the primary

question. The context is tokenized into different sentences once these important terms have been determined. The highest similarity score is used to select the sentence that best reflects the relevance of each sentence when it is compared to the key words using cosine similarity.

Next, we apply Named Entity Recognition (NER) on the chosen sentence using the identified question type (e.g., person, location, date) in order to extract the final response. In order to guarantee that the extracted response accurately answers the inquiry, NER is essential in reducing the number of pertinent entities and phrases. By concentrating on the most pertinent contextual elements, this method improves the accuracy of answer selection.

3.7. Performance Metrics

We choose performance metrics and confusion metrics to measure the model performance using standard classification metrics such as accuracy, precision, recall, and f1-score [51]. For multiple classification problems, such metrics are simple to obtain and can be computed as follows:

Accuracy is determined by dividing the total number of question type classifications by the sum of the accurate question type classifications. The following is the mathematical expression:

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

Precision: it is a measure of the true positive among all positives. Precision assesses how closely the returned response relates to the original query. The following is the mathematical expression:

$$Precision = \frac{TP}{TP + FP}$$

Recall: Often referred to as sensitivity, it indicates the number of accurate predictions you made out of actual positive facts. Recall assesses how many of the really pertinent answers were retrieved by the system. The following is the mathematical expression:

$$Recall = \frac{TP}{TP + FN}$$

F1-score: is the recall and precision weighted average. The expression in mathematics is shown below.

$$\mathrm{F1-score} = \frac{2 * (precision * Recall)}{Precision + Recall}$$

Where:

- TP (True Positives): The number of correctly predicted positive instances.
- TN (True Negatives): The number of correctly predicted negative instances.
- FN (False Negatives): The number of instances that were positive but predicted as negative.
- FP (False Positives): The number of instances that were negative but predicted as positive.

CHAPTER FOUR

4. Experiments and evaluation

4.1. Introduction

To validate the proposed Amharic multi-hop question answering, model a series of experiments were done in this chapter. This section of the research also discusses data collection and evaluation of the model. In this chapter, the basic procedures behind the implementation of question answering from Amharic contexts are investigated. The best fit procedures are depicted and the selection of network strategy is elaborated via showing the performance graphs, which visualize the casual design approaches.

4.2. Experimentation

4.2.1. Data Collection

The data collection of this research is not an easy task instead it is very tedious and time-consuming work to collect 1720 questions within its context in the history domain from Internet and some other sources. The collected question and answer pairs are given for one language expert to review and the question within its context which is passed by the language experts taken as a dataset for this research. After the review of the language expert only 1500 question and context pairs passed, and the remaining 220 questions and context pairs were rejected from our dataset because of misconnection of the question and the context. We used different data cleaning mechanisms like stop word removal, and normalization into the tokenized. The data cleaning step's main aim was to prepare the data for training and testing and minimize resource consumption.

4.2.2. Implementation

Since Python is a free open-source programming language that is utilized for data analysis and machine learning in the field of artificial intelligence, we used Anaconda software tools in this experiment. We utilized the Jupiter code editor from Anaconda. An object-oriented programming language that enables quick application development is Python. It is software that uses a high-level programming language that enables to create desktop

and web apps. While Jupiter is a Python environment for scientific creation, it also features editing, interactive debugging, and testing.

4.2.3. Hyperparameters

A number of deep learning hyperparameters influenced the experimental results; however, the loss function was the one we used in the current study. This is one of the most significant features of deep learning algorithms. Gradients are evaluated by predicting the model's error, which is generated by the loss function. To reduce the error rate, the weights of the neurons in a given layer are modified via error backpropagation. In this study, we used categorical cross-entropy loss functions due to the multi-class classification. Using the typical categorical Cross entropy [45], computed as follows:

$$J_{ccce} = -\frac{1}{M}\sum_{m=1}^{M} \sum_{k} y_{km} \log\left(h_{\theta}(x_m, k)\right) \quad \text{.....(4.1)}$$

Where:

- M = The number of training examples or data points.
- K = number of classes
- y_{km} = target label for training example m for class k
- x = input for training example m
- h θ = model with neural network weights θ

Another important aspect of deep learning is optimization, that helps in the reduction of the model's prediction errors. We used Adam, an optimizer that learns faster than others. Adam uses two averages to make better selections during training.

$$m^t = \frac{m_t}{1-\beta_1^t}.$$

$$v^t = \frac{v_t}{1-\beta_2^t} \quad \text{.....(4.2)}$$

$m\hat{}t = mt1 - \beta t1 v\hat{}t = vt1 - \beta t24.\ 2$

β1 = The exponential decay rate for the first moment estimates (0.9)

β2 = The exponential decay rate for the second − moment estimates (0.999)

To prevent overfitting, we also used a method known as dropout. Dropout randomly ignores some neurons during training, allowing the model to generalize better. We set a dropout rate of 0.39. Finally, we used L2 regularization to prevent overfitting and limit the model's weights to modest values. We applied a regularization rate of 0.001. We used Hyperparameter Optimization (HPO) to determine the optimal parameters. We used random search to experiment with different hyperparameter combinations and improve our model's performance.

4.3. Evaluation and Results

4.3.1. Experimental Result of CNN Model

The convolutional neural network (CNN) experiment produced important results for training and validation accuracy when the given hyperparameters were applied. The accuracy metrics shown in the training and validation graphs were used to assess the model's performance.

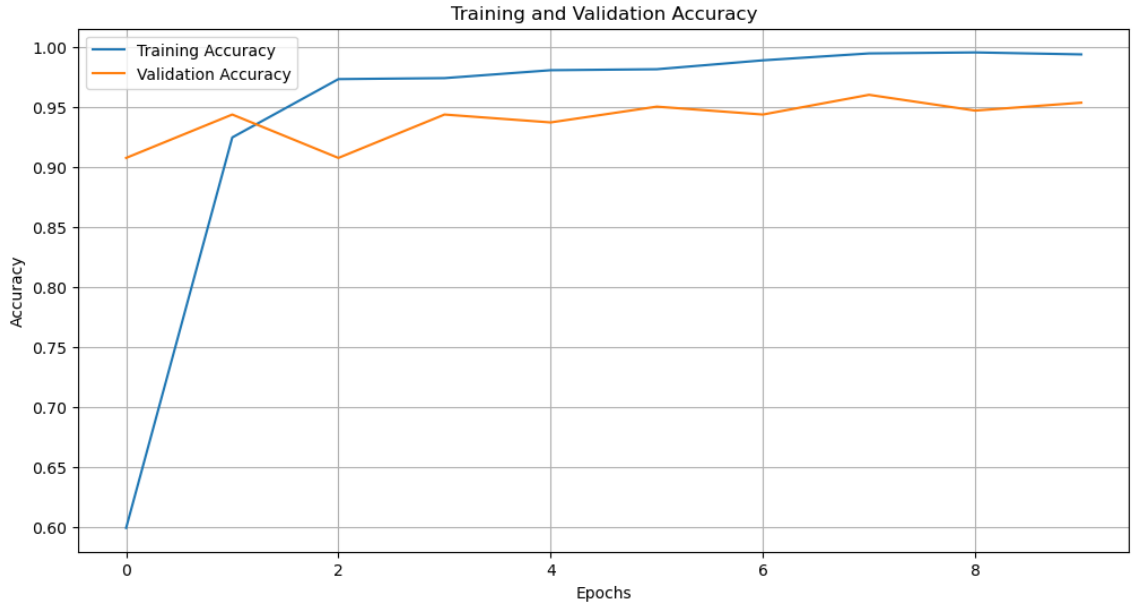

Figure 4. 1:Training and validation Accuracy curve of CNN Model

The training and validation accuracy over epochs is shown in Figure 4.1. After the initial epochs, the training accuracy peaked at almost 100%, demonstrating that the model had successfully learnt the training set. The validation accuracy, on the other hand, increased more gradually and remained at 95%. The trend indicates that although the CNN model

showed remarkable performance on the training dataset, there was a minor gap in its capacity to generalize to previously encountered data.

Convolutional neural network (CNN) experiment findings provide important insights into the learning behaviour of the model, especially when examining the training and validation loss curves. The loss metrics during the training phase are shown in Figure 4.2, which emphasizes the model's capacity to reduce error over epochs.

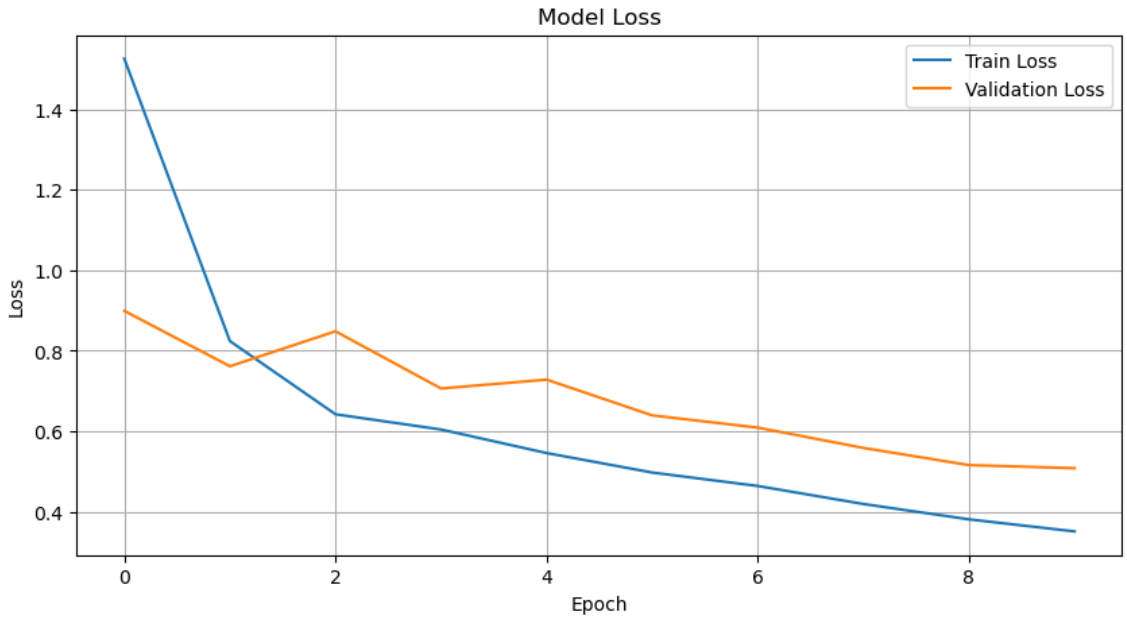

Figure 4. 2: Training and validation Loss curve of CNN Model

The training loss initially showed an immediate drop, which suggested that the training data was well learned. By the conclusion of the training epochs, this declining trend had reached a low point of roughly 0.4. On the other hand, the validation loss showed an increase that was more unstable, declining at first before getting at around 0.6. The lower training loss compared to the validation loss shows that the model can have a tendency to overfit the training data, according to the difference between training and validation loss. However, the total loss reduction for both training and validation shows that the CNN model was successful in extracting useful patterns from the dataset.

4.3.2. Experimental Result of LSTM Model

The performance of the LSTM model is shown in Figure 4.3, which displays the training and validation accuracy over epochs. In the early epochs, training accuracy rapidly improves, eventually reaching close to 100%. This indicates that the model has learned the training data well. The validation accuracy, while slightly lower than the training accuracy, stabilizes around 95%, suggesting the model generalizes reasonably well, though there is a small gap between training and validation accuracy. This gap could indicate some degree of overfitting, as the model performs slightly better on the training set than on the validation set.

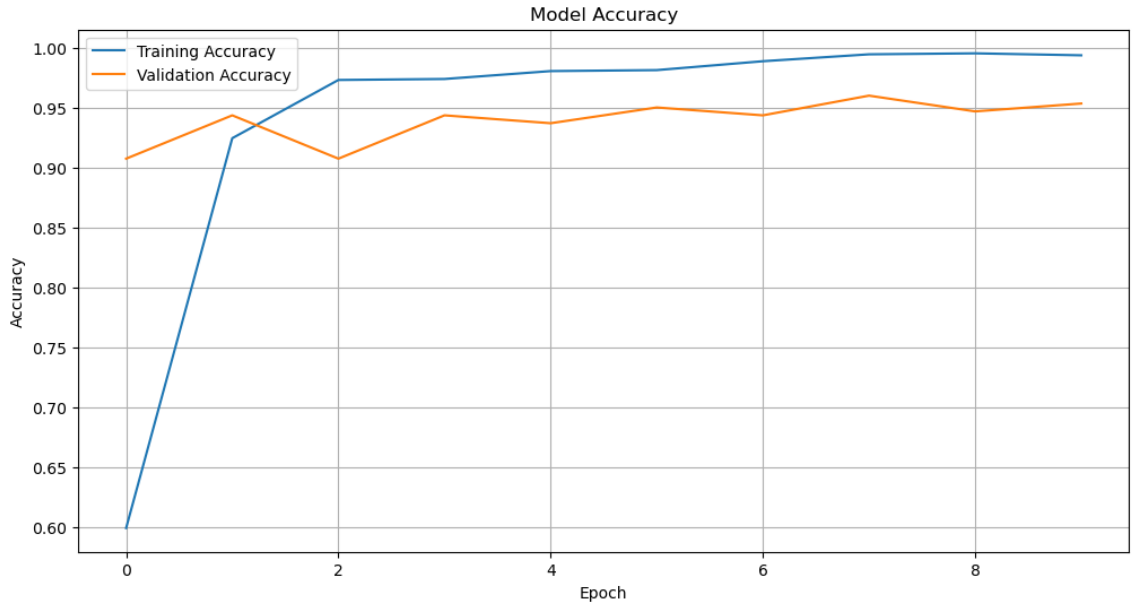

Figure 4. 3: Training and validation Accuracy curve of LSTM Model

While the validation loss did not reach the same low levels as the training loss, it did show a more irregular pattern and progressively declined as training went on. The model's performance on the training set was good, but it had trouble generalizing to the validation set, according to the validation loss curve overall. Even though it is quite small, the difference between the loss from training and validation suggests that there was a little overfitting, meaning that the model performed slightly better on the training set than on the testing set. The LSTM model was apparently learning efficiently and maintaining stable performance, however, based on the general decreasing trend in both loss curves.

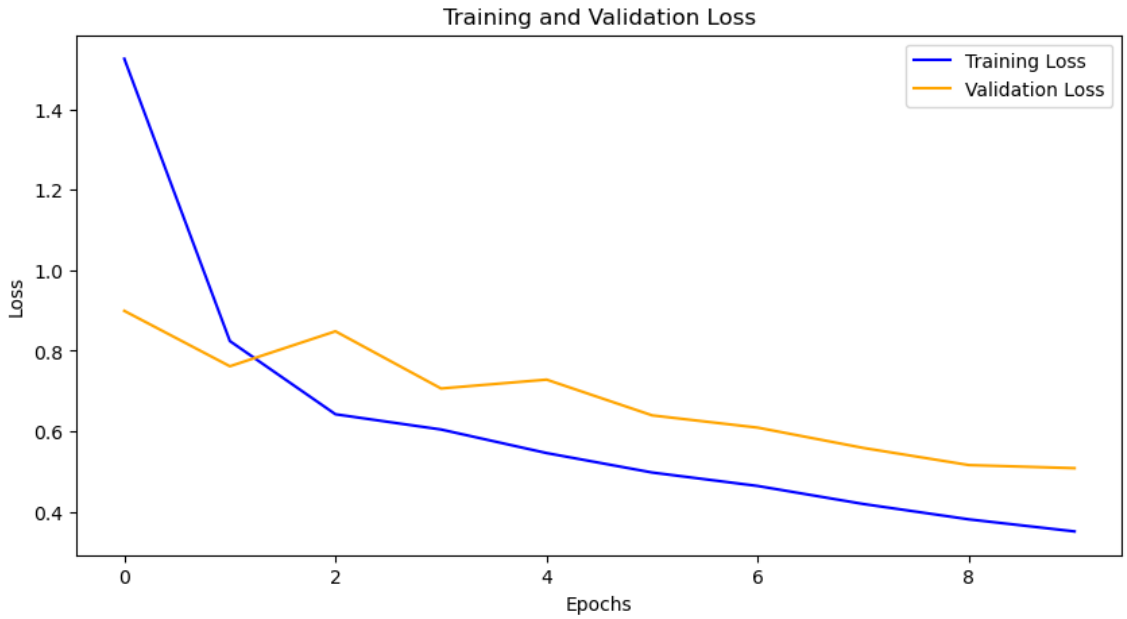

Figure 4. 4: Training and Validation Loss curve of LSTM Model

Figure 4.4 shows the training and validation loss for the LSTM model, which shows a gradual decrease over each of the epochs.

4.3.3. Experimental Result of BiLSTM Model

The training and validation accuracy curves for the Bi-LSTM model over several epochs are shown in Figure 4.5. After a few epochs, the model shows a notable increase in training accuracy, approaching 100% accuracy by the third epoch. This shows that over time, the model is lowering classification mistakes and effectively learns from the training set. The validation accuracy is still rather good, averaging approximately 95% following the initial increase, even though it is not as high as the training accuracy. The model is able to retain strong generalization over different epochs, even with slight differences in the validation accuracy.

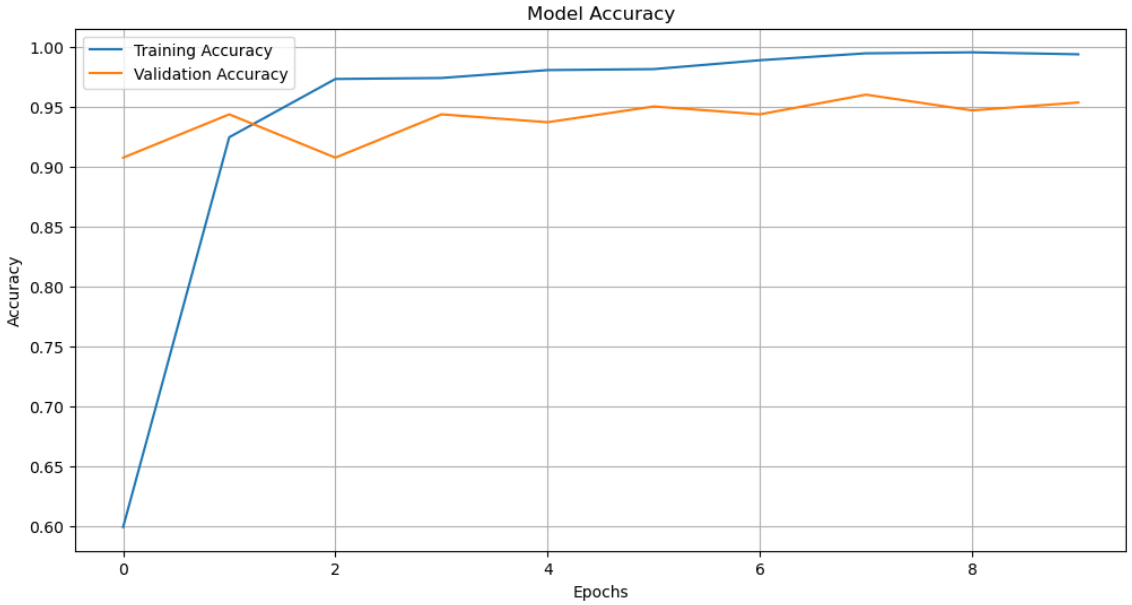

Figure 4. 5: Training and validation Accuracy curve of Bi-LSTM Model

The training and validation loss curves for the Bi-LSTM model over ten epochs are displayed in Figure 4.6. The model appears to be learning and minimizing the error on the training set when the training loss gradually drops as a result of the first few epochs. The training loss has decreased by the conclusion of the tenth epoch, indicating that optimization was successful.

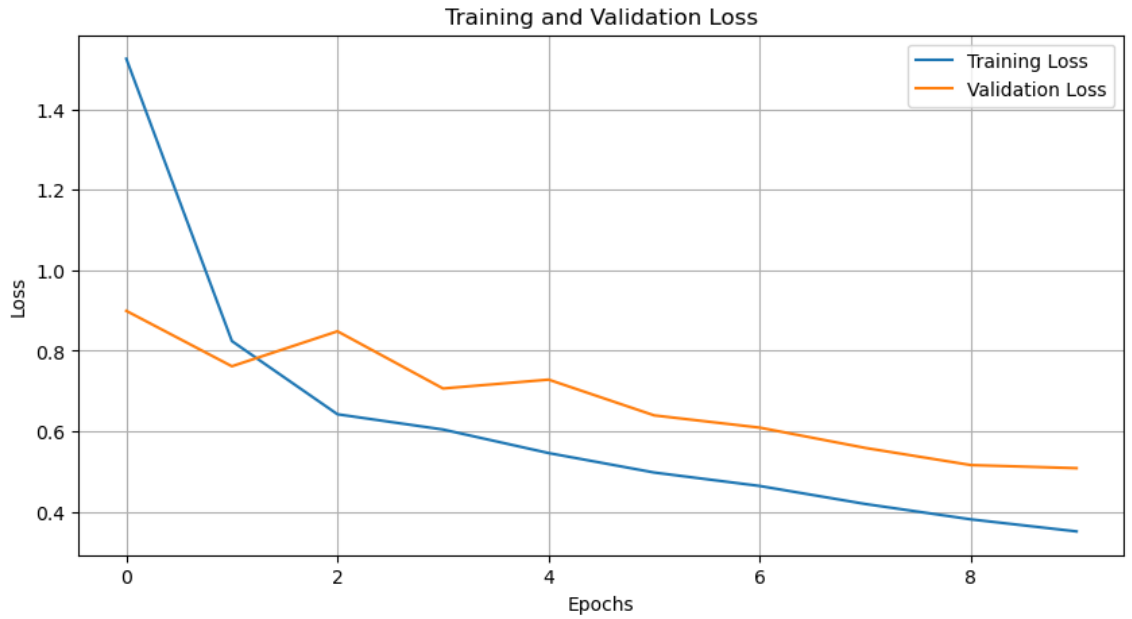

Figure 4. 6: Training and validation Loss Curve of Bi-LSTM Model

To summarize up, the Bi-LSTM model performs well in terms of loss minimization, showing a decrease in loss during both training and validation.

Table 4. 1:comparison of the Bi-LSTM, CNN, and LSTM models

Specifications	**CNN**	**LSTM**	**Bi-LSTM**
Validation Loss	Maybe more likely to show validation loss because long-term dependencies are harder to capture.	lowers	Small and closely matched with training loss (better fit)
Training Accuracy	High convergence, maybe overfitting	Excellent, but takes longer to achieve high accuracy	High performance that stays after a few epochs
Training Loss	Fast initial drop, but sequencing data may be challenging for it.	decreasing	Lower and steadily decreasing
Handling Sequential Data	Limited	Good	Excellent
Overfitting	Higher risk of overfitting	Moderate risk of overfitting	Lower

Table 4.2 displays the Bi-LSTM model's overall performance on AMHQAS. Based on performance metrics, this is the result of one experiment that outperformed the first.

Table 4. 2: Evaluation matrix of Bi-LSTM Model

Question_Type	Precision	Recall	F1-Score	Support
factoid_DATE	0.98	0.98	0.98	84
factoid_PERSON	0.97	0.92	0.95	78
factoid_LOCATION	0.90	0.97	0.94	68
factoid_ORGANIZATION	0.96	0.95	0.95	74
Accuracy			0.95	304
Macro Avg	0.95	0.95	0.95	304
Weighted Avg	0.96	0.95	0.95	304

It is not entirely constant across all categories as there are some small differences, particularly in recall and precision. However, the variations are small, and the model consistently exhibits high performance.

4.3.4. Performance Evaluation of AMHQAS Question Type Classification

We used datasets of 1500 Amharic history question to create a multiclass classification model for question_types. The four classes in the model are factoid_date, factoid_person, factoid_location, and factoid_organization. The total dataset is displayed in Figure 4.6. We adjusted our datasets to have almost equal question distributions. First-class labels go into factoid_date, second-class labels into factoid_person, third-class labels into factoid_location, and fourth-class labels into factoid_organization. As you can see, we're attempting to balance the distribution of questions for each class.

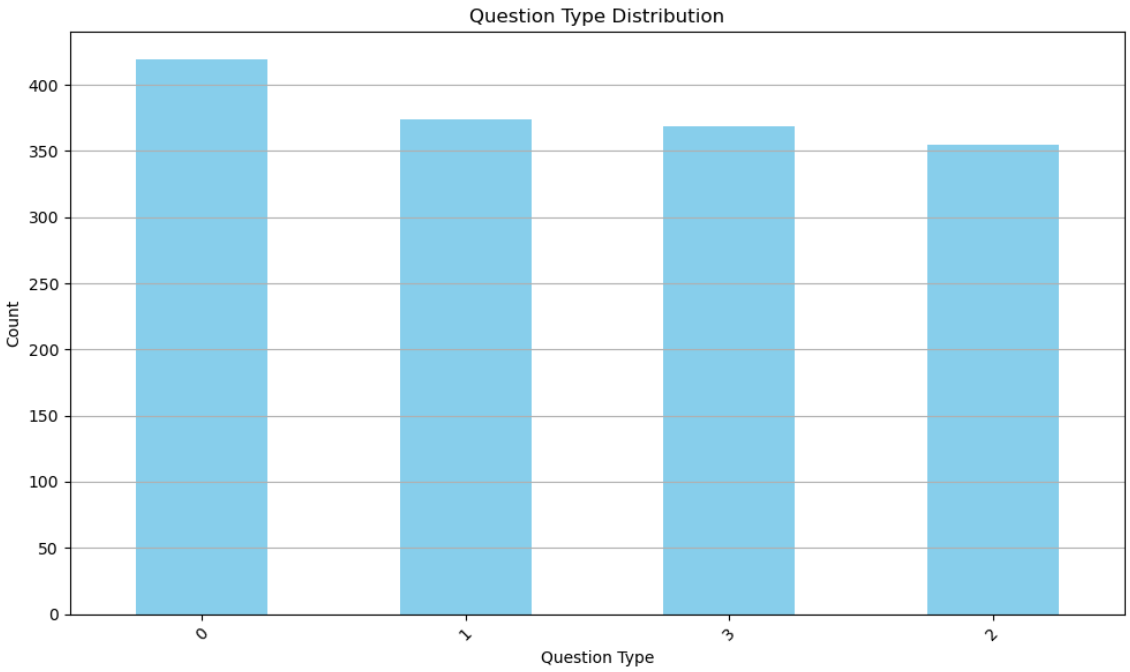

Figure 4. 7: Question type distribution

The next crucial stage is to use the model we trained to classify different types of questions after it has been developed fitted with data, and assessed using evaluation metrics. Before classification, the model needs to be loaded and stored. Thus, the model is saved using the save.model() method, which makes use of h5 extensions and loaded for testing using the models.load_model() Keras library. Our question type classifier has a testing accuracy of 97.04%.

Confusion matrix is an additional evaluation matrix used in this study for categorizing question types. Based on the actual labels in the dataset sentence and a predicted label, such metrics might be computed for each class to be used in the multi-label classification. In Figure 4.7, the AMHQA Question Type Classification's performance is evaluated using the confusion matrix. Confusion matrices are useful for showing which elements of each class are correctly and incorrectly classified. The confusion matrix result was generated using the Bi-LSTM model, and certain values that were connected to the actual value were not predicted by the model. The confusion matrix presented shows the performance of a classification model across four classes. The model is highly accurate, with the majority of instances that properly classified. For Class 0, 83 examples are accurately identified,

with only one instance misclassified as Class 2. Similarly, 76 occurrences of Class 1 are accurately detected, but two are mistakenly classed as Class 3. In the case of Class 2, 65 instances are correctly identified, whereas three are incorrectly classified as Class 3. Finally, Class 3 displays 71 right classifications, with three instances incorrectly displayed as Class 2. The model performs well overall, as shown by the darker diagonal squares indicating correct predictions, and the lighter off-diagonal squares highlighting instances of misclassification, including across neighbouring classes. These misclassifications are small, showing that the model is mainly accurate.

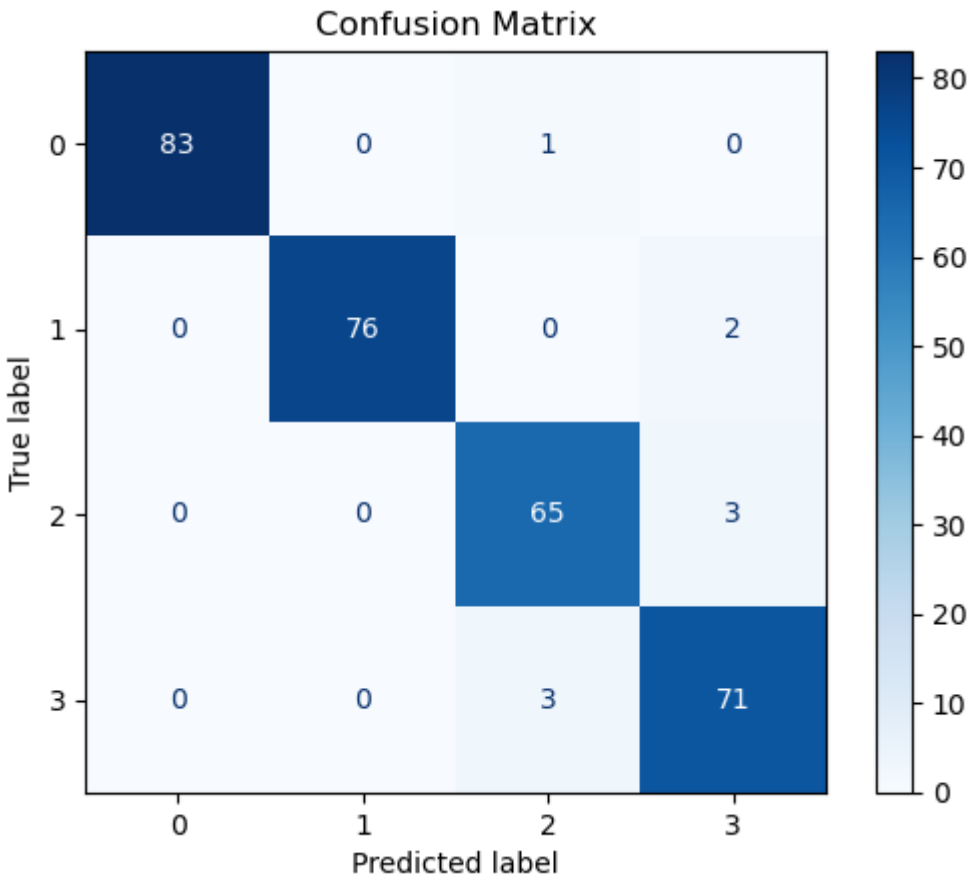

Figure 4. 8: Confusion Matrix of Bi-LSTM Model

Prediction of Question type classification

Five question and response pairs were chosen at random from each question class in order to assess the question classification task's prediction abilities. The summary of the output for question type categorization prediction is displayed in Figure 4.8. By appending the expected label to the question, the Bi-LSTM model predicts the question class of every input question.

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
Sample 1:
Actual Question Type: 0
Predicted Question Type: 0
----------
Sample 2:
Actual Question Type: 0
Predicted Question Type: 0
----------
Sample 3:
Actual Question Type: 1
Predicted Question Type: 1
----------
Sample 4:
Actual Question Type: 3
Predicted Question Type: 3
----------
Sample 5:
Actual Question Type: 0
Predicted Question Type: 0
```

Figure 4. 9: Bi-LSTM question type prediction result

4.3.5. Answer Extraction Evaluation of AMHQAS

To find an answer to a factual question, the key word from the input question searched for in the NE labelled corpus. The appropriate response will subsequently be presented. Otherwise, no answer (መልስ የለም) is returned.

The system's performance was evaluated in terms of answer extraction accuracy. Rather than creating a unique NER model, we used a transformer-based model fine-tuned on the MasakhaNER dataset, which contains news items in ten African languages. This pre-trained model had a considerable impact on the system's recall, assisting with the correct identification of named entity and the selection of the appropriate answer type. In our experiment, 15 questions were chosen, covering factoid_date, factoid_person, factoid_location, and factoid_organization questions. Out of these, 10 were answered correctly.

Table 4. 3: Questions Prepared for Testing

Question	Ground Truth	Extracted Answer	Correct (True
Q1	ሐምሌ 23 ቀን 1914	ሐምሌ23ቀን1914	Yes (TP = 1)
Q2	ኢሳቶ	ኢሳቶ	Yes (TP = 1)
Q3	ምኒልክ	ምኒልክ	Yes (TP = 1)
Q4	ቡልጋ መገዘዝ	ጎንደር	No (FP = 1)
Q5	ሶፍያ	No Answer	No (FN = 1)
Q6	እጅጋየሁ	ታይለሥላሴአምሃሥላሴ	No (FP = 1)
Q7	የእንግሊዝ	የእንግሊዝ	Yes (TP = 1)
Q8	እ.ኤ.አ. ነሐሴ 17 ቀን	No Answer	No (FN = 1)
Q9	በሊባኖስ ብሻሪ ከተማ	በሊባኖስብሻሪከተማ	Yes (TP = 1)
Q10	በሃይደልበርግዩኒቨርሲቲ	በሃይደልበርግዩኒቨርሲቲ	Yes (TP = 1)
Q11	እ.ኤ.አ መጋቢት 1 ቀን	እ.ኤ.አ መጋቢት 1 ቀን 1896	Yes (TP = 1)
Q12	የአድዋ ጦርነት	ምኒልክእቴጌጣይቱበጡል	No (FP = 1)
Q13	ኑር ኢብን ሙጃሂድ	ኑርኢብንሙጃሂድ	Yes (TP = 1)
Q14	በ1809	በ1809	Yes (TP = 1)
Q15	ተንታ ከተማ	ተንታ ከተማ	Yes (TP = 1)

- True Positives (TP): 10 (Q1, Q2, Q3, Q7, Q9, Q10, Q11, Q13, Q14, Q15)
- False Positives (FP): 3 (Q4, Q6, Q12)

- False Negatives (FN): 2 (Q5, Q8, because there was no predicted answer but a ground truth answer existed)

Precision Calculation:

$$Precision = \frac{True\ Positives(TP)}{True\ Positives(TP) + False\ Positives(FP)}$$

=10/ (10+3) = the Precision is approximately 77%

Recall Calculation:

$$\text{Recall} = \frac{True\ Positives(TP)}{True\ Positives(TP) + False\ Negatives(FN)}$$

=10/ (10+2) =83.33%

F1 Calculation:

$$\text{F1} = \frac{2 * \text{percision} * \text{recall}}{Percision + recall}$$

= 2*77*83.33/ (77+83.33) = 80.04%

The F1-Score considers both precision and recall, providing a balanced assessment of the Model's performance. For example, in our study the F1-Score of 80.04% represents the difference between precision and recall: while precision was relatively low (77%), recall was higher (83.33%), and the F1-Score falls somewhere in the centre.

Table 4. 4: Performance of model in terms of precision, accuracy, F1-Score measure, and recall

Evaluation metrics	**Performance**
Accuracy	66.6%
Precision	77%
Recall	83.33%
F1	80.04%

In conclusion, while the model performs well overall, its relatively low accuracy and precision scores show areas for development, particularly in reducing false positives. Improving precision while keeping high recall should be a priority in order to improve performance even further.

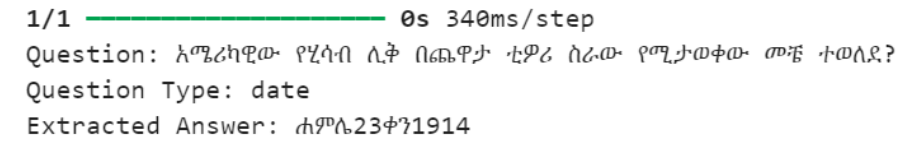

Figure 4. 10: screen shot of correct answers

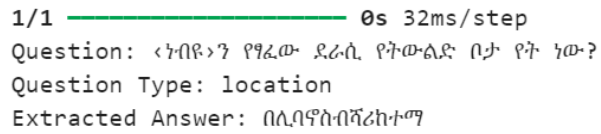

Figure 4. 11:screen shot of correct answers

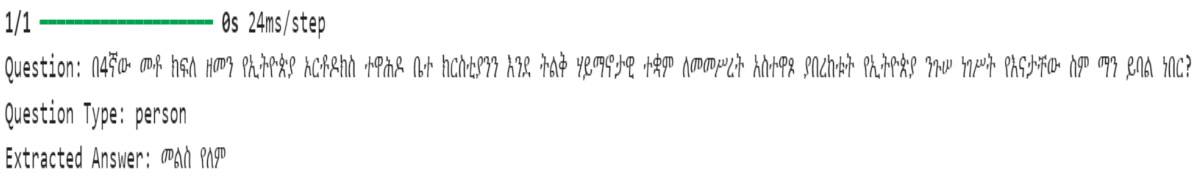

Figure 4. 12:screen shot of question with no answers

4.3.6. Discussions

In this section, we evaluated the model's performance using a testing dataset and several approaches. Three experiments were conducted using different deep learning algorithms with fixed and variable hyperparameters. We employed word interpretation and graphical analysis to evaluate the comparative results. The following section discusses the hyperparameters in the three studies and their impact on our experiment.

The constant (Emb =Word2vec Split ratio = 0.2 Epochs = 10) and variable hyperparameters (Random State, Optimizer, Activation Function, Batch Size, Dropout Rate, and Number of Neurons) were used in two separate tests as shows in Table 4.5, These hyperparameters control many aspects of model complexity, regularization, and learning processes.

Table 4. 5:Hyperparameters

Hyperparameter	**Value**	
Random State	Experiment 2	Experiment 1
	42	0
Optimizer	Experiment 2	Experiment 1
	Adam	Nadam
Activation Function	Experiment 2	Experiment 1
	ReLU	Tanh
Regularization	Experiment 2	Experiment 1
	L2(withfactor	L2(withfactor
Batch Size	Experiment 2	Experiment 1
	8	16
Dropout Rate	Experiment 2	Experiment 1
	0.39	0.2
Number of Neurons (LSTM)	Experiment 2	Experiment 1
	60	100
Dense Layer Neurons	Experiment 2	Experiment 1
	4	4

In experiment 1, 3 deep learning methods, including LSTM, BiLSTM, and CNN models, were evaluated under specific hyperparameter configurations: activation function set to "tanh," batch size = 16, random state = 0, and the number of neurons = 100. The optimizer used was NAdam. Throughout the experiments, we used the "tanh" activation function, which, while advantageous for non-linear transformations, has limitations due to the potential for vanishing gradients within the range of -1 to 1. This can make it difficult for the models to reach the global minimum efficiently. The results showed different performances across the models. While in this experiment, the LSTM model had quite lower test accuracy of 88% and a larger test loss of 0.4892, indicating possible overfitting during training. The BiLSTM model actually performed well, with a test accuracy of 96.38% and a loss of 0.2429, despite a little rise in loss during validation. However, the CNN model had somewhat lower test accuracy (95.00%) and higher test loss (0.4402),

indicating some instability in model training, probably due to the complex nature of the data. In this experiment, the accuracy of LSTM, CNN, and Bi-LSTM achieves 88%, 95%, and 96.38% respectively.

During the second experiment the constant hyperparameters were defined in the first experiment and used in the second. However, the variables were: random state = 42, optimizer = Adam, activation = ReLU, regularize = L2 (0.001), batch size = 4, epochs = 10, dropout = 0.39, and number of neurons = 60. The Adam optimizers employed in this experiment have a quite higher learning rate than Nadam. The random state, which is used to regularize and decrease the graph's overfitting concerns, is increased from zero to 42. The results showed distinct performances across the models. The LSTM model scored 94% accuracy on the test set with a loss of 0.1734. The BiLSTM model performed well, with a test accuracy of 97.04% and a test loss of 0.1616, demonstrating its robustness in capturing bidirectional dependencies. CNN models, while usually suitable for sequence classification applications, shown varying performance. It achieved test accuracy of 96.38% and a test loss of 0.3212. In this experiment, the accuracy of LSTM, CNN, and Bi-LSTM achieves 94%, 96.38%, and 97.04% respectively.

These results show the difficulties of modifying hyperparameters. While the CNN models performed reasonably well, CNN and BiLSTM models outperformed LSTM, and BiLSTM outperforming both of the models.

Table 4. 6: comparison result of the three deep learning algorithms

Model	Experiment 1	Experiment 2
LSTM	88%	94%
Bi-LSTM	96.38%	97.04%
CNN	95%	96.38%

If a question word is classed as a person type but is classified as a location, the question answering system may fail to give a correct response or produce a wrong answer. The effectiveness of passage retrieval has a considerable impact on the quality of question responses. If the retrieval component chooses wrong sentences as the top-ranked ones and the highest-ranked sentence does not contain the candidate answer, the performance of

answer selection will be affected. All of the components stated above help to improve the system's overall performance and accuracy. In our study, passage retrieval does not appear to be the problem, because we do not employ the passage retrieval component. Instead, the issue is in selecting keywords from the question. Because cosine similarity is calculated after recognizing keywords and sentences, if the solution is not found in the sentence with the highest similarity score, the algorithm will either return nothing or display wrong result.

Answer to Research Questions

I. Which of the deep learning algorithm is more effective for Amharic Multi-hop question answering in history?

Bi-LSTM correctively identified questions within 97.04% and provides the best performance in sequence-based tasks due to bidirectional learning, lower loss, higher accuracy, and greater generalization. LSTM has good performance but may not generalize as well as Bi-LSTM because of its unidirectional nature. CNN has fast convergence, but may struggle with complex temporal patterns, resulting in increased validation loss and perhaps overfitting in sequence tasks. Based on the experiment and taking corrective actions by controlling variable hyperparameters Bi-LSTM is the superior model for tasks containing sequence data, such as text, surpassing CNN and LSTM in terms of accuracy and generalization.

II. To what extent the Deep learning approach provide correct answer for user's query?

Our experiments indicate that the deep learning algorithms LSTM, CNN, and Bi-LSTM obtained 96%, 96.38%, and 97.04% accuracy, respectively.

III. What are the optimal values of hyperparameters that give best performance?

In the second experiment, the optimal hyperparameters were identified based on performance improvements. The following values showed the best results:

Random State: 42 **Optimizer:** Adam **Activation Function:** ReLU **Regularization:** L2 with a value of 0.001 **Batch Size:** 8 **Epochs:** 10 **Dropout Rate:** 0.39 **Number of Neurons:** 60

CHAPTER FIVE

5. Conclusion and Recommendation

5.1. Conclusion

QAS enables users to ask natural language questions and receive exact answers from a collection of documents. history information's are needed for researching historical events, people, and other history-related topics. History learners may struggle to find the relevant knowledge among the available materials. To address such issues, we use AMHQAS. This study aimed to provide a deep learning-based multi-hop QA for Amharic history domain, including factoid_date, factoid_person, factoid,location definitions, and factoid_organization in the closed domain. Experts reviewed 1720 Amharic question-and-context pairs in the history domain and fed the results into the developed model. After linguistic expert evaluation, only 1500 question and context pairs passed. The remaining 220 pairs were rejected due to different reasons. The trained model uses the dataset and hyper-parameters from 2 distinct tests for each mode to predict the predetermined class of questions. The model's effectiveness was assessed by confusion and performance metrics, including accuracy, precision, recall, and f1-score. The second experiment showed that LSTM, CNN, and Bi-LSTM attained accuracy rates of 96%, 96.38%, and 97.04%, respectively. Bi-LSTM showed great performance than other models, obtaining a state-of-the-art accuracy of 97.04%.

5.2. Recommendations

This thesis investigates the use of deep learning to develop AMHQAS. Based on the study findings, the researcher suggests improvements for future research:

- Our study aims to address factoid_date, factoid_person, factoid_loaction, and factoid_organization questions. Future research will address further types of questions like comparison.
- In our study, users type query onto keyboards to receive answers from the system. However, some people may not write Amharic and may instead prefer to utilize voices. So, for future it is good to have audio based multi-hop question answering.
- This study utilized deep learning approaches, which requires a huge number of datasets for optimal performance. Having a huge corpus is essential for creating NLP systems. The QAS performance improves with increasing corpus size. To enhance model performance, researchers should create a high-quality, larger corpus.
- Dataset collecting presents a challenge in task. For future work, we advocate automatically generating multi-hop question-answer combinations from specified context. This can be used to develop QA datasets without human annotations, providing a better option.
- In this study, we used pre-prepared contexts to generate replies, which limited the scope of answer extraction from existing papers. However, in the future, it is encouraged to investigate multi-hop information retrieval strategies. This strategy would allow for dynamic retrieval of relevant documents from a wider corpus, enhancing the system's ability to handle complex queries that involve reasoning across different sources of information.

Reference

[1] J. Allen, Natural language understanding. USA: Benjamin-Cummings Publishing Co., Inc., 1988.

[2] "A. Saini and P. K. Yadav, 'A Survey on Question – Answering System,' vol. 6, no. 3, pp. 20453–20457, 2017. - Google Search." Accessed: Sep. 25, 2024.

[3] "Andrew Greenwood Mark, 'Open-Domain Question Answering.' Unpublished PhD Dissertation, Department of Computer Science, University of Sheffield,Sheffield ,2005. - Google Search." Accessed: Sep. 25, 2024.

[4] V. Mavi, A. Jangra, and A. Jatowt, "Multi-hop Question Answering," May 31, 2024, arXiv: arXiv:2204.09140. Accessed: Oct. 03, 2024.

[5] E. Stroh and P. Mathur, "Question answering using deep learning," unpublished, 2016, Accessed: Sep. 25, 2024.

[6] "A Neural Question Answering System for Basic Questions about Subroutines | IEEE Conference Publication | IEEE Xplore." Accessed: Sep. 25, 2024.

[7] "Preena M. P. & Shibily J., 2019 - Google Search." Accessed: Sep. 25, 2024.

[8] "T. Yeshambel, J. Mothe, and Y. Assabie, 'Learned Text Representation for Amharic Information Retrieval and Natural Language Processing,' Information, vol. 14, no. 3, Art. no. 3, Mar. 2023, doi: 10.3390/info14030195. - Google Search." Accessed: Sep. 25, 2024.

[9] "Amharic," Wikipedia. Oct. 25, 2023. Accessed: Oct. 25, 2023.

[10] T. Abedissa, "ADDIS ABABA UNIVERISTY SCHOOL OF GRADUATE STUDIES".

[11] M. Getachew, "ADDIS ABABA UNIVERSITY SCHOOL OF GRADUATE STUDIES SCHOOL OF INFORMATION SCIENCE," 2019.

[12] D. Melesew, "MSc thesis on: Designing Amharic Question Answering model for Healthcare Using Deep Learning Approach".

[13] N. V. Otten, "Top 5 Ways To Implement Question-Answering Systems In NLP & A List Of Python Libraries," Spot Intelligence. Accessed: Mar. 21, 2024.

[14] A. Agrawal et al., "VQA: Visual Question Answering," Oct. 26, 2016, arXiv: arXiv:1505.00468. doi: 10.48550/arXiv.1505.00468.

[15] A. Abbasiantaeb, Z., & Momtazi, S. Z. ,. &. Momtazi, S., "Text-based question answering from information retrieval and deep neural network perspectives: A survey - Abbasiantaeb - 2021 - WIREs Data Mining and Knowledge Discovery - Wiley Online Library." Accessed: Mar. 22, 2024.

[16] K. Nassiri and M. Akhloufi, "Transformer models used for text-based question answering systems," Appl Intell, vol. 53, no. 9, pp. 10602–10635, May 2023, doi: 10.1007/s10489-022-04052-8.

[17] Yih et al., "Question Answering over Knowledge Graphs: Question Understanding via Template Decomposition" by Wen-tau Yih et al. (2015) - Google Search." Accessed: Mar. 22, 2024.

[18] E. G. Cortes, V. Woloszyn, D. Barone, S. Möller, and R. Vieira, "A systematic review of question answering systems for non-factoid questions," J Intell Inf Syst, vol. 58, no. 3, pp. 453–480, Jun. 2022, doi: 10.1007/s10844-021-00655-8.

[19] M. Breja and S. K. Jain, "A survey on non-factoid question answering systems," International Journal of Computers and Applications, vol. 44, no. 9, pp. 830–837, Sep. 2022, doi: 10.1080/1206212X.2021.1949117.

[20] S. M. Yimam and M. Libsie, “TETEYEQ: Amharic Question Answering For Factoid Questions”.
[21] A. M. N. Allam and M. H. Haggag, “The Question Answering Systems: A Survey.,” vol. 2, no. 3, 2012.
[22] H. Hu, “A study on question answering system using integrated retrieval method,” Unpublished Ph. D. Thesis, The University of Tokushima, Tokushima, 2006, Accessed: Sep. 26, 2024.
[23] J. L. V. González and A. F. Rodríguez, “A Semantic Approach to Question Answering Systems.,” in TREC, 2000. Accessed: Sep. 26, 2024.
[24] “Indurkhya N. and Damereau F.J., (Eds). Handbook of... - Google Scholar.” Accessed: Sep. 26, 2024.
[25] M. Drake, Encyclopedia of library and information science, vol. 1. CRC Press, 2003. Accessed: Sep. 26, 2024.
[26] “Query Reformulation Strategies | Elicit.” Accessed: Oct. 04, 2024.
[27] O. Kolomiyets and M.-F. Moens, “A survey on question answering technology from an information retrieval perspective,” Information Sciences, vol. 181, no. 24, pp. 5412–5434, 2011, Accessed: Sep. 26, 2024.
[28] J. Fukumoto, “Question Answering System for Non-factoid Type Questions and Automatic Evaluation based on BE Method.,” in NTCIR, 2007. Accessed: Sep. 26, 2024.
[29] L. V. Lita, “Instance-based question answering,” Dissertation Abstracts International, vol. 68, no. 01. 2006. Accessed: Sep. 26, 2024.
[30] P. Gupta and V. Gupta, “A survey of text question answering techniques,” International Journal of Computer Applications, vol. 53, no. 4, 2012, Accessed: Sep. 26, 2024.
[31] D. Melo, I. Pimenta Rodrigues, and V. Beires Nogueira, “A review on cooperative question-answering systems,” 2013, Accessed: Sep. 26, 2024.
[32] S. K. Dwivedi and V. Singh, “Research and reviews in question answering system,” Procedia Technology, vol. 10, pp. 417–424, 2013, Accessed: Sep. 26, 2024.
[33] A. Mone, I. Mete, P. Gangarde, and M. Kharad, “Automatic Answering System for English Language Questions,” Information Processing & Management, vol. 23, no. 5, pp. 495–505, 1987, Accessed: Sep. 26, 2024.
[34] D. Dominguez-Sal and M. Surdeanu, “A machine learning approach for factoid question answering,” Procesamiento del Lenguaje Natural, no. 37, pp. 131–136, 2006, Accessed: Sep. 26, 2024.
[35] H. T. Ng, L. H. Teo, and J. L. P. Kwan, “A machine learning approach to answering questions for reading comprehension tests,” in 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 2000, pp. 124–132. Accessed: Sep. 26, 2024.
[36] R. Poonguzhali and D. K. Lakshmi, “Evaluating the Performance of Recurrent Neural Network based Question Answering System with Easy and Complex bAbI QA Tasks,” International Journal of Advanced Science and Technology, vol. 29, no. 5s, Art. no. 5s, Apr. 2020, Accessed: Sep. 26, 2024.
[37] “Understanding LSTM Networks -- colah’s blog.” Accessed: Sep. 26, 2024.
[38] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language

Processing (EMNLP), Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1746–1751. doi: 10.3115/v1/D14-1181.
[39] “FIKIR SETIE TEZERA (2).pdf.”
[40] N. Reimers and I. Gurevych, “Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks,” Aug. 16, 2017, arXiv: arXiv:1707.06799. Accessed: Oct. 06, 2024.
[41] S. Bashetty, K. Raja, S. Adepu, and A. Jain, “Optimizers in Deep Learning: A Comparative Study and Analysis,” IJRASET, vol. 10, no. 12, pp. 1032–1039, Dec. 2022, doi: 10.22214/ijraset.2022.48050.
[42] “(PDF) Deep Learning Techniques: An Overview.” Accessed: Oct. 06, 2024.
[43] G. Moges, “Semantic-Aware Amharic Text Classification Using Deep Learning Approach,” Addis Ababa University, 2020. Accessed: Sep. 26, 2024.
[44] “Automated Amharic News Categorization Using Deep Learning Models - Endalie - 2021 - Computational Intelligence and Neuroscience - Wiley Online Library.” Accessed: Sep. 26, 2024.
[45] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, “Dive into Deep Learning,” Aug. 22, 2023, arXiv: arXiv:2106.11342. Accessed: Sep. 26, 2024.
[46] C. Wang, P. Nulty, and D. Lillis, “A Comparative Study on Word Embeddings in Deep Learning for Text Classification,” in Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval, Seoul Republic of Korea: ACM, Dec. 2020, pp. 37–46. doi: 10.1145/3443279.3443304.
[47] Y. T. Tessema, “NEXT WORD PREDICTION FOR AMHARIC LANGUAGE USING BI-LSTM”.
[48] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research,” Journal of Management Information Systems, vol. 24, no. 3, pp. 45–77, Dec. 2007, doi: 10.2753/MIS0742-1222240302.
[49] T. D. Belay, A. A. Ayele, and S. M. Yimam, “The Development of Pre-processing Tools and Pre-trained Embedding Models for Amharic”, Accessed: Sep. 30, 2024.
[50] R. Anhar, T. B. Adji, and N. A. Setiawan, “Question classification on question-answer system using bidirectional-LSTM,” in 2019 5th International Conference on Science and Technology (ICST), IEEE, 2019, pp. 1–5. Accessed: Sep. 30, 2024.
[51] “Maslej-Krešňáková Et Al. - 2020 - Comparison of Deep Learning Models and Various Text Pre-Processing Techniques For The Toxic Comments C-Annotated | PDF | Artificial Neural Network | Deep Learning.” Accessed: Oct. 01, 2024.

Appendices

Appendix A

Amharic scripts

Appendix B

Stop words list

እኔ	በኋላ	ብቻ
የእኔ	ከላይ	የራሱ
እኔ ራሴ	ከታች	ተመሳሳይ
እኛ	ወደከ	ስለዚህ
የእኛ	ወደ ላይ	ይልቅ
የእኛ	ታች	እንዲሁ
እኛ ራሳችን	ውስጥ	በጣም
አንቺ	ውጭ	እ.ኤ.አ.
ያንተ	ላይ	ት
ራስህን	ጠፍቷል	ይችላል
እራሳችሁ	በላይ	ያደርጋል
እሱ	በታች	ብቻ
የእሱ	እንደገና	ዶን
ራሱ	ተጨማሪ	ይገባል
እሷ	ከዚያ	አሁን
የእሷ	አንድ ጊዜ	ጥቂቶች
እራሷ	እዚህ	ተጨማሪ
እነሱ	እዚያ	በጣም
እነሱን	መቼ	ሌላ
የእነሱ	የት	አንዳንድ
ራሳቸው	እንዴት	እንደዚህ
ምንድን	እንዴት	አይ
የትኛው	ሁሉም	ወይም አይደለም
ማን	ማንኛውም	
ይህ	ሁለቱም	
የሚል ነው	እያንዳንዳቸው	
እነዚህ		
እነዚያ		