AMHARIC EXTRACTIVE TEXT SUMMARIZATION USING AmRoBERTa –BiLSTM MODEL

BY: EDEN AHMED

HAWASSA UNIVERSITY, HAWASSA, ETHIOPIA

October,2024

AMHARIC EXTRACTIVE TEXT SUMMARIZATION USING AmRoBERTa –BiLSTM MODEL

BY: EDEN AHMED

ADVISOR: MICHAEL MELESE (PhD)

A THESIS SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE

HAWASSA UNIVERSITY

INSTITUTE OF TECHNOLOGY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

HAWASSA, ETHIOPIA

October,2024

APPROVAL SHEET-I

This is to certify that the thesis entitled "**Amharic Extractive Text Summarization using AmRoBERTa-BiLSTM Model**" submitted in partial fulfillment of the requirements for the degree of Master's with specialization in Computer Science, the Graduate Program of the Department/School of Informatics, and has been carried out by Eden Ahmed. Therefore, we recommend that the student has fulfilled the requirements and hence hereby can submit the thesis to the department.

Michael Melese (PhD)	_____	_____
Name of major advisor	Signature	Date

EXAMINERS' APPROVAL SHEET

SCHOOL OF GRADUATE STUDIES

HAWASSA UNIVERSITY EXAMINERS' APPROVAL SHEET

We, the undersigned, members of the Board of Examiners of the final open defense by Eden Ahmed have read and evaluated his/her thesis entitled "Amharic Extractive Text Summarization using AmRoBERTa-BiLSTM Model", and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirements for the degree.

_____	_____	_____
Name of Major Advisor	Signature	Date
_____	_____	_____
Name of Internal Examiner-I	Signature	Date
_____	_____	_____
Name of Internal Examiner-II	Signature	Date
Wondwossen Mulugeta (PhD)	_____	_____
Name of External examiner	Signature	Date
_____	_____	_____
SGS Approval	Signature	Date

ACKNOWLEDGMENT

I would like to express my heartfelt gratitude to Almighty God and the Holy Virgin St. Mary, the mother of our Lord, for their blessings and unwavering support throughout my journey.

I am deeply grateful to my advisor, Dr. Michael Melese. His invaluable guidance, thoughtful feedback, and encouragement has been essential in shaping my research and helping me overcome the challenges I encountered.

I would also like to thank my beloved family for their constant love and encouragement. Their support has been a vital source of strength for me.

Additionally, I want to extend my appreciation to my beloved friends for their valuable advice and continuous support.

STATEMENT OF THE AUTHOUR

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

Name: Eden Ahmed Signature: _____

Place: Institute of Technology, Hawassa University, Hawassa

Date of Submission: October 14 ,2024

ABSTRACT

Extractive text summarization is a crucial task in natural language processing, allowing users to quickly grasp the main ideas of lengthy documents. The manual summarization process is often labor-intensive and time-consuming. As the volume of information in the Amharic language continues to grow, the need for effective summarization systems has become essential. While various summarization techniques have been developed for multiple languages, research specifically focused on Amharic remains limited. Most existing studies rely on traditional methods that often lack of contextual embeddings, which are crucial for understanding the meaning within the text. Additionally, current approaches often struggle to capture long-range dependencies among sentences and none of the existing studies have utilized hybrid deep models, which have demonstrated state-of-the-art performance in summarization tasks for other languages. This study addresses the challenge of extractive text summarization for Amharic news articles by proposing a hybrid deep learning model that combines the contextual understanding of AmRoBERTa with the sequential processing capabilities of Bidirectional Long Short-Term Memory. A dataset of 1,200 Amharic news articles, covering a variety of topics, was collected. Each article was segmented into sentences and labeled by experts to indicate their relevance for summarization. Preprocessing was conducted, including normalization and tokenization using AmRoBERTa, to prepare the data for modeling. The proposed model was trained using various hyperparameter configurations and optimization techniques. Its effectiveness was evaluated using ROUGE metrics. The results demonstrate that our model achieved significant performance, with a ROUGE-1 score of 44.48, a ROUGE-2 score of 34.73, and a ROUGE-L score of 44.47.

Key words: - Text Summarization, Deep Learning, Natural Language Processing,

LIST OF ABBREVIATIONS

ATS	Automatic text summarization
AmRoBERTa	Amharic Robustly Bidirectional Encoder Representation from Transformer
BERT	Bidirectional encoder representation from transformer
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
DL	Deep learning
HITS	Hyperlink-Induced Topic Search
IDF	Inverse Document Frequency
LSTM	Long short-term memory
LSA	Latent semantic analysis
MDS	Multi Document Summarization
NLP	Natural Language Processing
PDA	Personal Digital Assistant
RQ	Research Question
RNN	Recurrent Neural Network
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
SDS	Single Document Summarization
TF	Term Frequency

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.1. Background

Natural Language Processing is a field within artificial intelligence that emphasizes the interaction between computers and humans using natural language. The main goal of NLP is to develop systems that can efficiently process and analyze extensive amounts of natural language data. NLP allows machines to comprehend, interpret, and produce human language in a manner that is both significant and useful. This functionality is essential for numerous applications in our data-driven society, including text summarization, which enables users to condense large quantities of information into clear and concise summaries [1].

Since its introduction a decade ago, the internet has undergone remarkable changes and has become an integral part of our everyday lives. Each day, an immense volume of data is produced online. This exponential growth has created a critical need to quickly identify meaningful information within the vast quantities of data, prompted by the challenges of information overload [2]. Consequently, users spend considerable time searching for the information they need, often unable to read or comprehend all the text presented in search results. Many of these results contain repetitive or irrelevant sections. As a result, the need for summarizing and condensing textual resources has become increasingly urgent and significant. Manual summarization is not only time-consuming but also labor-intensive, making it challenging for individuals to effectively summarize such large volumes of text [3]. Automatic Text Summarization emerges as a vital solution to this problem.

The primary objective of a summary is to convey the key concepts of a document in a more concise form. Automatic Text Summarization (ATS) systems help users comprehend the core points of the original document without needing to read it in its entirety. These automatically generated summaries can significantly assist users, saving them both time and effort. According to [4], "summary" is defined as "a text that is produced from one or more texts, that conveys important

information in the original text(s), and that is no longer than half of the original text and usually significantly less than that."

Automatic text summarization has a wide range of valuable applications. For instance, it can be employed to optimize information for smaller mobile devices, such as PDAs, which require significant content reduction. Furthermore, search engines could index summaries rather than entire documents, which would decrease the resources consumed by indexing algorithms. This strategy also enhances the search engine performance by improving its ability to accurately address user queries [5].

Automated document summarization generates a concise version of the input document while maintaining its essential information. This process can be approached in two primary ways: [6] extractive and abstractive. Extractive methods create summaries by directly choosing the most relevant sentences from the original text. Conversely, abstractive methods strive to formulate new words and phrases, emulating human-like summarization. Furthermore, summarization techniques are classified by the number of documents involved: single-document summarization produces a summary from a single source, while multi-document summarization gathers information from multiple texts addressing the same subject.

Automatic Text Summarization is a particularly challenging task due to the complexity of the input text or documents. Amharic, a Semitic language commonly referred to as "Amarigna," is one of the most widely spoken languages in Ethiopia [7]. This paper presents a hybrid approach to Amharic extractive text summarization that leverages the strengths of AmRoBERTa, a state-of-the-art pre-trained language model for Amharic, and BiLSTM, a recurrent neural network architecture. By combining the contextual understanding capabilities of AmRoBERTa with the sequential modeling abilities of BiLSTM, this research aims to generate Extractive summaries of Amharic text.

1.2. Motivation of the Study

The Amharic language is one of the major languages in Africa and serves as the working language of the Federal Government of Ethiopia, being widely spoken across the nation. As a member of the Afro-Asiatic family, it belongs to the Semitic group and features its own distinct script [8]. With the rise of the Internet and an increase in mass storage media, the volume of electronic documents and textual data has surged significantly. This growth has been accompanied by a notable rise in the amount of online content from Amharic news sources. Unfortunately, the prevalence of repetitive and irrelevant information complicates users' ability to find the information they need, leading to wasted time and effort on unimportant material. To address this issue, there is a pressing need for the development of automatic text summarizers that can enhance system efficiency and effectiveness. Consequently, the researcher is motivated to create a text summarizer for the Amharic language.

1.3. Statement of the Problem

The rise of the internet and big data has resulted in a massive and exponential growth of information. As a consequence, people are now confronted with an overwhelming amount of textual material daily. This challenge is particularly significant for Amharic speakers, given the increasing number of Amharic news sources publishing content online. Readers often struggle to sift through large volumes of text to discern which pieces of information are genuinely useful. Therefore, robust automatic text summarization systems specifically designed for the Amharic language are highly needed.

Various researchers have conducted studies on automatic text summarization for several languages such as for English [9], [10] and for Arabic [11] using recent deep learning methods. In contrast, the study in Amharic extractive text summarization is using statistical methods such as TF-IDF [8] , [12], and Open Text Summarizer [13] , as well as techniques like latent semantic analysis [14], graph-based methods [15], NLP parsers [16], and some machine learning approaches. While these methods have contributed to the development of extractive text

summarization systems, there is a lack contextual embeddings, which are crucial for understanding the meaning within the text. Additionally, current approaches often struggle to capture long-range dependencies among sentences. Moreover, none of the existing studies have utilized hybrid deep models, which have demonstrated state-of-the-art performance in summarization tasks for other languages. These gaps are particularly critical in the context of extractive text summarization, where understanding the relationships between sentences is vital for accurately identifying and representing key information. To address these limitations and enhance the performance of Amharic text summarization, this research develops an extractive summarization method that uses a hybrid model combining AmRoBERTa and BiLSTM.

Research Questions

This study aims to explore and address the following research questions:

RQ1: How can the hybrid model (AmRoBERTa-BiLSTM) be effectively utilized for summarizing Amharic text?

RQ2: What hyperparameters significantly influence the performance of the AmRoBERTa-BiLSTM model in extractive summarization tasks?

1.4. Objective of the Study

1. 4.1. General Objective

The general objective of this research study is to design and develop automatic single document summarizer for Amharic news text using AmRoBERTa-BiLSTM hybrid model.

1.4.2. Specific Objectives

To achieve the above general objective, the following specific objectives of study are formulated.

- To review related research works in the area of text summarization and large language models.
- To collect, prepare, and preprocess a non-domain-specific Amharic news dataset.
- To design the general architecture of the AmRoBERTa-BiLSTM hybrid model for effective summarization.

- To train different configurations of the hybrid algorithms on the dataset.
- To assess the performance of the summarization model using suitable metrics.
- To report the findings of the study for future research in the area.

1.5. Significance of the Study

The study has the following benefits:

- It helps user to get most concise, fluent and useful information in a short time.
- The study would reduce the time and labor of the user by providing a short summary of Amharic document news.
- Various organizations and individuals who are users of news media benefit from the resulting research work which can be considered as a step towards producing high-performance Amharic document summarization system.
- It contributes towards the realization of robust Amharic document summarization systems.
- The study also be significant for future researchers who want to conduct their study in the area of automatic text summarization.

1.6. Scope and Limitation of the Study

1.6.1. Scope of the Study

The scope of this is to develop an automatic single document summarizer for Amharic news text using AmRoBERTa-BiLSTM hybrid model. There are different domains of text summarization. This study concerns the summarization of news articles only. In addition, as text summarization is a wide concept that includes multilingual summarization, this study is specifically for the Amharic language. Amharic text is an input to build the system. Moreover, the study focuses on a generic document summarization and an extractive text summarization approach.

1.6.2. Limitation of the Study

The study is limited with a constrained dataset of Amharic news articles due to time and resource restrictions. Additionally, the evaluation primarily focuses on intrinsic measures and does not address extrinsic qualities of the generated summaries.

1.7. Methodology

In order to achieve the main objective of this research, the following procedures were followed:

Literature Review

A comprehensive literature review was conducted to thoroughly examine the nature of summarization. This review encompassed existing methodologies, applications, and related works across various languages. Additionally, various resources were reviewed to enhance the understanding of the structure and characteristics of the Amharic language. This included an exploration of the writing system, word classes, and morphological features specific to Amharic. By examining these linguistic aspects, the review aimed to support the development of effective text summarization techniques tailored specifically for the Amharic language.

Research Design

The research methodology appropriate for this study is the experimental method. The main reason for selecting this method is its problem-solving nature. Using experimental research allows for the integration of theoretical frameworks with practical testing; every scientific theory requires experimentation to validate the functionality of the system. The theories adopted were based on the research questions that the study aimed to answer. Through testing, the research questions were systematically addressed.

Data Source

Amharic news articles were collected from various online news websites. The articles covered diverse domains, including sports, technology, business, politics, economics, and more. After gathering the documents, a preprocessing step was conducted to prepare the data for training and testing.

Tools and Techniques

The tools and techniques used in this study include programming languages, libraries, and frameworks that facilitate data processing and model training. Specific tools such as Python and TensorFlow were utilized for implementing the summarization model. Techniques for data preprocessing, including normalization, cleaning, and tokenization, were employed to ensure the dataset was suitable for analysis.

Evaluation

The evaluation of the summarization model was carried out using standard metrics to assess its performance. ROUGE scores were employed to measure the quality of the generated summaries compared to reference summaries. This evaluation process aimed to validate the effectiveness of the proposed summarization techniques.

CHAPTER TWO

LITERATURE REVIEW

2.1. Overview

This chapter provides an overview of the existing literature and related work on automatic text summarization. The discussion begins by exploring the fundamental concepts underlying text summarization, as well as examining the various types of ATS. Then, it delves into highlights its applications in various domains and approach of ATS based on distinct criteria. The chapter also examines the evaluation mechanisms used to assessed the quality of automatically generated summaries. Furthermore, this review explores existing ATS research, with a particular focus on work done in both foreign languages and local languages.

2.2. Automatic text summarization

Automatic text summarization involves generating a clear, fluent, and meaningful summary that encompasses the key information as thoroughly as possible [17]. The aim of an automatic text summarization system is to generate a summary that captures the main concepts from the input document concisely [4] while reduce repetition [18]. ATS systems help users in understanding the key idea of the original document without having to read the entire text [19]. This allows users to benefit from automatically generated summaries, saving them significant time and effort.

Maybury [20] described the automated summary as follows: “An effective summary distills the most important information from sources to produce an abridged version of the original information for a particular user and task”. [4] also characterized the summary in the following manner: “A summary can be loosely defined as a text that is produced from one or more texts, which conveys important information in the original text, and that is no longer than half of the original text and usually significantly less than that. Text here is utilized rather loosely and can involve to speech, multimedia documents, hypertext, and so on.” The resulting summary should

be shorter than the original text while maintaining the most essential detail contained within it [21].

The fundamental tasks that all summarization systems rely on have remained consistent: create an intermediate representation of the input text that highlights its main points, score the sentences according to this representation, and selecting a summary that consists of a specific number of sentences [22].

However, over the past few decades, both the approaches utilized in this discipline and the evaluation measures used to gauge the quality of summaries relative to human-written ones have undergone significant evolution. As technology has advanced, ATS has discovered various uses and has reemerged as a prominent area of research.

2.3. Applications of ATS

In the current fast-moving environment, getting up to date on the latest news is critical. However, because there is so much information available, it can be difficult to keep up with everything. This is where the art of summarizing news articles comes into play. Summarizing news articles involves distilling the key information from a news story into a more concise version that is easier to read. and digest. By summarizing news articles, readers can quickly gain an overview of the main points of a story and decide whether it is worth reading the complete article.

One of the key applications for developing summarization methods was to condense scientific texts [23]. The aim was to streamline the research process for readers by offering automatically generated abstracts that are valid and reliable, thereby reducing the manpower needed for manual abstract writing. These abstracts would remain unbiased, free from the abstractor's personal background and attitudes toward the topic, ensuring consistent quality.

In the current landscape, the expansion of social networks, email, blogs, and advancements in speech technology make the potential applications for summarization appear endless. One intriguing way summarization could enhance our daily conversations is by participating in ongoing discussions on forums, micro-blogs, or blogs. Rather than going through entire threads, summarization would allow us to quickly access the most important comments and dominant opinions, facilitating more informed contributions. This approach is also beneficial when joining email threads or discussions with multiple participants, as summarization tools can significantly

aid in making informed decisions, acquiring essential information, and keeping up with correspondence.

This would also be highly beneficial for businesses, particularly in situations where one needs to participate in a current meeting or email discussion. Other business applications might involve maintaining corporate memory, improving decision-making efficiency, and analyzing employee performance and communication trends. Summarization can streamline the flow of internal documents, facilitating the assessment and reuse of relevant information on specific topics and enabling quick report generation. In the realm of banking firms and stock trading, summarization tools can significantly improve the analysis of financial reports and market updates, making them a valuable resource for these companies.

ATS can be extremely beneficial for marketing by summarizing extensive online conversations, which aids in examining broad trends. Today, platforms like micro-blogs and Twitter have emerged as key venues for news sharing and opinion exchange, and summarizing these discussions can offer valuable insights into public sentiment [24].

2.4. Types of summaries

Summaries can be classified based on various criteria, such as the nature of the input size, approach, output summary, summary language, and content of the summary. Each of these categories is integrated to define a summarization task.

According to input size: ATS is categorized into single document and multi-document. Single Document Summarization focuses on a single text document to produce a condensed summary of its content while retaining the essential information [25]. On the other hand, Multi-Document Summarization aims to generate a summary from multiple input documents, eliminating redundant content [26]. Multi-Document Summarization is more complex than Single Document Summarization and encounters various significant challenges [27].

According to the approach: ATS is divided into extractive, abstractive, and hybrid methods [28]. In the extractive summarization method, the most important sentences are selected from the input document to create a summary by joining these sentences. Conversely, the abstractive summarization method transforms the input document into an intermediate representation, from

which the summary is generated. Unlike extractive summaries, abstractive summaries use sentences that are different from those in the original document. Hybrid summarization combines both extractive and abstractive techniques to create the summary.

According to the nature of the output: ATS can be grouped into generic or query. A generic text summarizer identifies and extracts essential information from one or more text to offer an overall understanding of their contents [29]. Conversely, query-based summarization utilizes a multi-document summarizer that operates on a set of related documents chosen from a larger corpus according to a specific query [30]. The resulting summary emphasizes content that is relevant to that query. A generic summary offers a more comprehensive view of the document's content than a query-based summary, which only displays the data most pertinent to the initial search [31].

According to the nature of the summary language: ATS is categorized into three: monolingual, multilingual, and cross-lingual. when both the input and output documents are in the same language it is consider as monolingual. It is classified as multilingual when the input text contains multiple languages, and the summary is produced in those same languages. Conversely, a cross-lingual summarization arises when the input text is in one language, and the summary is generated in a different language [32].

According to the nature of the summarization algorithm: ATS categorized into two types based on the summarization algorithm: supervised and unsupervised. The supervised approach necessitates a training phase that depends on annotated data, requiring human effort for labeling. In contrast, the unsupervised algorithm does not involve a training step or the need for training data [33].

According to the summary content: ATS is classified as either Indicative or Informative. An indicative summary offers a broad overview of the source material, assisting users in grasping its general scope and determining whether to engage with the complete document. In contrast, an informative summary distills the key information and concepts from the original text, aiming to communicate the main ideas without exploring intricate details [34]. Its goal is to convey the main contents without delving into details [35].

According to the summarization domain: ATS is categorized as either general or domain-specific. A general ATS system summarizes documents across multiple domains. Conversely, a domain-specific ATS system is designed to summarize documents within a particular field.

2.5. Word Embedding Techniques

Word embeddings, commonly referred to as word vectors, are a method employed to express words and documents numerically. This approach allows words with similar meanings to share similar representations in a vector space. By transforming words into a lower-dimensional framework, word embeddings are capable of effectively capturing semantic relationships and approximate meanings [36]. The most widely used word embedding techniques for extracting features from text are Bag of Words, Term Frequency-Inverse Document Frequency (TF-IDF), Word2Vec, GloVe embeddings, FastText, and the latest model, BERT.

Bag of Words

Bag of Words is a basic method used for representing text documents as numerical vectors. The approach involves counting the occurrences of each word in a document by considering it as a bag, or collection, of words. This method provides an easy way to convert text into vectors while disregarding the order of the words [37].

Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a machine learning technique utilized for word embedding in text analysis. It consists of two components: term frequency (TF) and inverse document frequency (IDF). This method relies on a statistical approach to determine the relevance of words within a text, which can be applied to either a single document or a collection of documents known as a corpus. Term Frequency measures how frequently a term occurs within a document and Inverse Document Frequency assesses the significance of a term across all documents in the corpus.

TF-IDF = TF × IDF

The TF-IDF score is utilized to evaluate the significance of words in a document, where higher scores signify greater importance.

Word2vec

Word2Vec is a technique for representing individual words as fixed-length vectors, enabling the encoding of semantic relationships among them. This vector-based representation captures the meanings of words within a particular context. Word2Vec utilizes two approaches: Continuous Bag of Words (CBOW) and Skip-gram. In the Skip-gram model, the context words are predicted based on a given target word, whereas in the Continuous Bag of Words model, the target word is inferred from the surrounding context information [38].

GloVe

GloVe, which stands for Global Vectors for Word Representation, is an unsupervised learning technique created at Stanford University that aims to produce word embeddings by integrating global word co-occurrence matrices from a corpus. The primary objective of GloVe embeddings is to analyze the statistical relationships between words. Unlike the occurrence matrix, which counts individual word frequencies, the co-occurrence matrix provides insights into how often specific pairs of words appear together. Each entry in this matrix signifies a pair of words that frequently co-occur. The GloVe model utilizes a word context matrix and employs matrix factorization in its construction [39].

Fast Text

FastText is a word embedding method developed to tackle the difficulties associated with rare and out-of-vocabulary words, which are often challenging for Word2Vec. Unlike Word2Vec, FastText leverages subword information to generate its word embeddings. It functions at the character level, enabling it to capture the internal structure and morphological characteristics of words. In this approach, each word is represented as the aggregate of its subword vector representations. This feature allows FastText to effectively handle rare and out-of-vocabulary terms, making it a powerful tool for language modeling and a variety of NLP [40].

BERT Embedding

BERT is a transformer-based language representation model trained on a diverse and large corpus. Unlike ELMo, which generates representations by pre-training bidirectional language models that combine left-to-right and right-to-left contexts, BERT utilizes a masked language model to predict words that are randomly concealed within a sequence. Furthermore, it includes a next-sentence prediction task to capture the relationships between sentences. Through fine-tuning, BERT has achieved state-of-the-art performance across various NLP tasks. In contrast to traditional word embeddings like Word2Vec or GloVe, which provide a single representation for each word, BERT generates distinct embeddings for each word depending on its surrounding context. This capability allows BERT to capture nuanced meanings and relationships in language more effectively [38].

AmRoBERTa Embedding

AmRoBERTa is a pre-trained language model specifically designed for the Amharic language. It is based on the Robustly Optimized BERT Pretraining Approach, which is a state-of-the-art language model for English. AmRoBERTa embeddings is their ability to capture contextual understanding. This means that the embeddings take into account the surrounding words and phrases when representing a particular word. This is crucial for many NLP tasks, as it allows the model to understand the meaning of a word in the context of a sentence or document.

2.6. Approaches to Text Summarization

Text summarization have two approaches: extractive and abstractive. Each types employs various methods. To generate a more accurate summary, a range of methodologies is utilized, including statistical, graph, machine learning, and deep learning techniques.

Statistical Based Methods

These approaches identify significant sentences and words from the original text through the application of statistical analysis on various characteristics. A sentence is deemed "most significant" based on criteria such as its "optimal placement" and "frequency of occurrence." The scoring mechanism employed by a statistical-based extractive summarizer involves two main phases: 1) the selection and computation of pertinent statistical and/or linguistic features, along

with the assignment of weights to these features, and 2) the calculation of a final score for each sentence present in the document [41].

Topic Based Methods

These approaches focus on determining the main subject of a document. Frequently utilized techniques for topic representation encompass TF-IDF, Term Frequency, lexical chains, and topic word methodologies, where the topic is displayed in a clear table alongside its associated weight. The process of a topic extractive summarizer involves converting the input text into an intermediate representation that encapsulates the topics addressed, subsequently assigning an importance score to each sentence within the document based on this representation [42].

Graph Based Methods

These approaches are affected by the PageRank algorithm, represent documents as interconnected graphs. In this representation, sentences act as the vertices, and edges between them indicate the similarity between those sentences. A common technique for establishing connections between vertices is to assess the similarity of two sentences; if this similarity exceeds a certain threshold, an edge is created. Cosine similarity, often calculated using TF*IDF weights for words, is the most frequently used method for this measurement. This graph representation produces two primary results: first, the partitions within the graph reveal the different topics addressed in the documents; second, it facilitates the identification of significant sentences. Sentences that have numerous connections to other sentences within a partition are likely to be pivotal within the graph, increasing their chances of being included in the summary. Additionally, because this method does not rely on language-specific linguistic analysis, it is applicable across various languages [43].

Graph-based methods are applicable for both SD and MD summarization. Since they only require basic language processing, such as detecting sentence and word boundaries, they can be used with a variety of languages.

Latent semantic analysis (LSA) method

This approach has developed within the realm of natural language analysis to investigate the the relationships between documents and the terminology they contain by creating related concepts [44]. Latent Semantic Analysis operates on the premise that terms with similar meanings tend to appear in similar contexts. A matrix is created from a substantial corpus of text, with rows

corresponding to unique words and columns representing each paragraph, capturing word frequencies. A mathematical technique referred to as singular value decomposition is utilized to decrease the number of columns while maintaining the similarity structure among the rows. The comparison of words is achieved by computing the cosine of the angle between two vectors derived from any two rows. Values approaching 1 signify a high degree of similarity between words, whereas values nearing 0 indicate considerable dissimilarity. LSA consists of three primary steps: constructing the input matrix, executing singular value decomposition, and selecting sentences [45].

Machine Learning Based Methods

Machine learning is a domain within artificial intelligence and computer science that focuses on utilizing data and algorithms to enable AI systems to mimic human learning processes. This approach allows the systems to improve their performance over time, gradually increasing their accuracy and effectiveness in various tasks [46].

Machine learning operates through three primary components.

1. **Decision Process**: Machine learning algorithms are designed to make predictions or classifications based on input data, which may be categorized as either labeled or unlabeled. These algorithms produce estimations concerning the patterns identified within the data.

2. **Error Function**: This function evaluates the predictions made by the model by contrasting them with established examples, thereby facilitating an assessment of the model accuracy.

3. **Model Optimization Process**: In order to enhance the model alignment with the training data, the algorithm adjusts weights to minimize discrepancies between actual examples and its estimates. This iterative process of evaluation and optimization continues automatically until the model reaches a predefined level of accuracy.

Machine Learning Methods

Machine learning methods can be categorized into three main types:

Supervised

Supervised learning employs labeled datasets to train algorithms for precise classification or prediction of outcomes. As data is input into the model, it adjusts its weights to achieve a proper fit. This adjustment is part of the cross-validation process, which helps prevent issues like overfitting or underfitting. Supervised learning is effective for addressing various real-world challenges, such as automatically filtering spam emails. Common techniques in supervised learning contain logistic regression, support vector machines, neural networks, linear regression, naïve Bayes and random forests [46].

Unsupervised

Unsupervised learning refers to the application of algorithms that analyze and categorize unlabeled datasets into distinct groups known as clusters. These algorithms are capable of detecting underlying patterns or associations within the data without the need for human intervention. This ability to reveal similarities and differences renders unsupervised learning especially valuable for tasks such as exploratory data analysis, customer segmentation, cross-selling strategies, and image or pattern recognition. Furthermore, it can assist in minimizing the number of features in a model through techniques for dimensionality reduction. Prominent methods in this area include Singular Value Decomposition and Principal Component Analysis. Additionally, other frequently utilized algorithms in unsupervised learning encompass neural networks, k-means clustering, and probabilistic clustering techniques [46].

Semi-Supervised

Semi-supervised learning strikes a balance between supervised and unsupervised learning. It employs a smaller labeled dataset to assist in classifying and extracting features from a larger, unlabeled dataset during training. This approach addresses the challenge of insufficient labeled data for supervised learning algorithms, especially when labeling data is too expensive or impractical [46].

Common machine learning algorithms

Neural Networks

Neural networks mimic the functioning of the human brain through a vast network of interconnected processing nodes. They excel at pattern recognition and are crucial in various applications.

Logistic Regression

Logistic regression is a supervised learning algorithm designed to predict categorical response variables, such as binary outcomes like "yes" or "no." This technique is applicable in scenarios such as spam classification and quality control in manufacturing processes.

Linear Regression

Linear regression is an algorithm used to forecast numerical values by establishing a linear relationship among various inputs.

Decision Trees

Decision trees are versatile tools used for both numerical predictions (regression) and categorical classification. They operate through a series of branching decisions that can be visually represented in a tree diagram. One of the key benefits of decision trees is their transparency, making them easy to validate and audit, in contrast to the opaque nature of neural networks.

Clustering

Clustering, an unsupervised learning technique enables algorithms to detect patterns in data for grouping purposes. These algorithms assist data scientists by highlighting distinctions between data points that might be missed by human analysis.

Deep Learning Based Methods

Information-driven ATS can improve in effectiveness, accessibility, and ease of use by utilization of deep learning models. These models hold great potential for ATS because they aim to replicate the functioning of human brain. Deep neural networks are often utilized in NLP tasks Because of their architecture, which aligns well with the intricate nature of language. For example, each layer of the network can carry out a particular task and transmit the results to the subsequent layer [47]. One of the primary applications of DL methods is to uncover hidden structures and features at various stages of abstraction, that are valuable for NLP tasks. DL algorithms include recurrent neural networks, LSTM, transformers, BERT, etc.

Artificial Neural Networks

Artificial neural networks serve as a foundation for the wide range of ML algorithms. The primary goal is to train a network of interconnected units, commonly known as neurons, to detect patterns in input data and generate predictions based on those patterns. These neurons are arranged into layers input, hidden, output and interconnected by different pathways. Each connection between neurons has a specific weight that influences the strength of the connection. Neural networks can vary in terms of the number of neurons, the arrangement and number of layers, and the pathways by which information is transmitted [48].

Training a neural network involves determining the optimal weights and biases for the connections within the network to maximize prediction accuracy for a specific task. mostly Most networks are trained using supervised learning, where the network is provided with input data alongside the true value of the expected output. loss function is employed to calculate the network's loss, which measures the difference between the predicted output and the actual output value for the provided input data [49]. This process is known as forward propagation. The selection of the loss function is significantly determined by the specific learning problem being addressed, such as whether it involves binary classification, multi-class classification, sequence prediction, or other tasks.

Convolutional Neural Networks

Convolutional Neural Networks are a prominent and effective algorithm within deep learning [50]. Their key advantage over traditional approaches lies in their capability to automatically identify essential features in data, eliminating the need for manual feature selection. This makes CNNs highly adaptable and efficient for a wide range of applications, such as image and audio analysis, as well as face recognition. Like traditional neural network, CNNs are modeled after the way neurons function in the brain [50]. For example, when processing an image, a CNN assigns weights to various elements within the image, which helps it differentiate between different objects. This ability allows CNNs to capture pixel relationships and grasp the overall context of a scene, making them less reliant on pre-processing steps than other methods.

Recurrent Neural Networks based Models

Recurrent Neural Networks are widely used architectures for processing sequential data, such as natural language. RNNs are organized as a series of feedforward networks where the output information at a specific time step t^i is stored in a hidden state h_t then passed as input to the next time step t^{i+1}. The key idea is that hidden state retains information from previous parts of the sequence, allowing the RNN to make predictions based on earlier context.

However, while this approach enables the network to remember information from previous states, RNNs struggle with learning long-term dependencies in sequential data [51]. his limitation arises from the variant of the backpropagation algorithm used for calculating gradients in RNNs, known as backpropagation through time, which encounters issues with vanishing and exploding gradients [52]. These problems can result in inaccurate weight adjustments, leading to difficulties in learning [53].

To retain information for either a short or long duration, the RNN possesses its own memory.

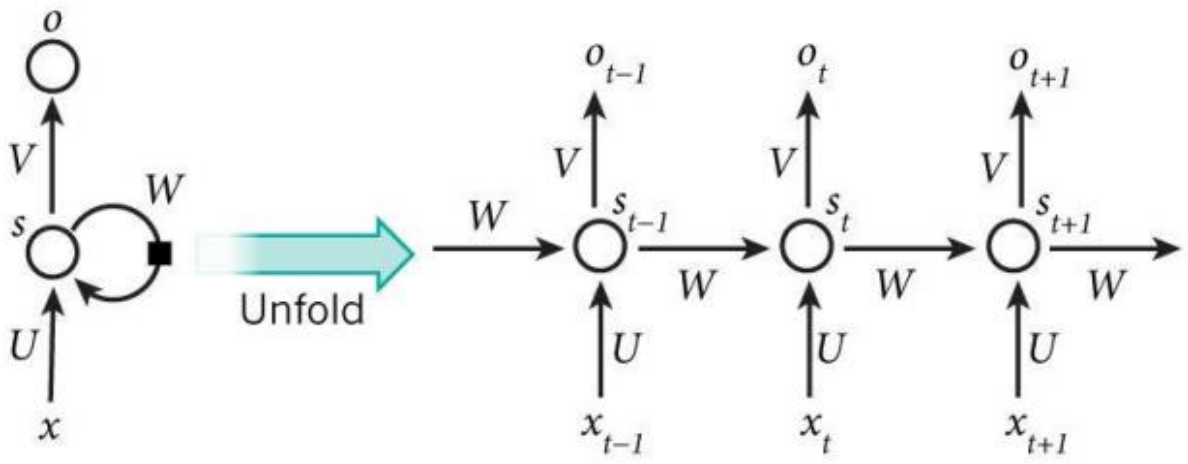

FIGURE 1: RNN UNFOLDING IN TIME FOR FORWARD COMPUTATION

This figure illustrates a recurrent neural network and the temporal unfolding of its forward computation. In the diagram, x_t denotes the input at a specific time step t while s_t represents the hidden state at that same time step. The network's memory can be computed using the formula: $s_t=f(u_{xt}+w_{st}-1)$. Here, f represents an activation function, such as sigmoid, ReLU, or tanh. The output layer at time t is indicated by O_t. Notably, the RNN shares the same parameters across a sequence of layers [54].

Long short-term memory (LSTM)

LSTM represents a distinct architecture within the realm of RNN, specifically engineered to proficiently capture long-term dependencies present in sequential data. This architecture mitigates the issues of vanishing and exploding gradients that conventional RNNs face when processing lengthy sequences. LSTMs employ a complex internal framework that facilitates effective learning over prolonged durations [55].

LSTM comprises a four-part cell architecture, which includes the forget gate, input gate, cell state, and output gate. The forget gate assesses which information from the preceding hidden state h_{t-1} should be preserved by integrating it with a new weight vector. The input gate determines which components of the current input x_t are to be incorporated, also through a connection to a new weight vector. The cell state functions as the long-term memory, which is modified using an

activation function that reflects the choices made by the forget and input gates. The current hidden state ht is derived by applying a different activation function to the updated cell state in conjunction with a weight vector.

LSTM networks, gates serve as miniature decision-making components that regulate the information flow within the LSTM cell. These gates are responsible for retaining only relevant data in long-term memory while discarding irrelevant or obsolete information. LSTMs are capable of processing not only single data points but also complete sequences, including audio and video. For example, they are especially proficient in applications such as handwriting recognition, speech recognition, machine translation, and text generation [56].

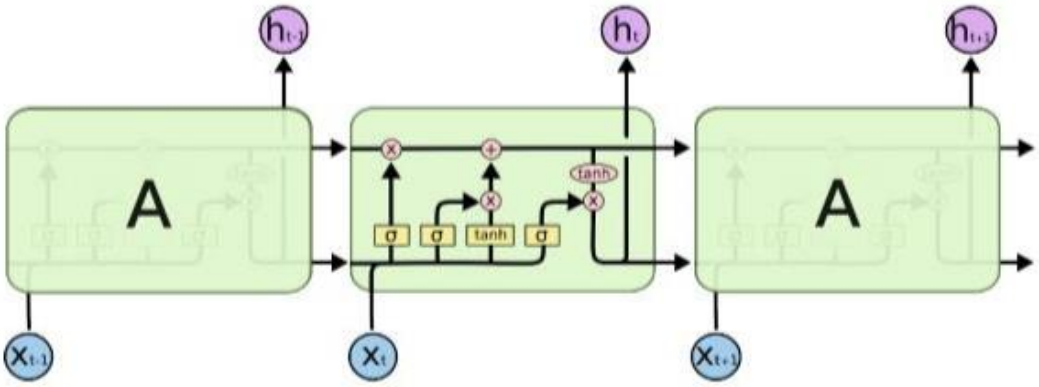

FIGURE 2: THE REPEATING MODULE OF AN LSTM CONSISTS OF FOUR INTERCONNECTED LAYERS.

Bidirectional Long Short-Term Memory

Bi-LSTM networks provide considerable advantages compared to traditional LSTMs in the realm of sequence prediction leading to enhanced accuracy and efficiency [57]. In contrast to standard LSTMs, which analyze data in a unidirectional manner, Bi-LSTMs perform two analyses: one progressing from the start to the end of the sequence and another retracing from the end to the start [57]. This bidirectional approach enables the network to leverage both past context (prior data) and future insights (upcoming data) during the prediction process, resulting in improved accuracy and effectiveness.

A notable feature of Bi-LSTM is its feedback input mechanism. Each layer not only receives data from the preceding layer but also from the corresponding layer during the reverse pass. This configuration facilitates a more comprehensive and expedited learning process, as the network can leverage both historical and forthcoming dependencies within the input sequence. In essence, Bi-LSTM ability to concurrently assess past data and imminent information offers a more enriched context for making predictions. By assimilating insights from the entire sequence, Bi-LSTM models can yield outputs that are both more accurate and contextually pertinent. Furthermore, the feedback loop improves the flow of information and enables the network to learn long-term dependencies with greater efficiency. As a result, the bidirectional processing and enhanced learning capabilities of Bi-LSTM establish it as a formidable tool for a variety of sequence prediction tasks, outpacing traditional LSTMs in accuracy and efficiency [57].

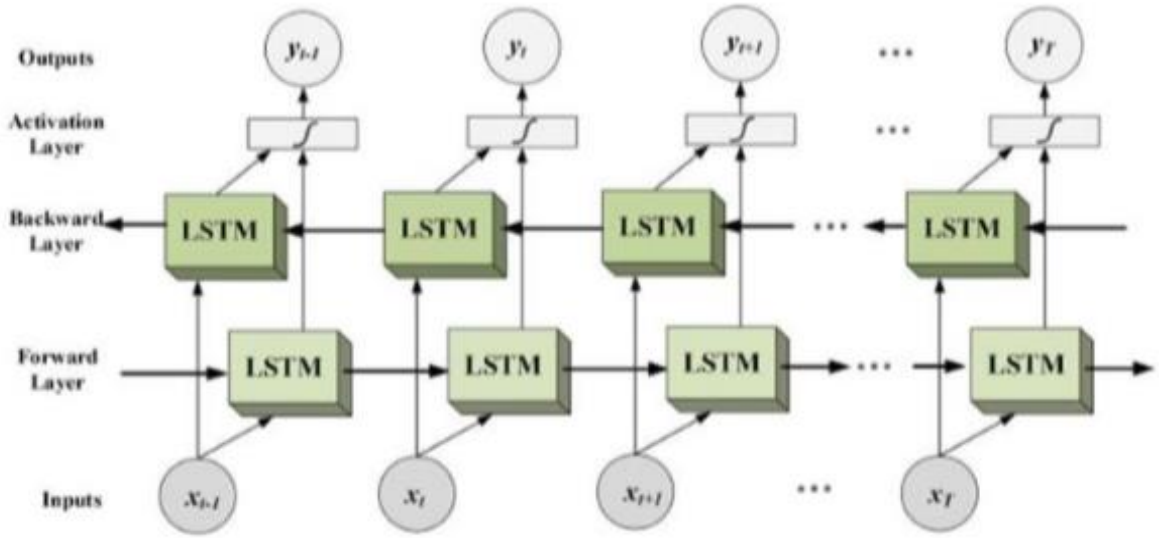

FIGURE 3: BILSTM ARCHITECTURE

Transformers

Transformers represent an architecture that relies on attention mechanisms and is composed of two main elements: the encoder and the decoder. This model was first presented by [58], To address the constraints of recurrent models, especially those that impede parallel processing, which leads to extended training durations and reduced efficiency for longer dependencies, transformers offer a solution. Their attention-based, non-sequential architecture allows for significant parallelization, maintaining a constant sequential and path complexity of O(1). Transformers are employed to tackle translation issues by emphasizing the relationships between words in an input sentence and integrating this with a pre-existing translation of the same sentence [59].

Encoder: The encoder comprises multiple layers, each consisting of two sub-layers. The first sub-layer includes a multi-head self-attention mechanism. Here, self-attention refers to the scenario where the target sequence matches the input sequence. In other words, self-attention is a specific type of attention mechanism that connects different positions within the same input sequence. The term "multi-head" signifies that rather than computing attention a single time, it employs scaled dot-product attention, enabling multiple attention calculations to occur simultaneously [58]. The second sub-layer is a straightforward position-wise, fully connected feed-forward network. Each sub-layer includes a residual connection and layer normalization. This design allows the output of each sub-layer to be combined with its previous input.

Decoder: The decoder is set up in a way that resembles the encoder, but it features an extra sub-layer called masked multi-head attention. This variant of multi-head attention prevents the model from attending to future positions, unlike self-attention. The masking is implemented to ensure that the predictions do not consider upcoming elements in the target sequence [58].

Bidirectional encoder representations from transformers (BERT)

BERT is a transformer-based model presented by [60]. The authors highlight that previous language representation models, such as RNNs, were limited in their token encoding by only considering tokens in one direction. Unlike RNNs, they utilize transformers to develop a Bidirectional Encoder Representation from Transformers, which can encode a token using information from both directions.

BERT was launched in 2018 and is capable of tackling various tasks, such as question answering, sentiment analysis, and text summarization. These tasks require a comprehensive understanding of language, which is facilitated through a two-step process: pre-training and fine-tuning. In the pre-training phase, BERT learns the intricacies of language, while the fine-tuning phase focuses on adapting BERT to perform specific tasks effectively.

Input and Output Embedding: - Like other language models, BERT converts each input token into a vector representation through a token embedding layer. In addition, BERT includes two other embedding layers known as segment and position embeddings. To form an input embedding, the token, segment, and position embeddings are combined for each token. Prior to being processed by the embedding layers, the input text is tokenized using WordPiece.

BERT utilizes WordPiece tokenization, introduced by [61], to improve the management of rare words. This method involves splitting words into sub-words using a predefined vocabulary set. For example, in the sentence "I went playing," if "playing" is absent from the vocabulary, it would be tokenized into the following components: {‘i’, ‘went’, ‘play’, ‘##ing’}. The hash symbols signify that this token is a continuation of a larger word.

BERT also incorporates two special tokens into the input: [CLS] and [SEP]. The [CLS] token is added at the beginning of the input sequence, and its output vector serves as a representation of the entire input sequence for classification tasks. The [SEP] token acts as a separator, enabling BERT to differentiate between pairs of input sentences, which is utilized in the pre-training tasks [62].

To format the data for the encoder layer, BERT constructs a vector for each token in the input by integrating three distinct types of embeddings: token embedding, segment embedding, and positional embedding. Token embedding serves as the vector representation for each word in the vocabulary. Segment embedding is utilized to differentiate which words are part of which sentence. Lastly, positional embedding, similar to the encoding used in the transformer model, encodes the relative position of each word within the sequence.

Pretraining & Fine Tuning

The BERT framework encompasses two separate phases: pre-training and fine-tuning. The pre-training phase is designed for the model to acquire a broad understanding of language, while fine-tuning involves adjusting the parameters learned during pre-training for a specific downstream task. BERT models undergo pre-training using two unsupervised tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NLP) [60].

Masked Language Modeling involves masking or hiding random words within a sentence. The model subsequently processes the output vector of the masked token and applies a softmax function over the vocabulary to predict the missing word. In the pre-training phase of BERT, 15% of all tokens were masked. This technique allows BERT to attain genuine bi-directionality, as it utilizes token embeddings from both the left and right contexts of the masked token while the sequence moves through each encoder layer [60].

Next sentence prediction This process occurs concurrently with the Masked Language Modeling (MLM) approach. However, it differs from MLM as it involves a binary classification task for sentence pairs. For a given pair of sentences (A, B), the model aims to predict whether sentence B follows sentence A. In 50% of cases, B is indeed the next sentence. In the remaining instances, a random sentence from the corpus is substituted. Through this training method, BERT gains an understanding of the relationships between sentences [60].

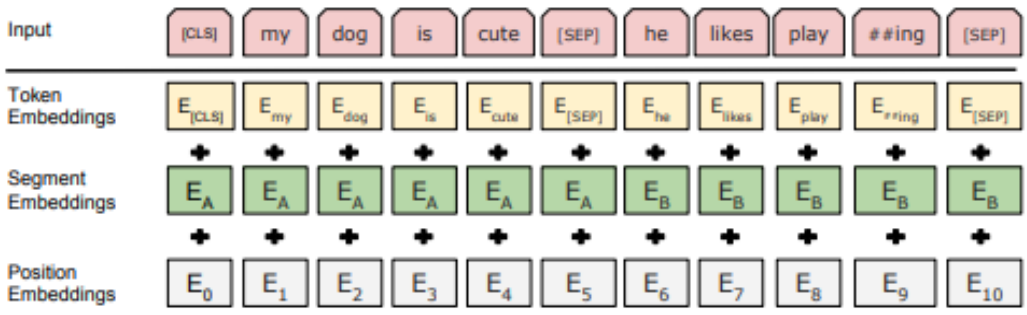

FIGURE 4: BERT INPUT REPRESENTATION

Robustly Optimized BERT Pretraining Approach (RoBERTa)

It is a state-of-the-art language representation model developed by Facebook AI. While it is built upon the foundational BERT architecture, it varies in several important aspects. Its goal is to enhance the original BERT model by increasing the model size, expanding the training dataset, and refining the training techniques to optimize the use of the Transformer architecture. This results in a language representation that is more expressive and robust, demonstrating state-of-the-art performance across numerous NLP tasks.

The RoBERTa model is composed of multiple self-attention and feed-forward layers. The self-attention layers enable the model to weigh the significance of various tokens in the input sequence and generate representations that consider the context of the entire sequence. The feed-forward layers are employed to convert the representations created by the self-attention layers into a final output representation [63].

Features of RoBERTa

Removing the Next Sentence Prediction (NSP): the Next Sentence Prediction task was eliminated in RoBERTa's development. In the NSP task, pairs of sentences were provided to the model, which had to determine whether the second sentence followed the first in the original text. However, it was found that the NSP task had a minimal impact on the model's performance in downstream tasks.

A Full-Sentence approach is utilized in RoBERTa, where sequences from multiple documents are packed together. Each input to the model consists of full sentences that are sampled consecutively from one or more documents, reaching a maximum sequence length of 512 tokens. This method enhances the model's ability to learn long-range dependencies more effectively.

Dynamically changing the masking pattern: dynamic masking is a pre-training technique that has been employed in some variants of RoBERTa to enhance performance on downstream NLP tasks. Unlike the static masking that is utilized in the original BERT model, which has the same tokens consistently masked during each epoch of pre-training, dynamic masking is characterized by the random selection of different tokens to be masked at various points throughout the pre-training process. This encourages the model to learn richer, context-aware representations that can differentiate between synonyms and ambiguous terms [64]**.**

2.7. Evaluation metrics for summarization

Evaluation is a vital aspect of any NLP application. As new and enhanced summarization techniques emerge, effective evaluation methods are essential for measuring the difference between human-created summaries and automatic summaries [65].

Evaluation measures can be classified into two main categories: intrinsic and extrinsic. Intrinsic evaluation methods focus on assessing the quality of the summaries produced, while extrinsic methods evaluate the summaries based on their effectiveness in the specific tasks for which they were generated [66]. In intrinsic evaluation, the automatically generated summaries are compared to human-created "gold" standard summaries, often referred to as reference summaries.

Intrinsic evaluation of text summaries can be divided into two primary categories: quality assessment and content assessment. The produced summary is emphasized for examination in quality assessment, focusing on various linguistic features such as grammatical accuracy, redundancy, clarity of references, and general structure and coherence. The content assessment approach is the method most frequently employed for summarization evaluation, which can be further split into two subcategories: co-selection and content-focused evaluation.

Co-selection measure

Co-selection evaluation methods are appropriate for summaries created by selecting sentences from the document being summarized, where the ideal summaries consist of extracted sentences from the original text. This category includes metrics such as precision, recall, and F-measure.

Precision is the proportion of sentences that are present in both the manually generated summary and the machine-generated summary, relative to the total number of sentences in the machine-generated summary. It can be expressed as:

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \ldots\ldots\ldots\ldots\ldots (1)$$

Recall is the proportion of sentences that are found in both the manually generated summary and the machine-generated summary, compared to the total number of sentences in the manually generated summary.

$$\text{Recall} = \frac{TP}{FN+TP} \qquad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Furthermore, The F1-score serves as a composite metric that combines both precision and recall. It is typically calculated as the harmonic mean of these two values [66].

$$F1 = \frac{\mathbf{2}*\boldsymbol{P}*\boldsymbol{R}}{\boldsymbol{P}+\boldsymbol{R}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

The co-selection measure offers the benefit that once a human judge has defined the ideal summary, it can be readily applied to evaluate automatic summaries through simple comparisons. However, a significant limitation is the challenge of establishing a gold-standard summary. The recall rate of a summary can fluctuate between 25% and 50%, depending on which of the two human-generated extracts is used for assessment. Consequently, this approach can result in two equally valid extracts being judged in contrasting ways [67].

Content-based Summary Evaluation

Content-based measures allow for the evaluation of summaries that may not exactly match the sentences in the ideal summaries. Relying solely on exact matches would be too limiting, as two sentences can convey similar information even if they are phrased differently. Various content-based similarity measures, such as cosine similarity and ROUGE, consider different textual properties to assess this similarity.

Cosine Similarity

Cosine similarity employs the vector space model to represent both the system-generated and ideal summaries. In this model, the axes correspond to the words found in either summary. As a result, each summary can be depicted as a point within this vector space.

ROUGE

ROUGE is a metric comprising several evaluation methods that assess the quality of automatically generated summaries compared to a reference summary. Among the frequently used methods for evaluating automatic summarization are ROUGE-1, ROUGE-L, and ROUGE-SU [66].

2.8. Introduction to Amharic Language

Amharic is the official working language of the Federal Democratic Republic of Ethiopia. It is often considered the only sub-Saharan language with its own unique writing system, positioning it as a significant ancient language in Africa. It is classified as one of the "Ethio-Semitic" languages, which are part of the Semitic branch of Afro-Asiatic languages. It shares connections with Arabic, Hebrew, Syrian and Aramaic. It is the second most widely spoken Semitic language, following Arabic and features its own unique writing system and numeral symbols. Amharic has served as a written language for around 600 years [13].

Amharic utilizes the Ge'ez script for writing, which comprises 33 basic characters. Each character has seven or eight variations, depending on the vowel sound to be pronounced in the syllable. It is one of the few languages in the world with its own indigenous script, even though it is derived from the Ge'ez script [68].

2.9. Amharic Writing System

Amharic is written using a system known as Fidel, which is adapted from the writing system of the Ge'ez language [69]. Amharic incorporates the entire Ge'ez alphabet, including redundant characters, into its writing system. For instance, the characters ሐ, ሀ, ኀ, and ኸ are pronounced as (hä), while ሰ and ሠ are pronounced as (sä). Similarly, አ and ዐ are both pronounced as (a), and ጸ and ፀ are pronounced as (tsä) [70].

It is organized in a tabular format with seven columns, where the first column displays the base characters and the subsequent columns show their derived vowel sounds. Additionally, there are approximately twenty characters representing labialized sounds, such as ሏ (lwa), ሟ (mua), and ሷ (sua). The Amharic alphabet consists of 33 base characters, each of which has six additional forms created by applying different vowels to the base character. The base characters use the vowel ኧ (ä), while the derived characters incorporate vowels in the following order: ኡ (u), ኢ (i), ኣ (a), ኤ (e), እ (), and ኦ (o). For example, starting with the base character ሰ (sä), the following six derived characters emerge: ሱ (su), ሲ (si), ሳ (sa), ሴ (se), ስ (s), and ሶ (so). With the exception of the two base forms (i.e., አ and ዐ), which represent the vowel sound (ä), the remaining 32 base characters consist of pairs of consonants and vowels. The vowels are not explicitly encoded; instead, they

function as modifiers of the base characters. As a result, these 33 basic characters, along with their six derived forms, yield a total of (33 + 6*33) syllable patterns [71].

2.10. Punctuation Marks and Numerals in Amharic Language

Amharic has its own distinct writing system, which includes unique numerals and punctuation marks [14]. In Amharic language, different punctuation marks are used for different purposes. The most known punctuations are:

TABLE 1: AMHARIC PUNCTUATION MARKS

Symbol	Punctuation name	Description
:	Colon	Used to separate two words
: :	Four dots	Acts as a sentence delimiter, equivalent to a period (.) in English.
?	Question mark	Indicates the end of a question.
!	Exclamation mark	Used to convey strong emotions, surprise, or emphasis; placed at the end of a sentence or phrase.
፣	Netela serez	Used to separate items or ideas, similar to a comma in English.
፤	Dirib serez	Functions like a semicolon in English, connecting related thoughts.
“ ”	Quotation marks	Used for denoting speech or quotations.
...	Three Dots	Functions similarly to an ellipsis in English, indicating an omission, an unfinished thought.
( )	Parenthesis	Used to enclose additional information or elaboration
፩-፼	Number	Ethiopic numeral symbols representing numbers from one to ten, twenty to ninety, and beyond.

2.11. Amharic Morphology

The Amharic language is known for its complexity and morphological richness, making it one of the most challenging languages. It demonstrates a root-pattern morphological structure, where a root consists of a set of consonants that convey a fundamental lexical meaning. A pattern involves a series of vowels inserted between the consonants of the root to create a stem. The verbal stems in Semitic languages, particularly in Amharic, follow a "root + vowels + template" structure. For example, the stem seber ('broke') is formed from the root verb sbr combined with ee and the CVCVC. Alongside these non-concatenated morphological features, Amharic employs various affixes to generate inflectional and derivational morphemes. Affixation can take the form of prefixes, infixes, suffixes, or circumfixes. The morphological complexity of the language can be better grasped by examining the formation process through inflection and derivation. Amharic nouns are inflected for number, definiteness, cases and gender. For instance, the noun ቤት (house) generates various forms through inflection and affixation, including ቤቶች (houses), ቤቱ (the house), ቤቴ (my house), and ቤትህ (your house) and others [72], [73].

Amharic adjectives can also be marked for number, definiteness, cases, and gender through an affixation process similar to that of nouns. The affixation of morphemes to indicate number aligns closely with nouns, though there are some variations in plural formation. In contrast, the inflections and derivations of Amharic verbs are even more complex than those of nouns and adjectives. A single verbal root can give rise to several verbs in their surface forms. Because verbs are marked for various grammatical units, one verb can effectively create a complete sentence. Furthermore, Amharic nouns can be derived from adjectives, verbal roots (by placing vowels between the consonants), stems, stem-like verbs, and other nouns [72].

There are only a few primary adjectives in the language that are not derived. However, numerous adjectives can be formed from nouns, stems, compound words, and verbal roots. Adjectives can

also be derived either from roots by intercalation of vocalic elements or attaching a suffix to bound stems [72], [73].

2.12. Amharic Word Classes

Amharic words are classified into five categories based on their morphological characteristics and their position within a sentence. These categories include nouns, verbs, adjectives, adverbs, and prepositions.

Nouns: Nouns are words that identify or name a class of people, places, things, ideas, etc., or any member of these classes [74]. In Amharic, nouns are categorized as either primitive or derived. Primitive nouns appear in their original form, whereas derived nouns come from different states or entirely different categories. For instance, the derived noun እግረኛ (pedestrian) originates from the primitive noun እግር (leg). A variety of morphemes, including '-ነት', '-ኛኛ', '-ኛት', '-ኣዊ', '-ተኛ', '-ኛ', and the prefix 'ባለ-', are used to form derived nouns [75].

Amharic has three sets of pronouns: independent pronouns, noun-possessive pronouns, and verb-object pronoun suffixes [76]. Amharic features three sets of pronouns: independent pronouns, noun-possessive pronouns, and verb-object pronoun suffixes. Independent pronouns can stand alone, replacing nouns that refer to people, places, or things. Examples include እሱ, እስዋ, እኔ, አንተ, and አንች. Noun-possessive pronouns are added as suffixes to the nouns they modify to indicate ownership. These pronouns distinguish between singular and plural forms, as well as masculine and feminine in the singular. For instance, የእኔ, የአንተ, and የእሱ illustrate this usage. Verb-object pronoun suffixes are attached to verbs to indicate the recipient of the action. Examples include ሰጠኝ, ሰጠህ, ሰጠሽ, and ሰጡት.

Verb: Any word that can appear at the end of a sentence and can take suffixes such as /ህ/, /ሁ/, and /ሽ/—which denote masculine, feminine, and plural forms—is classified as a verb.

Adjective: Amharic adjectives are always placed before the nouns or pronouns that they modify. In the example "ሰላማዊ ህዝብ," the adjective "ሰላማዊ" is used to describe the noun "ህዝብ." Most adjectives in Amharic are derived from nouns and verbs; however, some primitive adjectives, such as ባዶ, ብልህ, and ቢጫ, exist. The adjectives derived from nouns are formed by adding the suffixes (-እኛ), (-አዊ), (-አማ), and (-አም) [77], [78].

Adverb: An adverb is a word that qualifies a verb by adding additional context regarding time, place, or situation. While other words may appear between the adverb and the verb, the modified verb will immediately follow the adverb, coming before any other verbs in the sentence. For instance, in the phrase "እህቴ ዛሬ መጣች" (My sister came today), the word "ዛሬ" (today) specifies the timing of the action. Amharic adverbs are relatively limited in number and can be found in various forms. They may exist as primary words, as standalone terms that arise independently, or in compound forms, where two words combine to create a single expression [79].

Preposition: is a word that cannot accept any prefixes or suffixes, cannot be used to form other words, and does not carry meaning on its own. However, when used with nouns, it can convey various adverbial roles. Examples of such prepositions in Amharic include ከ, ለ, ወደ, ስለ, እንደ, and ወዘተ.

2.13. Related work

2.13.1. Foreign languages

Text Summarization for English

Liu [9] developed a single-document extractive summarization model called BERTSUM, which utilizes BERT, a pre-trained transformer model recognized for its effectiveness in various natural language processing tasks [60]. Since BERT functions as a masked language model, its output vectors correspond to tokens rather than complete sentences. To address this, the author adapted the input sequence and the embedding approach used in BERT. Liu employed interval sentence embeddings to capture multiple sentences within a document and opted for an inter-sentence transformer instead of a sigmoid classifier for processing the sentence representations. Additionally, the researcher incorporated an LSTM layer to enhance the outputs from BERT. The model was evaluated using the ROUGE metric on the CNN/DM corpus and the NYT dataset. The results from BERTSUM with Transformer on the CNN/DM dataset were impressive, achieving R-1: 43.25, R-2: 20.24, and R-L: 39.63. Overall, the models introduced by Liu outperformed previous studies, demonstrating significant advancements in extractive summarization.

Miller also employed a BERT model for extractive summarization [10]. The author acknowledged the significant role that summarization tools could play in educational contexts, particularly for summarizing lectures. In response to this need, Miller created a Python-based RESTful service

designed to facilitate lecture summarization. The study utilizes the BERT model to generate text embeddings and applies K-means clustering to identify the most pertinent sentences for inclusion in the summary.

Text Summarization for Arabic

Text summarization for Arabic was developed by [11], where the author introduced a novel approach that encompasses both abstractive and extractive summarization techniques using the BERT model. The study utilized KALIMAT, a comprehensive Arabic corpus articles along with their corresponding extractive summaries. Additionally, Multilingual BERT was employed in this research. The quality of the generated summaries was assessed using the ROUGE metric. The results indicated scores of R-1: 42.02, R-2: 24.59, and R-L: 41.99 for the extractive summarization method, while the abstractive approach yielded R-1: 12.21, R-2: 4.36, and R-L: 12.19. The researcher concluded that the use of pre-trained M-BERT enhances performance in both extractive and abstractive summarization, even when working with relatively small datasets.

Text Summarization for Danish

Automatic Text Summarization for Danish was developed by Nielsen and Veile [80]. They focused on implementing a systematic approach to automatic text summarization using BERT-based models, specifically targeting both extractive and abstractive summarization techniques. They utilized the DaNewsroom dataset, which consists of approximately 1.13 million Danish news articles, each paired with reference summaries categorized as extractive, abstractive, or mixed. The researchers employed two BERT models: DaBERT, a monolingual model pre-trained specifically on Danish text, and mBERT, a multilingual model trained on a diverse corpus of 104 languages. The models were fine-tuned for summarization tasks using approaches developed by Liu and Lapata (2019), adapting the pre-trained BERT models to the specific task of summarization.

For extractive summarization, the fine-tuning process involved a binary classification task where the model predicts whether a sentence from the source text should be included in the summary. In contrast, for abstractive summarization, the model generates new sentences that capture the essence of the original text, framed as a sequence-to-sequence problem where the input (source

text) is encoded into embedding vectors, and a decoder generates the output (summary). The results of the evaluation revealed that the mBERT model significantly outperformed the DaBERT model, particularly in abstractive and mixed summarization tasks. The following ROUGE scores were recorded: for the mBERT-Abstractive model, the scores were ROUGE-1: 48.33, ROUGE-2: 39.98, and ROUGE-L: 44.98. The mBERT-Extractive model achieved ROUGE-1: 43.93, ROUGE-2: 36.20, and ROUGE-L: 41.72, while the mBERT-Mixed model scored ROUGE-1: 49.99, ROUGE-2: 41.62, and ROUGE-L: 46.67.

Text Summarization for Norwegian

Moren developed effective extractive text summarization models for Norwegian news articles using two distinct approaches: a pre-trained Norwegian BERT model and a Multilingual BERT model [81]. The methodology involved utilizing a dataset sourced from Aftenposten, which provided the necessary labeled summaries for training the models. This dataset posed challenges due to its size, limiting the amount of training data available for the Norwegian language. The researchers aimed to explore how well these models could perform given the constraints of a low-resource language. They fine-tuned both the pre-trained Norwegian BERT and the Multilingual BERT models, with the Norwegian BERT designed to understand the nuances of the Norwegian language and the Multilingual BERT trained on a diverse set of languages, allowing it to leverage cross-lingual knowledge.

The models were evaluated using the ROUGE metric, which measures the overlap of n-grams between the generated summaries and human-written reference summaries, providing insights into the quality of the summaries produced. The evaluation results revealed that the pre-trained Norwegian BERT achieved a ROUGE-1 score of 35.62, ROUGE-2: 23.10, ROUGE-L: 30.49 while the Multilingual BERT outperformed it with a ROUGE-1 score of 39.60, ROUGE-2: 28.6, ROUGE-L: 34.75 indicating a significant gap in effectiveness. These findings suggest that while the Norwegian BERT is tailored for the language, the broader training of the Multilingual BERT provides significant advantages, particularly in scenarios where data availability is limited. The gaps identified in this study include limited data availability and the need for more comprehensive evaluation metrics.

Text Summarization for Somali

Text summarization for Somali was developed by the researcher [82] . To achieve this, the study utilized TF-IDF and BERT-SUM techniques to create an extractive summarization method for the Somali language. A total of 60 paragraphs were gathered from various sources, which were then summarized using both TF-IDF and BERT. The experimental results demonstrated reasonable outcomes. Ten evaluators were selected to assess the quality of the summarized texts. Based on their evaluations, the lowest score awarded was 70%, indicating a "Good" quality rating. The author noted that there had been no prior research conducted on this topic and highlighted the absence of lemmatizers or stemmers for the Somali language, which posed significant challenges for the task.

2.13.2. Local Languages

Text Summarization for Afan Oromo

The study [83] used deep learning approach to summarize Afan Oromo text. The primary objective of the study was to develop a system for implementing both extractive and abstractive summarization of Afan Oromo proclamation texts, aiming to create effective and efficient summarization methods while evaluating the suitability of various algorithms. The researcher gathered a dataset comprising twenty-seven Afan Oromo proclamations along with 583 related articles for the purpose of experimentation. For text summarization, the study employed abstractive models, specifically a Sequence-to-Sequence decoder with attention mechanisms, as well as extractive models such as Text Rank. To assess the quality of the generated summaries, various evaluation metrics were utilized including ROUGE-1 and ROUGE-2. The findings revealed that the ROUGE-1 and ROUGE-2 scores were higher for the abstractive summarization approach compared to the extractive method. The researcher concluded that the establishment of a standardized lexicon for the Afan Oromo language is essential for facilitating more effective text summarization in this language.

Afan Oromo news text summarization using sentence scoring method proposed by [84]. The researcher identified eight key features for this method: word frequency, thematic words, title words, the presence of capital letters, sentence position, numerical names, term weights, and cue

phrases. Trials were conducted on these features to assess their effects on the summarization system. The research utilized an extractive summarization approach, experimenting with ten topics selected from a list of thirty. The developed system calculates sentence scores by aggregating the scores of the individual words within each sentence, resulting in an overall sentence score. The summary is then generated by extracting the top n sentences based on three different extraction rates: 20%, 30%, and 40%. The findings indicated that the system's performance was notably lower when the sentence position feature was excluded. Additionally, the thematic words feature did not significantly influence the system's performance. However, it was observed that as the extraction rate increased, the system's performance also improved.

Afan Oromo text summarization utilizing word embeddings was developed by [85]. To create the summarizer, the researchers incorporated language-specific lexicons, including stop words and stemming techniques. A graph-based PageRank algorithm was employed to identify and select the most significant sentences from the document for inclusion in the summary. To assess the similarity between sentences, cosine similarity calculations were performed. Data for the study was gathered from both primary and secondary sources. The system was evaluated through three different experimental setups, encompassing both subjective and objective assessment methods. The system was evaluated through three different experimental setups, encompassing both subjective and objective assessment methods. The result of subjective evaluation is 83.33% informative, 78.8% referential integrity and grammar, and 76.66% structure and coherence. The result of objective evaluation also achieved 0.527 precision, 0.422 Recall and 0.468 F-measure.

Text Summarization for Tigrigna

Automatic Text summarization for Tigrigna language has been conducted by [86]**.** The researcher introduced two distinct methods for text summarization. The first method employed term frequency, which relies on the frequency of words to identify relevant sentences that contain frequently occurring terms. The second method focused on title words, identifying key words from the document's title and extracting sentences that include those words for the summary.

For experimentation, 30 news articles were collected from the Aiga Forum and Dmtsi Woyane Tigray websites. The results of the experiments indicated that the term frequency method achieved recall, precision, and F-Score values of 46% each. In contrast, the title word method reported recall at 46%, precision at 50%, and an F-Score of 48%. Overall, the findings revealed that the title word

method outperformed the term frequency method in both subjective and objective evaluations. In conclusion, the author emphasized the necessity of developing a standardized linguistic resource, or corpora, for Tigrinya text summarization to enhance performance evaluation outcomes.

Text summarization for Wolaita

Automatic text summarization for the Wolaita language was developed by [87]. The study introduced a graph-based extractive summarization method to tackle the challenges of condensing documents that contain extensive information. This process can be particularly very difficult and time-consuming, especially for less resourced and technologically unflavored languages. 92 documents were gathered, and various experiments were done on the documents. In addition, text preprocessing and page ranking algorithms were used. The experiment yielded values of 61.16% recall, 60.69% precision, and 60.46% f-measures. According to the researcher, the experiment's findings reveal encouraging results in the summarization of the Wolaita text. Finally, the researcher advised future researchers to conduct the study using deep learning to achieve the best results.

Text Summarization for Amharic

Kamil nuru [12] created the first summarization system for single documents in Amharic by employing statistical methods. The system utilized various features to assign weights to sentences, including titles, title words, leading sentences, keywords from head sentences, concluding sentences, initial sentences of paragraphs, cue phrases, and high-frequency keywords. Each of these features was associated with a specific weight, determined through training on manually created summaries of four news articles. These weights were then aggregated to generate an overall score for each sentence. Subsequently, the sentences with the highest scores were selected to construct the summary. The results indicated that the system achieved a precision of 70.4% and a recall rate of 58%, while effectively reducing the news content to 38.5% of its original size.

The author [88] proposed automatic text summarization using Naïve Bayes. The research had two steps. Those were training and testing. 480 Amharic news articles were used for experimentation. For initial experiment used a randomly selected 100 document/extract pairs and main experiment with the remaining 380 pairs. The training features included the presence of title words, positional indicators (such as first, middle, and last sentences), and frequent keywords. Summarization was

carried out with a 30% extraction rate. The system achieved a precision of 0.8, a recall rate of 0.833, and an F-measure of 0.816. The researcher proposed that utilizing a combination of multiple features could lead to improved performance in accurately predicting which sentences should be included in the summary.

The researcher [14] presented Amharic text summarization using latent semantic analysis. Two methodologies were used in their studies. These are LSAGraph and TopicLSA. In this approach, each text was modeled as a graph, where nodes represented terms and edges indicated the semantic relationships between phrases. The GB approaches were applied to the graph successively until they converged to obtain the relevance score of words

The author [13] developed Amharic multi-document text summarization using open text summarizer. The researcher used this open-source program after redesigning and modifying it, and the reason for doing so is that it supports more languages with some modifications. 35 single documents Amharic text news were collected form Addis Zena, Addis Admass, Walta information Centre, and Ethiopian news reporter and from this text news 10 multi document text news was organized and formulated. The experimentations were conducted with extraction rate of 20%, 30% and 40% for each of Amharic multi document text news. The performance of the summarization system was assessed using intrinsic evaluation techniques, which included both subjective and objective methods. The results from these evaluations indicated that the summarizer performed optimally at an extraction rate of 40%.

The study [16] proposed domain based single and multiple document summarization methods for Amharic text for both extractive and abstractive approaches using an NLP parser. According to the study, in the preprocessing stage, sentence segmentation, tokenization, part of speech tagging, named entity detection, and relation detection were performed, and lexical, syntactic, and semantic parsing were used in the analysis stages. Three evaluation metrics, precision, recall, and f-measure, were computed. However, the specific quantitative values for all the metrics were not provided. They suggest creating a summarization model that encompasses various document types, including images, videos, and other formats, in addition to text documents.

The researchers [15] conducted a generic and domain independent graph-based model for automatic single document summarization. They used PageRank and HITS link analysis algorithms with two sentence centrality measures: cumulative sum and discounted cumulative sum, for exploiting the relation between sentences in a text and nodes in a graph. 12 experiments have been conducted by pairing the link analysis algorithms with sentence centrality measures to provide system summaries with compression rates of 10%, 20%, and 30%. The researcher obtained 0.632 F-measure at a compression rate of 20% and a 0.697 F-measure at a compression rate of 30%.

The work [8] developed an Amharic text summarization system for news items shared on social media. Three components were employed to summarize the posted news items. First, the researchers calculated the similarity between each document within two pairs of sentences. Second, they grouped the documents using the K-means algorithm based on the similarity results. Third, they summarized the clustered documents individually using the TF-IDF algorithm. A total of 4,951 posted documents related to protests, droughts, sports, and floods from Twitter and Facebook were prepared for experimentation. In the first experiment, the clustered group of protest posts achieved the highest F-score of 87.07% for an extraction rate of 30%. In the second trial, the drought post groups reached an F-score of 84% at the same extraction rate. The third experiment with sports posts yielded the highest F-score of 91.37%, while the fourth trial for summarizing post texts achieved an F-score of 93.52% for an extraction rate of 30%. The researchers suggested using a dictionary file and additional lexicon rules for the Amharic language to address issues of over- and under-stemming of Amharic terms.

Executive Summary

Generally, most of the literature tries to develop Amharic extractive text summarization but there is a gap in solving the problems of text summarization. Most existing studies rely on traditional methods that often lack of effective embeddings, which are crucial for understanding the contextual meaning within the text. Additionally, current approaches often struggle to capture long-range contextual information and none of the existing studies have utilized hybrid deep models, which have demonstrated state-of-the-art performance in summarization tasks for other languages. However, recent advancements in natural language processing have introduced powerful contextual embedding models like AmRoBERTa, which have demonstrated significant improvements in text summarization tasks for various languages. AmRoBERTa ability to effectively understand the context of words and sentences makes it a promising tool for Amharic text summarization and BiLSTM layer captures both forward and backward dependencies in the sequence, which is particularly important for summarization tasks where the meaning of a sentence can depend on both preceding and succeeding words. This study integrates AmRoBERTa with BiLSTM. This hybrid approach leverages the strengths of both components: AmRoBERTa excels at capturing semantic meaning, while BiLSTM effectively handles sequential processing.

In addition, this study is the first to explore the potential of AmRoBERTa and BiLSTM hybrid models for Amharic extractive text summarization. By leveraging the strengths of AmRoBERTa and BiLSTM, the study aims to develop an effective solution for summarizing Amharic news articles.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1. Overview

In this chapter, we discussed the research methodology part for developing the extractive text summarizer model specifically designed for Amharic news articles. The methodology leverages the combination of AmRoBERTa and BiLSTM model for effective text summarization. We present the architecture of the proposed system and detailing the individual components. These components are essential for transforming raw Amharic texts into concise summaries that capture the central ideas of the original documents.

3.2. Research Design

A research methodology serves as a comprehensive action plan, strategy, process, or framework that informs the selection of methods and links them to their practical implementation [89]. In this study, we adopt experimental research methodologies, which are widely recognized in computer science, to identify the optimal model for our extractive text summarization task. Through a series of carefully designed experiments, we aim to determine the most suitable model parameters while evaluating the impact of various experimental setups on the effectiveness of our approach. This methodological framework allows us to systematically explore and refine our model, ensuring that the summarization process is both robust and effective.

3.3. Dataset Collection and Annotation

Amharic news articles were gathered from different online news websites, including Amhara Mass Media, Addis Admas news, Walta News, Fana news and others. After collecting an Amharic news dataset, we segmented the documents into individual sentences. These sentences were then annotated by experts, with each sentence labeled as either 0 or 1. A label of 0 indicates that the

sentence is not important for the summary, while a label of 1 signifies that it is important. The annotation process followed specific guidelines (annotation standards), which were developed to ensure consistency and reliability in identifying summary-worthy sentences.

The experts were labeled sentences based on the following criteria [90]:

- **Relevance:** Sentences that contribute directly to the main idea or key themes of the text were labeled as 1, while those that do not were labeled as 0.
- **Informativeness:** Sentences that encapsulate core ideas and provide essential information were labeled as 1, while those lacking substance were labeled as 0.
- **Novelty:** Sentences that present new or unique information were labeled as 1, while redundant or repetitive sentences were labeled as 0.
- **Conciseness:** Sentences that convey significant information without excessive detail or elaboration were prioritized and labeled as 1, while overly detailed sentences were labeled as 0.
- **Coherence:** Sentences with a logical flow and clear progression of ideas were labeled as 1, while disjointed or unrelated sentences were labeled as 0.
- **Clarity:** Sentences that are straightforward and easy to understand were labeled as 1, while ambiguous or unclear sentences were labeled as 0.

Each Amharic news sentence contains a unique News ID, a unique Sentence ID, and a summary label. In the annotated dataset, each news item, sentence, and its corresponding label are organized in an Excel table format, with entries separated by sequentially numbered IDs.

3.4. System Architecture

The proposed model architecture consists of several interconnected components designed to effectively summarize Amharic news articles. Initially, we collect Amharic news data from various news sources. Once collected, the data is segmented into individual sentences, which is crucial for transforming the raw text into a format that is suitable for labeling. Next, the dataset undergoes preprocessing, where they are cleaned to remove unnecessary characters and normalized to standardize the text. This process also includes tokenization, preparing the data for further analysis.

In the proposed system architecture, we utilized AmRoBERTa to generate contextual embeddings for each token in the input sentence. The resulting embeddings are then processed through a Bidirectional Long Short-Term Memory layer, enhancing the model ability to understand dependencies and nuances in the text by analyzing the embeddings in both forward and backward directions. The output from the BiLSTM is directed to the output layer to assigned a predicted probability score reflecting its relevance to the overall summary.

In the sentence selection phase, predefined thresholds are applied to these scores to identify and select the most relevant sentences for inclusion in the final summary. Finally, during the sentence compilation stage, the selected sentences are organized based on their associated News ID and concatenated to create coherent summaries.

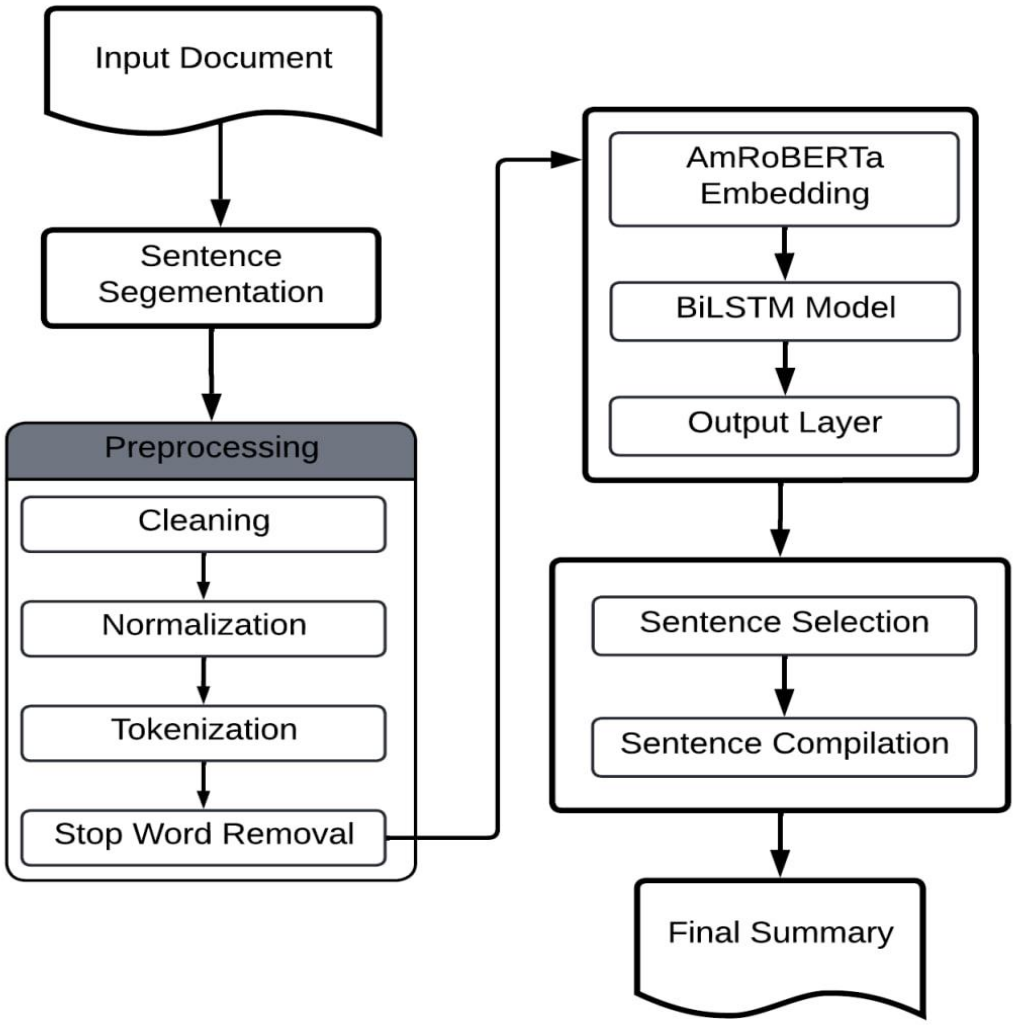

FIGURE 5: ARCHITECTURE OF AMHARIC EXTRACTIVE TEXT SUMMARIZATION

3.4.1. Preprocessing

Preprocessing is essential for adapting our dataset for neural network models and subsequent processing tasks. In natural language processing, text preprocessing serves as an initial phase that converts raw text into a format that machines can interpret, making it easier for machine learning algorithms to function effectively. For this study, the preprocessing phase consists of several key steps: cleaning, normalization, tokenization and stop word removal applied to the segmented sentences of Amharic news articles.

Cleaning

The data cleaning process enhances raw text data by removing unnecessary items that can hinder the performance of natural language processing models. This includes eliminating non-Amharic text, special characters, irrelevant symbols, emojis, and more. By filtering out these elements, the data cleaning process ensures that the text remains focused solely on the Amharic language, allowing NLP models to work more efficiently and accurately.

Normalization

Normalization is the process of transforming a list of words into a more uniform sequence, which is essential for extractive text summarization tasks. The Amharic language is morphologically rich, featuring characters such as ሀ, ሃ, ሐ, ሓ, ኀ, and ኃ, which can be used interchangeably without altering meaning. This variability underscores the necessity of normalization to enhance text matching and processing. By standardizing these characters, normalization simplifies the search process, enabling summarization models to recognize relevant terms more effectively. It ensures that characters like "ሠ" and "ሰ" or "ሀ" and "ሐ" are converted to a single representative form, thereby preparing the text for efficient processing.

	Sentence	normalized_data
0	ሚኒስቴሩ አዲስ የአምስት ዓመት የመንግስት ፋይናንስ አስተዳደር ሪፎርም ስ...	ሚኒስቴሩ አዲስ የአምስት አመት የመንግስት ፋይናንስ አስተዳደር ሪፎርም ስ...
1	የገንዘብ ሚኒስቴር ከ ባሉት አምስት ዓመታት ተግባራዊ የሚሆን የመንግስት...	የገንዘብ ሚኒስቴር ከ ባሉት አምስት አመታት ተግባራዊ የሚሆን የመንግስት...
2	የገንዘብ ሚኒስቴር የፊሲካል ጉዳዮች ዘርፍ ሚኒስትር ዴኤታኢዮብ ተካልኝ ...	የገንዘብ ሚኒስቴር የፊሲካል ጉዳዮች ዘርፍ ሚኒስትር ዴኤታኢዮብ ተካልኝ ...
3	የገንዘብ ሚኒስትር ዴኤታው አከለውም ይህ የሪፎርም ስትራቴጂክ ዕቅድ ይፋ ...	የገንዘብ ሚኒስትር ዴኤታው አከለውም ይህ የሪፎርም ስትራቴጂክ እቅድ ይፋ ...
4	አዲሱ የሪፎርም ስትራቴጅ ተግባራዊ ሆኖ ውጤታማ ሊሆን የሚችለው በባለድርሻ...	አዲሱ የሪፎርም ስትራቴጅ ተግባራዊ ሆኖ ውጤታማ ሊሆን የሚችለው በባለድርሻ...
5	በሚኒስቴሩ የመንግስት ፋይናንስ ማሻሻያ ዲቪዥን ኃላፊ የሆኑት ዳዊት ሽመል...	በሚኒስቴሩ የመንግስት ፋይናንስ ማሻሻያ ዲቪዥን ሀላፊ የሆኑት ዳዊት ሽመል...
6	በፋይናንስ ተቋማት ያለው የቁጠባ ገንዘብ መጠን ከ ትሪሊየን ብር በላይ መ...	በፋይናንስ ተቋማት ያለው የቁጠባ ገንዘብ መጠን ከ ትሪሊየን ብር በላይ መ...
7	በኢትዮጵያ የፋይናንስ ተቋማት ውስጥ ያለው የቁጠባ ገንዘብ መጠን ከ ትሪሊ...	በኢትዮጵያ የፋይናንስ ተቋማት ውስጥ ያለው የቁጠባ ገንዘብ መጠን ከ ትሪሊ...
8	የባንኩ ምክትል ገዥ ሰለሞን ደስታ የሕብረተሰቡ የቁጠባ ባህል ከጊዜ ወደ ...	የባንኩ ምክትል ገዥ ሰለሞን ደስታ የህብረተሰቡ የቁጠባ ባህል ከጊዜ ወደ ...
9	በመሆኑም ባንኮችን ጨምሮ በተለያዩ የፋይናንስ ተቋማት አጠቃላይ የቁጠባ ...	በመሆኑም ባንኮችን ጨምሮ በተለያዩ የፋይናንስ ተቋማት አጠቃላይ የቁጠባ ...

FIGURE 6: NORMALIZED DATA

Tokenization

Tokenization is the process of breaking down sentence or entire text document into smaller units known as tokens, which can be words or phrases. In our case, we have employed tokenization to split sentences into their constituent words. We used the AmRoBERTa tokenizer, a specialized tool designed for processing the Amharic language that is derived from the AmRoBERTa model.

The AmRoBERTa tokenizer effectively handles the unique characteristics of Amharic text and utilized for various natural language processing tasks.

```
tokenizer = RobertaTokenizer.from_pretrained('uhhlt/am-roberta')
```

The following table displays some tokenized Amharic news sentences from our dataset, generated by the AmRoBERTa tokenizer.

normalized_data	tokenized_data
ሚኒስቴሩ አዲስ የአምስት አመት የመንግስት ፋይናንስ አስተዳደር ሪፎርም ስ...	[ሚኒስቴሩ, አዲስ, የአምስት, አመት, የመንግስት, ፋይናንስ, አስተዳደር...
የገንዘብ ሚኒስቴር ከ ባሉት አምስት አመታት ተግባራዊ የሚሆን የመንግስት...	[የገንዘብ, ሚኒስቴር, ከ, ባሉት, አምስት, አመታት, ተግባራዊ, የሚሆን...
የገንዘብ ሚኒስቴር የፊሲካል ጉዳዮች ዘርፍ ሚኒስትር ዴኤታኢዮብ ተካልኝ ...	[የገንዘብ, ሚኒስቴር, የፊሲካል, ጉዳዮች, ዘርፍ, ሚኒስትር, ዴኤታኢዮብ...
የገንዘብ ሚኒስትር ዴኤታው አከለውም ይህ የሪፎርም ስትራቴጂክ እቅድ ይፋ ...	[የገንዘብ, ሚኒስትር, ዴኤታው, አከለውም, ይህ, የሪፎርም, ስትራቴጂክ,...
አዲሱ የሪፎርም ስትራቴጅ ተግባራዊ ሆኖ ውጤታማ ሊሆን የሚችለው በባለድርሻ...	[አዲሱ, የሪፎርም, ስትራቴጅ, ተግባራዊ, ሆኖ, ውጤታማ, ሊሆን, የሚችለ...
...	...
አነስተኛ የመብራት ሀይል የሚጠቀምና ለመያዝ ቀላል የሆነ ፕሮጀክተር ከጓደ...	[አነስተኛ, የመብራት, ሀይል, የሚጠቀምና, ለመያዝ, ቀላል, የሆነ, ፕሮ...
ፕሮጀክቱ የተለያዩ ፊልሞችን ለማየትና ለትምህርት አገልግሎት የሚውል እንደ...	[ፕሮጀክቱ, የተለያዩ, ፊልሞችን, ለማየትና, ለትምህርት, አገልግሎት, የ...
ውድድሩ ከበርካታ አዳዲስ የስራ ፈጠራዎች ጋር የተደረገና ሴት ልጆች እንዴ...	[ውድድሩ, ከበርካታ, አዳዲስ, የስራ, ፈጠራዎች, ጋር, የተደረገና, ሴት...
በዩኒቨርስቲው የአምስተኛ አመት የኢንጅነሪንግ ተማሪ ቁፋሮነት ህይወት እን...	[በዩኒቨርስቲው, የአምስተኛ, አመት, የኢንጅነሪንግ, ተማሪ, ቁፋሮነት, ...
በውድድሩ አሸናፊ ለሆኑ እቅዶች ገንዘብ የተሰጠ ሲሆን አንደኛ የሆነውን እ...	[በውድድሩ, አሸናፊ, ለሆኑ, እቅዶች, ገንዘብ, የተሰጠ, ሲሆን, አንደኛ...

FIGURE 7: TOKENIZED DATA

Stop word removal

Stop words are often considered ineffective for identifying the main points of a text, despite their significant contribution to co-occurrence counts. This can negatively impact the role of non-stop words in pinpointing the main ideas, ultimately affecting the performance of summarization algorithms. For example, in Amharic, words like ነው (new), ናቸው (they are), and ነበረ (was) contribute little value to the document's overall meaning and often introduce noise instead of clarity. Because of this, we remove stop words from our dataset since they are not important for analysis and do not help convey the meaning of the documents. Common stop words in Amharic include conjunctions, articles, and prepositions, which usually do not provide much useful information to the text.

3.4.2. AmRoBERTa Embedding

Word embedding is a language modeling technique that maps words to vectors of real numbers, representing them in a multi-dimensional vector space. These embeddings allow models to understand the semantic similarities between words and their usage in sentences, facilitating more effective language processing.

In this study, we utilize the AmRoBERTa model to generate these embeddings, leveraging its unique advantages in contextual understanding, particularly for processing Amharic text. AmRoBERTa produces a sequence of embeddings that encapsulate the contextual information of entire sentences. As a variant of the RoBERTa model, it has been specifically fine-tuned for the Amharic language. Pre-trained on a substantial corpus of Amharic text, AmRoBERTa effectively captures the linguistic nuances and contextual details unique to this language. RoBERTa, the foundation of AmRoBERTa, is a transformer-based model derived from BERT (Bidirectional Encoder Representations from Transformers) and is optimized for enhanced performance across various natural language understanding tasks.

3.4.3. BiLSTM Model

In our study, we employ Bidirectional Long Short-Term Memory networks, a type of recurrent neural network particularly effective for various natural language processing tasks, including extractive text summarization. BiLSTM enhances traditional LSTM networks by processing text in both forward and backward directions, significantly improving their ability to capture contextual information from both past and future contexts.

After the extraction of AmRoBERTa embeddings, these representations are passed through a BiLSTM layer configured with a specified number of units in both directions. This BiLSTM layer effectively captures dependencies within the sequence of embeddings, enriching the model understanding of the context in which words and sentences appear. To mitigate overfitting, dropout and recurrent dropout techniques are applied, randomly deactivating a fraction of the input units during training. This integration allows for a more nuanced understanding of word relationships and ultimately enhances overall model performance.

To convert the sequence of outputs from the BiLSTM layer into a fixed-size representation, a **GlobalAveragePooling1D layer** is employed. This layer computes the average of the embeddings across all time steps, effectively summarizing the sentence into a single vector that captures its essential features. The resulting sentence embedding is then refined through a series of dense layers. The first dense layer contains 128 units with ReLU activation and applies L2 regularization to mitigate overfitting by penalizing large weights. This is followed by batch normalization to stabilize the learning process and a dropout layer, which further prevents overfitting by deactivating some activations during training.

3.4.4. Output Layer

It serves as the final component of the neural network model, consisting of a dense layer that employs a sigmoid activation function. This layer generates a probability score, indicating the likelihood that the input sentence is relevant or irrelevant to the summary.

3.4.5. Sentence Selection

In this study, the sentence selection function identifies relevant sentences within a document for inclusion in the summary. This process applies a threshold to the predicted probabilities of each sentence. While the most common threshold used is 0.5, we also utilize a threshold of 0.4 for comparison purposes. Sentences with scores exceeding the chosen threshold are considered relevant and included in the summary, whereas those that do not meet these thresholds are deemed irrelevant and excluded. The selected relevant sentences are then compiled into a list of tuples, where each tuple contains the news ID, sentence ID, and the actual sentence text.

3.4.6. Sentence Compilation

After the selection of relevant sentences, the sentence Compilation function organize the selected sentences by their respective News_id and Sentences_id, grouping all selected sentences associated with each news item. For each news_id, the function concatenates the relevant sentences into a single summary. In cases where no sentences are selected for a particular news_id, it returns a default message indicating that no summary is available.

Output: The final output of the system is an extractive summary comprised of the most relevant sentences selected from the original document.

3.5. Model Optimization

K-Fold Cross-Validation is an essential technique used in machine learning to evaluate the performance and robustness of a model, particularly when dealing with limited data [91]. In this study, Extractive Amharic Text Summarization," K-Fold Cross-Validation ensures that the developed model for Amharic text summarization is both accurate and generalizable across different subsets of the data.

The primary idea behind K-Fold Cross-Validation is to partition the available dataset into 'K' equally sized folds or subsets. During each iteration, one of these folds is set aside as the test dataset, while the remaining 'K-1' folds are used to train the model. This process is repeated 'K' times, with each fold being used as the test set exactly once. After completing all 'K' iterations, the results are averaged to provide a more reliable and unbiased estimate of the model's performance.

3-Fold and 5-Fold Cross-Validation were utilized through the Stratified K-Fold method, which ensures that each fold preserves the original label distribution of the dataset. This approach involves training the model on various subsets of the data. Such a strategy aids in identifying potential overfitting and offers a thorough understanding of how the model is likely to perform on new, unseen data. The final metrics and the summaries generated across all folds provide valuable insights into the model's effectiveness in delivering accurate and contextually appropriate summaries in Amharic.

3.6. Evaluating Summaries

The performance of the model is evaluated by assessing the quality of generated summaries against reference summaries using ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics. This evaluation process provides a quantitative measure of the similarity between the generated and reference summaries. Specifically, it employs three types of ROUGE scores:

ROUGE-1: Measures the overlap of individual words between the generated and reference summaries.

ROUGE-2: Evaluates the overlap of pairs of consecutive words between the generated and reference summaries.

ROUGE-L: Evaluates the longest sequence of words that appear in the same order in both summaries.

These metrics collectively offer a comprehensive measure of similarity, helping to determine how effectively the generated summaries align with the reference summaries.

CHAPTER FOUR

EXPERIMENTATION, RESULTS AND DISCUSSION

4.1. Overview

In this chapter, we provide a detailed overview of the dataset utilized in our study, along with the implementation procedures and experimental setups. We also discuss the hyperparameters used, and in addition, we outline the results obtained from our experiments.

4.2. Dataset Description

A total of 1,200 Amharic news articles were gathered from a diverse online news website, including Amhara Mass Media, Addis Admas newspaper, Walta News, and others. The articles cover various categories such as sports, technology, business, and more. This diverse selection allows for capturing a wide range of viewpoints and types of news, ensuring that the dataset reflects various interests and issues. By using articles from multiple areas, a comprehensive dataset was created that will be valuable for summarization efforts.

After collecting the data, the initial step involved performing sentence segmentation on the articles to prepare for labeling. Then expert annotators labeled each segmented sentence as "1" for important to summary and "0" for not important to summary within each document.

Each document was assigned a unique ID, while each news segment received its own news ID. Additionally, every segmented sentence was given a specific sentence ID. This structured approach allowed for clear identification and organization of the data, facilitating the labeling process. The labeling indicates the role of each sentence, with "1" representing extractive summary sentences and "0" for those that do not contribute to the summary.

This systematic method ensures that all sentences are accurately categorized, making the dataset well-suited for our summarization efforts. 5-fold and 3-fold cross-validation technique was employed to ensure robust performance evaluation across different subsets of the data.

TABLE 2: DATA FORMAT

	News_id	Sentence_id	Sentence	Label
0	1	1	ለስንዴ ምርታቸው ተገቢውን የገበያ ዋጋ ማጣታቸውን አርሶ አደሮች ተናገሩ	0
1	1	2	ነባ ታህሳስ 4 2011 ለፖስታና መከሮኒ ስንዴ ምርታቸው ተገቢውን የገ...	1
2	1	3	አርሶ አደሮቹ ለኢዜአ እንዳሉት የፖስታና መከሮኒ የስንዴ ምርት ያዋጠናል...	1
3	1	4	ሆኖም እያመረቱት ላለው ምርት ተገቢው ዋጋ ማግኘት ባለመቻላቸው ተቸግረዋል	0
4	1	5	ከአርሶ አደሮቹ መካከል የአጋርፋ ወረዳ ነዋሪው አቶ ሙስጠፋ በከር እንዳተ...	1
...	...	...	...	...
4995	711	13	አነስተኛ የመብራት ኃይል የሚጠቀምና ለመያዝ ቀላል የሆነ ፕሮጀክተር ከ...	1
4996	711	14	ፕሮጀክቱ የተለያዩ ፊልሞችን ለማየትና ለትምህርት አገልግሎት የሚውል እንደ...	1
4997	711	15	ውድድሩ ከበርካታ አዳዲስ የስራ ፈጠራዎች ጋር የተደረገና ሴት ልጆች እንዴ...	0
4998	711	16	በዩኒቨርስቲው የአምስተኛ ዓመት የኢንጅነሪንግ ተማሪ ፍሬገነት ሀይወት እን...	0
4999	711	17	በውድድሩ አሸናፊ ለሆኑ እቅዶች ገንዘብ የተሰጠ ሲሆን እንደኛ የሆነውን ...	0

4.3. Experimental Setup

To evaluate the performance of the proposed hybrid model for Amharic text summarization, a comprehensive experimental setup was established. This setup was designed to ensure that the experiments could be conducted efficiently and effectively.

Hardware Specifications

The experiments were conducted on a dedicated HP laptop computer, specifically the HP 250 G7 Notebook PC. The hardware specifications of the machine are as follows:

TABLE 3: EXPERIMENTAL SETUP

Manufacturer	HP
Model	HP 250 G7 Notebook PC
Processor	Intel(R) Core (TM) i5-8265U CPU @ 1.6GHz 3.19 GHz
Memory (RAM)	8 GB
Operating System	Windows 10

Software Specifications

For the development and experimentation of the summarization model, we utilized the following tools:

Python is a high-level, general-purpose programming language that is extensively used for a number of purposes, such as text and image processing. It is easily expandable, portable, and free due to its open-source and freely available nature. Its prominence in the data science and machine learning domains can be attributed to its readability and simplicity.

TensorFlow is an open-source, Python-based library created by the Google Brain team for numerical computation and large-scale machine learning. It presents a range of deep learning and machine learning models and algorithms in a cohesive framework. TensorFlow is a tool that allows programmers to construct models using data flow graphs. This allows programmers to generate huge neural networks that are mostly used for tasks like generation, prediction, natural language interpretation, and classification.

Keras is a popular and user-friendly deep learning framework that simplifies the process of defining, developing, and evaluating the performance of models. It provides a high-level interface for building neural networks, making it accessible for rapid prototyping and experimentation.

4.4. Hyperparameters in the model

Hyperparameters are crucial factors that influence both the architecture of the network and the training process of the model. These parameters are set before training begins and remain fixed during the optimization of weights and biases. The hyperparameters that were considered for our hybrid model are as follows:

Epochs: This defines how many times the model will read through the entire dataset. Each experiment involved training the model for up to 20 epochs, with early stopping applied to prevent overfitting if the validation performance stopped improving after a certain number of epochs.

Batch Size: This refers to the number of samples processed before the model internal parameters are updated. A smaller batch size can provide a more accurate estimate of the gradient, while a larger batch size can speed up training. We experimented with batch sizes of 4, 8,16 and 32 to identify the most efficient size that balanced memory usage and performance.

Learning Rate: This controls the speed at which the optimizer descends the error curve. A well-tuned learning rate can significantly improve convergence. Different learning rates were explored, ranging from 0.0001 to 0.01, to ensure that the model converged effectively during training without getting stuck in local minima.

Dropout: This is a regularization technique used during training to prevent overfitting by randomly setting a fraction of the input units to zero at each update. Dropout was applied to prevent overfitting, with values of 0.4, and 0.5 tested to observe its impact on generalization performance.

Optimizer: The optimizer role is to minimize the loss function by updating the network parameters based on the gradients calculated during training. There are several deep learning optimizers available, each with unique strengths. For our experiment, we used the Adam optimizer, which is known for its adaptive learning rate capabilities.

Loss Function: The loss function measures the performance of the model by calculating the difference between the model predictions and the actual target values. This difference indicates

how accurately the model is predicting outcomes. In our experiments, we used the binary cross-entropy loss function, which is particularly suitable for binary classification tasks.

Activation Function: The activation function determines the output of a neural network node. In our model, we used the sigmoid activation function. The sigmoid function squashes the output to a range between 0 and 1, making it particularly useful for binary classification tasks.

Cross-Validation: By splitting the data into subsets, a statistical technique called cross-validation is used to evaluate how well a machine learning model is performing. To guarantee that the assessment is trustworthy and independent of a particular train-test split, the model is trained on a subset of the data and validated on the remaining portions several times. Three- and five-fold cross-validation were both used in this investigation.

4.5. Results of the experiments

In this experiment, we tested the hybrid AmRoBERTa-BiLSTM model using the hyperparameters defined earlier in this chapter. The results of the experiment were measured using ROUGE scores. We conducted various experiments by tuning different batch sizes, learning rate, threshold to evaluate their impact on the model performance.

Experiment 1: In the first experiment, we employed a 5-fold cross-validation approach. The model was configured with a dropout rate of 0.5 to mitigate overfitting, using the ReLU activation function in the hidden layer to introduce non-linearity and improve learning efficiency. In the final classification layer, the sigmoid activation function was used, which is well-suited for binary classification tasks. The Adam optimizer was selected with a learning rate of 0.001, which is a common choice for deep learning models due to its adaptability. The model was trained over 10 epochs with a threshold value of 0.4, and a batch size of 4 was chosen to evaluate how small batch sizes impact learning stability. Under these conditions, the model achieved average ROUGE scores of 40.21 for ROUGE-1, 29.38 for ROUGE-2, and 40.11 for ROUGE-L.

Experiment 2: For the second experiment, all hyperparameters remained the same as Experiment 1, except for an increased batch size of 8. This change allowed the model to process more data at once, potentially leading to better gradient estimation and faster convergence. The model showed a slight improvement in ROUGE-1 40.32 and ROUGE-L 40.29, while ROUGE-2 decreased marginally 29.25. This suggests that a moderate increase in batch size helped the model to better capture unigrams and Longest Common Subsequence, though it did not significantly improve bigram.

Experiment 3: In the third experiment, we increased the batch size further to 16 while reducing the learning rate to 0.0001 with threshold 0.4. Lowering the learning rate allowed the model to make smaller, more precise updates to the weights, leading to gradual learning. This change improved all ROUGE scores, with ROUGE-1 reaching 41.32, ROUGE-2 improving to 29.77, and ROUGE-L also achieving 41.28. This result suggests that larger batch sizes, combined with a smaller learning rate, helped the model better capture both unigrams and bigrams, as well as Longest Common Subsequence.

Experiment 4: In the fourth experiment, the model was adjusted to use a threshold value of 0.5 with 3-fold cross-validation and dropout rate of 0.4. This allowed more neurons to remain active during training, potentially improving the model generalization ability. The batch size was reverted to 4, and the learning rate remained at 0.001. This configuration yielded the best performance so far, with average ROUGE-1 reaching 42.50, ROUGE-2 improving to 33.07, and ROUGE-L also at 42.48. The shift to 3-fold cross-validation likely reduced the variability in the training data, leading to improved performance.

Experiment 5: In this experiment, the batch size was increased to 8, while keeping other parameters the same as Experiment 4. However, this led to a slight decline in performance, with average ROUGE-1 dropping to 40.55, ROUGE-2 decreasing to 31.70, and ROUGE-L also at 40.54. This suggests that increasing the batch size did not necessarily result in better performance under these settings, possibly due to overfitting or suboptimal gradient updates.

Experiment 6: For the sixth experiment, the batch size was further increased to 16 while keeping other parameters constant. This change resulted in a moderate performance improvement, achieving average ROUGE-1 of 42.16, ROUGE-2 of 33.04, and ROUGE-L of 42.11. This shows that under certain conditions, larger batch sizes can improve model performance, likely by stabilizing the learning process.

Experiment 7: In the seventh experiment, we tested a batch size of 32. This larger batch size led to a slight decline in performance, with ROUGE-1 at 41.45, ROUGE-2 at 32.74, and ROUGE-L at 41.42. While the model could process more data at once, the increased batch size might have caused the model to make less frequent updates, leading to suboptimal convergence.

Experiment 8: The eighth experiment explored the impact of increasing the learning rate to 0.01 while keeping the batch size at 4. This experiment resulted in a notable performance decrease, with ROUGE-1 dropping to 37.74, ROUGE-2 to 30.02, and ROUGE-L to 37.70. The larger learning rate likely caused the model to overshoot the optimal weight updates, leading to poorer performance.

Experiment 9: In the final experiment, the learning rate was reduced to 0.0001 with a batch size of 4. This configuration resulted in the best overall performance, with ROUGE-1 reaching 44.48, ROUGE-2 improving to 34.73, and ROUGE-L at 44.47. The smaller learning rate allowed the model to make finer updates, leading to better generalization and improved performance across all metrics.

The following table shows the overall summary of experimental results and hyperparameter values used in each experiment in detail.

TABLE 4: SUMMARY OF EXPERIMENTAL RESULTS

Experiment	Thershold	Cross-Validation	Dropout	Batch Size	Learning Rate	ROUGE-1	ROUGE-2	ROUGE-L
1	0.4	5-fold	0.5	4	0.001	40.21	29.38	40.11
2	0.4	5-fold	0.5	8	0.001	40.32	29.35	40.29
3	0.4	5-fold	0.5	16	0.0001	41.32	29.77	41.28
4	0.5	3-fold	0.4	4	0.001	42.50	33.07	42.48
5	0.5	3-fold	0.4	8	0.001	40.55	31.70	40.54
6	0.5	3-fold	0.4	16	0.001	42.16	33.04	42.11
7	0.5	3-fold	0.4	32	0.001	41.45	32.74	41.42
8	0.5	3-fold	0.4	4	0.01	37.74	30.02	37.70
9	0.5	3-fold	0.4	4	0.0001	44.48	34.73	44.47

Comparative Analysis

The experiments demonstrate that adjusting hyperparameters such as batch size, learning rate, dropout rate, and cross-validation strategy significantly impacts the model performance. Larger batch sizes (16 and 32) generally improved performance up to a certain point; however, excessively large batches can lead to performance degradation if they result in infrequent weight updates. Lowering the learning rate to 0.0001 allowed the model to achieve the best results, indicating that gradual weight updates are most effective for this task. Conversely, increasing the learning rate to 0.01 resulted in a notable drop in performance due to overshooting during weight updates. Additionally, the use of 3-fold cross-validation provided more stable and improved results compared to 5-fold cross-validation, as evidenced in Experiments 4 through 9. In conclusion, the

combination of smaller learning rates, moderate dropout rates, and 3-fold cross-validation generally yielded the best results for optimizing the performance of extractive Amharic text summarization models.

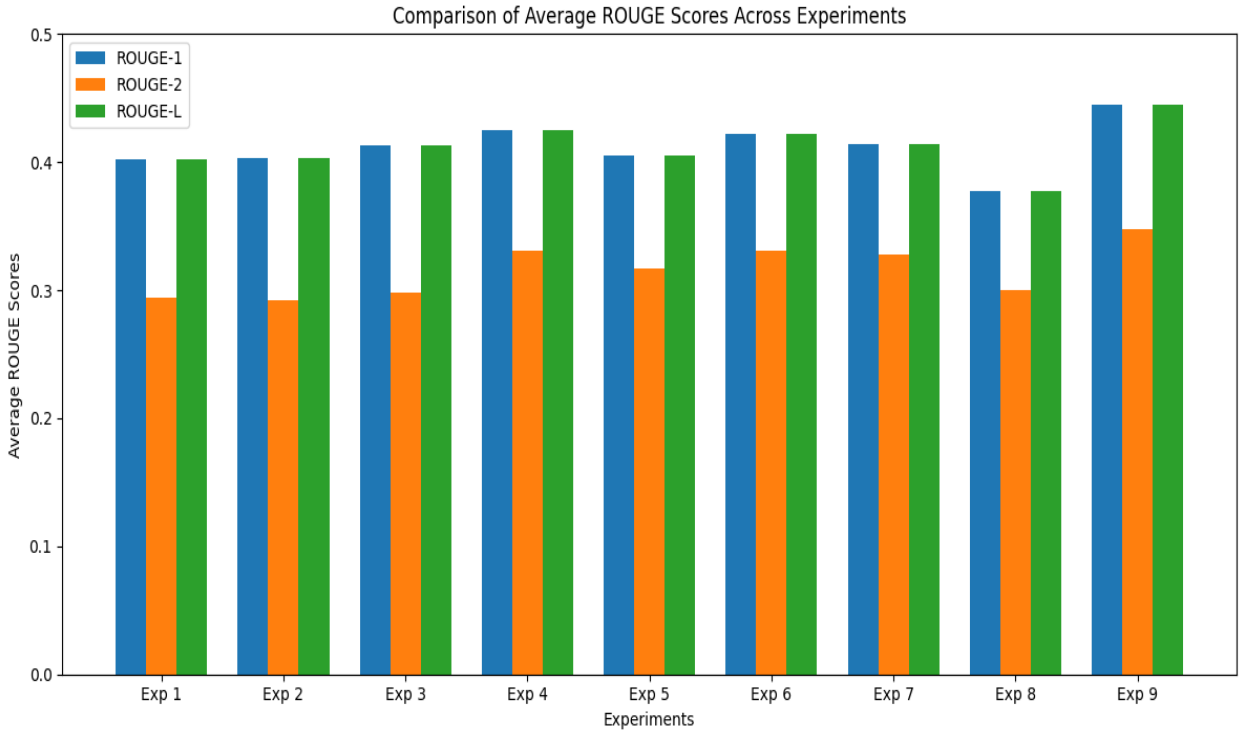

FIGURE 8: COMPARISON OF ROUGE SCORE ACROSS EXPERIMENTS

Generated Summary 2:

- ለረዥጋማ መሬትና ዝናብ አጠር አካባቢዎች የሚስማሙ ምርጥ የሰብል ዝርያዎች ተገኙ የማሽላ የሰሊጥና የጤፍ ዝርያዎቹ በአጭር ጊዜ ለፍራ የሚበቁ በሽታን የመቋቋም አቅም ያላቸውና ከፍተኛ ምርት የሚሰጡ ናቸው በተጨማሪም ማእከሉ በአነስተኛ የዝናብ መጠን ለምቶ በአጭር ጊዜ ለምርት የሚደርሰው አባ ሲና የተሰኘ የሰሊጥ ምርጥ ዝርያ እያስተዋወቀ መሆኑንም አስተባባሪው ገልፀዋል በሰሊጥ የሚካሄደው ምርምር በዝናብ ታግዞ ከሚለማው የማሽላ ዝርያ በተጓዳኝ በማልማት አርሶ አደሩን ተጠቃሚ ለማድረግ ታልሞ እንደሚካሄድም አስረድተዋል በአሁኑ ጊዜ የሰሊጥ ምርጥ ዝርያን ከማዳበሪያ ፓኬጅ ጋር በማቀናጀት በራያ አዘቦና ራያ አላማጣ ወረዳዎች በሚገኙ 20 ዝቡራዎች ማሳ ላይ እየለማ ይገኛል ብለዋል በራያ አላማጣ ወረዳ የጡሙጋ ቀበሌ ነዋሪ አርሶ አደር ዝናቡ ደርቦ የእርሻ መሬታቸው በረዥጋማ ውሀ በመሸፈኑ የዘሩትን ማሽላና ጤፍ ማሳ ላይ ይጠፋባቸው እንደነበር ይናገራሉ ከዚህ በፊት የአካባቢ ማሽላና ጤፍ በውሀ እጥረትና በአቀንጭራ አረም ሲጠቁ እንደነበር የሚገልፁት የዚሁ ቀበሌ ነዋሪ አርሶ አደር አብርሀ ዝናቡ ማእከሉ ያቀረባቸው ዝርያዎች አቀንጭራ አረምን ተቋቁመው በአነስተኛ ዝናብ ውሀ ፈጥነው ምርት ሰጥተዋል ብለዋል

Generated Summary 4:

- ከአማራ ብድርና ቁጠባ ተቋም ጋር በመተባበር የኅብርና ኅብለት አጅ በኢጅ ሽያጭ መዝሓት የማይችሉ አርሶ አደሮች ከተቋሙ ብድር በመውሰድ ገንዘቡን በወቅቱ እንዲመልሱ ማድረግ ያስቻለ አሰራር መሆኑንም አስረድተዋል ቡድን መሪው እንዳሉት ከዚህ በፊት የኅብለት ስርጭቱ በማህበራት በኩል በነበረበት ወቅት እስከ 3 አመት የሚቆይ ብድር ለአርሶ አደሩ ቢሰጥም ገንዘቡን ለማስመለስም ችግር ነበር ባለፈው አመት ብቻ ለምሽር እርሻ በብድር ከተሰራጨው ከ311ሚሊዮን ብር ውስጥ እስካሁን ከ260 ሚሊዮን በር በላይ መመለሱን በማሳያነት ጠቅሰዋል ከዚህ በፊት ካለብኝ የገንዘብ እጥረት ኅብት በብድር እንዲሰጠኝ ስጠይቅ ከቀበሌ አስመስክር እየተባለ ስለሚገተት ኅብት ያልት ነበር የሚሰጠኝም ከምፈልገው በታች ነበር ሲሉም ገልፀዋል አሁን ግን ብድሩን እስፍላሉሁ ብዙ ከአብቁተ እስከተበደርኩ ድረስ የሚያስፈልገኝን ያህል ኅብት ወስጄ መጠቀም እየቻልኩ ነው ብድሬንም በወቅቱ እየመለስኩ ነው ብለዋል

Generated Summary 7:

- የክልሉ የእርሻ መሬት የአፈር ባህሪ ሁሉም ሊባል በሚችል መልኩ አሲዳማ ቢሆንም አፈሩ ለእርሻ ስራ ችግር የማይፈጥር እንደሆነ በጥናቱ መለየቱን ገልፀዋል በዚህ ምክንያት ለክልሉ የእርሻ መሬት ስምንት የማዳበሪያ አይነት እንደሚያስፈልግ በጥናት መለየቱን ቡድን መሪው ተናግረዋል በጥናቱ ለክልሉ የእርሻ መሬት እንደሚያስፈልጉ ከተለዩት የማዳበሪያ አይነቶች መካከል ፆሪያ ፓታሽ ቦሮን እና ፎስፌት የተባሉት እንደሚገኙበት ተመልክቷል እንደ ቡድን መሪው ገለፃ ጥናቱን የክልሉ አመራሮች ባለሙያዎችና ሌሎች ባለድርሻ አካላት በተገኙበት ይፋ ማድረግ ያስፈለገው ተጨማሪ ኅብአቶችን ለማሰባሰብ ያለመ ነው በዚህም ምክንያት አርሶ አደሩ በሄክታር የሚያገኘው ምርት ከሌሎች የሀገሪቱ አካባቢዎች ጋር ሲነፃፀር በሶስት እጥፍ ዝቅ ያለ መሆኑን አስረድተዋል ጥናቱ ይፋ በተደረገበት ስነስርአት ወቅት የተገኙት የክልሉ ርእሰ መስተዳድር አቶ ኡሞድ ኡጁሉ በክልሉ የተካሄደውን የአፈር ለምነትና ማደበሪያ አጠቃቀም ምክረ ሀሳብ ጥናት ወደ ተግባር በመለወጥ የግብርናውን ዘርፍ ለማሳደግ በትኩረት ሊሰራ ይገባል ብለዋል የክልሉ የአፈር ለምነትና የማዳበሪያ አጠቃቀም ምክር ሀሳብ ጥናት ካርታ በቅርቡ ታትሞ ስራ ላይ ይውላል ተብሏል

FIGURE 9: SAMPLE GENERATED SUMMARY

Model: "functional_17"

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	(None, None)	0	-
attention_mask (InputLayer)	(None, None)	0	-
lambda_9 (Lambda)	(None, None, 768)	0	input_ids[0][0], attention_mask[0][0]
bidirectional_1 (Bidirectional)	(None, None, 512)	2,099,200	lambda_9[0][0]
global_average_pooling1d_9 (GlobalAveragePooling1D)	(None, 512)	0	bidirectional_1[0][0]
dense_27 (Dense)	(None, 128)	65,664	global_average_pooling1d_…
batch_normalization_19 (BatchNormalization)	(None, 128)	512	dense_27[0][0]
dropout_19 (Dropout)	(None, 128)	0	batch_normalization_19[0]…
dense_28 (Dense)	(None, 64)	8,256	dropout_19[0][0]
batch_normalization_20 (BatchNormalization)	(None, 64)	256	dense_28[0][0]
dropout_20 (Dropout)	(None, 64)	0	batch_normalization_20[0]…
Sentence_id (InputLayer)	(None, 1)	0	-
News_id (InputLayer)	(None, 1)	0	-
dense_29 (Dense)	(None, 1)	65	dropout_20[0][0]

Total params: 2,173,953 (8.29 MB)
Trainable params: 2,173,569 (8.29 MB)
Non-trainable params: 384 (1.50 KB)

FIGURE 10: MODEL SUMMARY

4.6. Discussion

In this work, we developed an extractive text summarization model using AmRoBERTa-BiLSTM. We followed several steps to prepare our data, starting by segmenting the input documents into sentences. Each sentence was labeled as either 1 (relevant to the summary) or 0 (not relevant). Next, we performed data cleaning. Once the data was cleaned, we normalized it to ensure consistency by converting similar-sounding characters into their canonical forms. We then proceeded to tokenization using the pre-trained AmRoBERTa tokenizer.

We utilized the AmRoBERTa-BiLSTM model for extractive summarization. The experimental results demonstrate the impact of various hyperparameters on the performance of the extractive Amharic text summarization model, as measured by ROUGE scores. We conducted experiments using both 3-fold and 5-fold cross-validation, which allowed for a comprehensive evaluation of the model robustness and generalization capabilities.

Overall, the findings underscore the importance of carefully tuning hyperparameters such as batch size, learning rate, and dropout rate, as they have a significant impact on the model ability to generate accurate performance, as indicated by the ROUGE scores across both 3-fold and 5-fold cross-validation setups. The highest ROUGE scores were achieved with a learning rate of 0.0001 and a batch size of 4, under the 3-fold cross-validation setup. This configuration produced ROUGE-1, ROUGE-2, and ROUGE-L scores of 44.48, 34.73, and 44.47, respectively, indicating that a lower learning rate is beneficial for optimizing model performance in this case. Additionally, the use of different cross-validation techniques provided valuable insights into the model generalization performance, with 3-fold cross-validation generally yielding slightly better results compared to 5-fold cross-validation. This suggests that the model may perform better with a smaller number of folds, potentially due to the increased training data available in each fold, leading to better learning outcomes.

RQ1. How can the hybrid model (AmRoBERTa-BiLSTM) be effectively utilized for summarizing Amharic text?

The hybrid model AmRoBERTa-BiLSTM effectively summarizes Amharic text by leveraging the strengths of both components. AmRoBERTa provides contextual embeddings that capture the Amharic semantics, while the BiLSTM processes these embeddings in both directions to identify key sentences. The model is trained on a labeled dataset of Amharic texts, enabling it to learn significant patterns relevant to summarization. Effective preprocessing techniques, such as tokenization and normalization, ensure that the input data is clean, enhancing the quality of the generated summaries. Performance is evaluated using ROUGE scores. By combining AmRoBERTa embeddings with BiLSTM sequential learning capabilities, the model generates high-quality extractive summaries.

RQ2. What hyperparameters significantly influence the performance of the AmRoBERTa-BiLSTM model in extractive summarization tasks?

The combination of a learning rate of 0.0001, a batch size of 4, a dropout rate of 0.4, and the use of 3-fold cross-validation proved to be the most effective configuration for optimizing the performance of the AmRoBERTa-BiLSTM model in extractive summarization tasks for Amharic text.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1. Conclusion

In this study, we thoroughly explored the field of extractive Amharic text summarization, with the objective of developing an effective model for summarizing Amharic news articles from a variety of domains. Our research journey encompassed multiple stages, including data collection, preparation, preprocessing, experimentation, and model evaluation. We began by gathering a rich dataset of 1,200 Amharic news articles from a variety of fields, including Sports, Politics, Science and Technology, and Agriculture. Each article was carefully segmented into individual sentences, which were then labeled as 1 or 0 by experts to indicate whether they should be included in the summary or not. This rigorous data preparation process established a solid foundation for the subsequent modeling tasks. To enhance the model performance, we undertook a series of preprocessing steps. This included cleaning, normalization, stop word removal. Additionally, we used the pre-trained AmRoBERTa tokenizer to prepare the text for modeling. The experimentation phase involved implementing a hybrid model that combined AmRoBERTa with BiLSTM. This approach aimed to leverage the strengths of both models to achieve optimal summarization results. From the conducted experiments Experiment 9 consistently outperforms the other experiments across all three ROUGE metrics. Experiment 9 achieved ROUGE-1 score of 44.48, ROUGE-2 score of 34.73 and ROUGE-3 score of 44.47.

In conclusion, this study contributes to the advancement of Amharic text summarization by leveraging state-of-the-art models and techniques. As we progress, ongoing research and development in this field have the potential to enhance effective text summarization in the Amharic language, addressing diverse applications and user needs.

5.2. Contribution of the Study

This study makes several significant contributions to the field of Amharic text summarization, particularly in Amharic extractive text summarization.

- We have collected a comprehensive dataset of 1,200 Amharic news articles from various domains, with labeling conducted by experts. This valuable resource provides a solid foundation for researchers interested in Amharic text summarization and supports future studies in this area.
- We have developed a hybrid deep learning model combining AmRoBERTa and BiLSTM, specifically tailored for Amharic text summarization.
- This study addresses common challenges in text summarization by improving the extraction of meaningful summaries from Amharic news articles, which can enhance various NLP applications, including information retrieval and content summarization.

.

5.3. Recommendation

Building on the findings and contributions of this study, several recommendations emerge for future researchers interested in advancing the field of Amharic text summarization and natural language processing:

Domain-specific Summarization: Given the diverse domains covered in the collected dataset, future research could focus on domain-specific summarization techniques. Tailoring summarization models to specific domains such as politics, science, technology, or agriculture could lead to more accurate and relevant summaries tailored to domain-specific requirements.

Integration of Multimodal Data: Incorporating multimodal data sources, such as images or audio transcripts associated with news articles, could enrich the summarization process. Future research could explore techniques for integrating and summarizing multimodal data to provide comprehensive and informative summaries across multiple modalities.

Evaluation Metrics and Benchmarking: Future research should continue to refine evaluation metrics and benchmarking standards for Amharic text summarization. Developing standardized evaluation protocols and benchmark datasets tailored to the nuances of Amharic language and summarization tasks would facilitate fair and comprehensive model comparisons.

User-centered Evaluation: In addition to automated evaluation metrics, future research could incorporate user-centered evaluation methodologies to assess the quality and utility of generated summaries from a human perspective. Conducting user studies and soliciting feedback from native Amharic speakers could provide valuable insights into the perceived effectiveness and relevance of summarization outputs.

By addressing these recommendations, future researchers can contribute to the ongoing development and refinement of Amharic text summarization techniques, ultimately enabling the creation of more accurate, informative, and contextually relevant summaries tailored to the needs of Amharic-speaking communities.

REFERENCES

[1] Liddy, e. D. (2001). natural language processing. Https://surface.syr.edu/istpub.

[2] N. Kumari. (2022). “hindi text summarization using sequence to sequence neural network,” pp. 1–12.

[3] G. C. V. Vilca and m. A. S. Cabezudo. (2017). “a study of abstractive summarization using semantic representations and discourse level information.” Lect. Notes comput. Sci. (including subser. Lect. Notes artif. Intell. Lect. Notes bioinformatics), vol. 10415.

[4] D. Radev, e. Hovy, and k. Mckeown. (2002). “introduction to the special issue on summarization.” Comput. Linguist., vol. 28, no. 4, pp. 399–408.

[5] A. Redi. (2020). “extractive amharic text summarization using latent dirichlet allocation.” Haramaya university.

[6] A. Ghadimi and h. Beigy. (2022). “hybrid multi-document summarization using pre-trained language models.” Expert syst. Appl., vol. 192, p. 116292.

[7] G. H. Gebremedhin and a. A. Mebrahtu. (2020). “linguistic evolution of ethiopic languages: a comparative discussion,” int. J. Intell. Syst. Appl., vol. 8, no. 1, pp. 1–9.

[8] A. Guadie, d. Tesfaye, and t. Kebebe. (2021). “amharic text summarization for news items posted on social media.” J. Intell. Inf. Syst., vol. 10, no. 6, pp. 125–138.

[9] Y. Liu and m. Lapata. (2019). “text summarization with pretrained encoders.” Arxiv prepr. Arxiv1908.08345.

[10] Miller, D. (2019). “leveraging bert for extractive text summarization on Lectures. ” Arxiv Preprint Arxiv:1906.04165.

[11] Elmadani, K. N., Elgezouli, M., & Showk, A. (2020). “ bert fine-tuning for arabic text summarization. ” Arxiv Preprint Arxiv:2004.14135.

[12] Nuru, k. (2004). amharic news text summarization. Msc thesis. Addis ababa university, addis ababa, ethiopia.

[13] T. D. Demis. (2018). “automatic amharic multi document news text summarization using open text summarizer.”

[14] M. Tamiru and m. Libsie. (2009). “automatic amharic text summarization using latent semantic analysis.” Unpubl. Msc thesis, addis ababa univ.

[15] K. D. Dessalegn and m. Y. Tachbelie. (2017). “graph-based automatic amharic text summarizer.” Preprints.

[16] Mekuria, G. T., & Jagtap, A. S. (2017). automatic amharic text summarization using NLP Parser. International Journal of Engineering Trends and Technology (IJETT), 53(1), 52–58.

[17] A. M. Abu nada, e. Alajrami, a. A. Al-saqqa, and s. S. Abu-naser. (2020). “arabic text summarization using arabert model using extractive text summarization approach.”

[18] N. Moratanch and s. Chitrakala. (2017). “a survey on extractive text summarization,” in 2017 international conference on computer, communication and signal processing (icccsp), pp. 1–6.

[19] N. Nazari and m. A. Mahdavi. (2019). “a survey on automatic text summarization,” j. Ai data min., vol. 7, no. 1, pp. 121–135.

[20] M. T. Maybury. (1995). “generating summaries from event data,” inf. Process. Manag., vol. 31, no. 5, pp. 735–751.

[21] M. Gambhir and v. Gupta. (2017). “recent automatic text summarization techniques: a survey,” artif. Intell. Rev., vol. 47, pp. 1–66.

[22] M. Allahyari et al. (2017). “a brief survey of text mining: classification, clustering and extraction techniques,” arxiv prepr. Arxiv1707.02919.

[23] H. P. Luhn. (1958). “the automatic creation of literature abstracts,” ibm j. Res. Dev., vol. 2, no. 2, pp. 159–165.

[24] G. Carenini, g. Murray, and r. Ng. (2011). “methods for mining and summarizing text conversations.” Synth. Lect. Data manag., vol. 3, no. 3, pp. 1–130.

[25] H. D. Kim, d. H. Park, y. Lu, and c. Zhai. (2012). “enriching text representation with frequent pattern mining for probabilistic topic modeling.” Proc. Am. Soc. Inf. Sci. Technol., vol. 49, no. 1, pp. 1–10.

[26] A. Khan, n. Salim, and h. Farman. (2016). “clustered genetic semantic graph approach for multi-document abstractive summarization.” In 2016 international conference on intelligent systems engineering (icise), pp. 63–70.

[27] T. Hofmann, “probabilistic latent semantic indexing. (1999). ” in proceedings of the 22nd annual international acm sigir conference on research and development in information retrieval, pp. 50–57.".

[28] El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). automatic Text Summarization: A Comprehensive Survey. In Expert Systems with Applications (Vol. 165). Elsevier Ltd. Https://Doi.Org/10.1016/J.Eswa.2020.113679.

[29] P. Gupta, r. Tiwari, and n. Robert. (2016). “sentiment analysis and text summarization of online reviews: a survey.” In 2016 international conference on communication and signal processing (iccsp), pp. 241–245.

[30] D. Sahoo, r. Balabantaray, m. Phukon, and s. Saikia. (2016). “aspect based multi-document summarization.” In 2016 international conference on computing, communication and automation (iccca), pp. 873–877.

[31] M. J. Mohan, c. Sunitha, a. Ganesh, and a. Jaya. (2016). “a study on ontology based abstractive summarization,” procedia comput. Sci., vol. 87, pp. 32–37.

[32] P. Agarwal and s. Mehta. (2018). “empirical analysis of five nature-inspired algorithms on real parameter optimization problems.” Artif. Intell. Rev., vol. 50, pp. 383–439.

[33] M. Mohd, r. Jan, and m. Shah. (2020). “text document summarization using word embedding,” expert syst. Appl., vol. 143, p. 112958.

[34] I. K. Bhat, m. Mohd, and r. Hashmy. (2016). “sumitup: a hybrid single-document text summarizer,” in soft computing: theories and applications: proceedings of socta 2016, volume 1, 2018, pp. 619–634.

[35] K. Takeuchi. (2002). “a study on operations used in text summarization.”

[36] Almeida, F., & Xexéo, G. (2019). word embeddings: a survey. Arxiv preprint arxiv:1901.09069.

[37] Bakarov, A. (2018). a survey of word embeddings evaluation methods. Arxiv preprint arxiv:1801.09536.

[38] Wang, c., nulty, p., & lillis, d. (2020). a comparative study on word embeddings in deep learning for text classification. Acm international conference proceeding series, 37–46. Https://doi.org/10.1145/3443279.3443304.

[39] Heidenreich, h. (2018). introduction to word embeddings. Towards data science.

[40] Eshetu, a., teshome, g., & abebe, t. (2020). learning word and sub-word vectors for amharic (less resourced language). International journal of advanced engineering research and science (ijaers), 7(8), 358-366.

[41] V. Gupta and g. S. Lehal. (2010). “a survey of text summarization extractive techniques,” j. Emerg. Technol. Web intell., vol. 2, no. 3, pp. 258–268.

[42] A. Nenkova and k. Mckeown. (2012). “a survey of text summarization techniques,” min. Text data, pp. 43–76.

[43] M. Allahyari et al. (2017). “text summarization techniques: a brief survey,” arxiv prepr. Arxiv1707.02268.

[44] S. A. Babar and p. D. (2015). Patil, improving performance of text summarization.” procedia comput. Sci., vol. 46, pp. 354–363.

[45] A. Jackson. (2006). “foresight,” drugs and the future: brain science, addiction and society. Pp. 7–10. Doi: 10.1016/b978-012370624-9/50005-0.

[46] Wang, H., Lei, Z., Zhang, X., Zhou, B., & Peng, J. (2016). machine learning basics. deep learning, 98-164.

[47] W. B. Demilie. (2022). “review article comparative analysis of automated text summarization techniques : the case of ethiopian languages,” vol. 2022, no. Dl.

[48] Russell, s. J. (2010). artificial intelligence a modern approach. Pearson education, inc.

[49] C. C. Aggarwal. (2018). “neural networks and deep learning,” springer, vol. 10, no. 978, p. 3.

[50] L. A. (2021). "review of deep learning: concepts, cnn architectures, challenges, applications, future directions," Journal of big data.

[51] T. Trinh, a. Dai, t. Luong, and q. Le. (2018). “learning longer-term dependencies in rnns with auxiliary losses,” in international conference on machine learning. pp. 4965–4974.

[52] S. Hochreiter and j. Schmidhuber. (`1997). “long short-term memory,” neural comput., vol. 9, no. 8, pp. 1735–1780.

[53] S. Hochreiter, y. Bengio, p. Frasconi, and j. Schmidhuber. (2001). “gradient flow in recurrent nets: the difficulty of learning long-term dependencies.” A field guide to dynamical recurrent neural networks. Ieee press in.

[54] Britz, d. (2015). recurrent neural networks tutorial, part 1 – introduction to rnns. Wildml.wordpress,17 sept.2015.web. 04 may 2016.

[55] S. S.-n. N. T. And a. S. N. (2019). "a comparative analysis of forecasting financial time series using arima, lstm, and bilstm," arxiv:1911.09512v1 [cs.lg].

[56] A. F. G. And f. K. (2019). "predicting next word using rnn and lstm cells: stastical language modeling," 2019 fifth international conference on image information processing , no. Aejaz farooq ganai.

[57] Sharma, R., Goel, N., Aggarwal, N., Kaur, P., & Prakash, C. (2019). next word prediction in hindi using deep learning techniques. In 2019 International conference on data science and engineering (ICDSE) (pp. 55-60). IEEE.

[58] A. Vaswani et al. (2017). “attention is all you need,” adv. Neural inf. Process. Syst., vol. 30.

[59] Q. Cao and d. Xiong. (2018). “encoding gated translation memory into neural machine translation,” in proceedings of the 2018 conference on empirical methods in natural language processing, pp. 3042–3047.".

[60] J. Devlin, m.-w. Chang, k. Lee, and k. Toutanova. (2018). “bert: pre-training of deep bidirectional transformers for language understanding,” arxiv prepr. Arxiv1810.04805.

[61] Y. Wu et al. (2016). “google’s neural machine translation system: bridging the gap between human and machine translation. Rapport technique. Yang q. Et wu x., 2006. 10 challenging problems in data mining research,” int. J. Inf. Technol. Decis. Mak., vol. 5, no.".

[62] L. Logeswaran, m.-w. Chang, k. Lee, k. Toutanova, j. Devlin, and h. Lee. (2019). “zero-shot entity linking by reading entity descriptions,” arxiv prepr. Arxiv1906.07348.

[63] Delobelle, P., Winters, T., & Berendt, B. (2020). Robbert: a dutch roberta-based language model. arXiv preprint arXiv:2001.06286.

[64] Yimam, s. M., ayele, a. A., venkatesh, g., gashaw, i., & biemann, c. (2021). introducing various semantic models for amharic: experimentation and evaluation with multiple tasks and datasets. Future internet, volume 13 number 11. Https://doi.org/10.3390/fi.

[65] D. Grantor. (2023). “extractive summarization of scientific publications.”

[66] Y. Dong. (2018). “a survey on neural network-based summarization methods,” arxiv prepr. Arxiv1804.04589.

[67] A. A. Teklewold. (2013). “automatic summarization for amharic text using open text summarizer.” Addis ababa university.

[68] Y. Firdyiwek and d. Yaqob, “the system for ethiopic representation in ascii z t av”.

[69] Y. T. Tessema. (2020). “next word prediction for amharic language using bi-lstm.”

[70] A. Salawu and a. Aseres. (2015). “language policy, ideologies, power and the ethiopian media,” communicatio, vol. 41, no. 1, pp. 71–89.

[71] B. Worku.(2015). “information extraction from amharic language text: knowledge-poor approach,” unpubl. Masters thesis, dep. Comput. Sci. Addis ababa university.

[72] M. Abate and y. Assabie.(2014). “development of amharic morphological analyzer using memory-based learning,” in advances in natural language processing: 9th international conference on nlp, poltal 2014, warsaw, poland, september 17-19.

[73] A. Temesgen and y. Assabie(2013).“development of amharic grammar checker using morphological features of words and n-gram based probabilistic methods,” in proceedings of the 13th international conference on parsing technologies, pp. 106–112.

[74] D. G. Agonafer.(2003). “an integrated approach to automatic complex sentence parsing for amharic text.” Addis ababa university.

[75] G. Tulu. (2020) “bidirectional amharic-afaan oromo machine translation using hybrid approach,” no. March.

[76] A. Teferra and g. Hudson. (2007). essentials of amharic, vol. 18. Rüdiger köppe.

[77] S. A. Teserra. (2007). “bilingual word and chunk alignment: a hybrid system for amharic and english.”

[78] S. M. Mersha, “computer synthesis and analysis of amharic words: an object oriented approach”.

[79] I. Feyera. (2020) “a master’s thesis.”

[80] Nielsen, l. C., & veile, s. L. (2020). automatic text summarization for danish using bert.

[81] Morén, a. (2021). using bert thomas indrias biniam using bert thomas indrias biniam adam morén.

[82] Seid, a. I. (2023). somali extractive text summarization. March. Https://doi.org/10.1109/ocit56763.2022.00063".

[83] Geleta, h. T. (2022). afan oromo text summarization with deep learning approach.

[84] Temesgen, g. (2021). afaan oromo news text summarization using sentence scoring method. St. Mary’s university.

[85] Fanache, l. T. (2020). college of natural sciences afaan oromo text summarization using word embedding college of natural sciences lamesa tashoma fanache advisor : yaregal assabie (phd).

[86] Birhanu, g. A. (2017). college of natural science school of information science automatic text summarizer for tigrinya language.

[87] Kurika, a. E., sundado, t. S., melesse, m., & eligo, w. M. (2022). extractive text summarization for wolaita. International journal, 10(6).

[88] Andarge, t. (2005). the application of machine learning technique(naive bayes) for automatic text summarization: the case of amharic news texts. M.sc thesis, addis ababa university, addiss ababa, ethiopia.

[89] Researchgate., desmond bala bisandu. (2019). design science research methodology in computer science and information systems.

[90] Ernst, o., shapira, o., dagan, i., & levy, r. (2019). re-examining summarization evaluation across multiple quality criteria.

[91] I. K. Nti, o. Nyarko-boateng, and j. Aning. (2021). "performance of machine learning algorithms with different k values in k-fold crossvalidation," int. J. Inf. Technol. Comput. Sci., vol. 13, no. 6, pp. 61–71, doi: 10.5815/ijitcs.

[92] T. Scialom, p.-a. Dray, s. Lamprier, b. Piwowarski, and j. Staiano. (2020). "mlsum: the multilingual summarization corpus," arxiv prepr. Arxiv2004.14900.

[93] A. Helen. (2006). "automatic text summarization for amharic legal judgments," addis ababa univ. Ethiopia.

[94] K. Y. Yang. (2014). "research on information ecosystem construction about e-business," inf. Sci., vol. 32, no. 3, pp. 37–41.

[95] Z. C. Lipton. (2015). "a critical review of recurrent neural networks for sequence learning, corr abs/1506.00019," arxiv prepr. Arxiv1506.00019.

[96] Nielsen, l. C., & veile, s. L. (2020). automatic text summarization for danish using bert. How to summarize news articles traditional methods and ai tools.pdf.

[97] M. Barbella and g. Tortora. (2022). "rouge metric evaluation for text summarization techniques," ssrn electron. J., doi: 10.2139/ssrn.4120317.

APPENDIX

List of Stop-words

ሁሉ	ሰሞኑን	ተባለ	አሳሰበ	እንደተናገሩት
ሁሉም	ሲሆን	ተገለጸ	አሳስበዋል	እንደአስረዱት
ኋላ	ሲል	ተገልጿል	አስፈላጊ	እንደገና
ሁኔታ	ሲሉ	ተጨማሪ	አስገነዘቡ	ወቅት
ሆነ	ቢሆን	ተከናውኗል	አስገንዝበዋል	እንዲሁም
ሆኑ	ብለዋል	ችግር	አብራርተዋል	እንጂ
ሆኖም	ብቻ	ታች	አበራርተው	
ሁል	ብዛት	ትናንት	አስረድተዋል	
ሁሉንም	ቦታ	ነበረች	እስከ	
ላይ	በርካታ	ነበሩ	እባክህ	
ሌላ	በሰሞኑ	ነበረ	እባክሽ	
ሌሎች	በታች	ነው	እባክዎ	
ልዩ	በኋላ	ነይ	አንድ	
መሆኑ	በኩል	ነገር	አንጻር	
ማለት	በውስጥ	ነገሮች	እስኪደርስ	
ማለቱ	በጣም	ናት	እንኳ	
መካከል	ብቻ	ናቸው	እስከ	
የሚገኙ	በተለይ	አሁን	እዚሁ	
የሚገኝ	በተመለከተ	አስታወቀ	እና	
ማድረግ	በተመሳሳይ	አስታውቀዋል	እንደ	
ማን	የተለያየ	አስታውሰዋል	እንደገለጹት	
ማንም	የተለያዩ	እስካሁን	እንደተገለጸው	