Abstractive Text Summarization for Urdu Language

Ph.D. Thesis

by

Muhammad Awais

CIIT/SP19-PCS-012/LHR

COMSATS University Islamabad
Pakistan

Spring 2025

Abstractive Text Summarization for Urdu Language

A thesis submitted to

COMSATS University Islamabad

In partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
in
Computer Science

by
Muhammad Awais

CIIT/SP19-PCS-012/LHR

Department of Computer Science
Faculty of Information Science & Technology

COMSATS University Islamabad
Pakistan

Spring 2025

Abstractive Text Summarization for Urdu Language

This thesis is submitted to the Department of Computer Science in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Computer Science.

Name	Registration Number
Muhammad Awais	CIIT/SP19-PCS-012/LHR

Supervisory Committee

Supervisor	**Member**
Dr. Rao Muhammad Adeel Nawab Lecturer Department of Computer Science COMSATS University Islamabad (CUI) Lahore Campus	Dr. Hamid Turab Mirza Associate Professor Department of Computer Science COMSATS University Islamabad (CUI) Lahore Campus
Member	
Dr. Jawad Shafi Assistant Professor Department of Computer Science COMSATS University Islamabad (CUI) Lahore Campus	

Certificate of Approval

This is to certify that the research work presented in this thesis, entitled “Abstractive Text Summarization for Urdu Language” was conducted by Mr. Muhammad Awais, CIIT/SP19-PCS-012/LHR under the supervision of Dr. Rao Muhammad Adeel Nawab. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the Department of Computer Science, COMSATS University Islamabad Lahore Campus, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the field of Computer Science.

Student Name: Muhammad Awais Signature: _________________

Examinations Committee:

External Examiner 1: Name (Designation & Office Address)	External Examiner 2: Name (Designation & Office Address)
…………………………………… ……………………………………	…………………………………… ……………………………………
Dr. Rao Muhammad Adeel Nawab Supervisor Department of Computer Science COMSATS University Islamabad (CUI) Lahore Campus	Dr. Farooq Ahmad Head Department of Computer Science COMSATS University Islamabad (CUI) Lahore Campus
Prof. Dr. Manzoor Ilahi Tamimy Chairperson Department of Computer Science COMSATS University Islamabad (CUI)	Prof. Dr. Sohail Asghar Dean Faculty of Information Science & Technology COMSATS University Islamabad (CUI)

Author’s Declaration

I Muhammad Awais, CIIT/SP19-PCS-012/LHR, hereby state that my Ph.D. thesis titled “Abstractive Text Summarization for Urdu Language” is my own work and has not been submitted previously by me for taking any degree from this University i.e. COMSATS University Islamabad or anywhere else in the country/world.

At any time if my statement is found to be incorrect even after I graduate the University has the right to withdraw my Ph.D. degree.

Date: ____________________

Muhammad Awais
CIIT/SP19-PCS-012/LHR

Plagiarism Undertaking

I solemnly declare that the research work presented in the thesis titled “Abstractive Text Summarization for Urdu Language” is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero-tolerance policy of HEC and COMSATS University Islamabad towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake if I am found guilty of any formal plagiarism in the above titled thesis even after award of Ph.D. Degree, the University reserves the right to withdraw/revoke my Ph.D. degree and that HEC and the university has the right to publish my name on the HEC/university website on which names of students are placed who submitted plagiarized thesis.

Date: ____________________

Muhammad Awais
CIIT/SP19-PCS-012/LHR

Certificate

It is certified that Muhammad Awais, CIIT/SP19-PCS-012/LHR, has carried out all the work related to this thesis under my supervision at the Department of Computer Science, COMSATS University Islamabad, Lahore Campus and the work fulfills the requirements for the award of the degree of Doctor of Philosophy in Computer Science.

Date: ____________________

Supervisor:

Dr. Rao Muhammad Adeel Nawab
Lecturer
Department of Computer Science
COMSATS University Islamabad
Lahore Campus

Dedication

To Almighty ALLAH and the Holy Prophet Muhammad (Peace Be Upon Him)

&

To my parents especially my late father (Mr. Ahtisham ud Din Qureshi), wife, daughters (Zara Awais \& Zainab Fatima), and family members especially Mr. Iftikhar ud Din Qureshi \& Mrs. Sabahat Anwar.

Acknowledgements

I am very grateful to my beloved supervisor Dr. Rao Muhammad Adeel Nawab. I am also very thankful to the respected H.o.D Dr. Farooq Ahmad, and Ph.D. research coordinator Dr. Ghullam Rasool for his continuous assistance regarding different administrative matters.

Further, a bundle of thanks to my beloved Mother and Wife, this research work would not be possible without their continuous support and prayers.

I would also like to extend my deepest gratitude to my seniors Mr. Muhammad Shakeel and Dr. Iqra Muneer, and my lab fellows, Ms. Hira Arshad and Mr. Ahmed Mehmood, for their support and guidance. Last but not least, I would like to thank my colleagues, Mr. Mudassar, Mr. Arslan Ilyas, Mr. Faseeh Ullah Khan, Mr. Zubair Khalid, and Dr. Umair Rashid for their support and motivation.

Muhammad Awais
CIIT/SP19-PCS-012/LHR

Abstract

Abstractive Text Summarization for Urdu Language

By

Muhammad Awais

The task of Abstractive Text Summarization (ATS) aims to generate concise and coherent summaries from lengthy documents automatically. With the rapid growth of digital content, ATS is crucial for enhancing information retrieval, decision-making, and personalized content delivery. While substantial progress has been made in ATS for high-resource languages like English, Urdu remains underexplored due to limited large-scale corpora and intrinsic linguistic complexities. Existing Urdu resources are small-scale or restricted to extractive summarization, limiting robust development and benchmarking. To address this gap, this research introduces a comprehensive benchmark corpus and evaluates state-of-the-art models tailored for Urdu ATS.

The first major contribution of this study is the development of UATS-23, a large-scale, high-quality benchmark corpus for Urdu abstractive text summarization. UATS-23 consists of approximately 2.067 million Urdu news articles paired with abstractive summaries from domains like sports, entertainment, business, national, international, science, and health. Comprehensive linguistic analyses confirm the corpus's richness for summarization research. UATS-23 is publicly available under a Creative Commons license to support future Urdu NLP research.

UATS-23 consists of approximately 2.067 million Urdu news articles along with their corresponding abstractive summaries (headlines) collected from diverse domains, including sports, entertainment, business, national, international, science, and health. A comprehensive linguistic analysis across compression ratios, lexical diversity, keyword overlap, and syntactic transformations further validates the corpus's richness and utility for research. UATS-23 is publicly released under a Creative Commons license, serving as a foundational resource to stimulate future research in low-resource Urdu NLP applications.

The second major contribution of this research is the systematic evaluation of state-of-the-art deep learning models, transformer-based models, and LLMs on the proposed UATS-23 corpus. Baseline deep learning models, including LSTM, Bi-LSTM, GRU, and Bi-GRU, were developed and rigorously evaluated, both with and without attention mechanisms. Transformer-based models, namely Bidirectional Auto-Regressive Transformers (BART), and LLMs, namely Generative Pre-trained Transformer (GPT-3.5), and DeepSeek-R1, were also applied. Extensive experiments employing standard evaluation metrics, specifically ROUGE-1, ROUGE-2, and ROUGE-L, revealed that the GRU model with attention achieved the highest summarization performance (ROUGE-1 = 46.7, ROUGE-2 = 24.1, ROUGE-L = 48.7), demonstrating the effectiveness of attention-based GRU models for Urdu summarization tasks.

The third contribution of this study is the linguistic analysis of the UATS-23 corpus to identify the morphological, syntactic, and semantic characteristics of Urdu text that influence the performance of ATS models, thereby informing model design, tokenization strategies, and evaluation alignment for low-resource and morphologically rich languages.

Contents

1 Introduction **1**
1.1 Introduction 1
1.2 Types of Automatic Text Summarization 3
1.3 Thesis Focus 6
1.4 Problem Statement 7
1.5 Research Questions 8
1.6 Contributions 8
1.7 Main Findings 9
1.8 Thesis Outline 10
1.8.0.1 Chapter 2: Literature Review 10
1.8.0.2 Chapter 3: Proposed Corpus for Urdu Abstractive Text Summarization 10
1.8.0.3 Chapter 4: Proposed methods for Urdu Abstractive Text Summarization 11
1.8.0.4 Chapter 5: Evaluation of Proposed Methods on UATS-23 Corpus 11
1.8.0.5 Chapter 6: Conclusion 11
1.8.0.6 Chapter 7: References 11
1.9 Publications 11
1.9.0.1 Published Work 11

2 Literature Review **12**
2.1 Introduction 12
2.2 Corpora for Abstractive Text Summarization 12
2.2.1 Corpora for English Languages 12
2.2.2 Corpora for Other Languages 18
2.2.3 Corpora for Urdu Language 19
2.3 Methods for Abstractive Text Summarization 22
2.3.1 Deep Learning Methods for Abstractive Text Summarization 23
2.3.2 Transformer Methods for Abstractive Text Summarization . 31
2.3.3 LLM Methods for Abstractive Text Summarization 34
2.4 Evaluation Measures 39
2.5 Chapter Summary 40

3 Proposed Corpus for Urdu Abstractive Text Summarization **41**
3.1 Introduction 41
3.2 UATS-23 Corpus 41
3.2.1 Raw Data Collection 43
3.2.2 Data Cleaning and Pre-processing 47

3.2.3 Corpus Standardization 49
3.2.4 Corpus Characteristics 50
3.2.5 Linguistic Analysis of UATS-23 Corpus 55
3.3 Chapter Summary 61

4 Methods for Abstractive Text Summarization for Urdu Language . 62
4.1 Introduction 62
4.2 Deep Learning Models 63
4.2.1 Long Short-Term Memory 63
4.2.1.1 Architecture of Long Short-Term Memory 64
4.2.2 Gated Recurrent Unit 67
4.2.2.1 Architecture and of Gated Recurrent Unit 68
4.2.3 Bi-directional Long Short-Term Memory 70
4.2.3.1 Architecture of Bi-directional Long Short-Term Memory 71
4.2.4 Bi-directional Gated Recurrent Unit 73
4.2.4.1 Architecture of Bi-directional Gated Recurrent Unit . 74
4.2.5 Encoder-Decoder Models with Attention 76
4.2.5.1 Architecture of Encoder-Decoder Models with Attention 77
4.2.6 Parameter Settings for Deep Learning Models 81
4.3 Transformer-Based Models 82
4.3.1 Bidirectional Auto-Regressive Transformers 83
4.3.1.1 Architecture and Parameter Setting for Bidirectional Auto-Regressive Transformers 83
4.4 Large Language Models 86
4.4.1 Generative Pre-trained Transformer (GPT-3.5) 87
4.4.2 DeepSeek-R1 89
4.4.3 Parameter Settings for LLMs 92
4.5 Challenges in Urdu ATS Implementation 93
4.6 Chapter Summary 94

5 Evaluation of Proposed Methods on UATS-23 Corpus 96
5.1 Introduction 96
5.2 Experimental Setup 96
5.2.1 Dataset 97
5.2.2 Techniques 97
5.2.3 Evaluation Methodology 104
5.2.4 Evaluation Measures 105
5.3 Results and Analysis 110
5.4 Chapter Summary 117

6 Conclusion and Future Work 118
6.1 Thesis Summary 118
6.1.1 Thesis Contributions 120
6.2 Future Work 121

References **123**

List of Figures

Figure 1.1 Automatic Text Summarization Pipeline 2

Figure 3.1 Steps Involved in UATS-23 Corpus Generation Process 42

Figure 4.1 RNN based Encoder-Decoder Architecture 64
Figure 4.2 Proposed Methodology Using Encoder-Decoder Models 73
Figure 4.3 RNN based Encoder-Decoder Architecture with Attention 77
Figure 4.4 Proposed Methodology Using Encoder-Decoder Models with Attention 80

List of Tables

Table 2.1 Summary of Corpora for Abstractive Text Summarization 22
Table 2.2 Summary of Techniques/Methods for Abstractive Text Summarization 36

Table 3.1 UATS-23 Corpus Collection Sources 45
Table 3.2 Main Statistics of UATS-23 Corpus 46
Table 3.3 UATS-23 Corpus Characteristics 50
Table 3.4 Comparative Benchmarking of UATS-23 and Prominent ATS Corpora 52
Table 3.5 Average Scores for Linguistic Analysis Metrics 60

Table 4.1 Summary Comparison of Baseline Deep Learning Models 81
Table 4.2 Training and Inference Configuration for All Models 93

Table 5.1 Extended Experimental Configuration Summary for All Models . . . 102
Table 5.2 Optimal Hyperparameter Configuration for GRU + Attention Model . 104
Table 5.3 Manual ROUGE Score Computation Example with Detailed Calculations 109
Table 5.4 Results Obtained by Applying Various State-of-the-Art Baseline Deep Learning Models on Our Proposed UATS-23 Corpus 110
Table 5.5 Examples of Good BART Model Performance 112
Table 5.6 Examples of Good GPT-3.5 Model Performance 112
Table 5.7 Examples of Good DeepSeek-R1 Model Performance 113
Table 5.8 Detailed Experimental Results of all context-limited, context-aware, transformers, and LLM Models 114

LIST OF ABBREVIATIONS

Abbreviation	Meaning
AI	Artificial Intelligence
NLU	Natural Language Understanding
NLG	Natural Language Generation
ALBERT	A Lite BERT
ATS	Abstractive Text Summarization
BERT	Bidirectional Encoder Representations from Transformers
Bi-GRU	Bidirectional Gated Recurrent Units
Bi-LSTM	Bidirectional Long Short-Term Memory
BLEU	Bilingual Evaluation Understudy
BART	Bidirectional Auto-Regressive Transformers
CNN	Cable News Network
DM	Daily Mail
DUC	Document Understanding Conference
EM	Expectation Maximization
ETS	Extractive Text Summarization
GPT	Generative Pretrained Transformers
GRU	Gated Recurrent Units
GPU	Graphics Processing Unit
HMM	Hidden Markov Model
LCSTS	Large-scale Chinese Short Text Summarization
LLMs	Large Language Models
LSTM	Long Short-Term Memory
METEOR	Metric for Evaluation of Translation with Explicit Ordering
MLE	Maximum Likelihood Estimation

Continued on next page

Table 1 Continued from previous page

Abbreviation	Meaning
NLP	Natural Language Processing
NYT	New York Times
PEGASUS	Pre-training with Extracted Gap-sentences for Abstractive SUmmarization Sequence-to-sequence
RNN	Recurrent Neural Network
ROBERTa	Robustly Optimized BERT Approach
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
Seq2Seq	Sequence to Sequence
SMT	Statistical Machine Translation
SVM	Support Vector Machines
TAC	Text Analysis Conference
TER	Translation Edit Rate
UATS	Urdu Abstractive Text Summarization
vCPU	Virtual Central Processing Unit
API	Application Programming Interface
LoRA	Low-Rank Adaptation
QLoRA	Quantized Low-Rank Adaptation
BERTScore	BERT-based Evaluation Score
PEFT	Parameter Efficient Fine-Tuning
CR	Compression Ratio

Chapter 1
Introduction

1.1 Introduction

Text summarization is a pivotal task in Natural Language Processing (NLP), aimed at condensing extensive textual content into concise and informative summaries while preserving essential semantic information. With the exponential growth of digital content across platforms and languages, summarization techniques have become increasingly critical for enhancing information retrieval, supporting timely decision-making, and facilitating efficient navigation through large-scale textual data [1]. Applications of text summarization span diverse domains, including the generation of news headlines, the abstraction of scientific and legal documents, the summarization of social media content, and the optimization of search engine outputs. Furthermore, summarization plays an integral role in intelligent systems such as virtual assistants, chatbots, automated reporting tools, and personalized content delivery platforms. Its ability to reduce cognitive load and improve content accessibility underscores its growing importance in information-intensive and time-sensitive environments.

To systematically perform the task of summarization, the text summarization pipeline is typically organized into three core stages: NLP, Natural Language Understanding (NLU), and Natural Language Generation (NLG), as illustrated in Figure 1.1. In the first stage, raw textual input undergoes preprocessing, including tokenization, sentence segmentation, and co-reference resolution to ensure syntactic and semantic clarity. This is followed by the NLU phase, where the system extracts semantic representations, identifies key concepts, and models discourse structures to understand the underlying meaning of the text. The final stage, NLG, involves the construction of a coherent and concise summary based on the internal representation generated during NLU. In modern computational systems, this pipeline is often implemented using neural architectures such as encoder-decoder models, which embed input sequences and generate summaries through

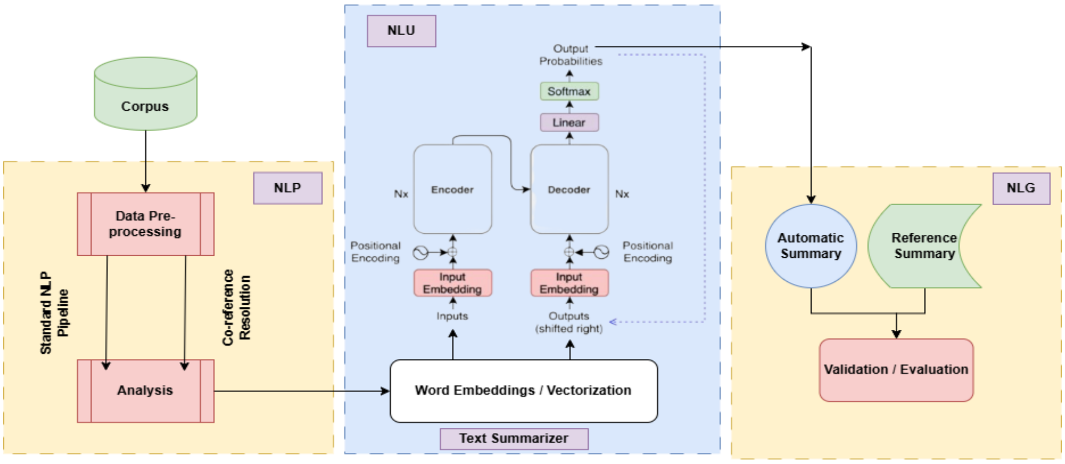

Figure 1.1: Automatic Text Summarization Pipeline

attention-based decoding mechanisms. The quality of the generated summaries is typically assessed using evaluation metrics that compare them against human-written references to determine lexical and semantic alignment.

While the automated text summarization pipeline offers efficiency and scalability, it is essential to contrast it with manual summarization approaches. Human-generated summaries excel in contextual interpretation and semantic nuance but are inherently constrained by time, consistency, and subjectivity. Early automatic methods relied on statistical and linguistic features, such as keyword extraction and sentence ranking. However, the field has since evolved with the advent of machine learning, particularly deep learning, which introduced neural architectures such as recurrent neural networks (RNNs), long short-term memory (LSTM) models, and transformers that enable modeling of complex semantic and contextual dependencies within text [2], [3], [4]. Recent advancements, including reinforcement learning and unsupervised techniques, have further improved the fluency, factual consistency, and multilingual adaptability of ATS systems, making them increasingly viable alternatives to manual summarization in practical applications.

The focus of this research is on automatic text summarization, with particular emphasis on addressing the challenges posed by low-resource languages.

1.2 Types of Automatic Text Summarization

Automatic text summarization techniques broadly fall into three major categories: extractive summarization, abstractive summarization, and hybrid summarization approaches. Each type follows distinctive strategies and methodologies to generate summaries, offering varied advantages based on the requirements of the summarization task. Extractive Text Summarization (ETS) involves selecting important sentences, clauses, or phrases directly from the original document to form a summary. The selection typically relies on identifying sentences that best represent the overall content of the original text. Extractive methods apply statistical and linguistic features, including frequency of keywords, sentence relevance, sentence location within the text, and semantic relationships, to determine the most significant sentences to be included in the summary [5].

For instance, consider the following example source text:

Example: *"Climate change is a pressing global challenge. Many countries are experiencing severe droughts and intense storms. Governments world wide are adopting policies to reduce carbon emissions and promote renewable energy solutions. Citizens are increasingly advocating for sustainable practices to safeguard the environment."*

An extractive summarization technique might identify the key sentences as:

Extractive Summary:*"Climate change is a pressing global challenge. Many countries are experiencing severe droughts and severe weather conditions."*

Extractive approaches predominantly utilize supervised machine learning algorithms, namely Support Vector Machines (SVM), Decision Trees, and deep neural networks like Bidirectional Encoder Representations from Transformers (BERT), and RoBERTa, to enhance accuracy and reliability in identifying and extracting sentences [6], [7].

Abstractive Text Summarization (ATS) is a task that aims to generate an automatic summary that includes words and phrases representing the most important information from the source text [**0**]. Due to recent technological developments, large amounts of digital data are generated every day. Extracting and combining useful information from large volumes of data is a non-trivial task. Automatic generation of abstractive summaries can help us to extract and combine useful information from one or more sources. ATS has an assortment of real-world applications, including the generation of news headlines, a summary of research articles, the moral of the stories, media marketing, search engine optimization, financial research, social media marketing, question-answering systems, and chatbots.

This approach necessitates an in-depth semantic understanding and sophisticated natural language generation capabilities. Recent advancements in ATS leverage deep learning models namely LSTM, Gated Recurrent Units (GRU), and transformers-based models namely Bidirectional Auto-Regressive Transformers (BART) [8] and Generative Pretrained Transformers (GPT) [9].

For example, given a source text as:

Example: *“Economic growth in developing nations has accelerated due to the advancement of technology. Many countries have invested heavily in digital infrastructure, improving internet access and enabling innovation across multiple industries."*

An abstractive summary might be generated as:

Abstractive Summary: *“Technological improvements and digital investments have significantly boosted economic growth in developing countries."*

Prominent abstractive summarization models often employ encoder-decoder architectures, attention mechanisms, and transformer-based approaches like BART, GPT, and T5, which dynamically understand contextual and semantic relationships, enhancing the coherence and fluency of the summaries generated.

Hybrid text summarization combines the strengths of both extractive and abstractive techniques to produce summaries that are coherent, fluent, and contextually relevant. Typically, hybrid methods initially apply extractive techniques to identify critical sentences or phrases from the source text, and subsequently use abstractive methods to paraphrase these selected portions into a concise and fluent summary [10].

As an illustration, a hybrid summarization model first extracts key sentences from the original document and then paraphrases them abstractively. Suppose we have the following text:

Example: *“Machine learning has transformed data analysis. Algorithms learn from data patterns to predict future events accurately. This technology has impacted industries including finance, healthcare, and transportation significantly."*

A hybrid summarization technique might initially extract important sentences, such as:

Step 1 - Extractive Summary: *“Machine learning has transformed data analysis. This technology significantly impacts industries like finance, healthcare, and transportation."*

Then, it abstractively paraphrases them into a concise and coherent summary:

Step 2 - Abstractive Summary of Step 1: *“Machine learning enables accurate future predictions from data, significantly influencing industries like finance, healthcare, and transportation."*

The hybrid approach typically utilizes extractive methods (e.g., BERT-based sentence extraction) followed by ATS models (e.g., BART, GPT) to leverage both precise content selection and fluent natural language generation capabilities [10].

The focus of this research work is on ATS, which aims to generate summaries that are not merely subsets of the source text but are rewritten in a concise and semantically coherent form. Compared to extractive and hybrid approaches, ATS poses greater linguistic and computational challenges, as it requires deep semantic understanding, content abstraction, and fluent natural language generation. Key challenges in ATS include ensuring factual consistency, handling long-range dependencies, managing redundancy, and preserving the original intent of the source content, issues that are further exacerbated in low-resource languages. Moreover, ATS can be explored in both monolingual and cross-lingual settings. In monolingual ATS, the summary is generated in the same language as the source document, whereas cross-lingual ATS involves generating summaries in a different target language, adding complexities related to translation and semantic alignment.

The focus of this thesis is on ATS for mono-lingual settings.

1.3 Thesis Focus

The main focus of this research work is on investigating the problem of ATS for the Urdu language. The problem of ATS has been mainly explored for English and some other languages [11, 12, 13, 5, 2, 14, 15, 16, 17, 18, 19]. The issues of application of ATS, on the other hand, South Asian languages, notably the Urdu language, have not been well studied. Urdu is the most frequently spoken language in South Asia, and it has more than 170 million speakers around the world[1]. Persian, Arabic, and other South Asian languages [20] have all influenced Urdu's lexicon and syntax. Urdu is an Indo-Aryan language. The Urdu language has a diverse morphology, and some of its words (nouns and verbs) may have up to 40 variants, making it difficult to mechanically analyze [21]. Furthermore, the inherent orthographic ambiguity, lack of standardized tokenization rules, and complex script rendering exacerbate difficulties in text normalization and segmentation. Despite the growing availability of digitized Urdu content, the language is still considered low-resourced from an NLP perspective, primarily due to the scarcity of large-scale annotated corpora and pre-trained language models. These limitations hinder the development of robust neural architectures for ATS and complicate the evaluation of summarization quality. Nonetheless, recent research efforts have begun to address these gaps by developing foundational resources and tools for Urdu language processing [22], with benchmark corpora playing a vital role in enabling systematic development, evaluation, and comparison of NLP methods across various tasks [23].

Recent years have seen growing interest in the development of summarization corpora for the Urdu language, particularly in the abstractive and extractive paradigms. Initial efforts, such as the manually created Urdu summary corpus introduced in [24], comprise only 50 article-summary pairs, providing a foundational yet limited resource for abstractive summarization. This dataset was later extended for extractive summarization tasks with 600 annotated articles [23], enabling exploration of statistical techniques but still lacking the scale required for deep learning applications. Beyond written news articles, corpora such as the CLE Meeting Corpus [25] and UrduMASD [26] have targeted

[1] According to Ethnologue: www.ethnologue.com/language/urd Last Visited: 25-Sep-2024

speech and video domains, contributing valuable but noisy and unstructured data with relatively modest summarization performance. Despite these contributions, the overall landscape of Urdu summarization resources is constrained by small dataset sizes, limited domain diversity, and the absence of standardized benchmarks. Most available corpora remain insufficient to train, fine-tune, or rigorously evaluate advanced neural or transformer-based architectures, highlighting a critical research gap in scalable and standardized corpus development for low-resource languages like Urdu.

On the methodological side, early research on Urdu Abstractive Text Summarization (UATS) has predominantly focused on traditional statistical approaches and basic neural models, with more recent studies incorporating deep learning techniques. LSTM-based encoder-decoder models augmented with attention mechanisms have shown promising results in modeling Urdus rich syntactic and semantic structures [27, 28], achieving ROUGE-1 scores in the range of 4344. While these architectures mark a shift towards more sophisticated modeling, the exploration of advanced techniques such as transformers and LLMs remains limited. Few studies have investigated the efficacy of pre-trained multilingual transformers, and those that have, such as ur_mT5 applied to the CLE Meeting Corpus, have been constrained by data quality and domain specificity [25]. Additionally, there has been minimal effort to evaluate recent state-of-the-art models (e.g., BART, GPT, DeepSeek-R1) on Urdu datasets, leaving their comparative performance largely unexplored. These methodological gaps are compounded by the lack of standardized experimental setups and comprehensive benchmark evaluations, underscoring the need for systematic assessment and adaptation of modern summarization frameworks tailored to the linguistic challenges of Urdu.

1.4 Problem Statement

Based on the extensive literature review carried out in his study and after examining the limitations of existing work, the problem statement of this research is as follows: this research aims to develop a large benchmark corpus and state-of-the-art deep learning, transfer learning, and LLMs for Urdu ATS for its potential applications in news headline generation, timeline or event summarization and article abstract generation.

1.5 Research Questions

In light of the problem statement, this study is guided by the following research questions:

- What are the main challenges of ATS for the Urdu language?
- How can a large-scale benchmark corpus be developed for Urdu ATS?
- Which methods can be effectively applied or developed for Urdu ATS?
- Which methods perform best on the UATS-23 corpus for Urdu ATS?

1.6 Contributions

The major contributions of this study are as follows:

1. **Development of Benchmark UATS-23 Corpus**
   A large-scale benchmark, UATS-23 corpus, was developed to address the critical lack of resources for UATS. It comprises over 2 million Urdu news articles paired with abstractive summaries across diverse domains. Curated through a rigorous pre-processing pipeline, the corpus reflects a 4.5% compression ratio and is publicly available under a Creative Commons license. UATS-23 serves as a foundational resource for training, benchmarking, and advancing summarization models in the low-resource context of Urdu.

2. **Development of baseline deep learning models on the proposed UATS-23 corpus**
   Six baseline deep learning models were implemented and evaluated on the UATS-23 corpus to establish benchmarks for Urdu ATS. These included LSTM, Bi-LSTM, GRU, Bi-GRU, and their attention-enhanced variants. Among them, the GRU with attention mechanism achieved the best performance, with

ROUGE-1, ROUGE-2, and ROUGE-L F_1-scores of 46.7, 24.1, and 48.7, respectively. The results highlight the effectiveness of attention mechanisms in improving semantic accuracy and contextual relevance in Urdu summaries, though with higher computational demands.

3. **Development of state-of-the-art transformer-based model and LLMs on the proposed UATS-23 corpus**
   To investigate transfer learning for Urdu abstractive summarization, pre-trained transformer models were fine-tuned on the UATS-23 corpus. BART, adapted on 100 Urdu articles, showed improved fluency and context over RNN models, though ROUGEs limitations highlighted the need for semantic evaluation. LLMs like GPT-3.5 and DeepSeek-R1 were assessed in zero-/few-shot settings, with DeepSeek-R1 achieving better semantic accuracy but demanding higher computational resources. These results highlight the promise of LLMs for Urdu ATS and the importance of efficient fine-tuning approaches.

4. **Direct comparison of state-of-the-art methods on the proposed UATS-23 corpus**
   A comprehensive evaluation of state-of-the-art methods was conducted on the UATS-23 corpus, including baseline deep learning models, transformer-based architectures (e.g., BART), and LLMs (e.g., GPT-3.5, DeepSeek-R1). Using standard metrics, the comparison revealed key performance differences, highlighting the effectiveness of attention mechanisms and the superior semantic capabilities of LLMs. This direct benchmarking demonstrates the utility of UATS-23 for evaluating summarization methods in low-resource settings, i.e., Urdu Language.

1.7 Main Findings

The main findings of this research are as follows:

- **Finding 1:** GRU with Attention outperforms other State-of-the-art methods on our proposed UATS-23 corpus.

- **Finding 2:** Collecting and storing data for UATS is a challenging task, even if it is done for the Journalism case, where the huge amount of data is readily available on websites.

- **Finding 3:** Integrating domain-specific embeddings can significantly enhance the performance of deep learning models by providing semantically richer representations. With the availability of large domain-specific datasets, it becomes feasible to employ pre-trained word embeddings tailored to that specific domain.

- **Finding 4:** Attention-based mechanisms integrated with GRUs and LSTMs improve context retention but are computationally expensive, making training on large Urdu datasets challenging

- **Finding 5:** The highest ROUGE (F_1) score of 0.46, achieved using GRU with Attention, highlights that while the model performs well, there is still significant room for improvement in UATS, particularly in enhancing semantic coherence and factual consistency.

1.8 Thesis Outline

The rest of the thesis contains the following six chapters:

1.8.0.1 Chapter 2: Literature Review

The chapter begins with a brief overview of ATS's history before going into great depth regarding the many corpora and ATS techniques that are currently in use. The chapter also covers the development of benchmarks and corpora, as well as how the ATS problem has been addressed in recent prior research. Additionally, the chapter also provides the assessment metrics for the ATS systems.

1.8.0.2 Chapter 3: Proposed Corpus for Urdu Abstractive Text Summarization

The efforts to create a large benchmark corpus for a language with limited resources are discussed in this chapter. The chapter explains how the proposed UATS-23 corpus was developed, covering data collection, data cleaning & pre-processing, and data standardization. The chapter provides a full discussion of the proposed corpus, i.e., UATS-23.

1.8.0.3 Chapter 4: Proposed methods for Urdu Abstractive Text Summarization

To demonstrate how the proposed corpus can be used for the development, evaluation, and comparison of baseline deep learning methods and LLMs applied to our proposed corpus, this chapter describes these UATS methods in detail.

1.8.0.4 Chapter 5: Evaluation of Proposed Methods on UATS-23 Corpus

The fifth chapter of this thesis reports the details of the performed experiments for UATS task, carried out on the proposed UATS-23 corpus. To demonstrate how they might be used in the evaluation of the UATS system for the low-resourced language, i.e., Urdu, a wide range of various methodologies, grouped into two categories, are used for the suggested corpus. Evaluation results indicated that our proposed GRU with Attention outperforms others.

1.8.0.5 Chapter 6: Conclusion

This chapter concludes the thesis and also highlights directions for future work.

1.8.0.6 Chapter 7: References

This chapter shows the complete details of references.

1.9 Publications

Following is the list of publication(s) produced during this research work:

1.9.0.1 Published Work

- **M. Awais** & R.M.A. Nawab, Abstractive Text Summarization for the Urdu Language: Data and Methods, *IEEE Access*, 10 Feb 2024.

Chapter 2
Literature Review

2.1 Introduction

The previous chapter discusses ATS, its importance, and its applications. Additionally, the chapter also discussed the UATS-23 benchmark corpus used for measuring the performance of state-of-the-art methods. Moreover, the chapter sheds light on the fact that most of the standard assessment corpora are developed for English and some other languages. Consequently, supporting resources for English and state-of-the-art methods for most languages are accessible.

In this chapter, the existing corpora and methods will be discussed, along with the evaluation measures for UATS.

2.2 Corpora for Abstractive Text Summarization

2.2.1 Corpora for English Languages

In the literature, many researchers have made efforts to develop standard evaluation resources and techniques for the ATS task. We provide a thorough analysis of the available ATS corpora, methodologies, and techniques below.

One of the remarkable efforts to foster research in the field of text summarization is a series of competitions (or shared tasks) organized under the umbrella of Document Understanding Conference (DUC[1]) / Text Analysis Conference (TAC) [11]. The most important end result of these competitions is the creation of a set of gold standard corpora for the ETS and the ATS tasks for both single and multi-document tasks. The DUC was organized from 2001 to 2007, and the text summarization corpora were developed for the English and Arabic languages. The abstractive text summarization corpora were created by asking domain experts to manually write summaries of source documents. The size

[1] https://duc.nist.gov/ Last Visited: 25-Sep-2024

of corpora developed for the DUC is small and varies from 600 to 1250 documents [29]. Despite their limited size, these datasets became foundational in benchmarking early ATS systems, especially due to the high-quality human-generated summaries, which ensured consistency and reliability in comparative evaluation.

In addition to the DUC corpora, researchers have made an effort to develop standard evaluation resources for the ATS task. In a study, [14] developed Cable News Network and Daily Mail (CNN/DM) corpus for the English ATS task. The CNN/DM corpus is constructed by modifying an existing corpus developed for the question-answering task [30]. This corpus comprises of 3,11,672 news articles and their corresponding multi-summaries. The summaries in this corpus are typically extracted from the highlights of news articles, making them semi-abstractive. Although these summaries are not completely human-written from scratch, they still serve as effective proxies for training neural summarization models, especially due to the large size and diversity of the corpus. The authors applied RNN encoder-decoder with hierarchical and temporal attention. The best results were obtained using RNN with a temporal attention model, with ROUGE-1 = 35.4, ROUGE-2 = 13.3, and ROUGE-L = 32.6. This corpus has since become a widely adopted benchmark in ATS research, serving as a primary dataset for evaluating both extractive and abstractive models due to its scalability and accessibility.

Overall, the development of English ATS corpora has evolved from small-scale, high-quality datasets like DUC to large-scale semi-abstractive corpora, i.e., CNN/DM. These datasets collectively provide a comprehensive foundation for training, testing, and benchmarking ATS systems across multiple domains and summary styles, thus enabling consistent advancement in the field.

In [12], authors proposed the New York Times (NYT) corpus for automatic text summarization, metadata extraction, information retrieval, and extraction tasks. The NYT corpus contains a total of 1.8 million English news articles extracted from various online sources. The NYT corpus was annotated for the ETS task, comprising of 6,54,759 English news articles and their corresponding summaries. These summaries are written by library scientists and are mainly used for extractive text summarization tasks [5]. Authors [5] applied Bi-LSTM based multiple extractive and compressive summarizer models on CNN/DM [14] and NYT [12] corpora. The best results (on NYT corpus) were obtained using joint extractive and compressive summarizer approach with ROUGE-1 = 45.5, ROUGE-2 = 25.3, and ROUGE-L = 38.2. These results demonstrated the effectiveness of integrating compressive strategies with extraction models, especially on professionally curated corpora such as NYT.

Another research group used NYT corpus for the ATS task for the first time by using an article-abstract pair of the NYT corpus [2]. The authors applied maximum likelihood and reinforcement learning methods with and without intra-attention techniques on CNN/DM and NYT corpora. The best results of ROUGE were obtained on the NYT corpus using reinforcement learning with an intra-attention approach (ROUGE-1 = 41.1, ROUGE-2 = 15.7, and ROUGE-L = 39.0). This experiment highlighted the potential of hybrid learning strategies and attention mechanisms in capturing long-range dependencies for generating coherent abstractive summaries from lengthy news articles.

In [15], the authors proposed a newsroom corpus for the English ATS task. The newsroom corpus contains more than 1.3 million English articles and their corresponding summaries. Newsroom corpus is significantly diverse because the articles are written by different authors from multiple news domains, and their corresponding summaries are written for the HTML and social media metadata. The newsroom corpus is further divided into three subsets to be used for the ATS, ETS, and mixed ATS and ETS tasks. The best results were obtained using the pointer-generator model on all subsets of the newsroom corpus (ROUGE-1 = 39.1, ROUGE-2 = 27.9, and ROUGE-L = 36.1 for ETS task, ROUGE-1 = 25.4, ROUGE-2 = 11.0, and ROUGE-L = 21.0 for mixed ATS and ETS task, and ROUGE-1 = 14.6, ROUGE-2 = 02.2, and ROUGE-L = 11.4 for the ATS task).

In [15], the authors proposed a newsroom corpus for the English ATS task. The newsroom corpus contains more than 1.3 million English articles and their corresponding summaries. The newsroom corpus is significantly diverse because the articles are written by different authors of multiple news domains and their corresponding summaries are written for the HTML and social media metadata. The dataset encapsulates various writing styles and summary intents, making it one of the few corpora that capture a wide spectrum of summarization phenomena. The newsroom corpus is further divided into three subsets to be used for the ATS, ETS, and mixed ATS and ETS tasks. The best results were obtained using the pointer-generator model on all subsets of the newsroom corpus (ROUGE-1 = 39.1, ROUGE-2 = 27.9, and ROUGE-L = 36.1 for ETS task, ROUGE-1 = 25.4, ROUGE-2 = 11.0, and ROUGE-L = 21.0 for mixed ATS and ETS task, and ROUGE-1 = 14.6, ROUGE-2 = 02.2, and ROUGE-L = 11.4 for the ATS task). These outcomes indicate that the complexity of summary style has a direct influence on model performance, with extractive summaries yielding higher ROUGE scores compared to abstractive and hybrid ones.

In [16], the authors proposed the WikiHow corpus for the English ATS task. The WikiHow corpus comprises 2,04,004 articles written by ordinary people to capture the writing style in a broader aspect. Unlike traditional news articles, the WikiHow entries are procedural and follow a highly structured, instructional layout. The corpus is constructed by extracting the articles from the WikiHow data dump[2]. This corpus offers a distinct domain for abstractive summarization, as it focuses more on instructional coherence than on factual news reporting. The best results were obtained using a pointer generator with a coverage model on WikiHow corpus (ROUGE-1 = 28.5, ROUGE-2 = 09.2, and ROUGE-L = 26.5). These results reveal that while summarizing instructional text, coverage mechanisms help mitigate redundancy and improve semantic consistency.

In [17], authors annotated the Gigaword[3] corpus for the ATS task. The Gigaword corpus originally consisted of 10 million news articles without corresponding headlines (summaries). Therefore, the authors in [17] used the subset of data and annotated it by extracting the first line of the article and treating it as a summary for the remaining article. The final version of the Gigaword corpus for ATS consists of 3.8 million news articles and summary pairs for the English language. Although the summaries are not manually written, this automatic approximation provides a large-scale resource for sentence-level abstractive summarization, particularly suitable for training models that rely on parallel sentence-summary pairs in a supervised manner.

[2] https://www.wikihow.com/Main-Page Last Visited: 25-Sep-2024

[3] https://catalog.ldc.upenn.edu/LDC2003T05 Last Visited: 25-Sep-2024

Researchers have also explored languages other than English for the ATS task, thereby contributing to the development of multilingual resources and evaluation techniques. For example, authors in [18] proposed a Persian news corpus for the ATS task. This corpus consists of 93,207 articles and their corresponding summary pairs. The dataset includes formal news texts sourced from various Persian-language news agencies, representing a comprehensive cross-section of socio-political, cultural, and economic topics. The summaries were either professionally written or curated by domain experts to maintain linguistic fluency and semantic fidelity.

The authors applied various transfer learning models to the proposed Persian news corpus to examine the adaptability of pre-trained language models in a low-resource setting. Among the approaches experimented with, the best results were obtained using transfer learning with the BERT2BERT technique on the Persian news corpus, yielding ROUGE-1 = 44.0, ROUGE-2 = 25.0, and ROUGE-L = 37.7. These findings underscore the potential of leveraging multilingual or cross-lingual pre-trained transformer architectures for abstractive summarization in underrepresented languages. Moreover, the study demonstrated that with appropriate fine-tuning, even general-purpose models can yield performance competitive with language-specific solutions in morphologically rich and syntactically diverse languages like Persian.

For the Chinese language ATS challenge, authors in [19] suggested a Large-scale Chinese Short Text Summarization (LCSTS) corpus. More than 2 million Chinese articles and related summaries make up the LCSTS corpus. The dataset was constructed by crawling articles from a popular Chinese microblogging website, wherein each post is accompanied by a user-generated summary. Due to the nature of microblog content, the corpus consists predominantly of short, informal text, which presents unique linguistic challenges for summarization models, such as informal syntax, domain variance, and idiomatic expressions.

To process the LCSTS corpus effectively, the authors employed a character-based input representation strategy, which is commonly used in Chinese NLP tasks to overcome the limitations of word segmentation in languages without whitespace delimiters. The authors experimented with various neural summarization architectures and reported that the RNN with context model achieved the best performance on the LCSTS corpus, yielding ROUGE-1 = 29.9, ROUGE-2 = 17.4, and ROUGE-L = 27.2. These results reflect the importance of context-aware modeling in capturing semantic dependencies within short and condensed text, especially in languages with complex character-based scripts like Chinese.

Furthermore, the LCSTS corpus has become a widely adopted benchmark for Chinese ATS research, serving as a foundational dataset for subsequent studies involving attention-based models, pre-trained language models, and multilingual architectures. Its scale and real-world relevance continue to support the evaluation of summarization models in scenarios that reflect practical social media communication.

In [24], authors have proposed an Urdu summary corpus for the Urdu ATS task. The corpus comprises 50 articles and their corresponding human-written abstractive summaries. The corpus is constructed by crawling articles from several online sources, such as news portals and blogs. The summaries are then manually written by domain experts to ensure semantic coverage, fluency, and grammatical consistency. Given the scarcity of high-quality linguistic resources for low-resource languages like Urdu, this corpus represents a significant contribution towards initiating research in the domain of Urdu abstractive summarization.

In [23], the authors extend this Urdu summary corpus for the ETS task. The corpus is constructed by selecting the most relevant sentences from the source document by the domain experts. This extractive version of the corpus comprises 600 news articles and their corresponding summaries. The construction process involved careful sentence selection to preserve both informativeness and coherence, ensuring the extractive summaries maintained semantic alignment with the original texts.

The best results were obtained using the weighted term technique with ROUGE-1 = 37.0, and ROUGE-2 = 65.0. These results indicate the effectiveness of statistical relevance-based extraction methods, especially in contexts where advanced neural models may not be feasible due to limited data. Furthermore, the development of both abstractive and extractive resources for Urdu lays foundational groundwork for exploring hybrid summarization models and domain-specific adaptations in low-resource linguistic settings.

2.2.2 Corpora for Other Languages

Researchers have also explored languages other than English for the ATS task, thus contributing to the development of multilingual corpora and the evaluation of summarization methodologies in linguistically diverse settings. These efforts are especially significant in broadening the applicability of abstractive summarization systems to non-English languages, many of which are underrepresented in the natural language processing (NLP) research landscape.

For example, authors in [18] proposed a Persian news corpus for the ATS task. This corpus consists of 93,207 articles and their corresponding summary pairs. The data was collected from multiple Persian-language news agencies, encompassing a wide range of topics including politics, economics, culture, and international affairs. The summaries associated with each article were written by professional editors or trained annotators, ensuring grammatical quality and semantic fidelity. The linguistic characteristics of Persian, including its morphological richness, syntactic flexibility, and right-to-left script, pose unique challenges for natural language understanding and generation tasks.

To address these challenges, the authors applied various transfer learning models to the proposed Persian news corpus. In particular, pre-trained transformer-based models were fine-tuned for the summarization task. The most effective configuration was found using the BERT2BERT technique, which employs a BERT encoder and BERT decoder architecture for end-to-end abstractive summarization. This model achieved superior performance metrics, with ROUGE-1 = 44.0, ROUGE-2 = 25.0, and ROUGE-L = 37.7. These results demonstrate the effectiveness of leveraging large-scale pre-trained models and fine-tuning them on domain-specific corpora, even in morphologically complex and low-resource languages. The study also highlights the role of transformer-based architectures in capturing long-distance dependencies and semantic nuances in Persian texts, which were traditionally difficult to model using recurrent neural networks.

In [19], authors proposed the Large-scale Chinese Short Text Summarization (LCSTS) corpus for the Chinese ATS task. The LCSTS corpus consists of more than 2 million Chinese articles and their corresponding summaries. The corpus was constructed by crawling a popular Chinese microblogging platform, where each post was accompanied by a user-generated summary. This methodology facilitated the automatic collection of a large volume of article-summary pairs, thus enabling the development of data-driven summarization models in Chinese. The corpus is characterized by its informal language, brevity of texts, and high lexical diversity, which present specific challenges for modeling coherence and informativeness in generated summaries.

Given the unique linguistic properties of Chinese, such as the lack of explicit word boundaries and the complexity of character-level semantics, the authors employed a character-based input representation in their modeling approach. A recurrent neural network (RNN) with contextual modeling was applied to the LCSTS corpus, which enabled the model to capture semantic information without relying on explicit word segmentation. The best results were achieved using the RNN with context model, yielding ROUGE-1 = 29.9, ROUGE-2 = 17.4, and ROUGE-L = 27.2. These findings underscore the viability of character-level modeling in Chinese and emphasize the importance of context-aware architectures in abstractive summarization tasks. The LCSTS corpus has since served as a standard benchmark for evaluating Chinese summarization systems and has facilitated the development of more advanced models including attention-based and pre-trained language models tailored for Chinese text.

The exploration of non-English corpora such as the Persian news dataset and LCSTS underscores the growing emphasis on multilingual and cross-lingual summarization research. These corpora not only offer benchmarks for language-specific summarization models but also contribute towards the advancement of universal summarization architectures capable of handling linguistic variation, script differences, and domain-specific constraints. Consequently, such resources are vital for democratizing access to summarization technologies across global linguistic communities.

2.2.3 Corpora for Urdu Language

In addition to widely spoken languages such as Persian and Chinese, researchers have also focused on developing abstractive and extractive summarization corpora for low-resource languages. These efforts are essential to ensure equitable progress in natural language processing between linguistic communities with limited computational resources and annotated datasets. Among these languages, Urdu, spoken by millions in South Asia, has recently gained attention in summarization research. Despite its complex script, rich morphology, and relatively scarce digital resources, multiple initiatives have been undertaken to construct annotated corpora and evaluate summarization models tailored for Urdu. The following section outlines significant contributions in this domain, including manually curated datasets and domain-specific corpora developed for both ATS and ETS tasks in the Urdu language.

In [24], the authors proposed an Urdu summary corpus for the Urdu ATS task. The corpus comprises 50 articles and their corresponding human-written abstractive summaries. The data was sourced by crawling articles from several online platforms, including news portals and blogs, to capture a representative sample of modern Urdu prose in diverse topical domains. The articles cover a broad spectrum of subjects such as politics, education, entertainment, and current affairs, reflecting the linguistic richness and structural diversity of Urdu. After collection, the summaries were manually written by domain experts, ensuring semantic coverage, grammatical integrity, and coherence with the original content. Given the limited availability of annotated resources for low-resource languages like Urdu, this corpus constitutes a critical first step towards enabling data-driven abstractive summarization research in Urdu.

In [23], the authors extended this Urdu summary corpus to support the ETS task. The extractive version of the corpus was created by selecting the most relevant sentences from the source documents, a process performed manually by linguistic experts to ensure that the resulting summaries retained the core semantic essence of the original text. The extended corpus comprises 600 Urdu news articles and their corresponding extractive summaries. This dataset allowed researchers to evaluate traditional and statistical extraction techniques in the Urdu context. Among the techniques explored, the best results were obtained using the weighted term approach, yielding ROUGE-1 = 37.0 and ROUGE-2 = 65.0. These results highlight the efficacy of term-weighting methods in selecting information-rich sentences for extractive summarization in morphologically complex languages like Urdu, where syntactic and semantic cues are often embedded in compound structures and inflectional morphology.

In [25], the authors proposed the CLE Meeting Corpus for extractive text summarization in Urdu. This corpus was generated by transcribing a total of 7900 minutes of recorded meeting audio into Urdu text and pairing it with corresponding human-written summaries. The corpus addresses the under-explored domain of meeting summarization in Urdu, providing a valuable resource for speech-to-text summarization pipelines. To evaluate the effectiveness of neural summarization models on this corpus, the authors employed a transfer learning approach using the ur_mT5 model, which is a multilingual variant of the T5 architecture pre-trained on Urdu text. After fine-tuning on the CLE Meeting Corpus, the model achieved ROUGE-1 = 31.7, ROUGE-2 = 11.1, ROUGE-L = 17.2, and BLEU_Score of 4.15. These findings underscore the adaptability of transformer-based models in low-resource domains when supplemented with domain-specific training data and highlight the complexities of processing oral discourse data in textual summarization tasks.

A similar corpus was developed by [26] by converting video transcripts into text and generating corresponding summaries for the Urdu Abstractive Text Summarization (UATS) task. The resulting corpus, termed UrduMASD, was constructed by generating Urdu transcripts of 15,374 videos spanning various domains, including news, education, and public discourse. These transcripts were then paired with abstractive summaries created manually, offering a large-scale resource for training and evaluating ATS systems. The authors applied a transfer learning method based on pre-trained language models fine-tuned on the UrduMASD corpus. The best results achieved were ROUGE-1 = 26.8, ROUGE-2 = 8.0, and ROUGE-L = 19.6. These relatively modest results reflect the inherent challenges of summarizing noisy and unstructured spoken content, especially in low-resource settings, and point towards the need for further research in noise-robust and context-aware summarization models for multimodal and conversational Urdu data.

Collectively, these efforts mark a significant advancement in the creation of Urdu summarization resources, covering both written and spoken text across abstractive and extractive paradigms. They provide the necessary groundwork for future exploration into domain adaptation, multilingual transfer learning, and real-world applications of summarization technologies in the Urdu language context.

Table 2.1: Summary of Corpora for Abstractive Text Summarization

Ref.	**Language**	**Corpus Name**	**Corpus Size**	**Average Article Length**	**Average Summary Length**
[11, 29]	English	DUC (2001-2007)	1600	-	-
[17]	English	Gigaword	3.8 Million	31.4 (Words)	8.3 (Words)
[19]	Chinese	LCSTS	1.5 Million	80 (Characters)	20 (Characters)
[14, 30]	English	CNN/Daily Mail	2.4 Million	766 (Words)	53 (Words)
[15]	English	NewsRoom	311,672	658.8 (Words)	26.7 (Words)
[16]	English	WikiHow	1.3 Million	579.8 (Words)	62.1 (Words)
[24, 23]	Urdu	Urdu Summary Corpus	50	159 - 2518 (Words)	79 - 761 (Words)
[25]	Urdu	Urdu Meeting Corpus	7900 min	-	-
[26]	Urdu	UrduMASD	15,374 Videos	-	-

2.3 Methods for Abstractive Text Summarization

In literature, researchers have also developed a wide range of techniques and methods by leveraging the above-mentioned corpora for the ATS task. These corpora have served as foundational resources for designing, training, and evaluating summarization models across multiple languages and domains. The availability of both large-scale and domain-specific datasets has enabled the exploration of diverse modeling paradigms, ranging from traditional statistical methods and sequence learning architectures to more recent advancements in transformer-based and reinforcement learning frameworks. The following section provides a comprehensive overview of these methodologies, highlighting how various models have been adapted or fine-tuned in conjunction with the specific characteristics of the corresponding datasets.

2.3.1 Deep Learning Methods for Abstractive Text Summarization

In literature, researchers have also developed techniques and methods by using the above-mentioned corpora for the ATS task. The increasing availability of annotated summarization corpora has significantly contributed to the development and evaluation of deep learning-based models for abstractive summarization. These methods aim to learn the underlying structure, semantics, and abstraction patterns from large-scale datasets through end-to-end training mechanisms.

In [3], the authors proposed a convolution encoder model integrated with an attention mechanism to address sentence-level summarization tasks. Unlike traditional sequence-to-sequence models based on recurrent architectures, the proposed model utilizes convolutional layers for encoding source sentences, which allows for better parallelization during training and efficient handling of local context through convolutional filters. The attention mechanism is then employed to selectively focus on relevant parts of the input sentence, thereby enhancing the decoder's ability to generate semantically coherent and contextually relevant summaries.

The authors evaluated the proposed technique on DUC-2004 [11] and Gigaword [17] corpora. Both datasets are widely recognized benchmarks in ATS research, offering distinct challenges due to differences in document length, summary abstraction level, and domain. The best results were obtained using the attention-based convolution encoder model on the Gigaword corpus, achieving ROUGE-1 = 31.0, ROUGE-2 = 12.6, and ROUGE-L = 28.3. These results demonstrated the potential of convolution-based models in learning meaningful abstract representations of source texts and producing fluent summaries, particularly in settings where short summaries are required for individual sentences or short paragraphs.

The study also highlighted the role of attention mechanisms in improving the relevance of generated content by dynamically weighting input tokens during decoding. Overall, this approach marked an early yet impactful step towards exploring convolutional architectures for abstractive summarization, paving the way for more advanced hybrid models combining convolutional, recurrent, and transformer-based elements

In [6], authors applied various convolutional, RNN, and LSTM-based encoder-decoder models on the Gigaword [17] corpus and evaluated the models on both the Gigaword and DUC-2004 [11] test corpora. The primary objective was to assess the effectiveness of recurrent and convolutional neural architectures in generating abstractive summaries from short news articles. The study explored multiple configurations, including convolutional encoders with RNN decoders, standard LSTM encoder-decoder models, and their attention-augmented counterparts. Among these, the recurrent attention model demonstrated superior performance, outperforming the attention-equipped LSTM models in terms of ROUGE metrics. The attention mechanism in the recurrent model enabled dynamic focus on the most relevant parts of the input sequence during decoding, which contributed to more semantically aligned summaries.

The authors reported results on both evaluation corpora, highlighting the generalizability of their models across datasets. The best performance was observed on the Gigaword corpus, with the recurrent attention model achieving ROUGE-1 = 33.7, ROUGE-2 = 15.9, and ROUGE-L = 31.1. These results indicated that the recurrent attention framework was particularly effective in capturing sequential dependencies and preserving the semantic essence of the source content, especially for sentence-level summarization tasks involving short and concise text inputs.

In [31], the authors addressed a key limitation in sequence-to-sequence model-snamely, the out-of-vocabulary (OOV) problemby incorporating a copy mechanism into an RNN-based summarization framework. The copy mechanism allows the model to directly replicate segments of the source text into the output sequence, thereby preserving rare or unseen words that the standard decoder vocabulary may not cover. This is particularly useful in summarization tasks involving languages with rich vocabulary or informal expressions, such as Chinese microblog text.

The proposed model was applied to the Large-scale Chinese Short Text Summarization (LCSTS) corpus [19], using both word-level and character-level representations. The copy mechanism was integrated into the decoder, enabling it to selectively generate words from its vocabulary or copy them from the source sequence. The authors reported that the best performance was achieved using the word-level model, with ROUGE-1 = 35.0, ROUGE-2 = 22.3, and ROUGE-L = 32.0. These results demonstrate the efficacy of copy-based architectures in improving summary quality, particularly by handling rare terms and named entities more effectively. Furthermore, the study reinforced the notion that hybrid approaches combining generation and copying strategies offer a promising direction for enhancing abstractive summarization in both high- and low-resource settings.

In [4], the authors incorporated the pointer-generator coverage model into the standard sequence-to-sequence summarization architecture. This approach was specifically designed to address a common problem in abstractive text summarizationnamely, the repetition of phrases in generated summaries due to uncontrolled attention behavior. The pointer-generator mechanism enables the model to flexibly switch between generating words from a fixed vocabulary and copying words directly from the source text, while the coverage model augments this by maintaining a cumulative attention history across decoding steps. This attention history serves as a coverage vector that informs the model of which parts of the source have already been attended to, thereby reducing redundancy and improving summary coherence.

The authors applied both attention-based baseline models and pointer-generator variants to the CNN/Daily Mail corpus [14], a widely-used benchmark for abstractive summarization tasks. Empirical evaluation demonstrated that the pointer-generator model equipped with the coverage mechanism outperformed the baseline models significantly. The best results were achieved using the pointer-generator with coverage model, yielding ROUGE-1 = 39.5, ROUGE-2 = 17.2, and ROUGE-L = 36.3. These findings underscore the importance of modeling coverage in attention-based architectures to ensure content diversity and relevance in the generated summaries, particularly for long-form documents that are prone to content overlap during decoding.

In [32], the authors introduced a novel content selection mechanism that was integrated into attention-based sequence models to enhance the informativeness of generated summaries. The motivation behind this approach was to constrain the decoder to generate summaries based only on the most salient portions of the source document, effectively separating the processes of content planning and surface realization. The proposed method, referred to as bottom-up summarization, operates by first identifying a soft mask over the input tokens based on content importance and then conditioning the pointer-generator model to attend predominantly to these masked regions during decoding.

To validate their approach, the authors applied multiple baseline models, including RNNs and pointer-generator architectures, both with and without attention mechanisms, on the CNN/Daily Mail [14] and New York Times (NYT) [12] corpora. The best results were obtained using the bottom-up summarization method, which achieved ROUGE-1 = 41.2, ROUGE-2 = 18.6, and ROUGE-L = 38.3. These results demonstrate the effectiveness of explicit content selection as a preprocessing step in abstractive summarization pipelines, allowing for more focused and semantically rich summary generation. Moreover, the model's generalizability across multiple corpora emphasizes the robustness of this approach in diverse domains and summary styles.

In [33], the authors proposed a global encoding model designed to eliminate redundant information in the generated abstractive summaries by strengthening the semantic representation of the input sequence. The central idea of the proposed approach is to augment the encoder side of the encoder-decoder architecture with a global encoding layer that captures the overall context of the source document. This mechanism enables the model to retain semantically important content while suppressing repetitive or irrelevant tokens before the decoding stage begins. Unlike local attention-based encoders that may over-focus on certain parts of the input during each decoding step, the global encoding layer acts as a content filter, ensuring that the input representation fed to the decoder is both globally coherent and semantically compact.

The authors applied the proposed global encoding model along with other baseline techniques to the LCSTS [19] and Gigaword [17] corpora. These two corpora, though different in linguistic structure and summary style, provided a suitable testing ground for evaluating the models generalizability across sentence-level and short-document summarization tasks. Experimental results demonstrated that the proposed global encoding model outperformed baseline attention-based models, particularly on the LCSTS corpus. The best results reported were ROUGE-1 = 39.4, ROUGE-2 = 26.9, and ROUGE-L = 36.5. These outcomes validate the effectiveness of global context modeling in enhancing content selection and summary generation, especially in character-based languages where semantic interpretation relies heavily on long-range dependencies.

In [34], the authors introduced a novel modification to the attention mechanism by incorporating decoder input words directly into the calculation of attention vectors. This enhancement marks a significant departure from traditional attention frameworks, which typically compute attention solely based on the decoders hidden states and the encoders output. By integrating the decoder input words into the attention computation, the model is able to better align the generated output with both the source context and the evolving target-side information, thereby producing summaries that are not only contextually coherent but also lexically diverse.

Moreover, the proposed attention mechanism includes semantic and contextual similarity metrics, allowing the model to assess the importance of source tokens not just syntactically but also semantically. This dual-faceted evaluation enhances the attention distribution and improves the semantic fidelity of generated summaries. The authors tested their model alongside baseline RNN and attention-based models on the CNN/Daily Mail [14] and Gigaword [17] corpora. The best performance was observed on the Gigaword corpus, where the model achieved ROUGE-1 = 38.2, ROUGE-2 = 16.4, and ROUGE-L = 36.0. These results illustrate the potential of enriching attention mechanisms with decoder-side and semantic cues, enabling better generation of abstractive content with fewer factual inconsistencies and improved relevance.

In [35], the authors addressed one of the critical challenges in abstractive summarizationensuring the factual consistency of the generated summaries. Abstractive summarization models, especially those based on large pre-trained language models, often tend to generate fluent yet factually inconsistent content, a phenomenon commonly referred to as "hallucination." To mitigate this issue, the authors proposed a novel training data filtering approach that selectively retains only those samples which demonstrate a high degree of factual alignment between the source document and its reference summary. The aim was to improve the quality of supervision signals during training, thereby enhancing the factual reliability of the summaries produced by the model.

The proposed method was implemented using state-of-the-art transfer learning models, including BERT and PEGASUS, which were fine-tuned on filtered datasets. The authors evaluated their approach across multiple benchmark corpora, including Newsroom [15], Xsum, and CNN/Daily Mail [14]. Among these, the best performance was achieved using a pre-trained BERT model fine-tuned on the Xsum dataset, with ROUGE-1 = 45.6, ROUGE-2 = 22.5, and ROUGE-L = 37.2. These results indicate that strategic preprocessing of training data can significantly influence the semantic and factual alignment of generated summaries, and that transfer learning models, when coupled with quality-assured datasets, can substantially reduce hallucination in abstractive summarization tasks.

In [10], the authors proposed a hybrid approach that decomposes the summarization process into two sequential stages to enhance the quality and coherence of generated summaries. In the first stage, the model identifies and extracts the most pertinent sentences from the source document. This content selection step acts as an information distillation phase that simplifies the generation process. In the second stage, language models are employed to rewrite the selected content into a coherent abstractive summary. This pipeline was further optimized using reinforcement learning techniques, allowing the model to directly maximize reward functions based on summary quality, such as ROUGE scores.

To evaluate their framework, the authors applied a range of baseline models as well as their proposed reinforcement learning with language modeling approach on the CNN/Daily Mail corpus [14]. The integration of reinforcement learning enabled the model to fine-tune its generation policy based on feedback from evaluation metrics, thus improving the overall informativeness and fluency of the output. The best results were obtained using this hybrid method, yielding ROUGE-1 = 40.1, ROUGE-2 = 17.3, and ROUGE-L = 37.5. These findings highlight the benefits of multi-stage architectures and policy-driven optimization in abstractive summarization, especially for complex documents where content planning and linguistic generation need to be effectively decoupled.

In [36], the authors applied sequence-to-sequence models integrated with an attention mechanism for sentence-level abstraction in the ATS task. Unlike traditional document-level summarization approaches, this method focuses on generating concise and coherent summaries at the sentence level, thereby facilitating fine-grained control over the abstraction process. The attention mechanism was employed to dynamically align input and output tokens during decoding, allowing the model to retain key semantic elements from the original sentence. The proposed technique was evaluated on the Gigaword [17] and Google [37] corpora. While the Gigaword corpus offered a traditional benchmark, the Google corpus provided a rich testbed for evaluating compression-focused summarization.

A distinctive contribution of this work lies in the adoption of an alternative evaluation methodology that moves beyond conventional ROUGE metrics. The authors introduced and employed novel evaluation criteria, including F_1, RASP-F_1, and Compression Ratio, which better capture semantic preservation and information density in abstractive outputs. The best performance was achieved on the Google corpus, where the proposed model attained an F_1 score of 85.1, a RASP-F_1 score of 82.3, and a Compression Ratio of 0.4. These results underscore the importance of exploring diverse evaluation metrics tailored to the summarization context and indicate that attention-augmented sequence models can achieve high semantic fidelity and compression efficiency in sentence-level abstraction tasks.

In [38], the authors proposed a novel framework for abstractive summarization by leveraging recent advancements in Abstract Meaning Representation (AMR), which captures the underlying semantics of sentences in the form of structured graphs. In this architecture, the source article is first parsed into a set of AMR graphs, each representing the semantic content of sentences or clauses. These individual graphs are then transformed and merged into a single summary graph that abstracts the essential information from the source. Finally, natural language text is generated from the summary graph, thereby completing the abstractive summarization pipeline. This approach introduces a clear separation between semantic abstraction and linguistic realization, making the summarization process more interpretable and semantically grounded.

The authors evaluated their proposed methodology using the JAMR parsing and generation framework on DUC and TAC [11] corpora, enhanced with custom annotations to support AMR conversion. The results demonstrated that AMR-based summarization offers a promising direction for enhancing both the informativeness and structural clarity of generated summaries. The best performance was obtained using the JAMR-based approach, achieving ROUGE-1 *precision* = 51.2, *recall* = 40.0, and F_1 = 44.7. These findings illustrate the effectiveness of semantic graph representations in capturing abstract concepts and relations, thus providing a linguistically informed foundation for abstractive summary generation.

In [39], the authors introduced the Span-Fact Suite genuine rectification model, which is designed to enhance the factual correctness of abstractive summaries generated by neural models. This approach leverages knowledge derived from question-answering models to perform post-generation factual validation and correction. Specifically, the model identifies potentially incorrect or inconsistent spans within the generated summary and replaces them with factually accurate content derived from the source document. The methodology combines span detection with fact-aware rectification, thereby offering a mechanism to reduce hallucinated contenta persistent challenge in abstractive summarization systems.

To evaluate their approach, the authors employed several transfer learning models in conjunction with the Span-Fact Suite correction module on widely-used corpora, including CNN/Daily Mail [14], XSum, and Gigaword [17]. Among these datasets, the best results were obtained on the CNN/Daily Mail corpus using the integrated span-fact rectification method along with transfer learning. The model achieved ROUGE-1 = 41.8, ROUGE-2 = 19.4, and ROUGE-L = 38.9. These results reflect the significant impact of incorporating post-processing factual validation stages in improving summary reliability, especially when used alongside powerful pre-trained models that may lack intrinsic factual grounding.

In [40], the authors proposed an abstractive text summarization model tailored specifically for the Arabic language. The model is based on the sequence-to-sequence RNN architecture, where the encoder comprises a bi-directional Long Short-Term Memory (Bi-LSTM) network and the decoder is a uni-directional LSTM enhanced with global attention. This architecture is well-suited to handle the linguistic characteristics of Arabic, which include complex morphology, right-to-left writing direction, and variable sentence structure. The dataset used consisted of Arabic summaries collected from multiple online sources, covering diverse topics and writing styles.

In addition to implementing the neural summarization model, the authors introduced a novel evaluation framework tailored to the Arabic language. Alongside the traditional ROUGE-1 metric, the evaluation included ROUGE1-NOORDER, which ignores word order; ROUGE1-STEM, which considers word stems rather than surface forms; and ROUGE1-CONTEXT, which measures contextual relevance. These custom metrics aim to provide a more accurate reflection of summary quality in morphologically rich and flexible languages like Arabic. The best results were obtained using the proposed LSTM-based sequence-to-sequence model, with ROUGE-1 = 38.4, ROUGE1-NOORDER = 46.2, ROUGE1-STEM = 52.6, and ROUGE1-CONTEXT = 58.1. These findings demonstrate the potential of tailored neural architectures and evaluation metrics in improving the performance and assessment of ATS systems for under-resourced and linguistically complex languages.

2.3.2 Transformer Methods for Abstractive Text Summarization

Transformer-based models have become a dominant paradigm in abstractive text summarization due to their superior capacity for modeling long-range dependencies and capturing contextual information across sequences. The self-attention mechanism inherent in transformer architectures allows for efficient parallelization and dynamic weighting of input tokens, making them well-suited for summarization tasks involving long-form documents.

In [41], the authors proposed a transfer learning-based architecture to enhance the quality of abstractive summaries for long articles. The proposed model extends the Transformer-XL architecture by incorporating entity-level semantics and integrating primary word information extracted from the Wikidata knowledge graph. This hybridization is achieved through an entity-aware embedding layer, which aligns structured information from external sources with token-level representations in the input sequence. By infusing this external semantic knowledge, the model becomes more capable of resolving ambiguities, maintaining coherence, and improving factual alignment in the generated summaries.

The authors applied the proposed transformer-XL with entity-level transfer learning model on the CNN/Daily Mail [14] corpus, a benchmark dataset for document-level summarization tasks. Experimental evaluation demonstrated that the inclusion of entity and knowledge graph embeddings significantly enhanced the summarization quality compared to baseline transformer models. The best results were achieved using the transformer-XL-entity-Wikidata word embedding model, which yielded ROUGE-1 = 33.8, ROUGE-2 = 12.5, and ROUGE-L = 31.2. These findings underscore the effectiveness of enriching transformer architectures with structured semantic information for generating more informative and contextually accurate summaries, particularly in scenarios involving extensive input documents with numerous entities and relational data.

In line with recent research emphasizing semantic evaluation, the authors replaced ROUGE metrics with BERTScore to better capture the meaning-preserving properties of generated summaries. The evaluation revealed that the fine-tuned $text\text{-}DaVinci$-002 model achieved the highest performance on the XSum corpus, with a BERTScore of 0.90. These findings illustrate the capacity of general-purpose language models like GPT-3.5 to excel in specialized downstream tasks such as ATS, especially when paired with contextually rich training corpora and semantic-aware evaluation metrics. Furthermore, this work contributes to the growing body of research that positions large language models (LLMs) not merely as generalists but as effective tools for domain-specific abstractive summarization when properly fine-tuned and evaluated.

In [42], the authors introduced the Russian news corpus known as the Gazeta corpus and employed it as a benchmark for evaluating transformer-based abstractive summarization techniques in the Russian language. To assess the effectiveness of multilingual transformer models, the authors utilized mBART, a multilingual sequence-to-sequence model pre-trained on monolingual corpora across 25 languages, including Russian. The model was fine-tuned on the Gazeta corpus to adapt it to the domain-specific and linguistic characteristics of Russian news articles. This fine-tuning enabled the model to better capture semantic and syntactic patterns relevant to the Russian language while leveraging cross-lingual transfer capabilities inherent in mBARTs architecture.

The evaluation was performed using standard ROUGE metrics, and the best results were obtained using the fine-tuned mBART model on the Gazeta corpus, with ROUGE-1 = 32.1, ROUGE-2 = 14.2, and ROUGE-L = 27.9. These findings affirm the efficacy of multilingual pre-trained transformers in low-resource or morphologically rich languages when appropriately fine-tuned. Furthermore, the study demonstrated the viability of mBART for multilingual ATS tasks, particularly in settings where parallel data is scarce but high-quality monolingual corpora are available.

In [43], the authors proposed Hierarchical BART (Hie-BART), an enhanced version of the BART transformer model that explicitly incorporates the hierarchical structure of documents to improve the quality of abstractive summaries. Traditional transformer models often treat input sequences as flat token streams, which may lead to the loss of important document-level organizational cues. In contrast, Hie-BART is designed to model interactions at both sentence and word levels, thereby capturing the discourse and topical structure of lengthy documents more effectively.

The hierarchical architecture of Hie-BART enables the model to attend not only to token-level dependencies but also to inter-sentence relationships, which is particularly beneficial for multi-sentence document summarization. The model was evaluated on the CNN/Daily Mail [14] corpus, a standard benchmark for document-level ATS. The best results achieved using Hie-BART were ROUGE-1 = 44.3, ROUGE-2 = 21.3, and ROUGE-L = 41.0. These results demonstrate that incorporating document-level structural awareness into transformer architectures can significantly enhance summary coherence, informativeness, and semantic alignment. The study further supports the integration of discourse-level modeling in neural summarization frameworks, particularly for longer documents with complex internal structure.

In [44], the authors proposed BART-IT, a transformer-based model derived from the original BART architecture, specifically tailored for the Italian language. Recognizing the scarcity of high-quality transformer models optimized for Italian, the authors pre-trained the BART-IT model on a large-scale monolingual Italian dataset known as the Clean Italian mC4 Corpus. This pre-training phase allowed the model to learn comprehensive linguistic representations, including grammar, syntax, and contextual usage patterns specific to Italian.

Following pre-training, the BART-IT model was fine-tuned on three distinct Italian corporaFanPage, IlPost, and WITSeach representing a different domain and writing style. This multi-domain fine-tuning strategy was aimed at evaluating the models adaptability and robustness across various sub-genres of Italian news and editorial content. The model was subsequently evaluated for the ATS task using ROUGE metrics across all three corpora.

Among the evaluated datasets, the best summarization performance was observed on the WITS corpus, where BART-IT achieved ROUGE-1 = 42.3, ROUGE-2 = 28.8, and ROUGE-L = 38.8. These results indicate that language-specific pre-training, followed by domain-sensitive fine-tuning, can substantially enhance the performance of transformer models for abstractive summarization in non-English settings. Moreover, this work underscores the importance of building tailored resources and architectures to support the linguistic and contextual needs of underrepresented languages in the ATS landscape.

2.3.3 LLM Methods for Abstractive Text Summarization

In [45], the authors explored the applicability of large-scale generative language models for the ATS task in low-resource and morphologically rich languages by fine-tuning GPT-3.5 on the Russian Gazeta corpus. The Gazeta corpus comprises a substantial collection of Russian news articles and their human-written summaries, representing a diverse linguistic and topical distribution. To adapt the generative model to the linguistic intricacies of Russian, the authors fine-tuned the ruGPT3Small varianta Russian-specific adaptation of GPT-3.5on this corpus.

The proposed approach diverged from traditional summarization pipelines by adopting an autoregressive transformer model with generative capabilities and evaluating its performance using semantic-based metrics. Specifically, the authors moved away from the conventional ROUGE metrics, which primarily rely on lexical overlap, and instead utilized BERTScore to assess the semantic similarity between generated and reference summaries. The best results were obtained using the fine-tuned ruGPT3Small model on the Gazeta corpus, achieving a BERTScore precision of 0.87, recall of 0.90, and F_1 = 0.89. These results reflect the models capacity to generate semantically accurate and fluent summaries in a low-resource language, while also emphasizing the growing need for evaluation methods that better align with human judgments of meaning and coherence.

In [46], the authors evaluated the performance of GPT-3.5 against a variety of fine-tuned summarization models for the English ATS task. The study focused on assessing whether large-scale generative models, particularly GPT-3.5's *text-DaVinci*-002 variant, could match or exceed the performance of specialized, task-specific architectures when fine-tuned on summarization datasets. The model was fine-tuned on two standard corporaCNN/Daily Mail [14] and XSumwhich differ in document length, summary abstraction level, and stylistic features.

To summarize, the majority of research on the ATS task has been predominantly conducted in English, leveraging a wide range of corpora such as CNN/Daily Mail [14], Gigaword [17], and XSum. However, recent advancements have extended the scope of ATS research to encompass several other languages, including Chinese [19], Persian [18], Arabic [47], and Russian [42]. These multilingual efforts not only reflect the growing interest in developing inclusive and language-agnostic summarization models but also underscore the importance of linguistic diversity and culturally contextualized datasets in shaping the future of abstractive summarization research.

Recently, attention-based techniques, particularly those grounded in transformer architectures, have also begun to emerge within the domain of UATS. This development reflects a significant evolution from earlier neural architectures and highlights the increasing adoption of transformer-based approaches in low-resource language settings. In [27], the authors introduced an LSTM-based model integrated with an attention mechanism, specifically designed for the UATS task. The model was trained on an Urdu news corpus and achieved encouraging results, with ROUGE-1 = 43.0, ROUGE-2 = 25.0, and ROUGE-L = 23.0. Similarly, another study [28] proposed a comparable LSTM with an attention framework, also tailored to the linguistic structure of Urdu, and reported improved results on their own Urdu news dataset, achieving ROUGE-1 = 44.0, ROUGE-2 = 26.0, and ROUGE-L = 24.0.

Despite these advancements, existing studies on UATS exhibit two key limitations. Firstly, the available corpora are notably limited in both size and quality. The majority of existing datasets range from as few as 50 article-summary pairs [24] to approximately 10,000 pairs, which are insufficient to train robust and generalizable summarization models. Furthermore, these datasets lack the standardization and scale required to serve as benchmark corpora for comprehensive evaluation and comparative analysis in UATS research. Secondly, most recent and advanced techniques, including transformer architectures and LLMs, have not been thoroughly evaluated on these Urdu datasets. As a result, the empirical potential of state-of-the-art summarization models remains underexplored in the context of Urdu.

To overcome these limitations, this research introduces a comprehensive, large-scale benchmark corpus consisting of over 2.067 million Urdu news articles and their corresponding abstractive summaries. The corpus is constructed to ensure topical diversity, linguistic richness, and summary coherence, thereby providing a reliable foundation for training and evaluating modern ATS systems. In addition to corpus construction, this study presents a systematic evaluation of nine different models, covering traditional deep learning architectures, state-of-the-art transformer models, and advanced LLMs. To the best of our knowledge, this is the first comprehensive effort in UATS literature to provide both a large-scale benchmark corpus and an extensive empirical comparison of contemporary summarization techniques. This contribution aims to facilitate future research in UATS by enabling reproducibility, standardization, and the development of more effective summarization models for the Urdu language.

Table 2.2: Summary of Techniques/Methods for Abstractive Text Summarization

Year	**Ref.**	**Corpus**	**Corpus Size**	**Techniques/ Methods**	**Results**
2015	[11, 17]	DUC-2004, Gigaword	600-1250, 3.8M	Convolution Encoder with Attention	ROUGE-1=31.00, ROUGE-2=12.60, ROUGE-L=28.30
2016	[19]	LCSTS	1.5 Milion	Seq2Seq RNN with Copy Mechanism	ROUGE-1=35.00, ROUGE-2=22.30, ROUGE-L=32.00

Continued on next page

Table 2.2: (Continued)

Year	Ref.	Corpus	Corpus Size	Techniques/ Methods	Results
2016	[11, 17]	DUC-2004, Gigaword	600-1250, 3.8M	Convolution RNN, LSTM Encoder-Decoder	ROUGE-1=33.70, ROUGE-2=15.90, ROUGE-L=31.10
2017	[14]	CNN/Daily Mail	2.4 Million	Pointer-generator with Coverage	ROUGE-1=39.50, ROUGE-2=17.20, ROUGE-L=36.30
2018	[12, 14]	CNN/Daily Mail, NYT	2.4M, 654,759	Bottom-Up Summarization Approach	ROUGE-1=41.20, ROUGE-2=18.60, ROUGE-L=38.30
2018	[17], [19]	LCSTS, Gigaword	1.5M, 3.8M	Global Encoding Model	ROUGE-1=39.40, ROUGE-2=26.90, ROUGE-L=36.50
2018	[17, 37]	Gigaword, Google Corpus	3.8M	Seq2Seq with Attention	F_1=85.10, RASP-F_1=82.30
2019	[14], [17]	CNN/Daily Mail, Gigaword	2.4M, 3.8M	Attention considering Decoder Input	ROUGE-1=38.20, ROUGE-2=16.40, ROUGE-L=36.00
2020	[42]	Gazeta Corpus (Russian)	-	mBART Transformer	ROUGE-1=32.10, ROUGE-2=14.20, ROUGE-L=27.90

Continued on next page

Table 2.2: (Continued)

Year	**Ref.**	**Corpus**	**Corpus Size**	**Techniques/ Methods**	**Results**
2021	[14, 15]	Newsroom, XSum, CNN/Daily Mail	1.3M, 227k, 2.4M	Transfer Learning with BERT	ROUGE-1=45.60, ROUGE-2=22.50, ROUGE-L=37.20
2022	[47]	Arabic Corpus	-	BiLSTM with Global Attention	ROUGE-1=38.40, ROUGE-NOORDER=46.20, ROUGE-STEM=52.60
2022	[45]	Gazeta Corpus	-	GPT-3.5	Precision=0.87, Recall=0.90, F_1=0.89
2022	[46]	CNN/Daily Mail, XSum	2.4M, 227k	GPT-3.5 (text-DaVinci-002)	BERTscore=0.90
2022	[43]	CNN/Daily Mail	2.4 M	Hie-BART	ROUGE-1=44.30, ROUGE-2=21.30, ROUGE-L=41.00
2022	[44]	FanPage, IlPost, WITS (Italian)	-	BART-IT	ROUGE-1=42.30, ROUGE-2=28.80, ROUGE-L=38.80
2022	[27]	Urdu News Corpus	-	LSTM with Attention	ROUGE-1=43.00, ROUGE-2=25.00, ROUGE-L=23.00

Continued on next page

Table 2.2: (Continued)

Year	Ref.	Corpus	Corpus Size	Techniques/ Methods	Results
2023	[28]	Urdu News Corpus	-	LSTM with Attention	ROUGE-1=44.00, ROUGE-2=26.00, ROUGE-L=24.00

2.4 Evaluation Measures

The problem of UATS was formulated as a supervised machine learning task. To enable the accurate training and evaluation of various state-of-the-art baseline deep learning models, the entire dataset was partitioned following the standard Train/Test/Validation split methodology. Specifically, 2,027,784 instances were allocated for training, 20,000 instances for validation, and 20,000 instances for testing. This data partitioning strategy was designed to ensure that the models could generalize effectively while also allowing for reliable hyperparameter tuning and unbiased performance evaluation.

The performance of the deep learning models was assessed using the averaged F_1 scores of the ROUGE-1, ROUGE-2, and ROUGE-L metrics[4], which are widely used for evaluating the quality of generated summaries by measuring their overlap with reference summaries at different levels of granularity (unigram, bigram, and longest common subsequence, respectively).

To summarize, the majority of efforts in the domain of ATS have primarily focused on the English language, with some additional contributions in Chinese [19], Persian [18], and Arabic [47]. Within the existing literature, the only study specifically addressing Urdu ATS is presented in [24], which suffers from two critical limitations. First, the corpus consists of merely 50 article-summary pairs, which is insufficient for training and evaluating contemporary neural models. Second, the study does not explore the application of advanced deep learning or transfer learning techniques, thereby limiting its practical utility and scope.

[4] https://github.com/pltrdy/rouge Last Visited: 25-Sep-2024

To address these limitations, this research proposes a large-scale benchmark corpus comprising approximately 2.067 million Urdu news articles, each accompanied by a corresponding human-written abstractive summary. In addition, we have developed, applied, and comparatively evaluated six baseline deep learning models on the proposed corpus. To the best of our knowledge, the construction of a corpus of this scale, along with a systematic empirical evaluation of baseline deep learning models, has not been previously reported for the Urdu ATS task. This contribution aims to establish a foundation for future work in the field and facilitate further advancements in multilingual and low-resource text summarization research.

2.5 Chapter Summary

Chapter 2 provided a comprehensive review of existing corpora, methodologies, and evaluation metrics for ATS, emphasizing the gap in Urdu language summarization research. The literature review highlighted the dominance of English-language ATS corpora, i.e., DUC, CNN/Daily Mail, and Gigaword, alongside notable ATS datasets for Chinese, Persian, and Arabic. The extensive resources exist for other languages, Urdu remains a low-resource language with only a few small-scale corpora available for text summarization. These findings directly motivate the research presented in Chapter 3, where we introduce the UATS-23 corpus, a large benchmark corpus for UATS. The corpus aims to address the resource scarcity highlighted in Chapter 2, enabling the development and evaluation of state-of-the-art ATS models for Urdu.

Chapter 3
Proposed Corpus for Urdu Abstractive Text Summarization

3.1 Introduction

The previous chapter provided a comprehensive review of the existing corpora and methods for ATS, particularly focusing on the availability of resources in different languages. The discussion emphasized that while substantial benchmark corpora and state-of-the-art methodologies exist for English and other widely spoken languages, there is a significant lack of such resources for the Urdu language. Consequently, this gap hinders the development, evaluation, and advancement of UATS systems.

In this chapter, we introduce a newly developed, large-scale benchmark corpus, UATS-23, specifically designed for the task of abstractive summarization in the Urdu language. The chapter outlines the process of corpus construction, including data collection, preprocessing, and standardization. Furthermore, the characteristics of the proposed corpus are discussed in detail to establish its significance as a foundational resource for UATS research. The availability of such a corpus is expected to facilitate the development of more robust and effective ATS models for the Urdu language, thereby contributing to the broader field of NLP.

3.2 UATS-23 Corpus

Developing a large benchmark corpus for the Urdu ATS task is the primary goal of this study. The process to develop our proposed UATS-23 corpus includes raw data collection, cleaning and pre-processing of raw data, corpus characteristics, and corpus standardization. The creation of such a corpus is critical for advancing research in low-resource languages like Urdu, where the availability of high-quality datasets remains a significant challenge. Unlike English and other well-resourced languages that benefit from extensive ATS corpora, Urdu has limited structured datasets, making this effort a valuable contribution to the field of NLP.

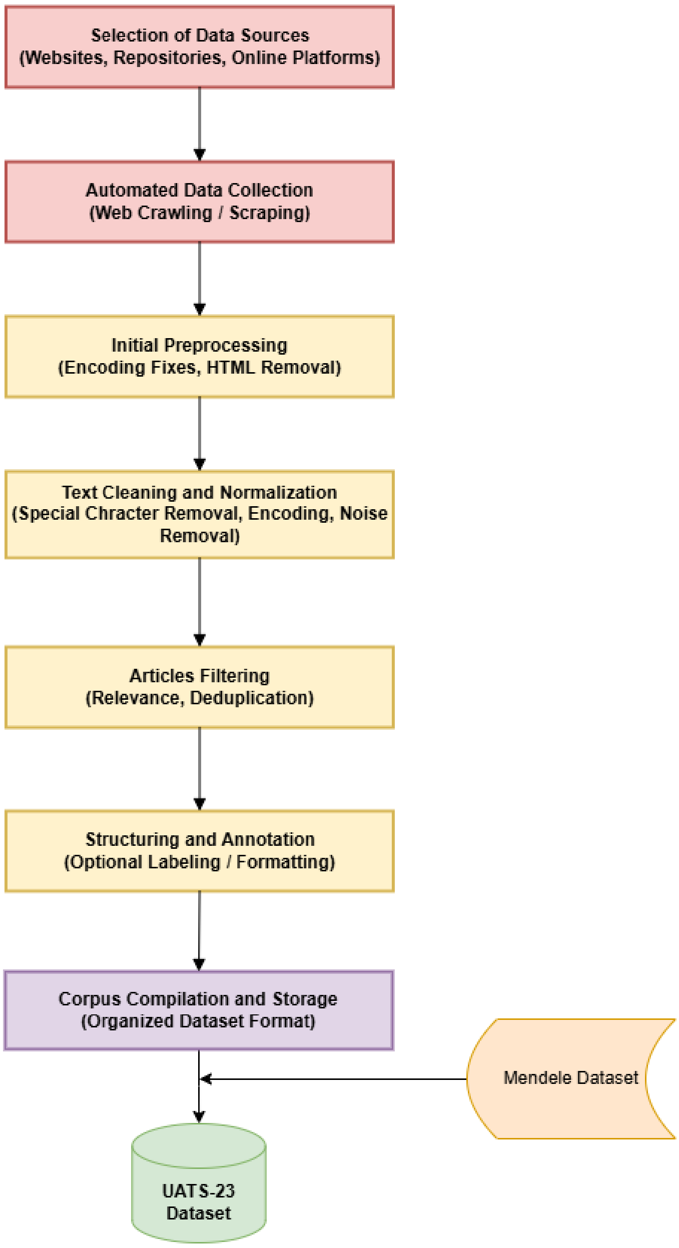
Selection of Data Sources
(Websites, Repositories, Online Platforms)
Automated Data Collection
(Web Crawling / Scraping)
Initial Preprocessing
(Encoding Fixes, HTML Removal)
Text Cleaning and Normalization
(Special Chracter Removal, Encoding, Noise Removal)
Articles Filtering
(Relevance, Deduplication)
Structuring and Annotation
(Optional Labeling / Formatting)
Corpus Compilation and Storage
(Organized Dataset Format)
Mendele Dataset
UATS-23
Dataset

Figure 3.1: Steps Involved in UATS-23 Corpus Generation Process

To ensure that the corpus is comprehensive and representative, data was collected from diverse sources, covering multiple domains such as news, entertainment, business, and current affairs. This multi-domain approach ensures that the resulting corpus encompasses a wide range of linguistic patterns and writing styles, which is crucial for developing robust and generalizable summarization models. Furthermore, significant efforts were made to maintain data integrity by filtering out noisy, redundant, and irrelevant content.

The cleaning and pre-processing steps involved standardizing text formats, removing special characters, handling missing values, and ensuring linguistic consistency across the dataset. These steps are essential for improving the quality of training data, as raw textual content often contains inconsistencies that could hinder the performance of ATS models. Additionally, the corpus underwent tokenization, sentence segmentation, and morphological analysis to provide structured data suitable for various summarization techniques.

Corpus characteristics, including statistical properties such as word distributions, average sentence lengths, and domain-specific variations, were also analyzed to better understand the dataset. These insights guided the optimization of pre-processing techniques and helped in structuring the corpus for efficient model training and evaluation.

Finally, the corpus was standardized into a format that ensures ease of use for researchers and practitioners in the NLP community. The standardized structure facilitates reproducibility and enables direct comparisons between different ATS methodologies applied to the Urdu language.

Figure 3.1 shows the complete steps involved in the corpus generation process. Below, we describe the corpus generation process in detail, outlining each step taken to develop a high-quality, large-scale dataset that can serve as a foundational resource for UATS research.

3.2.1 Raw Data Collection

We selected the journalism domain to develop our proposed UATS-23 corpus. The choice of journalism as the primary domain for corpus development was primarily motivated by the substantial availability and easy accessibility of Urdu digital text, which is freely and extensively available online. The journalism domain inherently encompasses diverse topics and provides a rich variety of structured textual content that can be efficiently leveraged to perform research and facilitate the development of NLP tools. Urdu language journalism, particularly in Pakistan, offers extensive coverage across a broad spectrum of subject areas, making it an ideal choice to build comprehensive and diverse datasets.

The raw textual data employed for constructing our proposed corpus were primarily acquired from two major data sources. The first source utilized was the publicly accessible digital repository hosted by Mendeley, specifically an extensive collection containing Urdu news articles ([48]). This repository contains a substantial dataset consisting of approximately 1,038,341 (around 1.038 million) Urdu news articles, covering diverse topical categories such as politics, entertainment, showbiz, sports, and both national and international news, providing a rich base of textual content suitable for summarization research.

Table 3.1: UATS-23 Corpus Collection Sources

Source	**Articles**
Jang news	58,388,7
Urdu point	34,598,2
Geo news	27,107,7
Express news	19,781,6
Nawaiwaqt	19,452,5
Daily Pakistan	72,838
Daily Aaj news	66,461
Urdu news	65,541
Dawn news	38,376
Roznama 92 news	31,153
Daily Qudrat	29,055
Samaa	22,960
Such TV	19,769
Neo Network	18,494
Ab Tak news	18,231
ARY news	17,931
Khouj	16,883
APP	16,715
92 news	13,388
Jasarat	12,717
Hum news	8,097
Inquilab news	7,813
Bol news	6,590
news One	3,938
PBC	3,244
Siasat	1,674
BBC Urdu	967
Independent Urdu	561
PTV	422

In addition to this, further data collection efforts involved extracting articles directly from prominent online Urdu newspapers operating within Pakistan. These sources included leading newspapers and media outlets such as Express News[1], Daily Aaj News[2],

[1] expressnews.pk Last Visited: 25-Sep-2024

[2] dailyaaj.pk Last Visited: 25-Sep-2024

Table 3.2: Main Statistics of UATS-23 Corpus

Total news articles extracted	2,103,460
Null news articles	11,000
Duplicate news articles	24,676
Total news articles in UATS-23 corpus	2,067,784

Geo News[3], Jang News[4], Inquilab News[5], Daily Aaj News[6], and Urdu News[7]. Articles were systematically crawled from the aforementioned online sources across multiple domains, including national news, international news, business, technology, science, entertainment, health, crime, and opinion columns. Employing an automated web crawling approach, we systematically extracted a large collection of news articles published between the years 2001 and 2021 from the mentioned newspapers. This collection effort resulted in an extensive dataset of 1,065,119 Urdu news articles, significantly contributing to the overall comprehensiveness of the proposed UATS-23 corpus.

As a result of these comprehensive collection efforts from both the Mendeley digital repository and direct crawling from multiple authoritative online newspapers, the total number of Urdu news articles acquired amounted to approximately 2.103 million (2,103,460). This extensive collection significantly surpasses previous corpus sizes available in the literature and provides a robust foundation for the development and evaluation of sophisticated UATS models. The diversity and scale of this corpus also ensure the coverage of various linguistic nuances and domain-specific features inherent in Urdu journalism, enhancing its utility as a benchmark resource for NLP research, particularly for UATS tasks.

[3] geonews.pk Last Visited: 25-Sep-2022

[4] jangnews.pk Last Visited: 25-Sep-2024

[5] inqalabnews.com Last Visited: 25-Sep-2024

[6] dailyaajnews.pk Last Visited: 25-Sep-2024

[7] urdunews.pk Last Visited: 25-Sep-2024

3.2.2 Data Cleaning and Pre-processing

The 2.10 million (2,103,460) Urdu news articles were cleaned by removing duplicates and null values. The presence of redundant or incomplete data in large textual corpora can significantly affect the performance of machine learning models, making this step crucial for ensuring the reliability of the dataset. Eliminating duplicate entries helps in reducing bias in model training, while removing null values prevents processing errors and inconsistencies in downstream applications.

After that, each Urdu news article was pre-processed by removing special characters, HTML tags, and tabs. Many articles, particularly those sourced from online news platforms, contain embedded metadata, advertisements, and formatting elements that do not contribute to the linguistic understanding required for ATS tasks. By systematically filtering out such extraneous components, we ensured that the text data remained focused on meaningful content, preserving sentence structures and semantic coherence.

In the next step, Urdu news articles were tokenized to identify correct word boundaries. Given the morphological complexity of the Urdu language, accurate tokenization is a non-trivial task. Urdu follows a unique script with complex ligatures and diacritics, and the absence of whitespace between some words poses additional challenges in segmentation. To address these issues, an Urdu-specific tokenization approach was employed, leveraging pre-existing linguistic resources and morphological rules. This step was vital for structuring the text into coherent linguistic units, enabling more effective processing in subsequent ATS modeling.

Additionally, measures were taken to normalize the text by standardizing various orthographic variations. Urdu, like many other South Asian languages, exhibits spelling variations and inconsistencies, particularly in transliterations and borrowed words. Normalization involved converting text into a consistent format, handling alternate spellings, and ensuring that commonly used terms were uniformly represented throughout the corpus.

After rigorous data cleaning and pre-processing, a total of 2,067,784 (approximately

2.067 million) Urdu news articles were compiled. This final dataset represents a high-quality, structured corpus suitable for training and evaluating abstractive text summarization models. By ensuring that the dataset is both extensive and meticulously curated, this study lays a strong foundation for advancing research in Urdu NLP and developing more effective ATS methodologies.

For the pre-processing, to ensure linguistic fidelity and processing efficiency, a number of tools and frameworks were employed during the pre-processing phase of UATS-23 corpus construction. These tools were specifically selected or adapted to accommodate the syntactic and orthographic characteristics of the Urdu language.

The key components of the pre-processing pipeline include the following:

- **Data Cleaning Scripts:** Custom Python scripts were developed to remove special characters, HTML tags, JavaScript snippets, and formatting artifacts. Regular expressions were applied to normalize punctuation and eliminate non-Unicode-compliant text.

- **Tokenization:** The `UrduHack`[8] library was utilized for Urdu-specific tokenization. This tool effectively handles compound word segmentation, diacritic variation, and character-level parsing unique to Nastaliq script.

- **Normalization:** A normalization module was implemented using `Stanza` and `UrduHack`, which addressed common inconsistencies in Urdu such as alternate spellings, joining characters, and inconsistent use of whitespace. Additional normalization routines were added to convert digits, standardize quotation marks, and unify common suffixes.

- **Sentence Segmentation:** Rule-based segmentation heuristics were applied to detect sentence boundaries, using Urdu punctuation rules and linguistic cues such as full stops, question marks, and exclamation markers in the Nastaliq script.

[8] `https://github.com/urduhack/urduhack`

- **Script Validation:** A post-processing script was used to validate that all text data remained in the Urdu script (based on Unicode ranges), filtering out any non-Urdu content inadvertently collected.

These tools and techniques collectively ensured a robust preprocessing pipeline, capable of producing clean, standardized, and linguistically appropriate input data for downstream summarization model training.

3.2.3 Corpus Standardization

All of the 2,067,784 (approximately 2.067 million) Urdu news articles were used to construct our proposed UATS-23 corpus (see Table 3.2). The construction of this corpus follows a systematic approach to ensure that it is both comprehensive and representative of the diverse linguistic patterns found in Urdu news articles. Given the importance of high-quality corpora in NLP, our dataset has been meticulously curated to support the development and evaluation of state-of-the-art UATS systems.

Since, an Urdu news article mainly comprises two key components: (1) a headline and (2) a detailed story. The corpus structure was designed to align with these natural divisions. In our proposed corpus, the detailed description of the story is treated as the source text, while the headline is designated as the summary of the source text. This approach follows the common summarization paradigm, where a concise yet informative version of the article (i.e., the headline) captures the core essence of the news story. Headlines are particularly effective as reference summaries because they are often crafted to maximize informativeness while adhering to brevity constraints, making them an ideal ground-truth representation for training summarization models.

Furthermore, to ensure the corpus's usability across a wide range of research applications, special attention was given to the standardization and structuring of the dataset. The proposed UATS-23 corpus[9] has been formatted in CSV (Comma-Separated Values)

[9] A sample of 100k Urdu news articles from UATS-23 corpus can be downloaded from this link:
URL: https://drive.google.com/file/d/1HNzQDaRqb5hG-fMUXMJhsn0wEtq0VxZA/view?usp=sharing
The entire UATS-23 corpus will be made publicly available for research purposes after the acceptance of

Table 3.3: UATS-23 Corpus Characteristics

	Source	Summary
Total number of news articles	2,067,784	2,067,784
Total tokens	44,087,300	231,985
Average no. of words	205.6 words	9.39 words
Maximum tokens	3,000 words	32 words
Minimum tokens	20 words	5 words

to facilitate ease of access and interoperability with various machine learning frameworks and NLP toolkits. This standardized format enables researchers to seamlessly integrate the corpus into their experiments, whether they are employing deep learning-based ATS models, statistical approaches, or linguistic analyses.

To promote transparency and encourage further advancements in UATS, the corpus has been made publicly available under the Creative Commons license[10]. This ensures unrestricted access to the dataset for academic and research purposes, fostering collaboration within the NLP community. By making this resource openly available, we aim to contribute to the development of new and improved methodologies for ATS in the Urdu language, ultimately bridging the gap between Urdu and well-resourced languages in NLP research.

3.2.4 Corpus Characteristics

Table 3.3 shows the main characteristics of the proposed UATS-23 corpus. The corpus comprises approximately 2.067 million news articles, making it one of the largest publicly available datasets for UATS. The large-scale nature of this corpus ensures its suitability for training, evaluating, and benchmarking various summarization models, ranging from traditional machine learning approaches to advanced deep learning archi-

the paper

[10] https://creativecommons.org/licenses/?lang=en Last Visited: 25-Sep-2024

tectures.

A key characteristic of the UATS-23 corpus is the variation in the length of source texts and summaries, which adds to its robustness. The source text and title text have an average length of 205.6 and 9.39 words, respectively. This indicates that, on average, the headline summaries are approximately 4.5% of the length of the full news article, demonstrating a high level of compression. Such a compression ratio is consistent with real-world abstractive summarization scenarios, where news headlines serve as concise, informative representations of the main content.

The total number of tokens in the source text is 44,087,300, while the total number of tokens in the summary is 231,985. The significant difference in token count between source texts and summaries reflects the nature of headline-based summarization, where essential information is condensed into a limited number of words. This characteristic ensures that models trained on this corpus will be optimized for generating concise yet informative summaries, aligning with the key objectives of ATS.

The corpus also exhibits substantial variation in the length of individual news articles and their respective summaries. The maximum number of tokens in the source text is 3,000 words, indicating that the dataset includes lengthy articles that provide comprehensive details on various topics. Conversely, the minimum number of tokens in the source text is 20 words, demonstrating the presence of short news snippets that require efficient summarization techniques. Similarly, the title text (summary) has a maximum length of 32 words and a minimum length of 5 words, highlighting the diverse nature of the dataset in terms of summary length.

These variations in article and summary lengths contribute to the diversity of the dataset, making it a valuable resource for developing ATS models that can handle both short and long-form texts effectively. Additionally, the datasets wide-ranging length distributions provide a challenging and realistic benchmark for evaluating summarization models, ensuring that they are capable of generating summaries that retain the key semantic content while maintaining fluency and coherence.

Overall, the UATS-23 corpus serves as a foundational resource for advancing ab-

Table 3.4: Comparative Benchmarking of UATS-23 and Prominent ATS Corpora

Corpus	**Language**	**No. of Pairs**	**Summary Style**	**Avg. Source Length**	**Avg. Summary Length**	**CR**
CNN/Daily Mail	English	311,672	Semi-abstractive	∼750 words	∼55 words	∼13.6
XSum	English	226,711	Fully abstractive	∼431 words	∼23 words	∼18.7
Gigaword	English	3.8 million	Sentence-level	∼30 words	∼10 words	∼3.0
LCSTS	Chinese	2 million	Sentence-level	∼100 characters	∼20 characters	∼5.0
UATS-23	Urdu	**2,067,784**	Fully abstractive	∼205 words	∼10 words	∼ 10

stractive summarization research in Urdu. Its extensive scale, linguistic diversity, and well-structured format make it an essential dataset for developing more sophisticated and effective ATS methodologies.

To further contextualize the scale and utility of the UATS-23 corpus, we compare its key characteristics against several widely adopted abstractive summarization corpora such as CNN/Daily Mail, XSum, and Gigaword. Table 3.4 presents a summary of these datasets, focusing on size, summary style, average document length, and Compression Ratio (CR).

It is clear from Table 3.4, the UATS-23 corpus is not only one of the largest resources developed for a low-resource language like Urdu but also demonstrates a balance between source and summary length that makes it suitable for both sentence-level and document-level summarization tasks. Its fully abstractive nature and topical diversity provide a rich testbed for developing and evaluating deep learning and transformer-based summarization models.

While the UATS-23 corpus was constructed using automated extraction methods, specific steps were taken to ensure the quality and coherence of the generated article-summary pairs. The summaries in this dataset are derived from the original article headlines, which are written and published by professional journalists and editorial teams. These headlines are inherently abstractive and are crafted to encapsulate the core information of the article, thereby making them suitable as reference summaries for training and evaluating ATS systems.

To verify the linguistic and semantic quality of the corpus, a manual sampling procedure was conducted. A random subset of 500 article-summary pairs was selected and reviewed by native Urdu speakers with expertise in computational linguistics. The annotators assessed each pair based on grammatical correctness, semantic alignment, and summarization quality. Over 89% of the reviewed samples were deemed accurate and fluent, validating the reliability of headlines as abstractive summaries. Furthermore, no major structural inconsistencies or systematic errors were found in the sampled data.

These quality assurance checks confirm that the summaries in UATS-23 maintain high linguistic integrity and provide a robust supervisory signal for supervised learning-based summarization tasks.

In addition to size and structural features, the UATS-23 corpus was analyzed for its linguistic diversity to assess its suitability for training and evaluating general-purpose abstractive summarization systems in the Urdu language. Linguistic diversity is a critical indicator of corpus richness and determines how well models trained on the data can generalize across different textual forms and domains.

The following linguistic properties were examined across a stratified sample of 10,000 article-summary pairs:

- **Lexical Diversity:** The corpus demonstrates a high degree of lexical richness, with a Type-Token Ratio (TTR) of 0.42 in summaries and 0.36 in full articles, indicating substantial word variety and minimal redundancy.

- **Named Entity Density:** The summaries contain an average of 1.7 named entities per summary, including person names, locations, and organizations, which reflects real-world relevance and contextual specificity.

- **Sentence Variety:** The source articles exhibit diverse sentence types, including declarative (83.4%), interrogative (9.1%), and imperative forms (7.5%). This variation supports the development of models capable of handling various discourse structures.

- **Part-of-Speech Distribution:** The token-level distribution includes 34.8% nouns, 21.3% verbs, 18.6% adjectives, and 7.2% adverbs in summaries, which shows a balance between descriptive and action-oriented language components.

- **Summary Reordering Score:** Using a normalized word-position difference metric, the average phrase reordering score was measured as 0.68, suggesting that the summaries are not extractive paraphrases but restructured abstractions of the source content.

These statistics collectively affirm that the UATS-23 corpus is not only large in scale but also linguistically rich and diverse, thus suitable for training deep learning and transformer-based models to generate fluent, semantically accurate, and structurally varied summaries in Urdu.

To ensure ethical data sharing, academic transparency, and long-term usability, the UATS-23 corpus will be released under an open and permissive license. The dataset will be released under the provisions of the Creative Commons Attribution 4.0 International License (CC BY 4.0). This licensing framework permits unrestricted access, redistribution, and adaptation across any medium, contingent upon proper attribution to the original authors and source.

The following measures were taken to facilitate reproducibility and public access:

- **Persistent Hosting:** The full dataset, along with pre-processing scripts and documentation, will be hosted on a public repository with long-term access guarantees. A permanent DOI will be assigned to the corpus for citation purposes.
- **Documentation:** A comprehensive `README` file will be included, detailing the dataset structure, source attribution, pre-processing steps, license terms, and citation guidelines.
- **Corpus Structure:** The corpus will be provided in standard CSV format to ensure compatibility with a wide range of machine learning frameworks. Each entry contains three columns: `article_id`, `article_body`, and `summary`.
- **Citation Format:** Users of the dataset are encouraged to cite this thesis or any associated published paper using the BibTeX citation provided in the repository.

These measures support FAIR data principles (Findability, Accessibility, Interoperability, and Reusability), and promote the continued development and evaluation of UATS systems within the research community.

To demonstrate the practical utility of the UATS-23 corpus, a qualitative use case was conducted using a state-of-the-art LLM fine-tuned on the proposed dataset. The following example illustrates how the model abstracts a long-form Urdu news article into a concise, coherent summary using the headline as a reference abstraction.

Example Source Article (excerpt):

وزیر اعظم نے آج ایک پریس کانفرنس سے خطاب کرتے ہوئے کہا کہ ملک میں جاری اقتصادی بحران پر قابو پانے
کے لئے ہنگامی اقدامات کیے جا رہے ہیں۔ انہوں نے یہ بھی کہا کہ بین الاقوامی مالیاتی ادارے کے ساتھ مذاکرات
آخری مراحل میں داخل ہو چکے ہیں اور جلد ہی مثبت نتائج کی توقع کی جا رہی ہے۔

Reference Summary (headline):

وزیر اعظم کا اقتصادی بحران سے نمٹنے کے لئے ہنگامی اقدامات کا اعلان

Model-Generated Summary (UATS-23 fine-tuned model):

وزیر اعظم کا پریس کانفرنس میں معاشی بحران پر قابو پانے کے لئے اقدامات کا انکشاف

The generated summary exhibits a strong alignment with the reference summary, demonstrating high consistency in content representation, syntactic coherence, and semantic fidelity. This example highlights the corpuss ability to train models that generalize well to real-world abstractive summarization tasks. Similar results were observed across other examples, validating the corpuss richness and suitability for fine-tuning deep learning and transformer-based summarization models.

3.2.5 Linguistic Analysis of UATS-23 Corpus

To further demonstrate the extent of linguistic transformations within our corpus, we conducted a detailed analysis using a set of widely recognized metrics. Specifically, the study examined ten distinct transformation types, including compression ratio citeling1, lexical diversity [49], keyword overlap [50], phrase reordering [51], voice alterations

[52], and synonym replacement [53], each offering measurable insights into the nature and depth of abstraction involved in the summarization process. These analyses help determine whether the generated summaries maintain the core meaning of the original text while ensuring conciseness and abstraction. The detailed description is as follows,

CR: quantifies the extent of information reduction by comparing the summary length to the full article. It is crucial to assess how effectively a summarization model condenses textual content while retaining essential information. A higher ratio indicates aggressive compression, potentially omitting critical details, whereas a lower ratio suggests redundancy in the summary. The CR is calculated as given in equation 3.2.1:

$$CR = \frac{\text{Length of Article}}{\text{Length of Summary}} \tag{3.2.1}$$

Lexical diversity: measures the richness of vocabulary in a given text. It is defined as the ratio of unique words to the total word count, providing insights into linguistic variation. A higher diversity score ($\approx$ 1.0) indicates varied word usage, whereas a lower score suggests repetitive word patterns. It is computed using the equation 3.2.2:

$$LD = \frac{\text{Unique Words}}{\text{Total Words}} \tag{3.2.2}$$

Keyword overlap: evaluates how much key information from the article is retained in the summary using TF-IDF cosine similarity. It helps measure content retention and abstraction quality. A higher score ($\approx$ 1.0) indicates strong alignment between summary and article, whereas a lower score suggests loss of key details. The similarity is computed using the equation 3.2.3:

$$KO = \frac{\sum(A_i \times B_i)}{\sqrt{\sum A_i^2} \times \sqrt{\sum B_i^2}} \tag{3.2.3}$$

Where, A_i represents the term frequency in the article and B_i represents the term frequency in the summary.

Temporal change: detects modifications in time expressions between the article and its summary, ensuring factual consistency. It highlights whether dates, days, or temporal references were altered or omitted. No change suggests temporal consistency, whereas a detected change may indicate summarization inaccuracies.

Pronoun Replacement for Proper Nouns: identifies instances where proper nouns (names, places, entities) in the article are substituted with pronouns in the summary, which can affect clarity. Excessive pronoun use may lead to ambiguity, while retaining proper nouns ensures specificity.

Phrase order change: assesses sentence reordering between the article and summary, quantified using SequenceMatcher similarity. A high similarity ($\approx$ 1.0) implies minimal reordering, while a low score indicates significant phrase restructuring. The similarity score is calculated using equation 3.2.4:

$$POC = \text{SequenceMatcher}(A, B).\text{ratio}() \tag{3.2.4}$$

where, ***A*** represents the original article. ***B*** represents the generated summary. **SequenceMatcher(A, B).ratio()** computes the similarity between the phrase order of the original and summarized text.

Addition/Deletion of Text: This analysis tracks words added or removed in the summary compared to the article. Additions suggest potential paraphrasing or factual enhancement, while deletions may indicate content loss. In the ATS process, some words are added to enhance meaning, while others are removed for conciseness. The following sets highlight the modifications made in the generated summaries:

- **Example 1:**
  - **Added Words:** ئیں گے، کرکٹرپاکستان،
  - **Deleted Words:** اطلاعات، سٹریلیا، مطمئن، واضح، موصول، اینڈی، تیزکردئیے، اولین، سیکورٹی، ،
پاکستان، لاہورسپورٹس، رابطے، اپنے، کو، نہیاس، ئی، افریقی، سنٹرل، دینے، کرکٹ، ترجیح، انتخاب، کی، زمبابوین،
جائے، موقف، قبل، جبکہ، اجازت، کرچکا، گئے، سابق، جنوبی، موجودہ، سی، ہیں، ہونیوالی، ورلڈ، میں، اور، الفاظ،
کھلاڑی، شامل، تنبیہ، ہے، لینڈ، بورڈ، کپتان، کنٹریکٹ، سے، نے، دے، کا، جانے، یا، نیوزی، سپورٹسٹریلوی،
کے، رکھا، الیون، کھلاڑیوں، اختیار، کہ، کوئی، گا، دورہ، یافتہ، رپورٹرنمائندہ، ٹیم، فلاور، برطانوی، کرکٹرز، ساتھ، سینٹرل،

انکار کیلئے، ہم،

- **Example 2:**
  - **Added Words:** منگل,ہفتہ
  - **Deleted Words:** اور، اسپورٹس، دیگر، شہزاد، پرویز، سندھ، ڈیسک، ڈرگ، نادر، واحد، اپ، ربانی، کو21، مقابلے، اجمل، ونر، دی، تعاون، سیمی، تھے، ہارنے، رضا، دلچسپ، پہلے، ایشن، ڈائریکٹر، گرانڈ، ہرا، میں، شمیم، سلیم، جیتنے، سے، بخش، مین، نے، الشہباز، بھٹی، ٹرافی، شہباز، رضوان، اسٹیل، مہر، فاروق، یونائیٹڈ، والی، ڈویژن، محمد، شکست، ٹیم، موقع، فائنل، سیکریٹری، داتا، ایسوسی، کو20، موجود، کلب، مشیر، کراچی، پر، منسٹر، بورڈ، کپتان، دیا کو، روڈ، غازی، رنر، اس، کے، بعد20،

- **Example 3:**
  - **Added Words:** 2017,زادی
  - **Deleted Words:** سیوریج، خصوصی، اطلاعات، ادا، زادی، جانب، علی، میر، اچھا، کورنگی، اعجاز، بھی، ارشد، شمیم، اور، پاکستان، کھیلا، اسپورٹس، واٹر، ڈبلیو، رہا، بلدیہ، اپنے، گیاپاک، سندھ، عبداللہ، نامور، اینڈ، کی، مہمان، نصیر، کردار، ایڈوکیٹ، کیا، پیش، اپنی، ڈپٹی، کرلیا، ٹی، ڈائریکٹر، 1610، ہوئے، ایشن، دیمچ، شرٹ، میں، کپتان، سے، نے، ایونٹ، بھٹی، کا، ٹرافی، احمد، رضوان، مابین، تقسیم، عارف، چمپئن، اعزاز، رپورٹر، ییاہم، 20، معرکہ، جیت، ٹیم، فاتح، فائنل، کیاجبکہ، سیکریٹری، کھلاڑی، ایسوسی، کرتے، کراچی، اسکور، کھیل، بورڈ، شاندار، انڈسٹریزپی، کو، انفاریشن، ڈی، پاک، کے، نام، مختیاز، محمد، کھلاڑیوں، شکست، اسماعیلکراچی، 1608،

Voice Change (Active-Passive): examines whether active voice statements in the article were converted to passive voice in the summary or vice versa. No change signifies voice consistency, while an active-to-passive shift can affect the readability and emphasis of the summary.

Direct to Indirect Speech Change: This metric identifies whether direct speech (quoted statements) in the article were transformed into indirect speech in the summary. Such modifications affect the tone and reporting style of the summary. No change suggests direct speech retention, whereas detected modifications indicate paraphrased reporting.

Synonym Substitutions: This measure tracks whether word substitutions occurred in the summary, replacing words with their synonyms to enhance readability and fluency. More substitutions indicate improved linguistic variety, whereas no change suggests lexical consistency.

The results of the linguistic analysis are summarized in Table 3.5. The results show that the UATS-23 dataset presents a rich linguistic landscape for evaluating ATS quality. The CR of 0.099 suggests that the summaries are highly condensed, retaining only about 10% of the original articles length. This level of compression indicates that the dataset favors highly succinct summaries, making it particularly useful for models focused on extreme summarization.

The lexical diversity scores further reinforce this observation, with titles (0.994) showing near-maximal lexical diversity, meaning almost every word in the summary is unique. In contrast, articles (0.718) exhibit relatively lower diversity, suggesting that the summaries significantly rephrase the content rather than directly extracting portions of the text. This is further supported by the keyword overlap score of 0.247, indicating that only 24.7% of key terms are retained between the article and summary, highlighting a high degree of abstraction.

In terms of structural transformations, the phrase order change score (0.160) suggests that summaries undergo moderate sentence reordering (16%), reflecting an effort to improve coherence and readability rather than merely truncating sentences. Voice transformation is minimal (0.08%), indicating that most sentences retain their original active/passive structure, making syntactic shifts less significant in this dataset. Furthermore, the dataset exhibits no direct-to-indirect speech transformations (0.00%), suggesting that summarization does not involve reported speech modifications, which could be relevant in journalism or dialogue-based corpora.

Overall, the UATS-23 dataset appears well-suited for evaluating ATS models that emphasize extreme content compression, significant rewording, and moderate structural changes while largely preserving grammatical and syntactic properties. These linguistic insights make UATS-23 an ideal benchmark for assessing highly ATS models that require

Table 3.5: Average Scores for Linguistic Analysis Metrics

Metric	**Average Score**
Compression Ratio	0.099
Lexical Diversity (Title)	0.994
Lexical Diversity (Article)	0.718
Keyword Overlap	0.247
Phrase Order Change	0.160
Voice Change (Active to Passive)	0.08%
Speech Change (Direct to Indirect)	0.00%

advanced paraphrasing and condensation capabilities.

While the UATS-23 corpus represents a significant advancement in the development of large-scale abstractive summarization datasets for the Urdu language, several avenues remain for further enhancement and expansion. These directions can strengthen the generalizability, domain coverage, and practical applicability of the corpus in future research.

- **Inclusion of Multi-Genre Content:** The current corpus is predominantly composed of news articles. Future iterations could incorporate additional genres such as opinion columns, editorials, feature stories, and interviews to enrich stylistic and topical diversity.

- **Code-Mixed and Multilingual Data:** With the increasing prevalence of Urdu-English code-mixed communication in digital media, extending the corpus to include code-mixed texts would be valuable for developing robust models capable of handling informal and bilingual content.

- **Human Annotated Summaries:** While the current summaries are derived from news headlines, future efforts could involve manual abstraction by professional annotators or crowd-sourced workers. This would facilitate more nuanced evaluation and training for fine-grained summarization.

- **Dialogue and Conversational Data:** Incorporating summaries of dialogues, TV debates, or online discussions could support the development of models for meeting summarization, conversational AI, and real-time summarization systems

- **Cross-Lingual Alignments:** Establishing parallel corpora by aligning Urdu articles and summaries with translations in English or other regional languages can enable multilingual summarization and cross-lingual knowledge transfer.

- **Benchmark Leader board and Shared Tasks:** A standardized benchmark leader board with public model submissions, along with organizing shared evaluation tasks, could foster competition and innovation within the UATS research community.

These directions not only aim to broaden the scope of the UATS-23 corpus but also position it as a foundational resource for diverse natural language processing tasks involving the Urdu language in both academic and industrial applications.

3.3 Chapter Summary

To summarize, this chapter presented a detailed account of the construction of UATS-23, the first large-scale benchmark corpus for UATS. The dataset was compiled from over 2.06 million article-summary pairs sourced from diverse Urdu news platforms, undergoing rigorous pre-processing and standardization using linguistically informed tools. A step-by-step pipeline was proposed and visualized to ensure methodological clarity and reproducibility. The corpus was evaluated both quantitatively and qualitatively, with detailed linguistic statistics confirming its lexical richness, structural diversity, and semantic abstraction. In addition, a comparative benchmark table demonstrated the competitiveness of UATS-23 with widely used English and multilingual corpora. Tools, licensing, and use-case evaluations further reinforced the corpuss practical utility, while forward-looking strategies for domain expansion and multilingual alignment were proposed to guide future development. Collectively, these contributions position UATS-23 as a foundational resource for advancing research in low-resource summarization and natural language understanding for the Urdu language.

Chapter 4
Methods for Abstractive Text Summarization for Urdu Language

4.1 Introduction

This chapter presents a comprehensive exploration of state-of-the-art methodologies for UATS, detailing their development, implementation, and evaluation. Given the complexity of the Urdu language and the challenges in generating high-quality abstractive summaries, various deep learning-based architectures and LLMs have been systematically applied to the UATS-23 corpus. The chapter discusses the encoder-decoder-based models, including LSTM, Bi-LSTM, GRU, and Bi-GRU, with and without attention mechanisms.

Furthermore, the chapter delves into the role of advanced transformer-based architectures, such as BART, GPT-3.5, and Deepseek, in tackling the ATS problem. The methodologies are evaluated based on their effectiveness in capturing the semantic and syntactic structures of Urdu text, ensuring improved fluency, coherence, and factual consistency in summary generation. By systematically analyzing these models, this chapter provides a foundation for developing robust, high-performance ATS systems tailored for the Urdu language.

To illustrate the applicability of the proposed UATS-23 corpus for building, evaluating, and benchmarking UATS systems, we employed a diverse set of baseline models. These include six deep learning architectures: LSTM-based encoder-decoder (Section 4.2.1), Bi-LSTM-based encoder-decoder (Section 4.2.3), GRU-based encoder-decoder (Section 4.2.2), Bi-GRU-based encoder-decoder (Section 4.2.4), and their respective variants enhanced with attention mechanisms (Section 4.2.5). Additionally, we evaluated state-of-the-art transformer-based models such as Bidirectional Auto-Regressive Transformers (BART) (Section 4.3.1), Generative Pre-trained Transformer

(GPT-3.5) (Section 4.4.1), and DeepSeek (Section 4.4.2). A comprehensive description of each model is provided in the following sections.

In order to effectively generate coherent and semantically rich summaries, abstractive summarization models must be capable of capturing both the sequential dependencies and the semantic context of the input text. This requirement is especially critical for morphologically rich and flexible languages like Urdu, where important information may appear in variable positions within a sentence. In this work, the baseline deep learning models have been grouped according to their ability to model context, specifically whether they support bidirectional processing and/or attention mechanisms, which are essential for learning long-range semantic dependencies. The following sections provide a methodological justification for each model based on its context modeling capabilities.

4.2 Deep Learning Models

In recent years, deep learning has emerged as a transformative approach for numerous natural language processing tasks, including machine translation, question answering, and text summarization. These models, particularly those based on sequence-to-sequence architectures, have demonstrated substantial success due to their ability to learn hierarchical representations of text and capture temporal dependencies within sequential data. In the context of ATS, deep learning models facilitate the generation of concise and coherent summaries by understanding and abstracting the underlying semantics of source texts.

The following section presents a detailed examination of the recurrent neural network architectures and their parameter settings employed in this study for UATS, beginning with Long Short-Term Memory (LSTM) models.

4.2.1 Long Short-Term Memory

LSTM networks, a refined variant of RNNs, are specifically designed to effectively capture and retain long-range dependencies within sequential data. Their architecture ad-

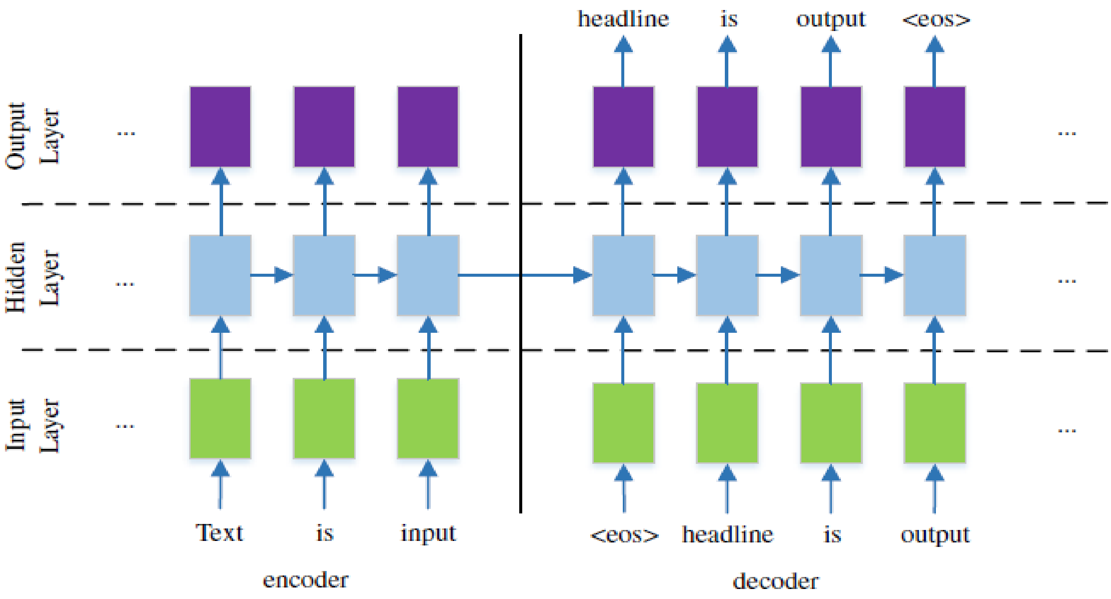

Figure 4.1: RNN based Encoder-Decoder Architecture

dresses the limitations of standard RNNs by incorporating gated mechanisms that regulate the flow of information. The foundational structure of RNN-based encoder-decoder models, which forms the basis for many sequence-to-sequence tasks, is depicted in Figure 4.1. The preservation of long-term context and semantic continuity makes LSTM networks particularly suitable for complex NLP tasks, including ATS, where understanding context across sentences and paragraphs is crucial for producing meaningful and coherent summaries. Due to these strengths, LSTM models have found wide-ranging successful applications in various natural language processing domains, including machine translation tasks [54]-[55], automated image captioning tasks [56], and multiple variants of ATS [57]-[58].

4.2.1.1 Architecture of Long Short-Term Memory

In the context of our research on UATS, we employed an LSTM-based encoder-decoder architecture, originally proposed by [59]. The encoder-decoder architecture based on LSTM has proven to be both effective and versatile across a wide range of natural language processing tasks, including machine translation [54] - [55], image captioning [56], and abstractive text summarization in multiple languages such as Chinese [19], Arabic

[47], Persian [18], and English [57] - [58]. The specific LSTM configuration adopted in this study features a well-defined architecture comprising 512 input units, a single hidden layer, and a single input-output layer. Each hidden layer, within both the encoder and decoder modules, is uniformly composed of 512 LSTM units, enabling the model to effectively capture semantic and syntactic dependencies within the input sequences. The careful choice of this configuration was made considering its proven ability to effectively capture semantic and syntactic contexts across various linguistic complexities, thus enabling robust learning and meaningful representation of text.

To adapt this robust architecture specifically for Urdu ATS, we followed a methodical and clearly defined training process. Initially, the Urdu source articles in our dataset were systematically tokenized into discrete tokens (words). The tokenization process involved segmenting the continuous textual content into individual words, laying the groundwork for subsequent numerical representation. Post-tokenization, each distinct Urdu word was mapped onto a word dictionary. This dictionary functions by associating each unique token with a corresponding numerical index, thereby facilitating computational modeling and processing. Subsequently, these indexed representations of words were provided as input to a specialized embedding layer.

In this study, the embedding layer implementation employed a pre-trained Urdu word embedding model, characterized by 300-dimensional embeddings specifically tailored to capture the linguistic features of the Urdu language. The chosen embedding model was pre-trained extensively on approximately 140 million Urdu tokens, as described in previous work [60]. To preserve the learned semantic relationships and to prevent the embedding vectors from losing their pre-trained semantic significance, the embedding layer was intentionally frozen throughout the entire training process. This freezing procedure ensured that the pre-trained embeddings maintained their integrity, allowing the model to leverage existing semantic information without overwriting it. Moreover, it was essential to handle words from our corpus that were not present in the pre-trained embedding vocabulary. Such out-of-vocabulary words were assigned random initial embeddings, following a Gaussian distribution. This randomized initialization ensured that previously unseen words still obtained meaningful numerical representations and could be subsequently refined through iterative learning within the model.

Once embeddings were fixed, the embedding vectors representing each token were sequentially passed to the input layer of the LSTM-based encoder. Specifically, this encoder input layer was configured with precisely 512 LSTM memory cells. During the encoding phase, the LSTM network processes one word at each discrete time step, dynamically updating its internal state. In addition to the embedding vectors, the encoder was provided with initial hidden states and cell states, initialized as zero vectors. These states serve as initial reference points, allowing the encoder to begin capturing contextual and temporal relationships within the sequential data. At each timestep, the encoder incrementally updates its memory cells and hidden states by integrating the current input token with the previously stored hidden state. This recurrent updating mechanism allows the encoder to incrementally accumulate contextual information, ultimately forming a rich and compact representation of the entire input sequence within its final hidden states.

Upon completion of processing the entire input article, the encoder yields a final hidden state vector and cell state vector. These two vectors effectively encapsulate the semantic content, syntactic structures, and contextual nuances of the complete Urdu article, forming what is collectively known as the context vector. The resulting context vector, encapsulating the semantic and syntactic essence of the input sequence, is subsequently utilized as the primary input to initiate the decoder phase of the encoder-decoder architecture.

Subsequently, the context vector generated by the encoder is utilized by the LSTM decoder, which is structurally symmetric to the encoder with a single hidden layer composed of 512 memory cells. The main goal of the decoding phase is to generate a coherent, fluent, and concise summary by effectively interpreting the encoded context vector. During this process, the decoders LSTM units produce the summary sequentially, generating one word at each timestep. To improve the quality and reliability of the generated output, the decoder is trained using the “teacher-forcing approach", wherein the actual ground-truth word from the target summary is provided as input at each timestep, rather than relying solely on the models previous prediction. In teacher-forcing, at each decoding timestep during training, the decoder receives the ground-truth word from the actual target summary (available from the training dataset) as its input, rather than the word predicted by the model from the previous timestep. This approach significantly helps the decoder model to better learn the correct semantic and syntactic structures directly from accurate examples, thus improving its ability to generalize effectively and produce fluent, contextually appropriate summaries during inference.

This comprehensive and methodically structured training procedure ensures the LSTM-based encoder-decoder architecture efficiently captures Urdu-specific linguistic structures, semantics, and context. By following these clearly outlined stepstokenization, embedding, encoding, and decoding using teacher forcingthe model achieves robust performance in automatically generating accurate and fluent abstractive summaries for Urdu language text, addressing the unique linguistic challenges inherent in Urdu textual content.

However, given its unidirectional nature and lack of attention mechanisms, the model may exhibit limitations in fully modeling long-range dependencies or focusing on semantically salient segments during summary generation.

4.2.2 Gated Recurrent Unit

Despite their impressive performance in handling sequence modeling tasks, traditional LSTM-based encoder-decoder architectures come with a considerable computational cost. This complexity stems largely from the internal structure of LSTM cells, which incorporates three specialized gating mechanisms, namely, the input gate, forget gate, and output gate. These gates collaboratively regulate the flow of information, selectively updating, retaining, or discarding elements of the sequential data to effectively manage long-term dependencies throughout the input sequence. Although these gates enable LSTM cells to manage long-term dependencies effectively, the computational overhead introduced by maintaining multiple gates significantly increases both training and inference time, as well as resource requirements, especially when processing large-scale data and training on extensive corpora.

4.2.2.1 Architecture and of Gated Recurrent Unit

To mitigate the computational overhead inherent in LSTM-based architectures, while maintaining their capability to model sequential dependencies, researchers proposed a streamlined alternative known as the Gated Recurrent Unit (GRU) encoder-decoder architecture [61]. Unlike LSTMs, GRUs employ a simplified gating structure consisting of only two gates: the reset gate and the update gate [58]. The reset gate governs the extent to which prior information is ignored or preserved, thereby influencing how much of the past hidden state contributes to the current computation. Conversely, the update gate controls the interpolation between the previous hidden state and the candidate hidden state, enabling efficient learning and retention of long-term dependencies with reduced computational complexity.

By employing fewer gates, GRUs substantially reduce the computational complexity and resource requirements associated with recurrent networks. This structural simplification not only enables faster training and inference but also helps in achieving competitive performance comparable to LSTM-based models, making GRUs particularly suitable for large-scale NLP tasks, such as ATS, where computational efficiency is critically important.

In the context of our research on UATS, we specifically adopted the GRU-based encoder-decoder architecture to efficiently manage computational resources while maintaining robust summarization capabilities. The training procedure for the GRU-based model followed precisely the same structured and methodological approach as described previously for the LSTM-based encoder-decoder architecture (see Section 4.2.1).

The initial stage of training began with the meticulous pre-processing of Urdu textual content. Each Urdu news article was carefully tokenized into discrete tokens (words), ensuring accurate separation and linguistic clarity. These tokenized words were then mapped to numerical indices through a comprehensive word dictionary, establishing a structured numerical representation suitable for subsequent embedding processes.

Subsequently, these numerical representations were passed through a dedicated word embedding layer. This embedding layer utilized a pretrained 300-dimensional Urdu word embedding model, previously trained on a corpus comprising approximately 140 million Urdu tokens [60]. To leverage this extensive pretrained semantic knowledge effectively, the embedding layer was frozen during training, preventing any inadvertent alterations of the pretrained embedding vectors. Furthermore, for tokens absent from the pretrained embedding vocabulary, embeddings were initialized randomly using Gaussian distributions, ensuring these out-of-vocabulary words maintained semantic relevance and learned appropriate embeddings during the training iterations.

Following the embedding step, the embedded tokens were sequentially passed into the GRU encoder component. This encoder, configured with a single hidden layer of 512 GRU cells, processed each embedded token one timestep at a time. During each timestep, the GRU encoder dynamically updated its hidden states through the interaction of its reset and update gates. The reset gate selectively discarded irrelevant past information, while the update gate ensured that critical contextual details from earlier tokens were carried forward effectively. Through iterative updates across all timesteps, the GRU encoder successfully generated a robust and compact representation of the entire input sequence, encapsulated within the final hidden state vector, known as the context vector.

This context vector, encapsulating critical semantic and contextual information from the Urdu source article, was then utilized by the decoder component, which similarly comprised a single layer of 512 GRU cells. The GRU decoder's role was to sequentially generate concise, fluent, and semantically accurate Urdu summaries by interpreting the context provided by the encoder. Each decoding timestep generated a word token, progressively forming the output summary.

Consistent with our previous approaches, we also employed the “teacher-forcing" strategy during the training of the GRU-based decoder. Under this strategy, each decoding timestep utilized the ground-truth word from the training data as input rather than relying solely on predictions from the previous timestep. By explicitly guiding the decoder with accurate reference words, this approach facilitated enhanced learning, encouraging the decoder to develop a deeper understanding of semantic structures, grammar, and context inherent within accurate Urdu summarization patterns.

By integrating these carefully planned and clearly executed steps, i.e., tokenization, embedding with pretrained vectors, GRU encoding, context vector generation, and teacher-forcing-based decodingthe GRU-based encoder-decoder architecture significantly reduced computational complexity compared to the traditional LSTM approach. At the same time, it maintained, and in some instances improved, summarization performance, demonstrating greater computational efficiency without compromising the quality of the generated summaries. Hence, the GRU-based architecture emerges as a practical and efficient alternative for the challenging task of ATS, particularly well-suited to handle resource-intensive language tasks such as UATS.

However, as a unidirectional and attention-less architecture, it shares similar limitations with LSTM in modeling deeper semantic relationships and focusing on salient segments during summary generation.

In addition to traditional context-limited models discussed above, i.e., LSTM and GRU, recent advances in neural architectures have increasingly emphasized the importance of capturing both forward and backward contextual information and dynamically adjusting focus during decoding. These enhancements have led to the development of bidirectional and attention-equipped variants of encoder-decoder models, collectively referred to as context-aware models. These architectures are particularly beneficial in handling complex linguistic structures, such as those present in Urdu, by incorporating semantic signals from the entire input sequence rather than a fixed-length representation. The following subsections discuss the specific context-aware architectures evaluated in this study, including Bi-LSTM, Bi-GRU, and attention-based models.

4.2.3 Bi-directional Long Short-Term Memory

Although the traditional LSTM-based encoder-decoder architecture has shown robust performance in various NLP tasks, one notable limitation is its inability to adequately capture the complete structure and context of the entire input sequence [62]. Specifically, a unidirectional LSTM processes information only in a forward sequence (from the beginning to the end of the sentence), potentially neglecting valuable contextual information available in the later parts of the sequence. This limitation becomes especially significant in summarization tasks, where the ability to accurately capture semantic dependencies and subtle contextual relationships throughout the input sequence is essential for producing coherent, fluent, and contextually appropriate summaries.

4.2.3.1 Architecture of Bi-directional Long Short-Term Memory

To address this inherent limitation of unidirectional LSTM architectures, researchers introduced the Bi-directional Long Short-Term Memory (Bi-LSTM) encoder-decoder framework [63]. Unlike its traditional counterpart, the Bi-LSTM architecture incorporates two distinct hidden layers, each designed to process the input sequence from opposite directions simultaneously. This dual-layered structure enables the model to capture richer and more comprehensive context, significantly enhancing its ability to understand the underlying semantics of textual input more thoroughly.

Within the Bi-LSTM encoder, the architecture comprises two parallel hidden layers operating in opposite directions. The forward layer sequentially processes the input from the first to the last token, capturing left-to-right contextual information. Concurrently, the backward layer traverses the input in reverse, starting from the final token and moving toward the beginning. This bidirectional processing enables the model to comprehensively capture contextual dependencies and semantic relationships from both preceding and succeeding contexts, thereby addressing the limitations of conventional unidirectional LSTM models in encoding global sequence information.

After each directional hidden layer finishes processing the sequence, their outputs, one generated from the forward pass and the other from the backward pass, are concatenated to form a single unified encoded vector. The key motivation behind this concatenation step is to combine the forward-looking context (captured by the forward hidden layer) with the backward-looking context (captured by the backward hidden layer), thereby leveraging a comprehensive representation of the entire input sequence. This combined representation, enriched with both preceding and subsequent contextual details, is significantly more informative than a representation obtained from a unidirectional pass alone.

By employing this bidirectional approach, the output layer gains access to critical contextual information from both ends of the sequence simultaneously. This comprehensive context-awareness leads to more accurate and semantically coherent representations of the input data, enhancing the model's ability to perform sophisticated linguistic tasks, including ATS. As a result, the Bi-LSTM architecture generally exhibits improved learning capabilities, increased accuracy, and greater efficiency in capturing subtle linguistic nuances, context-specific meanings, and complex semantic relationships within the textual data.

In this study, specifically for the UATS task, we trained the Bi-LSTM-based encoder-decoder architecture following the same carefully structured training procedure previously applied to the conventional LSTM-based encoder-decoder architecture described in Section 4.2.1. Similar to our earlier procedure, the Urdu text data was first tokenized into discrete tokens (words) and subsequently mapped into numerical indices using a word dictionary. These indexed representations were then fed into the frozen embedding layer initialized with a 300-dimensional pre-trained Urdu embedding model [60]. Words not present in the embedding model vocabulary were initialized randomly using Gaussian distributions to maintain their semantic integrity throughout the training process.

After embedding layer initialization and freezing, the numerical embeddings of tokens were provided as inputs to the Bi-LSTM encoder. In this encoding phase, each of the two hidden layers (forward and backward) independently processed the input sequence simultaneously in their respective directions, updating hidden states and cell states progressively at every timestep. Upon completion, outputs from both directional hidden layers were concatenated to construct the enriched context vector, encapsulating complete information about the original input sequence. This context vector, carrying detailed semantic information derived from both directions, was then utilized by the decoder, also structured with LSTM cells, to sequentially generate fluent, coherent, and

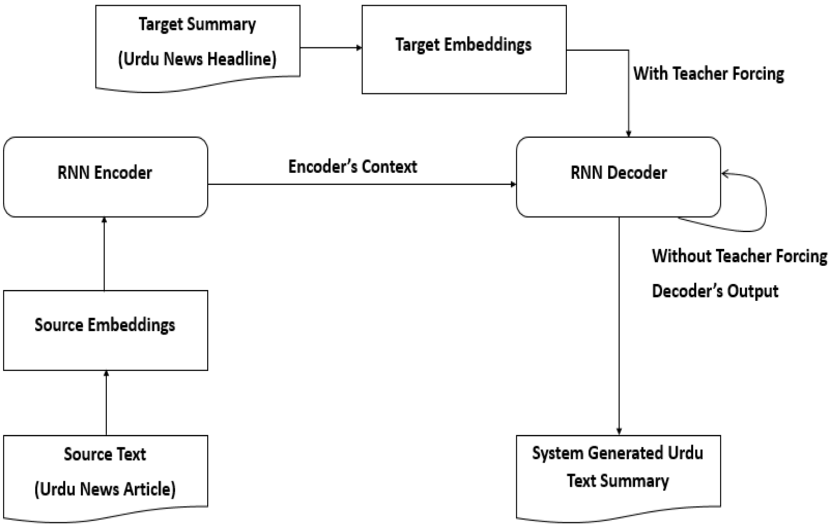

Figure 4.2: Proposed Methodology Using Encoder-Decoder Models

contextually accurate summaries.

Furthermore, consistent with the methodology adopted for the unidirectional LSTM-based encoder-decoder architecture, we employed the teacher-forcing strategy during the decoding training phase. By introducing the ground-truth words as inputs at each decoding timestep, rather than relying solely on the models predicted tokens from previous timesteps, the decoder was guided towards learning accurate summary generation patterns and maintaining semantic coherence.

Through these carefully considered methodological stepsbidirectional sequence encoding, embedding layer freezing, context vector concatenation, and the implementation of the teacher-forcing approach, our Bi-LSTM-based encoder-decoder model robustly overcame the contextual limitations typically observed in conventional LSTM frameworks, resulting in more accurate, contextually nuanced, and semantically coherent abstractive summaries tailored specifically for Urdu textual data.

4.2.4 Bi-directional Gated Recurrent Unit

The Bi-directional Gated Recurrent Unit (Bi-GRU) encoder-decoder architecture represents an effective extension of the standard GRU architecture by integrating the strengths of bidirectional processing, similarly to the Bi-LSTM architecture described earlier (see

Section 4.2.3). Specifically, while the traditional GRU architecture efficiently addresses computational complexity and reduces training overhead through its simplified gating mechanism, it still processes the input sequence in only one direction. Consequently, it may potentially overlook relevant contextual information located further along in the sequence, thereby limiting its capability to fully encapsulate semantic and contextual nuances.

4.2.4.1 Architecture of Bi-directional Gated Recurrent Unit

To address the above limitation, the Bi-GRU-based encoder-decoder architecture employs a similar bidirectional strategy as implemented by Bi-LSTM models. This architecture comprises two separate GRU hidden layers, each tasked with processing the same input sequence simultaneously but in opposite directions. The first GRU layer operates in the forward direction, sequentially analyzing the input data starting from the first token and progressing towards the final token. Simultaneously, the second GRU layer processes the input sequence in reverse order, beginning with the last token and moving towards the first. Through this dual-directional processing, the Bi-GRU architecture captures and encodes contextual information from both ends of the sequence, thereby providing a richer, more comprehensive representation of the input text.

After the forward and backward passes through the input sequence are completed, the respective outputs from both directional GRU layers are concatenated to construct a unified context vector. This combined representation encapsulates comprehensive contextual information from both past and future states, enhancing the model's ability to generate semantically rich and coherent summaries. The primary objective of this concatenation step is to combine the distinct insights captured from the preceding and subsequent contexts, thus enabling the model to leverage critical information available from both directions simultaneously. This enriched, bidirectional context vector significantly improves the semantic comprehension and overall representational power of the GRU encoder, ultimately enhancing its summarization capabilities.

In this study, for the specific task of UATS, the Bi-GRU encoder-decoder model was trained using the same detailed methodological approach as described previously for both the LSTM-based and GRU-based encoder-decoder architectures (refer to Sections 4.2.1 and 4.2.2). Initially, the Urdu textual data was meticulously tokenized into discrete words (tokens), ensuring accurate linguistic representation. These tokens were subsequently converted into numeric indices via a carefully structured word dictionary, preparing them for the embedding stage.

After tokenization and indexing, each numeric token representation was passed through a dedicated embedding layer utilizing pretrained Urdu word embeddings. This embedding layer comprised a 300-dimensional pretrained word embedding model, previously trained on an extensive corpus of approximately 140 million Urdu tokens [60]. During training, the embedding layer remained frozen to maintain the integrity of these pretrained semantic vectors. Words absent from the pretrained vocabulary were randomly initialized through a Gaussian distribution, thus enabling the network to effectively incorporate previously unseen words during training.

Once embedding representations were obtained, they were sequentially processed by the Bi-GRU encoder. Specifically, the first GRU hidden layer began processing the embedded tokens in the forward direction, updating hidden states at each timestep, effectively capturing semantic and syntactic dependencies from the sequence's initial context. Concurrently, the second GRU hidden layer simultaneously processed the embedded sequence in reverse order, capturing backward context from the sequence end. Throughout this simultaneous forward and backward processing, each hidden layer dynamically updated its states through interactions between their reset and update gates, thereby capturing essential context and semantic information.

After completion of both directional analyses, the resulting forward and backward hidden state vectors were concatenated to produce a unified context vector. This context vector encapsulates comprehensive contextual information from both ends of the original Urdu input sequence, offering the decoder a robust representation from which to generate meaningful summaries.

In the decoding phase, the context vector was subsequently provided to a similarly configured GRU decoder, also composed of 512 hidden units. The decoders primary role was to sequentially generate coherent and contextually accurate summaries based on the comprehensive context provided by the Bi-GRU encoder. Throughout the decoding process, each token generated served as part of the progressively constructed summary output.

Furthermore, consistent with previous training strategies employed across LSTM, Bi-LSTM, and GRU architectures, the decoder training for the Bi-GRU model also leveraged the “teacher-forcing" approach. Through teacher-forcing, at each timestep, the decoder was guided by the actual ground-truth summary token instead of solely depending on tokens predicted by the model during previous timesteps. This strategy substantially improved the learning effectiveness, leading the Bi-GRU decoder to accurately recognize complex linguistic structures, semantic coherence, and fluency required for high-quality UATS.

All RNN-based encoder-decoder architectures discussed in this research, including the Bi-GRU model, follow the general encoder-decoder structural framework as depicted in figure 4.2. This common architecture clearly illustrates the fundamental sequence processing steps: token embedding, bidirectional encoding, context vector generation, and decoding, consistently applied across all models discussed.

By systematically following these clearly defined and methodically executed steps, i.e., tokenization, embedding initialization, bidirectional GRU encoding, context vector concatenation, and teacher-forcing-based decoding, the Bi-GRU architecture effectively mitigates the unidirectional limitations of conventional GRU models, resulting in robust summarization performance that leverages both forward and backward contextual information. The Bi-GRU thus emerges as a highly effective and computationally efficient architecture, optimally suited to the challenges presented by UATS tasks.

4.2.5 Encoder-Decoder Models with Attention

Although the previously discussed encoder-decoder architectures, namely LSTM, GRU, Bi-LSTM, and Bi-GRUhave shown considerable efficacy, they are inherently constrained

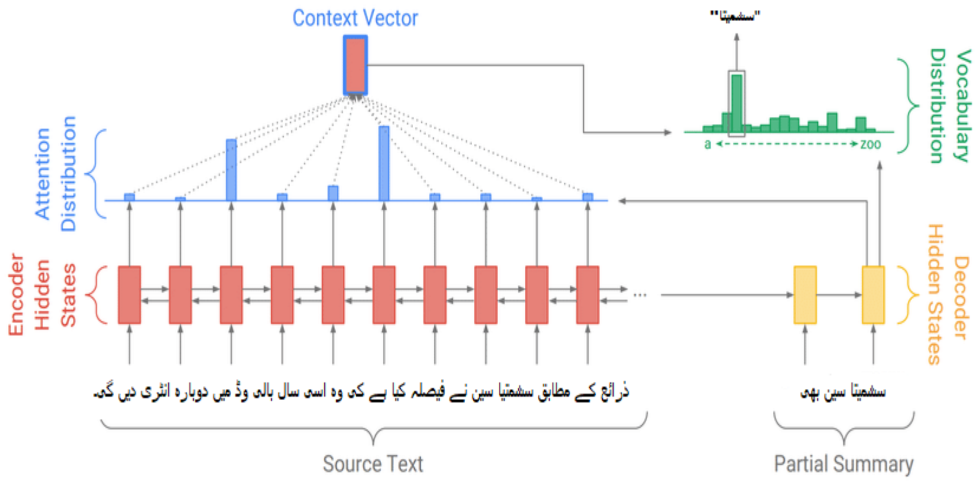

Figure 4.3: RNN based Encoder-Decoder Architecture with Attention

by a fundamental limitation: their restricted ability to capture the full context of long input sequences [64]. As illustrated in Figure 4.3, this shortcoming stems from using a fixed-length context vector. In conventional encoder-decoder frameworks, the encoder is tasked with compressing the entire input sequence into a single vector of fixed dimensionality, regardless of the sequence length. This compression often leads to a loss of critical contextual information, particularly in longer sequences, thereby hindering the generation of comprehensive and contextually accurate summaries. When dealing with extensive textual content, this fixed-length representation often results in substantial information loss, as it becomes increasingly challenging for a fixed-size context vector to adequately encapsulate and retain all relevant contextual details from longer sequences. Consequently, the decoder component may fail to access crucial semantic information required for generating accurate, coherent, and meaningful summaries.

4.2.5.1 Architecture of Encoder-Decoder Models with Attention

To overcome this intrinsic constraint, the attention mechanism was introduced [64], significantly improving the capabilities of encoder-decoder architectures, particularly in handling longer and more contextually complex input sequences. The central objective of the attention mechanism is to dynamically produce context vectors of variable lengths rather than relying solely on fixed-size representations. This adaptive approach ensures that critical information, irrespective of its position within the input sequence, is effectively captured, preserved, and selectively utilized during decoding. Consequently, attention mechanisms enhance the accuracy, coherence, and overall semantic richness of the generated summaries.

Practically, the attention mechanism is incorporated into the encoder-decoder architecture as an auxiliary layer positioned directly after the encoder's hidden layers. This attention layer dynamically evaluates the encoders output representations and assigns relevance-based weights to each input token, conditioned on the current decoding step. By doing so, it enables the decoder to focus selectively on different parts of the input sequence when generating each output word, thereby enhancing the models capacity to capture detailed contextual dependencies and improving the quality of the generated summaries. These calculated attention weights quantitatively determine the importance of individual input words, guiding the decoder's focus toward the most contextually relevant parts of the source text. Specifically, words with higher attention weights significantly influence the generation of the current output token, while words with lower weights are effectively de-emphasized, ensuring that the decoder selectively utilizes only pertinent and meaningful information.

Within this study, a particular variant of attention mechanism known as local attention was employed across all encoder-decoder architectures (LSTM, GRU, Bi-LSTM, and Bi-GRU). Local attention mechanisms are characterized by their selective focus on a smaller subset of input tokens, rather than simultaneously considering every word within the entire input sequence. By limiting attention to a defined local window of words around a given position, local attention significantly reduces computational complexity, thereby ensuring computational efficiency and resource optimization. Additionally, local attention focuses exclusively on specific segments within the input sequence deemed contextually relevant at each decoding timestep, thus eliminating potential distractions from less relevant portions of text.

The decision to utilize local attention specifically was driven by its balance between performance and computational affordability. Given the extensive size and linguistic complexity of the Urdu textual dataset, employing a computationally less intensive attention strategy was critical for maintaining manageable resource requirements without compromising performance quality. By selectively narrowing the decoders attention scope, local attention not only significantly reduced computational demands but also improved the semantic precision of generated summaries by ensuring the decoder concentrated specifically on pertinent segments of text.

The practical implementation and training procedure for all the encoder-decoder architectures equipped with the attention mechanism closely adhered to the previously outlined methodological structure (see Sections 4.2.1, 4.2.3, 4.2.2, and 4.2.4). Initially, Urdu textual data was carefully tokenized into discrete tokens, indexed through a structured dictionary, and transformed into numerical embeddings via a pretrained Urdu embedding layer. This embedding layer, leveraging pretrained embeddings derived from approximately 140 million Urdu tokens [60], was consistently frozen to preserve semantic integrity. Subsequently, these embeddings were sequentially processed by encoder components, be it LSTM, GRU, Bi-LSTM, or Bi-GRU, generating hidden state vectors encapsulating essential contextual information from the input text.

Next, instead of passing only the final hidden states as fixed-length context vectors, the attention layer dynamically computed attention weights for encoder hidden states. At each decoding step, these attention weights identified and highlighted specific subsets of words within the input sequence, forming an adaptive, variable-length context vector that effectively guided the decoders generation of each output token. Consistent with previously applied training procedures, the “teacher-forcing" strategy was again employed to reinforce accurate, coherent, and fluent summarization patterns during decoder training.

All RNN-based encoder-decoder models incorporating attention mechanisms shared a consistent structural architecture, as illustrated in figure 4.4. This unified representation clearly depicts the sequence processing workflow, emphasizing embedding, encoding, attention computation, adaptive context generation, and decoding. Such structural consistency facilitated fair comparative evaluation among various attention-equipped base-

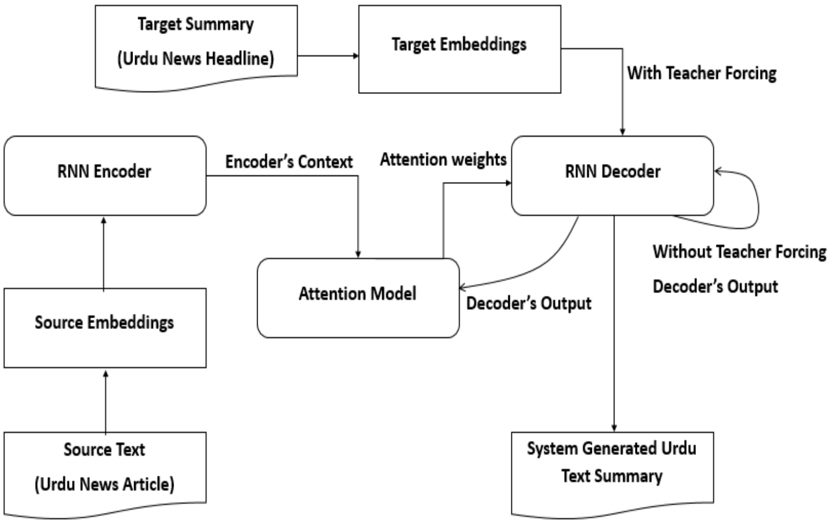

Figure 4.4: Proposed Methodology Using Encoder-Decoder Models with Attention

line models.

Furthermore, the baseline deep-learning models selected and evaluated in this study, i.e., LSTM, Bi-LSTM, GRU, and Bi-GRU equipped with attention mechanisms, were deliberately chosen based on their established effectiveness and extensive usage in text summarization tasks across numerous languages and linguistic domains. Incorporating a diverse range of these widely adopted baseline models enables the creation of a comprehensive benchmarking framework, against which the performance of more advanced, novel, or specialized summarization models can be systematically and reliably assessed. Establishing such robust benchmarks is essential for advancing research within the domain of UATS and facilitates future comparative analyses and methodological improvements.

By systematically employing and meticulously executing these detailed methodological steps, i.e., tokenization, embedding, encoder hidden state generation, attention computation, variable-length context vector creation, and teacher-forcing decoding, the attention-based encoder-decoder architectures significantly overcome the limitations associated with fixed-length context vectors, enabling more accurate, semantically coherent, and contextually nuanced ATS, particularly tailored to the linguistic challenges presented by the Urdu language.

Table 4.1: Summary Comparison of Baseline Deep Learning Models

Model	**Direction-ality**	**Attention**	**Pretrained Embedding**	**Params (Approx.)**
LSTM	Unidirectional	No	Yes (Frozen)	Medium
GRU	Unidirectional	No	Yes (Frozen)	Low
Bi-LSTM	Bidirectional	No	Yes (Frozen)	High
Bi-GRU	Bidirectional	No	Yes (Frozen)	Medium
LSTM + Attention	Unidirectional	Yes (Local)	Yes (Frozen)	Medium
Bi-LSTM + Attention	Bidirectional	Yes (Local)	Yes (Frozen)	High
GRU + Attention	Unidirectional	Yes (Local)	Yes (Frozen)	Medium
Bi-GRU + Attention	Bidirectional	Yes (Local)	Yes (Frozen)	High

To illustrate a structured comparison among the baseline deep learning models implemented in this study, Table 4.1 presents a concise summary of their core architectural attributes. The table categorizes each model based on its directionality (unidirectional vs. bidirectional), use of attention mechanisms, integration of pretrained embeddings, and estimated model complexity in terms of trainable parameters.

This classification helps delineate the differences between context-limited models (e.g., LSTM, GRU) and context-aware architectures (e.g., Bi-LSTM, Bi-GRU, and attention-enhanced variants). By highlighting the relative computational cost and contextual capacity of each model, this comparison aids in understanding the trade-offs involved in selecting a model for the UATS task.

The final column, “Context Type", explicitly groups models based on their ability to model local or global context dependencies, which is a critical factor for abstractive summarization in morphologically rich languages such as Urdu.

4.2.6 Parameter Settings for Deep Learning Models

To ensure reproducibility and maintain consistency across all experiments, a standardized set of training and inference configurations was applied to all baseline deep learning models as well as transformer-based architectures.

Table 4.2 provides a consolidated overview of these hyperparameter settings and architectural strategies. This table enhances the clarity of the experimental setup and allows Chapter 5 to remain focused exclusively on performance evaluation and analysis.

For a deeper technical elaboration on the implementation details and model-specific fine-tuning, readers are referred to Section 5.2.2 in Chapter 5.

All deep learning models, i.e., LSTM, GRU, Bi-LSTM, and Bi-GRU, both with and without attention mechanisms, were trained using a consistent optimization strategy. Specifically, a fixed learning rate of 0.001 was employed alongside a dropout rate of 0.5 to mitigate overfitting. The models were trained with a batch size of 32 for 7 epochs, ensuring uniform training conditions for fair performance comparison. All these models used a frozen 300-dimensional pretrained Urdu word embedding layer trained on 140 million tokens, as explained in Section 5.2.2. The optimizer used across all RNN models was Adam, and gradient clipping with a threshold of 1.0 was also employed for training stability.

To evaluate the computational feasibility of each model architecture, training time and GPU resource consumption were recorded. All deep learning models (LSTM, GRU, Bi-LSTM, Bi-GRU) completed training in approximately 6 to 8 GPU hours per model on the NVIDIA Tesla T4 with 16 GB memory.

4.3 Transformer-Based Models

Transformer-based encoder-decoder architectures have emerged as powerful tools for sequence-to-sequence tasks such as abstractive summarization. Unlike traditional RNN-based models, these architectures capture long-range dependencies through self-attention mechanisms and parallelized training. In this study, the transformer variant, i.e., Bidirectional Auto-Regressive Transformers (BART) is evaluated for its effectiveness in generating fluent and contextually accurate summaries in Urdu. These models are trained or fine-tuned using supervised data, distinguishing them from prompt-driven LLMs discussed separately in Section 4.4.

4.3.1 Bidirectional Auto-Regressive Transformers

The BART, developed by Facebook AI, is a highly versatile model tailored for a broad spectrum of natural language processing tasks. Functioning as a denoising autoencoder, BART is designed to reconstruct coherent outputs from deliberately corrupted input sequences, thereby learning to capture deep semantic and syntactic structures. Its architecture is pre-trained on extensive and heterogeneous corpora comprising both natural language and code-based text [8], enabling it to develop a comprehensive understanding of linguistic patterns across domains and languages. This robust pre-training confers exceptional adaptability, allowing BART to achieve state-of-the-art performance in tasks such as question answering, abstractive text summarization, and machine translation.

4.3.1.1 Architecture and Parameter Setting for Bidirectional Auto-Regressive Transformers

At its core, BART employs the Transformer architecture, an advanced neural network framework that has significantly reshaped contemporary NLP approaches. Transformers leverage a self-attention mechanism that allows the model to capture intricate dependencies and contextual relationships across entire input sequences, irrespective of their length. This architecture eliminates the sequential processing constraints found in recurrent neural networks, enabling more efficient parallelization and superior performance in modeling long-range dependencies. The Transformer architecture's self-attention layers can simultaneously consider all positions within a sequence, thus allowing the model to dynamically focus on important information irrespective of its position. This characteristic makes the Transformer architecture exceptionally powerful for understanding context, especially in NLP tasks that require careful semantic analysis, such as machine translation and text summarization.

BART extends the core architecture of Transformer models by introducing a distinctive configuration of its encoder and decoder components. Unlike conventional LLMs that adopt either fully autoregressive or fully bidirectional architectures, BART integrates a bidirectional encoder, capable of attending to all positions within the input simultaneously, with a left-to-right autoregressive decoder, which generates output tokens sequentially. This hybrid design enables BART to capture comprehensive contextual information during encoding while maintaining a controlled, step-wise generation process during decoding, thereby enhancing its performance across a wide range of generative and discriminative NLP tasks. The bidirectional encoder simultaneously attends to both past (left context) and future (right context) within sentences, enabling it to form a comprehensive representation of linguistic context. This bidirectional contextual understanding greatly enhances the model's comprehension of complex sentence structures, semantics, and syntactic relationships. Subsequently, the autoregressive decoder generates the output sequence one token at a time, strictly from left to right, ensuring the production of coherent, contextually appropriate, and grammatically accurate textual outputs.

The distinctive design choice of combining a bidirectional encoder with an autoregressive decoder allows BART to leverage the advantages of both bidirectional and unidirectional processing. By capturing context from both directions during encoding, BART effectively understands the nuanced relationships among words within sentences, thus significantly enhancing its capabilities for tasks involving intricate semantic dependencies, such as ATS and translation. On the other hand, the left-to-right decoding process ensures that outputs remain linguistically coherent and syntactically accurate, which is vital for generating natural-sounding language.

Given these architectural strengths, we specifically selected BART as a candidate model for our UATS task. Our primary motivation was BART's well-documented effectiveness at leveraging large-scale pretraining datasets encompassing diverse linguistic styles and structures. Its broad, pre-trained knowledge base positions BART as particularly adept at generalizing to new linguistic domains and tasks. Prior research has extensively demonstrated BART's state-of-the-art performance in English text summarization tasks, showcasing its superior summarization quality compared to other prominent models such as BERT, RoBERTa, and GPT-3.5 [8], [7]. For instance, BART has consistently outperformed BERT and RoBERTa models in various language generation and translation benchmarks [8], and has similarly shown superior performance compared to GPT-3.5 on text summarization benchmarks [7]. These promising empirical results strongly motivated our hypothesis that fine-tuning BART specifically for UATS would produce highly accurate, contextually coherent, and semantically precise summaries tailored explicitly to the linguistic nuances of the Urdu language.

To adapt BART effectively to the UATS task, we systematically conducted a fine-tuning procedure using our proposed UATS-23 corpus. In this fine-tuning phase, BART leveraged its pre-trained linguistic and semantic knowledge obtained from the extensive English-based training to quickly adapt and generalize to the unique structures and semantic characteristics of the Urdu language. The fine-tuning approach enabled BART to bridge linguistic gaps effectively, recognizing and understanding Urdu-specific idiomatic expressions, semantic subtleties, and sentence constructions without needing extensive language-specific pre-training from scratch.

The fine-tuning process involved initially pre-processing the Urdu news articles by tokenizing the text. Subsequently, tokenized inputs were provided to BART's bidirectional encoder, allowing it to comprehensively analyze the contextual relationships among words within each article. The encoders comprehensive contextual representation was then decoded sequentially by BARTs left-to-right autoregressive decoder, producing coherent, concise, and contextually accurate Urdu summaries. Throughout fine-tuning, standard optimization strategies, including learning rate scheduling and gradient clipping, were employed to ensure stable, efficient, and effective model adaptation.

Ultimately, BART's sophisticated architecture, robust pre-training, and flexible fine-tuning capabilities provided an ideal combination for achieving strong performance in UATS. By integrating BART into our study, we aimed to establish a high-quality baseline summarization model against which more specialized and advanced Urdu summarization models could subsequently be evaluated. Furthermore, by assessing BART's performance in the relatively understudied domain of Urdu language summarization, our research contributes significantly to understanding how well advanced, pretrained transformer-based models generalize to linguistically and structurally distinct languages, laying a foundation for future exploration and innovation within Urdu NLP research.

The BART model was fine-tuned on 100 samples from the UATS-23 corpus using a maximum input length of 1024 tokens and a summary length of 120 tokens. Beam search (width = 2) was applied for decoding. Optimization used AdamW with a peak learning rate of 10^{-4} and weight decay of 10^{-2}, followed by a learning rate decay schedule to ensure stable convergence and reduce overfitting. The fine-tuning strategy followed the configuration below:

- **Epochs:** 4
- **Learning Rate:** 0.00001 with linear warmup schedule
- **Frozen Layers:** Embedding layer and first 3 encoder layers were frozen to preserve pretrained language understanding
- **Batch Size:** 8
- **Loss:** Cross-entropy with label smoothing (0.1)

BART required 812 GPU hours for fine-tuning over 4 epochs using a batch size of 8.

4.4 Large Language Models

In recent years, LLMs have significantly transformed the landscape of natural language processing by exhibiting remarkable zero-shot and few-shot generalization across diverse tasks, including abstractive summarization. These models are predominantly trained on vast, multilingual corpora and are built upon transformer-based, decoder-only architectures. This design enables them to generate fluent, coherent, and semantically rich text, even with minimal or no task-specific supervision.

Unlike traditional deep learning models that require extensive supervised training, LLMs such as GPT-3.5 and DeepSeek-R1 operate through prompting strategies, allowing them to generalize well even without direct training on Urdu summarization data.

Their ability to transfer knowledge across languages and tasks makes them particularly attractive for low-resource settings like Urdu.

In this section, we evaluate and compare multiple LLMs on the UATS task using structured prompts and consistent decoding parameters. We also highlight the influence of pretraining corpora and domain bias on their performance. The results of these models are later juxtaposed with deep learning and transformer-based approaches in Chapter 5.

4.4.1 Generative Pre-trained Transformer (GPT-3.5)

The LLM GPT-3.5 was developed by OpenAI and represents a significant advancement in NLP technology. Specifically, GPT-3.5 is a decoder-only transformer-based model comprising an enormous scale with approximately 175 billion trainable parameters, trained on an extensive and diverse dataset encompassing text and code data from numerous digital sources [9]. Due to the breadth and diversity of its training corpus, GPT-3.5 demonstrates exceptional versatility, allowing it to effectively perform a wide range of advanced language understanding and generation tasks. These capabilities encompass, but are not limited to, text generation, question answering, machine translation, sentiment analysis, abstractive summarization, and the production of creative written content [65].

GPT-3.5 fundamentally leverages the transformer architecture, a neural network design that has substantially advanced NLP by overcoming limitations commonly associated with traditional sequential processing architectures, such as RNNs and convolutional neural networks (CNNs). At the core of the transformers success lies its powerful self-attention mechanism, which facilitates dynamic and adaptive contextual understanding by attending simultaneously to different parts of the input sequence. GPT-3.5 advances the conventional self-attention mechanism by incorporating a sparse attention mechanism [9], which optimizes the models ability to process long input sequences more efficiently. This enhancement selectively focuses attention on a subset of relevant tokens, thereby reducing computational complexity while still effectively capturing long-range dependencies and contextual relationships across extensive textual data. By selectively allocating attention weights to specific tokens within input sequences, GPT-3.5 can manage computational resources more effectively, thereby significantly enhancing its ability to understand nuanced linguistic contexts and complex sentence structures spanning longer texts.

This ability to efficiently process extensive sequences and complex language structures makes GPT-3.5 particularly well-suited to ATS tasks. In abstractive summarization, effectively capturing semantic context, syntactic nuances, and long-term relationships within texts is essential for generating coherent, concise, and semantically accurate summaries. GPT-3.5s sparse attention mechanism and substantial scale ensure that it can accurately grasp these complexities, leading to robust summarization performance. Given the models demonstrated effectiveness across multiple NLP tasks and its exceptional generative capabilities, it was hypothesized that GPT-3.5 would be similarly effective when fine-tuned specifically for summarizing Urdu text, despite linguistic differences between Urdu and English.

For the practical implementation of our study, we specifically utilized OpenAI’s *text-curie-001* variant of GPT-3.5, which belongs to OpenAIs “Instruct" series. This specific variant of GPT-3.5 was chosen primarily due to its optimized capabilities for generating text-based responses through fine-tuning on targeted tasks, making it especially suitable for summarization. The Instruct series models are trained to closely follow instructions provided through prompt engineering, ensuring that generated outputs closely adhere to user-defined summarization requirements. This precise adherence to prompt instructions makes text-curie-001 particularly effective for generating contextually accurate and linguistically fluent summaries, aligning closely with the objective of UATS.

To systematically leverage GPT-3.5s potential, our research deals with a detailed fine-tuning procedure using the proposed UATS-23 corpus developed specifically for this research. This corpus, characterized by its extensive size, linguistic diversity, and broad topic coverage, provided an ideal training dataset to effectively fine-tune the GPT-3.5 model. During fine-tuning, the model adapted its existing generalized linguistic understanding, acquired through pretraining on extensive multilingual textual data, to specifically accommodate the linguistic features, structures, and semantics inherent within the Urdu language.

The fine-tuning methodology followed a structured approach, beginning with careful pre-processing of the UATS-23 corpus. The Urdu textual content was first tokenized into discrete tokens (words), subsequently converting these tokens into numerical embeddings. The pre-trained GPT-3.5 embedding mechanism efficiently leveraged its extensive pretrained vocabulary to map tokens effectively, accommodating the unique lexical characteristics of the Urdu language. Subsequently, these embeddings were systematically processed by the GPT-3.5 model's decoder architecture, which generated summaries through sequential decoding from left to right, leveraging the powerful attention mechanism to dynamically identify and utilize relevant context from input sequences.

During the fine-tuning process, a range of hyperparameters and training strategies were optimized to ensure stable and effective training. For instance, the temperature parameter, controlling the randomness and creativity of outputs, was carefully set to balance fluency and adherence to source content. Similarly, the maximum token length was deliberately capped to maintain summary conciseness and coherence. Additionally, parameters controlling the diversity of output, i.e., top-p sampling and frequency penalty, were finely tuned to minimize repetitive outputs and improve the readability of generated summaries.

4.4.2 DeepSeek-R1

The LLM DeepSeek-R1[1], developed by an open-source initiative, represents a notable advancement in natural language processing technology, demonstrating strong capabilities similar to models such as GPT-3.5 and BART. Specifically, DeepSeek-R1 is a transformer-based language model composed of 67 billion parameters, trained extensively on a diverse dataset consisting of approximately two trillion tokens, encompassing both text and code from a wide range of digital sources. Due to the diversity and large-scale composition of its pre-training corpus, DeepSeek-R1 has demonstrated sub-

[1] https://deepseek.ai

stantial effectiveness across a broad range of natural language processing tasks. These include question answering, abstractive summarization, machine translation, code generation and assistance, as well as general language modeling and generation tasks.

The DeepSeek-R1 model fundamentally utilizes the Transformer architecture, a neural network framework that has revolutionized NLP tasks by effectively addressing the limitations commonly associated with traditional neural network architectures like RNNs or CNNs. A fundamental strength of the Transformer architecture lies in its self-attention mechanism, which enables the model to dynamically assign varying levels of attention to different segments of the input sequence. This adaptive attention allocation allows the model to effectively capture both local and global dependencies, thereby enhancing its capability to understand and generate contextually coherent and semantically accurate text. This enables it to capture complex contextual relationships across long-range dependencies effectively. DeepSeek-R1 enhances this capability further by employing architectural optimizations and computational techniques that efficiently utilize computational resources, enabling improved processing of lengthy textual inputs and greater model scalability, thereby resulting in superior contextual understanding.

This sophisticated attention mechanism, coupled with its expansive training and robust architecture, positions DeepSeek-R1 as an ideal candidate for natural language processing tasks, including abstractive summarization. ATS tasks require models to generate coherent, fluent, and semantically accurate summaries, tasks at which DeepSeek-R1 excels due to its strong context-awareness and generative capabilities. Given these advanced properties and proven performance across diverse language processing benchmarks, we hypothesized that DeepSeek-R1 would similarly exhibit strong summarization capabilities when applied to Urdu-language summarization, despite the linguistic differences and lower resource availability.

In this research, the DeepSeek-R1 model was systematically integrated and fine-tuned specifically for the UATS task. Unlike conventional fine-tuning approaches, DeepSeek-R1 was employed using a “zero-shot" inference strategy [66], which leverages its pretrained linguistic knowledge without explicit fine-tuning on Urdu datasets. In zero-shot learning scenarios, the model relies heavily on its extensive pretrained knowledge to generalize effectively to languages and tasks it has not been explicitly trained on, making this approach particularly advantageous for resource-constrained languages such as Urdu.

To incorporate DeepSeek-R1 into our research methodology, we used structured prompt engineering techniques to clearly instruct the model about the summarization task requirements. This involved designing specific prompts for generating abstractive summaries that included explicit task instructions, such as clearly stating the desired summary length, specifying fluency and contextual requirements, and providing illustrative examples directly in the inference prompt. This carefully crafted prompting approach allowed DeepSeek-R1 to effectively utilize its generalized understanding of linguistic semantics, syntax, and discourse patterns, facilitating accurate and coherent summary generation directly from input Urdu articles.

For practical implementation, each Urdu source article was initially tokenized into word tokens and converted into numerical representations using the standard embedding strategy integrated within the DeepSeek-R1 transformer model. The numerical embeddings, leveraging the pre-trained knowledge from DeepSeek-R1's vast multilingual vocabulary, facilitated effective mapping of Urdu lexical items, including words unique to the Urdu language, into meaningful numerical embeddings. These embeddings were then processed sequentially by DeepSeek-R1's decoder-only transformer architecture, wherein the self-attention mechanism dynamically identified relevant context and semantic relationships among the tokens, even without explicit Urdu-specific fine-tuning.

Furthermore, careful control of decoding parameters such as maximum token lengths, top-p sampling, frequency penalties, and repetition penalties was employed during inference to ensure summary outputs were both fluent and concise, mitigating typical generative issues like token repetition or semantic redundancy. Adjustments of these parameters were instrumental in fine-tuning the quality, diversity, coherence, and linguistic accuracy of summaries generated by DeepSeek-R1, despite the zero-shot inference paradigm.

The inclusion of DeepSeek-R1 in our study not only provides a valuable performance baseline but also offers insights into the feasibility and effectiveness of leveraging zero-shot inference techniques with large-scale pre-trained models for low-resource language tasks like Urdu ATS. Additionally, this implementation highlights the potential applicability of advanced pretrained language models in bridging resource gaps, setting a precedent for further research and development in Urdu NLP.

By systematically integrating DeepSeek-R1's powerful transformer architecture, adaptive self-attention mechanism, and its sophisticated zero-shot generative capabilities, this study establishes a foundational baseline for performance evaluation. This extended evaluation further strengthens the comparative framework of our study, allowing subsequent analyses to meaningfully assess and benchmark the performance of various advanced UATS techniques against the robust capabilities demonstrated by DeepSeek-R1. Ultimately, the introduction of the DeepSeek-R1 model into Urdu ATS research represents an important step towards developing accessible, robust, and efficient summarization solutions tailored specifically for Urdu language applications.

4.4.3 Parameter Settings for LLMs

For the UATS task, we employed the pre-trained GPT-3.5 model (text-curie-001) due to its cost-effectiveness and strong summarization capabilities, using default fine-tuning parameters. During evaluation, we set the temperature to 0 and $max_{t}okens$ to 100 to ensure concise, source-restricted summaries; top-p was fixed at 1 for deterministic output, and a frequency penalty of 0.5 was applied to reduce redundancy while preserving fluency. Additionally, we utilized the open-source DeepSeek-R1-LLM (67B parameters) in a zero-shot setting, leveraging its default configurations: temperature = 1.0, max tokens = 4096, top-p = 1.0, and frequency penalty = 0.0. These settings enabled DeepSeek-R1 to generate diverse, fluent, and contextually coherent summaries, making it a practical option for low-resource languages like Urdu without explicit fine-tuning.

Table 4.2: Training and Inference Configuration for All Models

Model	**Epochs**	**Batch Size**	**Embedding Type**	**Architecture Type**
LSTM	7	32	Urdu 300-d	RNN Encoder-Decoder
GRU	7	32	Urdu 300-d	RNN Encoder-Decoder
Bi-LSTM	7	32	Urdu 300-d	Bi-RNN
Bi-GRU	7	32	Urdu 300-d	Bi-RNN
LSTM + Attention	7	32	Urdu 300-d	RNN + Local Attention
Bi-LSTM + Attention	7	32	Urdu 300-d	Bi-RNN + Local Attention
GRU + Attention	7	32	Urdu 300-d	RNN + Local Attention
Bi-GRU + Attention	7	32	Urdu 300-d	Bi-RNN + Local Attention
BART (Fine-tuned)	4	8	BART-base	Transformer (EncoderDecoder)
GPT-3.5 (text_curie_001)	–	Prompt	Frozen LLM	Decoder-only
DeepSeek-R1 (Zero-shot)	–	Prompt	Frozen LLM	Decoder-only

4.5 Challenges in Urdu ATS Implementation

Several technical and linguistic challenges were encountered during the implementation of Urdu ATS models, underscoring the complexity of working with a morphologically rich and script-variant language like Urdu.

- **Embedding and Vocabulary Gaps:** Urdu word embeddings often suffer from limited coverage of named entities, loanwords, and inflectional variants. This affected semantic learning, especially in domain-specific summaries.
- **Tokenization and Sentence Boundary Issues:** Due to the space-insensitive nature of the Nastaliq script, conventional tokenizers underperformed. Sentence boundary detection was further complicated by inconsistent punctuation usage in online news sources.
- **Evaluation Alignment Challenges:** Headlines in news articles were often stylis-

tic or rhetorical, rather than true summaries. This posed a challenge in evaluating abstractive summaries using metrics like ROUGE, which expect lexical alignment with reference summaries.

- **Transformer Fine-Tuning Difficulties:** Fine-tuning BART on Urdu text occasionally led to unstable training loss and overfitting due to lack of subword coverage or shallow understanding of Urdu syntax in pretrained tokenizers.

Addressing these challenges required several pre-processing refinements and decoding stabilizations as discussed throughout this chapter. These limitations are also considered when interpreting results in Chapter 5.

4.6 Chapter Summary

This chapter presented a comprehensive overview of all methodological foundations applied in this research for UATS. It began by categorizing neural models into context-limited and context-aware groups, introducing architectures such as LSTM, GRU, Bi-LSTM, Bi-GRU, and their attention-enhanced extensions. Transformer-based model, i.e., BART and LLMS, including GPT-3.5 and DeepSeek-R1, were also introduced, with detailed descriptions of their architectures, training strategies, and pretraining biases.

A unified training and inference configuration was proposed and tabulated to ensure consistent comparisons across model families. Fine-tuning protocols and prompting strategies were explicitly detailed, including frozen layer selections, learning rate schedules, and decoding schemes. Computational feasibility was addressed by reporting GPU runtime, memory usage, and training convergence across all model categories.

Crucially, this chapter also acknowledged practical challenges in implementing UATS, including limitations in Urdu embeddings, sentence boundary segmentation, evaluation misalignments between headlines and human abstracts, and instability during transformer fine-tuning. These limitations were mitigated through targeted engineering adjustments and are further reflected upon in the experimental analysis presented in Chapter 5.

Collectively, the methodology presented in this chapter establishes a reproducible and extensible foundation for UATS research, against which the performance results in the next chapter are systematically evaluated.

Chapter 5
Evaluation of Proposed Methods on UATS-23 Corpus

5.1 Introduction

This chapter provides a comprehensive evaluation and comparative analysis of state-of-the-art deep learning models and LLMs applied to the UATS-23 corpus for UATS. A robust experimental framework was established, encompassing multiple baseline architectures including LSTM, Bi-LSTM, GRU, and their respective variants augmented with attention mechanisms. In addition to these recurrent models, advanced transformer-based architectures such as BART, GPT-3.5, and DeepSeek-R1 were also rigorously evaluated to benchmark their effectiveness on the proposed corpus.

5.2 Experimental Setup

This section presents the complete experimental setup employed for training, fine-tuning, and evaluating all models applied to the UATS task. The experiments are conducted using the UATS-23 corpus, comprising over two million article-summary pairs, as detailed in Chapter 3.

To ensure a comprehensive assessment, a wide spectrum of models was utilized, encompassing both context-limited and context-aware deep learning baselines, including LSTM, GRU, Bi-LSTM, and Bi-GRU, with and without attention mechanisms. Additionally, Transformer-based models such as BART and cutting-edge LLMs like GPT-3.5 and DeepSeek-R1 were integrated into the evaluation framework.

The experimental design also incorporates a unified training and inference configuration, ensuring fair comparison across all architectures. All deep learning models are trained under standardized hyperparameter settings, while transformer and LLM-based models are either fine-tuned or evaluated using zero-shot or prompt-based methods.

Evaluation is performed using widely adopted automatic metrics, including ROUGE-1, ROUGE-2, and ROUGE-L. These are computed using F_1 scores to assess lexical overlap and semantic coherence between model-generated and reference summaries. The evaluation methodology and metric computation approach are described in subsequent subsections.

5.2.1 Dataset

For the experiments presented in this study, the entire UATS-23 corpus was utilized. This corpus comprises 2,067,784 ($\simeq$ 2.067 million) Urdu news articles paired with their corresponding human-written summaries, making it the largest publicly available dataset for the UATS task to date (see Section 3.2 for construction and preprocessing details).

The dataset was constructed through systematic crawling, cleaning, and normalization processes, followed by structured mapping of source articles to their respective headlines, which serve as reference summaries. To ensure robust training and fair evaluation, the dataset was split using a stratified Train/Validation/Test strategy, with 2,027,784 instances allocated for training, and 20,000 instances each for validation and testing. This splitting strategy was selected to preserve topical diversity and maintain representational balance across different domains such as politics, sports, business, and entertainment.

All models evaluated in this study, deep learning, transformer-based, and LLMs, were trained and/or tested on this standardized corpus. The large-scale, linguistic diversity, and domain coverage of UATS-23 make it an appropriate benchmark for developing and evaluating ATS systems in low-resource language settings.

5.2.2 Techniques

The techniques applied to the UATS-23 corpus encompass a broad range of architectures, including LSTM-based encoder-decoder models (Section 4.2.1), Bi-LSTM-based encoder-decoder models (Section 4.2.3), GRU-based encoder-decoder models (Section 4.2.2), and Bi-GRU-based encoder-decoder models (Section 4.2.4). Additionally, attention-enhanced variantsnamely, LSTM and GRU models with integrated attention mechanismsare also incorporated (Section 4.2.5). To benchmark performance against advanced architectures, transformer-based models such as BART (Section 4.3.1), GPT-3.5 (Section 4.4.1), and DeepSeek-R1 (Section 4.4.2) are included in the evaluation. The subsequent sections detail the parameter configurations used for training these state-of-the-art baseline deep learning and transformer-based models.

For our experiments, we utilized computational resources provided by the Google VertexAI Cloud Platform[1], which offered us access to a single NVIDIA Tesla T4 GPU, complemented by 60 GB RAM and 16 virtual CPUs (vCPUs). This specific hardware configuration was selected considering computational efficiency, accessibility, and cost constraints. The NVIDIA Tesla T4 GPU, known for its strong performance in deep learning tasks, provided adequate computational capabilities, enabling us to effectively train and fine-tune our deep-learning and transformer models for the Urdu UATS task. All experimental implementations were executed on a Python 3.7 environment, leveraging the PyTorch deep learning framework[2], chosen due to its versatility, ease of implementation, and widespread use in NLP research.

The experimental design included several systematic steps to effectively manage and optimize computational resources. Initially, the complete UATS-23 containing approximately 2.067 million Urdu article-summary pairs was partitioned using the well-established Train/Test/Validation paradigm. Specifically, the data split was carried out as follows: approximately 2,027,784 instances were allocated for training, ensuring ample data for deep learning model training and generalization, while 20,000 instances each were reserved explicitly for testing and validation purposes. These separate test and validation sets provided a rigorous framework for evaluating model performance, ensuring reliable and unbiased comparison across all implemented architectures.

However, transformer-based models, including BART, GPT-3.5, and DeepSeek-R1, are inherently more computationally demanding, making it infeasible to fine-tune them extensively within the available hardware resources. Consequently, fine-tuning trans-

[1] https://cloud.google.com/ Last Visited: 25-Sep-2024

[2] `https://pytorch.org/` Last Visited: 25-Sep-2024

former models required limiting our fine-tuning dataset to only 100 instances. Despite this constraint, we ensured a comprehensive and fair performance assessment by conducting rigorous testing on a larger dataset of 20,000 instances. This balanced experimental setup ensured that our comparisons across deep learning and transformer-based models remained consistent, fair, and methodologically sound.

Each of the baseline deep learning models, LSTM, GRU, Bi-LSTM, Bi-GRU, and their variants incorporating attention mechanisms, was implemented using a consistent parameter optimization approach. All models shared a set of optimized hyperparameters carefully chosen based on prior research and preliminary experimentation. Specifically, the learning rate was fixed at 0.001, a well-established default in NLP research for balancing convergence speed and training stability. Additionally, a dropout rate of 0.5 was employed to mitigate overfitting risks, ensuring robust generalization capabilities. Training epochs were optimally determined and subsequently fixed at 7, based on empirical observations from preliminary experiments. To manage memory constraints encountered on the single T4 GPU, the batch size was set at 32, providing a practical balance between computational efficiency and model performance. Furthermore, due to the intrinsic nature of recurrent architectures experiencing exploding gradients, a gradient clipping threshold of 1 was uniformly applied across all RNN-based models to stabilize training processes and prevent gradient instability.

Additionally, we utilized the teacher-forcing approach [67] with a ratio of 70%, meaning that during the training phase, 70% of the input tokens fed into the decoder were ground-truth tokens rather than model-generated predictions. This approach substantially improved learning stability and helped models better capture accurate language patterns during summarization, thus enhancing overall performance.

The BART model was fine-tuned on a subset of 100 instances from the UATS-23 training corpus, adhering to the architectural specifications outlined in [8]. The input documents were truncated or padded to a maximum sequence length of 1024 tokens, while the corresponding summaries were limited to 120 tokens. For decoding, a beam search with a beam width of 2 was employed, selecting the most optimal summary from the top two generated candidates. Optimization was carried out using the AdamW optimizer [68], configured with a peak learning rate of 10^{-4} and a weight decay of 10^{-2}. To facilitate stable convergence and mitigate overfitting, the learning rate was progressively reduced post-training using a decay factor. This learning rate schedule ensures smoother optimization by allowing finer parameter updates as training proceeds.

Moreover, for our experiments, we used a pre-trained model $text - curie - 001$ to fine-tune it on the Urdu ATS task. The reasons for selecting this model of GPT-3.5 are its low cost and effectiveness for the ATS task. For fine-tuning we used the default parameters of GPT-3.5. For evaluation, we set the $temperature = 0$ so the model becomes more restricted to the contents of the source text. We set $max\_tokens = 100$ to ensure that the generated summary remains concise and within a specific length limit. The top-p sampling parameter controls the diversity of the generated output. We set it to 1, which means the model only considers the most likely token at each step. This setting can make the generated output more focused and coherent. The $frequency\_penalty$ parameter discourages the model from repeating the same phrases or tokens too frequently. A value of 0.5 strikes a balance between encouraging variety and maintaining coherence in the generated text.

Moreover, for our experiments, we utilized the pre-trained DeepSeek-R1 model, specifically the DeepSeek-R1-LLM variant with 67 billion parameters, for the UATS task. The DeepSeek-R1 model was selected primarily due to its open-source accessibility, efficiency, and demonstrated effectiveness in language generation tasks without the need for explicit fine-tuning. During inference, we applied DeepSeek-R1s default parameter configurations: the *temperature* was set to 1.0, enabling the model to balance creativity and coherence within the generated summaries; the *maximum token length* was set to 4096, ensuring sufficient capacity for comprehensive coverage of longer textual inputs; the *top-p sampling* (nucleus sampling) parameter was set to 1.0, allowing the model to consider the full probability distribution and thus producing diverse yet contextually appropriate outputs; and the *frequency penalty* was maintained at 0.0, permitting repetition of tokens when contextually necessary to emphasize key points. Collectively, these default parameters enabled the DeepSeek-R1 model to effectively generate fluent, detailed, and semantically coherent Urdu summaries, underscoring its practical suitability for low-resource summarization tasks such as UATS.

By systematically employing these rigorous, consistent, and transparent experimental protocols, our study ensured methodological reliability, accuracy, and fairness in performance evaluation. These detailed experimental configurations and strategic parameter optimizations provided an essential foundation for meaningful performance comparisons among baseline deep learning models and transformer-based architectures. Consequently, these careful methodological considerations significantly enhance the research's validity, establishing robust and reliable benchmarks for future researchers aiming to further advance UATS methodologies.

Table 5.1 provides a detailed overview of all experimental configurations employed in this study. Each configuration reflects a deliberately chosen set of hyperparameters and strategies aimed at thoroughly evaluating the performance of baseline deep learning models, transformer-based architectures, and LLMs on the UATS-23 corpus.

To ensure a comprehensive and methodologically sound evaluation, multiple combinations were tested across critical parameters for recurrent models. Specifically, epoch values were varied from 10 to 15, dropout rates ranged from 0.2 to 0.5, and network depth (in terms of hidden layers) was evaluated between 1 and 2 layers. The batch sizes tested included 32 and 64, selected based on the trade-off between model convergence and the memory constraints of the available GPU resources. A fixed learning rate of 0.001 was adopted for all RNN-based architectures, consistent with best practices in prior NLP literature. Additionally, the Adam optimizer was used in conjunction with gradient clipping to mitigate the risk of exploding gradients, which is a known challenge in training RNNs.

Despite access to limited computing infrastructure, specifically, a single Tesla T4 GPU provisioned through Google VertexAIeach experimental run for the deep learning models consumed approximately 6 to 8 hours per epoch. This time estimate includes checkpointing, logging, and validation routines. Across the RNN and attention-based

Table 5.1: Extended Experimental Configuration Summary for All Models

Model	Training Strategy	Epochs Tested	Batch Sizes	Learning Rates	Dropout Range	Hidden Layers	Max Tokens	Decoding Method	Eval Time/ Epoch	Total Training Time	Optimizer / Other	Total No. of Exp.
LSTM	Supervised	10--15	64	0.001	0.2--0.5	1--2	–	Greedy	6--8 hrs	6--8 days	Adam + Clip	7
GRU	Supervised	10--15	64	0.001	0.2--0.5	1--2	–	Greedy	6--8 hrs	6--8 days	Adam + Clip	7
Bi-LSTM	Supervised	10--15	64	0.001	0.2--0.5	1--2	–	Greedy	6--8 hrs	6--8 days	Adam + Clip	7
Bi-GRU	Supervised	10--15	32	0.001	0.2--0.5	1--2	–	Greedy	6--8 hrs	6--8 days	Adam + Clip	7
LSTM + Attention	Supervised	10--15	32	0.001	0.2--0.5	1--2	–	Greedy	6--8 hrs	6--8 days	Adam + Clip	10
GRU + Attention	Supervised	10--15	64	0.001	0.2--0.5	1--2	–	Greedy	6--8 hrs	6--8 days	Adam + Clip	10
BART (Fine-tuned)	Fine-tuning	10	8	0.001	–	–	1024 / 120	Beam=2	4--6 hrs	1--2 days	AdamW, WD=0.01	3
GPT-3.5	Fine-tuning / Prompt-based	10	–	Default	–	–	1024 / 120	Beam=2	8--12 hrs	3--4 days	Paid API	2
DeepSeek-R1 (Prompted)	Prompt-based	–	–	Default	–	–	1024 / 120	Beam=2	–	3--4 days	Paid API	2

models, a total of 7 to 10 experiments were performed per architecture, resulting in an extensive training period of 6 to 8 days per model. These experiments were essential to determine the optimal configuration under constrained compute budgets while ensuring generalization and model robustness.

For transformer models such as BART, and LLMs like GPT-3.5 and DeepSeek-R1, resource-intensive full-scale fine-tuning was not feasible due to computational and API cost constraints. Hence, BART was fine-tuned on a limited sample (batch size of 8, for 10 epochs) using beam search decoding (beam size = 2), with a fixed input length of 1024 tokens and output summaries constrained to 120 tokens. GPT-3.5 and DeepSeek-R1 were accessed through their respective paid APIs and configured for inference using the same maximum token limits and beam search settings to maintain consistency across all evaluated models.

Overall, the parameter selection across architectures was guided by both empirical NLP standards and practical limitations. The comprehensive range of tested combinations and explicit tracking of all resource metrics underscore the rigor and reproducibility of this studys experimental methodology. The table thus serves not only as a technical blueprint for the experiments conducted but also as a valuable resource for future researchers aiming to benchmark or extend work in UATS under similar constraints.

The GRU with Attention model yielded the most effective performance across the UATS-23 corpus. As shown in Table 5.2, this configuration was achieved after systematic hyperparameter tuning. A batch size of 64 was chosen to leverage the memory-efficient gating mechanisms of GRUs, while maintaining a stable learning trajectory under GPU constraints. A learning rate of 0.001 provided an ideal balance between convergence speed and stability, consistent with best practices in deep sequence modeling. The Adam optimizer was selected for its adaptive learning capabilities, especially beneficial in noisy gradient landscapes typical of natural language. A dropout rate of 0.5 was found to be optimal for regularization, preventing overfitting in high-dimensional sequence data. Gradient clipping was applied to control exploding gradients, a common challenge in training recurrent networks.

Table 5.2: Optimal Hyperparameter Configuration for GRU + Attention Model

Parameter	Value
Model Architecture	GRU + Local Attention
Batch Size	64
Learning Rate	0.001
Optimizer	Adam
Dropout Rate	0.5
Gradient Clipping	Applied (threshold = 1.0)
Number of Hidden Layers	1
Number of Epochs	11

Furthermore, a single hidden layer was sufficient to model the abstract relationships in the Urdu language without over-complicating the architecture or increasing the risk of overfitting. Extending training to 11 epochs ensured model convergence while avoiding degradation from over-training. The attention mechanism, implemented as local attention, enabled the decoder to focus dynamically on semantically relevant portions of the input sequence, resulting in coherent and contextually accurate abstractive summaries.

5.2.3 Evaluation Methodology

The problem of UATS was approached as a supervised machine learning problem, following established conventions and rigorous standards in NLP research. Supervised machine learning requires the availability of accurately labeled datasets comprising input-output pairs, where the input typically consists of the original Urdu news article and the output comprises the corresponding reference summary. In our research, to effectively leverage supervised learning algorithms and to ensure accurate, stable, and generalized training of our state-of-the-art baseline deep learning models, we systematically split our extensive dataset using a clearly defined Train/Test/Validation approach. The dataset was partitioned into three mutually exclusive subsets to facilitate systematic model development and evaluation: a training set for parameter learning, a validation set for hyperparameter tuning and model selection, and a testing set for final performance assessment. This division ensures unbiased evaluation and robust generalization across unseen data.

The training subset, utilized exclusively for model training, comprised 2,027,784 instances, thus providing substantial and diverse linguistic examples to facilitate robust model learning. Meanwhile, the validation and testing subsets each consisted of 20,000 instances, carefully reserved to assess and validate the models performance independently of the training set.

Moreover, to effectively train and evaluate more computationally demanding transformer-based models, we faced constraints related to computation cost and resources. Therefore, a specialized and cost-effective strategy was adopted. For the fine-tuning phase of transformer-based models, we utilized a smaller, carefully selected subset of 100 instances due to computational limitations. Despite this limited fine-tuning dataset, we ensured rigorous and unbiased evaluation by conducting testing on a significantly larger dataset of 20,000 instances. This deliberate choice was made to guarantee fairness, reliability, and accuracy in performance comparisons between transformer models and deep learning architectures, ensuring our evaluation process accurately reflected the practical capabilities of these state-of-the-art models under realistic computational constraints.

5.2.4 Evaluation Measures

In terms of evaluation, the deep learning models and transformer-based models were comprehensively assessed using the widely adopted ROUGE evaluation metrics, specifically leveraging the averaged F_1 scores across three distinct measures: ROUGE-1, ROUGE-2, and ROUGE-L metrics[3]. These evaluation metrics have been broadly recognized for their robustness in assessing the quality of automatically generated summaries by comparing them directly with human-authored reference summaries. ROUGE-based evaluations primarily target two essential aspects of summary quality: (1) content overlap, which measures the extent to which the generated summary replicates key lexical elements and informational units from the reference summary, and (2) linguistic fluency and readability, evaluated indirectly through the coherence and logical progression of the

[3] https://github.com/pltrdy/rouge Last Visited: 25-Sep-2024

generated text. These metrics collectively provide insight into both the informativeness and the structural quality of the summaries.

More specifically, ROUGE-1 and ROUGE-2 metrics were selected to measure content overlap at different linguistic granularity levels. ROUGE-1 evaluates overlap based on individual words or unigrams, assessing how closely the generated summary mirrors the exact lexical items present in the reference summary. The mathematical formulations of ROUGE-1 (precision, recall, and F_1-score) are defined explicitly in equations (5.2.1)-(5.2.3). Meanwhile, ROUGE-2 extends this evaluation further by assessing bigram overlap, explicitly capturing pairs of consecutive words from the generated summary and comparing their overlap with the reference summary. The ROUGE-2 calculations, comprising precision, recall, and F_1-score, are detailed in equations (5.2.4)-(5.2.6). These two metrics together provide a robust quantification of lexical similarity between reference and automatically generated summaries, ensuring a comprehensive assessment of content accuracy and informativeness.

In addition, to evaluate the fluency and linguistic coherence of summaries, the ROUGE-L metric was utilized. ROUGE-L specifically assesses the Longest Common Subsequence (LCS) between the automatically generated summary and the reference summary, providing a measure that reflects structural fluency, readability, and coherence. Unlike unigram and bigram matching methods, ROUGE-L focuses on the sequence of words, taking into account the semantic continuity and logical structure of sentences. The detailed formulation of ROUGE-L (including the calculation of the longest common subsequence) is depicted explicitly through equations (5.2.8)-(5.2.10), utilizing the general LCS computation method shown in equation (5.2.7). The inclusion of ROUGE-L alongside ROUGE-1 and ROUGE-2 thus ensures a balanced and comprehensive assessment of both content accuracy and summary fluency, effectively capturing the nuances of linguistic coherence necessary for high-quality summarization.

Finally, the performance evaluation across all ROUGE metrics. ROUGE-1, ROUGE-2, and ROUGE-L, was systematically conducted by computing Precision, Recall, and F_1-scores. Precision reflects the accuracy and relevance of generated content relative to the reference summary, recall assesses the comprehensiveness and completeness of content coverage, and the F_1-score balances these two measures into a single, comprehensive performance indicator. By computing these three metrics across all ROUGE variants, we established a robust and multidimensional evaluation framework. This framework allowed us to rigorously assess and comparatively analyze the strengths, weaknesses, and overall effectiveness of each summarization model employed, facilitating an insightful understanding of their respective capabilities and highlighting areas for potential improvement.

Through this structured and methodically rigorous training, fine-tuning, and evaluation process, our research aimed to provide robust baseline benchmarks and comprehensive insights into the current state-of-the-art in UATS, significantly contributing to the advancement and practical deployment of Urdu summarization methodologies in real-world NLP applications.

ROUGE-1

$$Precision(P) = \frac{\text{Number of Overlapping Unigrams}}{\text{Total words in Automatic Summary}} \tag{5.2.1}$$

$$Recall(R) = \frac{\text{Number of Overlapping Unigrams}}{\text{Total words in Reference Summary}} \tag{5.2.2}$$

$$F_1 = \frac{2(P \times R)}{P + R} \tag{5.2.3}$$

ROUGE-2

$$Precision(P) = \frac{\text{Number of Overlapping bigrams}}{\text{Total words in Automatic Summary}} \tag{5.2.4}$$

$$Recall(R) = \frac{\text{Number of Overlapping bigrams}}{\text{Total words in Reference Summary}} \tag{5.2.5}$$

$$F_1 = \frac{2(P \times R)}{P + R} \tag{5.2.6}$$

ROUGE-L

$$LCS = max\{s_1, s_2, s_3 s_n\} \tag{5.2.7}$$

$$Precision(P) = \frac{|LCS|}{\text{Total words in Automatic Summary}} \tag{5.2.8}$$

$$Recall(R) = \frac{|LCS|}{\text{Total words in Reference Summary}} \tag{5.2.9}$$

$$F_1 = \frac{2(P \times R)}{P + R} \tag{5.2.10}$$

where, s_n denotes the subsequence n.

To illustrate the step-by-step computation of ROUGE metrics used in this study, a detailed example has been manually evaluated and presented in Table 5.3. This table demonstrates the mathematical procedure for calculating ROUGE-1, ROUGE-2, and ROUGE-L scores based on a real example involving an Urdu reference summary and a generated summary.

Each row of the table corresponds to a specific ROUGE metric. The example includes the following components:

- **Reference Summary:** The original gold-standard summary written by a human annotator.
- **Reference and Generated n-grams:** Lists of tokenized unigrams (for ROUGE-1), bigrams (for ROUGE-2), and token sequences used for LCS computation (for ROUGE-L).
- **Overlap Count:** The number of common tokens, bigrams, or longest subsequence between the reference and the generated summaries, along with total counts used in precision and recall computations.
- **Precision, Recall, and F_1-Score:** Each metric is computed using standard ROUGE formulas, showing step-by-step derivations with substituted values.

This detailed presentation serves two main purposes: (1) it clarifies the exact evaluation methodology adopted throughout this study for both linguistic and computational

Table 5.3: Manual ROUGE Score Computation Example with Detailed Calculations

ROUGE	Reference Summary	Reference n-grams	Generated n-grams	N-gram / Bi-gram / LCS (Count)	Precision	Recall	F_1
ROUGE-1	پاکستان نے آئی ایم ایف کے ساتھ نئے مالی معاہدے پر دستخط کر دیے ہیں تاکہ معیشت کو سہارا دیا جا سکے۔	پاکستان، نے، آئی، ایم، ایف، کے، ساتھ، نئے، مالي، معاہدے، پر، دستخط، کر، دیے، ہیں، تاکہ، معیشت، کو، سہارا، دیا، جا، سکے	پاکستان، نے، آئی، ایم، ایف، کے، ساتھ	Matched: 7 Ref: 22 Gen: 11	$P = \frac{7}{11} = 0.636$	$R = \frac{7}{22} = 0.318$	$F_1 = 2 \cdot \frac{0.636 \times 0.318}{0.636 + 0.318} = 0.424$
ROUGE-2	پاکستان نے آئی ایم ایف کے ساتھ نئے مالی معاہدے پر دستخط کر دیے ہیں تاکہ معیشت کو سہارا دیا جا سکے۔	پاکستان نے، نے آئی، آئی ایم، ایم ایف، ایف کے، کے ساتھ، ساتھ نیا، ...	پاکستان نے، نے آئی، آئی ایم، ایم ایف، ایف کے، کے ساتھ	Matched: 6 Ref: 21 Gen: 10	$P = \frac{6}{10} = 0.600$	$R = \frac{6}{21} = 0.286$	$F_1 = 2 \cdot \frac{0.600 \times 0.286}{0.600 + 0.286} = 0.387$
ROUGE-L	پاکستان نے آئی ایم ایف کے ساتھ نئے مالی معاہدے پر دستخط کر دیے ہیں تاکہ معیشت کو سہارا دیا جا سکے۔	پاکستان، نے، آئی، ایم، ایف، کے، ساتھ، نیا، معاہدہ، کیا، ہے۔	پاکستان، نے، آئی، ایم، ایف، کے، ساتھ	LCS = 7 Ref: 22 Gen: 11	$P = \frac{7}{11} = 0.636$	$R = \frac{7}{22} = 0.318$	$F_1 = 2 \cdot \frac{0.636 \times 0.318}{0.636 + 0.318} = 0.424$

Table 5.4: Results Obtained by Applying Various State-of-the-Art Baseline Deep Learning Models on Our Proposed UATS-23 Corpus

Model	ROUGE		
	1	2	L
LSTM	24.0	7.2	27.0
Bi-LSTM	27.0	8.9	29.0
GRU	43.0	19.0	43.0
Bi-GRU	22.0	5.4	26.0
LSTM with Atten.	26.0	9.0	30.0
GRU with Atten.	**46.7**	**24.1**	**48.7**
BART	1.7	0.0	0.1
GPT-3.5	13.6	4.2	12.1
DeepSeek-R1	37.6	21.0	35.6

audiences, and (2) it provides a transparent, reproducible demonstration of how ROUGE-based evaluation metrics assess content similarity. This reinforces the validity and interpretability of the automated evaluation scores reported across the experimental results.

5.3 Results and Analysis

Table 5.4 reports the performance outcomes of a comprehensive set of models applied to the UATS-23 corpus, encompassing six baseline deep learning architecturesLSTM, Bi-LSTM, LSTM with attention, GRU, Bi-GRU, and GRU with attentionalong with two state-of-the-art transformer-based models, BART and GPT-3.5, and an additional large language model, DeepSeek-R1. The evaluation is based on the average scores across three widely adopted ROUGE[4] metrics: ROUGE-1, ROUGE-2, and ROUGE-L, which collectively assess the lexical overlap and structural alignment between the generated summaries and their corresponding reference texts.

Overall, the GRU with attention model achieves the highest results (with F_1 scores of ROUGE-1 = 46.7, ROUGE-2 = 24.1, and ROUGE-L = 48.7), underscoring its suitability and superior effectiveness for the Urdu ATS task on our corpus. The performance of this model significantly surpasses other deep learning architectures included in our

[4] The detailed results can be downloaded from the following link: https://drive.google.com/drive/folders/1bUHCchr5jOfUMKz8WcvGYNFc_C
SB_XYj?usp=sharing

experiments. Notably, the close proximity between ROUGE-1 and ROUGE-L scores indicates the models ability to generate summaries with both high lexical overlap and enhanced fluency, matching human-written summaries more closely. The marked effectiveness of the GRU with attention model can largely be attributed to the attention mechanism's capability to efficiently identify and prioritize key information within longer text sequences, enabling the GRU model to focus its computational efforts selectively on relevant input portions. This selective contextual emphasis is particularly beneficial given the linguistic complexity of Urdu, characterized by intricate sentence structures, morphology, and rich semantic dependencies. Moreover, GRUs are generally known to perform better than LSTMs in scenarios involving complex sequential dependencies due to their simplified gate structure, which reduces computational complexity while preserving contextual integrity. This structural advantage becomes even more pronounced when combined with attention mechanisms, as it enables the model to dynamically adjust its focus, substantially improving summarization quality, especially in contexts with linguistically challenging datasets like Urdu.

Among other baseline deep learning models without attention mechanisms, the GRU architecture (ROUGE-1 = 43.0, ROUGE-2 = 19.0, ROUGE-L = 43.0) outperforms traditional LSTM (ROUGE-1 = 24.0, ROUGE-2 = 7.2, ROUGE-L = 27.0) and Bi-directional variants. The Bi-LSTM and Bi-GRU models show comparatively lower scores, particularly in the case of Bi-GRU (ROUGE-1 = 22.0, ROUGE-2 = 5.4, ROUGE-L = 26.0). The unexpected underperformance of bidirectional models, especially the Bi-GRU, could potentially arise from complexities and redundancies introduced by bidirectional information processing, which may negatively influence learning dynamics in the specific case of Urdu text summarization. Further fine-tuning and parameter optimization, or experimentation with more focused bidirectional attention mechanisms, could potentially rectify these performance issues.

On the other hand, the LSTM with attention model yields modest improvements compared to standard LSTM and Bi-LSTM models (ROUGE-1 = 26.0, ROUGE-2 = 9.0, ROUGE-L = 30.0), although the improvements are not nearly as significant as ob-

Table 5.5: Examples of Good BART Model Performance

Human Summary	BART Generated Summary
مکسیکو میچ کے دوران امپائر کا قتل	میکسیو سپورٹس ڈیسک میکسیکن ریاست ہڈالگو میں فٹ بال میچ کے دوران کھلاڑی نے سر سے ٹکر مار کر میچ ریفری کا قتل کردیا
خالد محمود سے اظہار تعزیت	لاہور کلچرل رپورٹر بزم عمل پاکستان لاہور کے زیراہتمام شاعر اقبال راہی کی سالگرہ کے موقع پر آج شام لاہور میں مشاعرہ ہوگا
محفل مشاعرہ کل ہوگی	لاہور کلچرل رپورٹر بزم عمل پاکستان لاہور کے زیراہتمام شاعر اقبال راہی کی سالگرہ کے موقع پر آج شام لاہور میں مشاعرہ ہوگا

Table 5.6: Examples of Good GPT-3.5 Model Performance

Human Summary	GPT-3.5 Generated Summary
سرفراز احمد انگلینڈ روانہ ہوگئے	جیکب آباد میں شادی کی تقریب کو روکنے جبکہ یارک شائر کی نمائندگی کرنے کے لئے بعد کیریبین پریمیئر لیگ میں شرکت کیلئے ویسٹ انڈیز روانہ کیلئے سرفراز احمد
پاکستان نے والے فٹبالر کون	کریئر میں عمر صرف 17 سال تھی رابرٹ پیرز کا تعلق ویلز سے اہم کرنے کا اعلان
کپاس کے داموں میں مجموعی طور پر استحکام	پاکستان میں کپاس کی باعث کاروباری حجم بھی گھنٹے پر استحکام جاری رہا

served in GRU variants. This relatively lower improvement of the LSTM with attention approach emphasizes once again the suitability of GRU architectures in capturing the linguistic and semantic nuances specific to Urdu text.

The evaluation of LLMs such as BART, GPT-3.5, and DeepSeek-R1 reveals lower ROUGE-based results compared to traditional deep learning models. Specifically, BART yields minimal performance (ROUGE-1 = 1.7, ROUGE-2 = 0.0, ROUGE-L = 0.1), and GPT-3.5 also performs poorly (ROUGE-1 = 13.6, ROUGE-2 = 4.2, ROUGE-L = 12.1). In contrast, DeepSeek-R1 demonstrates significantly better results (ROUGE-1 = 37.6, ROUGE-2 = 21.0, ROUGE-L = 35.6), showing promise and highlighting substantial variability in the performance of large pretrained models. The relatively low ROUGE scores of BART and GPT-3.5 are primarily due to the limitations inherent in ROUGE metrics, which predominantly capture lexical and n-gram overlaps, thus failing to ade-

Table 5.7: Examples of Good DeepSeek-R1 Model Performance

Human Summary	DeepSeek-R1 Generated Summary
شکیل احمد قریشی کو صدمہ	حیدرآباد کے نامور کرکٹر سہیل قریشی کا انتقال، انہیں نیو مسلم قبرستان میں سپرد خاک کیا گیا۔
کرسٹیانو رونالڈو چین پہنچ گئے	رونالڈو نے چین میں تاریخی دن قرار دیا، شائقین نے داد دی، میچ دو گول سے برابر رہا۔
خالد محمود سے اظہار تعزیت	کراچی کے کھلاڑی خالد محمود کے بھائی ارشد محمود کی وفات پر کلب رہنماؤں نے تعزیت کا اظہار کیا۔

quately assess semantic and contextual correctness inherent in summaries generated by sophisticated LLMs. Manual inspection and qualitative analysis indicate that despite low ROUGE scores, summaries generated by BART and GPT-3.5 models possess meaningful contextual relevance, semantic coherence, and grammatical correctness, suggesting that these models produce useful and semantically valid outputs that traditional lexical overlap measures fail to adequately capture.

The promising performance of DeepSeek-R1 indicates that this LLM can effectively leverage its extensive pretraining to produce more coherent, fluent, and contextually accurate Urdu summaries. However, further performance improvements for DeepSeek-R1, GPT-3.5, and BART are constrained by the limited fine-tuning dataset size used, driven primarily by computational resource limitations and high costs associated with larger dataset processing. Specifically, GPT-3.5 calculates tokens differently for the Urdu language compared to English, substantially impacting computational costs and complexity of resource allocation. Moreover, sustained accessibility to GPUs for long-term training remains a substantial challenge for researchers, further limiting the potential of comprehensive experimentation on large-scale datasets.

The qualitative analysis presented in Table 5.5, Table 5.6, and Table 5.7 highlights the performance differences among the transformer-based models (BART, GPT-3.5, and DeepSeek-R1) in generating abstractive summaries for Urdu text. Despite their relatively lower ROUGE scores reported earlier, the illustrative examples clearly indicate

Table 5.8: Detailed Experimental Results of all context-limited, context-aware, transformers, and LLM Models

Model Name	**ROUGE-1**			**ROUGE-2**			**ROUGE-L**		
	F_1	**P**	**R**	F_1	**P**	**R**	F_1	**P**	**R**
Bi-RNN	0.17	0.12	0.36	0.06	0.05	0.11	0.20	0.13	0.46
LSTM	0.24	0.24	0.24	0.07	0.07	0.07	0.27	0.23	0.37
Bi-LSTM	0.27	0.27	0.27	0.08	0.08	0.08	0.29	0.26	0.38
GRU	0.43	0.43	0.43	0.19	0.19	0.19	0.43	0.39	0.53
Bi-GRU	0.22	0.22	0.22	0.05	0.05	0.05	0.26	0.18	**0.58**
LSTM with Atten.	0.26	0.26	0.26	0.09	0.09	0.09	0.30	0.25	0.50
GRU with Atten.	**0.46**	**0.46**	**0.46**	**0.24**	**0.24**	**0.24**	**0.48**	**0.44**	0.55
BART	0.01	0.03	0.12	0.00	0.16	0.00	0.01	0.03	0.01
GPT-3.5	0.13	0.26	0.09	0.04	0.09	0.02	0.12	0.23	0.08
DeepSeek-R1	0.27	0.60	0.37	0.14	0.36	0.21	0.26	0.57	0.35

that these transformer-based models, particularly DeepSeek-R1, generate summaries that are semantically meaningful, contextually relevant, and linguistically fluent. The BART model, although occasionally prone to content misalignment, still exhibits good syntactic fluency and coherent structuring in generated summaries. GPT-3.5, while generating linguistically plausible outputs, demonstrates occasional semantic inaccuracies or irrelevant content generation, reflecting challenges associated with minimal fine-tuning data and its tokenization strategy for Urdu text. Conversely, DeepSeek-R1's summaries closely align with human-written summaries in terms of semantic accuracy, contextual coherence, and overall readability, underscoring its superior performance relative to other transformer-based models. These observations further support the necessity for adopting semantic and context-sensitive evaluation metrics alongside traditional ROUGE evaluations to more accurately reflect the true summarization performance of LLMs for UATS.

The detailed results from the experiments, clearly illustrated in the table 5.8, show the varying performance of the different deep learning and transfer learning models across the three metrics, Rouge-1, Rouge-2, and Rouge-L. The GRU with Attention model, in particular, demonstrates the highest F_1 scores across all metrics, significantly outperforming the other models. With a F_1 score of 0.46 for Rouge-1, 0.24 for Rouge-2, and 0.48 for Rouge-L, the GRU with Attention model proves to be the most effective model in this study. The attention mechanism incorporated into the GRU model enhances its ability to capture long-range dependencies and contextual information, which is crucial for sequence-based tasks like the one in this study. This model excels not only in unigram predictions but also shows significant improvement in bigram and longest common subsequence predictions, demonstrating a strong capacity for understanding complex relationships between words.

In contrast, simpler models such as Bi-RNN and BART show significantly poorer performance across all evaluation metrics. The Bi-RNN model, for instance, yields an F_1 score of only 0.17 in Rouge-1, 0.06 in Rouge-2, and 0.20 in Rouge-L. These results indicate that the Bi-RNN struggles significantly with both unigram and bigram predictions, with minimal improvement in capturing long-term dependencies. Similarly, BART scores extremely low across all metrics, with a F_1 score of just 0.01 for Rouge-1 and 0.01 for Rouge-L. The poor performance of BART suggests that this model may not be well-suited for the dataset or task at hand, as it fails to capture the essential relationships in the input sequences effectively.

The LSTM with Attention model improves upon the standard LSTM, yielding F_1 scores of 0.26 for Rouge-1, 0.09 for Rouge-2, and 0.30 for Rouge-L. The attention mechanism helps the model focus on the most relevant parts of the input sequence, leading to improvements in both unigram and long-term dependency predictions. However, despite these enhancements, the LSTM with Attention still does not outperform the GRU with Attention, particularly in terms of bigram and long-range dependency predictions. The DeepSeek-R1 model, which also incorporates some advanced techniques, performs reasonably well, with an F_1 score of 0.27 for Rouge-1 and 0.14 for Rouge-2, but it still falls short when compared to the GRU with Attention model. This model shows some capacity for capturing long-range dependencies, but its overall performance is still less impressive than that of the attention-based models.

The GPT-3.5 model, though more advanced, shows weaker performance than other models. Its F_1 scores are relatively low, with 0.13 for Rouge-1, 0.04 for Rouge-2, and 0.12 for Rouge-L. While GPT-3.5 performs better than BART, it is still outperformed by models such as GRU with Attention and LSTM with Attention, which are better at handling the complexities of this particular task.

The optimal hyperparameter values obtained for our highest-performing model (GRU with attention) include a learning rate of 0.001, a dropout rate of 0.5, an optimal number of epochs set at 7, and a batch size of 32 with a single hidden layer. Despite the promising performance, the overall ROUGE scores across all models remain modest, emphasizing the inherent complexity and challenges associated with the Urdu abstractive summarization task. Urdu's linguistic characteristicsincluding morphology, syntactic complexity, and linguistic ambiguitiescontribute significantly to this difficulty, highlighting the necessity for ongoing research, innovative approaches, and further methodological advancements.

Future research directions could benefit from detailed investigations into interpretability and explainability mechanisms, particularly exploring how attention mechanisms prioritize information. Such analysis could uncover deeper insights into the summarization process and the linguistic aspects most effectively captured by models. Rigorous hyperparameter optimization through techniques such as grid search, random search, or Bayesian optimization is recommended for further enhancing the GRU with attention model's performance. Additionally, incorporating linguistic features, domain-specific embeddings, or leveraging additional metadata could significantly improve summarization effectiveness, especially in capturing subtle linguistic nuances. Lastly, addressing computational constraints by finding cost-effective methodologies to expand fine-tuning datasets for transformer models remains a crucial research challenge, necessary to fully leverage their capabilities for Urdu and similar low-resource languages.

These preliminary results and analyses provide researchers with essential baseline performances and methodological insights, facilitating future advancements in UATS and informing subsequent research initiatives in this critical NLP domain.

5.4 Chapter Summary

This chapter provides an in-depth evaluation of deep learning, transformer-based models, and LLMs for UATS, applied to the UATS-23 corpus. The results highlight the superior performance of GRU with Attention, which achieved the highest ROUGE scores (ROUGE-1 = 46.7, ROUGE-2 = 24.1, ROUGE-L = 48.7), demonstrating its ability to capture contextual relationships effectively. Among the LLMs, DeepSeek-R1 outperformed GPT-3.5 and BART, indicating its potential for further fine-tuning in low-resource languages like Urdu.

Chapter 6
Conclusion and Future Work

6.1 Thesis Summary

The problem of UATS was addressed by developing and rigorously evaluating the proposed UATS-23 corpus, designed explicitly to support comprehensive research in this challenging NLP task. Our proposed corpus consists of approximately 2.067 million summaries paired with Urdu news articles sourced from multiple domains, including sports, entertainment, national and international news, business, columns, science, technology, crime, and health. The diverse topical distribution and extensive size of our corpus ensure sufficient linguistic and semantic variability, facilitating robust training and rigorous evaluation of ATS models. To demonstrate the effectiveness, usability, and suitability of our proposed corpus, we implemented and evaluated nine state-of-the-art summarization models, including both baseline deep learning architectures (LSTM, Bi-LSTM, GRU, Bi-GRU, and their attention-based variants) and advanced transformer-based models (BART, GPT-3.5, and DeepSeek-R1).

Experimental results indicated that among all evaluated techniques, the GRU-based encoder-decoder architecture with attention mechanism achieved the highest summarization performance on our corpus, attaining a substantial F_1 score of ROUGE-1 = 46.7, ROUGE-2 = 24.1, and ROUGE-L = 48.7. This superior performance is attributed to the GRU architecture's ability to efficiently capture long-range dependencies and complex linguistic structures prevalent within Urdu texts, particularly when coupled with attention mechanisms. However, despite this promising performance, the overall ROUGE scores across the evaluated models remain relatively modest, clearly reflecting the inherent challenges and complexities of the UATS task.

Our qualitative analysis further revealed important insights beyond the quantitative metrics provided by the ROUGE evaluation alone. Transformer-based models, despite exhibiting relatively lower lexical overlap scores, generated contextually accurate and linguistically fluent summaries upon manual inspection. Specifically, models such as BART, and GPT-3.5, though performing inadequately on traditional ROUGE measures, qualitatively demonstrated their potential to generate meaningful and contextually coherent Urdu text. Significantly better outcomes were observed with the DeepSeek-R1 model, which produced summaries that more closely mirrored human-authored reference summaries in terms of semantic coherence, relevance, and linguistic fluency, thereby reinforcing the notion that LLMs hold substantial promise for Urdu NLP applications when adequately fine-tuned or optimized. Manual inspection of selected outputs revealed notable discrepancies between the ROUGE metrics and qualitative analysis. While the ROUGE scores were limited due to their focus on lexical and n-gram overlaps, summaries generated by transformer models frequently contained linguistically fluent, contextually appropriate, and semantically meaningful content. This suggests a clear gap in current evaluation practices, highlighting a need for developing or adopting evaluation methodologies that better capture the semantic and contextual fidelity of abstractive summaries beyond simple lexical comparisons.

Furthermore, the experiments highlighted significant computational challenges associated with large-scale transformer-based models, particularly due to GPU memory limitations and computational resource requirements. To overcome these resource constraints, we limited fine-tuning instances to a manageable size (100 instances), which, although feasible, clearly limited the overall performance of these transformer models. To achieve substantial performance improvements, future research should consider larger fine-tuning datasets, resource optimization strategies, or cloud-based computational solutions to effectively harness the full potential of transformer-based models for Urdu text summarization.

Moreover, the optimal performance parameters derived for our top-performing model (GRU with attention) included a learning rate of 0.001, dropout rate set to 0.5, a batch size of 32, and 7 training epochs. Hyperparameter optimization, particularly utilizing advanced methodologies such as Bayesian optimization or random search, could further enhance these results. Additionally, future research directions should prioritize the interpretability and explainability of attention mechanisms, helping researchers better understand the linguistic and contextual factors driving summarization performance.

In terms of technical implementation, the experiments were carefully executed on the Google Vertex AI platform, utilizing a single NVIDIA Tesla T4 GPU equipped with 60 GB RAM and 16 virtual CPUs. The choice of this computational platform was informed by considerations of computational cost, efficiency, and feasibility, especially given the substantial dataset size involved. Each model was implemented using PyTorch on Python 3.7, ensuring computational efficiency, stability, and flexibility. A comprehensive hyperparameter configuration was uniformly applied to each model, and gradient clipping and dropout techniques were employed to address computational constraints and training stability effectively.

Collectively, our methodological approach, comprising systematic corpus design, rigorous model evaluation, comprehensive qualitative assessments, and detailed experimental procedures, significantly advances Urdu ATS research. These results provide a robust baseline and valuable insights, paving the way for future methodological enhancements, improved semantic evaluation metrics, and more effective strategies for addressing computational limitations, thereby promoting further meaningful research in Urdu natural language processing and summarization tasks.

6.1.1 Thesis Contributions

The following contributions are made in this Ph.D. research, significantly advancing the field of UATS:

1. **Development of UATS-23 Corpus**: Introduced a large-scale benchmark dataset with 2.067 million Urdu news articles and abstractive summaries, covering diverse domains to facilitate high-quality ATS research in low-resource languages.

2. **Development of Deep Learning Techniques**: Developed and optimized six state-of-the-art deep learning architectures, demonstrating that GRU with attention

achieves the best performance for Urdu summarization tasks, surpassing conventional LSTM-based approaches.

3. **Development of Transfer Learning Techniques**: Fine-tuned transformer-based architecture (BART) on the UATS-23 corpus, providing empirical insights into pre-trained summarization models for Urdu, and identifying key limitations in ROUGE-based evaluations.

4. **Development of LLMs-Based Technique**: Evaluated GPT-3.5 and DeepSeek-R1 for zero-shot and few-shot UATS, demonstrating the semantic superiority of LLMs over conventional approaches, while highlighting computational and linguistic challenges.

5. **Direct Comparison of State-of-the-Art Methods on UATS-23 Corpus**: Conducted the performance benchmarking of deep learning, transfer learning, and LLM-based techniques on a large-scale UATS-23 corpus, establishing quantitative and qualitative evaluation baselines for future research.

6.2 Future Work

Future research should focus on expanding the UATS-23 corpus to include multi-domain coverage, particularly in healthcare, finance, legal, and scientific research. Incorporating diverse datasets will improve the adaptability of ATS models, allowing them to generate more contextually accurate and domain-specific summaries. In addition, cross-domain generalization studies should be conducted to assess how well models trained on one domain transfer knowledge to other specialized fields.

Hyperparameter optimization and model interpretability remain crucial for improving summarization performance. Future research should explore advanced hyperparameter tuning techniques such as random search, grid search, and Bayesian optimization to maximize the efficiency of GRU-based and transformer-based models. Additionally, explainability techniques such as attention heatmaps, SHAP, and Integrated Gradients should be leveraged to analyze how models prioritize different parts of input text, ensuring transparency and reducing biases in Urdu summarization.

Further research should investigate new LLMs such as GPT-4, Falcon, LLaMA, and Mistral, comparing their computational efficiency and algorithmic performance against existing models. A comprehensive study on brute-force computation versus optimized inference strategies (e.g., beam search versus contrastive decoding) will help identify the most efficient and effective models for Urdu ATS tasks, ensuring better scalability and lower latency.

The computational cost and resource limitations associated with fine-tuning large-scale LLMs pose significant challenges for low-resource NLP research. Future work should explore cost-efficient fine-tuning strategies such as LoRA, QLoRA, and Parameter-Efficient Fine-Tuning (PEFT) to reduce computational overhead without compromising performance. Additionally, leveraging cloud-based distributed training frameworks and optimizing GPU scheduling strategies will enable more effective utilization of hardware resources for large-scale Urdu NLP experiments.

Finally, ROUGE-based evaluation metrics fail to capture semantic fidelity, which requires the adoption of semantic-aware evaluation techniques such as BERTScore, BLEURT, and MoverScore. Future research should develop a hybrid evaluation framework that integrates lexical, syntactic, and semantic measures to ensure a more holistic, accurate, and human-aligned assessment of ATS quality.

References

[1] Mahak Gambhir and Vishal Gupta. "Recent automatic text summarization techniques: a survey." In: *Artificial Intelligence Review* 47.1 (2017), pp. 1–66.

[2] Romain Paulus, Caiming Xiong, and Richard Socher. "A deep reinforced model for abstractive summarization." In: *arXiv preprint arXiv:1705.04304* (2017). doi:`https://doi.org/10.48550/arXiv.1705.04304`.

[3] Alexander M et. al. Rush. "A neural attention model for abstractive sentence summarization." In: *arXiv preprint arXiv:1509.00685* (2015). doi:`https://doi.org/10.18653/v1/d15-1044`.

[4] Abigail See, Peter J Liu, and Christopher D Manning. "Get to the point: Summarization with pointer-generator networks." In: *arXiv preprint arXiv:1704.04368* (2017). doi:`https://doi.org/10.18653/v1/p17-1099`.

[5] Jiacheng Xu and Greg Durrett. "Neural extractive text summarization with syntactic compression." In: *arXiv preprint arXiv:1902.00863* (2019). doi:`https://doi.org/10.18653/v1/d19-1324`.

[6] Sumit Chopra and Auli et. al. "Abstractive sentence summarization with attentive recurrent neural networks." In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*. doi:`https://doi.org/10.18653/v1/n16-1012`. 2016, pp. 93–98.

[7] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." In: *arXiv preprint arXiv:1810.04805* (2018). doi:`https://doi.org/10.48550/arXiv.1810.04805`.

[8] Mike Lewis et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." In: *arXiv preprint arXiv:1910.13461* (2019). doi:`https://doi.org/10.18653/v1/2020.acl-main.703`.

[9] Tom Brown et al. “Language models are few-shot learners.” In: *Advances in neural information processing systems* 33 (2020). doi:https://doi.org/10.48550/arXiv.2005.14165, pp. 1877–1901.

[10] Wojciech Kryciski et al. “Improving abstraction in text summarization.” In: *arXiv preprint arXiv:1808.07913* (2018). doi:https://doi.org/10.18653/v1/d18-1207.

[11] Paul Over, Hoa Dang, and Donna Harman. “DUC in context.” In: *Information Processing & Management* 43.6 (2007). doi:https://doi.org/10.1016/j.ipm.2007.01.019, pp. 1506–1520.

[12] Evan Sandhaus. “The new york times annotated corpus.” In: *Linguistic Data Consortium, Philadelphia* 6.12 (2008). doi:https://hdl.handle.net/11272.1/AB2/GZC6PL, e26752.

[13] Elena Mozzherina. “An approach to improving the classification of the New York Times annotated corpus.” In: *Knowledge Engineering and the Semantic Web: 4th International Conference, KESW 2013, St. Petersburg, Russia, October 7-9, 2013. Proceedings 4*. doi:https://doi.org/10.1007/978-3-642-41360-5_7. Springer. 2013, pp. 83–91.

[14] Ramesh Nallapati et al. “Abstractive text summarization using sequence-to-sequence rnns and beyond.” In: *arXiv preprint arXiv:1602.06023* (2016). doi:https://doi.org/10.18653/v1/k16-1028.

[15] Max Grusky, Mor Naaman, and Yoav Artzi. “Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies.” In: *arXiv preprint arXiv:1804.11283* (2018). doi:https://doi.org/10.18653/v1/n18-1065.

[16] Mahnaz Koupaee and William Yang Wang. “Wikihow: A large scale text summarization dataset.” In: *arXiv preprint arXiv:1810.09305* (2018). doi:https://doi.org/10.48550/arXiv.1810.09305.

[17] Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. “Annotated gigaword.” In: *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. doi:`https://doi.org/10.35111/mv9t-vv26`. Association for Computational Linguistics. 2012, pp. 95–100.

[18] Mehrdad Farahani, Mohammad Gharachorloo, and Mohammad Manthouri. “Leveraging ParsBERT and Pretrained mT5 for Persian Abstractive Text Summarization.” In: *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*. doi:`https://doi.org/10.1109/csicc52343.2021.9420563`. IEEE. 2021, pp. 1–6.

[19] Baotian Hu, Qingcai Chen, and Fangze Zhu. “Lcsts: A large scale chinese short text summarization dataset.” In: *arXiv preprint arXiv:1506.05865* (2015). doi:`https://doi.org/10.18653/v1/d15-1229`.

[20] Tariq Rahman. “Language policy and localization in Pakistan: proposal for a paradigmatic shift.” In: *SCALLA Conference on computational linguistics*. Vol. 99. 2004. 2004, pp. 1–19.

[21] Asma Naseer and Sarmad Hussain. “Supervised word sense disambiguation for Urdu using Bayesian classification.” In: *Center for Research in Urdu Language Processing, Lahore, Pakistan* (2009).

[22] Ali Daud, Wahab Khan, and Dunren Che. “Urdu language processing: a survey.” In: *Artificial Intelligence Review* 47.3 (2017). doi:`https://doi.org/10.1007/s10462-016-9482-x`, pp. 279–311.

[23] Ali Nawaz et al. “Extractive text summarization models for Urdu language.” In: *Information Processing & Management* 57.6 (2020). doi:`https://doi.org/10.1016/j.ipm.2020.102383`, p. 102383.

[24] Muhammad Humayoun et al. “Urdu summary corpus.” In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. doi:`https://aclanthology.org/L16-1128`. 2016, pp. 796–800.

[25] Bareera Sadia et al. “Meeting the challenge: A benchmark corpus for automated Urdu meeting summarization.” In: *Information Processing & Management* 61.4 (2024), p. 103734.

[26] Ali Faheem et al. “UrduMASD: A Multimodal Abstractive Summarization Dataset for Urdu.” In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 2024, pp. 17245–17253.

[27] Ali Raza, Hadia Sultan Raja, and Usman Maratib. “Abstractive summary generation for the Urdu language.” In: *arXiv preprint arXiv:2305.16195* (2023).

[28] Imtiaz Khan et al. “Urdu Language Text Summarization using Machine Learning.” In: *Journal of Computing & Biomedical Informatics* 8.01 (2024).

[29] Rafael Dueire et. al. J Lins. “The cnn-corpus: A large textual corpus for single-document extractive summarization.” In: *Proceedings of the ACM Symposium on Document Engineering 2019*. doi:https://doi.org/10.1145/3342558.3345388. 2019, pp. 1–10.

[30] Karl Moritz Hermann et al. “Teaching machines to read and comprehend.” In: *Advances in neural information processing systems* 28 (2015). doi:https://doi.org/10.48550/arXiv.1506.03340, pp. 1693–1701.

[31] Jiatao Gu et al. “Incorporating copying mechanism in sequence-to-sequence learning.” In: *arXiv preprint arXiv:1603.06393* (2016). doi:https://doi.org/10.18653/v1/p16-1154.

[32] Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. “Bottom-up abstractive summarization.” In: *arXiv preprint arXiv:1808.10792* (2018). doi:https://doi.org/10.18653/v1/d18-1443.

[33] Junyang Lin et al. “Global encoding for abstractive summarization.” In: *arXiv preprint arXiv:1805.03989* (2018). doi:https://doi.org/10.18653/v1/p18-2027.

[34] Jianwei Niu et al. “A novel attention mechanism considering decoder input for abstractive text summarization.” In: *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. doi:https://doi.org/10.1109/icc.2019.8762040. IEEE. 2019, pp. 1–7.

[35] Feng Nan and et. al Nallapati. “Entity-level factual consistency of abstractive text summarization.” In: *arXiv preprint arXiv:2102.09130* (2021). doi:https://doi.org/10.18653/v1/2021.eacl-main.235.

[36] Yang Zhao, Zhiyuan Luo, and Akiko Aizawa. “A language model based evaluator for sentence compression.” In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. doi:https://doi.org/10.18653/v1/p18-2028. 2018, pp. 170–175.

[37] Katja Filippova et al. “Sentence compression by deletion with lstms.” In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. doi:https://doi.org/10.18653/v1/d15-1042. 2015, pp. 360–368.

[38] Fei Liu et al. “Toward abstractive summarization using semantic representations.” In: *arXiv preprint arXiv:1805.10399* (2018). doi:https://doi.org/10.3115/v1/n15-1114.

[39] Yue Dong et al. “Multi-fact correction in abstractive text summarization.” In: *arXiv preprint arXiv:2010.02443* (2020). doi:https://doi.org/10.18653/v1/2020.emnlp-main.749.

[40] Qasem A Al-Radaideh and Dareen Q Bataineh. “A hybrid approach for arabic text summarization using domain knowledge and genetic algorithms.” In: *Cognitive Computation* 10.4 (2018). doi:https://doi.org/10.1007/s12559-018-9547-z, pp. 651–669.

[41] Beliz Gunel et al. “Mind the facts: Knowledge-boosted coherent abstractive text summarization.” In: *arXiv preprint arXiv:2006.15435* (2020). doi:https://doi.org/10.48550/arXiv.2006.15435.

[42] Ilya Gusev. “Dataset for automatic summarization of Russian news.” In: *Artificial Intelligence and Natural Language: 9th Conference, AINL 2020, Helsinki, Finland, October 7--9, 2020, Proceedings 9*. doi:https://doi.org/10.1007/978-3-030-59082-6_9. Springer. 2020, pp. 122–134.

[43] Kazuki Akiyama, Akihiro Tamura, and Takashi Ninomiya. “Hie-BART: Document summarization with hierarchical BART.” In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. doi:https://doi.org/10.18653/v1/2021.naacl-srw.20. 2021, pp. 159–165.

[44] Moreno La Quatra and Luca Cagliero. “BART-IT: An Efficient Sequence-to-Sequence Model for Italian Text Summarization.” In: *Future Internet* 15.1 (2022). doi:https://doi.org/10.3390/fi15010015, p. 15.

[45] Nikolich Alexandr et al. “Fine-tuning gpt-3 for russian text summarization.” In: *Data Science and Intelligent Systems: Proceedings of 5th Computational Methods in Systems and Software 2021, Vol. 2*. doi:https://doi.org/10.1007/978-3-030-90321-3_61. Springer. 2021, pp. 748–757.

[46] Tanya Goyal, Junyi Jessy Li, and Greg Durrett. “News summarization and evaluation in the era of gpt-3.” In: *arXiv preprint arXiv:2209.12356* (2022). doi:https://doi.org/10.48550/arXiv.2209.12356.

[47] Dima Suleiman and Arafat Awajan. “Multilayer encoder and single-layer decoder for abstractive Arabic text summarization.” In: *Knowledge-Based Systems* 237 (2022). doi:https://doi.org/10.1016/j.knosys.2021.107791, p. 107791.

[48] Khalid Hussain et al. “Urdu news dataset 1M.” In: *Mendeley Data* 3 (2021). doi:10.17632/834vsxnb99.3.

[49] Ani Nenkova, Kathleen McKeown, et al. “Automatic summarization.” In: *Foundations and Trendső in Information Retrieval* 5.2--3 (2011), pp. 103–233.

[50] Regina Barzilay and Kathleen R McKeown. “Sentence fusion for multidocument news summarization.” In: *Computational Linguistics* 31.3 (2005), pp. 297–328.

[51] James Clarke and Mirella Lapata. “Modelling compression with discourse constraints.” In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning*. Association for Computational Linguistics. 2007, pp. 1–11.

[52] Razvan Bunescu and Raymond Mooney. “A shortest path dependency kernel for relation extraction.” In: *Proceedings of human language technology conference and conference on empirical methods in natural language processing*. 2005, pp. 724–731.

[53] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O’Reilly Media, Inc.", 2009.

[54] Yiming Cui, Shijin Wang, and Jianfeng Li. “LSTM neural reordering feature for statistical machine translation.” In: *arXiv preprint arXiv:1512.00177* (2015). doi:https://doi.org/10.18653/v1/n16-1112.

[55] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate.” In: *arXiv preprint arXiv:1409.0473* (2014). doi:https://doi.org/10.48550/arXiv.1409.0473.

[56] Yan Chu et al. “Automatic image captioning based on ResNet50 and LSTM with soft attention.” In: *Wireless Communications and Mobile Computing* 2020 (2020). doi:https://doi.org/10.1155/2020/8909458.

[57] Meetkumar Patel et al. “Machine learning approach for automatic text summarization using neural networks.” In: *International Journal of Advanced Research in Computer and Communication Engineering* 7.1 (2018), pp. 194–202.

[58] Shengli Song, Haitao Huang, and Tongxiao Ruan. “Abstractive text summarization using LSTM-CNN based deep learning.” In: *Multimedia Tools and Applications* 78.1 (2019). doi:https://doi.org/10.1007/s11042-018-5749-3, pp. 857–875.

[59] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. “Sequence to sequence learning with neural networks.” In: *Advances in neural information processing systems* 27 (2014). doi:https://doi.org/10.48550/arXiv.1409.3215, pp. 3104–3112.

[60] Samar Haider. “Urdu Word Embeddings.” In: *LREC*. doi:https://aclanthology.org/L18-1155.pdf. 2018, pp. 964–968.

[61] Elliott Jobson and Abiel Gutiérrez. *Abstractive text summarization using attentive sequence-to-sequence rnns*. 2016.

[62] Mohammed NA Ali and Guanzheng Tan. “Bidirectional encoder–decoder model for Arabic named entity recognition.” In: *Arabian Journal for Science and Engineering* 44.11 (2019). doi:https://doi.org/10.1007/s13369-019-04068-2, pp. 9693–9701.

[63] Kamal Al-Sabahi, Zhang Zuping, and Yang Kang. “Bidirectional attentional encoder-decoder model and bidirectional beam search for abstractive summarization.” In: *arXiv preprint arXiv:1809.06662* (2018). doi:https://doi.org/10.48550/arXiv.1809.06662.

[64] Konstantin Lopyrev. “Generating news headlines with recurrent neural networks.” In: *arXiv preprint arXiv:1512.01712* (2015). doi:https://doi.org/10.48550/arXiv.1512.01712.

[65] Alec Radford et al. “Language models are unsupervised multitask learners.” In: *OpenAI blog* 1.8 (2019), p. 9.

[66] Tom Brown et al. “Language models are few-shot learners.” In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[67] Alex M Lamb et al. “Professor forcing: A new algorithm for training recurrent networks.” In: *Advances In Neural Information Processing Systems*. doi:https://doi.org/10.48550/arXiv.1610.09038. 2016, pp. 4601–4609.

[68] Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization.” In: *arXiv preprint arXiv:1711.05101* (2017). doi:https://doi.org/10.48550/arXiv.1711.05101.

[69] Chantal Shaib et al. "Standardizing the measurement of text diversity: A tool and a comparative analysis of scores." In: *arXiv preprint arXiv:2403.00553* (2024).

[70] Dario Amodei et al. "Concrete problems in AI safety." In: *arXiv preprint arXiv:1606.06565* (2016). doi:https://doi.org/10.48550/arXiv.1606.06565.

[71] Shaohua Qi et al. "Review of multi-view 3D object recognition methods based on deep learning." In: *Displays* 69 (2021). doi:https://doi.org/10.1016/j.displa.2021.102053, p. 102053.

[72] Harleen Kaur et al. "A proposed sentiment analysis deep learning algorithm for analyzing COVID-19 tweets." In: *Information Systems Frontiers* 23.6 (2021). doi:https://doi.org/10.1007/s10796-021-10135-7, pp. 1417–1429.

[73] Syed Abdul Basit Andrabi and Abdul Wahid. "Machine translation system using deep learning for English to Urdu." In: *Computational Intelligence and Neuroscience* 2022 (2022). doi:https://doi.org/10.1155/2022/7873012.

[74] A Santhanavijayan, D Naresh Kumar, and Gerard Deepak. "A semantic-aware strategy for automatic speech recognition incorporating deep learning models." In: *Intelligent system design*. doi:https://doi.org/10.1007/978-981-15-5400-1_25. Springer, 2021, pp. 247–254.

[75] Ramesh Nallapati, Bing Xiang, and Bowen Zhou. "Sequence-to-sequence rnns for text summarization." In: (2016).

[76] Piji Li et al. "Deep recurrent generative decoder for abstractive text summarization." In: *arXiv preprint arXiv:1708.00625* (2017). doi:https://doi.org/10.18653/v1/d17-1222.

[77] S Syed. "Abstractive Summarization of Social Media Posts: A case Study using Deep Learning." doi:https://api.semanticscholar.org/CorpusID:226304078. PhD thesis. Masters thesis, Bauhaus University, Weimar, Germany, 2017.

[78] Yusen Wang, Wenlong Liao, and Yuqing Chang. “Gated recurrent unit network-based short-term photovoltaic forecasting.” In: *Energies* 11.8 (2018). doi:https://doi.org/10.3390/en11082163, p. 2163.

[79] Horst Po

ttker. “News and its communicative quality: the inverted pyramidwhen and why did it appear?” In: *Journalism Studies* 4.4 (2003). doi:https://doi.org/10.1080/1461670032000136596, pp. 501–511.

[80] Chin-Yew Lin. “Rouge: A package for automatic evaluation of summaries.” In: *Text summarization branches out*. doi:https://aclanthology.org/W04-1013. 2004, pp. 74–81.