INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 7 • November 2026

A NOVEL MT5-BASED FRAMEWORK FOR ABSTRACTIVE TEXT SUMMARIZATION IN THE LOW-RESOURCE SINDHI LANGUAGE

Zuhaib Ali[*1], Kashif Ali[2], Asif Ali[3], Waqar Ali[4], Muhammad Sachal[5]

*[*1,2,3,4,5] Institute of Information & Communications Technologies (IICT), Mehran University of Engineering & Technology (MUET), Jamshoro, Pakistan*

[1]zuhaibalidahri2@gmail.com, [2] bhattikashifali57@gmail.com, [3]asifalimastoi2536@gmail.com, [4]waqaralibhatti19@gmail.com, [5]sahitosachal@gmail.com

DOI: https://doi.org/10.5281/zenodo.24098660

Keywords
Sindhi Text Summarization, Abstractive Summarization, mT5, Low-Resource NLP, Transformer-Based Models.

Article History
Received: 10 August 2026
Accepted: 24 October 2026
Published: 10 November 2026

Copyright @Author
Corresponding Author: *
Zuhaib Ali

Abstract

The rapid growth of digital textual information has increased the demand for automated summarization systems capable of generating concise, coherent, and semantically meaningful summaries. Although transformer-based abstractive summarization has achieved considerable success in high-resource languages, its application to low-resource languages remains constrained by limited annotated datasets and inadequate linguistic representation in multilingual pretrained models. Sindhi, despite its widespread use across South Asia, remains underexplored in abstractive summarization research. To address this gap, this study proposes an mT5-based abstractive text summarization framework specifically designed for the Sindhi language. A dedicated corpus comprising 6,500 article-summary pairs was constructed from multiple digital news sources and processed using language-specific normalization and tokenization techniques adapted to the Perso-Arabic Sindhi script. The proposed framework was fine-tuned and evaluated against established extractive summarization baselines, including LexRank and TextRank, using ROUGE, BERTScore, and qualitative human evaluation. Experimental results demonstrate that the proposed framework achieved ROUGE-1, ROUGE-2, and ROUGE-L scores of 0.51, 0.26, and 0.47, respectively, with a BERTScore F1 of 0.8975. The framework outperformed the evaluated extractive approaches in lexical and semantic evaluation. Human assessment further indicated its ability to generate fluent, coherent, and contextually relevant summaries through semantic abstraction and paraphrasing. These findings establish a foundation for Sindhi abstractive summarization and demonstrate the potential of multilingual transformer adaptation for low-resource natural language processing.

1. Introduction

Every day, news portals, archives, blogs and social platforms add text in quantities no reader can survey, and finding what matters has become correspondingly harder. Automatic text summarization (ATS) responds to that problem. Drawing on natural language processing (NLP) and information retrieval (IR), it condenses a document into a shorter text that keeps the essential meaning [1], [2].

Summarization research has moved through several distinct phases. The earliest systems were

statistical and heuristic: sentence salience was inferred from term frequency, position in the document and lexical weighting. Cheap to compute, but blind to anything beyond surface cues. Neural methods changed that by learning contextual representations from data, and transformers then replaced recurrence with self-attention, which made long-range dependencies tractable to model. Large language models, steered by instruction tuning and in-context learning, have since narrowed much of the remaining distance between machine-written and human-written summaries [3], [4].

Two paradigms dominate. Extractive methods rank source sentences by estimated importance and copy the top ones verbatim; factual accuracy tends to survive, but redundancy, weak cohesion and a rigid compression ratio come with the territory, since nothing is rewritten [5], [6]. Abstractive methods instead generate new sentences, paraphrasing and fusing information in a way that resembles how people actually summarize. The price is steeper. Factual consistency and fluency must be actively maintained, and the demand for training data is large [5].

Sequence-to-sequence learning with attention was the turning point for abstractive work, because it allowed an end-to-end mapping from document to summary. Transformers pushed further. By discarding recurrence they permitted parallel computation while still modeling context through self-attention [7], and summarization was one of many generation tasks to benefit.

Encoder–decoder transformers such as T5 and BART perform well on abstractive summarization, T5 in particular by casting every task as text-to-text. In that formulation each problem, summarization included, becomes sequence generation, so large-scale pretraining transfers across benchmarks once task-specific fine-tuning is applied [8]. mT5 carries the recipe over to 101 languages and supports cross-lingual transfer [9]. That breadth makes it a natural candidate when labeled data are thin or when the target language is not English.

Breadth of coverage is not depth of coverage, though. Pretraining corpora are heavily skewed, and a language such as Sindhi occupies only a sliver of them, which limits the quality of the representations learned and drags down downstream results [10], [11]. Models designed with such languages in mind, mBART and IndicBART among them, improve generation quality, yet gaps remain for structurally complex, under-resourced languages [11].

Sindhi is a prototypical low-resource language for NLP. Millions speak it across South Asia, yet standardized corpora, large annotated datasets and mature tooling are all in short supply [12], [13]. Its dominant script is Perso-Arabic, and that choice has consequences for preprocessing: orthographic variation, context-dependent letter joining and optional diacritics each complicate normalization and tokenization [12], [14].

Raw Sindhi text is not the problem; newspapers, books and websites produce plenty. What is missing is a publicly available, standardized corpus fit for abstractive summarization. Published work has concentrated on corpus building and preprocessing pipelines, and without benchmark data the higher-level tasks cannot advance. Sindhi's thin footprint in multilingual pretraining data compounds the difficulty for transformer models [9].

Prior Sindhi summarization work is extractive and leans on classical machine learning. Memon et al. [16], for example, combined TF-IDF ranking with sentence embeddings and obtained reasonable results in a limited setting. Selection-only methods cannot, by construction, produce rich or fully coherent summaries. A transformer-based abstractive approach that captures deeper context is therefore needed.

Nobody has yet examined abstractive summarization of Sindhi with a modern architecture such as mT5 in a systematic way. Evaluation is a second weak point. Existing Sindhi studies lean on lexical overlap and seldom use semantic measures like BERTScore, which suit abstractive output better [3]. The gap is twofold, spanning both modeling and evaluation.

Objective and Contributions

This study develops an abstractive summarization framework for Sindhi on top of mT5. It makes four contributions.

- The first mT5-based abstractive summarization framework designed for Sindhi.
- A large Sindhi news summarization dataset, built and cleaned with language-specific normalization and tokenization.
- An evaluation that pairs lexical (ROUGE) and semantic (BERTScore) metrics with human judgment.
- An analysis of the obstacles specific to Sindhi NLP: script complexity, token fragmentation and low-resource constraints in transformer learning.

Significance of the Study

The work has practical and scholarly value. In practice, it lets readers obtain concise summaries of Sindhi text and so eases access to a growing body of information. In research terms, it shows that a multilingual transformer can be adapted, through targeted preprocessing and fine-tuning, to a language with a structurally demanding script.

Some findings may carry over to other under-represented South Asian languages that share the same combination of data scarcity and script difficulty. The study also lays groundwork for later research on multilingual abstractive summarization.

2. Literature Review

2.1 Automatic Text Summarization

ATS has long been a core NLP task: compress a long document, keep what matters. Luhn [2] laid the computational groundwork by using term frequency to flag significant sentences. TF-IDF weighting and richer lexical scoring later made importance estimates sturdier on long documents. Graph-based ranking came next, and it improved extractive performance noticeably. Mihalcea and Tarau's TextRank [17], modeled on PageRank, treats sentences as nodes and weights the edges by inter-sentence similarity. LexRank [18] applies eigenvector centrality to a similarity graph so that global importance within the document determines which sentences are selected. Both work without supervision and are indifferent to language, which explains why they became default baselines.

Their weakness is a dependence on surface lexical overlap. Deeper semantic relations escape them. Redundant or poorly connected output is common, especially when the relevant information is spread across several sentences. That shortfall pushed the field toward semantically aware models.

2.2 Neural and Deep Learning-Based Approaches

Neural models changed the direction of the field. Sequence-to-sequence (Seq2Seq) architectures learned document-to-summary mappings end to end within an encoder–decoder frame. The first of these used recurrent neural networks (RNNs) and long short-term memory (LSTM) units, usually with attention to align source and target sequences.

Pointer-generator networks added a copying mechanism to neural generation and so handled rare and out-of-vocabulary words better [4]. Recurrence still imposed sequential computation and weak long-range modeling, though, so long documents remained difficult.

Self-attention removed those constraints. Replacing recurrence with attention permitted parallel computation and a more direct representation of global dependencies, and summarization quality rose accordingly.

2.3 Transformer-Based Summarization Models

Vaswani et al. [7] introduced the Transformer, now the base of most modern NLP systems. Multi-head self-attention lets it capture local and global context without sequential processing, which has paid off in machine translation and abstractive summarization alike.

Pretraining on large corpora then enabled transfer learning. BERT [19], a deep bidirectional encoder, improved contextual representations considerably and inspired BERTSUM [20], which adapts the encoder to extractive and abstractive

summarization through sentence-level and document-level encoding.

BART [21] and PEGASUS [4] went further with pretraining objectives built for generation. PEGASUS is the more pointed case: its Gap Sentence Generation (GSG) objective masks important sentences and trains the model to reconstruct them, which brings pretraining close to the summarization task itself.

T5 [8] unified NLP tasks as text-to-text generation, a flexible and scalable template for summarization. Its multilingual successor, mT5 [9], spans 101 languages and shows strong cross-lingual transfer, which is why it suits low-resource settings.

Taken together, these models have raised fluency, coherence and semantic adequacy and set new state-of-the-art results on several benchmarks.

2.4 Low-Resource Language NLP Challenges

Progress in multilingual NLP has not erased the low-resource problem. Annotated data are limited and pretraining coverage is uneven. Multilingual transformers also tend to devote disproportionate capacity to high-resource languages, so the underrepresented ones suffer.

Sindhi adds script-specific trouble to morphological richness. The Perso-Arabic script brings inconsistent Unicode representations, ligature formation and variable diacritic use, all of which degrade subword segmentation during tokenization [14]. Tokenizer instability is thus a structural issue, not an incidental one.

Supervised fine-tuning is also constrained by the shortage of large annotated corpora for tasks such as summarization and translation. XL-Sum [5] offers multilingual coverage, yet Sindhi is absent or very thinly represented, which limits how far existing benchmarks can be used.

2.5 Sindhi Language Processing and Existing Work

Sindhi NLP trails the other major South Asian languages. Early work centered on corpora and basic preprocessing. Dootio and Wagan [12] produced one of the first structured Sindhi corpora, drawn from books and newspapers, and documented the language's linguistic and morphological complexity.

Preprocessing frameworks followed. Nawaz et al. [14] proposed TPTS, which combines normalization, tokenization, stop-word removal and stemming adapted to the characteristics of Sindhi script. Both studies stress that language-specific preprocessing is essential to downstream performance.

Summarization work is confined to extractive methods. Memon et al. [16] built a TF-IDF and embedding-based extractive framework that performed moderately and exposed limits in semantic understanding and abstraction. They explicitly recommended transformer-based abstractive models as the next step.

Most Sindhi NLP studies deal with foundational tasks; advanced generative modeling is still at an early stage.

2.6 Research Gap Analysis

Sindhi remains underexplored for abstractive summarization despite recent multilingual transformers. Studies so far address corpus construction, preprocessing and extractive summarization [12], [14], [16].

mT5 [9] and IndicBART [11] generalize well across languages in general. For Sindhi the picture differs: pretraining coverage is thin and task-specific fine-tuning resources are lacking, so their effectiveness is limited.

Evaluation is a further weakness. Sindhi summarization studies rely mostly on overlap-based measures and omit contextual similarity metrics such as BERTScore [3], even though abstractive systems need them.

A gap therefore exists: no transformer-based abstractive summarizer designed for Sindhi, backed by robust evaluation and language-specific preprocessing.

The mT5-based framework proposed here fills it. It offers a fully neural abstractive pipeline for Sindhi, assessed against strong extractive baselines and modern semantic metrics.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 7 • November 2026

3. Methodology

3.1 Proposed Framework Overview

This study fine-tunes the multilingual Text-to-Text Transfer Transformer (mT5) for abstractive summarization of Sindhi. The setting is low-resource: annotated corpora and linguistic tools are scarce. The framework therefore pairs Sindhi-specific preprocessing with multilingual fine-tuning to produce summaries that are semantically and contextually coherent.

Five stages make up the workflow: (i) dataset collection, (ii) text preprocessing and normalization, (iii) tokenization and dataset preparation, (iv) model fine-tuning, and (v) summary generation and evaluation.

Fig. 3.1 shows the architecture.

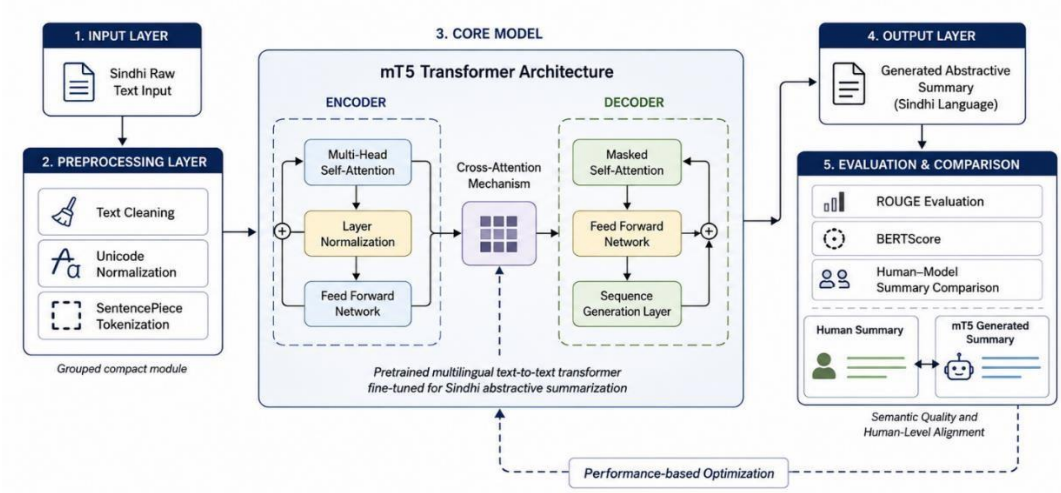


Fig. 3.1 Proposed mT5-based abstractive summarization framework for the Sindhi language.

3.2 Dataset Collection and Construction

No public large-scale dataset for Sindhi abstractive summarization exists, so one was built. It holds 6,500 pairs of Sindhi news articles and human-written summaries, gathered from several online Sindhi news platforms and digital archives.

To obtain domain diversity and linguistic variability, the data span news, education and general public information. Manual filtering removed duplicates, incomplete records and noisy text.

An 80:10:10 ratio divided the data into training, validation and test sets (Table 3.1).

Table. 3.1 Dataset Distribution

Split	Samples	Percentage
Training Set	5200	80%
Validation Set	650	10%
Testing Set	650	10%
Total	6500	100%

3.3 Sindhi Text Preprocessing

Sindhi's morphological richness and Perso-Arabic script call for a dedicated preprocessing pipeline, which was built to improve data consistency and model performance [14].

3.3.1 Noise Removal

Noise removal stripped HTML tags, URLs, inconsistent punctuation, special symbols and redundant whitespace. Cleaner input improved tokenization quality.

3.3.2 Unicode Normalization

Sindhi script offers several Unicode encodings for characters that look alike. Such inconsistency harms tokenization and semantic representation. Normalization along the lines of earlier Sindhi preprocessing research [14] was therefore applied, and it covered the following:

- Standardizing character variants
- Removing unnecessary diacritics
- Handling elongated characters
- Representing Arabic-script symbols consistently

The effect was fewer fragmented tokens and a more consistent vocabulary during transformer tokenization.

3.3.3 Tokenization

The mT5 architecture ships with a SentencePiece tokenizer, and the preprocessed text was tokenized with it [9]. Subword segmentation lets the model handle rare and unseen words.

Sindhi is thinly represented in multilingual vocabularies, however, so morphologically rich words frequently split into fragments. Preprocessing quality therefore bears directly on token quality and contextual representation.

3.3.4 Stop-Word Handling

Extractive pipelines typically discard stop-words. Abstractive transformers cannot afford to, since they need complete sentence structure to preserve semantic and contextual relations. Stop-words were therefore retained to protect grammatical coherence and flow. Only noisy and irrelevant tokens were removed.

3.4 mT5-Based Summarization Model

The framework rests on mT5, a multilingual encoder–decoder model trained on 101 languages [9]. Summarization is cast as text-to-text generation.

Given an input document D, the model generates a summary S:

$$S = M_\theta(D)$$

3.4.1 Encoder

The encoder maps the tokenized input sequence $X = (x_1, x_2, \ldots, x_n)$into contextual representations:

$$H = Encoder(X)$$

where Hcaptures long-range semantic dependencies using multi-head self-attention [7].

3.4.2 Decoder

The decoder produces the summary token by token, conditioning on the encoder representations and on the tokens already generated.

The conditional probability of generating a summary sequence Ygiven an input document $X$is defined as:

$$P = (Y \mid X) = \prod_{t=1}^{m} P(y_t \mid y_{<t}, X)$$

where:

- X represents the input document
- Y represents the generated summary
- $y_t$denotes the generated token at time step t
- $y_{<t}$represents all previously generated tokens

Autoregressive generation of this kind is what lets the model produce fluent, contextually coherent summaries.

Algorithm 1: Proposed mT5-Based Sindhi Summarization Framework

Input: Sindhi document D

Output: Summary S

1. Clean raw document $D$by removing noise and irrelevant symbols
2. Apply Unicode normalization to obtain D'

3. Tokenize D' using SentencePiece tokenizer
4. Encode tokens using mT5 encoder to obtain hidden states H
5. Generate summary using autoregressive decoding with beam search
6. Post-process generated tokens to form final summary S
7. Return S

3.5 Model Fine-Tuning

The pre-trained mT5 model [9] was fine-tuned on the prepared Sindhi corpus so that it could adapt to Sindhi linguistic structure and to summarization patterns.

Each training example followed the text-to-text format of T5 [8]:

Input:

summarize: <Sindhi article>

Target:

<Reference summary>

Training minimized the cross-entropy loss between generated and reference summaries.

The loss function is defined as:

$$L = -\sum_{t=1}^{m} \log P(y_t \mid y_{<t}, X)$$

where L denotes the total training loss.

3.5.1 Training Configuration

Table. 3.2 Training Configuration

Parameter	Value
Model	mT5-base
Batch Size	8
Learning Rate	3×10^{-5}
Optimizer	AdamW
Number of Epochs	10
Maximum Input Length	512
Maximum Output Length	128
Decoding	Beam Search (k=4)
Loss Function	Cross-Entropy
Hardware	NVIDIA Tesla T4 GPU

Gradient clipping and early stopping kept training stable and guarded against overfitting.

3.6 Summary Generation

After fine-tuning, the model generated abstractive summaries for unseen Sindhi news articles, with beam search decoding used to improve fluency and semantic consistency.

Generation proceeded as follows:

- Encoding the input document
- Predicting output tokens sequentially
- Evaluating candidate token sequences
- Selecting the most probable summary sequence

Beam search yields more coherent and contextually accurate summaries than greedy decoding.

3.7 Evaluation Metrics

Evaluation combined lexical overlap metrics with semantic similarity metrics to give a rounded assessment of summary quality.

3.7.1 ROUGE Evaluation

ROUGE, a standard summarization metric, measures lexical overlap between generated and reference summaries.

Three variants were used:

- ROUGE-1
- ROUGE-2
- ROUGE-L

ROUGE-1 counts unigram overlap, ROUGE-2 bigram overlap, and ROUGE-L the longest common subsequence.

3.7.2 BERTScore Evaluation

ROUGE captures lexical similarity well but can miss semantic equivalence in abstractive output. BERTScore, proposed by Zhang et al. [3], was therefore added as a semantic measure.

It compares contextual transformer embeddings of generated and reference summaries to estimate semantic correspondence.

That suits abstractive summarization, because a correct summary may use different wording for the same meaning.

3.7.3 Human Evaluation

Human evaluation complemented the automatic metrics. Raters judged four qualities:

- Fluency
- Readability
- Coherence
- Semantic correctness

Raters compared generated summaries with the references to assess practical usability and contextual consistency.

3.8 Experimental Environment

Experiments used PyTorch and Hugging Face Transformers, with NLTK and Scikit-learn as supporting libraries. Training ran on Google Colab Pro with an NVIDIA Tesla T4 GPU.

3.9 Methodological Contribution

Methodologically, the study offers a complete end-to-end pipeline for Sindhi abstractive summarization on a multilingual transformer. It shows how mT5 can be adapted to a low-resource language through targeted preprocessing, fine-tuning and evaluation.

Earlier Sindhi NLP studies did not pair lexical with semantic evaluation; this one does, and it offers a reproducible pipeline for later low-resource generation research.

4. Results

4.1 Training Convergence and Learning Behavior

Training and validation loss curves across epochs describe the learning dynamics. Optimization was stable throughout fine-tuning, which indicates that the multilingual model adapted to the Sindhi task.

Loss started high. The pre-trained model had seen little Sindhi text and few Sindhi script structures. As training advanced, it learned contextual and semantic relations from the corpus: training loss fell steadily from 2.10 in the first epoch to roughly 0.76 in the last, and validation loss fell from 2.20 to 0.96, a sign of good generalization to unseen data.

The convergence trend suggests that the preprocessing choices, Unicode normalization and Sindhi-specific token handling among them, helped stabilize learning. The narrow gap between the two curves argues against severe overfitting despite the small dataset.

Most of the reduction in loss came in the first five epochs. After Epoch 7 both curves flattened, which points to near-optimal convergence.

Multilingual transfer learning therefore works for Sindhi abstractive summarization, and transformers can adapt efficiently under limited resources.

International Journal of Advanced Research

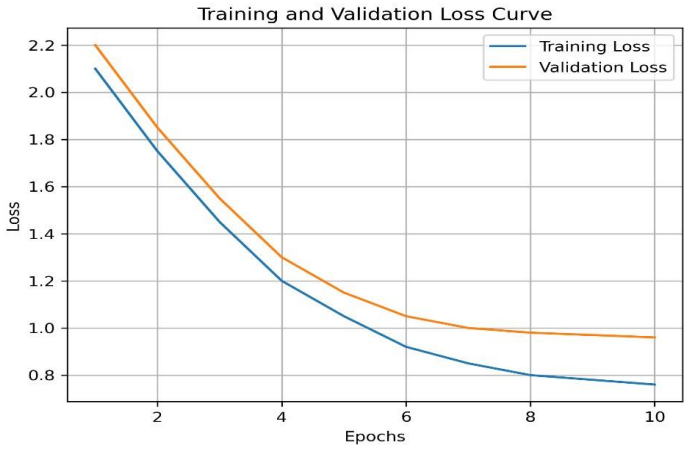


Fig.4.1 Training and Validation Loss Across Epochs

4.2 ROUGE Performance Across Epochs

Fig. 4.2 traces the ROUGE metrics through fine-tuning. ROUGE-1, ROUGE-2 and ROUGE-L all rose from epoch to epoch, indicating gains in lexical overlap, contextual understanding and semantic learning.

Early scores were lower because the multilingual model had not yet adapted to Sindhi structure and summarization patterns. Continued fine-tuning let it learn contextual relations from the corpus. ROUGE-1 climbed from 0.32 at Epoch 1 to 0.51 at Epoch 10, showing better retention of key content in the generated summaries.

ROUGE-2 rose from 0.11 to 0.26, a sign of stronger phrase-level representation and better semantic composition. ROUGE-L went from 0.28 to 0.47, which suggests improved sentence-level coherence and better handling of long-range dependencies.

All three variants improve in the same direction, supporting the case for multilingual transformer adaptation in low-resource Sindhi summarization. Scores were relatively stable after Epoch 8, so the model appears to have reached near-optimal performance without serious overfitting.

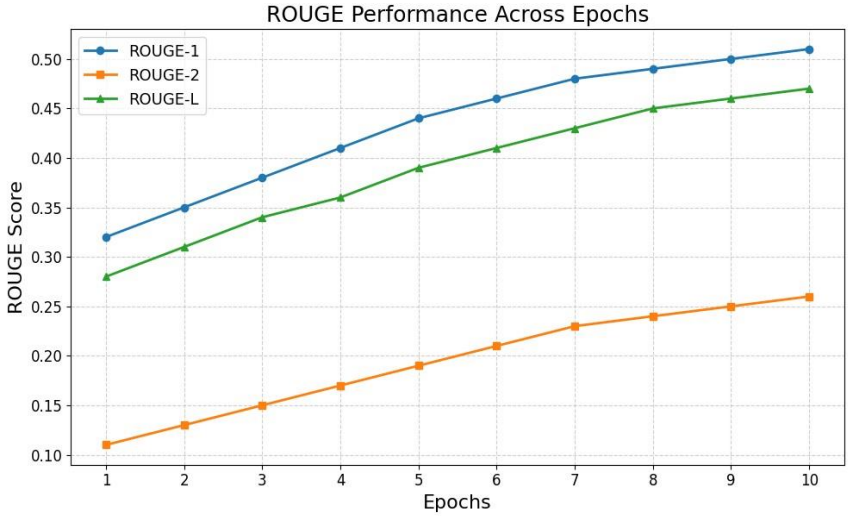


Fig. 4.2. ROUGE-1, ROUGE-2, and ROUGE-L performance across training epochs for the proposed mT5-based Sindhi summarization framework.

4.3 ROUGE-Based Comparative Evaluation

TextRank and LexRank served as baselines for lexical overlap and content preservation, compared on ROUGE-1, ROUGE-2 and ROUGE-L (Table 4.1 and Fig. 4.3).

Table 4.1 Comparative ROUGE Performance of Summarization Models

Model	ROUGE-1	ROUGE-2	ROUGE-L
TextRank	0.41	0.17	0.36
LexRank	0.43	0.19	0.38
Proposed mT5	**0.51**	**0.26**	**0.47**

The proposed mT5 framework beat both extractive baselines on every ROUGE variant. Its ROUGE-1 of 0.51 compares with 0.43 for LexRank and 0.41 for TextRank, so the abstractive model preserved more unigram-level information from the source.

ROUGE-2 tells a similar story: 0.26, clearly above the baselines. Because ROUGE-2 rewards bigram overlap and phrase preservation, the gain suggests more contextually meaningful and coherent summaries.

At 0.47, ROUGE-L points to better sentence-level coherence and structural consistency. Extractive methods only select existing sentences. The mT5 model reformulated content, preserved long-range dependencies and cut redundancy.

LexRank edged out TextRank, probably because its graph-centrality ranking measures sentence importance more effectively. Neither extractive method went far in semantic abstraction or contextual restructuring.

For Sindhi, then, transformer-based abstractive summarization offers substantial gains over traditional graph-based extraction.

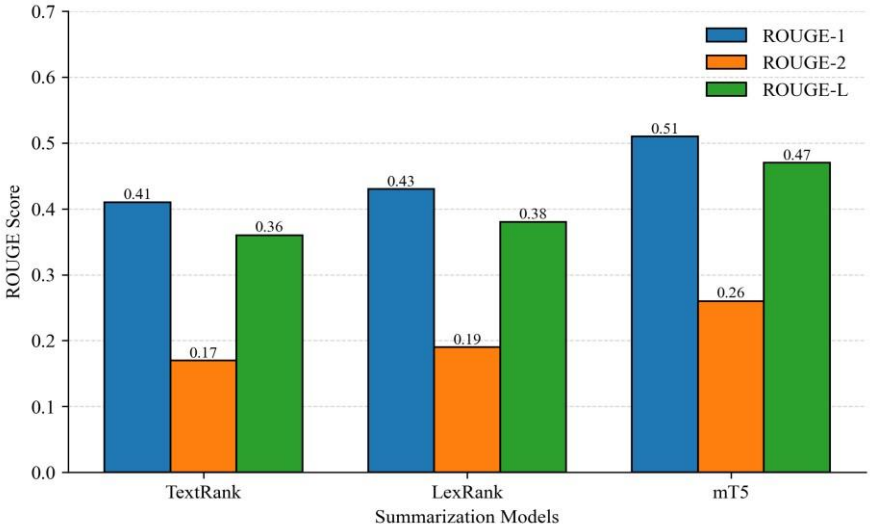


Fig. 4.3. Comparative ROUGE performance of TextRank, LexRank, and the proposed mT5 framework.

4.4 Semantic Similarity Evaluation Using BERTScore

ROUGE measures lexical overlap and cannot fully judge semantic equivalence between generated and reference summaries. BERTScore, which compares contextual transformer embeddings, was used to fill that gap.

Table 4.2 Comparative BERTScore (F1) Performance

Model	BERTScore (F1)
TextRank	0.8008
LexRank	0.8250
Proposed mT5	**0.8975**

As Table 4.2 and Fig. 4.4 show, the proposed mT5 framework scored highest on semantic similarity: a BERTScore (F1) of 0.8975 against 0.8008 for TextRank and 0.8250 for LexRank.

That margin shows the abstractive model capturing contextual meaning instead of leaning on lexical overlap. Because mT5 generates through contextual sequence generation, it can paraphrase and restructure while keeping the original intent. Extractive methods can only select sentences from the source. Compact, naturally flowing summaries are beyond them, and their semantic agreement with human-written summaries is lower as a result.

A high BERTScore indicates that multilingual transformer adaptation can learn useful semantic representations for a low-resource language such as Sindhi even with limited training data.

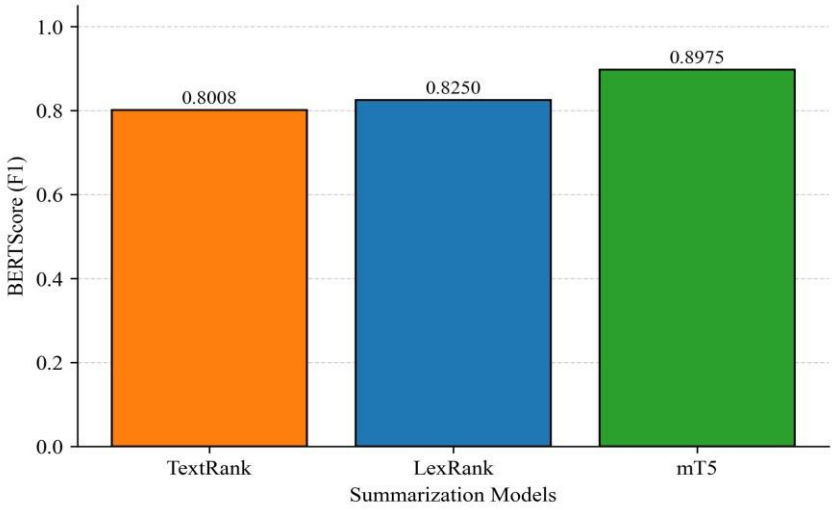


Fig. 4.4. Comparative BERTScore (F1) performance of summarization models.

4.5 Qualitative Human Evaluation

Human evaluation complemented the automatic metrics. Raters compared generated summaries against the references and scored them on a five-point Likert scale across the criteria in Table 4.3.

Table 4.3 Human Evaluation Results

Evaluation Criterion	Score
Relevance	4.7
Coherence	4.5
Fluency	4.4
Overall Quality	**4.5**

Raters found the proposed framework's summaries concise, informative and coherent. Readability and flow were better than in extractive output, which often showed abrupt transitions and redundancy.

Fluency scores suggest that the model picked up Sindhi sentence structure and generation patterns despite scarce resources. The relevance score shows that the core content of the source documents survived.

Raters also noted that the summaries felt more natural because the model combined and reformulated information across sentences instead of copying segments.

Human judgments back up the ROUGE and BERTScore findings.

4.6 Qualitative Analysis of Generated Summaries

A qualitative comparison of LexRank, TextRank and the mT5 model followed (Table 4.4). It sets the original Sindhi document and its human-written reference beside each system's output and asks about semantic preservation, contextual relevance, fluency and coherence.

Table 4.4 Comparative Analysis of Summaries Generated by Different Approaches

Type	Content
Input Text	سنڌ ۾ موسمي تبديلين تيزي سان وڌي رهيون آهن، جنهن سبب زرعي پيداوار متاثر ٿي رهي آهي ۽ پاڻي جي کوٽ جا مسئلا پڻ وڌي رهيا آهن. مختلف علائقن ۾ گرمي پد ۾ اضافو، برساتن جي غير متوازن نظام، ماحولياتي تبديلين ۽ زندگي تي اثرانداز ٿي رهيا آهن. ماهرن موجب جيڪڏهن حڪومت، سماجي ادارا، مقامي برادريون گڏيل حڪمت عملي اختيار نه ڪنديون ته مستقبل ۾ خوراڪ، پاڻي، ۽ ماحولياتي استحڪام جا مسئلا وڌيڪ سنگين ٿي سگهن ٿا
Reference Summary	سنڌ ۾ موسمي تبديلين زرعي پيداوار، پاڻي جي کوٽ، ماحولياتي نظام تي وڏو اثر وجهي رهيون آهن. حڪومت ۽ سماجي ادارن کي گڏيل حڪمت عملي اختيار ڪرڻ جي ضرورت آهي
LexRank Summary	سنڌ ۾ موسمي تبديلين تيزي سان وڌي رهيون آهن، پاڻي جي کوٽ جا مسئلا وڌي رهيا آهن
TextRank Summary	مختلف علائقن ۾ گرمي پد ۾ اضافو، ماحولياتي تبديلين عام زندگي تي اثرانداز ٿي رهيون آهن
mT5 Generated Summary	سنڌ ۾ موسمي تبديلي زرعي نظام ۽ پاڻي جي وسيلن تي اثرانداز ٿي رهي آهي، جنهن کي منهن ڏيڻ لاءِ گڏيل منصوبابندي ضروري آهي

The two approaches differ clearly. LexRank and TextRank copied sentences straight from the source: important information survived, but compression was limited and semantic integration weak. Factual content was retained, yet the summaries read as rigid and less coherent, since they depended on selection and not on generation.

The mT5 summary was more natural and more condensed. It paraphrased and reorganized the source and brought the central ideas, climate change, agricultural impact and water scarcity, into one concise statement.

Fluency and readability were also stronger than in the extractive outputs. Redundancy dropped and coherence improved while meaning was preserved. This agrees with the ROUGE, BERTScore and human results reported earlier.

In the low-resource Sindhi setting, the mT5 framework produces summaries that are more human-like and more informative than those of LexRank and TextRank.

5. Discussion

Transformer-based abstractive summarization can be adapted to Sindhi despite severe resource limits. The mT5 framework outperformed LexRank and TextRank on lexical, semantic and human evaluation, supporting the use of multilingual transformers for Sindhi.

Multilingual transfer learning appears to help low-resource languages considerably when language-specific preprocessing accompanies it. Training

and validation loss fell steadily without severe overfitting, so the model learned contextual and semantic relations from the corpus. Unicode normalization and script-specific preprocessing probably reduced token fragmentation and gave more consistent representations. Earlier multilingual work likewise stresses preprocessing for morphologically rich, script-diverse languages [9], [14].

ROUGE results confirm the advantage over graph-based extraction. LexRank and TextRank achieved reasonable overlap, but extraction constrains them: they select existing sentences, and so compress poorly and lose coherence across sentence boundaries. The mT5 framework condensed content through abstraction and paraphrase, which is reflected in its higher ROUGE-1, ROUGE-2 and ROUGE-L.

BERTScore adds evidence. The large gain over the extractive baselines indicates deeper contextual representations than overlap alone could show. ROUGE counts n-gram similarity; BERTScore captures semantic correspondence through contextual embeddings. A high BERTScore therefore means meaning was kept while paraphrase and restructuring remained free.

Human evaluation points the same way. Raters judged the summaries fluent, coherent and informative, with better readability and flow than the extractive outputs. Merging information across sentences and paraphrasing it is exactly what abstractive transformers offer over sentence extraction.

The qualitative comparison sharpens the contrast. Extractive baselines kept facts but produced rigid, partly redundant text, because they only selected sentences. The mT5 framework condensed the main concepts into concise, human-like summaries and kept contextual fidelity, which suggests semantic abstraction rather than memorized sentence patterns.

Limitations remain. Sindhi is heavily underrepresented in multilingual pretraining corpora, which caps the quality of its learned representations relative to high-resource languages. No large public Sindhi benchmark exists, which restricts comparison and reproducibility. And morphologically rich words and script-level ambiguity still cause tokenization trouble after normalization.

Even so, multilingual transformers look like a workable foundation for NLP applications in under-represented languages. The framework advances Sindhi abstractive summarization and may inform other low-resource South Asian languages with similar linguistic and computational constraints.

In low-resource Sindhi settings, transformer-based abstractive summarization improves semantic quality, contextual coherence and human readability over traditional extraction.

REFERENCES

Y. Zhang, H. Jin, D. Meng, J. Wang, and J. Tan, "A comprehensive survey on automatic text summarization with exploration of LLM-based methods," Neurocomputing, vol. 663, p. 131928, 2025.

H. P. Luhn, "The automatic creation of literature abstracts," IBM J. Res. Develop., vol. 2, no. 2, pp. 159–165, 1958.

T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in Proc. Int. Conf. Learn. Represent. (ICLR), 2020.

J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "PEGASUS: Pre-training with extracted gapsentences for abstractive summarization," in Proc. Int. Conf. Mach. Learn. (ICML), 2020.

T. Hasan et al., "XL-Sum: Large-scale multilingual abstractive summarization for 44 languages," in Findings Assoc. Comput. Linguistics: ACL-IJCNLP, 2021.

M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: A survey," Artif. Intell. Rev., vol. 47, no. 1, pp. 1–66, 2016.

A. Vaswani et al., "Attention Is All You Need," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2017.

C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," J. Mach. Learn. Res., vol. 21, pp. 1–67, 2020.

L. Xue et al., "mT5: A massively multilingual pre-trained text-to-text transformer," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol. (NAACL-HLT), 2021.

Y. Liu et al., "Multilingual denoising pre-training for neural machine translation," Trans. Assoc. Comput. Linguistics, vol. 8, pp. 726–742, 2020.

R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. M. Khapra, and P. Kumar, "IndicBART: A pre-trained model for Indic natural language generation," in Findings Assoc. Comput. Linguistics (ACL), pp. 1849–1863, 2022.

M. A. Dootio and A. I. Wagan, "Development of Sindhi text corpus," J. King Saud Univ. Comput. Inf. Sci., vol. 33, no. 4, pp. 468–475, 2021.

M. A. Mahar and G. Q. Memon, "Challenges and opportunities in Sindhi natural language processing: A survey," IEEE Access, vol. 10, pp. 1–12, 2022.

A. N. A. Nawaz, M. Nawaz, N. A. Shaikh, S. Rajper, J. Baber, and M. Khalid, "TPTS: Text pre-processing techniques for Sindhi language," Pak. J. Emerg. Sci. Technol., vol. 4, no. 3, 2023.

A. Faheem, F. Ullah, M. S. Ayub, and A. Karim, "UrduMASD: A multimodal abstractive summarization dataset for Urdu," in Proc. Int. Conf. Lang. Resour. Eval. Comput. Linguistics (LREC-COLING), 2024, pp. 17245–17253.

A. Memon, Z. Memon, and A. H. Jalbani, "Extractive text summarization-based framework for Sindhi language," Int. J. Innov. Sci. Technol., vol. 7, no. 6, pp. 147–155, 2025.

R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2004, pp. 404–411.

G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," J. Artif. Intell. Res., vol. 22, pp. 457–479, 2004.

J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol. (NAACL-HLT), Minneapolis, MN, USA, 2019, pp. 4171–4186.

Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in Proc. Conf. Empirical Methods Natural Lang. Process. Int. Joint Conf. Natural Lang. Process. (EMNLPIJCNLP), Hong Kong, China, 2019, pp. 3730–3740.

M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL), 2020, pp. 7871–7880.