Text Summarization using Pre-Trained Models on Tamil, English, Gujarati and Bengali

Tanisha Sriram[*,†], Ananya Raman[†], Sowmya Anand[†] and Durairaj Thenmozhi[†]

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, Chennai-603110

Abstract

This paper explores machine learning models for the Indian Language Summarization (ILSUM 2024) shared task, with a specific focus on generating summaries from news articles in four languages: Bengali, English, Gujarati, and Tamil. Representing team "SynopSizers" in this task, we addressed the gap of underrepresented Indian languages in NLP, particularly in text summarization. Though there is an abundant availability of large-scale datasets for languages like English and French, there has been a severe underrepresentation of NLP modelling of Indian languages, specifically in the field of text summarization. The central aim is to address this gap and narrow it. A key challenge of this process was the presence of code-mixing and script-mixing, where English phrases and Latin scripts were embedded in articles written in Indian languages. Popular English-trained models struggled with these challenges and hence required the use of multilingual models. Several models were tested and trained during the process. The models were evaluated using standard ROUGE metrics. Among the models tested, an extractive frequency-based model demonstrated the most consistent performance across all languages.

Keywords

Indian Languages, Automatic Text Summarization, Article Summarization, Bengali, English, Gujarati, Tamil

1. Introduction

In recent years, Natural Language Processing (NLP) has seen huge leaps, transforming how we interact and understand text-based data. It has integrated itself into the way we learn and process, from basic tokenization to more complex processes like detecting hate speech, retrieving and summarizing legal documents, analyzing sentiment, and identifying fake news [1], to name a few. The accuracy of machines imitating humans has reached a scarily stunning level [2]. And, with the sheer volume of digital content, be it social media, magazines, or even newspapers, NLP plays an important role in language comprehension as well. Thus, NLP models play an important role in text summarization, which focuses on distilling large amounts of information into summaries [3]. This allows human readers to grasp concepts briefly and concisely.

Extensive research and development have gone into languages like English, Chinese, German, French, and Spanish, having large-scale datasets and advanced models [4]. Unfortunately, the same cannot be said for Indian languages — very little attention has been given to these languages. Despite the millions who speak these languages, efforts in creating effective NLP tools for them, particularly for Automatic Text Summarization (ATS) [5], remain scarce. Most available datasets are either too small or inaccessible to the public, limiting their utility for meaningful research and development [1, 6].

In an attempt to narrow this chasm, the Indian Language Summarization (ILSUM) shared task was initiated. For the ILSUM 2024 edition, the dataset (publicly available corpora specifically for summarization) has been compiled from leading national newspapers and features more than 15,000 article-headline pairs for each language, including Bengali, English, Gujarati, and Tamil. However, these datasets contain the presence of code-mixing and script-mixing, where English phrases and Latin scripts

Forum for Information Retrieval Evaluation, December 12-15, 2024, India

[*]Corresponding author.

[†]These authors contributed equally.

tanisha2310538@ssn.edu.in (T. Sriram); ananya2310278@ssn.edu.in (A. Raman); sowmya2310543@ssn.edu.in (S. Anand); theni\TU_d@ssn.edu.in (D. Thenmozhi)

0009-0009-6936-4316 (T. Sriram); 0009-0003-7762-2756 (A. Raman); 0000-0001-7116-9338 (S. Anand)

are interwoven with Indian-language content [7, 8]. Tackling these challenges requires a nuanced approach to the rich linguistic diversity that Indian languages represent.

This research aims to foster the development of NLP tools that can handle the complexities of multilingual and code-mixed content, thus making an attempt to pave the way for more inclusive and wide-reaching innovations in the field of natural language processing.

In recent years, Natural Language Processing (NLP) has seen huge leaps, transforming how we interact and understand text-based data. It has integrated itself into the way we learn and process, from basic tokenization to more complex processes like detecting hate speech, retrieving and summarizing legal documents, analyzing sentiment, and identifying fake news [1], to name a few. The accuracy of machines imitating humans has reached a scarily stunning level [2]. And, with the sheer volume of digital content, be it social media, magazines, or even newspapers, NLP plays an important role in language comprehension as well. Thus, NLP models play an important role in text summarization, which focuses on distilling large amounts of information into summaries [3]. This allows human readers to grasp concepts briefly and concisely.

Extensive research and development have gone into languages like English, Chinese, German, French, and Spanish, having large-scale datasets and advanced models [4]. Unfortunately, the same cannot be said for Indian languages — very little attention has been given to these languages. Despite the millions who speak these languages, efforts in creating effective NLP tools for them, particularly for Automatic Text Summarization (ATS) [5], remain scarce. Most available datasets are either too small or inaccessible to the public, limiting their utility for meaningful research and development [1, 6].

In an attempt to narrow this chasm, the Indian Language Summarization (ILSUM) shared task was initiated. For the ILSUM 2024 edition, the dataset (publicly available corpora specifically for summarization) has been compiled from leading national newspapers and features more than 15,000 article-headline pairs for each language, including Bengali, English, Gujarati, and Tamil. However, these datasets contain the presence of code-mixing and script-mixing, where English phrases and Latin scripts are interwoven with Indian-language content [7, 8]. Tackling these challenges requires a nuanced approach to the rich linguistic diversity that Indian languages represent.

This research aims to foster the development of NLP tools that can handle the complexities of multilingual and code-mixed content, thus making an attempt to pave the way for more inclusive and wide-reaching innovations in the field of natural language processing.

2. Related Works

The following were some of the research papers that were referred while involving in the task.

Text summarization for Indian languages [9] paper by Aishwarya Krishnakumar et al. explores the evolution of text summarization, from ancient uses to modern NLP models. While summarization is advanced for English, Indian languages are underrepresented. The authors, participating in the FIRE 2022 ILSUM task, address this gap by comparing models like mT5_m2m_CrossSum, XL-Sum, and Bert for code-mixed text summarization in English, Gujarati, and Hindi. They found that mT5_m2m_CrossSum produced the most accurate summaries, earning a top-ten validation set ranking for each language. This work highlights the effectiveness of mT5-based models for multilingual summarization in Indian languages.

A paper on text summarization techniques by Allahyari et al. (2017) [10] provides a comprehensive review of automatic text summarization techniques, addressing the growing need for concise representations of vast text data from the Internet and other digital sources. The authors examine a range of summarization methods, particularly focusing on extractive approaches for both single- and multi-document summarization. These methods include topic modeling, frequency-based strategies, graph-based approaches, and machine learning techniques, each evaluated for their effectiveness and limitations in different contexts. The paper emphasizes the challenges in automatic summarization due to the lack of human-like language understanding in machines and highlights significant advancements and trends in the field, offering a valuable state-of-the-art overview of summarization technology.

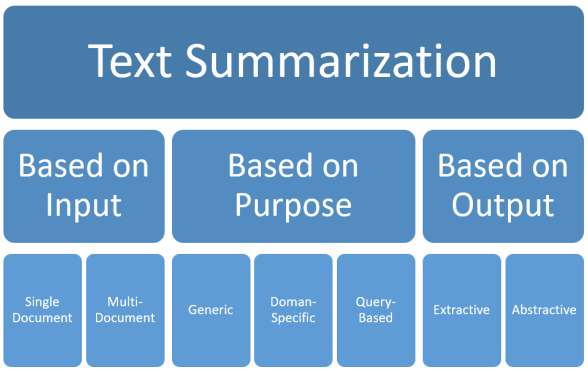

Figure 1: Types of Text Summarization.

Hahn and Mani (2000) [11] explored the complexities of creating coherent summaries from diverse sources, given the explosion of online information in their paper. Existing extraction-based tools like Microsoft's AutoSummarize are limited in coherence and scope. The authors discuss knowledge-poor and knowledge-rich methods—basic rules versus extensive background knowledge—to enhance summary quality. Summaries are classified as extracts or abstracts, with functions such as indicative, informative, or critical, and a growing focus on user-specific needs. They highlight key challenges, including summarizing non-textual media, multiple sources, and achieving high compression rates, essential for advancing summarization tools.
Awasthi et al. (2021) [12] provided an overview of extractive and abstractive methods in automatic text summarization in their paper on natural language processing. They emphasized unsupervised extractive approaches, including K-Means clustering for sentence selection and the SummCoder framework, which ranks sentences based on relevance and novelty. The study also discusses EdgeSumm, a graph-based method using nouns as nodes for text representation. This work highlights the need for effective summarization techniques to manage the growing volume of online information and the critical role of NLP in advancing these methods.

3. Exploration on Summarization

Understanding the types of text summarization is crucial before delving into Natural Language Processing (NLP) for several reasons. Different summarization types (extractive vs. abstractive) require distinct approaches and algorithms. By understanding these differences, we can choose the most suitable models and techniques for their specific needs, leading to more effective and efficient NLP solutions. The different types of text summarization is represented in Figure 1.

3.1. Based on Output

The table presents the distribution of data across training, validation, and test sets for four languages—Bengali, English, Gujarati, and Tamil. It highlights the number of records allocated to each phase for each language, providing insight into the dataset's structure for model training, hyperparameter tuning, and performance evaluation in a multilingual context.

4. Task Description and Dataset

The aim of the task is to generate a meaningful fixed length summary, either extractive or abstractive, for each article. The dataset for this task is built using articles and headline pairs from several leading newspapers of the country. The Table 1 presents the distribution of data across training, validation, and test sets for four languages—Bengali, English, Gujarati, and Tamil. It highlights the number of records

Table 1
Details of Dataset.

	No. Train Records	No. Val Records	No. Test Records
Bengali	12400	2950	2206
English	9376	1500	2500
Gujarati	33630	2999	1457
Tamil	4109	456	1955

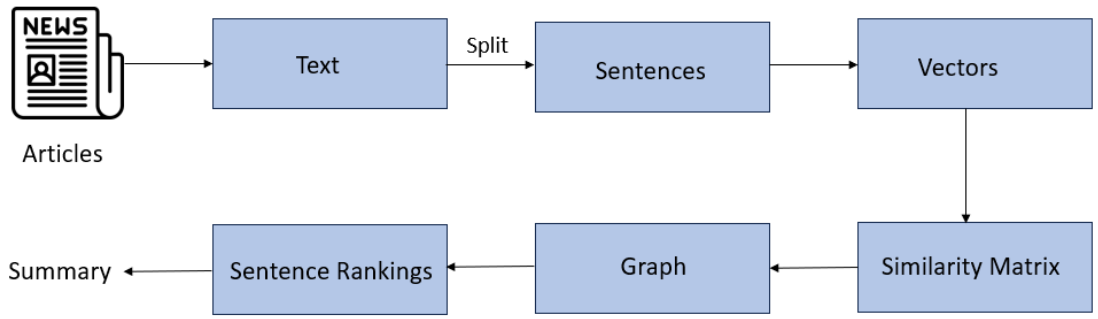

Figure 2: Flow of Text Summarization.

allocated to each phase for each language, providing insight into the dataset's structure for model training and performance evaluation in a multilingual context. The train and val dataset contained id, Heading, Summary, and Article for each language, whereas the test dataset contained id, Heading and Article alone. More details about the dataset is presented in Table 1. The overview of the task can be found in Findings of the First Shared Task on Indian Language Summarization (ILSUM): Approaches, Challenges and the Path Ahead [13] and FIRE 2022 ILSUM Track: Indian Language Summarization [14]. More details on the dataset and additional documents [15], [16], [17], [18] were also referred.

5. Methodology

The articles are split into individual sentences. Each sentence is transformed into a vector representation, and a similarity matrix is created by comparing the vectors. This matrix forms the basis of a graph where sentences are nodes and edges represent sentence similarity. A ranking algorithm, such as PageRank, is applied to the graph to rank the sentences based on their importance. Finally, the highest-ranked sentences are selected to create a concise summary of the original text. The basic flow of Summarization is given in Figure 2.

5.1. Pre-processing

We used proper pre-processing of the text data in different languages- Bengali, Tamil, Gujarati and English. The quality and uniformity required for efficient summarization were preserved in our experiment. The raw text files are read into binary mode in order to anticipate encoding problems when reading text. The content was decoded using UTF-8 with error handling, to replace any problematic characters.
We used a function cleaning that was to remove the white spaces and words that had nothing to do with the target language. We used regular expressions in many places, which meant we replaced all sequences of whitespaces by one space and also stripped the leading and trailing spaces of the text. This is how this step was really essential in maintaining the original integrity of content while

providing a clean dataset.
In applying normalization techniques, we convert all text to lowercase and removed punctuation in the case of Tamil and other Indic languages. Standardization was further required for eliminating variability arising due to case sensitivity and non-alphanumeric characters to not interfere with the summarization algorithms.
Finally, after all of these cleaning processes and normalizations had been done on the data, we saved them as new CSV files for easy access in subsequent phases of our research. This phase, that was full of detail concerning cleaning and normalizing the data, was very important to having the best performance of our summarization models, that is, to generate outputs which are more accurate and relevant in their context.

5.2. Models

We used several models for summarization, and each of them was chosen because of their unique strengths and the application in different languages involved: English, Tamil, Bengali, and Gujarati.

We began with **SumBasic**, ie., using the very basic nature of summary creation through frequency analysis of words. The model gives the calculation of how many times a word appears in the text and identifies the most frequent ones amongst them. We selected sentences that contained these high-frequency words, with a bias toward those that contributed the most to the understanding of the document as a whole. Although we experienced SumBasic to be efficient and straightforward to employ, we realized that frequency alone sometimes could not succeed in capturing textual subtleties as many times this resulted in trading off the ability of summation.

We now used **TF-IDF**, Term Frequency-Inverse Document Frequency. This model measures the importance of each term in relation to the whole document collection. The model that we consider consists of two main aspects- Term Frequency (TF), which counts how many times a word appears in a document, and Inverse Document Frequency (IDF), which evaluates the importance of a word in the whole dataset. We then scored terms by these metrics, picking those sentences that have terms with the highest score for summary generation. This approach was able to effectively balance local relevance with global context and thus was particularly strong in capturing the flavor of the text.

We leveraged the use of the **mT5** model for summarization work, which works on the transformer architecture and has been pre-trained on various language tasks using a large multilingual dataset. We framed summarization as a problem of text-to-text and thus allowed mT5 to natively transform an input text into a summary. With the use of self-attention mechanism in the model, it was easy to down-weight other words and phrases based on contextual relations. Through fine-tuning, mT5 became highly effective at producing coherent summaries while maintaining the original meaning and context of the text. This is an excellent advantage over traditional extractive methods.

We further developed the **XLSum** [19] long-form content-specific model. An XLSum model makes use of the encoder-decoder architecture highly suited for understanding and summarizing large documents. We preprocessed the input text in chunks capturing fine-grained details including broad themes. Training XLSum on a wide variety of lengthy documents helped it to very efficiently condense lengthy stories into nutshell summaries without losing any contextual information that was important. The decoder actually chose the sentences and phrases most relevant to work, ensuring the produced summaries were coherent and informative.

We fine-tuned the variant **mT5-Tamil**, focusing on an exhaustive Tamil corpus while retaining the core functionality and further enhancing its ability to understand unique syntactic and semantic features of Tamil. With this adaptation, mT5-Tamil improved its capability to summarize better. The self-attention mechanism was inherently important as it enabled mT5-Tamil to assess and decide about

the importance of each word in its context. This thematic training permitted summaries which were perfectly accurate and contextual, centered on the intricacies of Tamil literature and the modalities of communication.

We also employed a multi-Indic transformer-based model, **MultiIndic** [20], that is trained on multiple Indic languages. The nuances of the language in the model were very helpful in our research. Patterns and linguistic structure found in various kinds of textual data help MultiIndic learn, creating coherent summaries while showing respect to the linguistic context in which they were written. Of course, its effectiveness was really clear in summarizing texts in languages with drastically divergent structures from the English language.

We also leveraged a language-specific variation of the BERT architecture for text in Tamil, which we call **Tamil-BERT** [21]. The model employs a bidirectional attention mechanism that enables it to look at words on either side of a token as it processes one token. This made it easier for Tamil-BERT to capture the intricate relationships between words and phrases that define Tamil. These played an important role in arriving at coherent, contextually rich summaries. Its training on Tamil datasets made it learn all kinds of idiomatic expressions and other nuances of language, which further elevated its effectiveness for summarization tasks.

We further used **Indic-BERT** [22], which utilizes the BERT architecture to serve multiple Indic languages. After being pre-trained with a diverse set of texts, this learned the unique characteristics of each language. The model's bidirectional nature allowed it to process words in context, hence greatly improving its capacity to generate relevant summaries. This focus on understanding interactions in words within the much larger text made Indic-BERT particularly effective for summarization tasks in languages like Tamil, with multilingual capabilities that ensured high-quality outputs in a wide range of contexts.

It was through this multilateral approach that we attempted to achieve the very rich and multifaceted summarization process, reflecting the quality of languages involved in our research.

6. Analysis

6.1. Performance Metrics

One of the main aspects of text summarization is the assessment of quality in the produced summaries. The most commonly applied metrics to evaluate the produced summaries in this area are known as ROUGE, or Recall-Oriented Understudy for Gisting Evaluation. ROUGE consists of a set of measures comparing the generated summaries to one or more reference summaries prepared by human beings. This assessment accounts for the overlap of n-grams, or contiguous sequences of words, between the summaries generated and the reference, providing valuable insights into content coverage, fluency, and coherence overall.

ROUGE-1

ROUGE-1 specifically measures the overlap of unigrams, or single words, in the generated and reference summaries.

$$\text{ROUGE-1 Recall} = \frac{\text{Number of overlapping unigrams}}{\text{Total unigrams in reference}} \tag{1}$$

$$\text{ROUGE-1 Precision} = \frac{\text{Number of overlapping unigrams}}{\text{Total unigrams in generated summary}} \tag{2}$$

$$\text{ROUGE-1 F1} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{3}$$

As shown in Equations 1, 2, and 3, the ROUGE-1 metrics include Recall, Precision, and F1.

In the equations, **Matching Unigrams** is the number of overlapping unigrams between the generated summary and the reference summaries. While **Generated Unigrams** and **Reference Unigrams** are the total number of unigrams in the generated and reference summaries, respectively. ROUGE-1 is of use for general lexical overlap; therefore, it is the foundation measure used in summarization evaluation.

ROUGE-2

ROUGE-2 pushes the evaluation further through to bigrams, which in turn gives an enriched view of the relations of the context between and among the words of generated text. It uses the same precision and recall formulas except that they zero in on bigram matching rather than individual words. What it captures is the words and the relation that a bigram might hold where its relationship with the consecutive words has improved the effectiveness in judging coherence and flow during generated summaries. Calculations follow ROUGE-1's approach except this now is in the count and numbers of bigrams.

$$\text{ROUGE-2 Recall} = \frac{\text{Number of overlapping bigrams}}{\text{Total bigrams in reference}} \tag{4}$$

$$\text{ROUGE-2 Precision} = \frac{\text{Number of overlapping bigrams}}{\text{Total bigrams in generated summary}} \tag{5}$$

$$\text{ROUGE-2 F1} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{6}$$

As shown in Equations 4, 5, and 6, the ROUGE-2 metrics are based on bigram overlaps.

ROUGE-L

ROUGE-L measures LCS between the extracted and target summaries. Here, an LCS is measured as "the longest subsequence common to both and of matching words." Thereby comparing the word order in addition to the structure, ROUGE-L tends to give coherence a wider sense. Calculation of ROUGE-L precision, recall and F1 is as indicated below:

$$\text{ROUGE-L Recall} = \frac{\text{LCS length}}{\text{Total words in reference}} \tag{7}$$

$$\text{ROUGE-L Precision} = \frac{\text{LCS length}}{\text{Total words in generated summary}} \tag{8}$$

$$\text{ROUGE-L F1} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{9}$$

As shown in Equations 7, 8, and 9, the ROUGE-L metrics use the longest common subsequence (LCS) length.

Here, **LCS** is the length of the longest common subsequence, and **Generated Summary** and **Reference Summary** represent the number of words in each of the summaries. ROUGE-L is very effective in verifying the structural cohesion of generated summaries since it is sensitive to content and order.

6.2. Evaluation Procedure

The evaluation process with ROUGE scores follows some systematic steps. Before that, researchers would prepare by gathering a set of reference summaries along with their generated summaries. After the tokenization of generated and reference summaries into their constituent n-grams, each metric will count the number of matching n-grams.

Following this, precision, recall, and F1-scores for ROUGE-1, ROUGE-2, and ROUGE-L are calculated according to the formulas above. This ultimately produces scores for comparison over the quality of summaries generated against the reference summaries that were used.

With ROUGE scores, the quantitative analysis of summarization algorithm performance can be done for even better models to be developed. As a result, this means there would be improvement in natural language processing automated summarization. The scheme is a full evaluation in pushing summarization technologies while making sure that produced summaries will not be substandard to certain levels of accuracy and coherence.

Table 2 summarizes ROUGE-1 scores for various models on four languages: Tamil, English, Gujarati, and Bengali. The scores here measure the ability of each model to match individual words between the generated summaries and the reference summaries. The models include classical methods such as SumBasic and Freq Based, as well as transformer-based models like mT5, XLSum, and various language-specific models like Tamil-BERT and Indic-BERT. The results show a variation in performance across languages and models, with Freq Based achieving the highest ROUGE-1 scores in English and Gujarati, while models like mT5 and XLSum perform better in certain languages. Tamil-BERT and MultiIndic have relatively low scores, which may mean that they need further optimization for these tasks. In general, the table indicates how various summarization techniques perform in different languages. Both traditional and modern models give valuable insights into multilingual summarization tasks.

Table 3 reports the ROUGE-2 scores, which measure the overlap of bigrams (two consecutive words) between the summaries generated and the reference summaries. This metric is a stricter measure than ROUGE-1, requiring better understanding and generation of context. In this table, the results show that Freq Based and mT5 consistently deliver higher ROUGE-2 scores, particularly in languages like English and Gujarati, indicating that these models are better at capturing contextual relationships between words. SumBasic also does reasonably well across languages, though its scores are typically all lower than more advanced models. Tamil-BERT and MultiIndic fare worse in this test, particularly in English, which suggests the models are less effective at creating coherent bigrams in these languages. Overall, the ROUGE-2 scores indicate that the higher-advanced models, Freq Based and mT5 have a stronger capability of capturing the syntactic relationship between words in more than one language.

The ROUGE-L scores of Table 4 present the number of LCS between the summaries created through the model and those manually created in the references. ROUGE-L considers the structure and order of the entire summary produced. Hence, it is a better quality measure for summarization. Results in the table indicate that ROUGE-L scores have been more or less in similar trend with the scores obtained by ROUGE-1 and ROUGE-2. In this context, models like Freq Based and TF-IDF scored more significantly for both English and Gujarati. Notably, all languages of the mT5 model have a relatively poor score, which indicates that it fails in maintaining the sentence structure and coherence. Tamil-BERT and Indic-BERT also show lower effectiveness in other languages except the Tamil and Gujarati languages. Overall, the ROUGE-L scores depict how different models handle summary coherence and structures, where classical methods Freq Based and TF-IDF have proven to maintain the quality at the sentence level from diverse languages.

Table 2
ROUGE-1 scores for all models.

	Tamil	English	Gujarati	Bengali
SumBasic	0.0860	0.2257	0.0676	0.0740
Freq Based	0.0955	0.3210	0.0912	0.0820
TF-IDF	0.0916	0.1125	0.0674	0.0735
mT5	0.081	0.1290	0.0438	0.076
XLSum	0.0606	0.2010	0.0751	0.0703
mT5-Tamil	0.0963	-	-	-
MultiIndic	0.0312	-	0.0312	0.0426
Tamil-BERT	0.0306	-	-	-
Indic-BERT	0.0242	-	0.0420	0.0312

Table 3
ROUGE-2 scores for all models.

	Tamil	English	Gujarati	Bengali
SumBasic	0.0247	0.098	0.0376	0.0740
Freq Based	0.0333	0.1510	0.0942	0.0820
TF-IDF	0.0324	0.1024	0.0572	0.0735
mT5	0.0021	0.0972	0.0198	0.076
XLSum	0.0089	0.0997	0.0089	0.0703
mT5-Tamil	0.0349	-	-	-
MultiIndic	0.0045	-	0.0050	0.0426
Tamil-BERT	0.0042	-	-	-
Indic-BERT	0.0032	-	0.0403	0.0312

Table 4
ROUGE-L scores for all models.

	Tamil	English	Gujarati	Bengali
SumBasic	0.0844	0.0850	0.0788	0.0836
Freq Based	0.0952	0.0925	0.0905	0.0810
TF-IDF	0.0910	0.0890	0.0843	0.0851
mT5	0.0081	0.0120	0.0115	0.0130
XLSum	0.0606	0.0634	0.0579	0.0595
mT5-Tamil	0.0948	-	-	-
MultiIndic	0.0312	-	0.0325	0.0317
Tamil-BERT	0.0305	-	-	-
Indic-BERT	0.0245	-	0.0270	0.0282

6.3. Performance Analysis

This paper proved the effectiveness of various text summarization models that work with four languages, which are in this case English, Tamil, Bengali and Gujarati. Amongst these models, the **Frequency Based** model turned out to be the best-working model for English, Gujarati and Bengali whereas the mt5-Tamil produced the highest scores for Tamil. The Frequency Based summarization model garnered impressive ROUGE-1, ROUGE-2, and ROUGE-L scores. The success of Frequency Based is attributed to the fact that it simply works on significant word occurrences. Hence, it is capable of successfully distilling key information without losing contextual relevance. This feature makes it pretty suitable for richly morphological languages, where the discovery of major words may greatly determine the quality of the summary. The value of rogue scores that were obtained by the val dataset is given in Table 2, Table 3 and Table 4.

In Tamil, the best result was depicted by the **mT5-Tamil** at 0.0963 ROUGE-1; the Frequency Based model having a ROUGE-1 score of 0.0955 showed the second-best score. Such a specially tailored

Table 5
Performance of the Frequency Based model with a validation dataset for Gujarati.

Metric	Rouge-1	Rouge-2	Rouge-4	Rouge-L
F1-Score	0.2109	0.0835	0.0437	0.1958
Precision	0.320	0.150	0.065	0.285
Recall	0.157	0.063	0.035	0.146

Table 6
Performance of the Frequency Based model with a validation dataset for Bengali.

Metric	Rouge-1	Rouge-2	Rouge-4	Rouge-L
F1-Score	0.1957	0.1224	0.0938	0.1693
Precision	0.290	0.185	0.140	0.250
Recall	0.150	0.093	0.073	0.130

Table 7
Performance of the mT5 model with a validation dataset for Tamil.

Metric	Rouge-1	Rouge-2	Rouge-4	Rouge-L
F1-Score	0.1547	0.0877	0.0561	0.1468
Precision	0.230	0.130	0.083	0.218
Recall	0.120	0.070	0.045	0.113

training on data specific to Tamil proves to be quite helpful for the model when understanding the nuances in the language, underlining the case for language-specific adaptations. The advanced architecture along with the contextual understanding makes mT5 a great tool for Tamil summarization.

For Gujarati and Bengali, Frequency Based summarization has shown endurance consistently, thereby further solidifying its capabilities across languages. The results point towards the need to utilize summarization methods suited to the fine-tuned syntactic and semantic characteristics of each language.

However, it is also to be noted that this study does have some limitations. The frequency-based methods used may result in summaries that, although accurate on key terms, are often shallow and superficial, perhaps missing important contextual information. Also, models may vary based on the quality and size of training datasets for each language, which may adversely affect low-resource languages with fewer resources.

With advanced models, user feedback, and hybrid approaches combining extractive and abstractive techniques, we see wide potential in the improvement of summarization techniques. Future research can work on neural networks that, sooner rather than later, could push to deliver deeper context and semantics awareness to foster better quality in summaries. Additionally, user-centric features and interaction with summarization tools can improve the practical applicability of these models as much as possible and make it even more responsive to the user's needs.

7. Results

The submission for the Gujarati data was ranked 4th. The performance results are recorded in Table 5.
The submission for the Bengali data was ranked 4th. The performance results are recorded in Table 6.
The submission for the Tamil data was ranked 4th. The performance results are recorded in Table 7.
The submission for the English data was ranked 7th. The performance results are recorded in Table 8.

Table 8
Performance of the Frequency Based model with a validation dataset for English.

Metric	Rouge-1	Rouge-2	Rouge-4	Rouge-L
F1-Score	0.2295	0.0894	0.0513	0.1895
Precision	0.330	0.125	0.073	0.280
Recall	0.180	0.067	0.038	0.145

8. Conclusion

In general, this work has proven that the approaches based on frequency are powerful in terms of generative summaries but limited in someway and can be complemented by integrating them with some more sophisticated models. Techniques such as TF-IDF and basic n-gram approaches do an excellent job for lexical frequency to retrieve vital content but lose focus from context, semantics, and coherence of the generated summaries. We can further strengthen more complex models such as mT5, by using transformer architectures and attention mechanisms, in the interest of combining the best characteristics of frequency-based techniques. These result in subtler and information-rich summaries that hold up the content but also maintain the subtlety of language and meaning.
Models like mT5 can introduce the ability of having a deeper understanding of complex linguistic structures and relations in text, hence allowing it to contextualize and therefore produce coherent summaries. Integrating frequency-based methods with models is thus a hybrid approach to stand to draw strength from both sides. For instance, it can highlight highly frequent key phrases and concepts that the transformer model will use for good. The summary can then be in coherent narrative with much more depth and meaning for the source text than just the aggregate of the high frequency terms. The present research has developed new ways in which such an integrated approach might further find application across the entire range of linguistic contexts. With changing requirements of text summarization, adaptation to various languages, dialects, and genres is needed. Through frequency-based techniques on mT5 models, we can unlock more robust and adaptive summarization solutions that could potentially reach a larger audience. The methodology proposed here marks the beginning of future work with increased sophistication in summarization techniques in terms of how well they function and the nuances of the human language. Ultimately, the results here might enable information to become more readily available across domains and make the job of users easier in discerning meaning from large amounts of textual data.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] R. Satapara, P. Jain, Key advances in natural language processing: A 2023 review, Journal of Artificial Intelligence Research 67 (2023) 34–56.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, in: Advances in Neural Information Processing Systems, volume 33, 2020, pp. 1877–1901.

[3] P. Joshi, S. Santy, A. Budhiraja, K. Bali, M. Choudhury, The state and fate of linguistic diversity and inclusion in the nlp world, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 6282–6293. URL: https://aclanthology.org/2020.acl-main.557.

[4] O. Bojar, C. Buck, C. Callison-Burch, D. Dyer, M. Federico, Y. Graham, B. Haddow, P. Koehn, J. Leveling, C. Monz, et al., Findings of the 2014 workshop on statistical machine translation, in: Proceedings of the Ninth Workshop on Statistical Machine Translation, Association for Computational Linguistics, 2014, pp. 12–58. URL: https://aclanthology.org/W14-3301.

[5] R. Wijayanti, M. L. Khodra, K. Surendro, D. H. Widyantoro, Learning bilingual word embedding for automatic text summarization in low resource language, Journal of King Saud University-Computer and Information Sciences 35 (2023) 224–235.

[6] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, D. Klakow, A survey on recent approaches for natural language processing in low-resource scenarios, arXiv preprint arXiv:2010.12309 (2020).

[7] V. Srivastava, M. Singh, Challenges and considerations with code-mixed nlp for multilingual societies, arXiv preprint arXiv:2106.07823 (2021).

[8] S. Thara, P. Poornachandran, Code-mixing: A brief survey, in: 2018 International conference on advances in computing, communications and informatics (ICACCI), IEEE, 2018, pp. 2382–2388.

[9] M. Kl, A. Krishnakumar, F. Naushin, B. Bharathi, Text summarization for indian languages using pre-trained models, 2023.

[10] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, K. Kochut, Text summarization techniques: A brief survey, 2017. URL: https://arxiv.org/abs/1707.02268. arXiv:1707.02268.

[11] U. Hahn, I. Mani, The challenges of automatic summarization, Computer 33 (2000) 29–36. doi:10.1109/2.881692.

[12] I. Awasthi, K. Gupta, P. S. Bhogal, S. S. Anand, P. K. Soni, Natural language processing (nlp) based text summarization - a survey, in: 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 1310–1317. doi:10.1109/ICICT50816.2021.9358703.

[13] S. Satapara, B. Modha, S. Modha, P. Mehta, Findings of the first shared task on indian language summarization (ILSUM): approaches challenges and the path ahead, in: K. Ghosh, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2022 - Forum for Information Retrieval Evaluation, Kolkata, India, December 9-13, 2022, volume 3395 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 369–382. URL: https://ceur-ws.org/Vol-3395/T6-1.pdf.

[14] S. Satapara, B. Modha, S. Modha, P. Mehta, FIRE 2022 ILSUM track: Indian language summarization, in: D. Ganguly, S. Gangopadhyay, M. Mitra, P. Majumder (Eds.), Proceedings of the 14th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE 2022, Kolkata, India, December 9-13, 2022, ACM, 2022, pp. 8–11. URL: https://doi.org/10.1145/3574318.3574328. doi:10.1145/3574318.3574328.

[15] S. Satapara, P. Mehta, S. Modha, D. Ganguly, Indian language summarization at fire 2023, 2024, pp. 27–29. doi:10.1145/3632754.3634662.

[16] S. Satapara, P. Mehta, S. J. Modha, D. Ganguly, Key takeaways from the second shared task on indian language summarization (ilsum 2023), in: Fire, 2023. URL: https://api.semanticscholar.org/CorpusID:269791803.

[17] S. Satapara, P. Mehta, S. Modha, A. Hegde, S. HL, D. Ganguly, Overview of the third shared task on indian language summarization (ilsum 2024), in: K. Ghosh, T. Mandl, P. Majumder, D. Ganguly (Eds.), Working Notes of FIRE 2024 - Forum for Information Retrieval Evaluation, volume CEUR-WS.org of *CEUR Workshop Proceedings*, Gandhinagar, India, 2024.

[18] S. Satapara, P. Mehta, S. Modha, A. Hegde, S. HL, D. Ganguly, Key insights from the third ilsum track at fire 2024, in: Proceedings of the 16th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE 2024, ACM, Gandhinagar, India, 2024.

[19] T. Hasan, A. Bhattacharjee, M. S. Islam, K. Mubasshir, Y.-F. Li, Y.-B. Kang, M. S. Rahman, R. Shahriyar, XL-sum: Large-scale multilingual abstractive summarization for 44 languages, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 2021, pp. 4693–4703. URL: https://aclanthology.org/2021.findings-acl.413.

[20] A. Kumar, H. Shrotriya, P. Sahu, R. Dabre, R. Puduppully, A. Kunchukuttan, A. Mishra, M. M. Khapra, P. Kumar, Indicnlg suite: Multilingual datasets for diverse nlg tasks in indic languages, 2022. URL: https://arxiv.org/abs/2203.05437.

[21] R. Joshi, L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages, arXiv preprint arXiv:2211.11418 (2022).
[22] D. Kakwani, A. Kunchukuttan, S. Golla, G. N.C., A. Bhattacharyya, M. M. Khapra, P. Kumar, Indic-NLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages, in: Findings of EMNLP, 2020.