Text Summarization for Resource-Poor Languages: Datasets and Models for Multiple Indian Languages

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in Computer Science and Engineering by Research

by

Vakada Lakshmi Sireesha
20171137
lakshmi.sireesha@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
May 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Text Summarization for Resource-Poor Languages: Datasets and Models for Multiple Indian Languages" by Sireesha Vakada, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. Radhika Mamidi

To My Family and Friends

Acknowledgments

I want to seize this opportunity to express my heartfelt gratitude to all those who have provided unwavering support and endless motivation throughout my journey at IIIT-Hyderabad. Each and every one of you holds a special place in my heart.

I am immensely thankful to my supervisor, Dr. Radhika Mamidi, for her invaluable support. She has been a constant pillar of strength throughout my degree program. Her genuine belief in me and her guidance has allowed me the freedom to explore different techniques and have steered me in the right direction. I consider myself fortunate to be her research student.

I am fortunate to be a part of the Language Technologies and Research Center IIIT Hyderabad. This is where I got the interest and confidence to start my research journey. I am fortunate to interact with faculty like Dr. Vasudev Varma and Dr. Manish Shrivatsava. I would specially thank Dr.Vasudev Varma for taking me in their Telugu Wikipedia Project, which served as one of the root motivations for part of my problem statement. Thank you for sharing the knowledge to improve my academic, project, and research skills.

I want to thank my family for their support throughout this journey. Despite the fact that my graduation was a year late, they never pestered me about the status of my research at any stage. Special thanks to my brother, who enthusiastically congratulates me on every minor accomplishment.

I'd want to thank my mentors, Mounika Marreddy and Subbareddy Oota, as well as my comrade Charan Chinni, for always assisting me with new ideas and hoping the best for me. I began working with them on a couple of projects, which was invaluable learning in the early stages of my research career. I'll never forget our late-night meetings before each paper submission. I want to thank my friend, motivator, and comrade Anudeep for always being there during my research adventure. I can't thank you enough for all the effort you have put into supporting and motivating me; thank you, Aonuu!!

Special thanks to my father for encouraging me to choose IIIT-Hyderabad. I can proudly say that I have zero regrets of not joining any other NIT/IIT. The experience I had at IIIT-Hyderabad was exceptional. I thank all my friends m Megs, Shreya, Srikar, Ravi, Amul, Rupa, Yippie, Aravind, Zoo, Bheemala, JD, Mayukha, Varsha, Rasna, Jessy, Vishwath, Bijjam, Sushanth, Snehi, Vijay, Gayatri, Chintoo, PK, Pavan, Harshitha, Tejaswi, and PV for cheering me up in the smallest of any achievement. Thank you, Varsha, for being an amazing roomie. I would also thank my beloved dual-degree friends for making the fifth year memorable. I would also thank my seniors, Vasant and Raghava, for patiently bearing me throughout the years and supporting me at every turn. I thank each of you sincerely for

letting me grow in such a healthy and lovely environment. This journey seemed smooth despite many ups and downs because of your unconditional support and affection. Lastly, I conclude my journey at IIIT-Hyderabad, armed with many memories to cherish forever.

Abstract

Document summarization aims to create a precise and coherent summary of a text document. There exist a plethora of deep learning summarization models that are developed mainly for English, which often requires (i) a large training corpus and (ii) efficient pre-trained language models and tools. However, English summarization models for low-resource languages like Indian languages are often limited by rich morphological variation and syntactic and semantic differences. Also, the restricted form of supervision limits the generality and usability of low-resource languages due to the lack of annotated corpora. The Graph Autoencoder (GAE) model has recently shown superior performance on several NLP tasks, even with limited resources. In this work, we propose **GAE-ISUMM**, an unsupervised **I**ndic **Summ**arization model that extracts summaries. In particular, our proposed model uses GAE and leverages: (i) learning document representations and (ii) jointly learning sentence representations and summary of the document. For evaluation purposes, we introduce **TELSUM**, a manually annotated summarization dataset comprised of 501 document-summary pairs. Extensive experiments on existing low-resource datasets (XL-Sum) and TELSUM provide the following insights: (i) our proposed model displays state-of-the-art results on XL-Sum and report benchmark results on TELSUM, (ii) Surprisingly, the inclusion of positional and cluster information in the proposed model further improved the performance of summaries. We open-source our dataset and code [1].

On the other hand, with the advancement of various deep-learning methodologies and transformer-based models, summarization has advanced to a new level. However, consistent and standard datasets must be produced to benefit from these deep learning algorithms fully. The creation of dedicated resources is rarely seen for low-resource Indian languages, unlike English which hinders the progress of summarization. To this end, we create summarization resources for Indian languages by introducing **ISummCorp** (**I**ndic **Summ**arization **Corp**ora) and **IndicSumm** (**Indic** Language **Summ**arization Models). IsummCorp is a highly abstractive summarization dataset sourced from the Times Of India (TOI). It is manually annotated by experts across eight Indian languages. Human and intrinsic evaluations demonstrate the high quality, abstraction, and compactness of ISummCorp. IndicSumm is a set of diverse monolingual and multilingual models based on ISummCorp. We refined IndicSumm, by fine-tuning the sophisticated, multilingual pre-trained mT5 model. With ISummCorp, we show that a model can perform better in a monolingual environment when trained with enough monolingual data than in a multilingual finetuning scenario. To investigate the potential of monolingual models, we finetuned mT5

[1] https://github.com/scsmuhio/Summarization

using ISummCorp in both monolingual and multilingual situations and achieved better performance in a monolingual setting. Furthermore, we compare IndicSumm to other multilingual summarization models (XL-Sum and IndicBART) and achieve the state-of-the-art results.

Contents

Chapter Page

1 Introduction 1
1.1 Why is text summarization important? 1
1.2 Summarization 2
1.2.1 Extractive Summarization 2
1.2.2 Abstractive Summarization 2
1.2.3 Multilingual Summarization 2
1.3 Motivation and Overview 3
1.4 Challenges 4
1.5 Contribution of this thesis 5
1.6 Organization of the Thesis 6

2 Related Work 7
2.1 Summarization techniques 7
2.1.1 Traditional Approaches 8
2.1.1.1 Graph Based Approaches 8
2.1.1.2 Deep Learning approaches 8
2.1.2 Summarization for Indian languages 9
2.2 Summarization Datasets 9
2.2.1 Monolingual Datasets 10
2.2.2 Multilingual Datasets 10
2.2.3 Indian language Summarization Datasets 10
2.3 Multilingual Models for Indian Languages 11
2.4 Background Literature 11
2.4.1 Graph Autoencoder (GAE) 11
2.4.2 GRU 12

3 GAEISUMM: An unsupervised model 13
3.1 Introduction 13
3.2 Datasets 16
3.2.1 TELSUM Dataset 17
3.2.1.1 Dataset Collection and pre-processing 17
3.2.1.2 Annotation Details 17
3.2.2 Other Indian Langauge Datasets 17
3.2.2.1 NCTB and BNLPC: 17
3.2.2.2 XL-Sum: 18

3.2.2.3 Marathi, and Hindi Short Summaries: . . . 18
3.3 GAEISUMM Model . . . 19
3.3.1 Graph Construction: . . . 19
3.3.2 Document Level Graph Construction and Encoding: . . . 19
3.3.3 Sentence Encoding and Summary generation . . . 19
3.3.3.1 Sentence Level Graph Construction: . . . 20
3.3.3.2 Sentence Encoding: . . . 20
3.3.3.3 Clustering and Cluster Embeddings: . . . 21
3.3.3.4 Sentence Scoring and Selection: . . . 22
3.3.3.5 Loss calculation: . . . 23
3.4 Experimental Setup . . . 23
3.4.1 Model Training Setup & Hyperparameters: . . . 23
3.4.2 Evaluation Metrics: . . . 24
3.5 Results and Analysis . . . 25
3.5.1 Performance on TELSUM Dataset: . . . 25
3.5.2 Analysis of Clusters: . . . 26
3.5.3 Ablation Studies: . . . 27
3.5.4 Performance on XL-Sum (Seven Indian Languages): . . . 29
3.5.5 Does Cross-Lingual Models Improve Summarization Performance? . . . 30
3.5.6 Experiments in Cross-lingual Setting: . . . 30
3.5.7 Performance on Other Summarization Datasets: . . . 31
3.6 Conclusion . . . 31
3.7 Ethical Statement . . . 31

4 ISummCorp: an abstractive summarization dataset . . . 33
4.1 Introduction . . . 33
4.2 ISummCorp . . . 34
4.2.1 Dataset Collection . . . 34
4.2.2 Dataset Creation . . . 35
4.2.3 Dataset Pre-processing . . . 35
4.2.4 Uniqueness of ISummCorp . . . 36
4.3 Quality Analysis of Dataset . . . 37
4.3.1 Intrinsic Evaluation . . . 38
4.3.2 Human Evaluation . . . 39
4.4 Challenges faced . . . 39
4.5 Analysis of Summaries . . . 41

5 IndicSumm: language models in Indian context . . . 44
5.1 IndicSumm . . . 44
5.2 Training Details . . . 45
5.2.1 Monolingual IndicSumm . . . 45
5.2.2 Multilingual IndicSumm . . . 45
5.3 Results and Analysis . . . 46
5.3.1 Analysis of IndicSumm . . . 46
5.3.2 Baselines . . . 46
5.3.3 Comparison with existing multilingual models . . . 48

5.4 Analysis of Summaries 50
5.5 Conclusion 51

6 Conclusion and Future Work 53
6.1 Future Work 53
6.2 Other Publications 56
6.2.1 Journal Proceedings 56
6.2.2 Conference Proceedings 56

Bibliography 58

List of Figures

Figure Page

3.1 Outline of GAE-ISUMM. Our model GAE-ISUMM involves two phases: Document Encoding - a) Document Graph Construction and b) Obtaining Graph-based Representations; Sentence Embedding and Summary Generation involves seven steps starting from 1) Sentence Graph Construction to 7) Loss calculation. 14
3.2 GAE model 15
3.3 Working of GRU 15
3.4 An Example of human annotated summary and GAE-ISUMM predicted summary from TELSUM 26
3.5 Analysis of clusters: the clusters formed by GAE-ISUMM for an example article from TELSUM. 27
3.6 Clusters formed by a sample article from TELSUM dataset(English version). 28
3.7 An Example of Actual and Predicted summary on XL-sum. 28

4.1 An Example of Article-summary pair from ISummCorp 40
4.2 Another Example of Article-summary pair from ISummCorp 41
4.3 TOI website article page in Telugu (Samyam). 43

5.1 A Sample article-summary pair and Predicted summary from ISummCorp. 50
5.2 A Sample article-summary pair and Predicted summary 51

List of Tables

Table Page

3.1 Statistics of TELSUM and different available summarization datasets in Indian languages. Here, the Compression ratio is the average length of the summary to the average document length. . . . 18
3.2 Comparison of ROUGE score results of GAE-ISUMM with other methods on TELSUM dataset. . . . 24
3.3 GAE-ISUMM on XL-sum: ROUGE score results on seven different Indian languages. These results are compared with mBART (MB) and IndicBART (IB) results from [8]. 25
3.4 ROUGE score results on Other Summarization Datasets using GAE-ISUMM. Here '-' indicates that the dataset has no state-of-art results. . . . 29
3.5 Cross-lingual experiments of GAE-ISUMM: Bilingual setting of "hi" and "en" with the other Indian languages from XL-sum. Here '-' represents no cross-lingual experiment for that permutation of languages. . . . 29
3.6 Ablation studies of GAE-ISUMM on TELSUM dataset. . . . 30

4.1 ISummCorp dataset statistics . . . 36
4.2 Intrinsic Evaluation of ISummCorp and other datasets. All the values are reported in percentages. Redundancy cannot be calculated for the XL-Sum dataset because their summaries are of a single sentence. . . . 37
4.3 Manual evaluation of different languages from ISummCorp on average. Abstractivity is the percentage of novel 1-grams in the summary. The Consistency metric is rated out of 1, and the remaining human evaluation metrics are rated out of 5. . . . 38

5.1 Comparison of IndicSumm Monolingual and IndicSumm Multilingual models. . . . 46
5.2 Comparison of IndicSumm with few baselines with Rouge-1 metric . . . 47
5.3 Comparison of IndicSumm with few baselines with Rouge-2 metric . . . 47
5.4 Comparison of IndicSumm with few baselines with Rouge-L metric . . . 48
5.5 Comparison of IndicSumm models with the existing multilingual models with Rouge-1 metric . . . 48
5.6 Comparison of IndicSumm models with the existing multilingual models with Rouge-2 metric . . . 49
5.7 Comparison of IndicSumm models with the existing multilingual models with Rouge-L metric . . . 49

Chapter 1

Introduction

1.1 Why is text summarization important?

The past few years have seen a tremendous increase in the growth of digital data. Social media platforms such as Facebook, Twitter, Quora, and Reddit have made online material accessible to everyone. According to the International Data Corporation (IDC), the total amount of digital data traveling annually around the world will increase from 4.4 zettabytes in 2013 to 180 zettabytes in 2025. Web pages, blogs, news articles, status updates, and other forms of content have all contributed to the dramatic surge. With so much unstructured data available, selecting only the most essential information from each is crucial.

Search engines are one of the crucial applications of summarization. Search engines have been the primary source of knowledge in any field over the last two decades. Understanding every relevant document related to a query is a complicated and impossible process with billions of users. As a result, it is necessary to provide information to the users in a concise format given a query, which is where summarization becomes handy.

Later comes the summarization of news articles. The news is the most reliable source of information in everyone's daily life. It involves different domains such as politics, sports, business, entertainment, social awareness, and many others. News helps us gather knowledge from different parts of the world. It can be used as a resource to learn more about a subject or, on occasion, to increase general knowledge. It is also capable of impacting society to any extent. In the modern era, the news is being read online to save time. To gain more knowledge or awareness about the surroundings, it is necessary to provide the users with information concisely. Summarization aims to comprehend the main informative content quickly and concisely. Text summarization enables us to achieve this goal by producing more focused and shorter summaries that capture the key information. There are several other applications to summarization, such as Text classification, Question Answering, Paraphrasing, and Entity Timelining.

1.2 Summarization

Text Summarization is one of the focused fields of the Natural Language Processing(NLP) community, posing several challenges in aspects like quality resources, quality models, and consistent text generation. Summarization provides a brief synopsis of the input text containing the key information. Summarization is mainly of two kinds: Abstractive summarization and Extractive summarization. We briefly discuss Extractive and Abstractive techniques below. Further, we also explain the difference between a usual summarization task and multilingual summarization.

1.2.1 Extractive Summarization

Extractive summarization picks the most important segments from the input text and concatenates them to form a summary. Extractive methods were dominant in the early era of summarization, majorly based on techniques such as Frequency-based approaches, Keyword extraction, sentence length and position, cue words, and many others [36, 21]. In extractive summarization, the sentences in an article are often ranked based on the features, and a subset is chosen to create a clear and brief extractive summary. Initially, graph-based approaches such as TextRank [46] and LexRank [11] became more famous under extractive summarization techniques. Further, researchers have explored both supervised and unsupervised techniques for extractive summarization. Recent works have explored deep-learning techniques that mainly include transformers, auto-encoders [21], and seq-to-seq [48] models to make the extractive summarization more effective.

1.2.2 Abstractive Summarization

Abstractive summarization reproduces essential information in a new way after thoroughly analyzing and interpreting the text. The main goal is to create a new, concise text that captures the essence of the original. Early abstractive summarization methods focused on text compression, information extraction, and clustering [67]. With the recent advancements in deep learning and NLG (Natural Language Generation), abstractive approaches shifted their interest toward seq2seq models and transformer-based models [49].

1.2.3 Multilingual Summarization

Multilingual summarization is a task where we train the models in multiple languages instead of one. Multilingual summarization aims to create a more generic platform for different languages. The NLP community is recently seeing a surge in the development of multilingual models and datasets. Various multilingual models were developed with Transformers [71] as the base architecture. BERT [9], BART [31], T5 [59] are some of the fruits of the transformer architecture which are further used to develop models for multiple languages. Recently contributed multilingual models include mBERT [58], mBART [37], mT5 [77]. These models were mainly trained for translation tasks. However, when

provided with task-specific datasets, these models can be fine-tuned to downstream NLP tasks such as classification, summarization, question-answering, and many more. Here, we will briefly discuss the base models of different multilingual models.

- **T5:** T5 is an encoder-decoder model that was trained on various tasks that are both supervised and unsupervised and are each translated into a text-to-text format. T5 is effective in different tasks.

- **BART:** Bart employs a typical seq2seq/machine translation architecture with a left-to-right decoder and a bidirectional encoder (like BERT) (like GPT). BART performs well for comprehension tasks but is especially effective when fine-tuned for text generation.

- **BERT:** BERT is a transformers model that was self-supervised and pre-trained on a sizable corpus of multilingual data. BERT was pre-trained on the unlabeled texts, which allowed it to use a large amount of data that is readily accessible. The texts were then used as inputs for an autonomous process that generated labels and inputs. To be more precise, it was pre-trained with the MLM (Masked language Modelling) and NSP objectives (Next Sentence Prediction).

1.3 Motivation and Overview

Indian languages (Languages that are native to India) are spoken by about 10% of people all around the globe. Eight Indian languages secure a place among the top twenty spoken languages in the world, and around thirty Indian languages with more than a million speakers [26]. With the drastic increase in digital usage content by the Indian population, it is essential to build the necessary resources and tools for Indian languages. However, Indian languages cannot reap the benefits of emerging deep learning models due to a lack of standard and massive annotated summarization datasets. Additionally, Indian languages have significant structural and morphological variations from high-resource languages like English, and techniques developed for English cannot be extended to Indian languages. The primary motivation for research in the summarization of Indian languages is to develop techniques and resources that help generate summaries more effectively and accurately. However, very few works discuss the summarization of low-resource Indian languages, which cannot be scaled relative to high-resource languages like English.

Applying supervised deep learning techniques for extractive text summarization is currently complicated by the need for manually produced large-scale extractive summaries that serve as the networks' training data. So, to overcome the limitations of low-resource languages, we propose an unsupervised summarization technique GAE-ISUMM. For any low-resource language that does not have adequate data, GAE-ISUMM can be used to produce summaries since it is unsupervised.

Further, even with the exponential growth of deep learning techniques in summarization, we realized we could not reap the benefits of these techniques due to the lack of proper summarization datasets in

Indian languages. We then created an Indian language-specific multilingual dataset (ISummCorp) and models (IndicSumm). We also created monolingual summarization models specific to Indian languages taking a notch high in low-resource languages. We also prove that a language model performs better when fine-tuned with enough data in a single language setting than in a multilingual setting with the help of ISummCorp.

1.4 Challenges

Text summarization is challenging since it is heavily dependent on the context and is very abstract. Given an article to a single person in two different instants, even then, the output summaries turn out to be different. So, it is highly improbable that we will arrive at a single, universal solution for summarization. Here, we briefly discuss the challenges faced in the process of resource creation towards summarization.

- Lack of resources: The main limitation for resource-poor languages in summarization is the need for more resources. Languages with limited resources need more standardized datasets and language-specific approaches. Due to the morphological structure and syntax difference, summarization techniques created for English cannot be used for other languages. In addition, resources like corpora, language models, standard datasets, and pre-processing tools are necessary to explore the potential of summarization in a given language fully.

- Automated text generation: Text generation is a complex process and needs lots of training data for text generation. It needs a complete understanding of the language and complete contextual information about the article. Regarding text generation of summaries, the model must first understand the language and learn about the contextual information. The model also has to learn to prioritize the crucial elements of the article.

- Processing long documents: Understanding and processing long documents would be challenging compared to shorter documents. Even in the world of deep learning, there are limitations regarding the input and output sizes and the number of parameters. Additionally, processing long documents with neural methods are always computationally expensive and time-taking.

- Summary size variations: The summary length depends on the extraction process and differs for different datasets. Summaries for a few datasets are of a single sentence. Furthermore, some other datasets might have the summary size to be half of the article. With different datasets having different criteria for summaries, it is challenging to idealize the summary size.

- Evaluation metric: Summary generation cannot be evaluated objectively and needs an abstract perspective. The evaluation metrics generally used for summarization are ROUGE [32] and BLEU [55] scores. However, since summarization is an abstract task, human evaluation for summaries is also required, which might be time-consuming and expensive.

1.5 Contribution of this thesis

As mentioned earlier, there is a need to create annotated datasets, different summarization techniques, and develop machine learning and deep learning models to understand text generation of low-resource languages better. To our knowledge, no such resources are available for Indian languages, and we are the pioneers in creating Indian-language-specific resources for summarization. At first, with limited resources and dataset creation being time-consuming, we developed an unsupervised summarization technique, GAE-ISUMM. To evaluate our technique, we created TELSUM, a summarization dataset, manually annotated by professional annotators. Later, learning about the drastic increase of datasets and models for all the languages worldwide, we attempted to create an Indian-language-specific multilingual dataset (ISummCorp) and models (IndicSumm). We will briefly discuss all the work here and explain each in detail later. The main contributions of this thesis are as follows:

- **GAE-ISUMM**: GAE-ISUMM is an unsupervised Indic summarization model that extracts summaries from text documents that leverage the idea of Graph AutoEncoder. GAE-ISUMM is a model where it learns the summary from the document. GAE-ISUMM, being an unsupervised technique, does not require any labeled data for training.

- **TELSUM**: TELSUM Dataset is a summarization dataset created in Telugu(a resource-poor Indian language) to evaluate the unsupervised GAE-ISUMM technique. The TELSUM dataset consists of 501 document-summary pairs. The summaries were manually written by two professional annotators who are native Telugu speakers. Unlike other summarization datasets, TELSUM is a dataset where annotators manually create summaries rather than retrieve them straight from a news source.

- **ISummCorp**: ISummCorp, a large-scale, multilingual summarization dataset for eight Indian languages extracted from Times Of India(TOI). Our corpus provides 376k article-summary pairs from eight different Indian languages: Hindi(Hi), Tamil(Ta), Telugu(Te), Bengali(Bn), Gujarati(Gu), Marathi(Ma), Malayalam (Ml), and Kannada(Kn). ISummCorp is the first Indian language-specific dataset created to date.

- **IndicSumm**: IndicSumm is a set of monolingual and multilingual summarization models trained on ISummCorp. These models are finetuned based on mT5 architecture. The multilingual model supports all 8 Indian languages of ISummCorp. The monolingual models are designed for each of the eight languages to explore the potential of low-resource languages when trained with adequate data.

The TELSUM, ISummCorp, and IndicSumm models are publicly available under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. We open-source our data and our models. The data and code related to the unsupervised technique can be accessed from the reposi-

tory [1]. Furthermore, the content related to the multilingual datasets and models can be accessed from the mentioned repository [2].

1.6 Organization of the Thesis

There are seven chapters in the study. The first chapter is the introduction to the thesis; the subsequent chapters are summarised as follows:

- **Chapter 2. Related work**

  In this chapter, we discuss the literature survey of available summarization datasets for Indian languages. Also, we present several multilingual deep learning models and datasets developed using multiple languages to date.

- **Chapter 3. GAEISUMM**

  In this work, we propose an architecture, the unsupervised summarization Model. The model mainly depends on the idea that the summary is learned from the document itself. This chapter explains the different components of the model in detail. We also explain how the summary is learned from the document through contrastive loss.

- **Chapter 4. ISummCorp**

  This is the first work to create an Indian language-specific multilingual summarization dataset on such a large scale. This chapter explains the whole extraction process and the pre-processing steps involved. We also explain how we evaluated our summaries with detailed guidelines and examples. Additionally, we discuss how ISummCorp is different from other datasets.

- **Chapter 5. IndicSumm**

  In this chapter, we present a set of monolingual and multilingual models finetuned on ISummCorp. We explain the experimental setup and performance in detail compared with the existing multilingual models.

- **Chapter 6. Conclusion and Future Work**

  In this chapter, we present the conclusion of our contributions towards low-resource Indian languages. Further, we mention the future scope of our contributed datasets and models in the field of Natural Language Processing.

[1] https://github.com/scsmuhio/Summarization
[2] https://github.com/sireeshasummarization/samayam_data

Chapter 2

Related Work

Summarization tries to minimize the content present in a document by obtaining only the essential information. It is one of the most captivating problem statements for researchers from many decades. [40] first openly discussed the problem of summarization in mid 20th century. With the significant increase in digital data, summarization has recently received more attention. Additionally, text generation became more straightforward with the emergence of deep learning techniques.

In this chapter, we have extensively studied summarization techniques and datasets in monolingual and multilingual settings. We first discuss the emergence of various summarization techniques through time. And then mention various benchmark datasets released for monolingual and multilingual summarization tasks. Later, we will discuss recent state-of-art models and their performance on different datasets. We also include the progress of summarization for Indian languages in terms of techniques and datasets parallelly.

2.1 Summarization techniques

Summarization can be of two types: 1) Extractive summarization and 2)Abstractive summarization. Extractive summarization is where we select the crucial text, concatenate and present it as the summary. In the case of abstractive summarization, we paraphrase the important information in the document to obtain the summary. Extractive summarization involves selecting key phrases and essential sentences from the document and combining them to form a summary. This summarization preserves critical information by leaving out redundant or less important details. However, Abstractive summarization produces text summaries that involve generating a new, condensed version of the document. Abstractive summarization is more challenging as it has to understand the content of the original text and rephrase them.

2.1.1 Traditional Approaches

At the start, most of the methods proposed for summarization are based on statistical or linguistic features such as sentence position and length [11], named entities [19], the number of keywords present in a sentence, term frequency [40]. Most traditional approaches follow a scoring mechanism where a score is assigned to each sentence based on these features or rule-based techniques and ranked accordingly to generate a summary.

2.1.1.1 Graph Based Approaches

The graph-based approaches for summarization started with inspiration from the PageRank algorithm [46]. In graph-based approaches, the document is usually considered a connected graph where the sentences play the role of vertices (or) nodes. The edge weights tell us about the connectivity in a graph, which is calculated based on the similarity between nodes [11]. LexRank technique [11] computes the centrality of the sentences by using degrees of similarity present between words or phrases. Other techniques that use similarity scores for edge weights are Luhn summarization [40] and KL greedy summarization [16]. They may also consider various linguistic features to calculate the graph's edge weights. [82] considers discourse-level features and similarity score to obtain the edge weights.

However, seq2seq models lack in capturing the global context and the long-distance sentence relationships present in a document. Modeling long-range inter-sentence relationships with transformer-based models are still challenging [75] and requires massive computation and memory. By capturing the long-term dependencies and treating the document as a graph[24], graph-based approaches assist in overcoming these limitations. Later, with the new idea of Graph Convolutional Networks (GCN), graph-based approaches have become more popular recently. They have also proven their dominance in other fields such as classification [78] and semantic role labelling [43]. Graph-based models are capable of drawing syntactic information, exploiting long-range multi-word relations, and have been deployed on document-word relationships [38, 20]. [74] proposes a Graph-based selective attention mechanism preserving the document's syntactic and semantic structure, achieving state-of-art on CNN/Daily Mail dataset [20].

For more effective and computation friendly, we explored the idea of a recently proposed method, Graph Autoencoder (GAE) [25]. GAE here captures the hidden semantic information between documents and sentences by using the idea of an autoencoder (AE) [62] to graph-structured data. Few recent papers have also obtained benchmark results on text classification using GAEs [73]. [73] uses GAE, outperforming existing benchmark results in text classification. Nevertheless, the application of GAE in summarization still needs to be explored.

2.1.1.2 Deep Learning approaches

The literature survey shows that most extractive summarization techniques rely on learning efficient text representations using these deep learning approaches. Deep neural models, such as Recurrent

Neural Networks (RNN) [48], Convolutional neural networks (CNN) [81] and attention-based models [60] have been successful in summarization with their strong representation power. [48] uses a GRU(Gated Recurrent Unit)-based RNN(Recurrent Neural Network) in order to obtain extractive summaries of given document. Here, the summarization is treated as a sequence classification problem, and the sentences are selected by training the model. [81] treats summarization as a regression problem, where they rank sentences with the help of CNNs (Convolutional Neural Networks).

In abstractive summarization, most models follow encoder-decoder-based approaches to generate summaries either in a supervised or unsupervised setting. Several researchers also apply the idea of RNN and attention to the abstractive summarization models [57, 6]. By using attention encoder-decoder-based models, state-of-the-art performance was achieved by [60]. [76] proposes a hierarchical transformer model to extract summaries. Similarly, [38] introduces a document-level encoder using transformers to generate summaries, minimizing the reconstruction loss.

2.1.2 Summarization for Indian languages

The progress of summarization for Indian languages has been low compared to other non-English languages around the globe. A few years back, summarization for Indian languages was restricted to traditional approaches such as keyword extraction, frequency-based approaches, and PageRank-based techniques [47, 61]. [27] studied different approaches developed for Indian languages and a few other resource-poor languages. From the study of [27], we observe several techniques developed for Indian languages, such as data clustering, graph-based approaches, rule-based approaches, and fuzzy logic. However, irrespective of the technique, the datasets used for the evaluation are of tiny size, and the techniques used are of the existing baselines. However, the past 2-3 years have seen an exponential increase in the number of works for Indian languages. Recently, many works used seq2seq models for extractive and abstractive summarization tasks of Indian language [2, 28]. [23] proposes a deep learning architecture based on attention-based LSTM models. They propose this technique or Hindi and Marathi languages.

2.2 Summarization Datasets

Several benchmark datasets have been introduced to evaluate the performance of summarization models. These datasets test the accuracy and effectiveness of different summarization algorithms and approaches. They help researchers and developers understand the strengths and limitations of different models. The extraction process for these datasets typically involves collecting many documents and manually creating summaries for them. The summaries are then used as the ground truth for evaluating the performance of summarization models. The quality of the summaries is often assessed using various evaluation metrics, such as ROUGE [32] or BLEU [55].

2.2.1 Monolingual Datasets

With English being the most spoken language in the world, most of the benchmark datasets are dedicated to the English language only. We will discuss some of them and their summary extraction process in brief. One of the most used summarization datasets to evaluate any model is CNN/Daily Mail dataset introduced by [49]. The CNN/DailyMail dataset was created from news stories on CNN and Daily Mail websites, where the human-written bullet points are considered the summary. Another most used dataset in recent times is NEWSROOM [13]. NEWSROOM is extracted from search and social media metadata. This dataset is a mix of both abstractive and extractive summaries extracted from different news publications. A few other benchmark datasets that are extensively used are DUC2002 [1] and XSum [50]. DUC2002 has 567 articles with two different gold summaries for each article. Here, professionals write the summary to create a summarization dataset. XSum dataset is extracted from the BBC news website to create a short, one-sentence summary.

2.2.2 Multilingual Datasets

Recently, multilingual models or datasets have gained huge attention in the NLP community. The multilingual tradition on a large scale was started by [63] with a dataset named MLSUM in 5 different languages(French, German, Russian, Turkish, and Spanish) obtained from online newspapers. [29] extracted around 770k article-summary pairs from WikiHow in 18 different languages. The summaries are extracted by aligning the images in WikiHow.

Other recently released datasets are XL-Sum [18] and MassiveSumm [70]. MassiveSumm is a vast multilingual dataset for 92 languages from 16 language families, comprising about 28.8 million articles. Similar to XSum, [18] released the XL-Sum summarization dataset for 44 languages sourcing from the BBC news website. [53] released the GlobalVoices multilingual dataset from a single source for 15 languages. The dataset consists of parallel data, which makes it one of the benchmark datasets for cross-lingual summarization. Apart from the large-scale datasets, other multilingual datasets are published in other European and Asian languages. [65] releases corpora for low-resource languages Catalan and Spanish. [68] released a dataset WikiMulti of around 230k article-summary pairs for 15 languages inspired by the concept of Feature Articles of Wikipedia [2].

2.2.3 Indian language Summarization Datasets

We have seen several summarization datasets till now, most of which are dedicated to English. There is a lack of summarization datasets available for Indian languages, and even those that do exist often have low-quality summaries. This is in contrast to the abundance of several English-language summarization datasets. Additionally, many multilingual summarization datasets do not include support for Indian languages. This lack of resources challenges those working on summarization in Indian languages.

[1] https://www-nlpir.nist.gov/projects/duc/guidelines/2002.html
[2] https://en.wikipedia.org/wiki/Wikipedia:Featured_articles

Even in the case of Multilingual datasets, many of them do not support Indian languages. There are very few summarization datasets available for Indian languages, of which the quality of the summaries is also very concerning.

[64] gives an overview of the datasets available for Indian languages to date. Most of the available datasets are obtained from news websites and books [5, 72]. IndicNLG released IndiccSentenceSummarization dataset as part of the summarization task where the first sentence (or) headline of the article is considered to be the summary [3]. Another recently released summarization dataset is TeSum [69], where they crowdsource summary generation for different Telugu news articles.

Apart from these, there are two multilingual datasets, XL-Sum and MassiveSumm, of where Indian languages are part. However, XL-Sum is merely a single-sentence summary of a large article. Furthermore, the MassiveSumm dataset has multiple resources where the summary of the article can be extractive or abstractive, headline or context of the article. In contrast, we would like to introduce a summarization corpora ISummCorp, a multilingual dataset with a concise summary.

2.3 Multilingual Models for Indian Languages

With many datasets available for English, it has received the majority of attention in summarization research so far. Additionally, due to the structural and morphological differences across languages, models developed using English datasets cannot be extended to Indian languages. For Indian languages, there have been relatively few attempts at summary, in contrast to English. Regarding multilingual summarization of Indic languages, [8] has introduced a sequence-to-sequence pre-trained model trained on 11 Indian languages, inspired by mBART [37] architecture. Another competitive model that can be used for the summarization of Indian languages is XL-Sum mT5 model [4]. XL-Sum mT5 model uses mT5 pre-trained model to finetune on XL-Sum dataset. We compare our IndicSumm with the multilingual models that support Indic languages and still achieve better results.

2.4 Background Literature

2.4.1 Graph Autoencoder (GAE)

Graph Autoencoder (GAE) is a neural network that learns the graph's structure. It encodes the graph's nodes into low-dimensional latent space and decodes them back to the graph structure. This makes GAE compress the information of graphs into low-dimensional latent space and learn them. One of the main benefits of GAEs is that they can capture the complex relationship between the nodes in a graph. So GAE can be trained in an unsupervised manner by minimizing the reconstruction error between the original and reconstructed graphs.

[3] https://huggingface.co/datasets/ai4bharat/IndicSentenceSummarization
[4] https://huggingface.co/csebuetnlp/mT5_multilingual_XLSum

Graph Autoencoder (GAE) [25] takes input, an undirected weighted graph $G := (V, A, X)$, where V is a set of N nodes $(v_1; v_2; ::v_N)$, $A \; 2 \; \mathbb{R}^{NXN}$ is a symmetric adjacency matrix representing node relationships and $X \; 2 \; \mathbb{R}^{NXD}$ is the node feature matrix. GAE obtains an encoding $Z \; 2 \; \mathbb{R}^{NXP}$, a reduced dimension space of X. The GAE model tries to reconstruct an Adjacency matrix $A^{'}$ close to the empirical graph (A) while Z captures the essential components of G. We stack an inner product decoder to reconstruct the graph as follows:

$$A^{'} = g(AH^{(1)} \quad _1) \tag{2.1}$$

$$Z = H^{(1)} = f(AX \quad _0) \tag{2.2}$$

where g is an activation function, and $\quad _0$, $\quad _1$ are the weights learned from the graph reconstruction. Fig. 3.2 briefly describes the workflow of the GAE model. The above equations help us to understand the working of GAE.

2.4.2 GRU

GRU(Gated Recurrent Unit) is a Recurrent Neural Network(RNN) type used in natural language processing and other sequence modeling tasks. It helps in capturing long-term dependencies. It has a more straightforward structure when compared to LSTM, as it has only two gates reset gate and an update gate, making it easier to train and faster to run. The reset gate and update gate are used to control the flow of information in the GRU. They are calculated using the below equations.

$$Resetgate : r_t = \quad (W_r \quad [h_t \quad _1; x_t] + b_r) \tag{2.3}$$

$$Updategate : z_t = \quad (W_z \quad [h_t \quad _1; x_t] + b_z) \tag{2.4}$$

$$h_t = (1 \quad z_t) \quad h_t \quad _1 + z_t \quad \tilde{h_t} \tag{2.5}$$

$$\tilde{h_t} = \tanh(W \quad [r_t \quad h_t \quad _1; x_t] + b) \tag{2.6}$$

where r_t and z_t are the reset and update gates at time step t respectively, $h_t \quad _1$ is the hidden state at the previous time step, x_t is the input at time step t, W_r and W_z are the weight matrices for the reset and update gates. b_r and b_z are the bias terms for the reset and update gates. As said, the reset gate and update gates control the flow of information, the reset gate determines how much past information to forget, and the update gate determines how much new input is to be included in the hidden state. The hidden state at the current time step h_t is then calculated as shown in the above equation where $\tilde{h_t}$ is the candidate hidden state. Moreover, b and W are bias term and weight matrices for the candidate hidden state. The main GRU is used to capture long-term information in sequence data. This makes GRU more powerful for NLP tasks such as summarization and machine translation.

Chapter 3

GAEISUMM: An unsupervised model

The creation of summarization datasets is an expensive and time-taking process. The lack of large-scale extractive summarization datasets made manually and needed as ground truth for training the networks are currently the most challenging barrier in implementing supervised deep learning algorithms. So, here we tackle this gap by utilizing techniques that do not require labeled data for training, i.e., an unsupervised methodology for summarization. We propose an unsupervised deep learning strategy based on graph auto-encoders and language embeddings.

3.1 Introduction

Document summarization aims to minimize the content in a text document and preserve the salient information. There are usually two categories of summarization techniques: Extractive [48] and Abstractive [56]. Extractive summarization extracts the document's salient text (e.g., words, phrases, or sentences). Whereas Abstractive summarization concisely paraphrases the information contained in the document.

Most extractive summarization methods rely on sentence scoring [1], keyword extraction [36], and clustering to identify the most important sentences in a document and output a summary. Sentence scoring is a technique that helps to rank sentences based on various features, such as the presence of specific keywords, the position of the sentence in the document, and the length of the sentence. The sentence with a high score is part of the summary. Keyword extraction involves generating a summary based on the presence of the most important words or phrases in the document. Coming to the clustering technique involves similar grouping sentences and selecting a representative sentence from each group that accounts for a summary.

Abstractive summarization methods involve natural language generation and neural network-based model techniques to generate new sentences that cover the document's meaning. One such approach is using Seq2Seq models [49], which are trained on large datasets of human-generated text to learn the structure and grammar of the natural language. These models generate a summary that conveys the article's primary motive.

Understanding the text's contextual and semantic representation for effective summaries is one of the main challenges in extractive document summarization. The traditional methods extract summary based on hand-crafted features, including Term Frequency [40], Sentence Position and Length [11], Keyword Extraction [10] and largely depend on the availability of NLP tools.

Significant progress has been made in single document extractive summarization by using recent popular deep learning models such as RNNs [48], CNNs [81], attention-based models [60], and sequence to sequence models [49]. The literature survey manifests that majority of the extractive summarization techniques rely on learning efficient text representations. Also, most of these models follow encoder-decoder-based approaches to generate summaries either in a supervised or unsupervised setting. [76] proposes a hierarchical transformer model to extract summaries. Similarly, [38] introduces a document-level encoder using transformers to generate summaries, minimizing the reconstruction loss. However, these models lack in capturing the global context and the long-distance sentence relationships present in a document. Graph-based methods help overcome these limitations by capturing the long-term dependencies when the document is modeled into a graph [24].

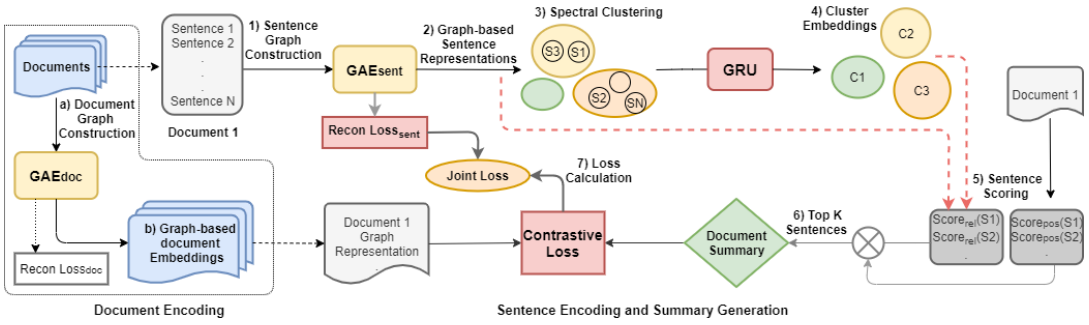

Figure 3.1: Outline of GAE-ISUMM. Our model GAE-ISUMM involves two phases: Document Encoding - a) Document Graph Construction and b) Obtaining Graph-based Representations; Sentence Embedding and Summary Generation involves seven steps starting from 1) Sentence Graph Construction to 7) Loss calculation.

Recently, Graph Convolutional Networks (GCN) have been successful in NLP and applied to various tasks such as text classification [78], semantic role labelling [43] and summarization [75]. GCN-based models can draw syntactic information, exploit long-range multi-word relations, and have been deployed on document-word relationships [38]. With the recently proposed method, Graph Autoencoder (GAE) [25] is used to capture the hidden semantic information between documents and sentences by using the idea of an autoencoder (AE) [62] to a graph-structured data. [21] proposes an unsupervised framework that learns sentence representations using a deep auto-encoder model. Few recent works have also obtained benchmark results on text classification using GAEs [73]. However, applying GAE for document summarization is an unexplored area. Also, adopting an AE or GAE trained on English

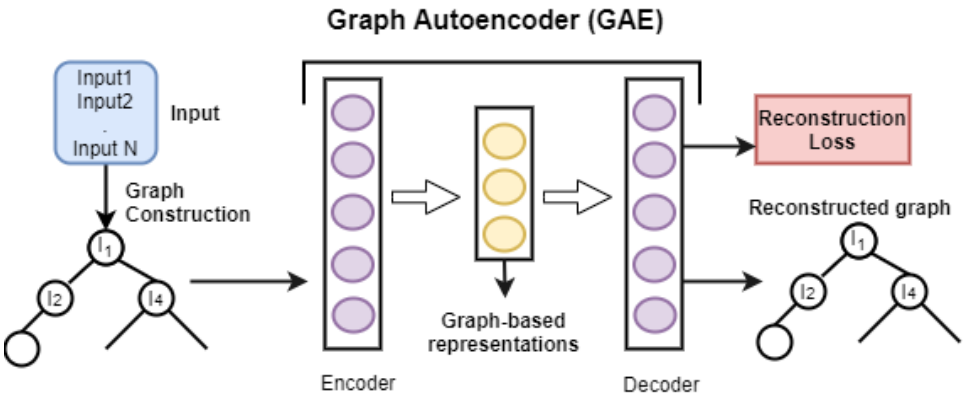

Figure 3.2: GAE model

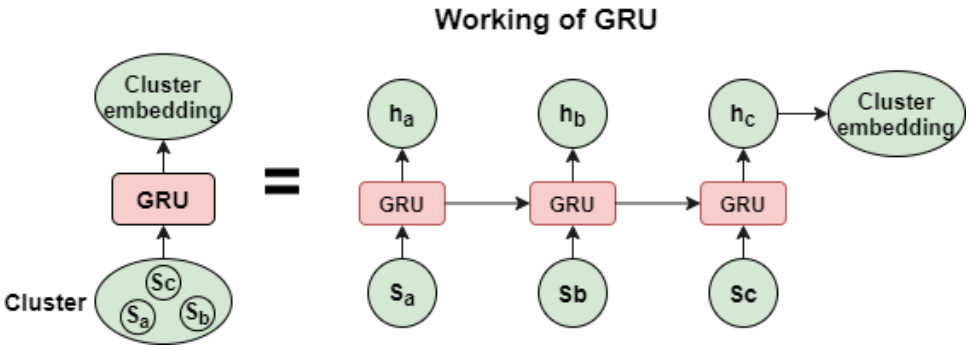

Figure 3.3: Working of GRU

corpora for Indian languages have significant limitations: (i) lack of such large-scale human-annotated datasets, (ii) rich morphological variation, and (iii) syntax and semantic differences.

Moreover, due to the dearth of various qualitative tools and scarcity of annotated data, summarization models are not well-studied for low-resource Indian languages.

Unlike English, few preparatory works have studied text summarization for Indian languages. In these works, the authors use traditional or baseline methods such as keyword extraction [47], hand-crafted features [14], and TextRank algorithm [42] to extract the summaries. In another recent work, [41] proposes a heuristic model based on the frequency score of named entities and the vocabulary of the document to produce summaries. Unfortunately, all these works were limited to existing baseline models that rank sentences based on term frequency or similarity heuristics. Also, the works focused on something other than learning sentence or document representations, which is crucial for an effective summarization model. In this scenario, unsupervised approaches are alluring as they do not require any labelled data for training.

Inspired by the GAE [25], SummPip [82], and Salience Score Estimation [79], we propose GAE-ISUMM: an unsupervised extractive summarization model that jointly optimizes the loss between doc-

ument representations, sentence representations and generated summaries to showcase better performance.

Most importantly, our proposed model simultaneously learns the sentence representations and summary of the document, while earlier methods extract summaries after learning the sentence representations. Fig. 3.1 illustrates our proposed method, GAE-ISUMM.

Our main contributions are as follows:

- We propose GAE-ISUMM, an unsupervised model that learns a summary from the input document. Also, to the best of our knowledge, we are the first to apply GAE for the summarization task.
- We formulate the problem as a graph network to learn sentence and document representations using GAE and perform text summarization jointly with our proposed method.
- To the best of our knowledge, we are the first to investigate the effectiveness of graph-based embeddings for different Indian languages in an unsupervised setting.
- We further introduce TELSUM, a manually annotated Telugu summarization dataset of 501 document-summary pairs.

In this chapter, we aim to bridge the gap by creating models and resources for the summarization task of Indian languages. The proposed method, GAE-ISUMM, can be extended to other resource languages that are closer to Indian languages culturally and linguistically by translating this resource without losing the rich morphological variations.

Further in this chapter, we first discuss TELSUM and the different datasets we used for experiment purposes. Later, we explain all the components of GAE-ISUMM in detail, followed by Experimental Results and Analysis.

3.2 Datasets

Most of the summarization datasets in Indic languages are web scraped, and they consider the text's headline or gist as summary [18]. Table 3.1 reports the statistics of different summarization datasets available for Indian languages. From Table 3.1,we notice that the summary length is either too long or too short compared to the document size for many Indian language datasets. This indicates that the summary needs to be more representative of the document. Unlike existing datasets, we introduce a dataset TELSUM for Telugu that is manually annotated (human-written summaries) with the sole aim of creating a gold-standard summarization dataset to evaluate GAE-ISUMM. [1] We first discuss in detail about TELSUM and later briefly explain the different summarization datasets used for experiments.

[1]Dataset is created only for the Telugu language due to time and resource constraints.

3.2.1 TELSUM Dataset

TELSUM dataset is a manually-annotated summarization dataset for Telugu created to evaluate GAE-ISUMM. Here, we describe the dataset collection process, preprocessing steps, and annotation details of TELSUM.

3.2.1.1 Dataset Collection and pre-processing

For TELSUM, we scraped a total of 4020 documents (news articles) from a Telugu news website *samyam*[2]. After crawling the documents, we cleaned and preprocessed the data by removing the unwanted URLs, hashtags, hyperlinks, English text, and documents with very few sentences (<5). From this, we obtained a total of 3098 documents, of which 2597 documents are used in training the GAE-ISUMM in an unsupervised setting, and the remaining documents (501) form the test dataset, henceforth referred to as TELSUM[3]. Our TELSUM dataset consists of 501 document-summary pairs. These summaries were manually written by two professional annotators who are native Telugu speakers. Three highly proficient Telugu native speakers verified the written summaries regarding readability, relevance, and coverage. Further, we use these annotated summaries to evaluate our model GAE-ISUMM. The privacy details, fair compensation for annotators, and ethical concerns are discussed in Section 3.7.

3.2.1.2 Annotation Details

Each annotator was given a set of guidelines and asked first to write sample summaries. The written summaries were verified and asked to annotate the whole dataset TELSUM. Set of Guidelines: The data is randomly selected and divided among the two annotators. Each of the annotators was provided with very detailed guidelines on how to annotate summaries. We briefly mention here all the factors that are considered while annotating: Relevance and Coverage (if the summary contains all the essential aspects of the article), Compression ratio (length of summary with respect to article), Creativity (how abstract the summary is). The annotators should not include bias in any form in summary from their perspective.

3.2.2 Other Indian Langauge Datasets

3.2.2.1 NCTB and BNLPC:

NCTB [5] is a Bengali abstractive dataset collected from Bangladesh textbooks. Similarly, BNLPC [17] is a Bengali extractive dataset from daily newspapers. These are the only datasets publicly available for different languages. BNLPC and NCTB summaries are handwritten by professionals.

[2]https://telugu.samayam.com/

[3]https://github.com/scsmuhio/Summarization

Dataset	Lang	#Docs	Avg len of doc	Compression ratio
TELSUM	Telugu (te)	501	17.43	0.170
NCTB	Bengali (bn)	200	08.06	0.518
BNLPC	Bengali (bn)	139	12.78	0.129
Marathi	Marathi (mr)	100	14.80	0.513
Hindi Short	Hindi (hi)	66,367	17.91	0.056
XL-sum	Telugu (te)	11,308	31.14	0.032
	Bengali (bn)	8,226	41.53	0.024
	Tamil (ta)	17,846	33.59	0.030
	Gujarati (gu)	9,665	49.68	0.020
	Punjabi (pa)	8,678	41.02	0.020
	Marathi (mr)	11,164	55.13	0.018
	Hindi (hi)	51,715	29.79	0.036

Table 3.1: Statistics of TELSUM and different available summarization datasets in Indian languages. Here, the Compression ratio is the average length of the summary to the average document length.

3.2.2.2 XL-Sum:

It is a comprehensive and diverse dataset [18] extracted from the BBC website. The XL-SUM dataset can be an extension of the XSum [50] dataset, where they extract only English articles from the BBC website. The dataset covers 44 languages, of which we consider seven Indian languages for our monolingual experiments. We also use the English XL-sum dataset for our cross-lingual experiments. The dataset uses contextual information or the gist as the summary. The XL-sum summaries are of a single sentence that sometimes fails to convey the summary of the article or any contextual information required to understand the article.

3.2.2.3 Marathi, and Hindi Short Summaries:

Marathi dataset is extractive and collected from a news website [4]. Hindi short summarization dataset consists of 330k articles scraped from a news Hindi website [5]. This dataset considers the headline as the summary with the article as the input text document.

[4]https://tinyurl.com/mtz473dr
[5]https://tinyurl.com/4pd86b55

3.3 GAEISUMM Model

The overall pipeline of our proposed model GAE-ISUMM is described in Fig. 3.1. Our proposed model involves training in two phases: 1) document encoding and 2) sentence encoding and summary generation. Each of these two phases is trained separately. Before that, we briefly try to explain the construction of different graphs. The following subsections explain all the key components of GAE-ISUMM.

3.3.1 Graph Construction:

The GAE-ISUMM incorporates graph construction at two levels - one at the sentence level (each node representing a sentence in the sentence graph) and another at the document level (each node representing a document in the document graph), as shown in Fig. 3.1. We use cosine similarity to identify the relationship between the nodes, which helps us build the graph's adjacency matrix. The goal of graph construction is to capture the global context either within the document (sentence-level graph) (or) across all the documents (document-level graph).

3.3.2 Document Level Graph Construction and Encoding:

To build a document-level graph across all the documents, first, we obtain sentence representations for each document using available pre-trained languages models [9, 7, 77, 22, 44, 45] as described in sentence encoding and summary generation. To get the document representation, we map each node or document D_j to a fixed-length vector (X_{doc}) by averaging all the sentence representations in D_j. Each document D_i (or) node is mapped to a fixed-length vector using the pre-trained language models [44]. First, We obtain sentence mappings as described in 3.3.3. Finally, the document-level graph is fed into GAE$_{doc}$ to obtain the graph-based latent document representations (Z_{doc}). GAE$_{doc}$ is a Graph Auto-encoder at the document level. It takes input from a graph and outputs graph-dependent values for each node. The graph is constructed based on similarities of the embeddings of the node, i.e., document here. GAE_{doc} helps in learning representations at a global level from all other documents. The GAE$_{doc}$ model is trained independently by minimizing the reconstruction loss of document level graph (Recon loss$_{doc}$). These obtained latent document representations (Z_{doc}) are further used in the GAE-ISUMM model while calculating contrastive loss.

3.3.3 Sentence Encoding and Summary generation

Here, we present the following details: (i) the sentence-level graph construction, (ii) sentence encoding, (iii) clustering and cluster embeddings, (iv) sentence scoring and selection, and (v) loss calculation. The sentence encoding and summary generation steps are processed at a single document level.

3.3.3.1 Sentence Level Graph Construction:

We build a sentence-level graph for each document D := $(S_1; S_2; ::; S_N)$ where each sentence S_i is considered a node. To obtain a sentence-level graph, we use existing Indic pre-trained language models [22, 44, 45] and various multilingual pre-trained models [9, 7, 77] to get a fixed-length vector representation for each sentence. The pre-trained language models help us detect high-level contextual features capturing precise semantic and syntactic relationships. We also investigate monolingual distributed word embeddings and pre-trained language models available for Telugu language [45]. After mapping each node to a fixed-length vector, the sentence-level graph is fed into GAE_{sent} to obtain latent graph-based representations (Z_{sent}) of each sentence in the document. GAE_{sent} is a Graph Auto-encoder at the sentence level. It takes input from a graph and outputs graph-dependent values for each node(sentence). The graph is constructed based on similarities of the embeddings of the node, i.e., a sentence here. GAE_{sent} learns the sentence embeddings from the surrounding sentences, which helps to gain more contextual information inside a document.

3.3.3.2 Sentence Encoding:

Word embeddings represent words in a continuous vector space so that semantically similar words are close to each other in vector space. There are various types of word embeddings, some of which are listed below. The details of different types of sentence encoding are discussed below:
Distributed Telugu Word Embeddings:

- **Word2vec-Te**: It is the short form for Word2vec Transfer Embedding, a variant of popular Word2vec embedding specifically designed for transfer learning. In transfer learning, the goal is to transfer knowledge from a pre-trained model to a new task to improve the new model's performance. Word2vec-Te was designed to be easy for transfer learning by providing pre-trained word embeddings that can be fine-tuned on a new task.
- **Glove-Te**: It is similar to Word2vec but a variant of GloVe embedding and designed for transfer learning. It is a count-based word embedding and was developed by Stanford University. It can also be fine-tuned for a specific task.
- **FastText-Te**: It is a variant of a pre-existing word embedding FastText designed for transfer learning. FastText was developed by Facebook, which can be fine-tuned for the desired tasks.
- All the above-distributed word embedding models are trained on a large Telugu dataset of 8 million sentences [45]. We average the word embeddings in the sentence to obtain the sentence representation.

Pre-trained Telugu Transformer Language Models: We use the monolingual pre-trained Transformer models such as BERT-Te, Albert-Te, Roberta-Te, and DistilBERT-Te available for Telugu [45]. These Pre-trained language models directly output sentence embeddings.

- **BERT-Te:** BERT-Te is based on the BERT model, which is a transformer-based model. BERT-Te is that it can capture the context-dependent meaning of words in the Telugu language.
- **Albert-Te:** Albert-Te is a variant of the ALBERT model, a transformer-based language representation model developed for the Telugu language. It captures the context-dependent meaning of words in the language.
- **Roberta-Te:** Roberta-Te is a variant of the RoBERTa model, a transformer-based model developed specifically for the Telugu language.
- **DistilBERT-Te:** It is similar to BERT and based on the DistilBERT model. DistilBERT-Te is trained on large amounts of Telugu text data and can capture the context-dependent meaning of words in the language.

Multilingual Pre-trained embeddings

- **IndicBERT**: IndicBERT [22] is a multilingual pre-trained Transformer language model created for twelve Indian languages. It is trained on an Indic-Corp [22]. We use these IndicBERT embeddings to obtain sentence-level representations. Here, the sentence representation is obtained by taking the average of the token representations from the last hidden state.
- **mBERT**: mBERT [9] is a multilingual version of BERT trained in 104 languages from Wikipedia. By jointly conditioning on both left and right context in all layers, BERT is intended to pre-train deep bidirectional representations from unlabeled text, in contrast to previous language representation models.
- **XLM-R**: XLM-R [7] is a transformer-based multilingual masked language model pre-trained on Common Crawl [7] data in 100 languages. XLM-R is created by training a Transformer-based masked language model on a hundred languages.
- **mT5**: mT5 [77] is a multilingual variant of T5 model pre-trained on CommonCrawl dataset [77] which contains 101 different languages. A multilingual variant of T5 was created to leverage a unified text-to-text format.

3.3.3.3 Clustering and Cluster Embeddings:

A document usually consists of multiple events (or) a series of events; we believe that clustering on the document helps to segregate better and understand the document. We generate cluster representations to incorporate the cluster information in the final summary. We applied spectral clustering [52] on the latent sentence representations Z_{sent} obtained from GAE_{sent}, where the spectral clustering method partitions a document of N sentences into M clusters $(c_1; c_2; :; c_M)$.

For each cluster c_i with jc_ij sentences, the GRU [4] mechanism outputs a cluster embedding C_i on top of sentence embeddings in cluster c_i (please refer Fig. 3.3). Here, the sentences are passed into

GRU according to their relative position in the document. Finally, we extract the last hidden state h_{jc_ij} of GRU to obtain cluster embedding C_i, as shown in Equation 3.2. This cluster embedding has a semantic overview of the entire cluster, which helps to capture significant text.

$$h_t = \text{GRU}(h_{t-1}; S_t^i) \tag{3.1}$$

$$C_i = h_{jc_ij} \tag{3.2}$$

where S_t^i represents sentence at t^{th} time unit in cluster c_i.

3.3.3.4 Sentence Scoring and Selection:

For each sentence S_i of cluster c_j in document D, we estimate the sentence score using two criteria, (i) Sentence relevance score ($score_{rel}$) and (ii) Sentence position score ($score_{pos}$). To estimate $score_{rel}$, we first calculate weighted relevance scores using the $f(S_i; D)$ in Equation (3.3) similar to attention mechanism [71]. Later, the scores are normalized via Softmax to obtain $score_{rel}$ as shown in Equation (3.4).

$$f(S_i; D) = !^T \tanh(W_1 \quad Z_{sent}^{S_i} + W_2 \quad C_j) \tag{3.3}$$

$$score_{rel}(S_i; D) = \frac{f(S_i; D)}{\sum_{S_p 2 C_j} f(S_p; D)} \tag{3.4}$$

where $Z_{sent}^{S_i}$ denotes the graph-based latent sentence representation of S_i, C_j represents its cluster embedding, and $f!$, $W1$, $W2g$ are trainable parameters. Inspired from [21], $score_{pos}$ (refer Equation (3.5)) is calculated based on the relative position $P(S_i)$ 2 [1,2,..,N] of sentence S_i in the Document D := $(S_1; S_2; :::; S_N)$. Sentences at the start of the document are given high priority than the rest as they provide more relevant information about the entire document [33]. The final sentence score is calculated in Equation (3.6).

$$score_{pos}(S_i; D) = \max\left(0{:}5; exp\frac{P(S_i)}{\frac{}{N}}\right) \tag{3.5}$$

$$\text{score}(S_i; D) = \quad \text{score}_{rel}(S_i; D) + \quad \text{score}_{pos}(S_i; D) \tag{3.6}$$

The variables in Equation (3.6): , 2 [0,1] with + =1, assign relative weights to $score_{rel}$ and $score_{pos}$, respectively. In every iteration, we sort the sentences in descending order of their sentence scores, and the top K sentences are considered our predicted summary ($\hat{S}$) of the document. We obtain our final candidate summary whenever our model reaches the local minima solution.

3.3.3.5 Loss calculation:

We estimate the contrastive loss [15] between the graph-based latent document representations (Z^D_{doc}) and the average of all sentence representations in the candidate summary. The final loss L is calculated as follows:

$$L = \text{Reconstruction loss}(GAE_{sent}) + \text{Contrastive loss}(\hat{S}; Z^D_{doc}) \tag{3.7}$$

3.4 Experimental Setup

This section describes GAE-ISUMM training setup, hyper-parameter tuning, evaluation metrics, and analysis of results.

3.4.1 Model Training Setup & Hyperparameters:

Here, we describe the training details of document encoding (GAE_{doc}), sentence encoding (GAE_{sent}), and summary generation over all documents. In our GAE-ISUMM, we train each phase (GAE_{doc}, GAE_{sent} and summary generation) separately. The final training was performed by minimizing the joint loss (Reconstruction loss + Contrastive loss) over all the documents, as mentioned in Equation (3.7).

The model trainable parameters in GAE_{doc}, GAE_{sent}, GRU, and sentence scoring weight parameters $f!$, $W1$, $W2g$ are described below. To perform document summarization using GAE-ISUMM, we set the first convolution layer's embedding size as 128 (for Word2Vec-Te and FastText-Te), 32 (for Glove-Te) and 256 (for remaining embeddings) for both GAE_{doc}, GAE_{sent}. The input feature vector for the summarization task is extracted from Telugu pre-trained embeddings (Word2Vec-Te, FastText-Te, Glove-Te, BERT-Te, RoBERTa-Te, ALBERT-Te) as well as from multilingual language models (mBERT(multilingual BERT), IndicBERT, XLM-R). Since IndicBERT shows superior performance over other multilingual models [22], in this chapter, we use IndicBERT for input feature extraction for all other experiments on Indian languages.

We use Adam optimizer with an initial learning rate of 0.001 to train GAE_{doc} with a two-layer of GCNConv with 768,256 as input and output dimensions, respectively. We use the scikit-learn package of Spectral Clustering to perform sentence graph clusterization. After experimenting with various values, the number of clusters is considered the average number of sentences in annotated summaries. The joint loss function is optimized with an Adam optimizer, a learning rate of 0.0005, weight decay of 0.0005, a hidden filter size of 128 and an output dimension of 384. It involves 4 GCNConv layers. We experimented with a range of values to determine the choice of and . The model was effective when =0.6 and =0.4. To extract the summary, we chose the value of K (number of sentences in the predicted summary) based on the number of clusters. We trained each phase of the model to a maximum

Model		R-1	R-2	R-L
TextRank		36.77	22.14	34.30
LexRank		38.89	26.65	36.45
SumBasic		34.97	20.24	33.65
KL Greedy		33.85	18.71	31.77
GAE-ISUMM	Word2Vec	42.81	30.13	41.53
	Glove	42.81	30.13	41.53
	FastText	42.49	33.42	41.49
	BERT-Te	45.12	34.81	43.83
	ALBERT-Te	44.45	31.78	43.20
	Robert-Te	45.84	37.19	44.69
	distilbert	43.96	33.99	42.87
	XLM-R	45.24	**38.49**	44.14
	Indicbert	**46.29**	36.6	**44.9**
	mbert	44.12	34.72	41.95
	mT5	45.47	35.08	43.83
	mbart	44.86	32.9	42.27

Table 3.2: Comparison of ROUGE score results of GAE-ISUMM with other methods on TELSUM dataset.

of 40 epochs which took around two days. The experiments were performed on a single V100 16GB RAM GPU machine.

3.4.2 Evaluation Metrics:

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a metric to evaluate the quality of summaries of the text. ROUGE compares a generated summary to one or more reference summaries and produces a score based on the overlap between the two. We use ROUGE [34] F1-metric (ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L)) to evaluate our model. ROUGE-N score refers to the N-grams overlap between the candidate and gold summary (human reference summary). In contrast, ROUGE-L refers to the longest matching sub-sequence of the candidate and gold summary. The higher ROUGE score indicates that the candidate summary is more similar to the gold summary.

Metric	R-1			R-2			R-L		
Language# / Method !	MB	IB	GAE-Summ	MB	IB	GAE-Summ	MB	IB	GAE-Summ
bn	**26.81**	25.27	22.10	**10.57**	9.55	9.20	**22.45**	21.51	20.10
gu	21.49	21.66	**22.20**	7.48	7.43	**8.40**	19.08	19.39	**19.43**
hi	**39.72**	38.25	30.10	**17.46**	16.51	13.83	**32.46**	31.48	26.99
mr	21.46	22.26	**22.75**	9.53	9.94	**9.61**	19.26	20.08	**20.14**
pa	28.15	**30.28**	23.50	10.30	**11.88**	8.7	22.75	**24.38**	17.98
te	16.16	16.39	**16.92**	4.95	5.40	**6.87**	14.36	14.71	**15.25**
ta	22.47	21.79	**22.60**	**10.22**	9.75	10.1	20.33	19.67	**20.90**

Table 3.3: GAE-ISUMM on XL-sum: ROUGE score results on seven different Indian languages. These results are compared with mBART (MB) and IndicBART (IB) results from [8].

3.5 Results and Analysis

The effectiveness of our proposed model is evaluated by comparing it with several existing baselines such as Textrank [46], LexRank [11], SumBasic [51], and KL Greedy [16]. In particular, we analyze our GAE-ISUMM method on TELSUM as well as on other existing Indian language datasets. We also describe the performance of each component of our proposed method in the ablation study.

3.5.1 Performance on TELSUM Dataset:

Table 3.2 reports the ROUGE scores of the baseline methods and our proposed method GAE-ISUMM with various pre-trained language model features as input. We make the following observations from Table 3.2: (i) All the baseline methods showcase lower ROUGE scores compared to GAE-ISUMM because these methods follow simple heuristics based on sentence similarity. (ii) The LexRank model reports a higher ROUGE score among all the baseline models. (iii) GAE-ISUMM displays its effectiveness by outperforming all the baselines with any of the Telugu or multilingual pre-trained models. Henceforth, we argue that graph-based representations capture global (document-level) and local context (sentence-level) information. (iv) All the pre-trained transformer-based models performed similarly, while IndicBERT and XLM-R report the highest ROUGE scores for R-1, R-L, and R-2, respectively.

Fig. 3.4 shows the human-annotated summary and the summary predicted by GAE-ISUMM for an example article from TELSUM dataset. From Fig. 3.4, we observe that GAE-ISUMM extracts all the essential named entities and highly coincide with the manual summaries in terms of coverage and consistency.

Human reference summary	కిందటి సంవత్సరంతో కన్నా ఈ సంవత్సరం భారత లో కరోనా మరింత భయంకరం గా మారింది. ఇటీవలే శ్రవణ్ కుమార్ రాథోడ్ (66) అనే ప్రముఖ బాలీవుడ్ సంగీత దర్శకుడు కరోనా బారిన పడి మృతిచెందారు. ఆయన మరణ వార్త తెలిసి బాలీవుడ్ స్టార్ హీరో అక్షయ్ కుమార్ సోషల్ మీడియా వేదికగా ప్రగాఢ సంతాపం తెలిపారు. Compared to previous year, corona has become more awful this year in India. Recently, Shravan Kumar Rathore(66), a popular music director has died affected by corona. Bollywood star hero Akshay Kumar has taken to social media to mourn the news of his death.
Predicted summary	దేశంలో కరోనా మహమ్మారి ఉగ్రరూపం దాల్చుతుంది. ఈ పరిస్థితుల్లో కరోనా బారిన పడి ప్రముఖ బాలీవుడ్ మ్యూజిక్ డైరెక్టర్ శ్రవణ్ కుమార్ రాథోడ్ (66) కన్నుమూశారు. ఈ విషాద వార్త విని పలువురు బాలీవుడ్ సినీ ప్రముఖులు ఆయన మృతి పట్ల సంతాపం తెలుపుతున్నారు. The corona epidemic is raging in the country. Leading Bollywood music director Shravan Kumar Rathore, 66, has died after suffering from corona. Many Bollywood celebrities are mourning his death on hearing this tragic news.

Figure 3.4: An Example of human annotated summary and GAE-ISUMM predicted summary from TELSUM

3.5.2 Analysis of Clusters:

As mentioned in the previous section, the number of clusters formed for each document depends on the dataset's average length of the annotated summaries. Fig. 3.5 represents the clusters formed for an example article from TELSUM dataset. The example article is about a movie that was yet to release. Of the 3 clusters formed, the first cluster talks about the expected release date of the movie and the main crew involved. The second cluster talks about the male lead and the characterization of the male lead. The third cluster talks about the female-lead cast and her previous career details. The clusters formed are specific to certain content of the article and don't overlap with the other cluster. Translation of the 3 clusters in the English language is shown in Fig. 3.6 From these clusters formed, we can clearly state that the inclusion of cluster formation helps us get full coverage of the article and helps us remove redundant information (sentences with similar semantics) present in the article.

Clusters	Sentences
Cluster 1	1) నేచురల్ స్టార్ నాని రీసెంట్‌గా దేవదాస్ అనే చిత్రంతో మంచి హిట్ కొట్టగా , తన తదుపరి చిత్రాన్ని ప్రముఖ చిత్ర నిర్మాణ సంస్థ సితార ఎంటర్‌టైన్‌మెంట్‌లో జెర్సీ అనే టైటిల్‌తో చేస్తున్నాడు. 2) మళ్ళీరావా ఫేం గౌతమ్ తిన్ననూరి దర్శకత్వంలో ఈ చిత్రం రూపొందుతుంది. 3) ఈ నెల 18న లాంచింగ్ కానున్న ఈ చిత్రం చివరి వారంలో రెగ్యులర్ షూటింగ్ జరుపుకోనుంది. 4) ఈ చిత్రానికి అనిరుధ్ రవిచంద్రన్ సంగీతం అందిస్తున్న సంగతి తెలిసిందే.
Cluster 2	1) క్రికెటర్‌గానే కాదు పెళ్ళయిన నడివయస్కుడు, ముసలి వ్యక్తిగా కూడా నాని ఈ చిత్రంలో కనిపించనున్నాడని చెబుతున్నారు. 2) ఈ పాత్ర కోసం నాని రోజుకు 3గంటలు క్రికెట్లో శిక్షణ తీసుకుంటున్నాడట. 3) నాని మూడు పాత్రలని చాలెంజ్‌గా తీసుకొని నటిస్తున్నాడని చెబుతున్నారు. 4) కలని అందుకోవాలంటే ఆలస్యం చేయొద్దు అని జెర్సీకి ట్యాగ్ లైన్‌గా ఉంచారు.
Cluster 3	1) అయితే పీరియాడిక్ చిత్రంలో నాని సరసన ఇద్దరు భామలు నటించనున్నట్టు తెలుస్తుంది. 2) కన్నడలో యూటర్న్ సినిమాతో సూపర్ హిట్ అందుకున్న శ్రద్దా శ్రీనాథ్‌తో పాటు మలయాళ నటి రెబ్బా మోనికా జాన్ మరో హీరోయిన్‌గా నటించనుందట. 3) మలయాళం, తమిళంతో కలిపి నాలుగు సినిమాల్లో నటించిన మోనిక 'జెర్సీ' సినిమా ద్వారా తెలుగులో పరిచయం కానున్నారు.

Figure 3.5: Analysis of clusters: the clusters formed by GAE-ISUMM for an example article from TELSUM.

3.5.3 Ablation Studies:

To investigate the importance of each of the components present in our model GAE-ISUMM, we conduct several ablation experiments and compare them with the benchmark result obtained in Table 3.6. The ablation studies we conducted are: (i) GAE-ISUMM without GAE_{sent} and GAE_{doc}, where the model doesn't include GAE at any stage. The whole model uses pre-trained embeddings extracted from the pre-trained models instead of graph-based embeddings. (ii) without clustering i.e., there is no cluster information sent for sentence scoring, and (iii) removal of sentence position scores, i.e., the top K sentences are scored only based on Sentence relevance score ($score_{rel}$).

From Table 3.6, we observe that, in the first case, with the removal of GAE_{sent} and GAE_{doc} components; the model was not able to learn effective representations of the article and display lower ROUGE scores. In the second aspect, we train our model without clustering, i.e., we estimate $score_{rel}$ without any cluster information in Equation (3.3). The results reflect that, without clustering, the model fails to capture the complete significant information from the document. In the third aspect, we remove the sentence position score from the final sentence score estimation in Equation (3.6). We observe that removing sentence position yields a relative drop of 6.8% in R-1, proving the importance of sentence position while generating summaries. Previous studies highlight the importance of sentence position while extracting a summary. [54]

Clusters	Sentences
Cluster 1	1) Natural star Nani recently received massive success with his film Devdas and is doing his next film titled Jersey under the famous production company Sitara Entertainment. 2) Gautam Tinnamuri, who received fame with the Malliraava movie, will be directing this film. 3) The film will be launched on the 18th of this month and will have regular shooting in the last week. 4) It is known that Anirudh Ravichandran composed music for this film.
Cluster 2	1) It is said that Nani will be seen not only as a cricketer but also as a married man and an older man in this film. 2) Nani is training in cricket for 3 hours daily for this role. 3) It is said that Nani is taking all three roles as a challenge. 4) The tagline of the jersey is "Don't delay to achieve your dream."
Cluster 3	1) But it is known that two female leads will act along with Nani in the period film. 2) With Shraddha Srinath, who got a super hit in Kannada with the movie U-turn, Malayalam actress Rebba Monica John will play the other female lead. 3) Monica, who has acted in four films in Malayalam and Tamil, will make her debut in Telugu through the film 'Jersey.'

Figure 3.6: Clusters formed by a sample article from TELSUM dataset(English version).

Human reference summary	చైనా ప్రభుత్వం నుంచి అమెరికాకు అతిపెద్ద దీర్ఘకాలిక ముప్పు పొంచి ఉందని అమెరికా నిఘా ఏజెన్సీ ఎఫ్‌బీఐ డైరెక్టర్ క్రిస్టఫర్ రే అన్నారు. Christopher Ray, director of the U.S. intelligence agency FBI, said the biggest long-term threat to the U.S. was posed by the Chinese government.
Predicted summary	ఎఫ్‌బీఐ డైరెక్టర్ క్రిస్టఫర్ రే "చైనా ప్రభుత్వ గూఢచర్యం, డేటా చోరీ వల్ల అమెరికా భవిష్యత్తుకు ఎప్పుడూ లేనంత దీర్ఘకాలిక ముప్పు ఉంది" అని వాషింగ్టన్ లోని హడ్సన్ ఇన్స్టిట్యూట్ లో మాట్లాడిన ఎఫ్‌బీఐ డైరెక్టర్ క్రిస్టఫర్ రే అన్నారు. Christopher Ray, director of the FBI says "China's government act of spying and data theft pose an unprecedented threat to the U.S. future," at the Hudson Institute in Washington.

Figure 3.7: An Example of Actual and Predicted summary on XL-sum.

Metric	R-1		R-2		R-L	
Dataset# / Method *!*	GAE-ISUMM	State-of-art	GAE-ISUMM	State-of-art	GAE-ISUMM	State-of-art
BNLPC [5]	**71.8**	61.6	**66.8**	56.5	**71.2**	61.1
NCTB [5]	**12.33**	12.17	**3.19**	1.92	10.99	**11.35**
Marathi	**80.5**	64.8	**75.5**	59.1	**80.0**	66.1
Hindi Short	**20.2**	-	**12.0**	-	**16.1**	-

Table 3.4: ROUGE score results on Other Summarization Datasets using GAE-ISUMM. Here '-' indicates that the dataset has no state-of-art results.

Metric	R-1		R-2		R-L	
Lang-2# / Lang-1 *!*	hi	en	hi	en	hi	en
bn	**23.26**	21.49	**9.6**	9.21	**21.45**	19.5
gu	**22.49**	19.95	**7.48**	7.03	**19.16**	16.14
hi	-	29.56	-	12.78	-	24.07
mr	**21.89**	19.13	**8.78**	8.26	**19.72**	18.93
pa	23.76	**24.1**	8.68	**8.89**	**18.34**	17.41
te	**17.81**	16.55	**7.89**	7.83	**15.43**	14.85
ta	**22.36**	21.24	**9.79**	9.1	**20.1**	19.54

Table 3.5: Cross-lingual experiments of GAE-ISUMM: Bilingual setting of "hi" and "en" with the other Indian languages from XL-sum. Here '-' represents no cross-lingual experiment for that permutation of languages.

3.5.4 Performance on XL-Sum (Seven Indian Languages):

We compare the performance of our model GAE-ISUMM on the multilingual XL-Sum [18] dataset. For this dataset, the summaries and the articles were extracted from the BBC news website for 44 different languages. We observe that these summaries don't capture the complete information in the document since most of the summaries from this dataset are of single line and the article length is huge [69]. To evaluate our model performance, we consider seven Indian language datasets ("te, ta, gu, pa, bn, mr, hi") from XL-Sum and compare our GAE-ISUMM performance with the current state-of-art multilingual approach proposed by [8]. We use IndicBERT as our feature extraction model to obtain the sentence and document representations and consider the top one (or) two scored sentences (according to the dataset average summary length) predicted by our model. Table 3.3 illustrates the results obtained

Ablation studies on GAE-ISUMM	R-1	R-2	R-L
Without GAE_{sent} and GAE_{doc}	42.94	33.04	39.19
Without clustering	44.61	31.27	41.09
Without $score_{pos}$	43.14	32.97	39.95

Table 3.6: Ablation studies of GAE-ISUMM on TELSUM dataset.

on the XL-Sum dataset. From Table 3.3, we observe that GAE-ISUMM outperformed the state-of-the-art models for four Indian languages ("gu, mr, te, and ta"). In particular, we notice that our proposed model performed well on the Dravidian languages ("te, ta") and two of the Indo-Aryan languages ("gu, mr"). Although R-2 scores of GAE-ISUMM showcase close performance to mBART and IndicBART for "bn, hi, and pa", however, GAE-ISUMM report lower scores in the case of R-1 and R-L. The lower performance can be mainly due to the fact that the document size of XL-Sum datasets ranges from 6 to 110 sentences. However, their gold summaries extracted are confined to one or two sentences (under-representation of the document). Overall, we compare our unsupervised model with their supervised multilingual setting and yet achieve competitive or better results.

3.5.5 Does Cross-Lingual Models Improve Summarization Performance?

Generally, multilingual pre-trained models are usually evaluated by their capacity for knowledge transfer across languages. To investigate the cross-lingual language transfer, we also experiment with our GAE-ISUMM in bilingual settings on the XL-Sum dataset. Bilingual summarization can be done either by training the model on a single language (high-resource language) alone and testing on the second language (low-resource language) or by training on both languages and testing on the second language. This allows the model to benefit from the high-resource languages. To better assess the usefulness of the proposed dataset TELSUM and the existing multilingual dataset (XL-Sum), we evaluate our GAE-ISUMM in a cross-lingual setting.

3.5.6 Experiments in Cross-lingual Setting:

Here, we performed our cross-lingual experiments by considering "Hindi (hi)" and "English (en)" from XL-Sum dataset as the high-resource languages and "bn, gu, mr, pa, ta, and te" to be the low-resource languages. We trained on both the high and low-resource languages and tested on the low-resource languages. We also experimented by considering "hi" as the low-resource language and "en" as the high-resource language. From Table 3.5, we observe that training in a bilingual setting has improved ROUGE scores. The bilingual setting of "hi" with "te, bn, gu, pa" has improved ROUGE scores compared to the monolingual setting. Also, the bilingual setting of "hi" with other languages

performed better than "en" due to the morphological and structural similarity of Indian languages. We report the cross-lingual transfer results of the TELSUM dataset (refer Table 3.6).

3.5.7 Performance on Other Summarization Datasets:

We employ our GAE-ISUMM model on other existing summarization datasets of different Indian languages, as reported in Table 3.4. Observations from Table 3.4 that GAE-ISUMM yield superior performance on all the datasets with high ROUGE scores (R-1, R-2, and R-L). Therefore, we argue that our GAE-ISUMM model can be generalizable to any resource-poor language dataset.

This work proposes a novel unsupervised summarization model that explores the strong representation power of neural networks, RNNs, and graph representations. We build a model that learns sentence and document representations using GAE and cluster representations from GRU. Further, our model ranks the sentences based on their position, significance, and semantic value. Our experiments prove that GAE-ISUMM outperforms all the baseline models and reports benchmark results on the TELSUM. We also test the performance of our model in other Indian languages with the help of existing summarization datasets. In the future, we plan to introduce an abstractive model taking summarization a step forward in low-resource Indian languages.

3.6 Conclusion

This chapter addresses the robust representational capabilities of graph-based techniques, RNNs, and neural networks by proposing a unique unsupervised summarization model. Using cluster representations from GRU and GAE, we create a model that learns text representations and a document summary. The positions, relevance, and semantic value of the sentences are also considered while ranking them in our approach. Our experiments showcase that GAE-ISUMM outperforms all baseline models and provide benchmark results on the TELSUM. With the help of existing summarization datasets, we examine the effectiveness of our model for other Indian languages in both monolingual and bilingual settings.

3.7 Ethical Statement

We reused the publicly available datasets (XL-Sum [6], BNLPC, NCTB [5], Marathi summarization dataset [7], Hindi-short summarization dataset [8]) to compare our state-of-art models.

Fair Compensation: We provided the data to *Elancer IT Solutions Private Limited* [9] company for getting the annotated summary. In order to perform the annotation process, *Elancer IT Solutions*

[6] https://github.com/csebuetnlp/xl-Sum

[7] https://github.com/pratikratadiya/marathi-news-document-dataset

[8] https://www.kaggle.com/datasets/disisbig/hindi-text-short-summarization-corpus

[9] http://elancerits.com/

Private Limited chose five native speakers of Telugu with excellent fluency. The company itself properly remunerates all the annotators.

Privacy Concerns: We have gone through the privacy policy of samyam website [10]. We do not foresee any harmful uses of using the data from the website.

[10]https://telugu.samayam.com/privacy-policy/privacypolicy/64302688.cms

Chapter 4

ISummCorp: an abstractive summarization dataset

With the exponential growth of resource and model creation towards summarization all around the globe, we realized the lack of proper datasets hinders the progress of summarization in Indian languages. We created a dataset for Indian languages using the Times of India(TOI) platform to address this issue. Additionally, we created vivid monolingual and multilingual summarization models specific to Indian languages.

4.1 Introduction

Text summarization is one of the most focused areas of the Natural Language Processing (NLP) community. It presents several hurdles regarding high-quality resources, models, and accurate text generation. The scientific community has shown great interest in summarization due to the expansion of digital data and deep-learning innovations over the past few years [66, 12]. High-quality, meticulously extracted, sizable, annotated datasets are required to benefit from these deep learning techniques fully. Recently, the NLP community has contributed large-scale datasets and seen significant growth in summarization. However, datasets on such a large scale must be appropriately retrieved and evaluated. If not, the models trained on these datasets will likely be at stake.

High-resource languages like English provide many datasets to train effective models. On the other hand, low-resource languages currently benefit from high-resource languages when trained in multilingual settings. It is crucial to explore the potential of these low-resource languages when trained alone and with an appropriate amount of language-specific data. Here, we try to develop datasets for a few low-resource languages in the Indian subcontinent.

Indian languages (Languages that are native to India) are spoken by about 10% of people all around the globe. Eight Indian languages secure a place among the top twenty spoken languages in the world, and around thirty Indian languages with more than a million speakers [26]. With the drastic increase in digital usage content by the Indian population, it is essential to build the necessary resources and tools for Indian languages. However, Indian languages cannot reap the benefits of emerging deep learning models due to a lack of standard and huge annotated summarization datasets. Additionally, Indian

languages have significant structural and morphological variations from high-resource languages like English, and techniques developed for English cannot be extended to Indian languages.

This work aims to contribute resources toward Indian languages by providing a new benchmark dataset ISummCorp. **ISummCorp** is a manually annotated abstractive summarization dataset sourced from the Times Of India(TOI). It is a professionally annotated dataset of about 376k article-summary pairs from eight Indian languages. It is one of the largest summarization datasets and the first publicly available dataset for a few languages(Malayalam, Marathi). We make ISummCorp publicly available [1]. The main aim of releasing this dataset is to aid in building efficient and more robust models in the field of Indian language summarization.

4.2 ISummCorp

To develop more standardized resources for Indian languages, we developed ISummCorp, a large-scale, multilingual summarization dataset for eight Indian languages. Our corpus provides 376k article-summary pairs from eight different Indian languages, namely Hindi(Hi), Tamil(Ta), Telugu(Te), Bengali(Bn), Gujarati(Gu), Marathi(Ma), Malayalam (Ml), and Kannada(Kn).

4.2.1 Dataset Collection

ISummCorp is extracted from the Times Of India (TOI) platform. TOI is one of India's most used and reliable news websites. The TOI publishes unbought stories from all over the country on its trustworthy website. It displays diversity in subjects, including science, technology, politics, current events, economics, finance, and health. For the news to reach every part of the country, TOI started publishing online news articles in native languages through various websites such as Samyam(Telugu)[2], Navabharat Times(Hindi)[3], VijayKarnataka(Kannada)[4], Samayam(Malayalam)[5], Samayam(Tamil)[6], Maharashtra Times(Marathi)[7] , Eisamay(Bengali)[8], iamgujarat(Gujarati)[9]. We developed our IndicSumm data for all languages under the TOI umbrella. Moreover, we extracted news articles from all the domains to include diversity in data.

[1] https://github.com/sireeshasummarization/samayam_data
[2] https://telugu.samayam.com/
[3] https://navbharattimes.indiatimes.com/
[4] https://vijaykarnataka.com/
[5] https://malayalam.samayam.com/
[6] https://tamil.samayam.com/
[7] https://maharashtratimes.com/
[8] https://eisamay.com/
[9] https://www.iamgujarat.com/

4.2.2 Dataset Creation

The process of creating summarization datasets is different across different datasets. Few datasets consider headlines as the summary of the article. Few other assumes the first sentence of the article to be the summary. CNN/Dailymail dataset [49] considers the highlights of the article, written by professionals, to be the summary. There are also datasets where multiple annotators write summaries to create a summarization dataset [69]. Since this is a very time-consuming and expensive process, this methodology does not fit for creating large-scale datasets. It may be argued that all of the most popular or large-scale benchmark datasets take into account the highlights, which are typically 3–4 sentences (or) 60–75 tokens long.

After carefully examining the extraction procedures, we discovered that the structure and content of the websites hosting TOI news articles are reliable. We have developed crawlers because TOI websites do not offer archiving or equivalent mechanisms. We provide language- and domain-specific crawler scripts to efficiently extract article-summary pairs. Here, we briefly describe the websites' structure and extraction process. The website's front page has various domains, each containing various sub-domains. The website's sub-domain sites feature several articles with headers that, when clicked, direct us to a page with the final content. Figure 4.3 gives us a glimpse of the article structure, and `shorturl.at/vBPS9` can be considered a reference for an example article page.

The content present in the article pages is written by professionals specializing in journalism and domain-specific fields. All the articles on TOI websites follow a similar structure which eases the extraction task. The headline of the article page is followed by a brief paragraph that succinctly summarises the information in the article, which is what we consider a summary. The input article is then presented after the brief paragraph. For a few of the articles, there also exist 3-4 bullet points that point out all the important information present in the article.

The summaries and the articles were written by professionals specialized in domain-specific fields. The sentence at the beginning is what we think of as the article's summary. The bullet points are organized as points; when combined for a summary, the summary loses coherence which is why we do not consider them as a summary. Additionally, not all articles have bullet points, and they convey identical information to the brief paragraph. So, the short opening paragraph serves as the article's summary. Figure 4.3 shows an example of the article.

4.2.3 Dataset Pre-processing

When extracted on such a large scale, it is typical for datasets to have noisy data or undesired text. So, we have employed a few techniques to eliminate the extraneous samples or any wanted text from the dataset. Before tokenizing each sentence, we split the article into sentences using Polyglot-tokenizer (https://pypi.org/project/polyglot-tokenizer). We eliminated any existing URLs, non-Unicode characters, or text written in a language other than the required one. In addition to text pre-processing, we

Features	Bengali (bn)	Gujarati (gu)	Marathi (mr)	Kannada (kn)	Malayalam (ml)	Hindi (hi)	Tamil (ta)	Telugu (te)
Total #samples	32093	25787	45988	82866	15198	123526	12297	38269
Avg length of document(sentences)	24.66	23.01	21.89	21.36	18.88	17.35	18.07	19.24
Avg #tokens in a document	397.0	389.75	317.38	278.30	254.45	384.71	289.32	251.48
Avg length of summary(sentences)	3.07	2.36	2.32	2.29	1.811	2.91	1.66	2.97
Avg #tokens in a summary	40.11	38.63	33.42	29.21	20.73	62.82	21.61	29.71
Total vocabulary (document)	7.7M	5.6M	9M	14.7M	2.75M	22M	2.2M	6.2M
Total vocabulary (summary)	1.1M	840k	1.3M	2.2M	291k	5.8M	244k	992k

Table 4.1: ISummCorp dataset statistics

developed other heuristics to produce a high-quality, uniform dataset. A few of them are mentioned below:

- We eliminate articles that are extremely brief and do not offer coherent or consistent information. They are eliminated by limiting the minimum number of tokens for an article.
- We filtered out the appropriate summary of article content by limiting the compression ratio to a specific bandwidth.
- It is a common practice to set a minimum length for summaries or text extracts to ensure that they are sufficiently informative and concise. Setting a minimum length of 20 tokens (words or word pieces) is a reasonable choice that can help ensure that the summary contains enough information.
- Any text regarding the images or image caption is excluded from the article content as it hinders the coherency of the article.

Any article that does not obey the above heuristics is eliminated from the dataset.

4.2.4 Uniqueness of ISummCorp

ISummCorp differs from the existing datasets in size, extraction, and scope of extension to other languages. These days, [49, 50, 69] serve as the primary source of extraction for summarization. Here, we will briefly discuss the various kinds of datasets and how ISummCorp stands out from the rest.

Wikipedia Source Extraction:

Many summarization datasets are derived from Wikipedia of different languages [29, 68]. Wikipedia is a huge crowd-sourcing platform for summarization tasks for multiple languages. The first paragraph of the article is considered the summary of the rest of the article. However, this is true only in the case of featured articles [10] of Wikipedia, but not all.

[10] https://en.wikipedia.org/wiki/Wikipedia:Featured_articles

Dataset	ISummCorp	XL-Sum
Compression	14.92	7.58
Abstractivity	20.79	41.8
Topic Similarity	33.99	23.8
Redundancy	3.05	NA

Table 4.2: Intrinsic Evaluation of ISummCorp and other datasets. All the values are reported in percentages. Redundancy cannot be calculated for the XL-Sum dataset because their summaries are of a single sentence.

Other News Websites:

Online News websites act as the other source of source for summarization datasets. Summaries are graphically extracted for several datasets from news website-based sources. Some often used techniques include extracting the headline, and the opening few phrases [80]. In addition to these, other datasets use a different summary extraction method dependent on the article's structure. These two datasets are Massivesum [70] and XL-Sum [18]. According to XL-Sum, the article summary is defined as a single statement that encompasses all of the article's context. Massivesum uses various resources, and the summary extraction differs for different articles. XL-Sum and Massivesum, however, received poor evaluations from [69] when it came to their summaries.

Our summary extraction procedure for the dataset ISummCorp consists of two stages: 1) filtering during extraction and 2) evaluation following extraction. Domain- and language-specific assessment scripts were built in the initial phase, so we could only retrieve the most pertinent information. Any article that deviates from the intended structure is discarded. We report all the statistics related to ISummCorp in table 4.1.

4.3 Quality Analysis of Dataset

When attempting to create a dataset on such a huge scale, its quality must be ensured. We follow a two-step evaluation to ensure that the summaries extracted are reliable and concise. The article-summary pairs are first automatically evaluated by a few factors, i.e., compression, Topic similarity, Abstractivity, Redundancy, and Semantic Coherence introduced by [3]. Using these metrics can provide a more comprehensive evaluation of a summarization dataset and allow for better comparisons between different datasets. However, it is important to note that these metrics may only capture some aspects of a summary and may not be sufficient to evaluate the quality of a summarization dataset fully. So,

Lang	Consistency	R&C	Fluency	Coherence
Bengali (bn)	0.98	4.3	4.7	4.7
Gujarati (gu)	0.98	4.6	4.8	4.6
Hindi (hi)	0.93	4.4	4.9	4.5
Kannada (kn)	0.94	4.5	4.9	4.6
Malayalam (ml)	0.97	4.7	4.7	4.7
Marathi (mr)	0.98	4.5	4.9	4.6
Tamil (ta)	0.97	4.4	4.8	4.5
Telugu (te)	0.99	4.7	4.7	4.8

Table 4.3: Manual evaluation of different languages from ISummCorp on average. Abstractivity is the percentage of novel 1-grams in the summary. The Consistency metric is rated out of 1, and the remaining human evaluation metrics are rated out of 5.

we also follow the manual evaluation of our datasets in terms of Consistency, Relevance and Coverage, Fluency, and Coherence.

4.3.1 Intrinsic Evaluation

To evaluate a summarization dataset, [3] proposed four metrics, i.e., Compression, Topic similarity, Abstractivity, and Redundancy. To make better comparisons, we prefer to assess our ISummCorp and other summarization datasets based on these parameters.

- Compression: Compression reveals how condensed a summary is in reference to the main article.
- Topic similarity: Using the Jensen-Shannon distance [35], topic similarity calculates the degree of similarity between the summary and the article. Topic similarity refers to how well the summary represents the main topics of the original text.
- Abstractivity: Including novel vocabulary in the summary while training a summarization model is crucial. We measure this novelty with the abstraction measure. Abstractivity measures the level of abstraction in summary, with a higher level of abstractivity indicating a greater degree of generalization and less specific detail.
- Redundancy: Redundancy measures the repetition of information in summary, with a lower level of redundancy indicating a more concise summary.

Table 4.2 gives an insight into the article-summary extraction of different datasets based on these features. We observe that the XL-Sum dataset cannot be included for redundancy metric as the summaries are just one sentence.

4.3.2 Human Evaluation

Summarization is a very abstractive task. Two people can never produce the same summary of the same article. So, a summary cannot be thoroughly evaluated with only intrinsic measures. Human evaluation is necessary because it enables a subjective assessment of the summaries by people who can take a variety of aspects into account. To ensure the quality of our dataset, we also manually examine it. We assigned the evaluation task to professional annotators, which we will discuss now in detail.

It is a difficult and almost impossible task to proofread every document. So, we take random samples of articles from each language for evaluation. The evaluation process is carried forward by asking the annotators to rate 200 randomly selected articles from each language on four factors: Consistency, Relevance and Coverage, and Fluency, Coherence. The consistency factor is evaluated on a binary basis, while the other factors are rated on a scale of 1-5. These factors were chosen because they can provide insight into the overall quality of the summaries. In addition to these factors, we also included coherence and relevance, which overlap with intrinsic evaluation measures. This allows us to cross-check our results and get a more comprehensive understanding of the strengths and weaknesses of our dataset. Overall, the manual evaluation procedure aids us in verifying the quality of our summaries and implementing any necessary adjustments.

- **Consistency(Yes/No):** Consistency refers to the degree to which the summary and the source text, i.e., the document, are similar.
- **Relevance and Coverage (1-5):** Relevance talks about the extent to which the summary represents the document's main content. And, Coverage is about if the summary can capture all the crucial aspects of the document.
- **Fluency (1-5):** Fluency is a subjective factor that refers to the overall smoothness and clarity of the language used in a text. It is crucial to consider a summary's grammatical and syntactic soundness as part of fluency. Most native language speakers should be able to understand the summary if it is written in an approachable manner. This entails ensuring that the language used is appropriate for the target audience and that any technical terminology present is clearly explained.
- **Coherence (1-5):** When we try to train an abstractive model, we must not input a summary where the sentences are disjoint. The paragraph must have some coherence or relevance for the reader. Using the coherence factor, we attempt to analyze this aspect of text generation.

4.4 Challenges faced

Here, we discuss the challenges we faced during dataset curation. Producing a dataset on such a huge scale is a difficult task. The data collected is not uniform for all domains. We had to write different

Article	ఇండియన్ రైల్వే తీపి కబురు అందించింది. రైలు ప్రయాణికులకు ఊరట కలిగే ప్రకటన చేసింది. ట్రైన్ ప్యాసింజర్ల కోసం కొత్త సేవలు అందుబాటులోకి తీసుకువస్తున్నట్లు వెల్లడించింది. దీని వల్ల ఇకపై రైల్వే ప్రయాణికులకు ట్రైన్ జర్నీ బోర్ కొట్టదు. ఆ సర్వీసులు ఏంటని ఆలోచిస్తున్నారా? ట్రైన్స్ లో రేడియో ఎంటర్ టైన్ మెంట్ సేవలు అందుబాటులోకి రాబోతున్నాయి. నార్త్ రైల్వే తాజాగా ట్రైన్స్ లో కస్టమైజ్డ్ మ్యూజిక్ ఎక్స్‌పీరియన్స్, ఆర్ జే ఎంటర్ టైమ్ అందించాలని భావిస్తోంది. అయితే ఈ సేవలు కొన్ని ట్రైన్స్ లో మాత్రమే అందుబాటులో ఉండనున్నాయి. ముందుగా వందే భారత్ ఎక్స్ ప్రెస్, శతాబ్ది ట్రైన్స్ లో ఈ ఎంటర్ టైన్ మెంట్ సేవలు అందుబాటులో ఉంచాలని రైల్వే భావిస్తోంది. ప్రయాణంతోపాటు పాటలు అనేది మంచి కాంబినేషన్ అని రైల్వే ఒక ప్రకటనలో తెలిపింది. ఇప్పటికే నార్త్ రైల్వే ప్రయాణికుల కోసం ఎంటర్ టైన్ మెంట్ సేవల కోసం కాంట్రాక్ట్ కూడా కుదుర్చుకుంది. ఢిల్లీ, లక్నో, భోపాల్, చండీఘర్, అజ్మీర్, డెహ్రాడూన్, కాన్పూర్, వారణాసి, కట్రా వంటి పలు ప్రాంతాలకు ప్రయాణించే ప్యాసింజర్లకు ఈ సేవలు అందుబాటులో ఉండొచ్చని తెలుస్తోంది. ఏడాదికి రూ.43 లక్షలకు పైగా ఆదాయం రావచ్చనే అంచనాలు ఉన్నాయి. అందువల్ల రైల్వే ఈ ఎంటర్టైన్మెంట్ సర్వీసులను మరిన్ని ట్రైన్స్ లో కూడా అందుబాటులోకి తీసుకురావచ్చనే అంచనాలు నెలకున్నాయి. Indian Railways has announced some good news. The announcement was made to bring relief to the train passengers. It has been revealed that new services are being made available for train passengers. Due to this, the train journey will no longer be boring for railway passengers. Wondering what those services are? Radio entertainment services are going to be available in trains. Northern Railways intends to offer a customized music experience in trains, RJ Entertainment. But these services will be available only in some trains. Railways hopes to make these entertainment services available in Vande Bharat Express and Shatabdi trains first. The railways said in a statement that songs are a good combination of travel and music. Northern Railways has already entered into a contract for entertainment services for passengers. It seems that these services may be available for passengers traveling to many places like Delhi, Lucknow, Bhopal, Chandigarh, Ajmer, Dehradun, Kanpur, Varanasi, Katra. It is expected that the income will be more than Rs.43 lakhs per year. Therefore, there are expectations that the railways may make these entertainment services available in more trains.
Actual summary	ట్రైన్ ప్యాసింజర్లకు కొత్త సేవలు అందుబాటులోకి రానున్నాయి. రైళ్లలో ఇకపై మ్యూజిక్ సేవలు లభించనున్నాయి. దీని వల్ల ప్రయాణికులకు జర్నీ బోర్ కొట్టక పోవచ్చు. పాటలు వింటూ ప్రయాణం సాగించవచ్చు. New services will be available for train passengers. Music services will be available in trains from now on. Due to this, the journey may not be boring for the passengers. You can travel while listening to songs.

Figure 4.1: An Example of Article-summary pair from ISummCorp

scripts for each language and domain to remove the noise present at a minute level. These are the challenges we faced at a higher level. We list a few other challenges that we faced at an article level.

- **Different language text:** There are a few articles that contain English text in between, and to extract the monolingual datasets, the whole text has gone through a check to remove any unwanted English text present.

- **Unwanted URL data:** Since we extracted our data from an online website. Several ads are present as part of clickbait, given an article page. And, for a few articles, some links refer to ads/other articles. The URLs are also followed by text in the same language, making this a more challenging job.

- **Unrecognized characters:** When the whole data is crawled from the website, there are several non-UTF-8 characters present. We manually verified the decoded version of all these characters and replaced the text accordingly.

- **Evaluation of extracted datasets:** As summarization is a subjective task, it does not have a ground truth or a reference summary for comparison. Since our dataset contains articles from different domains, we needed people familiar with all the domains and able to judge the summary. Moreover, it is also a time-taking and expensive procedure.

Article	भोजपुरी इंडस्ट्री की नंबर वन एक्ट्रेसेस में से एक अक्षरा सिंह हर दिन सफलता की तरफ एक कदम बढ़ा रही हैं। उनकी जबरदस्त फैन फॉलोइंग है और उनकी फिल्मों को देखना लोग पसंद करते हैं। यही वजह है कि उन्होंने अपनी फीस बढ़ा ली है। बताया जा रहा है कि वो रवि किशन के बाद अब अक्षरा भी अपकमिंग प्रोजेक्ट्स के लिए पहले से ज्यादा चार्ज करेंगी। उन्होंने डार्लिंग मूवी के लिए दोगुना पैसा लिया है। इस तरह वो भोजपुरी इंडस्ट्री की सबसे महंगी हिरोइन बन गई हैं। अक्षरा को महंगी कीमत पर साइन करने को लेकर फिल्म "डार्लिंग" के निर्माता प्रदीप के शर्मा ने कहा कि अक्षरा सिंह डिजर्व करती हैं। वे सिलेक्टेड काम करती हैं और वे हमारी स्क्रिप्ट की डिमांड है। ये तो अच्छी बात है। इसमें कोई बड़ी बात नहीं है। उन्होंने कहा कि 'डार्लिंग' की स्क्रिप्ट में फीमेल लीड सिंगर हैं, जिसको वो पूरी तरह सूट करती हैं। अक्षरा के साथ इस फिल्म में निर्माता प्रदीप के शर्मा के बेटे राहुल शर्मा लांच हो रहे हैं।'अक्षरा से बेहतर कोई नहीं हो सकता' वहीं, इसको लेकर डायरेक्टर रजनीश मिश्रा ने कहा कि इस कहानी में अक्षरा से बेहतर कोई नहीं हो सकता है। इसलिए हमने उन्हें फिल्म की कहानी के अनुसार उनको महंगे दाम पर साइन किया। अक्षरा ने भी माना कि वे अब सेलेक्टिव काम करना चाहती हैं। इसलिए उन्होंने फीस बढ़ाई है। भले ही उन्होंने अपनी फीस बढ़ा दी है, लेकिन लोग उन्हें बढ़ी हुई फीस भी देने के लिए तैयार हैं। उन्होंने कहा, 'ये भोजपुरी इंडस्ट्री के लिए भी अच्छी बात है कि हिरोइन भी अपनी प्रतिभा और काम के हिसाब से फिल्मों के चार्ज करने के लिए प्रेरित हों। इसमें बुरा कुछ भी नहीं है। हर कोई जिंदगी में आगे बढ़ाना चाहता है। मैंने कई प्रोजेक्ट्स फ्रेंडली नोटिस पर किये हैं और आज भी कर रही हूं। वो एक अलग बात है, लेकिन अब मैं और भी अच्छे काम करना चाहती हूं, इसलिए पैसे तो खर्च करने होंगे। Akshara Singh, one of the top actresses of Bhojpuri industry, is taking one step towards success every day. He has a huge fan following and people love to watch her movies. This is the reason why they have increased her remuneration. It is being told that after Ravi Kishan, Akshara will also charge more than before for her upcoming projects. She has taken double the money for Darling movie. In this way she has become the most expensive heroine of Bhojpuri industry. Regarding signing Akshara at an expensive price, Pradeep K Sharma, producer of the film "Darling", said that Akshara Singh deserves. She does selected work and she is the demand of our script. This is a good thing. There is no big deal in this. She said that the script of 'Darling' has a female lead singer, which she suits perfectly. Along with Akshara, the film marks the launch of Rahul Sharma, son of producer Pradeep K Sharma. On the other hand, director Rajneesh Mishra said that there can be no one better than Akshara in this story. So we signed her at an exorbitant price as per the story of the film. Akshara also agreed that she now wants to do selective work. That's why they have increased the fees. Even though he has increased his fees, but people are ready to pay him the increased fees as well. He said, 'It is also a good thing for Bhojpuri industry that heroines are also motivated to charge films according to their talent and work. There is nothing bad in this. Everyone wants to move ahead in life. I have done many projects on friendly notice and am still doing it. That is a different thing, but now I want to do more good work, so money has to be spent.
Actual summary	अक्षरा सिंह भोजपुरी इंडस्ट्री की फेमस ऐक्ट्रेसेस में से एक हैं। उनसे जुड़ी एक बड़ी खबर सामने आ रही है कि उन्होंने अपनी फीस बढ़ा दी है। अब वो एक फिल्म के लिए पहले से दोगुना चार्ज वसूल करेंगी। Akshara Singh is one of the famous actresses of Bhojpuri industry. A big news related to her came out that she has increased her remuneration. And, now she will charge twice as much as before for a film.

Figure 4.2: Another Example of Article-summary pair from ISummCorp

4.5 Analysis of Summaries

Here, we display a few examples of ISummCorp from different languages. Figures 4.1 and 4.2 display the article-summary pairs from different languages. The example article-summary pair 4.1 tries to cover the central idea of the whole article in its summary in a very concise format. If you observe clearly, the summary does not dive deep into the details of the article, such as "which all trains have already started the service, what is the expected income from these services". The gold summary only displays the essential information from the whole article.

Another example article-summary pair is displayed in fig 4.2. The article is about an actress who has increased her remuneration. The summary begins by attempting to provide background information, such as the actress's name, level of fame, and industry. The summary then jumps right into the main motive of the article, which is about her compensation. Contextual information is required in articles like these because the article is domain-specific and geographically also specific. The need for pre-contextual information is required because we cannot assume that everyone knows the actress.

The observations we made from the summaries of ISummCorp are listed below:

- The summary tries to minimize the content present into a concise format covering all the essential aspects of the article.

- For an article, if there is any pre-contextual information required, the summary provides it with minimal and sufficient details.
- The summary covers all the critical aspects of the article and mentions them in a minimal way.
- , In short, the summaries can be regarded as short summaries of the article.

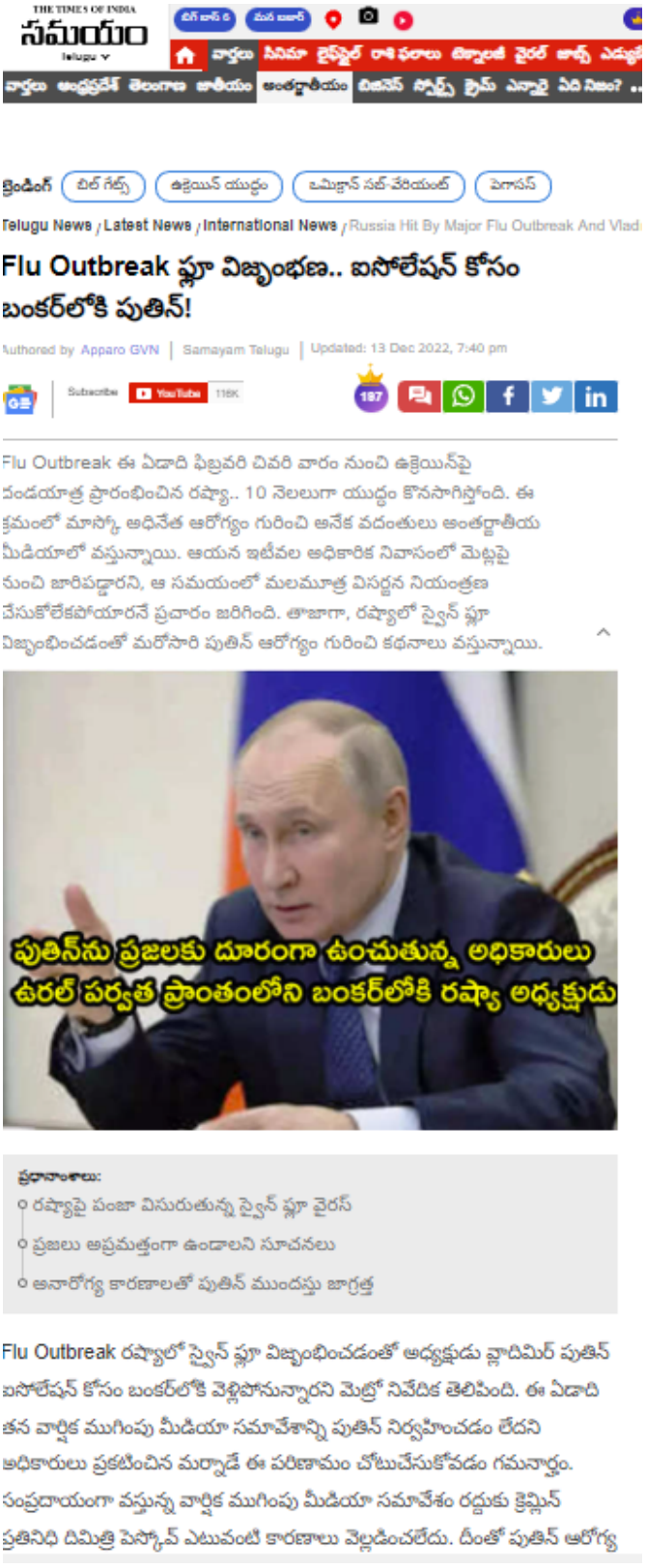

Figure 4.3: TOI website article page in Telugu (Samyam).

Chapter 5

IndicSumm: language models in Indian context

Finetuning transformer-based and sequence-to-sequence-based models have achieved state-of-the-art results on many abstractive summarization datasets. Recently, a wide variety of pre-trained models for multiple languages were contributed to the research community. This work aims to contribute resources towards Indian languages by providing new benchmark models.

Here, we introduce **IndicSumm:** a set of monolingual and multilingual models trained on ISummCorp. ISummCorp is a huge multilingual dataset spread across eight Indian languages. With the help of ISummCorp, we build various summarization models based on mT5 [77] specific to Indian languages. We are more inclined towards creating monolingual models that are known to be effective given the morphological richness of the Indian languages. We later train a multilingual model on all eight languages to create a generic summarization model specific to Indian languages. In summary, we make the following contributions through this work:

- We are the first to present monolingual summarization models that have been trained in Indian languages.
- We develop eight monolingual summarization models and a multilingual Indic summarization model and publicly release these models.
- We prove that a language, when finetuned with an adequate amount of data in a monolingual setting, outperforms the multilingual fine-tuning strategy.
- We achieve state-of-art results with IndicSumm over the existing Indic multilingual models like XL-Sum and IndicBART.

5.1 IndicSumm

IndicSumm is a set of models based on the T5 [59] family, which is mainly a transformer model. T5 introduces a unified framework that can handle any problem statement in a text-to-text format. Further,

[77] introduced a multilingual variant of T5 trained on the Common Crawl corpus [1] in 101 languages. With the help of transfer learning, we finetune the mT5 model using ISummCorp in monolingual and multilingual settings. We chose mT5 over other pre-trained models because of its massive coverage over our eight Indian languages and its unique training objective for all downstream tasks.

5.2 Training Details

Here, we discuss in detail the training details and finetuning process of IndicSumm in monolingual and multilingual settings.

5.2.1 Monolingual IndicSumm

In this study, we finetuned the mT5 model for each language. The mT5 model consisted of 8 blocks, with two layers of encoder and decoder settings in each block. The hidden and filter sizes were set to 512 and 1024, respectively, with six attention heads. We applied a dropout rate of 0.1 and used the AdamW [39] optimizer with a maximum learning rate of 3E-4. The batch size was set to 2048 tokens. The training was performed on 4 NVIDIA GeForce GTX 1080Ti GPUs and took approximately five days for 12 epochs. The tokenizer used was Google's pre-trained mT5 tokenizer, which covers 101 languages and has a vocabulary of 250k words. Due to computational limitations, we had to reduce the input and output to 512 and 50 tokens, respectively. We reserved 20% of the data for testing and 10% for validation and used the remaining 70% to train the model. We followed a similar training strategy(in terms of hyperparameters) as of [30].

5.2.2 Multilingual IndicSumm

Moving to the multilingual setting, the Multilingual model is created by inputting the article-summary pairs from all eight existing languages. This mode of training has gained immense importance in recent times. [8]. This model is created for the multilingual setting by giving all the article-summary pairs from eight languages as input. The input batch size, learning rate, and optimizer remained the same as in monolingual models. The multilingual model setting is the same as the monolingual setting. However, for each batch, the input article-summary pairs are from multiple languages, unlike in a monolingual setting.

The multilingual model took five days to train 12 epochs. The tokenizer used was bert-base-multilingual-cased, pretrained on 104 languages and has a vocabulary size of 110,000. We used four NVIDIA GeForce GTX 1080Ti graphics cards. The base model has seven blocks and two layers with 120M parameters, which constrained our training inputs and outputs to 512 and 50 tokens, respectively.

[1] https://commoncrawl.org/

Lang Setting	Monolingual			Multilingual		
Language	R-1	R-2	R-L	R-1	R-2	R-L
Bengali(bn)	39.38	28.1	37.66	32.73	19.32	31.55
Gujarati(gu)	33.22	22.97	31.67	30.5	22.56	30.49
Hindi(hi)	65.14	55.77	62.63	57.84	25.11	57.69
Kannada(kn)	59.39	52.13	58.3	53.91	45.52	53.88
Malyalam(ml)	26.18	15.33	25.04	21.18	18.80	21.03
Marathi(mr)	59.16	52.04	58.36	51.67	21.67	50.63
Tamil(ta)	44.89	27.69	41.57	37.53	23.86	37.32
Telugu(te)	36.95	22.46	35.88	32.95	19.68	32.93

Table 5.1: Comparison of IndicSumm Monolingual and IndicSumm Multilingual models.

5.3 Results and Analysis

Here, we conduct an empirical evaluation of different summarization techniques and IndicSumm when tested on ISummCorp. We first try to analyze the performance of IndicSumm monolingual and multilingual models. Later, we compare the performance of IndicSumm with standard baselines and then with recently released multilingual summarization models.

5.3.1 Analysis of IndicSumm

Our main objective behind finetuning monolingual models is to explore the potential of low-resource languages. Table 5.1 reports the F1-score of ROUGE [32] metric for the monolingual and multilingual models finetuned. As observed, all the monolingual models outperform the multilingual model for all languages. Our main observation is that low-resource languages can outperform the multilingual strategy when trained separately with enough resources. However, the multilingual and monolingual model scores are competitive for the languages Gujarati, Marathi, and Tamil. This can be accounted for relatively fewer training samples of the languages. Hindi and Kannada languages have showcased higher ROUGE scores of all the languages, which might account for their large number of training samples and relatively less abstractive summaries. Figure 5.1 demonstrates a sample example from ISummCorp and the predicted summary.

5.3.2 Baselines

Here, we compare IndicSumm monolingual setting with a few standard baseline techniques. We considered three baseline systems which we will discuss in brief.

Metric	ROUGE-1			
Languages# / Methodology!	Random	LEAD-3	LexRank	IndicSumm
Bengali (bn)	14.96	27.99	20.37	**39.38**
Gujarati (gu)	14.70	24.27	22.49	**33.22**
Hindi (hi)	22.58	42.05	34.79	**65.14**
Kannada (kn)	16.72	50.08	32.61	**59.39**
Malayalam (ml)	15.14	20.74	22.65	**26.18**
Marathi (mr)	14.98	32.26	30.01	**59.16**
Tamil (ta)	16.19	29.73	23.64	**44.89**
Telugu (te)	15.61	31.57	25.00	**36.95**

Table 5.2: Comparison of IndicSumm with few baselines with Rouge-1 metric

Metric	ROUGE-2			
Languages# / Methodology!	Random	LEAD-3	LexRank	IndicSumm
Bengali (bn)	5.08	16.3	10.66	**28.1**
Gujarati (gu)	4.84	12.62	9.75	**22.97**
Hindi (hi)	9.07	29.13	20.38	**55.77**
Kannada (kn)	6.70	42.20	21.68	**52.13**
Malayalam (ml)	4.71	7.74	11.55	**15.33**
Marathi (mr)	6.03	22.42	19.20	**52.04**
Tamil (ta)	3.79	14.24	9.59	**27.69**
Telugu (te)	3.23	17.52	10.81	**22.46**

Table 5.3: Comparison of IndicSumm with few baselines with Rouge-2 metric

- **Random Baseline:** Here, any k random sentences are selected from the article to form the summary. The summary will be extractive, and k depends upon the expected output summary size.
- **LEAD-K:** LEAD-K summarization is a technique where the first K sentences of the article concatenate to form a summary. It can also be a set of first K keywords to form a summary. However, we consider the first K sentences to be the summary for our baseline.
- **LexRank:** LexRank [11] is a graph-based approach that ranks sentences based on the weights of a TF-IDF graph generated from the input. This is also extractive in nature, where it ranks the sentences based on centrality score and forms the summary.

Table 5.2 shows the ROUGE scores for the baselines compared with IndicSumm. We report different ROUGE [32] metrics (ROUGE-1, ROUGE-2, ROUGE-L) on all the experiments. Table 5.2 reports the

Metric	ROUGE-L			
Languages# / Methodology!	Random	LEAD-3	LexRank	IndicSumm
Bengali (bn)	13.72	26.03	16.09	**37.66**
Gujarati (gu)	13.33	22.32	16.62	**31.67**
Hindi (hi)	19.10	34.09	25.73	**62.63**
Kannada (kn)	15.85	49.49	27.18	**58.3**
Malayalam (ml)	14.26	19.53	18.75	**25.04**
Marathi (mr)	14.28	31.17	24.48	**58.36**
Tamil (ta)	14.95	27.57	18.42	**41.57**
Telugu (te)	14.84	31.43	20.06	**35.88**

Table 5.4: Comparison of IndicSumm with few baselines with Rouge-L metric

Metric	R1				
Languages# / Methodology!	XL-Sum mT5	IndicBART-IndicSS	IndicBART-XLSum	IndicSumm(Multilingual)	IndicSumm(Monolingual)
Bengali (bn)	20.80	0.24	19.7	32.73	39.38
Gujarati (gu)	20.24	0.57	16.88	30.5	33.22
Hindi (hi)	33.57	24.15	30.85	57.84	65.14
Kannada (kn)	4.8	1.84	33.80	53.91	59.39
Malayalam (ml)	4.70	0.94	14.67	21.18	26.18
Marathi (mr)	32.91	16.17	27.15	51.67	59.16
Tamil (ta)	25.11	0.95	22.29	37.53	44.89
Telugu (te)	26.02	2.52	19.57	32.95	36.95

Table 5.5: Comparison of IndicSumm models with the existing multilingual models with Rouge-1 metric

ROUGE-1 metrics of comparison of IndicSumm with other baselines. similarly, table 5.3 and table 5.4 reports the ROUGE-2 and ROUGE-L metrics of the same. According to the table 5.2, IndicSumm outperforms all the established baselines in terms of all the ROUGE metrics. Of all the baselines, LEAD-K tends to outperform the Random and LexRank models. Random baseline performed similar ROUGE scores for all the datasets irrespective of their abstractiveness.

5.3.3 Comparison with existing multilingual models

We compare the performance of IndicSumm monolingual models with the existing finetuned multilingual models. We compare IndicSumm with the extensively trained multilingual transformer models such as the XL-Sum mT5 model [18] and IndicBART variants [8].

- **XL-Sum mT5**: [18] have released a multilingual variant of T5 model finetuned on XL-sum dataset. XL-Sum dataset supports 44 languages trained over 1 million article-summary pairs.

Metric	R2				
Languages# / Methodology↓	XL-Sum mT5	IndicBART-IndicSS	IndicBART-XLSum	IndicSumm(Multilingual)	IndicSumm(Monolingual)
Bengali (bn)	9.73	0.02	10.35	19.32	28.1
Gujarati (gu)	9.01	0.04	7.19	22.56	22.97
Hindi (hi)	17.55	9.96	17.71	25.11	55.77
Kannada (kn)	0.06	0.01	24.35	45.52	52.13
Malayalam (ml)	0.04	0.007	6.04	18.80	15.33
Marathi (mr)	21.77	7.06	18.53	21.67	52.04
Tamil (ta)	11.95	0.06	10.01	23.86	27.69
Telugu (te)	10.27	0.02	9.22	19.68	22.46

Table 5.6: Comparison of IndicSumm models with the existing multilingual models with Rouge-2 metric

Metric	RL				
Languages# / Methodology↓	XL-Sum mT5	IndicBART-IndicSS	IndicBART-XLSum	IndicSumm(Multilingual)	IndicSumm(Monolingual)
Bengali (bn)	20.00	0.22	18.34	31.55	37.66
Gujarati (gu)	18.87	0.46	15.56	30.49	31.67
Hindi (hi)	27.89	20.31	27.43	57.69	62.63
Kannada (kn)	4.87	1.63	32.30	53.88	58.3
Malayalam (ml)	4.70	0.7	13.6	21.03	25.04
Marathi (mr)	31.50	15.37	26.06	50.63	58.36
Tamil (ta)	27.09	0.67	20.54	37.32	41.57
Telugu (te)	24.90	1.91	19.31	32.93	35.88

Table 5.7: Comparison of IndicSumm models with the existing multilingual models with Rouge-L metric

Here, we compare the XL-Sum multilingual trained mT5 model on our dataset. This model is trained on all the 44 languages present in the dataset. However, XL-Sum does not support languages like Marathi and Malayalam.

- **IndicBART variants**: IndicBART [8] is a multilingual pre-trained sequence-to-sequence model trained specifically on Indian languages based on mBART [31] architecture. IndicBART evaluates the summarization task by finetuning IndicSentenceSummarization and XL-Sum datasets. It also has a sister model named IndicBARTss, similar to IndicBART but trained based on single script representation. Moreover, these models are evaluated on Machine Translation and Summarization tasks. For the summarization task, [8] have finetuned IndicBART on IndicSentenceSummarization and XL-Sum datasets. IndicSentenceSummarization dataset is a dataset extracted from different news websites in Indian languages where the headline is considered as the summary. We

Article	https://telugu.samayam.com/sports/cricket/news/india-vs-sri-lanka-virat-kohli-becomes-6th-indian-batter-to-score-8000-test-runs/articleshow/89989203.cms
Actual summary	టెస్టుల్లో ఇప్పటి వరకూ ఐదుగురు భారత క్రికెటర్లు మాత్రమే 8వేల పరుగుల మార్క్ కి చేరుకోగా.. తాజాగా విరాట్ కోహ్లీ ఈ రికార్డ్ లో చోటు సాధించాడు. So far only five Indian cricketers have reached the 8000 run mark in Tests. And Virat Kohli has recently achieved this record.
Predicted summary	విరాట్ కోహ్లీ టెస్టుల్లో మరో అరుదైన మైలురాయిని అందుకున్నాడు. కెరీర్ లో 100వ టెస్టులో 8వేల పరుగుల మార్క్ ని అందుకున్న రెండో క్రికెటర్ గా విరాట్ కోహ్లీ నిలిచాడు. Virat Kohli has achieved another rare milestone in Tests. He became the second cricketer to reach the 8000-run mark in the 100th Test of his career.

Figure 5.1: A Sample article-summary pair and Predicted summary from ISummCorp.

compare our IndicSumm models with IndicBART-XLSum [2] model and MultiIndiCSentenceSummarization [3].

Tables 5.5 showcases the performance of different finetuned models on ISummCorp. We report different ROUGE [32] metrics (ROUGE-1, ROUGE-2, ROUGE-L) on all the experiments. Table 5.5 reports the ROUGE-1 metrics of comparison of IndicSumm with other multilingual models. similarly, table 5.6 and table 5.7 reports the ROUGE-2 and ROUGE-L metrics of the same. From the table 5.5, we observe that IndicSumm outperformed the existing multilingual models in all aspects. We observe that the IndicBART-XLSum performed better than the XL-Sum mT5 for most languages except Malayalam and Kannada. We also observe that both LexRank and XL-Sum mT5 performed similarly. However, the inferior performance of XL-Sum mT5 on Kannada and Malayalam datasets is due to the absence of these language datasets in the XL-Sum dataset. Also, IndicSumm multilingual model has outperformed all the existing baseline and multilingual models. We should be able to deduce how critical it is to have a suitable summarization dataset from the outcomes of IndicBART-IndicSentenceSummarization. Otherwise, no matter how many hyperparameters are pre-trained into a model, the outcomes from finetuning are always substandard.

5.4 Analysis of Summaries

Figures 5.1 and 5.2 shows us a few examples of the articles with the gold summary and predicted summary. First, talking about 5.1, the actual and predicted summary seems similar in terms of length

[2] https://huggingface.co/ai4bharat/IndicBART-XLSum
[3] https://huggingface.co/ai4bharat/MultiIndicSentenceSummarization

Article	ఇండియన్ రైల్వే తీపి కబురు అందించింది. రైలు ప్రయాణికులకు ఊరట కలిగే ప్రకటన చేసింది. ట్రైన్ ప్యాసింజర్ల కోసం కొత్త సేవలు అందుబాటులోకి తీసుకువస్తున్నట్లు వెల్లడించింది. దీని వల్ల ఇకపై రైల్వే ప్రయాణికులకు ట్రైన్ జర్నీ బోర్ కొట్టదు. ఆ సర్వీసులు ఏంటని ఆలోచిస్తున్నారా? ట్రైన్స్ లో రేడియో ఎంటర్ టైన్ మెంట్ సేవలు అందుబాటులోకి రాబోతున్నాయి. నార్త్ రైల్వే తాజాగా ట్రైన్స్ లో కస్టమైజ్డ్ మ్యూజిక్ ఎక్స్‌పీరియన్స్, ఆర్ జే ఎంటర్ టైమ్ అందించాలని భావిస్తోంది. అయితే ఈ సేవలు కొన్ని ట్రైన్స్ లో మాత్రమే అందుబాటులో ఉండనున్నాయి. ముందుగా వందే భారత్ ఎక్స్ ప్రెస్, శతాబ్ది ట్రైన్స్ లో ఈ ఎంటర్ టైన్ మెంట్ సేవలు అందుబాటులో ఉంచాలని రైల్వే భావిస్తోంది. ప్రయాణంతోపాటు పాటలు అనేది మంచి కాంబినేషన్ అని రైల్వే ఒక ప్రకటనలో తెలిపింది. ఇప్పటికే నార్త్ రైల్వే ప్రయాణికుల కోసం ఎంటర్ టైన్ మెంట్ సేవల కోసం కాంట్రాక్ట్ కూడా కుదుర్చుకుంది. ఢిల్లీ, లక్నో, భోపాల్, చండీఘర్, అజ్మీర్, డెహ్రాడూన్, కాన్పూర్, వారణాసి, కట్రా వంటి పలు ప్రాంతాలకు ప్రయాణించే ప్యాసింజర్లకు ఈ సేవలు అందుబాటులో ఉండొచ్చని తెలుస్తోంది. ఏడాదికి రూ.43 లక్షలకు పైగా ఆదాయం రావచ్చనే అంచనాలు ఉన్నాయి. అందువల్ల రైల్వే ఈ ఎంటర్టైన్మెంట్ సర్వీసులను మరిన్ని ట్రైన్స్ లో కూడా అందుబాటులోకి తీసుకురావచ్చనే అంచనాలు నెలకున్నాయి. Indian Railways has announced some good news. The announcement was made to bring relief to the train passengers. It has been revealed that new services are being made available for train passengers. Due to this, the train journey will no longer be boring for railway passengers. Wondering what those services are? Radio entertainment services are going to be available in trains. Northern Railways intends to offer a customized music experience in trains, RJ Entertainment. But these services will be available only in some trains. Railways hopes to make these entertainment services available in Vande Bharat Express and Shatabdi trains first. The railways said in a statement that songs are a good combination of travel and music. Northern Railways has already entered into a contract for entertainment services for passengers. It seems that these services may be available for passengers traveling to many places like Delhi, Lucknow, Bhopal, Chandigarh, Ajmer, Dehradun, Kanpur, Varanasi, Katra. It is expected that the income will be more than Rs.43 lakhs per year. Therefore, there are expectations that the railways may make these entertainment services available in more trains.
Actual summary	ట్రైన్ ప్యాసింజర్లకు కొత్త సేవలు అందుబాటులోకి రానున్నాయి. రైళ్లలో ఇకపై మ్యూజిక్ సేవలు లభించనున్నాయి. దీని వల్ల ప్రయాణికులకు జర్నీ బోర్ కొట్టక పోవచ్చు. పాటలు వింటూ ప్రయాణం సాగించవచ్చు. New services will be available for train passengers. Music services will be available in trains from now on. Due to this, the journey may not be boring for the passengers. You can travel while listening to songs.
IndicSumm predicted summary	రైలు ప్రయాణికులకు తీపి కబురు అందించింది. ట్రైన్ ప్యాసింజర్ల కోసం కస్టమైజ్డ్ మ్యూజిక్ ఎక్స్‌పీరియన్స్, ఆర్ జే ఎంటర్ టైన్ మెంట్ లాంటి కొత్త సేవలు అందుబాటులోకి రాబోతున్నాయి. దీని వల్ల ఇకపై రైల్వే ప్రయాణికులు బోర్ కొట్టక పోవచ్చు . The train passengers received some good news. New services like customized music experience, RJ entertainment are going to be available for train passengers. Due to this, railway passengers may not get bored anymore.

Figure 5.2: A Sample article-summary pair and Predicted summary

and number of sentences. Though the actual and predicted summary is not the same, both try to convey the same information "Virat Kohli has recently reached the 8000 run milestone in his test career", which is the central idea of the whole article. The predicted summary tries to extract and paraphrase similar information as the actual summary from the input article.

Figure 5.2 shows another example of Indicsumm predicted summary. Regarding this article, the number of sentences in the predicted and gold summary is quite different. The gold summary is four sentences, whereas the predicted summary is only three. However, both summaries convey the same information, the article's primary motive.

5.5 Conclusion

Indian languages are resource-poor, relative to English, in terms of available datasets, feature representations, and machine learning models. We tried to bridge the gap by creating Indian language-specific datasets and models. We present IndicSumm, a set of Indian language-based summarization models created to promote the development of the Natural Language Generation for Indian languages. We are the

first to develop monolingual summarization models for Indian languages, to enhance the performance of summarization tasks. We also explore the potential of low-resource monolingual models by training them with enough data. In conclusion, we hope that the IndicSumm helps Indian languages to step ahead in summarization.

Chapter 6

Conclusion and Future Work

The main aim of this thesis is to contribute resources towards the summarization of low-resource Indian languages. We take the first step towards it by proposing an unsupervised summarization technique GAE-ISUMM. With almost negligible resources for low-resource Indian languages, we intended to shed some light on sophisticated deep-learning techniques for Indian languages. GAE-ISUMM addresses the robust representational capabilities of graph-based techniques, RNNs, and neural networks. Using cluster representations from GRU and GAE, we create a model that learns text representations and a document summary. Additionally, to evaluate our approach, we have created TELSUM, a Summarization dataset in Telugu. With the help of existing summarization datasets, we examine the effectiveness of our model for other Indian languages in both monolingual and bilingual settings.

Further, the Indian languages needed standard summarization datasets to explore emerging transformer-based models or transfer learning techniques. So, the next step in our work focused on creating standard datasets for Indian languages. We created ISummCorp, a multilingual summarization dataset for eight Indian languages comprising 376k article-summary pairs. We explain in detail how our dataset ISummCorp is unique and can stand as a benchmark dataset for Indian languages. We also introduce IndicSumm, a set of monolingual and multilingual models finetuned on ISummCorp with mT5 as the base model. We argue that a language can perform better when finetuned in a monolingual setting with an adequate amount of data than in a multilingual setting. We proved this with the help of ISummCorp. We also explore the potential of our IndicSumm models by experimenting with other multilingual models present and achieve state-of-art results in every aspect.

6.1 Future Work

The thesis work mainly focused on the summarization of Indian languages. With the increased digital data in Indian languages on various platforms, resources for summarization for these resource-poor languages must be improved. In the process of creating resources for the summarization of Indian languages, we realized that there is a huge scope for our work in the coming years. the unsupervised technique we proposed at the start can be a starting point for sophisticated summarization models for

Indian languages ahead. We have created different types of datasets, which will take a position as a standard dataset for Indian languages. Apart from it, the models we have created can be used can be vividly used to extract summaries. The scope of our work on the summarization of Indian languages is the following:

- **Extending to other low-resource languages:** GAE-ISUMM being unsupervised and not requiring labeled data, it can be extended to other resource-poor languages and obtain high-quality summaries.
- **Cross-lingual Dataset:** ISummCorp has eight different Indian languages, which cover different domains such as sports, national, international, business, and entertainment. These few domains cover the same or similar news all around the country. This helps us build parallel articles from ISummCorp by analyzing the article's composition. This allows us to build a cross-lingual dataset of 56 combinations which further helps build cross-lingual learning.
- **Multi-document Summarization:** Extending the idea of Crosslingual datasets, if we can draw parallel corpora from ISummCorp, we can further create a multi-document summarization dataset with the help of translation.
- **Domain Specific Analysis:** ISummCorp contains domain-specific data which can be further used to create domain-specific models. A mainstream domain identification can be created with the data, or domain-specific words can be extracted from such a huge dataset. Further, an analysis can be done based on the objectivity of each domain.
- **Code Mixed Data:** ISummCorp can also act as a huge source to obtain code mixed data for different Indian languages. The news articles are written in an understandable format for the locals, and a few English transliterated words can be further used to create code-mixed data.
- **Analysis of Different Indian language families:** IsummCorp consists of 8 different languages(Hindi, Marathi, Bengali, Gujarati, Telugu, Tamil, Malayalam, Kannada). Hindi, Marathi, Bengali, and Gujarati belong to the Indo-Aryan language family, and Tamil, Telugu, Malayalam, and Kannada are from the Dravidian family. Experiments can explore the cross-learning transfer within a language family and different language families with abundant data available from both language families.
- **Extending the number of languages:** India is a multilingual country with more than 30 languages being spoken by million plus people. Digital data is also growing exponentially in other Indian languages. Hence, creating resources for other Indian languages would help analyze and design better models.
- **Different multilingual and monolingual models:** Recent studies have introduced different multilingual models trained on around 100 different languages. But the Indian language-specific tasks

might benefit better when trained alone with adequate data. The other multilingual models can be pre-trained o finetuned on Indian datasets for better performance.

- **Online Tools for automated summarization:** There are a lot of summarization tools available for high-resource languages like English. However, when it comes to resource-poor languages, there are no such summarization tools. The models we have created can be used to develop an automatic summarization user interface for resource-poor Indian languages.

Publications

1. Vakada Lakshmi Sireesha, Anudeep Ch, Subbareddy Oota, Mounika Marreddy, Radhika Mamidi: GAE-ISUMM: Unsupervised Graph-based Summarization for Indian Languages

   International Joint Conference on Neural Networks (IJCNN), 2023

2. Vakada Lakshmi Sireesha, Anudeep Ch, Mounika Marreddy, Radhika Mamidi: IndicSumm: Summarization Resource Creation for Eight Indian Languages

   10th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics

3. Vakada Lakshmi Sireesha, Charan Chinni, Mounika Marreddy, Oota Subba Reddy, Radhika Mamidi: Unsupervised Graph based Telugu News Articles Text Summarization

   NeurIPS, 15th Women in Machine Learning Workshop(WiML 2020)

6.2 Other Publications

6.2.1 Journal Proceedings

1. Mounika Marreddy, Oota Subba Reddy, Vakada Lakshmi Sireesha, Chinni Venkat Charan, Radhika Mamidi: Am I a Resource-Poor Language? Data Sets, Embeddings, Models and Analysis for four different NLP tasks in Telugu Language

   ACM Transactions on Asian and Low-Resource Language Information Processing, 2022

6.2.2 Conference Proceedings

1. Mounika Marreddy, Oota Subba Reddy, Vakada Lakshmi Sireesha, Chinni Venkat Charan, Radhika Mamidi: Clickbait Detection in Telugu: Overcoming NLP Challenges in Resource-Poor Languages using Benchmarked Techniques

   International Joint Conference on Neural Networks (IJCNN), 2021

2. Mounika Marreddy, Oota Subba Reddy, Vakada Lakshmi Sireesha, Chinni Venkat Charan, Radhika Mamidi: Multi-Task Text Classification using Graph Convolutional Networks for Large-Scale Low Resource Language

   International Joint Conference on Neural Networks (IJCNN), 2022

Bibliography

[1] R. M. Aliguliyev. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications*, 36(4):7764–7772, 2009.

[2] G. Babu and S. Badugu. Deep learning based sequence to sequence model for abstractive telugu text summarization. *Multimedia Tools and Applications*, pages 1–22, 2022.

[3] R. Bommasani and C. Cardie. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8075–8096, 2020.

[4] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014.

[5] R. R. Chowdhury, M. T. Nayeem, T. T. Mim, M. S. R. Chowdhury, and T. Jannat. Unsupervised abstractive summarization of bengali text documents. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2612–2619, 2021.

[6] T. Chowdhury, S. Kumar, and T. Chakraborty. Neural abstractive summarization with structural attention. *arXiv preprint arXiv:2004.09739*, 2020.

[7] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

[8] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. M. Khapra, and P. Kumar. Indicbart: A pre-trained model for natural language generation of indic languages. *arXiv preprint arXiv:2109.02903*, 2021.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] H. P. Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.

[11] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *JAIR*, 22:457–479, 2004.

[12] Q. Grail, J. Perez, and E. Gaussier. Globalizing bert-based transformer architectures for long document summarization. In Proceedings of the 16th conference of the European chapter of the Association for Computational Linguistics: Main volume, pages 1792–1810, 2021.

[13] M. Grusky, M. Naaman, and Y. Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. arXiv preprint arXiv:1804.11283, 2018.

[14] A. N. Gulati and S. Sawarkar. A novel technique for multidocument hindi text summarization. 2017 International Conference on Nascent Technologies in Engineering (ICNTE), pages 1–6. IEEE, 2017.

[15] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In CVPR'06, volume 2, pages 1735–1742. IEEE, 2006.

[16] A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. NAACL, pages 362–370, 2009.

[17] M. M. Haque, S. Pervin, and Z. Begum. Automatic bengali news documents summarization by introducing sentence frequency and clustering. 2015 18th International Conference on Computer and Information Technology (ICCIT), pages 156–160. IEEE, 2015.

[18] T. Hasan, A. Bhattacharjee, M. S. Islam, K. Mubasshir, Y.-F. Li, Y.-B. Kang, M. S. Rahman, and R. Shahriyar. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. In ACL-IJCNLP, pages 4693–4703, 2021.

[19] M. Hassel. Exploitation of named entities in automatic text summarization for swedish. NODALIDA'03–14th Nordic Conferenceon Computational Linguistics, Reykjavik, Iceland, May 30–31, 2003, pages 9, 2003.

[20] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. NeurIPS, 28:1693–1701, 2015.

[21] A. Joshi, E. Fidalgo, E. Alegre, and L. Fernández-Robles. Summcoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. Expert Systems with Applications, 129:200–215, 2019.

[22] D. Kakwani, A. Kunchukuttan, S. Golla, N. Gokul, A. Bhattacharyya, M. M. Khapra, and P. Kumar. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4948–4961, 2020.

[23] R. Karmakar, K. Nirantar, P. Kurunkar, P. Hiremath, and D. Chaudhari. Indian regional language abstractive text summarization using attention-based lstm neural network. 2021 International Conference on Intelligent Technologies (CONIT), pages 1–8. IEEE, 2021.

[24] T. N. Kipf and M. Welling. Semi-supervised classi cation with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

[25] T. N. Kipf and M. Welling. Variational graph auto-encoders. arXiv preprint arXiv:1611.07308, 2016.

[26] A. Kumar, H. Shrotriya, P. Sahu, R. Dabre, R. Puduppully, A. Kunchukuttan, A. Mishra, M. M. Khapra, and P. Kumar. Indicnlg suite: Multilingual datasets for diverse nlg tasks in indic languages. arXiv preprint arXiv:2203.05437, 2022.

[27] Y. Kumar, K. Kaur, and S. Kaur. Study of automatic text summarization approaches in different languages. Arti cial Intelligence Review, 54(8):5897–5929, 2021.

[28] N. Kumari and P. Singh. Hindi text summarization using sequence to sequence neural network. 2022.

[29] F. Ladhak, E. Durmus, C. Cardie, and K. McKeown. Wikilingua: A new benchmark dataset for cross-lingual abstractive summarization. arXiv preprint arXiv:2010.03093, 2020.

[30] G. Lample and A. Conneau. Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291, 2019.

[31] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461, 2019.

[32] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. Text Summarization Branches Out, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[33] C.-Y. Lin and E. Hovy. Identifying topics by position. In Fifth Conference on Applied Natural Language Processing, pages 283–290, 1997.

[34] C.-Y. Lin and E. Hovy. Manual and automatic evaluation of summaries. In ACL-02 Workshop on Automatic Summarization, pages 45–51, 2002.

[35] D. Lin and P. Pantel. Jensen-shannon distance as a measure of distributional similarity for natural language. In Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing, pages 388–395, 2001.

[36] M. Litvak and M. Last. Graph-based keyword extraction for single-document summarization. In Coling 2008: Proceedings of the workshop multi-source multilingual information extraction and summarization, pages 17–24, 2008.

[37] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer. Multilingual denoising pre-training for neural machine translation. Transactions of the Association for Computational Linguistics, 8:726–742, 2020.

[38] Y. Liu and M. Lapata. Text summarization with pretrained encoders. EMNLP-IJCNLP, pages 3730–3740, 2019.

[39] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017.

[40] H. P. Luhn. The automatic creation of literature abstracts. IBM Journal of research and development, 2(2):159–165, 1958.

[41] K. K. Mamidala et al. A heuristic approach for telugu text summarization with improved sentence ranking. TURCOMAT, 12(3):4238–4243, 2021.

[42] K. U. Manjari. Extractive summarization of telugu documents using textrank algorithm. In *ICSMAC*, pages 678–683. IEEE, 2020.

[43] D. Marcheggiani and I. Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *EMNLP*, pages 1506–1515, 2017.

[44] M. Marreddy, S. R. Oota, L. S. Vakada, V. C. Chinni, and R. Mamidi. Clickbait detection in telugu: Overcoming nlp challenges in resource-poor languages using benchmarked techniques. In *IJCNN*, pages 1–8. IEEE, 2021.

[45] M. Marreddy, S. R. Oota, L. S. Vakada, V. C. Chinni, and R. Mamidi. Am i a resource-poor language? data sets, embeddings, models and analysis for four different nlp tasks in telugu language. *Transactions on Asian and Low-Resource Language Information Processing*, 2022.

[46] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *EMNLP*, pages 404–411, 2004.

[47] R. Naidu, S. K. Bharti, K. S. Babu, and R. K. Mohapatra. Text summarization with automatic keyword extraction in telugu e-newspapers. In *Smart computing and informatics*, pages 555–564. Springer, 2018.

[48] R. Nallapati, F. Zhai, and B. Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*, 2017.

[49] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[50] S. Narayan, S. B. Cohen, and M. Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*, 2018.

[51] A. Nenkova and L. Vanderwende. The impact of frequency on summarization.

[52] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NeurIPS*, pages 849–856, 2002.

[53] K. Nguyen and H. Daumé III. Global voices: Crossing borders in automatic news summarization. *arXiv preprint arXiv:1910.00421*, 2019.

[54] Y. Ouyang, W. Li, Q. Lu, and R. Zhang. A study on position information in document summarization. In *Coling 2010: Posters*, pages 919–927, 2010.

[55] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[56] R. Paulus, C. Xiong, and R. Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

[57] R. Paulus, C. Xiong, and R. Socher. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*, 2018.

[58] T. Pires, E. Schlinger, and D. Garrette. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*, 2019.

[59] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, et al. Exploring the limits of transfer learning with a uni ed text-to-text transformer. J. Mach. Learn. Res., 21(140):1–67, 2020.

[60] P. Ren, Z. Chen, Z. Ren, F. Wei, J. Ma, and M. de Rijke. Leveraging contextual sentence relations for extractive summarization using a neural attention model. SIGIR, pages 95–104, 2017.

[61] K. Sarkar. Bengali text summarization by sentence extraction. arXiv preprint arXiv:1201.2240, 2012.

[62] J. Schmidhuber. Deep learning in neural networks: An overview. Neural networks, 61:85–117, 2015.

[63] T. Scialom, P.-A. Dray, S. Lamprier, B. Piwowarski, and J. Staiano. Mlsum: The multilingual summarization corpus. arXiv preprint arXiv:2004.14900, 2020.

[64] S. Sinha and G. N. Jha. An overview of indian language datasets used for text summarization. arXiv preprint arXiv:2203.16127, 2022.

[65] E. S. Soriano, V. Ahuir, L.-F. Hurtado, and J. Gónez. Dacsa: A large-scale dataset for automatic summarization of catalan and spanish newspaper articles. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5931–5943, 2022.

[66] D. Suleiman and A. Awajan. Deep learning based abstractive text summarization: approaches, datasets, evaluation measures, and challenges. Mathematical problems in engineering, 2020, 2020.

[67] K. Thadani and K. McKeown. Supervised sentence fusion with single-stage inference. In Proceedings of the Sixth International Joint Conference on Natural Language Processing, pages 1410–1418, 2013.

[68] P. Tikhonov and V. Malykh. Wikimulti: a corpus for cross-lingual summarization. arXiv preprint arXiv:2204.11104, 2022.

[69] A. Urlana, N. Surange, P. Baswani, P. Ravva, and M. Shrivastava. Tesum: Human-generated abstractive summarization corpus for telugu. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 5712–5722, 2022.

[70] D. Varab and N. Schluter. Massivesumm: a very large-scale, very multilingual, news summarisation dataset. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10150–10161, 2021.

[71] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin. Attention is all you need. In NeurIPS, pages 5998–6008, 2017.

[72] S. Vijay, V. Rai, S. Gupta, A. Vijayvargia, and D. M. Sharma. Extractive text summarisation in hindi. In 2017 International Conference on Asian Language Processing (IALP), pages 318–321. IEEE, 2017.

[73] Q. Xie, J. Huang, P. Du, M. Peng, and J.-Y. Nie. Inductive topic variational graph auto-encoder for text classi cation. In NAACL: Human Language Technologies, pages 4218–4227, 2021.

[74] H. Xu, Y. Wang, K. Han, B. Ma, J. Chen, and X. Li. Selective attention encoders by syntactic graph convolutional networks for document summarization. In ICASSP, pages 8219–8223. IEEE, 2020.

[75] J. Xu, Z. Gan, Y. Cheng, and J. Liu. Discourse-aware neural extractive text summarization. In *ACL*, pages 5021–5031, 2020.

[76] S. Xu, X. Zhang, Y. Wu, F. Wei, and M. Zhou. Unsupervised extractive summarization by pre-training hierarchical transformers. In *EMNLP*, pages 1784–1795, 2020.

[77] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

[78] L. Yao, C. Mao, and Y. Luo. Graph convolutional networks for text classification. In *AAAI*, volume 33, pages 7370–7377, 2019.

[79] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. Radev. Graph-based neural multi-document summarization. *arXiv preprint arXiv:1706.06681*, 2017.

[80] L. Zhang and H. Chen. Generating summaries of long texts using the titles and first sentences. In *International Conference on Natural Language Processing and Knowledge Engineering*, pages 442–447. Springer, 2010.

[81] Y. Zhang, J. E. Meng, and M. Pratama. Extractive document summarization based on convolutional neural networks. In *IECON*, pages 918–922. IEEE, 2016.

[82] J. Zhao, M. Liu, L. Gao, Y. Jin, L. Du, H. Zhao, H. Zhang, and G. Haffari. Summpip: Unsupervised multi-document summarization with sentence graph compression. In *SIGIR*, pages 1949–1952, 2020.