Enhancing Text Summarization for Indian Languages: Mono, Multi and Cross-lingual Approaches

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

Master of Science in
Computer Science and Engineering
by Research

by

ASHOK URLANA
2020701023
ashok.urlana@research.iiit.ac.in

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

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Enhancing Text Summarization for Indian Languages: Mono, Multi and Cross-lingual Approaches" by Ashok Urlana, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Dr. Manish Shrivastava

This work is dedicated to my family, friends and guide for their love, boundless support and encouragement.

Acknowledgments

I hereby express my sincere gratitude to my supervisor, Dr. Manish Shrivastava, for his unwavering support throughout the course of my Master's thesis. I am immensely grateful for his guidance and motivation, starting from the first day of our discussion, which helped me formulate the problem statement and gain a deeper understanding of the research area. His technical expertise and valuable suggestions helped shape this thesis at every stage. I also gained his personal and ethical insights into research, which will be invaluable for my future endeavors.

I would like to thank my collaborators for this thesis, Priyanka Ravva, Nirmal Surange, Pavan Baswani, Lokesh Madasu, and Gopichand Kanumolu. Their help was invaluable in completing this thesis. I would also like to acknowledge my colleagues Pruthwik Mishra and Vandan Mujadia, constantly supported me during my entire Master's program. Our countless hours of intense discussions provided me with a clear understanding of best research practices. I would like to express my gratitude to K. Ravikanth, J. Chakravarthi, and Satish Kumar Ch, faculty members at RGUKT-Basar, RGUKT-Nuzvid, and RGUKT-Srikakulam respectively, for their unwavering support in coordinating internship activities.

I express my heartfelt thanks to IIIT Hyderabad for providing the coursework that laid the foundation for my research work. I thank Prof. C.V Jawahar, Prof. Dipti Misra Sharma, Dr. Radhika Mamidi, and Dr. Pawan Kumar for their insightful lectures, assignments, and quizzes. The concepts I learned in the courses helped me enhance my technical skills, and the projects I completed provided the basis for my research work. I am also grateful to my friends Hiranmai, Ananya, Prashanth, Palash, Sai Teja, Srikar, Karthik, Vijay, and Saideep for making my time at IIIT Hyderabad memorable.

Finally, I express my deepest gratitude to my family members for their constant support. The immense support I received from my brother was instrumental in helping me complete my Master's degree.

Abstract

The internet serves as a vast repository of information covering a diverse array of topics, ranging from blogs and articles to websites. However, it is important to note that not all of this information is valuable or relevant. Navigating through the plethora of content in order to gain a comprehensive understanding of a particular topic can be a daunting and time-consuming task. Furthermore, it is all too common to invest time in reading content that ultimately proves to be unimportant or irrelevant. Given the inherent limitations of the human cognitive capacity to process large quantities of information, concise and relevant summaries are highly sought after in order to efficiently and effectively comprehend complex subjects.

Summarization is a computational task that condenses textual information into a concise version by including only the most essential and relevant information. There are two main approaches to summarization: extractive and abstractive. Extractive summarization involves selecting sentences based on their importance, while abstractive summarization involves introducing new words or phrases in the summary. Document summarization has been studied for over three decades by the NLP community. However, progress in Indian language summarization has been limited due to the lack of high-quality datasets and benchmark models, which has motivated us to work towards developing resources and benchmarks for Indian languages. In this thesis, we have developed text summarization resources for Indian languages in three different settings: mono-lingual, cross-lingual, and multi-lingual.

The initial focus of this thesis is on mono-lingual summarization, specifically creating a high-quality dataset for the popular south Indian language, Telugu. We propose a pipeline that crowd-sourced summarization data and then aggressively filtered the content via: automatic and partial expert evaluation. Using the pipeline, we create a high-quality Telugu abstractive summarization dataset (TeSum). The dataset consists of 20,329 document-summary pairs, which were created by 347 annotators and evaluated by 3 raters. We carefully designed annotation guidelines that consider the parameters of Relevance, Readability, and Creativity. Additionally, we compared our dataset with existing Telugu summarization datasets.

By training a summarization system on multiple languages, the system can learn to represent concepts in a shared space, regardless of the language in which they are expressed. This shared representation learning can be useful for transfer learning, as it enables the model to apply knowledge gained from one language to another language. To achieve this, we perform the multi-lingual and cross-lingual summarization for Indian languages.

For multi-lingual summarization, we utilized the Indian Language Summarization (ILSUM) dataset to create baselines, which includes Hindi, Gujarati, and Indian English. We test the proposed filters on ILSUM data to perform the quality assessment. We conducted experiments with different pre-trained sequence-to-sequence models to identify the best-performing model for each language. Our work also involved an in-depth analysis of the impact of k-fold cross-validation when dealing with limited data. Additionally, we performed experiments using a combination of the original and filtered versions of the data to assess the effectiveness of the pre-trained models.

We present the PMIndiaSum, a new cross-lingual and highly parallel summarization dataset for languages in India. The dataset covers 4 language families, 14 languages, and 196 language pairs. We detail the approaches taken to derive this dataset, including data acquisition, cleaning, quality assurance, and inspection. In addition, we publish benchmarks for various methodologies, such as fine-tuning pre-trained language models and summarization-and-translation. Experimental results suggest that the provision of multilingual data enhances cross-lingual summarization between Indian languages.

Furthermore, this thesis also delves into multi-perspective scientific document summarization. Our objective is to develop a model that can generate a generic summary encompassing various aspects covered by multiple reference summaries of a scientific document. We describe the different pre-trained models used in this task, as well as the challenges encountered during the process.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In today's world, information is primarily found online, with the World Wide Web containing billions of documents that continue to increase exponentially. As a result, people are facing information overload and require tools that provide timely access to and a digest of various sources to alleviate this issue. To address these concerns, automatic summarization systems have been developed. These systems can take a single article, cluster of news articles in one or more languages, broadcast news show, or an email thread and produce a concise and fluent summary of the most critical information. In recent years, many summarization applications have been created for various types of information, such as news, medical information, scientific articles, spontaneous dialogues, and more. Although these systems are not perfect, they have been shown to help users and improve other automatic applications and interfaces. The availability of processed text data is currently limited to certain languages, such as English, which restricts the usage of natural language processing (NLP) applications to specific communities. In order to expand the user base of NLP applications, it is crucial to develop NLP systems for a wide range of languages. This thesis aims to build the motivation for the development of text summarization resources and systems specifically for Indian languages.

1.1 Motivation

The field of text summarization has undergone significant developments over the past few decades, driven by the continuous efforts of NLP researchers. Early experiments primarily relied on surface-level phenomena such as sentence position and word frequency counts, with a focus on extracting text from the source rather than generating new text. In the early years of NLP literature, there are numerous descriptions of summarization tasks with different goals. According to the DUC (2003-2007) guidelines, summarization is defined as the task of generating a concise text that provides a general idea of the source article. In a previous study, Hovy and Lin [29] pondered the precise definition of a summary and proposed an answer:

A summary is a text that is produced out of one or more (possibly multimedia) texts, that contains (some of) the same information of the original text(s), and that is no longer than half of the original text(s).

Hovy and Lin [29] and KS Jones [72] endeavored to establish a definition of summarization that encompasses more than just a brief indication of the source article's content. In fact, Hovy and Lin [29] builds upon and expands upon the work of KS Jones [72] to categorize summaries based on broader aspects such as input, purpose, and output in a more refined manner. In this thesis, we present a diverse range of datasets and summarization systems to cater to the following types of summaries.

1.2 Typology of Summaries

Summarization often involves various distinctions, which are commonly discussed in the literature [72, 29, 56]. In this context, we provide definitions for key terminologies that are frequently used in summarization research. These terminologies are further classified based on source (document), target (summary) and purpose.

1.2.1 Source Oriented

Single *vs.* **Multi-document**: Single-document summarization is commonly used when the aim is to obtain a brief overview of a single piece of text. On the other hand, multi-document summarization involves condensing information from multiple thematically similar documents into a concise summary. This approach is useful when seeking a comprehensive understanding of a topic or event by consolidating information from multiple resources. The key distinction between single and multi-document summarization lies in the scope of the input or source document size.

Domain specific *vs.* **General:** A domain-specific summary is generated from one or more documents that are focused on a particular domain or field. Domain-specific summarization models are specifically designed to capture the unique terminologies, jargon, and writing styles commonly used within that domain. On the other hand, a general domain summary is capable of providing a summary of input text(s) from any domain, without being limited to a specific field or subject matter.

1.2.2 Target Oriented

Extractive *vs.* **Abstractive Summaries:** *Extractive summaries* are created by combining important snippets, sentences, or passages from documents. These extractions are taken verbatim from the original document. On the other hand, *abstractive summaries*, aim to convey relevant information by rephrasing and reusing phrases or clauses, but they are expressed in the words chosen by the summary author.

Guided *vs.* **Controllable summarization:** Guided summarization involves using different types of guidance signals to minimize the deviation of the summary from the source document [18]. It incorporates the additional information to guide the summarization process. This allows for controllability

by incorporating user-specified inputs. On the other hand, controllable summarization empowers the reader to control essential aspects of the generated summary, such as desired length, focus on specific entities, and the style based on their preferred source of information [20].

1.2.3 Purpose Oriented

Generic *vs.* **Query focused summaries:** A generic summary provides the author's view of the input text by giving equal importance to all aspects of the source text. On the other hand, a query-oriented (user-oriented) summary prioritizes specific aspect(s) or themes based on the user's desire to learn about a particular aspect. This type of summary may omit certain themes or aspects in order to directly address the user's query. To generate a useful summary in this context, the summarizer model needs to take the query as well as document into account.

Indicative *vs.* **Informative:** An indicative summary provides the high level overview of the input text, with out covering all the contents. It provides a general idea or indication of what the input text is about with out covering specific details or supporting evidence. Where as after reading the informative summary one can explain what the input article is about but not necessarily what was contained in it. An informative summary often consists of specific details, data, and the examples from the original content and typically written in a formal and objective tone.

Role based or Multi-perspective: A multi-perspective summary provides a summary of a given text, focusing on specific themes or aspects of the input text. For example, for a scientific document, there could be multiple valid summaries, each providing a summary from a different perspective. One summary could focus on the "abstract and introduction" sections of the scientific document, while another could concentrate on the "results and conclusions" sections of the same document.

1.3 Summarizing Across Languages

Formally, given a source document D, the process of summarization should produce a target summary S with a shorter length, yet conveying the most important message in D. We explore three types of summarization tasks defined by language directions.

1. Monolingual: document D_L and summary S_L are in the same language L.
2. Cross-lingual: document D_{L1} and summary S_{L2} are in different languages $L1$ and $L2$.
3. Multilingual: monolingual and cross-lingual summarization from $D_{\{L1,L2,...,Ln\}}$ to $S_{\{L1,L2,...,Lm\}}$ within a single model.

In our context, we limit cross-lingual models to ones that summarize from one single language to another. Mono- and cross-lingual models are considered to be more focused on the target language and thus more accurate, whereas a multilingual model can significantly save storage and computation, and leverage

the potential to share knowledge across languages. This thesis offers a comprehensive collection of summarization systems and datasets that encompass mono, multi, and cross-lingual approaches.

1.4 Progress in Indian Language Summarization

The development of summarization datasets for Indian languages has been the focus of several attempts, but one of the major challenge is ensuring public accessibility of these datasets [71]. These datasets can be monolingual or multilingual, enabling summarization in either a monolingual way, with separate models for each language, or a single model for all languages. However, the existing datasets have limitations in terms of language coverage and accessibility.

1.4.1 Language Coverage and Datasets Availability:

The early statistical and probabilistic models paved the way for the emergence of deep learning models that have led to significant progress in various automatic summarization tasks. This progress has extended to Indian languages in recent years, where the lack of reliable datasets posed a significant challenge to the development of text summarization models. The XL-Sum[27] dataset supports eight Indian languages (Bengali, Gujarati, Hindi, Marathi, Punjabi, Tamil, Telugu and Urdu). Another dataset, MassiveSumm[79], covers 12 languages, but the corpus is not publicly accessible, and the license is unknown. Recently, released Indian language summarization (ILSUM)[68] dataset only covers Hindi, Gujarati, and Indian English.

For cross-lingual summarization between Indian languages, the WikiLingual[37] dataset supports only Hindi. However, the CrossSum dataset is currently the only publicly accessible dataset that supports summarization between Indian languages, covering 56 pairs formed by pairing up 8 languages in XL-Sum.

Efforts to create and share reliable datasets for Indian languages can greatly contribute to the development of text summarization models, enabling researchers and practitioners to build more accurate and effective summarization systems for a wider range of Indian languages, and ultimately benefiting various natural language processing (NLP) applications such as news summarization, content generation, and information retrieval in these languages.

1.4.2 Pre-trained Models:

The field of text summarization for Indian languages has been significantly impacted by recent advancements in neural-based multi-lingual pre-trained models. Specifically, Google's mT5 [82], Facebook's mBART-50 [74], and IndicBART [16] have emerged as notable models in this domain. mT5 currently provides support for 10 Indian languages, while mBART-50 supports nine Indian languages, and IndicBART is designed to cater exclusively to Indian languages, supporting 12 of them. These models have opened up new opportunities to enhance text summarization capabilities for Indian languages, marking a significant transformation in the field.

1.5 Thesis Contribution

The main objective of this thesis is to advance the field of text summarization for Indian languages, addressing the challenges posed by the lack of reliable datasets while improving the quality of existing resources and models. Through our research, we hope to provide valuable insights into the field of text summarization and contribute to its growth, especially in the context of Indian languages. This thesis discusses the challenges in developing mono, multi, and cross-lingual resources for Indian languages and provides several benchmarks. The following are the contributions of this thesis:

- We provide a high-quality Telugu abstractive summarization dataset. This is achieved by using a carefully designed filtration techniques and benchmark models.
- This thesis offers mono-lingual summarization benchmarks for Hindi, Gujarati, and Indian English.
- It also provides a massively parallel cross-lingual summarization dataset for 14 Indian languages, along with corresponding baselines.
- Additionally, it explores various modeling approaches for multi-perspective scientific document summarization.

1.6 Thesis Workflow

The thesis is structured as follows: Chapter 2 briefly discusses existing summarization datasets and systems for Indian languages. Chapter 3 demonstrates the need for monolingual summarization resources and provides benchmarks and details on quality comparisons among various summarization datasets. Moreover, proposes the pipeline to create high-quality datasets for low-resource languages. Chapter 4 explains the setup for performing multilingual summarization and the challenges involved in it. Chapter 5 discusses the need for cross-lingual summarization and presents a novel dataset and corresponding benchmark experiments. Chapter 6 provides details on the novel task of multi-perspective scientific document summarization, including corresponding experiments and analysis. Chapter 7 details the conclusions and future works.

Chapter 2

Literature Review

Text summarization in low resource languages has long been a persistent challenge in the field of Natural Language Processing. There are distinct methods for abstractive and extractive summarization, but they share common features such as salience and coverage. Various approaches have been employed for summarizing Indian languages, ranging from statistical and linguistic-based techniques to pure machine learning and deep learning methods. In this chapter, we discuss different approaches for summarization, including extractive, abstractive, guided, and multi-perspective summarization. We also provide a detailed analysis of existing summarization datasets for Indian languages, along with the corresponding challenges in creating high-quality datasets. Additionally, we cover the language support of these datasets with existing pre-trained models, and the availability of resources for mono, multi, and cross-lingual summarization approaches.

2.1 Summarization Approaches

2.1.1 Extractive Summarization

The main challenge in extractive summarization is to effectively select and organize important sentences. Initially, heuristic-based methods [17], frequency-based techniques, and clustering approaches [32] were used to summarize Telugu. However, the k-means extraction method suffered from out-of-order extraction, although the summaries generated through the frequency-based approach were more fluent. To address the issues with sentence ranking, TextRank [44] and PageRank [17] algorithms were applied.

Several works have attempted multi-document summarization in Telugu, as described in the works of [62] and [83]. It is worth noting that most summarization systems for Indian languages are designed for extractive summarization, as it is relatively easier to implement. Examples of such extractive summarizers can be found in [30, 65, 59, 67, 76, 26, 7, 9].

2.1.2 Abstractive Summarization

Abstractive summarization methods are known to generate more fluent and coherent summaries compared to extractive methods. In earlier work on abstractive summarizers for Indian languages, information extraction techniques [31] and automatic keyword extraction methods [50] were used. These summarizers utilize parts-of-speech tagging to create headlines and comprise various components, such as word cues, keyword extraction, sentence selection, sentence extraction, and summary generation modules. These components analyze the textual data to identify the main features of the summary.

Recent abstractive approaches to summarization, such as neural attention models [66], Seq2Seq RNNs [52], Pointer-Generator networks [69] focus on generating summaries that capture the meaning of the input text without necessarily choosing sentences directly from the text. With the emergence of large neural language models for generation tasks, abstractive approaches have become more popular and generate high-quality summaries. While there have been various improvements in model architectures and summarization techniques, a large part of the progress in English text summarization can be attributed to the availability of large-scale datasets, such as CNN/DailyMail [52, 28], Gigaword [66, 23], XSum [55], etc.

2.1.3 Guided and Controlled Summarization

Abstractive summaries, while they may sound fluent, often suffer from issues due to their unconstrained nature. These issues include unfaithful summaries that may contain factual errors or hallucinated content. Moreover, it is challenging to train models to focus on specific aspects and control the summaries. To address these problems, a recent study by Dou et al. (2021) [18], proposed guided summarization as a solution. This approach constrains the output generation of models to minimize deviation from the original source and allows for controllability through user-specified inputs.

Previous research has also explored ways to guide neural abstractive summarization models. For instance, Kikuchi et al. (2016)[33] focused on specifying the length of summaries, while Li et al. (2018)[39] provided models with keywords to avoid omitting crucial information. Cao et al. (2018)[11] proposed models that retrieve and reference relevant summaries from the training set. In their recent work, Dou et al. (2021)[18] introduced a framework to investigate four types of guidance signals, including highlighted sentences, keywords, salient relation triples (subject, relation, object), and retrieved summaries.

2.1.4 Multi-perspective Summarization

Research on summarizing scientific documents has been widely explored in recent years. It is pertinent to note that there is a great deal of variation in the density of information covered [58], the level of detail, and the organization of the content within the scientific document summaries. Recent work by Fabbri [19] uses question threads from the Yahoo forum to build the multi-perspective answer summarization

corpus. Meng et al., [46] present FactSum that contains four summaries for each paper covers different aspects, they can provide summaries based on user requests.

A number of scholarly document summarization datasets, including PubMed and arXiv [13], were used for training neural models ScisummNet [85] and SciTLDR for extreme summarization [10]. Unlike these datasets, Multi-perspective summarization shared task organizers released a multi-perspective summarization dataset for scientific documents. Various generation models, including BART, T5, ProphetNet, and PEGASUS, have shown great performance in summarization tasks. In particular, models like Big Bird [86] and Longformer [5] were released to handle long documents.

2.2 Available Datasets

2.2.1 Mono and Multi-lingual Datasets

Documents and summaries are the key components of a summarization dataset. A typical workflow to gather such is to make use of publicly available articles, especially news, given its online availability and language coverage. When an article is regarded as a document, a valid summary can be either human-produced, or some text accompanying the document, such as the headline, highlights, or the first sentence. The latter can be much cheaper but more error-prone. There have been various attempts to develop summarization datasets for languages in India, but one of the primary challenges is ensuring their public accessibility [71].

In recent years, several datasets for automatic abstractive summarization have been proposed. Many of these datasets were inspired by the DUC tasks, which involve generating a summary for one or more given articles based on a given topic. The CNN/Daily Mail corpus for summarization [28, 53] is created by using the CNN and Daily Mail news stories as the document, paired with human-generated summaries. This dataset was soon followed by other datasets such as Gigaword, Newsroom, and XSUM. These monolingual datasets led the way to multilingual datasets in the form of XL-Sum, MassiveSumm, IndicnNLG etc.

Hasan et.al [27] created XL-Sum by collecting articles from the BBC website. They used the first sentence of the article body as the summary and the rest as the document. [79] developed MassiveSumm by scraping news articles from a wide range of domains. More recently, the Indian Language Summarization dataset (ILSUM) dataset was obtained by scraping news articles and corresponding headlines from several leading newspapers in India [68].

While this approach of scraping news and highlights does lead to large numbers which are suitable for training large deep learning models, it is safe to assume that if the quality of the data is not up to the mark, the model outputs would also suffer. For Telugu, we evaluated XL-Sum and MassiveSumm datasets, and came to the conclusion that the dataset qualities could not be considered as human summarization, therefore we set up for a task of creating human generated and curated summarization dataset for

Telugu. We find that the curation policies can even be extended to scraped data such as XL-Sum and MassiveSumm.

2.2.2 Cross-lingual Datasets

Attempts to exploit articles for summarization usually result in monolingual or multilingual datasets, but not cross-lingual. One consideration is to have existing datasets undergoing secondary processing. [89] proposed to translate the summaries in existing monolingual datasets into another language. They perform quality control using round-trip translation to make sure the translation of a translated summary does not deviate from the original too much. The CrossSum dataset [6] was derived from XL-Sum by computing the semantic similarity between cross-lingual document-summary pairs in XL-Sum. A different paradigm proposed by [37] is to align summaries and documents in different languages using image pivots in the how-to articles on the WikiHow website.

Our proposed dataset PMIndiaSum falls into the spectrum of utilizing publicly available news article-headline pairs. The data is massively parallel and cross-lingual thanks to the natural multilingualism of the published articles. We greatly benefit from the raw data and tools released by PMIndia [25], which is a parallel corpus between English and Indian languages, extracted from the Prime Minister of India website.

2.3 Indian language coverage

Monolingual and multilingual datasets enable summarization in a monolingual fashion, with either separate models for each language, or a single model for all. Regarding language coverage, TeSum is only for Telugu, and ILSUM consists of Hindi, Gujarati, and Indian English. Moreover, XL-sum supports eight Indian languages, however, its assumption of using the first sentence of a document as a summary may not be valid for Indian languages as revealed by [78]. The IndicNLG [35] benchmark dataset provides support for sentence summarization in 11 Indian languages. The dataset is created by utilizing the first sentence of a document as input and its corresponding headline as the summary. The IndicNLG Suite also provides the datasets for several Indic language NLG tasks, such as headline generation, parapharse generation, question and biography generation.

In terms of summarizing into a different language, WikiLingual [37] is cross-lingual, but it involves merely one Indian language, Hindi. Further, MassiveSumm supports 12 Indian languages but the corpus is not publicly accessible[1]. To the best of our knowledge, CrossSum is currently the only publicly available dataset that supports cross-lingual summarization between Indian languages: in total 56, formed by pairing up the 8 different languages in XL-Sum. Meanwhile, our proposed PMIndiaSum dataset supports 13 Indian languages and in total $13 \times 12 = 156$ cross-lingual pairs. Assuming the zero-shot train-test paradigm, multilingual datasets can be used for cross-lingual summarization.

[1] https://github.com/danielvarab/massive-summ

To the best of our knowledge, PMIndiaSum is the first massively parallel dataset that supports both multilingual and cross-lingual summarization for Indian languages, setting it apart from existing datasets.

Chapter 3

Monolingual Summarization

The task of monolingual summarization involves generating summaries in the same language as the original document. This chapter discusses the process of creating and validating datasets and benchmark models for monolingual summarization in Indian languages. The focus of this chapter is on the creation of a human-annotated abstractive summarization dataset for the widely spoken south Indian language 'Telugu'. Additionally, the chapter covers the filtration techniques employed to ensure the quality of the datasets.

3.1 Introduction

In recent years, much work has been done to advance the state of the art of summarization for multiple languages across the world. But, most of these works adhere more closely to the DUC summarization challenge rather than the nuanced definitions presented by Hovy and Lin [29]. We find that collection of summarization data is reduced to mass scraping of various news sources across the world, in order to source the articles from the real world. At the same time, the expensive task of summary creation is reduced to clever partitioning of the content already available on online news media . In Hermann [28], we first saw, the usage of news articles along with their highlights available on certain reputed news outlets for the purpose of cloze-kind question answering. This data was later re-purposed for the summarization task by using the highlights as a proxy to the summary itself, trusting an implicit assumption which is based on strict editorial policies implemented by some news publications.

The underlying assumption is that these highlights or bullet points preceding an article are editorially required to convey a broad idea about the content of the article. While this assumption is an inspired one, and makes sense for a large number of news sources with strict editorial quality control, unfortunately, the assumption cannot be blindly extended to a vast majority of news sources. And it can be easily found that this assumption fails even for a large number of English news summarization datasets. Even so, many recent works have followed this strategy for creating massive summarization data sets fit for the training of deep neural networks. There are other problems as well. While such an approach might give

	Train		Validation		Test	
# Pairs	16295		2017		2017	
Avg Compression%	58.26		58.08		58.28	
	Text	Summary	Text	Summary	Text	Summary
# Unique Words	183641	113723	49038	28873	49620	28777
Avg Unique Words	88.93	42.78	89.19	43.14	90.74	43.81
(Min, Max) Words	(30, 536)	(12, 213)	(32, 685)	(10, 248)	(36, 592)	(12, 261)
Avg Words	120.8	50.02	122.56	50.8	124.82	51.69
Avg Sentences	9.23	3.22	9.50	3.19	9.51	3.17

Table 3.1: TeSum Statistics

us some “summary", one cannot guarantee if 1) the summary is abstractive, unless explicitly measured for it, and 2) the summary is coherent.

We look at a particular Indian language, Telugu, for which such datasets are available as XL-Sum [27] and MassiveSumm [79] which have been collected from various sources. Both rely on above mentioned assumptions. Even with a casual observation, we find that these assumptions, and therefore these datasets do not stand up to the test. Therefore, we find that there is a urgent and immediate need for a dataset and a dataset creation methodology which stays true to the essence of the task of summarization as defined in [29]. Such a dataset must be created with human involvement and thorough evaluation. While we acknowledge that purely human-generated summaries may be costly, we propose a methodology that ensures high-quality datasets without relying heavily on human involvement. We recognize that both creation and evaluation of summaries are expensive processes, with evaluation often being more costly than creation. Taking this into consideration, we propose a Human-generated summary creation pipeline. We propose a combination of automated and human evaluations to ensure a high quality dataset for Telugu (can be extended to other languages). We present, TeSum, a Human-Generated, curated Abstractive Summarization data set for Telugu[1] (Table 3.1).

We compare the resultant dataset with the existing datasets for the language and show that in the light of some well-motivated criteria, both XL-Sum[2](Telugu) and MassiveSumm [3](Telugu) do not live up to the expectations.

[1] https://ltrc.iiit.ac.in/showfile.php?filename=downloads/teSum/

[2] XL-Sum data is taken from the publicly available repository at https://github.com/csebuetnlp/xl-sum

[3] We thank the authors of MassiveSumm for graciously providing us with the entire MassiveSumm datasets, for our experiments.

3.2 Crowd-Sourced Corpus Creation

We suggest a two-step process for creating summaries, involving a crowd-sourced phase followed by curation by trained experts. In the crowd-sourced phase, we work with 347 “creators" and 3 expert “raters" who are provided with specific guidelines to ensure the quality of the generated content. The content is then rigorously filtered to retain high-quality article-summary pairs. In the second phase, human experts evaluate a subset of the collected article-summary pairs to remove any substandard tuples. This two-step approach, involving both crowd-sourcing and expert curation, helps to ensure the quality and reliability of the dataset.

3.2.1 Source

We collect source articles from Telugu news sites using a fair usage policy and divide them into sets of 50 articles each. The original articles remain the copyright of the original authors/publishers. The TeSum dataset is released as a list of URLs and summary pairs. The collected articles are then processed to remove HTML tags, non-Telugu content, and common irrelevant phrases such as article dates and city names, ensuring that the dataset contains only relevant and high-quality content for summarization.

3.2.2 Manual Summarization

The human summarization task is posed as a crowd-sourcing activity. Each HIT (Human Intensive Task) for a creator consists of 50 news articles set created earlier. The creator is expected to create summaries for all the articles in the HIT following the guidelines given below. Each HIT submitted by the creator undergoes thorough automatic and human evaluation steps in order to ensure quality based on a criteria which maintains the essence of the task of summarization. The creator needs to ensure that:

1. Relevance: All or most of the relevant information contained in the article should be present in the summary.

2. Readability and Coherence: The summary should be coherent, readable and free of any grammatical errors.

3. Creativity: The summary should have novel sentential and phrasal structures.

The human summary creators were given the guidelines presented in Section 3.2.2.1 based on the above 3 properties.

3.2.2.1 Guidelines for Abstractive Summary Creation

Summary creators were instructed to carefully follow these guidelines and write one abstractive summary per article.

1. **Relevance and Coverage:** All the pertinent information conveyed in the source article should be captured in summary while discarding any irrelevant information. Redundant information or information unrelated to the major topic of the article may be considered irrelevant.

    - **Missing important information:** A summary has to cover all the important aspects of the original article.
    - **Including irrelevant information:** A summary should not include any irrelevant information. No personal opinion(s) or non-factual details should be included.
    - **Redundant information:** Summary should not contain any repetitive phrases/sentences.

2. **Readability:** If the summary is understandable by a native speaker without looking at the source article, it is considered "Readable". Bad grammar, pronouns that cannot be resolved within the summary, and unnatural sentential/phrasal structures would make the summary difficult to understand. Also, creators are instructed that the summary should stand as an independent article, and the reader should not need the original article to understand it fully.

    - **Disjoint sentences:** While paraphrasing, sentences should be joined in such a way that the composite sentence must be meaningful.
    - **Anaphora issue:** In summary, pronouns should be used only after the antecedent has appeared at least once.
    - **Disordering of sentences:** The summary should be coherent to convey the proper context of the original article.
    - **Not readable:** The summary should be free from any syntactic and semantic errors.

3. **Creativity:** Since this is an abstractive summarization task, we require the summaries to have novelty in terms of sentential structures such as lexical choices (vocabulary used, is other than the given article), phrasal constructions, and sentence formations.

    - **Missing novel sentence structure:** The summary should contain novel sentence structures (using some novel words) compared to the original article.
    - **Lengthy summary:** The summary should be a new shorter text that conveys the most crucial information of the original article.
    - **Sentence level summary:** The summary should not be created by just altering words/phrases in individual sentences.

This exercise resulted in a collection of 92941 article-summary pairs. The full set of guidelines with examples are available here[4].

[4] https://tinyurl.com/TesumCreationGuidelines

3.3 Corpus Curation Process

As expected with any crowd-sourced text annotation task, the summaries generated by the creators had a wide variety of errors. As shown in the guidelines, we have not instructed the creators to create their summaries within a pre-specified character or word limit. This is done to ensure that the creators do not feel restricted while writing the summaries in order to fit within a pre-specified limit. An artificial limit to the length of the summary at the creation phase might introduce unnatural structures or phrasing in the sentences of summaries. Instead, we decide to filter these generated summaries on the basis of multiple automated criteria after the summaries have already been created. The crowdsourcing activity resulted in a collection of 92941 article-summary pairs. These articles were then filtered based on the following two stages.

3.3.1 Automatic Filtering

Even with basic sanity checks at the crowdsourcing stage, we encounter a large number of errors in these submitted HITs. These article summary pairs need to be filtered out in order to maintain the high quality of the dataset.

1. *Remove Empty:* We remove any pairs where either the summary/article or both are empty.
2. *Remove Duplicates:* Duplicate pairs and duplicate summaries were removed. We do not want duplicate article-summary pairs. Two distinct articles should not have the same summary. We find it is unlikely that two distinct articles would share the same summary, therefore we also remove the pairs which share a common summary.
3. *Remove Prefixes:* We remove all prefix cases, that is any pair where the summary is just the first few sentences of the article. We should note that though MassiveSumm has claimed to follow steps $1 - 3$, we find in Table 3.2 that applying these steps to MassiveSumm, a large volume of their samples still fell in to these categories. Duplicate summaries case holds true for XL-Sum also.
4. *Remove Article Length $<$ 4 Sentences:* We removed 4452 pairs with less than 4 sentence articles.
5. *Remove Article Length $<$ 40 Tokens and / or Summary Length $<$ 10 Tokens:* Very small article lengths are not indicative of the general distribution of news article data.

3.3.2 Automatic Quality Control

- *Compression ranges:* If an article is compressed [8] too much then we loose significant amount of information from the article, which contradicts the first property of summarization that all/most of the relevant information of article must be present in the summary. Though, a summary should also result in a significant amount of reduction in the size of the article but not at the cost of relevance.

	XL-Sum	MassiveSumm	TeSum
Dataset Size	**13025**	**119282**	**92941**
Empty	2	5579	2
Duplicate Pairs	0	10456	515
Duplicate Summary	141	2698	135
Prefixes	3	30741	1330
Article $<$ 4 Sentences	10	4953	4195
Article $<$ 40 Tokens **Summary $<$ 10 Tokens**	374	5446	10
Compression $<$ 50%	10	1641	52802
Compression $>$ 80%	11920	46776	456
Abstractivity $<$ 10	0	6683	5942
Abstractivity $>$ 80	227	303	42
Human-Eval(TeSum)	-	-	7183
Final Valid	**338**	**4006**	**20329**
Valid %	**2.6%**	**3.36%**	**21.9%**
Avg Abstractivity scores	68.23	36.44	31.08
Avg Compression(%)	71.71%	73.34%	58.24%

Table 3.2: Pre-processing and Filtration Details. Here, the bottom 2 rows show the average abstractivity scores and average compression% for the final valid pairs of the 3 datasets.

	Relevance	Readability	Creativity
Score 0	0 - 10% relevant information	Not understandable	Copied verbatim from the 'original' article
Score 1	10 - 40% relevant information	Largely ungrammatical	Most of the sentences copied from the 'original article'
Score 2	40 - 60% relevant information	Approx 50% ungrammatical	Half the summary is copied from the original article
Score 3	60 - 90% relevant information	Minor grammatical error	Most of the summary is novel, but some is copied verbatim
Score 4	Everything is relevant and all the relevant information is covered	Free from any grammatical, spelling and punctuation errors.	The entire summary, except the factual information (names, dates etc.), is novel

Table 3.3: Abstractive Summarization Evaluation Criteria

Therefore, we set compression% limits to be between 50-80. While the upper limit is higher than many previous datasets, (which, usually, set this to 30%) we find, particularly in news domain which is information dense, there can be large number of examples where slightly more content is required in the summary. Figure 3.1 shows the article counts of TeSum for compression% ranges.

- *Abstractivity ranges:* We want novelty in the summary, the content should be different from the source on both sentential as well as phrasal levels. We often find that even with the best editorial practices, the content in the highlights is often a conjunction of multiple disjoint phrases or absolute copy of phrases from the article. Which apart from being non-coherent, beats the third property of creativity. Therefore, we take the measure of abstractivity from [8] and apply 10-80 range of filtration. Even though we want the summary to be abstractive, we still need to copy some n-grams from the article which corresponds to factual information (names etc.) as presented in the article. Therefore, we restrict Abstractivity at 80 which is still a fairly lenient limit.

  On the lower side, for shorter articles which are information dense, it is possible that the copied unigrams or bigrams will constitute a large chunk of the summary in order to retain facts. This is especially true in the news domain where the summary creator has to copy smaller n-grams but would change the phrasal structure for novelty (paraphrasing). If abstractivity as proposed by Bommasani and Cardie [8] goes very close to zero then we get very high degree of copying in higher n-grams also. And on the other side, if the abstractivity is very high then we loose important information in terms of verbs, nouns, etc also which need to be there.

At this stage, after filtering by compression and abstractivity, we are left with 27512 article-summary pairs. Table 3.2 shows the number of article-summary pairs getting affected by each filter.

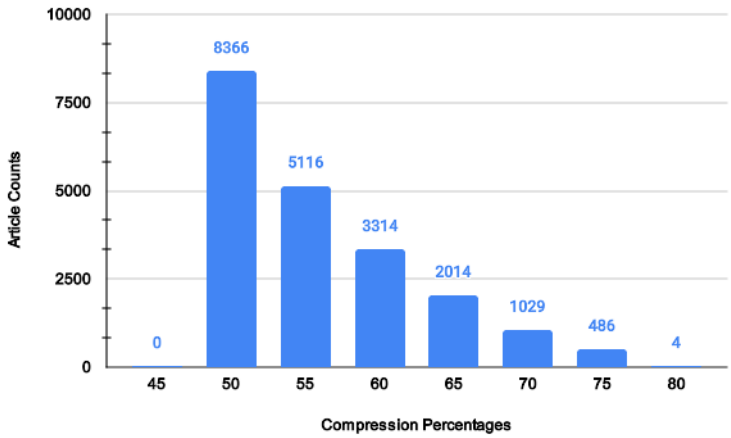

Figure 3.1: Article Count Vs. Compression

3.4 Human Evaluation

To maintain quality, one has to ensure that the human summarization guidelines are well understood by the creators and creators are, by an large, sticking to the guidelines. Though, it is impossible in any such task to have all the submissions manually evaluated, if a reasonable percentage of all the submissions are evaluated and found to be of high quality, it can be safely assumed that the rest of the submissions are also of high quality. Over the course of large number of evaluations, the expected percentage of lower quality samples in the total data can be estimated.
For human evaluation, the raters were asked to rate a minimum of 25% of the pairs from each HIT, for the 3 parameters *Relevance*, *Readability* and *Creativity* as per the Table 3.3, full set of guidelines can be accessed here[5]. Each rater is supposed to rate a sample by giving scores, ranging between 0 to 4, for each parameter.

3.4.1 Special Cases

- If all the sentences are copied verbatim from the original article, scores are [0 0 0] for Relevance, Readability, and Creativity.
- In case of syntactic errors (spelling, spacing, punctuation), if that particular word/phrase deviates the overall meaning/context significantly, then scores will be deducted in Readability as well as Relevance.
- In case of tense issues, simultaneously, the scores can be reduced in Creativity and Relevance.
- The addition of irrelevant information or outside the context of the article leads to obtaining less scores in Creativity and Relevance.

[5] `https://tinyurl.com/TeSumEvaluationGuidelines`

	Avg Scores			# Samples >= 3		
	XL-Sum	**MassiveSumm[Te]**	**TeSum**	**XL-Sum**	**MassiveSumm[Te]**	**TeSum**
Relevance	1	2	3	4	43	185
Readability	3.5	2.9	3.27	176	144	188
Creativity	0.98	1.58	3.28	12	51	170
All 3 parameters rated >= 3				**4**	**35**	**154**

Table 3.4: Human Evaluation of XL-Sum[Te], MassiveSumm[Te] and TeSum on 200 samples each.

- For anaphora-related issues, both Readability and Creativity scores will be reduced.
- Improper usage of novel words/phrases causes a reduction in Creativity score. If that particular word/phrase deviates from the original article's meaning, there will also be a reduction in the Relevance score.

3.4.2 Inter Rater Reliability:

The inter-rater-reliability was established by following the guidelines (as mentioned in section 3.4). We ramdomly extracted 500 samples from the total collected articles. These 500 samples were then rated by 3 expert raters to compute the ICC3 scores. The agreement scores were then computed using the Intra-class Correlation Coefficient (ICC) [70] following the guidelines given by Koo and Li [34]. We report **ICC3** scores, which correspond to fixed raters and individual (single) reliability. We specifically chose this model (ICC3), because each sampled article-summary pair, from the HITs, is then evaluated by one rater, and not all 3.
For our three parameters: Relevance, Readability and Creativity, our raters achieved **0.89**, **0.94** and **0.90** reliability scores respectively. These scores indicate good to excellent reliability.

3.4.3 Human Evaluation Process

Each HIT was evaluated by one rater, by randomly selecting a minimum of 25% from the HIT and distributing among the 3 raters, such that each pair of this 25% was evaluated by a single rater. If on an average the combination of these 25% pairs do not rate 3 or above for each individual parameter, then the entire HIT is rejected based on the assumption that there is a higher percentage of low quality submissions in this HIT. This process resulted in a total reduction of 7183 pairs. Giving us the final 20329 pairs.
Since we are evaluating only a percentage of the samples submitted in each HIT, we need to be aware of the possibility of some errors in the final dataset. To estimate this, we take the 5089 evaluated samples

(25% of the final 20329) and find individual samples which have lower scores. These were found to be 3.6%. As, 25% is a fair enough sample size, we can safely extend the same error percentages to the entire dataset. Therefore resulting in a dataset which, while being smaller than other existing datasets, is of high quality. But if we subject the existing datasets to the same high standards that we expect from our dataset, we find that our dataset size is not low at all, in comparison with their resultant dataset sizes.

3.5 Evaluating Existing Datasets

For comparison, when we applied the filters mentioned in Section 3.3, to the existing datasets MassiveSumm and XL-Sum, we found some surprising results. As mentioned in the Table 3.2, MassiveSumm ended up with only 3.36% of their original dataset size. Similarly, XL-Sum also reduced to only 2.6% of their original size. Even if we relax the constraints a little bit, it does not help their end results much. We will be releasing all the filtering scripts along with the lists of IDs/URLs for basic problem cases from both the datasets.

Human Evaluation of MassiveSumm and XL-Sum:
Due to surprising final numbers of the XL-Sum and MassiveSumm datasets after the filtration, we decided to validate this finding by manually evaluating randomly selected 200 article-summary pairs from each of the 3 datasets using the same raters. All samples were completely annonymized/randomized in order to avoid dataset bias. We found that the summaries are of low quality for XL-Sum and MassiveSumm on almost all parameters, Table 3.4 shows the average numbers obtained by each dataset for each parameter individually. Also, the counts of article-summary pairs from each dataset which gained 3 or above ratings are shown. The bottom line shows final count of valid pairs (out of the 200) for each dataset which were rated 3 or above, for all the 3 parameters.

Other Languages: As the numbers on the existing datasets were too surprising, we wondered if it was for this particular language. Therefore, we extended our analysis to some other languages (Hindi, Gujarati and Marathi from both XL-Sum and MassiveSumm) that we could evaluate (could read). We found similar issues in all of these datasets. We show the detailed filtration counts in Table 3.5.

3.6 Baseline Models

We present some common baselines used for summarization by other authors, to demonstrate the impact of the datasets on summarization using various models.

3.6.1 Models

To demonstrate the quality of the TeSum dataset for the task of summarization and establish baselines, we trained and tested multiple existing summarization models using the TeSum dataset. This allows us

	HINDI		MARATHI		GUJARATI	
	XL-Sum	**MassiveSumm**	**XL-Sum**	**MassiveSumm**	**XL-Sum**	**MassiveSumm**
Dataset Size	**88472**	**563477**	**13627**	**127838**	**11397**	**43830**
Empty	5	20936	1	1488	0	3797
Duplicate Pairs	4	48461	0	614	1	525
Duplicate Summary	698	5626	465	4507	59	878
Prefixes	19	4225	3	4015	6	99
Article $<$ 4 Sentences	164	27845	6	6811	104	5307
Article $<$ 40 Tokens **Summary $<$ 10 Tokens**	377	125372	154	60489	97	14303
Compression $<$ 50%	13	1990	6	145	5	163
Compression $>$ 80%	85028	286696	11985	47659	10611	18481
Abstractivity $<$ 10	4	10668	1	843	0	55
Abstractivity $>$ 80	29	643	411	128	91	11
Final Valid	**2131**	**31015**	**595**	**1139**	**423**	**211**
Valid %	**2.4%**	**5.5%**	**4.37%**	**0.89%**	**3.71%**	**0.48%**

Table 3.5: Filtration counts of XL-Sum and MassiveSumm for the other 3 languages; Hindi, Marathi and Gujarati.

to evaluate the performance and effectiveness of the dataset in comparison to different summarization models, showcasing its potential for various applications and use cases.

Pointer-Generator (PG): This model is implemented using sequence-to-sequence Recurrent Neural Networks (RNN) [73] with attention mechanism [4]. Further, we also implemented the pointer-generator[69] with coverage mechanism model. Additionally, we have implemented the pointer-generator mechanism, as proposed by See et al [73], which allows the model to determine whether to copy words from the source text or generate from the vocabulary. This helps address the issue of Out Of Vocabulary (OOV) words effectively. Furthermore, we have also incorporated a coverage mechanism to prevent the model from attending to the same phrases multiple times, which mitigates redundancy in summary generation.

MLE+RL, with intra-attention: This model utilizes the intra-attention mechanism proposed by Paulus [60] to attend to the input document and generate decoder output independently, mitigating the issue of repetitive and incoherent phrases in the summaries. Additionally, the model incorporates a novel training approach that combines supervised and Reinforcement Learning (RL) to overcome exposure bias problems and generate readable summaries.

Text summarization with Pretrained Encoders (BertSumAbs): This model is based on the novel document-level encoder by Liu and Lapata [42] which uses Bidirectional Encoder Representations from Transformers (BERT). For abstractive summarization, this method adopts the encoder-decoder architecture with a new fine-tuning approach where the encoder is a pre-trained BERT and the decoder is a randomly initialized Transformer. For this model, we have used the embeddings by Marreddy et al. [45] (trained on 8M+ Telugu sentences).

mT5: We fine-tuned the Multi-lingual Text To Text Transfer Transformer (mT5) model by Xue et al. [82] on TeSum dataset. The mT5 model is a multi-lingual variant of the T5 model [64] that has been trained on a large English dataset from common crawl. Specifically, we used the mT5-small variant for our experiments.

3.6.2 Experimental Setup

To create train, dev and test splits of TeSum dataset, we divide the total 20329 pairs into carefully selected sub-parts of about 80%, 10% and 10% respectively. The selection of pairs is done in a way that preserves the balance in terms of length of articles, compression(%) and abstractivity levels across the three splits. Table 3.1 details the statistics for the 3 splits.

For the experiments and baseline training, we have used Word2Vec [47] (Telugu Wikipedia pre-trained) embeddings. Apart from mT5, which was fine-tunned using 2 GPUs and 20 CPUs, the rest had system config of 1 GPU and 10 CPUs. Further details on hyper-parameter settings and configuration is listed in Table 3.6. Here, 'PG+' represents PG and PG+Coverage models, and 'MLE+' represents MLE, MLE+RL and RL models.

Necessary Concessions: As, after our filtration steps, the originally large-scale existing-datasets ended up with a very low percentage of their total article-summary pairs. Which extrinsically does not make for a fair comparison. Therefore, before going ahead with the model training and experiments, for evaluating

Parameters	PG+	MLE+	BertSumAbs	mT5
Max source length	400	400	512	512
Max target length	100	100	200	256
Min target length	35	35	50	30
Batch Size	8	8	140	2
Epochs/Iterations	100k iter	100k iter	50k iter	10 epochs
Vocab Size	50k	50k	28996	250112
Beam Size	4	4	5	4
Learning Rate	0.15	0.001 (MLE) 0.0001 Others	lr_bert = 0.002 lr_dec = 0.2	5e-4

Table 3.6: Experimental setup and parameter settings

the effect of these curations of the datasets for the task of summarization, we are forced to make some concessions for XL-Sum and MassiveSumm.

As a concession for MassiveSumm, we decided to concede compression from 80% to 90% and we find that it added a fairly high number of articles to the valid set for MassiveSumm (giving us a total of 17248 pairs, which we then divide into about 80%-10%-10% to get the train, dev and test splits). Relaxing the compression further would increase the numbers, but we also note that the authors themselves have presented their results on a randomly selected 12633 pairs (not made available by the author), therefore we take a comparative number, which according to us should be of a better quality due to the aggressive quality control.

For XL-Sum, the only option was to remove all constraints, as the original size itself was quite small. Therefore, we considered the original splits of XL-Sum (Telugu) for our experiments.

3.7 Results and Analysis

For better comparison, experiments were conducted by training on each dataset's training split and then testing on all 3 datasets' test set. Table 3.7 shows the ROUGE scores[6] for some of the selected best

[6]Multilingual Rouge from XL-Sum `https://tinyurl.com/MLTERouge`

Trained on TeSum

	Tested on TeSum			Tested on MassiveSumm			Tested on XL-Sum		
Model	**T-R1**	**T-R2**	**T-RL**	**M-R1**	**M-R2**	**M-RL**	**XL-R1**	**XL-R2**	**XL-RL**
Pointer Generator	**39.37**	**22.72**	**32.15**	25	**13.8**	20.74	**9.73**	**2.29**	**7.32**
MLE + RL- wo	38.09	21.9	31.77	**25.05**	13.41	**20.78**	8.56	2.03	6.54
RL- wo	31.19	17.6	24.86	20.16	10.58	16.75	8.28	1.96	6.45
BertSumAbs	26.49	12.55	19.6	18.61	8.24	14.69	6.21	1.34	4.96
mT5-small	37.42	20.82	30.88	24.37	12.5	20.2	8.8	2.06	6.7

Trained on MassiveSumm

	Tested on TeSum			Tested on MassiveSumm			Tested on XL-Sum		
Model	**T-R1**	**T-R2**	**T-RL**	**M-R1**	**M-R2**	**M-RL**	**XL-R1**	**XL-R2**	**XL-RL**
Pointer Generator	26.31	14.45	22.16	28.38	16.33	24.95	**9.85**	**2.18**	**8.34**
MLE + RL- wo	26.3	14.62	22.36	**30.46**	**18.69**	**27.17**	9.82	2.03	8.23
RL- wo	15.59	7.93	13.75	13.82	7.76	12.82	4.27	0.89	3.88
BertSumAbs	**29.73**	13.59	22.02	23.76	11.47	19.28	7.69	1.54	6.11
mT5-small	27.67	**15.08**	**23.03**	29.43	17.41	26.25	9.67	1.91	8.14

Trained on XL-Sum

	Tested on TeSum			Tested on MassiveSumm			Tested on XL-Sum		
Model	**T-R1**	**T-R2**	**T-RL**	**M-R1**	**M-R2**	**M-RL**	**XL-R1**	**XL-R2**	**XL-RL**
Pointer Generator	17.13	2.2	10.06	12.45	1.59	8.73	5.41	0.28	4.35
MLE + RL- wo	3.7	0.44	3.18	2.88	0.43	2.63	1.17	0.04	1.13
RL- wo	2.4	0.28	2.14	1.61	0.17	1.49	0.68	0.04	0.66
BertSumAbs	13.8	2.41	10.26	11.46	2.06	9.19	6.55	1.44	5.73
mT5-small	**18.42**	**9**	**15.88**	**19.44**	**9.12**	**17.46**	**12.24**	**3.6**	**11.18**

Table 3.7: ROUGE scores achieved by various baseline models.

performing model configurations. Here, 'wo' with MLE+RL and RL models stands for 'without intra attention', and 'Pointer Generator' represents the PG+Coverage model.
Looking at this table our first observation is that models trained on TeSum end up performing well across the board, but do not end up beating models trained and tested on the same dataset for almost all models. We surmise that this is because the fundamental nature of these summarization datasets is different. While MassiveSumm and XL-Sum summaries are primarily small number of disjoint sentences, TeSum summaries are coherent discourses in themselves. This means that a model trained to avoid copying and trained to generate coherent discourse would fail on MassiveSumm and XL-Sum.
While we accept the contributions made by XL-Sum and MassiveSumm, which bring value to this field for any given language, we claim that this scraping and the initial pre-processing is just the first step. The data need to be held to higher standards. Even if it is achieved by scraping, filtering and then evaluating a percentage of randomly selected samples of the resultant, it would ensure a much more valuable dataset than just scraping.

3.8 Conclusion

Dataset creation for any task is an expensive and complex activity. With the increased demand for data for deep-learning models, it is often infeasible to create datasets which reach the desired sample counts. It then does make sense to make do with data collected "from the wild". It is our opinion that such collected data, while useful, should also be subjected to quality control. At the same time, we should adopt pipelines which can establish a balance between quality control and cost. This is especially critical for Low Resource Languages which need to make do with low sample numbers.
To this effect, we constructed a high quality Human-curated Abstractive summarization dataset for Telugu. We also compared the dataset properties with existing Telugu summarization datasets and claim that these existing datasets can also benefit from the quality control measures that we have proposed.
Though, purely on the basis of size, our work also started with a huge collection of 92k+ article-summary pairs like the existing datasets, but by making use of human expertise at both annotation and quality assessment stages, we show that after applying the same quality measures our dataset performs significantly better then the automated ones. And as a result we out-perform the other datasets in terms of final size as well.

Chapter 4

Multilingual Summarization

This chapter demonstrates the need of multi-lingual summarization systems. Formally, multilingual summarization deals with more than one languages documents and corresponding documents. We presented the benchmark results for two Indian languages Hindi, Gujarati and Indian English. We explored various pretrained sequence-to-sequence models to perform the multi-lingual summarization task.

4.1 Recent Advancements in Multilingual Text Summarization

Automatic text summarization is a technique for obtaining a condensed version of a long document while retaining its relevance. The NLP community has become more interested in text summarization for Indian languages in recent years. The progress of text summarization has, however, been hindered due to the lack of high-quality datasets. Nevertheless, the availability of large-scale multilingual datasets such as XL-Sum[27] and MassiveSumm[79] have led to substantial progress in natural language generation and summarization tasks. Even though quality-wise, these datasets are far from perfect[78], they do serve as a good starting point in terms of quantity. Additionally, recent advancements in neural-based pretrained models have transformed the field significantly.

We have performed the multi-lingual summarization task as a part of ILSUM shared task. The goal of the shared task is to create reusable corpora for Indian language summarization. The dataset is created by scraping the news articles and corresponding descriptions from publicly available news websites. ILSUM data[68] consists of a summarization corpus for two major Indian languages- Hindi and Gujarati, along with Indian English.

This chapter provides a comprehensive overview of the existing sequence-to-sequence models for Indian language and English summarization. For Hindi and Gujarati, we used multilingual models such as MT5[82], mBART[41] and IndicBART[16] variants. We fine-tuned the PEGASUS[87], BART[38], T5[64] and ProphetNet[63] models on English data. Out of all the models, for English, PEGASUS outperformed others, while for Hindi, MT5 gave us the best results, and for Gujarati, MBart performed the best. In order to avoid overfitting, we have performed k-fold cross-validation on the training dataset.

	English		Hindi		Gujarati	
#Pairs	12564		7957		8457	
	Text	Summary	Text	Summary	Text	Summary
#Avg Words	595	36.24	553	40.17	414.43	32.26
(Min, Max) Words	(1, 5717)	(1, 113)	(17, 5034)	(6, 113)	(25, 2839)	(1, 408)
#Avg Sentences	10.29	1.26	18.1	1.7	21.28	1.57
(Min, Max) Sentences	(1, 169)	(1, 17)	(1, 157)	(1, 9)	(1, 187)	(1, 46)

Table 4.1: ILSUM Train Data Statistics

We have observed that Hindi k-fold experiments had better scores than the experiments performed with the full version of the released data. We have applied several filters to assess the quality of the released datasets. Various combinations of filtered and original data were used to determine the efficacy of the pretrained generation models. We talk about our models, experiments and dataset filters later in this chapter.

4.2 Corpus Description

The dataset released for this task has been collected from several leading Indian newspaper websites. The English and Hindi datasets were scraped from indiatvnews[1], and the Gujarati data was created by scraping the divyabhaskar[2] and gujarati.news18[3] websites. The Hindi and Gujarati datasets include articles/summaries which contain English words or phrases which have been code-mixed and script-mixed. Note that we have observed a few samples of English and Gujarati datasets, where the summaries consists of only one word. The ILSUM training data statistics are mentioned in Table 5.3. We have used the Indic[36] tokenizer to generate the counts in Table 4.1.

4.3 Model Description

The pretrained language models (PLMs) used for downstream tasks are pretrained using massive amounts of unlabeled text data. A PLM encodes extensive linguistic knowledge into a vast amount of parameters[1], which stimulates universal representations and improves generation quality. We have experimented with various pretrained generation models to find the optimal architecture.

T5 [64] model proposes defining every NLP task in a text-to-text format. The model consists of an

[1] https://www.indiatvnews.com/

[2] https://www.divyabhaskar.co.in/

[3] https://gujarati.news18.com/

			Validation Scores		
Lang	**Model**	**Full Data / k-fold**	**R-1**	**R-2**	**R-4**
English	PEGASUS	Full Data	**56.85**	**45.92**	**43.36**
	$T5_{large}$	Full Data	56.05	45.03	42.36
	$BART_{large}$	k-fold	54.83	43.58	40.71
	PEGASUS xsum	Full Data	54.66	43.48	40.64
	BRIO	Full Data	53.57	41.86	38.81
	$BART_{large}$ xsum	k-fold	53.35	41.74	38.75
	$T5_{base}$ + Adapter	k-fold	51.91	40.07	37.1
	ProphetNet	k-fold	49.51	36.98	33.83
Hindi	IndicBART	k-fold	**60.73**	**51.26**	**47.57**
	$MT5_{base}$	k-fold	60.04	50.72	46.82
	$MT5_{base}$*	Full Data	58.65	49.09	45.08
	IndicBART-SentSumm	k-fold	58.09	47.99	43.72
	$MBart_{large}$50 + Adapters	Full Data	56.26	45.56	41.21
	$MBart_{large}$50	Full Data	55.76	44.96	40.59
Gujarati	$MBart_{large}$50	Full Data	**26.20**	**16.44**	**12.16**
	$MT5_{base}$	Full Data	25.11	15.81	11.68
	$MT5_{base}$*	Full Data	24.16	14.68	10.79
	IndicBART	k-fold	23.38	13.34	9.35
	$MBart_{large}$50 + Adapter	Full Data	21.63	13.04	9.56

Table 4.2: ILSUM Experiments on Validation Data. *Finetuned on the combination of Hindi and Gujarati Data

encoder-decoder Transformer architecture finetuned on the C4 corpus. In our experiments, we use both the T5-Base (220M parameters) and T5-Large (770M parameters) versions of the model. Since T5 is trained on an English-only dataset, we also look at the multilingual variants of the model for our experiments in Hindi and Gujarati. The MT5 model[82] uses an architecture very similar to T5, and is trained on 101 languages, as described in the mC4 dataset. Owing to the large size of the models, we only finetuned the base version (580M parameters) of the MT5 model (the large version has 1.2B parameters).

BART [38] is a denoising autoencoder for pretraining seq2seq models, which is similar to both BERT and GPT. Since it uses a bidirectional encoder like BERT, and an autoregressive decoder like GPT. The model was trained by corrupting the text using a noising function, and reconstructing the original text. We experiment with the BART-large model (406M parameters), and then also try out versions of the BART model finetuned on different datasets, namely the BART-Large-CNN and BART-Large-XSUM model, finetuned on the CNN-Daily Mail and XSUM datasets respectively. We try out multilingual variants[41] of the BART model for Hindi and Gujarati summarization experiments, namely the MBart-Large-50 (610M parameters) model[74], trained on 50 languages.

PEGASUS [87] uses the extracted gap sentences (GSG) self-supervised objective strategy to train the encoder-decoder model. Rather than masking a smaller text span as in BART and T5, PEGASUS masks the entire sentence. Later, it concatenates the gap sentences into pseudo summaries. It chooses the sentences based on importance. In the same way as T5, PEGASUS does not reconstruct full sequence of inputs but only masked sentences. The pretraining is performed with C4[64] and HugeNews corpus. We finetune the PEGASUS-large model on the ILSUM English corpus.

BRIO [43] is a novel training paradigm to achieve neural abstractive summarization, wherein a contrastive learning component is introduced to reinforce the abstractive model's ability to estimate the probability of system-generated summaries more precisely instead of using MLE training alone. Two stages are involved in this approach: the first stage generates the candidates using a pretrained sequence-to-sequence model, and next stage selects the best one.

ProphetNet [63] introduced a novel self-supervised objective, wherein the goal is to predict the next-n tokens, instead of just optimizing for one-step ahead predictions. We experiment with ProphetNet in our English summarization experiments.

IndicBART[16] is a pretrained sequence-to-sequence model trained on 11 Indic languages and English. It follows the masked span reconstruction objective similar to MBart. In contrast to available generation models, IndicBART utilizes the orthographic similarity between the Indian languages to achieve better cross-lingual transfer learning capabilities. This model size (244M) is much smaller than MBart and MT5 models with compact vocabulary. We finetune the IndicBART model on Hindi and Gujarati datasets.

Adapters: Recently proposed lightweight adapters[61] are effective at mitigating the overhead of pretrained language models for downstream tasks. We can update the adapters during finetuning and freezing most of the PLM parameters. In recent work[88], adapters were applied to perform Gujarati text summarization. Adapters can not only speed up training time but are also storage efficient since they require saving only adapter weights instead of entire finetuned model weights.

			Test Scores		
Lang	**Model**	**Full Data / k-fold**	**R-1**	**R-2**	**R-4**
English	PEGASUS	Full Data	**55.83**	**44.58**	**41.8**
	$T5_{large}$	Full Data	54.73	43.08	40.12
Hindi	$MT5_{base}$	k-fold	**60.72**	**51.02**	**47.11**
	IndicBART	k-fold	58.38	48.31	44.25
Gujarati	$MBart_{large}50$	Full Data	**26.11**	16.51	12.41
	$MBart_{large}50$	Full Data (dropout=0.2)	26.07	**16.60**	**12.58**

Table 4.3: ILSUM scores on Test Data

4.4 Experiments and Results

We have performed experiments under two different settings: the first is with the entire released dataset (full data), and the other is where we split the dataset into 10 folds and utilize 90% data (9 folds) for training and 10% data (1 fold) for validation. In both settings, the released data in the validation phase was used for testing purposes and we report these results in Table 4.2. Note that doing such k-fold cross validation experiments were also essential to evaluate our models' performance because validation summaries were not provided to us.

We use the standard ROUGE metric[40] to compute all the scores. We observed that PEGASUS yields the best results for English when finetuned on the full data version in the validation phase. We achieved the best results when we finetuned IndicBART and MBart using k-fold and full data during the validation phase. Finetuning a model on k-fold data might sometimes lead to better results than finetuning it on the entire dataset, which indicates that the dataset needs to be studied more and appropriate filters need to be applied, to see which examples in the dataset contribute to the model learning something useful. We discuss this in the next section.

Based on the results of the validation phase, we submit results from the best models in the test phase. While PEGASUS and MBart still give us the best results for English and Gujarati respectively, MT5 performs better than IndicBART for Hindi when finetuned on k-fold data. Hyper-parameter settings are listed in Table 4.4.

The multilingual models have been pretrained on large amounts of data, and they are sufficiently capable of handling the presence of code-mixing in the dataset, which we observe in the outputs as well. The models generate good summaries and can add relevant English text in Hindi and Gujarati examples where appropriate. For instance, the average number of English words in Hindi and Gujarati training summaries is 0.25 and 1.91 respectively. For the test set released for Hindi and Gujarati, the summaries generated by our models have an average of 0.23 and 1.44 English words per summary. Note that the average number

Parameters	BART	T5	ProphetNet	PEGASUS	BRIO	MBart	MT5	IndicBART
Max source length	512	512	512	512	512	512	512	512
Max target length	75	75	75	75	75	75	100	75
Batch Size	2	1	1	2	2	4	2	2
Epochs	5	5	5	5	5	5	10	10
Vocab Size	50265	32128	30522	96103	50264	250054	250112	64015
Beam Size	4	4	5	4	4	4	4	4
Learning Rate	5e-5	5e-5	5e-5	5e-4	5e-5	5e-5	5e-5	5e-5

Table 4.4: Experimental setup and parameters settings

of English words in Hindi summaries is less because a large number of training samples are purely in Hindi and do not contain any English words or characters. As shown in Table 4.2 (indicated with *), we have trained our models with the combination of Hindi and Gujarati languages data, we have obtained the comparable scores with respect to the best performing models.

4.5 Data Quality Assessment

To verify the quality of the data, we have applied some of the filters mentioned in TeSum[78]. Filters were applied include checking whether there are:

1. Empty instances
2. Duplicate pairs and summaries within the dataset
3. Cases where the first few sentences of the article itself are taken as the summary
4. Check whether the summary is 'compressed enough', i.e., we should not have summaries comparable in size to the text that has to be summarized. Compression is a good measure of telling us if the summary provided is a shortened version of the input document/text or not.

Filters counts for all the languages can be found in Table 4.5. It is important to note that, based on our filters, only about 68% of the Hindi summaries are valid since many are simply the first few sentences of the article. It could also be one of the reasons for models giving better results on k-fold data. Some of the folds in the training data might contain a large percentage of high-quality, valid summaries while leaving out a significant number of summaries which we consider invalid. Note that for Gujarati and English, the number of final valid article-summary pairs is comparable to the original dataset size, which is why the top-performing models give better results when finetuned on the whole dataset as compared to k-fold subsets.

Filters	Hindi	Gujarati	English
Dataset Size	7957	8457	12565
Empty	0	0	1
Duplicate Pairs	23	0	0
Duplicate Summary	15	113	117
Prefixes	2518	135	486
Compression <50%	11	37	182
Final Valid	5390	8172	11779
Valid %	**67.74%**	**96.63%**	**93.74%**

Table 4.5: Filtration counts of ILSUM data

4.5.1 Data Variation Experiments

The unavailability of large datasets is one of the main bottlenecks for neural models for text generation. The existing summarization datasets for Indian languages are quite small. To improve the model generation capabilities on limited dataset, we did k-fold cross-validation on the best performing models (see Table 4.2). The mean ROUGE scores and standard deviation scores over 10 runs are reported in Table 4.6. We did 10-fold cross-validation using the released training dataset with the following combinations:

1. **Original data:** Fine-tuned for 5 epochs with released training dataset

2. **Original + Filtered data:** Finetuned for 3 epochs with original + 2 epochs with Filtered data

3. **Filtered data:** Fine-tuned for 5 epochs with only filtered dataset

4. **Filtered + Original data:** Finetuned for 3 epochs with filtered data + 2 epochs with original data

To perform all the experiments, we used the 'filtered data' obtained after applying filters mentioned in Table 4.5. To compare the models' performance on different variations of the training dataset, we have not made any changes in the validation data. As observed in Table 4.6, the experiments performed with 'original' data produce better scores than the 'filtered' data. Also, the models finetuned on the combination of the 'filtered + original' dat aset performed better compared to the 'original+filtered' combination.

Lang	Model	Data composition	R-1	R-2	R-L
English	PEGASUS	Original Data	52.51 ± 1.1	40.91 ± 1.36	47.81 ± 1.16
		Original + Filtered Data	51.65 ± 1.14	40.07 ± 1.25	46 ± 3.67
		Filtered Data	51.88 ± 1.25	40.37 ± 1.39	47.32 ± 1.31
		Filtered + Original Data	53.28 ± 1.18	41.82 ± 1.3	48.67 ± 1.2
	T5-large	Original Data	**53.45 ± 0.95**	**42.16 ± 1.13**	**48.97 ± 1.05**
		Original + Filtered Data	53.22 ± 1.23	42.04 ± 1.41	48.85 ± 1.31
		Filtered Data	51.9 ± 1.37	40.49 ± 1.53	47.38 ± 1.46
		Filtered + Original Data	53.33 ± 0.83	42.1 ± 0.96	48.92 ± 0.86
	BART-large	Original Data	50.25 ± 1.52	38.15 ± 1.85	45.46 ± 1.63
		Original + Filtered Data	51.42 ± 0.88	39.85 ± 1.11	46.93 ± 1
		Filtered Data	51.21 ± 1.3	39.83 ± 1.57	46.79 ± 1.38
		Filtered + Original Data	52.45 ± 1.05	40.98 ± 1.29	48 ± 1.17
Hindi	IndicBART	Original Data	26.36 ± 1.02	12.66 ± 0.73	26.28 ± 0.98
		Original + Filtered Data	21.58 ± 0.66	9.84 ± 0.76	21.45 ± 0.6
		Filtered Data	21.27 ± 0.88	9.75 ± 0.56	21.12 ± 0.86
		Filtered + Original Data	25.67 ± 1.04	12.16 ± 0.82	25.57 ± 1
	MT5-base	Original Data	**27.04 ± 1.22**	**13.21 ± 0.61**	**26.96 ± 1.22**
		Original + Filtered Data	20.33 ± 0.91	9.26 ± 0.8	20.2 ± 0.92
		Filtered Data	20.61 ± 1.55	9.47 ± 0.67	20.51 ± 1.53
		Filtered + Original Data	26.73 ± 1.11	12.83 ± 0.61	26.64 ± 1.1
Gujarati	MBart Large 50	Original Data	20.36 ± 0.67	11.65 ± 1.13	20.01 ± 0.72
		Original + Filtered Data	16.04 ± 1.12	9.23 ± 0.76	15.83 ± 1.15
		Filtered Data	12.82 ± 2.28	6.6 ± 1.54	12.38 ± 2.36
		Filtered + Original Data	19.55 ± 0.74	11.42 ± 0.43	19.2 ± 0.72
	MT5-base	Original Data	**21.55 ± 0.77**	**11.81 ± 0.78**	**21.19 ± 0.83**
		Original + Filtered Data	18.63 ± 0.93	9.23 ± 0.5	18.19 ± 0.92
		Filtered Data	9.66 ± 0.97	4.84 ± 0.56	9.53 ± 0.92
		Filtered + Original Data	20.29 ± 0.62	10.7 ± 0.52	19.84 ± 0.56

Table 4.6: Validation set ROUGE scores on ILSUM corpus. This table reports the mean ROUGE scores and its standard deviation over 10 runs

4.6 Discussion and Conclusions

While having better models finetuned exclusively on Indian languages might benefit research in the area of Indian Language Summarization, creating larger, high-quality datasets for such languages will surely lead to progress in this field. It might be interesting to look at sources other than news websites as well, and to keep in mind the filters discussed earlier while creating the dataset. For the ILSUM task, PEGASUS, MT5 and MBart give us the best results for English, Hindi and Gujarati respectively. We conclude that the transformer-based pretrained seq2seq models are capable of generating high-quality summaries for the ILSUM dataset.

Chapter 5

Cross-lingual Summarization

This chapter demonstrates the need for cross-lingual summarization datasets and provides a novel dataset titled as 'PMIndiaSum'. Formally cross-lingual summarization accepts the source and target in different languages. The development of cross-lingual summarization datasets is an essential strategy in addressing the dearth of resources in low-resource languages, and it serves as a means to enhance natural language processing tasks. However, it can be challenging due to the lack of parallel documents or the reliability of translations in multilingual websites. In this work, we utilized the preprocessing and filtration strategies mentioned in the chapter 3 to maintain the dataset quality. PMIndiaSum supports monolingual, multilingual, and cross-lingual summarization tasks.

5.1 Introduction

In recent years, natural language processing (NLP) has witnessed remarkable advancements in the development of large-scale datasets for various NLP tasks. However, such progress is often limited to high-resource languages, with low-resource languages remaining under-represented. This is particularly true for Indian languages, which are low-resource and have been historically overlooked in NLP research [35].

Summarization is the task of generating a short description from a longer text. In the context of Indian languages, the availability of large-scale summarization datasets has been limited in terms of language coverage and data size. Furthermore, the quality of existing datasets has been criticized [78], which accentuates the need for high-quality and diverse datasets for Indian languages.

To address these limitations, we introduce PMIndiaSum, a new summarization dataset for Indian languages[1] that is cross-lingual and massively parallel. The data is extracted from an Indian governmental website: the Prime Minister of India[2]. The website publishes a diverse range of government news articles, usually available in multiple languages and covering the same content.

[1]These are the commonly seen languages in India and nearby countries, but they belong to several language families and are widely spread around the globe. Our research impact is not limited to a single country geographically.

[2]https://www.pmindia.gov.in

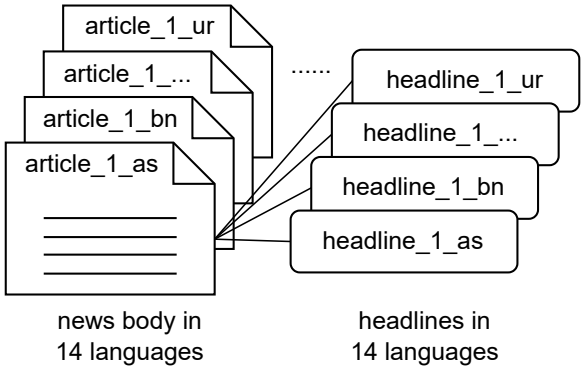

Figure 5.1: Sourced from massively parallel articles, PMIndiaSum supports 14 languages and 196 language pairs.

The development of cross-lingual summarization datasets is crucial in addressing the lack of resources in low-resource languages. However, it can be challenging due to the lack of parallel documents or the reliability of translations in multilingual websites. To overcome these challenges, we leverage the parallel nature of the Prime Minister of India's website, which allows us to gather massive amounts of parallel documents across various languages and language families.

The construction of our dataset follows the convention of using article headline-body pairs as summarization data [54, 12, 35]. The dataset contains document-summary pairs, where the body of the news article is used as the document, and the headline serves as the summary. The massive parallelism between articles in different languages enables us to construct many-to-many parallelism across different languages, as illustrated in Figure 5.1.

Our dataset supports the largest number of Indian languages for monolingual, multilingual, and cross-lingual tasks compared to existing datasets [79, 6]. Table 5.1 lists all supported languages. Specifically, the data contains Manipuri, an often overlooked language in Indian NLP research. Alongside the dataset, we also provide benchmarks covering methodologies like fine-tuning pre-trained language models and summarization-and-translation, which can serve as a reference for future research.

This chapter gives a comprehensive account of approaches to addressing the concerns raised by [71] with regard to developing a summarization dataset for Indian languages. Explicitly, we 1) ensure transparency in data acquisition and processing; 2) carry out comprehensive quality assurance; 3) currently offer coverage for the largest number of languages and language families; 4) provide reasonably sized data for each language pair to facilitate system building. We anticipate our work to aid in research progress in Indian language summarization and dataset construction. We release our corpus under the CC-BY-4.0 licence.[3]

[3]https://github.com/ashokurlana/PMIndiaSum

Family	Language
Dravidian	Kannada (kn), Malayalam (ml), Tamil (ta), Telugu (te)
Indo-Aryan	Assamese (as), Bengali (bn), Gujarati (gu), Hindi (hi), Marathi (mr), Odia (or), Punjabi (pa), Urdu (ur)
Indo-European	English (en)
Tibeto-Burman	Manipuri (mni)

Table 5.1: PMIndiaSum covers 14 widely used languages in India from 4 language families.

5.2 Preparation of PMIndiaSum

In this section, we describe our workflow to build a reliable summarization dataset for languages in India, from an official Indian government website: the Prime Minister's Office of India[4]. The website publishes governmental news articles consisting of a headline and a body, which enables us to follow the convention of using headline-body pairs to construct summary-document data. Additionally, articles are usually available in the most widely used languages in India, across different language families, covering the same topic and content. This makes the website an ideal source of a diverse cross-lingual dataset.

5.2.1 Data acquisition

Upon inspection, an article on the PMIndia website is often available in multiple languages, with the default being English; articles come with a language indicator in the HTML structure. We first accumulate headline-body pairs, in all available languages for all articles. These come from two sources:

1. We gathered readily crawled headline-body pairs from the PMIndia parallel corpus release [25], which belong to articles published between 2014 and 2019.

2. We then newly crawled the website for more articles up to early 2023, and extracted documents and corresponding headlines from the HTML data. We used PMIndia's crawler[5] in our workflow.

In total, we harvested 94,036 headline-body pairs for all languages, which we preliminarily regard as monolingual summary-document pairs. Figure 5.2 illustrates the distribution of English articles across years and sources.

[4] https://www.pmindia.gov.in
[5] https://github.com/bhaddow/pmindia-crawler

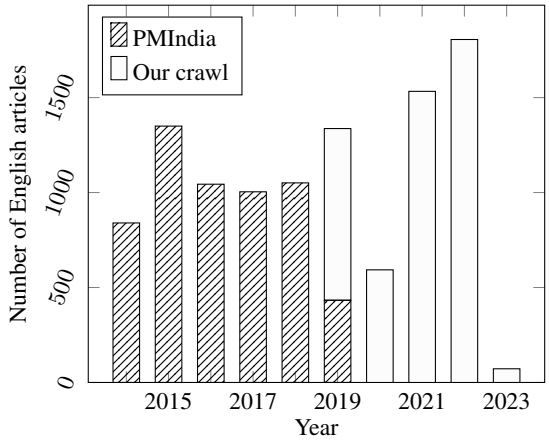

Figure 5.2: PMIndiaSum acquisition statistics for English articles, where 56% are from PMIndia and 44% are newly crawled.

5.2.2 Data processing

To create a high-quality summarization dataset, the collected headline-body pairs undergo rule-based processing in the context of monolingual summarization. We describe the rules, and list them in Table 5.2, detailing the number of invalid samples filtered at each step. Some of the steps follow [78]'s work. The final monolingual data contains 76,680 samples, 81.5% of the raw size.

Language mismatch. While we place high confidence in the Indian governmental website's effort to ensure language correctness, we still made our language filtering: we remove an entire document-summary pair if either the document or the summary contains text outside of its designated language's Unicode.[6] Code-mixed samples are also eliminated as a side effect.

Duplicates and empty. To maintain a dataset containing solely unique document-summary pairs, we remove duplicate document-summary pairs. Next, samples with identical summaries are all eliminated since we deem it improbable for two different documents to have the same summary. Moreover, pairs with either an empty document or an empty summary are discarded.

Prefix. To ensure that PMIndiaSum is abstractive in nature, we remove all samples where the summary is repeated as the first, or first few sentences in the document.

Length. Documents or summaries that are exceedingly short may not be representative of the general distribution of data. We opt to filter out any pairs if the document contains less than two sentences or the summary contains less than three tokens. Technically, for both word and sentence segmentation, we used the `indic_nlp_library` [7].

[6] Defined under the South and Central Asia-I languages on `https://unicode.org/versions/Unicode13.0.0/`

[7] `https://github.com/anoopkunchukuttan/indic_nlp_library`

	as	bn	gu	hi	kn	ml	mni	mr	or	pa	ta	te	ur	en
raw size	2925	9250	7393	7642	5743	5106	5432	6034	4760	7315	6314	6656	5622	13844
- language mismatch	68	3418	302	80	361	344	271	93	40	2367	68	422	1144	5288
- duplicate pairs	4	11	11	5	16	7	1	12	8	11	23	8	6	2
- duplicate summaries	8	28	34	110	38	25	14	54	26	53	54	34	24	202
- empty	0	1	0	0	0	0	2	0	1	0	0	0	0	0
- prefixes	33	37	28	49	13	52	92	28	13	30	73	46	133	169
- doc <2 sentences	723	191	216	35	5	12	113	30	21	40	13	19	0	19
- sum <3 tokens	0	7	0	0	3	1	3	1	0	0	4	1	0	4
final (monolingual)	2089	5557	6802	7363	5307	4665	4936	5816	4651	4814	6079	6126	4315	8160

Table 5.2: Data preprocessing and filtering statistics for document-summary pairs in each language.

5.2.3 Monolingual statistics

We present the statistics of our PMIndiaSum corpus for qualitative inspection in Table 5.3. We utilize token-based metrics as suggested by [24] and [55]. As most metrics are not well-defined for cross-lingual data, we report monolingual statistics.

Vocabulary. We report vocabulary as the number of unique tokens for both documents and summaries in each language, under the same tokenization introduced earlier. These counts imply the morphological richness of the languages covered.

Length and compression. The average document length in PMIndiaSum is 27 sentences with 518 tokens, with the maximum being 719 tokens and the minimum being 344 tokens. On the other hand, summaries are on average 12 tokens, with nearly all being a single sentence. We then compute the compression [8] rate which measures how concise a summary S is compared to its document D as $1 - \frac{len(S)}{len(D)}$. We report the average compression across all data samples; a high compression of around 90% implies an extreme summarization task.

Density. [24] introduced extractive fragments $\mathcal{F}(D, S)$, which reflect the degree to which a summary is composed of text taken from the document. Formally, density is defined as $\frac{1}{|S|} \sum_{f \in \mathcal{F}(D,S)} |f|^2$, measuring the average length of extractive fragments to which each word in the summary belongs. It has been observed that languages such as hi, pa, ur, and mni tend to have higher density scores, while other languages tend to have lower scores. This suggests that the words in the summary may not necessarily come from the same phrases as those in the document.

Novelty. We show the novelty of summaries by computing the percentage of n-grams present in a summary but not in its document. We report average novelty scores across all samples in each language for 1-4 n-grams.

Redundancy. The level of redundancy in a summary can be determined by the percentage of repetitive n-grams [27]. We present the average redundancy for unigram and bigram, and the low scores are desirable.

	as	bn	gu	hi	kn	ml	mni	mr	or	pa	ta	te	ur	en
doc vocab	73k	130k	125k	94k	166k	211k	143k	147k	98k	80k	150k	202k	61k	68k
sum vocab	6k	10k	14k	10k	12k	12k	12k	12k	9k	9k	13k	16k	8k	11k
word/doc	522	517	505	719	405	336	489	510	544	688	344	430	754	498
sent/doc	24.7	31.2	26.0	31.0	27.0	26.0	27.1	30.2	31.4	28.9	23.8	26.7	34.0	22.2
word/sum	11.9	11.5	12.3	15.4	12.3	10.1	13.5	11.4	10.8	18	11.7	14.1	17.7	13.4
sent/sum	1	1	1	1	1	1	1	1	1	1	1	1	1	1
compression	91.5	92.1	89.7	91.3	91.0	90.8	90.4	90.9	91.7	90.1	90.5	90.5	91.6	91.6
density	6.06	3.91	5.45	8.67	4.86	3.98	7.97	4.50	3.74	8.12	5.35	4.75	8.13	7.31
novelty, n=1	27.9	30.0	27.4	12.7	28.5	30.7	22.4	29.5	29.2	15.2	24.7	26.9	13.1	22.9
novelty, n=2	52.5	59.8	53.5	38.7	58.4	59.2	44.9	59.0	60.8	41.3	52.9	55.8	39.7	44.9
novelty, n=3	63.5	73.3	65.9	54.1	70.9	72.2	59.8	71.1	73.5	57.6	66.1	71.9	56.3	56.9
novelty, n=4	71.4	81.3	73.6	62.7	78.5	79.9	67.5	78.8	81.4	68.4	74.0	79.7	66.7	64.9
redundancy, n=1	1.4	1.3	1.5	5.5	2.8	2.5	5.1	1.5	1.3	4.9	2.1	2.6	5.7	4.0
redundancy, n=2	0.1	0.1	0.2	0.3	0.4	0.4	0.8	0.2	0.1	0.9	0.3	0.3	0.4	0.4

Table 5.3: PMIndiaSum monolingual document-summary data analysis.

5.2.4 Multilingualism

On top of monolingual data, our PMIndiaSum features massive parallelism between documents and summaries across languages, which enables summarization between any languages covered in the data. As shown in Figure 5.3, most of the articles are available in at least two languages, and 232 articles are available in all languages. This allows the creation of cross-lingual and multilingual summarization data.

Technically, we align documents and summaries in different languages pivoting through their default English articles. Such multi-way parallelism results in $14 \times 13 = 182$ cross-lingual pairs in addition to the 14 monolingual data pairs. The average data size is 5477 for monolingual and 3408 for cross-lingual summarization. Table 5.4 lists data sizes for all 196 language directions.

5.2.5 Data split

To prevent data leakage, which is exposing models to test data (even in another language given our nature of parallelism) in the training phase, we carefully prepared consistent validation and test splits for each target summary language. The aforementioned 232 articles, available in all languages, were divided equally into validation and testing sets, with 116 document-summary pairs for each language in each set. Furthermore, to ensure model generalization to unseen data, no articles used for validation or testing appear in the training split.

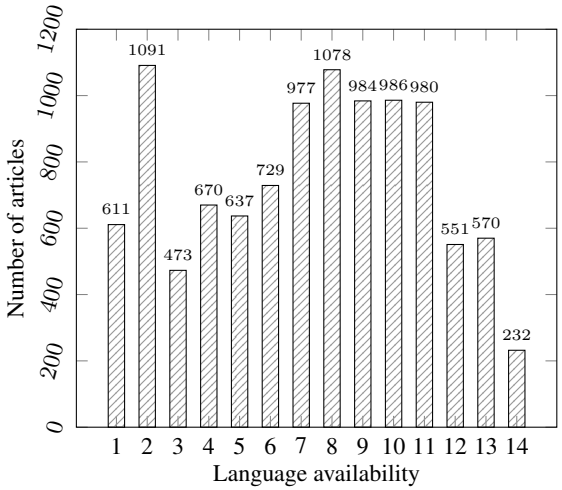

Figure 5.3: Plot of number of articles against the number of languages they are available in.

5.2.6 Quality consideration

We assess the quality of our dataset from three perspectives.

5.2.6.1 Text quality

The PMIndiaSum corpus boasts high-quality text featuring carefully curated texts from reliable sources written by native speakers. The removal of extraneous elements, such as HTML tags and potentially tweets, indicates a deliberate effort to maintain text integrity. Furthermore, the fact that the data are sourced from a governmental website adds to the corpus's credibility. The PMIndiaSum corpus offers well-structured, informative, and linguistically accurate texts, making it a valuable resource for various natural language processing tasks and research endeavours.

5.2.6.2 Summary quality

To estimate and compare the quality of the headline versus the first sentence as a summary, we performed a human evaluation. We randomly sample 50 document-summary pairs from Hindi and Telugu monolingual data. For each language, we invite three native speakers to evaluate the accuracy, informativeness, and quality of the summaries based on the following guidelines.

Human Evaluation Guidelines: In this task you will look at a set of articles and summaries. Each article will be presented with two (2) summaries. The aim of the evaluation is to determine which summary is better at conveying the main points of the article to the audience while providing enough information. Here are the steps you need to take in order to do the annotation:

1. Read the news article to understand the context of the summaries.

D_{L1} \ S_{L2}	as	bn	gu	hi	kn	ml	mni	mr	or	pa	ta	te	ur	en
as	2089	1153	1728	1716	1411	1166	1737	1415	1711	1332	1607	1570	1571	1743
bn	1153	5557	4655	3987	3573	3332	2898	4268	3079	3479	3845	4239	2345	4075
gu	1728	4655	6802	4896	4411	3901	3998	4940	4021	4052	4731	5111	3185	5216
hi	1716	3987	4896	7363	3680	3723	3163	4340	3784	3426	4817	4376	3061	5899
kn	1411	3573	4411	3680	5307	3176	3405	3800	3439	3526	3720	4151	2742	4325
ml	1166	3332	3901	3723	3176	4665	2423	3547	3037	2881	3957	3737	2299	3769
mni	1737	2898	3998	3163	3405	2423	4936	3120	2901	3198	3181	3596	3199	3807
mr	1415	4268	4940	4340	3800	3547	3120	5816	3450	3573	4219	4488	2582	4411
or	1711	3079	4021	3784	3439	3037	2901	3450	4651	3224	3722	3667	2646	3873
pa	1332	3479	4052	3426	3526	2881	3198	3573	3224	4814	3356	3772	2489	3615
ta	1607	3845	4731	4817	3720	3957	3181	4219	3722	3356	6079	4514	3139	4909
te	1570	4239	5111	4376	4151	3737	3596	4488	3667	3772	4514	6126	2847	4955
ur	1571	2345	3185	3061	2742	2299	3199	2582	2646	2489	3139	2847	4315	3415
en	1743	4075	5216	5899	4325	3769	3807	4411	3873	3615	4909	4955	3415	8160

Table 5.4: Size of document-summary pairs in PMIndiaSum for all language directions. Each cell corresponds to a dataset (D_{L1}, S_{L2}), with the source document D in language $L1$ and the target summary S in language $L2$.

2. Read both summaries carefully and compare them with each other. Consider accuracy, informativeness, and quality of the content. We have provided the definition of these criteria in the below bullet points:

    - **Accuracy:** This refers to how closely the summary reflects the factual content of the news article. An accurate summary should not include any false information or misrepresentations of the article's content.
    - **Informativeness:** This refers to how much information the summary provides about the main points of the news article. An informative summary should provide enough detail to give the reader a good understanding of the article's content.
    - **Quality:** This refers to the overall standard of the summary. A high-quality summary should be fluent, well-written, free of errors, and easy to read.

3. Provide your binary decision by selecting the summary that you believe is better. Base your decision solely on the quality, accuracy, and informativeness of the content, without being influenced by

factors such as writing style, personal preferences, or biases. Please note that one of the summaries may be an incomplete sentence, and this is acceptable as long as it does not compromise the quality, accuracy, or informativeness of the summary. Please ensure that your decision is solely based on the evaluation criteria and not on other factors. This will help ensure that the evaluation is objective and consistent.

4. If there is no clear winner, choose the summary that you believe is more suitable for the target audience.

The evaluators need to answer whether the text presented is a summary and whether the headline or the first sentence is a better summary. Out of the three evaluations, the best summary was chosen. In Telugu, 33 out of 50 headlines were rated better than the first sentence, while in Hindi it was 17 out of 50. Evaluators were also asked to rate whether the headline or first sentence is a summary. For Telugu, 49 out of 50 rated the headline as a summary, and the same was observed for Hindi. However, for the first sentence, 45 out of 50 rated it as a summary in Telugu, while in Hindi the score was 35 out of 50. Human evaluation indicates that all headlines are summaries of the article, but not all first sentences are summaries. Potentially this is because headlines from a governmental website are very informative.

5.2.6.3 Parallelism

Finally, we assert the validity of cross-lingual document-summary pairs. We measure the degree of parallelism by calculating cosine between neural presentations (specifically LaBSE by [21]) of texts in two languages. We compute LaBSE scores between documents, as well as between summaries, for each language pair. The numbers are then presented in Table 5.5.

The average LaBSE score was 0.86 for cross-lingual summaries and 0.88 for cross-lingual documents. These scores indicate the high parallelism between documents and between summaries. These also significantly exceeds the 0.74 threshold used to extract CrossSum from the XL-Sum [27, 6].

5.3 Benchmark Experiments

In accompanying our dataset, we also conduct benchmark experiments using various methods, spanning monolingual, cross-lingual, and multilingual summarization. We release our scripts alongside the dataset, hoping these to serve as experimental baselines for future work.

5.3.1 Task and evaluation

Formally, given a document D, the process of summarization should produce a summary S with a shorter length, yet conveying the most important information in D. We explore three types of summarization models defined by language directions.

$L1$ \ $L2$	as	bn	gu	hi	kn	ml	mni	mr	or	pa	ta	te	ur	en
as	-	0.88	0.86	0.82	0.87	0.87	0.80	0.85	0.90	0.87	0.85	0.88	0.85	0.87
bn	0.89	-	0.86	0.85	0.87	0.86	0.73	0.87	0.88	0.87	0.85	0.86	0.87	0.86
gu	0.85	0.90	-	0.92	0.91	0.88	0.70	0.90	0.90	0.92	0.88	0.90	0. 91	0.89
hi	0.81	0.89	0.93	-	0.89	0.86	0.65	0.91	0.87	0.91	0.88	0.88	0.90	0.89
kn	0.86	0.91	0.94	0.92	-	0.89	0.72	0.90	0.89	0.91	0.89	0.91	0.89	0.90
ml	0.88	0.90	0.91	0.89	0.93	-	0.73	0.88	0.89	0.89	0.87	0.88	0.88	0.88
mni	0.84	0.75	0.71	0.68	0.73	0.75	-	0.69	0.74	0.71	0.71	0.74	0.69	0.74
mr	0.85	0.90	0.93	0.93	0.92	0.90	0.71	-	0.89	0.90	0.88	0.89	0.89	0.89
or	0.90	0.91	0.92	0.90	0.92	0.92	0.76	0.91	-	0.90	0.88	0.89	0.88	0.90
pa	0.87	0.90	0.94	0.93	0.93	0.91	0.73	0.92	0.92	-	0.89	0.90	0.91	0.90
ta	0.85	0.90	0.91	0.90	0.93	0.91	0.73	0.91	0.91	0.91	-	0.89	0.88	0.88
te	0.87	0.91	0.92	0.90	0.94	0.92	0.75	0.91	0.92	0.93	0.92	-	0.89	0.90
ur	0.84	0.89	0.92	0.90	0.91	0.90	0.72	0.90	0.90	0.92	0.90	0.91	-	0.90
en	0.84	0.89	0.90	0.91	0.91	0.90	0.72	0.90	0.90	0.90	0.90	0.90	0.89	-

Table 5.5: LaBSE scores between parallel documents (D_{L1}, D_{L2}, upper right) and parallel summaries (S_{L1}, S_{L2}, lower left).

1. Monolingual: document D_L and summary S_L are in the same language L.

2. Cross-lingual: document D_{L1} and summary S_{L2} are in different languages $L1$ and $L2$.

3. Multilingual: monolingual and cross-lingual summarization from $D_{\{L1,L2,...,Ln\}}$ to $S_{\{L1,L2,...,Lm\}}$ within a single model.

In our context, we limit cross-lingual models to ones that summarize from one single language to another. Mono- and cross-lingual models are considered to be more focused on the target language and thus more accurate, whereas a multilingual model can significantly save storage and computation, and leverage the potential to share knowledge across languages.

For evaluation, we report F1 scores of ROUGE-1/2/L [40], as implemented in XL-Sum[8] because it supports all languages we are interested in. We use the default language-specific settings such as segmentation and stemming.

[8] https://github.com/csebuetnlp/xl-sum/tree/master/multilingual_rouge_scoring

5.3.2 Methodology

We intend to provide a benchmark of four conventional summarization approaches. We introduce the paradigms below and also label the language directions tested with each method in Table 5.6.

As learning-free baselines, we consider extractive summarization methods. One way is to select the first sentence (lead), and another way is to score each sentence in the document against the reference and pick the best-scoring sentence (oracle). Fine-tuning refers to loading a pre-trained language model (PLM) and then fine-tuning it for the task of summarization for intended languages. Summarization-and-translation schemes are designed for cross-lingual summarization only. In summarization-then-translation (sum-trans), the document undergoes monolingual summarization followed by translation into the target language; similarly, translation-then-summarization (trans-sum) means to translate the document into the target language followed by monolingual summarization in the language. Finally, in zero-shot summarization, we fine-tune a PLM on monolingual data for all available languages, but perform cross-lingual summarization instead. We tick the methods for available language directions in Table 5.6.

	mono	**cross**	**multi**
extractive: lead	✓		
extractive: oracle	✓		
sum-trans		✓	
trans-sum		✓	
fine-tuning: full	✓	✓	✓
fine-tuning: zero-shot		✓	

Table 5.6: Summarization approaches and language directions.

Extractive. We include two training-free baselines: 1) selecting the lead sentence, and 2) scoring each sentence in the document against the reference and picking the best-scoring one in an oracle way.

Fine-tuning. We load a pre-trained language model (PLM) and continue training it for summarization in any designated languages.

Summarization-and-translation. For cross-lingual summarization, it might be easier to delegate the language conversion to a translation system. With summarization-then-translation (sum-trans), a document undergoes monolingual summarization followed by translation into the target language; conversely, translation-then-summarization (trans-sum) refers to translating a document into the target language followed by monolingual summarization.

Zero-shot. We fine-tune a PLM on monolingual data in all available languages but perform cross-lingual summarization instead.

5.3.3 Models

Recent advances in PLM fine-tuning have shown promising progress in summarization for Indian languages [75, 77]. We base our fine-tuning paradigm on two such models: IndicBART[9] [16] and mBART-50[10] [41]. We followed each PLM's convention to add language identification tokens to inform the PLM of the source and target languages.

Regarding language coverage, `mni` and `ur` are not available in indicBART, whereas `as`, `kn`, `mni`, `or`, and `pa` are not supported by mBART-50. Since neither PLM covers the Manipuri language, our workaround is to add a randomly initialized embedding entry as the `mni` language token in both indicBART and mBART. As a result, we are able to cover all languages in our dataset.

Specifically in the summarization-and-translation workflow, monolingual summarization is done using the above-mentioned monolingual fine-tuned PLMs, and utilized the translation system deployed by LTRC-IIIT-Hyderabad, utilizing the methods presented in [49, 48]. The MT systems[11], achieved BLEU scores of 36.33, 21.61, 18.73, 18.36 and 15.89 for English-Hindi, English-Telugu, English-Gujarati, English-Punjabi, and English Marathi language pairs, respectively, on the Flores benchmarks.

	Lead			**Oracle**			**IndicBART**			**mBART**		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
as	43.18	31.65	41.62	46.39	34.47	44.73	**58.22**	**41.72**	**56.05**	-	-	-
bn	43.40	27.39	40.28	47.32	**30.98**	43.74	48.88	27.80	46.15	**51.37**	30.45	**48.19**
gu	46.50	33.73	45.25	52.33	38.40	50.94	**66.29**	49.82	64.61	66.26	**50.09**	**64.80**
hi	60.56	44.80	58.16	64.74	49.08	62.51	68.85	53.10	66.90	**71.67**	**55.92**	**69.37**
kn	41.18	26.35	37.42	46.78	31.11	42.79	**58.62**	**39.94**	**56.25**	-	-	-
ml	38.05	23.77	37.06	47.38	**32.72**	45.82	48.58	30.33	**47.47**	**48.79**	30.24	47.35
mni	52.18	38.30	50.50	56.39	**42.01**	54.15	55.09	38.65	52.98	**58.04**	41.32	**56.40**
mr	47.90	31.02	45.21	51.26	33.53	48.70	62.27	42.87	59.58	**63.46**	**43.20**	**61.05**
or	41.93	21.83	37.26	47.05	26.15	42.00	**58.25**	**38.56**	**55.45**	-	-	-
pa	61.23	43.95	57.71	64.58	47.09	60.73	**72.03**	**54.21**	**68.50**	-	-	-
ta	42.28	27.90	41.64	54.48	38.03	52.80	63.90	**47.51**	62.17	**64.90**	47.42	**63.33**
te	47.64	31.20	41.03	**51.43**	**34.42**	**45.21**	33.84	16.25	32.70	34.37	15.99	33.42
ur	33.49	21.59	30.15	57.52	39.94	52.73	-	-	-	**68.92**	**52.64**	**64.83**
en	44.26	33.30	42.09	50.82	38.68	48.18	76.11	63.45	74.47	**79.27**	**66.88**	**77.75**

Table 5.7: Monolingual summarization benchmark results.

[9] `https://huggingface.co/ai4bharat/IndicBARTSS`. We used the IndicBARTSS variant, which deals each language in its own script without the need of mapping to or from Devanagari.

[10] `https://huggingface.co/facebook/mbart-large-50`

[11] `https://ssmt.iiit.ac.in/translate`

5.3.4 Experimental setup

For fine-tuning IndicBART and mBART for monolingual, cross-lingual and multilingual experiments, we always set a training budget of 100 epochs and also apply early stopping if there are three consecutive non-improving validation cross-entropy. We use an effective batch size of 96 by a combination of different batch sizes, numbers of GPUs, and gradient accumulation to utilize different GPUs. For document-summary pairs, we set the maximum lengths to 1024 and 64 respectively. Specifically, IndicBART and mBART-large-50 models have 244M and 610M parameters correspondingly. All other configurations follow the default in the Hugging Face trainer[12].

	IndicBART			**mBART**		
	R1	R2	RL	R1	R2	RL
as	0.26	0.61	0.49	-	-	-
bn	1.00	0.98	0.83	0.39	0.15	0.34
gu	1.26	1.53	1.40	1.22	1.26	1.46
hi	0.83	0.77	0.78	0.42	0.53	0.35
kn	0.85	0.63	0.77	-	-	-
ml	0.78	0.82	0.59	1.39	1.63	1.41
mni	0.81	0.25	0.73	3.00	1.90	2.68
mr	1.51	1.40	1.43	1.49	1.33	1.12
or	1.19	1.48	1.19	-	-	-
pa	0.60	0.65	0.60	-	-	-
ta	0.64	0.88	0.53	0.67	0.66	0.40
te	0.67	0.44	0.86	1.10	1.28	1.13
ur	-	-	-	1.95	2.80	2.06
en	0.09	0.40	0.04	1.12	2.02	1.08

Table 5.8: Monolingual summarization benchmarks standard deviation scores.

5.3.5 Results and Analysis

Monolingual. Results for monolingual summarization are listed in Table 5.7 and Table 5.8. Regarding two extractive baselines, oracle extractions are considerably better than picking the first sentence, indicating that the summary information is scattered across a document. Our PLM fine-tuning yields higher numbers than extractions, and mBART is generally better than IndicBART, but it supports fewer languages.

[12] https://huggingface.co/docs/transformers/main_classes/trainer

	Summarization-Translation						Translation-Summarization						Fine-tuning						Zero-shot					
	IndicBART			mBART			IndicBART			mBART			IndicBART			mBART			IndicBART			mBART		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
hi-en	45.07	24.94	41.61	46.16	25.51	43.09	**61.48**	40.52	**59.78**	59.55	36.25	56.80	60.73	**40.58**	58.73	57.16	35.33	54.88	1.05	0.03	1.05	3.55	0.32	3.45
en-hi	44.73	24.26	39.64	47.17	26.65	41.66	60.26	38.80	56.48	**61.01**	**39.49**	**57.48**	54.23	33.88	51.26	58.89	37.87	55.62	0.98	0.00	0.98	0.97	0.02	0.96
gu-te	21.00	6.10	19.54	20.93	6.38	19.60	**34.09**	**16.37**	**32.79**	26.08	9.06	24.93	31.38	14.30	30.40	31.19	14.32	30.25	1.22	0.06	1.13	0.92	0.04	0.89
te-gu	15.47	4.36	14.56	15.83	4.43	14.83	37.90	18.02	36.64	23.74	8.72	22.50	38.16	16.92	37.15	**41.94**	**20.39**	**40.68**	1.05	0.06	0.98	1.20	0.08	1.16
ml-mni	**27.00**	**14.90**	**24.74**	25.82	13.54	23.72	16.98	8.16	16.16	17.17	8.81	16.41	20.19	8.20	18.94	19.70	7.63	18.88	0.00	0.00	0.00	0.00	0.00	0.00
mni-ml	23.16	**7.48**	**22.28**	**23.26**	7.38	22.08	19.77	5.78	19.42	15.68	3.33	14.71	15.36	4.67	14.60	6.91	0.65	6.79	0.05	0.00	0.05	0.00	0.00	0.00
mr-bn	35.80	13.96	32.19	**36.96**	**14.08**	**33.00**	35.42	12.72	31.66	34.53	11.49	31.21	33.95	12.89	30.37	34.43	12.76	31.51	0.00	0.00	0.00	0.00	0.00	0.00
bn-mr	34.98	13.35	31.33	36.72	14.42	32.80	39.32	**19.14**	36.66	37.50	17.84	34.91	38.41	16.59	35.46	**39.78**	19.06	**37.07**	0.00	0.00	0.00	0.00	0.00	0.00
te-ta	32.43	14.34	30.46	32.77	13.49	31.09	43.27	24.30	41.49	**44.97**	**25.10**	**42.96**	35.91	17.89	34.76	36.81	16.51	35.67	1.17	0.06	1.10	1.22	0.06	1.16
ta-te	40.81	**22.54**	37.77	**41.29**	22.31	**38.38**	30.77	14.00	29.57	31.28	13.21	30.08	32.04	15.74	30.88	30.31	14.15	29.36	1.50	0.11	1.41	1.40	0.12	1.33
mni-en	32.40	15.36	28.37	33.82	16.74	29.97	**39.53**	**19.33**	**38.14**	33.03	14.45	31.63	37.41	18.10	35.71	24.38	9.25	22.16	0.05	0.00	0.05	1.55	0.20	1.56
en-mni	**38.95**	**23.99**	**35.73**	38.73	23.66	35.30	23.31	12.91	21.99	25.23	14.23	23.90	19.85	7.76	18.38	15.21	5.94	14.86	0.00	0.00	0.00	0.00	0.00	0.00

Table 5.9: Cross-lingual summarization benchmark.

	Summarization-Translation						Translation-Summarization						Fine-tuning						Zero-shot					
	IndicBART			mBART			IndicBART			mBART			IndicBART			mBART			IndicBART			mBART		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
hi-en	0.53	0.35	0.47	0.43	0.50	0.34	0.47	0.52	0.56	1.00	0.89	1.08	0.29	0.26	0.20	6.41	6.87	6.03	0.05	0.02	0.05	4.51	0.44	4.34
en-hi	0.63	0.42	0.82	0.10	0.29	0.49	0.64	0.77	0.81	1.07	1.27	1.03	0.19	0.28	0.22	1.36	1.77	1.74	0.05	0.00	0.05	0.18	0.03	0.19
gu-te	0.45	0.22	0.41	0.44	0.29	0.42	0.40	0.36	0.35	0.97	0.72	1.07	0.45	0.41	0.46	1.26	0.46	1.08	0.02	0.00	0.02	0.13	0.03	0.08
te-gu	0.21	0.44	0.16	0.93	0.64	0.96	0.39	0.09	0.38	2.02	0.85	1.93	0.24	0.33	0.19	0.56	0.68	0.16	0.00	0.00	0.00	0.09	0.03	0.05
ml-mni	0.42	0.41	0.44	1.55	1.51	1.54	0.38	0.39	0.24	0.83	0.40	0.86	0.56	0.23	0.43	1.52	1.05	1.50	0.00	0.00	0.00	0.00	0.00	0.00
mni-ml	0.41	0.68	0.55	2.27	1.51	1.95	0.62	0.48	0.64	0.40	0.09	0.21	0.25	0.25	0.26	3.22	0.77	3.08	0.00	0.00	0.00	0.00	0.00	0.00
mr-bn	0.53	0.65	0.39	1.79	1.50	1.50	0.08	0.31	0.13	0.07	0.42	0.02	0.11	0.17	0.18	2.27	1.08	1.86	0.00	0.00	0.00	0.00	0.00	0.00
bn-mr	0.32	0.18	0.17	1.21	0.78	1.24	0.98	0.68	0.76	2.08	1.08	1.81	0.13	0.18	0.08	2.40	1.68	2.40	0.00	0.00	0.00	0.00	0.00	0.00
te-ta	0.68	0.45	0.64	0.35	0.55	0.36	1.01	0.87	1.04	1.16	1.14	1.08	0.67	0.38	0.62	1.87	1.30	2.18	0.00	0.00	0.01	0.04	0.00	0.03
ta-te	0.27	0.11	0.31	1.47	1.51	1.14	0.40	0.30	0.45	0.24	0.41	0.08	0.40	0.34	0.34	1.13	0.76	0.90	0.01	0.00	0.01	0.09	0.00	0.04
mni-en	0.38	0.54	0.58	0.48	0.22	0.58	0.16	0.08	0.12	1.04	1.52	1.21	0.47	0.54	0.50	9.35	6.04	9.44	0.00	0.00	0.00	2.69	0.35	2.69
en-mni	0.85	0.54	0.83	0.22	0.39	0.56	0.28	0.32	0.20	2.82	1.65	2.80	0.24	0.24	0.30	3.72	1.35	3.54	0.00	0.00	0.00	0.00	0.00	0.00

Table 5.10: Cross-lingual summarization benchmarks standard deviation scores.

Cross-lingual. Testing all 182 cross-lingual directions is intractable given our resource constraint. Thus, we shortlist language pairs to fulfill as much as possible: 1) high and low resource availability, 2) combinations of language families as listed in Table 5.1, and 3) languages supported by both IndicBART and mBART for comparison purposes. According to the cross-lingual results in Table 5.9 and Table 5.10, we observe that the summarization-and-translation approaches outperform fine-tuning. Furthermore, results suggest the inability of zero-shot models to perform cross-lingual summarization with our data.
Multilingual. In this setting, both PLMs are fine-tuned with data for all language pairs. Results are reported in Table 5.11, 5.12, 5.13, 5.14 and Table 5.15, 5.16, 5.17, 5.18 correspondingly for IndicBART and mBART. While mBART supports fewer languages, we observe that it performs better than IndicBART, especially for cross-lingual summarization. When tested on cross-lingual, multilingual fine-tuned IndicBART seems to only produce reasonable numbers for summarization into `hi`, `pa`, `te`, and `en`. Next, multilingual fine-tuning does not help most of the monolingual directions, except for a few metrics on `bn`, `gu`, and `mni`. Moreover, for 7 out of 12 cross-lingual pairs, our multilingual mBART surpasses cross-lingual mBART, implying that the availability of other language pairs helps cross-lingual summarization. Finally, despite `mni` being an unseen language during pre-training of both PLMs, IndicBART copes less well than mBART during our fine-tuning, especially when summarizing from `mni`.

5.4 Conclusions and Future work

In this study, we have presented PMIndiaSum, a corpus created for 13 Indian languages, supporting 182 language pairs and encompassing mono, multi, and cross-lingual summarization. We have also detailed the preprocessing and filtration steps undertaken to ensure high-quality data. The dataset is publicly accessible and includes extractive baselines, summarization-translation, and fine-tuning benchmarks. Our experiments demonstrate the significance of multi-lingual and cross-lingual datasets for low-resource languages.

As a natural extension, future work could involve continuously integrating new articles from the source website to keep the dataset updated. However, care must be taken to ensure that experiment comparisons remain fair by maintaining consistent data size. Another promising direction for future research could involve pivoting through Indian languages to create a more comprehensive dataset.

Limitations

Our articles are scraped from a governmental website, leading to a domain bias towards political news, and a style bias towards headline-like summaries. Also, we selected the articles that are available in all languages to be validation and testing samples, in order to prevent information leakage. However, being available in all languages, those articles might have latent bias such as having been deemed as more important for all readers or easier to translate. Finally, the baseline numbers for some languages are

relatively high, leaving a small room for improvement. We suggest that future comparisons can omit ROUGE-1 and add ROUGE-3 numbers.

Ethics Statement

We place trust in the Indian governmental website to eliminate inappropriate content, but we could not inspect every data sample ourselves. On the other hand, the values promoted by a governmental agency might not align with every potential user of the data. Also, the provision of data in widely spoken languages might encourage research which edges out lesser-used languages.

	as			bn			gu			hi			kn			ml			mni		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
as	38.85	26.34	37.55	10.38	3.01	9.88	10.93	7.28	10.73	12.96	9.02	12.49	6.78	3.33	6.50	2.92	1.55	2.92	25.83	9.97	23.37
bn	7.95	2.32	7.74	30.78	17.34	29.19	11.48	6.21	11.04	10.58	6.12	9.92	7.81	4.44	7.42	2.41	0.97	2.41	25.08	10.34	23.21
gu	2.64	0.36	2.47	4.57	1.31	4.34	59.30	45.42	57.93	6.85	3.91	6.36	5.22	2.35	4.95	2.60	1.00	2.52	13.28	5.08	11.88
hi	5.82	1.74	5.5	4.20	0.87	4.02	9.75	4.96	9.51	62.77	48.46	60.56	11.10	5.71	10.37	6.51	2.75	6.30	17.96	7.03	16.17
kn	4.74	1.64	4.48	5.06	1.58	5.06	14.66	7.69	14.36	14.34	9.28	13.8	50.58	35.57	48.45	3.58	1.66	3.39	17.23	7.11	15.72
ml	4.86	1.52	4.67	4.83	1.25	4.78	17.39	9.34	16.96	18.30	11.44	17.56	14.9	6.36	13.95	41.21	23.97	39.97	16.11	6.19	14.70
mni	3.15	0.50	2.97	4.35	0.51	4.30	0.45	0.22	0.45	0.49	0.38	0.49	1.16	0.91	1.16	0.11	0.00	0.11	56.84	39.35	54.19
mr	5.53	1.88	5.25	7.52	2.51	7.17	19.90	11.57	19.50	35.01	21.93	33.61	12.34	6.44	11.71	9.83	4.50	9.51	19.31	7.77	17.53
or	5.96	2.49	5.94	7.75	2.85	7.49	16.2	8.62	15.62	18.56	11.46	17.84	8.32	3.97	8.00	4.93	2.16	4.86	20.64	8.09	18.88
pa	3.11	0.68	2.94	4.89	1.73	4.68	9.45	5.06	9.07	16.44	11.26	15.96	9.32	4.91	8.75	3.35	1.66	3.35	15.86	5.84	14.15
ta	2.75	0.84	2.62	4.98	1.73	4.91	17.37	9.46	16.84	17.26	11.00	16.73	8.54	4.67	8.15	5.3	2.14	5.07	16.39	6.45	14.81
te	1.02	0.19	1.02	1.98	0.49	1.93	3.86	1.46	3.69	3.74	2.24	3.52	1.40	0.40	1.40	0.87	0.16	0.87	7.28	2.79	6.61
en	2.37	0.58	2.13	5.23	1.08	5.04	14.04	7.68	13.67	15.92	10.78	15.30	11.22	5.53	10.35	4.44	2.09	4.28	18.07	7.17	16.70

Table 5.11: Multilingual summarization benchmark with IndicBART for as, bn, gu, hi, kn, ml and mni languages.

	mr			or			pa			ta			te			en		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
as	4.67	2.27	4.23	7.82	5.04	7.55	17.00	10.62	16.08	1.58	0.63	1.55	10.02	4.16	9.73	19.21	12.79	18.58
bn	5.70	3.20	5.51	9.22	5.29	9.02	17.5	10.78	16.46	2.01	1.07	1.99	9.48	3.44	8.95	22.27	14.10	21.23
gu	6.46	3.30	6.08	4.53	2.13	4.41	8.39	5.07	8.10	4.80	2.42	4.73	10.08	3.81	9.74	14.74	10.05	14.55
hi	14.39	6.55	13.80	9.55	5.43	9.31	26.86	17.15	25.19	7.68	4.06	7.51	15.27	6.44	14.78	24.45	16.46	23.97
kn	5.99	2.76	5.53	7.77	3.91	7.20	22.21	14.10	20.94	5.88	3.48	5.88	17.25	6.87	16.68	27.32	16.84	26.56
ml	8.58	3.80	8.22	10.91	5.69	10.71	29.39	19.05	27.85	9.90	4.70	9.64	22.04	9.71	21.27	30.48	20.17	29.75
mni	0.40	0.31	0.40	2.25	1.37	2.22	0.86	0.56	0.86	0.32	0.04	0.32	1.64	0.56	1.61	2.51	1.74	2.43
mr	47.47	32.36	45.23	13.89	8.03	13.54	30.19	19.25	27.90	7.00	3.41	6.95	15.99	7.07	15.35	29.84	19.96	28.89
or	9.23	5.24	8.78	50.9	35.58	49.22	24.73	15.76	22.84	6.22	3.79	6.22	14.49	6.69	13.95	23.50	14.86	22.77
pa	9.62	5.18	8.81	5.57	3.11	5.51	67.90	50.84	64.22	3.65	2.17	3.55	14.85	6.68	14.41	20.45	13.5	19.88
ta	5.80	2.71	5.34	7.27	4.40	7.20	18.24	11.67	17.03	57.99	41.02	56.35	16.43	7.79	15.85	25.36	17.10	24.48
te	1.12	0.13	0.93	1.08	0.10	1.08	4.82	2.96	4.75	1.12	0.38	1.05	30.82	13.28	29.69	7.94	4.91	7.75
en	8.40	3.77	8.19	6.16	3.13	6.01	19.70	12.23	18.35	6.94	3.99	6.91	15.35	6.72	14.69	70.38	57.85	69.22

Table 5.12: Multilingual summarization benchmark with IndicBART for mr, or, pa, ta, te and en languages.

	as			bn			gu			hi			kn			ml			mni		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
as	0.75	0.77	0.86	0.62	0.47	0.53	2.11	1.33	2.06	0.17	0.25	0.30	0.52	0.29	0.53	0.23	0.34	0.23	1.74	1.05	1.63
bn	1.27	0.64	1.24	0.93	0.64	0.95	1.74	0.73	1.74	1.16	0.80	1.22	1.70	0.63	1.49	1.08	0.45	1.08	0.10	0.39	0.17
gu	0.13	0.07	0.12	0.53	0.31	0.54	0.66	0.41	0.68	1.70	1.29	1.63	0.33	0.71	0.30	0.36	0.14	0.33	1.35	0.63	1.21
hi	1.09	0.58	1.10	0.69	0.17	0.62	0.89	0.44	0.88	0.60	0.37	0.70	2.25	1.66	2.26	0.21	0.39	0.25	0.47	0.08	0.39
kn	0.57	0.26	0.50	0.27	0.27	0.27	0.45	0.89	0.42	1.85	0.92	1.72	1.24	0.83	1.03	1.24	0.82	1.21	0.93	0.56	0.80
ml	0.77	0.22	0.74	1.25	0.66	1.21	1.57	0.70	1.51	0.89	0.78	0.89	1.32	0.59	1.25	0.58	0.53	0.82	0.62	0.24	0.48
mni	0.16	0.10	0.14	0.14	0.04	0.14	0.26	0.19	0.26	0.42	0.33	0.42	0.59	0.48	0.59	0.07	0.00	0.07	0.45	0.58	0.24
mr	0.35	0.21	0.42	1.45	0.64	1.36	1.04	0.58	1.07	1.39	1.13	1.39	1.87	0.99	1.87	2.19	0.90	2.09	1.51	0.73	1.34
or	0.42	0.21	0.44	0.26	0.23	0.27	1.18	0.57	1.17	1.63	1.07	1.61	1.25	0.70	1.18	0.41	0.25	0.30	1.19	0.40	0.97
pa	0.38	0.34	0.39	0.68	0.23	0.62	1.29	0.85	1.30	4.48	3.07	4.25	0.83	0.78	0.91	0.24	0.10	0.24	1.08	0.49	0.78
ta	0.47	0.17	0.45	0.15	0.18	0.15	1.44	1.14	1.36	2.34	1.60	2.21	0.16	0.16	0.23	0.70	0.44	0.71	0.75	0.89	0.80
te	0.18	0.06	0.18	0.11	0.03	0.11	0.47	0.10	0.48	0.50	0.33	0.47	0.72	0.40	0.72	0.40	0.10	0.40	0.53	0.17	0.53
en	0.59	0.19	0.48	0.91	0.12	0.86	1.19	0.81	1.13	1.94	1.37	1.80	0.92	0.22	0.72	0.61	0.53	0.60	1.01	0.26	0.85

Table 5.13: Multilingual summarization benchmarks standard deviation scores with IndicBART for as, bn, gu, hi, kn, ml and mni languages.

	mr			or			pa			ta			te			en		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
as	0.52	0.21	0.46	1.51	0.71	1.44	0.66	0.62	0.68	0.56	0.36	0.59	1.19	0.73	1.08	1.08	0.67	1.13
bn	0.57	0.50	0.63	0.86	0.69	0.67	1.87	1.19	1.71	1.55	1.04	1.52	0.82	0.56	0.81	0.48	0.35	0.43
gu	0.40	0.31	0.56	1.25	0.89	1.30	0.34	0.33	0.44	0.95	0.76	0.85	1.36	0.91	1.32	2.15	1.19	2.13
hi	2.18	1.34	2.02	2.76	1.69	2.54	2.32	1.41	2.27	0.83	0.60	0.90	0.49	0.36	0.56	1.24	0.65	1.21
kn	1.06	0.44	1.03	1.12	0.25	0.92	1.83	1.00	1.45	1.08	0.43	1.08	2.06	0.95	1.95	2.73	1.40	2.50
ml	2.15	0.98	1.97	1.89	1.09	1.89	1.74	0.98	1.60	0.68	0.25	0.59	1.59	0.88	1.45	0.31	0.36	0.23
mni	0.46	0.34	0.46	1.55	1.20	1.52	0.28	0.07	0.28	0.43	0.07	0.43	0.87	0.48	0.82	0.99	0.83	0.95
mr	1.51	1.17	1.53	1.40	0.96	1.25	2.20	1.48	2.14	0.34	0.62	0.38	1.44	0.63	1.40	3.29	1.58	3.18
or	0.84	0.68	0.85	1.84	1.50	1.60	0.84	0.82	0.80	0.52	0.57	0.52	0.46	0.18	0.39	0.55	0.19	0.49
pa	0.88	0.80	0.83	0.67	0.58	0.66	0.54	0.64	0.56	0.15	0.16	0.16	0.45	0.33	0.45	3.76	2.66	3.80
ta	0.37	0.42	0.42	0.88	0.36	0.77	1.03	0.52	1.05	1.63	0.97	1.58	1.03	0.42	0.83	1.59	0.98	1.39
te	0.77	0.22	0.72	0.53	0.17	0.53	0.72	0.58	0.77	0.79	0.55	0.79	0.33	0.25	0.43	1.64	1.02	1.63
en	0.88	0.60	0.88	0.63	0.66	0.66	1.53	1.06	1.31	2.13	0.98	2.08	0.27	0.23	0.29	1.21	1.05	1.18

Table 5.14: Multilingual summarization benchmarks standard deviation scores with IndicBART for mr, or, pa, ta, te and en languages.

	bn			gu			hi			ml			mni		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
bn	52.10	29.19	47.86	45.71	23.49	44.41	57.3	34.25	53.9	33.06	14.08	31.73	33.85	15.39	31.65
gu	38.00	16.36	35.13	66.85	48.29	65.07	58.42	36.63	55.27	32.48	14.47	31.08	33.82	15.62	31.36
hi	37.22	15.9	33.91	47.69	24.76	46.03	71.18	54.48	68.95	32.65	14.03	31.39	34.41	15.72	31.76
ml	37.65	15.36	34.68	44.75	22.74	43.42	58.20	36.20	54.84	47.12	27.14	45.55	33.74	15.86	31.44
mni	33.89	11.98	30.31	42.68	21.23	41.31	54.31	32.26	51.26	29.31	10.96	27.86	58.11	39.77	55.50
mr	37.20	15.57	34.47	47.02	24.53	45.67	60.56	38.23	57.04	32.72	14.40	31.18	34.08	15.58	31.44
ta	35.93	15.18	32.97	43.62	21.19	41.91	55.94	34.01	52.93	31.41	13.07	29.79	31.62	13.38	29.07
te	35.61	15.11	32.93	44.20	21.63	42.72	57.98	36.05	55.11	31.38	13.08	30.17	34.06	16.2	31.74
ur	36.19	14.45	32.81	46.3	23.79	44.57	57.02	34.71	54.04	30.60	12.65	29.50	32.70	14.49	30.60
en	37.26	15.73	34.26	47.99	25.10	46.41	59.84	38.05	56.85	32.37	13.62	31.09	35.71	16.92	33.27

Table 5.15: Multilingual summarization benchmark with mBART for bn, gu, hi, ml and mni languages.

	mr			ta			te			ur			en		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
bn	41.55	20.39	38.95	40.20	18.70	39.06	29.55	12.62	28.47	49.27	27.86	45.59	61.48	39.81	59.11
gu	42.43	21.28	39.84	40.68	19.1	39.09	30.62	13.12	29.50	49.23	28.15	45.77	62.97	42.87	60.95
hi	43.77	22.37	40.85	42.32	19.81	40.38	32.56	14.98	31.68	50.15	28.71	46.54	64.16	43.92	62.10
ml	40.40	19.63	38.44	40.05	18.60	38.50	30.62	13.59	29.64	48.97	27.32	45.24	62.80	41.29	60.45
mni	38.76	17.77	36.52	37.66	17.09	36.07	28.88	12.33	27.90	46.73	25.89	43.05	59.69	38.11	57.40
mr	60.73	40.18	58.01	41.51	18.81	39.53	31.11	13.74	30.04	48.42	26.72	44.47	63.31	41.89	61.04
ta	40.40	19.55	37.81	61.14	41.59	59.24	30.67	14.35	29.75	47.72	26.42	44.23	62.13	40.69	59.67
te	42.16	21.09	39.47	41.03	19.08	39.67	34.45	15.63	33.44	48.23	26.70	44.65	62.15	40.47	59.67
ur	41.67	20.27	38.97	40.81	19.13	39.57	30.91	14.26	29.92	66.49	48.26	62.18	62.39	41.87	60.32
en	42.51	20.7	39.43	41.95	20.20	40.56	31.38	14.27	30.47	51.43	29.92	47.96	78.19	64.03	76.66

Table 5.16: Multilingual summarization benchmark with mBART for mr, ta, te, ur and en languages.

	bn			gu			hi			ml			mni		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
bn	0.89	0.61	0.94	0.19	0.34	0.16	0.75	0.47	0.49	1.82	0.77	1.84	0.46	1.07	0.62
gu	1.37	2.01	1.77	0.79	0.37	0.85	0.49	0.55	0.36	2.44	2.32	2.45	0.82	0.31	0.66
hi	1.53	1.26	1.47	0.95	0.88	1.22	1.43	1.59	1.56	2.07	1.69	1.68	0.44	0.33	0.69
ml	0.85	1.15	0.79	1.66	0.57	1.26	0.39	0.29	0.13	0.72	1.83	0.77	2.00	1.52	1.99
mni	0.71	0.54	0.55	0.46	0.27	0.51	0.49	0.68	0.58	2.19	2.05	2.28	0.60	0.74	0.42
mr	1.47	1.52	1.47	0.78	1.18	0.60	0.79	0.77	1.03	2.24	1.79	2.20	0.47	0.60	0.6
ta	1.21	1.56	1.50	0.94	1.45	1.09	0.57	1.05	0.75	1.61	1.31	1.52	0.46	0.60	0.44
te	1.72	1.64	1.46	0.51	0.65	0.23	0.89	0.98	0.71	1.77	0.86	1.60	0.46	0.63	0.55
ur	1.20	1.23	1.26	0.55	0.83	0.64	0.76	0.99	1.02	1.59	1.38	1.64	0.17	0.73	0.50
en	1.60	1.85	1.71	0.62	1.43	0.66	1.31	0.83	1.37	1.90	1.16	1.41	0.29	0.15	0.48

Table 5.17: Multilingual summarization benchmarks standard deviation scores with mBART for bn, gu, hi, ml and mni languages.

	mr			ta			te			ur			en		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
bn	0.78	0.47	1.02	0.56	1.09	0.85	0.89	0.45	0.78	1.11	1.52	1.36	1.63	2.06	1.63
gu	1.16	1.62	0.70	0.85	1.09	0.54	1.94	1.78	1.92	0.38	0.93	0.58	1.12	1.70	1.12
hi	1.12	1.40	0.82	1.19	1.09	1.08	1.76	1.88	1.68	1.04	1.18	0.95	1.26	2.36	1.26
ml	2.28	2.40	1.93	1.93	2.74	1.97	1.71	1.81	1.59	0.97	0.73	1.02	2.89	3.55	2.99
mni	1.22	1.52	1.37	0.74	1.29	1.07	1.08	0.80	0.78	0.96	1.07	1.04	2.11	2.87	2.17
mr	1.54	1.75	1.12	1.36	1.19	1.42	0.73	1.04	0.82	0.64	0.50	0.65	0.72	0.94	0.60
ta	0.30	1.26	0.47	0.86	0.98	0.79	0.60	0.33	0.36	1.18	1.03	1.07	1.62	2.07	1.69
te	1.89	1.44	1.73	1.34	1.75	1.48	0.65	0.56	0.65	0.89	0.32	0.76	1.16	2.13	1.16
ur	0.79	1.19	0.57	0.44	0.05	0.65	1.83	1.79	1.84	0.90	0.97	1.09	0.78	1.50	1.06
en	1.15	1.22	1.12	0.83	1.00	0.98	1.02	1.05	0.72	0.56	0.69	0.41	1.18	2.34	1.26

Table 5.18: Multilingual summarization benchmarks standard deviation scores with mBART for mr, ta, te, ur and en languages.

Chapter 6

Multi-Perspective Scientific Document Summarization

This chapter deals with a specific novel problem for the scientific document summarization task. In general, summary writing is very subjective. Such that, having more than one summary is essential to capture the different perspectives of scientific document. We used the Multi-perspective scientific document summarization (MuP) dataset to tackle the problem of subjective bias. This chapter also explores various approaches to perform the multi-perspective summarization task.

6.1 Scientific Document Summarization: Approaches and Challenges

With the rapidly growing research community, the volume of scientific papers being published every year is also going up. Which makes it nearly impossible for researchers to stay on top of the latest research. Scientific document summarization plays a crucial role in mitigating this problem. However, generating generic summaries for scientific documents is a non-trivial task due to their specific structure, varied content and inclusion of citation sentences. Scientific articles often represent salient information through tables, figures, and pseudo-codes [3] and mathematical equations. And, generic text does not usually contain such elements.

The two widely used approaches for scientific document summarization are content-based [15, 57] and citation-based [51, 2, 85]. The former relies on traditional extractive and abstractive methods whereas, the latter locates the target paper by matching a portion of text with the citation sentences.

Almost all traditional summarization models, whether extractive or abstractive, follow supervised learning approach. That means, given a document the model learns to generate its summary based on its given gold (target) summary. However, in real world, summary writing is very subjective. For a given document, there could be multiple different yet valid summaries where each summary writer has written a summary of the same document from their perspective of the document. This subjectivity raises concerns about the evaluation ability of the model that is presented with only one gold summary. The MuP-2022 shared task is a novel attempt to address this concern. The goal of multi-perspective summarization task is to develop models that are capable of leveraging multiple gold summaries to generate one generic summary.

	Train		Validation	
#Pairs	18934		3604	
#Unique Pairs	8382		1060	
	Text	Summary	Text	Summary
#Avg Words	2671.41	113.57	2671	115.13
#Avg Sentences	122.35	4.78	121.14	4.82

Table 6.1: MuP Data Statistics

MuP-2022 shared task data contains a collection of scientific documents with multiple summaries. These summaries were collected by first taking (one or) multiple scientific peer reviews for each document and then extracting the introductory paragraph that summarizes the key contributions of the paper from the reviewer's perspective.

For this task, we explored several pretrained sequence-to-sequence models such as BART [38], T5 [64], and ProphetNet [63]. We also experimented with: a two-stage fine-tuning approach using the SciTLDR dataset [10] and the divide and conquer approach, by [22], that first divides the document into multiple sections to obtain section-wise summaries, and then aggregates all partial summaries to form the complete summary.

For the MuP 2022 shared task dataset, our fine-tuned BART$_{large}$ model remained the best among all our experiments by achieving 40.68 ROUGE-1 F1-score and 26.04 average ROUGE F1-score.

6.2 Corpus Description

The multi-perspective scientific document summarization task aims to generate a summary that covers various aspects of the document. Evaluating such a system with just one gold (or reference) summary negatively impacts the goal, as summaries are usually very subjective. Considering the fact that multiple summaries would help cover more different perspectives of the scientific document, which a single summary might have missed.

MuP2022 [14] shared task data[1] contains multiple reference summaries for majority of the training set documents, and all of the development set documents also had a minimum of 3 reference summaries. The corpus consists of around 10K papers and 26.5K summaries. The average length of the summaries is 114.3 words long.

[1] https://github.com/allenai/mup

6.3 Methodology

Self-supervised pretrained models like BART [38], T5 [64], XLNet [84], ProphetNet [63], PEGASUS [87] have been effective for many generative tasks. We experiment with these pre-trained models and fine-tune them on MuP dataset for this task.

BART is a transformer-based [80] standard sequence-to-sequence model modified to work as an auto-encoder [38]. A self-supervised autoencoder is trained on the corrupted text (addition of noise) and uses a language model to reconstruct the original text with the true replacement of corrupted tokens. BART uses five "noising" methods: token masking, token deletion, text infilling, sentence permutation, and document rotation.

T5 or **Text to Text Transfer Transformer** [64] is a transformer-based approach that converts all the text-based language problems into the text-to-text format. This strategy allows the use of the same model architecture across a diverse set of tasks. T5 is pretrained on a multi-task mixture of supervised and unsupervised tasks using the common crawled corpus. We fine tune T5 base model on MuP corpus.

ProphetNet [63] is a sequence-to-sequence pretraining model. The unique objective of this model is to predict the future n-grams as the self-supervised training strategy. Unlike the traditional sequence-to-sequence models, ProphetNet is optimized by n-step ahead prediction instead of one-step-ahead prediction. We experimented with ProphetNet models with and without fine-tuned on the CNN/DailyMail dataset.

Utilizing SciTLDR The TLDR [10] approach aims at creating extremely short summaries (TLDRs) for scientific documents. For this task, the authors introduced a SciTLDR dataset of 5400 TLDRs over 3200 papers.

DANCER [22] Most of the extractive and abstractive methods for scientific document summarization typically consider the input as abstract and/or full text of the article to generate the abstract-like summary. In contrast, DANCER divides the source text into multiple sections, generates an individual summary for each section, and aggregates the partial summaries to form the target summary.

6.4 Experiments

All of our experiments were performed on the same splits of train, validation and test sets as provided by the organizers. Table 6.1 shows the data statistics. We used NLTK tokenizer and the simplified version data released by the task organizers to report all the counts mentioned in Table 6.1.

The following subsections detail various categories of experiments. We hypothesise that various sections of the source document may contribute in multi-perspective reviews of the document reviews. The subsection 6.4.3 and 6.4.4 detail the experiments conducted, specifically, to capture various sections of the document.

Model	R-1	R-2	R-L	Avg R-f
Baseline	**40.8**	12.3	24.5	25.8
BART$_{large}$ cnn	40.68	**12.47**	**24.99**	**26.05**
DistilBART cnn	39.36	11.79	24.47	25.21
BART$_{base}$ cnn	39.12	11.42	23.8	24.78
T5$_{base}$	38.35	11.26	24.64	24.75
ProphetNet	38.15	11.45	24.25	24.62
BART$_{base}$	38.53	11.39	23.92	24.61
ProphetNet cnn	37.59	10.91	24.09	24.2
DANCER + BART	33.07	9.06	18.2	20.11
BART + Two-stage	32.51	6.82	20.64	19.99

Table 6.2: ROUGE scores for models fine-tuned on MuP2022 dataset

Parameters	BART	T5	ProphetNet
Max source length	1024	1024	512
Max target length	150	128	128
Min target length	56	30	56
Batch Size	1	1	1
Epochs	2	10	1
Vocab Size	50265	32128	30522
Beam Size	4	4	5
Learning Rate	5e-5	1e-4	5e-5

Table 6.3: Experimental Setup and Parameters Settings

				Train & Val Data				Test Data			
R-1	R-2	R-L	Avg R-f	1	2	3	4	1	2	3	4
40.68	**12.47**	**24.99**	**26.05**	✓					✓		
40.67	12.5	24.93	26.03	✓						✓	
40.47	12.29	24.76	25.84			✓			✓		
40.34	12.28	24.79	25.8				✓				✓
40.33	12.28	24.75	25.79		✓				✓		
40.39	12.25	24.73	25.79			✓				✓	
40.23	12.32	24.77	25.77	✓							✓
40.23	12.17	24.6	25.67				✓		✓		
40.1	12.25	24.63	25.66		✓			✓			
40.22	12.13	24.54	25.63	✓				✓			

Table 6.4: Impact of Data Variations

6.4.1 Existing Pre-trained Generation models

We experimented with existing SOTA generation models like BART [38], T5 [64] and ProphetNet [63]. Table 6.3 details the general experimental setup for each.

Experiments were conducted with different versions of these models, such as DistilBART-cnn, BART$_{base}$, BART$_{base}$-cnn (base model of BART fine-tuned on CNN dataset), and ProphetNet-cnn.

Among all these, BART$_{large}$ achieved better performance for the MuP task. We use the BART$_{large}$ model fine-tuned on the CNN/DailyMail dataset [28] to initialize our model.

6.4.2 Two Stage Fine-tuning

In order to follow TLDR [10] approach, we attempted two stage fine tuning. Using the available checkpoints in the Hugging Face Transformers Library [81], first we fine-tune the BART model on the SciTLDR dataset for 10 epochs with the max source and target token lengths of 1024 and 150 respectively. In the second stage, we fine-tune this model on the MuP dataset, with the same settings. However, as the bottom line of the Table 6.2 shows, this approach did not help with this MuP task.

6.4.3 Data Variation

The entire MuP dataset was released in two formats: one that consisted of the full-text of the scientific document along with meta-data and second, a simplified version of the source document. This simplified content is basically the pre-processed initial 2000 tokens of the documents' introduction sections.
We conducted a few experiments, with our submitted model, to investigate the contribution of various sections of these documents in the target summaries. For this, we created four categories of training, validation and test sets such that each category's source content consisted of one of the following combinations of sections of the source document:

1. ***Introduction*:** Only the introduction section of the document was used as the input to the BART model.

2. ***Abstract + Introduction*:** Both abstract and introduction sections, in concatenation, were utilized as the input for the BART model.

3. ***Abstract + Introduction + Conclusion*:** The BART model was fed with a combination of abstract, introduction and conclusion sections (if available) of the document.

4. ***Abstract + Conclusion*:** A combination of abstract and conclusion section was used as the input to the BART model.

First, we separately fine-tuned our BART$_{large}$ model using the training and validation data of each of these categories. Next, in each of these experiments all 4 models were tested with all 4 categories of test data. Table 6.4 shows the respective ROUGE f1-scores. Where, the checkmarks (✓) indicate the selected combination of training and test data category.
As shown in Table 6.4, the combination of '1' & '2' (i.e. only-introduction section for the training data and abstract + introduction for test data) outperforms all the rest. All these models were fine-tuned for two epochs and with the max source and target lengths of 1024 and 150, respectively.

6.4.4 Divide and Conquer Approach

Following the DANCER approach, we prepare the training, validation and test inputs by dividing each corresponding source documents into four sections: **Abstract, Introduction, Results and Discussion, and Conclusion**. We fine-tuned the BART model on each section of information separately and combined all the summaries at the end to get the final generated summary.

6.4.5 Impact of Hyperparameters

In order to find the optimal architecture for our BART$_{large}$ model, we experimented with number-of-epochs (1, 2, 3, 5) with default max-target-length of 128, where fine-tuning with 2 epochs showed better performance. We then tested for max-target-lengths (128, 150, 200) with 2 epochs. Where

Epochs	R-1	R-2	R-L	Avg R-f
1	40.5	12.48	24.88	25.95
2	**40.57**	**12.49**	**24.98**	**26.01**
3	40.31	12.23	24.8	25.78
5	40.35	12.02	24.59	25.65

Table 6.5: Impact of number-of-Epochs Variation

Epochs	Max Target Length	R-1	R-2	R-L	Avg R-f
2	128	40.57	**12.49**	24.98	26.01
	150	**40.68**	12.47	**24.99**	**26.05**
	200	40.67	12.47	24.99	26.04
5	128	**40.35**	12.02	24.59	25.65
	150	40.31	**12.1**	**24.66**	**25.69**

Table 6.6: Impact of Max-Target-Length Variation

max-target-length 150 gave slightly better performance than the remaining. Tables 6.5 and 6.6 detail the corresponding ROUGE f1-scores.

6.5 Results & Discussion

For the MuP task, we experimented with various pre-trained generation models, a couple of scientific document summarization approaches, methods to cover different sections of the document and parameter settings. As shown in Table 6.2, among all of these the BART$_{large}$cnn (our submitted) model performed the best. This model was fine-tuned for 2 epochs with max-target-length 150 and data combination 1-2 (as mentioned in section 6.4.3). With this model, we secured 3rd rank in the MuP-2022 shared task.
While the MuP task considers summaries from multiple reviewers as different "perspectives", most of these summaries cover only the major contributions of the paper. These summaries, though diverse in their construction, do not look at the research paper from different points-of-view. We see a validation of this claim from the results in table 6.4, where model trained on "introduction" section alone outperforms all other combinations.

Chapter 7

Conclusions and Future work

In this thesis, we have addressed the challenges involved in the creating resources for Indian languages. This thesis begins with a broad literature review to address the common practices followed to create summarization datasets and requirement of high-quality datasets. We proposed a pipeline that crowd-sources summarization data and then aggressively filter the content via: automatic and partial expert evaluation. Using the pipeline we create a high-quality Telugu abstractive summarization dataset (TeSum) which we validate with sampling-based human evaluation. A number of recently released datasets for summarization, scraped the web-content relying on the assumption that summary is made available with the article by the publishers. While this assumption holds for multiple resources (or news-site) in English, it should not be generalised across languages without thorough analysis and evaluation. Our analysis clearly shows that this assumption does not hold true for most Indian language news resources. We also provide the baseline numbers for various models commonly used for summarization. We show that our proposed filtration pipeline can even be applied to these large-scale scraped datasets to extract better quality article-summary pairs.

We tested the proposed filters on the Indian language summarization (ILSum) dataset, which includes Hindi, Gujarati, and English languages. To demonstrate the efficacy of the filters, we fine-tuned models using various combinations of filtered and noisy data through k-fold cross-validation. Additionally, we implemented multi-lingual models by incorporating data from more than one language.

We introduce the PMIndiaSum, a text summarization corpus for 14 languages in India, supporting monolingual, cross-lingual, and multilingual summarization between 196 language pairs. We detailed the processing steps, reported data statistics, conducted quality evaluations, and made the dataset publicly accessible. We also published benchmark results for extractive baselines, summarization-and-translation, and fine-tuning involving two pre-trained language models. Our experiments emphasize the value of a multilingual dataset for languages in India.

For the multi-perspective scientific document summarization (MuP) task, we experimented with various pre-trained generation models, a couple of scientific document summarization approaches, methods to cover different sections of the document and paramter settings. We observed that BART$_{large}$ cnn model outperformed the rest. While MuP task considers summaries from multiple reviewers as

different "perspectives", most of these summaries cover only the major contributions of the paper. These summaries, though diverse in their construction, do not look at the research paper from different perspectives. Where model trained on “introduction" section alone outperforms all other combinations.

7.1 Future Work

1. *Focus on User-Centric Summarization:* It is recommended to develop summarization systems that take into account specific user characteristics and needs. Instead of building generic summarization models, the focus should be on creating tailored summaries that cater to individual users' requirements.

2. *Explore Query-Oriented, Multi-Perspective, and Role-Oriented Summarization Tasks:* Further research should be conducted in the areas of query-oriented, multi-perspective, and role-oriented summarization tasks. These tasks have the potential to enhance the effectiveness and versatility of summarization systems.

3. *Improve Datasets Quality:* While scraping data from the internet is a common practice, it should be followed by proper filtration and minimal human evaluation to ensure high-quality summarization datasets. This is particularly important for producing reliable summaries in low-resource languages. Another direction is to include websites available in multiple languages to expand the data size, domain coverage, and language availability.

4. *Robust Multilingual Models for Low-Resource Languages:* To avoid factual errors and hallucinated summaries, it is necessary to develop robust multilingual summarization models for low-resource languages. This will enable the generation of accurate and coherent summaries in languages with limited resources.

5. *Robust Summarization Evaluation Metrics:* While n-gram based evaluation metrics like ROUGE are widely used, there is a need for more context-aware evaluation metrics. The development of robust evaluation metrics that take into account the context and coherence of summaries will facilitate more accurate and comprehensive evaluation of summarization systems.

   In conclusion, future work in the field of summarization should prioritize user-centric approaches, explore diverse summarization tasks, improve dataset quality, develop robust multilingual models, and enhance evaluation metrics to ensure the advancement of the field and the production of high-quality summarization systems.

Related Publications

- Ashok Urlana., Surange, N., Baswani, P., Ravva, P., & Shrivastava, M. (2022, June). **TeSum: Human-Generated Abstractive Summarization Corpus for Telugu**. In Proceedings of the Thirteenth Language Resources and Evaluation Conference (pp. 5712-5722).
- Ashok Urlana, Nirmal Surange, and Manish Shrivastava. **Ltrc@ mup 2022: Multi-perspective scientific document summarization using pre-trained generation models.** Proceedings of the Third Workshop on Scholarly Document Processing. 2022.
- Ashok Urlana, Sahil Manoj Bhatt, Nirmal Surange, and Manish Shrivastava. 2022. **Indian Language Summarization using Pretrained Sequence-to-Sequence Models.** In Working Notes of FIRE 2022 - Forum for Information Retrieval Evaluation, Kolkata, India, December 9-13, 2022 (Kolkata, India) (CEUR Workshop Proceedings). CEUR-WS.org.
- Ashok Urlana, Chen, Pinzhen and Zhao, Zheng and Cohen, Shay B. and Shrivastava, Manish and Haddow, Barry "**PMIndiaSum: Multilingual and Cross-lingual Headline Summarization for Languages in India**" submitted at ArXiv.

Other Publications

- Priyanka Ravva, Ashok Urlana, and Manish Shrivastava. **AVADHAN: System for Open-Domain Telugu Question Answering.** Proceedings of the 7th ACM IKDD CoDS and 25th COMAD. 2020. 234-238.

Bibliography

[1] *Pretrained Language Models for Text Generation: A Survey*, 2021.

[2] A. Abu-Jbara and D. Radev. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 500–509, 2011.

[3] N. I. Altmami and M. E. B. Menai. Automatic summarization of scientific articles: A survey. *Journal of King Saud University-Computer and Information Sciences*, 2020.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *proceedings of international conference of learning representations*, 2015.

[5] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.

[6] A. Bhattacharjee, T. Hasan, W. U. Ahmad, Y.-F. Li, Y.-B. Kang, and R. Shahriyar. Crosssum: Beyond english-centric cross-lingual abstractive text summarization for 1500+ language pairs. *arXiv preprint arXiv:2112.08804*, 2021.

[7] S. Bhosale, D. Joshi, V. Bhise, and R. Deshmukh. Marathi e-newspaper text summarization using automatic keyword extraction technique. *International Journal of Advance Engineering and Research Development*, 5(3):789–792, 2018.

[8] 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.

[9] A. Burney, B. Sami, N. Mahmood, Z. Abbas, and K. Rizwan. Urdu text summarizer using sentence weight algorithm for word processors. *International Journal of Computer Applications*, 46(19):38–43, 2012.

[10] I. Cachola, K. Lo, A. Cohan, and D. Weld. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online, Nov. 2020. Association for Computational Linguistics.

[11] Z. Cao, W. Li, S. Li, and F. Wei. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161, 2018.

[12] S. Chopra, M. Auli, and A. M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 93–98, San Diego, California, June 2016. Association for Computational Linguistics.

[13] A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[14] A. Cohan, G. Feigenblat, T. Ghosal, and M. Shmueli-Scheuer. Overview of the first shared task on multi-perspective scientific document summarization (mup). In *Proceedings of the Third Workshop on Scholarly Document Processing*, Online, Oct. 2022. Association for Computational Linguistics.

[15] E. Collins, I. Augenstein, and S. Riedel. A supervised approach to extractive summarisation of scientific papers. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 195–205, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics.

[16] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. Khapra, and P. Kumar. IndicBART: A pre-trained model for indic natural language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1849–1863, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[17] R. Damodar, A. Ramineni, and R. Konda. Telugu text summarization. *International Research Journal of Modernization in Engineering Technology and Science, India*, 2021.

[18] Z.-Y. Dou, P. Liu, H. Hayashi, Z. Jiang, and G. Neubig. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online, June 2021. Association for Computational Linguistics.

[19] A. R. Fabbri, X. Wu, S. Iyer, and M. Diab. Multi-perspective abstractive answer summarization. *arXiv preprint arXiv:2104.08536*, 2021.

[20] A. Fan, D. Grangier, and M. Auli. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[21] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[22] A. Gidiotis and G. Tsoumakas. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040, 2020.

[23] D. Graff, J. Kong, K. Chen, and K. Maeda. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34, 2003.

[24] M. Grusky, M. Naaman, and Y. Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[25] B. Haddow and F. Kirefu. PMIndia—A collection of parallel corpora of languages of India. *arXiv preprint arXiv:2001.09907*, 2020.

[26] M. Hanumanthappa, M. Narayana Swamy, and N. Jyothi. Automatic keyword extraction from dravidian language. *International Journal of Innovative Science Engineering and Technology*, 1(8):87–92, 2014.

[27] 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 *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online, Aug. 2021. Association for Computational Linguistics.

[28] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28:1693–1701, 2015.

[29] E. Hovy and C.-Y. Lin. Automated text summarization and the summarist system. In *TIPSTER TEXT PROGRAM PHASE III: Proceedings of a Workshop held at Baltimore, Maryland, October 13-15, 1998*, pages 197–214, 1998.

[30] J. S. Kallimani, K. Srinivasa, et al. Information retrieval by text summarization for an indian regional language. In *Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering (NLPKE-2010)*, pages 1–4. IEEE, 2010.

[31] J. S. Kallimani, K. Srinivasa, et al. Information extraction by an abstractive text summarization for an indian regional language. In *2011 7th International Conference on Natural Language Processing and Knowledge Engineering*, pages 319–322. IEEE, 2011.

[32] M. H. Khanam and S. Sravani. Text summarization for telugu document. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 18(6):25–28, 2016.

[33] Y. Kikuchi, G. Neubig, R. Sasano, H. Takamura, and M. Okumura. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

[34] T. K. Koo and M. Y. Li. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of chiropractic medicine*, 15(2):155–163, 2016.

[35] A. Kumar, H. Shrotriya, P. Sahu, A. Mishra, R. Dabre, R. Puduppully, A. Kunchukuttan, M. M. Khapra, and P. Kumar. IndicNLG benchmark: Multilingual datasets for diverse NLG tasks in Indic languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5363–5394, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.

[36] A. Kunchukuttan. The IndicNLP Library. `https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf`, 2020.

[37] F. Ladhak, E. Durmus, C. Cardie, and K. McKeown. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online, Nov. 2020. Association for Computational Linguistics.

[38] 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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[39] C. Li, W. Xu, S. Li, and S. Gao. Guiding generation for abstractive text summarization based on key information guide network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 55–60, 2018.

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

[41] 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.

[42] Y. Liu and M. Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[43] Y. Liu, P. Liu, D. Radev, and G. Neubig. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[44] K. U. Manjari. Extractive summarization of telugu documents using textrank algorithm. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 678–683. IEEE, 2020.

[45] 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 *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021.

[46] R. Meng, K. Thaker, L. Zhang, Y. Dong, X. Yuan, T. Wang, and D. He. Bringing structure into summaries: a faceted summarization dataset for long scientific documents. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1080–1089, Online, Aug. 2021. Association for Computational Linguistics.

[47] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[48] V. Mujadia and D. Sharma. The LTRC Hindi-Telugu parallel corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3417–3424, Marseille, France, June 2022. European Language Resources Association.

[49] V. Mujadia and D. M. Sharma. English-Marathi neural machine translation for LoResMT 2021. In *Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021)*, pages 151–157, Virtual, Aug. 2021. Association for Machine Translation in the Americas.

[50] 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.

[51] P. Nakov, A. Schwartz, and M. Hearst. Citation sentences for semantic analysis of bioscience text. In *Proceedings of the SIGIR'04 workshop on Search and Discovery in Bioinformatics*, 2004.

[52] R. Nallapati, B. Xiang, and B. Zhou. Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023, 2016.

[53] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

[54] C. Napoles, M. Gormley, and B. Van Durme. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montréal, Canada, June 2012. Association for Computational Linguistics.

[55] 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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

[56] A. Nenkova, S. Maskey, and Y. Liu. Automatic summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 2011.

[57] N. I. Nikolov, M. Pfeiffer, and R. H. Hahnloser. Data-driven summarization of scientific articles. In *WOSP 2018 Workshop Proceedings*, page 2_W24. EuropeanLanguage Resources Association, 2018.

[58] P. Over and J. Yen. An introduction to duc-2004. *National Institute of Standards and Technology*, 2004.

[59] S. Pattnaik and A. K. Nayak. A simple and efficient text summarization model for odia text documents. *Indian journal of computer science and engineering*, 11, 2020.

[60] R. Paulus, C. Xiong, and R. Socher. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*, 2017.

[61] J. Pfeiffer, A. Rücklé, C. Poth, A. Kamath, I. Vulić, S. Ruder, K. Cho, and I. Gurevych. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online, Oct. 2020. Association for Computational Linguistics.

[62] P. Pingali, J. Jagarlamudi, and V. Varma. A dictionary based approach with query expansion to cross language query based multi-document summarization: Experiments in telugu-english. *National Workshop on Artificial Intelligence*, 2006.

[63] W. Qi, Y. Yan, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang, and M. Zhou. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online, Nov. 2020. Association for Computational Linguistics.

[64] 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 unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[65] S. Renjith and P. Sony. An automatic text summarization for malayalam using sentence extraction. In *proceedings of 27th IRF International Conference*, 2015.

[66] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. *CoRR*, abs/1509.00685, 2015.

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

[68] S. Satapara, B. Modha, S. Modha, and P. Mehta. Findings of the first shared task on indian language summarization (ilsum): Approaches, challenges and the path ahead. In *Working Notes of FIRE 2022 - Forum for Information Retrieval Evaluation, Kolkata, India, December 9-13, 2022*, CEUR Workshop Proceedings. CEUR-WS.org, 2022.

[69] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[70] P. E. Shrout and J. L. Fleiss. Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, 86(2):420, 1979.

[71] S. Sinha and G. N. Jha. An overview of indian language datasets used for text summarization. *ICT with Intelligent Applications: Proceedings of ICTIS 2022, Volume 1*, pages 693–703, 2022.

[72] K. Spärck Jones. Automatic summarizing: factors and directions. In *Advances in automatic text summarization*, volume 1, pages 1–12, Cambridge, Mass, USA, 1998. MIT Press.

[73] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[74] Y. Tang, C. Tran, X. Li, P. Chen, N. Goyal, V. Chaudhary, J. Gu, and A. Fan. Multilingual translation with extensible multilingual pretraining and finetuning. *CoRR*, abs/2008.00401, 2020.

[75] D. Taunk and V. Varma. Summarizing indian languages using multilingual transformers based models. *arXiv preprint arXiv:2303.16657*, 2023.

[76] C. Thaokar and L. Malik. Test model for summarizing hindi text using extraction method. In *2013 IEEE Conference on Information & Communication Technologies*, pages 1138–1143. IEEE, 2013.

[77] A. Urlana, S. M. Bhatt, N. Surange, and M. Shrivastava. Indian language summarization using pretrained sequence-to-sequence models. *arXiv preprint arXiv:2303.14461*, 2023.

[78] 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, Marseille, France, June 2022. European Language Resources Association.

[79] 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, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[80] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[81] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics.

[82] 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. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online, June 2021. Association for Computational Linguistics.

[83] D. N. S. Y Madhavee Latha. Multi-document abstractive text summarization through semantic similarity matrix for telugu language. *International Journal of Advanced Science and Technology*, 29(1):513–521, 2020.

[84] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[85] M. Yasunaga, J. Kasai, R. Zhang, A. R. Fabbri, I. Li, D. Friedman, and D. R. Radev. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7386–7393, 2019.

[86] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed. Big bird: Transformers for longer sequences, 2021.

[87] J. Zhang, Y. Zhao, M. Saleh, and P. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

[88] Z. Zhao and P. Chen. To adapt or to fine-tune: A case study on abstractive summarization. In *Proceedings of the 21st Chinese National Conference on Computational Linguistics*, pages 824–835, Nanchang, China, Oct. 2022. Chinese Information Processing Society of China.

[89] J. Zhu, Q. Wang, Y. Wang, Y. Zhou, J. Zhang, S. Wang, and C. Zong. NCLS: Neural cross-lingual summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3054–3064, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.