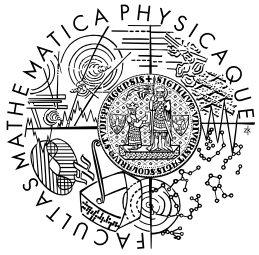

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Miftahul Jannat

Cross-Language Summarization

Institute of Formal and Applied Linguistics

Supervisors of the master thesis: RNDr. Jiří Hana, Ph.D.
Prof. Claudia Borg, Ph.D.

Study programme: Language Technologies and
Computational Linguistics

Prague 2025

I declare that I carried out this master thesis on my own, and only with the cited sources, literature and other professional sources. I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

To the Almighty for this life, for giving me strength, hope when I had none.

To my Ammu and Abbu, your love is the foundation of all I am. Your sacrifices, your prayers, your faith in me, they carried me further than I ever imagined. To Bhaiya, Apu, Bhabi, Dulabhai, my Niece and Nephews, thank you for always taking care of me even when the distance is 4500 miles. You are my home, always.

To Prof. Claudia Borg and Dr. Jiří Hana, your guidance was more than supervision. You showed kindness and belief when I needed it most. I will always be grateful for your mentorship.

To the Erasmus Mundus Program, the LCT administration, coordinators, and the teachers at Charles University and the University of Malta, thank you for opening the doors that led me here and for trusting in my journey.

To my LCT friends on this unforgettable journey, each of you left a mark on my story. Especially Kate, Sara, Mariam, Michelle, Keenu, Anna, and Jehad for all the good memories. To my friends from back home, your love reached me. Fatema, Akash, thank you so much.

And last but not least, to my beloved Husband, my best friend, my anchor, my safe place, your love gave me the courage to finish what I started.

This Master's thesis is for all of you. With love, with gratitude; always.

Title: Cross-Language Summarization

Author: Miftahul Jannat

Institute: Institute of Formal and Applied Linguistics

Supervisors: RNDr. Jiří Hana, Ph.D., Institute of Formal and Applied Linguistics, Charles University, Czechia. Prof. Claudia Borg, Ph.D., Department of Artificial Intelligence, University of Malta, Malta

Abstract: Cross-Language Summarization (CLS) is a crucial task of NLP in which the goal is to generate a summary in a target language that differs from the language of the input document. This thesis investigates the capabilities of a two-stage Extractive–Abstractive framework for CLS across three languages—English, Hindi, and Bengali; spanning four domains. The finding highlights, even with limited training data, how training in a multilingual environment with a high-resource language uplifts the performance of low-resource languages like Hindi. In contrast, Bengali, a lower-resourced language, exhibits lower performance under similar conditions. It is also evident that Strategic curation of datasets plays a crucial role; specifically, the use of curated data enhances the performance of mT5, highlighting the impact of well-balanced training samples in cross-lingual settings. We evaluated the quality of summaries using automatic metrics (e.g., ROUGE-L), and we found an alignment with human judgment and ChatGPT-based evaluation.

Keywords: Cross-Language Summarization, Multilingual, Multidomain, Low-resource Languages

Contents

Introduction

In Cross-Language Summarization (CLS), the goal is to generate a summary in a target language that differs from the language of the input document. This NLP task is challenging as it requires solving two complex NLP tasks: Summarization and Translation.

In recent years, there has been growing interest in the CLS task for low-resource languages [1]. In this thesis, we focus on three languages: English, which is a high-resource language; Hindi, a low-resource language; and Bengali, which has even fewer available resources. The three languages in this study differ in their writing scripts, which adds another layer of complexity to the Cross-Lingual Summarization task.

Most existing CLS work for low-resource languages is limited to a single domain, typically news articles [1, 2]. In contrast, our work explores both language and topic diversity, which adds complexity to the task but also makes it more realistic and practical. For instance, a multinational company that offers a wide range of products globally might want to understand regional customer preferences, product feedback, or overall satisfaction across different languages and domains. Our research setting better reflects such real-world needs.

The key research questions of this thesis are:

- **RQ1: To what extent can a two-stage Extractive–Abstractive framework effectively support cross-lingual summarization in low-resource, Multi-Lingual, and Multi-Domain scenarios?**
  With this question, we aim to evaluate the capability of a two-stage pipeline where an extractive model selects salient content and an abstractive model generates fluent summaries, in a situation of limited language resources, linguistic, and subject diversity.

- **RQ2: What are the relative advantages and performance differences between mBART and mT5 in handling Abstractive Summarization tasks across diverse languages such as Bengali, Hindi, and English?**
  With this question, we aim to evaluate and compare the performance of these two models in the same experimental setups for a Cross-Lingual task.

- **RQ3: Can the performance of a low-resource language improve when trained alongside a high-resource language in a Multilingual setting?**
  The potential of Cross-Lingual Transfer Learning in Multilingual Models is explored with this question. By training low-resource and high-resource language pairs jointly, we aim to assess whether the shared representations and multilingual pretraining can help the model generalize better and improve summarization quality in the low-resource language.

- **RQ4: What is the extent of correlation between Human judgments and ChatGPT-based evaluations in assessing CLS quality?**
  Following the same criteria, we sought to identify the agreement and disagreement between human evaluation and AI evaluation.

To address these research questions, we constructed two subsets from the publicly available XWikiRef dataset [3]. These subsets, referred to as Initial-20 and Curated-20 throughout this thesis, cover three languages and span four distinct domains (*Films*, *Books*, *Sportsman*, *Writers*). Following the work of [3], we trained and evaluated our models under three configurations: (1) Multidomain, which explores subject variability within each language; (2) Multilingual, which examines linguistic diversity across each subject; and (3) a combined setup, referred to as Multilingual-Multidomain, which incorporates both language and domain variation.

This thesis comprises four chapters. Chapter 1 provides an overview of the Cross-Language Summarization task, including various approaches, prior research on this topic, and available datasets. In Chapter 2, I discuss the dataset selection strategy and the various experimental setups employed in this work. Chapter 3 contains performance evaluation of every experiment, error analysis, and the key findings. In Conclusion, I provide a summary of this thesis, its limitations, and future work. The materials for this work can be found online[1].

[1]https://github.com/Miftahul7/Cross-Language-Summarization

1 Background

This chapter presents an overview of the Cross-Language Summarization task.

1.1 Text Summarization

Text summarization is the process of generating a shorter version of a text that preserves its most important content and overall meaning. The goal is to produce a coherent and concise summary that captures the main ideas from the source [4]. In the following sections, we outline several different variations of the text summarization task.

1.1.1 Monolingual Summarization

This task handles only one language in a single model where a summary is generated in the same language $\mathcal{L}_T$ as the language of the input document $\mathcal{L}_I$; $\mathcal{L}_I = \mathcal{L}_T$.

1.1.2 Multilingual Summarization (MLS)

Multilingual Summarization (MLS) handles multiple languages in a unified model where a summary is generated in the same language $\mathcal{L}_T$ as the language of the input document $\mathcal{L}_I$; $\mathcal{L}_I = \mathcal{L}_T$ [5].

1.1.3 Cross-Language Summarization (CLS)

In Cross-Language Summarization (CLS), a summary is generated in a target language $\mathcal{L}_T$ that differs from the language of the input document $\mathcal{L}_I$. Formally, given an input document $D^{\mathcal{L}_I} = \{x_1, x_2, \ldots, x_m\}$, the goal is to generate a summary $S^{\mathcal{L}_T} = \{y_1, y_2, \ldots, y_n\}$, and $\mathcal{L}_I \neq \mathcal{L}_T$. Following the formal definition introduced by Wang et al. [6], the model learns to estimate the conditional probability:

$$P_\theta(S^{\mathcal{L}_T} \mid D^{\mathcal{L}_I}) = \prod_{t=1}^{n} P_\theta(y_t \mid D^{\mathcal{L}_I}, y_{<t}) \tag{1.1}$$

The fundamental difference between MLS and CLS approaches is represented in the diagrams in Figure 1.1.

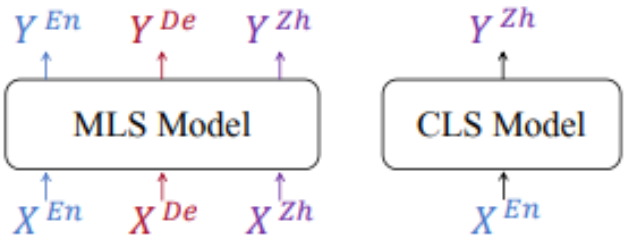

Figure 1.1 Difference in MLS and CLS models, from [5]

1.1.4 Early CLS Approaches

Early approaches often relied on pipeline architectures, where translation and summarization were performed in sequence, either by translating the source text first and then summarizing it [7, 8, 9, 10] or by summarizing the text first and then translating the summary [11, 12]. Although this strategy was adopted in several early works, such methods often introduce error propagation and repeated processing delays, making them less effective and inefficient for real-time applications [13].

1.1.5 End-to-End CLS Models

In recent years, there has been a noticeable change in research focus from traditional pipeline architectures toward end-to-end CLS models. The goal is to jointly handle both the translation and summarization tasks within a single unified framework, thereby reducing cascading errors and improving efficiency. Several advanced strategies have emerged in this domain:

- Multi-task Learning: This approach uses shared representations across related tasks such as summarization, translation, and language modeling, to improve cross-lingual generalization [14, 15].
- Knowledge Distillation Methods: These methods enhance model training with knowledge transference from high-capacity teacher models (often trained on high-resource languages) to smaller or student models (specific to the target language), which helps to improve performance in low-resource settings [16].
- Resource-Enhanced Approaches: These approaches incorporate external resources to augment the input document's information, which allows output summaries to be generated based on both the original encoded content and the added contexts [17].
- Pre-training Strategies: These utilize large-scale Multilingual corpora and Masked Language Modelling objectives to initialize models with cross-lingual capabilities before fine-tuning on summarization tasks [18, 19].

The strategies outlined above are illustrated in Figure 1.2.

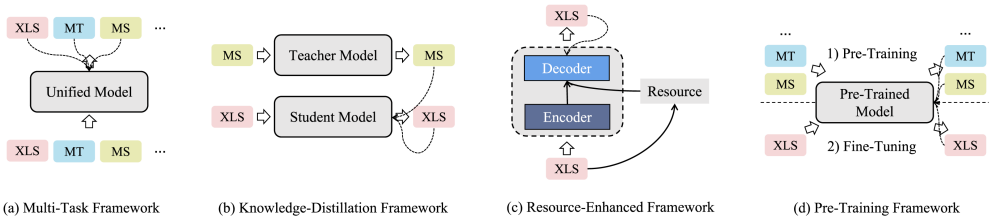

Figure 1.2 End-to-End CLS Approaches from [20]. MT= Machine Translation, XLS= CLS, MS= Monolingual Summarization.

1.2 Two-Stage Summarization Architecture

Inspired by the work of [3], in this work, we followed a two-stage architecture, consisting of an *Extractive* stage followed by an *Abstractive* stage.

1.2.1 Extractive Approach

Extractive summarization is an approach to automatic text summarization where the summary is generated by selecting and concatenating the most salient sentences or text units directly from the original document(s) [21]. The main goal is to preserve the overall meaning of the content while significantly reducing the text length. This method does not involve rewriting or paraphrasing but instead identifies and lifts important content exactly as it is from the input. Unlike abstractive methods, which may generate new sentences or paraphrase content, extractive techniques rely entirely on the existing textual material.

Traditional methods

Earlier Extractive Summarization approaches heavily relied on manually crafted features to estimate sentence importance. Each sentence is evaluated based on these characteristics and assigned a relevance score. Summaries are then formed by selecting the highest-scoring sentences using various algorithms. These features are typically categorized as follows:

- **Structural Features:[22]**
  - Location of the sentence within the document (e.g., lead sentences).
  - Sentence length and title word overlap.
  - Presence of named entities
- **Event-Based Indicators:** Detection of action-related nouns to capture dynamic content [23].
- **Content-Oriented Features:** Term frequency measures, such as Term Frequency - Inverse Document Frequency (TF-IDF)[24].

Neural approaches

Recent neural approaches to extractive summarization, such as PriorSum [25], use Convolutional Neural Networks (CNNs) to turn each sentence into a vector, which acts as a prior score for how suitable that sentence is for a summary, without needing to look at surrounding context. Another method, R2N2 [26], applies Recursive Neural Networks (RNNs) that consider the sentence's grammar structure (via its parse tree) to build vector representations. Both methods outperform traditional, feature-based systems, showing that neural models can effectively capture hidden semantic features for summarization.

Salience-Based Extractive Techniques

In this work, we follow [3], where Salience-based Extractive Summarization (ES) is employed to identify the Top-K most relevant sentences from a set of input references, to a given section heading. The process begins by segmenting the reference corpus into individual sentences. Each sentence is then paired with the target section title and fed into the pretrained XLM-RoBERTa model [27], which operates as the model parameters remain unchanged.

Relevance scores are derived from the likelihood estimates produced by the language model, which reflect how well a sentence aligns with the contextual meaning of the section title. Based on these scores, the Top-K highest-scoring sentences are selected.

1.2.2 Abstractive Approach

The Abstractive Summarization (AS) stage of our framework involves generating a fluent, coherent, and concise text from the selected filtered content from the first stage. The goal of AS is to create a high-level semantic representation of the input text and then apply Natural Language Generation (NLG) methods to produce a coherent summary [28]. To facilitate multilingual abstractive summarization, we experimented with two powerful multilingual pre-trained sequence-to-sequence generation models, both of which have been widely adopted for cross-lingual NLP tasks due to their robust architecture and multilingual capabilities.

mBART

mBART *(Multilingual Bidirectional and Auto-Regressive Transformer)*, proposed by Liu et al.[29], is a sequence-to-sequence model pre-trained using a multilingual denoising autoencoding objective. It extends the BART [30] architecture to the multilingual setting by jointly pre-training on large-scale monolingual corpora from multiple languages. The model employs a standard Transformer encoder-decoder structure [31], where the input text is corrupted using noise functions such as sentence permutation and token masking. The objective is to reconstruct the original sentence from the noisy version, encouraging the model to learn both syntactic structure and semantic coherence across languages [29]. Their results also demonstrate the transferability of learned representations obtained through multilingual pre-training. mBART-50 [32] advanced mBART's capabilities by expanding its language coverage from 25 to 50 languages.

mT5

The second model we consider is mT5 *(Multilingual Text-to-Text Transfer Transformer)* [33], a multilingual adaptation of the T5 model [34]. mT5 treats every NLP task as a text-to-text problem, using a unified framework that allows both the input and output to be plain text sequences. Trained on the mC4 corpus—a multilingual variant of the Common Crawl dataset mT5 has been shown to perform strongly on a variety of cross-lingual benchmarks [33].

1.2.3 Extractive-Abstractive Pipeline

One of the key difficulties in multi-document AS lies in effectively handling a large volume of input documents. To mitigate this issue, an Extractive stage is used to pre-select a subset of relevant paragraphs, thereby reducing input size before applying the AS. Although substantial advancements have been made to improve the scalability of abstractive models, many recent approaches still rely on an extractive pre-processing step for this reason [35].

1.3 Existing Datasets for CLS and MLS

Table 1.1 shows some of the available datasets for the CLS and MLS tasks. Among them, MultiLing'13 [36] and MultiLing'15 [37] are MLS datasets, containing 40 and 38 languages, respectively. CrossSum [1], a low-resource CLS dataset, is based on online news articles with 1500+ language pairs, containing 1.68M article-summary pairs. As part of the dataset creation process of Global Voices [7], two summary types were gathered:

1. brief descriptions extracted from social media platforms (referred to as *gv-snippet*), and

2. 50-word summaries manually written by humans (referred to as *gv-crowd*).

WikiMulti [38] contains ≈23K English articles. On average, each non-English language contains ≈9.64K articles that are aligned with corresponding English articles. XWikiRef [3] is a multi-document CLS dataset containing data from 5 domains (films, politicians, sportsmen, books, writers), each for 8 languages, containing ≈69K articles from Wikipedia.

Dataset	**Source**	**Languages**
XWikis [39]	Wikipedia articles	4
CrossSum [1]	BBC News	45
MLSUM [2]	Online news	6
MultiLing'13 [36]	Wikipedia articles	40
MultiLing'15 [37]	Wikipedia articles	38
WikiMulti [38]	Wikipedia articles	15
Global Voices [7]	gv-snippet and gv-crowd	15
CLIDSUM [19]	Dialogue Documents	3
WikiLingua [13]	From WikiHow [40]	18
EUR-LEX-SUM [41]	Legal acts of EU and human-generated summaries	24
NCLS [42]	Derived from pre-existing monolingual dataset	2
XWikiRef [3]	Various sources (list of URLs) covering 5 domains	8

Table 1.1 Some existing datasets for the CLS and MLS tasks

1.4 Applications of CLS

CLS has a wide range of practical applications across domains where communication in several languages and accessibility of content are crucial. Importantly, CLS breaks language barriers, enabling people to access news, information, knowledge, and more. For example, a student can read an abstract of a ground-breaking research that was published in another language. A book lover can read the abstract of a famous foreign book before buying an expensive translated copy. News portals from different countries can automatically summarize international articles. User reviews or product descriptions can be summarized for the intended language. Customer service chats or call transcripts can be summarized in a different language for analysis. For this example, a company could use the CLIDSUM [19] dataset to summarize conversations with a foreign customer. A company with branches in multiple countries can summarize its product and employee reviews in any target language. The use of the EUR-LEX-SUM dataset [41] might ensure transparency of laws for all member states of the EU.

1.5 Evaluation Methods for CLS

Various NLG metrics are used to evaluate CLS tasks. ROUGE score [43] is widely used to evaluate CLS tasks[19, 42, 2]. BERTScore is used in [19], which compares semantic similarity. LaSE, an embedding-based metric used in [1], is designed to evaluate summaries using references in a different language, making it useful for low-resource settings. METEOR [44] and chrF++[45] are also commonly used in many works across the literature [3].

In addition to these automatic metrics, many studies incorporate human evaluation to assess summary quality. For example, a 1-5 rating evaluation on the basis of accuracy, coverage, and understandability was carried out by [46, 7]. Informativeness, conciseness, and fluency are evaluated by [17, 42]. According to [47], in many cases, ChatGPT-based evaluation aligns closely with human assessments, demonstrating a high correlation in summary quality judgments.

1.5.1 Automated evaluation

In this work, we primarily use the ROUGE *(Recall-Oriented Understudy for Gisting Evaluation)* metric suite [43] for evaluation, which measures n-gram overlap between generated and reference summaries. We evaluate our system using Rouge-L, which focuses on measuring the *Longest Common Subsequence (LCS)*. Unlike ROUGE-1 or ROUGE-2, which compute overlap based on exact unigrams and bigrams, ROUGE-L captures sentence-level structure by identifying the longest sequence of words that appear in the same order (though not necessarily consecutively) in both texts, making it suitable to use for evaluating CLS tasks.

1.5.2 Evaluation using GPT-4 and Human Judgement

To complement the limitations of reference-based automatic metrics, we further evaluate a subset of outputs using OpenAI's GPT-4 [48] for qualitative judgment. In addition, we perform a human evaluation on selected test samples using

similar criteria to gain deeper insights into model strengths, weaknesses, and error patterns, especially in low-resource and cross-lingual scenarios. We also investigate if there is a correlation between human judgment and AI's judgment (LLM judgment). To assess the quality of generated summaries, we adopt four standard evaluation criteria: informativeness, faithfulness, coherence, and grammatical correctness. Below, we describe each of these criteria in detail.

- **Informativeness:** A high-quality summary should not only be factually accurate but also convey the most important content from the original text. This attribute, often referred to as informativeness, is crucial for user trust and task relevance in multilingual summarization tasks [49].
- **Faithfulness:** In neural summarization, hallucinations occur when the model produces information that is not verifiable in the original input document. These inaccuracies are especially problematic in multilingual scenarios, where faithfulness metrics are often lacking [49].
- **Coherence:** Coherence refers to the flow and arrangement of ideas in the summary. A coherent summary should present information in a structured manner that is logical and understandable [50].
- **Grammatical Correctness:** This criterion assesses if the generated summary adheres to the grammatical conventions of the target language. Well-formed sentences are expected from the model, which improve readability and reduce the cognitive load on the reader.

2 Methodology

In this chapter, we provide details on the preparation of the dataset to train CLS models and explain the architecture of the CLS, incorporating both Extractive and Abstractive methods. Moreover, we provide details on the three experimental setups of this thesis: Multidomain, Multilingual, and Multilingual-Multidomain.

2.1 Dataset selection and Preparation

We utilize subsets of the XWikiRef dataset [3]. The XWikiRef comprises Wikipedia sections across five domains and eight languages. The dataset provides cleaned section texts, extracted using MediaWikiParserFromHell, along with citation URLs filtered to retain only HTML and PDF formats. Reference content is scraped via BeautifulSoup and pdfminer, then tokenized using IndicNLP tools. Each Article includes the section title, citation URLs, and corresponding Wikipedia text. Figure 2.1 shows the data structure. Each article has multiple sections, each section having a section title, content which acts as a gold summary of that section, and a list of References which acts as input for our model. The number of sections is not the same for each article, which is why it is denoted as n number of sections.

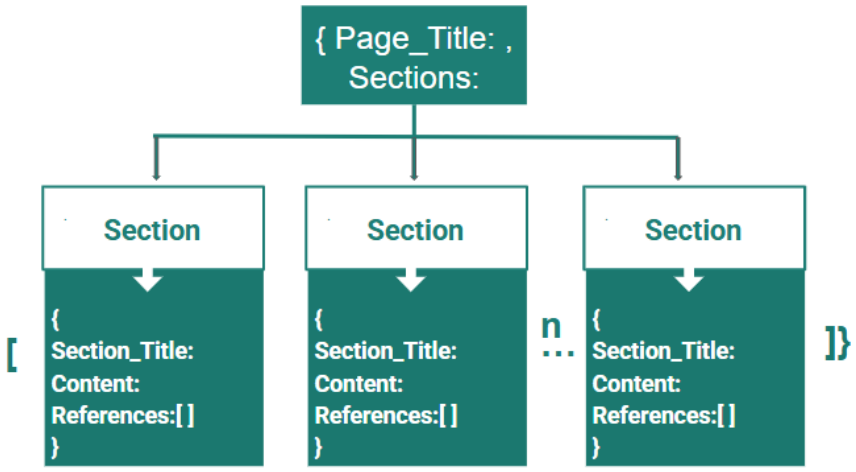

Figure 2.1 Structure of dataset

2.1.1 Language

For our experiments, we selected three languages from the XWikiRef dataset. They are Bengali, Hindi, and English. The choice of these languages was motivated by both linguistic diversity and resource availability. Bengali and Hindi are considered low-resource languages in the context of natural language processing, especially when compared to English, which has extensive digital and annotated linguistic resources. This selection allowed us to explore the challenges and capabilities of cross-lingual summarization systems in handling both high and low-resource languages.

We aimed to investigate performance variations across different cross-lingual settings, specifically focusing on English to Bengali and English to Hindi cross-lingual summarization tasks, and also English to English as a monolingual summarization task. One of the reasons for this choice is that all three of these languages belong to the Indo-European language family [51], despite their differing resource levels. In contrast, Bengali and Hindi, which belong to the Indo-Aryan language family [51], exhibit differences in script and usage patterns that add complexity to direct cross-lingual summarization. Through this setup, we aimed to evaluate how well the model generalizes across closely and distantly related language pairs under varying resource constraints.

2.1.2 Domain Selection

Initially, our goal was to include all five available domains— books, films, politicians, sportsman, and writers, for each of the three languages (Bengali, Hindi, and English). However, during preliminary data inspection and quality analysis, we encountered significant challenges with the *politicians* domain. Specifically, this domain exhibited a scarcity of sections with crawlable and non-empty reference URLs, which are essential for training and evaluating our summarization model. In many cases, the cited references were either missing, inaccessible, or pointed to irrelevant or non-textual resources such as punctuation marks. Furthermore, the available content in this domain often lacked the richness and coherence necessary for meaningful summarization, especially in the low-resource languages (Bengali and Hindi), where coverage for political figures is comparatively limited and inconsistent across language editions. Given these limitations, we excluded the *politicians* domain from our final dataset. We retained the four remaining domains—books, films, sportsman, and writers for each of the three languages.

2.1.3 Initial-20: Dataset Subsampling

To simulate a low-resource scenario and maintain experimental consistency, we decided to limit our selection to 20 instances per domain for each chosen language. This controlled sampling allowed us to assess the effectiveness of cross-lingual summarization under constrained data conditions. We created a dataset comprising the first 20 articles from each area, hereafter referred to as **Initial-20**. This yielded a smaller dataset consisting of 240 samples in total (3 languages $\times$ 4 domains $\times$ 20 Articles), providing a more reliable basis for evaluating cross-lingual summarization performance while still adhering to a low-resource setup.

Comparing tables 2.1, 2.2, and 2.3, we can see, unsurprisingly, that the English dataset exhibits higher values for average reference sentences per article, particularly in domains such as films (13,793) and sportsman (11,348). These values sharply drop in Bengali (e.g., films: 201, sportsman: 2025) and are further reduced in the Hindi corpus (films: 98, sportsman: 924). This reflects more concise source texts on the Bengali and Hindi Wikipedia pages compared to the more well-structured English versions. The number of words per reference block also varies across languages. While English maintains high average lengths (e.g., writers: 2056 words), Hindi shows surprisingly high values in the writers domain (3979 words). Bengali reference sentence lengths are more moderate, with peaks

in books (2598) and writers (3563).

Attributes	Domains			
	Books	**Films**	**Sportsman**	**Writers**
Total Sections	96	94	99	85
Empty References	37	20	7	20
Avg Ref Length (Sentences)	5,530	13,793	11,348	5,032
Avg Ref Length (words)	1,914	1,988	1,736	2,056
Avg Gold Summary Length (words)	998	882	967	1,401
Vocabulary Size	95,478	1,20,192	1,59,128	1,27,738

Table 2.1 Domain-Wise Statistics of Initial-20 (English Language)

Attribute	Domains			
	Books	**Films**	**Sportsman**	**Writers**
Total Sections	50	52	58	52
Empty References	26	27	23	22
Avg Ref Length (sentences)	387	201	2,025	1,200
Avg Ref Length (words)	2,598	866	1,781	3,563
Avg Gold Summary Length (words)	180	171	339	482
Vocabulary Size	30,907	20,021	56,864	40,983

Table 2.2 Domain-Wise Statistics of Initial-20 (Bengali Language)

Attributes	Domains			
	Books	**Films**	**Sportsman**	**Writers**
Total Sections	60	51	53	74
Empty References	33	28	18	31
Avg Ref Length (Sentences)	172	98	924	1808
Avg Ref Length (words)	1,435	832	1,638	3,979
Avg Gold Summary Length (words)	380	142	339	398
Vocabulary Size	16,355	10,583	27,386	38,599

Table 2.3 Domain-Wise Statistics of Initial-20 (Hindi Language)

Gold summary lengths follow an interesting pattern. In English, writers have the longest summaries (1402 words), whereas in Bengali and Hindi, summaries are shorter and more evenly distributed, with a peak in writers again. This suggests that writer biographies, regardless of language, tend to be more elaborate. However, the summaries in Bengali and Hindi remain significantly shorter overall. The number of empty references is fewer in English (except the Books domain) while having more sections, which indicates the data richness in the English language. On the other hand, approximately half of the references are empty in Bengali and Hindi. The domain Sportsman shows fewer empty references in all three languages.

English datasets have the highest vocabulary sizes (up to 159k) due to higher references available, and the average length of references is more than 12 times higher than in the other languages, except for the Writers domain. We also notice

that English gold summaries are almost 4 times longer than Bengali and Hindi. Bengali and Hindi show smaller vocabulary ranges (Hindi films: ∼10k, sportsman: ∼27k), likely due to smaller corpora and fewer reference pages found on Wikipedia. For this reason, the English dataset was downsampled for Multilingual (M-L) and the Multilingual-Multidomain model (ML-MD), ensuring that the model is not biased towards English.

2.1.4 Curated-20: Strategic Data Curation

We first evaluated our method on Initial-20, and then on a manually selected set of 20 higher-quality samples per pair, hereafter referred to as **Curated-20**. To improve the quality and relevance of our cross-lingual summarization experiments, we took a more hands-on approach and decided to cherry-pick examples that looked cleaner, more coherent, and overall better suited for meaningful evaluation. We performed this curation exclusively for Bengali and Hindi samples, leaving English samples unchanged from Initial-20.

During our initial exploration, we noticed that not all Wikipedia sections were equally usable. Some had a lot of noise, such as leftover wiki markup, broken or missing references, advertisements, fragmented text, or even list-based content without any narrative structure. To avoid feeding such noisy inputs into our models, we aimed to pick sections that had a clean paragraph structure and a more readable flow of information.

In choosing the data, we followed a few basic principles:

- **Clarity**: Mostly clean and understandable text without excessive wiki formatting.
- **Completeness**: Full paragraphs rather than random fragments or lists.
- **Relevance**: Content that stayed on topic and didn't contain irrelevant topics.
- **Good References**: Sections that had valid, crawlable URLs pointing to real content, not broken pages or images.
- **Domain Variety**: A mix of different topics within each domain, to avoid bias.
- **Better Content**: Section content, which works as a Gold summary of this dataset, was also noisy in some instances, containing very small or very large gold summaries. We chose those instances that seem relatively fair.

However, given the size of the original dataset, manually going through every article would have been extremely time-consuming. To keep things practical, we reviewed a manageable portion of the data and selected the 20 articles per domain and language that looked relatively cleaner, more informative, and more useful based on our judgment. This method does not guarantee the absolute best examples out of the original dataset, yet it provides a balanced and reasonably high-quality subset for efficient training and evaluation. As a result, this curated collection strikes a balance: it is not completely exhaustive, but it is clean enough to allow meaningful experimentation while reflecting the real-world challenges of working with imperfect and noisy multilingual data.

Preprocessing Curated-20

After selecting the Curated-20 dataset, pre-processing steps were applied before feeding it into the pipeline. Unicode characters, excessive whitespace were normalized. We removed the template, wiki artifacts, repeated punctuation marks, along with email addresses, URLs, website links, etc. Exact duplicate sentences, which commonly appear in Wiki-style corpora, were removed. Repetitive bigrams that appeared multiple times across the text were filtered out. These steps help in cleaning the input without losing meaningful content and provide a better input to the extractive stage, thus increasing efficiency.

Attributes	**Books**		**Films**		**Sportsman**		**Writers**	
	Raw	Cleaned	Raw	Cleaned	Raw	Cleaned	Raw	Cleaned
Total Sections	47	47	50	50	45	45	49	49
Avg Ref Length (sentences)	245	135	173	50	230	76	217	92
Avg Ref Length (words)	1,598	891	886	258	1,087	311	953	424
Avg Gold Summary Length (words)	255	236	121	116	129	121	179	170
Vocabulary Size	23,354	20,737	13,528	10,154	18,335	11,426	20,123	15,899

Table 2.4 Domain-Wise Statistics of Bengali Dataset (Raw vs Cleaned) for Curated-20

Attributes	**Books**		**Films**		**Sportsman**		**Writers**	
	Raw	Cleaned	Raw	Cleaned	Raw	Cleaned	Raw	Cleaned
Total Sections	49	49	44	44	56	56	44	44
Avg Ref Length (sentences)	100	33	39	28	2,691	314	23	17
Avg Ref Length (words)	750	257	733	417	3851	478	534	279
Avg Gold Summary Length (words)	335	294	133	119	206	187	206	191
Vocabulary Size	12,026	8,569	6,811	5,848	29,252	23,234	6,241	5,846

Table 2.5 Domain-Wise Statistics of Hindi Dataset (Raw vs Cleaned) for Curated-20

Attributes	**Books**		**Films**		**Sportsman**		**Writers**	
	Raw	Cleaned	Raw	Cleaned	Raw	Cleaned	Raw	Cleaned
Total Sections	96	94	94	94	99	99	85	85
Avg Ref Length (sentences)	5,530	3,849	13,793	3,854	11,348	2,960	5,032	1,946
Avg Ref Length (words)	1,914	1,210	1,988	617	1,736	433	2,056	864
Avg Gold Summary Length (words)	998	795	882	720	967	814	1,401	1,064
Vocabulary Size	95,478	77,775	120,192	90,600	159,128	123,948	1,27,738	108,169

Table 2.6 Domain-Wise Statistics of English Dataset (Raw vs Cleaned) for Curated-20

Comparing the properties of the raw and cleaned data in Tables 2.4, 2.5, and 2.6, we can see that the datasets had lots of redundancy even after selecting a curated dataset. The greatest amount of noise is presented in the sportsman domain in all language subsets. Even with the same number of articles, the English subset was much larger. To avoid bias towards the English language in the Multilingual and Multilingual-Multidomain settings, we had to remove some instances of English to create a balance.

2.1.5 Data Splitting Strategy

The dataset provides raw data organized by language and domain, but lacks ready-to-use splits for our experimental configurations. To address this gap and enable consistent evaluation, we constructed splits tailored to the requirements of the Multidomain (M-D), Multilingual (M-L), and Multilingual-Multidomain

(ML-MD) settings for both the Initial-20 and Curated-20 datasets. These splits were created with the output of our extractive stage.

Firstly, we split per domain-language pair using an 80:10:10 ratio for training, validation, and testing, respectively. For the M-D setup, from the per-pair split, training data across all four domains within each language were combined and shuffled to create the training set of the M-D setup. In a similar way, we created validation sets and test sets. For the M-L configuration, we followed a similar approach. We merged and shuffled training data from the per-pair split for different languages of each domain, thus creating a training set for the M-L experiment. Validation sets and test sets were created similarly. For the ML-MD setup, the training set was constructed by merging and shuffling the training portions from all M-L configurations across domains to ensure a diverse and balanced dataset. The same procedure was applied to the validation and test sets.

2.2 Summarization Framework

Tables 2.1, 2.2, and 2.3 show the average length of references of a section for each domain and language in Initial-20. We can see that the inputs are large. Therefore, we followed a two-stage summarization framework to effectively handle the challenge of long input sequences, particularly prevalent when dealing with multiple reference documents. This architecture aims to balance the efficiency of extractive techniques with the expressiveness of abstractive generation, thereby enabling high-quality summaries even under the length constraints imposed by most transformer-based models.

In the subsequent sections, we elaborate on the specific mechanisms, models, and design decisions involved in each of the two stages.

2.2.1 Extractive Summarization

For the extractive approach, we implemented a salience-based sentence selection strategy. The input for this stage was a raw per-pair dataset for the Initial-20 dataset, and for the Curated-20 dataset, it was a preprocessed per-pair input. The working process is described in Figure 2.2.

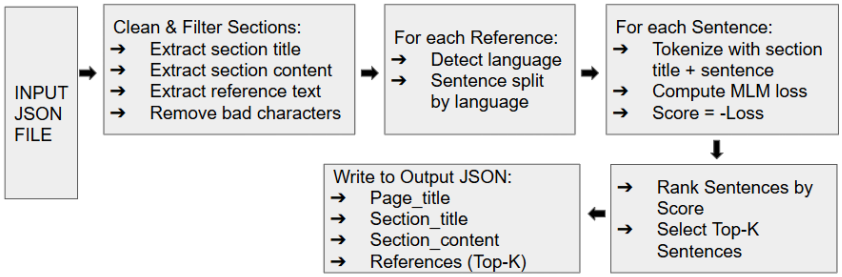

Figure 2.2 Diagram Showing the working process of Salience-Based Extractive summarization

Given a Wikipedia section in the source language and its associated reference content, we ranked the sentences in the reference based on their relevance to

the input section. The model uses a Masked language model (MLM), such as `xlm-RoBERTa-base` [27], to score how well each reference sentence aligns with a section's title and content. Each sentence is tokenized along with the section-title and then passed through the MLM to compute a negative log-likelihood loss. The lower the loss, the higher the relevance score. Top-ranked reference sentences were selected from every non-empty reference block after cleaning invalid characters. The reference is skipped if language detection or sentence splitting failed. This method required no supervised training, making it particularly suitable for low-resource scenarios. Implementing the extractive module involved integrating several third-party packages (e.g., language detection, tokenization).

The output of this stage is used to split the dataset into train, validation, and test sets.

2.2.2 Abstractive Summarization

Both mBART and mT5 offer complementary strengths. In our experiments, we treat both models as viable options and evaluate their performance across multiple language pairs and domains. The input to each model is prepared by feeding the training sets of filtered content from Stage 1 in the desired source language, and the target summary is generated in the designated output language, allowing us to assess the efficiency of each model in real-world, low-resource multilingual summarization settings.

- **mBART** The version we use contains 24 transformer layers, with 12 layers each in the encoder and decoder stacks. mBART supports a wide range of languages, including Bengali, Hindi, and English, which are central to our study. Its ability to understand and generate content in low-resource languages makes it a strong candidate for cross-lingual summarization tasks.

- **mT5** The variant used in our experiments also comprises 24 transformer layers, allowing for deep contextual understanding and powerful generation capabilities. Like mBART, mT5 is equipped to handle the three target languages in our study and can generate summaries in a linguistically diverse and semantically faithful manner.

Given the model complexity and sequence length limitations, abstractive summarization required substantial computing resources. This stage particularly stressed the system under HPC constraints, necessitating the use of smaller subsets of the dataset and the deployment of containerized environments.

Design

This stage is built as a modular pipeline to train and evaluate text summarization models using transformer architectures, such as mBART and mT5. The input data is output from the extractive stage. It starts by preparing the input data, including article titles, section headings, reference sentences, and summaries, all stored in JSON format. These data are tokenized using the appropriate language settings, and training, validation, and test sets are efficiently loaded in batches through a language-aware data module. A pretrained sequence-to-sequence model is then initialized and fine-tuned on the training data. During training, the model

learns to generate summaries by minimizing the difference between its outputs and the reference summaries. Its performance is regularly checked on the validation set using ROUGE scores. After training, the best model is used to generate summaries for the test set, and its output is compared with the reference summaries to compute the final evaluation metrics. These results, including the input text, predicted summaries, and reference summaries, are saved for qualitative and quantitative analysis. For analysis, each data sample includes a *domain ID* and a *lang ID*, used respectively for Multidomain (M-D) and Multilingual (M-L) setups, and both for the Multidomain-Multilingual (ML-MD) setup.

2.2.3 Implementation Challenges

While implementing the two-stage summarization framework comprising extractive filtering and subsequent abstractive generation, we encountered several practical difficulties that required substantial troubleshooting and adaptation. Below, we outline the key challenges and the steps taken to address them.

- **Incomplete and Poor Documentation** The publicly available source code suffered from inadequate documentation, particularly regarding system dependencies. The provided requirements file was insufficient and did not include several essential packages. During setup, we had to manually identify and install a range of additional dependencies that were not listed, including but not limited to: pycld2, morfessor, polyglot, icu, tkinter, indic-nlp-library, regex, langdetect, and sentencepiece. This significantly delayed the development process and introduced considerable friction in reproducing the original system environment.

- **Compatibility Issues with Older Package Versions** Some of the modules referenced in the codebase were outdated or incompatible with current library versions. For example, an error—ImportError: cannot import name 'Locale' from 'icu'—persisted despite installing the icu module. This was eventually resolved by using the command pip install –no-binary :all: PyICU, followed by the installation of pycld2 and morfessor. Identifying such version-specific fixes was a time-intensive task, underscoring the importance of maintaining up-to-date and well-documented codebases.

- **Computational Constraints on HPC Environment** Due to the large size of the dataset and the computational intensity of the summarization models—particularly during the abstractive generation phase—we faced several memory allocation issues when running the system on a high-performance computing (HPC) environment. To mitigate this, we pre-processed the dataset to limit it to 20 articles per domain across three languages. We deployed the system within a dedicated container to ensure better resource isolation and stability.

2.3 Experimental Settings

To systematically evaluate the performance of both extractive and abstractive cross-lingual summarization approaches, we design experiments under three

distinct configurations. These configurations are structured to test models in both controlled and generalized multilingual settings, enabling a comprehensive analysis of their adaptability and robustness.

2.3.1 Multi-Domain Setting (M-D)

This configuration explores the summarization task within multiple topics of a language simultaneously. The goal is to evaluate whether models can handle topic diversity. We construct this configuration by combining all four domains of a language, resulting in a total of **4 distinct domain-language pairs**. This results in 3 experiments with mBart and 3 experiments with mT5 for each of our datasets, resulting in a total of 12 experiments.

2.3.2 Multi-Lingual Setting (M-L)

This configuration explores the summarization task within a single domain but across multiple languages simultaneously. The goal is to evaluate whether models can handle language diversity within a controlled topical context (e.g., *films*). We construct this configuration by combining one domain for all three languages, resulting in a total of **3 distinct domain-language pairs**. This results in 4 experiments with mBart and 4 experiments with mT5 for each of our datasets, resulting in a total of 16 experiments.

2.3.3 Multilingual and Multidomain Setting (ML-MD)

In this most general and challenging configuration, both the *language* and *domain* dimensions vary simultaneously. This setup closely mirrors real-world scenarios where summarization systems are expected to operate robustly across a wide range of topics and multiple languages, often without fine-tuning for each specific combination. The goal of this setting is to evaluate the generalization capabilities of our two-stage framework when faced with diverse, noisy, and low-resource inputs.

We construct this configuration by combining each domain with each language, resulting in a total of **12 distinct domain-language pairs**, and two experiments for each of our datasets, resulting in a total of 4 experiments. This setting provides insights into each model's generalization performance and cross-domain transfer capabilities.

2.4 Experimental Setup

Table 2.7 outlines the core hyperparameters used in both the extractive and abstractive stages of our summarization framework. For the extractive stage, we employed the XLM-RoBERTa-base [27] model to encode sentence representations. To ensure computational efficiency and consistent input size for the subsequent stage, we limited the number of candidate sentences to a maximum of 50 per sample in the Initial-20 dataset. However, for Curated-20, the top 20 sentences per sample were taken.

Component	Setting	Details
Extractive Stage	Sentence Encoder	XLM-RoBERTa-base (max input: 512 tokens)
	Sentence Limit	Top-50 sentences (Initial-20), Top-20 Sentences per sample (Curated-20)
Abstractive Stage	Models	mBART (facebook/mbart-large-50), mT5 (google/mt5-base)
	Fine-tuning	10 epochs, batch size 1
	Input/Output Length	Max input: 512 tokens, max output: 256 tokens
	Optimizer	AdamW, learning rate: 1e-5
	Decoding Strategy	Greedy decoding

Table 2.7 Core hyperparameters used for extractive and abstractive summarization stages.

In the Abstractive stage, we fine-tuned two multilingual sequence-to-sequence models, mBART and mT5 using Hugging Face's `facebook/mbart-large-50`[52] and `google/mt5-base` [33] checkpoints, respectively. Both models were trained for 10 epochs with a batch size of 1, and maximum input and output lengths were fixed at 512 tokens and 256 tokens, respectively. Optimization was carried out using the AdamW optimizer with a learning rate of 1e-5. For decoding, we used a greedy strategy, where the most probable token is selected at each step without stochastic sampling or beam search. These hyperparameter settings were applied consistently across all experimental configurations to ensure comparable results. Due to the limitation of computation resources, we had to keep the batch size 1.

2.5 Evaluation Strategy

We evaluate our proposed two-stage cross-lingual summarization framework using both automated and human-centric evaluation approaches. Automated metrics described in Section 1.5.1, will be used to analyze performance of our different experimental setting. We also compare the performance of two state-of-the-art multilingual models mBART and mT5. The criteria this work follows for human evaluation and ChatGPT-based evaluation is discussed on Section 1.5.2.

2.6 Summary

We propose a two-stage cross-lingual summarization framework focused on low-resource multilingual settings. In the first stage, we apply a salience-based extractive method using XLM-RoBERTa [27] to select the most relevant sentences from citation sources. These sentences are then passed to a second, abstractive stage, where we fine-tune mBART and mT5 to generate summaries.

All the experiments were done using the Initial-20 and Curated-20 datasets, which cover three languages (Bengali, Hindi, English) and four domains (books, films, sportsman, writers).

We evaluated our framework in three settings: Multidomain, Multilingual,

and Multilingual-Multidomain. Since standard splits were unavailable, we created 80:10:10 train-validation-test partitions through merging and shuffling relevant subsets.

3 Results and Evaluation

This chapter presents the outcomes of our Cross-Language Summarization (CLS) experiments across various experimental settings. It includes a detailed analysis of model performance and an error analysis. .

3.1 Extractive Stage Evaluation

Although the extractive stage takes input in the form of articles with multiple sections, the output dataset is flattened; each input article may produce several output entries, depending on the number of sections that have non-empty references. For the Initial-20 dataset, after performing the evaluation, we noticed some concerning issues. An example of extractive output from a (Hindi, book) pair is shown. The extracted reference sentences, which serve as input to the next stage, contain mostly noise and no meaningful information. While analyzing the data, we found that some references were much longer than 50 lines. The issue stemmed from a lack of sentence separators in the input. As a result, when the salience-based extractor was asked to select the top 50 sentences, it either:

- selected all available sentences (if there were 50 or fewer), or
- selected a large block of text when sentence separators were missing.

This analysis heavily inspired the decision to build the Curated-20 dataset and also to extract the top 20 sentences per section.

Extracted Output Preview (From Initial-20 Dataset): com Books-A-Million IndieBound Find in a library All sellers Pańcatantra Pańcatantra The Book of India's Folk Wisdom The Book of India's Folk Wisdom Patrick Olivelle OUP Oxford Nov 25, 1999 Fiction 195 pages 0 ReviewsReviews aren't verified, but Google checks for and removes fake content when it's identified 0 Reviews aren't verified, but Google checks for and removes fake content when it's identified Preview this book - Introduction Introduction Note on the Translation Note on the Translation Bibliography Bibliography Guide to the Pronunciation of Sanskrit Words Guide to the Pronunciation of Sanskrit Words PAŃCATANTRA PAŃCATANTRA THE PRELUDE TO THE STORY THE PRELUDE TO THE STORY Appendices Appendices Explanatory Notes Explanatory Notes Glossary of Names Glossary of Names Index Index Copyright Copyright Pańcatantra: The Book of India's Folk Wisdom Limited preview - 1999 Limited preview - 1999 Limited preview Pańcatantra: The Book of India's Folk Wisdom Patrick OlivelleLimited preview - 2009 Patrick Olivelle Patrick Olivelle Limited preview - 2009 Limited preview

3.2 Abstractive Stage Evaluation

In this section, we present a detailed analysis of the performance of our two datasets, the Initial-20 and the Curated-20, using two different summarization

pipelines, namely Salience + mBART and Salience + mT5. The evaluation is conducted across three distinct experimental settings: Multilingual, Multidomain, and Multilingual-Multidomain. This comparative analysis aims to highlight how variations in dataset quality and structure influence model performance across different languages and domains.

3.2.1 Multidomain setting

While preparing the train-test-validation splits for the Multidomain setting using the Initial-20 dataset, we merged all domain instances within each language, followed by shuffling and splitting. However, this approach led to an unintended situation where certain domains, such as *Books* in Hindi and Bengali, were absent from the test set due to random shuffling.

To address this issue in the Curated-20 dataset, we adopted a more controlled strategy. Each domain was first individually split into train, validation, and test subsets. These were then merged across domains for each language, followed by shuffling. This method ensured balanced representation of all domains across all data splits.

Tables 3.1 and 3.2 show Multidomain performance for each language. Also, it reflects each domain's performance within this setup. For the same configuration and dataset, a comparison between mBart and mT5 is highlighted.

3.2.2 Rouge-L Based Evaluation

Pipeline	Language	Books	Writers	Films	Sportsman	W. Avg
Salience + mBART	bn	-	6.52	1.50	**10.41**	**7.70**
	hi	-	**1.43**	0	0	0.55
	en	**8.40**	7.46	7.01	6.43	7.17
Salience + mT5	bn	-	**1.44**	0	0.7	0.85
	hi	-	0.44	0.7	**1.72**	0.92
	en	**3.02**	1.63	1.62	1.44	1.92

Table 3.1 ROUGE-L F1 scores reported as percentages (%) for Multidomain Experiment for Initial-20 dataset

Pipeline	**Language**	**Books**	**Writers**	**Films**	**Sportsman**	**W. Avg**
Salience + mBART	bn	3.46	**5.43**	2.75	5.03	4.17
	hi	3.91	**14.16**	8.57	3.06	7.24
	en	4.62	6.58	**19.88**	9.09	**10.04**
Salience + mT5	bn	0	1.43	0	**3.28**	1.18
	hi	4.17	4.65	**6.14**	0.95	**3.49**
	en	3.18	0.8	**4.05**	1.57	2.4

Table 3.2 ROUGE-L F1 scores reported as percentages (%) for Multidomain Experiment for Curated-20 dataset

Overall, the Salience + mBART pipeline consistently outperformed Salience + mT5 across all languages and domains. Notably, the Curated-20 dataset

achieved higher weighted average scores for most language-domain combinations, particularly for Hindi and English, suggesting improved domain coverage and better data quality. For instance, Salience + mBART achieves a significant increase in English *Film* summaries (from 7.01% to 19.88%) and Hindi *Writers* (from 1.43% to 14.16%). One possible explanation for Bengali achieving a higher weighted average in the Initial-20 dataset is the uneven domain distribution caused by the data splitting strategy, which may have resulted in a higher proportion of better-aligned samples, such as those from the *Sportsman* domain. Since domains like *Sportsman* scored significantly higher, this likely increased the overall average for Bengali. Curated-20 enables more balanced performance across domains, highlighting the importance of controlled data splitting and domain representation.

The result of Initial-20 shows poor performance while generating. As we discussed already in the section 3.1, the issue comes from problems with extracting references which act as input to the abstraction models. Given poor input text, it is expected to produce such a prediction text.

3.2.3 Error Analysis

Here is an example from the Initial-20 dataset, a (Bengali, Films) pair generated with the mBart model:

Gold summary: অমানুষ একটি জনপ্রিয় বাংলা-হিন্দি দ্বিভাষিক সুপারহিট চলচ্চিত্র। এই ছবিটি মুক্তি পায় ১৯৭৫ সালে। এই ছবির পরিচালক ছিলেন শক্তি সামন্ত। সুরকার ছিলেন শ্যামল মিত্র। এই ছবির মুখ্য ভূমিকায় অভিনয় করেন উত্তম কুমার, শর্মিলা ঠাকুর, অনিল চট্টোপাধ্যায় এবং উৎপল দত্ত। এই চলচ্চিত্রে বাংলা এবং হিন্দিতে গাওয়া কিশোর কুমারের গানগুলি খুবই জনপ্রিয় হয়েছিল তখনকার দিনে। ছবিটির মূল কাহিনীকার শক্তিপদ রাজগুরু। তিনি দীর্ঘদিন সুন্দরবনের বাদাবন এলাকায় ছিলেন। সেখানকার মানুষের দৈনন্দিন জীবনযাপনকে তুলে ধরেছিলেন তার নয়া বসত উপন্যাসে। সেই উপন্যাস থেকেই শক্তি সামন্ত এই ছবি তৈরি করেন। ছবিটি বাংলা এবং হিন্দি দুটি ভাষায় আলাদা করে তৈরি হয়। দুটো চলচ্চিত্রই ভাল সফল হয়েছিল। ছবিটি এদুরিতা নামে তেলুগু ভাষাতেও রিমেক হয়েছিল। নায়কের ভূমিকায় ছিলেন বিখ্যাত এন টি রামা রাও।

Prediction text: অমানুষ (চলচ্চিত্র) Introduction Try the new Google Books Routledge Rediff Books Flipkart Infibeam All sellers Television in India Television in India Satellite television identity and globalisation in contemporary India A television producers view on cricket and satellite TV in contemporary India A television producers view on cricket and satellite TV in contemporary India Inserting TV into the transforming text of post1980 Bengali cinema

In this example, the Prediction text, which was expected to generate the Bengali summary, could only generate the page title, which was present in the input text, demonstrating an uninformative output. There was one more concerning issue we noticed from our ROUGE-based evaluation. We observed that ROUGE-1 and ROUGE-L scores were nearly identical across maximum predictions. Upon investigation, we found that this similarity was not due to an error in computation but rather due to the fact that the model generated very few words which were present in the gold summary. Furthermore, most of the correctly predicted words were proper nouns, such as a writer's name, a city name, or a film name (as in

this case). For example, if the model generates a first and last name correctly (e.g., Albert Einstein), both ROUGE-1 and ROUGE-L would report a match count of 2, ultimately yielding the same F1 score.

Since we got a comparatively better score for the (Bengali, sportsman) pair, we decided to investigate this further.

Gold Summary: ১০ ফেব্রুয়ারি, ১৯৯৬ সালে ফিলাডেলফিয়া, পেনসিলভানিয়ায় শুরু হওয়া যন্ত্র বনাম মানুষের মধ্যকার আনুষ্ঠানিক খেলা শুরু হয়। এর মাধ্যমেই ডীপ ব্লু প্রথমবারের মতো ঐ সময়ের বিশ্ব চ্যাম্পিয়ন ও শীর্ষস্থানীয় ইলো রেটিংধারী দাবাড়ু হিসেবে গ্যারি কাসপারভের বিরুদ্ধে প্রতিযোগিতায় অবতীর্ণ হয়। এতে সাদা ঘুঁটি ও সিসিলিয়ান ডিফেন্স (বি২২) ১ম খেলায়ই এটি নিয়মিতভাবে সময় নিয়ন্ত্রণ করে ডীপ ব্লু। ফলে গ্যারী কাসপারভ পরাভূত হন। পরবর্তীতে বাকী ৫টি খেলায় কাসপারভ নিজের নিয়ন্ত্রণে এনে নেন। তিনি ৩টিতে জয়ী এবং বাকী ২টিতে ড্র করেন। সামগ্রীকভাবে ফলাফল ছিল কাসপারভ (৪) - ডীপ ব্লু (২)। (দাবায় জয়ী হলে ১ পয়েন্ট, ড্র ১/২ পয়েন্ট এবং পরাজিত ০ পয়েন্ট হিসেবে নিরূপণ করা হয়।) এর মাধ্যমে নিঃসন্দেহে তিনি যন্ত্রচালিত দাবা কম্পিউটার হিসেবে ডিপ ব্লু....

Predicted Summary: গ্যারি কাসপারভ বনাম ডিপ ব্লু Watson demonstrated that a whole new generation of human - machine interactions will be possible. The technology in Watson was a substantial step forward from Deep Blue and earlier machines because it....

The page title was a long sequence, and that was generated by the predicted summary, leading to a better score. Although the input in this example was clearer and more semantically meaningful than the previous case, the model failed to perform the CLS task. Despite Bengali being the target language, the model generated a summary in English. Notably, the content of the generated summary was factually accurate and aligned with the source text. This observation suggests that while the model is capable of producing faithful summaries when the input is well-structured, it struggles with adhering to the designated output language in cross-lingual scenarios.

3.2.4 Multilingual Setting

In the extractive stage, we provided 20 articles as input to each pair. However, this stage flattens the articles into contained sections. Since the English subset contains a higher number of sections with non-empty references, it resulted in the English subset becoming nearly three times larger than the other languages. To ensure balance in the multilingual setting, where each domain is evaluated across three languages, we subsampled the English subset. This helped mitigate dataset bias and made the experiments more computationally manageable under limited memory constraints. For example, in the training set of the domain Sportsman of the Initial-20 dataset, there were 82 instances from English, while Hindi and Bengali had 29 and 21, respectively. We subsampled the dataset so that each language had 20 instances.

3.2.5 Rouge-L Based Evaluation

Pipeline	Domain	Bengali	Hindi	English	W.Avg	Overall
Salience + mBART	Books	5.04	4.76	**11.67**	**7.80**	**5.20**
	Writers	0	1.92	**6.11**	2.77	
	Films	0	5.38	**6.37**	4.98	
	Sportsman	0.79	6.14	**10.23**	5.67	
Salience + mT5	Books	0	**2.50**	1.73	1.46	**0.54**
	Writers	0	0	**0.67**	0.25	
	Films	**1.04**	0	0.51	0.43	
	Sportsman	0	0	**0.58**	0.22	

Table 3.3 ROUGE-L F1 scores reported as percentages (%) for Multilingual Experiment for Initial-20 dataset

Pipeline	Domain	Bengali	Hindi	English	W.Avg	Overall
Salience + mBART	Books	4.18	3.38	**5.42**	4.31	**5.40**
	Writers	0.69	**8.72**	6.62	5.18	
	Films	0	8.62	**20.57**	**8.34**	
	Sportsman	2.17	3.20	**9.12**	4.17	
Salience + mT5	Books	0	**4.17**	0.80	1.41	**0.89**
	Writers	0	**2.01**	1.74	**1.45**	
	Films	0	0	**2.29**	0.65	
	Sportsman	0	0	**0.74**	0.22	

Table 3.4 ROUGE-L F1 scores reported as percentages (%) for Multilingual Experiment for Curated-20 dataset

Salience + mBART achieves relatively consistent performance across both datasets, with a slight increase in the weighted average (W.Avg) from 5.20% to 5.40%, suggesting that the Curated-20 dataset may have supported slightly more stable outputs. Salience + mT5 shows more noticeable improvement from 0.54% to 0.89%—in the Curated-20 dataset, indicating that data curation positively influences mT5, which is more sensitive to training signal quality.

Language-wise, English summaries performed best, especially in the Films domain, where the Salience + mBART pipeline achieved a notable increase from 6.37% to 20.57% in Curated-20. Hindi also improved, particularly in the Writers domain. However, Bengali performance remained low across both datasets and pipelines, highlighting the continued challenge of summarization in low-resource settings.

In summary, Curated-20 provided more stable and improved results, particularly for mT5 and in domains with well-structured input, demonstrating the importance of dataset quality in multilingual summarization.

3.2.6 Error Analysis

For the same test example mentioned earlier, corresponding to the (Bengali, sportsman) pair, the summary generated by the mBART model in the Multilingual setting is shown below:

Predicted Summary: IBM. Deep Blue Blue Blue.

The poor performance of mT5 suggests that the model may struggle more to handle the same language summarization tasks effectively when trained in a multilingual setting, even within a single domain. Here is an example from the (English, Films) pair under the mT5 Multilingual configuration. Despite being an English-to-English summarization task, the model failed to generate a coherent output, highlighting the limitations of multilingual training when not guided by a specific language.

Gold Summary: "spinner" (police variant) on display at Disney-MGM Studios in the 1990s "Spinner" is the generic term for the fictional flying cars used in the film. A spinner can be driven as a ground-based vehicle, and take off vertically, hover, and cruise much like vertical take-off and landing (VTOL) aircraft. They are used extensively by the police as patrol cars, and wealthy people can also acquire spinner licenses. The vehicle was conceived and designed by Syd Mead who described the spinner as an aerodyne – a vehicle which directs air downward to create lift, though press kits for the film stated that the spinner was propelled by three engines: "conventional internal combustion, jet, and anti-gravity". A spinner is on permanent exhibit at the Science Fiction Museum and Hall of Fame in Seattle, Washington. Mead's conceptual drawings were transformed into 25 vehicles by automobile customizer Gene Winfield; at least two were working ground vehicles, while others...
Predicted Summary: a sampling machine

3.2.7 Multilingual-Multidomain Setting

We assembled all the training data from each language-domain pair to create a training set for our ML-MD setting and shuffled it. Similar procedures were done with validation and test sets. This way, we ensure data is trained and tested with every language-domain pair.

Table 3.5 and 3.6 show that for the Salience + mBART pipeline, the Initial-20 set yields slightly higher overall scores. However, Curated-20 shows more consistent gains in low-resource settings, especially with mT5. For Salience + mT5, Curated-20 performs substantially better, with a sixfold increase (from 0.15% to 0.89%), suggesting that cleaner and more aligned data has an impact on mT5's ability to generalise.

- Domain-Wise Comparison: In both datasets, the performance of the Books domain in Bengali remains near zero, indicating persistent difficulty in this language-domain pairing. For the Books domain, Hindi and English scores in Curated-20 with mT5 (4.17% and 0.77%) show improvement over zeros

Pipeline	Domain	Bengali	Hindi	English	W.Avg	Overall
Salience + mBART	Books	0	8.25	**9.94**	6.06	**6.83**
	Writers	0	**15.38**	10.6	**8.66**	
	Films	0	**14.46**	2.85	6.85	
	Sportsman	2.38	2.81	**15.56**	6.91	
Salience + mT5	Books	0	0	0	0	0.15
	Writers	0	0	0	0	
	Films	0	0	0	0	
	Sportsman	0	0	**1.75**	**0.58**	

Table 3.5 ROUGE-L F1 scores reported as percentages (%) for Multilingual-Multidomain Setting for Initial-20 dataset

Pipeline	Domain	Bengali	Hindi	English	W.Avg	Overall
Salience + mBART	Books	1.55	3.92	**4.62**	3.10	**5.64**
	Writers	0.72	**13.61**	6.62	**7.03**	
	Films	0	**8.62**	**20.57**	**8.34**	
	Sportsman	2.16	3.42	**9.12**	4.27	
Salience + mT5	Books	0	**4.17**	0.77	1.41	0.89
	Writers	0.72	**2.01**	1.74	**1.46**	
	Films	0	0	**2.30**	0.65	
	Sportsman	0	0	**0.74**	0.16	

Table 3.6 ROUGE-L F1 scores reported as percentages (%) for Multilingual-Multidomain Setting for Curated-20 dataset

in Initial-20, demonstrating better cross-lingual adaptation due to more curated samples. For the Writers domain, Salience + mBART shows a significant drop from 15.38% (Initial-20) to 13.61% (Curated-20) for Hindi, but this drop is offset by mT5's improvement, rising from zero to 2.01%. For the Films domain, English summarization is stronger in Curated-20 with mBART (20.57%) compared to (2.85%) in Initial-20, reflecting a major gain. mT5 also improved from zero to 2.30%.

- Language-Wise Comparison: In every domain, for Bengali, the ROUGE-L score is low or zero across both datasets and pipelines, indicating a strong need for improved Bengali training data. The Hindi subset of Curated-20 sees improvements for mT5 across domains, with better token alignment and possibly clearer structure in curated samples. Also, a joint training with domain and language diversity improved the performance of Hindi. The English subset is stronger in mBART across both, but Curated-20 allows mT5 to generate at least minimally workable summaries, where it was non-functional before.

Curated-20 reduces noise and inconsistencies, which is beneficial for transformer-based multilingual transfer. It supports better generalization in low-resource conditions.

3.3 Gpt-4 Based Evaluation

3.3.1 Setup

To assess the quality of system-generated summaries, OpenAI's GPT-4 [48] via the ChatGPT interface was used for comparison of summaries. We evaluated a total of 10 samples using GPT-4 from different experimental settings and our two datasets. For each test case, the following was provided to GPT-4:

- A reference summary: Considered as the gold standard,
- A system-generated summary: output from the pipeline under evaluation,
- A standardized evaluation prompt with criteria explained.

The prompt was repeated consistently across examples to ensure fair comparison. An example of the prompt used is:

> Prompt: Here are two documents related to a summarization task. The first is a reference summary, and the second is the system output. Can you please compare the two documents and rate the system output on the basis of the reference output?
> Criteria:
> Coherence (logical flow and structure),
> Informativeness (coverage of important content),
> Grammatical Correctness (language fluency and correctness)
> Faithfulness (Not generating information that is not present in the input)
>
> Please provide a score from 1 (very poor) to 5 (excellent) for each category, and a brief explanation for each score.

The summaries are inserted clearly with the tags of *Reference summary* and *System Output*.

Ethical Consideration

All the interactions were carried out in a neutral tone with a consistent approach to avoid directing the model. The evaluation was conducted using a paid subscription, under the condition that the data would not be used for training purposes.

3.3.2 Evaluation

In Table 3.7, the first three system outputs are evaluated from the Initial-20 dataset, and the rest are from the Curated-20 dataset. The evaluation shows that almost all the instances lack coherence as well as in delivering information. The best score is seen for a domain-language pair from films-English, from the Multidomain experiment. It has achieved an excellent rating for faithfulness, and a very good rating for Coherence, Informativeness, and Grammatical Correctness. The score for Initial-20 appears to be poor throughout.

Id	Pipeline	Setting	Pairs	Coherence	Informativeness	Correctness	Faithfulness
1	Salience+ mBART	ML-MD	(bn,books)	1	1	1	1
2	Salience+ mBART	ML-MD	(en,books)	1	1	2	1
3	Salience+ mBART	M-D	(bn,sportsman)	1	1	1	1
4	Salience+ mBART	M-L	(bn,sportsman)	2	2	3	2
5	Salience+ mBART	M-D	(en,films)	4	4	4	5
6	Salience+ mBART	M-D	(hi,films)	1	1	1	1
7	Salience+ mT5	ML-MD	(bn,books)	1	1	1	1
8	Salience+ mT5	ML-MD	(hi,books)	1	1	3	2
9	Salience+ mBART	ML-MD	(hi,writers)	2	1	2	2
10	Salience+ mBART	ML-MD	(en,writers)	1	1	2	1

Table 3.7 GPT-4 evaluation results for different pipelines and settings across datasets.

3.4 Human Evaluation

A human evaluation was done with 5 samples from different settings. The evaluation only focused on Bengali and English instances, ensuring diversity while remaining within the evaluator's proficiency. These five examples were also part of the ten samples we later gave to GPT-4 for evaluation, which are recognizable by their ID numbers. We only chose mBart-generated summaries as most of the text generated by mT5 was too short and fragmented to be meaningful.

Id	Setting	Pairs	Coherence	Informativeness	Correctness	Faithfulness
1	ML-MD	(bn,books)	1	1	1	1
3	M-D	(bn,sportsman)	1	2	2	3
4	M-L	(bn,sportsman)	2	2	2	3
5	M-D	(en,films)	3	3	4	5
10	ML-MD	(en,writers)	1	1	2	1

Table 3.8 Human evaluation results for different pipelines and settings across datasets.

The first two samples in Table 3.8 are drawn from the Initial-20 dataset, while the remaining examples are from the Curated-20 dataset.

Ethical Consideration

The human evaluation was performed prior to GPT-4, so that the AI's output would not influence the reviewers' opinions.

3.4.1 Correlation between GPT-4 and Human assessments

When compared with GPT-4's assessments, a general agreement can be observed, indicating a positive correlation between human and AI-based evaluations. GPT-4 demonstrated strong performance even in non-English languages, which is particularly valuable given that human reviewers may have limitations in their proficiency across all target languages. This suggests that large language models like GPT-4 can serve as reliable complementary evaluators, especially in multilingual evaluation settings.

However, with ID numbers 3, 4, and 5 in Tables 3.7 and 3.8, there is a noticeable divergence between GPT-4 and human evaluations. The primary reason for disagreement in Informativeness and Faithfulness appears to be that GPT-4 did not have access to the original input text. As a result, it relied on the reference summary, which may have omitted certain details present in the source.

Consequently, GPT-4 classified some of these valid details as hallucinations, despite their presence in the input. Additionally, in some cases, residual noise remained in the input even after preprocessing, which may have influenced the model's outputs.

The disagreement for ID 4 under Grammatical Correctness can be attributed to the language mismatch. Although the target language was Bengali, the model generated a summary that was mostly in English. While the structure of the English output was mostly correct, the human evaluator focused on the appropriateness of the target language and penalized the summary accordingly for not standing up to Bengali grammatical standards.

3.5 Key Findings

From the analysis, we can summarize our key findings as follows:

- **mBART outperformed mT5:** Across most language-domain pairs, mBART [29] consistently achieved higher ROUGE scores than mT5 [33], suggesting its stronger capacity for multilingual summarization, especially when pretrained with language-specific tokens. mT5-generated texts were minimal and mostly noise. It performed worse in an M-L situation, suggesting it has limitations in low-resource conditions. However, with the introduction of the cleaner and more focused Curated-20 dataset, mT5 exhibited improved performance, especially in Hindi, notably within the books and writers domains. Curated-20 shows more meaningful gains for mT5, the more sensitive and data-dependent model. This indicates that higher-quality input-output pairs in Curated-20 improve the training signal. Despite this improvement, the model consistently failed to generate coherent outputs in Bengali, indicating that mT5 remains highly sensitive to language resource availability and benefits significantly from curated, high-quality data.

- **Joint training boosts generalization:** Our ML-MD combined setting shows better performance than the M-D and M-L settings. The unification of language and domain diversity in one training setup helped the model to leverage Cross-Lingual and Cross-Domain harmony more efficiently. The improvement means the model is learning patterns that work well across languages and various subjects. By seeing more variety in the data, the model gains a deeper understanding and learns robust and transferable features.

- **Resource availability impacts performance:** The results demonstrate that high-resource languages, such as English, and to a certain extent Hindi, perform reasonably well even with limited training data. In contrast, Bengali, being a lower-resourced language, exhibits significantly lower performance under similar conditions, highlighting the disparity caused by poor resources. However, it shows a comparatively better performance with the Curated-20 dataset than Initial-20, suggesting that data quality, consistency, and domain selection can reduce resource limitations. The structured and noise-reduced nature of Curated-20 appears to help the model generalize better, even with

fewer samples, compared to the noisier and more heterogeneous Initial-20 dataset. This observation highlights that strategic curation of multilingual datasets can play a crucial role in uplifting the performance of languages like Hindi and narrowing the performance gap for very low-resource languages like Bengali, especially in complex Multilingual-Multidomain summarization pipelines.

- **Domain performance:** The performance in every experimental setup with Curated-20 shows superiority of the films and the writers domain. Domain books show an overall poor performance, especially in the ML-MD model. These observations highlight that while certain domains benefit from shared training (e.g., films and writers), others, like books, may require domain-specific modeling or additional pretraining to improve summarization quality in multilingual, multidomain contexts.

- **Correlation of Human and GPT-4 evaluations:** We analyze how closely the automatic assessments made by GPT-4 align with human judgments across multiple qualitative aspects such as informativeness, faithfulness, grammatical correctness, and coherence. Overall, there was a high level of similarity between the automated and human assessments. However, some disagreement was observed in the dimension of faithfulness, where in some cases, GPT-4 underestimated the factual consistency of generated summaries compared to human evaluators. One important factor contributing to this disagreement is likely the absence of the source input text during GPT-4's evaluation, which limits its ability to detect unsupported or hallucinated content.

- **Repetition in Generated Summary:** The generated summaries sometimes repeated words, punctuation marks, or even whole sentences, which disrupted their coherence and grammatical correctness.

- **Hallucination:** Hallucination, a common issue in text generation. It was minimal for mBART. As defined by [49], it happens when the model includes information that cannot be confirmed from the input text.

- **Language Mismatch** In most cases, the generated cross-lingual summaries did not effectively capture the source information. However, summaries in the English subset showed promising results. In some instances, the model produced factually correct summaries in English, even when the target language was different.

4 Conclusion

In this thesis, we analyzed the Cross-Language Multi-Document Summarization task for low-resource languages. We created two subsets of the dataset from a publicly available dataset, XWikiRef [3], which contains data of five distinct subjects of eight languages. We selected Bengali, Hindi, and English data for our work. The language choice was made motivated by both topologically linguistic diversity and differences in their resource availability. Bengali and Hindi are considered low-resource languages in the context of natural language processing, especially when compared to English, which has extensive digital and annotated linguistic resources. In comparison to Hindi, Bengali is a lower-resource language. All three languages belong to the Indo-European family, and both the low-resource languages belong to the Indo-Aryan family, and all three languages have their own script. Among the five domains in XWikiref[3], we selected to work with four, such as *Books*, *Writers*, *Sportsman*, *Films*.

We first created a subset named `Initial-20`, which comprises the very first 20 Articles of each domain-language pair from the original dataset, each article having multiple sections. We then, with strategic data curation, selected 20 samples for each domain-language pair, with each article comprising multiple sections, and created `Curated-20`. Curated-20 was preprocessed before being used in the two-stage framework.

While performing the Extractive stage with the Initial-20 dataset, the model chose the top 50 salient contents per input section. following [3]. In this case, the model selected all available sentences if there were 50 or fewer in the input. For that, we decided to extract the top 20 sentences for each section while working with the Curated-20 dataset.

Using the output from the Extractive stage, we created our train-val-test split. Then we trained and evaluated the abstractive stage using mBart and mT5 in three different experimental settings, such as Multidomain (M-D), Multilingual (M-L), and Multilingual-Multidomain (ML-MD). In total, 16 experiments were done for each dataset.

The result analysis suggests that, across most language-domain pairs, mBART consistently outperformed mT5, suggesting its stronger capacity for multilingual summarization, especially when pretrained with language-specific tokens. A significant improvement was noticed in mT5 performance with the Curated-20 dataset, suggesting its sensitivity. Results show that Multilingual training boosted the performance of Hindi, suggesting that transfer learning helped Hindi to some extent, while Bengali significantly showed poor performance. Both datasets exhibit a better performance in the ML-MD experimental setup, where all of the 12 domain-language pairs were trained together. The unification of language and domain diversity in one training setup helped the model to leverage Cross-Lingual and Cross-Domain harmony more efficiently. Although there was a correlation between the automated and human assessments, some disagreement was observed

in the dimension of faithfulness. The reason behind that might be the absence of the source input text during GPT-4's evaluation, which limits its ability to detect hallucinated content.

4.1 Limitations

The data and resource scarcity of languages like Bengali and Hindi were our biggest limitations. The dataset was heavily noisy, full of repeated lines, advertisements. Even with a curative approach and preprocessing, the input to the Abstractive model was not of very good quality, which impacts training.

We have done a total of 32 experiments using either `mbart-large-50` [52] or `mT5` [33], which are large-scale Multilingual transformers that demand substantial memory resources, posing GPU memory constraints during training and inference, especially with longer inputs. This is one of the reasons why we had to choose 20 articles per domain-language pair. We were required to restrict the batch size to 1, which impacted training efficiency. Furthermore, limitations in computational resources prevented us from performing an extensive hyperparameter tuning.

4.2 Future Work

Our future work will be focused on expanding the number of languages, particularly those with extremely limited resources, to help evaluate the generalization ability of the framework at a larger scale. Introduction of new scripts with morphological variations would further challenge the system and reveal new insights.

We can incorporate an intermediate planning step suggested by [50]. Incorporating an entity-based intermediate step across source–target pairs may enhance the abstraction summarization process.

We would like to evaluate and compare more samples with human evaluation and ChatGPT-based evaluation. To evaluate the faithfulness of our Bengali and Hindi generated summaries, `mFact` [49] can be used to detect hallucinated content.

Bibliography

1. Bhattacharjee, Abhik; Hasan, Tahmid; Ahmad, Wasi Uddin; Li, Yuan-Fang; Kang, Yong-Bin; Shahriyar, Rifat. CrossSum: Beyond English-centric cross-lingual summarization for 1,500+ language pairs. *arXiv preprint arXiv:2112.08804*. 2021.
2. Scialom, Thomas; Dray, Paul-Alexis; Lamprier, Sylvain; Piwowarski, Benjamin; Staiano, Jacopo. MLSUM: The multilingual summarization corpus. *arXiv preprint arXiv:2004.14900*. 2020.
3. Taunk, Dhaval; Sagare, Shivprasad; Patil, Anupam; Subramanian, Shivansh; Gupta, Manish; Varma, Vasudeva. Xwikigen: Cross-lingual summarization for encyclopedic text generation in low resource languages. In: *Proceedings of the ACM Web Conference 2023*. 2023, pp. 1703–1713.
4. Wang, Guanghua; Wu, Weili. Surveying the landscape of text summarization with deep learning: A comprehensive review. *Discrete Mathematics, Algorithms and Applications*. 2024, vol. 16, no. 03, p. 2330004.
5. Wang, Jiaan; Meng, Fandong; Zheng, Duo; Liang, Yunlong; Li, Zhixu; Qu, Jianfeng; Zhou, Jie. Towards unifying multi-lingual and cross-lingual summarization. *arXiv preprint arXiv:2305.09220*. 2023.
6. Wang, Liangming; Wan, Xiaojun; Wan, Xiaoxue; Li, Lei; Zhang, Min-Yen. A Survey on Cross-Lingual Summarization. *Transactions of the Association for Computational Linguistics*. 2022, vol. 10, pp. 171–189.
7. Nguyen, Khanh; Daumé III, Hal. Global Voices: Crossing borders in automatic news summarization. *arXiv preprint arXiv:1910.00421*. 2019.
8. Leuski, Anton; Lin, Chin-Yew; Zhou, Liang; Germann, Ulrich; Och, Franz Josef; Hovy, Eduard. Cross-lingual c* st* rd: English access to hindi information. *ACM Transactions on Asian Language Information Processing (TALIP)*. 2003, vol. 2, no. 3, pp. 245–269.
9. Wan, Xiaojun. Using bilingual information for cross-language document summarization. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, pp. 1546–1555.
10. Boudin, Florian; Huet, Stéphane; Torres-Moreno, Juan-Manuel. A graph-based approach to cross-language multi-document summarization. *Polibits*. 2011, no. 43, pp. 113–118.
11. Orăsan, C; Chiorean, Oana Andreea. Evaluation of a cross-lingual Romanian-English multi-document summariser. 2008.
12. Wan, Xiaojun; Li, Huiying; Xiao, Jianguo. Cross-language document summarization based on machine translation quality prediction. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. 2010, pp. 917–926.
13. Ladhak, Faisal; Durmus, Esin; Cardie, Claire; McKeown, Kathleen. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. *arXiv preprint arXiv:2010.03093*. 2020.

14. Cao, Yue; Liu, Hui; Wan, Xiaojun. Jointly Learning to Align and Summarize for Neural Cross-Lingual Summarization. In: Jurafsky, Dan; Chai, Joyce; Schluter, Natalie; Tetreault, Joel (eds.). *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 6220–6231. Available from doi: 10.18653/v1/2020.acl-main.554.

15. Bai, Yu; Huang, Heyan; Fan, Kai; Gao, Yang; Zhu, Yiming; Zhan, Jiaao; Chi, Zewen; Chen, Boxing. Unifying cross-lingual summarization and machine translation with compression rate. In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2022, pp. 1087–1097.

16. Nguyen, Thong Thanh; Luu, Anh Tuan. Improving neural cross-lingual abstractive summarization via employing optimal transport distance for knowledge distillation. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2022, vol. 36, pp. 11103–11111. No. 10.

17. Zhu, Junnan; Zhou, Yu; Zhang, Jiajun; Zong, Chengqing. Attend, translate and summarize: An efficient method for neural cross-lingual summarization. In: *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*. 2020, pp. 1309–1321.

18. Xu, Ruochen; Zhu, Chenguang; Shi, Yu; Zeng, Michael; Huang, Xuedong. Mixed-lingual pre-training for cross-lingual summarization. *arXiv preprint arXiv:2010.08892*. 2020.

19. Wang, Jiaan; Meng, Fandong; Lu, Ziyao; Zheng, Duo; Li, Zhixu; Qu, Jianfeng; Zhou, Jie. Clidsum: A benchmark dataset for cross-lingual dialogue summarization. *arXiv preprint arXiv:2202.05599*. 2022.

20. Wang, Jiaan; Meng, Fandong; Zheng, Duo; Liang, Yunlong; Li, Zhixu; Qu, Jianfeng; Zhou, Jie. A Survey on Cross-Lingual Summarization. *Transactions of the Association for Computational Linguistics*. 2022, vol. 10, pp. 1304–1323. issn 2307-387X. Available from doi: 10.1162/tacl_a_00520.

21. Cheng, Jianpeng; Lapata, Mirella. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*. 2016.

22. Radev, Dragomir R; Allison, Timothy; Blair-Goldensohn, Sasha; Blitzer, John; Celebi, Arda; Dimitrov, Stanko; Drabek, Elliott; Hakim, Ali; Lam, Wai; Liu, Danyu, et al. MEAD-a platform for multidocument multilingual text summarization. 2004.

23. Filatova, Elena; Hatzivassiloglou, Vasileios. Event-based extractive summarization. 2004.

24. Nenkova, Ani; Vanderwende, Lucy; McKeown, Kathleen. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 2006, pp. 573–580.

25. Cao, Ziqiang; Wei, Furu; Li, Sujian; Li, Wenjie; Zhou, Ming; Wang, Houfeng. Learning summary prior representation for extractive summarization. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 2015, pp. 829–833.

26. Cao, Ziqiang; Wei, Furu; Dong, Li; Li, Sujian; Zhou, Ming. Ranking with recursive neural networks and its application to multi-document summarization. In: *Proceedings of the AAAI conference on artificial intelligence*. 2015, vol. 29. No. 1.

27. Conneau, Alexis; Khandelwal, Kartikay; Goyal, Naman; Chaudhary, Vishrav; Wenzek, Guillaume; Guzmán, Francisco; Grave, Edouard; Ott, Myle; Zettlemoyer, Luke; Stoyanov, Veselin. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*. 2019.

28. Lin, Hui; Ng, Vincent. Abstractive summarization: A survey of the state of the art. In: *Proceedings of the AAAI conference on artificial intelligence*. 2019, vol. 33, pp. 9815–9822. No. 01.

29. Liu, Yinhan; Gu, Jiatao; Goyal, Naman; Li, Xian; Edunov, Sergey; Ghazvininejad, Marjan; Lewis, Mike; Zettlemoyer, Luke. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*. 2020, vol. 8, pp. 726–742.

30. Lewis, Mike; Liu, Yinhan; Goyal, Naman; Ghazvininejad, Marjan; Mohamed, Abdelrahman; Levy, Omer; Stoyanov, Ves; Zettlemoyer, Luke. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*. 2019.

31. Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N; Kaiser, Łukasz; Polosukhin, Illia. Attention is all you need. *Advances in neural information processing systems*. 2017, vol. 30.

32. Tang, Yuqing; Tran, Chau; Li, Xian; Chen, Peng-Jen; Goyal, Naman; Chaudhary, Vishrav; Gu, Jiatao; Fan, Angela. Multilingual Translation from Denoising Pre-training. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, 2021, pp. 3450–3466. Available from doi: 10.18653/v1/2021.findings-acl.304.

33. Xue, Linting; Constant, Noah; Roberts, Adam; Kale, Mihir; Al-Rfou, Rami; Siddhant, Aditya; Barua, Aditya; Raffel, Colin. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*. 2020.

34. Raffel, Colin; Shazeer, Noam; Roberts, Adam; Lee, Katherine; Narang, Sharan; Matena, Michael; Zhou, Yanqi; Li, Wei; Liu, Peter J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*. 2020, vol. 21, no. 140, pp. 1–67.

35. Beltagy, Iz; Peters, Matthew E; Cohan, Arman. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*. 2020.

36. Steinberger, Josef. The UWB summariser at multiling-2013. In: *Proceedings of the MultiLing 2013 workshop on multilingual multi-document summarization*. 2013, pp. 50–54.

37. Giannakopoulos, George; Kubina, Jeff; Conroy, John; Steinberger, Josef; Favre, Benoit; Kabadjov, Mijail; Kruschwitz, Udo; Poesio, Massimo. Multiling 2015: multilingual summarization of single and multi-documents, on-line fora, and call-center conversations. In: *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 2015, pp. 270–274.

38. Tikhonov, Pavel; Malykh, Valentin. Wikimulti: a corpus for cross-lingual summarization. In: *Conference on Artificial Intelligence and Natural Language*. Springer, 2022, pp. 60–69.

39. Perez-Beltrachini, Laura; Lapata, Mirella. Models and datasets for cross-lingual summarisation. *arXiv preprint arXiv:2202.09583*. 2022.

40. Koupaee, Mahnaz; Wang, William Yang. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*. 2018.

41. Aumiller, Dennis; Chouhan, Ashish; Gertz, Michael. EUR-lex-sum: A multi-and cross-lingual dataset for long-form summarization in the legal domain. *arXiv preprint arXiv:2210.13448*. 2022.

42. Zhu, Junnan; Wang, Qian; Wang, Yining; Zhou, Yu; Zhang, Jiajun; Wang, Shaonan; Zong, Chengqing. NCLS: Neural cross-lingual summarization. *arXiv preprint arXiv:1909.00156*. 2019.

43. Lin, Chin-Yew; Och, Franz Josef. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In: *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)*. 2004, pp. 605–612.

44. Banerjee, Satanjeev; Lavie, Alon. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In: Goldstein, Jade; Lavie, Alon; Lin, Chin-Yew; Voss, Clare (eds.). *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, 2005, pp. 65–72. Available also from: `https://aclanthology.org/W05-0909/`.

45. Popović, Maja. chrF++: words helping character n-grams. In: *Proceedings of the second conference on machine translation*. 2017, pp. 612–618.

46. Wang, Jiaan; Liang, Yunlong; Meng, Fandong; Zou, Beiqi; Li, Zhixu; Qu, Jianfeng; Zhou, Jie. Zero-shot cross-lingual summarization via large language models. *arXiv preprint arXiv:2302.14229*. 2023.

47. Wang, Jiaan; Liang, Yunlong; Meng, Fandong; Sun, Zengkui; Shi, Haoxiang; Li, Zhixu; Xu, Jinan; Qu, Jianfeng; Zhou, Jie. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*. 2023.

48. OpenAI. *GPT-4 Technical Report*. 2023. Available also from: https://openai.com/research/gpt-4. Accessed: 2025-07-14.

49. Qiu, Yifu; Ziser, Yftah; Korhonen, Anna; Ponti, Edoardo; Cohen, Shay. Detecting and Mitigating Hallucinations in Multilingual Summarisation. In: Bouamor, Houda; Pino, Juan; Bali, Kalika (eds.). *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, 2023, pp. 8914–8932. Available from DOI: 10.18653/v1/2023.emnlp-main.551.

50. Huot, Fantine; Maynez, Joshua; Alberti, Chris; Amplayo, Reinald Kim; Agrawal, Priyanka; Fierro, Constanza; Narayan, Shashi; Lapata, Mirella. uPLAN: Summarizing using a Content Plan as Cross-Lingual Bridge. *arXiv preprint arXiv:2305.14205*. 2023.

51. Thieme, Paul. The indo-european language. *Scientific American*. 1958, vol. 199, no. 4, pp. 63–78.

52. Tang, Yuqing; Tran, Chau; Li, Xian; Chen, Peng-Jen; Goyal, Naman; Chaudhary, Vishrav; Gu, Jiatao; Fan, Angela. Multilingual Translation with Extensible Multilingual Pretraining and Finetuning. 2020. Available from arXiv: 2008.00401 [cs.CL].

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Full Form
CLS	Cross-Language Summarization
MLS	Multilingual Summarization
M-L	Multilingual
M-D	Multidomain
ML-MD	Multilingual-Multidomain
ES	Extractive Summarization
AS	Abstractive Summarization
NLP	Natural Language Processing
NLG	Natural Language Generation

Table 4.1 List of Abbreviations Used in This Thesis

A Developer documentation

The materials for this work can be found online[1]. List of scripts and other material:

- `readme.md` – The readme file
- `requirements.txt` – A list of libraries needed to be installed for this work.
- `Data`
  - `Initial-20` – The Initial-20 dataset organized by language folders (Bengali (bn), English (en), and Hindi (hi)), and each folder contains four domains (films, sportsman, books, writers) in JSON format.
  - `Curated-20` – The Curated-20 dataset organized by language folders (Bengali (bn), English (en), and Hindi (hi)), and each folder contains four domains (films, sportsman, books, writers) in JSON format. The `cleaned` subdirectories contain a version created with the `Data/clean.py` script.
  - `clean.py` – Preprocesses each (language-domain) pair (e.g., bn_writers.json).
  - `mD_split.py` – Creates train/val/test set splits for "Multidomain" experiment. For a particular language, it takes each domain as input and splits it in an 80:10:10 ratio (e.g., bn_writers_train.json, bn_films_train.json, .., bn_films_test.json), and saves it in `perDomainperLang` folder. Then merges and shuffles all the domain's training sets to create a train set (e.g. bn_train.json). A similar approach is used to get validation sets and test sets. This makes sure every domain is present in all the splits.
  - `mL_split.py` – Creates splits for Multilingual experiment. From `perDomainperLang` folder, it merges train sets of all languages (e.g. bn_writers.json, en_writers.json, hi_writers.json) for each domain to create train sets(e.g. writers_train.json). A similar approach is used to get validation sets and test sets.
  - `mLmD_split.py` – Creates splits for Multilingual-Multidomain experiment. It uses splits from the Multilingual task, merges the train sets of all domains to get a full training set of 12 domain-language pair. A similar approach is used to get validation sets and test sets.
- `extractive`
  - `extractive.[py, sh]` – Script to extract top-k sentences. Input of this stage is a cleaned language-domain pair subset (e.g., Data/bn/-cleaned/bn_writers.json)
- `abstractive` – This folder contains a folder for each of the experimental settings.

[1] https://github.com/Miftahul7/Cross-Language-Summarization

- `Multidomain` – Folder of Multidomain summarization
  - `model`
    - `dataloader.py` – This code defines a PyTorch Lightning data module. It loads JSONL data, tokenizes inputs and targets with language-specific handling, and provides DataLoaders for training, validation, and testing.
    - `model.py` – Summarizer model uses either mBart or mT5.
  - `train_mD.[py,sh]` – Script training the summarizer model.
  - `testing`
    - `testing.[py,sh]` – loads trained model checkpoints, evaluates the mBart or mT5 summarization against the gold summary. Output a CSV file with the results, including the Rouge score.

- `Multilingual` – Folder of Multilingual summarization
  - `model`
    - `dataloader.py` – This code defines a PyTorch Lightning data module. It loads JSONL data, tokenizes inputs and targets with language-specific handling, and provides DataLoaders for training, validation, and testing.
    - `model.py` – Summarizer model uses either mBart or mT5.
  - `train_mL.[py,sh]` – Script training the summarizer model.
  - `testing`
    - `testing_mbart.[py,sh]` – Loads trained model checkpoints, evaluates the mBart summarization against the gold summary. Output a CSV file with the results, including the Rouge score.
    - `testing_mt5.[py,sh]` – Loads trained model mT5 checkpoints, evaluates the mT5 summarization against the gold summary. Output a CSV file with the results, including the Rouge score.

- `ML-MD` – Folder of Multilingual-Multidomain summarization
  - `model`
    - `dataloader.py` – This code defines a PyTorch Lightning data module. It loads JSONL data, tokenizes inputs and targets with language-specific handling, and provides DataLoaders for training, validation, and testing.
    - `model.py` – summarizer model uses either mBart or mT5.
  - `train_mLmd.[py,sh]` – script training the summarizer model.
  - `testing`
    - `testing_mbart.[py,sh]` – loads trained mBart model checkpoints, evaluates the mBart summarization against the gold summary. Output a CSV file with the results, including the Rouge score.
    - `testing_mt5.[py,sh]` – loads trained mT5 model checkpoints, evaluates the mT5 summarization against the gold

summary. Output a CSV file with the results, including the Rouge score.

- `evaluation` – This folder contain 3 python scripts that produce an overall evaluation summary. We get a clear idea of how each language-domain pair performed for each experimental settings.
  - `evaluate_mD.py` – Take each CSV file of the Multidomain experiment returned by test script and returns a JSON file summarizing average ROUGE-L F1 scores across multiple domains and languages. Summarizes for mBart and mT5 separately.
  - `evaluate_mL.py` – Take each CSV file of Multilingual experiment returned by test script and returns a JSON file summarizing average ROUGE-L F1 scores across multiple domains and languages. Summarizes for mBart and mT5 separately.
  - `evaluate_mLmD.py` – Take each CSV file of ML-MD experiment returned by test script and returns a JSON file summarizing average ROUGE-L F1 scores across multiple domains and languages. Summarizes for mBart and mT5 separately.