Grounded Content Automation: Generation and Verification of Wikipedia in Low-Resouce languages.

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

Master of Science
in
Computational Linguistics
by Research

by

SHIVANSH S
2019114003
shivansh.s@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Grounded Content Automation: Generation and Verification of Wikipedia in Low-Resouce languages.**" by **Shivansh S**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Advisor: Prof. Vasudeva Varma
Date	Co-advisor: Dr. Manish Gupta

To my loving family, and to my amazing friends.

Acknowledgments

As I reflect on my journey of the past five years leading up to completing my B. Tech and Master by Research degrees, I am grateful for the various experiences and the support system essential to my growth and learning. The challenges I have faced and the knowledge I have gained have contributed to my personal growth, and I am deeply thankful to each individual who has played a part in this journey, leading to my development as a researcher and a person.

It would not have been possible for me to complete my work on time, in a manner that I am proud of, without the support from my family. They have always understood my challenges, helped me face them, and made me accountable towards the work I want to do. Without them and their love and support by my side, I would not have been able to complete these five years with the success I have been able to. So, a special thanks to my parents, sister and grandmothers for always being there for me.

I thank my advisor, Professor Vasudeva Varma, for welcoming me into the iREL family and providing valuable guidance and support throughout my three years working with him. His availability, encouragement and ideas have been pivotal in shaping my approach towards research.

Equally, I am grateful to Dr. Manish Gupta, my co-advisor, whose attention to detail, vast knowledge, and readiness to help have significantly influenced my research methodology, making it more rigorous and exciting. He has guided me through every step of my research projects and helped me transform my abstract ideas into published research papers.

My journey would not have been the same without my amazing co-authors: Dhaval, Shivprasad, Ankita, Aakash, Lakshya and Harshit. Their insights, discussions and contributions have enhanced the quality of our collective work and made me more creative, professional and open to ideas. I hold their contributions in high esteem, and collaborating with them has been a pleasure. A special thanks go to my friends from iREL, Aditya, Gokul, Nirmal, Pavan, Sagar, Bhavyajeet, Tathagata and Tushar, for enriching discussions on NLP and research with creativity, rigour and knowledge.

I am forever grateful to have a friend group as amazing, helpful and hardworking as mine. Special thanks to my friends Arjo, Tanishq, Shreyas, Zeeshan, Kush, Rutvij and the entirity of room 333 for being there throughout my five years. Also, I would like to thank Priyanka, for being my support and for helping me be a better version of myself.

Lastly, I acknowledge the vibrant and amazing community at IIIT, with Professors and peers having expertise in various domains, but still being extremely helpful in teaching, collaborating and working with. The multiple interactions have collectively shaped me into the individual I am today. The research-

oriented culture and emphasis on skill development at our college have helped my academic and personal growth.

This acknowledgement, though comprehensive, cannot fully capture the depth of my gratitude to everyone who has been a part of my journey. I offer my sincere thanks to all of you.

Abstract

In this thesis, we work towards improving the representation of low-resource languages in the digital world by easing the access and participation of these communities to reliable information hubs like Wikipedia. Although the internet has brought in an information age, it is disproportionately distributed amongst language communities since content and tools for low-resource languages are less readily available. Recognizing the importance of Wikipedia as the primary source of reliable, unbiased information, we seek to improve the information available by automatically generating Wikipedia articles in low-resource languages to improve the quality and quantity of articles available.

Our work begins with XWikiGen, a cross-lingual multi-document summarization task that aims to generate Wikipedia articles using reference texts and article outlines. We propose the XWikiRef dataset to facilitate this, which spans eight languages and five distinct domains, laying the groundwork for our experimentation. We observe that existing Wikipedia text generation tools rely on Wikipedia outlines to provide a structure for the article. Hence, we also propose Multilingual Outlinegen, a task focused on generating Wikipedia article outlines with minimal input in low-resource languages. To support this task, we introduce another novel dataset, WikiOutlines, which encompasses ten languages over eight domains, further enriching available multilingual tools for further research work.

An important question with text generation is the reliability of the generated information. For this, we propose the task of Cross-lingual Fact Verification (FactVer). In this task, we aim to verify the facts in the source articles against their references, addressing the growing concern over hallucinations in Language Models. We manually annotate the FactVer dataset for this task to benchmark our results against it. By exploring these three tasks, we highlight the disparity in content and tools available in low-resource languages, underscore the importance of multilingual and cross-lingual tools in global participation and propose innovative solutions to enhance Wikipedia's accessibility and reliability for low-resource languages.

Overall, we contribute multiple novel datasets and methodologies to automatic text generation and highlight the importance of inclusivity in the Internet age. By tackling the challenges of article generation, outline generation and fact verification, we pave the way for future advancements that promise to improve the quality and quantity of information available to low-resource language communities of the world.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In an era where internet and technologies effect every aspect of our lives, equitable access to information remains a challenging task, especiall for low-resource languages. While in the current information age we see abundance of web-pages and content available in English, languages spoken by smaller communities struggle for representation on the web. Among the many endeavors to bridge this gap, automated text generation is a promising avenue, offering a easily scalable solution while taking advantage of existing information.

This thesis delves into the problem of automated Wikipedia text generation in low-resource languages. We highlight the importance of Wikipedia within these communities, review previous research in this area, and explore various scalable methods to enhance the representation of these languages online. As the foundational chapter of this thesis, we stress the need for developing robust natural language processing (NLP) tools for low-resource languages, aiming to reduce the digital divide and preserve linguistic diversity on the internet.

We also emphasize the significance of cross-lingual and multi-lingual tools in improving access to information across diverse linguistic groups. These tools not only enable the sharing of existing internet resources across languages but also enrich the digital landscape globally. Although text generation remains our main focus, ensuring the reliability of content generation is extremely important. In this context, we focus on the role of Wikipedia as a trusted source of information, motivating our exploration of automated Wikipedia text generation in this thesis.

Finally, within this chapter, we outline our contributions to this field and provide an overview of the structure that guides the subsequent sections of this thesis.

1.1 Motivation for work in Low-Resource Languages

1.1.1 Bridging the Digital Divide

In today's world, access to the internet and information plays a crucial role in socio-economic development, education and cultural preservation. Most of the information available on the internet is in

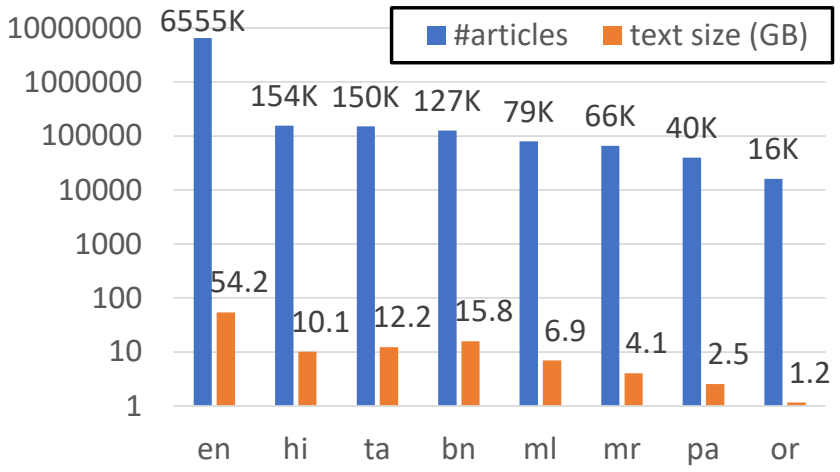

Figure 1.1: No. of Wikipedia pages and size of text in GBs, using 20220926 Wikidump. Y axis is in the Log Scale.

English and other high-resource languages, and this is reflected in Wikipedia as well. As is highlighted in Figure 1.1, there are around ∼6.56 million articles in English, compared to an average of ∼90 thousand articles for low-resource languages.

However, a significant portion of the world's population speaks low-resource languages, which are marginalized in the digital landscape. Despite the widespread availability and accessibility of the internet, content in low-resource languages is scarce, limiting the ability of these communities to engage with online resources, preserve their culture or access educational resources.

In Figure 1.1, we highlighted one example of where the digital divide exists, favoring content in high-resource languages and perpetuating inequalities in access to knowledge and information. While speakers of high-resource languages benefit from a wealth of digital resources, those who speak low-resource languages face barriers to accessing information online, hindering their ability to participate fully in the digital world. The majority of the content available on the internet being in high-resource languages excludes people from low-resource communities from understanding and consuming information. Furthermore, any subsequent research/tools made with the information on the internet as its base predominantly caters to high-resource language, increasing the digital divide.

1.1.2 Preserving Linguistic Diversity

Having information available in low-resource languages is important for the preservation of linguistic diversity, cultural heritage and traditional knowledge. However, the lack of digital content in these languages poses a threat to their preservation and revitalization. We observe Wikipedia acting as a source of preserving cultural heritage, where a lot of low-resource entities have pages without having a lack of equivalent English Wikipedia pages. Across seven low-resource languages, on average, ∼42% of entities

	hi	ta	bn	ml	mr	pa	or
Percentage	50.60	46.70	31.50	36.30	42.0	38.70	39.40

Table 1.1: Percentage of pages in Low-Resource languages without equivalent English Wikipedia page.

do not have a corresponding English Wikipedia page, implying that they are entities of local importance. More specifically, we see the spread in Table 1.1, which tells us how low-resource language communities use mediums like Wikipedia to store and convey their cultural history and knowledge about important local entities.

Hence, it is important to work on low-resource languages, to increase content in LR languages directly and to develop tools to help people from low-resource communities to share knowledge, culture and information with the world.

1.2 Need for Multi-lingual and Cross-lingual tools

1.2.1 Accessibility of Information

In today's interconnected world, where communication transcends linguistic boundaries, the ability to understand and contribute content in multiple languages is becoming increasingly crucial. Since the internet primarily has information in high-resource languages, it becomes difficult for low-resource communities to contribute and learn from the internet. The language barrier also poses challenges related to accessing information and collaborations across linguistic communities while ensuring the accuracy and reliability of content.

The motivation to work on multilingual and cross-lingual research stems from the need for effective communication and information access in diverse linguistic environments. Developing tools catering to multiple languages allows low-resource communities to contribute better to the digital world, preserve their culture and have access to reliable information. Research in cross-lingual tools is important since it develops tools that can use the vast amount of information available in high-resource languages to improve the content and its availability in low-resource languages.

1.2.2 Knowledge sharing in Cross-Lingual context

A possible method to improve existing knowledge in low-resource languages is to leverage existing generic web content in those languages. It is not easy to do so, since even generic content exists very sparsly in low-resource languages, as can be seen in Wikipedia (show in Figure 1.1) and large publically available dumps like CommonCrawl [3].

In Wikipedia, the references to an article can be in a language different than the article's language. This enables authors and editors to verify information for articles in low-resource languages by having

references in high-resource languages. Table 1.2 shows statistics on the percentage of reference texts in Wikipedia that are in the same language as the source Wikipedia article. We see that for the low-resource languages, on average, the number is quite low, highlighting that there is not enough information in low-resource languages available on the internet. The table also highlights the need for cross-lingual research, which utilizes existing information in English to enhance available information in other languages.

Domain/Languages	**bn**	**hi**	**ml**	**mr**	**or**	**pa**	**ta**	**en**
books	16.5	14.9	9.9	12.3	0.0	5.2	28.2	94.8
films	21.5	10.4	21.0	6.5	0.0	1.2	10.9	96.8
politicians	21.4	31.2	8.4	25.0	0.0	1.9	8.7	90.0
sportsmen	1.4	1.7	1.2	2.5	0.0	0.2	1.1	87.2
writers	11.0	18.3	4.6	27.2	0.0	6.0	7.7	94.7

Table 1.2: Percentage of reference texts which are in the same language as source Wikipedia article.

1.3 Need for automatic Wikipedia text generation and verification.

1.3.1 Automatic Wikipedia text generation

Wikipedia is a vast repository of knowledge, but it constantly requires updates and expansions to reflect the latest information and developments. Figure 1.2 displays the number of edits and new articles made to Wikipedia in a period of 16 years (from 2006 up to 2022). We observe that the changes made to English are significantly higher compared to changes made to low-resource languages.

To assist the contributors, and to fill gaps in low-resource languages, it is imperative to make use of reliable information available in English and develop an automated cross-lingual tool to generate Wikipedia articles reliably and efficiently. Naïve methods like translation do not work, since there are many entities in low-resource languages without a corresponding page in English as shown in Table 1.1. Hence, a generative method is required for the creation and verification of Wikipedia pages, to ensure reliable information is available to all.

1.3.2 Fact Verification for Text Generation

As previously illustrated, there is a growing need for cross-lingual tools capable of generating content in Low-Resource languages. Multiple attempts have been made to automatically generate Wikipedia content using information available on the net using language models. However, one of the well-known issues with using language models for text generation is hallucination, which makes it difficult to be confident about the reliability of automatically generated texts. Given that Wikipedia is used widely as a

source of truthful information, it becomes imperative for any generated text to be supported by facts and references.

Existing work has addressed fact verification at the sentence level in both monolingual and multilingual settings, but no previous work is done at fact-level verification. Fact-level verification is necessary since it ensures that each fact within a sentence is validated. Based on Table 1.3, the average number of facts (indicated by a fact-triplet) within a sentence across all languages is observed to be approximately two. Hence, the development of cross-lingual fact-level verification for Wikipedia articles becomes necessary to allow for including automatically generated text in Wikipedia.

1.4 Cross-Lingual Multi-Document Wikipedia Article Generation

As stated above, Wikipedia as a source of information requires frequent updates to include relevant new entities. Low-resource Wikipedia has a general lack of contributors, which makes it necessary for us to develop a cross-lingual automated pipeline for article generation. Previous work on automated Wikipedia generation focuses primarily on English, utilizing the English reference text to summarize and generate the article. However, as shown in Table 1.2, monolingual summarization can not be a solution to this problem.

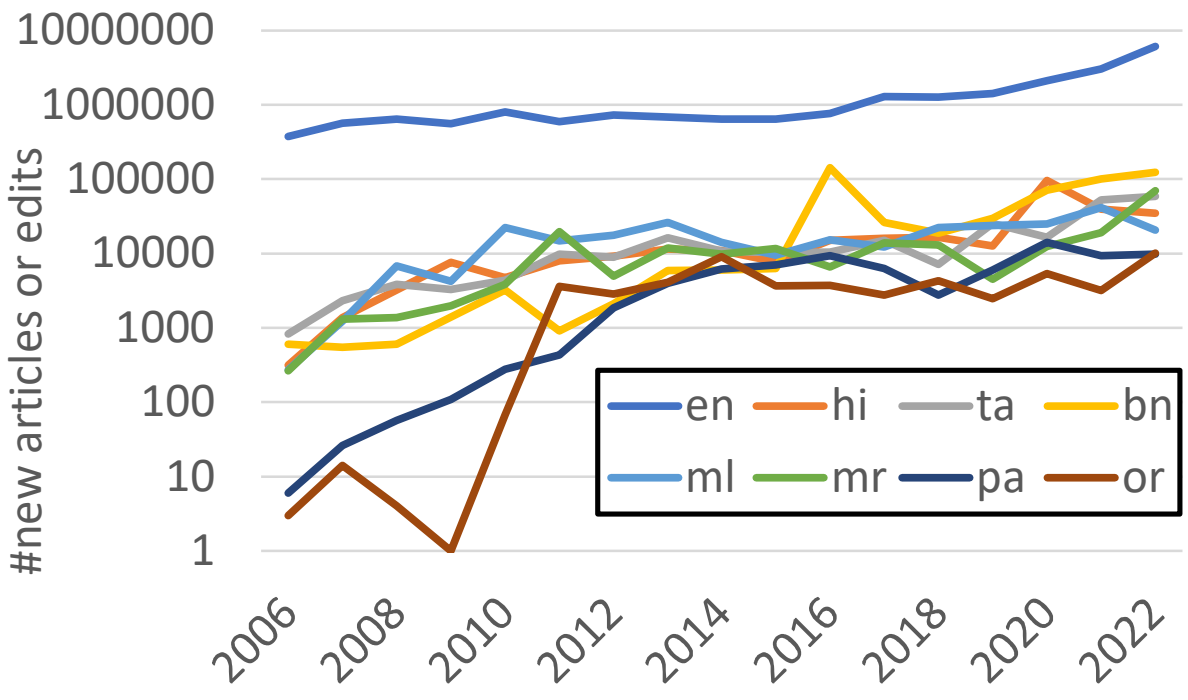

Figure 1.2: Number of edits or new articles on Wikipedia, 2006-2022 according to the 20220926 Wikipedia dump. Y axis is in the log scale.

	Articles	Sent	Fact	Avg Facts per Sent
or	2943	14575	24777	**1.700**
hi	15640	57424	112543	**1.960**
te	4896	25078	43278	**1.726**
en	51475	133054	298468	**2.243**
gu	1615	9561	17723	**1.854**
kn	3771	26083	48593	**1.863**
bn	25221	122008	238978	**1.959**
ml	13865	55750	103637	**1.859**
as	1492	10344	17190	**1.662**
ta	16477	57363	101089	**1.762**
pa	8835	30665	63664	**2.076**
mr	8858	20144	45017	**2.235**

Table 1.3: Data Statistics about Wikipedia Articles, Sentences and Number of Facts according to XAlign Dataset [1] in People Domain

For these reasons, we propose the novel task of Cross-Lingual Wikipedia Article Generation (XWikiGen), in which we generate the Wikipedia article by summarizing multiple reference documents. Since this is a novel task, we also contribute a Cross-Lingual Wikipedia Reference Dataset (XWikiRef), covering ∼69k Wikipedia articles across five domains and eight languages. Using this dataset, we create a two-stage summarization pipeline, where we input Article Title, Article Outline, References, Target Language to generate the Wikipedia page in the target language section by section. The pipeline is based on unsupervised Extractive Summarisation followed by Cross-Lingual Supervised Abstractive Summarisation to generate text specific to each section.

Successfully addressing these challenges contributes to the enhancement of information available in low-resource languages on the web and develops cross-lingual tools for low-resource languages as well.

1.5 Multilingual Wikipedia Outline Generation

Automated Wikipedia text generation is an important step towards improving the representation of low-resource (LR) languages, and keeping information on Wikipedia up to date and reliable. An important step towards easing the creation of Wikipedia pages is to have a structured outline as the reference. Humans and automated generators both would benefit from having an automatically generated structured outline to help them plan and write the article. To aid outline generation, we also impose the constraint of doing so with minimal information in the form of entity name, language and domain, and

propose the task of OutlineGen. We do this so that the user has to provide minimal information to get started with writing an article.

There has been work previously done primarily on Wikipedia page generation (in English), and even for work related to outline generation, it has been in English and includes a lot of information as input. We are the first to propose Wikipedia Outline Generation as a task in a multi-lingual setting. The challenges with OutlineGen tasks derive from the diversity in Wikipedia outlines across articles, even for a given (language, domain) pair. The outlines seen across languages and domains are not consistent, making it difficult to generate automated outlines.

Since we propose a novel problem, we also created a dataset WikiOutlines, to benchmark our work, and encourage work on this work in the future. Our dataset contains Wikipedia outlines from $\sim$ 166k pages across eight domains and ten languages. We then propose two methods, first based on finite state automata, and the other based on multilingual transformers for this task, for which we give scores across multiple metrics.

Addressing these problems, and finding an effective solution for them will be useful for planning Wikipedia articles better for both human author/editor, and for automated pipelines. An existing outline will also reduce humans' work for editing since an automatically generated article will follow the outline, generating relevant text.

1.6 FactVer Dataset and Task

In an era where the internet is filled with non-reliable information, Wikipedia stands out as a source of verifiable and reliable content. With more automated methods of text generation, even for Wikipedia, it has become important to automate the verification of facts represented in generated texts. To do this, we present a novel approach for cross-lingual fact extraction and verification called FactVer.

Formally, we extract facts as a factoid (subject, relation, object) from the article, and label whether it is supported by factoids extracted from citations and references. We keep the granularity of verification at fact-level, instead of sentence-level like existing work, since we observe that many sentences have multiple facts within them. Any method proposed for this task must be cross-lingual, since references need not be in the same language as that of thesource article.

Since this is a novel task, we also created FactVer dataset, a manually annotated cross-lingual dataset for fact extraction and verification. Having an annotated public dataset will help future work on this task, as well as help us benchmark our methods on it. Our dataset spans $\sim$33k articles across six languages in the people's domain in Wikipedia.

Using the FactVer dataset, we work on the Fact Extraction and Verification task. This task is essential since it streamlines the fact-checking process for authors and editors. Moreover, it allows us to verify automatically generated Wikipedia articles and isolate instances of hallucinations. Successfully working on this task would allow us to enhance existing web content in low-resource languages by ensuring the reliability of the information.

1.7 Thesis Contributions

Overall, this work highlights the need for more reliable information in low-resource languages on the internet. To address this problem, we have developed tools to help automate the Wikipedia generation and verification process. Formally, we make the following contributions:

- **XWikiRef and XWikiGen:** A cross-lingual multi-document dataset and task for generating Wikipedia articles from their outline and references.
- **WikiOutlines and OutlineGen:** A multilingual dataset and task across eight domains and ten languages for generation Wikipedia article's outline given minimal information.
- **FactVer:** A cross-lingual automated pipeline for verification of a Wikipedia article from its references.

1.8 Organization of Thesis

This thesis is structured as follows:

- **Chapter 1: Introduction and Motivation** (this chapter): Here, we motivate the need to increase and enhance reliable information on the internet in low-resource languages. We consider Wikipedia as the main source of reliable information and develop tools for it.
- **Chapter 2: Related Work and Background**: We discuss past works on Outline Generation, Multilingual Generation, Cross-Lingual tools, Article Generation, Fact Extraction and Verification to gain relevant background and basis for our work.
- **Chapter 3: Dataset Preparation and Analysis**: We discuss the need for outline generation and article generation datasets and the process and details of curating our multilingual datasets.
- **Chapter 4: Cross-lingual Generation of Wikipedia Articles**: Using the XWikiRef dataset, we create a pipeline for automatically generating Wikipedia articles from article outlines and references.
- **Chapter 5: Multilingual Generation of Wikipedia Outlines**: We utilise the WikiOutlines dataset to generate outlines for Wikipedia articles using minimal information (article title, language and domain).
- **Chapter 6: Cross-lingual Fact Extraction and Verification**: We discuss the motivation and creation of the FactVer dataset and propose an automated pipeline for verifying Wikipedia article facts from its references.

- **Chapter 7: Conclusion and Future Work**: Here, we summarise our work, re-iterating the motivation and effect of our work. We also discuss possible future work that can be built on top of it to enhance low-resource language content online.

Chapter 2

Related Work

Automated text generation has been a subject of keen exploration and study within academics working on improving linguistic representation. Given our focus on text generation, information representation, multilingual/cross-lingual generation, and fact verification, it is imperative to delve into the existing research in these realms. This section offers a comprehensive review of previous works, spanning various methods in text generation techniques, approaches to multilingual text generation, strategies for summarization, and methodologies for fact verification. By delving into this body of literature, we aim to better situate our own contributions within the broader landscape of advancements in Wikipedia text generation for low-resource languages.

This chapter is dedicated to exploring text generation within the context of Wikipedia. Initially, we delve into the realm of short-text generation for Wikipedia, which involves structured information in various formats like factoids and tables into text, typically limited to a sentence or two. Following this, our attention turns to long-text generation, where we aim to automate the creation of either sections or entire articles on Wikipedia. While much of the existing research has focused on English, recent efforts have started to embrace multilingual approaches.

Understanding the significance of a Wikipedia Outline as a useful structural tool for both humans and automated systems, we go through past work concerning its automatic generation. Wanting a more general understanding of multilingual and cross-lingual text generation, we survey various works on summarization, drawing on multiple methodologies and available datasets. Lastly, we study the well-established field of multilingual fact extraction and verification. Through a review of previous studies in this domain, we seek to gain inspiration, deepen our understanding, and improve our own research.

2.1 Wikipedia Short Text Generation

Since Wikipedia has been such an importance source of content for information, there have been lot of attempts to automate the process of generating information for Wikipedia. Exploration of automated Wikipedia text generation has been a problem of interest for the past 5-6 years.

Initial efforts in this field include [4], which introduces the WikiBio dataset (spanning over ∼700k articles). In this, the authors use the infobox of articles as factual context, and the first sentence of Wikipedia pages is used as the desired biography. Using neural seq-2-seq methods, they establish that more context results in improved results. In [5] work on the WeatherGov dataset, which converts table-like weather data into coherent sentences. They create a complext seq-2-seq architecture using LSTMs [6] as encoder and decoder with an alignment layer. [7] and [8] also works on converting ordered data (tables) to text using neural seq2seq architectures.

Work related to converting graphical structured data set include [9], which takes semantic triplets from DBPedia and WikiData to convert it into texts. A notable contribution in this realm, [10], uses state-of-the-art pretrained Transformer models [10] like BART [11] and T5 [3], which have found widespread adoption in Fact-to-Text tasks.

Majority of the previous work for ordered-data to text generation have focused in English. Recently, cross-lingual data to text generation has gained more attention, with works such as [1, 12] working on it. In [1], they introduce a dataset, XAlign, with 0.45Million fact-to-text aligned samples across eight languages. Acknowledging the lack of reliable information in low-resource setting, they also manually anotate ∼5k samples to give gold dataset. On this data, they train multiple transformer models on this with finetuned mt5 [13] giving the best results. To extend work on this dataset, [12] propose XAlignV2, including annotated data for four more languages. They also conduct extensive experiments on best models for the fact-to-text task, determining mt5 [13] with fact and structure aware inputs perform the best.

2.2 Wikipedia Long Text Generation

Another important task for Wikipedia text generation is to generate Wikipedia articles automatically. The significance of this task stems from the important role Wikipedia plays as a provider of reliable information. Since each article in Wikipedia requires significant manpower to ensure quality, reliability and significance, multiple studies have been done to help automate this process. In Table 2.1, we provided a comparison between all existing datrasets to generate Wikipedia articles. [14, 15, 17] work on taking external articles as input to generate full or parts of the Wikipedia article. Wiki Current Events Portal (WECP) [17] takes in multiple news articles and an article title to generate the summary section for a Wikipedia article. Compared to that, [14] take in a set of non-Wikipedia URLs as references to generate the whole Wikipedia article. All of these studies have been on English with varying multi-document inputs, highlighting the need for more work in a multilingual setting.

More recently, there have been studies done for generating summaries and paragraphs for Wikipedia articles in a multilingual setting. Existing works like Multiling [18] and WikiMulti [19] take in the whole Wikipedia articles as input to generate few sentences for articles in different languages. Although this seems to be a good start, there is a marked difference between historical literature and models available for English versus other low-resource languages.

Dataset	Input	Output
WikiSum [14]	Set of citation URLs	Whole Wiki article
WikiAsp [15]	Set of citation URLs	One section in same language
GameWikiSum [16]	Professional video game reviews	Gameplay Wikipedia sections
Wiki Current Events Portal (WCEP) [17]	Set of news articles	WCEP Summary
MultiLing'15 [18]	Whole Wikipedia article	First few Wikipedia sentences in same language
WikiMulti [19]	Whole Wikipedia article	Intro paragraph in other language
XWikiRef (Ours)	Set of citation URLs	One section in another language

Table 2.1: Input-Output format of popular Wikipedia Summarization datasets.

A common limitation to these works has been to exclude section-specific intent during summarization, generating article as a whole instead. Hayashi et al. [15] addresses this, where they work on section-specific summarization by recognizing recognizing the main topics in the input text. While effective, the reliance on the model to figure out latent subtopics introduces challenges in content selection. To tackle this, our work related to Wikipedia text generation provides section-specific citations as input. This allows us to study the summarization abilities of the model better since noisy references belonging to other sections are excluded.

2.3 Wikipedia Outline Generation

In an article, its outline serves as a structured document playing acting as the spine of the article's content. The importance of article outlines are specifically high for authors and automated systems which want to write or generate articles. The first paper highlighting the importance of automatic outline generation and working on it was proposed by Zhang et al. [2]. In it, the task was to take in the whole Wikipedia article as the input (without any delimiting information like sections or section-titles) and to produce the section boundary and section titles. Formally, the first task was to identify the boundaries for sections given the entire document text and the second task was to generate the required section heading

for that specified section. They built a complex architecture using LSTMs [6] to perform both the tasks simultaneously.

Recently, Maheshwari et al. [20] introduced a related task—creating a Table of Contents (or Outline) for lengthy documents such as contracts and financial documents.In their case, the section distinctions were already given, and the task was to generate the section titles in a way relevant to the varied personalities of the readers. They model the section-title generation task as a question generation task using BART [11], along with custom-defined rewards to nudge the generated output in the correct direction.

Notably, both these efforts focused exclusively on English text. The extension of Outline Generation for multiple low-resource languages introduces a different set of challenges which we address in our work.

2.4 Multilingual Generation and Summarization

As highlighted in previous sub sections, majority of historical work in text generation has been in English. Multi-Lingual and cross-lingual text generation has been a complex but important area where much work has been done in the past 5-10 years. Since multi-lingual Transformer models like mT5 [13] and mBART [21], amongst others, have been created, we have seen approaches towards many common problems in a multi-lingual setting. Popular multi-lingual and cross-lingual NLG tasks include machine translation [22, 21], question generation [23], summarization [24], style transfer [25], and multilingual neural dialogue generation [26]. All these tasks showcase challenges often faced in multilingual context, and propose baselines and dataset to improve future works on these.

In our case, we focus on summarization of texts in a multilingual context. In the past, relatively less work has been done for this task in low-resource languages. One of the first papers on this was MultiLing'15 [18], which introduced multi-lingual summarization in 30 languages. Later, datasets like XLSum [27], MLSum [28], CrossSum [29], Global Voices [30], WikiLingua [31], WikiMulti [19] have been created for cross-lingual summarization.

Consisting of $\sim$1.35M professionally annotated articles from BBC, XL-Sum [27] covers 44 of low to high resource languages. CrossSum, released by Hasan et al. [29], extends the multi-lingual XL-Sum with $\sim$1.7M instances of cross-lingual summaries. Another recent datasets are MLSum and GlobalVoices, having $\sim$1.5M and $\sim$300K summaries in cross-lingual context covering 5 and 15 languages respectively. Outside of the news domain, WikiLingua [31] covers $\sim$770k summaries extracted in 18 languages from WikiHow.

We enrich this line of work by contributing a new cross-lingual multi-document summarization dataset, a new multi-lingual outline generation dataset, and a cross-lingual multi-document fact verification dataset.

2.5 Multilingual Fact Extraction and Verification

With more and more work happening in the space of text generation, it has become important to verify the content generated. Although there have been multiple studies on hallucinations [32, 33], we want to focus on more grounded approaches highlighting areas with unsupported facts. The challenge of structured fact extraction, often represented as a tuple of ¡subject, relation, object¿, from unstructured textual data ahs been widely studied. Existing works in this field in multilingual context include [34] and [35]. We take inspiration from the work done in [34], which works on fact extraction in Telugu and Hindi, to extract facts in four other Indian languages without resorting to translation. Our proposed approach for fact-extraction aligns closely with the approach presented in [35], but we extend this by exploring alternative methods and including the challenging and important task of fact verification.

Fact Verification has been a problem of historical interest, with most old methods involving creation of manually annotated datasets. The limitations of such approach is easy to see, requiring huge manpower when including any new language or domain. Hence, automatic fact-extraction has become an important problem to solve, and there have been multiple attemps to do so. One of the most famous benchmarks for fact extraction is FEVER [36], which is a large-scale manually annotated dataset. There have been multiple models and pipelines created to solve this task, including [37, 38, 39]. Amongst these, the works working on a fact-level verification is [37], in which the authors propose a seq2seq model to generate logical operations between spans of claims to determine claim verification. [39] involves using language models for claim-generation in a zero-shot setting, generating questions to then verify the claim based on existing context. Other works in this domain [38, 40] have been mostly monolingual and focussing on sentence-level verification rather than at fact-level verification (which is more important to isolate and identify wrong facts at a granular level). Works have also included Knowledge Graphs, where individual facts can be added to the main graph [41], to then compare such represented facts between two graphs and verify them [42, 43, 44].

Chapter 3

Dataset Preparation and Analysis

We discuss the need and process of creating multilingual and cross-lingual datasets to improve the encyclopedic content creation process in low-resource languages on the web. To create an automated pipeline for Wikipedia article generation, we divide the problem into Outline Generation and Article Text Generation. Since existing works on both problems were primarily in English, we must create a new dataset to assist our work and future work on these topics.

3.1 XWikiRef: Cross-lingual Multi-document Summarization Dataset

3.1.1 Dataset Description and Creation

We first tackle the problem of generating a Wikipedia article, giving an article outline and relevant references. We know that there are multiple references to a Wikipedia article and that they may be in languages different from the source articles. As we had seen in Table 1.2, different languages of references are especially observed in articles of low-resource languages; hence, our datasets must be cross-lingual.

XWikiRef is a multi-document, cross-lingual, multi-domain summarization dataset with samples across eight languages and five domains. The dataset was created to tackle the novel problem of XWikiGen and aid future works on this topic. The task is to generate a Wikipedia article section-by-section, using just the article outline and its references. Each sample in XWikiRef consists of an article title, domain, source language, article outline and scraped reference texts. It also consists of the original Wikipedia text, which is the target for each source sample.

First, we narrow down our language of interest to English (en), Hindi (hi), Bengali (bn), Marathi (mr), Malayalam (ml), Tamil (ta), Odia (or) and Punjabi (pa) and our domains of interest to writers, books, sportsmen, politicians and films. We use Wikipedia's language-specific 20220926 XML dump to extract relevant information. Before that, we filter content based on domains using the Wikidata API[1], which gives us a list of entities.

[1] `https://query.wikidata.org/`

Once we get our extracted Wikipedia articles per language and domain, we prepare them for our dataset. A few articles of high information in Wikipedia will have multi-depth subsections since they are important entities. We consider the section title level 1 and the subsection title level 2. For any article which is of more depth than 2, we merge it with its parent subsection. Now that we have our article outline, language, domain, and article content, we will work on scraping references.

To get URLs mentioned in the reference section of Wikipedia, we use MediaWikiParserFromHell[2], a Python package. We want to extract data only from PDFs and web pages since most content is in those two formats. We exclude any other format and keep a five-second cut-off time to scrape PDF/web pages. After scraping, we use IndicNLP [45] to sentence-tokenize the text as phrases. We exclude any article which does not have a single scrapable URL reference.

We split the dataset into train, test, and validation in 80:10:10 ration, and make it available publically.

	bn	**hi**	**ml**	**mr**	**or**	**pa**	**ta**	**en**	**Total**
Books	313	922	458	87	73	221	493	1,467	4,034
Film	1,501	1,025	2,919	480	794	421	3,733	1,810	12,683
Politicians	2,006	3,927	2,513	988	1,060	1,123	4,932	1,628	18,177
Sportsmen	5,470	6,334	1,783	2,280	319	1,975	2,552	919	21,632
Writers	1,603	2,024	2,251	784	498	2,245	1,940	714	12,059
Total	10,893	14,232	9,924	4,619	2,744	5,985	13,650	6,538	**68,585**

Table 3.1: XWikiRef: Total number of articles per domain per language

3.1.2 Dataset Analysis

Our curated XWikiRef dataset has approximately 69k articles across all languages and domains, details of which are presented in Table 3.1. Since Wikipedia has a differing number of articles depending on language and domain, we get an observable difference in the number of articles per language per domain due to it. In total, we have curated the dataset for 105k sections across all articles, the distribution for which is also presented in Table 3.2.

[2]https://pypi.org/project/mwparserfromhell/

Domain/Lang	bn	hi	ml	mr	or	pa	ta	en	Total
Books	434	987	557	111	88	238	598	2,972	5,985
Film	2,139	1,363	3,737	676	1,351	476	4,781	4,766	19,289
Politicians	3,261	4,478	3,719	1,384	1,404	1,524	6,431	4,780	26,981
Sportsmen	9,485	8,118	2,642	3,056	485	2,624	3,769	2,698	32,877
Writers	2,598	2,743	3,435	1,166	896	3,034	3,113	2,409	19,394
Total	17,917	17,689	14,090	6,393	4,224	7,896	18,692	17,625	**104,526**

Table 3.2: XWikiRef: Total number of sections per domain per language

A Wikipedia article has multiple references in different languages, so our dataset is a cross-lingual multi-document summarization dataset. Presented earlier, Table 1.2 shows the percentage of references in the same languages as that of the article, which comes to an average of 15% for non-English Wikipedia articles. We also see in Table 3.3 that the distribution of the number of references per article comes out to approximately 5. For domain-wise distribution of number of references, Figure 3.1 shows how the majority of domains have 5+ references, truly making it a challenging multi-document summarization task.

	bn	hi	ml	mr	or	pa	ta	en	Average
Books	3.62	2.61	2.59	2.07	3.46	2.30	2.40	6.34	3.17
Film	4.85	7.14	3.34	2.96	3.81	4.10	3.83	12.74	5.35
Politician	4.98	4.09	3.75	3.87	2.07	3.59	3.91	14.21	5.06
Sportsmen	6.37	8.30	6.96	4.20	3.93	4.49	6.38	21.88	7.81
Writers	5.20	5.46	4.16	3.74	2.85	3.34	4.20	17.61	5.82
Average	5.00	5.52	4.16	3.37	3.22	3.56	4.14	14.56	**5.44**

Table 3.3: XWikiRef: Individual and Average number of references per domain per language.

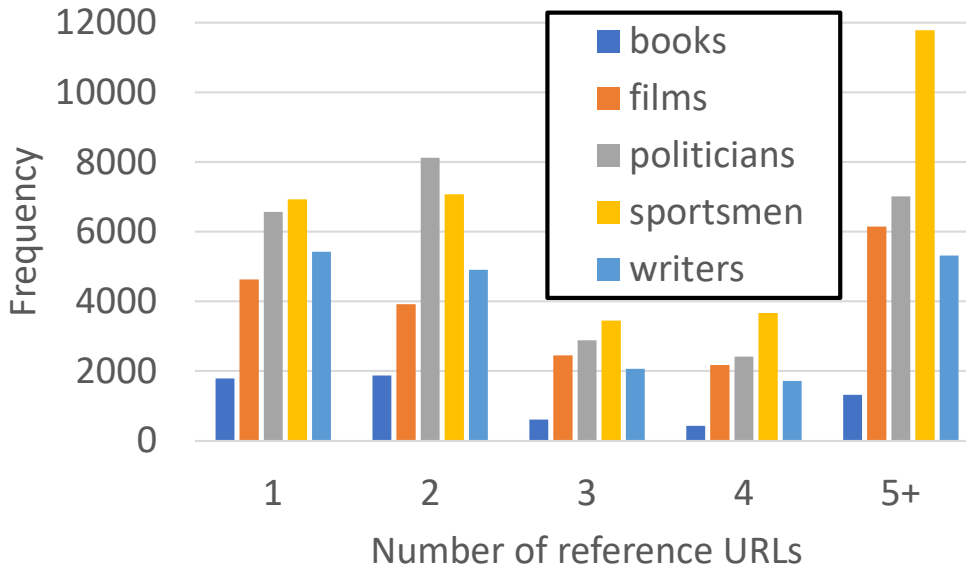

Figure 3.1: Distribution of number of reference URLs across domains in our XWikiRef dataset

3.2 WikiOutlines: Multilingual Outline Generation Dataset

3.2.1 Motivation

The need for creating a multi-lingual outline generation dataset like WikiOutlines is deeply rooted in enhancing the quality and scalability of Wikipedia text generation for low-resource languages. Although existing dataset and work exists (WikiOG by Zhang et al. [2]), the motivation for WikiOutlines is driven by the significant limitations in current datasets and methodologies for low-resource outline generation. In our dataset analysis, we observe significant diversity in article structures across multiple domains and lanuages which we hope to address and contribute to by creating a standard benchmarking dataset.

We assemble a dataset spanning ten languages and eight domains, focussing on low-resource indic languages. We ensure that our dataset is not dominated by English articles due to it having more articles, and instead ensure all languages are represented fairly. In our dataset and proposed task, our goal is to generate a Wikipedia Outline using (entity, language, domain) as the input to ensure minimal information from the users side is required. This initiative paves the way for models that can understand and replicate the intricate structure of Wikipedia pages, making the process of creating and expanding Wikipedia entries more efficient and inclusive.

3.2.2 Dataset Description and Creation

Works focusing on automatic Wikipedia text generation and our proposed dataset XWikiRef use article outlines to improve the generation quality. Generating an entire Wikipedia page's structural outline

	hi	mr	bn	or	ta	en	ml	pa	kn	te	Avg
politicians	53.4	63.7	36.6	35.3	32.4	9.4	20.5	27.9	17.7	13.0	31.0
cities	17.5	49.8	15.5	22.8	20.0	18.0	37.0	56.1	24.2	10.7	27.1
books	74.1	40.5	17.8	31.0	12.6	10.6	29.7	32.7	30.5	7.9	28.7
writers	33.9	36.8	15.7	15.2	11.9	7.3	15.0	16.7	11.7	8.3	17.2
companies	28.0	50.8	26.5	36.8	25.0	28.8	26.2	45.7	15.6	20.6	30.4
sportsman	44.2	69.2	22.6	22.8	39.3	14.7	22.7	39.1	16.2	10.5	30.1
films	17.4	20.8	19.9	38.4	17.1	21.8	17.1	28.0	40.9	30.3	25.2
animals	35.0	22.5	18.9	18.3	28.5	12.4	38.4	29.0	16.8	12.7	23.2
Avg	37.9	44.3	21.7	27.6	23.3	15.4	25.8	34.4	21.7	14.2	**26.6**

Table 3.4: Percentage of articles with outline same as the most-frequent outline of (language, domain) pair

is the first step to automating the whole process. Given a (language, domain) pair, a naive approach is to use the most frequently occurring article outline. Table 3.4 shows the percentage of articles with outlines the same as the most frequent outline per (language, domain) for eight domains and ten languages. We observe that, on average, only 26.6% of articles follow the templatized outline. Hence, we see that Wikipedia has a huge diversity across outlines for various entities.

We propose the novel WikiOutlines, a multilingual multidomain dataset across eight domains and ten low-resource languages. The dataset was created to tackle the problem of OutlineGen, which is the task of generating the Wikipedia section outline by condition on (entity, language, domain) triple, and to promote future work on this topic. Each sample in this dataset contains a Wikipedia entity, along with the (language and domain) pairing it belongs to. Corresponding each (entity, language, domain) triple, we provide the article outline in that language, which acts as the target generation for the task.

We start by narrowing down our languages of interest: English (en), Hindi (hi), Bengali (bn), Marathi (mr), Malayalam (ml), Tamil (ta), Telugu (te), Kannada (kn), Punjabi (pa) and Odia (or), and domains of interest: politicians, cities, sportsmen, books, writers, companies, animals and films using Wikidata API. We retrieve the corresponding Wikipedia pages in our chosen languages using language-specific 20221201 XML dumps. The text on a Wikipedia page follows a standardized structure from which we extract sections and subsections.

We split the dataset into train, test, and validation in 80:10:10 ration, and make it available publically.

3.2.3 Comparitive Analysis

Past works for Wikipedia Outline Generation has been limited. As described in Chapter 2, there has only been one paper which has worked on generating article outlines for Wikipedia articles, [2] by Zhang et al. The importance of automatic outline generation has gained more traction as ability of language models to generate texts have improved. We can see a detailed comparison in Table 3.6 between our proposed dataset, WikiOutlines, and existing dataset from WikiOG. We observe that WikiOG is an English-only dataset, containing multiple articles across three domains: cities, celebrities and music. Compared to that, our dataset spans over ten languages and eight domains, with the intersecting domains being cities. Another important detail about WikiOG dataset is the huge number of article size they are covering, with the total being close to $\sim$1.8M. WikiOutlines, on the other hand, only covers about $\sim$166k articles. The main reasoning for this is due to the huge difference in number of articles between English and other low resource languages. To ensure that we are able to represent this difference, as well as ensure that our trained transformer models do not suffer from long-tail classification problem, we reduce the total number of articles.

3.2.4 Data Analysis

We analyze our prepared dataset across several parameters, the details of which are in the following tables and figures.

We show the overall distribution of our dataset in Table 3.5 across all languages and domains, coming out to $\sim$166k articles. We observe that sportsmen and politicians have the largest number of samples, while books and animals have the lowest number of samples. From a language perspective, Odia and Kannada have the lowest number of samples, while Bengali and Tamil are the richest languages.

	hi	**mr**	**bn**	**or**	**ta**	**en**	**ml**	**pa**	**kn**	**te**	**Total**
politicians	6,617	3,815	8,071	1,336	5,885	566	3,405	1,589	699	1,808	33,791
cities	1,048	827	854	268	851	3,550	554	526	256	290	9,024
books	1,428	148	805	87	1,988	762	740	468	105	215	6,746
writers	3,474	1,882	3,605	564	3,005	758	3,475	3,320	1,128	1,339	22,550
companies	683	366	679	38	644	4,546	431	138	212	180	7,917
sportsman	9,476	11,556	13,154	408	9,808	177	2,583	2,327	660	640	50,789
films	4,959	1,033	3,655	920	6,504	1,165	3,934	618	1,704	3,621	28,113
animals	472	395	1,261	142	1,556	427	2,317	200	315	205	7,290
Total	28,157	20,022	32,084	3,763	30,241	11,951	17,439	9,186	5,079	8,298	**166,220**

Table 3.5: WikiOutlines: Total number of samples per domain per language

	WikiOG	WikiOutlines
Number of Articles	1,757,145	166,220
Domains Covered	3 (cities, celebrities, music)	8 (animals, books, cities, companies, films, politicians, sportsmen, writers)
Languages Covered	1 (en)	10 (en, hi, bn, ta, te, ml, kn, mr, or, pa)

Table 3.6: Comparison between WikiOG[2] and WikiOutlines dataset.

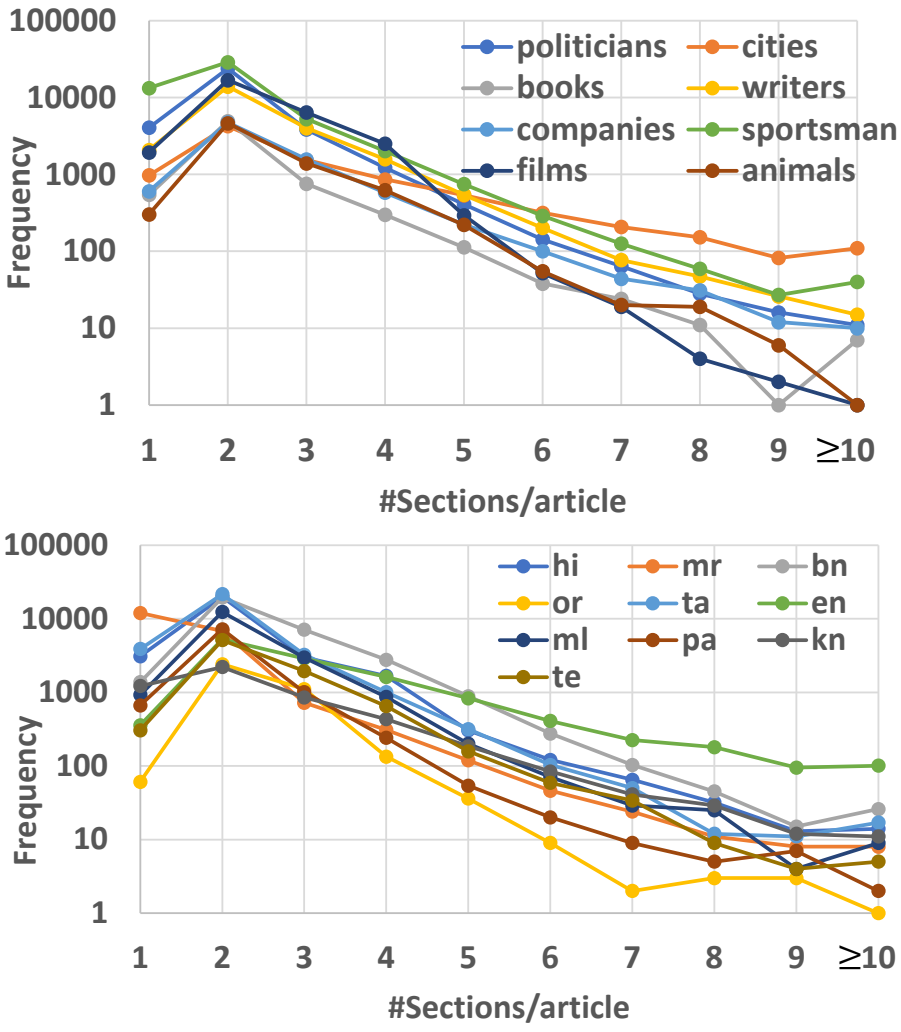

Figure 3.2: Distribution of number of sections across various domains and languages in the WikiOutlines dataset

Next, Fig. 3.2 shows the distribution of number of sections across various domains and languages in the WikiOutlines dataset. Notice that the y-axis is drawn in log scale. The figures show that for every language and every domain, most samples have 2 sections, except for Marathi most samples have just 1 section. Amongst the languages, the distribution is flatest for English, where the number of samples with $\geq$10 sections is the highest. Amongst the domains, cities has a similar behavior.

Finally, we show word clouds of the most frequent Wikipedia section titles for each of the eight domains in Fig. 3.3. Each word cloud contains the five most frequent titles per language. Section titles for one language are shown using a single color. Font size indicates relative frequency. The word clouds show the variety of section titles per (language, domain) pair.

3.3 Conclusion

In this chapter, we introduce two novel datasets: XWikiRef and WikiOutlines.

XWikiRef is designed as a cross-lingual multi-document summarization dataset aimed at facilitating the generation of Wikipedia articles through reference summarization. It comprises of approximately 69,000 articles spanning five domains and eight languages. We envision XWikiRef to become a benchmark for research and future endeavors in automatic Wikipedia Text Generation.

On the other hand, WikiOutlines is a multilingual dataset crafted to streamline the Wikipedia generation process by generating outlines based on minimal information. With around 166,000 articles spanning eight domains and ten languages, WikiOutlines is poised to support authors, editors, and automated tools in their efforts to create outlines for Wikipedia articles across multiple languages and domains.

Figure 3.3: Word clouds of most frequent Wikipedia section titles per domain. Each word cloud contains titles across all languages. Section titles for one language are shown using a single color. Font size indicates relative frequency.

Chapter 4

Cross-lingual Generation of Wikipedia articles

4.1 Introduction

Wikipedia is a repository of reliable knowledge on various topics in multiple languages. However, since each Wikipedia article requires authors to be fluent in the language and multiple editors to ensure the quality, we observed a significantly reduced number of articles in low-resource languages than in English.

To remedy the lack of articles in low-resource languages while not being dependent on native speakers, it is imperative to create an automated pipeline for the cross-lingual generation of Wikipedia articles. Cross-lingual generation helps in leveraging knowledge existing in high-resource or other languages to generate articles in low-resource languages, improving access to reliable information for all.

Generating articles from multi-document summarization helps in the generation of more varied, less biased articles having a lot more reliable information based on source documents than it would if it generated on its own. A cross-lingual multi-document summarisation-based approach makes information more accessible to people, helps in the preservation of their culture and linguistic heritage, as well as bridges the gap in information available in low-resource and high-resource languages.

Our proposal for an automated pipeline is displayed in Figure 4.1, where our input is (Article Title, Article Outline, Target Language, Reference URLs) and output is the generated article.

Overall, we make the following contributions:

- We create a large cross-lingual multi-document dataset, XWikiRef (Chapter 3), and motivate the XWikiGen problem, where the task is to use (Article Title, Article Outline, Target Language, Reference Text) as input to generate a Wikipedia Article.
- We experiment with multiple methodologies and observe that the best-performing model is a two-stage extractive-abstractive summarization pipeline utilizing HipoRank and mBART.

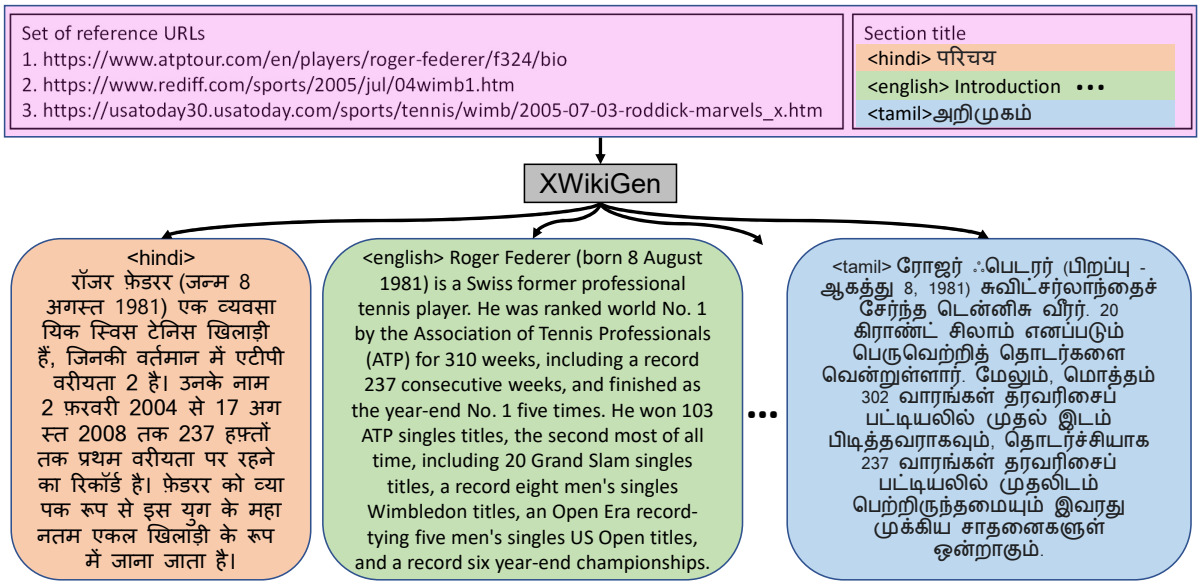

Figure 4.1: XWikiGen examples: Generating Hindi, English, and Tamil text for the Introduction section from cited references.

4.2 Two-Stage Approach for XWikiGen

A Wikipedia article often has multiple references, in different languages spanning over thousands of words. Table 3.3 shows the total number of references per section, and Table 4.1 shows the average number of sentences per reference. Across all (language, domain) pairs, we see that the total number of sentences, hence total text input, comes out to be very large. There exists work like Longformer [46] and Reformer [47] which work on increasing context size and decreasing computation time by reducing quadratic complexity of attention mechanism in transformers. Even with such works, multilingual transformers are not able to practically handle huge context lengths of multiple references in a single go, often leading to large time or poor performance.

To address the problem, we devise a two-stage process. Our goal is to reduce the large amount of general text from input to focused and important sentences after the first stage, which serves as the input to the second stage after which we get our abstractive summary. We think of two possible solutions for the first stage, either create an intermediary dataset on which we can train supervised models or to have a stage of unsupervised summarization. Creating a dataset is expensive, and very hard since manual labeling requires people to know most languages present in the references. Hence, unsupervised summarisation serves as the only viable option in our case.

We discuss the two stages in detail and other experiments we do in the following sections.

	bn	hi	ml	mr	or	pa	ta	en
Books	200.2	117.9	1232.0	225.8	51.9	246.7	302.7	940.8
Films	223.9	320.6	91.9	105.6	345.9	172.6	192.5	1253.6
Politicians	1318.3	467.1	513.3	394.0	54.5	255.4	614.1	1540.9
Sportsmen	335.7	1166.3	406.9	167.5	724.0	253.5	714.0	1535.0
Writers	643.2	2032.5	800.1	385.5	118.5	351.0	1279.0	2061.3

Table 4.1: Average number of sentences in references of a section for each domain and language in XWikiRef.

4.3 Unsupervised Extractive Summarization

For the extractive stage, our input is the set of citation texts, and the output is a subset of those sentences. These sentences are ideally supposed to be the most relevant or salient based on which the next stage of summaries are produced. For reasons mentioned in section 4.2, we experiment with unsupervised extractive summarization. Our first approach is based on QA-GNN [48], and the second is based on HipoRank [49]. We expand on the details of the approach in the following sections.

4.3.1 Using QA-GNN for Salience-Based Extractive Summarization

QA-GNN [48] solves the Question-Answering problem by calculating the relevance score for each answer with respect to the Question-Answer context. We use the same structure to calculate the relevance of each sentence by appending it with the section title (derived from the article outline provided in the input) and calculating the relevance score for it. For Question Answering, they used a monolingual Language Model to calculate the relevance. We change it to a cross-lingual pre-trained XLM-RoBERTa [50] model to better suit our needs.

The relevance score with respect to the section title for each sentence in the reference acts as the saliency of the sentence. We then greedily select the top-K sentences from our ranking to create our extractive summary.

4.3.2 Using HipoRank for Importance-based Extractive Summarization

HipoRank [49] is a methodology used to get extractive summaries of long scientific documents. It generates unsupervised summaries by creating hierarchical graphs using sentences and sections as the two levels of nodes. To calculate the importance score of each sentence node, they asymmetrically weigh edges connecting intra-sectional and inter-sectional nodes and combine them to get a node-level score.

HipoRank was developed for English scientific document summarization, hence, to tweak it for our use case, we use mBERT [51] for sentence-level node representation. Similar to HipoRank, we compute section-level node representation using the mean representation of all its sentence-level representations. We connect the sentence-level nodes to signify the local level importance of each node and connect sentence and section-level nodes to signify the global level importance of each node. To reduce complexity, edges connected two sentence-level nodes across sections are not allowed.

The nodes are given a score based on a combination of intra-section and inter-section level edge weight, which in turn are calculated based on cosine similarity of node representations. We also add a parameter based on closeness to the boundary of a section (beginning or end) based on the heuristic that sentences closer to the section ending and beginning are more important. Combining edge connections and boundary parameter, we get a score that represents the importance of each sentence node. We then greedily select top-K nodes based on these scores, which act as our extractive summary.

4.4 Supervised Cross-lingual Abstractive Summarization

Although necessary, the first stage of unsupervised extractive summarization also brings up a few problems. Firstly, since it is purely an extractive summary, the sentences extracted remain in the language as that of the reference text. Secondly, these sentences are selected with no order in mind, hence the extracted summary is not coherent. To solve these problems, we need a cross-lingual abstractive summarization step, to create a coherent summary in our target language.

We look at two of the best cross-lingual seq-2-seq models for our use case, mBART [21] and mT5 [13]. mBART is an encoder-decoder transformer model pretrained on CommonCrawl dataset [3], while mT5 is also an encoder-decoder transformer model pretrained on mC4 dataset [1]. Both models have proven to perform effectively across various multilingual and cross-lingual NLP tasks, and we use both of them in our experiments. We use mBART-large variant having 12 layers each for the encoder and decoder, with a total parameter size of 610.87 million, and mt5-base has 12 layers for the encoder and decoder with a parameter size of 582.4 million. Hence, we ensure similar parameter sizes between both models for fair experimentation.

Input to both models is from the training set of XWikiRef dataset, where the input is (Article Title, Section Title, Target Language, Reference Texts). We use this input to generate the Wikipedia article section by section.

[1] https://www.tensorflow.org/datasets/catalog/c4#c4multi-lingual_nights_stay

4.5 Experimental Setup

4.5.1 Training Setup

For our first stage unsupervised extractive summarization, we perform computations on one GPU, NVIDIA 2080Ti with 12GB of GPU RAM. QA-GNN based extractive summarization (subsection 4.3.1) requires us to use XLM-RoBERTa-base [50] for getting sentence-section title relevance score. In HipoRank based extractive summarization (subsection 4.3.2), we use mBERT [51] to get sentence representations. In both cases, we keep a maximum input length of 512 tokens, and a value of k to select top-K sentences as fifty.

In our abstractive stage, we train our supervised model in a multi-GPU setting, utilizing NVIDIA V100 with 32GB of GPU RAM. We use 'google/mt5-base' and 'facebook/mbart-large-50' from huggingface to get checkpoints of mT5 [13] and mBART [21] respectively. We keep a similar training configuration in both cases, training the model for twenty epochs on a batch size of four on AdamW optimizer with 1e-5 as the learning rate. In this case, as well, we keep a maximum input length of 512 tokens.

4.5.2 Metrics Used

To measure our model's performance on the XWikiRef dataset, we benchmark our performance using standard text generation metrics.

- Rouge-L: It is an n-gram overlap-based metric to determine the correctness of generated text as compared to gold text. In Rouge-L, the value of n for n-gram is the longest co-occurring n-grams (Longest Common Subsequence) between prediction and reference.
- CHRF++: Another word overlap-based metric, but in this case CHRF++ includes F-score for character level n-gram overlaps. It is built on top of chrf metric, with additional provisions to account for cases number of characters differs in prediction and reference.
- METEOR: It is a more complicated metric, often used to represent more information than simple token matching. It improves word matching between prediction and reference using synonyms, stemming, word-order swapping and paraphrasing.

4.6 Multi-domain and Multilingual Setups

Our dataset spans eight languages and five domains, which leads to the question of how to train our model. Possible options include training language-wise, domain-wise, (language, domain) pair-wise or all together. In the case of having individual model per (language, domain) pair, we see that having 8X5, or 40, models will be too inefficient, and will not benefit cross-lingual or cross-domain learning. Going

through other options, we see that others are more feasible, with language-wise models needing eight models in total and domain-wise models needing five models in total.

Based on previous literature, a single multi-lingual model performs better than multiple individual models due to cross-lingual learning. With that in mind, we conduct the following experiments for language-wise, domain-wise and combined setups to get the best possible results.

4.7 Results

To recall, we have multiple parameters across which we conduct our experiment. They include the choice of extractive summarization (between QA-GNN and HipoRank), the choice of model of abstractive summarization (between mT5 and mBART), different setups based on language and domain choices (multilingual, multi-domain, multilingual and multi-domain). For all these experiments, we compute the score for three standard text-generation metrics, Rouge-L, METEOR and CHRF++ across XWikiRef, and present it in Table 5.1.

Based on scores across all parameters and training setups, we observe that multilingual and multi-domain setup together consistently outperforms only multilingual and only multi-domain methods. This lies in with observations from previous work related to knowledge learned across languages and domains.

	Extractive	Abstractive	ROUGE-L	chrF++	METEOR
Multi-lingual	Salience	mBART	15.59	17.20	10.98
	Salience	mT5	14.66	15.45	8.92
	HipoRank	mBART	**16.96**	**19.11**	**12.19**
	HipoRank	mT5	15.98	17.11	10.08
Multi-domain	Salience	mBART	**19.88**	**22.82**	**15.00**
	Salience	mT5	12.13	13.66	7.27
	HipoRank	mBART	18.87	20.79	14.10
	HipoRank	mT5	12.29	13.93	7.36
Multi-lingual-multi-domain	Salience	mBART	20.50	22.32	14.81
	Salience	mT5	17.31	18.77	11.57
	HipoRank	mBART	**21.04**	**23.44**	**15.35**
	HipoRank	mT5	17.65	19.04	11.74

Table 4.2: Results of XWikiGen across all different training setups. Highlighted in bold are the best results per block, and underlined results are the best results overall.

	bn	en	hi	mr	ml	or	pa	ta
ROUGE-L	14.49	7.46	29.01	20.67	12.25	25.54	16.89	17.09
chrF++	18.58	10.55	28.38	20.41	15.30	27.31	13.49	21.90
METEOR	9.71	5.90	25.24	13.72	6.42	22.69	10.12	9.87

Multi-lingual HipoRank+mBART

	bn	en	hi	mr	ml	or	pa	ta
ROUGE-L	15.30	12.07	36.16	31.25	14.22	29.53	16.91	15.00
chrF++	19.40	17.41	34.34	32.50	18.34	32.20	14.10	21.65
METEOR	10.34	9.59	31.02	24.86	8.89	26.86	10.01	9.29

Multi-domain Salience+mBART

	bn	en	hi	mr	ml	or	pa	ta
ROUGE-L	15.21	16.32	36.38	22.71	15.50	27.41	18.64	18.87
chrF++	19.50	21.34	34.55	21.93	18.65	28.83	16.27	23.99
METEOR	10.24	12.74	31.24	14.88	8.84	23.93	11.6	11.26

Multi-lingual-multi-domain HipoRank+mBART

Table 4.3: Detailed per-language results on test part of XWikiRef, for the best model per training setup.

Overall, we see that the multi-lingual multi-domain method with HipoRank and mBART provides the best results across all metrics. We see that HipoRank also mostly outperforms the QA-GNN based extractive summarization, although the latter performs better in the case of multi-domain setup.

We also provide more specific results for language-wise and domain-wise scores in Tables 4.3 and 4.4 respectively. We observe that in most cases, multi-domain models perform much better than multilingual models, with the exception of Tamil (ta). Perhaps counter-intuitively, we observe that languages like English (en) and Hindi (hi) which are better represented on the internet benefit more from shifting to a multilingual multi-domain setup from a pure multi-lingual setup. Generally, multilingual multi-domain setup proves to be beneficial compared to multilingual or multi-domain setup, but for Marathi (mr) and Odia (or) among languages and sportsmen in domains, which show slight losses when shifting to multilingual multi-domain setup.

We also provide scores specific to each (language, domain) pair for our best-performing model in Tables 4.5, 4.6 and 4.7. We see that overall, our works best for Hindi (hi), and reasonably well for Marathi (mr) and Oriya (or). We also note the need for improved performance in Bengali (bn) and Malayalam (ml).

	writers	books	sportsmen	politicians	films
ROUGE-L	10.12	3.65	20.61	22.01	14.60
chrF++	10.76	3.58	22.94	24.34	18.36
METEOR	5.77	1.93	14.66	17.61	10.04

Multi-lingual HipoRank+mBART

	writers	books	sportsmen	politicians	films
ROUGE-L	14.21	20.17	20.65	22.77	20.82
chrF++	17.24	21.86	22.75	26.14	24.30
METEOR	10.06	16.26	14.71	18.88	14.81

Multi-domain Salience+mBART

	writers	books	sportsmen	politicians	films
ROUGE-L	14.67	22.03	20.44	23.70	21.60
chrF++	16.65	22.81	21.57	25.75	24.51
METEOR	9.81	17.55	13.84	18.92	15.11

Multi-lingual-multi-domain HipoRank+mBART

Table 4.4: Detailed per-domain results on test part of XWikiRef, for the best model per training setup.

	ROUGE-L				
	writers	**books**	**sportsmen**	**politicians**	**films**
bn	10.61	9.43	15.78	17.46	15.75
en	13.04	15.62	18.53	13.32	20.15
hi	33.23	58.71	28.48	53.18	21.46
mr	15.37	17.00	26.77	20.06	24.15
ml	8.96	10.93	12.97	14.36	24.19
or	13.15	12.31	9.38	43.76	26.66
pa	14.96	12.35	24.54	16.59	17.15
ta	10.62	11.85	18.94	19.18	24.90

Table 4.5: Detailed results (ROUGE-L) for every (domain, language) partition of the test set of our XWikiRef dataset, for our best XWikiGen model: Multi-lingual-multi-domain HipoRank+mBART.

	chrF++				
	writers	**books**	**sportsmen**	**politicians**	**films**
bn	14.72	14.19	20.28	21.21	20.03
en	19.71	18.90	22.80	20.00	24.13
hi	31.05	51.99	26.99	52.05	19.64
mr	14.68	16.24	26.84	18.12	21.82
ml	13.35	12.18	15.42	18.01	26.51
or	14.44	15.16	10.51	44.17	29.27
pa	13.42	12.39	21.32	14.02	13.82
ta	16.43	17.63	23.98	23.77	29.94

Table 4.6: Detailed results (chrF++) for every (domain, language) partition of the test set of our XWikiRef dataset, for our best XWikiGen model: Multi-lingual-multi-domain HipoRank+mBART.

	METEOR				
	writers	**books**	**sportsmen**	**politicians**	**films**
bn	6.13	5.66	10.56	12.99	10.39
en	10.65	11.62	13.89	11.47	15.09
hi	28.49	53.78	21.46	51.65	15.30
mr	7.40	9.50	20.14	10.74	14.30
ml	3.92	4.77	6.14	7.73	16.16
or	5.67	9.14	5.28	40.89	23.30
pa	8.59	7.48	16.54	9.80	9.63
ta	4.89	6.29	10.03	11.24	17.05

Table 4.7: Detailed results (METEOR) for every (domain, language) partition of the test set of our XWikiRef dataset, for our best XWikiGen model: Multi-lingual-multi-domain HipoRank+mBART.

We also qualitatively verify the outputs from our best model across languages and domains. We show these in Table **??**, 4.9 and 4.10, where we observe that our model, in general, is coherent and fluent to an extent. We see common errors associated with Transformers in them, like repeated string tokens that break sentence structure, and hallucinations causing inconsistent facts mentioned in references

Domain	Reference URLs	Entity/ Section	Lang	Reference Text	Generated Summary
Films	• http://www.thehindu.com/features/cinema/cinema-columns/blast-from-the-past-chandra-mohana-or-samooga-thondu-1936/article5668076.ece	சந்திர மோகனா (திரைப்படம்)/அறிமுகம்	ta	சந்திரமோகனா என்பது 1936 ஆம் ஆண்டு வெளிவந்த தமிழ்த் திரைப்படமாகும். முதுரை மோகன் மூவிடோன் நிறுவனத்தினரின் தயாரிப்பில் வெளிவந்த இத்திரைப்படத்தில் எம். கே. ராதா, எஸ். வி. வெங்கட்ராமன் மற்றும் பலரும் நடித்துள்ளனர். இப்படமானது கொத்தமங்கலம் எஸ். எம். சுப்ரமண்யம் என்று அப்போது அழைக்கப்பட்ட கொத்தமங்கலம் சுப்பு எழுதிய சந்திரமோகனா அல்லது சமுதாயத் தொண்டு என்ற புதினத்தை அடிப்படையாகக் கொண்டு எடுக்கப்பட்டது.	சந்திர மோகனா (Chandra Mohana) 1947 ஆம் ஆண்டு வெளிவந்த தமிழ்த் திரைப்படமாகும். எஸ். எஸ். வாசன் இயக்கத்தில் வெளிவந்த இத்திரைப்படத்தில் எம். கே. ராதா, டி. ஆர். ராமச்சந்திரன் மற்றும் பலரும் நடித்திருந்தனர்.
Sportsmen	• https://www.icc-cricket.com/news/1939383 • https://www.icc-cricket.com/media-releases/1212091 • http://www.espncricinfo.com/story/_/id/25033717/associates-pathway-2023-world-cup-undergoes-major-revamp	2021 पापुआ न्यू गिनी त्रिकोणी सीरीज (मई)/परिचय	hi	2021 पापुआ न्यू गिनी त्रिकोणी सीरीज़ 2019–2023 आईसीसी क्रिकेट विश्व कप लीग 2 क्रिकेट टूर्नामेंट का 8 वां दौर होने वाला था, जो मई 2021 में पापुआ न्यू गिनी में खेला जाना था। यह नामीबिया, पापुआ न्यू गिनी और संयुक्त राज्य अमेरिका की क्रिकेट टीमों के बीच एक त्रिकोणीय राष्ट्र श्रृंखला होती, जिसमें मैच एक दिवसीय अंतर्राष्ट्रीय (वनडे) जुड़नार के रूप में खेले जाते थे। आईसीसी क्रिकेट विश्व कप लीग 2 2023 क्रिकेट विश्व कप के लिए योग्यता मार्ग का हिस्सा है। हालाँकि, 12 फरवरी 2021 को, कोविड-19 महामारी के कारण श्रृंखला को स्थगित कर दिया गया था।	2021 पापुआ न्यू गिनी ट्राई-नेशन सीरीज़ एक क्रिकेट टूर्नामेंट था जो सितंबर 2021 में पापुआ न्यू गिनी में खेला गया था। यह पापुआ न्यू गिनी क्रिकेट टीम और पापुआ न्यू गिनी क्रिकेट बोर्ड (पीएनसीबी) के बीच एक त्रिकोणीय राष्ट्र श्रृंखला थी, जिसमें पापुआ न्यू गिनी और संयुक्त राज्य अमेरिका के बीच एक दिवसीय अंतर्राष्ट्रीय (वनडे) मैच खेला गया था। यह श्रृंखला सितंबर 2021 में होने वाली थी, लेकिन कोविड-19 महामारी के कारण इसे स्थगित कर दिया गया था।

Table 4.8: Examples of XWikiGen using our best model in Sportsmen and Films domain.

and predictions. Both problems are well documented when it comes to cross-lingual generation, and increasing the training data is known to alleviate it.

4.8 Conclusion

In this chapter, we motivate the problem XWikiGen, a cross-lingual multi-document summarization task for generation Wikipedia articles automatically using references. Since references can be large and in multiple languages, our methods and models must be cross-lingual in nature. Due to the large context size of references, we propose a two-stage summarization pipeline, where the first stage is an unsupervised extractive summarization to reduce the length of references to the most important few sentences. We then use these sentences as input to our multilingual transformers (mT5 and mBART) for supervised abstractive summarization to generate the article section by section.

We also introduce the dataset XWikiRef across eight languages and five domains, spanning approximately 69,000 articles. Using this dataset, we benchmark our results with Rouge-L, CHRF++ and METEOR as our metrics. We observe that in most cases, HipoRank + mBART performs the best for generating the Wikipedia article. We also provide analysis of both quantitative and qualitative results. To encourage further research in this topic, we make our code and dataset public.

Domain	Reference URLs	Entity/ Section	Lang	Reference Text	Generated Summary
Books	• https://www.nytimes.com/books/97/07/13/reviews/970713.13polklt.html	കോൾഡ് മൌണ്ടൻ/ആമുഖം	ml	കോൾഡ് മൌണ്ടൻ 1997 ൽ പ്രസിദ്ധീകരിക്കപ്പെട്ട ചാൾസ് ഫ്രേസിയർ എഴുതിയ ഒരു ചരിത്ര നോവലാണ്. ഈ നോവലിന് യു.എസ്. നാഷണൽ അവാർഡ് ഫോർ ഫിക്ഷൻ എന്ന പുരസ്കാരം ലഭിച്ചിട്ടുണ്ട്. അമേരിക്കൻ ആഭ്യന്തരയുദ്ധത്തിൻറെ അവസാനകാലത്ത് യുദ്ധത്തിൽ പങ്കെടുത്ത കോൺഫെഡറേറ്റ് സേനയിലെ ഡബ്ലിയു. പി. ഇൻമാൻ എന്ന മുറിവേറ്റ സൈനികനാണ് ഈ കഥയിലെ കേന്ദ്രകഥാപാത്രം. അദ്ദേഹം അഡ മൺറോയെന്ന തൻറെ പ്രണയിനിയുടെ അടുത്തു തിരിച്ചെത്തുവാനായി മാസങ്ങളോളം ഏകനായി നടത്തുന്ന യാത്രയാണ് കഥയുടെ ഇതിവൃത്തം. ഹോമറിൻറെ ഒഡീസിയുമായി ഈ നോവലിന് അനേകം സാദൃശ്യങ്ങളുണ്ട്.	ഒരു അമേരിക്കൻ നോവലിസ്റ്റാണ് കോൾഡ് മൌണ്ടൻ (ജീവിതകാലം: 1798–1831). അദ്ദേഹത്തിന്റെ ഏറ്റവും പ്രശസ്തമായ നോവലായ കോൾഡ് മൌണ്ടൻ, ന്യൂയോർക്ക് ടൈംസിന്റെ 1997 ലെ മികച്ച നോവലിനുള്ള അവാർഡ് നേടി.
Politicians	• https://timesofindia.indiatimes.com/city/patna/all-eyes-on-4-seats-in-khagaria-district/articleshow/78939785.cms • https://web.archive.org/web/20201204042405/https://www.jansatta.com/lifestyle/bihar-election-ljp-khagaria-candidate-renu-kumari-kushwaha-spent-24-and-half-lakhs-on-jewelry-and-cars-owns-this-much-property/1564879/ • http://164.100.47.194/loksabha/Members/memberbioprofile.aspx?mpsno=537&lastls=13	ਰੇਣੂ ਕੁਸ਼ਵਾਹਾ/ਜਾਣ-ਪਛਾਣ	pa	ਰੇਣੂ ਕੁਸ਼ਵਾਹਾ (ਜਿਸ ਨੂੰ ਰੇਣੂ ਕੁਮਾਰੀ ਸਿੰਘ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ), ਇੱਕ ਭਾਰਤੀ ਸਿਆਸਤਦਾਨ, ਲੋਕ ਜਨਸ਼ਕਤੀ ਪਾਰਟੀ ਦੀ ਆਗੂ ਅਤੇ ਬਿਹਾਰ ਦੀ ਇੱਕ ਸਾਬਕਾ ਰਾਜ ਮੰਤਰੀ ਹੈ। ਉਹ ਖਗੜੀਆ ਦੀ ਰਹਿਣ ਵਾਲੀ ਹੈ। ਉਹ ਅਤੀਤ ਵਿੱਚ ਸਮਤਾ ਪਾਰਟੀ ਅਤੇ ਜਨਤਾ ਦਲ (ਯੂਨਾਈਟਿਡ) ਤੋਂ ਲੈ ਕੇ ਭਾਰਤੀ ਜਨਤਾ ਪਾਰਟੀ ਤੱਕ ਕਈ ਸਿਆਸੀ ਪਾਰਟੀਆਂ ਨਾਲ ਜੁੜੀ ਰਹੀ ਹੈ। 2015 ਵਿੱਚ ਉਸਨੇ ਭਾਜਪਾ ਦੀ ਟਿਕਟ 'ਤੇ ਸਮਸਤੀਪੁਰ ਵਿਧਾਨ ਸਭਾ ਸੀਟ ਤੋਂ ਚੋਣ ਲੜੀ, ਜਿੱਥੇ ਉਹ ਆਰ.ਜੇ.ਡੀ. ਉਮੀਦਵਾਰ ਤੋਂ ਹਾਰ ਗਈ।	ਰੇਣੂ ਕੁਸ਼ਵਾਹਾ ਇੱਕ ਭਾਰਤੀ ਸਿਆਸਤਦਾਨ ਹੈ। ਉਹ ਭਾਰਤੀ ਜਨਤਾ ਪਾਰਟੀ ਦਾ ਮੈਂਬਰ ਹੈ ਅਤੇ ਭਾਰਤੀ ਜਨਤਾ ਪਾਰਟੀ ਦੀ ਨੁਮਾਇੰਦਗੀ ਕਰਦਾ ਹੈ।

Table 4.9: Examples of XWikiGen using our best model in Books and Politicians domain.

Domain	Reference URLs	Entity/ Section	Lang	Reference Text	Generated Summary
Writers	• http://www.columbia.edu/itc/mealac/pritchett/00ambedkar/timeline/graphics/youth.html • http://www.columbia.edu/itc/mealac/pritchett/00ambedkar/timeline/1890s.html • http://www.columbia.edu/itc/mealac/pritchett/00ambedkar/txt_ambedkar_waiting.html	ভীমরাও রামজী শাকপাল/প্রথম জীবন এবং শিক্ষা	bn	ভীমরাও রামজী শাকপাল যুবক থাকাকালীন 'মোহ' (Mhow) অঞ্চলের (বর্তমান মধ্য প্রদেশ) এবং কেন্দ্রীয় সামরিক সেনানিবাসে ব্রিটিশ কর্তৃক স্থাপিত শহরে আম্বেদকর জন্মগ্রহণ করেছিলেন। তিনি ছিলেন রামজী মালোজী শাকপাল (Ramji Maloji Sakpal) এবং ভীমাবাইের (Bhimabai) ১৪তম তথা সর্বকনিষ্ঠ পুত্র। তার পরিবার ছিলেন মারাঠী অধ্যুষিত বর্তমান কালের "মহারাষ্ট্র"-এর রত্নগিরি জেলার "আম্বোভাদ" (Ambavade) শহরে। তারা হিন্দু সম্প্রদায়ের অধিভুক্ত ছিল (মহর জাতি), যারা অস্পৃশ্য জাতি হিসেবে এবং প্রচণ্ড আর্থ-সামাজিক বৈষম্যের শিকার হত। আম্বেদকরের পূর্বপুরুষেরা ছিলেন ব্রিটিশ ইস্ট – ইন্ডিয়া কোম্পানির সেনা এবং তার পিতা "রামজী শাকপাল" মোহ সেনানিবাসের ভারতীয় সেনা হিসেবে নিযুক্ত ছিলেন, তিনি সেকালের গৎবাঁধা শিক্ষাপদ্ধতিতে মারাঠী এবং ইংরেজিতে ডিগ্রি লাভ করেছিলেন এবং সেইসাথে তিনি প্রাথমিক বিদ্যালয়ের শিক্ষা লাভে কঠোর পরিশ্রমে সন্তানদের উদ্বুদ্ধ করেন। কবির পান্ডের মতে, রামজী শাকপাল তার সন্তানদের হিন্দু সংস্কৃতি সম্পর্কে অধ্যয়ন করতে উদ্বুদ্ধ করতেন। যদিও আম্বেদকর বিদ্যালয়ে যেতেন, তাকে অন্যান্য অস্পৃশ্য শিশুর ন্যায় আলাদা করে দেয়া হত। শিক্ষকগণ তাদের প্রতি অমনোযোগী ছিলেন এবং কোনোরূপ সহযোগিতাপূর্ণ মনোভাব পোষণ করতেন না। তাদের শ্রেণিকক্ষের ভেতরে বসার অনুমতি ছিলো না, এমনকি তাদের যদি তৃষ্ণা পেতো উচ্চবর্ণের কোনো একজন এমন উচ্চতা হতে সেই পানি ঢেলে পান করাতো, যাতে নিচুজাতের শিক্ষার্থীরা বা পানি বা পানির পাত্র স্পর্শ না করতে পারে । এই কাজটি সাধারণত আম্বেদকরের অন্য করতো বিদ্যালয়ের চাপরাসী (Peon) এবং যদি পিওন না থাকত বা না আসত, তখন সারাদিন পানি ছাড়াই কাটাতে হতো, আম্বেদকর এই অবস্থাকে এভাবে আখ	ভীমরাও রামজি আম্বেদকর ১৯৪০ সালের ২৪শে নভেম্বর কলকাতায় জন্মগ্রহণ করেন। তার পিতার নাম ফারুক আম্বেদকর এবং মাতার নাম অঞ্জলিনা আম্বেদকর। তিনি কলকাতা বিশ্ববিদ্যালয় থেকে স্নাতক ডিগ্রি অর্জন করেন। এরপর তিনি কলকাতা বিশ্ববিদ্যালয়ে অধ্যাপনা শুরু করেন। এরপর তিনি কলকাতা বিশ্ববিদ্যালয় থেকে স্নাতকোত্তর ডিগ্রি অর্জন করেন।
Films	• https://www.rogerebert.com/reviews/stranger-than-paradise-1984	Stranger Than Paradise/Introduction	en	Stranger Than Paradise is a 1984 American black-and-white absurdist deadpan comedy film, co-written, directed and co-edited by Jim Jarmusch, and starring jazz musician John Lurie, former Sonic Youth drummer-turned-actor Richard Edson, and Hungarian-born actress and violinist Eszter Balint. It features a minimalist plot in which the main character, Willie, is visited by Eva, his cousin from Hungary. Eva stays with him for ten days before going to Cleveland. Willie and his friend Eddie go to Cleveland to visit her, and the three then take a trip to Florida. The film is shot entirely in single long takes with no standard coverage.	Stranger Than Paradise is a 1984 American drama film directed and co-written by Roger Ebert and starring John Lurie and John Travolta. The film tells the story of Willie (Lurie) and his cousin Eva (Travolta) traveling from Hungary to New York City in search of a place to stay. The film received positive reviews from critics, with praise for Lurie's performance, his direction, and the film's screenplay. It was nominated for the Academy Award for Best Original Screenplay.

Table 4.10: Examples of XWikiGen using our best model in Writers and Films domain.

Chapter 5

Multilingual Generation of Wikipedia Outlines

5.1 Introduction

Wikipedia has been the most popular source of factual and neutral encyclopedic information for millions of users. Although English Wikipedia is rich with ∼7M articles, number of Wikipedia pages in nine low resource (LR) languages which we consider in our work add to ∼100K. Unfortunately, recent efforts towards enriching LR Wikipedia over the years have also not been as encouraging as for English as mentioned in Chapter 4. Hence, automated text generation for low-resource Wikipedia is critical.

One of the methods discussed for automatic Wikipedia article generation in Chapter 4 for low-resource languages was translation. A key reason, as mentioned previously, why it does not work is due to the number of pages existing in low-resource languages without a corresponding page in English or other high-resource languages. Table 1.1 in Chapter 1 shows this in detail, and expands on the need for automatic generation of articles.

To generate an entire Wikipedia page, it is important to first generate the structural outline and then fill the sections with LR language text using these existing methods. We discuss the frequency of templatized outlines observed in Wikipedia pages across languages and domains in Chapter 3, Table 3.4, which shows that 26.6% of all articles follow templatized outlines. Thus, we need to design a method that generates the Wikipedia section outline by conditioning on (entity, language, domain) triple. Again, translating the outline from the corresponding English Wikipedia page is not effective because (1) several LR pages on Wikipedia do not have equivalent pages on English Wikipedia, and (2) Often, LR Wikipedia exists for LR communities. Here, the pages are written by LR language editors for people of the LR communities. Hence, their outlines differ significantly from the outlines for corresponding Wikipedia pages (if they exist).

Hence, we propose the task of Outline Generation, OutlineGen, for Wikipedia articles, which is a novel task to generate Wikipedia-styled outlines given an article's (entity, language, domain) triple. Since our goal is to generate Wikipedia outlines for entities where no Wikipedia page already exists, we take minimal inputs (entity, language, domain) for the task. Fig. 5.1 shows examples of OutlineGen task for the "Roger Federer" entity (which belongs to the sportsman domain) for English, Bengali and Telugu

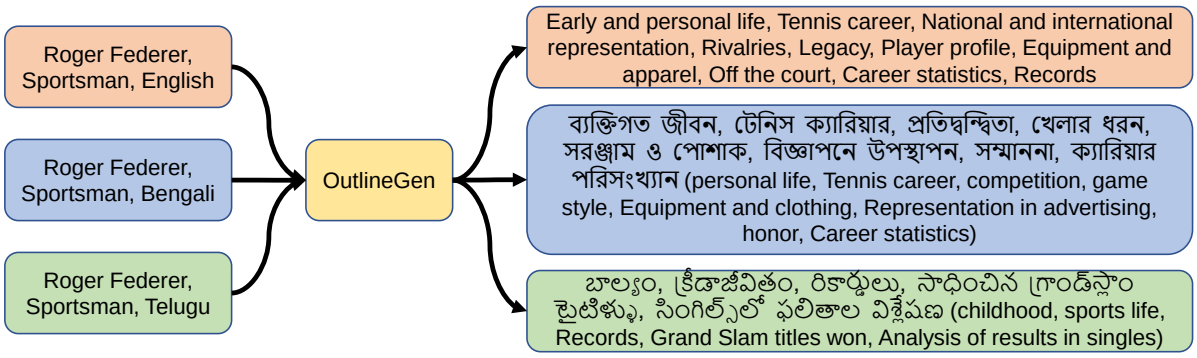

Figure 5.1: OutlineGen examples: Generating outlines for the "Roger Federer" entity (which belongs to the sportsman domain) for English, Bengali and Telugu Wikipedia pages.

Wikipedia pages. These outlines could help human editors to plan the article content better. These outlines could also help improve the quality of the automatically generated text (using methods like [52, 14, 53, 54]) and hence reduce human post-editing efforts.

For the OutlineGen task, we benchmark our results against the WikiOutlines dataset mentioned in Chapter 3. WikiOutlines, which contains Wikipedia section outlines from ∼166K Wikipedia pages across 8 domains and 10 languages. The domains include politicians, cities, books, writers, companies, sportsman, films and animals. Languages include Hindi (hi), Marathi (mr), Bengali (bn), Odia (or), Tamil (ta), English (en), Malayalam (ml), Punjabi (pa), Kannada (kn) and Telugu (te).

Overall, we make the following contributions:

- We define and motivate the need for the novel OutlineGen task, where the input is minimal (entity, target language, and domain). The output is the wikipedia-style outline.
- We experiment with multiple methodologies, observing that our best-performing model is a multilingual generative model using mt5 [13].

5.2 Approaches for OutlineGen Task

Two promising methods can act as reasonable solutions for the OutlineGen task. The first solution involves building and usage of (language, domain) specific weighted finite state automata (WFSA). The second solution involves providing (entity, language, domain) as input to a Transformer [55]-based encoder-decoder multi-lingual model to generate outlines. Since the second solution is conditional on the entity, we expect it to perform better compared to the first solution.

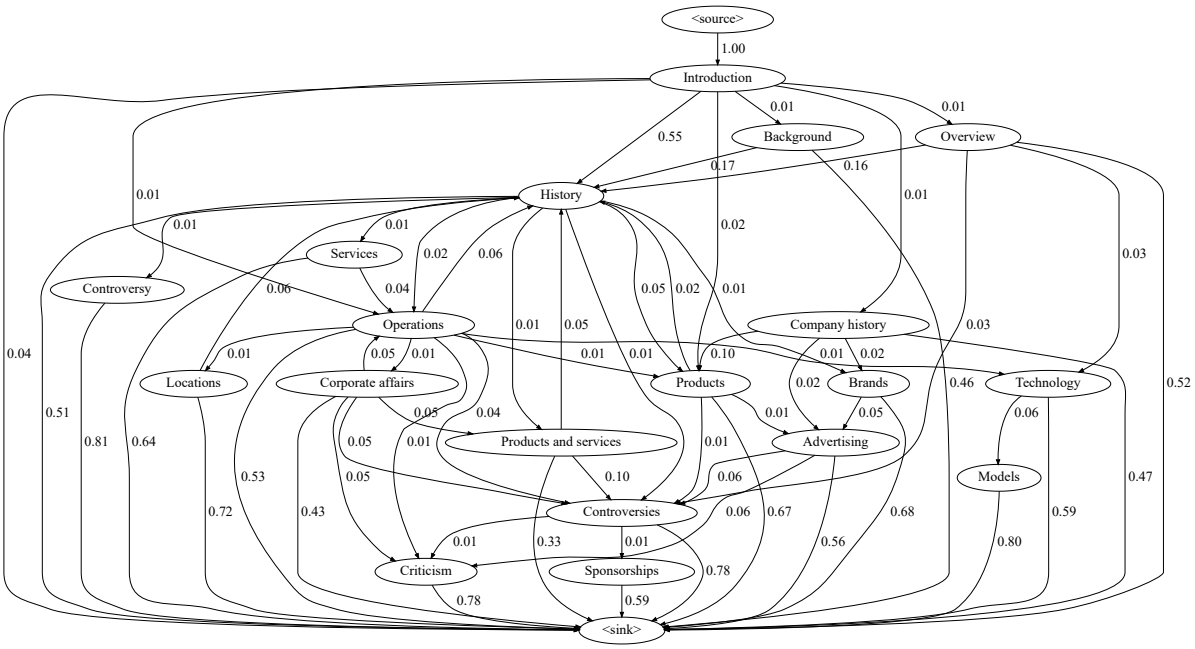

Figure 5.2: Example of generated weighted finite state automata, where section-titles are the nodes, and transition probability is written on the edges. This is for (en, companies).

5.2.1 Weighted Finite State Automata (WFSA)

Table 3.4 shows that many articles share the same outline. These article outlines are often specific for a language over a particular domain, and the section transition patterns can potentially be found via simple statistical models instead of large generative ones. Hence, instead of defining static outlines based simply on frequency, we learn a weighted finite state automata for all articles belonging to a (language, domain) pair. The source node for the WFSA is ⟨source⟩, and the sink node is represented by ⟨sink⟩. The nodes between the source and the sink contain the section titles, and the transition probability from node A to node B is the conditional probability of section title B following section title A in an article outline. Fig. 5.2, 5.3 and **??** shows three examples of WFSA learned for (en, companies), (hi, sportsman) and (ml, films) pairs. These are drawn using top 20 most frequent section titles for that (language, domain) pair as nodes. Also, edges with a weight more than 0.005 are shown.

WFSA involves two hyper-parameters: (i) a *beam-size* (samples from top-k instead of choosing the most probable next state), and (ii) *token-level* (word or section-title level WFSA).

The WFSA is used for inference as follows. We start from ⟨source⟩ state and select *beam-size* number of next possible states. We base our selection either greedily (selecting the most probable next states) or by sampling them from a probability distribution over the next states. We repeat these instructions recursively (in a breadth-first manner), maintaining the visited part of the outline in a queue and the total accumulated transition probability. Once we reach the ⟨sink⟩ state, we terminate the recursion and

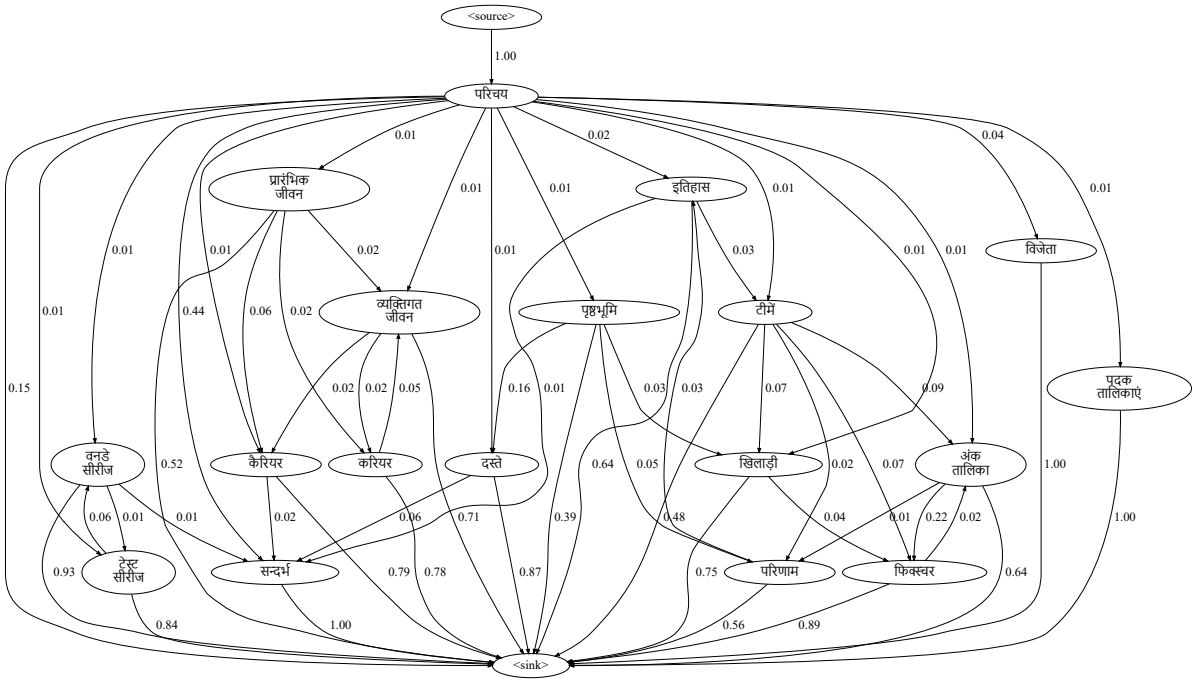

Figure 5.3: Example of generated weighted finite state automata, where section-titles are the nodes, and transition probability is written on the edges. This is for (hi, sportsman).

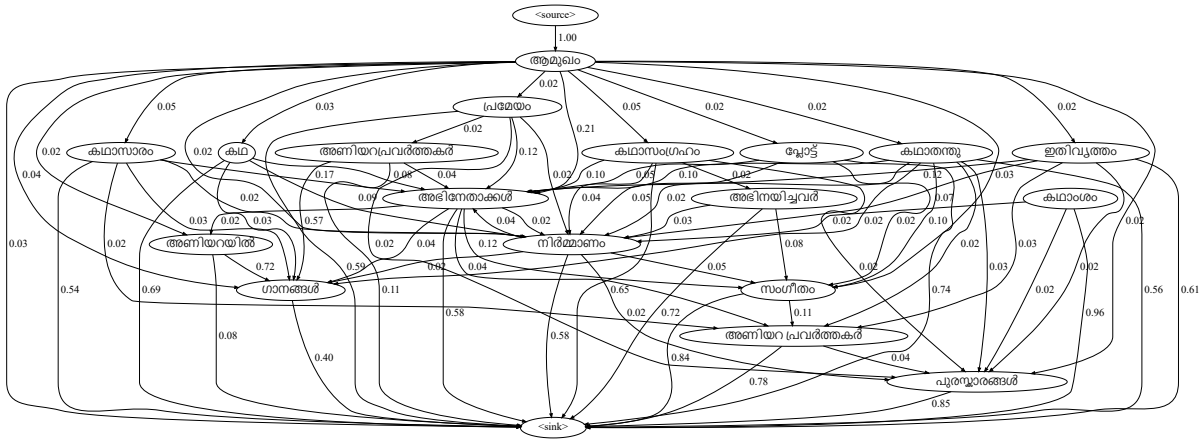

Figure 5.4: Example of generated weighted finite state automata, where section-titles are the nodes, and transition probability is written on the edges. This is for (ml, films).

store the generated outline with the geometric mean of transition probabilities signifying the probability of that outline occurring. We terminate when the breadth-first search queue is empty. Outline with the highest probability is selected as the output. Of course, we ensure that the generated outline does not have repeated section titles.

We also experiment with another kind of WFSA, QueryBlazer [56], a query completion model used to predict query completion to the user's incomplete query input. It uses an n-gram model at a subword level to create its FSA and predict the user output. We model our data in a query format, where the query consists of language and domain, and the outline is set as the 'completion' of the query. At inference time, we generate prediction 'completions' as outlines for each (language, domain) pair. We observe that QueryBlazer performs worse than normal WFSA, hence we do not compute other metrics for it.

5.2.2 Multi-lingual Transformer Generative Models

WFSA-based methods are not entity-specific. This restricts them to predict the same outline for all entities belonging to the same (language, domain) pair. Hence, we also experimented with popular multi-lingual Transformer encoder-decoder generative models like mT5 [13] and mBART [21]. The language and domain are passed as input with a separator token. The models are fine-tuned to generate outlines. The model is now supposed to automatically decide the number of sections in the outline and the actual section titles in the outline as well.

5.3 Experiments and Results

5.3.1 Training Setup

We trained both WFSA and Transformer-based models in training data and tuned hyper-parameters on the validation set. For WFSA, we found beam size=4 to provide best result on validation set. For mT5 and mBART, we trained using AdamW optimizer for 10 epochs on a machine with 4 NVIDIA V100 GPUs. We used a batch size of 8, a learning rate of 2e-5, and a beam size of 3.

5.3.2 Metrics Used

To measure our model's performance on the WikiOutlines dataset, we benchmark our performance using syntactic and semantic text generation metrics.

- Rouge-L: It is an n-gram overlap-based metric to determine the correctness of generated text as compared to gold text. In Rouge-L, the value of n for n-gram is the longest co-occurring n-grams (Longest Common Subsequence) between prediction and reference.
- METEOR: It is a more complicated metric, often used to represent more information than simple token matching. It improves word matching between prediction and reference using synonyms, stemming, word-order swapping and paraphrasing.
- BLEU: The score is computed by comparing n-grams of the machine-translated text to n-grams of the reference text, calculating precision for each n-gram, and then applying a brevity penalty to discourage overly short translations.

	XLM-Score	BLEU	METEOR	ROUGE-L
WFSA (section-level)	70.0	45.0	37.1	56.8
WFSA (word-level)	69.1	43.4	36.1	55.9
mBART	70.2	39.1	31.9	52.2
mT5	**76.2**	**48.5**	**40.3**	**59.4**

Table 5.1: Comparison of WFSA and Transformer-based methods for multi-lingual outline generation. Best scores are bolded.

- XLM-Score: It is a variation of traditional BERT-Score, using XLM to calculate the contextual embeddings instead of BERT to cater to cross-lingual settings. It compares the similarity of individual tokens in machine-generated and reference texts, providing a measure of semantic equivalence rather than relying on exact word matches.

5.3.3 Main Results

Table 5.1 shows the comparison of WFSA and Transformer-based methods for multi-lingual outline generation using popular natural language generation metrics like XLM-Score, BLEU, METEOR and ROUGE-L. We observe that (1) mT5 outperforms other methods by large margins across all metrics. (2) WFSA at section-title level leads to better results compared to WFSA at word level.

Further, we show detailed results for our best model (mT5) at a (language, domain) level using the four metrics (XLM-Score, BLEU, METEOR, and ROUGE-L) in Tables 5.2-5.5. From these tables, we observe that (1) The model performs best for films and sportsman domains, and worst for writers and animals domains. This is justified because of the large number of training samples in films and sportsman domains and low number of samples in animals domain. However, it is surprising that the model does not perform well on writers domain inspite of the large number of training samples. (2) The model performs best for Punjabi, and worst for Telugu and Kannada. The worse performance for Telugu and Kannada can perhaps be because of low number of training samples for those languages.

5.3.4 Qualitative Analysis

Lastly, we show some examples of generated outlines using our best method in Table **??**. These examples show that our method can generate reasonably usable outlines.

	en	mr	hi	kn	ta	bn	pa	te	ml	or	Avg
companies	80.4	80.7	64.8	69.1	82.2	73.4	92.2	72.6	69.8	80.4	**76.6**
writers	73.4	70.7	71.3	63.3	75.2	66.1	86.9	68.4	72.2	84.0	**73.2**
cities	77.0	72.3	62.9	66.0	78.6	66.9	92.0	63.5	76.1	92.0	**74.7**
politicians	71.6	81.8	79.2	65.4	81.0	75.2	87.5	69.2	70.5	88.4	**77.0**
books	71.6	73.3	87.9	70.1	78.9	69.4	89.9	63.4	72.8	83.3	**76.1**
films	80.7	76.1	72.7	81.1	80.3	71.3	91.1	76.5	72.5	91.8	**79.4**
animals	71.9	62.0	67.0	69.0	80.9	68.9	88.9	68.2	78.0	81.4	**73.6**
sportsman	81.1	88.2	83.0	66.7	86.1	74.2	90.8	64.2	75.0	83.5	**79.3**
Avg	**75.9**	**75.6**	**73.6**	**68.9**	**80.4**	**70.7**	**89.9**	**68.2**	**73.4**	**85.6**	**76.2**

Table 5.2: XLM-Score for mT5 across various (language, domain) pairs.

	en	mr	hi	kn	ta	bn	pa	te	ml	or	Avg
companies	52.8	66.7	43.8	44.1	57.9	57.0	67.2	45.6	47.5	37.6	**52.0**
writers	31.5	46.2	54.3	29.6	37.1	40.6	45.5	33.0	50.2	41.1	**40.9**
cities	39.4	48.0	28.9	33.3	48.0	35.0	67.7	26.8	60.5	62.4	**45.0**
politicians	35.3	68.3	67.5	31.9	53.0	58.2	50.1	37.6	46.6	51.9	**50.0**
books	38.3	54.2	81.5	46.4	48.3	44.4	58.0	38.2	53.9	49.2	**51.2**
films	52.6	57.8	40.2	64.9	52.0	49.2	62.0	56.8	52.6	55.5	**54.4**
animals	32.0	37.4	37.3	35.6	54.9	38.6	50.9	44.7	59.3	31.9	**42.3**
sportsman	41.9	79.2	72.0	31.7	62.4	45.5	62.9	32.8	53.2	44.4	**52.6**
Avg	**40.5**	**57.2**	**53.2**	**39.7**	**51.7**	**46.0**	**58.1**	**39.4**	**53.0**	**46.7**	**48.5**

Table 5.3: BLEU for mT5 across various (language, domain) pairs.

5.4 Conclusion

In this chapter, we motivate and proposed the problem of OutlineGen, which is the task of multilingual outline generation using minimal information (Article Title, Language and Domain). We want to use minimal information so that it requires minimal intervention from humans, and requires no information from authors, editors or automated systems to generate an outline. Due to the similarity in outlines we observe, we propose the statistical method of Weighted Finite State Automata (WFSA). A drawback of

	en	mr	hi	kn	ta	bn	pa	te	ml	or	Avg
companies	64.9	80.5	54.3	58.1	64.4	62.6	69.3	57.0	54.6	45.2	**61.1**
writers	44.1	62.5	64.0	44.8	46.8	51.0	52.6	44.9	56.8	50.0	**51.7**
cities	59.9	68.5	43.7	48.6	55.9	46.7	73.2	44.9	66.7	77.3	**58.5**
politicians	44.3	80.9	74.3	52.1	59.8	66.8	57.6	52.0	54.4	64.5	**60.7**
books	51.1	72.8	84.9	64.3	55.7	51.2	64.4	46.8	58.5	56.1	**60.6**
films	67.1	66.7	59.0	75.8	57.7	59.1	65.4	63.9	59.6	70.7	**64.5**
animals	49.8	55.8	47.5	46.7	59.8	50.0	58.1	51.6	65.2	44.0	**52.9**
sportsman	73.5	86.7	77.5	54.4	71.9	59.2	68.9	45.7	60.1	53.5	**65.1**
Avg	**56.8**	**71.8**	**63.2**	**55.6**	**59.0**	**55.8**	**63.7**	**50.9**	**59.5**	**57.7**	**59.4**

Table 5.4: ROUGE-L for mT5 across various (language, domain) pairs.

	en	mr	hi	kn	ta	bn	pa	te	ml	or	Avg
companies	49.3	38.6	35.0	39.9	47.9	45.2	52.8	39.5	32.9	24.9	**40.6**
writers	28.5	33.5	45.1	26.0	29.5	31.7	32.5	30.6	37.6	29.9	**32.5**
cities	46.4	30.6	34.5	23.4	37.2	31.1	58.1	25.7	49.8	70.9	**40.8**
politicians	26.6	39.3	59.0	26.7	42.5	52.7	39.6	36.7	34.3	53.4	**41.1**
books	40.1	31.7	74.1	28.2	37.0	31.4	46.7	24.1	39.8	37.9	**39.1**
films	52.3	53.0	54.1	42.8	36.5	36.2	47.2	45.9	37.2	68.4	**47.4**
animals	43.9	27.1	39.5	25.7	41.7	29.6	41.3	32.0	48.1	22.6	**35.1**
sportsman	63.8	46.6	62.9	31.5	58.6	42.8	54.0	25.7	39.6	34.3	**46.0**
Avg	**43.9**	**37.6**	**50.5**	**30.5**	**41.4**	**37.6**	**46.5**	**32.5**	**39.9**	**42.8**	**40.3**

Table 5.5: METEOR for mT5 across various (language, domain) pairs.

wfsa is that it generates a single outline for a (language, domain) pair instead of it being specific to each article title. To tackle this, we use finetuned mT5 and mBART as well.

We also introduce the dataset of WikiOutlines, which covers eight domains and ten languages spanning approximately 166k articles. We benchmarked our models against WikiOutlines using XLM-Score, BLEU, METEOR and Rouge-L as the metrics. Our best performing model is finetuned mT5, giving results of 76.2 XLM Score, 48.5 BLEU Score, 40.3 METEOR Score and 59.4 ROUGE-L respectively. We also provide quantitative and qualitative analysis of our results in this chapter. The high scores

Info\Entities	Takin	Dresden	इन्साफ का मंदिर (1969 फ़िल्म)	ଓଟକା ଭାଉ ପାକା ଓରେ ପାକା	তমাল নদী
Language	en	en	hi	or	bn
Domain	animals	cities	films	films	sportsman
Ground-truth outline	Introduction, Appearance, Distribution and habitat, Behaviour and ecology	Introduction, History, Geography, Governance, Culture, Economy, Education and science, Notable people	परिचय, मुख्य कलाकार, परिणाम, नामांकन और पुरस्कार	ପରିଚୟ, ଅଭିନୟ, ପାଟ ଏବଂ ପାକାର	ভূমিকা, প্রাথমিক, প্রশিক্ষণ বিবরণ,
Generated outline	Introduction, Taxonomy, Distribution and habitat, Behaviour and ecology	Introduction, History, Geography, Demographics, Culture, Notable People	परिचय, मुख्य कलाकार, परिणाम, नामांकन और पुरस्कार	ପରିଚୟ, ଅଭିନୟ, ପାଟ ଏବଂ ପାକାର	ভূমিকা, প্রাথমিক-মৌলিক তথ্য, প্রশিক্ষণ বিবরণ

Table 5.6: Examples of generated outlines using our best method

observed as well as the qualitative analysis of our results show that the proposed system is practically usable to generate candidate outlines.

Chapter 6

Cross-lingual Fact Extraction and Verification

6.1 Introduction

As discussed in previous chapters, the importance of Wikipedia in low-resource communities is very high, especially since it provides reliable information covering multiple domains and languages. Ensuring the reliability of Wikipedia is important, and it has become tougher for editors to verify each sentence in multiple low-resource languages since reference articles can be of different languages and the number of references can go up to hundreds. Hence, automatic fact verification for Wikipedia articles is required.

Existing work on fact verification has worked on sentence level, where multiple sentences are given as context, and one must predict if a given sentence follows the context. Such work is mostly monolingual and does not have the noise that multiple Wikipedia references and their articles will have. Another reason these solutions do not work directly in the context of Wikipedia is that they verify predicates on sentence level granularity, but as previously mentioned in Chapter 1 in Table 1.3, average number of facts is 2 for sentences in Wikipedia. Hence, we must have a fact-level granularity fact verification to better judge the correctness of the articles proposed.

To help with automatic Fact Verification, we proposed the task of FactVer, which is a cross-lingual fact verification task meant to verify the Wikipedia article against its references at a factoid-level granularity. The factoid is defined as a triple of (subject, relation, object). The task is cross-lingual since the reference can be in languages different than that of the source article. Figure 6.1 demonstrates how factver works. Article title, source language and reference texts act as our input to the pipeline, and the output is a set of factoids from the article with supporting labels from factoids from references mentioning whether the article factoid is supported, or is in need of citation.

The pipeline for cross-lingually extracting factual information can also be used for multiple purposes, like automatically populating knowledge graphs such as Wikidata or utilising natural language text from multiple sources to create a common knowledge graph. Once the facts are extracted, we pass each of the facts along with semantically selected sentences from the reference through a classifier pipeline, which predicts if the citations support the fact or if the fact is in need of further citation. Such a pipeline can be used for automatically citing text on the low-resource editions of Wikipedia and reducing the

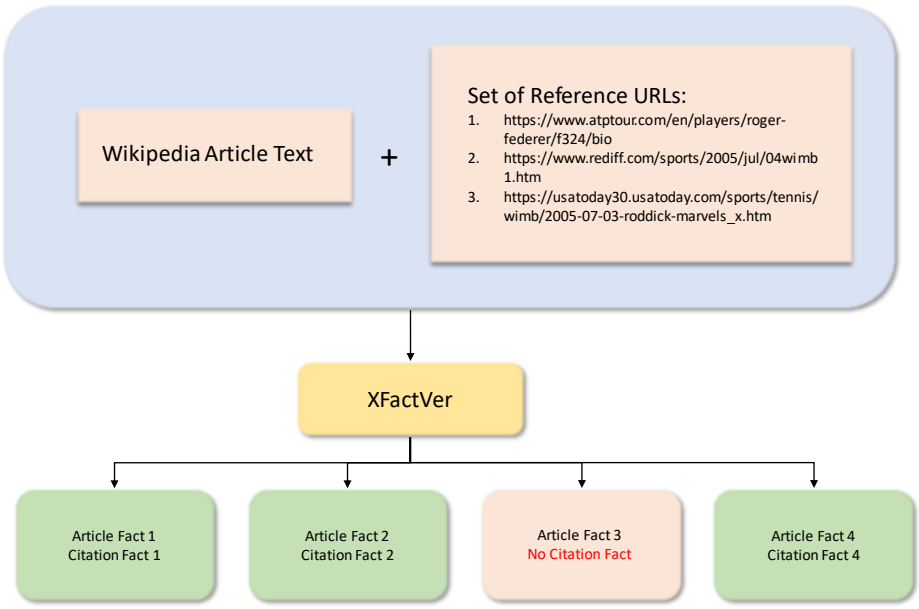

Figure 6.1: Description of Cross-lingual Fact Verification process.

manual efforts needed to identify sentences needing citations. This becomes particularly important for the low-resource versions of Wikipedia, which have a lower quality of articles and fewer editors.

Thus, the major contributions of this paper include:

- A cross-lingual dataset for fact extraction and verification, covering English and five Indian languages.
- A pipeline for automated cross-lingual fact extraction and verification, focussing on fact-level granularity instead of sentence-level.

6.2 Dataset Preparation and Analysis

Currently, there are popular datasets for Fact Verification like the FEVER benchmark, but they work on sentence-level granularity. In our case, we want factoid-level granularity, and for that, we need to create our own dataset. Since the first step to creating our dataset involves knowing the factoids from Wikipedia articles, we look at existing work in that domain. XAlign [1] is a work on Wikipedia in low-resource languages on people's domain with factoids extracted from the Wikipedia article. Table 1.3 describes details of the dataset, which covers over 12 low-resource languages spanning $\sim$155k articles.

Language	Articles	Sentences	Facts
bn	11,468	53,522	106,165
or	1,635	7,601	13,035
en	4,715	17,326	39,540
pa	3,491	12,324	25,758
ta	6,003	21,937	38,100
hi	5,796	20,277	40,062
Total	**33,108**	**132,987**	**262,660**

Table 6.1: Dataset statistics for each of the languages.

XAlign manually aligns its test dataset, and hence we can use it in our test dataset as gold-extracted factoids from Wikipedia articles.

Another need for our proposed problem is getting references from Wikipedia articles. We utilize XWikiRef for this, which we have described in detail in Chapter 3. To summarize, it is a cross-lingual dataset across eight languages and five domains with reference texts for an article extracted. The domains it covers is writers, politicians, sportsmen, films and books.

Since we need both, the extracted factoid from Wikipedia article, and the reference texts of articles, we need to find an intersection between these two datasets. To do that, first, we need to find intersecting domains, which leads us to choose a subset of XWikiRef containing the domain of writers, sportsmen and politicians, the domains that come under the people domain. We also need to find an intersection between languages, so we choose six languages: Bengali (bn), Odia (or), English (en), Hindi (hi), Punjabi (pa) and Tamil (ta). Once we have narrowed down language and domains of interest, we find the intersection between the two dataset. It is worth noting that XWikiRef has Wikipedia Article Title in each sample, whereas XAlign only has Wikipedia page QID. Hence, to find set intersection between the two, we use Wikimapper [1], which converts article title to qid and vice versa.

Dataset statistics related to the combined dataset are described in Table 6.1. We get a total of $\sim$33k articles across six languages in peoples domain (more specifically, in writers, sportsmen and politicians domain). To create our test set, we have two steps that we have to take: gold fact extraction and gold fact verification. For the first step, XAlign already has a gold test dataset which is manually annotated which we can use as gold fact extraction dataset. For fact verification, we need to manually go through all references for an article and assign a label for each factoid mentioned in XAlign dataset. It is infeasible for humans to manually go through multiple references in various languages to accurately assign labels, hence we semi-automate this task. For each article factoid, we find ten most similar sentences across all reference sentences. We do this using semantic similarity between the factoid and reference sentence,

[1] `https://github.com/jcklie/wikimapper`

using LABSE [57] to give us contextual representation. Once we have the top-10 most similar sentence for a given factoid, we translate all of them to English for ease of understanding by manual annotators. We then ask the annotaters to either label supported or not supported based on if the article factoid was supported by any of the ten recommended reference sentences. In this way, we get our manually annotated dataset. Description of this dataset can be seen in Figure 6.2.

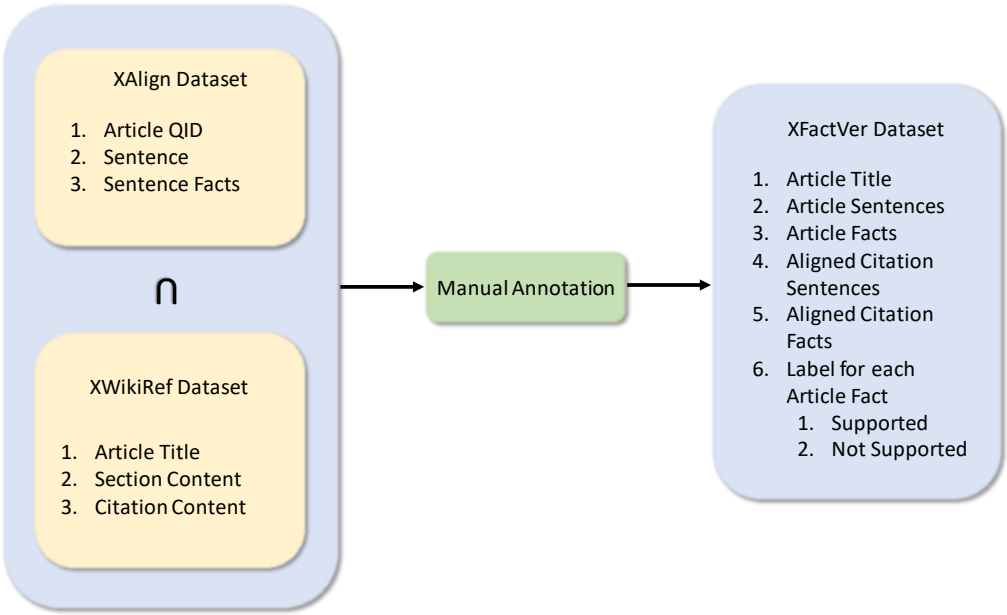

Figure 6.2: Components of the XFactVer dataset.

6.3 Methodology and Pipeline

Direct fact verification of a Wikipedia article is improbable due to the context size of multiple references. We encounter this problem with XWikiGen in Chapter 4, and we use a two-stage process to reduce the context length in it. Directly using such large context as input, especially in case of fact verification where the reference sentence may not be related to the entity itself, leads to poor extraction of facts. Hence, we divide our pipeline into two stages: Fact Extraction and Fact Alignment. Figure 6.3 shows how we plan to approach this problem.

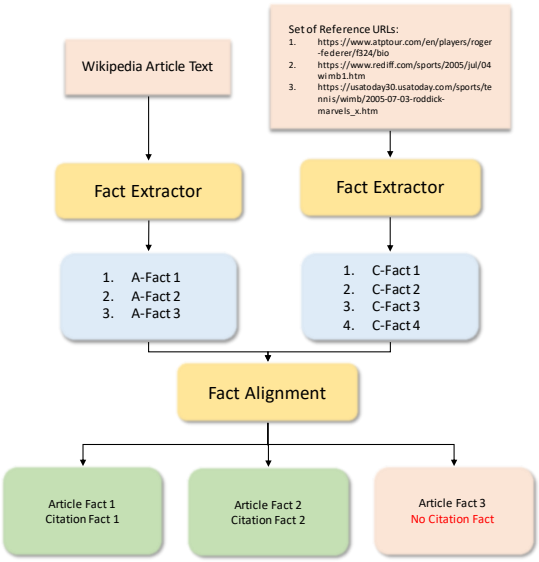

Figure 6.3: Pipeline for automated fact extraction and verification.

6.3.1 Fact Extraction using Multilingual Transformers and LLMs

The first step of our pipeline requires us to extract facts from both, sentences from Wikipedia article and sentences from references. For extraction, firstly, we train our cross-lingual mT5 [13] model on XAlign [1] train dataset. Once we do that, we now want to use it as an inference to get factoids from sentences in article and references. Before that, we must note that blindly inputting sentences into the model will often lead to incorrect factoid formation and extraction. Hence, to improve performance and reduce complexity, we first perform a data preparation set.

In XAlign, the authors used POS tagging along with other heuristics to determine sentences that most probably do not have a factoid. Doing this in our case would allow us to reduce complexity since the number of sentences on which we run inferences would be reduced. Hence, we keep a heuristics of wanting a sentence with 5 words and having at least a Verb and a Noun. We do the POS-tagging using Stanza [58] and other low-resource POS taggers to filter possible sentences.

After our initial step of data preparation, we then pass them through our Fact Extractor. For this, we experiment with two options: (1) supervised mT5 trained on XAlign dataset, (2) few-shot GPT-4. In the first experiment, we train mT5-small, which is an encoder-decoder transformer with eight encoder layers and eight decoder layers. We train it for ten epochs, using AdamW optimizer and a learning rate of 2e-5. We train it on a multi-GPU setting on 4 GPUs with a batch size of 4. In our second experiment, we try out GPT-4 to test out feasibility of LLMs for cross-lingual tasks and to see how well they understand and extract factoids.

Specifically, for GPT-4, we use few-shot prompting to get results. In few-shot prompting, a few examples are provided within the prompt of input and expected output so that the model learns by example what to do. In our case, we give examples from our val dataset while running inference on our test dataset. The initial prompt used was:

You must extract all facts in English from the following {LANG} sentence. A fact consists of a relation and tail entity present in the sentence. Return the extracted facts in the form of a list of lists.

6.3.2 Fact Verification via Alignment

As described in our proposed pipeline, the second stage of our verification process is Fact Alignment. In this case, we have already extracted two sets of facts, one from the Wikipedia article, and another from its reference texts in our previous step. Now that we have two sets of facts, we want to align the appropriate reference fact to article fact to generate labels of supported or unsupported appropriately.

Initially, we wanted to label our dataset according to three labels, supported, unsupported and contradictions. But during manual annotations, we observed that the number of contradictions observed is very few, and mostly they occur due to failure in our heuristic of selecting top-10 most similar sentence. Hence, we ignore that metric, and instead choose to go with only supported and unsupported. Due to our initial attempt to include contradiction, we also tried a heuristic for labeling by matching the relation and object label of factoids. However, we observe this performs poorly, and we do not proceed with it.

Finally, we go with a similarity metric using LABSE [57] to get semantic representation. In this case, for each factoid in the article set, we find similarity of it against all factoids in the reference set. We set a threshold of 0.7 as our similarity threshold, and if there is a similarity 0.7 between factoids of article and references, then we set the label as supported, otherwise, we set the label as unsupported. Even this method is a simplistic approach to the fact alignment problem, and we hope that in proposing the task, dataset and benchmarks, there is more work on this problem leading to improved approaches towards it.

6.4 Results

We display metrics across both the steps of our method. For Fact Extraction stage, we choose to take ROUGE-L in line with existing work CLFE [59], and also include BERTScore. ROUGE-L is a commonly used metric for text generation tasks, and it measures the overlap of n-gram (in this case, largest common subsequence) to give us the score. We include BERTScore as well since it is a semantic representation, and we observed that a few times, facts extracted had relations which meant the same thing but strict overlap did not consider them equal. Hence, we add a semantic measure to have a more comprehensive understanding of our model's performance on our dataset.

We show the results for fact extraction in Table 6.2. We observe that in almost all cases, mT5-small performs better, except in case of Bengali (bn). We also observe that in most cases, the scores are quite high, which implies that we can be confident about the performance of our models in first stage of

	mT5-small		GPT-4	
Metric	**ROUGE-L**	**BERTScore**	**ROUGE-L**	**BERTScore**
bn	0.838	0.890	**0.902**	**0.954**
or	**0.711**	**0.860**	0.600	0.822
en	**0.768**	**0.883**	0.656	0.868
pa	**0.692**	**0.865**	0.601	0.847
ta	**0.842**	**0.924**	0.766	0.902
hi	**0.854**	**0.932**	0.596	0.833
avg	**0.784**	**0.893**	0.687	0.871

Table 6.2: Results of Fact Extraction stage across different metrics and methods.

	bn	**or**	**en**	**pa**	**ta**	**hi**	**avg**
Accuracy	66.59	70.52	61.90	60.39	66.43	57.76	**63.93**

Table 6.3: Results of Fact Verification by Alignment stage across all languages.

our pipeline. We also see that GPT-4 performs well in cross-lingual context, but is still not as good as fine-tuned cross-lingual model like mT5. Surprisingly, mT5 performs better even for high-resource language like English (en).

For our Fact Verification via Alignment step, we use accuracy as our metric. We decide on a strict and simplistic metric since it better fits our use-case of fact verification. We can see the details of our model's performance in Table 6.3. We observe that for most languages, we get an almost equal accuracy, which implies that in our pipeline, there is not much of a language divide. One reason for the lower accuracy scores as compared to high fact extraction scores could be the similarity matching being too simplistic of a method. Another possibility is error propagation due to a two-stage method.

6.5 Conclusion

In this chapter, we motivate and propose the problem of Cross-lingual Fact Extraction and Verification, which is the task of extracting factoids from Wikipedia article and references, and verifying if wikipedia facts are supported by the reference facts. A notable difference from existing work in this field is that we work on fact-level granularity to account of sentences having multiple facts. Due to the large number of references, we propose a two-stage process as the pipeline. The first stage involves Fact Extraction in

which we extract factoids (subject, relation, object) from the article and reference in English. We compare different extraction methods involving few-shot GPT4 and finetuned mT5, and observe finetuned mT5 performs better with 0.784 Rouge-L score and 0.893 BERTScore. For fact verification we use a simple similarity based alignment, and observe an average of approximately 64% accuracy.

We also introduce the FactVer dataset, which is a dataset across six languages in the people domain of Wikipedia, spanning 33k articles. The dataset is formed as an intersection of XWikiRef and XAlign [1], after which we manually annotate the test split for fact verification. We make this dataset and our code publically available to enable further research in low resource fact extraction and verification.

Chapter 7

Conclusion & Future Work

In this thesis, we explore three main problems when it comes to text generation for Wikipedia articles. Firstly, we explore generation of Wikipedia article using its outline and references as context. We also try to automate the process of generating article outlines, generating the outline using minimal information. Lastly, we explore the difficult problem of fact extraction and verification for Wikipedia articles, using the references as grounding and determining if facts mentioned in the articles are supported by references or not.

In **Chapter 1** we introduce and motivate the problems we work on. Here, we highlight the disparity in content between content and available tools in low-resource languages vs English on the internet. For a lot of communities, it is not easy to participate in the global digital world due to the disparity in content and tools available, and hence it is important for us to work on problems in low-resource languages. Possible solutions involve developing datasets and tools which are multilingual or cross-lingual in nature, taking advantage of information available in multiple languages to enhance performance in each. We also talk about importance of Wikipedia when it comes to providing reliable information, and how to improve accessibility to it. The chapter also discusses the overall structure of the thesis and the main contributions made through it.

After defining our problem, we explore the existing literatue on related topics in **Chapter 2**. We go over multiple papers on Wikipedia text generation, both in the long and short format. Although most of the work for text generation has been done in English, we observe that recently, there has been more focus on multi-lingual and cross-lingual generation. Short-text generation for Wikipedia often utilizes information available as a factoid or graph or info-box as its base to generate one to two sentences about the main article. Long-text generation focusses on generating either a section or the whole article using references as its base. We then explore the previous works done for Outline Generation, and observe limited number of papers in it. Majority of the work done for automated outline-generation has been in English, and often uses the whole article content as its context. Hence, it reinforces our motivation to work on the problem of multilingual outline generation with minimal information. Lastly, study past literatue for Fact Extraction and Verification. Here, we see that a lot of work has been done on these problems, even in multi-lingual context. Most of the existing work focus on sentence-level fact

verification, where they verify a sentence's validity based on multiple reference sentence provided. We aim to work on fact-level granularity instead of sentence-level granularity since one sentence ends up having multiple facts in it. Overall, we learn about the work, the gaps and the popular methodologies used in all the problems we will be tackling in this chapter.

We start with describing the datasets we propose and create to accompany our tasks better in **Chapter 3**. We talk about two datasets in this chapter, XWikiRef and WikiOutlines. XWikiRef is a cross-lingual multi-document summarization dataset across five domains (writers, politicians, sportsmen, books, films) and eight languages (English, Hindi, Bengali, Oriya, Punjabi, Marathi, Malayalam, Tamil). It has $\sim$69k samples where each sample has its article title, article outline, reference texts, target language, domain and article content. We use this dataset for the task of automatic Wikipedia article generation. Next, we propose the WikiOutlines dataset, which is a multilingual outline generation dataset across eight domains (writers, politicians, sportsmen, books, films, cities, animals, companies) and ten languages (English, Hindi, Bengali, Oriya, Punjabi, Marathi, Malayalam, Tamil, Telegu and Kannada). It has $\sim$166k samples, where each sample has its article title, language, domain and article outline. We use this dataset for the task of automatic Wikipedia outline generation. Besides describing the two datasets, we also provide a preliminary analysis of them in the chapter.

In **Chapter 4**, we discuss the problem of automatic generation of Wikpedia articles using references as context. We define XWikiGen: a cross-lingual multi-document summarization task for generating Wikipedia articles section-by-section using article outline and references as context. We observe that due to large number of references, and each reference having multiple sentences, it is infeasible to directly fine-tune a model for our summarization task. Hence, we propose a two-stage pipeline of unsupervised extractive summarization (to reduce total number of sentences to only the most salient ones) followed by supervised abstractive summarization. We experiment with two models for unsupervised summarization stage: using QA-GNN for saliency based and HipoRank for importance based summarization. For the abstractive stage, we experiment with mT5 and mBART, both being state-of-the-art multilingual transformer models of about the same size. We benchmark our results with Rouge-L, CHRF++ and METEOR as our metrics, and provide both qualitative and quantitative analysis. We observe that HipoRank + mBART pipeline performs the best across all languages and domains.

We discuss our work on multilingual outline generation in **Chapter 5**. We define OutlineGen as an outline-generation task over multiple languages using minimal information (Article Title, Language, Domain). The reason for having minimal information as input is so that it requires almost no intervention required from author's or editor's end for them to get a structural outline to get started with. We propose two main methodologies here, Weighted Finite State Automata (WFSA) and Finetuned Multilingual Transformer. The reason for using WFSA is due to repition in patterns observed in outlines of articles belonging to a (language, domain) pair. Taking advantage of this, we build our WFSA, using transition probabilities as the weight of edges, and traverse the graph. A main drawback of this method is that it generates a single outline per (language, domain) pair, and hence we add article-title as additional input and finetune mT5 and mBART over it. We benchmark our approach using Rouge-L, METEOR, BLEU

and XLM-Score to measure performance of our models syntactically and semantically. We also provide qualitative and quantitative analysis and results for our models. Overall, we observe that finetuned mT5 performs the best across all metrics, although results from WFSA are quite close.

Lastly, we discuss our work on Cross-lingual fact extraction and verification in **Chapter 6**. Formally, FactVer is a cross-lingual task where the goal is to extract factoids (subject, relation, object) from Wikipedia articles, and verify them against factoids from the article's references. We do this at a fact-level granularity to account for sentences having multiple facts. We create a new dataset for this task combining XWikiRef and XAlign [1], then manually annotate the supported or not-supported labels for all samples in the test split. We cover six languages (Hindi, English, Bengali, Odia, Punjabi and Tamil) in three domains (Writers, Politicians and Sportsmen). Since the size of the article and reference texts are huge, we use a two-stage approach for this task. We first extract the facts from the sentences of both article and references, after which we align the facts to match them. We experiment with few-shot GPT4 and finetuned mT5 for the Fact Extraction part, where we observe better results for finetuned mT5 in most cases. For Fact-Verification we perform a simple semantic match for alignment, and calculate the accuracy against the manually annotated dataset.

Future Work

We also highlight possible extensions of this work that can be done in the future:

- In case of XWikiRef and WikiOutlines, we can increase the number of languages and domains covered to also include other high-resource and non-indic low-resource languages to better measure the effectiveness of our methods.
- Better utilize newer methodologies like Retrieval Augmented Generation (RAG), which has a direct use case for XWikiGen and FactVer. As well as utilize LLMs to improve generation quality for the abstractive stage of XWikiGen and OutlineGen.
- Additional context can be provided for OutlineGen, including relevant documents and references to better judge and generate the outline required per article title. We can add these data sources from existing pages of different languages, or can use search query results for the same.
- Utilize methods like Reinforcement Learning to devise custom rewards that work better for low-resource languages for all three tasks to better nudge the existing models in the required direction.
- Build knowledge graphs from factoids extracted from Wikipedia Articles to represent knowledge across languages in a singular graph and use it for downstream tasks like generation and question answering.

Overall, we have made important contributions towards improving the quality and quantity of low-resource content on Wikipedia and the Internet. More work can be done to improve participation and representation of low-resource communities which we outline above. Our existing work, and any future work in this domain will help in the ultimate goal of improving access to reliable information to the maximum number of people.

Appendix A

More Experiments with FSA and RL for Outline Generation.

In this chapter, we delve into additional experiments conducted for Outline Generation. Our exploration includes testing various types of FSA, enhancing data informativeness, and exploring Reinforcement Learning, along with other contextual augmentation methods for outline generation.

We will be discussing the following experiments:

1. Different types of Weighted Finite State Automata (WFSA)
2. Reinforcement Learning with FSA
3. Using reference text as context

A.1 Different types of WFSA

In Chapter 5, we explore both word-level and sentence-level FSA experiments. Here, additionally, we investigate various approaches to sampling the next node and determining beam size. We also vary across the min-size parameter which determines the minimum size of outline we require. Our FSA construction remains conventional, but during our breadth-first search, we experiment with sampling the next k-nodes through a combination of probabilistic methods.

Weighted Probablity Sampling: In weighted random probability sampling, each item in the population is assigned a weight (or probability) that reflects its likelihood of being selected. Here, the probability of being selected is defined as the weight of the node divided by sum of weights of all nodes.

Cumulative Probablity Sampling: Cumulative random probability sampling, on the other hand, involves creating a cumulative distribution of the weights and then selecting an item based on where a randomly generated number falls within this distribution. This method also ensures that items with higher weights are more likely to be selected, but it does so by comparing the random number against the cumulative distribution rather than directly against the weights.

For all sampling scenarios, we compare results across validation dataset using only Rouge-L as metric. We average over ten runs for all probabilistic methods.

Rouge-L Score	Min Size = 3	Min Size = 2	Min Size = 1
K = 1	0.285	0.448	0.48
K = 2	0.34	0.489	0.526
K = 3	0.339	0.514	0.55
K = 4	0.333	0.529	0.563

Rouge-L Score	Min Size = 3	Min Size = 2	Min Size = 1
K = 1	0.292	0.455	0.469
K = 2	0.352	0.489	0.522
K = 3	0.342	0.514	0.553
K = 4	0.33	0.528	0.572

Table A.1: Rouge-L Score on Val dataset for Cumulative and Weighted Sentence level Sampling respectively. Here k = number of nodes sampled at each level.

Rouge-L Score	Min Size = 3	Min Size = 2	Min Size = 1
K = 1	0.359	0.463	0.482
K = 2	0.357	0.469	0.51
K = 3	0.35	0.501	0.544
K = 4	0.371	0.524	0.551

Rouge-L Score	Min Size = 3	Min Size = 2	Min Size = 1
K = 1	0.37	0.457	0.473
K = 2	0.343	0.469	0.507
K = 3	0.353	0.499	0.54
K = 4	0.374	0.519	0.558

Table A.2: Rouge-L Score on Val dataset for Cumulative and Weighted Word level Sampling respectively. Here k = number of nodes sampled at each level.

In Table A.1 and Table A.2 we show the result of our proposed probabilistic FSA methods on the Validation dataset of WikiOutlines. We observe that in most cases, Sentence-level FSA performs better than Word-level FSA, and weighted and cumulative sampling both have almost the same results. We also note that lower the min size restriction on the outline, the better it performs, while the opposite is true for K (number of nodes sampled). It is to note that although we show Rouge-L here, the decision to select the optimal model was made by observing val results across multiple metrics. We show Rouge-L only since other experiments show cased in this chapter were only measured for Rouge-L, and hence we want this to be an apt comparison.

Additionally, we also experiment with Query Blazer[56]. It is an extremly fast FSA based method which uses ngram modelling for text completion of queries. We use this for our outline generation task with the input as (Language, Domain). We experiment going language-wise versus language, domain-wise for QueryBlazer, and observe that having specific FSA for each (Lang, Dom) pair is better. This is in line with results seen for our FSA as well. The scores for QueryBlazer are mentioned in Table A.3.

A.2 Reinforcement Learning with FSA

While experimenting with FSA, we observe that FSA gives us a good model of probablity of each outline. We use this model by combining FSA with our fine-tuned generative model by using Reinforcement Learning. We end up using FSA as a reward, where after each generative step we calculate the probability of that outline, which we end up using as a reward. Figure A.1 shows how our proposed model will work.

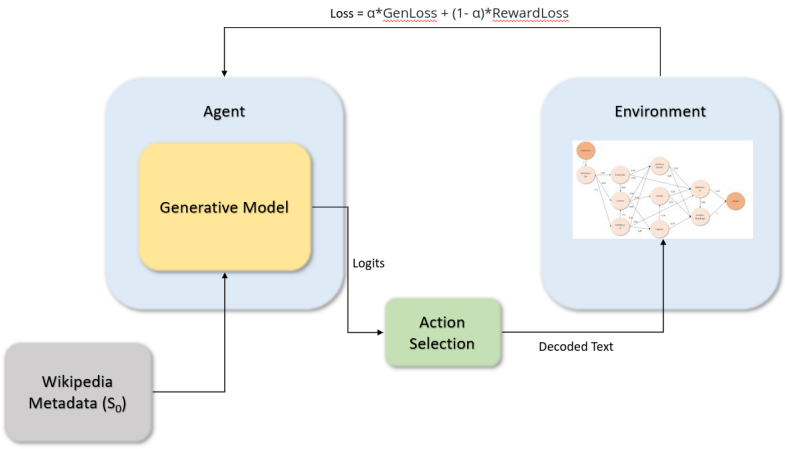

Figure A.1: Reinforcement Learning with FSA as reward.

	hi	mr	bn	or	ta	en	ml	pa	kn	te	AVG
politicians	0.744	0.833	0.620	0.626	0.629	0.423	0.536	0.578	0.539	0.511	**0.604**
cities	0.555	0.752	0.505	0.460	0.517	0.570	0.653	0.823	0.526	0.593	**0.595**
books	0.884	0.709	0.567	0.606	0.521	0.473	0.684	0.661	0.607	0.631	**0.634**
writers	0.631	0.657	0.517	0.494	0.462	0.401	0.528	0.539	0.512	0.453	**0.520**
companies	0.618	0.788	0.577	0.488	0.583	0.652	0.628	0.714	0.540	0.512	**0.610**
sportsman	0.710	0.852	0.559	0.554	0.659	0.543	0.588	0.655	0.535	0.442	**0.610**
films	0.526	0.670	0.578	0.688	0.601	0.651	0.559	0.700	0.755	0.643	**0.637**
animals	0.528	0.643	0.535	0.510	0.609	0.374	0.648	0.621	0.543	0.470	**0.548**
AVG	**0.650**	**0.738**	**0.557**	**0.553**	**0.573**	**0.511**	**0.603**	**0.661**	**0.570**	**0.532**	**0.595**

	bn	en	hi	kn	ml	mr	or	pa	ta	te	AVG
politicians	0.506	0.346	0.597	0.539	0.434	0.576	0.541	0.467	0.508	0.442	**0.496**
cities	0.510	0.440	0.555	0.526	0.608	0.752	0.421	0.678	0.587	0.528	**0.560**
books	0.461	0.405	0.707	0.607	0.546	0.512	0.491	0.528	0.420	0.472	**0.515**
writers	0.425	0.334	0.507	0.512	0.429	0.474	0.408	0.436	0.378	0.390	**0.429**
companies	0.469	0.535	0.498	0.540	0.566	0.557	0.387	0.611	0.471	0.424	**0.506**
sportsman	0.456	0.466	0.567	0.535	0.472	0.401	0.501	0.528	0.556	0.361	**0.484**
films	0.472	0.541	0.485	0.755	0.454	0.496	0.620	0.530	0.482	0.522	**0.536**
animals	0.442	0.316	0.466	0.543	0.523	0.478	0.412	0.497	0.493	0.381	**0.455**
AVG	**0.442**	**0.316**	**0.466**	**0.543**	**0.523**	**0.478**	**0.412**	**0.497**	**0.493**	**0.381**	**0.455**

Table A.3: Rouge-L score on Val dataset for QueryBlazer across (Lang, Dom) and Lang respectively.

	hi_IN	mr_IN	bn_IN	or_IN	ta_IN	en_XX	ml_IN	pa_IN	kn_IN	te_IN	AVG
politicians	0.747	0.805	0.621	0.390	0.485	0.374	0.536	0.589	0.542	0.304	**0.539**
cities	0.461	0.555	0.488	0.454	0.393	0.569	0.653	0.675	0.524	0.410	**0.518**
books	0.885	0.657	0.543	0.606	0.461	0.438	0.684	0.646	0.607	0.377	**0.590**
writers	0.642	0.628	0.512	0.421	0.414	0.387	0.528	0.571	0.512	0.315	**0.493**
companies	0.621	0.645	0.578	0.488	0.424	0.642	0.552	0.679	0.540	0.374	**0.554**
sportsman	0.723	0.823	0.555	0.610	0.656	0.437	0.587	0.616	0.523	0.341	**0.587**
films	0.456	0.662	0.524	0.365	0.507	0.420	0.557	0.631	0.755	0.533	**0.541**
animals	0.533	0.387	0.389	0.432	0.606	0.415	0.647	0.632	0.543	0.410	**0.500**
AVG	**0.634**	**0.645**	**0.526**	**0.471**	**0.493**	**0.460**	**0.593**	**0.630**	**0.568**	**0.383**	0.540

	hi_IN	mr_IN	bn_IN	or_IN	ta_IN	en_XX	ml_IN	pa_IN	kn_IN	te_IN	AVG
politicians	0.747	0.833	0.644	0.617	0.629	0.421	0.536	0.578	0.517	0.510	**0.603**
cities	0.440	0.752	0.530	0.662	0.517	0.569	0.653	0.823	0.414	0.593	**0.595**
books	0.885	0.731	0.566	0.365	0.523	0.421	0.684	0.661	0.555	0.497	**0.589**
writers	0.642	0.656	0.509	0.480	0.462	0.402	0.531	0.539	0.452	0.453	**0.513**
companies	0.597	0.788	0.589	0.488	0.583	0.638	0.552	0.770	0.510	0.425	**0.594**
sportsman	0.737	0.841	0.569	0.618	0.669	0.549	0.601	0.655	0.485	0.442	**0.617**
films	0.538	0.649	0.578	0.688	0.601	0.633	0.557	0.659	0.746	0.643	**0.629**
animals	0.530	0.643	0.535	0.504	0.606	0.385	0.647	0.629	0.481	0.470	**0.543**
AVG	**0.639**	**0.737**	**0.565**	**0.553**	**0.574**	**0.502**	**0.595**	**0.664**	**0.520**	**0.504**	0.585

Table A.4: Rouge-L scores for RL methods for mBART and mT5 respectively.

We adopt a semi-training approach, initially fine-tuning the model for five epochs, followed by an additional five epochs of training with the reinforcement learning (RL) function. For this experiment, we employ mT5 [13] and mBART [60], with (lang, dom) based FSA. The outcomes are detailed in Table A.4. Our observation indicates that the overall results are inferior compared to a standard ten-epoch fine-tuning approach, although its better than employing pure FSA alone. Furthermore, we note that mT5 outperforms mBART, consistent with observations from standard fine-tuning experiments. In summary, we find that RL fails to yield performance improvements over the traditional fine-tuning method.

A.3 Using Reference text as Context

Inspired by XWikiGen and Dynamic-ToC [20], we employ a custom RL reward based methodoly using references as the input for Outline Generation. Similar to XWikiGen, we perform a two-stage process, where the first is unsupervised extractive summarization and the second step is RL-based finetuning. Since we know that HipoRank outperforms QA-GNN based summarization, we use HipoRank for our extractive summarization step. Once we get our summarized sentences, we then perform a finetuning step. The proposed pipeline can be seen in Figure A.2.

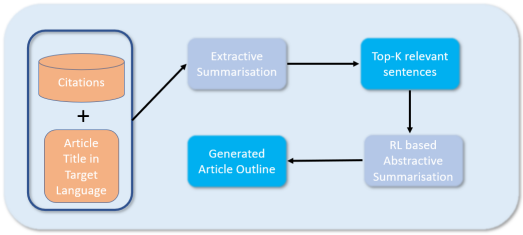

Figure A.2: Reinforcement Learning with 2-stage summarization.

In the initial stage of our information extraction pipeline, the output may sometimes appear disjointed and expressed in the language different than that of the source text. To refine this output into a coherent form, we employ a secondary stage. In this phase, we conducted experiments using two multilingual natural language generation models: mBART-large [60] and mT5-base [13].

Given that the input for the second stage can vary significantly in terms of writing styles and subject matter, we have to ensure that the generated outline aligns well with the source text. To address this, we drew inspiration from [20] and introduced two reward functions into our generation model within a reinforcement learning framework.

The first reward function is the Section-title compatibility reward. It involves fine-tuning an XLM-RoBERTa-based binary classifier [50] to evaluate the coherence between the generated section title and the input reference text. This classifier helps determine whether the generated title accurately reflects the content of the source text.

The second reward function is the Entity Correctness Reward. It focuses on ensuring the accuracy of named entities mentioned in the generated title. To achieve this, we utilized IndicNER [61] to extract named entities from both the generated title and the input reference sentences. This approach helps identify any discrepancies or hallucinations in the generated title, ensuring its alignment with the source material.

Table A.5 displays the scores for Rouge-L for mT5 and mBART trained along with RL objective with the custom reward functions. We observe that mT5 outperforms mBART by a lot, which is in line with other non-RL experiments as well. We also note that the scores seen on using references as context is a

lot lesser than when using minimal information. Our hypothesis is that it happens since we give it too much and too varying input as compared to somewhat similar output. Hence, it is not able to generalise well and instead increases variance by a lot.

	bn_IN	en_XX	kn_IN	te_IN	hi_IN	ml_IN	mr_IN	or_IN	pa_IN	ta_IN	Average
animals	0.467	0.548	0.053	0.044	0.464	0.572	0.322	0.170	0.071	0.197	**0.291**
books	0.430	0.438	0.018	0.406	0.913	0.337	0.293	0.537	0.190	0.529	**0.409**
cities	0.446	0.566	0.054	0.067	0.523	0.111	0.117	0.144	0.102	0.077	**0.221**
companies	0.420	0.537	0.069	0.174	0.348	0.275	0.339	0.000	0.000	0.211	**0.237**
films	0.641	0.529	0.532	0.646	0.510	0.700	0.439	0.767	0.277	0.599	**0.564**
politicians	0.642	0.346	0.163	0.371	0.730	0.458	0.365	0.581	0.261	0.433	**0.435**
sportsman	0.472	0.526	0.261	0.237	0.593	0.329	0.541	0.168	0.457	0.314	**0.390**
writers	0.405	0.370	0.139	0.262	0.543	0.289	0.450	0.430	0.344	0.212	**0.345**
	0.490	**0.482**	**0.161**	**0.276**	**0.578**	**0.384**	**0.358**	**0.350**	**0.213**	**0.321**	**0.481**

	bn_IN	en_XX	kn_IN	te_IN	hi_IN	ml_IN	mr_IN	or_IN	pa_IN	ta_IN	Average
animals	0.368	0.359	0.044	0.056	0.219	0.646	0.307	0.177	0.071	0.120	**0.237**
books	0.361	0.487	0.015	0.333	0.914	0.288	0.355	0.686	0.032	0.530	**0.400**
cities	0.125	0.530	0.100	0.064	0.498	0.036	0.250	0.409	0.023	0.066	**0.210**
companies	0.283	0.429	0.100	0.154	0.338	0.158	0.245	0.000	0.000	0.209	**0.192**
films	0.536	0.556	0.566	0.601	0.482	0.650	0.400	0.741	0.083	0.568	**0.518**
politicians	0.576	0.267	0.263	0.319	0.746	0.385	0.402	0.601	0.105	0.408	**0.407**
sportsman	0.321	0.509	0.295	0.227	0.580	0.318	0.577	0.204	0.350	0.298	**0.368**
writers	0.319	0.301	0.211	0.220	0.534	0.217	0.426	0.504	0.122	0.197	**0.305**
	0.361	**0.430**	**0.199**	**0.247**	**0.539**	**0.337**	**0.370**	**0.415**	**0.098**	**0.300**	**0.436**

Table A.5: Rouge-L scores of mT5 and mBART respectively for custom reward based RL with references as context.

Related Publications

- Dhaval Taunk, Shivprasad Sagare, Anupam Patil, **Shivansh Subramanian**, Manish Gupta, and Vasudeva Varma. **XWikiGen: Cross-Lingual Summarization for Encyclopedic Text Generation in Low Resource Languages**. In Proceedings of the ACM Web Conference 2023 (WWW '23). Association for Computing Machinery, New York, NY, USA, 1703-1713.
- **Shivansh Subramanian**, Dhaval Taunk, Manish Gupta, Vasudeva Varma. **OutlineGen: Multilingual Outline Generation for Encyclopedic Text in Low Resource Languages**. Submitted to International Conference on Advances in Social Networks Analysis and Mining, 2024.
- **Shivansh Subramanian**, Dhaval Taunk, Manish Gupta, Vasudeva Varma. **XOutlineGen: Cross-lingual Outline Generation for Encyclopedic Text in Low Resource Languages**. In Proceeding of the Wiki Workshop ′23.
- **Shivansh Subramanian***, Ankita Maity*, Aakash Jain*, Bhavyajeet Singh*, Harshit Gupta*, Lakshya Khanna* and Vasudeva Varma, **Cross-Lingual Fact Checking: Automated Extraction and Verification of Information from Wikipedia using References**, ICON 2023 Main Conference.

Other Publications

- Sravani Boinepelli, **Shivansh Subramanian**, Abhijeeth Singham, Tathagatha Raha and Vasudeva Varma. **Towards Capturing Changes in Mood and Identifying Suicidality Risk**. In Proceedings of the Eighth Workshop on Computational Linguistics and Clinical Psychology 2022.
- Harshit Gupta, Manav Chaudhary, Tathagatha Raha, **Shivansh Subramanian** and Vasudeva Varma. **Improving Conventional Prompting Methods for Brain Teasers**. Accepted at SemEval 2024.

Bibliography

[1] Tushar Abhishek, Shivprasad Sagare, Bhavyajeet Singh, Anubhav Sharma, Manish Gupta, and Vasudeva Varma. Xalign: Cross-lingual fact-to-text alignment and generation for low-resource languages. In *Companion Proceedings of the Web Conference 2022*, pages 171–175, 2022.

[2] Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng. Outline generation: Understanding the inherent content structure of documents. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 745–754, 2019.

[3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[4] R Lebret, D Grangier, and M Auli. Neural text generation from structured data with application to the biography domain. In *EMNLP*, pages 1203–1213, 2016.

[5] H Mei, M Bansal, and M R Walter. What to talk about and how? selective gen. using lstms with coarse-to-fine alignment. In *NAACL-HLT*, pages 720–730, 2016.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[7] H Shahidi, M Li, and J Lin. Two birds, one stone: A simple, unified model for text generation from structured and unstructured data. In *ACL*, pages 3864–3870, 2020.

[8] P Nema, S Shetty, P Jain, A Laha, K Sankaranarayanan, and M M Khapra. Generating descriptions from structured data using a bifocal attention mechanism and gated orthogonalization. In *NAACL-HLT*, pages 1539–1550, 2018.

[9] P Vougiouklis, H Elsahar, L-A Kaffee, C Gravier, F Laforest, J Hare, and E Simperl. Neural wikipedian: Generating textual summaries from knowledge base triples. *J. Web Semantics*, 52:1–15, 2018.

[10] L F R Ribeiro, M Schmitt, H Schütze, and I Gurevych. Investigating pretrained language models for graph-to-text generation, 2021.

[11] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.

[12] Shivprasad Sagare, Tushar Abhishek, Bhavyajeet Singh, Anubhav Sharma, Manish Gupta, and Vasudeva Varma. Xf2t: Cross-lingual fact-to-text generation for low-resource languages. *arXiv preprint arXiv:2209.11252*, 2022.

[13] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, 2021.

[14] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*, 2018.

[15] Hiroaki Hayashi, Prashant Budania, Peng Wang, Chris Ackerson, Raj Neervannan, and Graham Neubig. Wikiasp: A dataset for multi-domain aspect-based summarization. *Transactions of the Association for Computational Linguistics*, 9:211–225, 2021.

[16] Diego Antognini and Boi Faltings. Gamewikisum: a novel large multi-document summarization dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6645–6650, 2020.

[17] Demian Gholipour Ghalandari, Chris Hokamp, John Glover, Georgiana Ifrim, et al. A large-scale multi-document summarization dataset from the wikipedia current events portal. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1302–1308, 2020.

[18] George Giannakopoulos, Jeff Kubina, John Conroy, Josef Steinberger, Benoit Favre, Mijail Kabadjov, Udo Kruschwitz, and Massimo Poesio. 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*, pages 270–274, 2015.

[19] Pavel Tikhonov and Valentin Malykh. Wikimulti: a corpus for cross-lingual summarization. *arXiv preprint arXiv:2204.11104*, 2022.

[20] Himanshu Maheshwari, Nethraa Sivakumar, Shelly Jain, Tanvi Karandikar, Vinay Aggarwal, Navita Goyal, and Sumit Shekhar. Dynamictoc: Persona-based table of contents for consumption of

long documents. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5133–5143, 2022.

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

[22] Z Chi, L Dong, S Ma, S Huang, X-L Mao, H Huang, and F Wei. Mt6: Multilingual pretrained text-to-text transformer with translation pairs, 2021.

[23] Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. Cross-lingual natural language generation via pre-training. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7570–7577, 2020.

[24] J Zhu, Q Wang, Y Wang, Y Zhou, J Zhang, S Wang, and C Zong. Ncls: Neural cross-lingual summarization. In *EMNLP-IJCNLP*, pages 3054–3064, 2019.

[25] Eleftheria Briakou, Di Lu, Ke Zhang, and Joel Tetreault. Olá, bonjour, salve! xformal: A benchmark for multilingual formality style transfer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3199–3216, 2021.

[26] Qingyu Zhang, Xiaoyu Shen, Ernie Chang, Jidong Ge, and Pengke Chen. Mdia: A benchmark for multilingual dialogue generation in 46 languages. *arXiv preprint arXiv:2208.13078*, 2022.

[27] Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. *arXiv preprint arXiv:2106.13822*, 2021.

[28] Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. Mlsum: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, 2020.

[29] Tahmid Hasan, Abhik Bhattacharjee, Wasi Uddin Ahmad, Yuan-Fang Li, Yong-Bin Kang, and Rifat Shahriyar. Crosssum: Beyond english-centric cross-lingual abstractive text summarization for 1500+ language pairs. *arXiv preprint arXiv:2112.08804*, 2021.

[30] Khanh Nguyen and Hal Daumé III. Global voices: Crossing borders in automatic news summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 90–97, 2019.

[31] Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. Wikilingua: A new benchmark dataset for cross-lingual abstractive summarization. *arXiv preprint arXiv:2010.03093*, 2020.

[32] Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models, 2023.

[33] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), mar 2023.

[34] Keshav Kolluru, Muqeeth Mohammed, Shubham Mittal, Soumen Chakrabarti, and Mausam . Alignment-augmented consistent translation for multilingual open information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2502–2517, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[35] Bhavyajeet Singh, Siri Venkata Pavan Kumar Kandru, Anubhav Sharma, and Vasudeva Varma. Massively multilingual language models for cross lingual fact extraction from low resource Indian languages. In *Proceedings of the 19th International Conference on Natural Language Processing (ICON)*, pages 11–18, New Delhi, India, December 2022. Association for Computational Linguistics.

[36] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification, 2018.

[37] Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. ProoFVer: Natural logic theorem proving for fact verification. *Transactions of the Association for Computational Linguistics*, 10:1013–1030, 2022.

[38] Shyam Subramanian and Kyumin Lee. Hierarchical Evidence Set Modeling for automated fact extraction and verification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7798–7809, Online, November 2020. Association for Computational Linguistics.

[39] Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. Zero-shot fact verification by claim generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 476–483, Online, August 2021. Association for Computational Linguistics.

[40] Kung-Hsiang Huang, ChengXiang Zhai, and Heng Ji. CONCRETE: Improving cross-lingual fact-checking with cross-lingual retrieval. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1024–1035, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[41] Abhishek Nadgeri, Anson Bastos, Kuldeep Singh, Isaiah Onando Mulang', Johannes Hoffart, Saeedeh Shekarpour, and Vijay Saraswat. KGPool: Dynamic knowledge graph context selection for relation extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 535–548, Online, August 2021. Association for Computational Linguistics.

[42] Qi Zhu, Hao Wei, Bunyamin Sisman, Da Zheng, Christos Faloutsos, Xin Luna Dong, and Jiawei Han. Collective multi-type entity alignment between knowledge graphs. In *Proceedings of The Web Conference 2020*, WWW '20, page 2241–2252, New York, NY, USA, 2020. Association for Computing Machinery.

[43] Ishani Mondal, Yufang Hou, and Charles Jochim. End-to-end construction of NLP knowledge graph. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1885–1895, Online, August 2021. Association for Computational Linguistics.

[44] Mingyang Chen, Wen Zhang, Yushan Zhu, Hongting Zhou, Zonggang Yuan, Changliang Xu, and Huajun Chen. Meta-knowledge transfer for inductive knowledge graph embedding. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 927–937, New York, NY, USA, 2022. Association for Computing Machinery.

[45] Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, 2020.

[46] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[47] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2019.

[48] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, 2021.

[49] Yue Dong, Andrei Mircea, and Jackie CK Cheung. Discourse-aware unsupervised summarization of long scientific documents. *arXiv preprint arXiv:2005.00513*, 2020.

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

[51] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[52] Christina Sauper and Regina Barzilay. Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 208–216, 2009.

[53] Siddhartha Banerjee and Prasenjit Mitra. Wikiwrite: Generating wikipedia articles automatically. In *IJCAI*, pages 2740–2746, 2016.

[54] Dhaval Taunk, Shivprasad Sagare, Anupam Patil, Shivansh Subramanian, Manish Gupta, and Vasudeva Varma. Xwikigen: Cross-lingual summarization for encyclopedic text generation in low resource languages. In *Proceedings of the ACM Web Conference 2023*, pages 1703–1713, 2023.

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

[56] Young Mo Kang, Wenhao Liu, and Yingbo Zhou. Queryblazer: efficient query autocompletion framework. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1020–1028, 2021.

[57] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[58] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A python natural language processing toolkit for many human languages. In Asli Celikyilmaz and Tsung-Hsien Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online, July 2020. Association for Computational Linguistics.

[59] Bhavyajeet Singh, Siri Venkata Pavan Kumar Kandru, Anubhav Sharma, and Vasudeva Varma. Massively multilingual language models for cross lingual fact extraction from low resource Indian languages. In Md. Shad Akhtar and Tanmoy Chakraborty, editors, *Proceedings of the 19th International Conference on Natural Language Processing (ICON)*, pages 11–18, New Delhi, India, December 2022. Association for Computational Linguistics.

[60] Kaushal Kumar Maurya, Maunendra Sankar Desarkar, Yoshinobu Kano, and Kumari Deepshikha. Zmbart: An unsupervised cross-lingual transfer framework for language generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2804–2818, 2021.

[61] Arnav Mhaske, Harshit Kedia, Sumanth Doddapaneni, Mitesh M. Khapra, Pratyush Kumar, Rudra Murthy V au2, and Anoop Kunchukuttan. Naamapadam: A large-scale named entity annotated data for indic languages, 2023.