# References as Building Blocks: Investigating their Significance in Encyclopedic Text Generation

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

*Master of Science in* ***Computer Science and Engineering*** *by Research*

by

Dhaval Taunk
2021701028
dhaval.taunk@research.iiit.ac.in

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

International Institute of Information Technology
Hyderabad, India

# CERTIFICATE

It is certified that the work contained in this thesis, titled "**References as Building Blocks: Investigating their Significance in Encyclopedic Text Generation**" by **Dhaval Taunk**, 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 beloved family, for your unwavering love and support throughout this journey.*

# Acknowledgments

I would like to express my heartfelt gratitude to everyone who has been a part of my journey in last two years of my pursuit of MS by Research degree. It has been an incredible journey, filled with invaluable knowledge, personal growth, and unforgettable memories.

First and foremost, I would like to extend my deepest appreciation and gratitude for Prof. Vasudeva Varma sir for his invaluable guidance, support, and mentorship throughout the entire process of completing my thesis. I would also like to acknowledge Vasudeva sir's unwavering support and availability whenever I needed assistance or had questions. Thank you Vasudeva sir for allowing me to be a part of iREL family and giving me the chance to collaborate with many amazing people.

I am very much grateful and fortunate to have Dr. Manish Gupta sir as my co-advisor. His dedication towards his work, focusing on every minute detail, ready to take up any challenge attitude always motivates me to do more work . Thank you Manish sir, for helping with every point of my journey whether it be on exploring ideas, performing experiments or paper writing and many other things.

I would also like to thank Makarand Tapaswi sir and Charu Sharma ma'am for their guidance in the course Topics in Deep Learning and helping in getting a paper published in that course. Without their support and guidance, it would have a very difficult task in getting the paper through it.

Next, I would like to thank my parents (my father, mother, bua), my sister and newly added family member (my bother-in-law) for believing in me and my decisions and supporting me at every point of my life.

My time at IIIT was made more memorable by the many people I met here who supported me on my MS journey. I first met Sagar when I started doing my research. We collaborated on two shared tasks together, which greatly aided in my learning. Without my great co-authors Shivprasad, Shivansh, Pavan, and Lakshya, I would have found it extremely challenging to complete my research. I would also like to thank my senior Tushar for giving me his insightful feedback at various points during my research. Last but not least, I want to thank my labmates Bhavyajeet, Tathagata, Ankita, Aditya, Nirmal, and obviously there are more.

Lastly, I could not have survived on campus without the unwavering support of my pals Aditya, Amruth, Bhoomeendra, Dhruv, Nayan, Prateek. My time at IIIT was made unforget-

table by our random conversations about both research and non-research topics. Also, thanks to Siddhant for being a friendly & helpful next door neighbor who introduced me to an interesting game called CATAN.

# Abstract

Automated text generation for *low resource (LR) languages* is a critical area of research because of lack of contributors to encyclopedic texts, notably on Wikipedia. The majority of the work done so far on Wikipedia text generation has concentrated on creating English-only Wikipedia articles by summarizing English reference articles. Monolingual text generation is unable to address this issue for low-resource languages due to the lack of reference materials.

To start addressing these problems, we propose a benchmark dataset called XWikiRef that consists of ∼69K in Wikipedia articles from five different domains and eight different languages. Utilizing this dataset, we train a two-stage system that outputs a section-specific LR summary from an input of a set of citations and a section title.

One crucial aspect of content organization is the creation of article outlines, which summarize the primary topics and subtopics covered in an article in a structured manner. We introduce a pipeline called XOutlineGen, which generates cross-lingual outlines for encyclopedic texts from reference articles. XOutlineGen uses the XWikiRef dataset, which consists of encyclopedic texts generated from reference articles and section titles. Our pipeline employs this dataset to train a two-step generation model, which takes the article title and set of references as inputs and produces the article outline.

Commonsense question-answering (QA) methods combine the power of pre-trained Language Models (LM) with the reasoning provided by Knowledge Graphs (KG). A typical approach collects nodes relevant to the QA pair from a KG to form a Working Graph (WG) followed by reasoning using Graph Neural Networks (GNNs). This faces two major challenges: (i) it is difficult to capture all the information from the QA in the WG, and (ii) the WG contains some irrelevant nodes from the KG. To address these, we propose GrapeQA with two simple improvements on the WG: (i) Prominent Entities for Graph Augmentation identifies relevant text chunks from the QA pair and augments the WG with corresponding latent representations from the LM, and (ii) Context-Aware Node Pruning removes nodes that are less relevant to the QA pair. We evaluate our results on OpenBookQA, CommonsenseQA and MedQA-USMLE and see that GrapeQA shows consistent improvements over its LM + KG predecessor (QA-GNN in particular) and large improvements on OpenBookQA. We utilize the idea of relevance scoring from this work in our next work which is called XWikiGen for performing neural extractive summarization.

With this study, we propose XWikiGen, a task of cross-lingual multi-document summarization of text from numerous reference articles written in different languages to produce Wikipedia-style material. The suggested approach is built on the novel idea of using neural unsupervised extractive summarization to roughly select salient information and then using a neural abstractive model to produce the section-specific text. Extensive experiments have revealed that multi-domain training generally outperforms a multi-lingual and multi-lingual-multi-domain perform best, even better then previous two settings.

Overall, we propose a new dataset called XWikiRef for the task of encyclopedic text generation, a 2 stage pipeline XOutlineGen to generate article outline from references and a cross-lingual multi-document summarization based 2 stage pipeline XWikiGen to generate Wikipedia style text. Along with these, we also explore the idea of relevance scoring first in the domain of question answering with reasoning (GrapeQA) and then in the context of unsupervised extractive summarization.

# Contents

# List of Figures

# List of Tables

*Chapter 1*

# Introduction

## 1.1 Motivation

### 1.1.1 Inequality in volume of encyclopedic content present in Low Resource Languages vs English

For millions of people, Wikipedia is their go-to source for encyclopedic reference. Unfortunately, the number of articles in Wikipedia for low-resource (LR) languages is incredibly low. With ∼6.56 million articles conveyed in 54.2 GB of text, English Wikipedia displays plenty while Wikipedia is in bad shape, with only ∼90K worth of articles represented using an average of 7.5 GB of text across seven low-resource languages, as illustrated in Fig 1.1. Additionally, as shown in Fig 1.2, manual efforts to enrich the LR Wikipedia over the years have not been

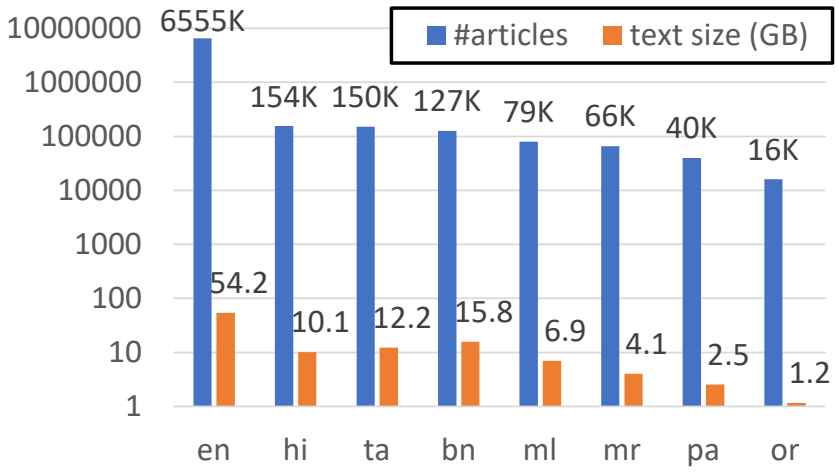


Figure 1.1: Number of Wikipedia articles and text size in GBs across eight languages, using 20220926 Wikipedia dump. Note that the Y axis is in log scale.

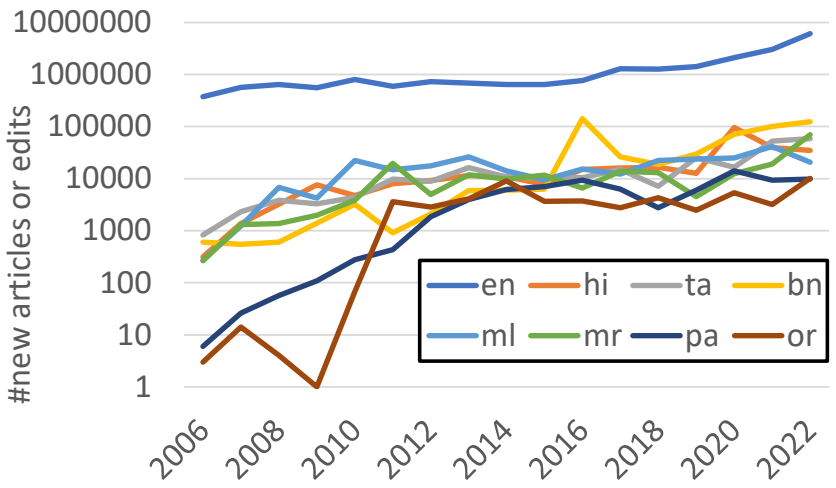


Figure 1.2: Number of new articles or edits on Wikipedia across eight languages from 2006 to 2022. This is obtained using a publication date from the 20220926 Wikipedia dump. Note that the Y axis is in the log scale.

as successful as they were for the English version. These findings suggest that automated text generation is essential for Wikipedia's low-resource languages based articles.

### 1.1.2 Possible approaches for automated encyclopedic text generation

Text from equivalent English Wikipedia pages can be translated as a possible naïve technique for the automatic generation of articles in low-resource Wikipedia. Unfortunately, a lot of low-resource entities of interest have a tendency to be local in nature. As a result, there are, on average, only 42.1% of entities in seven low-resource languages that have equivalent English Wikipedia pages. The percentages of Wikipedia entities that do not have an English-language equivalent Wikipedia pages are as follows: Hindi (50.60%), Tamil (46.70%), Bengali (31.5%), Malayalam (36.30%), Marathi (42.00%), Punjabi (38.70%), and Oriya (39.40%) all have higher percentages than English. In order to generate Wikipedia text using LR, we must therefore investigate other inputs.

Utilizing generic Web content to generate Wikipedia text is another strategy. This strategy faces a hurdle because low-resource languages typically have relatively little online information, as seen in publicly accessible huge dumps like CommonCrawl [60]. As a result, creating monolingual parallel datasets in LR languages is not viable. This encourages us to investigate the usage of cross-lingual strategies for our task.

## 1.2 Cross-lingual, Multi-Document, Multi-Domain dataset (XWikiRef)

With the help of the langdetect library[1], we examined the language of cited references on existing Wikipedia pages for eight languages and five domains. The quantities are negligible for the majority of (domain, LR language) combinations, as shown by Table 1.1, despite the fact that English Wikipedia pages contain more than 85% references in English. This inspires us to create XWikiRef, a cross-lingual, multi-document, multi-domain dataset to generate the encyclopedic text in low resource languages.

| **Domain** | **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 1.1: Percentage of cited references with language same as Wikipedia article language (for 8 languages and 5 domains which are a part of our XWikiRef dataset).

The dataset was gathered from Wikipedia pages that correspond to eight languages and five domains. Bengali (bn), English (en), Hindi (hi), Malayalam (ml), Marathi (mr), Oriya (or), Punjabi (pa), and Tamil (ta) are among the languages. Books, films, politicians, sportsmen, and writers are among the several domains. The dataset includes ∼69K Wikipedia articles with ∼105K sections. The average number of references cited per section is 5.44.

## 1.3 Cross-lingual Outline Generation using references (XOutlineGen)

One crucial aspect of content organization is the creation of article outlines [82], which summarize the primary topics and subtopics covered in an article in a structured manner. The articles in low-resource languages often contain entities that are specific to the region and not well-known globally. To create an outline for such an article, one approach is to translate or copy the outline from a similar article in English or the same language within the same domain.

[1] `https://pypi.org/project/langdetect/`

However, these methods require the user to be familiar with the entity and to be able to identify other similar articles with outlines that can be used as a reference.

In this work, we propose a cross-lingual outline generation task XOutlineGen for low resource languages where we utilize the reference URL's as input and generate the corresponding outline in the targeted low resource language.

## 1.4 Cross-lingual Encyclopedic Text Generation (XWikiGen)

This work proposes a novel task XWikiGen where we utilize XWikiRef dataset to create a pipeline to automatically generate the encyclopedic text in low resource languages.

XWikiGen accepts a set of reference URLs, the title of the target section, and the language in which it should be shown as input. The text that is appropriate for that Wikipedia article in the target language is the final generated text. Section-wise text generation is what XWikiGen entails as opposed to creating the complete Wikipedia page, just like generic summarizing differs from query-based summary. XWikiGen is cross-lingual, in contrast to earlier efforts to create English-only Wikipedia texts. Finally, unlike some other works that generate cross-lingual text using English Wikipedia pages, XWikiGen focuses on producing cross-lingual text using reference URLs in various languages.

The task XWikiGen is extremely difficult because it aims for the generation of long, cross-lingual encyclopedic text. Long text input can be challenging. As a result, we adopt a two-stage strategy. Important phrases are picked up in the first extraction stage across several reference texts. The section text is generated during the second abstractive stage. In both phases, neural models are used. For the extractive stage, we test unsupervised techniques like salience ([81] & GrapeQA), and hiporank [20], and for the abstractive step, mT5 [79] and mBART [46]. We test out multilingual, multi-domain, and multi-lingual-multi-domain training settings. Using standard text generation measures like ROUGE-L, METEOR, and chrF++, we provide our findings.

## 1.5 Question-Answering using language model and knowledge graphs with reasoning (GrapeQA)

For our salience based extractive summarization technique, we explored an approach where we find relevance of a reference text sentence w.r.t. to the section title. We then rank these sentences based on the salience score to extract top-k sentences.

In this work, we first explore the idea of relevance scoring mechanism inspired from QAGNN [81] in our work GrapeQA and propose two techniques viz. *Prominent Entities for Graph Augmentation (PEGA)* and *QA Context-Aware Node Pruning (CANP)* to improve the model perfor-

mance on the question-answering task. Thereafter, we explore this relevance scoring mechanism in our XWikiGen task.

## 1.6 Thesis Key Contributions

With this work, we highlight the scarcity of encyclopedic content in low resource languages and motivate research in this direction by making use of the English based rich web content to enhance encyclopedic content in low resource languages using cross-lingual based text generation approaches. We make the following key contributions with this work.

1. **XWikiRef**, a cross-lingual, multi-document, multi-domain dataset for the task of cross-lingual automatic outline and encyclopedic text generation.

2. **XOutlineGen**, a pipeline to generate the outline of a encyclopedic article in low resource language from references in cross-lingual manner.

3. **XWikiGen**, a pipeline to generate Wikipedia style text from references using cross-lingual, multi-document summarization based approach.

4. **GrapeQA**, an approach to improve question-answering with reasoning and to explore the effect of relevance scoring mechanism in context of unsupervised extractive summarization.

## 1.7 Thesis Outline

Overall, this thesis is divided into 7 chapters, of which a small description of each chapter is given below:

- In **chapter 1** (this chapter), we highlight the problem of lack of information content in low resource languages and motivate the need to create methods to enhance the content in these low resource languages.

- **Chapter 2** discusses about some of the related work that has been done in the past for outline generation, encyclopedic text generation tasks and some of the related datasets.

- In **chapter 3**, we discuss the need of a relevant encyclopedic text generation dataset and propose our own dataset XWikiRef.

- In **chapter 4**, we utilize the XWikiRef dataset to generate the outline of a Wikipedia article from references by proposing a pipeline called XOutlineGen.

- **Chapter 5** proposes a methodology GrapeQA to enhance task of question-answering with reasoning and to explore the benefit of using relevance scoring mechanism in XWikiGen task.
- In **chapter 6**, we propose XWikiGen, a cross-lingual multi-document summarization based pipeline build using XWikiRef to generate encyclopedic text in low resource languages.
- **Chapter 7** concludes this thesis by discussing overall contributions and impact of the work. We also discuss possible future work in this direction to enhance content in low resource languages.

At last, appendix section of this thesis discusses some of the methods that had been explored in solving various challenges are faced in different types of NLP tasks. The chapter discusses challenges like summarization of Indian languages, profiling irony and stereotypes spreaders in Twitter and multilingual news article similarity.

*Chapter 2*

# Related work

## 2.1 Outline Generation

A very small amount of work is being done in the domain of outline generation for encyclopedic text and that too in English only. The idea of generating outline of a Wikipedia article is first explored in Outline Generation: Understanding the Inherent Content Structure of Documents [82]. With this work, the authors proposed a hierarchical bidirectional GRU based framework in generating outline of an existing English Wikipedia based article. They want to create a outline for a given document by first predicting a sequence of section boundaries and then a succession of section headings.

In the past couple of years, some of the works [51, 6] explored the idea of outline generation in the context of financial domain datasets like SEC filing [4]. The authors in [51] explore the idea of reinforcement learning by adding reward functions in generating a persona based table of content. They first perform aspect detection to filter out the region of interest in the document. Then they generate outline of that filtered document using transformers based generative models by adding RL based reward functions.

With the idea of performing topic segmentation, [3] proposed a new dataset called WikiSection which contained 38K full-text articles from English and German Wikipedia articles annotated with sections.

## 2.2 Wikipedia based Short Text Generation

For the past five to six years, the automated generation of Wikipedia text has been a topic of interest. The initial attempts in the fact-to-text (F2T) line of research were primarily concerned with producing short text, generally the first phrase of Wikipedia entries using structured fact tuples.

Ample content overlap and aligned data are necessary for training F2T models. Some earlier studies, such as WebNLG [22], collected aligned data through crowdsourcing, while others used

heuristics like TF-IDF to perform automatic alignment. For F2T, seq-2-seq neural approaches have been widely employed [37, 52]. Examples of these include plain LSTMs [74], LSTM encoder-decoder models with copy mechanisms [70], LSTMs with hierarchical attentive encoders [57], and pretrained Transformer based models [62], such as BART [39] and T5 [60].

The majority of the prior work on fact-to-text was solely done in English. The Cross-lingual F2T (XF2T) problem was very recently put up in the works of [1] and [63]. This work's emphasis is on producing longer text as opposed to all of these previous works, which have concentrated on short text generation. In contrast to F2T literature, where the input is structured, the input in our instance simply a collection of reference URLs.

## 2.3 Wikipedia based Long Text Generation

In addition to producing short Wikipedia text, efforts have also been made to produce Wikipedia articles by condensing lengthy sequences [45, 23, 27, 2, 24, 73], as seen in Table 2.1 & 2.2. The generated text for each of these datasets either corresponds to the entire Wikipedia page or a selected section. The majority of these investigations [45, 27, 2, 23] were conducted in English exclusively. Additionally, these studies employ a variety of input formats, including single documents (an existing Wikipedia article in the same or a different language) and multi-document input formats (a collection of citation URLs, review pages).

| Dataset | #Summaries | XL? | ML? | #Langs | MD? | SS? |
|---|---|---|---|---|---|---|
| WikiSum [45] | ∼2.3M articles | No | No | 1 | Yes | No |
| WikiAsp [27] | ∼400K sections | No | No | 1 | Yes | Yes |
| GameWikiSum [2] | ∼26K gameplay Wikipedia sections | No | No | 1 | Yes | No |
| Wiki Current Events Portal (WCEP) [23] | ∼10.2K WCEP event summaries | No | No | 1 | Yes | No |
| MultiLing'15 [24] | ∼1.5K paragraphs | No | Yes | 38 | No | No |
| WikiMulti [73] | ∼150K intro paragraph | Yes | Yes | 15 | No | No |
| XWikiRef (Ours) | ∼105K sections | Yes | Yes | 8 | Yes | Yes |

Table 2.1: Statistics of popular Wikipedia Summarization datasets. XL=Cross-lingual. ML=Multi-Lingual. MD=Multi-document. SS=Section-specific.

| Dataset | Input | Output |
|---|---|---|
| WikiSum [45] | Set of citation URLs | Whole Wiki article |
| WikiAsp [27] | Set of citation URLs | One section in same language |
| GameWikiSum [2] | Professional video game reviews | Gameplay Wikipedia sections |
| Wiki Current Events Portal (WCEP) [23] | Set of news articles | WCEP Summary |
| MultiLing'15 [24] | Whole Wikipedia article | First few Wikipedia sentences in same language |
| WikiMulti [73] | Whole Wikipedia article | Intro paragraph in other language |
| XWikiRef (Ours) | Set of citation URLs | One section in another language |

Table 2.2: Input-Output format of popular Wikipedia Summarization datasets.

Most of these works produce an article as a whole instead of summarizing the section-specific text. Hayashi et al. [27] introduced section-specific summarization, which identifies the main subjects in the input text and then constructs a summary for each, in order to capture the section-specific intent while summarizing. The model is used by authors to identify the latent subtopics, however the content selection process is difficult. In order to overcome this problem, we use section-specific citations as input in our dataset. By doing so, we can explore the model's summarization abilities more thoroughly and avoid using noisy references from other parts.

It is interesting that no dataset currently available summarizes Wikipedia text from many languages. However, this setup is essential for Wikipedia text generation for LR languages, as explained in the previous section.

## 2.4 Multi-lingual and cross-lingual summarization

Thanks to models like XNLG [12], mBART [46], mT5 [79] etc., there has been a lot of work recently on multi-lingual and cross-lingual NLG tasks like machine translation [11, 46], question generation [12, 56], news title generation [40], blog title generation [9], and summarization [85, 29].

Past research on summarizing for low-resource languages has been scant. MultiLing'15 [24] introduced a novel task for multi-lingual summarization in 30 languages. In the past 2–3 years, a few datasets have been proposed for cross-lingual summarization mainly in the news domain: XLSum [26], MLSum [69], CrossSum [25], Global Voices [58], WikiLingua [35], WikiMulti [73]. A set of carefully crafted heuristics was used to extract the ∼1.35 million professionally annotated article-summary pairs that make up XL-Sum [26] from the BBC. It covers 44 languages, ranging in resource level from low to high. By publishing CrossSum, a cross-lingual summary dataset with ∼1.7 million occurrences, Hasan et al. [25] expand the multilingual XL-Sum dataset. CrossSum and XL-Sum, however, are exclusively applicable to the news domain. WikiLingua [35] is a multilingual dataset with ∼770K summaries that includes article and summary pairs that were taken from WikiHow in 18 different languages. The cross-lingual summary databases MLSum and GlobalVoices, which are based on news stories, each contain about ∼1.5M and ∼300K worth of summaries in 5 and 15 languages, respectively. We add to this body of work by supplying a fresh cross-lingual, multi-document summarizing dataset, referred to as XWikiRef and by suggesting a two-stage method for the related XWikiGen problem.

## 2.5 Question Answering using LMs and KGs

Answering questions is a challenging NLP problem as it involves understanding the question context and sifting through relevant information to identify the answer. Question-answering models have evolved from rule-based [31] to RNN-based sequence models [53] and now to Transformer-based Language Models (LM) such as RoBERTa-large [47]. However, commonsense question-answering adds a layer of complexity as the model needs to reason about questions relating diverse topics, making the task challenging for LMs that may not have seen something similar in the pre-training data.

While LMs capture the implicit patterns and contextual information within the data, KGs are able to capture explicit relations between the text entities. KGs such as Freebase [10], Wikidata [75], or ConceptNet [71] store knowledge in the form of graph triplets (topic-relationship-topic) and are well suited for Graph Neural Networks (GNNs), *e.g.* [78]. Thus, commonsense QA in particular has attracted interest in combining LMs and KGs with the reasoning ability of GNNs [41, 80].

Most works on LM + KG extract a sub-graph or Working Graph (WG) from the KG based on concepts mentioned in the QA pair [41, 21, 80] and focus on improving reasoning. For example, [41] propose a graph network to score answers while [21] focus on a multi-hop message passing framework that allows each node to attend to multi-hop neighbors in a single layer, combining interpretable path-based reasoning with scalable GNNs. [80] improve the extracted WG through a relevance scoring mechanism followed by joint reasoning and [83] fuse information from both the modalities (LM, KG) by mixing their tokens and nodes.

*Chapter 3*

# Creating a multi-document, cross-lingual and multi-domain dataset

## 3.1 Overview

In this work, we propose a novel cross-lingual - multi-document summarization dataset XWikiRef, for the advancement of encyclopedic text generation. This dataset encompasses 8 languages and 5 domains. With a diverse collection of high-quality Wikipedia based texts from various, this dataset opens up opportunities for training models that can generate comprehensive and concise summaries in multiple languages. Each document in the dataset contains Wikipedia specific article text divided into its sections with each section comprising of the section text and corresponding reference text, enabling the development of robust algorithms capable of distilling key information from multiple sources. We believe that this new dataset will significantly contribute to the development of encyclopedic text generation approaches in low resources languages, empowering the creation of sophisticated systems that bridge language barriers and foster global knowledge sharing. We also provide a thorough analysis of the dataset in this chapter.

## 3.2 Data Collection and Pre-processing

The XWikiRef dataset comprises Wikipedia articles for eight different languages (bn, en, hi, ml, mr, or, pa, ta) and five different fields (books, movies, politicians, sportsmen, and writers). In order to find the entities that have Wikipedia pages in our collection of languages, we first use the Wikidata API[1] to filter the domains of interest. The Wikipedia pages of the filtered items are then extracted using language-specific Wikipedia 20220926 XML dump. The text on Wikipedia is organized into sections and subsections. We take the section and subsections from the text. Text in articles that are deeper than two levels is combined into parent sub-sections.

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

Using wiki markup, we also retrieve the citation URLs from each part. To remove all wiki markup from a specific section and get clean section content, we utilize the Python package `MediaWikiParserFromHell`[2]. In order to exclude file types other than HTML and PDF, we filter the URLs. We use `pdfminer`[3] to extract the paragraph content from pdf files and `BeautifulSoup`[4] in Python to scrape the paragraph text from the related webpages for each reference URL. Some files (hundreds of PDF pages) are too big. Therefore, we set a time limit of 5 seconds for each URL that we scrape. We also deal with common scraping errors like pages that do not exist. The scraped text is then tokenized into individual phrases using the IndicNLP [32] module's universal sentence tokenizer. We only keep the portions of the dataset that have at least one (crawlable) reference URL with text that is not empty.

The domain, language, section title, set of reference URLs, and Wikipedia section text are all included in each sample in the dataset. Then, stratified by domain and language, this dataset is divided into train, validation, and test in a 60:20:20 ratio.

## 3.3 Data Analysis and Stats

The following tables contain the specifics of our analysis of our curated dataset across several parameters. The entire number of articles in the XWikiRef dataset are displayed in Table 3.1. The amount of articles varies across domains per language due to the distribution of Wikipedia articles across domains. The total number of articles from which we extract section text for the dataset is ∼69K.

| **Domain/Lang** | **bn** | **hi** | **ml** | **mr** | **or** | **pa** | **ta** | **en** | **Total** |
|---|---|---|---|---|---|---|---|---|---|
| **Books** | 313 | 922 | 458 | 87 | 73 | 221 | 493 | 1467 | 4034 |
| **Film** | 1501 | 1025 | 2919 | 480 | 794 | 421 | 3733 | 1810 | 12683 |
| **Politicians** | 2006 | 3927 | 2513 | 988 | 1060 | 1123 | 4932 | 1628 | 18177 |
| **Sportsmen** | 5470 | 6334 | 1783 | 2280 | 319 | 1975 | 2552 | 919 | 21632 |
| **Writers** | 1603 | 2024 | 2251 | 784 | 498 | 2245 | 1940 | 714 | 12059 |
| **Total** | 10893 | 14232 | 9924 | 4619 | 2744 | 5985 | 13650 | 6538 | **68585** |

Table 3.1: XWikiRef: Total #articles per domain per language

The distribution of sections across different (domain, language) pairs is shown in Table 3.2 of the XWikiRef dataset. Furthermore, XWikiRef is a dataset of multiple document summaries,

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

[3]https://pypi.org/project/pdfminer/

[4]https://pypi.org/project/beautifulsoup4/

as was already mentioned. The average number of references per section for each (domain, language) pair is shown in Table 3.3. Every (domain, language) pair in the dataset has at least two references on average, as shown in the table, even though many of these references are not in the LR language.

| **Domain/Lang** | **bn** | **hi** | **ml** | **mr** | **or** | **pa** | **ta** | **en** | **Total** |
|---|---|---|---|---|---|---|---|---|---|
| **Books** | 434 | 987 | 557 | 111 | 88 | 238 | 598 | 2972 | 5985 |
| **Film** | 2139 | 1363 | 3737 | 676 | 1351 | 476 | 4781 | 4766 | 19289 |
| **Politicians** | 3261 | 4478 | 3719 | 1384 | 1404 | 1524 | 6431 | 4780 | 26981 |
| **Sportsmen** | 9485 | 8118 | 2642 | 3056 | 485 | 2624 | 3769 | 2698 | 32877 |
| **Writers** | 2598 | 2743 | 3435 | 1166 | 896 | 3034 | 3113 | 2409 | 19394 |
| **Total** | 17917 | 17689 | 14090 | 6393 | 4224 | 7896 | 18692 | 17625 | **104526** |

Table 3.2: XWikiRef: Total #sections per domain per language

| **Domain/Lang** | **bn** | **hi** | **ml** | **mr** | **or** | **pa** | **ta** | **en** |
|---|---|---|---|---|---|---|---|---|
| **Books** | 3.62 | 2.61 | 2.59 | 2.07 | 3.46 | 2.30 | 2.40 | 6.34 |
| **Film** | 4.85 | 7.14 | 3.34 | 2.96 | 3.81 | 4.10 | 3.83 | 12.74 |
| **Politicians** | 4.98 | 4.09 | 3.75 | 3.87 | 2.07 | 3.59 | 3.91 | 14.21 |
| **Sportsmen** | 6.37 | 8.30 | 6.96 | 4.20 | 3.93 | 4.49 | 6.38 | 21.88 |
| **Writers** | 5.20 | 5.46 | 4.16 | 3.74 | 2.85 | 3.34 | 4.20 | 17.61 |

Table 3.3: XWikiRef: Average number of references per section for each domain and language

The dataset's distribution of reference URLs across domains is seen in Fig. 3.1. The image demonstrates that multi-document summarization is critical by displaying multiple samples when there are 5+ reference URLs across all domains.

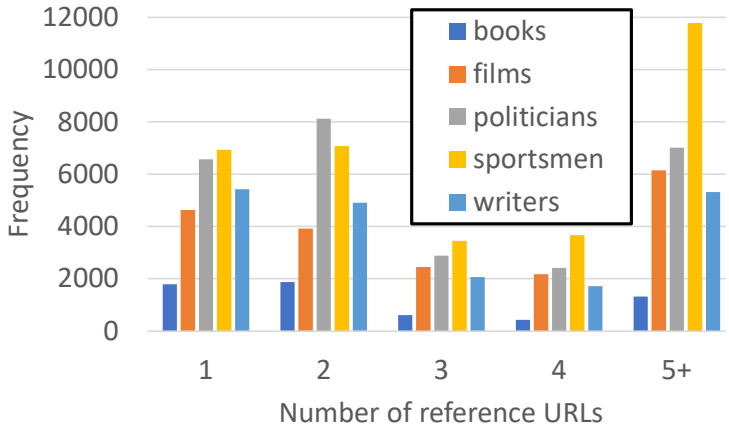


Figure 3.1: Distribution of number of reference URLs across domains in our XWikiRef dataset


Figure 3.2: 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.

Finally, in Fig. 3.2, we provide word clouds of the most popular Wikipedia section names for each of the five categories. The five most common titles in each word cloud are broken down by

language. Section titles are displayed in a single color for each language. Indicated by font size is relative frequency. The section titles' diversity for each (language, domain) pair is displayed in word clouds.

## 3.4 Summary

We propose a novel dataset XWikiRef with this work which includes ∼69K Wikipedia articles and ∼105K section specific summaries. It spans across 8 languages and 5 domains. We also propose two tasks of cross-lingual outline generation (XOutlineGen) and cross-lingual encyclopedic text generation (XWikiGen) which utilizes this dataset (discussed in coming chapters). We hope that this dataset helps in bridging the gap of encyclopedic content availability in low resource languages as compared to resourceful languages like English and other foreign languages.

*Chapter 4*

# Generating cross-lingual outline of encyclopedic articles

## 4.1 Overview

The rapid growth of Wikipedia as a comprehensive source of information in multiple languages has led to an increasing need for cross-lingual content organization. A critical aspect of content organization is the creation of article outlines, which provide a structured summary of the main topics and subtopics covered in an article. However, manually creating article outlines in multiple languages is time-consuming and resource-intensive. This work proposes XOutlineGen, which generates cross-lingual outlines for encyclopedic texts from reference articles. The work utilizes XWikiRef(Chapter 3) dataset, where encyclopedic texts were generated from the reference articles and section title. Our pipeline utilizes this dataset to train a two-step generation model, taking in the page title and the set of references and producing the outline for the article. The first stage is a neural unsupervised extractive summarization. The second stage is the outline generation stage which uses a Reinforcement Learning setup with two novel reward functions to nudge the output in the required direction.

The entities of low-resource articles are often local to the region; hence they may be unknown to the world. To generate an outline of such an article, a simplistic method may be to translate the outline from a similar article in English or to copy it from a similar domain in the same language. Both methodologies come with the problem of the user being aware of the entity and in such a manner that they can come up with other similar Wikipedia articles from which outlines can be determined. Another problem with such approaches is the lack of reference information in low-resource languages. An outline of an article will depend on the amount of information available online and what the information conveys. Hence, we propose XOutlineGen, a novel problem of cross-lingual Wikipedia outline generation using references. Fig. 4.1 shows an overview of our pipeline XOutlineGen which takes article title and reference URLs as input and generates the corresponding outline in the target language.

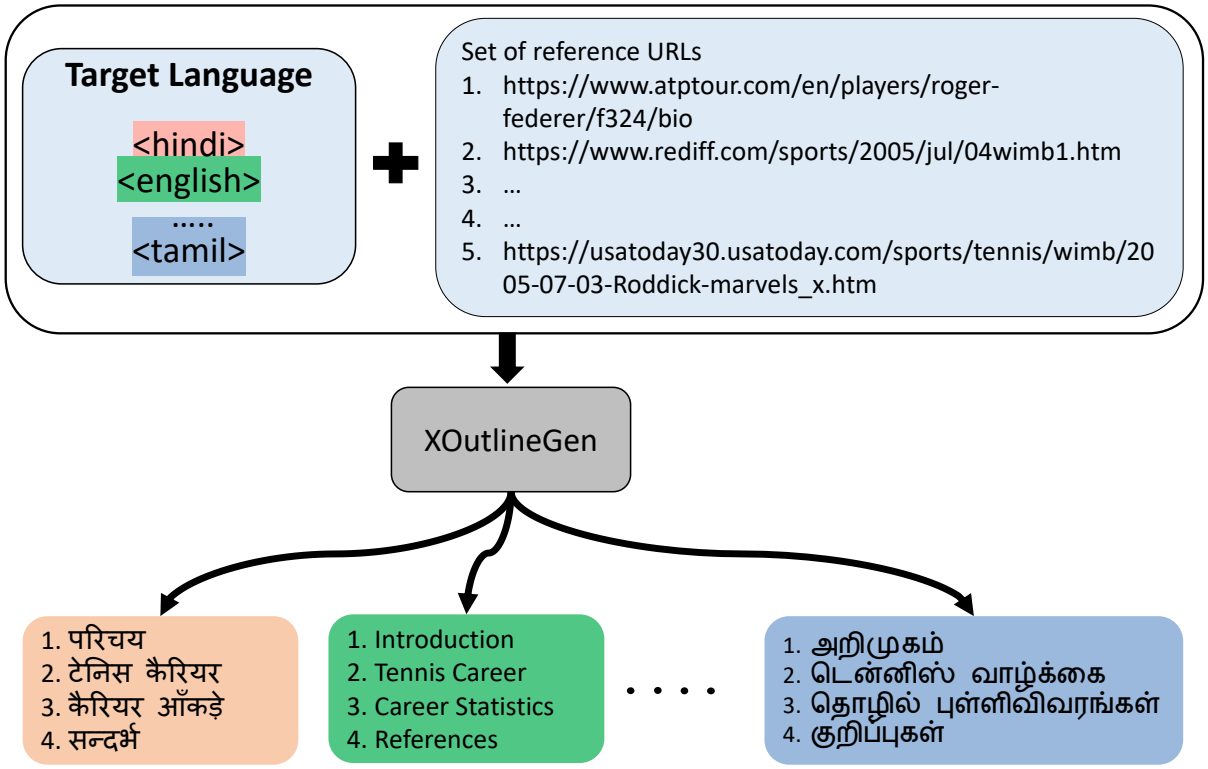


Figure 4.1: XOutlineGen examples: Generating Hindi, English, and Tamil Outline from cited references.

Overall we make the following contributions with this work:

- We motivate the need for the problem XOutlineGen, a cross-lingual outline generation task where the input is (article title, reference text) and the output is the outline for a Wikipedia article.
- We model XOutlineGen as a multi-document cross-lingual outline generation problem and propose a two-stage system with reward functions in a Reinforcement Learning based setting.

## 4.2 Methodology

The consistent structure of Wikipedia for particular language domain pairs gives us information about possible outline structure. More information can be gathered from the article title, reference text etc. In our methodology, we generate the article outline given the article title and reference text. Here, the key idea is that large multi-lingual language models have been trained on huge data in all languages. Because of it, they will have contextual knowledge of

the title to generate an appropriate outline. We use multi-lingual encoder-decoder transformer architecture like mT5 [79] and mBART [46] as our generation model. Our input to the model is the language domain pair, the article title and the reference text.

Multiple issues arise when providing large amounts of information as context to a model. Firstly, it is just too much information, and it greatly increases the time and computational cost. It is infeasible to have such long texts as input due to computational and time constraints on Transformers and other Transformer variants. In order to address this problem, we will select sentences based on importance from the reference text, which we then feed to the generation model to get the article outline. Hence, we reduce the text given to a model and only provide it with important sentences relevant to the article title. We select these sentences using a neural unsupervised extractive summarizer, HipoRank [20].

Another issue with generation from a lot of contextual information is that there is no guarantee that the output generated will be of the entity we need and in the format we need. One of the cons of the previous attempt of generating an outline by article title was how prone it would be to hallucinations, where it could generate an outline based on wrong world-knowledge understanding. Another area for improvement with generation is style compatibility. Since we are using reference text as context, which can occur in multiple styles, and can be about different things, we need to ensure that the outline generated is compatible with the source reference text. Hence, we need to ensure that the model does not hallucinate and generate an entity not present in the reference text and that the outline generated is compatible with the reference. We add rewards to the multi-lingual generation model in an RL setting to ensure that generated output is what we require. While training, these rewards are added to the loss of the multi-lingual encoder-decoder Transformers like mT5 and mBART.

The articles on Wikipedia can have references in various languages. To create an outline of an article that is cross-lingual, we used a two-stage approach. We first performed extractive summarization in stage 1 to extract the relevant information from reference text. This information was then used in stage two to create the overall outline. Additionally, we included RL based reward functions in stage 2 of the pipeline to ensure that the model produces a coherent and relevant outline.

### 4.2.1 Extractive Summarization Stage

1. **HipoRank based extractive summarization**: The unsupervised graph-based technique called Hierarchical and Positional Ranking technique (HipoRank) [20] is used to extractively summarize long input text. It generates a directed hierarchical network containing sentence and section nodes, as well as sentence-sentence and sentence-section edges with asymmetrically weighted edges, given a document with many sections. Then, a sentence node's score is calculated using a weighted sum of the edges incident on the node.

Using mBERT [17], we compute sentence node representations. To determine the representation for each section node, we use the mean of all the sentence representations within that section.

The intra-sectional and inter-sectional edges between each phrase node and the other nodes form the network. All of the sentences in a section are connected intra-sectionally, modeling the local significance of each sentence. The main point is that sentences that are comparable to the majority of sentences inside a segment are more significant. Contrarily, inter-sectional connections are made between section nodes and sentences and are designed to simulate the sentences' overall significance. Here, the concept is that the most significant sentences are those that most closely resemble previous sections. For the sake of efficiency, no edges are permitted between sentences that are in distinct parts.

To calculate edge weights, node embeddings are compared using their cosine similarity. Intra-sectional edges are given more weight if they are incident to a boundary sentence based on the supposition that significant sentences are located close to the borders (start or end) of a text. Similar to this, key portions are located close to the document's edges. This theory is applied to properly weigh inter-sectional edges. Finally, a weighted sum of the edges (both intra-sectional and inter-sectional) incident on the sentence node is used to calculate the importance score for the node. The top-K sentences are then carefully chosen to serve as our extractive summary after we organize these sentences according to importance score.

### 4.2.2 Outline Generation Stage

The first stage of the pipeline of extracting information may produce output that is incoherent and in the reference text language. Therefore, the second stage is required to create coherent output. We experimented with two different multi-lingual natural language generation models, mBART-large [46] and mT5-base [79] and compared the performance of both the models. Since the input for stage 2 is reference text, which can have varying styles and subjects, it is important to ensure that the generated outline is compatible with the source text. To achieve this we took inspiration from [51] and added 2 reward functions to the generation model in a reinforcement learning setting to ensure that the generated outline meets the desired criteria.

1. **Section-title compatibility reward**: We train an XLM-RoBERTa [14] model to classify positive and negative samples for the Section-Title compatibility reward. For the training dataset for this model, we create positive samples by extracting sections and their section titles from Wikipedia articles. To generate the negative samples, we randomly sample 100 sections which do not have the same section title as the current title. Then, we compute the similarity of the sample sections against the current section to measure semantic similarity. We then select the two most similar sections and pair them with the

current title to create the negative sample. The model is trained on this data for Binary Classification to judge whether or not a given title belongs to a section.

2. **Entity Correctness Reward**: We then create an Entity Correctness Reward, which looks for hallucinations in the generated title and rewards the model accordingly. We use IndicNER [54] in inference more to extract the named entities from the generated title and the input reference sentences. Once we get the named entities from the two texts, we decide to reward based on the entities covered in the title set and reference set.

## 4.3 Experiments

### 4.3.1 Training Configuration

For every stage of our method, different computational resources are needed. The extraction stage was carried out on a system that had one NVIDIA 2080Ti GPU and 12GB of GPU RAM. On the other hand, we fine-tuned the model for the outline generation stage using a system with an NVIDIA V100 and 32GB of GPU RAM in multi-GPU setting. This device has PyTorch 1.7.1 and CUDA 11.0 installed.

We used the multi-lingual BERT (mBERT [17]) model to get phrase representations for building the graph in order to perform the extractive summarization step based on HipoRank. This model's maximum input length was similarly set to 512. We set a cap of 50 maximum phrases per sample for the extraction stage output.

The mBART [46] and mT5 [79] models were fine-tuned for 20 epochs with a batch size of 16 during the outline generation stage. We used checkpoints from the facebook/mbart-large-50 and google/mt5-base models from huggingface to initialize the models. For all of our studies, we limited the maximum input length of 512 and output length to 32. A learning rate of 1e-5 was utilized using the AdamW optimizer. Finally, greedy decoding was utilized to create the outline.

### 4.3.2 Metrics

We used standard Natural Language Generation (NLG) metric ROUGE-L [42] to evaluate our system performance.

#### 4.3.2.1 ROUGE-L

ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation, Longest Common Subsequence) is a statistic for assessing the effectiveness of natural language generating systems. The metric is based on the longest common subsequence (LCS), which is a group of words that appear in the same order in both the generated text and the reference text. In order to determine

the ratio of the length of the LCS to the length of the reference text, ROUGE-L first computes the LCS between the computer-generated text and the reference text. The metric analyzes lengthier word sequences that exist in the same order in both the generated and reference text, as well as the sentence structure.

## 4.4 Results

We performed out experiment on XWikiRef which has ∼69K articles. We divide the dataset into train, val and test splits in 70:10:20 ratio and trained our pipeline in a multi-lingual - multi-domain setting. We experimented with both mBART-large and mT5-base models and achieve an average ROUGE-L score of 42.3 and 46.0 respectively. Table 4.1 and 4.2 show results on our initial experiments using mBART and mT5 respectively. Lastly, in Table [4.3, 4.4], we present some instances of our top model's outputs to qualitatively assess its performance.

| **Domain/Lang** | **bn** | **en** | **hi** | **ml** | **mr** | **or** | **pa** | **ta** | **Average** |
|---|---|---|---|---|---|---|---|---|---|
| **books** | 36.1 | 48.7 | 91.4 | 28.8 | 35.5 | 68.6 | 3.2 | 53.0 | **45.7** |
| **films** | 53.6 | 55.6 | 48.2 | 65.0 | 40.0 | 74.1 | 8.3 | 56.8 | **50.2** |
| **politicians** | 57.6 | 26.7 | 74.6 | 38.5 | 40.2 | 60.1 | 10.5 | 40.8 | **43.6** |
| **sportsmen** | 32.1 | 50.9 | 58.0 | 31.8 | 57.7 | 20.4 | 35.0 | 29.8 | **39.5** |
| **writers** | 31.9 | 30.1 | 53.4 | 21.7 | 42.6 | 50.4 | 12.2 | 19.7 | **32.8** |
| **Average** | **42.3** | **42.4** | **65.1** | **37.2** | **43.2** | **54.7** | **13.8** | **40.0** | **42.3** |

Table 4.1: XOutlineGen results on HipoRank + mBART methodology

| **Domain/Lang** | **bn** | **en** | **hi** | **ml** | **mr** | **or** | **pa** | **ta** | **Average** |
|---|---|---|---|---|---|---|---|---|---|
| **books** | 43.0 | 43.8 | 91.3 | 33.7 | 29.3 | 53.7 | 19.0 | 52.9 | **45.8** |
| **films** | 64.1 | 52.9 | 51.0 | 70.0 | 43.9 | 76.7 | 27.7 | 59.9 | **55.8** |
| **politicians** | 64.2 | 34.6 | 73.0 | 45.8 | 36.5 | 58.1 | 26.1 | 43.3 | **47.7** |
| **sportsmen** | 47.2 | 52.6 | 59.3 | 32.9 | 54.1 | 16.8 | 45.7 | 31.4 | **42.5** |
| **writers** | 40.5 | 37.0 | 54.3 | 28.9 | 45.0 | 43.0 | 34.4 | 21.2 | **38.0** |
| **Average** | **51.8** | **44.2** | **65.8** | **42.2** | **41.8** | **49.7** | **30.6** | **41.7** | **46.0** |

Table 4.2: XOutlineGen results on HipoRank + mT5 methodology

| Domain | Reference URL's | Entity | Lang | Reference Outline | Generated Outline |
|---|---|---|---|---|---|
| films | • https://catalog.afi.com/Catalog/moviedetails/59605<br>• https://boxofficemojo.com/movies/?id=mrsdoubtfire.htm<br>• https://www.bustle.com/articles/9597-13-facts-you-didnt-know-about-mrs-doubtfire<br>..... | Mrs. Doubtfire | en | 1. Introduction<br>2. Release<br>3. Production<br>4. Reception | 1. Introduction<br>2. Production<br>3. Reception |
| politicians | • http://www.newindianexpress.com/states/karnataka/2018/oct/08/d-k-shivakumar-stepping-on-many-toes-upsets-congress-brass-in-karnataka-1882541.html<br>• https://www.thehindu.com/news/national/karnataka/shivakumars-father-passes-away/article5523618.ece<br>• http://www.thehindu.com/news/cities/bangalore/it-raids-at-karnataka-ministers-house-resort-housing-gujarat-congress-mlas/article19406787.ece<br>..... | डी. के. शिवकुमार | hi | 1. परिचय<br>2. विवाद और भ्रष्टाचार के आरोप<br>3. राजनीतिक कैरियर | 1. परिचय<br>2. विवाद<br>3. राजनीतिक कैरियर |

Table 4.3: Some examples of XOutlineGen using our best model.

| Domain | Reference URL's | Entity | Lang | Reference Outline | Generated Outline |
|---|---|---|---|---|---|
| sportsmen | • https://www.cricbuzz.com/cricket-news/120636/vivo-to-transfer-ipl-title-rights-to-tata<br>• https://www.iplt20.com/news/3724/bcci-announces-schedule-for-tata-ipl-2022<br>• https://island.lk/await-ten-team-ipl-in-2021/<br>..... | २०२२ इंडियन प्रीमियर लीग | mr | 1. परिचय<br>2. पार्श्वभूमी<br>3. खेळाडू बदल<br>4. मैदाने<br>5. आकडेवारी | 1. परिचय<br>2. पार्श्वभूमी<br>3. खेळाची शैली<br>4. यजमान आणि कार्यक्रम |
| writers | • http://www.webcitation.org/72wTQv4PP<br>• http://www.webcitation.org/72wTHXiGD<br>• https://sambadenglish.com/former-mla-ratnamali-jema-passes-away/<br>..... | ପ୍ରତିଜ୍ଞା ଦେବୀ | or | 1. ପରିଚୟ<br>2. ପ୍ରାରମ୍ଭିକ ଜୀବନ<br>3. ଓଡ଼ିଆ ସାହିତ୍ୟରେ ଅବଦାନ | 1. ପରିଚୟ<br>2. ପ୍ରାରମ୍ଭିକ ଜୀବନ<br>3. ଜନ୍ମ, ପରିବାର ଓ ଶିକ୍ଷା |
| books | • https://archive.org/details/godofsmallthings0000roya_j3i0<br>• https://www.nytimes.com/books/97/05/25/reviews/970525.25truaxt.html<br>• http://news.bbc.co.uk/2/hi/uk_news/179131.stm<br>..... | দ্য গড অব স্মল থিংস | bn | 1. ভূমিকা<br>2. চরিত্রাবলি<br>3. মূল্যায়ন | 1. ভূমিকা<br>2. চরিত্রাবলি<br>3. সম্মান এবং পুরস্কার |

Table 4.4: Some more examples of XOutlineGen using our best model.

## 4.5 Summary

With this work, we presented a pipeline of outline generation on XWikiRef using HipoRank + mBART/mT5 models. We got the best results from HipoRank + mT5 combination. This was a preliminary work which we did before actually generating the encyclopedic content. A lot of work can be done in this domain to generate more concrete outlines by experimenting with different methodologies. We hope that our work will help the community generate more useful outlines of articles, aiding in creation of Wikipedia articles in LR languages.

*Chapter 5*

# Question-Answering with graph augmentation and pruning techniques

## 5.1 Overview

The domain of question-answering (QA) has expanded a lot with the integration of language models and knowledge graphs, bringing about a new frontier in commonsense question-answering. While traditional QA systems excel at factual inquiries, answering questions requiring common sense reasoning has posed a significant challenge. However, with the combined power of language models and knowledge graphs, a new era of intelligent QA systems has emerged. Language models possess the ability to understand and generate human-like text, while knowledge graphs capture structured representations of common knowledge and relationships. By leveraging these two together, researchers and developers have made significant strides in addressing commonsense reasoning in QA. This fusion enables systems to tap into vast amounts of contextual information, understanding nuanced relationships between entities and leveraging commonsense reasoning to provide accurate and coherent answers to complex questions. Therefore, in this work, we are proposing two modifications to the existing work [80]. We call over methodology GrapeQA which is discussed in the next sections.

Our emphasis with GrapeQA lies in improving the working graph (WG) with two simple ideas. (i) We augment the WG with useful information from the question-answer pair reducing the burden on a single QA context node used in previous works. (ii) Instead of keeping all nodes of the WG, or simply scoring relevance, we drop less relevant information (nodes) from the WG simplifying the graph reasoning process. The improvements to the WG are combined with the reasoning process of QA-GNN [80] and evaluated on three datasets, where we see especially large improvements on domain-specific OpenBookQA dataset.

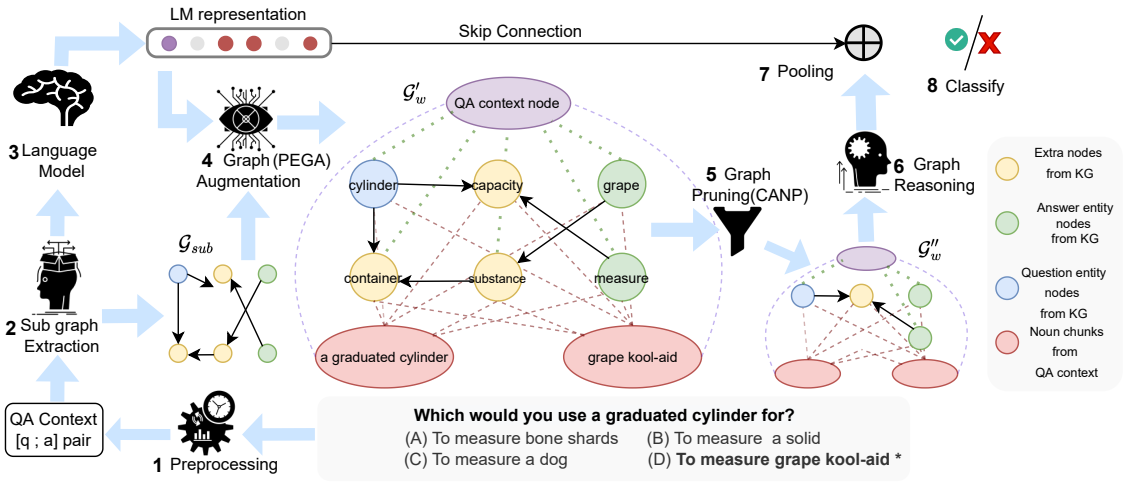


Figure 5.1: Method overview showing the approach to score the question with each answer option. GrapeQA improves QA-GNN [80] by augmenting the Working Graph with additional nodes that capture information from the QA pair (step 4: PEGA) and then pruning the graph to remove the least relevant nodes (step 5: CANP).

## 5.2 GrapeQA Methodology

We briefly describe the QA-GNN approach before our graph augmentation and pruning strategies.

### 5.2.1 LM + KG: QA-GNN as a case study

The objective of QA-GNN [80] is to use both LM and KG for commonsense QA tasks. Each multiple-choice QA consists of a question $q$ and $O$ answer options $\{a_o\}_{o=1}^{O}$ where only one is correct. We create one Working Graph (WG) per answer option and reason over the graph to produce a score. During training, cross-entropy loss is applied to scores of all answer options while we pick the highest scoring answer for inference.

We discuss the WG creation process starting with the KG. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the KG with $\mathcal{V}$ nodes and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ with $\mathcal{R}$ relation types. For a given question-answer pair $[q; a_o]$, all nodes in the KG may not be relevant. Hence, *Question / Answer entity nodes*, referred as $q_{\text{KG}}$ or $a_{\text{KG}}$, that have some text matching with the question $q$ or answer option $a_o$ are picked. Indirect relations between Question and Answer entity nodes are captured through common neighbors (2-hop away) by including them as *Extra nodes* $s_{\text{KG}}$. The sub-graph $\mathcal{G}_{\text{sub}}$ is formed together with the edges in $\mathcal{E}$ that connect the chosen KG nodes. In summary, the nodes of the sub-graph are $\{q_{\text{KG}}\} \cup \{a_{\text{KG}}\} \cup \{s_{\text{KG}}\}$.

Next, a relevance scoring mechanism is used to prune irrelevant nodes that may appear in the sub-graph. Scores are computed by encoding the *QA context* (concatenated question and

answer option text) and node label using an LM followed by a linear projection. The relevance score influences the node representation in the sub-graph. Finally, to create the Working Graph $\mathcal{G}_w$, *QA context* is added as a node to the sub-graph and connected with other nodes using a new edge type.

Question, Answer, and Extra nodes in $\mathcal{G}_w$ are initialized by creating sentences based on triplets from the KG, feeding them to a pretrained LM, and average pooling over relevant tokens (see [21] for details). The QA context node is initialized as $\mathbf{z}$, an encoding of the $[q; a_o]$ text using an LM. To perform reasoning, a relation type aware Graph Network is adopted. The output representations for all nodes are pooled and added to the LM's original encoding of the QA context. Finally, an MLP is used to predict a score for the correctness of the answer option. Fig. 5.1 illustrates QA-GNN along with our proposed modifications.

### 5.2.2 Graph Augmentation and Pruning

GrapeQA proposes two improvements to the WG and corresponding adaptations to QA-GNN. We overcome the limited capacity of the WG to exchange useful information between the QA context and the KG with Prominent Entities for Graph Augmentation (PEGA) that introduces additional nodes from the QA pair to the WG. We also propose QA-Context-Aware Node Pruning (CANP), a pruning method that removes least relevant nodes.

#### 5.2.2.1 Prominent Entities for Graph Augmentation (PEGA)

Graph augmentation begins by extracting noun phrase chunks $c$ from the question and answer pair $[q; a_o]$. We use spaCy's [28] *noun* chunk extractor $f_{\text{ext}}$ to obtain

$$\mathcal{V}' = \{c \mid c \in f_{\text{ext}}([q; a])\}\,. \tag{5.1}$$

The QA context is fed as input to the LM and representations of all the sub-word tokens are obtained. Each extracted noun phrase is represented by averaging over the embeddings of its sub-word tokens. As part of augmentation, these *noun chunks nodes* ($\mathcal{V}'$) are added as new nodes of type $n$ to the working graph $\mathcal{G}_w$. *Noun chunk nodes* also have two types of edges: $r_{no}$ between all the new ($\mathcal{V}'$) and old $\mathcal{G}_w$ nodes, and $r_{nn}$ among the noun chunks themselves resulting in an augmented WG, $\mathcal{G}'_w$:

$$\mathcal{E}' = \{\mathcal{V}' \times r_{nn} \times \mathcal{V}'\} \cup \{\mathcal{V}' \times r_{no} \times \mathcal{V}\}\,, \tag{5.2}$$

$$\mathcal{G}'_w = (\mathcal{V} \cup \mathcal{V}', \mathcal{E} \cup \mathcal{E}')\,. \tag{5.3}$$

#### 5.2.2.2 QA Context-Aware Node Pruning (CANP)

CANP aims to remove the less relevant nodes from the WG. Our intuition is that some Extra nodes (i.e. 2-hop neighbors from the KG which do not match the QA text) may be less relevant to the QA as compared to the Question / Answer entity nodes.

To perform pruning, we first associate and cluster Extra nodes with *Answer entity nodes*. CANP is only applied when there are more than one Answer entity nodes. Recall that the WG is created for one answer option (or one QA pair) and the number of Answer entity nodes (and clusters) depends on the number of nodes with text similar to the answer option in the KG. Similar to relevance scoring in QA-GNN, we calculate the relevance score for each Extra node $s_{\text{KG}}$ against each Answer entity $a_{\text{KG}}$ by encoding the concatenated text of the QA pair, the Answer entity, and the Extra node.

$$\psi_{sa}^{\text{KG}} = f_{\text{head}}\left(\text{LM}\left([\text{text}(\mathbf{z}); \text{text}(\mathbf{a}_{\text{KG}}); \text{text}(\backslash)]\right)\right), \tag{5.4}$$

where text$(\cdot)$ corresponds to the node's label text: $[q; a_o]$ pair for $\mathbf{z}$, Extra node's label $s_{\text{KG}}$ for \, and the Answer entity label $a_{\text{KG}}$ for $\mathbf{a}_{\text{KG}}$. Thus, each Extra node $s_{\text{KG}}$ is assigned to the cluster $\mathcal{V}_x$ corresponding to the highest relevance score,

$$\mathcal{V}_x = \{s_{\text{KG}} \,|\, x = \arg\max_{a_{\text{KG}}} \psi_{sa}^{\text{KG}}\}. \tag{5.5}$$

We compute the average relevance score for each cluster and identify the least relevant cluster $\mathcal{V}_r$ as

$$\psi_x^{\text{KG}} = \sum_{s_{\text{KG}} \in \mathcal{V}_x} \psi_{sx}^{\text{KG}} / |\mathcal{V}_x|, \tag{5.6}$$

$$\mathcal{V}_r = \mathcal{V}_x \text{ s.t. } r = \arg\min_x \psi_x^{\text{KG}}. \tag{5.7}$$

Finally, we remove the cluster with lowest average relevance score from the WG before continuing with graph-based reasoning. The PEGA augmented WG can be pruned as

$$\mathcal{G}''_w = (\mathcal{V} \cup \mathcal{V}' - \mathcal{V}_r, \mathcal{E} \cup \mathcal{E}' - \{\mathcal{V}_r \times R \times \mathcal{V}\}). \tag{5.8}$$

## 5.3 Experiments

### 5.3.1 Datasets

We evaluate GrapeQA on three QA datasets:

1. **CommonsenseQA (CSQA)**: CSQA is 5-way multiple-choice QA (MC-QA) dataset of 12,102 questions that requires commonsense reasoning to answer questions. We use standard splits [41] and report results on the in-house test (IHtest).

2. **OpenBookQA (OBQA)**: OBQA is a 4-way multiple choice-QA dataset of 5,957 questions based on elementary science knowledge; splits by [55].

3. **MedQA-USMLE**: MedQA is a 4-way multiple choice-QA dataset based on biomedical and clinical knowledge and has 12,723 questions from United States Medical License Exams, with splits by [30].

## 5.4 Working Graph Statistics

Given a question and corresponding answer option, KG nodes with matching text entities are identified. These matched nodes along with the Extra nodes that fall in 2-hop paths from them form the sub-graphs for each $[q; a]$ pair. Working Graphs are constructed by joining these sub-graphs with QA context nodes initialized with the representation from LM. In each Working Graph, the QA context node is connected to all the concept nodes in it which are extracted from the KG.

### 5.4.1 Node counts

Table 5.1 shows the number of nodes added to the WG on average. We see that general KGs (ConceptNet) afford a large number of extra nodes (100+) while MedQA with a smaller KG only adds a few extra nodes (∼20). The large number of noun chunks added in the MedQA is explained by the fact that the questions in MedQA are very large as they include patient's description. Table 5.2 presents the average number of words in the question and answer option.

| **Node Type** | **OBQA** | **CSQA** | **MedQA** |
|---|---|---|---|
| Question entity $q_{\text{KG}}$ | 6.52 | 7.36 | 6.1 |
| Answer entity $a_{\text{KG}}$ | 2.79 | 2.05 | 0.55 |
| Extra nodes $s_{\text{KG}}$ | 107.17 | 112.04 | 20.82 |
| Noun chunk nodes $\mathcal{V}'$ | 3.88 | 4.13 | 33.46 |

Table 5.1: Average number of nodes of each type in WGs.

| | **Question** | **Answer** |
|---|---|---|
| OBQA | 13.5 | 2.8 |
| CSQA | 13.8 | 1.5 |
| MedQA | 116.2 | 3.6 |

Table 5.2: Average number of words in the question $q$ and answer option $a_o$ for the different datasets.

### 5.4.2 Noun chunks are unique

Table 5.3 shows the number of unique nodes present in each dataset. It can be observed that the total number of *unique* nodes selected from the KG is low as compared to the total number of unique noun chunk nodes extracted. Even though Table 5.1 shows that a large number of nodes are added to the graph, they are not all unique. Thus, even if the average number of noun chunk nodes for each WG are low, they are more diverse compared to nodes from KG. A small overlap between noun chunk nodes and nodes from the KG indicates that this way of constructing the WG may provide better opportunity for graph reasoning to exchange information effectively between the QA (LM) and the KG.

| Dataset | Noun chunk nodes | Nodes from KG | Overlapping nodes |
|---|---|---|---|
| OBQA | 14470 | 7506 | 1958 |
| CSQA | 23881 | 12485 | 4023 |
| MedQA | 69370 | 2753 | 1268 |

Table 5.3: Number of unique nodes across all WG of the dataset. Even though more nodes are added from the KG on average (see Table 5.1), they are not all unique across the dataset and result in a smaller count.

### 5.4.3 Implementation & training details.

The LM adopted in our work is RoBERTa-large [47] for CSQA and OBQA, and SapBERT [43] for MedQA. ConceptNet [71] is our KG for generating the WG in CSQA and OBQA. For MedQA, we use the graph constructed by QA-GNN [80]. Our model consists of an LM and a GNN with dim 200. RADAM [44] optimizer is used with a learning rate of $10^{-5}$ for the LM and $10^{-3}$ for the GNN. OBQA & MedQA are trained for 50 epochs with a batch size of 128 and CSQA for 20 epochs with a batch size of 64. All models are a single run trained on 2 RTX 2080 Ti GPUs and take about 28 hours for OBQA and 16 hours for CSQA and MedQA.

### 5.4.4 Comparisons with Baselines

We use accuracy as a metric and compare our results primarily against other works that also adopt LM + KG methods (see Table 5.4). GrapeQA builds on top of QA-GNN (for direct comparison) and improving the WG results in highest performance on OBQA & MedQA and comparable performance on CSQA. For a fair comparison, we use the same LM for all methods unless noted.

| Model | OBQA Test | CSQA IHTest |
|---|---|---|
| RGCN [68] | 62.45 | 68.4 |
| GconAttn [77] | 64.75 | 68.6 |
| RN [65] | 65.20 | 69.1 |
| MHGRN [21] | 66.85 | 71.1 |
| GreaseLM (AristoRoBERTa) [83] | <u>84.8</u> | <u>74.05</u> |
| QA-GNN (RoBERTa-large ) [80] | 67.80 | 73.4 |
| GrapeQA: CANP (**Ours**) | 66.20 | **74.94** |
| GrapeQA: PEGA (**Ours**) | 82.0 | 73.41 |
| GrapeQA: PEGA+CANP (**Ours**) | **90.0** | 74.05 |

| Model | MedQA Test |
|---|---|
| BERT-base [18] | 34.3 |
| BioBERT-base [38] | 34.1 |
| RoBERTa-large [47] | 35.0 |
| BioBERT-large [38] | 36.7 |
| SapBERT [43] | 37.2 |
| GreaseLM [83] | <u>38.5</u> |
| QA-GNN [80] | 38.0 |
| GrapeQA: (PEGA) (**Ours**) | **39.51** |

Table 5.4: Comparison of Accuracy between LM+KG methods on the OpenBookQA, CommonsenseQA (left) and MedQA (right).

LM only methods tend to perform worse than the baseline QA-GNN. RoBERTa-large [47] for CSQA provides 72.1% while RoBERTa-large and AristoRoBERTa [13] for OBQA show 64.80% and 77.8%, respectively. For MedQA, the LM only model results are also shown in Table 5.4 (right); we see that LMs trained on medical data (*e.g.* SapBERT [43]) outperform generic LMs on this domain-specific task. GrapeQA outperforms all these approaches.

#### 5.4.4.1 OBQA

CANP applied to the original QA-GNN WG is unable to improve performance (-1.6%), probably because the WG is not rich. However, PEGA provides a 14.2% accuracy improvement over QA-GNN (82% vs. 67.8%). Interestingly, CANP when used together with PEGA boosts the accuracy to 90% (+22.2%); surpassing GreaseLM that uses an improved LM (AristoRoBERTa) and better integration of LM + KG by 5.2%. For the *domain-specific* OBQA, PEGA adds relevant information while CANP effectively cleans up irrelevant nodes resulting in large improvements.

#### 5.4.4.2 MedQA

PEGA achieves an improvement of 1.5% over QA-GNN, and 1% over GreaseLM, the previous SoTA. A reason for the small improvement (compared to OBQA) could be that the WG for

MedQA has fewer nodes (see Table 5.1). Additionally, the small number of Answer entity nodes in the WG also means that CANP is not applicable.

#### 5.4.4.3 CSQA

On *generic commonsense* questions, the WG can have large amounts of irrelevant information that CANP can simplify. We see an improvement of 1.5% over QA-GNN when using CANP only. However, unlike OBQA, PEGA shows comparable performance to QA-GNN as it may lead to stuffing the WG with common terms (noun chunks) that do not provide discriminatory information. Nevertheless, CANP alone also improves over GreaseLM by 0.9% (all in absolute points).

## 5.5 Ablation experiments & additional results

### 5.5.1 Noun chunk extraction.

While PEGA is an effective graph augmentation strategy, it relies on the noun chunk extraction method. We evaluate automatic noun chunk extraction methods spaCy and NLTK against a simple baseline that randomly adds 20% of the QA pair's words to the WG. Table 5.5 shows that extracting meaningful chunks is important and may lead to large performance change (on OBQA). Interestingly, even random chunks of the QA pair provides a 4.5% boost over QA-GNN that only includes one node to encode the entire QA context.

| **Noun chunk extraction method** | **Accuracy** |
|---|---|
| 20% random words | 72.32 |
| NLTK [48] | 78.40 |
| spaCy [28] | **82.00** |

Table 5.5: PEGA Ablations: Impact of different noun chunk extraction methods on OBQA.

### 5.5.2 Number of GNN layers

Tables 5.6 and 5.7 show ablation studies by varying the number of GNN layers over PEGA+CANP and PEGA-only respectively. 5 layer GNNs seem to be a suitable for both methods, while CSQA with PEGA-only shows highest performance with 4 layers.

| | Accuracy | |
|---|---|---|
| **#layers** | **OBQA** | **CSQA** |
| 4 | 88.38 | 72.60 |
| 5 | **90.00** | **74.05** |
| 6 | 88.96 | 71.88 |

Table 5.6: Impact of the number of GNN layers using the PEGA+CANP model.

| | Accuracy | |
|---|---|---|
| **#layers** | **OBQA** | **CSQA** |
| 4 | 83.20 | **74.62** |
| 5 | **82.00** | 73.41 |
| 6 | 81.40 | 73.17 |

Table 5.7: Impact of the number of GNN layers using the PEGA only model.

### 5.5.3 CANP is not necessary on MedQA

Table 5.1 shows the average number of nodes of different types in a WG. The number of extra concept nodes is much higher than the QA concept nodes except in the MedQA dataset. This makes it necessary to prune these nodes to keep only the relevant ones. In case of MedQA since the number of extra nodes in WG are already quite low, and the nodes from the KG are often meaningful (domain-specific) we do not perform CANP pruning.

### 5.5.4 Results on CSQA

Table 5.8 shows the results of our model on the official test set for CommonsenseQA. We compare our results with other existing approaches, both using powerful LMs (e.g., UnifiedQA) or LM+KG methods (QA-GNN, GreaseLM, etc.). Unfortunately we were unable to evaluate our best performing model on the in-house test set (GrapeQA: CANP-only) due to limited number of submissions indicated for the evaluation. Even on the in-house test set, we see no performance change between PEGA-only and QA-GNN (73.41% vs. 73.4%) while a ±1% variation exists due to random seeds.

| Model | Test Acc. |
|---|---|
| RoBERTa [47] | 72.1 |
| RoBERTa + FreeLB (ensemble) [84] | 73.1 |
| RoBERTa + HyKAS [50] | 73.2 |
| RoBERTa + KE (ensemble) | 73.3 |
| RoBERTa+KEDGN (ensemble) | 74.4 |
| XLNet+GraphReason [49] | 75.3 |
| RoBERTa+MHGRN [21] | 75.4 |
| Albert+PG [76] | 75.6 |
| QA-GNN [55] | 76.1 |
| Albert (ensemble) [36] | 76.5 |
| UnifiedQA* [33] | 79.1 |
| GRAPEQA (PEGA) (**Ours**) | **73.5** |

Table 5.8: Comparison on CommonSenseQA official test set using RoBERTa-large model. The best result is in **bold** and second best is underlined. Due to limited entries for evaluation, we were unable to evaluate our best method on CSQA: CANP-only. *UnifiedQA has 11B parameters and is about 30x larger than QA-GNN and our model and is trained on much more data.

## 5.6 Summary

We presented GrapeQA, an effective approach to integrate information from QA (LM) and KG for commonsense QA. We proposed two simple improvements to the working graph: PEGA, a graph augmentation that improves information flow between the QA and the KG; and CANP that prunes less relevant information. Our approach led to new SoTA results on three datasets OBQA, CSQA, and MedQA, with a large 22% increase on OBQA.

*Chapter 6*

# Cross-lingual encyclopedic text generation by utilizing references

## 6.1 Overview

In the context of encyclopedic text generation, there is a crucial need for a cross-lingual multi-document summarization approach specifically tailored for low-resource languages when it comes to Wikipedia-based content. Wikipedia serves as a vast repository of knowledge, encompassing a wide range of topics in numerous languages. However, low-resource languages often face limitations in terms of available linguistic resources and training data. To address this challenge, a cross-lingual multi-document summarization approach becomes invaluable. By leveraging multiple documents and applying summarization techniques that transcend language barriers, we can generate concise and informative summaries in low-resource languages. This approach not only promotes accessibility to knowledge for speakers of these languages but also aids in the preservation and cultivation of their linguistic and cultural heritage. Through a cross-lingual multi-document summarization approach, we can bridge the information gap, empower low-resource language communities, and foster inclusivity in the global knowledge sharing landscape.

With this work, we make the following contributions.

- We motivate and propose the XWikiGen problem which utilizes XWikiRef (Chapter 3) dataset where the input is (set of reference URLs, section title, language) and the output is a text paragraph.

- We model XWikiGen as a multi-document cross-lingual summarization problem and propose a two-stage extractive-abstractive system. Our multi-lingual-multi-domain models using HipoRank (extractive) and mBART (abstractive) lead to the best results.

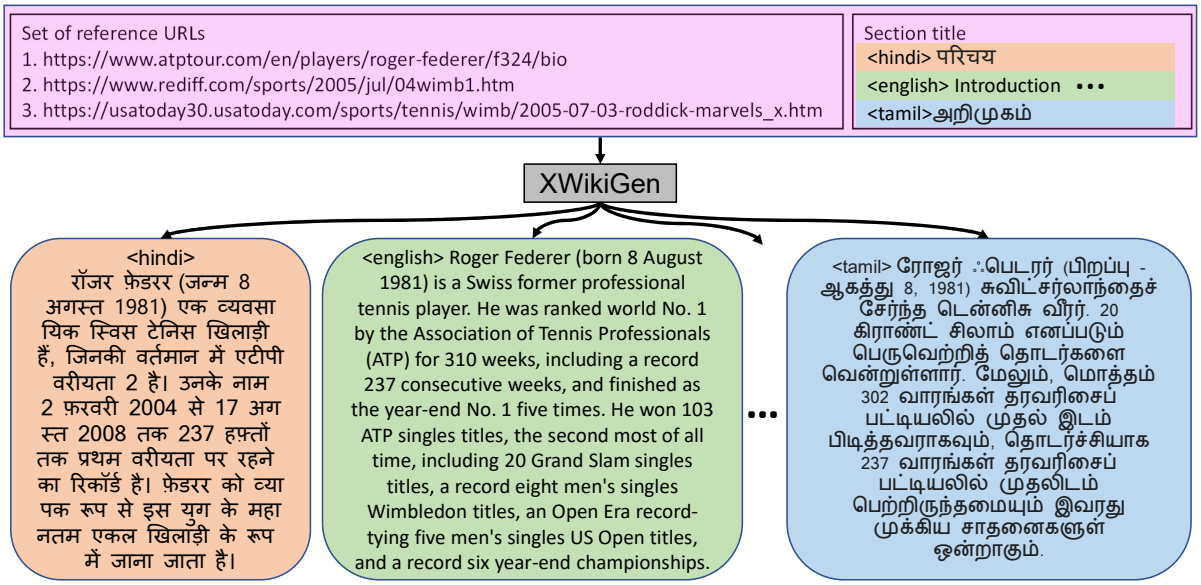


Figure 6.1: XWikiGen examples: Generating Hindi, English, and Tamil text for the Introduction section from cited references.

Figure 6.1 displays an overview of our pipeline XWikiGen, which takes as input a list of reference URLs, the title of the target section, and the target language. The expected result is the text that is appropriate for the specific Wikipedia section in the desired language.

## 6.2 Two-Stage Approach for XWikiGen

In this section, we first explain why a two-stage methodology for the XWikiGen task of cross-lingual multi-document summarization is necessary. The intricate details of the two stages - extractive and abstractive are then covered. Finally, we offer a variety of training configurations.

| **Domain** | **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 6.1: Average number of sentences in references of a section for each domain and language in XWikiRef.

For each domain and language in our dataset, the average number of sentences in a section's references are displayed in Table 6.1. The total amount of text input is rather significant when combined with the average number of references per section as displayed in Table 3.3. It is impossible to provide such lengthy inputs to an encoder-decoder model and expect it to be able to produce appropriate summaries given the quadratic complexity of Transformer-based approaches. Research on the sub-quadratic complexity of transformers is ongoing, with models like the Longformer [7] and Reformer [34] being used. But as part of our ongoing research, we intend to examine them.

We suggest a two-step technique to solve the issue of long inputs, with the first stage identifying promising candidate sentences from all the reference citations for a sample. The candidate sentences with the highest scores are fed into the second stage, which creates an abstractive summary. The two stages will be thoroughly covered in the paragraphs that follow.

### 6.2.1 Extractive Summarization Stage

The extraction stage, given a set of reference URLs, seeks to choose a subset of sentences from these URLs that best summarizes the set of URLs. While lexical chains or position-based methods were used earlier for extractive summarization, neural methods have gained popularity in recent years. Salience and HipoRank are two different extractive summarization-based algorithms that we test. The section title and a list of reference URLs serve as the input for both techniques. Each statement in these reference URLs generates a summary worthiness score using either method.

#### 6.2.1.1 Salience based extractive summarization

Finding the top-K salient sentences from the input references based on each sentence's connection to a certain section title is the fundamental goal of salience-based extractive summarization. The relevance scoring approach used in GrapeQA and QAGNN [81], where a language model was employed to determine the relevance score of each answer entity related to the QA (question-answer) context, served as an inspiration for our salience method. We first divided the reference text into sentences in order to extract the top-K sentences. After that, a section title is appended to each sentence before being provided as input to an XLM-RoBERTa [14] language model. Based on the likelihood from the language model, we assign a score to each sentence. The top-K sentences with the best relevance ratings are output to the following stage.

Keep in mind that we employ a pretrained language model in this approach. Only probe mode is used with the model.

#### 6.2.1.2 HipoRank based extractive summarization

The unsupervised graph-based technique called Hierarchical and Positional Ranking technique (HipoRank) [20] is used to extractively summarize long input text. It generates a directed hierarchical network containing sentence and section nodes, as well as sentence-sentence and sentence-section edges with asymmetrically weighted edges, given a document with many sections. Then, a sentence node's score is calculated using a weighted sum of the edges incident on the node.

Using mBERT [17], we compute sentence node representations. To determine the representation for each section node, we use the mean of all the sentence representations within that section.

The intra-sectional and inter-sectional edges between each phrase node and the other nodes form the network. All of the sentences in a section are connected intra-sectionally, modeling the local significance of each sentence. The main point is that sentences that are comparable to the majority of sentences inside a segment are more significant. Contrarily, inter-sectional connections are made between section nodes and sentences and are designed to simulate the sentences' overall significance. Here, the concept is that the most significant sentences are those that most closely resemble previous sections. For the sake of efficiency, no edges are permitted between sentences that are in distinct parts.

To calculate edge weights, node embeddings are compared using their cosine similarity. Intra-sectional edges are given more weight if they are incident to a boundary sentence based on the supposition that significant sentences are located close to the borders (start or end) of a text. Similar to this, key portions are located close to the document's edges. This theory is applied to properly weigh inter-sectional edges. Finally, a weighted sum of the edges (both intra-sectional and inter-sectional) incident on the sentence node is used to calculate the importance score for the node. The top-K sentences are then carefully chosen to serve as our extractive summary after we organize these sentences according to importance score.

### 6.2.2 Abstractive Summarization Stage

It should be noted that the extractive stage produces output in the language of the reference text, which can result in an incoherent summary due to sentences being taken from multiple documents. To generate a coherent summary in the target language, an abstractive stage is needed. This stage utilizes two advanced multi-lingual natural language generation models - mBART-large [46] and mT5-base [79]. These models are both transformer models with encoder-decoder architecture and have proven to be effective for various natural language processing tasks, such as named entity recognition, question answering, and natural language inference. Both models have 24 layers, consisting of 12 layers for the encoder and 12 layers for the decoder.

We provide the target language ID, article title, section title and the top-k sentences from the extractive stage (sorted by score in descending order) as input to these models.

The mT5 [79] model was trained on the mC4 dataset[1], which consists of web data in 101 different languages and follows a consistent text-to-text format. On the other hand, mBART [46] was trained on the CommonCrawl corpus, where the BART objective was utilized to mask phrases and permute sentences, with a single Transformer model recovering the original text. The mT5-base model has 12 layers for both the encoder and decoder, with 12 heads per layer, a feed-forward size of 2048, 64-dimensional keys and values, $d_{model}$ of 768, and a vocabulary size of 250112. It contains 582.40M parameters in total. The mBART-large-50 model [46], which is also multi-lingual, has the same number of layers for both the encoder and decoder as the mT5-base model. It has 16 heads per layer, a feed-forward size of 4096, $d_{model}$ of 1024, a vocabulary size of 250054, and 610.87M parameters. It should be noted that both models are almost the same size.

We utilized the training portion of our XWikiRef dataset to fine-tune both of these models on the output of the extractive stage.

## 6.3 Multi-lingual, Multi-domain, and Multi-lingual-Multi-domain setups

Our XWikiRef dataset comprises data from five domains and eight languages, offering various training possibilities. For instance, we could train one model for each pair of language and domain, resulting in 40 models that would require management and deployment. However, since the amount of training data for each pair of language and domain is not extensive, such models may not leverage cross-language or cross-domain knowledge effectively.

There are multiple methods for training models, one of which is a multi-lingual approach. This involves training a single model per domain, utilizing training data across all languages, resulting in five models. Another approach is a multi-domain method, in which we train a single model per language using training data across all domains, leading to eight models.

A final approach to training models is a multi-lingual-multi-domain method. This involves combining training data across all languages and domains and training a single model. Such a model is capable of leveraging cross-language and cross-domain information to learn more robust and accurate representations.

Existing research in multi-lingual cross-lingual natural language generation has indicated that multi-lingual models are often superior to individual models, particularly for low-resource languages. Given that this study concentrates on low-resource languages, we conduct experiments utilizing multi-lingual, multi-domain, and multi-lingual-multi-domain methods.

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

## 6.4 Experiments

### 6.4.1 Training Configuration

Our approach requires different computational resources for the two stages. The extractive step was conducted on a machine with one NVIDIA 2080Ti that has a GPU RAM of 12GB. On the other hand, for the abstractive stage, we fine-tuned the model on a machine with NVIDIA V100 that has a GPU RAM of 32GB. This machine is equipped with CUDA 11.0 and PyTorch 1.7.1.

To perform the extractive stage based on salience, we utilized the XLM-RoBERTa-base [14] model, which is capable of extracting sentence representations with a maximum input length of 512. On the other hand, for HipoRank, we employed the multi-lingual BERT (mBERT [17]) model to obtain sentence representations for constructing the graph. The maximum input length for this model was also set to 512. We limited the number of sentences outputted by the extractive stage to a maximum of 50 per sample.

The abstractive stage was performed by fine-tuning the mBART [46] and mT5 [79] models for 20 epochs, using a batch size of 4. We initialized the models with checkpoints from the google/mt5-base and facebook/mbart-large-50 huggingface models. We kept the maximum input and output length to 512 for all our experiments. AdamW optimizer was used with a learning rate of 1e-5. Finally, we used greedy decoding for generating the abstractive summaries.

### 6.4.2 Metrics

We used standard Natural Language Generation (NLG) metrics, including ROUGE-L [42], METEOR [5] and chrF++ [59], to evaluate our models. However, we did not use the PARENT [19] metric because it relies on word overlap between input and output text, which is not applicable to our task XWikiGen since the input and output are in different languages.

#### 6.4.2.1 ROUGE-L

ROUGE-L is a metric used for evaluating the quality of natural language generation systems. It stands for Recall-Oriented Understudy for Gisting Evaluation, Longest Common Subsequence. The metric is based on the longest common subsequence (LCS), which is a sequence of words that appear in the same order in both the generated and reference text. ROUGE-L computes the LCS between the machine-generated text and the reference text, and then calculates the ratio of the length of the LCS to the length of the reference text. The metric takes into account the structure of the sentences and considers longer sequences of words that appear in the same order in both the generated and reference text.

#### 6.4.2.2 chrf++

chrF++ is an extension of the traditional F-score metric that calculates the F-score based on the matching of character n-grams instead of word n-grams. The "+" in the name refers to additional features that are used to account for character position information and handle cases where the number of predicted characters is different from the number of reference characters. The chrF++ metric is often used for evaluating machine translation and summarization systems.

#### 6.4.2.3 METEOR

METEOR, an automated metric evaluation, is based on a generalized idea of unigram matching between the text generated by the machine and the reference text created by a human. It uses various techniques to align the machine-generated text and the reference text, such as stemming, synonyms, paraphrasing, and word order swapping, to improve the match between the two texts. In addition to unigram matching, METEOR also considers precision, recall, and alignment penalties to compute a score that reflects the quality of the generated text.

## 6.5 Results

Table 6.2 shows the results from different experiment settings for XWikiGen. The settings include two extractive approaches (salience and HipoRank), two abstractive approaches (mBART and mT5), three training setups (multi-lingual, multi-domain, and multi-lingual-multi-domain), and three metrics (ROUGE-L, METEOR, and chrF++). The metrics are evaluated on all test examples in XWikiRef and the results are presented as a micro-average.

| | 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 6.2: XWikiGen Results across multiple training setups and (extractive, abstractive) methods on test part of XWikiRef. Best results per block are highlighted in bold. Overall best results are also underlined.

The table shows that the best results are achieved by using the multi-lingual-multi-domain training setup, and in this setup, combining HipoRank with mBART yields the best overall performance. These results are significantly better than those obtained with other models. It is not surprising that the multi-lingual-multi-domain setup performs well since it combines learning across all languages and domains in the dataset. HipoRank was also expected to perform well because it combines the knowledge of the pretrained mBERT model with the hierarchical document structure. The HipoRank+mBART combination still yields the best results even for the multi-lingual setup. On the other hand, when it comes to the multi-domain setup, Salience+mBART performs better. Mann-Whitney U-Test reveals that among the best models in the Multi-lingual-multi-domain block, our best model (Hiporank+mBART) outperforms the second best model (Salience+mBART) with p-values of 5.47e-07 (ROUGE-L), 2.76e-26 (chrF++), and 6.76e-09 (METEOR). Additionally, Hiporank+mBART is superior to the worst model in the block (Salience+mT5) with p-values of 3.16e-143 (ROUGE-L), 1.69e-268 (chrF++), and 8.67e-201 (METEOR). It is worth noting that Salience employs XLM-RoBERTa with 270M parameters, whereas HipoRank uses mBERT with 110M parameters. Furthermore, mT5 and mBART have 580M and 610M parameters, respectively. Finally, the average output length for our best model is 221 words.

| | 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 6.3: Detailed per-language results on test part of XWikiRef, for the best model per training setup.

Additionally, we want to examine the performance of the best models for each training setup in more detail. Therefore, we present micro-averaged metrics per language and per domain for the test set in Tables 6.3 and 6.4 for these three models. Table 6.3 reveals the following: (1) Multi-domain training produces significantly better results than multi-lingual training, except for Tamil (ta). (2) Interestingly, relatively richer languages such as English (en) and Hindi (hi) benefit the most from transitioning from multi-lingual to multi-lingual-multi-domain setup. (3) To elaborate further, we found that there were improvements in most languages when comparing multi-domain training with multi-lingual-multi-domain, but there were losses in the mr and or languages. Additionally, Table 6.4 shows that across all domains, the results improved as we moved from multi-lingual training to multi-domain training and finally to multi-lingual-multi-domain setup, except for a slight drop in the sportsmen domain in the multi-lingual-multi-domain case.

| | 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 6.4: Detailed per-domain results on test part of XWikiRef, for the best model per training setup.

Lastly, we provide a detailed breakdown of the performance of our best model on a per (domain, language) basis in Table [6.5, 6.6, 6.7]. We note that the highest results are achieved for the combination of Hindi and books. In general, the model performs the best for Hindi across all domains. It also performs reasonably well for Marathi and Oriya, but there is room for improvement in Bengali and Malayalam.

| | 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 6.5: 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 6.6: 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 6.7: 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.

To qualitatively evaluate the performance of our top model, we provide some examples of its outputs in Table [6.8, 6.9, 6.10]. Overall, our model generates coherent text up to a certain length. However, as the length of the output increases, we observe repetitive patterns that disrupt sentence structure. This issue of generating repeated n-grams is common in pre-trained language models, and increasing the size of the training dataset has been shown to improve it. Additionally, we note the accuracy of the generated text compared to the reference text. Although the model produces sentences with correct structure, it occasionally predicts incorrect value strings such as birth-dates, personal names, and related entities. This issue of generating incorrect information is a common problem in pre-trained language models, and training on more data could help mitigate it.

## 6.6 Summary

We have identified a need for cross-lingual multi-document summarization to produce Wikipedia content for low-resource languages. To address this, we have created a unique dataset with approximately ∼105K summaries across five domains and eight languages. We also proposed a two-stage system that uses an extractive and abstractive approach, and tested various training setups. Our multi-lingual-multi-domain model using HipoRank (extractive) and mBART (abstractive) performed the best. We have made our dataset and code publicly available for use as a benchmark and encourage further research in this important area.

| 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 மூவீடோன் 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<br>• https://www.icc-cricket.com/media-releases/1212091<br>• 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 6.8: Some examples of XWikiGen using our best model.

| 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<br>• 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/<br>• 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 6.9: More examples of XWikiGen using our best model.

| Domain | Reference URLs | Entity/ Section | Lang | Reference Text | Generated Summary |
|---|---|---|---|---|---|
| Writers | • http://www.columbia.edu/itc/mealac/pritchett/00ambedkar/timeline/graphics/youth.html<br>• http://www.columbia.edu/itc/mealac/pritchett/00ambedkar/timeline/1890s.html<br>• 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 ভীমাবাইর (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 পাস্থের 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 জন্য করতো বিদ্যালয়ের চাপরাশী (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 6.10: More examples of XWikiGen using our best model.

*Chapter 7*

# Conclusion and future work

In this work, we aim to tackle the issue of lack in encyclopedic content for low resource languages especially in context of Indian languages. To mitigate this issue, we proposed a dataset called XWikiRef that can be used to create systems to generate the encyclopedic content in low resource languages. In addition to this, we proposed a framework called XWikiGen that utilizes XWikiRef for automatically generating encyclopedic text in these low resource languages. Apart from generating encyclopedic content, we also utilized this dataset to generate outline (XOutlineGen) of an encyclopedic article in cross-lingual manner. Additionally, we explored the idea of relevance scoring in context of commonsense question answering (GrapeQA) and thereafter studied its effect in unsupervised extractive summarization domain.

We start this thesis by highlighting the lack of encyclopedic content in low resource languages in **Chapter 1** by providing an comparative analysis of number English Wikipedia articles vs low resource Indic languages, number of edits made in these language based Wikipedia etc. This chapter also discusses the overview of dataset and different tasks proposed in this work which are discussed in the next chapters of this thesis.

Before coming to the discussion of proposed dataset and different tasks which we tried to solve in this work, we discuss some of the related works done in the past in **Chapter 2** by discussing some of the relevant datasets in context of encyclopedic text generation. We also discuss some of the works being done in the context of outline generation, commonsense question-answering. Lastly, we discuss some of the works related to short and long text generation in connection to the encyclopedic content.

**Chapter 3** discusses the details of our proposed dataset called XWikiRef which is a cross lingual, multi-document and multi-domain dataset covering 8 languages (Bengali, English, Hindi, Marathi, Malayalam, Odia, Punjabi and Tamil) and 5 domain (Books, Films, Politicians, Sportsmen and Writers). We also provide a detailed analysis of the proposed dataset in this chapter along with comparison with existing related datasets and highlight the issues our dataset tackles which other datasets we not able to tackle.

XOutlineGen in **Chapter 4** proposes a framework to automatically generate outline of encyclopedic article from references in cross-lingual manner. It proposes a 2 stage system where first stage perform unsupervised extractive summarization (HipoRank) to extract top K relevant sentences. The second stage uses these top-K relevant sentences as input along with the article title and target language information and generates the corresponding article outline. The second stage is fine-tuned using mBART/mT5 models by adding RL based reward functions to the system. We found HipoRank + mT5 combination to be the best in our experiment setting.

**Chapter 5** explores the idea of relevant scoring in the domain of commonsense question-answering. In this work, we propose two modification to the existing system viz. *Prominent Entities for Graph Augmentation (PEGA)* and *QA Context-Aware Node Pruning (CANP)* to improve the model performance on the question-answering task. We tested our proposed modifications on 3 datasets viz. CommonSenseQA, OpenBookQA and MedQA. We got the improved results on OpenBookQA and MedQA datasets as compared to baseline methodology (QAGNN) while a comparable performance in CommonSenseQA dataset. Thereafter, we explore this relevance scoring mechanism in our XWikiGen task by utilizing relevance scoring as a technique to perform unsupervised extractive summarization.

Finally, **Chapter 6** discusses about the main work of this thesis which is an pipeline to automatically generate encyclopedic text in low resource languages. We call this pipeline XWikiGen which is essentially a 2 stage cross-lingual, multi-document summarization based approach. It utilizes our XWikiRef dataset in which the system takes the reference text, section title information, and target languages as input and generates the corresponding section specific summary in the desired language. It employs a 2 stage framework where in first stage we experimented with Salience based and extractive summarization and HipoRank. In stage 2, we finetuned mBART and mT5 models and then perform a quantitative analysis of the results. We performed experiments in 3 different settings i.e. multi-lingual, multi-domain and multi-lingual - multi-domain. We found multi-lingual - multi-domain setting and the combination HipoRank + mBART to be the best performing experiment. With these work, we also make our code of XWikiGen and our dataset XWikiRef publicly available so that it can be utilized by the community in this area of research and help in enhancing the encyclopedic content in low resource languages.

## Future Work

Multiple possible future works are possible in the work presented in this thesis like improvement of the results, expansion of dataset etc. Below points discuss these points in details.

1. In our work, we proposed XWikiRef dataset which covers 8 languages and 5 domains. The dataset size can further increased by adding more languages (Indic as well as non-

Indic low resource languages) and domains. This will broaden the scope of automatic encyclopedic text generation to other languages.

2. While generating the encyclopedic content, apart from reference text, other sources of information can also be added and experimented with. Some of them could be Wikidata triples, knowledge graphs etc. Data sources from different modalities can also be tried out so as to cover more diverse set of input.

3. More set of experiments can be performed in outline generation work as the current work utilizes only HipoRank + mBART/mT5 models in RL based setting. Non RL based settings, different rewards can be tried out improve the results more.

4. In GrapeQA, we proposed modifications in sub-graph creation level while no modifications were done on the model part. By trying different new models, various experiments can be done and results can be improved.

5. Finally, in XWikiGen different new techniques can be tried out to improve the results like instruction-tuning, prompt engineering etc. New multilingual LLM's like BLOOM etc. can also be tried to check whether it helps in improving the results or not. Additionally thesis language models suffer with problems like hallucination, repetition etc. To mitigate these issues can also be one of the task that can be tried out in the future.

Overall, there is scope in some areas where this work can be extended and further research in this area can be done. It will help in achieving the ultimate goal i.e. to make the encyclopedic content in low resource languages as rich as resourceful languages.

*Appendix A*

# Challenges in different text based problems and their mitigation

This chapter explores different avenues to solve the various challenges faced in problems involving text as input. There are different tasks that were handled with these different works like text summarization, textual similarity and social media analysis. Below list indicates the problem solved with these works:

1. Indian Language Summarization
2. Profiling irony and stereotype spreaders on Twitter
3. Multilingual News Article Similarity

In this chapter, different types of modelling techniques were explored with these problems. The intrinsic details of these techniques along with their results are explained in the below sections.

## A.1 Indian Language Summarization

Automatic text summarization has a lot of potential applications in the current technological era like summarizing news articles, research articles etc. A lot of work has already been done in summarizing English languages text. But very little work is being done in summarizing Indian Languages. Therefore, summarizing text in these languages apart from English has become an essential task. India has approximately 350 million and 50 million Hindi and Gujarati speakers respectively. So building a summarization model in these languages will play a crucial role for this task. Recently, transformers based models like mBart[46], mT5[79] and IndicBart[16] have gained a lot of attention because of their multilingual capabilities including various Indic Languages.

Summarization can be performed in 2 ways: extractive summarization and abstractive summarization. In extractive summarization, a subset of sentences from the input text is taken as output summary. While in abstractive summarization, the entire summary is generated from scratch with the source text as input. Since text in abstractive summarization, summary is

generated from scratch, this makes it more human like generated text. But at the same time, it becomes more difficult to perform abstractive summarization as compared to extractive summarization.

In this work, we aim to perform abstractive summarization on these languages as a part of the FIRE shared task 2022 - ILSUM [66][67] using the dataset provided by the organizers. We used IndicBART and mT5 models for our experiments. We also performed data augmentation and tested the performance of the models. In the last, we report the ROUGE-1, ROUGE-2, ROUGE-3 and ROUGE-4 scores.

### A.1.1 Experiment Name

This subsection defines the experiment name with their details which are mentioned in the below mentioned tables:

#### A.1.1.1 English Experiments

1. **da_en_mt5**: mT5-small was finetuned in this approach along with data augmentation to 3 times of the actual english data.
2. **da_en_ibart**: IndicBART was finetuned in this approach along with data augmentation to 3 times of the actual english data.
3. **da5_en_ibart**: IndicBART was finetuned in this approach along with data augmentation to 5 times of the actual english data.
4. **en_ibart**: IndicBART was finetuned in this approach on the actual english dataset.
5. **en_mt5**: mt5-small was finetuned in this approach on the actual english dataset.

#### A.1.1.2 Hindi Experiments

1. **da5_hi_ibart**: IndicBART was finetuned in this approach along with data augmentation to 5 times of the actual hindi data.
2. **da_hi_ibart**: IndicBART was finetuned in this approach along with data augmentation to 3 times of the actual hindi data.
3. **da_hi_mt5**: mT5-small was finetuned in this approach along with data augmentation to 3 times of the actual hindi data.
4. **hi_ibart**: IndicBART was finetuned in this approach on the actual hindi dataset.
5. **hi_mt5**: mT5-small was finetuned in this approach on the actual hindi dataset.

#### A.1.1.3 Gujarati Experiments

1. **gu_ibart**: IndicBART was finetuned in this approach on the actual gujarati dataset.
2. **da_gu_ibart**: IndicBART was finetuned in this approach along with data augmentation to 3 times of the actual gujarati data.
3. **da5_gu_ibart**: IndicBART was finetuned in this approach along with data augmentation to 5 times of the actual gujarati data.
4. **gu_mt5**: mT5-small was finetuned in this approach on the actual gujarati dataset.

### A.1.2 Validation set results

Below 3 tables shows results of our experiments on the validation set.

Table A.1: ROUGE F1 scores on English Validation set

| Experiment | ROUGE-1 | ROUGE-2 | ROUGE-3 | ROUGE-4 |
|---|---|---|---|---|
| da_en_mt5 | 0.54 | 0.43 | 0.41 | 0.40 |
| da_en_ibart | 0.51 | 0.38 | 0.36 | 0.35 |
| da5_en_ibart | 0.51 | 0.38 | 0.36 | 0.35 |
| en_ibart | 0.49 | 0.36 | 0.33 | 0.32 |
| en_mt5 | 0.47 | 0.34 | 0.32 | 0.31 |

Table A.2: ROUGE F1 scores on Hindi Validation set

| Experiment | ROUGE-1 | ROUGE-2 | ROUGE-3 | ROUGE-4 |
|---|---|---|---|---|
| da5_hi_ibart | 0.6104 | 0.515 | 0.488 | 0.475 |
| da_hi_ibart | 0.604 | 0.508 | 0.482 | 0.470 |
| da_hi_mt5 | 0.595 | 0.49 | 0.473 | 0.46 |
| hi_ibart | 0.594 | 0.497 | 0.471 | 0.458 |
| hi_mt5 | 0.54 | 0.438 | 0.412 | 0.398 |

Table A.3: ROUGE F1 scores on Gujarati Validation set

| Experiment | ROUGE-1 | ROUGE-2 | ROUGE-3 | ROUGE-4 |
|---|---|---|---|---|
| gu_ibart | 0.246 | 0.146 | 0.118 | 0.105 |
| da_gu_ibart | 0.239 | 0.144 | 0.118 | 0.105 |
| da5_gu_ibart | 0.235 | 0.137 | 0.11 | 0.096 |
| gu_mt5 | 0.206 | 0.114 | 0.09 | 0.079 |

### A.1.3 Test set results

The below 3 tables shows the results of top 3 experiments per language on official test set.

Table A.4: ROUGE F1 scores on English Test set

| Experiment | ROUGE-1 | ROUGE-2 | ROUGE-3 | ROUGE-4 |
|---|---|---|---|---|
| da5_en_ibart | 0.521 | 0.401 | 0.378 | 0.369 |
| da_en_ibart | 0.512 | 0.389 | 0.366 | 0.358 |
| en_ibart | 0.493 | 0.367 | 0.344 | 0.336 |

Table A.5: ROUGE F1 scores on Hindi Test set

| Experiment | ROUGE-1 | ROUGE-2 | ROUGE-3 | ROUGE-4 |
|---|---|---|---|---|
| da5_hi_ibart | 0.592 | 0.491 | 0.464 | 0.451 |
| da_hi_ibart | 0.586 | 0.485 | 0.458 | 0.445 |
| hi_mt5 | 0.544 | 0.438 | 0.41 | 0.397 |

### A.1.4 Analysis

From the above results, we can say that data augmentation is a useful step as it has shown significant improvement of results over other experiments. Also, on comparing IndicBART and mT5, we can say that IndicBART performed better in most of the cases than mT5 for the

Table A.6: ROUGE F1 scores on Gujarati Test set

| Experiment | ROUGE-1 | ROUGE-2 | ROUGE-3 | ROUGE-4 |
|---|---|---|---|---|
| da5_gu_ibart | 0.242 | 0.146 | 0.119 | 0.106 |
| da_gu_ibart | 0.241 | 0.145 | 0.120 | 0.107 |
| gu_mt5 | 0.203 | 0.115 | 0.094 | 0.084 |

summarization task. Further improvement can be made by using larger models like mbart-large or mt5-base/mt5-large models.

## A.2 Profiling irony and stereotype spreaders on Twitter

Metaphorical and figurative style of writing presents a subtle way of communicating across a message on social media. The nature of the message being conveyed can be distinguished based on the use of such linguistic nuances in a message being propagated. Their usage in directed ways can make the message to be either generally harmless, potentially hurtful, or even inherently toxic in nature. The identification of such content is beneficial not only for shielding often targeted demographic groups, but also for reasons such as a better understanding of the textual content on social media. As pointed out in [72], a better understanding of sarcasm and irony in text can help improve sentiment analysis because of the difficulty in semantic understanding of text introduced by sarcasm. Apart from the understanding of sarcastic content, a profiling of users who tend to propagate such content can benefit to understand differences in the patterns of sarcasm originating from different sources. It can also ease tracking users indulging in the spread of toxic content through subtle means.

In our work, we focus on the task of profiling spreaders of ironical and stereotypical content on Twitter, as a part of the PAN [8] shared task [61] in CLEF 2022. In this task, we work on a Twitter feed of a set of users in English containing user-level annotations to indicate if the user is a spreader of ironical and stereotypical content. In our implementation of the solution, we treat all the user tweets as a single input and experiment with basic text pre-processing followed by a simple TF-IDF representation. This essentially models the task with simple term-frequency based information from past tweets. Inspite of the simplicity of modeling, however, experiments with simple, lightweight machine learning models gave encouraging results on this task.

Table A.7: Results obtained by different machine learning models

| Model | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|
| Logistic Regression | 87 | 85 | 89 | 82 |
| K Neighbors Classifier | 77 | 75 | 78 | 72 |
| SVM | 88 | 87 | 89 | 85 |
| **Random Forest Classifier** | **90** | **89** | **91** | 87 |
| XGBoost Classifier | 89 | **89** | 86 | **92** |

### A.2.1 Results

Table A.7 shows results of experiments performed using different models. Based on our ML-based experiments on term frequency representation of user tweets, we were able to achieve a respectable performance which was consistent across datasets used for validation and testing. Hence, if the dataset distribution used for the task matches with the data encountered in actual, in-the-wild tweets, a user-profiling with system good performance can be achieved with minimalistic lightweight techniques.

## A.3 Multilingual News Article Similarity

The objective of the multilingual news article similarity task is to determine how similar a given pair of news articles are, regardless of the language they are written in. This task focuses on assessing the similarity based on the entities and events discussed in the articles, rather than subjective aspects of the language used. We chose to utilize the encoder representations from models that have demonstrated superior performance in various natural language processing (NLP) tasks across different languages, as this task does not specifically target a particular set of languages. To model the similarity task using these representations, we employed a Siamese architecture as the underlying framework. Throughout our experiments, we explored different aspects, such as the features provided to the encoder model, data augmentation, and ensembling techniques. Among these experiments, we found that data augmentation yielded the most effective results. We employed Multilingual DistilBERT (DB) [64] and XLM-RoBERTa (XLM) [15] pretrained cross-lingual encoder representations. Along with the MSE value, evaluation was conducted using the Pearson Correlation Coefficient (PCC) and Mean Absolute Percentage Error (MAPE) metrics.

| Experiment | Validation set | | |
|---|---|---|---|
| | **PCC** | **MAPE** | **MSE** |
| XLM, only text | 0.53 | 0.39 | 0.98 |
| DB, only text | 0.55 | 0.41 | 0.93 |
| XLM, only metadata | 0.46 | 0.47 | 1.03 |
| XLM, metadata with text | 0.52 | 0.41 | 0.94 |
| XLM, extracted named entities with metadata | 0.45 | 0.43 | 1.05 |
| DB, extracted named entities with metadata | 0.47 | 0.43 | 1.04 |
| XLM, with data augmentation | 0.54 | 0.37 | 0.99 |
| **DB, with data augmentation** | **0.58** | **0.41** | **0.94** |

Table A.8: Results of the experiments performed on validation set

| Experiment | Test PCC |
|---|---|
| DB, data augmentation 3 times | 0.436 |
| **DB, data augmentation 4 times** | **0.441** |

Table A.9: PCC for the experiments performed on test set.

### A.3.1 Results

Table A.9 shows the final result on test set while table A.8 shows results of different experiments performed on test set.

As seen in the table A.8, the most effective strategy in our experiments, with the best performance across metrics, was data augmentation. In similar trials, DistilBERT regularly beat the XLM-RoBERTa equivalent when comparing the multilingual encoders employed. Named entities, metadata, and other derived features were not as helpful for the task as the simple original news text piece.

## 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.
- *__Dhaval Taunk__, *Lakshya Khanna, *Siri Venkata Pavan Kumar Kandru, Vasudeva Varma, Charu Sharma, and Makarand Tapaswi. **GrapeQA: GRaph Augmentation and Pruning to Enhance Question-Answering**. In Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion). Association for Computing Machinery, New York, NY, USA, 1138-1144.
- 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.

## Other Publications

- **Dhaval Taunk**, and Vasudeva Varma. **Summarizing Indian Languages using Multilingual Transformers based Models**. In Forum for Information Retrieval Evaluation (FIRE), December 9-13, 2022, India, 2022.
- **Dhaval Taunk**, Sagar Joshi, and Vasudeva Varma. **Profiling irony and stereotype spreaders on Twitter based on term frequency in tweets**. In Conference and Labs of the Evaluation Forum (CLEF) 2022.
- *__Sagar Joshi__, *Dhaval Taunk, and Vasudeva Varma. **IIIT-MLNS at SemEval-2022 Task 8: Siamese Architecture for Modeling Multilingual News Similarity.** In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1145-1150, 2022.

# Bibliography


[1] T. Abhishek, S. Sagare, B. Singh, A. Sharma, M. Gupta, and V. Varma. Xalign: Cross-lingual fact-to-text alignment and generation for low-resource languages. In *The World Wide Web Conference*, pages 171–175, 2022.

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

[3] S. Arnold, R. Schneider, P. Cudré-Mauroux, F. A. Gers, and A. Löser. SECTOR: A neural model for coherent topic segmentation and classification. *Transactions of the Association for Computational Linguistics*, 7:169–184, 2019.

[4] M. J. B. I. au2, D. M. Katz, and E. M. Detterman. Openedgar: Open source software for sec edgar analysis, 2018.

[5] S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[6] J. Barrow, R. Jain, V. Morariu, V. Manjunatha, D. Oard, and P. Resnik. A joint model for document segmentation and segment labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 313–322, Online, July 2020. Association for Computational Linguistics.

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

[8] J. Bevendorff, B. Chulvi, E. Fersini, A. Heini, M. Kestemont, K. Kredens, M. Mayerl, R. Ortega-Bueno, P. Pezik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, and E. Zangerle. Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, and Style Change Detection. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Thirteenth International Conference of the CLEF Association (CLEF 2022)*, volume 13390 of *Lecture Notes in Computer Science*. Springer, 2022.

[9] S. M. Bhatt, S. Agarwal, O. Gurjar, M. Gupta, and M. Shrivastava. Tourismnlg: A multi-lingual generative benchmark for the tourism domain. In *ECIR*, page To appear, 2023.

[10] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD International Conference on Management of Data*, pages 1247–1250, 2008.

[11] 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.

[12] Z. Chi, L. Dong, F. Wei, W. Wang, X.-L. Mao, and H. Huang. Cross-lingual natural language generation via pre-training. In *AAAI*, volume 34, pages 7570–7577, 2020.

[13] P. Clark, O. Etzioni, D. Khashabi, T. Khot, B. D. Mishra, K. Richardson, A. Sabharwal, C. Schoenick, O. Tafjord, N. Tandon, S. Bhakthavatsalam, D. Groeneveld, M. Guerquin, and M. Schmitz. From 'F' to 'A' on the N.Y. Regents Science Exams: An Overview of the Aristo Project. *arXiv:1909.01958*, 2019.

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

[15] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale, 2020.

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

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

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019.

[19] B. Dhingra, M. Faruqui, A. Parikh, M.-W. Chang, D. Das, and W. Cohen. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, 2019.

[20] Y. Dong, A. Mircea, and J. C. K. Cheung. Discourse-aware unsupervised summarization for long scientific documents. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1089–1102, Online, Apr. 2021. Association for Computational Linguistics.

[21] Y. Feng, X. Chen, B. Y. Lin, P. Wang, J. Yan, and X. Ren. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Nov. 2020.

[22] C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini. The webnlg challenge: Generating text from rdf data. In *INLG*, pages 124–133, 2017.

[23] D. G. Ghalandari, C. Hokamp, J. Glover, G. 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.

[24] G. Giannakopoulos, J. Kubina, J. Conroy, J. Steinberger, B. Favre, M. Kabadjov, U. Kruschwitz, and M. 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.

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

[26] T. Hasan, A. Bhattacharjee, M. S. Islam, K. Samin, Y.-F. Li, Y.-B. Kang, M. S. Rahman, and R. Shahriyar. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. *arXiv preprint arXiv:2106.13822*, 2021.

[27] H. Hayashi, P. Budania, P. Wang, C. Ackerson, R. Neervannan, and G. Neubig. Wikiasp: A dataset for multi-domain aspect-based summarization. *Transactions of the Association for Computational Linguistics*, 9:211–225, 2021.

[28] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. spaCy: Industrial-strength Natural Language Processing in Python. *10.5281/zenodo.1212303*, 2020.

[29] N. Jhaveri, M. Gupta, and V. Varma. clstk: The cross-lingual summarization tool-kit. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 766–769, 2019.

[30] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

[31] H. Kahaduwa, D. Pathirana, P. L. Arachchi, V. Dias, S. Ranathunga, and U. Kohomban. Question answering system for the travel domain. In *Moratuwa Engineering Research Conference (MERCon)*, pages 449–454, 2017.

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

[33] D. Khashabi, S. Min, T. Khot, A. Sabharwal, O. Tafjord, P. Clark, and H. Hajishirzi. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv:2005.00700*, 2020.

[34] N. Kitaev, L. Kaiser, and A. Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2019.

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

[36] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*. OpenReview.net, 2020.

[37] 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.

[38] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 09 2019.

[39] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880, 2020.

[40] Y. Liang, N. Duan, Y. Gong, N. Wu, F. Guo, W. Qi, M. Gong, L. Shou, D. Jiang, G. Cao, et al. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv:2004.01401*, 2020.

[41] B. Y. Lin, X. Chen, J. Chen, and X. Ren. KagNet: Knowledge-aware graph networks for commonsense reasoning. In *Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2829–2839, 2019.

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

[43] F. Liu, E. Shareghi, Z. Meng, M. Basaldella, and N. Collier. Self-alignment pretraining for biomedical entity representations. *arXiv:2010.11784*, 2020.

[44] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han. On the variance of the adaptive learning rate and beyond, 2021.

[45] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

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

[47] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692, 2019.

[48] E. Loper and S. Bird. NLTK: The Natural Language Toolkit. *CoRR*, cs.CL/0205028, 2002.

[49] S. Lv, D. Guo, J. Xu, D. Tang, N. Duan, M. Gong, L. Shou, D. Jiang, G. Cao, and S. Hu. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. *arXiv:1909.05311*, 2019.

[50] K. Ma, J. Francis, Q. Lu, E. Nyberg, and A. Oltramari. Towards generalizable neuro-symbolic systems for commonsense question answering. *arXiv:1910.14087*, 2019.

[51] H. Maheshwari, N. Sivakumar, S. Jain, T. Karandikar, V. Aggarwal, N. Goyal, and S. 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, Seattle, United States, July 2022. Association for Computational Linguistics.

[52] 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.

[53] Y. Meng, A. Rumshisky, and A. Romanov. Temporal information extraction for question answering using syntactic dependencies in an lstm-based architecture. *arXiv:1703.05851*, 2017.

[54] A. Mhaske, H. Kedia, S. Doddapaneni, M. M. Khapra, P. Kumar, R. Murthy, and A. Kunchukuttan. Naamapadam: A large-scale named entity annotated data for indic languages, 2022.

[55] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv:1809.02789*, 2018.

[56] R. Mitra, R. Jain, A. S. Veerubhotla, and M. Gupta. Zero-shot multi-lingual interrogative question generation for" people also ask" at bing. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3414–3422, 2021.

[57] 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.

[58] K. Nguyen and H. 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.

[59] M. Popović. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618, 2017.

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

[61] O.-B. Reynier, C. Berta, R. Francisco, R. Paolo, and F. Elisabetta. Profiling Irony and Stereotype Spreaders on Twitter (IROSTEREO) at PAN 2022. In *CLEF 2022 Labs and Workshops, Notebook Papers*. CEUR-WS.org, 2022.

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

[63] S. Sagare, T. Abhishek, B. Singh, A. Sharma, M. Gupta, and V. Varma. Xf2t: Cross-lingual fact-to-text generation for low-resource languages. *arXiv preprint arXiv:2209.11252*, 2022.

[64] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[65] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Neural Information Processing Systems (NeurIPS)*, 2017.

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

[67] S. Satapara, B. Modha, S. Modha, and P. Mehta. Fire 2022 ilsum track: Indian language summarization. In *Proceedings of the 14th Forum for Information Retrieval Evaluation*. ACM, December 2022.

[68] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607, 2018.

[69] T. Scialom, P.-A. Dray, S. Lamprier, B. Piwowarski, and J. Staiano. Mlsum: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, 2020.

[70] 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.

[71] R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence*, 2017.

[72] M. Sykora, S. Elayan, and T. W. Jackson. A qualitative analysis of sarcasm, irony and related #hashtags on twitter. *Big Data & Society*, 7(2):2053951720972735, 2020.

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

[74] 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.

[75] D. Vrandečić. Wikidata: A new platform for collaborative data collection. In *International Conference on World Wide Web (WWW)*, pages 1063–1064, 2012.

[76] P. Wang, N. Peng, F. Ilievski, P. Szekely, and X. Ren. Connecting the dots: A knowledgeable path generator for commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4129–4140, Online, Nov. 2020. Association for Computational Linguistics.

[77] X. Wang, P. Kapanipathi, R. Musa, M. Yu, K. Talamadupula, I. Abdelaziz, M. Chang, A. Fokoue, B. Makni, N. Mattei, et al. Improving natural language inference using external knowledge in the science questions domain. In *AAAI Conference on Artificial Intelligence*, pages 7208–7215, 2019.

[78] M. Welling and T. N. Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[79] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, 2021.

[80] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, June 2021.

[81] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. 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.

[82] R. Zhang, J. Guo, Y. Fan, Y. Lan, and X. Cheng. Outline generation: Understanding the inherent content structure of documents, 2019.

[83] X. Zhang, A. Bosselut, M. Yasunaga, H. Ren, P. Liang, C. D. Manning, and J. Leskovec. GreaseLM: Graph REASoning enhanced language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[84] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu. FreeLB: Enhanced Adversarial Training for Natural Language Understanding. *arXiv:1909.11764*, 2019.

[85] 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.