

Automatic Generation of Hindi Wikipedia Pages

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

Master of Science
in
Computational Linguistics by Research

by

Aditya Agarwal

20161104

`aditya.agarwal@research.iiit.ac.in`



International Institute of Information Technology

Hyderabad - 500 032, INDIA

June 2024

Copyright © Aditya Agarwal, 2024
All Rights Reserved

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “**Automatic Generation of Hindi Wikipedia Pages**” by **Aditya Agarwal**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Radhika Mamidi

To my parents

Acknowledgments

As I pen down this acknowledgment, a whole bag of emotions comes rushing at me. I am emotional, grateful, and exuberant, and these are to name a few. First and foremost, from the bottom of my heart, I would like to thank Dr. Radhika Mamidi for her exceptional guidance and insights during this journey. Her subject matter expertise, coupled with her constant push towards excellence, has helped me to shape this thesis into what it is now.

I want to express my gratitude to IIIT-Hyderabad and the LTRC Lab for giving me this opportunity to work on this topic and for providing me with the necessary requirements and infrastructure. The flexible schedule provided helped me to work on this thesis at my own speed, thus enabling me to take my time to work on the problem. This institute has provided me with a platform into the world of Natural Language Processing and Computational Linguistics, for which I will always be thankful.

I am grateful to my professors, namely, Dr. Dipti Misra Sharma, Dr. Manish Shrivastava, and Dr. Vasudeva Varma, who have been instrumental in helping me reach here with their impactful contributions during this journey. Their little anecdotes of wisdom and their teaching prowess have helped me strengthen my core fundamentals in Computational Linguistics.

I also want to thank Ph.D. student, Pruthwik Mishra, at the LTRC lab who was there for me during any doubts that I had during this long journey.

Most importantly, I am indebted to my dear friends Aniruddha, Saurabh, Animesh, Sahil, Abhishek, Kunal, Vaibhav, and all my wingmates for their support during this journey. I could not have completed this journey without your constant push and understanding during this tough and long journey. You were there for me when I was at my lowest and my highest, and for that, I will always be eternally thankful.

A particular acknowledgment goes to my friend, Aniruddha Deshpande, who was just there, at any time I called or wanted to talk about anything. His brilliant insights, subject matter understanding, and constant support have been instrumental in helping me reach my final destination. You were my pillar, and I hope I was for you as well.

Lastly, I would like to extend my deepest gratitude to my parents for their unwavering support and understanding. They have been so understanding of this journey, that it baffles me. The constant words of encouragement and their belief that I can achieve this milestone have helped me in ways I cannot even thank them for. This is for you, Mom and Dad!

In conclusion, I do not think that this thesis would have been possible if it weren't for all the people mentioned above and the Institution. The guidance and support have been incredible. I am truly indebted and thankful for their contributions to my academic and personal growth.

I will always remember the life lessons that I have learned from all my time at IIIT-Hyderabad and hope to apply them wherever life leads me. Once again, thank you everyone. Signing off.....

Abstract

Natural Language Generation (NLG) is a computer process that uses artificial intelligence to produce written or spoken language from structured or unstructured data [3]. Its purpose is to enable computers to communicate with users in a way that is understandable, rather than in computer language. NLG focuses on creating coherent written content in human languages like English, based on underlying data. Given the vast amount of text data available, NLG techniques are crucial for organizing and presenting information, with Wikipedia being a leading resource in this effort. [15]

Wikipedia is an online encyclopedia that's available in multiple languages and is freely accessible, thanks to contributions from volunteers known as Wikipedians. This collaborative platform uses a wiki-based editing system called MediaWiki. It holds the title of being the most extensive and most accessed reference work in history. Consistently ranked among the top 10 most popular websites by Similarweb and previously by Alexa, Wikipedia is hosted by the Wikimedia Foundation, a non-profit organization based in the United States, which relies on donations to operate. Natural Language Generation in Wikipedia involves creating articles in various languages, either through WikiBot or manual efforts. The linking of language versions on Wikipedia has been improved with the introduction of **Wikidata**, a unified system that uses unique identifiers for entities and their attributes.[8].

English Wikipedia sees the addition of about 500 articles daily, but Hindi Wikipedia lacks such growth, with only 150,000 pages compared to English's 54 million articles. To enhance Natural Language Generation and maintain Wikipedia's multilingual aspect, creating more detailed Hindi pages is crucial. This thesis proposes a method for automatically generating Hindi Wikipedia articles using Wikidata as a knowledge source [26]. The process involves extracting structured data from Wikidata, including entity names, properties, and relationships, and then generating natural language text based on predefined templates for the subject area. We tested our method by generating articles about scientists and compared them to machine-translated ones. Results show over 70% of the articles produced using our method are superior in coherence, structure, and readability. This approach has the potential to significantly reduce the time

and effort needed to create Hindi Wikipedia articles and can be extended to other languages and domains.

Contents

Chapter	Page
1 Introduction	1
1.1 Wikipedia	2
1.2 Wikidata	2
1.3 Knowledge Graph	3
1.4 Motivation	5
1.5 Key Contributions	5
1.6 Thesis Overview	6
2 Related Work	8
2.1 Initial Work	8
2.2 WikWrite: Generating Wikipedia Pages Automatically	8
2.3 Latest Approaches	10
3 Dataset Creation & Data Retrieval Techniques	11
3.1 Domain Selection	11
3.2 Preprocessing	12
3.2.1 Understanding how Wikidata stores data	12
3.2.2 Converting English key-value pairs to Hindi and combining both	13
3.3 Data Retrieval Techniques	14
3.3.1 TF-IDF	15
3.3.2 Frequency Filtering	16
3.4 Conclusion	16
4 Rule Based Template Sentences Model	18
4.1 Introduction	18
4.2 Model	19
4.2.1 Combine keys that align with each other in a specific way	19
4.2.2 Making Double Pair and Single Pair Sentences	23
4.2.3 Gender Nuances in Hindi	24
4.2.4 Features Addition & Final Wikipedia Page Generation	25
4.2.4.1 Feature Addition	25
4.2.4.2 Final Hindi Wikipedia Page Generation	28
4.3 Conclusion	29

5	Error Analysis & Evaluation	31
5.1	Introduction	31
5.2	Error Handling & Observations	32
5.2.1	Observation 1	32
5.2.2	Observation 2	33
5.2.3	Observation 3	34
5.2.4	Error Analysis	36
5.3	Results	37
5.4	Evaluation	37
6	Conclusions and Future Work	43
6.1	Conclusion	43
6.2	Future Work	44
	Bibliography	46

List of Figures

Figure		Page
1.1	A diagram that displays qualifiers, items, and properties in Wikidata. These are used to store all information in Wikidata about a certain entity.	3
1.2	This is how an actual Wikidata page looks on the internet	4
2.1	The proposed framework for WikiWrite	9
2.2	Different techniques and their rouge scores for WikiWrite	9
2.3	The Unified Graph Attention Network Structure	10
3.1	A screenshot showing the WDQS in action displaying the SPARQL syntax used in Wikidata to fetch data.	11
3.2	How Wikidata stores QID information of a scientist.	13
3.3	How english and hindi keys are intermingled in the data for a particular scientist.	14
3.4	A flowchart representing the entire pre-processing pipeline followed.	17
4.1	The top 20–25 key–value pairs with their respective TF–IDF and Frequency Filtering scores combined and arranged in descending order.	20
4.2	The three types of sentences created to accommodate for all kinds of keys about each scientist	24
4.3	The key, "Award Received," in Wikidata with the information about why the award was awarded.	26
4.4	Two screenshots showing the different ways date and time are stored on Wikidata and when we parse the data downloaded from Wikidata.	27
4.5	The entire model pipeline followed to generate the template sentences.	30
5.1	A screenshot that shows the spouse information mentioned in Wikidata for which we needed the gender to decide the appropriate sentence ending for the template sentence.	33
5.2	The list of awards won by Irene Curie which we needed to combine when writing the sentence pertaining to the awards she won during her lifetime.	35
5.3	Another screenshot that displays the multiple occupations for an example Scientist on Wikidata, which were required to be combined to make the template sentence.	36
5.4	The 1st page of the Research Survey circulated to evaluate our approach over the traditional Machine Translation approach	41

5.5	The 2nd page of the Research Survey circulated to evaluate our approach over the traditional Machine Translation approach	42
6.1	The final generated Hindi Wikipedia page on Benjamin Thompson published on Wikipedia	44

List of Tables

Table	Page
5.1 Final Results of Evaluation	39

Chapter 1

Introduction

Human language is vital for human progress, enabling the sharing of ideas and driving societal advancement. While people naturally understand and use language, teaching machines to do so is complex. Natural Language Processing (NLP) is a field in computer science that focuses on computers interacting with human language. It aims to teach machines language basics, including syntax, morphology, semantics, and pragmatics. NLP involves two main tasks: Natural Language Understanding, where computers grasp the structure and meaning of human language, and Natural Language Generation, where computers create meaningful human-like language. Natural Language Generation (NLG) involves machines automatically producing narratives that describe, summarize, or explain input data in a human-like way, enhancing their ability to communicate meaningfully.

NLG [15] is a multi-stage process that involves refining data and generating natural-sounding language content. The six stages of NLG can be summarized as follows:

1. Content analysis: The data is carefully examined to identify the relevant information that should be included in the final content. This stage focuses on determining the main topics in the source document and understanding the relationships between them.
2. Data understanding: The data is interpreted and analyzed to understand its meaning and context better. This stage often involves using machine learning techniques to uncover patterns and insights within the data
3. Document structuring: A document plan is created, and a narrative structure is chosen based on the type of data being processed. This stage focuses on organizing the content logically and coherently.

4. Sentence aggregation: Relevant sentences or parts of sentences are selected and combined to summarize the topic at hand accurately. This stage involves synthesizing information from different sources to create concise and informative sentences.
5. Grammatical structuring: Grammatical rules are applied to ensure the generated text is grammatically correct and coherent. The program analyzes the syntactical structure of the sentences and rewrites them accordingly.
6. Language presentation: The final output is generated based on a selected template or format chosen by the user or programmer. This stage focuses on presenting the content in a way that is visually appealing and accessible to the intended audience.

1.1 Wikipedia

Wikipedia is a remarkable example of human collaboration, housing one of the largest repositories of online knowledge. This widely utilized, multilingual encyclopedia relies on the contributions of volunteers and an open editing system. Despite being available in an impressive 326 languages, Wikipedia’s content distribution is uneven. Languages with larger internet user bases tend to have more content, while others face challenges due to a smaller editor pool, potentially leading to lower quality control and shorter articles. For example, as of April 2024, Hindi Wikipedia boasts **161,266**¹ articles, with an average length closer to 1000 words per article. This is still significantly less than the staggering **6,811,064**² articles in English Wikipedia. The dearth of content in specific languages can diminish the appeal of those versions to readers, making it more challenging to attract new editors. However, Wikidata serves as a valuable resource in addressing these imbalances by serving as the structured data foundation for Wikipedia.

1.2 Wikidata

Wikidata is a knowledge base hosted by the Wikimedia Foundation, where users edit and contribute information collaboratively [29]. It provides open data under a public domain license, which Wikimedia projects and others can use [28]. Data in Wikidata is stored with specific IDs serving as the base for the platform. Each entity has a unique ID, starting with a letter prefix: Q for items (e.g., Albert Einstein (Q937)), P for properties (e.g., an instance of (P31)), and L for lexemes (e.g., L1). This setup is illustrated in Figure 1.1. Additionally, Wikidata features a query service called WDQS³, enabling users to run queries on its extensive database

¹https://en.wikipedia.org/wiki/Hindi_Wikipedia

²https://en.wikipedia.org/wiki/English_Wikipedia

³<https://rb.gy/bv8of>

using an RDF triple store for SPARQL⁴ queries. With an active international community of volunteers, Wikidata contains information on over 55 million entities, covering people, places, and events. Its multilingual design allows data translation and presentation in the user's preferred language. This feature makes Wikidata a preferred tool for various content integration functions in Wikipedia, including cross-language article linking and infobox display.

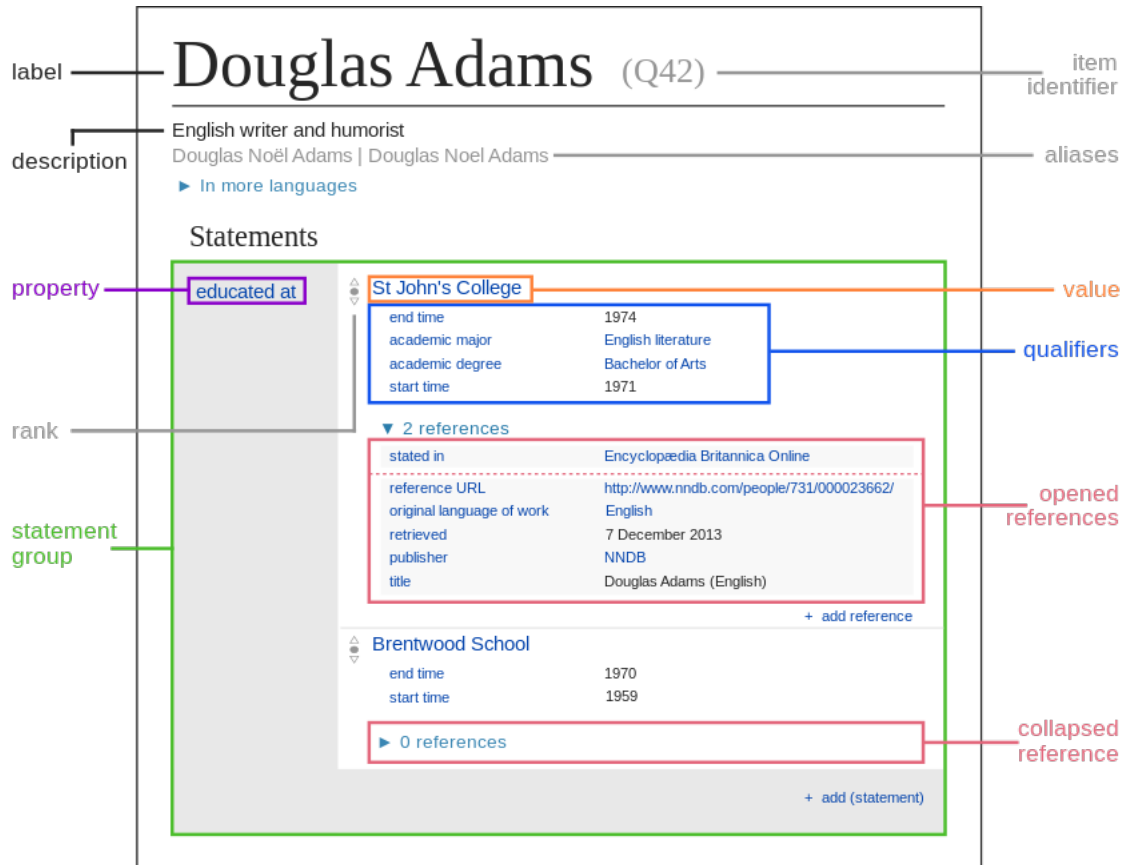


Figure 1.1 A diagram that displays qualifiers, items, and properties in Wikidata. These are used to store all information in Wikidata about a certain entity.

1.3 Knowledge Graph

A knowledge graph, also known as a semantic network, is a visual representation of connections among real-world entities, such as objects, concepts, events, or situations [6]. The fundamental components of a knowledge graph are nodes, edges, and labels. Nodes represent any entity,

⁴<https://www.w3.org/TR/rdf-sparql-query/>

whether it be a person, place, or thing. Edges, on the other hand, indicate the association between two nodes. Knowledge graphs are essential tools for effective knowledge management, and Wikidata is a prime example of a knowledge graph. In Wikidata, scientists (in our case) are the nodes, with information on the scientist as another node and the property as the edge. Figure 1.2 shows an actual Wikidata page of a Scientist. As can be seen in Figure 1.2, the property of the Scientist is **"instance of"**, and the information on the scientist is **"human"** in this case. For future reference in this thesis, we will represent these properties as **keys** and the respective information pertaining to that property as **values**.

The screenshot displays the Wikidata page for Guido van Rossum (Q30942). The page layout includes a sidebar with navigation links, a main content area with a title and description, a table of labels in various languages, and a 'Statements' section.

Wikidata Logo: WIKIDATA

Navigation Links: Main page, Community portal, Project chat, Create a new Item, Create a new Lexeme, Recent changes, Random Item, Query Service, Nearby, Help, Donate, Print/export, Create a book, Download as PDF, Printable version, Tools, What links here, Related changes, Special pages, Permanent link, Page information, Concept URI, Cite this page.

Item: Discussion | Read | View history | Search Wikidata

Guido van Rossum (Q30942)

Dutch programmer and creator of Python

In more languages

Configure

Language	Label	Description	Also
English	Guido van Rossum	Dutch programmer and creator of Python	
Spanish	Guido van Rossum	científico de la computación, conocido por ser el autor del lenguaje de programación Python	
Traditional Chinese	No label defined	No description defined	
Chinese	吉多·范罗苏姆	No description defined	Guid 吉多

All entered languages

Statements

instance of human edit

1 reference

+ add value

Figure 1.2 This is how an actual Wikidata page looks on the internet

1.4 Motivation

Hindi is one of the 22 official languages of India and is spoken by over 600 million people, yet the Hindi Wikipedia lacks in articles over the English Wikipedia. There is also no proper dataset available for the task of enriching Hindi Wikipedia automatically. Even though generating coherent and discourse-related sentence-length natural language text in different languages is now possible due to improved computing power and model capacity, generating multiple sentences that display coherence and relevance to a topic remains a challenge. This is especially true for Scientific domains, with minimal research done in Indian languages like Hindi. Our approach focuses on generating such human-like Hindi Wikipedia pages in the Scientist domain with a minimum length of 500 words. This project aims to surpass existing projects like LSJbot⁵ by generating longer documents that encompass all relevant information. We aim to address the scarcity of comprehensive and up-to-date information in Hindi on Wikipedia by leveraging automated content creation. This thesis examines the challenges and opportunities involved in implementing NLG for Hindi and explores the effectiveness of template-based approaches for generating high-quality, coherent, and contextually relevant Hindi Wikipedia articles.

1.5 Key Contributions

In this thesis, we describe a model that generates template sentences using a dataset specifically created from scratch. This dataset incorporates data points from the Scientist domain sourced from Wikidata. The template sentences are manually crafted with key-value placeholders filled using the dataset’s specific data points. Following that, the sentences undergo rearrangement based on a rule-based system to generate an article. The thesis also introduces this dataset created in Hindi and provides detailed insights into the nuances of the template sentences model along with the dataset construction process. This dataset is comprehensive, containing Hindi key-value pairs for 17,000 scientists who do not yet have a Hindi Wikipedia page. We also believe that our approach can be extended to other domains provided relevant translations and data are scraped for processing. The key contributions are:-

- **Data Analysis and Understanding:** We conducted an in-depth analysis of existing Wikidata to understand its structure and content. This exploration allowed us to grasp the nuances of the Scientist domain data available on Wikidata.
- **Requirement Analysis from Hindi Wikipedia:** By examining existing Hindi Wikipedia pages, we identified the specific content needed for scientist entries. This analysis guided our approach in sourcing relevant data from Wikidata.

⁵<https://en.wikipedia.org/wiki/Lsjbot>

- **Data Collection and Preprocessing:** Utilizing SPARQL and WDQS software, we systematically downloaded, extracted, and preprocessed the data from Wikidata. Our efforts were aimed at consolidating the dataset into a single, manageable resource.
- **Data Cleaning and Enrichment:** We meticulously cleaned and enriched the dataset, ensuring its accuracy and usability. Furthermore, we engaged Hindi experts to translate certain entities into Hindi with complete precision, enhancing the dataset’s quality.
- **Creation of a Comprehensive Dataset:** Recognizing the absence of a suitable dataset for Hindi Wikipedia content in our domain, we meticulously curated a dataset from scratch. This dataset encompasses 17,000 scientists for whom Hindi Wikipedia pages are currently unavailable, addressing a significant gap in the existing resources.
- **Development of the Template Sentences Model:** We developed a template sentences model based on the curated dataset. This model employs a combination of rules and Hindi syntactic dependencies to generate article content for Hindi Wikipedia entries effectively.

1.6 Thesis Overview

The remaining chapters are organized as follows:

Chapter 2: In this chapter, we look at the previous work done in the field of creating articles in Wikipedia, early efforts employing machine learning techniques, along with providing English Datasets and using various other methods to improve and enrich Wikipedia.

Chapter 3: This chapter will talk about our dataset in depth. We will look at the structure of how Wikidata stores data and the intricacies involved in converting this data into a human-readable format. We will also talk about how we used data extraction and retrieval techniques like TF-IDF and frequency filtering to pre-process this data.

Chapter 4: In this chapter, we present the template sentences model and show how we used the dataset to create these sentences. We also present a novel approach to creating complex sentences and using rule-based methods to create the final article. We also explore ways for further improvements to the data to make it as comprehensive as possible.

Chapter 5: We then move on to the Error Analysis part, wherein we talk about how we resolved the errors, as Hindi has various syntactic dependencies that need to be carefully handled. We also talk about the Evaluation Scheme undertaken by us to ensure the articles we generate beat the existing Wikibot system. We present the Research Survey conducted, and the evaluation results from 50 Hindi-English bilinguals.

Chapter 6: In this chapter, we conclude by presenting a brief summary of the topics discussed in the thesis and discussing the directions for possible future work.

Chapter 2

Related Work

2.1 Initial Work

Existing methods like [24] use the high-level structure of human-authored texts to automatically induce a domain-specific template for the topic structure of a new overview. While [27], [20], and [5] focus on generating sentences, a more challenging and interesting scenario emerges when the goal is to generate multi-sentence texts. [19] used Wikipedia articles in nine languages to identify word translations through keywords and a word alignment algorithm. [25] proposed to use links to retrieve Wikipedia articles in English, similar to an article in German. [16] established a structure for crafting a Wikipedia entry about a specific entity, ensuring not only visual consistency with other articles in the same category but also encompassing various aspects associated with the entity gathered from the web. [23] also investigates fully automatic methods to generate info-boxes for Wikipedia from the Wikidata knowledge graph.

2.2 WikWrite: Generating Wikipedia Pages Automatically

[1] introduces WikiWrite, a system to author new articles on Wikipedia automatically by obtaining vector representations of the red-linked entities using a paragraph vector model [9] that computes continuous distributed vector representations of varying-length texts. As can be seen from Figure 2.1, the paper uses the entire Wikipedia to obtain D-dimensional representations of words/entities as well as documents using the paragraph vector distributed memory (PV-DM) model from [9]. Similar articles are identified using cosine similarity between the vector representations of the missing entity and representations of the existing entities (entities that have corresponding articles). Content from similar articles is used to train multi-class classifiers that can assign web-retrieved content on the red-linked entity to relevant sections of the article.

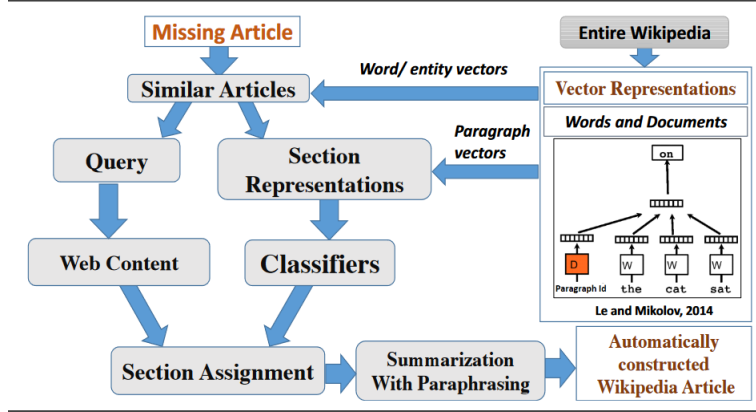


Figure 1: WikiWrite: Our Proposed Framework

Figure 2.1 The proposed framework for WikiWrite

Table 2: Content Selection Results

Technique	ROUGE-1	ROUGE-2
WikiWrite	0.441	0.223
WikiWrite (Ref)	0.520	0.257
WikiKreator	0.371	0.183
Perceptron-ILP	0.342	0.169

Table 3: Statistics of Wikipedia stub content addition

Statistics	WikiKreator	WikiWrite
No. of stubs appended	40	40
Entire edit retained	15	32
Modification of content	5	5
Content Removed	20	3
Avg. change in size	287 bytes	424 bytes
Avg. no of edits	3.82	1.39

Figure 2.2 Different techniques and their rouge scores for WikiWrite

From Figure 2.2, WikiWrite outperforms both WikiKreator and Perceptron-ILP according to both the ROUGE scores. WikiWrite (Ref) from the table performs better than WikiWrite because we use more reliable and verified references from Wikipedia articles. All the systems except Perceptron-ILP consist of a summarization component. Therefore, even with the same

or similar web sources, the systems equipped with summarizers retain more informative (higher ROUGE scores) content.

2.3 Latest Approaches

The research conducted by [21] shows the latest work that introduces a unified graph attention network structure for investigating graph-to-text models that combine global and local graph encoders in order to improve text generation. An extensive evaluation of their models demonstrated that the global and local contexts are empirically complementary, and a combination can achieve state-of-the-art results on two datasets. The same can be seen in Figure 2.3. [12] uses extractive summarization to coarsely identify salient information and a neural abstractive model to generate the articles. These models substantially help in providing and enriching Wikipedia Pages. Although these works carry out some matching across languages and improve English Wikipedia, we could not find references on creating Wikipedia Pages for the Hindi Language. To the best of our knowledge, we are the first to propose a dataset and evaluate a method in this field.

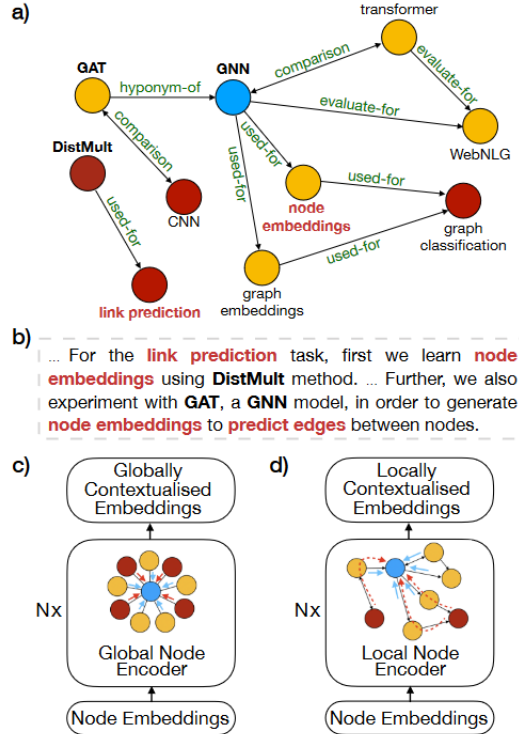


Figure 2.3 The Unified Graph Attention Network Structure

Chapter 3

Dataset Creation & Data Retrieval Techniques

3.1 Domain Selection

Before we understand how we collected the domain-specific data and created the dataset, it's essential to understand the intricacies of choosing the domain. Initially, we chose monuments as our domain due to the low number of Hindi Wikipedia articles in this category, with only **284** Hindi Wikipedia Pages compared to **11,524** English Wikipedia Pages. However, as we reviewed the available data points, we discovered that the content was inconsistent, lacking coherence and detail, and there were too few data points to establish a reliable template.

```
#defaultView:BubbleChart
SELECT ?occupationLabel (count(DISTINCT ?author) as ?count)
WHERE
{
    ?object wdt:P31 wd:Q13442814
    ; wdt:P50 ?author .
    ?author wdt:P106 ?occupation .
    SERVICE wikibase:label { bd:serviceParam wikibase:language "en,fr" }
}
GROUP BY ?occupationLabel
ORDER BY DESC(?count)
```

Figure 3.1 A screenshot showing the WDQS in action displaying the SPARQL syntax used in Wikidata to fetch data.

After conducting research on various domains, such as animals, films, birds, and trees, we ul-

timately selected the Scientific Person domain. This domain was ideal because it contained English Wikipedia pages for prominent scientists, botanists, zoologists, and other scientific personalities but no corresponding Hindi Wikipedia pages. Additionally, this domain had a wealth of existing Hindi Wikipedia data compared to the Monuments domain. Once we finalized the domain, we used Wikidata’s query service called WDQS to form a preliminary dataset. Querying data using WDQS and its SPARQL technology requires unique identification of domain properties and items, making query writing a task that requires careful attention to syntax dependencies. Figure 3.1 explains these dependencies and intricacy for a sample query in Wikidata.

The query service provided us with a JSON file containing data on nearly **30,000** Wikipedia pages, of which **13,000** already had existing Hindi Wikipedia pages. This allowed us to focus on creating Hindi Wikipedia pages for the remaining **17,000** entities within the Scientific Person domain.

3.2 Preprocessing

In this section, we look at the various steps undertaken to pre-process the data we obtained earlier and how we carefully administered techniques to clean and refurbish the data to create the dataset containing data on 17,000 Scientists.

3.2.1 Understanding how Wikidata stores data

To obtain the key-value pairs for each scientist, we had to understand how data is stored in Wikidata and find the correct approach to retrieve it. We found that for each scientist, the pairs were embedded so deeply that it required 6-7 nested iterations to obtain the values. Figure - 3.2 below shows an example of how Wikidata stores information on one such Scientist(**Benjamin Thompson is represented in Wikidata as "Q44645"**) [28]. As can be seen below, **"P19"** is one of the keys for this Scientist, and the value for this specific key is a nested dictionary as part of this key.

As can be seen in the example above, there were many more keys whose data was deeply embedded, and although this process was time-consuming, we successfully obtained all the pairs for all the scientists. We then used various libraries like QWikidata to convert these key-value pairs into a human-readable format; for example, if we closely look at the image above, we can see that the value that we want from the **P19** key is **Q54174**, but the key **P19** and the value **Q54174** are in the Wikidata format, which we as humans do not understand. The QWikidata module converts these keys and values to **Place of Birth(P19)** and **Woburn(Q54174)**, respectively.

```
>>> from qwikidata.linked_data_interface import get_entity_dict_from_api
>>> x = get_entity_dict_from_api("Q44645")
>>> x['claims']
{'P19': [{'mainsnak': {'snaktype': 'value', 'property': 'P19', 'hash': 'ff3e089ba9478989288598be73aa7588ebe0cac2', 'datavalue': {'value': {'entity-type': 'item', 'numeric-id': 54174, 'id': 'Q54174'}, 'type': 'wikibase-entityid'}, 'datatype': 'wikibase-item'}, 'type': 'statement', 'id': 'q44645$042981CA-1553-4004-BD21-4CFE2648F11B', 'rank': 'normal', 'references': [{'hash': '017fd1438416833026fbb94c3afc47676ee4dfff', 'snaks': {'P854': [{'snaktype': 'value', 'property': 'P854', 'hash': '175b9a6d48a225e7b0b0557116b5288e132f7e79', 'datavalue': {'value': 'http://ieeexplore.ieee.org/iel5/6/5215503/05215534.pdf?arnumber=5215534', 'type': 'string'}, 'datatype': 'url'}]}, 'snaks-order': ['P854']}]}, {'hash': 'ac1634c4254b00954835640157d12fd40784fab7', 'snaks': {'P854': [{'snaktype': 'value', 'property': 'P854', 'hash': '29a2d8bae10e1a7a38c84d2e5a75bec3292f7ea7', 'datavalue': {'value': 'http://www.sothebys.com/en/auctions/ecatalogue/2013/collections-l13311/lot.360.lotnum.html', 'type': 'string'}, 'datatype': 'url'}]}, 'snaks-order': ['P854']}]}, {'P27': [{'mainsnak': {'snaktype': 'value', 'property': 'P27', 'hash': '52d0408c7915122e0519a22577f0cbdc28f749b', 'datavalue': {'value': {'entity-type': 'item', 'numeric-id': 30, 'id': 'Q30'}, 'type': 'wikibase-entityid'}, 'datatype': 'wikibase-item'}, 'type': 'statement', 'id': 'q44645$587C48FF-624E-4FAA-A1B4-A4C3C815B116', 'rank': 'normal', 'references': [{'hash': 'fb836f8656389190ad476f8ed499ae01b20fb640', 'snaks': {'P854': [{'snaktype': 'value', 'property': 'P854', 'hash': '0b1d9174328db403bfd626e1baa3209594fb22e4', 'datavalue': {'value': 'http://www.nytimes.com/2009/04/06/books/06davis.html?pagewanted=all', 'type': 'string'}, 'datatype': 'url'}]}, 'snaks-order': [
```

Figure 3.2 How Wikidata stores QID information of a scientist.

3.2.2 Converting English key-value pairs to Hindi and combining both

After we obtained all pairs for the scientists, we then created a main dictionary for each scientist, with their name as the key and their key-value pairs as a nested dictionary. Some of these nested dictionaries contained English key-value pairs, which were translated manually and combined with the pre-existing Hindi pairs. As can be seen in Figure 3.3, in the nested dictionary of **Q7504(Irene Curie)**, we have "माता"¹ as a pre-existing Hindi key, but along with many other such Hindi keys, we also have keys such as **"nuclear physicist"** which don't have a pre-existing Hindi word to define them; hence, these keys also had to be converted to Hindi to ensure that our Wikipedia page was completely in Hindi and all the necessary information was part of said page.

To ensure greater accuracy in translation, a Hindi Domain Expert was consulted to translate the English key-value pairs, as relying solely on Google/Bing Translate would have resulted in an approximate accuracy of 85% [2], leading to inconsistent translations in the final dataset. Since these English key-value pairs were intermingled with the Hindi and English Pairs, a separate dictionary was created to store the pairs that required translation. Translations were recorded in an Excel file with the corresponding sentence context, allowing for accurate contextual translation.

An interesting example where the sentence context played an important role would be: **Given Word: "leaves"**. Now, this word, if given no context, could be translated as "पत्तियां" whereas if the context is given saying **"He leaves for work"**, the Hindi translation for the same

¹Mother

word comes out to be completely different, i.e. "निकल जाता है" , and hence sentence context was used. The task of back-propagating the translated English key-value pairs from the Excel Sheet to the original Hindi key-value pairs was anticipated to be tedious and involved mapping and clear demarcations for each entity. Despite incorporating these demarcations, some errors were encountered while using the pandas module. After extensive coding, we were ultimately successful in placing the translated English key-value pairs with the existing Hindi key-value pairs in Wikidata and were able to complete the dataset. The dataset was complete in terms of all pairs being in Hindi, but there were some additional features that we implemented to enhance the quality of our dataset, which we will discuss in the next subsection and continue in Chapter 4. Considering we had collated all the pairs of the Scientist in Hindi, our next step involved

```
{'Q7504': {'Hindi': {'माता': ['मेरी क्यूरी'], 'पिता': ['पियरे क्यूरी'], 'नागरिकता': ['फ्रांस'], 'जीवन साथी': ['Frédéric Joliot-Curie'], 'व्यवसाय': ['भौतिक विज्ञानी', 'रसायनशास्त्र वैज्ञानिक', 'प्रोफेसर', 'राजनीतिज्ञ'], 'nuclear physicist', 'शोधकर्ता'], 'संतान': ['Pierre Joliot', 'Hélène Langevin-Joliot'], 'पुरस्कार प्राप्त': ['लौरेन ऑफ ऑनरके अधिकारी', 'रसायन शास्त्र में नोबेल पुरस्कार'], 'honorary doctor of the Jagiellonian University of Krakow', 'Matteucci Medal', 'Order of the Cross of Grunwald, 3rd class', 'Barnard Medal for Meritorious Service to Science', 'honorary doctor of the Maria Curie-Skłodowska University', 'Commander with Star of the Order of Polonia Restituta'], 'मृत्यु का कारण': ['कैंसर'], 'मातृ संस्था': ['Science Faculty of Paris', 'Collège Sévigné'], 'पेरिस विश्वविद्यालय', 'जिसका उदाहरण है': ['मनुष्य'], 'CANTIC ID (former scheme)': ['a11202828'], 'का सदस्य': ['Academy of Sciences of the GDR', 'रूसी विज्ञान अकादमी', 'Royal Academy of Medicine of Belgium', 'Royal Netherlands Academy of Arts and Sciences', 'Polish Academy of Sciences'], 'दिया गया नाम': ['Irène'], 'पद पर आसीन': ['undersecretary'], 'doctoral advisor': ['Paul Langevin'], 'मौत का कारण': ['रक्त का कैंसर'], 'परिवार का नाम': ['Joliot-Curie'], 'निवास': ['पेरिस'], 'nominated for': ['भौतिकी में नोबेल पुरस्कार', 'भौतिकी में नोबेल पुरस्कार', 'रसायन शास्त्र में नोबेल पुरस्कार'], 'सहोदर': ['Ève Curie'], 'कार्य क्षेत्र': ['रसायन शास्त्र', 'रेडियोजीव विज्ञान'], 'जन्म तिथि': [{'time': '+1897-09-12T00:00:00Z', 'timezone': 0, 'before': 0, 'after': 0, 'precision': 11, 'calendar': 'http://www.wikidata.org/entity/Q1985727'}], 'मृत्यु तिथि': [{'time': '+1956-03-17T00:00:00Z', 'timezone': 0, 'before': 0, 'after': 0, 'precision': 11, 'c
```

Figure 3.3 How english and hindi keys are intermingled in the data for a particular scientist.

generating template sentences, but first, we needed to identify the crucial key-value pairs for the Scientist Domain so that we could make the page as comprehensive as possible. Keys such as **Doctoral Advisor**, **Student**, **Doctoral Student**, **Awards Won**, and **Field of Work** were considered essential and important for a page belonging to a Scientist. To extract these pairs, we utilized two highly effective relevance algorithms: TF-IDF and frequency filtering. We'll examine these techniques in depth, detailing each of their contributions toward identifying the most significant key-value pairs for each scientist.

3.3 Data Retrieval Techniques

In this subsection, we look at the two techniques mentioned above, namely TF-IDF and Frequency Filtering, which we used to gather and ascertain the most important key-value pairs for a Scientist.

3.3.1 TF-IDF

TF-IDF is a statistical approach that measures the relevance of a word to a document in a collection of documents. It calculates the score by multiplying two metrics: the frequency of the word in a document and the inverse document frequency of the word across the entire document set. A higher score indicates greater relevance of the word in the document [17]. Mathematically, the TF-IDF score of word t in document d from document set D is computed as follows:

$$tfidf(t, d, D) = tf(t, d).idf(t, D)$$

where

$$tf(t, d) = \log(1 + freq(t, d))$$

$$idf(t, D) = \log(N / (count(dED : tEd)))$$

As TF represents the frequency of a word within a document, in our scenario, the words corresponded to the keys, and the document pertained to each scientist. As the keys could either exist or not with a probability of 0 or 1, we assigned a value of 1 to our data if a particular key existed and 0 if it did not. Consequently, utilizing TF was not required for our data, and hence, we redirected our attention toward document frequency. In document frequency, we needed to determine the relevance of each key-value pair for a given scientist, and hence, we decided to calculate the frequency of each key-value pair across all the scientists, dividing each frequency by the total frequency of all keys in the data. We then took the log of this value and sorted all the values in decreasing order to identify the key-value pairs with the highest frequency. This approach allowed us to gain a better understanding of the significance of each key-value pair, given the low likelihood of two scientists sharing the same number of keys.

To prioritize the importance of least occurring keys, we reviewed approximately 200 Hindi Wikipedia pages of scientists and compiled a list of keys that were not frequently mentioned across all pages but were crucial for a complete scientist profile. Examples of such keys include: "नामांकित किया गया" ² or "छात्र" ³ were important, and we decided to use the IDF concept to include such keys as well. To get rid of the other keys that did not affect the quality of the page and also those for which the frequency was extremely low and not important, we used Frequency Filtering [22], which we will discuss next.

²Nominated for

³Student

3.3.2 Frequency Filtering

Frequency filtering is a technique used to eliminate stopwords, which are commonly used words that do not provide much meaning in a text [10]. The objective is to avoid diluting the importance of less frequent but more meaningful words. TF-IDF indirectly employs it to determine the significance of a word in a document.

We applied the concept of frequency filtering to our data by examining the list of relevant keys sorted by frequency using TF-IDF. To utilize frequency filtering, we established a threshold by analyzing 200 Hindi Wikipedia pages, similar to our earlier approach. Following a comprehensive analysis, we determined a limit for the number of keys to include in our dataset. We set the threshold for the maximum number of keys to 25, aligning with our primary goal of ensuring that each scientist’s profile comprised at least 500 words (provided there was sufficient information available on Wikidata). Any keys exceeding the limit were excluded from our dataset.

Upon completion of the aforementioned procedures, we successfully compiled a list of the top 20-25 most relevant and essential key-value pairs for each scientist. However, for some scientists, due to limited information available on Wikidata, only 10-15 pairs could be extracted. Nevertheless, we ensured that all available information on Wikidata for such scientists was incorporated into their Wikipedia page. Figure 3.4 represents the entire pre-processing pipeline.

3.4 Conclusion

In this chapter, we saw the ways we took to shortlist our domain and how we obtained the data from Wikidata in this domain using SPARQL and WDQS technology. We next discussed the tedious preprocessing steps needed to clean and refurbish this data to create a dataset from scratch. We first analyzed how Wikidata stores data, employed Python modules to decode the same, converted English Key-Value pairs to Hindi using human help, and used Data Retrieval Techniques like TF-IDF and Frequency Filtering to arrive at our final 20-25 key-value pairs for all Scientists.

In the next chapter, we will introduce the model we made on the data obtained so far. This chapter will analyze the model and the intricacies involved in the steps taken to ensure the data is used to its maximum efficiency, and the final Wikipedia Page collates all the information correctly and is a delight to read for the readers.

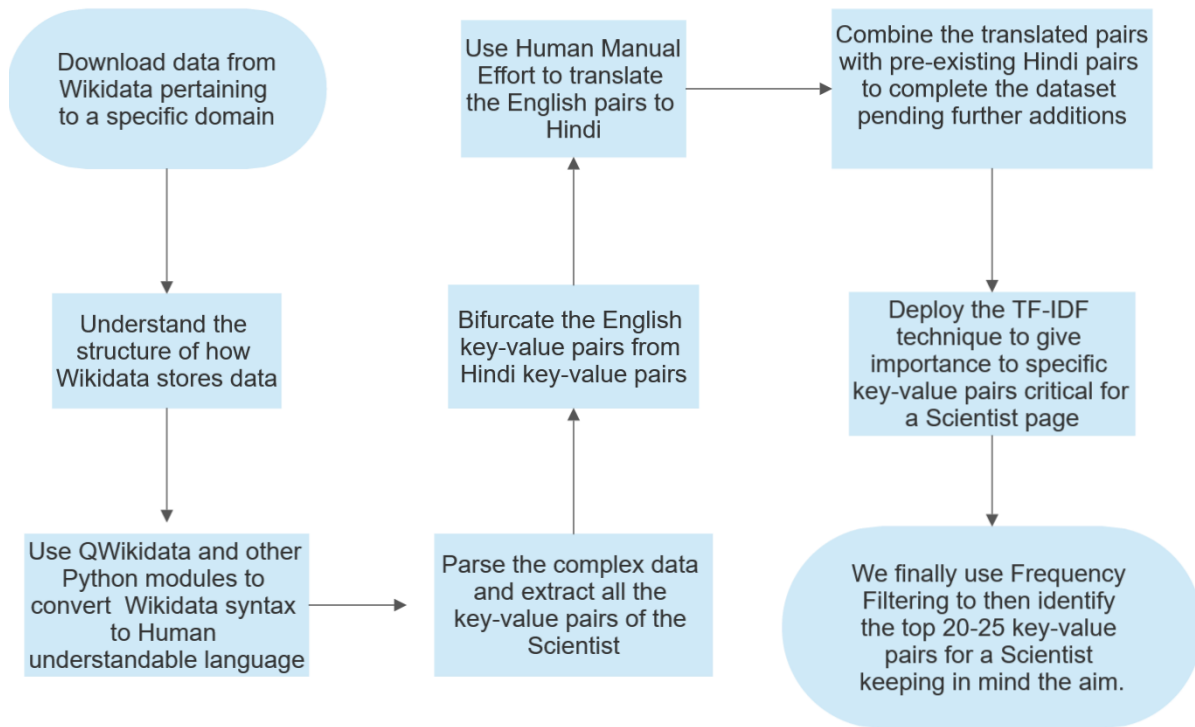


Figure 3.4 A flowchart representing the entire pre-processing pipeline followed.

Chapter 4

Rule Based Template Sentences Model

4.1 Introduction

The field of natural language processing (NLP) has witnessed significant advancements in recent years, with a multitude of applications ranging from chatbots to machine translation. Among the various approaches used to tackle the challenges in NLP, rule-based methods have proven to be effective in many scenarios [4]. In particular, the rule-based template sentences method has emerged as a powerful technique for generating coherent and contextually appropriate sentences.

The rule-based template sentences method involves the creation of a set of predefined rules and templates that govern the generation of sentences. These rules capture the syntactic and semantic patterns that are commonly observed in the target language. By leveraging these templates, the system can generate sentences that adhere to the desired structure and convey the intended meaning.

One of the key advantages of the rule-based template sentence method is its simplicity and interoperability [4]. Unlike more complex neural network-based approaches, rule-based methods offer a transparent framework where each rule can be inspected and understood by human experts. This transparency enables easy rule refinement and customization, making the method highly adaptable to specific domains or applications.

Furthermore, the rule-based template sentences method excels in scenarios where generating diverse but consistent sentences is crucial. By leveraging a set of predefined templates, the method can quickly generate a large number of variations while ensuring that each sentence follows the specified rules. This capability is particularly useful in tasks such as text generation for chatbots, data augmentation for training NLP models or generating personalized responses based on user input.

In this thesis chapter, we aim to delve into the rule-based template sentences method, exploring its underlying principles, design considerations, and practical applications. We will investigate various aspects of the method, including rule creation, template formulation, rule optimization, and system evaluation. Additionally, we will showcase the versatility and effectiveness of the rule-based template sentences method through experimental results and case studies.

By gaining a comprehensive understanding of the rule-based template sentences method, we hope to contribute to the growing body of knowledge in NLP and provide valuable insights for researchers and practitioners seeking efficient and interpretable approaches to text generation. Ultimately, this thesis chapter aims to shed light on the potential of rule-based methods in NLP and inspire further advancements in this exciting field. The next section explains our model.

4.2 Model

As discussed in section 3.3.2, we managed to finalize the top 20-25 most significant key-value pairs for all 17,000 scientists. Figure 4.1 shows these top 20-25 key-value pairs arranged in descending order according to their TF-IDF and frequency filtering scores. This filtering made the process of constructing template sentences more straightforward, as we only had to focus on these keys to create the sentences. The placeholders in these sentences would then be substituted with the unique values of the keys for each scientist. To generate the template sentences, we adopted a unique approach, starting with the most complicated sentences, followed by less complicated ones, and so on. We will explain this further in the upcoming paragraphs.

4.2.1 Combine keys that align with each other in a specific way

We will be able to explain this method better if we take any 3 keys from the top most frequently occurring 20-25 keys as an example. We will then extrapolate the same method applied to these keys to all the keys for each scientist. For instance, the 3 keys are:

{{व्यवसाय}}¹, {{जन्म तिथि}}² and {{जन्म स्थान}}³

which, when used separately, would result in 3 different sentences like

1. वह एक प्रसिद्ध {{व्यवसाय}} थे |,⁴

¹Occupation

²Date of Birth

³Place of Birth

⁴He/She was a famous {{Scientist}}

व्यवसाय 0.04681378968910739
जन्म तिथि 0.04040366243642403
दिया गया नाम 0.03405973766547023
नागरिकता 0.029494644794901832
जन्म स्थान 0.02881534976498119
मातृसंस्था 0.02836344586795771
उदहारण 0.023418408319637096
का उदहारण है 0.023378111156845193
नियोक्ता 0.022799559033904307
मृत्यु तिथि 0.022353411874422528
बोली या लेखी भाषा 0.01754941439587357
परिवार का नाम 0.016884511209807176
पुरस्कार प्राप्त 0.014302614422354563
मृत्यु का स्थान 0.013888129319352138
कार्यक्षेत्र 0.01189629812992381
शैक्षिकदृष्टि/उपाधि 0.010730558777729484
का सदस्य 0.004890348541674462
अंतिम विश्राम का स्थान 0.004055621598127909
मातृभाषा 0.004009567697794306
शैक्षिक डिग्री 0.0036843120266882353
डॉक्टर/सलाहकार 0.003333151036644513
स्थानिक भाषा में नाम 0.0029042740897878352
राजनीतिक दल के सदस्य 0.0020925740964080836
पद पर आसीन 0.0019688042392615257
आधिकारिक वेबसाइट 0.0016176432492178033

Figure 4.1 The top 20-25 key-value pairs with their respective TF-IDF and Frequency Filtering scores combined and arranged in descending order.

2. वह {{जन्म तिथि}} को पैदा हुई थे |,⁵ and

3. उनका जन्म {{जन्म स्थान}} देश में हुआ था |⁶

where '{{व्यवसाय}}', '{{जन्म तिथि}}' and '{{जन्म स्थान}}' are placeholders for the respective scientist Key-Value pairs. To ensure that the Wikipedia page is as informative and linguistically sound as possible, we opt to merge some of the related keys and create a sentence out of them. Employing this technique, we can get one sentence that can tell us the same information as mentioned in the above sentences, along with being more natural and linguistically sound like :

• 'वह एक प्रसिद्ध {{व्यवसाय}} थे जिनका जन्म {{जन्म तिथि}} को {{जन्म स्थान}} देश में हुआ था |' ⁷

This sentence, when read, is coherent, more natural, and can convey all information more efficiently.

Recognizing the advantages of utilizing complex sentences, we embarked on identifying pairs of keys that could be combined to form coherent and meaningful sentences like the 3 keys we used as examples earlier. Through our analysis, we discovered several pairs that aligned well together. Here are a few examples:

1. {{नागरिकता}}⁸, {{Scientist}}, {{मातृसंस्था}}⁹, and {{शैक्षिक दर्जा/उपाधि}}¹⁰,

2. {{Scientist}}, {{कार्य स्थल}}¹¹, {{नियोक्ता}}¹², and {{पद पर आसीन}}¹³, and

3. {{Scientist}}, {{के छात्र}}¹⁴, {{छात्र}}¹⁵, {{डॉक्टरेट सलाहकार}}¹⁶, and {{डॉक्टरेट छात्र}}¹⁷

Based on the above keys, below are the sentences using these keys:

⁵He/She was born on {{Date of Birth}}

⁶He/She was born in {{Place of Birth}}

⁷He/She was a famous {{Scientist}} who was born on {{Date of Birth}} in {{Place of Birth}}

⁸Citizenship

⁹Alma Mater

¹⁰Academic Degree

¹¹Work Location

¹²Employed

¹³Position

¹⁴Student of

¹⁵Students

¹⁶Doctoral Advisor

¹⁷Doctoral Student

1. {{नागरिकता}} में पैदा {हुए/हुई} {{Scientist}} {{मातृसंस्था}} {के/की} पूर्व छात्र {alivestatus/wgop} और आगे चलके उन्होंने {{शैक्षिक दर्जा/उपाधि}} की डिग्री भी प्राप्त की | , ¹⁸
2. {{Scientist}} का कार्यस्थल {{कार्य स्थल}} {alivestatus}, और वह {{नियोक्ता}} में एक {{पद पर आसीन}} के रूप में भी कार्यरत {alivestatus/wgop} | ,¹⁹, and
3. {{Scientist}} के शिक्षक {{के छात्र}} {alivestatus/wgok} और वह {{छात्र}} {के/की} शिक्षक भी {alivestatus/wgop} और इसके आलावा उनके डॉक्टरेट एडवाइजर {{डॉक्टरेट सलाहकार}} {alivestatus/wgok} और वह स्वयं {{डॉक्टरेट छात्र}} के/की डॉक्टरेट एडवाइजर भी {alivestatus/wgop} | ²⁰

We notice here that in the pairs of keys (1, 2, and 3) that align, there are 2 pairs of 4 keys that align and one pair with 5 keys that align. **Thus, we did not try and limit the number of keys that we would align together because the more the number of keys that we could collaborate together, the more complex the sentence would be, which in turn would give us more information.** There was a lower limit of a minimum of three keys that we took, and there was no upper limit.

We make another interesting observation: while making the above template sentences, we had to take care of various Hindi Syntactic Rules. For example, to compare, in English, the translation for Sentence 1 would be "Born in {Place}, {Scientist} was an alumnus of the {Alma-Mater} and went on to earn a {Academic Degree}" where the placeholders {Place}, {Scientist}, {Alma-mater} and {Academic Degree} are the English translations of the main four keys in Sentence 1.

Here, we see an interesting difference; in the case of the English Sentence, **the gender of the Scientist will not play any role whatsoever in the formation of the sentence. Be it a male or a female scientist, the sentence remains the same. However, the same sentence in Hindi changes drastically with the gender as {थे/थी}** (This is represented by {alivestatus} in our case), {हुए/हुई} and {के/की} placeholders also need to be added and changed according to the gender of the scientist [7]. While it is important to consider these nuances, for this explanation, we will temporarily set them aside and address them in a later section (Section 4.2.3) . For now, let us focus on the four major keys, namely: {{नाग-

¹⁸{{Scientist}} is a citizen of {{Citizenship}} whose alma mater is {{Alma Mater}} and who has attained a degree in {{Academic Degree}}

¹⁹{{Scientist}}'s workplace was in {{Work Location}} and he/she was employed at {{Employed}} at the position of {{Position}}

²⁰{{Scientist}} was a student of {{Student of}} and he/she was a student to {{Students}}. Apart from this, his/her doctoral advisor is/was {{Doctoral Advisor}} and his/her doctoral student is/was {{Doctoral Student}}

रिक्ता}}, {{Scientist}}, {{मातृसंस्था}}, and {{शैक्षिक दर्जा/उपाधि}} which have been embedded in Sentence 1 within double curly brackets ({{}}). We ignore the rest of the information embedded in the single curly brackets ({}) for now, as mentioned above.

4.2.2 Making Double Pair and Single Pair Sentences

As we observed that making sentences with multi-pair keys that align offered a deeper understanding of the language’s complexity by conveying multiple points of information, we decided to use this approach for all 20-25 keys and generated 11 coherent sentences that combined multiple keys. Additionally, we also saw an opportunity to extend this complexity by using two keys that align. We thus applied P&C concepts to create sentences with fewer complexities than multi-pair keys with two keys. Sentence 1 above contains three keys, and using P&C concepts, we could create three sentences by using any two of the three keys. Therefore, we obtained the following three sentences with every two out of the three keys: ({{नागरिकता}}, {{मातृसंस्था}}, and {{शैक्षिक दर्जा/उपाधि}}) :

1. वह {{नागरिकता}} के नागरिक {alivestatus/wgop} और वह {{मातृसंस्था}} {के/की} पूर्व छात्र भी {alivestatus/wgop} |²¹
2. उन्होंने {{शैक्षिक दर्जा/उपाधि}} की डिग्री भी प्राप्त की {alivestatus/gen} और वह {{मातृसंस्था}} {के/की} पूर्व छात्र भी {alivestatus/wgop} |²²
3. वह {{नागरिकता}} {के/की} नागरिक {alivestatus/wgop} और उन्होंने {{शैक्षिक दर्जा/उपाधि}} की डिग्री भी प्राप्त की {alivestatus/gen} |²³

Since these sentences also sounded naturally coherent and provided a deep understanding of the Hindi language, they were deemed suitable for use on the Hindi Wikipedia page. To replicate this success, we created multiple variations of the original 11 multi keys sentences. For example, If a sentence had three keys originally, we made three sentences with two out of those three keys, or if one of the 11 sentences had five keys, we made ten sentences ($5C2 = 10$ sentences), and so on. After this process, we generated 80 template sentences created through P&C of the original 11 sentences. Upon reviewing the dataset, we realized that certain scientists did not possess even two out of the three keys. Consequently, if we failed to create a sentence using the one key they did have, we would lose valuable information about those individuals.

²¹He/She was a citizen of {{Citizenship}} and he/she's alma mater was {{Alma Mater}}

²²He/She obtained a degree in {{Academic Degree}} and his/her alma mater is/was {{Alma Mater}}

²³He/She was a citizen of {{Citizenship}} and he/she also obtained a degree in {{Academic Degree}}

वह {{के छात्र }} {के/की} छात्र {alivestatus/wgop} |
 एक प्रोफेसर के रूप में, उनके छात्रों में {{छात्र}} भी शामिल {alivestatus/wgok} |
 वह {{डॉक्टरेट छात्र}} {के/की} डॉक्टरल एडवाइजर्स {alivestatus/wgop} |
 उनके डॉक्टरल एडवाइजर्स {{डॉक्टरेट सलाहकार}} {alivestatus/wgok} |

'{{Scientist}} के शिक्षक {{के छात्र }} {alivestatus/wgok}, डॉक्टरेट एडवाइजर {{डॉक्टरेट सलाहकार}} {alivestatus/wgok} और वह स्वयं {{डॉक्टरेट छात्र}} {के/की} डॉक्टरेट एडवाइजर भी {alivestatus/wgop}',
 '{{Scientist}} के डॉक्टरेट एडवाइजर {{डॉक्टरेट सलाहकार}} {alivestatus/wgok} और वह स्वयं {{डॉक्टरेट छात्र}} {के/की} डॉक्टरेट एडवाइजर होने के साथ साथ {{छात्र}} के शिक्षक भी {alivestatus/wgop}',
 '{{Scientist}} के शिक्षक {{के छात्र }} {alivestatus/wgok}, डॉक्टरेट एडवाइजर {{डॉक्टरेट सलाहकार}} {alivestatus/wgok} और वह {{छात्र}} के शिक्षक भी {alivestatus/wgop}',
 '{{Scientist}} के शिक्षक {{के छात्र }} {alivestatus/wgok} और वह स्वयं {{डॉक्टरेट छात्र}} {के/की} डॉक्टरेट एडवाइजर होने के साथ साथ {{छात्र}} के शिक्षक भी {alivestatus/wgop}',
 '{{Scientist}} के डॉक्टरेट एडवाइजर {{डॉक्टरेट सलाहकार}} {alivestatus/wgok} और वह स्वयं {{डॉक्टरेट छात्र}} {के/की} डॉक्टरेट एडवाइजर भी {alivestatus/wgop}',
 '{{Scientist}} के डॉक्टरेट एडवाइजर {{डॉक्टरेट सलाहकार}} {alivestatus/wgok} और इसके अलावा, वह {{छात्र}} {के/की} शिक्षक भी {alivestatus/wgop}',
 '{{Scientist}} स्वयं {{डॉक्टरेट छात्र}} {के/की} डॉक्टरेट एडवाइजर {alivestatus/wgop}, और इसके अलावा, उनके शिक्षक {{के छात्र }} {alivestatus/wgok}',

{{Scientist}} के शिक्षक {{के छात्र }} {alivestatus/wgok} और वह {{छात्र}} {के/की} शिक्षक भी {alivestatus/wgop} और इसके अलावा उनके डॉक्टरेट एडवाइजर {{डॉक्टरेट सलाहकार}} {alivestatus/wgok} और वह स्वयं {{डॉक्टरेट छात्र}} {के/की} डॉक्टरेट एडवाइजर भी {alivestatus/wgop} |

Figure 4.2 The three types of sentences created to accommodate for all kinds of keys about each scientist

Therefore, we concluded that in addition to the 11 triple-key and 80 double-key sentences we had generated, we needed to develop additional sentences that accounted for such scenarios. To minimize the level of risk, we opted to create single key sentences based on the 11 sentences we initially developed, resulting in nearly 60 additional sentences. In total, we ended up with 160 template sentences.

4.2.3 Gender Nuances in Hindi

Figure 4.2 illustrates the three types of sentences we created - Single Pair, Double Pair, and the original Multi Pair sentences based on one of the 11 multi-pair sentences we had previously discussed.

Once we had all the 160 template sentences, we needed to consider the nuances of gender in Hindi and English before creating the Wikipedia pages. We examined the Wikidata pages and found the "लिंग" key for each scientist, which we used to determine whether a scientist was male or female. Based on this, we created placeholders for words that varied with gender,

such as 'थे/थी' (represented by `alivestatus` in our case), 'हुए/हुई', and 'के/की' . We then filled these placeholders with the appropriate gender-based choice for each scientist. The following examples illustrate this process. For instance, we took two scientists from our dataset, **Frank Malina** and **Rosina M. Bierbaum**. Frank Malina's country of citizenship/place is the **USA**; alma mater is **Texas A&M University**, and academic degree is **Doctor of Philosophy**, while Rosina M. Bierbaum's country of citizenship/place is the **USA**, alma mater is **Stony Brook University**, and academic degree is **Doctor of Philosophy**. After filling in this information, we obtained the following sentences:

1. USA में पैदा हुए "फ्रैंक मलीना" "टेक्सास A&M यूनिवर्सिटी" के पूर्व छात्र थे और आगे चलके उन्होंने "डॉक्टर ऑफ फिलॉसफी" की डिग्री भी प्राप्त की | ²⁴
2. USA में पैदा हुई "रोसिना म बैरभौम" "सटोनी ब्रूक यूनिवर्सिटी" की पूर्व छात्र थी और आगे चलके उन्होंने "डॉक्टर ऑफ फिलॉसफी" की डिग्री भी प्राप्त की | ²⁵

As one can notice, there are stark differences in the way Hindi handles gender [18], with placeholders like 'हुई', 'हुए', 'के', 'की' changing according to whether the Scientist is a male or female. We coded the same for all 17,000 Scientists and identified all the Gender information for the same. Thus, finally, after all such nuances were dealt with, we had 160 template sentences in our hands, and we now moved on and were ready for the Feature Addition and Final Template Page Generation Step.

4.2.4 Features Addition & Final Wikipedia Page Generation

This section is divided into two parts: Feature Addition, which covers the additional features added to complete the template sentences, and Final Wikipedia Page Generation, which explains the rule-based system used to determine the order of the template sentences and how the page was ultimately created.

4.2.4.1 Feature Addition

We reviewed existing Hindi Wikipedia pages of scientists and compared them to our template sentences. We also searched Wikidata to find additional information to make our pages more

²⁴English Translation: Born in the USA, "Frank Malina" was an alumnus of "Texas A&M University" and later earned the degree of "Doctor of Philosophy".

²⁵English Translation: Born in the USA, "Rosina M. Bierbaum" was an alumnus of "Stony Brook University" and later earned the degree of "Doctor of Philosophy".

informative. We discovered that certain keys, such as **Award Received**, had values and references that linked to other Wikidata pages with valuable information. For instance, Nobel Prize information provided details on why and for what reason the award was given. Though accessing information through Wikidata's complex format was challenging, we persevered to access other Wikidata pages to obtain the reason for the award received.




award received		Officer of the Legion of Honour
	point in time	1939
		1 reference
		Nobel Prize in Chemistry
	point in time	1936
	together with	Frédéric Joliot-Curie
	prize money	79,958 Swedish krona
	award rationale	in recognition of their synthesis of new radioactive elements (English) såsom ett erkännande för deras gemensamt utförda syntes av nya radioaktiva grundämnen (Swedish)
		2 references
	reference URL	http://www.nobelprize.org/nobel_prizes/chemistry/laureates/1935/joliot-curie-bio.html
		honorary doctor of the Jagiellonian University of Krakow
	point in time	1951
		1 reference

Figure 4.3 The key, "Award Received," in Wikidata with the information about why the award was awarded.

As can be seen in Figure 4.3 that refers to the Wikidata page of Irene Curie, we see that the page has a key called "award received" with various values such as "Officer of the Legion of Honour," "Nobel Prize in Chemistry" etc. We also notice that the "Nobel Prize in Chemistry" has another section titled "award rationale" that displays the reason she received the award. To access this information, we need to access the Wikidata page of the "Nobel Prize in Chemistry" and then access the Q7504(Irene Curie) key inside the data and finally arrive at the reason. We then embed this information into our existing dataset and reformat the template sentence to accommodate this information as well. We also had to reformat keys such as **Date of Birth and Date of Death** to comply with accepted standards. Similar to the Award Received key, we had to locate the linked information, scrape it, and retrieve the necessary data. Additionally,

we decided to add the "Alive Status" key to our data, as the Hindi language encodes a person's living or deceased status, which affects the sentence endings. 'है', 'था', 'थे' or 'थी'. Figure 4.4

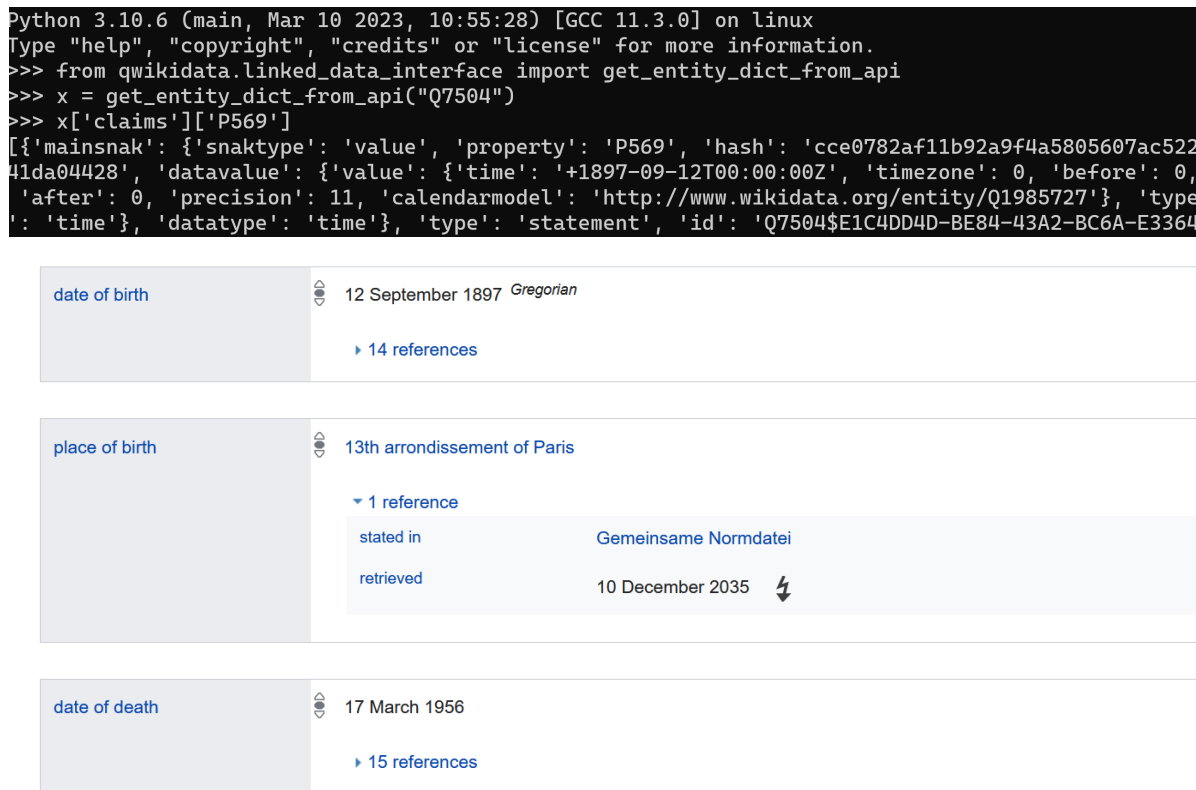


Figure 4.4 Two screenshots showing the different ways date and time are stored on Wikidata and when we parse the data downloaded from Wikidata.

displays two screenshots, with the second screenshot displaying the date of birth and death of Q7504(Irene Curie's) Wikidata page. As can be seen, the Date of Birth is 12th September 1897, and the Date of Death is 17th March 1956, respectively. Now, if we parse the Wikidata page using Python modules, as in the first screenshot, we can see the output in the complicated format that Wikidata stores information. We see that the same date of birth is written as '+1897-09-12T00:00:00Z' in the data downloaded, and it needs to be reformatted to be bought into the format that the second screenshot represents. To do this, we use a certain Python module called `datetime` that can help us achieve this particular goal.

Building on these nuances, we can also see that if we talk about a person who is no longer living, there are three types of the ending of a sentence, namely 'था' or 'थे' or 'थी' which further encodes gender and respect as well. For Females, we take 'थी' . For Males, we use 'था' . Even further, Hindi also has a respect honorific it uses to give respect to either a reputed personality or a great scientist. We use the third type to display respect: 'थे.' To address this issue of which

sentence ending to choose, we added Date of Death and Birth information to our template sentences to detect appropriate sentence endings for each scientist automatically. Since we had already obtained this information, we only needed to check if the Date of Death key existed for each scientist. If not, we assumed they were still alive. Using this information, we added the final feature to the template sentences, completing the task.

4.2.4.2 Final Hindi Wikipedia Page Generation

Before creating the final Hindi Wikipedia page from our template sentences, we developed a rule-based system to determine the order and type of sentences to use. We used the different kinds of sentences we created to help us achieve this.(Refer section 4.2.2).

We decided to start the Wikipedia page with a complex, multi-key sentence to showcase our natural language understanding. To account for situations where a scientist did not have all the keys required for the multi-key sentence, we had two-pair and single-pair sentences as backups. If the scientist did not have the key at all, we excluded that information from the sentences. This ensured that all available information was used to create sentences for the Wikipedia page.

To summarize, the order of the sentences was determined based on a weighted metric that assigned higher points to important keys such as Award Received, Date of Birth and Death, Doctoral Advisor, Student, and Academic Degree. Keys like spouses and children were given lower points. Additionally, the natural flow of information was taken into consideration, starting with introducing the scientist’s profession, then providing their Date of Birth and Death, followed by their academic qualifications and awards. If the scientist had received any awards or nominations, the reasons behind them were explained next. Finally, their family and eventual death were discussed, with rules in place to correlate the two.

When this was mathematically ascertained, we came up with an order of sentences that we felt justified our observations and gave a deeper natural understanding of the Hindi Language. We also followed the system to go for the double pair sentences and single if needed. An interesting case describes our thought process:

In the sentence order, we determined that the first sentence for a scientist would include Date of Birth, Place of Birth, and Occupation. The second sentence would contain Academic Degree, Country of Citizenship, and Alma Mater. However, if the scientist doesn’t have the Place of Birth key, we prioritize the double pair sentence that combines Profession and Country of Citizenship, writing it as the first sentence instead. The second sentence remains the same. Similarly, if the Date of Birth key is also missing, we select the single pair sentence that includes Occupation as the starting sentence, followed by the second most complex sentence. If

none of these three keys exist for the scientist, we choose the second most complex sentence as the starting sentence. This process continues until all 11 triple-pair sentences are utilized

Finally, we utilized these sentences to generate the final automatic Hindi Wikipedia page using a program. By inputting a scientist's name from our dataset, the program would automatically create a file that filled in all the relevant information for that scientist.

4.3 Conclusion

In this chapter, we introduced our novel Template Sentence model built on the data obtained using the processing pipeline done in the earlier chapter. It involves combining multiple keys that exhibit similarities and building three types of sentences, namely single-pair, double-pair, and Multi-Pair keys. These sentences help us decide the order of the information that will be presented to the reader using the mathematical weights assigned.

We then looked at the various features that were added that were missing from Wikidata and were needed to ensure the Wikipedia page created displayed all information pertinent to the Scientist. We studied the various nuances in the Hindi language and looked at ways we dealt with these to ensure a smooth and meticulous process of collating the information.

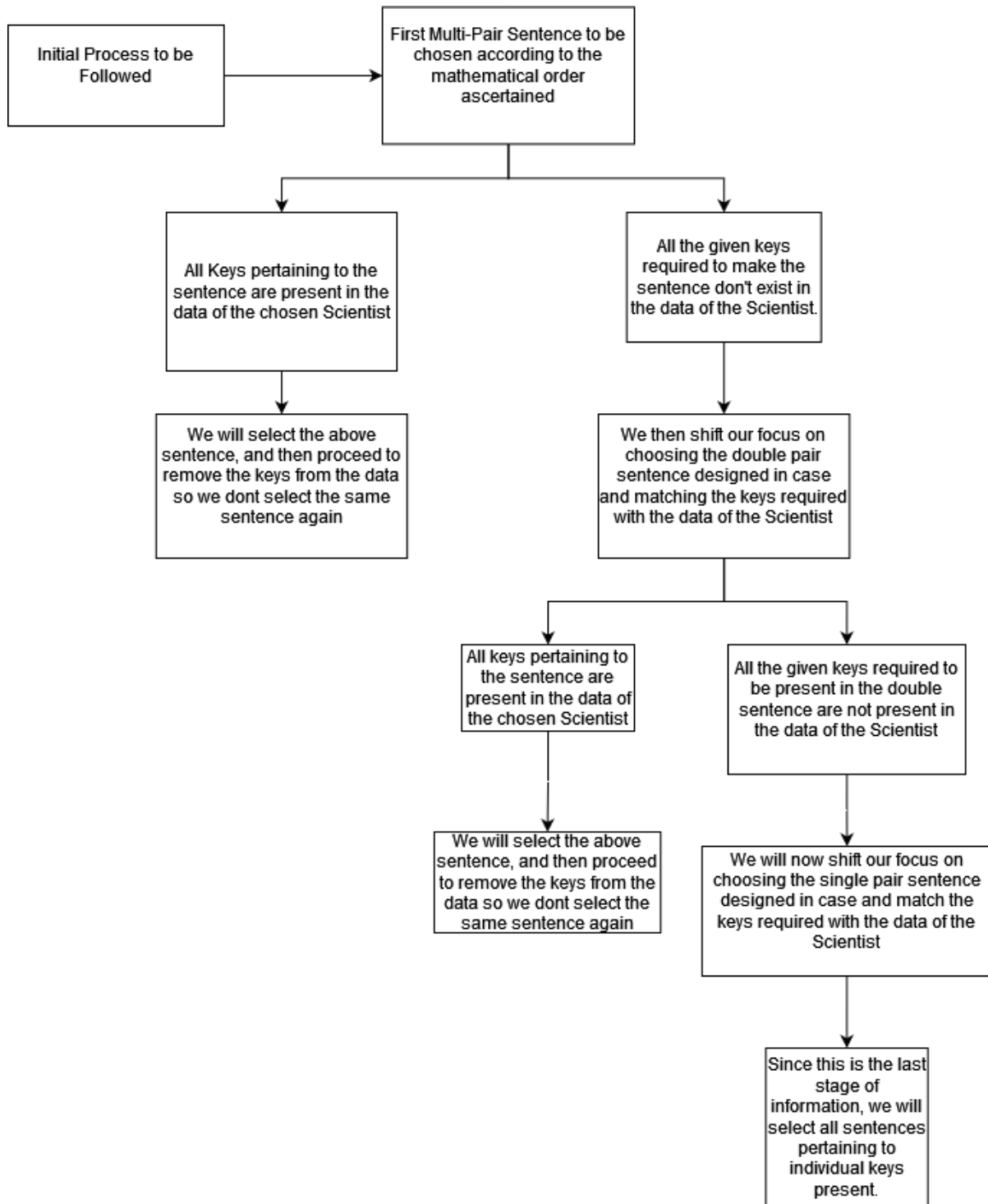


Figure 4.5 The entire model pipeline followed to generate the template sentences.

Chapter 5

Error Analysis & Evaluation

5.1 Introduction

In the preceding chapter, we outlined the methodology employed to address the challenge of generating template sentences for the Scientific domain in Hindi. Our approach involved combining relevant keys, delving into the intricacies of key combinations, addressing gender nuances in Hindi, and augmenting the Hindi Wikipedia page with additional features. Although seemingly straightforward, we encountered several errors and challenges along the way, which we promptly resolved. Throughout this process, we made noteworthy observations, which we will share in this chapter.

Furthermore, we devised an evaluation framework to compare our output with the standard pre-existing Wiki Inter-Language Translation System (Machine Translation). Unlike automated-driven evaluation schemes like BLEU [14] or ROGUE scores [11], we opted for a manually-driven evaluation approach. This decision was motivated by the understanding that assessing a system’s usefulness and user satisfaction surpasses the fulfillment of predetermined requirements or specifications. As our objective entails enriching the Hindi Wikipedia and catering to a broader audience, comprehending public reception and sentiments towards our generated outputs becomes of paramount importance.

In this chapter, we aim to present the errors we faced, how we tackled them, and finally discuss the results of our evaluation, shedding light on how our system fares against the established benchmark. Additionally, we intend to provide insight into the users’ perceptions and satisfaction levels, as their feedback contributes significantly to gauging the system’s overall effectiveness.

5.2 Error Handling & Observations

We break this section down into the errors we faced that were unforeseen and which required extensive handling and also how we observed certain intricacies that helped us improve our techniques:

5.2.1 Observation 1

One of the main observations we noticed is again based on the gender aspect in Hindi. Hindi, as a language, needs the gender of the object in a sentence to mark certain words. In the earlier paragraphs, we had talked about Sentences in which the Subject gender mattered, but there were certain sentences in which Object gender mattered. Consider the following two examples illustrating both the gender aspects respectively:

1. {{Scientist}} {{का भाग}} का हिस्सा होने के साथ-साथ, {{का सदस्य}} का हिस्सा भी {alivestatus/wgop}
| ¹
2. {{Scientist}} {{के/की}} जीवन साथी {{(जीवन साथी)}} {alivestatus/wgok} और उनके {बच्चे/बच्चों} का नाम {{संतान}} {alivestatus} | ²

The 'alivestatus/wgop' placeholder in Sentence 1 means it depends on the gender of the person, hence 'wgop,' and in Sentence 2, the 'alivestatus/wgok' placeholder refers to the gender of the key (which means object). Now, let us look into each example in a little detail. In Sentence 1, we see that the 'alivestatus/wgop' placeholder will be replaced with either 'थे' or 'थी' depending on whether the Scientist we talk about is Male or Female, but in Sentence 2, we see that the alivestatus/wgok or the '{के/की}' placeholder is independent of the gender of the Scientist. Instead, it depends on the gender of the '{{जीवन साथी}}' key. Thus, if the gender of '{{जीवन साथी}}' key turns out to be male; we will use 'थे' or if the person is a female, we will use 'थी' . We also were able to write sentences that did not need the gender of the subject. For example:

1. उनको {{दिया गया नाम}} के नाम से भी जाना जाता {alivestatus} | ³

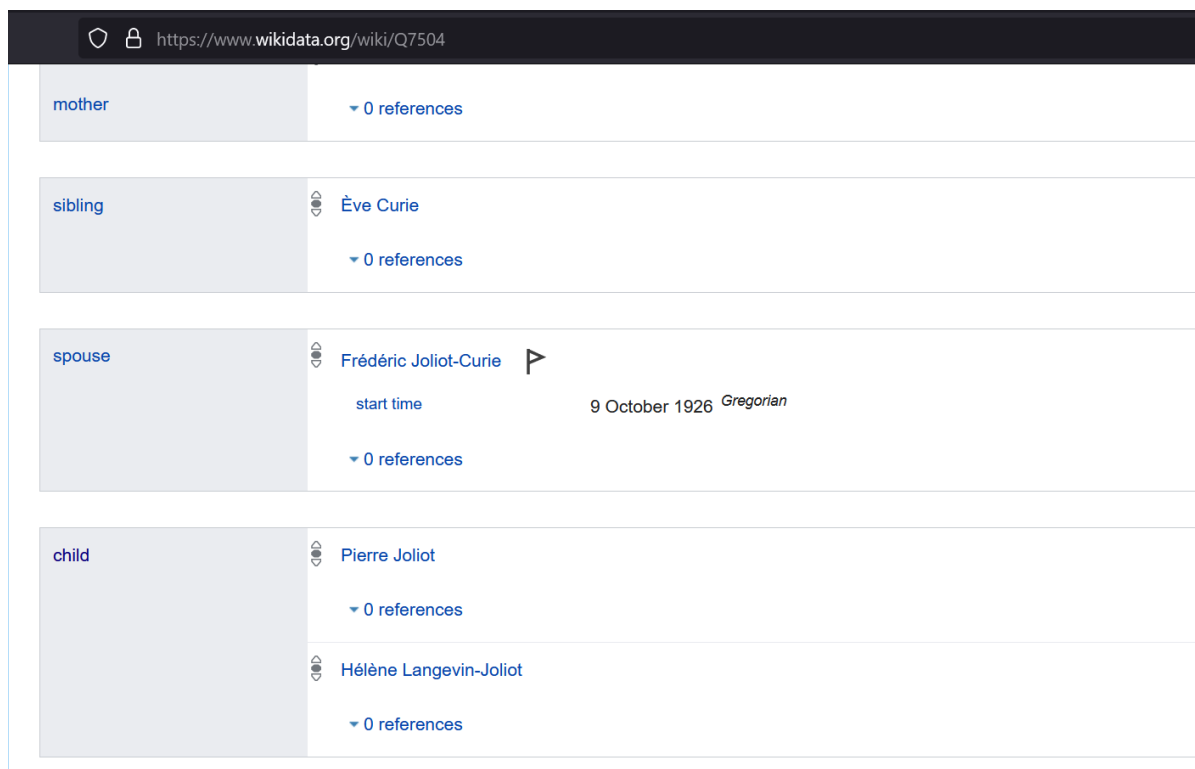
¹English Translation: Apart from being a part of {Part of}, he/she was also a part of {member}

²English Translation: {Scientist}'s husband/wife was {Spouse} and his/her children's name was/were {Children's name}

³He/She was also known by the name of {{Given Name}}

Here, the 'alivestatus' placeholder will be 'है', 'था', 'थे' or 'थी' depending on whether the Scientist is alive or not. It is independent of the gender of the Scientist.

Based on the above observation, we had to very carefully write the template sentences, keeping in mind the requirements based on the sentence. If there was a gender requirement from either the subject or object or even if it was independent of the subject & object, we needed to accommodate the sentence ending accordingly.



The screenshot shows the Wikidata page for Q7504. The page displays a list of relationships (properties) and their corresponding values (objects). The relationships shown are 'mother', 'sibling', 'spouse', and 'child'. The 'spouse' relationship is highlighted, showing the value 'Frédéric Joliot-Curie' with a start time of '9 October 1926' in the 'Gregorian' calendar. The 'child' relationship shows two values: 'Pierre Joliot' and 'Hélène Langevin-Joliot'.

Property	Value(s)	References
mother		0 references
sibling	Ève Curie	0 references
spouse	Frédéric Joliot-Curie start time: 9 October 1926 <i>Gregorian</i>	0 references
child	Pierre Joliot Hélène Langevin-Joliot	0 references

Figure 5.1 A screenshot that shows the spouse information mentioned in Wikidata for which we needed the gender to decide the appropriate sentence ending for the template sentence.

5.2.2 Observation 2

Based on the above nuances, we were not able to obtain the gender of some objects because that information was missing from the Wikidata Pages. A Wikidata page of a specific individual can tell you all information about the Subject, but the value of the key about the individual, more specifically the object, has no separate link to another Wikidata Page where we can fetch the gender information. Hence, we decided to employ some Python modules like gender-guesser [13] that can guess the gender based on pre-trained data.

This was, of course, not 100% accurate, and hence, we were unable even then to get the gender information for some objects. For clearly identifiable ones, we employed the codebase, but for those that showed "unknown," we decided to leave the placeholders in the format of थे/थी and when the user reads the sentences, he/she can use either of the placeholder words depending on the gender of the object, which they can find out using other resources and world knowledge.

5.2.3 Observation 3

When we downloaded our Wikidata dump using SPARQL queries, and when we parsed through the data, there were some pre-existing Hindi Labelled Key-Value pairs, which we used, but there were some English Key-Value Pairs for which we had to transliterate/translate depending on the requirement (Refer Section 3.2). We thus used the Anuvaad ⁴ to help us transliterate and translate the English Key-Value Pairs with excellent accuracy. When we walked through the values for every key of the Scientist, we noticed that it was not necessary that the value would be a single unit. We noticed keys like Award Received, Occupation, Names of their Children, etc., have multiple values, so we needed to accommodate all of them in our template sentences, but since we had made only one placeholder for all the keys, we had to improvise and combine all the values barring the last value, using the delimiter ',' and then use और to combine the last value of the list with the rest of the combined values.

Let us explain the above with the following examples:

1. {{Scientist}} को {{पुरस्कार प्राप्त}} से सम्मानित किया गया और उन्हें {{नामांकित किया गया}} के लिए नामांकित भी किया गया {alivestatus} | ⁵
2. {{Scientist}} एक प्रसिद्ध {{व्यवसाय}} {alivestatus/wgop} जिनका जन्म {{जन्म तिथि}} को {{जन्म स्थान}} देश में हुआ {alivestatus} | ⁶

Let us look at the example sentence 1. As we can see in the template sentence, we have made a single placeholder for 'पुरस्कार प्राप्त' but in reality, if we were to write the template sentence in a Hindi Wikipedia page for, let's say, Irene Curie as an example, we would see that in her Wikidata page from Figure 5.2, during her lifetime, she won multiple awards and was even nominated for a lot of awards. There was no way to include so many awards using multiple brackets because it would unnecessarily make the sentence dependent on each Scientist, and

⁴<https://anuvaad.org/>

⁵{{Scientist}} was nominated for {{Nominated Awards}} and was awarded {{Awards List}}

⁶{{Scientist}} was a famous {{Occupation}} who was born in {{Place of Birth}} on {{Date of Birth}}




award received		Officer of the Legion of Honour	
		point in time	1939
		▶ 1 reference	
		Nobel Prize in Chemistry	
		point in time	1936
		together with	Frédéric Joliot-Curie
		prize money	79,958 Swedish krona
		award rationale	in recognition of their synthesis of new radioactive elements (English) såsom ett erkännande för deras gemensamt utförda syntes av nya radioaktiva grundämnen (Swedish)
		▼ 2 references	
		reference URL	http://www.nobelprize.org/nobel_prizes/chemistry/laureates/1935/joliot-curie-bio.html
		reference URL	https://www.nobelprize.org/nobel_prizes/about/amounts/
		honorary doctor of the Jagiellonian University of Krakow	
		point in time	1951
		▶ 1 reference	

Figure 5.2 The list of awards won by Irene Curie which we needed to combine when writing the sentence pertaining to the awards she won during her lifetime.

one would have to create multiple template sentences depending on the Scientist. To avoid this situation, we decided to combine all the awards, and the example sentence for Irene Curie becomes

1. इरेने जोएलट-कुरी को "नए रेडियोधर्मी तत्वों के उनके संश्लेषण की मान्यता में" के लिए रसायन शास्त्र में नोबेल पुरस्कार से सम्मानित किए जाने के साथ-साथ मतेउची मेडल, विज्ञान को मेरिटायर्ड सर्विस के लिए बर्नार्ड मेडल, पोलोनिया रेस्टिटुआ के आदेश के स्टार के साथ कमांडर, ग्रैन्वल्ड के क्रॉस का आदेश, 3 वीं कक्षा, सेना के अधिकारी सम्मान के, मारिया क्यूरी-स्कलोडोवास्वा विश्वविद्यालय के मानद डाक्टर और क्रावो के जगेलोनियन यूनिवर्सिटी के मानद डाक्टर से सम्मानित किया गया | ⁷

⁷Irene Jolot-Curie was awarded the Nobel Prize in Chemistry "in recognition of her synthesis of new radioactive elements", in addition to being awarded the Matteucci Medal, the Bernardi Medal for Meritorious Service to Science Medal, Commander with Star of the Order of Polonia Restitua, Order of the Cross of Grundwald, 3rd Class, Army of the

We applied the same logic to all such keys that had multiple values associated with them, and two such examples are given above. We can see that this approach enables us to combine all the values into one single entity and use that as a placeholder for a template sentence. Now, this template sentence could be used for all scientists covering all aspects of multiple values associated. This observation, and consequently the approach, helped us immensely in collating the multiple values data and making the sentence complex and rich in Hindi.

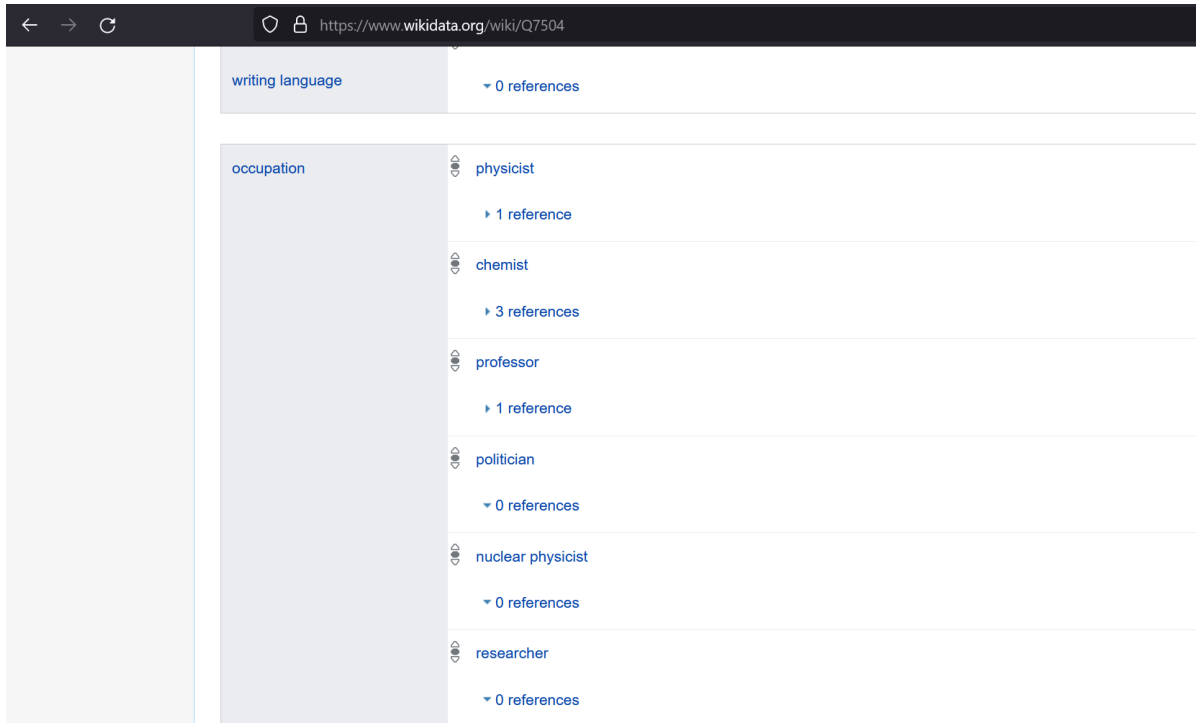


Figure 5.3 Another screenshot that displays the multiple occupations for an example Scientist on Wikidata, which were required to be combined to make the template sentence.

5.2.4 Error Analysis

During the vast amount of preprocessing we did on the 17000 Scientists' data, there were a lot of duplicate names of Scientists that had the same occupation, along with certain other keys, and hence our QIDs (Unique for every item in Wikidata) faced a mismatch, and our code ended up referring QIDs to the wrong recipient. This error was unforeseen as the Occupation Key had been mapped using SPARQL, and we were now facing an information mismatch for some Scientists. To resolve the same, we reiterated through all Scientists and re-ran the QWikidata

Official Honors, Honorary Doctor of Maria Curia-Skłodovas University and Jagiellon Niyan of Kraków Awarded Honorary Doctorate of the University.

Module to update any mismatched QIDs and were able to resolve the issue. For specific keys like the Number of Children, Wikidata stored information in an extremely complicated structure, and our code was unable to parse the data. Without the information, we would not be able to make the correct sentence. To resolve this, we had to improve the code to delve even deeper into the codebase of Wikidata, and finally, after using specific libraries to improve our efficiency, we were able to parse the data to get the desired information.

5.3 Results

After processing the above observations and correcting the errors, we were successful in generating Hindi Wikipedia pages for scientists who did not have one, despite having pages in other languages. The sample template Hindi Wikipedia page is publicly available at this link⁸. We compiled all available information on each scientist and incorporated it into their respective Hindi Wikipedia pages.

We have also created a valuable resource in the form of a dataset consisting of 17,000 entities. The dataset is divided into 1,700 files, each containing information on 10 scientists presented as key-value pairs under their respective names. The dataset and the corresponding code can be found at this link. During this entire process, we have created a method wherein a user can enter any Scientist name from our 17,000 rich Scientist dataset, and our code and algorithms will automatically generate the corresponding Hindi Wikipedia Page for that Scientist. The corresponding page will contain all the data available on Wikidata for the same scientist, and the sentences will be arranged coherently with the rule that the most complex sentence according to the number of keys and other intricacies as discussed in section 4.2.4.2 will occur first. In the next section, we demonstrate how we evaluated our work by comparing it to existing machine translation outputs from English to Hindi.

5.4 Evaluation

To evaluate our work, we enlisted 20 English-Hindi bilinguals and provided them with 2 sets of 50 articles each of 50 scientists. One set is machine-translated using Wikipedia’s in-built translator, while the second set was created using our template approach. Each Scientist has been vetted by 3 different workers. We then did a comparative analysis by creating a survey that hinged on 4 key points on a scale of 1-5. These points were based on the word level, sentence level, discourse level, and overall level of the articles, and the results were tabulated. Figure 5.4 and Figure 5.5 show the questionnaire for the research survey conducted.

⁸For anonymity, we have uploaded the article using an anonymous identity.

Out of the 50 articles given, 40 articles that were generated using our method received more points on the scale compared to the machine-translated articles. The following Table 5.1 displays some of these Scientists, with the last column displaying the better output between Template Driven Output & Machine Translation Output. The entire list of Scientists evaluated can be found at [this link](#). Based on the questionnaire that details the intricacies of word, sentence, and overall context level, the scores are compared, and the results show that our method has indeed produced better results in terms of readability, coherence, and structure of the articles.

Table 5.1: Final Results of Evaluation

Scientist	Output Type	Word Level Score	Sentence Level Score	Dis-course Level Score	Overall Score	Total Added Score	Better Approach
Cassiano Dal Pozzo	Machine Translation Output	2	1	5	1	9	Template Driven Approach
		2	2	4	3	11	
		2	2	3	2	9	
	Total Score	6	5	12	6	29	
	Template Driven Approach Output	4	5	4	4	17	
		4	5	5	5	19	
		4	5	5	4	18	
	Total Score	12	15	14	13	54	
John Houghton	Machine Translation Output	3	4	5	4	16	Machine Translation Output
		5	5	5	5	20	
		4	2	3	4	13	
	Total Score	12	11	13	13	49	
	Template Driven Approach Output	4	4	5	4	17	
		4	5	5	4	18	
		3	3	4	3	13	
	Total Score	11	12	14	11	48	
Yuri Mikhailovich Luzhkov	Machine Translation Output	4	5	5	5	19	Machine Translation Output
		4	5	5	4	18	
		4	5	5	4	18	
	Total Score	12	15	15	13	55	
	Template Driven Approach Output	4	3	5	4	16	
		5	5	5	5	20	
		2	5	5	3	15	
	Total Score	11	13	15	12	51	
Nicolas Cabrera	Machine Translation Output	4	3	3	3	13	Template Driven Approach
		4	4	3	4	15	
		4	3	3	3	13	
	Total Score	12	10	9	10	41	
	Template Driven Approach Output	3	4	4	4	15	
		3	2	4	2	11	
		3	5	3	5	16	
	Total Score	9	11	11	11	42	

Continuation of Table 5.1							
Scientist	Output Type	Word Level Score	Sentence Level Score	Dis-course Level Score	Overall Score	Total Added Score	Better Approach
Harrison Schmitt	Machine Translation Output	1	2	1	1	5	Template Driven Approach
		2	1	1	2	6	
		1	4	4	3	12	
	Total Score	4	7	6	6	23	
	Template Driven Approach Output	3	2	3	3	11	
		2	3	3	3	11	
		2	5	3	4	14	
	Total Score	7	10	9	10	36	
Leon Rosenfeld	Machine Translation Output	3	3	5	3	14	Template Driven Approach
		4	3	4	4	15	
		2	5	4	5	16	
	Total Score	9	11	13	12	45	
	Template Driven Approach Output	4	5	4	5	18	
		5	4	4	5	18	
		4	5	3	4	16	
	Total Score	13	14	11	14	52	
Benjamin Thompson	Machine Translation Output	4	4	4	4	16	Template Driven Approach
		4	4	4	4	16	
		4	4	5	3	16	
	Total Score	12	12	13	11	48	
	Template Driven Approach Output	3	5	5	4	17	
		4	5	5	4	18	
		3	5	5	4	17	
	Total Score	10	15	15	12	52	
Robert Lucas	Machine Translation Output	5	4	5	4	18	Template Driven Approach
		5	3	4	3	15	
		3	1	2	1	7	
	Total Score	13	8	11	8	40	
	Template Driven Approach Output	4	5	5	5	19	
		3	3	4	3	13	
		3	5	5	4	17	
	Total Score	10	13	14	12	49	
End of Table							

Research Survey

Instructions: You will be given 2 Sheets for every Scientist(One will be a Machine Translated Copy, and one will be my method). You have to read both the sheets and answer the following questions. Do mention any feedback you might have.

- Please rate the following information on a scale of 1 to 5, with 5 being “Most Accurate/Best” and 1 being “Least Accurate/Worse”. Please tick the appropriate box.



- Sheet 1

1. Rate the paragraph in Sheet 1 based on correct word usage, including spelling and words such as है, थे, था, or थी.

1	2	3	4	5
---	---	---	---	---

Feedback(If any)(Please write N/A if no feedback)

2. Rate the paragraph in Sheet 1 based on Sentence Level Fluency, i.e. (If the Words in the sentence indeed go in the order as specified - Also known as word order)

1	2	3	4	5
---	---	---	---	---

Feedback(If any)(Please write N/A if no feedback)

3. Rate the paragraph in Sheet 1 based on coherence, i.e., When you read the sentences, is there coherence between each? Does one follow the other naturally?

1	2	3	4	5
---	---	---	---	---

Feedback(If any)(Please write N/A if no feedback)

4. Rate the paragraph in Sheet 1 based on overall Grammar, Fluency, Coherence, Structure, and how well the information is understandable and able to tell the whole picture.

1	2	3	4	5
---	---	---	---	---

Feedback(If any)(Please write N/A if no feedback)

Figure 5.4 The 1st page of the Research Survey circulated to evaluate our approach over the traditional Machine Translation approach

- Sheet 2

1. Rate the paragraph in Sheet 2 based on correct word usage, including words such as है, थे, था, or थी.

1	2	3	4	5
---	---	---	---	---

Feedback(If any)(Please write N/A if no feedback)

2. Rate the paragraph in Sheet 2 based on Sentence Level Fluency, i.e. (If the Words in the sentence indeed go in the order as specified)The instructors were courteous, and they treated each person with dignity and respect.

1	2	3	4	5
---	---	---	---	---

Feedback(If any)(Please write N/A if no feedback)

3. Rate the paragraph in Sheet 1 based on coherence, i.e., When you read the sentences, is there coherence between each? Does one follow the other naturally?

1	2	3	4	5
---	---	---	---	---

Feedback(If any)(Please write N/A if no feedback)

4. Rate the paragraph in Sheet 1 based on overall Grammar, Fluency, Coherence, Structure, and how well the information is understandable and able to tell the whole picture.

1	2	3	4	5
---	---	---	---	---

Feedback(If any)(Please write N/A if no feedback)

Figure 5.5 The 2nd page of the Research Survey circulated to evaluate our approach over the traditional Machine Translation approach

Chapter 6

Conclusions and Future Work

6.1 Conclusion

Thus, we present a study on creating Hindi Wikipedia Pages in the Scientific person domain for those Scientists who do not yet possess a Hindi Wikipedia page by creating template sentences from information collected via Wikidata that serve to fill in information for any Scientist. We also present a rule-based system that works by analyzing various aspects and syntax present in the Hindi language to order accurately and present information in a sophisticated, meticulous, and articulate manner. Reading about a Scientist in their mother tongue will help a reader in their acquisition of scientific knowledge. Figure 6.1 presents such an article that has also been published on the Hindi Wikipedia.

The method works on the principle of data collection, cleansing the data, formulation, and a deep dive analysis into the intricacies involved in data present in Wikidata. The refurbished and cleaned dataset is then used to identify top frequently and most important keys without which any Scientist Wikipedia Page is incomplete. Once identified, we proceed with creating sentences based on these keys and their respective values keeping in mind the rules of the Hindi language. These sentences then will be filled in with different information for different Scientists based on a rule-based method.

Using our evaluation methods, we have been able to demarcate a significant difference in the outputs achieved using our method over the existing Wikipedia’s internal translation system. We also have been instrumental in creating a 17,000 datapoints dataset that can serve as a starting point for multiple other research projects to improve Natural Language Processing.



Figure 6.1 The final generated Hindi Wikipedia page on Benjamin Thompson published on Wikipedia

6.2 Future Work

We recognize that there is a significant lack of Wikipedia pages in other Indian languages, such as Tamil, Telugu, and Gujarati, among others. We believe that our methodology can be extended to these languages if the appropriate data is available, pre-processed per our code requirements, and, if necessary, the template sentences are written or transliterated from Hindi to the target language. We also anticipate that a Table-To-Text machine learning model can be applied to the dataset to generate articles in the required language, which would speed up the process even further. This would eliminate the need to create template sentences as the model would automatically generate the article based on the information in the dataset. However, these generated articles would still require manual vetting due to learning bias. In addition to creating Wikipedia pages, we believe that our dataset can be utilized for various linguistic tasks, such as enhancing current machine translation tasks and improving natural language generation models since we provide preprocessed data for 17,000 entities.

Related Publications

- Aditya Agarwal and Radhika Mamidi. 2023. Automatically Generating Hindi Wikipedia Pages Using Wikidata as a Knowledge Graph: A Domain-Specific Template Sentences Approach. In Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing, pages 11–21, Varna, Bulgaria.

Bibliography

- [1] S. Banerjee and P. Mitra. Wikiwrite: Generating wikipedia articles automatically. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16*, page 2740–2746. AAAI Press, 2016.
- [2] O. Dhariya, S. Malviya, and U. S. Tiwary. A hybrid approach for hindi-english machine translation. *2017 International Conference on Information Networking (ICOIN)*, pages 389–394, 2017.
- [3] C. Dong, Y. Li, H. Gong, M. Chen, J. Li, Y. Shen, and M. Yang. A survey of natural language generation. *ACM Computing Surveys*, 55(8):1–38, Dec. 2022.
- [4] M. Dorash. Machine Learning vs. Rule Based Systems in NLP - Friendly Data - Medium. 4 2018.
- [5] Z. Guo, Y. Zhang, Z. Teng, and W. Lu. Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association for Computational Linguistics*, 7:297–312, 2019.
- [6] A. Hogan, E. Blomqvist, M. Cochez, C. D’amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann. Knowledge graphs. *ACM Computing Surveys*, 54(4):1–37, July 2021.
- [7] A. Khandelwal, S. Swami, S. S. Akhtar, and M. Shrivastava. Gender prediction in english-hindi code-mixed social media content: Corpus and baseline system. *Computación y Sistemas*, 22, 12 2018.
- [8] M. Klang and P. Nugues. Pairing wikipedia articles across languages. In *Proceedings of the Open Knowledge Base and Question Answering Workshop (OKBQA 2016)*, pages 72–76, 2016.
- [9] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.

- [10] D. Li, M. Rastegar-Mojarad, R. Elayavilli, Y. Wang, Y. Yu, S. Mehrabi, N. Afzal, S. Sohn, Y. Li, and H. Liu. A frequency-filtering strategy of obtaining phi-free sentences from clinical data repository. 09 2014.
- [11] 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.
- [12] 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.
- [13] B. Onikoyi, N. Nnamoko, and I. Korkontzelos. Gender prediction with descriptive textual data using a machine learning approach. *Natural Language Processing Journal*, 4:100018, 2023.
- [14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL ’02, page 311–318, USA, 2002. Association for Computational Linguistics.
- [15] R. Perera and P. Nand. Recent advances in natural language generation: A survey and classification of the empirical literature. *Computing and Informatics*, 36:1–31, 01 2017.
- [16] Y. Pochampally, K. Karlapalem, and N. Yarrabelly. Semi-supervised automatic generation of wikipedia articles for named entities. *Proceedings of the International AAAI Conference on Web and Social Media*, 10(2):72–79, Aug. 2021.
- [17] S. Qaiser and R. Ali. Text mining: Use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181, 07 2018.
- [18] K. Ramesh, G. Gupta, and S. Singh. Evaluating gender bias in hindi-english machine translation. 06 2021.
- [19] R. Rapp, S. Sharoff, and B. Babych. Identifying word translations from comparable documents without a seed lexicon. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 460–466, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).
- [20] L. F. Ribeiro, C. Gardent, and I. Gurevych. Enhancing amr-to-text generation with dual graph representations. *arXiv preprint arXiv:1909.00352*, 2019.
- [21] L. F. Ribeiro, Y. Zhang, C. Gardent, and I. Gurevych. Modeling global and local node contexts for text generation from knowledge graphs. *Transactions of the Association for Computational Linguistics*, 8:589–604, 2020.

- [22] M. Sadat, M. M. A. Aziz, N. Mohammed, S. Pakhomov, H. Liu, and X. Jiang. A privacy-preserving distributed filtering framework for nlp artifacts. *BMC Medical Informatics and Decision Making*, 19, 09 2019.
- [23] T. Sáez and A. Hogan. Automatically generating wikipedia info-boxes from wikidata. In *Companion Proceedings of the The Web Conference 2018*, pages 1823–1830, 2018.
- [24] C. Sauper and R. Barzilay. Automatically generating Wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 208–216, Suntec, Singapore, Aug. 2009. Association for Computational Linguistics.
- [25] S. Schamoni, F. Hieber, A. Sokolov, and S. Riezler. Learning translational and knowledge-based similarities from relevance rankings for cross-language retrieval. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 488–494, 2014.
- [26] K. Shenoy, F. Ilievski, D. Garijo, D. Schwabe, and P. Szekely. A study of the quality of wikidata. *Journal of Web Semantics*, 72:100679, 2022.
- [27] L. Song, Y. Zhang, Z. Wang, and D. Gildea. A graph-to-sequence model for amr-to-text generation. *arXiv preprint arXiv:1805.02473*, 2018.
- [28] K. Tharani. Much more than a mere technology: A systematic review of wikidata in libraries. *The Journal of Academic Librarianship*, 47(2):102326, 2021.
- [29] D. Vrandečić and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, sep 2014.