Automatically Generating Hindi Wikipedia Pages using Wikidata as a Knowledge Graph: A Domain-Specific Template Sentences Approach

Aditya Agarwal
IIIT Hyderabad
Gachibowli, Hyderabad - 500032
aditya.agarwal@research.iiit.ac.in

Radhika Mamidi
IIIT Hyderabad
Gachibowli, Hyderabad - 500032
radhika.mamidi@iiit.ac.in

Abstract

This paper presents a method for generating Wikipedia articles in the Hindi language automatically, using Wikidata as a knowledge base. Our method extracts structured information from Wikidata, such as the names of entities, their properties, and their relationships, and then uses this information to generate natural language text that conforms to a set of templates designed for the domain of interest. We evaluate our method by generating articles about scientists, and we compare the resulting articles to machine-translated articles. Our results show that more than 70% of the generated articles using our method are better in terms of coherence, structure, and readability. Our approach has the potential to significantly reduce the time and effort required to create Wikipedia articles in Hindi and could be extended to other languages and domains as well.

1 Introduction

Being one of the largest collections of Human Knowledge, Wikipedia is a widely-used, multilingual online encyclopedia that relies on volunteer contributors and an open collaboration model using a wiki-based editing system. While research has shown success in the multilingual aspect of Wikipedia, its local language pages, particularly in Hindi, are lacking. There are only **149,464** Hindi Wikipedia pages, with an average length of fewer than 500 words, compared to **54,121,996** pages in English Wikipedia.

The interlinking of language versions on Wikipedia has undergone a significant overhaul with the introduction of **Wikidata**, a unified scheme that utilizes unique numbers to identify entities and their properties. Wikidata is a collaboratively edited knowledge base hosted by the Wikimedia Foundation, offering a common source of open data under a public domain license that can be used by Wikimedia projects and others. The data in Wikidata is stored in the form of specific IDs that serve as the base for the platform. Each entity has a unique entity ID, which is a number prefixed by a letter. Items are prefixed with Q (e.g., Albert Einstein (Q937)), properties are prefixed by P (e.g., an instance of (P31)), and lexemes are prefixed by L (e.g., L1). This can be seen in [1]. The platform also includes a query service called **WDQS**[1], which allows users to run queries on Wikidata's extensive database using an RDF triple store for SPARQL[2] queries against the current data version.

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. The fundamental components of a knowledge graph are nodes, edges, and labels. Nodes represent any entity, 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.

Generating coherent and discourse-related sentence-length natural language text in different languages is now possible due to improved computing power and model capacity. However, generating multiple sentences that display coherence and relevance to a topic remains a challenge, especially in Scientific domains, with minimal research done in

[1] https://rb.gy/bv8of
[2] https://www.w3.org/TR/rdf-sparql-query/

Proceedings of Recent Advances in Natural Language Processing, pages 11–21
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_002

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 LSJbot[3] by generating longer documents that encompass all relevant information.

This paper describes 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 paper 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.

2 Related Work

Existing methods from Sauper and Barzilay (2009) 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 Song et al. (2018), Ribeiro et al. (2019), and Guo et al. (2019) focus on generating sentences, a more challenging and interesting scenario emerges when the goal is to generate multi-sentence texts. Banerjee and Mitra (2016) 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 (Le and Mikolov, 2014) that computes continuous distributed vector representations of varying-length texts. The representations are then used to identify similar articles that currently exist on Wikipedia. Rapp et al. (2012) used Wikipedia articles in nine languages to identify word translations through keywords and a word alignment algorithm. Schamoni et al. (2014) proposed to use links to retrieve Wikipedia articles in English, similar to an article in German.

To the best of our knowledge, the research conducted by Ribeiro et al. (2020) 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. These models substantially help in providing and enriching Wikipedia Pages. Although these works carry out some kind of 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.

3 Method

Our paper aims to make a Hindi Wikipedia page citing all vital information important for any domain-specific data point. We assumed that relevant information on the particular data point requested is available on the Internet but scattered among several pages.

A specific domain (Scientists) is selected, and a search for entities within that domain is conducted in the desired language (Hindi) using WDQS. This search yields a list of data points related to scientists in the specified domain, which can be downloaded in various formats, such as JSON (in this case). To make the data more easily readable, the Python libraries JSON and QWikidata are utilized. Upon decoding the data, each entity consists of two main components: the child and the child name. The child part contains the Wikidata link associated with the entity, while the child name corresponds to the name of the entity as documented in Wikidata. To extract the relevant details, the QID is separated from the child and subjected to further processing.

We have followed a four-tier process to generate the article: **Collecting Domain Specific Key-Value pairs from Wikidata**, **Preprocessing**, **Template Sentence Generation with Data Retrieval Techniques**, **Features Addition & Final Wikipedia Page Generation**. Let us look into each of

[3] https://en.wikipedia.org/wiki/Lsjbot

these in detail.

3.1 Collecting Domain Specific(Biological Sciences) Key-Value Pairs from Wikidata

Before we understand how we collected the domain-specific data, it's important 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.

After conducting research on various domains, such as animals, films, birds, and trees, we ultimately 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.

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. An image showing how an actual Wikidata page looks can be seen in Figure 1 in the Appendix. This allowed us to focus on creating Hindi Wikipedia pages for the remaining 17,000 entities within the Scientific Person domain.

3.2 Preprocessing

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. An example can be seen in Figure 2 in the Appendix. Although this process was time-consuming, we successfully obtained all the pairs for the 17,000 scientists. We then used various libraries like QWikidata to convert these key-value pairs into a human-readable format. We 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.

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% (Dhariya et al., 2017), 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 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.

3.3 Template Sentence Generation with Data Retrieval Techniques

Next, we focused on generating template sentences, but first, we identified the crucial Key-Value pairs for the Scientist Domain. Key attributes such as Doctoral Advisor, Student, Doctoral Student, Awards Won, and Field of Work were considered essential. 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.

3.3.1 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. As our data had binary values (0 or 1) for the presence or absence of a key-value pair for a scientist, we shifted our focus to document frequency. To determine the relevance of each key-value pair for a given scientist, we calculated the frequency of each key-value pair across all 17,000 scientists, dividing each frequency by the total frequency of all keys in the data. We then sorted these 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"[4] or "छात्र"[5] were important, and we decided to use the IDF concept to include such keys as well. To get rid of the other keys which did not affect the quality of the page and also those for which the frequency was extremely low, we used Frequency Filtering, which we will discuss next.

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. The objective is to avoid diluting the importance of less frequent but more meaningful words. It is indirectly employed by TF-IDF 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. We now proceed to the Template Sentence Generation section.

3.3.3 Template Sentence Generation

With the top 20-25 most significant key-value pairs in hand, we proceeded to generate template sentences. This selection 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. The upcoming paragraph elaborates on this method.

To ensure that the Wikipedia page was as informative and linguistically sound as possible, we opted to merge some of the related keys and create a sentence out of them. For instance, keys such as:

[4] Nominated for

[5] Student

{{व्यवसाय}}[6], {{जन्म तिथि}}[7] and {{जन्म स्थान}}[8]

when used separately would result in 3 different sentences like

1. वह एक प्रसिद्ध {{व्यवसाय}} थे |,[9]
2. वह {{जन्म तिथि}} को पैदा हुई थे |, [10] and
3. उनका जन्म {{जन्म स्थान}} देश में हुआ था |[11]

where {{व्यवसाय}}, {{जन्म तिथि}} and {{जन्म स्थान}} are placeholders for the respective scientist key-value pairs, but if we employ our technique, we will get one sentence that is able to tell us the same information as mentioned in the above sentences:

वह एक प्रसिद्ध {{व्यवसाय}} थे जिनका जन्म {{जन्म तिथि}} को {{जन्म स्थान}} देश में हुआ था | [12]

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

1. {{नागरिकता}}[13], {{Scientist}}, {{मातृसंस्था}}[14], and {{शैक्षिक दर्जा/उपाधि}}[15],
2. {{Scientist}}, {{कार्य स्थल}}[16], {{नियोक्ता}}[17], and {{पद पर आसीन}}[18]

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

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 | [19]
2. {{Scientist}} का कार्यस्थल {{कार्य स्थल}} {alivestatus}, और वह {{नियोक्ता}} में एक {{पद पर आसीन}} के रूप में भी कार्यरत {alivestatus/wgop} | [20]

An important thing to note here is that while making the above template sentences, we had to take care of various Hindi Syntactic Rules. For example, just 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. While it is important to consider these nuances, for the purpose of this explanation, we will temporarily set them aside and address them in a later section. For now, let us focus on the four major keys, namely: {{नागरिकता}}, {{Scientist}}, {{मातृसंस्था}}, and {{शैक्षिक दर्जा/उपाधि}} which have been embedded in Sentence 1 within double curly brackets ({{}}).

These two sentences in a Hindi Wikipedia page offer a deeper understanding of the language's complexity by conveying multiple points of information. We used this approach for all 20-25 keys and generated 11 coherent sentences that combined multiple keys. Additionally, we applied P&C concepts to create sentences with fewer complexities but multiple 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}},

[6] Occupation

[7] Birth Date

[8] Birth Place

[9] He/She was a famous {Occupation}

[10] He/She was born on {Birth Date}

[11] He/She was born in {Birth Place}

[12] He/She was a famous {Occupation} who was born on {Birth Date} in {Birth Place}

[13] citizenship

[14] Alma mater

[15] Academic Degree

[16] Work Place

[17] Employed

[18] Position of Employment

[19] {Scientist}, born in {citizenship}, was an alumnus of {Alma Mater} and went on to receive a degree in {Academic Degree}

[20] {Scientist}'s place of work was {Work Place} and she was employed at {Employer} as a {Position of Employment}

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 sounded naturally coherent and were linguistically sound, they were deemed suitable for use on the Hindi Wikipedia page. To replicate this success, we created multiple variations of the original 11 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 ($5_{C_2} = 10$ 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. 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. An image, for another example, showing the three types of sentences created can be seen in Figure 3 in the Appendix.

We now needed to consider the nuances of gender we talked about previously before creating the Wikipedia pages. We examined the Wikidata pages and found the "लिंग" [24] 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 ऑफ़ फलसफा" की डिग्री भी प्राप्त की |,[25]
2. USA में पैदा हुई "रोसिना म बैरभौम" "सटोनी ब्रूक यूनिवर्सिटी" की पूर्व छात्र थी और आगे चलके उन्होंने "डॉक्टर ऑफ़ फलसफा" की डिग्री भी प्राप्त की |[26]

As one can notice, there are stark differences in the way Hindi handles gender, with the placeholders like 'हुई', 'हुए', 'के', 'की' changing according to if the Scientist is a male or female. We coded the same for all 17000 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 Pages Generation Step.

3.4 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

[21] He/She was a citizen of {Citizenship} and he/she was an alumnus of {Alma Mater}

[22] He/She obtained a degree in {Academic Degree} and he/she was an alumnus of {Alma Mater}

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

[24] gender

[25] Born in USA, "Frank Malina" was an alumnus of "Texas A&M university" and later went on to obtain a degree in "Doctor of Philosophy"

[26] Born in USA, "Rosini M Bierbaum" was an alumnus of "Brook University" and later went on to obtain a degree in "Doctor of Philosophy"

the order of the template sentences and how the page was ultimately created.

3.4.1 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 informative. We discovered that certain keys, such as **”Award Received”** or **”Date of Birth and Date of Death”**, had values and references that linked to other Wikidata pages with valuable information. For instance, the Wikidata page for Nobel Prize was linked to the Wikidata page for the **Award Received** key and 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 these Wikidata pages.

We also decided to add the ”Alive Status” key to our data, as the Hindi language encodes a person's living or deceased status, which affects sentence endings like 'है', 'था', 'थे' or 'थी'. Building on this information, 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, 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, successfully completing the task.

3.4.2 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 (Section 3.3.3).

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 determine the order of the sentences, we used 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 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. This page can be further enriched by the Wikipedian community.

4 Results

We successfully generated 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 link[27]. 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. In the next section, we demonstrate how we evaluated our work by comparing it to existing machine translation outputs from English to Hindi.

5 Human 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. The following link [28] shows the questionnaire for the research survey conducted.

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 link displays some of these Scientists, with the last column displaying the better output between Template Driven Output & Machine Translation Output. Based on the questionnaire that details the intricacies on 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.

6 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 in accordance with 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 pre-processed data for 17,000 entities.

References

Siddhartha Banerjee and Prasenjit Mitra. 2016. Wikiwrite: Generating wikipedia articles automatically. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 2740–2746. AAAI Press.

Omkar Dhariya, Shrikant Malviya, and Uma Shanker Tiwary. 2017. A hybrid approach for hindi-english machine translation. *2017 International Conference on Information Networking (ICOIN)*, pages 389–394.

Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. 2019. Densely connected graph convolutional

[27] For anonymity, we have uploaded the article using an anonymous identity.

[28] It is ensured the linked document does not breach the anonymity clause of the double-blind review process

networks for graph-to-sequence learning. *Transactions of the Association for Computational Linguistics*, 7:297–312.

Quoc V. Le and Tomás Mikolov. 2014. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Reinhard Rapp, Serge Sharoff, and Bogdan Babych. 2012. 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. European Language Resources Association (ELRA).

Leonardo FR Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing amr-to-text generation with dual graph representations. *arXiv preprint arXiv:1909.00352*.

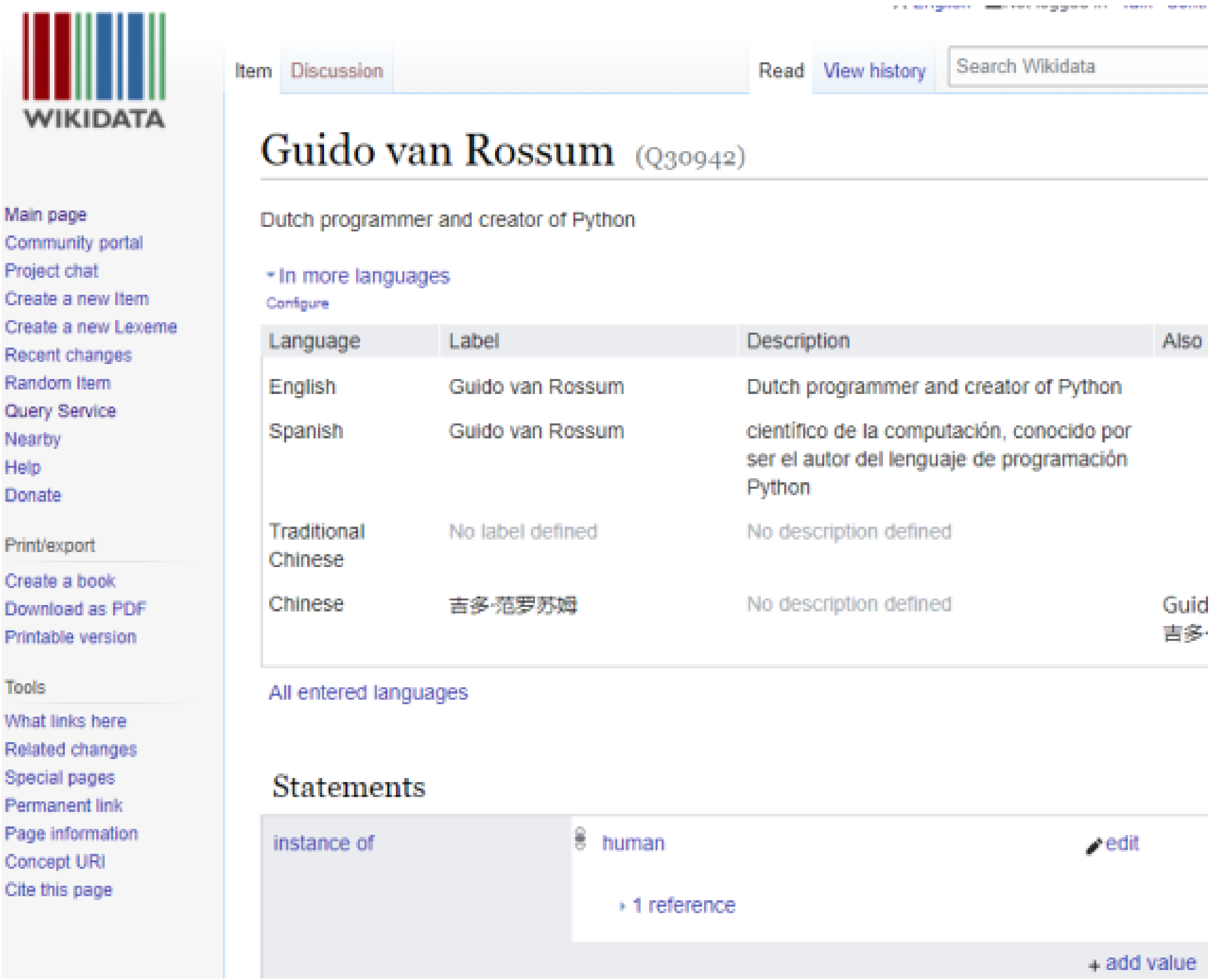
Leonardo FR Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. 2020. Modeling global and local node contexts for text generation from knowledge graphs. *Transactions of the Association for Computational Linguistics*, 8:589–604.

Christina Sauper and Regina Barzilay. 2009. 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. Association for Computational Linguistics.

Shigehiko Schamoni, Felix Hieber, Artem Sokolov, and Stefan Riezler. 2014. 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.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for amr-to-text generation. *arXiv preprint arXiv:1805.02473*.

A Appendix



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

Guido van Rossum (Q30942)

Dutch programmer and creator of Python

▾ In more languages
Configure

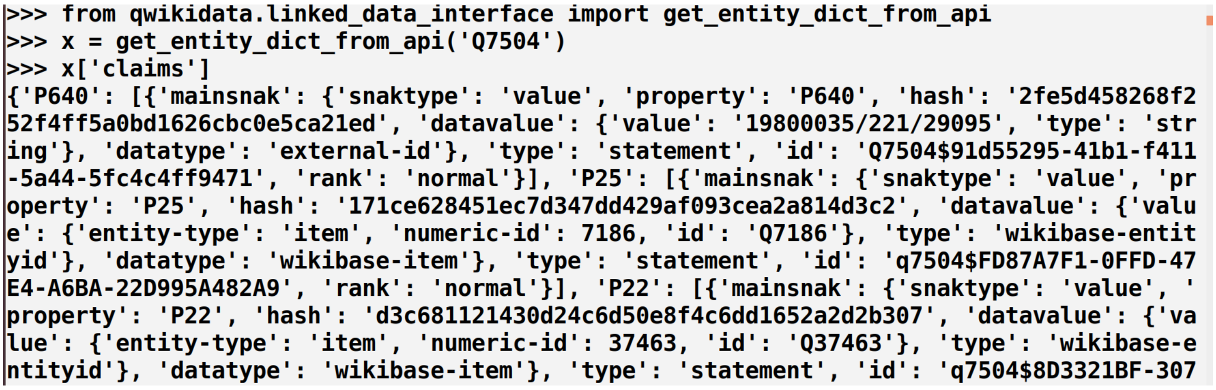
Language	Label	Description	Also
English	Guido van Rossum	Dutch programmer and creator of Python	
Spanish	Guido van Rossum	científico de la computación, conocido por ser el autor del lenguaje de programación Python	
Traditional Chinese	No label defined	No description defined	
Chinese	吉多·范罗苏姆	No description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

All entered languages

Statements

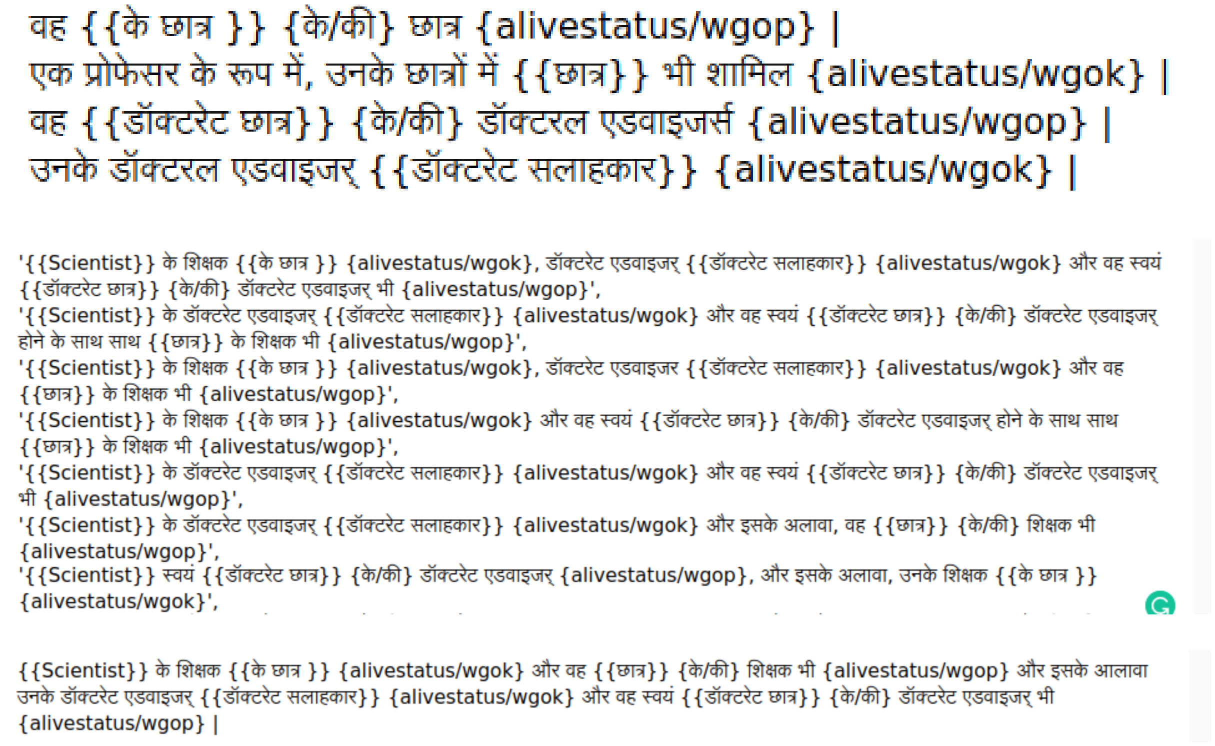
instance of	human › 1 reference	edit
		+ add value

Figure 1: This is how an actual Wikidata Page looks on the internet



```
>>> from qwikidata.linked_data_interface import get_entity_dict_from_api
>>> x = get_entity_dict_from_api('Q7504')
>>> x['claims']
{'P640': [{'mainsnak': {'snaktype': 'value', 'property': 'P640', 'hash': '2fe5d458268f2
52f4ff5a0bd1626cbc0e5ca21ed', 'datavalue': {'value': '19800035/221/29095', 'type': 'str
ing'}, 'datatype': 'external-id'}, 'type': 'statement', 'id': 'Q7504$91d55295-41b1-f411
-5a44-5fc4c4ff9471', 'rank': 'normal'}], 'P25': [{'mainsnak': {'snaktype': 'value', 'pr
operty': 'P25', 'hash': '171ce628451ec7d347dd429af093cea2a814d3c2', 'datavalue': {'valu
e': {'entity-type': 'item', 'numeric-id': 7186, 'id': 'Q7186'}, 'type': 'wikibase-entit
yid'}, 'datatype': 'wikibase-item'}, 'type': 'statement', 'id': 'q7504$FD87A7F1-0FFD-47
E4-A6BA-22D995A482A9', 'rank': 'normal'}], 'P22': [{'mainsnak': {'snaktype': 'value', '
property': 'P22', 'hash': 'd3c681121430d24c6d50e8f4c6dd1652a2d2b307', 'datavalue': {'va
lue': {'entity-type': 'item', 'numeric-id': 37463, 'id': 'Q37463'}, 'type': 'wikibase-e
ntityid'}, 'datatype': 'wikibase-item'}, 'type': 'statement', 'id': 'q7504$8D3321BF-307
```

Figure 2: A screenshot showing how Wikidata stores QID information of a Scientist and the level of pre-processing required to attain the important information.



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 एडवाइजर् {{डॉक्टरेट सलाहकार}} {alivestatus/wgok} |

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

Figure 3: A screenshot displaying the three different kinds of sentences created using P&C combinations of the keys from the main sentence at the bottom.