IndicWiki - Tech Summer Internship

Version: 2.0
Contact: (to be added)

Name of Contributors

Prasanna Saripudi
Sai Teja Kondapalli
Saimegha Kenguru
Manaswini Chintalapudi
Nirmai Aloori
Pratyusha Mayura
Praveen Garimella

Table of Contents

Goal & Purpose

Wikipedia is perhaps the most widely used resource of the Encyclopedic Knowledge, Education and E-Literacy platform on the Web. It is a free, online encyclopedia with over 7 million articles only in English. India is in second place with 117 million views per day behind the USA 189 million views as of March 2022.

90% of the people in India who can read in only their native language or Hindi are not able to benefit from this amazing tool for access to knowledge and learning. International Institute of Information Technology, Hyderabad has started the Project IndicWiki to enhance the content in Indian Language Wikipedias starting with Telugu and Hindi language Wikipedia.

The goal of this project is to generate approximately 3M wiki articles in telugu language given a particular domain of interest. Each wiki article generated should meet a minimum word count of 500 words and also have references, infobox, categories, image(optional) etc.

A systematic process - Scrum

In this internship you will be working in a team of 2 members for a particular domain. Your team will be assigned to a mentor for this project. You will be using scrum methodology for this project and the project duration is of 2 months i.e., 8 weekly sprints which include the below 4 ceremonies.

- *Sprint Planning*
    - This scrum meeting happens at the beginning of a new sprint and shouldn't last more than 2 hours.
    - This is designed to make sure the team is prepared to get the right things done every sprint. Sprint Planning allows the scrum team to answer the questions, “What can be delivered in this next sprint?” and “How will we accomplish that work?”
- *Daily Scrum (Daily Standup)*
    - This scrum meeting happens everyday and shouldn't last more than 15 minutes.
    - The Daily Scrum is the team's chance to get together, and discuss “What did you do yesterday?” “What will you do today?” and “Are there any impediments in the way?”
- *Sprint Review*
    - This scrum meeting happens at the end of the week and shouldn't last more than an hour.
    - This scrum meeting is the platform where all work completed during the sprint can be showcased to the stakeholders (mentors).
- *Sprint Retrospective*
    - This scrum meeting happens after the sprint review and shouldn't last more than 30-45 minutes.
    - This retrospective meeting is to discuss “What went well over the last sprint?” “What didn't go so well? and “What could we do differently to improve?”
    - Ultimately, this scrum ceremony should provide a blameless space for members of the team to provide their honest feedback and recommendations for improvements.

Please refer to this link for more information on the scrum methodology. Your mentors will guide you throughout the project. Slack will be used as a communication channel between you and your mentors for instance messaging and zoom for video-based conversations. The teams should update their work status to their mentors every 15 minutes on slack and the mentors validate to see if the teams are progressing well.

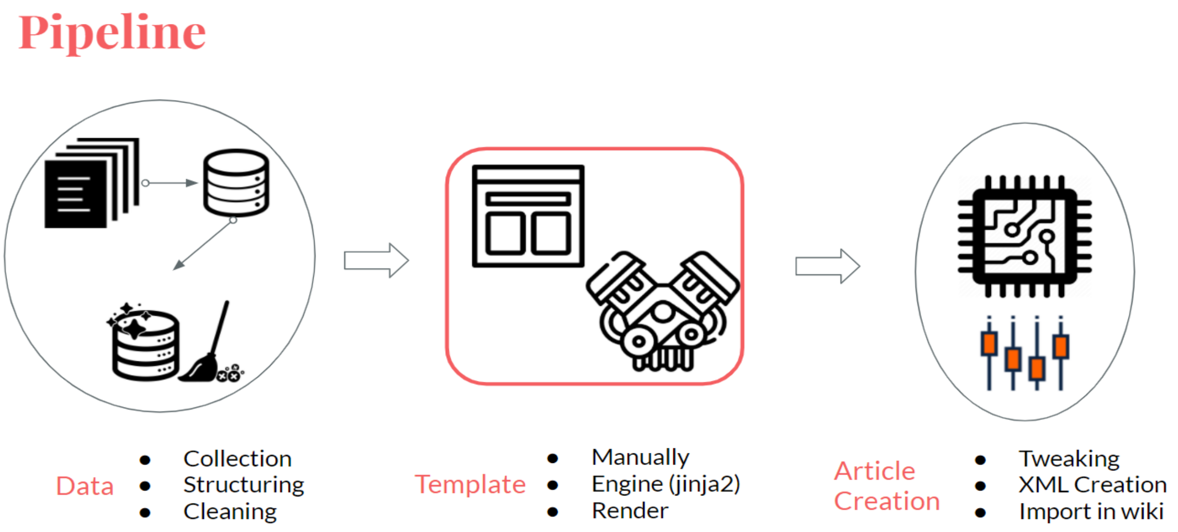
Version Control

Version control allows you to keep track of your work and helps you to easily explore the changes you have made, be it data, coding scripts, notes, etc. You are probably already doing some type of version control, if you save multiple files, such as file_v1.py, file_v2.py, etc. This approach will leave you with tens or hundreds of similar files, making it rather cumbersome to directly compare different versions, and is not easy to share among collaborators. With version control software such as Git, version control is much smoother and easier to implement. Using an online platform like Github to store your files means that you have an online backup of your work, which is beneficial for both you and your collaborators.

For this project, having a version control system is necessary as you will have to perform the tasks iteratively, it will help you to maintain versions and have control on your work. Please feel free to refer to this for a detailed explanation of version control using git.

The below sections describe the steps to be followed to generate the wiki articles.

Basic Pipeline for Article Generation:-



Pipeline

Data
- Collection
- Structuring
- Cleaning

Template
- Manually
- Engine (jinja2)
- Render

Article Creation
- Tweaking
- XML Creation
- Import in wiki

About Domains

First step would be to select/assign a domain i.e, to select a particular area of work or specific area to be worked upon. For example, colleges, schools, hospitals, temples could be domains.

Coming to the selection criteria, the domains should have adequate open data to make use of. You should also be a bit familiar with the chosen domain/ topic and that would ease the process of collecting the data as you would have an idea of what you are looking for. Even if you are not familiar with the domain, you can research it by looking at different resources like the existing wiki articles or the domain related websites to understand what goes into that domain.

Indicwiki has identified a list of domains that will be circulated to you. Your assigned mentor will discuss with you to assign a domain based on your interest.

Data Collection

The second step is to gather domain-specific data and the most critical objective of data collection is ensuring that information-rich and reliable data is collected. The approach of data collection might not be the same for all domains, it would mostly depend on the type of information/ data available for that particular domain. For example, some domains might have easily available data sources like spreadsheets, csv files, etc, while others would require scraping from different sources to form a reliable database. Irrespective of that, you will now see a generic approach that you could incorporate while collecting data.

- You need to first estimate the number of attributes (columns) required to generate an article for that domain. The number of attributes should be enough to generate a 500 word wiki article. For example, for a domain like movies, the attributes can be the year of release, title, country, language, budget, production company, plot etc.
  Note: Attribute count can vary based on the selected domain and the data.
- Start searching for data sources. These can be websites (for example, all government websites are trusted websites), API's, Spreadsheets, CSV files, PDF's, Wiki dumps, Databases, WikiData, WikiCommons etc. When a data source is found,
  - Check whether the data is meaningful and can be used for article generation. If you find it meaningful, then check for the attributes that can be used.

        - While checking for the meaningfulness of data, you could try to create sentences from the collected attributes and could discard the attributes that won't be useful for the article.
    - Extract the data.
        - Scraping libraries/ tools: Beautifulsoup, scrapy, tabula, selenium etc. These are some of the starter libraries/ tools you can make use of for scraping i.e., Beautifulsoup, Scrapy, Tabula, Selenium, etc (Please refer to the tools/ libraries reference section for more info)

      Note: Apart from the above mentioned libraries/ tools, there are many other ways to perform this task. Feel free to explore and make use of them.

        - You can also rely on using wikipedia articles, wiki db for scraping the data that you want to make use of in your dataset.
        - Incase of API's, Spreadsheets, CSV files we can use the data directly after analyzing it (which will be discussed later).
        - Images can also be collected if required for the data (WikiCommons or non-copyrighted images).
    - The above steps of searching for datasources, extracting data would repeat until you have enough attributes to meet the wiki minimum word count requirement i.e., 500 words.
- If the data is collected from multiple sources, then in order to create a unified knowledge base we merge all the data collected (based on a primary key like the ID, name, etc).
- The data can be stored as json, excel sheets or any of the key-value data types.
- After collecting the data, you need to analyze, clean the data before making use of it.
    - You need to look for missing values, duplicate entries in the dataset. You have to make sure that there are no duplicate entries. For handling the missing values, you can either remove the records with missing rows or fill the missing values depending on the data.
    - To perform Exploratory Data Analysis (EDA), we can make use of libraries like Sweetviz (open-source python library). You can refer to that here.

Scraping Tools:-

Beautiful Soup:- **https://www.crummy.com/software/BeautifulSoup/bs4/doc/**

It is a python library and it is used for scraping data out of HTML and XML files.
It lags when compared to Scrapy in terms of speed.

Issues to be taken into consideration:

- If Webpage content is available merely as strings, and hence not helpful for dynamic actions like selecting options on a drop down etc. → Selenium library was used for such cases.
- In some cases links will not be in the exact same format for every entity →
  For such cases, additional handling and conditional checks were needed in code.

***Selenium*:-** https://www.selenium.dev/documentation/
For Ajax and Pjax files(deals with Javascript elements).
It will better interact with the websites that have mouse clicks and filling forms.

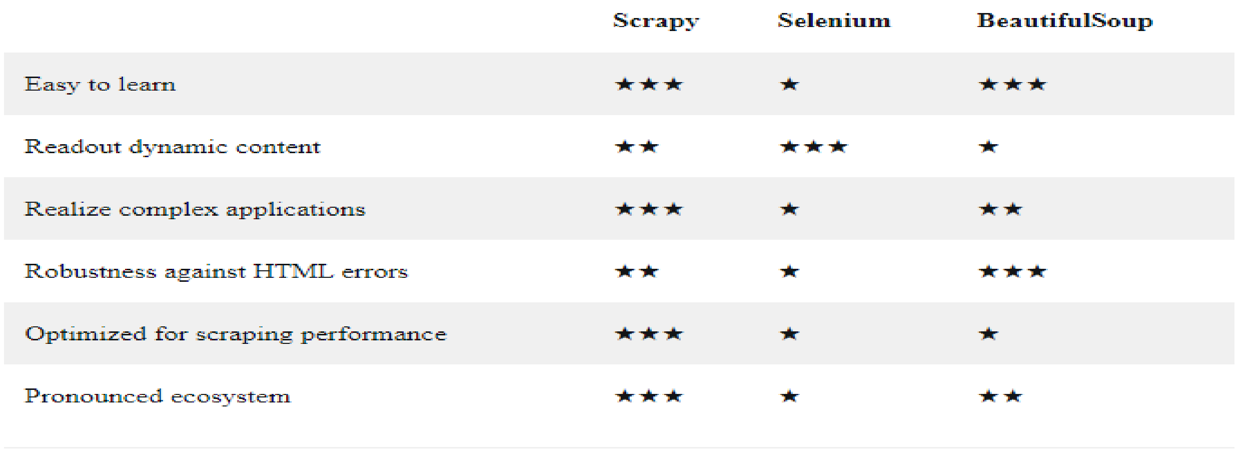
Issues to be taken into consideration:

- While facing issues regarding web drivers,
  - Sometimes it may not load pages correctly resulting in misplacing of data in the csv file
  - If dynamic actions like accessing a dropdown doesnt work, we can try scraping again in json format from a different webpage by sending requests

Scrapy:- https://docs.scrapy.org/en/latest/

It is used for large-scale data sets. Compared to beautiful soup it is faster. Multiple pages can be scraped at once.

For static and disabled javascript elements we can use beautiful soup and scrapy.
For javascript elements, selenium can be used.



	Scrapy	Selenium	BeautifulSoup
Easy to learn	★★★	★	★★★
Readout dynamic content	★★	★★★	★
Realize complex applications	★★★	★	★★
Robustness against HTML errors	★★	★	★★★
Optimized for scraping performance	★★★	★	★
Pronounced ecosystem	★★★	★	★★

Other scraping tools:

***Tabula*: https://schoolofdata.org/extracting-data-from-pdfs/**

- used to extract data from PDFs

- allows you to upload a PDF file and extract a selection of rows and columns from any table it may contain.

- Can select the portion of the PDF containing the data tables, and then easily extract the data from the tables into a CSV file or a Microsoft Excel spreadsheet.

- If you are at ease with the command line, and would like to use Tabula on a batch of similar documents, then you could use the tabula-extractor library directly. All information about this can be found here:

  https://github.com/tabulapdf/tabula-extractor/wiki/Using-the-command-line-tabula-extractor-tool

***Parse Hub:* https://www.parsehub.com/**

- It is a GUI based data extraction tool, it takes very long time to extract data and also gives no option to switch between html pages

Note:

- While scraping the data, any server shouldn't be overwhelmed by sending multiple requests. That means we have to make the crawler sleep for a certain time in order to reduce the multiple requests.

Step	**Tools/ Libraries**	**Criteria/ Description**
Data Scraping	Beautifulsoup	Python library for pulling data out of HTML and XML files
Data Scraping	Tabula	It can read tables from a PDF and convert them into a pandas DataFrame
Data Scraping	Selenium	Python library and tool used for

		automating web browsers to do a number of tasks. One of such is web-scraping to extract useful data and information that may be otherwise unavailable.
Data Scraping	Scrapy	Open-source web-crawling framework written in Python. It can also be used to extract data using APIs or as a general-purpose web crawler.
Data Scraping	Parse Hub	GUI based data extraction tool

Data Storing

- Data can be stored in format - xlsx, csv and pkl

Which format data should be stored and When?

- Csv- Using pandas library, datasets can be updated via code into the csv file.
- pkl- We pickle the data for more efficient loading and storing, which saves time.
- Xlsx- As we do manual edits in our data , it is easier to edit in this format rather than in csv format. Nevertheless, manual edits in the dataset can be mostly done with the help of Google Sheets , csv or excel files .

Template creation

Before proceeding to create a template, a sample article should be written by the team to create a structure for the template.

Sample article:

A thorough understanding of attributes is required to write a sample article. You should try generating meaningful sentences with the attributes by taking care of ordering of sentences to make a meaningful article.

- Based on the data collected, content, you can have different sections when required. For example, for a domain like movies the sections can be Plot, Cast, Production, Budget, References, etc.
- You also need to figure out how to represent the attributes data in the article. For example: Is tabular representation better than writing sentences, Should be represented as a list or bullets, etc.
- Analysis on the data collected can also be added as sentences to the sample article. For example, for a domain like schools if the data collected has numbers like x boys, y girls. Then you can perform some analysis on the data i.e., like the boys to girls ratio, comparing these stats to nationwide stats and adding them as sentences to the article.

For example, for the movies domain consider that you are trying to create an article for the titanic movie. For an attribute like ‘year of release’, the sentence formulation could be ‘టైటానిక్ అనే చిత్రం *1997*లో విడుదల అయింది’. This is just an example showing how an attribute could be used to formulate a meaningful sentence.

Now, you have to get the sample article reviewed by language experts before proceeding with the template creation.

Technology:

Jinja2 template engine is used for creating templates for domain specific Wiki articles. Jinja2 is a powerful template engine which allows users to create diverse templates. Check the Jinja documentation here, and this for installation. Generally, you can have text files for templates, but Jinja provides control for the template creation and much more uses. Hence using Jinja is a better option.

Creating a Template:

Based on the sample article created earlier, the domain expert must create a template for every target domain. This template needs to be given as an input to the rendering engine later for generating articles. The template contains domain specific attributes that you have collected whose values need to be supplied from a python program while rendering.

Details of template creation:

- Macros are to be created for organizing the template into sections according to the context. We pass the attributes relevant to that macro and use them for filling the template sentences.

- If there is only one sentence style for a datapoint in every article, then all of them would be so monotonous. Inorder to overcome this, we include randomization during the template creation. This is done by having multiple sentence variations for a single datapoint and then to choose one randomly out of them for an article.
- There should be Reflist, Categories, Infobox sections in our template that matches the wiki article style. You can refer to the hyperlinks for more info/ syntaxes on each of them.
- When required, inter wiki links/ references should be added in the article. So, all these inline references will be automatically listed under the references section using Reflist.
- Infobox contains important facts, statistics presenting a basic summary of the article. There is also an option to add an image to the infobox.
- Categories are intended to group together pages on similar subjects. Search for categories similar to the domain that you are working on and then you have to add those categories separately in the respective section of the template.
- Edge cases like missing data should also be taken care of during template creation for a datapoint by having necessary conditions.

Refer the sample templates created for hospitals, schools, ragas.

Translation/ Transliteration

Now, the task is to translate/ transliterate the data values depending on which is required. So you need to figure out what columns should be translated or transliterated. For example, if the word is the same in both the languages i.e., English, Telugu, then it is transliterated. Hyderabad → హైదరాబాద్ is transliteration and School → పాఠశాల is translation since school has its own telugu word పాఠశాల and not స్కూల్.

Some of the libraries which you can make use of at this step are Anuvaad, DeepTranslit, Google translator, etc (Please refer to the tools/ libraries reference section for more info). Feel free to explore more libraries and make use of them if needed.

Translation/Transliteration Tools:-

Deep Translit(python library):

- Issues:

    - Transliteration for some words like The, To , Into,USA,UK etc.. Wrongly transliterated words:
        - The - తె → ది
        - To - టో → టు
        - Into - ఇంటో → ఇంటు
        - USA - ఉసా → యు.స్.ఎ
        - UK - ఉక్ → యు.కె
    - These mis-transliterated words should be manually found and replaced using find and replace (VS Code GUI) Translation

Step	**Tools/ Libraries**	**Criteria/ Description**
Translation/ Transliteration	Bing Translator	Online microsoft translation tool(translation character limit-2M)
Translation/ Transliteration	DeepTranslit	For better transliteration of Indic languages
Translation/ Transliteration	Google_translator	Free and unlimited python API for google translate
Translation/ Transliteration	Anuvaad	Open-source translation models for Indic languages

Other Tools used in Corner Cases:-

- ***google.transliteration*** - Invoked only in rare cases where deeptranslit produced errors.
    - It produced errors where there were symbols like hyphen (‘-’) etc in the string to be transliterated.
    - It converted numbers to telugu while transliterating, which wasn’t ideal for Readability.
- ***google_trans_new*** **-** It is used for better translation but only for a limited number of strings.
- If stills errors persist, we can use ***translators.google*** and ***deep_translator***

Article Generation

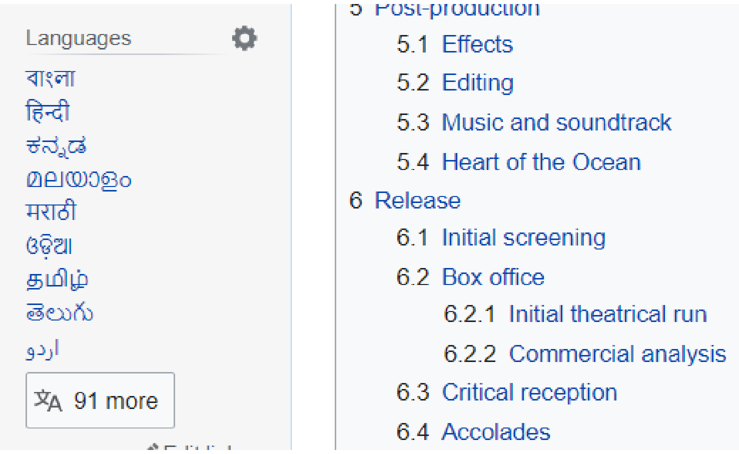
For generating articles, you need to have the data collected and the template created. For a particular record (single row), after fetching the column details as key-value pairs and importing the template, you should be rendering the template with the record details to generate an article for that record.

So for generating an article and to push that to the IndicWiki sandbox you have the below two routes in front of you.

- Using pywikibot to create articles.
- Generate xml files which are ready to be imported in mediawiki. For a detailed explanation of that, look at the steps below.
    a. If you goto special pages option in mediawiki, you will see the export pages option.
    b. Click on that and input the titles of a couple of pages existing in mediawiki.
    c. You will be able to download an XML file by doing this. In the XML file, you will see the header and the format of how a page is stored. You have to replicate it for new pages that you are creating and generate your own XML file with the information of new pages that you want to create.
    d. After that, you can import this new xml file in mediawiki (again an option in special pages) and all the articles will be created in mediawiki.

Future domain article development

As part of further developing the article, one of the enhancements is to link the Telugu article generated to the respective English Wikipedia article as shown below.



Languages

বাংলা

हिन्दी

ಕನ್ನಡ

മലയാളം

मराठी

ଓଡ଼ିଆ

தமிழ்

తెలుగు

اردو

文A 91 more

5 Post-production

5.1 Effects

5.2 Editing

5.3 Music and soundtrack

5.4 Heart of the Ocean

6 Release

6.1 Initial screening

6.2 Box office

6.2.1 Initial theatrical run

6.2.2 Commercial analysis

6.3 Critical reception

6.4 Accolades

Inorder to achieve this, we need to follow these basic steps:

- Collect the existing English Wikipedia links for the records in the database
- Maintain these collected English-wiki links in a separate column in the current database
- Use the english wikipedia links collected [if present for a record] for linking the respective generated telugu articles.

Another such is to use the already maintained data of existing english wikipedia links for the records in the database and import content from the existing english wikipedia links. The content being infobox, images, categories, data, etc.

Reference for the work done

During the first summer tech internship, multiple teams have worked on different domains and all their work is detailedly documented and provided in the github repositories in the below organization. This could provide a reference for the work that was already done using the same pipeline.
Github organization - https://github.com/indicwiki-iiit