Indicwiki Internship
Members of Parliament

Domain

The domain we worked on was **Members of Parliament**, the aim of the project being generating comprehensive articles for Telugu Wikipedia on 5000 articles, comprising details of the Members of Parliament like which constituency, which party, etc.

Team

Team Member	Email Id
Phaneendra Bejawada	phaneendra622@msitprogram.net
Suma Kola	sumakola1106@msitprogram.net

Data Collection

Sources/ Sites

We have collected data from Wikipedia articles and also the official website to gather information on the Members.

Websites:

- **Lok sabha**
- **Wikipedia**
- **Myneta**

We found out that the most appropriate data was from the official website that comprised the details of both personal life and professional life.

We have scraped the following attributes from the website:

- Name of member
- Constituency
- Party
- Lok Sabha Experience
- Gender

- Email id
- Father’s name
- Mother’s name
- Date of birth
- Place of Birth
- Marital Status
- Date of Marriage
- Spouse’s name
- Number of Daughters
- Number of Sons
- Educational Qualifications
- Profession
- Permanent Address
- Present Address

Tools used for Data collection

- **BeautifulSoup**
    - This module was used to obtain text/information from them, hence, no major issues were observed.

- **Requests**
    - This was used to get the response of the webpage, hence, no major issues were observed.
    - It is used together with BeautifulSoup.

- **Pandas**
    - Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.
    - It is used to merge, clean and store the Datasets.

- **SPARQLWrapper**
    - This is a wrapper around a SPARQL service. It helps in creating the query URI and, possibly, convert the result into a more manageable format.
    - It is used to get the data from the Dbpedia page of the members of Parliament

- **Sweetviz**
    - Sweetviz is an open-source Python library that generates beautiful, high-density visualizations to kickstart EDA (Exploratory Data Analysis) with just two lines of code. Output is a fully self-contained HTML application.
    - It is used to get the statistics of the Data found.

Infoboxes

- **InfoBox of English Wikipedia Article**
    - Using requests and BeautifulSoup we extracted the whole infobox from the english article.
    - Issue → No major issue was found.

Data Storing

- **Format #1**: Pickle, .pkl
- **Why** -
    - pkl prevents automatic conversion of null values into ‘nan’ and other default data type conversions which usually happens with .csv or .xlsx files, and is hence, the format we’ve preferred to ensure data is kept intact and unadulterated.

- **Format #2**: Comma Separated Value, .csv
- **Why** -
    - .csv is not a new/unreadable format to computers like .pkl is. Ease of access and editing of these files define why we’ve used .csv files for minor edits.

Data Cleaning

Cases taken care of

- **Data which was not useful was also scraped**
    - **Issue** : Some other values also got scraped along with the useful data for which we had to remove them manually. There were many values that got extracted with html tags which we had to remove. We tried to automate the process by finding all html tags using Python.

- **Data had some missing some values**
    - **Issue :** Data collected had some missing values. We used the Sweetviz report. We had to filter the missing values and search for the values manually.

- **Data had some duplicates**
    - **Issue :** Data collected had some duplicates. We used the Sweetviz report. We had to filter the values and delete the duplicates

- **Correcting Inconsistent/Irregular Data**
    - **Issue** : Data collected, especially Wikipedia Links was occasionally irregular in around a hundred records, owing to irregularity of Search functionality. So we found those links manually and searched for them.

Version Control

For version control of files used in the project, we used the IIIT GitHub repository, which was especially useful when tasks associated with data cleaning and editing the jinja template had to be performed, when multiple people were working on various versions of the template/data which were sequentially updated.

Sample Article

- Link to Sample article : MPS sample article

Sections

- **Approach to forming sections**

    - Initially, articles on Members of Parliament from the English wikipedia were analyzed to identify sections common to most of the members, and were identified, such as Introduction, Personal life and Professional Life. Most of these sections consist of personal and professional details of the politicians.

- **Sections and their Description**

    - **Introduction**
        - This section comprises the name of the Politician and which constituency does he belong to and also which party does he belong to.

    - **Personal life**
        - This section comprises the Personal details of the Politician like his place of birth, date of birth, father's name/mother's name, profession, education details.

    - **Political life**
        - This section contains the Loksabha experience of the politician. The party from which he belonged and also the constituency from which he was.

Jinja Template Creation

- Link to Jinja Template: Template

Edge cases

- **Making Sure all Edge Cases have been covered**
  - The first step was to prevent the rendering of sections/sentences if the attribute the sentence has been constructed on the basis of has a null value for that record. So in the initial stages of designing the template, all occurring null values in the database were identified and these were the very first edge cases added.
  - After this, at the end of every iteration of updating the sample article, and subsequently, the template, we tried rendering the article for tens of records, few with dense and few with sparse data. This would lead to the discovery of hidden edge cases which we would then include in the template, and repeat the process.

- **Possible/ Common edge cases**
  - Null Values
  - Making grammatical corrections to sentences used in the template on the basis of singular/plural value corresponding to the attribute forming it, if the attribute is numerical.
  - Making grammatical corrections to sentences based on gender like నాయకుడు for a male leader and నాయకురాలు for a female leader

- **Any edge cases rendering issues**
  - No issues have been found

Categories, References

- **How did you come up with the categories**
  - To identify categories for the domain of members of parliament, we visited existing English Wikipedia articles on the domain to obtain some ideas. On the basis of this observation, we identified the attributes that would define the category : members of Lok Sabha
  - However, we also tried to add one more category called {పార్టీ} రాజకీయ నాయకులు that also categorizes the politicians of a particular party

- **List out the categories**
    - Category : తెవికీ లోక్ సభ సభ్యులు (Tewiki Members of Lok Sabha)
    - Category : తెవికీ {పార్టీ} రాజకీయ నాయకులు({party} politicians)

- **What to keep in mind while mentioning a particular source as reference**
    - Every source from which data has been collected has been added as a reference.
    - The reference links appear at the end of the Wikipedia article, listing out all links enclosed in <ref></ref>, and these links have been appended at the end of sentences containing data obtained from that source, to add a citation. Hence, the reference tags listed at the end of the article are listed only if non-null data has been obtained from that reference link.

Infobox

- **Used an existing infobox template or created a new one ?**
    - We scraped the existing infobox from the English wikipedia article.
    - At first we planned to use the existing one but it was difficult to translate the infoboxes scraped.
    - So instead of using that we wrote a new infobox

- **Any infobox rendering issues**
    - We had one issue related to translation while rendering the infobox scraped from the English wikipedia article.
    - Some of the values are translated but some are not translated.

Translation/ Transliteration

Libraries

[List all the libraries used/explored at every stage irrespective of its success]

- DeepTranslit
    - Not Used for Transliteration
    - Issue: Did not have accurate transliterations of many independent English letters like “to”, “there”, “the”, etc. The workaround was manual correction.
    - Also As most of our data is proper nouns and as it is not that much accurate in translating proper nouns, we didn’t use that.

- GoogleTrans
    - Used for the translation of some words

    - Issue: Did not support mass translation (the limit was just 2MB can be translated at once.)

- Bing Translator
    - Used for Translation of Names, places, party names. The maximum translation had to be done using Bing because the other translators could not give proper translations/transliterations
    - Issue: Would support a translation of upto 2 million characters per account. But we need a Microsoft Azure account for using this which we don't have. So we had to use the Bing Translator without account that translated only 1000 characters at a time

Common Issues while translation, transliteration

- Incorrect Transliteration of abbreviated words, state names
    - An example of this is "AP" is not getting translated
    - Google translate was not at all helpful
    - Workaround :
        - So for these we used Bing which gave proper translations

- Incorrect Transliteration caused by English data that wasn't completely suitable for transliteration
    - An example of this is "etc." getting replaced by ඇæş
    - Workaround:
        - Manual replacement of such transliterations with meaningful Words.

XML Generation

Using the template we have created we have generated the XML for Members of Parliament.

Quality Review

As part of the quality review of the article, multiple iterations of reviewing were done to keep the quality and vocabulary of the article intact. Few instances of the quality review done are as follows -

- Better Vocabulary

This was done to replace English versions of words (or) Telugu words with accurate Telugu versions of the same.

- Using Attributes only of Significant Value (in terms of content)

The attributes which had more than 80% missing values were eliminated, as they were found to have added no value to the article.

- Keeping the article concise

Shifting the emphasis from keeping the article populated with the maximum possible word count, to keeping the article concise was one of the objectives of the quality review.