IndicWiki Summer Internship - Olympic Games

Domain

The domain we worked on was “Olympic Games”, the aim of the project being generating comprehensive articles for Telugu Wikipedia on 30,000+ Olympic Athletes, comprising all possible details of a particular athlete, such as their personal life, professional life etc., which we have been successfully able to do.

Team

Team Member	Email Id
Santosh Jonnakuti	santoshjonnakuti@gmail.com
Arjun Karthikeya	arjunkarthikeya2002@gmail.com
Divya Dosapati	divyadosapati24@gmail.com

Data Collection

Sources/ Sites

- Kaggle Dataset
  - Link - https://www.kaggle.com/datasets/mysarahmadbhat/120-years-of-olympic-history
  - Format of data available - CSV File
  - Tools used - Pandas
  - Attributes found
    - ID Unique number for each athlete
    - Name Athlete's name
    - Sex Male (M) or Female (F)
    - Age Integer
    - Height In centimeters
    - Weight In kilograms
    - Team Team name
    - NOC National Olympic Committee 3-letter code
    - Games Year and season
    - Year Integer
    - Season summer or Winter
    - City Host city
    - Sport Sport
    - Event Event
    - Medal Gold, Silver, Bronze, or NA

- Dbpedia Page of Athletes
    - Link - https://dbpedia.org/page/ followed by Athlete Label
    - Format of data available - WebPages (HTML)
    - Tools used - rdflib, SPARQLWrapper (Python Libraries)
    - Attributes found
        - Date of Birth
        - Place of Birth
        - State Of Origin
        - Thumbnail
        - Club
        - Coach
        - Education
        - Residence
        - Caption (related to Thumbnail)

Note : Athlete Label was extracted from the English wikipedia article link of that athlete. To extract wikipedia links of athletes we used requests and BeautifulSoup python libraries.

- English Wikipedia Articles
    - Link - https://wikipedia.org/wiki/ followed by Athlete Label
    - Format of data available - WebPages (HTML)
    - Tools used - requests, Wikipedia API, BeautifulSoup (Python Libraries)
    - Attributes found
        - Infobox was scraped from the English Article

Tools used for Data collection

- **Selenium**
    - An automation testing based Python library, primarily used for web scraping
    - Issues : No significant issues were encountered in the course of working with this, except for the time delay issue - sometimes, web pages would not load elements on time and that would result in timeouts, exceptions, and end of execution of the program. To avoid this, an additional method was added to keep Selenium webdriver's execution on hold, until some element would appear on the webpage.
    - But we didn't use this tool as it was not that useful as the names of athletes are proper nouns and also they are not unique, while searching

for a particular athlete we found so many others with the similar name which lead to wrong information of the athlete.

- **BeautifulSoup**
  - This was used as an additive to Selenium to navigate through HTML elements and 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 athlete.
- **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.

Images

- **Dbpedia**
  - Using SPARQLWrapper we extracted the thumbnail attribute from dbpedia. And we found around 6000+ images.
  - Issue → Since a lot of images weren't available on Dbpedia, (the number of thumbnails found is just 1/5rd the size of the database).
- **English Wikipedia**
  - Using requests and BeautifulSoup we extracted the thumbnail from the infobox of the english article.
  - Issue → No major issue was found.

- **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

- **Special Characters in Values**
    - **Issue** : Not all values (spanning across all attributes) had pure English characters, and not correcting them to their English equivalent characters would lead to a failure of translation/transliteration. These values were huge in magnitude, however, the only workaround to this was to correct them manually.

- **The String-List Anomaly**
    - **Issue** : Data enclosed in a list was intended to be collected as a list. However, such data was enclosed in a string containing the list, because data collection in .csv files does not support the list data type. The workaround to this was to use the method literal_eval() from the ast module in Python to convert such literals into actual strings, which would

however, fail if there were recurring inappropriate ‘, “ or other symbols. Hence, the final workaround was to use this method along with manual removal of apostrophes from such list based string stored attributes.

- **Filling Null Values**
    - **Issue** : There were many attributes with highly sparse data, owing to lack of information in the listed sources, such as the residence of Athlete, Education, Coach and Club, etc. The best workaround could have been using Selenium to search for such values on Google, but this didn’t work because of multiple dynamic elements in Google search results, so the final workaround was to manually search for such values and fill associated null values in the database.
- **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.

Data Merging

- Primary key - ID attribute
- Process followed, Tool used - Pandas
- Final KB format - .csv
- Final KB rows X columns - 35574 X 31681
- Final KB Link
- Issue → No major Issue found

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

- Sample Article Link

Sections

- **Approach to forming sections**

    - Initially, articles on athletes from English wikipedia were scrutinized to identify sections common to most of the athletes, and were identified, such as పరిచయం, వ్యక్తిగత జీవితము, క్రీడా జీవితం. Most of these sections consist of personal and professional details of athletes.
    - క్రీడా జీవితం was further divided into small sections with respect to olympic games (2004 ఒలింపిక్స్, 2008 ఒలింపిక్స్, ..).

- **Sections and their Description**

    - **Introduction (unnamed section)**
        - Contains basic details about the Athlete such as title, country, sport, discipline, NOC.

    - వ్యక్తిగత జీవితము
        - Contains personal details such as date of birth, place of birth, state of origin, nationality.

    - క్రీడా జీవితం
        - Contains professional details such as height, weight, coach, club and sections with respect to olympic games (2004 ఒలింపిక్స్, 2008 ఒలింపిక్స్, ..).

Jinja template creation

- Link

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 క్రీడాకారుడు for male athletes and క్రీడాకారిణి for female athletes.
- **Any edge cases rendering issues**
    - No issues have been found

Categories, References

- **How did you come up with the categories**
    - To identify categories for athletes, we visited existing Wikipedia articles on athletes to obtain some ideas. On the basis of this observation, we identified the attributes that would define the first two categories :olympic athlete, sport.
    - However, it was later understood that some infrequent attributes may also be used to categorize these articles. On the basis of suggestions we obtained from our mentors, we also added categories based on country.
- **List out the categories**
    - Category : ఒలింపిక్ అథ్లెట్
    - Category : {country} అథ్లెట్
    - Category : {sport} అథ్లెట్
- What to keep in mind while mentioning a particular source as reference
    - Every source from which data on a particular book has been collected has been added as a reference.
    - To obtain this reference link, a generalized format of every link was looked up for, such that, if any of the athlete label is appended to the link, it would redirect to a page containing details of that particular athlete. For each athlete, these generalized links with the primary key appended have been added as reference links.

Infobox

- Used an existing infobox template or created a new one ?
    - We have used the existing infobox from the English wikipedia article.
    - At first we planned to make a new one but the English wikipedia article infobox is quite long enough and it has a lot of information not only about olympics but also about other world championships.
- 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
    - Was used for the final translation.
    - Issue: Did not support mass translation (the limit was just 2MB can be translated at once.)

- Bing Translator
    - Not Used for Translation
    - 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.

Common Issues while translation, transliteration

- **Incorrect Transliteration of abbreviated words, country names**
    - An example of this is “UK” getting transliterated to ఉక్
    - **Workaround** :
        - **Translating** instead of **Transliterating** such data, if the data consisted of untranslatable words i.e. proper nouns, as any translating module would identify the difference between words to be abbreviated and words to be actually translated.
        - Performing manual corrections for values which did not contain proper nouns.
- **Incorrect Transliteration/Translation caused by non English characters**
    - As mentioned above in a lot of other sections, transliteration and translation would heavily take a hit because of non English characters used in various values.
    - **Workaround**
        - Manual identification and correction of such values.

XML Generation

Using the template we have created we have generated the XML for athletes.