Final Report Of IndicWiki Summer Internship For Movie Artists Team

---

Domain

Movie Artists is the domain we have worked on. We tried to generate around 7300 articles on different artists in which we have includes about 5000+ Indian Artists and also 2000+ Oscar Artists

Team

- Shahid Sameer - shahidsameer329@gmail.com
- P V S K Yashant - yashant.pvsk@gmail.com
- T Geethika - tiruveedulageethika@gmail.com

Data Collection

The data we have collected is Indian actors (wiki data) as well as Oscar award winner actors (through IMDB)

Sources/ Sites

- **IMDB**
  - Format of data available - tabular, text, numerical, image.
  - Tools used - Selenium, Parsehub, Beautiful soup
  - Attributes found - Name, IMDBid, Height, Alternate_Name, Star_Sign, filmo_occupation, filmo_all_film_ocupations, No_of_Awards, Year_Of_The_Award, Award_Outcome, Award_Category, Award_Description, latest_films, debut_films, occup1_num, occup2_num, occup3_num, occup4_num, occup1_films, occup2_films, occup3_films, occup4_films

- **Wiki Data**
  - Format of data available -
  - Tools used - Sparql, Selenium, Wptools
  - Attributes found- wiki_data_id, wiki_spouse_name, insta, twitterid, Father_Name, Mother_Name, Alternate_Name, Child_Name,Citizenship, wiki_siblings, Actor_POB, Work_period_start, Work_period_end, wiki_DOB, DOD, wiki_family_name, Relatives, Native_language, Languages_spoken, Birth Name, wiki_occupation, facebook id

Tools used for Data collection

- **ParseHub:**

  Used ParseHub to extract data from IMDB

  - Issue → Workaround
    - Using ParseHub we can visit only 200 pages, after we cross the 200 pages, it stopped running and asks for payment.
    - This was helpful for us in retrieving data for Oscar artists as the number of pages to retrieve the Oscar data was less than 200 (Actual count of Oscar artists=2068 | 100 per page).
    - As the number of pages for artists other than Oscar artists is more than 200 we shifted to Selenium.
  - We found Oscar artists data 100 per page using advanced search in IMDB as that is not the case for the Indian artists we had to use Selenium.

- **Selenium:**

  Used Selenium to extract data from IMDB (for Indian artists & awards for Oscar artists) and Wiki Data (for Oscar artists)

  - Issue → Workaround
    - As we couldn't find a command for extracting Oscar artists wiki data with Sparql we used Selenium but that resulted in a lot of unnecessary words/characters for the attributes. And when we are going to retrieve the data for especially Oscar artists the Tag names for the same attribute in wiki data are unique for each Oscar artist, so
    - We had to use the whole box Tag name which resulted in unnecessary data.
    - Most of the unnecessary data was cleaned by using an option in Excel for cleaning
  - Issue → Workaround
    - Extracting data using Selenium can be done with the help of HTML tags of that particular website. In the filmography section in IMDB, the Html tag names for all the occupation tables were the same , which is the reason for not being able to extract all the occupation's film data for the artists.
    - Due to the above issue we use Beautiful Soup to extract filmography data for all the artists.

- **Beautiful Soup**

  Used Beautiful Soup to extract filmography data for all the artists

  No issues faced

- **Wptools**

  Used Wptools to extract Wiki ids for Oscar artists so that we can extract the attributes data through selenium and beautiful soup

  - Issue → Workaround
    - we have to search with artist names which we got from IMDB to find wiki ids, due to this wiki id for the duplicate name were not extracted
    - As the count of the missing artists due to duplicate names was less (around <100 among 2000+ records), so we ignored those records

- **Sparql**

  Used Sparql to extract Wiki data for Indian artists

  There is no issue for Indian artists

  - Issue → Could not extract wiki data for Oscar artists as we did not find perfect command for Oscar artists, hence used Selenium for Oscar artists wiki data extraction

Images

- **Wiki Commons**

Used Sparql tool to extract images for Indian artists, Selenium for Oscar artists

- Issue → Workaround (for Indian artists)
  - While extracting images for Indian artists from sparql, we were able to extract the wiki common links, but to use them in articles we needed the wiki common name of the image.
  - Cleaned the image links in the spreadsheet. The image links had some special characters which we replaced, so we got the Wiki Commons image names.
- Issue → Workaround (for Oscar artists)
  - While extracting images for Oscar artists from Selenium, we got some extra characters and unnecessary data.
  - Most of the unnecessary data was cleaned manually and used text to column split option in Excel for cleaning a few attributes.

Data Storing

- .csv - easier to manipulate and can be edited easily
- .xlsx - As it is easy to clean the unnecessary data in the excel

- .pkl - This format is used for faster loading and better storage of versions and also Pickle will only write out any single object once, making it effective to store recursive structures ,and can serialize pretty much any python object.

Data Cleaning

Cases taken care of

- Gender neutral → As we used gender specific sentences in our articles, we found 381 Null values for gender so we manually filled gender values by searching through the Internet.
- Spouse → The spouse names for Oscar artists were multiple and their marital status was inappropriate from our data. We manually corrected them by specifying only one current spouse name, or ex spouse name.
- Removal of unnecessary data → For the Images, Twitter id, and for all data scraped through selenium from wikidata, some unnecessary characters were recognised. Cleaned them by using the text to column feature in excel.
- Data for some attributes we scraped from imdb like filmography and known for are intended to be in a list of lists and dictionaries → we needed to convert them into appropriate data types. Using a python library called ast with the help of command **literal_eval**.
- Transliterated/translated → for this database some of the single Telugu characters were wrong like the, to I, id names etc. we manually corrected them by the find and replace option in excel.

Data Merging

- Primary key - Wiki ID, Imdb ID.
- Process followed, Tool used - used primary key to merge the database.
- FinalKB format - final_data_set.xlsx format
- Final KB rows X columns - 7295 X 54
- Final KB link - [github link] -
- Storage → Huge data is not stored in a single cell, The artists like above 1000 movies so we have taken upto the extent of the row cell.
- Date formatting → The format of dates will be changing so we have to change it in the render file

Version control

- We used IIIT GitHub repository for the version control to maintain the versions and stay clear.
- Repository - https://github.com/indicwiki-iiit/Movie_artists

Sample article

Link - https://github.com/indicwiki-iiit/Movie_artists/blob/main/stats/sample%20article%20for%20movie%20artists.docx

Approach

Checked some of the artists in the wikipedia to get a clear idea about how they formed the sections like inbox and their respective career sections. Then finally decided the sections to be included in the article and divided that respective sections in the team.

Sections (Name → Description)

Infobox:

Name, Image , Birth Name, wiki Dob, Dod, place of birth, citizenship, alternate_name, spouse name, child name, Father_name, Mother_name, wiki_siblings, Signature image in the infobox section.

Intro :
Specified the artist name. mentioned their occupations, and their works with which they are popularly known for (movies).

Career:
The artists work period starts, Their first occupations debut and latest films. Their latest and debut films of other occupations with occupation specific sentences, number of films in each occupation, total no of awards as a winner, nominee. One of the award details was mentioned.

Personal life:
Birthplace, dob, dod, languages spoken, birth name, alternate_name, family name, father name, mother name,siblings , relative , spouse details are mentioned in this section.

Filmography:
All the films with year, movie name, imdb link of movie, are given in a tabular format for the occupations as mentioned in the Intro.
Mentioned the occupation of the artists with the gender specific sentence.

చిత్రం విడుదల సంవత్సరం	చిత్రం పేరు	చిత్రం ఐయండీబీ లింకు
-	లాబర్ ఆఫ్ లవ్ (Labor of Love)	లాబర్ ఆఫ్ లవ్
2021	ఓల్డ్ (Old)	ఓల్డ్
2019-2021	సర్వంత్ (Servant)	సర్వంత్

Awards:

సంవత్సరం	అవార్డు పేరు	అవార్డు వివరణ	ఫలితం
1971	ఆస్కార్ (Oscar)	బెస్ట్ ఫిల్మ్ ఎడిటిటింగ్ :టోరా! టోరా! టోరా! (1970) :షరేడ్ విత్సెంబ్రోకే జ్. హెరింగ్ :శిన్య ఇనో	పేర్కొనబడ్డారు
1951	ఆస్కార్ (Oscar)	బెస్ట్ ఫిల్మ్ ఎడిటిటింగ్ :అన్నియ్ గెట్ యూర్ గన్ (1950)	పేర్కొనబడ్డారు
1945	ఆస్కార్ (Oscar)	బెస్ట్ ఫిల్మ్ ఎడిటిటింగ్ :సిన్స్ యు వెంట్ అవే (1944) :షరేడ్ విథల్ క్. కెర్న్	పేర్కొనబడ్డారు
1940	ఆస్కార్ (Oscar)	బెస్ట్ ఫిల్మ్ ఎడిటిటింగ్ :గోన్ విథ్ ది విండ్ (1939) :షరేడ్ విథల్ క్. కెర్న్	విజేత
1971	ఎద్డీ (Eddie)	బెస్ట్ ఎడిటెడ్ ఫియేచర్ ఫిల్మ్ :టోరా! టోరా! టోరా! (1970) :షరేడ్ విత్షిణ్య ఇనో :పెంబ్రోకే జ్. హెరింగ్	పేర్కొనబడ్డారు

- Mentioned the artists name and their awards

External links: Imdb link, Twitter, Facebook, Instagram links if available.

Jinja template creation

- Link - https://github.com/indicwiki-iiit/Movie_artists/blob/main/Template/final_template.j2

Edge cases

- Covered all the edges cases possible for the artists according to the occupations and their filmography
- Occupation rendering : there are max of 10 occupations for some movie artists so we have to render and have written conditions for each and every occupation
- Gender :
    - We have covered the all edge cases and written conditions for the gender specific values for all the occupations we have taken for the template generation. We wrote sentences according to the gender of the artists

- Filmography:
    - Covered every movie for the artists and added a scroll bar for the person who has above 15 movies and set a default table for those who have 15 movies or less than that.
- Awards:
    - Same as filmography added a scroll bar for the person who have above 15 awards and default table who have 15 or less.
- Gender specific occupation:
    - Created a dictionary for both male and female to render the gender specific values into the template.
- Generalised occupations :
    - Created a common dictionary for both male and female it will not render the gender specific occupation but this will be common occupation heading for both the genders
- Secondary/unnecessary occupations:
    - We just did not include these occupations into the dictionary, giving them an empty string so that the database will remain same according to the attribute values but when we are calling the dictionary values we only take selected occupations into the article remaining will be unchanged in the database.

female specific occupations

```
'Casting director':'',
'Location management':'',
'Musician':'',
'Casting':'కాస్టింగ్ డైరెక్టర్',
'Additional Crew':'',
'Editor':'ఎడిటింగ్',
'Editorial' : 'సంపాదకీయం',
'Art director':'ఆర్ట్ డైరెక్షన్',
'Art':'ఆర్ట్ డిపార్ట్మెంట్',
'Producer':'నిర్మాణం',
'Animation':'యానిమేషన్',
'Writer':'కథా రచన',
'Actor':'నటన',
'Actress':'నటన',
'Makeup':'మేకప్ ఆర్టిస్ట్',
'Music':'సంగీతం',
'special effects':'స్పెషల్ ఎఫెక్ట్స్',
'Costume and Wardrobe':'కాస్ట్యూమ్ మరియు వార్డ్ రోబ్',
'Sound':'సౌండ్',
'Visual effects':'విసువల్ ఎఫెక్ట్స్',
'Script and Continuity':'స్క్రిప్ట్ అండ్ కంటిన్యూయిటీ',
'Composer':'సంగీత దర్శకత్వం',
'Production manager':'ప్రొడక్షన్  మేనేజర్',
'Set decorator':'సెట్ డెకొరేషన్',
'Production designer':'ప్రొడక్షన్ డిజైనింగ్',
'Second Unit Director or Assistant Director':'సహాయ దర్శకత్వం',
'Comedian':'హాస్యనటన',
'Costume designer':'కాస్ట్యూమ్ డిజైనర్',
'Stunts':'స్టంట్స్',
'Soundtrack':'సౌండ్ ట్రాక్',
'Cinematographer':'ఛాయాగ్రహణం',
'Director':'దర్శకత్వం',
```

male specific occupations

```
'Editor':'ఎడిటర్',
'Editorial':'సంపాదకీయురాలు',
'Locationmanagement':'',
'transgender female':'',
'Musician':'',
'Casting':'కాస్టింగ్ డైరెక్టర్',
'Model':'',
'AdditionalCrew':'',
'Cricketer':'',
'Politician':'',
'Soundtrack':'సంగీత విభాగంలో ప్రదర్శకురాలి',
'Cinematographer':'ఛాయాగ్రాహకురాలి',
'Director':'దర్శకురాలి',
'':'',
'Artdirector':'కళా దర్శకురాలి',
'Art':'అసిస్టెంట్ ఆర్ట్ డైరెక్టర్',
'Producer':'నిర్మాత',
'Animation':'అనిమేషన్ ఆర్టిస్ట్',
'Writer':'కథా రచయిత',
'Actor':'నటి',
'Actress':'నటి',
'Makeup':'మేకప్ ఆర్టిస్ట్',
'Music':'గాయకురాలి',
'Specialeffects':'స్పెషల్ ఎఫెక్ట్స్ ఆర్టిస్ట్',
'CostumeandWardrobe':'వార్డ్రోబ్ డిజైనర్',
'Sound':'సౌండ్ ఇంజినర్',
'Visualeffects':'విసువల్ ఎఫెక్ట్స్ ఆర్టిస్ట్',
'ScriptandContinuity':'స్క్రిప్ట్ రైటర్',
'Composer':'సంగీత దర్శకురాలి',
'Productionmanager':'ప్రొడక్షన్ మేనేజర్',
'Setdecorator':'సెట్ డెకరేటర్',
'Productiondesigner':'ప్రొడక్షన్ డిజైనర్',
'SecondUnitDirectororAssistantDirector':'సహాయ దర్శకురాలి',
'Comedian':'హాస్యనటి',
'Costumedesigner':'కాస్ట్యూమ్ డిజైనర్',
'Stunts':'స్టంట్ డైరెక్టర్',
'CameraandElectrical':'',
'Castingdirector':'',
'Transportation':''
```

Generalised occupations

```
'Editor':'ఎడిటర్',
'Editorial':'సంపాదకీయుడి',
'Locationmanagement':'',
'Musician':'',
'Casting':'కాస్టింగ్ డైరెక్టర్',
'Model':'',
'AdditionalCrew':'',
'Cricketer':'',
'Politician':'',
'Soundtrack':'సంగీత విభాగంలో ప్రదర్శకుడి',
'Cinematographer':'ఛాయాగ్రాహకుడి',
'Director':'దర్శకుడి',
'':'',
'Artdirector':'ఆర్ట్ డైరెక్టర్',
'Art' :'అసిస్టెంట్ ఆర్ట్ డైరెక్టర్',
'Producer':'నిర్మాత',
'Animation':'అనిమేషన్ ఆర్టిస్ట్',
'Writer':'కథా రచయిత',
'Actor':'నటుడి',
'Makeup':'మేకప్ ఆర్టిస్ట్',
'Music':'గాయకుడి' ,
'Specialeffects':'స్పెషల్ ఎఫెక్ట్స్ ఆర్టిస్ట్',
'CostumeandWardrobe':'వార్డ్రోబ్ డిజైనర్',
'Sound':'సౌండ్ ఇంజినర్',
'Visualeffects':'విసువల్ ఎఫెక్ట్స్ ఆర్టిస్ట్',
'ScriptandContinuity':'స్క్రిప్ట్ రైటర్',
'Composer':'సంగీత దర్శకుడి',
'Productionmanager':'ప్రొడక్షన్ మేనేజర్',
'Setdecorator':'సెట్ డెకరేటర్',
'Productiondesigner':'ప్రొడక్షన్ డిజైనర్',
'SecondUnitDirectororAssistantDirector':'సహాయ దర్శకుడి',
'Comedian':'హాస్యనటుడి',
'Costumedesigner':'కాస్ట్యూమ్ డిజైనర్',
'Stunts':'స్టంట్ డైరెక్టర్',
'CameraandElectrical':'',
'Castingdirector':'',
'Transportation':''
```

Categories, References

- Listing categories : Based on the attributes we have in the data set and referred some of the articles in wikipedia which are similar to our data set and listed them
- categories : family_name, actor place of birth, date of birth, date of death, occupations such as actor,director,producer,writer. Awards according to their first winning award and any renowned awards like oscar and national awards
- categories should be relevant to the article and it should be reliable content.

List of categories in the article:

- {{pob}}నుండి ప్రముఖులు
- నటులు
- దర్శకులు
- నిర్మాతలు
- సంపాదకీయులు
- తారాగణ దర్శకులు
- కళా దర్శకులు
- మేకప్ ఆర్టిస్ట్
- స్థాన నిర్వాహకులు
- సంగీతకారులు

- ఉత్పత్తి నిర్వాహకులు
- హాస్య నటులు
- యానిమేటర్లు
- ఛాయాగ్రాహకులు
- వస్త్ర రూపకర్తలు
- సహాయ దర్శకులు
- రచయితలు
- సంగీత దర్శకులు
- {{dob}}జననాలు
- {{DOD_year}} మరణాలు
- {{native language}}సినీ {{occup}}
- {{awards}} ఫిలింఫేర్ అవార్డుల విజేతలు
- ఆస్కార్ అవార్డుల విజేతలు
- తెవికీ సినిమా కళాకారులు

Infobox

- Used the existing Infobox Person.
- infobox rendering issues --- Images in the Infobox were taking lots of space, so mentioned '|thumb' after the image name, so it adjusted to the Infobox size. And also same problem with the signature image its not aligned perfectly centered so added '|center' to that image to be aligned to the centre of the infobox.

Translation/ Transliteration

Libraries

- **Textblob - Translation**

  The textblob online rendering was crashing for a large number of records so shifted to an alternative method.

- **Deep Translit - Transliteration**

  The accuracy was not up to the mark. Moreover, we used this library for almost all attributes.

  Issues:

  - Coming to transliteration we used deep translit library to transliterate the database of the artists who have above 500 or 1000 movies. It was not transliterating at some point so cross checked the data and took a limited number of movies for some artists.
  - Transliteration for some words like The, To etc.. Wrongly transliterated words:
    - The-తె → ది
    - To - టో → టు
  - These type of words are manually filled by find and replace method

- **Google Translator - Translation**

  Used for translation to some attributes like citizenship, occupation etc. it can convert any large file by giving the text file to google translator

XML Generation

- Render.py generates the wikitext for every artist. This wikitext is dumped into a single xml file with the help of genxml.py.
- This single xml file contains wikitext of each movie artist with unique page ids.
- This single xml file is imported to wikipedia to generate the articles for different movie artists.
- It is observed that there were duplicate movie artists titles in the dataset which will cause a problem while generating an XML file, as they might overwrite the previous movie artists with the same name. Since these can override the artists so that we made sure that there were no duplicates in the dataset.

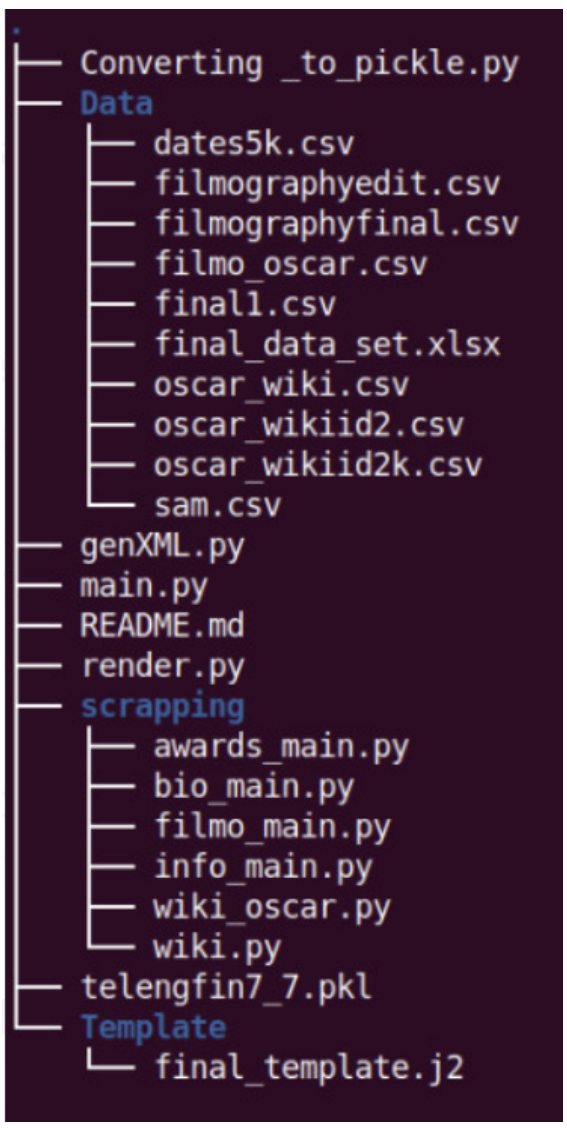
- While creating the XML, predefined entities(< > &’ “) were also taken care of by replacing them with appropriate strings(< > &"&apos) respectively. These entities are replaced with the strings mentioned above in all of the Wikitext and the title of the movie artists. Same process is done for title names as a safe side if any predefined present in it.

Quality review

- Generated some random articles among 7000 artists to check for any rendering mistakes or randomization.
- We took help from our mentor (prasanna mam) to improve the quality of our articles by random rendering of articles and reviewing them.
- After getting the article reviewed by the language experts, following their feedback we made some changes to the article, be it restructuring to refine the article structure or correct the randomization of sentences, etc.
- We made sure that the uniformity in the article is maintained, and also by taking care of any line breaks, extra lines, spaces by removing them.
- When we were reviewing the database, we found some records in the database that have little to no data available on them. We removed those records from the database.

Github Structure

Repository link:- https://github.com/indicwiki-iiit/Movie_artists



Template:

GitHub folder link: https://github.com/indicwiki-iiit/Movie_artists/blob/main/Template/final_template.j2

This folder contains the templates that are used for article generation

- final_template.j2 -- Contains the final Jinja2 Template for the article generation.

Data:

GitHub link : https://github.com/indicwiki-iiit/Movie_artists/tree/main/Data

- dates5k.csv : This data set consists of dates like date of birth and date of death which we have scrapped for the Indian artists about 5k persons.
- Filmography edit : Filmography which we scrapped have stored in a particular format in the form of list of list
- Filmography final : The final filmography for all the artists
- Final_data_set.csv : For both Indian and Oscar artists merged the whole data into single file
- Oscar wiki : For Oscar artists we have scrapped the data separately for almost 2000+ people

Scrapping:

Github link : https://github.com/indicwiki-iiit/Movie_artists/tree/main/scrapping

- awards_main.py : this will be useful in scrapping awards for the artists
- bio_main.py : this is used to scrape the bio of the artists from the imdb
- filmo_main.py : used to scrape the filmography data from the imdb
- info_main.py : for scrapping the info from imdb
- wiki_oscar.py : extracting known for from imdb
- wiki.py : Scrapping for the required attributes with unique ids from the wiki data

genXML.py:

GitHub link : https://github.com/indicwiki-iiit/Movie_artists/blob/main/genXML.py

- Contains the base code required for XML generation of the movie artists

render.py:

GitHub link : https://github.com/indicwiki-iiit/Movie_artists/blob/main/render.py

- Rendering code for the movie artists and for corrections of errors. The sample template will be generated from this code.

main.py:

GitHub link : https://github.com/indicwiki-iiit/Movie_artists/blob/main/main.py

- For generating the XML file we wrote this code by importing both the code files(render.py & genXML.py)

Converting_to_pickle.py:

GitHub link : https://github.com/indicwiki-iiit/Movie_artists/blob/main/Converting%20_to_pickle.py

- This will help to convert the dataset into pickle form.

telengfin7_7.pkl:

GitHub link : https://github.com/indicwiki-iiit/Movie_artists/blob/main/telengfin7_7.pkl

- Final pickle file for the movie artists to generate the articles.

Possible Extensions:

These are the possible extensions that can be developed on the current work in this domain [Movie artists].

- Education details of the artist, like their year of study, place of study etc.
- Personal life, family and marriage clear details of the artist like their spouse details and their marriage life (divorce details if any).

- Remuneration details of the artist, like their highest remuneration amount and the movie details for which highest remuneration was taken.
- Highest box office collection details, like the movie which made the highest collection in the artist's career.
- Their achievements in other fields rather than film careers etc.