Softwares :

The outcome of this domain was to collect data and articulate the information about various antiviruses and softwares for wikipedia. There are numerous softwares which are currently in use, although there was no particular website which describes the information in a native language (i.e Telugu). Therefore we took up the task to generate the essay's for about 11500 softwares and nearly 100 antiviruses. Furthermore, we've used python at each and every stage of the process(i.e scrapping,cleaning,merging) apart from that we've used few libraries to translate and transliterate the data (i.e anuvad,deep translit).

Team :

The team comprises of 4 members, whose whereabouts are as follows:

- Prathusha (Mentor) - cpmayura7@msitprogram.net
- Nirmai (Mentor) - nirmaialoori@gmail.com
- Veena - veenakadari@gmail.com
- V Vikas Reddy - vikasreddy270@gmail.com

Data Collection :

- https://download.cnet.com
    1. Format of the data available - columns
    2. Tools used - beautiful soup,pandas library
    3. Attributes scrapped - description ,whats_new ,additional_requirements and related_softwares
- Wikipedia (wikidata)
    1. Format of the data available - columns
    2. Tools used - beautiful soup,pandas library
    3. Attributes scrapped - all the attributes excluding the ones from the website download cnet
- The attributes are :

    1. Name
    2. Original_authors
    3. Developers
    4. Initial_release
    5. Stable Release Version
    6. Stable Release Date
    7. Preview_release
    8. Written_in
    9. Repository
    10. Middleware
    11. Engine
    12. Operating_system
    13. Platform
    14. Available_in
    15. Type
    16. Licence
    17. As_of
    18. Website
    19. Included_with
    20. Predecessor
    21. Successor
    22. Service_name
    23. Size
    24. Description
    25. Link
    26. Related_softwares
    27. Additional_requirments
    28. Whats_new

29. References
30. externalLinks
31. category

Tools used for data collection :

- **Beautiful soup** :
    1. We implemented a simple if else block to check if the value of the attribute is present or not.
    2. Speed was an issue and there were a few connection time-out errors. Inorder to overcome that, we've divided the attributes among ourselves which were required to be scrapped and that has considerably saved the time.

- **Wikipedia query page**

    1. We've collected the list of softwares from the wikipedia query page.

- **Images :**

    1. We've scrapped the image links alongside the remaining attributes from the English wikipedia articles.

- **Data storing** :

    1. Data is primarily stored in the csv, xlsx files.
    2. As we were constantly involved in updating,cleaning the data, we needed a structure which was agile. As a result we've used csv. Furthermore, wps office, microsoft word were the preferred softwares to analyse the data and they represented the data in the form of xlsx. On the other hand after scraping the data, we had to store and merge the data as well, so as a common platform for all the team members we've extensively used google docs as well as spreadsheets.

- **Data cleaning :**

    1. Some of the softwares was having abbreviations as their official name which after transliteration produced erroneous data, so we had to manually replace them with the appropriate one's
       Ex: AOL was transliterated as 'అమోల్ ' .

2. The scraped external links were concatenated directly as a string i.e there was no differentiation between the links ,So we had differentiated the links from one another.
3. Some abbreviations like LLC, Inc etc., remained the same in the transliterated data.
4. There were few spelling mistakes in the transliterated and translated data, Which were cleaned.
5. Some of the links from the dataset were not having any information, so we had removed the software.
6. Few Softwares appeared more than once, so we've removed the duplicate records.

- **Data merging** :

1. Firstly, we had the script ready to scrape all the attributes, and then divided the number of rows to be scrapped among us. Importantly, We've used a pandas script to merge the data, Inorder to reduce the redundancies we ensured that the files to be merged had the number of columns.
2. Final format was always csv.
3. As we've scrapped a few attributes from other websites, which were a handful, we have manually copied the entire column and pasted on the actual dataset. Moreover, it was done on google docs as it was user friendly.

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 :**

1. The link of the sample is as follows: https://tinyurl.com/yc6p2drj

- **Sections :**
1. We took the reference from the articles in the wikipedia, to get an idea on the structuring of the article.
2. **Infobox :** It is basically a table which contains the values for the attributes that are available for that software

3. **Overview :** It consists of all the general information about the software like the platform used, language used etc.,
4. **Introduction :** Contains the information about the name of the version, date released, the repository the software is linked to etc.,
5. **Additional requirements** : It describes the additional requirements that are needed for the proper functioning of the software.
6. **References :** This section of the article outlines the references that we've used to source the data.
7. **External links :** These are the Websites of organisations mentioned in an article.
8. **Categories :** Categories are used in Wikipedia to link articles under a common topic and are found at the bottom of the article page.

- # Jinja template creation :

Link : https://tinyurl.com/2p97curb

- ## Edge cases

● For numeric attributes, -1 was placed for stats that didn't have valid values, and hence was handled in template generation implementation.
● Null values, NoneType objects, empty strings, 'nan' values are all completely handled such that information would be displayed only when the content in that cell didn't correspond to any such above mentioned categories.
● Singular - plural values for numeric attributes were also handled in the template itself.
● Consistent spacing and line-breaks are ensured while generating templates.
● For sub paragraphs, only those sentences are displayed for a software, if there is at least one non-null value for attributes in the sentence.
● We had to be cautious about attributes available in the infobox, and attributes we collected.So, we created a custom infobox template and filled it with available values.

- ## Categories and References :

1. We've categorised the softwares in terms of the operating system that they can be used in, Moreover they can also be categorised by the service they provide.
2. Different categories listed are **Windows, macOS, Linux, Android, FreeBSD, NetBSD, OpenBSD,Unix, Unix-like, Plan 9, Inferno, MSX-DOS, IBM i, etc**.,
3. We've articulated all the links of the websites through which we have collected the data. As we have used wikipedia also to scrape the data there

was a dedicated table, where they displayed the references in a table which we were able to scrap . Furthermore download cnet is an excellent website we've used, and more importantly the website links were added to the references.

- **Infobox :**

1. Generally, infobox is one of the most important components to have in an article, because it gives away all the essential information by directly looking at it.
2. We've tried to put most of the attributes in the infobox, but the attributes which were consuming a lot of space like description,related softwares etc, were not included in the infobox.
3. As the value of each and every attribute is not present, we've displayed the attributes with the non-null values.
4. We've designed the infobox in such a manner that it goes on to describe the attributes of a software from general to most-specific, this ensures systematic flow of information.

Translation and Transliteration :

LIbraries

- **Anuvaad**
  - Primarily used for translation
  - As it was too slow and was producing a lot of redundancies, we had to look for an alternative

- **Google Translate**
  - It was used for translation of the data
  - It was rather faster than anuvaad but was also producing discrepancies.

- **Deep translit**
  - Used for the transliteration of the text
  - It was slow and also produced errors while transliteration.

Issues with Translation and Transliteration :

- Initially, we had to classify the data which needs to be translated and transliterated and as a matter of concern some of the attributes were in need of both translation and transliteration

- Deep Translit was used for transliteration, It didn't work well when abbreviations were in use, In other words it produced misspellings. Ex: AOL was transliterated as 'అమోల్ ' .
- Anuvaad was initially used for translation, It was too slow and didn't get the desired results, So as an alternative we had to use google translate.
- Google Translate was comparatively agile than anvaad, Though its translation was not satisfying. There were a few discrepancies.
- Description needed both translation and transliteration,

XML generation :

- Executing the https://tinyurl.com/48ha36r3 would generate wikitext for every software, whose details are present in the finalised dataset. This wikitext is dumped into a single xml file with the help of file https://tinyurl.com/mr25abjn
- The single XML file https://tinyurl.com/48ha36r3 contains wikitext of every Software, whose details were collected. This xml file is imported to wikipedia to generate the articles for different softwares.

Quality Review :

- We've added Randomization to the sentences, so that the sentences don't remain the same for all the softwares.
- Values of each and every attribute were manually checked for possible discrepancies.
- The structuring of the attributes in the infobox and structuring of sentences are organised, which provides the systematic flow of information to the reader.
- Furthermore, we've generated sample articles to check the soundness of the article.
- To maintain the quality of the article, we've only used the attributes which were significant and appropriate.
- Overall, There was no compromise made for the quality of the article. The structure that we've produced is not only informative but also enhances user readability and experience.

Github Structure

Visit our GitHub repository https://github.com/indicwiki-iiit/Softwares-and-Antiviruses