Project Overview:

The primary goal of this project was to gather and compile data on various antivirus and software applications for Wikipedia, with a focus on the Telugu language. Despite the numerous software applications in use, there was a lack of comprehensive native-language (Telugu) information available on them. Consequently, our team undertook the task of creating essays for approximately 11,500 software applications and nearly 100 antivirus programs. Python played a crucial role in every phase of the project, including web scraping, data cleaning, and data merging. We also leveraged various libraries for translation and transliteration, such as Anuvaad and Deep Translit.

Team:

Our team consisted of four members:

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

Data Collection:

We collected data from two primary sources:

1. **https://download.cnet.com**
   - Format of the available data: Columns
   - Tools used: Beautiful Soup, Pandas library
   - Attributes scraped:** Description, what's new, additional requirements, and related software

2. **Wikipedia (wikidata)**
   - Format of the available data: Columns
   - Tools used: Beautiful Soup, Pandas library
   - Attributes scraped: All attributes excluding those from the website download.cnet

The collected attributes include:

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. License
17. As of
18. Website
19. Included with
20. Predecessor
21. Successor
22. Service name
23. Size
24. Description
25. Link
26. Related software
27. Additional requirements
28. What's new
29. References
30. External links
31. Category

Tools Used for Data Collection:

1. Beautiful Soup: We used conditional statements to check if attributes were present, dividing the scraping tasks among team members to improve efficiency.

2. Wikipedia Query Page: We collected the list of software from Wikipedia query pages.

3. Images: We scraped image links alongside other attributes from English Wikipedia articles.

4. Data Storage: Data was primarily stored in CSV and XLSX files. CSV files were chosen for their agility and ease of updating and cleaning data. Microsoft Word and WPS Office were used for data analysis, with data represented in XLSX format. Google Docs and spreadsheets served as a common platform for team members during data storage and merging.

5. Data Cleaning:

Data cleaning involved several tasks:

1. Correcting erroneous transliterations for abbreviations.
2. Differentiating between concatenated external links.
3. Handling abbreviations like LLC and Inc.
4. Correcting spelling mistakes in transliterated and translated data.
5. Removing software with non-informative links.
6. Eliminating duplicate software records.

6. Data Merging:

Data was merged using a Pandas script to reduce redundancies. We ensured that the files to be merged had the same number of columns, and the final format was always CSV. For attributes scraped from other websites, we manually copied and pasted entire columns onto the actual dataset using Google Docs.

Version Control:

GitHub was used for version control, especially when multiple team members were working on various versions of templates and data files.

Sample Article:

A sample article is available at the following link: https://tinyurl.com/yc6p2drj

- **Sections of the Article:**

  We followed the structure of Wikipedia articles to create our essays. The main sections included:

  1. Infobox: A table containing essential attributes of the software.
  2. Overview: General information about the software, including platform and programming language.
  3. Introduction: Details about the software version, release date, and repository.
  4. Additional Requirements: Information about software prerequisites.
  5. References: Citations used for sourcing data.
  6. External Links: Links to related organizations.
  7. Categories: Categorization of software by operating system and service provided.

Jinja Template Creation:

A Jinja template was created for generating articles. The template can be accessed at this link: https://tinyurl.com/2p97curb

Edge Cases:

Various edge cases were handled during data processing:

1. Numeric attributes with no valid values were set to -1.
2. Null values, NoneType objects, empty strings, and 'nan' values were handled to display information only when the cell content was meaningful.
3. Singular-plural values for numeric attributes were addressed in the template.
4. Consistent spacing and line-breaks were ensured.
5. Subparagraphs displayed only if at least one non-null value existed for attributes in the sentence.

Translation and Transliteration:

We used the following libraries for translation and transliteration:

- Anuvaad: Primarily used for translation but was slow and produced redundancies.

- Google Translate: Used for translation and was faster than Anuvaad but had some discrepancies.

- Deep Translit: Used for transliteration but was slow and produced errors, especially with abbreviations.

Issues with Translation and Transliteration:

- Deep Translit did not work well with abbreviations, resulting in misspellings.
- Anuvaad was slow and did not yield satisfactory results, leading to the adoption of Google Translate.
- Google Translate, while faster, also had some translation discrepancies.
- Description required both translation and transliteration.

XML Generation:

The execution of the provided URL generated wikitext for each software in the dataset. This wikitext was compiled into a single XML file using the specified file, which was then imported into Wikipedia to generate articles for various software.

Quality Review:

To maintain article quality, we introduced randomization to sentences, ensuring that they do not remain the same for all software. We manually checked the values of each attribute for possible discrepancies. The structure of attributes in the infobox and sentence structuring were organized to provide systematic information flow to readers. Sample articles were generated to ensure the soundness of the content. Only significant and appropriate attributes were used, with no compromise on article quality.

GitHub Structure:

Our GitHub repository can be accessed at https://github.com/indicwiki-iiit/Softwares-and-Antiviruses