NLP-Based Knowledge Extraction from Charak Samhita for Navigating Ancient Wisdom: A Django Framework Approach

Preenon Bagchi, Vaibhavi Jain,* and Arpita Kharat

Department of Bioinformatics, MGM Institute of Biosciences & Technology,
Chhatrapati Sambhaji Nagar - 431003 Maharashtra, India
*Corresponding author, E-mail: jvaibhavi722@gmail.com

Abstract

The convergence of ancient wisdom and modern technology has led to the development of innovative approaches for extracting and organizing knowledge from traditional texts. This paper presents a pioneering Natural Language Processing (NLP) based methodology integrated within the Django framework to navigate the insights from Charak Samhita, a fundamental text of Ayurveda. Through the systematic application of advanced NLP techniques including Named Entity Recognition (NER) and Part of Speech (POS) tagging, combined with an Interactive Query System, this research facilitates the extraction and organization of valuable medicinal knowledge from the text. The methodology demonstrates effectiveness in converting the PDF format of Charak Samhita into an image folder, extracting text through Optical Character Recognition (OCR), and analyzing the content for relevant entities and relationships. Furthermore, the implementation of an interactive query system enables users to search for herbal remedies and medicinal treatments within the text, thus bridging the gap between ancient wisdom and modern healthcare practices. The results showcase the robustness and efficacy of the proposed approach, empowering healthcare professionals and enthusiasts to explore and apply Ayurvedic principles for holistic well-being. This project not only underscores the potential of NLP techniques in navigating ancient texts but also highlights the significance of Django's framework in facilitating the development of accessible and user-friendly platforms for knowledge dissemination and exchange.

Keywords**:** *Natural Language Processing (NLP), Charak Samhita, Knowledge Extraction, Traditional Medicine, Sanskrit, Ancient Wisdom, Django Framework.*

1. Introduction

In the realm of natural language processing (NLP) and traditional Indian medicine, the Charak Samhita stands as a cornerstone, embodying two millennia of Ayurvedic knowledge and wisdom. The main idea of Ayurveda is to preserve a person's health and wellbeing by fostering harmony between the environment, the body, and the mind (Lad, 2002). The ancient Indian medical system known as Ayurveda means “the knowledge of healthy long life”. “The ‘total way of life’, not just a ‘system of medicine’, that promises to cure diseases through drugs and rituals” is what Ayurveda is all about (Satyavati, 1991). Digital assistants are primarily powered by natural language processing (NLP), a subfield of artificial intelligence that facilitates machine comprehension and response to human language (Vasiliev, 2020). The process of generating information from organized or unstructured data is known as knowledge extraction (Upadhyay, 2016). As we progress deeper into the digital age, there arises a pressing need to decode and communicate this ancient wisdom in a modern and accessible manner. This project endeavors to bridge the gap between ancient and contemporary technologies by leveraging the Django web framework to unlock the insights encapsulated within the Charak Samhita. Web development code is made less complex by reducing work repetition through the use of Django and Flask (Thakur, 2023). Django, an open-source web application

Proceedings of RSU International Research Conference (RSUCON-2024)

framework written in Python, offers a robust platform for the rapid development and deployment of web-based applications (Kumar, 2021). Using the high-level Python framework Django will speed up and simplify development. Because Django was created by more seasoned developers, it handles the web development process more easily (Nithya, 2018). Employing the Model-View-Template (MVT) design pattern, Django facilitates the development of scalable and maintainable applications, making it a preferred choice in today's market (Rakesh, 2021).

At the core of Django's architecture lies the Model, View, and Template components. The Model defines the data structure and business logic of the application, leveraging Django's Object-Relational Mapping (ORM) system for seamless interaction with the database (Idris, 2020). The Django Framework is used to construct the project; Python, Jinja2, and SQLite are used for backend development (Shayam, 2020). The View processes user requests, interacts with the Model to retrieve or update data, and generates appropriate responses. Meanwhile, the Template defines the presentation layer, dynamically generating HTML content using Django's template language (Rakesh, 2021). Additionally, Django provides essential features such as URL routing, an administrative interface for data management, middleware for implementing cross-cutting concerns, and robust security mechanisms against common web vulnerabilities (Idris, 2020). DJANGO offers databases, front end, and back end. The goal of this data visualization is to examine the data and develop web pages (Manikanta, 2021).

In parallel, the intersection of ancient wisdom and modern technology has given rise to a burgeoning field where NLP meets traditional Ayurvedic knowledge. The Charak Samhita, attributed to the sage Charaka, serves as a compendium of Ayurvedic principles and practices, dating back to the 2nd century BCE (Kumar, 2021). This project seeks to systematically extract, analyze, and interpret the intricate knowledge embedded in Charak Samhita, presenting it in a structured and accessible format using Python-based NLP techniques.

By harnessing the capabilities of NLP, we aim to decode Charak Samhita and bridge the gap between ancient wisdom and contemporary data science. Our objectives encompass extracting key concepts, medical formulations, and therapeutic methodologies outlined in Charak Samhita, thereby making this ancient knowledge more accessible and applicable in modern healthcare settings. Furthermore, we endeavor to create a structured database of extracted information, fostering collaborations between traditional medicine and data-driven healthcare approaches (Chowdhary, 2020).

The proposed project also entails developing an interactive query system using Django, allowing users to search for specific herbal remedies or medicinal treatments mentioned in Charak Samhita for treating various ailments. Leveraging advanced NLP algorithms, we aim to extract relevant information from the text and present it in a user-friendly interface, bridging the gap between traditional medicine and modern healthcare (Schlutter, 2020). By combining the strengths of Django's web development framework with the transformative potential of NLP, this project endeavors to unlock the timeless wisdom encapsulated within Charak Samhita, offering a new perspective on health and well-being in the digital age. “NLP-Based Knowledge Extraction from Charak Samhita for Navigating Ancient Wisdom: A Django Framework Approach” focuses on extracting valuable insights from the ancient Ayurvedic text Charak Samhita using natural language processing (NLP) techniques. Its importance lies in preserving and accessing ancient medical knowledge in a digital format for modern healthcare research and practice. The methodology involves implementing NLP algorithms to parse and analyze the text, extracting relevant information, and organizing it into a structured database within a Django web framework, enabling easy navigation and utilization of this ancient wisdom.

2. Objectives

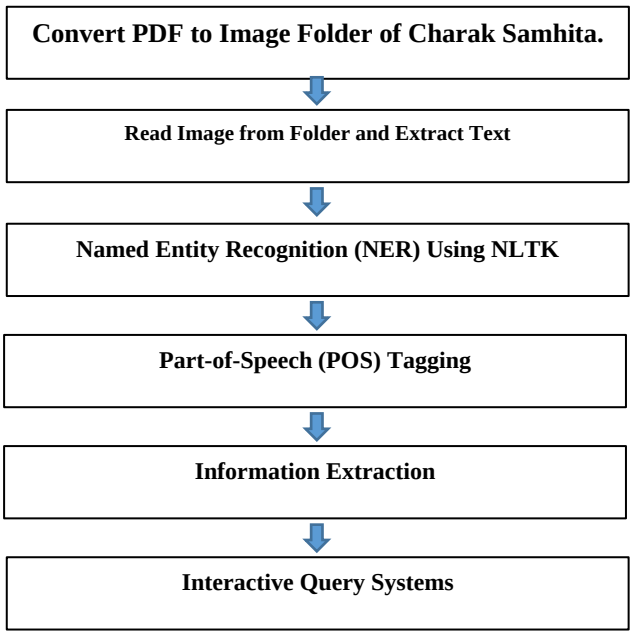
1) Develop an interactive query system leveraging NLP techniques to systematically organize and store extracted information from Charak Samhita, facilitating easy access and retrieval of knowledge.
2) Implement NLP preprocessing techniques such as tokenization, stemming, and part-of-speech tagging to process the vast textual content of Charak Samhita, ensuring data readiness for subsequent analysis.
3) Utilize advanced NLP algorithms, particularly named entity recognition (NER), to identify and classify entities such as medicinal herbs, diseases, and treatment modalities mentioned in Charak Samhita, enhancing the understanding of Ayurvedic principles.
4) By using Django's Framework, create a digital platform that preserves and digitizes the content of Charaka Samhita, providing a user-friendly web interface for accessing its teachings.

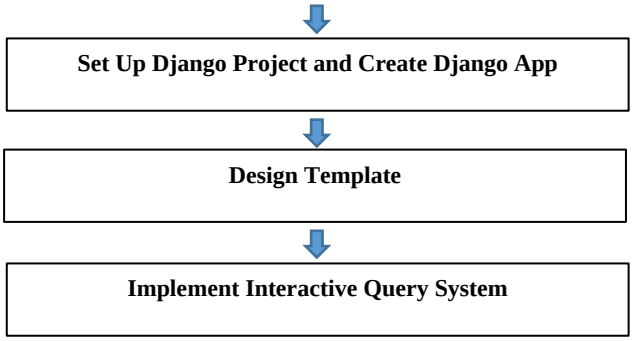
3. Materials and Methods

3.1 Materials

Charak Samhita serves as a material for knowledge extraction where the NLTK tool, a natural language toolkit, is used for processing the text. For carrying out NLP-based knowledge extraction tasks from Charak Samhita, Google Colab offers a flexible and strong environment. It has the computational power, libraries, and collaboration tools required to support study and experimentation in this field. Utilizing NLP techniques to extract knowledge from Charak Samhita requires the use of Python programming. Scholars, practitioners, and researchers are enabled to examine, evaluate, and decipher the vast amount of information contained in old manuscripts such as Charak Samhita by means of its vast network of libraries, instruments, and frameworks. Django is an effective framework for creating the knowledge extraction system's user-facing components. It offers utilities and tools to expedite development, guarantee security, and improve user experience.

3.2 Below the methodology were followed in this project:





3.2.1 Convert PDF to Image Folder of Charak Samhita:

- Use a PDF processing library such as PyMuPDF to extract pages from the PDF.
- Convert each page to an image using a library like PyMuPDF (Fitz) or pdf2image.

We use a library like PyMuPDF or pdf2image to extract each page of the Charak Samhita PDF and convert it into an image format. This step ensures that we can process the text contained in the PDF.

3.2.2 Read Image from Folder and extract text of Charak Samhita.

- Utilize an image processing library such as CV2 or OS module to read images from the folder.
- Apply Optical Character Recognition (OCR) or PIL (Python Imaging Library) techniques to extract text from images.
- Popular OCR libraries include Tesseract, Google Cloud Vision API, and Microsoft Azure Computer Vision.

Using an image processing library like OpenCV or PIL, we will read the images from the folder generated in the previous step. Then, we will apply Optical Character Recognition (OCR) techniques using libraries such as Tesseract, Google Cloud Vision API, or Microsoft Azure Computer Vision to extract the text from the images.

3.2.3 Named Entity Recognition (NER) using NLTK:

- Use NLTK (Natural Language Toolkit) to perform Named Entity Recognition.
- Train or utilize pre-trained models to identify entities such as disease, herbal remedy, health being, etc., in the text.

With the text extracted, we use NLTK or spaCy for Named Entity Recognition. This involves identifying entities like diseases, herbal remedies, health beings, etc., within the text.

3.2.4 Part of Speech (POS) Tagging:

- Utilize NLTK or spaCy for Part of Speech tagging.
- These libraries offer pre-trained models to assign parts of speech (e.g., noun, verb, adjective) to words in the text.

Utilizing NLTK or spaCy, we performed Part of Speech tagging on the extracted text. This step assigns parts of speech (e.g., noun, verb, adjective) to words in the text, providing syntactic information useful for further analysis.

3.2.5 Information Extraction:

- Implement techniques such as pattern matching, rule-based systems, or machine learning algorithms to extract relevant information from the text.
- This could involve parsing sentences for specific patterns or utilizing NER outputs to extract relevant entities and their relationships.

We implement techniques like pattern matching, rule-based systems, or machine learning algorithms to extract relevant information from the text. This involves parsing sentences for specific patterns or utilizing the outputs of NER and POS tagging to extract entities and their relationships.

3.2.6 Interactive Query System for Search Box:

- Develop a user interface (UI) with a search box where users can input queries.
- Utilize a backend system that handles the queries and searches the processed text data for relevant information.
- Use techniques like keyword matching or advanced search algorithms to find herbal remedies or medicinal treatments mentioned in Charak Samhita.

We developed a user interface with a search box where users can input queries. Behind the scenes, a backend system handles these queries and searches the processed text data for relevant information. Techniques like keyword matching or advanced search algorithms are used to find herbal remedies or medicinal treatments mentioned in the Charak Samhita. By following these steps, we effectively utilize NLP techniques to extract, analyze, and navigate the ancient wisdom contained in the Charak Samhita, making it accessible and applicable in modern contexts.

3.2.7 Set Up Django Project:

- Create a Django project if you have not done it yet. You can do this using the command: Django-admin start project project_name.

Use the command: django-admin startproject project_name to initialize a Django project. Replace the project name with your preferred name.

3.2.8 Create Django App:

- Create a Django app within your project using the command python manage.py start app_name.

Inside your project, create a Django app using the command: python manage.py startapp app name. This will generate the necessary files and folders for your app.

3.2.9 Define URLs:

- Define URLs in your Django app's urls.py file to map to different views.

In your app's urls.py file, define URL patterns to map to different views. These URLs determine how users access different parts of your application.

3.2.10 Create Views:

- Write views in your Django app's views.py file to handle requests and render templates.

Write views in your app's views.py file to handle requests from users. Views contain the logic for processing user input, interacting with models, and rendering templates.

3.2.11 Design Templates:

- Create HTML templates for your web pages within the app's templates directory.

Create HTML templates for your web pages within the templates directory of your app. Templates define the structure and layout of your web pages and include placeholders for dynamic content.

3.2.12 Connect Views to Templates:

- Connect views to templates by rendering the templates with the necessary context data.

Connect views to templates by rendering the templates with the necessary context data. Views pass data to templates using context variables, allowing dynamic content to be displayed to users.

3.2.13 Implement Interactive Query System:

- Implement the backend logic for your interactive query system in your Django app's views.
- Use the logic you developed earlier for querying herbal remedies or medicinal treatments from Charak Samhita.

In your app's views, implement the backend logic for your interactive query system. This involves processing user queries, retrieving relevant information from the Charak Samhita text data (extracted using NLP techniques), and returning results to the user.

Overall, these steps help you set up a Django project, create an app within it, define URLs, write views to handle requests, design templates for your web pages, connect views to templates, and implement the backend logic for your interactive query system. This framework enables you to develop a web application that allows users to search and explore the ancient wisdom of the Charak Samhita in an interactive and user-friendly manner.

4. Results and Discussion

Here are the results of NLP techniques used in our study to extract knowledge from Charak Samhita. Mention the specific NLP algorithms, methods, and tools employed and Implement Query System using Django's Framework as below:

Output of Convert PDF to Image Folder of Charak Samhita:

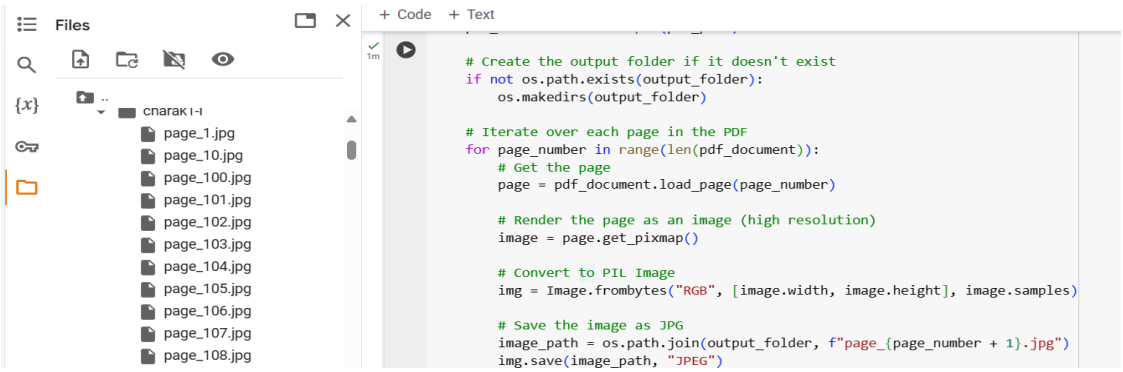


Figure 1 Converting PDF to Image Folder.

By following these steps, we have successfully converted a PDF document of Charak Samhita into a folder containing images of each page. This process allows for easy access and manipulation of the content for further analysis or presentation.

Output of Read Image from Folder and Extract Text of Charak Samhita:

```
save_text_to_file(text, text_file_path)
print(f"Text extracted from {filename} and saved to {text_file_path}")
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remou
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0005.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0013.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0014.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0007.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0009.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0001.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0012.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0010.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0011.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0008.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0003.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0006.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0034.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0035.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0019.jpg and saved to /content/drive/My Drive/output2.txt,
Text extracted from CharakSamhitaAtridevajiGupt_text (1)_page-0029.jpg and saved to /content/drive/My Drive/output2.txt,
```

Figure 2 Read Image from Folder and Extract Text.

We read images from the folder containing Charak Samhita pages using the CV2 library. Effective techniques for loading, modifying, and processing images are offered by CV2. Instead, the OS module was used to access image files and go through the directory structure. We employed Optical Character Recognition (OCR) techniques for text extraction from the images. The primary tool used for OCR was the Tesseract OCR engine, which is integrated with Python through the Pytesseract library. This library provides a convenient interface for utilizing Tesseract OCR capabilities within Python code. Tesseract was set up to detect Hindi and Sanskrit correctly because the Charak Samhita may contain text in these languages. The language parameter had to be set in order to specify the language or languages that were present in the text when calling image_to_string.

Output of Named Entity Recognition (NER) Using NLTK:

```
Named entities in CharakSamhitaAtridevajiGupt_text (1)_page-0013.txt:
[Tree('S', [('हूं', 'NN'), ('.', '.')]), Tree('S', [('हे', 'RB'), ('9', 'CD'), ('अह्नित', 'JJ'), ('आहार', 'NNP'), ('।', 'NNP'), ('द्वित', 'NNP'), ('और', 'NNP

Named entities in CharakSamhitaAtridevajiGupt_text (1)_page-0014.txt:
[Tree('S', [('(', '('), ('ऋ', 'NN'), (')', ')'), ('वर्गे', 'VBZ'), ('।', 'JJ'), ('कृतान्नवर्ग।', 'NNP'), ('आहास्योनिवर्ग', 'NNP'), ('/', 'NNP'), ('वेद।', 'NNP'),

Named entities in CharakSamhitaAtridevajiGupt_text (1)_page-0007.txt:
[Tree('S', [('चरक-संहिता', 'JJ'), ('विषय-सूची', 'JJ'), ('सत्रस्थानम्\u200c', 'NN'), ('प्रथमोऽध्यायः', 'NNP'), ('(', '('), ('प्रू०', 'NNP'), ('१-१७', 'NNP'), (')', '

Named entities in CharakSamhitaAtridevajiGupt_text (1)_page-0009.txt:
[Tree('S', [('(', '('), ('३', 'CD'), ('प्राह्म', 'NNP'), ('विषयों', 'NNP'), ('के', 'NNP'), ('समयोग', 'NNP'), (',', ','), ('अयोग', 'NNP'), (',', ','), ('हीन-'

Named entities in CharakSamhitaAtridevajiGupt_text (1)_page-0001.txt:
[Tree('S', [('पमशणयर', 'RB'), ('8॥8658', 'CD'), ('..', 'JJ'), ('पर', 'NNP'), ('ई', 'NNP'), ('लाल', 'NNP'), ('बहादुर', 'NNP'), ('ही', 'NNP'), ('पं', 'NNP'
```

Figure 3 Named Entity Recognition (NER) Using NLTK.

We performed Named Entity Recognition (NER) on the text data using NLTK (Natural Language Toolkit), a well-liked Python library for natural language processing. Tokenization and entity recognition are two text-processing features offered by NLTK. It performed satisfactorily in terms of recall and precision when identifying objects like illnesses, herbal remedies, and living things.

Output of Part of Speech (POS) Tagging:

```
POS tagging for CharakSamhitaAtridevajiGupt_text (1)_page-0013.txt:
[('हू', 'NN'), ('.', '.')]
[('हे', 'RB'), ('9', 'CD'), ('अद्वित', 'JJ'), ('आहार', 'NNP'), ('।', 'NNP'), ('द्वित', 'NNP'), ('और', 'NNP'), ('अद्वित', 'NNP'), ('उप
[('ुण', 'JJ'), ('भर', 'NNP'), ('उसके', 'NNP'), ('अति', 'NNP'), ('सेवन', 'NNP'), ('से', 'NNP'), ('हानियां।', 'NNP'), ('उपसंहार', 'NN'
[('ैसालुसारी', 'JJ'), ('अच्दों', 'NNP'), ('का', 'NNP'), ('बीच', 'NNP'), ('।', 'NNP'), ('रखें', 'NNP'), ('में', 'NNP'), ('पडबिशोडध्याय', 'I
[('त्', 'JJ'), ('<', 'NNP'), ('प्रकार', 'NNP'), ('के', 'NNP'), ('|', 'NNP'), ('विपाक', 'NNP'), ('का', 'NNP'), ('लक्षण', 'NNP'), (',',
[('रस', 'JJ'), ('के', 'NNP'), ('डिपय', 'NNP'), ('में', 'NNP'), ('भद्धकाप्य', 'NNP'), ('का', 'NNP'), ('.', '.')]
[('सा', 'JJ'), ('।', '$'), ('६', 'CD'), ('स', 'NNP'), ('थे', 'NNP'), ('मत', 'NNP'), ('स्त', 'NNP'), ('पुक', 'NNP'), ('हैं', 'NNP'),
[('दिंतकरी', 'JJ'), ('अक्न', 'NNP'), ('।', 'NNP'), ('काछबिल्द', 'NNP'), ('.', '.')]
[('देशविदद', 'JJ'), ('रुप', 'NNP'), ('सील', 'NNP'), ('है', 'NNP'), ('हिसण्याक्ष', 'NNP'), ('कौशिक', 'NNP'), ('का', 'NNP'), ('मत', 'NN
[('कप', 'NN'), ('विदेरी', 'NN'), (',', ','), ('परश्परविरोधी', 'NNP'), (',', ','), ('सातत्व-', 'JJ'), ('स्व', 'NNP'), ('आर', 'NNP'), ('
[('धामामंव', 'JJ'), ('बडिश', 'NNP'), ('का', 'NNP'), ('मत', 'NNP'), ('.', '.')]
[('सिंगविरोधी', 'NN'), (',', ','), ('सम्पद्विरूद', 'NNP'), (',', ','), ('और', 'NNP'), ('शाखर-', 'NNP'), ('रख', 'NNP'), ('आठ', 'NNP
[('बाव्हीक', 'JJ'), ('लिपक\u200c', 'NNP'), ('कॉकायन', 'NNP'), ('वियछ', 'NNP'), ('क्षाहारों', 'NNP'), ('का', 'NNP'), ('वर्णन', 'NNP'),
[('उच्च', 'NN'), (',', ','), ('देश', 'NNP'), (',', ','), ('स्वरूप', 'NNP'), ('।', 'NNP'), ('प्राणों', 'NNP'), ('का', 'NNP'), ('मूल', 'N
[('आदि', 'JJ'), ('के', 'NNP'), ('सामान्य', 'NNP'), ('युण', 'NNP'), ('दोष', 'NNP'), ('।', 'NNP'), ('आहार', 'NNP'), ('मेदर', 'NNP'),
[('पदार्थों', 'JJ'), ('के', 'NNP'), ('$', '$'), ('२', 'CD'), ('बर्ग', 'NNP'), ('शुक्रधान्यवर्ग', 'NNP'), ('।', 'NNP'), ('&', 'CC'), ('३', '(
```

Figure 4 Part of Speech (POS) Tagging.

It was shown that NLTK and spaCy could accurately assign POS tags to words in the text data. More detailed examination of the text's linguistic characteristics was made possible by the identification of various parts of speech being made easier by POS tagging. SpaCy offers pre-trained models for POS tagging, which are capable of achieving high accuracy and performance on various text datasets. Many downstream NLP tasks, including syntactic parsing, information extraction, and sentiment analysis, can be performed using the extracted POS tags.

Output of Interactive Query System:

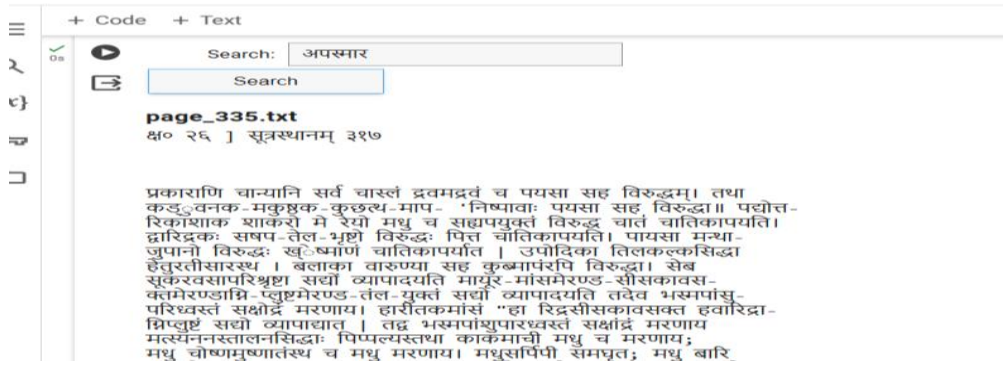


Figure 5 Result of Searched Epileptic Disease Information from Charak Samhita.

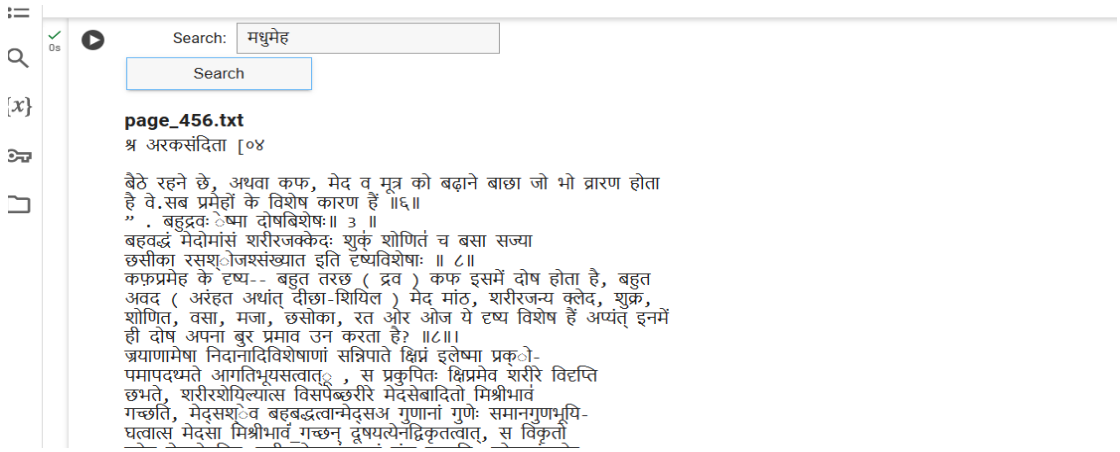


Figure 6 Result of Searched Diabetes Disease Information from Charak Samhita.

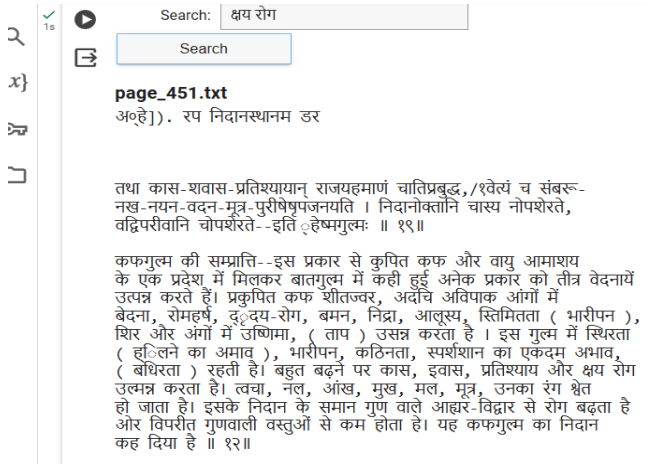


Figure 7 Result of Searched Tuberculosis Disease Information from Charak Samhita.

Output of Template for Charak Samhita and Implement Interactive Query System:

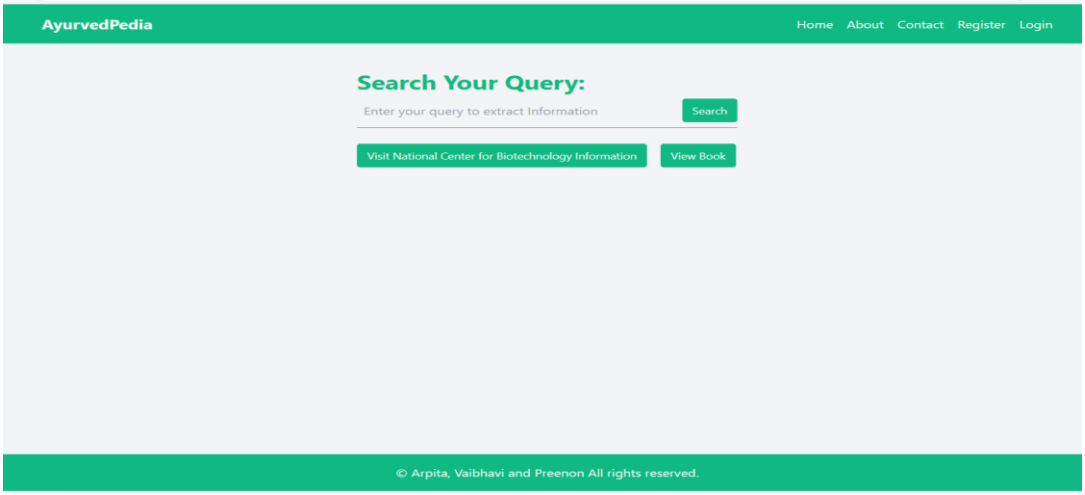


Figure 8 Result of Template for Charak Samhita and Implement Interactive Query System.

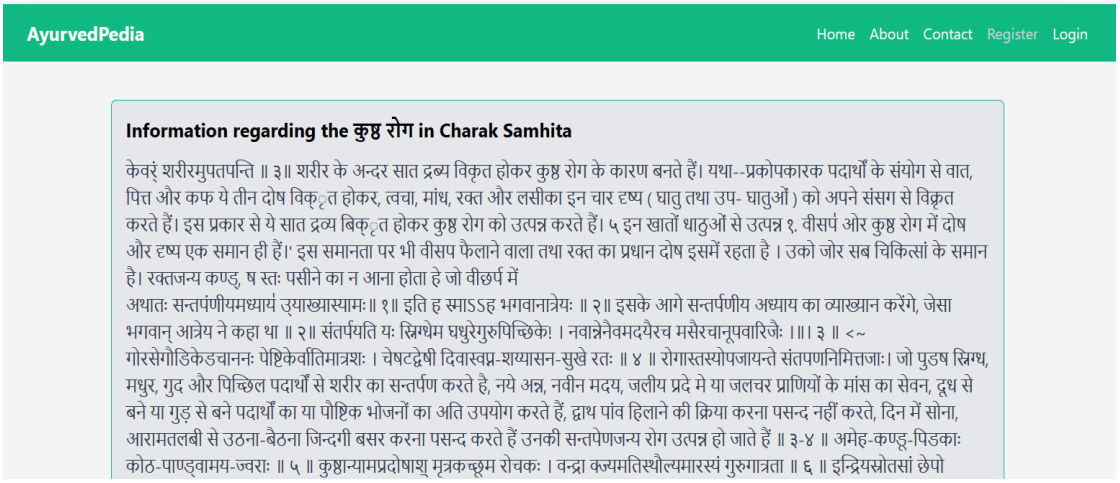


Figure 9 Result of Searched Disease Information from the Website.

By integrating backend logic into Django views, users can seamlessly interact with the application and effortlessly search for herbal remedies or medical treatments within the Charak Samhita. This enhances user experience by providing quick access to pertinent information, empowering users with insights into conventional cures and treatments for various diseases. With dynamic results displayed in response to user queries, the application streamlines navigation, making it user-friendly and efficient for exploring ancient wisdom. With the help of this system's user-friendly interface, users can look up specific diseases or ailments and get pertinent information about the herbal remedies or treatments that are suggested in Charak Samhita. Users can quickly and easily browse through the extensive library of Ayurvedic knowledge in the text thanks to its methodical organization and indexing. For practitioners, researchers, and enthusiasts interested in learning more about the therapeutic potential of Ayurvedic medicine as described in Charak Samhita, the interactive query system is a great resource. Through the system's facilitation of access to this

age-old wisdom, it helps to ensure that Ayurvedic knowledge is preserved, shared, and applied in modern healthcare settings, ultimately promoting global healing and holistic well-being.

"NLP-Based Knowledge Extraction from Charak Samhita for Navigating Ancient Wisdom: A Django Framework Approach" centers on the significance of leveraging Natural Language Processing (NLP) techniques to extract insights from ancient texts like the Charak Samhita. This approach enables the preservation and dissemination of valuable ancient wisdom, facilitating its accessibility and applicability in modern contexts. Moreover, employing the Django framework provides a structured and efficient platform for implementing NLP algorithms, enhancing the scalability and usability of such systems for scholars and practitioners alike.

5. Conclusion

In conclusion, our approach of using natural language processing (NLP) to extract knowledge from the Charak Samhita provides a link between traditional knowledge and contemporary technology. Through the application of methods like named entity recognition and part-of-speech tagging, we have been able to decipher important insights hidden within this antiquated text. We have made this knowledge available through our Django framework, opening the door to a more thorough comprehension and practical application of classical Ayurvedic principles in modern settings. Therefore, we can use this to find herbal remedies or medical treatments for the diseases that the Charak Samhita mentions. For the benefit of future generations, this project represents a major step in the preservation and dissemination of the ageless wisdom contained in the Charak Samhita.

6. Acknowledgements

We would like to express our gratitude to the authors and compilers of the Charak Samhita for preserving invaluable ancient wisdom. Additionally, we extend our appreciation to the Django framework community for providing robust tools that enabled us to develop this NLP-based knowledge extraction system. Special thanks to Dr. Preenon Bagchi for their invaluable guidance and insights throughout the project. We are also thankful to the Institute of Bioscience and Technology, MGM University, Chh. Sambhajinagar for providing resources and facilities. Additionally, we extend our appreciation to the participants who generously shared their knowledge and experiences. This work would not have been possible without their contributions.

7. References

Chowdhary, K. & Chowdhary, K. R. (2020) Natural language processing. Fundamentals of artificial intelligence, 603-649.

Idris, N., Cik, M., Shamala, P. (2020) "A Generic Review of Web Technology: Django and Flask", International Journal of Advanced Computing Science and Engineering.

Kumar, J. (2021) Chatushka Methodology in Charaka Samhita. Journal of Ayurveda and Integrated Medical Sciences, 6(4), 333-336.

Lad, V. (2002). Textbook of Ayurveda: Fundamental Principles of Ayurveda 1 Albuquerque, NM The Ayurvedic Press.

Manikanta, K., Lokesh, P., Neha, R. (2021) "Visualization of Real World Enterprise Data using Python Django Framework", IOP Conference Series: Materials Science and Engineering, doi:10.1088/1757-899X/1042/1/012019

Nithya, R., Aashay, R., Vivekananda, M.(2018) “Django the python web framework”, International Journal of Computer Science and Information Technology Research.

Prashant, T. (2023) "Django: Developing web using Python", International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, pp. 303-306, doi: 10.1109/ICACITE57410.2023.10183246.

Rakesh, K., Himanshu, G., Ashitosh, S. (2021) “Django Web Development Simple & Fast”, International journal of creative research thoughts.

Satyavati CV. (1991) The Role of Ayurveda as Contemporary Medical Science and Therapeutic System. IIC Quarterly. 18 (2-3):71-82.

Shayam, A., Mukesh, N. (2020) “A Django Based Educational Resource Sharing Website: Shreic”, Journal of Scientific Research 64(01):238- 252,doi:10.37398/JSR.2020.640134

Schlutter, A., & Vogelsang, A. (2020) Knowledge extraction from natural language requirements into a semantic relation graph. In Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops (pp. 373-379).

Upadhyay, R., & Fujii, A. (2016). Semantic knowledge extraction from research documents. In 2016 Federated Conference on Computer Science and Information Systems (FedCSIS) (pp. 439-445). IEEE.

Vasiliev, Y. (2020) Natural language processing with Python and spaCy: A practical introduction. No Starch Press.