# Entity Extraction from Medical Manuscripts

## Preserving Ancient Wisdom for Modern Medicine

# ABOUT

The objective of this project is to develop an efficient system for extracting entities from Sanskrit texts using advanced natural language processing (NLP) techniques. By leveraging pretrained models specifically designed for Sanskrit, the system aims to facilitate better understanding and analysis of ancient texts, thus contributing to research in linguistics, history, and cultural studies. The output includes various linguistic annotations, which can be used for further analysis or applications in academic research.

# INTRODUCTION

Sanskrit, one of the oldest languages in the world, holds a wealth of knowledge in its ancient texts, including religious scriptures, philosophical works, and historical documents. However, extracting meaningful information from these texts poses significant challenges due to the complexity of the language and the richness of its grammatical structure. This project seeks to address these challenges by utilizing state-of-the-art NLP models for entity extraction, focusing on tasks such as lemma generation and morphological analysis.

The need for this project arises from:

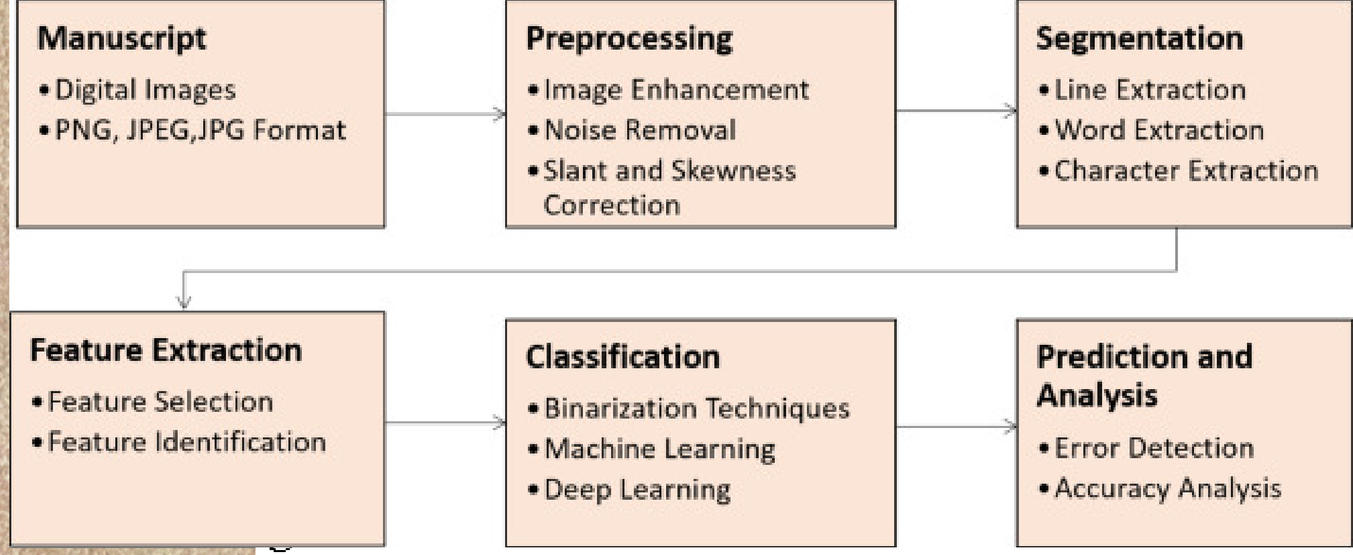
- Preservation of Knowledge: Digitizing and analyzing Sanskrit texts can help preserve cultural heritage.
- Facilitating Research: Efficient extraction of entities can assist researchers in linguistic studies, providing insights into ancient civilizations and languages.
- Educational Applications: The extracted data can be used in educational tools to teach Sanskrit linguistics and grammar.

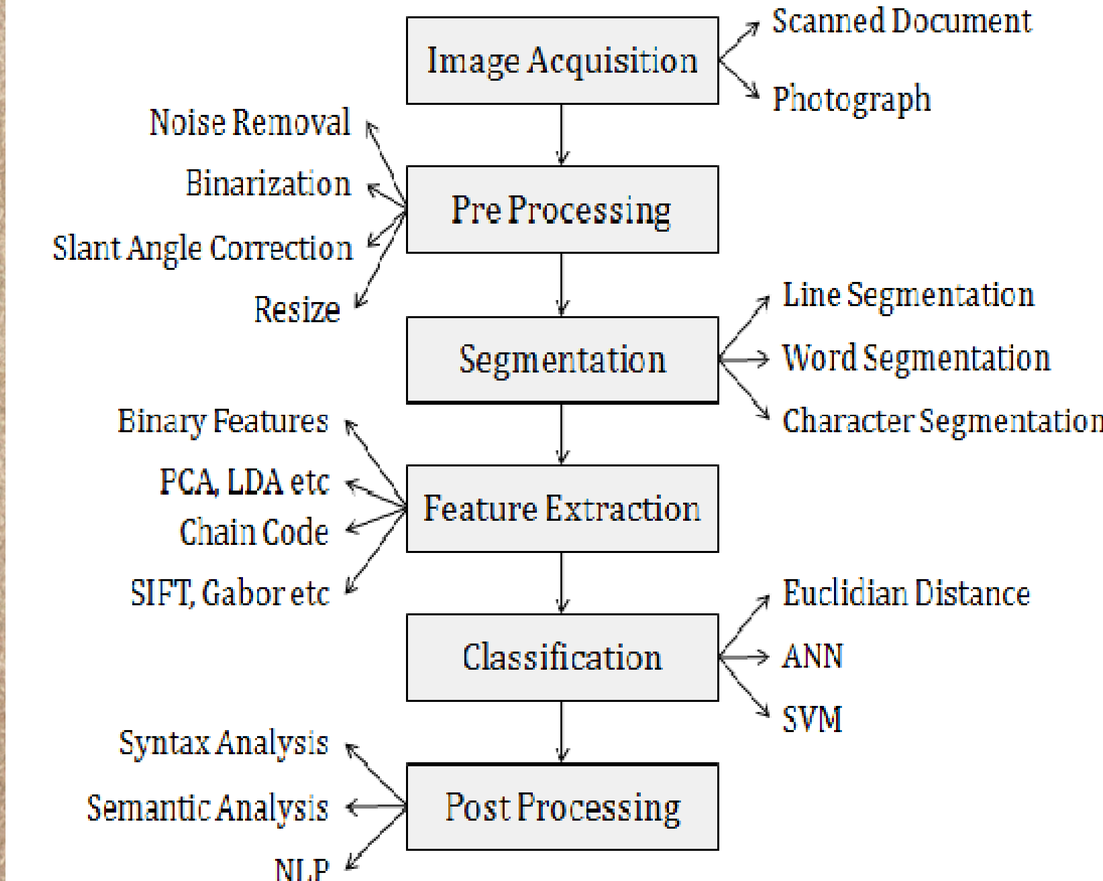
Overall Process

1. Data Preparation

Input Text: Prepare a dataset containing Sanskrit texts from which entities will be extracted. This data may include various manuscripts or literary sources.

Annotation: Ideally, a ground truth dataset is prepared, consisting of annotated entities for evaluation purposes. This dataset serves as a benchmark to measure the model's performance.



- Text Cleaning and Tokenization Process
- Objective: The purpose of the script is to clean and tokenize Sanskrit text by removing unwanted characters and standardizing the format for further processing.
- Steps Involved:
- Load Extracted Text:
- The script reads the Sanskrit text from a provided file (extracted_text.txt).
- Remove Non-Sanskrit Characters:
- Unwanted characters, such as punctuation, numbers, and other non-Sanskrit characters, are filtered out. Only characters in the Unicode range for Devanagari script (Sanskrit) are retained.
- Normalize Whitespace:
- Extra spaces, newlines, and other forms of irregular whitespace are removed, leaving the text in a clean format with consistent spacing.
- Tokenization:
- The cleaned text is split into individual tokens (words) using whitespace as a delimiter.
- Save Cleaned Output:
- The cleaned and tokenized text is saved into a new file (cleaned_tokens.txt) for further use in processing, analysis, or linguistic tasks.
- Benefits:
- Prepares raw text for linguistic analysis: This step is crucial for preparing data to be used in natural language processing (NLP) tasks.
- Removes noise: Ensures only relevant text is kept, improving the accuracy of downstream tasks like translation, sentiment analysis, or other text-based computations.

# 1.Segmentation

Segmentation: The scanned images are segmented into smaller, manageable sections. This step is crucial for isolating individual characters or lines of text, which helps in more accurate recognition and processing .

# 2. Preprocessing

Preprocessing techniques, such as resizing, grayscale conversion, noise reduction, and binarization, are applied to enhance the legibility of the text. These methods improve the quality of the images and make the text clearer.

# 3. Augmentation

Data augmentation methods are used to create modified versions of the scanned images. This step helps increase the robustness and accuracy of the OCR model by providing a more diverse set of training data.

# 4. Feature Extraction

Important features, such as edges, contours, and shapes, are extracted from the preprocessed images. These features are critical for recognizing the characters and distinguishing between different symbols and scripts.

# PROCESS

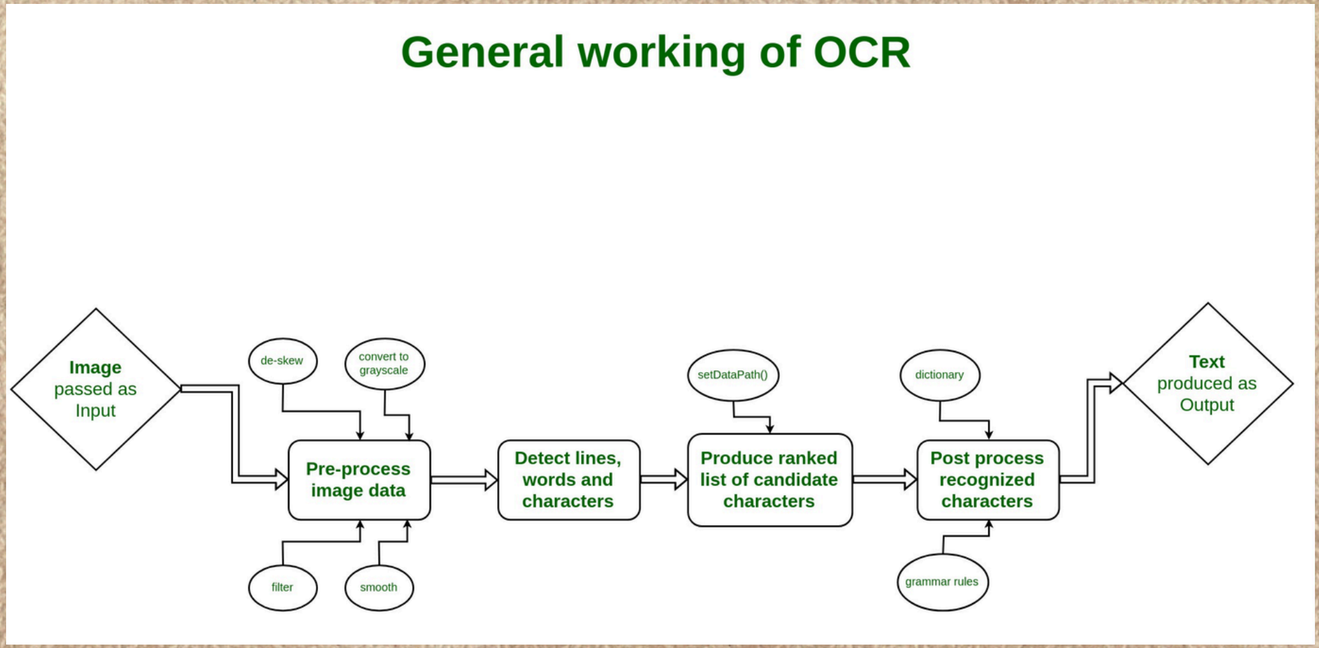
Tokenization - Handling Sanskrit-specific challenges like sandhi

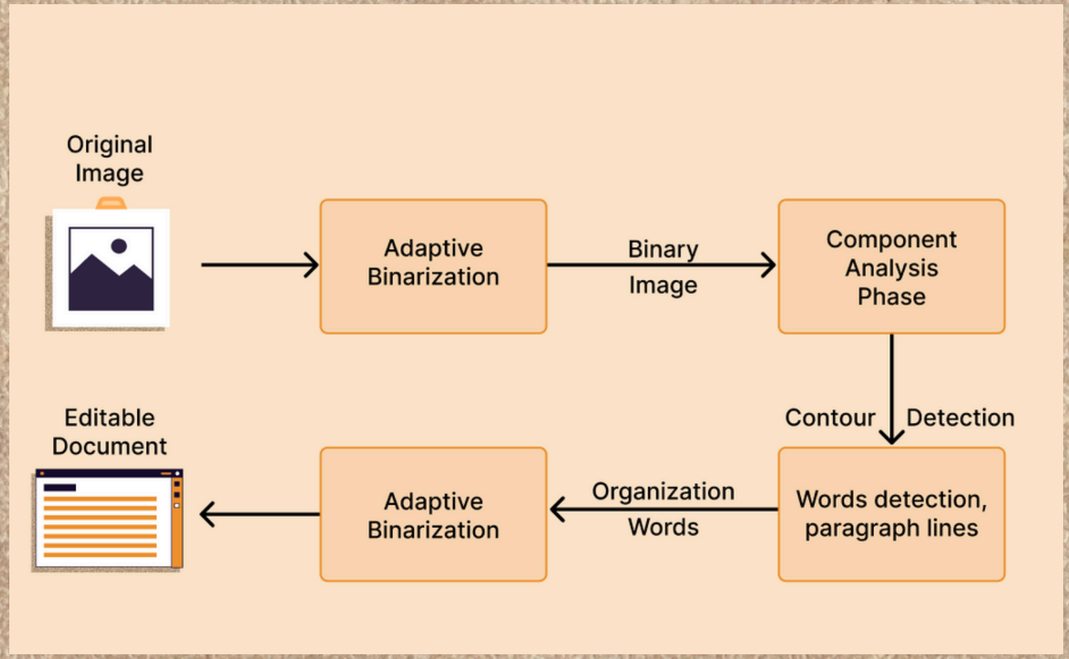
Annotated datasets - Training resources for better accuracy

Normalization - Ensuring consistency in character encoding

Sanskrit medical lexicons - Contains lists of plants, diseases, and treatments

Optical Character Recognition (OCR) technology is pivotal for digitizing and preserving ancient manuscripts. The OCR model's role is to convert images of manuscripts into digital text, making these valuable texts accessible for research and archiving. For ancient Indian manuscripts, especially Dravidian palm leaf scripts, OCR models must handle unique challenges such as degraded text, complex script styles, and missing characters. Key Challenges in OCR for Ancient Manuscripts: Degraded Quality: Manuscripts may be faded, damaged, or have irregularities due to age. Complex Scripts: Dravidian scripts have intricate and diverse characters. Non-Standard Fonts and Styles: Variations in handwriting styles and historical fonts. Missing Text: Incomplete or missing characters due to damage. Key Considerations: Preprocessing: Enhance image quality through binarization, noise removal, and contrast adjustment.



# Various ocr techniques results so we used best one in those

| | |
|---|---|
| Sample Image 1 | स्थितौ मध्ये समाप्तौ पितृसोमकौ' इति शुक्लादिमाससामान्यलक्षणम् । ३—वृष्त्याद्यर्थमित्यत्रादिशब्देन |
| Ground Truth | स्थितौ मध्ये समाप्तौ पितृसोमकौ ' इति शुक्लादिमाससामान्यलक्षणम् । ३ - वृष्त्याद्यर्थमित्यत्रादिशब्देन |
| **IndicOCR-v2** | स्थितौ मध्ये समाप्तौ पितृसोमकौ ' इति शुक्लादिमाससामान्यलक्षणम् । ३ - वृष्त्याद्यर्थमित्यत्रादिशब्देन |
| CNN-RNN | स्थितौ मध्ये समाप्तौ पितृसोमकौ ' इति शुक्लादिमाससामान्यलक्षणम् । ३ - वृष्त्याद्यर्थमित्यत्रादिशब्देन |
| Google-OCR | स्थिती मध्ये समाप्तौ पितृसोमकौ ' इति शुक्लादिमाससामान्यलक्षणम् । ३ - वृष्त्याद्यर्थमित्यत्रादिशब्देन |
| Ind.senz | स्थितौ मध्ये समाप्तौ पितृसोमकौ इति शुक्लादिमाससामान्यलक्षेणम् । ३ - वृष्त्याद्यर्थमित्यत्रादिशब्देन |
| Sample Image 2 | उत्तरा, संकल्पकाले सत्त्वात् । वैषम्येण तदेकदेशस्पर्शे तु तदाधिक्यवती ग्राह्येति । 'मन्दवारे प्रदोषोऽयं |
| Ground Truth | उत्तरा , संकल्पकाले सत्त्वात् । वैषम्येण तदेकदेशस्पर्शे तु तदाधिक्यवती ग्राह्येति । ' मन्दवारे प्रदोषोऽयं |
| **IndicOCR-v2** | उत्तरा , संकल्पकाले सत्त्वात् । वैषम्येण तदेकदेशस्पर्शे तु तदाधिक्ययवती ग्राह्येति । ' मन्दवारे प्रदोषोऽयं |
| CNN-RNN | उत्तरा , संकल्पकाले सत्त्वात् । वैषम्येण तदेकदेशस्पर्शे तु तदाधिक्यवती ग्राह्येति । ' मन्दवारे प्रदोषोऽ यं |
| Google-OCR | उत्तरा , संकल्पकाले सत्त्वात् । वैषम्येण तदेकदेशस्पर्शे तु तदाधिक्यवती प्रायति । ' मन्दवारे प्रदोषोऽयं |
| Ind.senz | ?ए संकल्पकाले सब्बुवोत् । वैषड्येण तदेकदेशस्पर्शे तु तदाधिक्यवती ग्राह्येति । 'मन्दवारे प्रदोषोऽयं |
| Sample Image 3 | 'अरुणोदये तु संप्राप्ते स्नानकाले विचक्षणः । माधवाङ्घ्रियुगं ध्यायन् यः स्नाति सुरपूजितः ॥' |
| Ground Truth | ' अरुणोदये तु संप्राप्ते स्नानकाले विचक्षणः । माधवाङ्घ्रियुगं ध्यायन् यः स्नाति सुरपूजितः ॥ ' |
| **IndicOCR-v2** | ' अरुणोदये तु संप्राप्ते स्नानकाले विचक्षणः । माधवाङ्घ्रियुगं ध्यायन् यः स्नाति सुरपूजितः ॥ ' |
| CNN-RNN | ' अरुणोदये तु संप्राप्ते स्नानकाले विचक्षणः । माधवाङ्घ्रियुगं ध्यायन् यः स्नाति सुरपूजितः ॥ ' |
| Google-OCR | ' अरुणोदये तु संप्राप्ते स्नानकाले विचक्षणः । माधवाङ्घ्रियुगं ध्यायन् यः स्नाति सुरपूजितः ॥' |
| Ind.senz | ' अरुणोदये तु संप्राप्ते स्नानकाले विचक्षणः । माधवाङ्घ्रियुगं ध्यायन् यः स्नाति सुरशइजतः ।।' |
| Sample Image 4 | ग्निश्च न शिष्टग्रहणोचितः ॥' मनुः—'दक्षिणेन मृतं शूद्रं पुरद्वारेण निर्हरेत् । पश्चिमोत्तरपूर्वै- |
| Ground Truth | ग्निश्च न शिष्टग्रहणोचितः ॥' मनुः - ' दक्षिणेन मृतं शूद्रं पुरद्वारेण निर्हरेत् । पश्चिमोत्तरपूर्व- |
| **IndicOCR-v2** | ग्निश्च न शिष्टग्रहणोचितः ॥' मनुः - ' दक्षिणेन मृतं शूद्रं पुरद्वारेण निर्हरेत् । पश्चिमोत्तरपूर्व- |
| CNN-RNN | निश्च न शिष्टग्रहणोचितः ॥' मनुः - ' दक्षिणेन मृतं शूद्रं पुरद्वारेण निर्हरेत् । पश्चिमोत्तरपूर्वै- |
| Google-OCR | मिश्च न शिष्टग्रहणोचितः ॥' मनुः-'दक्षिणेन मृतं शूद्रं पुरद्वारेण निर्हरेत् । पश्चिमोत्तरपूर्वै |
| Ind.senz | ग्निश्च न शिष्टग्रहणोचितः ।।' मनुः-'दक्षिणेन मृत स्फुं पुरद्वारेण ओइन्र्हेरेत् ए । पश्चिमोत्तरश्मैं |

# ByT5-Sanskrit Model

The ByT5-Sanskrit model is a variant of Google's ByT5 architecture, designed to handle byte-level inputs instead of token-based inputs. This approach is particularly useful for Sanskrit, where word formation is highly complex. The model processes raw byte sequences, making it effective for tasks like word segmentation, lemmatization, and morphosyntactic tagging. Its unified framework allows it to handle multiple tasks without needing task-specific architectures, making it efficient for Sanskrit NLP tasks.

works as a unified model that handles a range of tasks, such as word segmentation, lemmatization, and morphosyntactic tagging, all under one framework. ByT5 leverages token-free byte-level inputs, making it adaptable to the unique structure of Sanskrit, which has complex word formations and inflections. This approach simplifies preprocessing and allows the model to generalize well across tasks, eliminating the need for separate models for each.

The model demonstrates high performance across tasks like word segmentation, lemmatization, and morphosyntactic tagging by employing a single architecture for multiple functions. This efficiency reduces the necessity for separate models and streamlines the application of NLP techniques to Sanskrit texts.

a serialization process for morphosyntactic tagging in which specific tags are appended to words based on their grammatical features. Each word is annotated with detailed information about its case, gender, number, tense, mood, person, etc. The abbreviated tags for each feature are highlighted in red, and spaces are used as separation tokens between words.

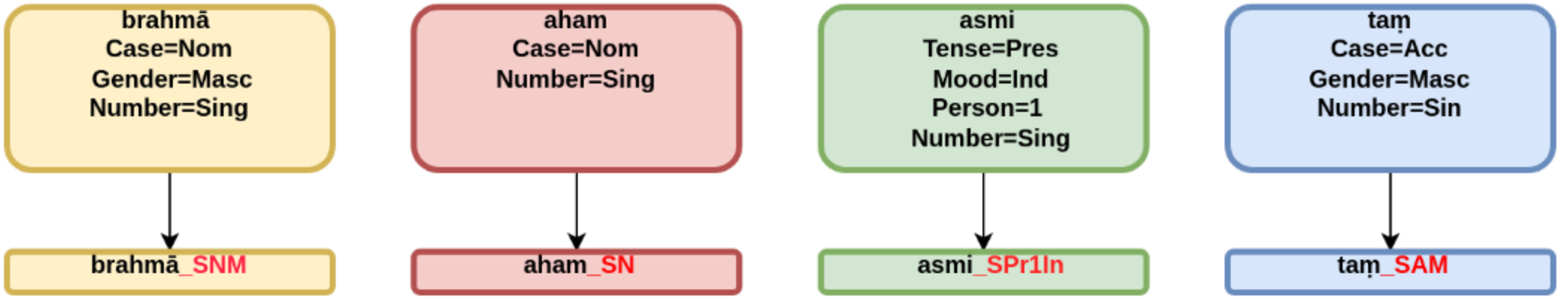



Figure 1: Serialization for the morphosyntactic tagging task. The abbreviated tags are highlighted in red. We use spaces as separation token between words.

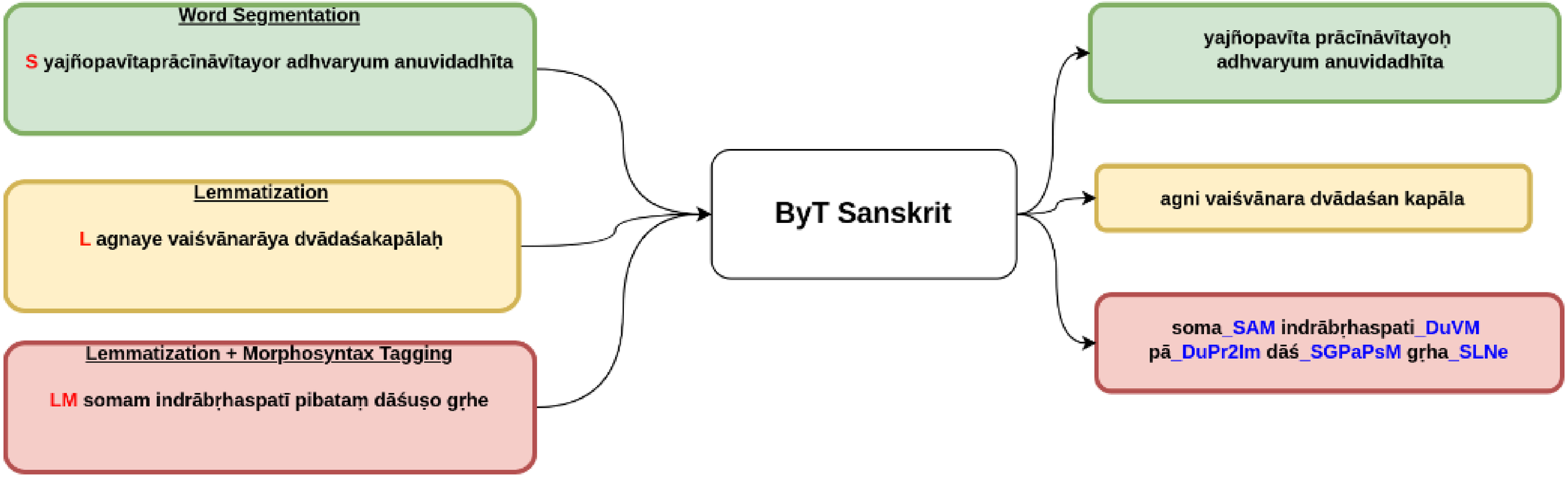
Key points from the image:

- Brahmā (Nominative, Masculine, Singular): Serialized as brahmā_SNM.
- Aham (Nominative, Singular): Serialized as aham_SN.
- Asmi (Present, Indicative, 1st Person, Singular): Serialized as asmi_SPr1In.
- Tam (Accusative, Masculine, Singular): Serialized as tam_SAM.

Each abbreviation represents specific morphosyntactic features such as:

- S = Singular
- N = Nominative case
- M = Masculine
- Pr = Present tense
- In = Indicative mood

This type of tagging is useful for tasks like syntactic parsing or machine translation, where understanding the grammatical roles of words is critical for accurate processing.




- **Word Segmentation (S):** This involves splitting Sanskrit sentences into their constituent words. For example:
- **Input:** yajñopavita prācīnāvītayoḥ adhvaryum anuvidadhīta
- **Output:** yajñopavita prācīnāvītayoḥ adhvaryum anuvidadhīta
- **Lemmatization (L):** This refers to the process of converting words to their base or dictionary form. Example:
- **Input:** agnaye vaiśvānarāya dvādaśakapālaḥ
- **Output:** agni vaiśvānara dvādaśan kapāla
- **Lemmatization + Morphosyntactic Tagging (LM):** This includes both lemmatization and the addition of morphosyntactic tags to indicate grammatical information such as case, number, and gender.
- Example:
- **Input:** somam indrābṛhaspati pibatam dāśuṣo gṛhe
- **Output:** soma_SAM indrābṛhaspati_DuVM pā_DuPrZIm dāś_SGPaPsM gṛha_SLNe

https://github.com/ihdia/sanskrit-ocr
https://arxiv.org/html/2409.13920

# THANK YOU