1. **Data Collection through Web Scraping**

Target Sources: Identify target websites that provide high-quality textual content in the Indic languages. These could include:

* News websites in Tamil, Hindi, Telugu, etc.
* Government portals providing official documentation.
* Blogs and articles in Indic languages.
* Literature and academic repositories (e.g., Tamil epics, Hindi poetry, etc.).
* Indic language Wikipedia pages.

Web Scraping Setup: Write web scraping scripts in Python using libraries like BeautifulSoup, Scrapy, or Selenium for dynamic content. Key components:

Text extraction: Extract titles, paragraphs, and any relevant textual data from the pages.

Language-specific Filters: Use filters to ensure that only content in the target Indic language is scraped. langdetect can be helpful to detect the language if needed.

Pagination Handling: Handle multiple pages, categories, and archives on news websites or blogs to extract more comprehensive data.

**⁠Preprocessing Scraped Text:**

Normalize the text by removing any HTML tags, advertisements, or unnecessary elements.

Remove noise such as comments, advertisements, and irrelevant sections.

Language validation: Ensure the extracted content is in the desired Indic language using libraries like langdetect.

Tokenization and Cleaning: Clean and tokenize text data specific to the Indic language. Use libraries like indic-nlp-library for tokenization and text normalization in Indic languages.

**2. PDF Preprocessing**

Source PDFs: PDFs in Indic languages can be obtained from:

* Government archives (e.g., policies, notifications in Hindi/Tamil).
* Academic papers and literature available in Indic languages.
* Online repositories offering books, historical documents, or poetry in the native languages.

Text Extraction from PDFs: Use PyPDF2, pdfplumber, or Tesseract OCR for extracting text from PDFs, especially for scanned documents.

Handling scanned PDFs: For scanned PDFs that don’t contain selectable text, use OCR (Optical Character Recognition) to extract text. Tesseract can handle Tamil, Hindi, and other Indic scripts.

Table extraction: If PDFs contain tables (e.g., government statistics), use camelot-py or tabula-py to extract tabular data.

**Cleaning and Preprocessing:**

Remove unwanted characters and noise (e.g., page numbers, footnotes, etc.).

Text normalization: Use Indic language-specific libraries to ensure proper character encoding.

Stopwords removal: Create or use pre-built stopword lists for the Indic language being used (e.g., stop words for Hindi, Tamil).

Segmentation: For large documents, segment text by sentences or paragraphs, using NLP techniques suitable for the target language.

**Metadata:**

Collect metadata for each document scraped or processed, including:

* Source URL (for web scraped data).
* Document type (news article, government document, academic paper, etc.).
* Language (detected using langdetect or predefined by the source).
* Date of extraction.