نہیں ہے نا امید
اقبال
اپنی کشت ویراں سے
ذرا نم ہو
تو یہ مٹی بہت
زرخیز
ہے
ساقی

Urdu Notebook LLM: AI-Powered Urdu Summarizer & Reader Assistant

Abstract

Urdu Notebook LLM is a personal AI assistant designed to make Urdu literature, scanned PDFs, and handwritten notes accessible and intelligent. The project integrates OCR, language modeling, summarization, and retrieval features to build a system that enables summarizing, searching, and interacting with Urdu texts in a notebook-style environment. By combining OCR with natural language processing, the assistant allows users to upload Urdu PDFs, digitize them, and generate summaries, translations, or Q&A responses. The project bridges a gap in current tools that heavily favor English, creating a specialized platform for Urdu readers, students, and researchers.

Table of Contents

Introduction

While AI summarizers and assistants exist widely for English and other major languages, tools for Urdu remain limited. Most Urdu PDFs are scanned images and cannot be processed directly by modern LLMs. This project aims to solve that challenge by creating a notebook-like assistant specifically for Urdu. Users will be able to upload Urdu text (books, poetry, academic material), process it with OCR, and then interact with it through summarization, Q&A, and keyword-based retrieval. The goal is to make Urdu literature as digitally accessible as English, while preserving linguistic and cultural richness.

Project Description

Core Components:

- **OCR & Text Input:** Convert scanned Urdu PDFs/images into text. Additionally, allow direct ingestion of digital Urdu text (episodes/chapters from novels collected via websites or personal archives).
- **Corpus Building:** Create a growing dataset of Urdu literature (e.g., serialized novels, personal collections). This dataset will be indexed for retrieval and used to improve the system's contextual understanding.
- **LLM Summarization:** Summarize Urdu text (chapters/episodes) into concise versions.
- **Semantic Search & Q&A:** Enable the user to ask questions about specific novels or episodes (*"What happened in episode 14 of [novel name]?"*).
- **Notebook Interface:** Save summaries, notes, and Q&A history, building a personalized "reading notebook."

Project Objectives:

- Develop a prototype Urdu-first AI notebook assistant.
- Enable OCR for scanned PDFs **and direct ingestion of episodic Urdu novels**.
- Implement summarization and Q&A for Urdu text, focused on the fed corpus.
- Provide search and retrieval across the collected episodes and texts.
- Build a simple notebook-style interface for interaction.

Methodology / Approach

Step-by-Step Development Plan:

1. **Requirement Analysis** – Identify user needs (summarization, Q&A, search).
2. **OCR Module** – Integrate Tesseract OCR with `urd` language pack.
3. **Summarizer Module** – Use a pre-trained Urdu summarizer (BART Urdu) or Hugging Face Urdu LLM.
4. **Q&A Module** – Implement retrieval-augmented generation (RAG) pipeline for Urdu.

5. **Notebook UI** – Build a minimal interface for uploading PDFs, viewing summaries, and searching.
6. **Testing & Refinement** – Validate accuracy of OCR and quality of summaries.

System Design:

- Input: PDF/Image → OCR → Clean Urdu Text
- Processing: Summarizer & Retriever → Notes / Answers
- Output: Urdu Summaries, Searchable Notebook, Highlighted Passages

Expanded

The methodology for developing the Urdu Notebook LLM is designed to create a personalized AI assistant capable of understanding, summarizing, and answering questions about Urdu novels. The approach integrates data collection, model training, and a user-facing notebook interface in a structured workflow.

1. Data Collection and Corpus Building

- **Sources:** The dataset includes scanned PDFs, images of Urdu novels, digital text from websites, personal archives, and episodic novel collections. Additionally, reader reviews, discussions, and personal interpretations are collected to enrich contextual understanding.
- **OCR and Text Input:** Scanned documents are processed using OCR tools (e.g., Tesseract, EasyOCR) to convert images into editable Urdu text. Digital text is ingested directly.
- **Data Organization:** Each novel is segmented by episodes or chapters and stored in a structured format (e.g., JSON), including fields such as episode number, novel name, text, summary, and notes.

2. Preprocessing and Preparation for Model Training

- **Text Cleaning:** Extracted text is normalized, removing noise, fixing encoding issues, and standardizing formatting.
- **Segmentation:** Long chapters are split into smaller chunks to ensure compatibility with model input limits.
- **Question-Answer Pair Creation:** Reviews, discussions, and summaries are converted into structured Q&A pairs to provide the model with supervised learning examples.

3. Model Training and Fine-Tuning

- **Model Selection:** A multilingual or Urdu-specific LLM (e.g., mT5, UrduT5, BLOOMZ) is used as the base model.
- **Fine-Tuning:** The model is trained on the collected corpus, including episode texts, user-generated summaries, and review-based question-answer pairs.
- **Iterative Learning:** As more episodes are processed and more summaries are added, the model is incrementally updated to improve accuracy and contextual understanding.

4. Summarization and Semantic Understanding

- The fine-tuned model generates concise summaries of episodes or chapters, capturing key events, character developments, and plot points.
- Semantic search capabilities enable retrieval of specific episodes or sections based on user queries, allowing targeted question-answering about novels.

5. Notebook Interface

- A personalized notebook interface records summaries, notes, and Q&A interactions.
- Users can access and search through past episodes, track novel progress, and maintain a comprehensive reading history.
- The notebook serves both as a knowledge repository and as a feedback loop for further model refinement.

6. Feedback Loop and Model Enhancement

- User-generated summaries and insights from reviews/discussions form the high-quality training data that guides the model.
- This iterative process ensures that the Urdu Notebook LLM continuously improves, producing summaries and answers that reflect both accurate content and contextual interpretations.

Workflow Summary:

1. Collect and OCR novel episodes.
2. Preprocess and segment text.
3. Generate summaries and extract Q&A from reviews/discussions.
4. Fine-tune the model on this data.
5. Use the model for summarization, Q&A, and semantic search.
6. Store outputs in a notebook interface.
7. Continuously update the model with new summaries and insights.

Technology Stack

- **Operating System:** Linux / Windows
- **Programming Language:** Python (OCR, LLM integration, UI)
- **OCR:** Tesseract OCR (Urdu model)
- **AI Models:** Hugging Face (`bart-urdu-summarizer`, `urdu-ocr`, multilingual LLMs)
- **Frameworks:** PyTorch, Hugging Face Transformers, LangChain (for RAG)
- **Database:** SQLite or FAISS (for vector search)
- **Interface:** Streamlit (prototype), Flask/React (expandable)
- **Version Control:** GitHub

Development Methodology

Agile & Iterative — Build the Urdu Notebook in increments:

- First, OCR + Summarizer MVP.
- Then, add search + Q&A.
- Finally, notebook interface & saving highlights.

Testing Approach

- **OCR Testing:** Accuracy on printed vs handwritten Urdu.
- **Summarizer Testing:** Compare human summaries with model summaries.
- **Search Testing:** Verify correct retrieval of Urdu passages.
- **System Testing:** Full workflow (upload → summary → Q&A).

Roadmap

Week 1: Environment setup, Python libraries, Tesseract Urdu OCR test.
Week 2: Build OCR-to-text pipeline, clean output formatting.
Week 3: Integrate Urdu summarizer model, test with short stories/articles.
Week 4: Implement keyword search + vector search (FAISS).
Week 5: Add Q&A chatbot layer (LangChain + LLM).
Week 6: Build Streamlit notebook UI, integrate modules, final testing.

Resources and Tools

Hardware:

- PC with Python + GPU (optional but recommended).

Software:

- Python, Tesseract OCR, Hugging Face Transformers, Streamlit, FAISS.

References:

- Urdu OCR datasets, Hugging Face Urdu models, research papers on Urdu summarization.

Deliverables

- MVP Urdu Notebook AI assistant (OCR + Summarizer + Search).
- Streamlit interface for uploading PDFs and viewing summaries.
- Modular Python codebase for future extensions (Q&A, highlights, TTS).
- Documentation of setup, usage, and limitations.