OpenBookLM Processing Flow

This document describes the data flow between different processing scripts in the OpenBookLM project.

Directory Structure

```
ROOT/
├── input/              # Input PDFs
│   ├── *.pdf          # Source PDF files
└── output/
    ├── text/          # Extracted text from PDFs
    │   └── *.txt
    ├── summaries/     # Generated summaries
    │   └── *.txt
    ├── podcast_dialogue/  # Generated podcast dialogue scripts
    │   └── *.txt
    └── podcast_audio/     # Final podcast audio files
        └── *.wav
```

Processing Flow

1. **PDF to Text** (`pdf_to_text.py`)
   - Input: `ROOT/input/*.pdf`
   - Output: `ROOT/output/text/*.txt`
   - Function: Extracts text content from PDF files
2. **Text to Summary** (`text_to_summary_400words.py`)
   - Input: `ROOT/output/text/*.txt`
   - Output: `ROOT/output/summaries/*.txt`
   - Function: Generates ~400 word summaries from text content

3. **Summary to Podcast Dialogue** (`summary_to_podcast_dialogue.py`)
    - Input: `ROOT/output/summaries/*.txt`
    - Output: `ROOT/output/podcast_dialogue/*.txt`
    - Function: Creates conversational dialogue scripts from summaries
4. **Podcast Dialogue to Audio** (`podcast_dialogue_to_podcast_audio.py`)
    - Input: `ROOT/output/podcast_dialogue/*.txt`
    - Output: `ROOT/output/podcast_audio/*.wav`
    - Function: Converts dialogue scripts to audio using text-to-speech

Usage

Run the scripts in sequence:

```
./backend/pdf_to_text.py
./backend/text_to_summary_400words.py
./backend/summary_to_podcast_dialogue.py
./backend/podcast_dialogue_to_podcast_audio.py
```

Each script will process all files in its input directory and generate corresponding output files in its output directory.