Chapter 1: Introduction

This is the introduction to our test document. It contains several paragraphs of text that should be extracted and processed by the PDF-to-audiobook pipeline.

The pipeline consists of four stages: parsing, cleaning, chunking, and synthesis. Each stage processes the text in a specific way to prepare it for audio output.

In this chapter, we will discuss the motivation behind building such a tool. Converting PDF documents to audio format allows people to consume content while commuting, exercising, or doing household chores.

Chapter 2: Methodology

Our approach uses a combination of rule-based and ML-based techniques. For simple PDFs, we use PyMuPDF for fast text extraction. For complex layouts with multi-column text, tables, and figures, we use Docling.

The text cleaning stage uses large language models (LLMs) to normalize abbreviations like Dr., Fig. 3, and e.g. into their spoken forms. It also removes artifacts like page numbers, headers, and citation markers [1].

After cleaning, the text is split into chunks at sentence boundaries using spaCy's natural language processing capabilities. Each chunk respects the character limits of the chosen TTS engine.

Chapter 3: Results

We tested our pipeline on a variety of PDF documents, including academic papers with complex layouts and simple ebook-style documents.

The results show that the auto-detection mechanism correctly identifies complex layouts in approximately 95% of cases. The PyMuPDF parser handles simple documents with near-perfect accuracy.

Audio quality was evaluated by 50 listeners who rated the output on a scale of 1 to 5. The average rating was 4.2 for Kokoro TTS and 4.6 for ElevenLabs.

We conclude that the pipeline successfully converts PDF documents into natural-sounding audiobooks with minimal manual intervention required.