Report

PDF (Portable Document Format) is a versatile file format created by Adobe that encapsulates a complete description of a fixed-layout document, including text, fonts, graphics, and other information needed to display it. The structure of a PDF document is hierarchical and consists of several key components crucial for accessibility:

1. **Root Object:** Contains information about the document structure, including references to other objects like the page tree, metadata, and outlines.
2. **Page Tree:** An index that organizes the pages of the PDF document, allowing navigation and access to individual pages.
3. **Tags (Structure Elements):** Special markers within a PDF that define the logical structure of the content. They are crucial for assistive technologies as they help in navigating and reading the document.
4. **Structure Root (StructTreeRoot):** The top of the structure tree that contains the logical order of the content. It organizes the tagged elements and ensures they follow a meaningful sequence, essential for screen readers.
5. **MarkInfo:** Provides information about whether the PDF contains structure elements and whether it is tagged.
6. **Content Streams:** Sequences of PDF operators that describe the content on a page, including text, images, and graphical elements.

Methodology

Tools and Libraries

- **PikePDF:** An open-source library for reading and modifying PDF files. It allows the script to open the PDF document and check its structure and metadata.
- **PDFMiner:** A library for extracting text from PDF documents. It is used to extract text content for further analysis.

Accessibility Checks

The code performs the following basic accessibility checks:

Tagged PDF:

Tags in a PDF document provide a logical structure, allowing screen readers and other assistive technologies to navigate and interpret the content correctly. When a document lacks tags, users who rely on these technologies face difficulties in reading and understanding the document. The

absence of tags poses a significant barrier to accessibility, making the document challenging for individuals with disabilities. To ensure accessibility, it's crucial to verify the presence of tags using the `/MarkInfo` dictionary, which indicates whether a document is tagged and structured appropriately for assistive technologies.

Logical Reading Order:

A logical reading order ensures that the content flows meaningfully, which is crucial for users relying on assistive technologies. The absence of a `/StructTreeRoot` element means that the document lacks a defined reading order, making it difficult for screen readers to convey the content coherently. This absence results in a disjointed experience for users relying on screen readers, as they are unable to follow the document's content in a logical sequence. To ensure a coherent reading experience, it is essential to check for the existence of the `/StructTreeRoot` element, which defines the document's logical structure.

Alternate Text for Images:

Alternate text for images is essential for visually impaired users to understand the content conveyed by images. When images lack alternate text descriptions, these users miss out on important information, leading to an incomplete understanding of the document. In this case, several images in the document do not have alternate text descriptions. Without this alt text, visually impaired users cannot grasp the information the images are intended to convey. To ensure full accessibility, it is crucial to inspect image elements and verify that they contain appropriate alternate text descriptions.

Conclusion

The automated script provides a robust method for evaluating the accessibility of PDF documents. In this instance, the PDF under review failed to meet several basic accessibility standards. It was neither tagged nor had a logical reading order, and it lacked alternate text for images. These findings highlight significant barriers for users with disabilities, underscoring the need for improvements to make the document accessible.