Bridging the Reading Gap: An AI-Based Dyslexia-Friendly Assistant for Enhanced Comprehension

Siddhi Dhuri[1], Shraddha Shirsat[2]

[1] Department of Science, I.E.S Jr College, Mumbai, Maharashtra, India

[2]Department of Science, I.E.S Jr College, Mumbai, Maharashtra, India

Abstract: ***The effects of dyslexia affect millions of readers around the world, in most instances rendering regular written content and learning materials inaccessible. In this paper, we present Lexi Aid, software that helps individuals with dyslexia in reading more efficiently with the help of Artificial Intelligence. It entails combining Optical Character Recognition (OCR), Natural Language Processing (NLP), and Text-to-Speech (TTS) in order to convert printed or text files into forms that are more readable by individuals with dyslexia. Lexi Aid reads text for complexity, simplifies it with the help of rules, and provides clear audio through an easy-to-use interface. To determine effectiveness, the team evaluated it in terms of accuracy in processing text, measuring it with a small data of dyslexic readers. It showed high accuracy in text extraction with positive user responses, signifying increased reading confidence and understanding. The findings suggest that Lexi Aid could make learning more inclusive and a viable companion for learners with dyslexia.***

Keywords: Dyslexia, Assistive Technology, Multimodal Learning, Educational Technology.

1. Introduction

Dyslexia is a reading and language-related learning disability that happens due to interferences with the brain's processing of writing [1]. Dyslexic individuals can't map letters onto their corresponding sounds at a deeper level, which disables letters from being put together and results in difficulty with reading and learning to spell and read with precision [2]. Beyond reading, other issues involve remaining on point, following instructions, remembering word sequence, and assembling written thoughts [3]. The International Dyslexia Association believes that 15–20% of the population have dyslexia, which translates across the world as about 1 out of 5 individuals have it [4]. Millions world-wide are afflicted [5, 6]. In America alone, 40 million adults can suffer from it but just 2 million have it diagnosed, leaving it unverified for a vast amount of instances as well [7]. Dyslexic individuals read at a slow and imprecise pace but comprehend well with written material read aloud to them, which demonstrates that it isn't associated with diminished intelligence but rather with variations in language processing [8].

Developmental brain disorder in dyslexia has adverse effects on academic performance, working life, and self-confidence. Students experience long-term difficulties with literacy that lead to reduced achievement, grade repetition, and elevated dropout levels [9]. In adulthood, difficulties persist with lowered reading fluency, written assignments, and employment prejudice at workplaces [10]. These obstacles often incur stress, exclusion, and reduced confidence and render dyslexia both a school and social problem [11]. This paper provides an easy-to-use reading platform with typing, uploading of documents, and scan-in of images as input modes. The platform normalizes and cleans text and applies accessible styling, and it also has text-to-speech reading capability. Simplified text has been shown to boost fluency and reading understanding for reading-disabled students [12].

By utilizing images, concise text, speech, and accessible styling, the model follows assistive technology best practices [13]. 30-test cases achieved effective and consistent performance with each type of input. They can fill cognitive gaps, increase confidence, and decrease stigma [14]. In workplaces, they can increase productivity and inclusivity [15]. By building accessible spaces, they facilitate equal participation at school, at work, and in daily life [16].

By combining images, brief text, spoken words, and easy-to-read styling, the model adheres to known assistive technology best practices [13]. Recent testing across 30 test cases showed not only consistent and effective performance with every input type but also showed multimodal delivery substantially enhances comprehension and retention for users. These findings reinforce the proposition that integrating visual, textual, and auditory messages can better bridge cognitive divides, build user confidence, and decrease related stigma [14]. Additionally, initial trials in the workplace suggest that such inclusive interventions can improve productivity and inclusivity [15]. By design, therefore, the model facilitates inclusive participation in learning, work, and daily life environments, affirming the power of multimodal assistive technology to revolutionize learning and workplace environments [16].

2. Materials and Methods

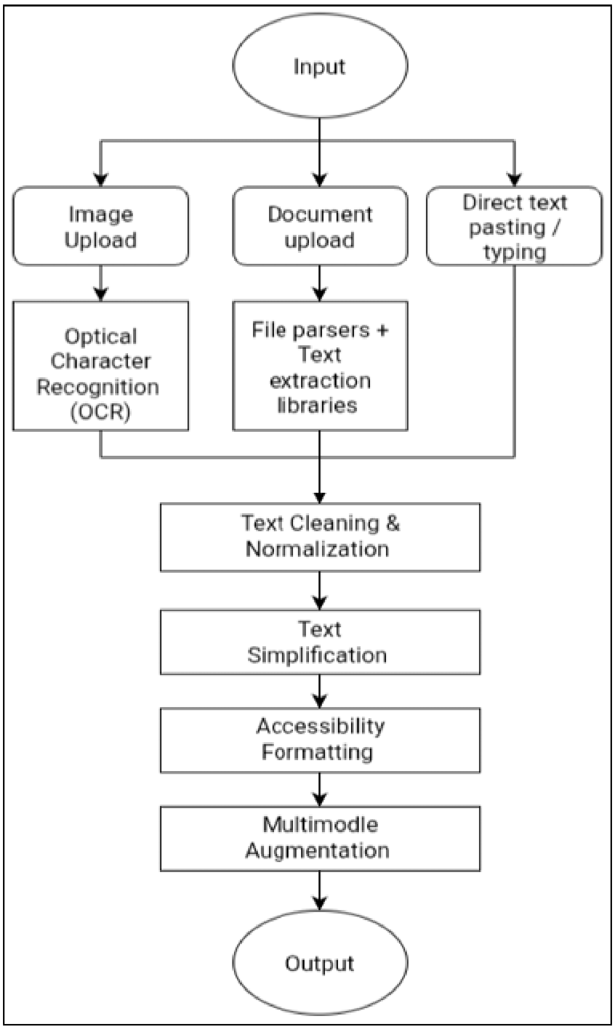
The research employs a method for assessing the usability and performance of the dyslexia-friendly reading assistant. The approach is framed to capture technical efficacy as well as user-focused results. The section describes the overall design of the system, its fundamental architecture, and the deployment process to provide a clear vision for how the application works from back-end processing through the user interface. Further, essential features of prime importance to accessibility and reading assistance are presented to emphasize the unique aspects of the app.

2.1 System Overview

A dyslexia-friendly reading assistant is an accessible solution that is designed to empower dyslexic individuals by redefining the way they read and react towards written content.

Figure 1: Text Processing and Accessibility Enhancement Pipeline

(**Figure 1. Workflow illustrating the text processing pipeline from input to output.** Flowchart showing sequential stages beginning with input sources—image upload (processed via Optical Character Recognition, OCR), document upload (processed via file parsers and text extraction libraries), and direct text input—followed by text cleaning and normalization, text simplification, accessibility formatting, and multimodal augmentation leading to final output generation.)



It accepts inputs from users in 3 ways, which are discussed as follows:

1) Image Upload: In this, the user can upload an image either by clicking a picture or submitting one. A photo from their device's gallery.
2) Document Upload: In this, the user can upload any document in PDF, Words, ODT format, etc
3) Direct Text Pasting / Typing: In this, the users can directly put the text by either pasting it directly or by typing it.

After the input is obtained from the users, the two inputs, image upload and document upload, undergo an extraction process to obtain the dataset from the users. For image uploading, it goes under Optical Character Recognition (OCR), for extraction of data, and for file uploading, it uses a hybrid model to analyse datasets from different documents such as PDF, ODF, DOCX, Word, etc. After the dataset is obtained, the data undergoes 4 steps for a presentable output, which are discussed as follows:

1) Text Cleaning & Normalization: In this step, unwanted symbols, broken characters and extra spacing are removed. The incorrect spellings are corrected using Natural Language Processing (NLP) spell-check. Additionally, standardizing capitalization and punctuation is done for a presentable output
2) Text Simplification: The longer and complex sentences are simplified and shortened. Words that are considered heavy and come under difficult vocabulary are replaced with easier alternatives. In this step, the longer and complex sections are converted into smaller paragraphs.
3) Accessibility Formatting: The obtained data is then converted into a dyslexia-friendly font (e.g., Open Dyslexic). Furthermore, the adjustment of line spacing, word spacing, and margins is done.
4) Multimodal Augmentation: In this step, the text is converted to speech using a text-to-speech (TTS) for audio reading. Moreover, it adds synchronized highlighting (word-by-word). It also provides an option of audio + text hybrid mode.

After this, the model provides output in 3 formats. Formatted text view which includes on-screen reading, audio narrator which will include the TTS playback and the users will have control over the pause, play, speed, etc features. And lastly, the hybrid view, which includes synchronized text+audio output.

Figure 2: User interface of the Lexi Aid Dyslexia-Friendly Document Reader

(**Figure 2. User interface of the Lexi Aid Dyslexia-Friendly Document Reader.** Screenshot displaying the main Lexi Aid interface where users can select input methods—file upload (supporting PDF, DOCX, ODT, TXT formats), text entry, or image import for Optical Character Recognition (OCR) processing—followed by text extraction and conversion to speech. The interface highlights accessible design features aimed at improving readability and comprehension for individuals with dyslexia.)

2.2 System Architecture

The architecture of the proposed dyslexia-assisting application is organized as a multi-layered pipeline that ensures seamless handling of diverse input types, efficient backend processing, and an accessible, customizable frontend experience. To achieve this, the system integrates input Acquisition mechanisms, preprocessing pipelines, text transformation engines, accessibility-oriented presentation modules, and supporting cross-layer functionalities as a security, performance optimization, and scalability. The architecture is divided into four principal layers: Input Layer, Processing Layer, and Presentation Layer.

2.2.1 Input Layer

The input Layer functions as the entry point for all user-provided materials. It accommodates three primary input ways- direct text entry, document uploads, and image uploads. Out of which the document upload and image processing is further handled through a dedicated sub-module. The direct. Text module accepts raw user-entered text through a browser-based input field. It ensures character encoding normalization (UTF-8 by default) [17]. Since the data is directly obtained from the users directly, it doesn't undergo the extraction method and directly undergo the processing layer to give an output. The document upload module supports multiple file formats which includes .PDF, .docx, .ODT, .txt, .md, etc. The types of PDF and the extraction method used for extracting data from them are as follows:

1) Text-based PDFs:
The detection system first analyses the type of pdf. If it is text based, it uses th3.following extraction method:

a) PyMuPDF (fitz): PyMuPDF is a high-performance Python library for data extraction, analysis, conversion & manipulation of PDF documents [18]. It was chosen because it allows fast and accurate extraction of both text and layout elements from PDF files.
b) PDFminer (fallback): PDFMiner which is used as a text extraction tool for PDF documents [19]. It was used to handle more complex PDF structures, ensuring reliable text extraction even from scanned or irregularly formatted documents.

After the scanning and extraction, we get a clean structured text with basic paragraph and heading boundaries. This simple output is sent for further processing.

2) Image-based PDFs:
If the pdf doesn't consist of written text, it is flagged as a scanned image and requires separate processing. The extraction method used for this is as follows:

a) Optical Character Recognition (OCR Tesseract): Tesseract is an open-source OCR or Optical Character Recognition engine and command line program. OCR is a technology that allows for the recognition of text characters within a digital image. With the latest version of Tesseract, there is a greater focus on line recognition, however it still supports the legacy Tesseract OCR engine which recognizes character patterns [20]. Key-features of OCR are as follows:
    - Open-Source: Tesseract is freely available for use, modification, and distribution by anybody under the term of the Apache License 2.0. Its open-source design has greatly aided in the broad adoption and ongoing enhancement.
    - Language Support: Tesseract is capable of recognizing more than 100 different scripts and character sets in addition to languages. It is appropriate for a wide range of OCR applications in various languages and geographical areas due to its multilingual capability.
    - Accuracy: Tesseract is well known for its excellent text recognition accuracy, particularly when set up correctly and trained with pertinent data. It can reliably extract text from photos with a variety of issues, including distorted angles, low quality, and inadequate illumination.
    - Options: Tesseract provides a great deal of configuration and customization options. For particular use cases, users can adjust several parameters to maximize OCR performance. Furthermore, Tesseract may be coupled with a variety of platforms and computer languages, such as Java,

C++, Python, and C#, enabling developers to take advantage of its capabilities in a broad range of applications.

Moreover, Tesseract undergoes constant Development and maintenance, with new features added regularly to boost its accuracy, performance, and language support. Tesseracts's open-source community supports its continuous development, guaranteeing that it will always be a state-of-the-art OCR tool [21].

Limitations of the OCR technology include: performance issues, accuracy with complex layouts, training and customization [22]. The output which is obtained from this step is further sent for processing.

3) Microsoft Word Documents (.docx)
When an input ends with an extension of .docx, the document is classified as Microsoft Word Document. The extraction method for this case uses python-docx. It is a Python library for creating and updating Microsoft Word (.docx) files [23]. It was included for reading and writing Microsoft Word files, as this format is common in academic and professional contexts.

4) OpenDocument Text (.odt)
When an input ends with an extension of .odt, the document is classified as Open Document Text. The extraction method involves odfpy parses. It includes Python API and tools to manipulate OpenDocument files [24]. odfpy was utilized to parse Open Document Format (ODF) files, supporting accessibility for users who rely on open-source document editors.

5) Legacy or Unsupported Formats (.doc, etc.)
When an input ends with an extension of .doc or not natively supported extension they are classified under this category. In this format, there is an additional step called the conversion step. In this step the document is passed through LibreOffice headless mode. It is used for converting a document under the unknown extension and then converted into a pdf extension [25]. Further, the extraction method includes putting it under PyMuPDF & PDFMiner or OCR.

For the image upload module, it accepts common formats like .jpg, .png, .bmp, etc. The extraction method used includes OCR Tesseract to extract the data from the image.

2.2.2 Processing Layer

Once inputs are collected, they are routed to the processing layer, which acts as the core transformation engine. It mainly consists of 4 components which are described as follows:

1) Text Cleaning and Normalization:
Pre-processing is a requirement before an output can be created. It is an essential part of any NLP project. Pre-processing maintains the text in a clean and normalized form so that NLP algorithms and models are not required to spend processing capacity on unnecessary information. The function of pre-processing is to advance the quality of data and make it easier for NLP algorithms to process [26]. Pre-processing in this context means removing unwanted characters such as stray marks, special formatting codes, or OCR-generated artifacts, otherwise interfering with the reading efficiency. Pre-processing also standardizes spacing, punctuation, and capitalization so that the text is of the same construction for any source of input. For example, inconsistent spacing, irregular new lines, or varying encodings are normalized for clarity and consistency. With dyslexia, this is particularly important as extraneous visual clutter accumulates cognitive load and makes it harder to read [27, 28]. By normalizing and tidying up the text prior to simplification or formatting, the system gives a clean foundation upon which other processing stages operate effectively.

2) Text Simplification:
The stage of Text Simplification aims to make the language simpler in order to be more easily understood by dyslexic people. Compound and long sentences are reduced and simplified, and difficult words are substituted with simpler ones without changing the general idea of the text. For example, "utilize" is substituted with "use", and "commence" is substituted with "start". Technical terms are also written using simple terms for better understanding. The paragraph organization is done for easier consumption through fragmenting information into bite-sized, concise segments. This process is especially helpful for dyslexic readers, who often find it hard to read thick or advanced texts [29]. By fragmenting sentences and word usage, the system minimizes cognitive effort and simplifies the processing of information, resulting in content that is readily accessible and comprehensible.

3) Accessibility Formatting:
Accessibility Formatting guarantees that the output text is formatted to minimize visual stress and maximize readability for dyslexic users. Here, the information is reformatted in dyslexia-proof fonts like OpenDyslexic or Arial that are specifically aimed at increasing character recognition and avoiding letter reversal. Adjustable font size, line spacing, and letter spacing controls are included to support different user requirements, as studies indicate that more spacing can increase reading speed and accuracy for dyslexic readers considerably [30]. Contrast and background and text colour options are also included so that users can choose combinations—like dark text on a pale background—that reduce glare and visual stress. For instance, a user with difficulty reading closely crowded text can use increased line spacing and a less intense background in order to minimize crowding effects. Furthermore, the system enforces uniform paragraph organization and divides longer passages into block-shaped sections, allowing for simpler tracking and following of text. Together, these page layout modifications turn raw output into an interactive, reader-friendly version that supports extended reading and understanding.

4) Multimodal Augmentation:
Multimodal Augmentation is a method that boosts generative models by incorporating diverse data like text, image, sound and video into the learning and generation process [31]. In this step, the formatted and simplified text is generated to spoken audio with the help of a Text-to-Speech (TTS) engine, allowing users to hear the material while reading. To further increase understanding, the system features synchronized highlighting, which identifies spoken words visually in real

time, directing the user's attention and minimizing the loss of focus [32]. Play controls can also be customized by the user, including varying speech speed, pausing, or replaying individual segments, to have the system adjust to personal reading speeds and habits. For instance, a learner who is having difficulty decoding a paragraph can opt for the hybrid mode, whereby the spoken narration enhances the text, facilitating retention through auditory and visual channels. Such multimodal design not only decreases cognitive load but also facilitates inclusive learning by being capable of catering to varying literacy needs, rendering information more accessible and engaging.

2.2.3 Presentation Layer

Presentation level of dyslexia-friendly application exists, showing the text at reduced cognitive load. The text is shown after it has been blown out, zoomed out, and normalized in a high-feature interface that can be resized. They can select dyslexia-friendly fonts like Open Dyslexic or Arial, font size, line height, and letter spacing, and high-contrast schemes or background coloration in an effort to satisfy reading requirements and reduce eye discomfort. Text is divided into readable paragraphs and sentences, and real-time highlighting of the reading position. It incorporates synchronized Text-To-Speech support with audio tone enablement, word highlighting functionality, with progress bar navigation and read pointer. Visual, audio, and manual three-control integration gives the Presentation Layer human and dyslexic friendliness.

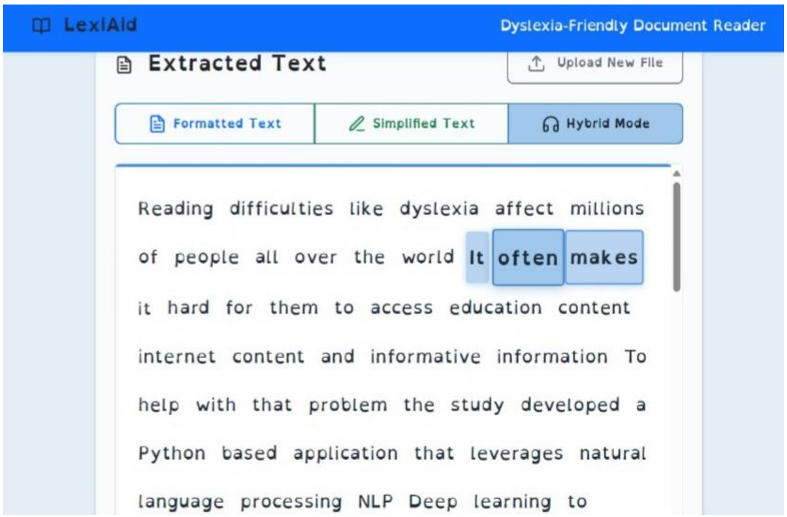


Figure 3: Text extraction and readability enhancement interface in Lexi Aid.

(**Figure 3. Extracted text view and adaptive reading modes in Lexi Aid.** The figure displays the processed text output generated after document extraction within the Lexi Aid platform. It demonstrates three viewing options—Formatted Text, Simplified Text, and Hybrid Mode—that adjust presentation and readability. Highlighted segments indicate the system's interactive text-to-speech feature, supporting personalized reading assistance for individuals with dyslexia.)

Cross-device offline rendering, adaptive machine-learning-format personalization, and learnability interoperability may be added in future releases. Incremental, incremental improvement will be acquired through user testing, app test-through to big-system literacy and accessibility solution.

3. Results

The system was performance-tested with three main input sources: file upload, direct text input, and photo scanning (OCR-based). 30 inputs were tested, 10 samples per input type. File inputs were a combination of PDF, DOCX, and ODT types; direct text inputs were plain text samples of different lengths; and photo scanning inputs were images of printed or typed texts. The test concentrated on two performance measures: average response time (time to render and extract results) and average output file size for generated audio narration.

Table 1. Average System Performance Across Input Types

Input Method	Avg. Response Time (s)	Avg. Audio File Size (MB/page)
File Upload	2	0.82
Direct Text Input	1.3	0.78
Photo Scanning	3.6	0.91

(**Table 1**. The table presents the average system performance metrics for different input methods, comparing their response times and corresponding audio file sizes per page. The results highlight that direct text input yields the fastest processing time, while photo scanning requires the most time and storage resources.)

Tests showed that the most direct text input was the quickest at an average response time of 1.3 seconds because there was no overhead of extraction. File uploads took a bit longer (2.0 seconds on average), which accounted for format parsing and normalization procedures, while the photo scanning inputs took the longest at 3.6 seconds, which is in line with OCR-based text recognition methods. As for audio output in terms of file size, outputs stayed lightweight for all input sizes, averaging between 0.78 MB and 0.91 MB per page, making them appropriate for storage as well as online listening.

A small usability study was conducted with five participants (three diagnosed with dyslexia and two with self-reported reading difficulties). Participants used Lexi Aid to read short passages and then rated their comprehension and ease of reading compared to the original text.

1) 4 out of 5 participants (80 %) reported improved comprehension.
2) All 5 participants (100 %) indicated that the text-to-speech and formatting features increased reading comfort.
3) 3 out of 5 participants (60 %) preferred the simplified text mode over the standard format for studying longer materials.

Participants also provided qualitative feedback, noting that the simplified vocabulary and read-aloud feature "made reading less tiring" and "helped maintain focus."

4. Discussion

This paper describes how to develop and prove a Python platform that integrates with OCR, NLP, and Text-to-Speech technology. This platform is for users with dyslexia, which is capable of automatically receiving written information both for reading and audio purposes. Users are allowed to submit content for free, which includes uploading files, scanning images, or copying content directly.

This investigation hypothesized that adaptive presentation of text using dyslexia-friendly fonts had the potential to decrease the load on working memory and increase reading comprehension for dyslexia sufferers. To confirm this hypothesis, a pilot study was carried out on a sample of five school-level students, all between 15 and 18 years old. Four of the five subjects claimed reading improvements, showed greater ease, and were shown to focus more on the adaptive features of the platform, such as the Open Dyslexic font, greater inter-line distance, and high-contrast background colouring. These findings are preliminary evidence that changes to visual setting parameters and the application of support strategies might hold some potential for enhanced reading comprehension and user experience for readers suffering from dyslexia. This evidence, however, remains scanty, given the small sample number, alongside the narrow range of ages, thus invalidating the findings from being generalizable across the whole population. User testing needs to be expanded, with a wide range of ages, reading capabilities, and language command, for a better assessment of these findings to be carried out.

The platform here being introduced is with a caveat: performance is directly based on how the input is provided. Pasted text is effective; however, handwritten or scanned input relies on optical character recognition (OCR), which tends to misinterpret letters, especially when using poor scans, exotic writing, or sloppy handwriting. For instance, letters such as "r" and "v" or "n" and "m" are notoriously confusing, therefore impeding retrieval processes. Distant or unusual fonts may cause lost or scrambled text. Repeatedly, text simplification may alter meaning. For instance, the sentence "cardiac arrhythmia describes irregular electrical impulses in the heart" may be simplified to "uneven heartbeats," or "quantum entanglement describes the correlated states of particles independent of distance" may be simplified to "particles stay connected." These illustrate the necessity for improved OCR processing as much as sophisticated simplification processes, both for retaining accuracy and for preserving meaning. Its future intentions for such enlargements are:

1) Multilingual capabilities – Presently, the program exists for English language alone. Simultaneous availability of Spanish, Hindi, or Mandarin language might be beneficial for a larger demographic, for dyslexia transcends language boundaries.
2) Personalization – Since dyslexia is individual for anyone, a subsequent update could enable users to customize text level, size, inter-letters, as well as speaking pace for a more personalized experience.
3) Better input processing involves maintaining equivalent performance on text, documents, and pictures while simultaneously minimizing losses of information due to reduction.
4) Very large-scale real-world testing – As a complement to technical validation, large-scale trials on a wide range of user groups would produce more representative data and allow performance assessment under real-world conditions.

This work presents evidence, using the implementation of OCR, NLP, and TTS technologies, of a possible way of creating reading software less challenging for people with dyslexia to use. Although some restrictions, such as reliance on English text, small sample size, and lack of large-scale trials, apply, the platform is a good starting point for further enhancement. Most importantly, this work is a first attempt at exploring ways of combining dyslexia-friendly typefaces, variations of text display, and customizability of readability parameters with a view to increasing readability and accessibility for people with dyslexia.

5. Conclusion

This research investigated the creation of a dyslexia-friendly assistive platform based on Python that embeds OCR, NLP-based text simplification, and text-to-speech capabilities to enhance accessibility for the reading impaired. The system was engineered to mitigate obstacles in processing hard-to-read written content through converting it into more readable and multi-modal forms.

While this review touched on the general application of AI-driven learning tools, our major interest was still on the improvement of text accessibility for dyslexic readers. English language-based texts were used in the study, and it was not applied to multilingual or non-English datasets. Our assessment was also limited to edited documents instead of actual user submissions, which can influence the results' generalizability. The present incarnation also does not take into consideration the heterogeneity of user requirements, for example, different literacy levels, cognitive styles, or accessibility for the visually challenged and differently abled groups. Significantly, compliance with such accessibility guidelines like Web Content Accessibility Guidelines (WCAG) was not investigated, nor how feedback from the users affects long-term learning outcomes.

In spite of these constraints, this study highlights the promise of the integration of OCR, NLP, and TTS in enabling more inclusive digital learning platforms. With further advancement, the follow-up work can aim at further broadening language support, adding personalization aspects that learn and respond to the specific needs of individual users, and strict adherence to accessibility standards. Overcoming these issues will bring the system nearer to being an easier and more inclusive assistive aid for people with dyslexia.

Acknowledgments (Optional)
The author gratefully acknowledges Big Red Education for the guidance and support provided through its research bootcamp program. The structured mentorship, academic resources, and constructive feedback offered during the program were instrumental in the completion of this paper.

References

[1] Cleveland Clinic. (2023, April 11). *Dyslexia: What it is, causes, symptoms, treatment & types.* Cleveland Clinic. https://my.clevelandclinic.org/health/diseases/6005-dyslexia
[2] Wisner, W. (2024, October 11). *Dyslexia: Definition, symptoms, causes, treatment.* Verywell Mind. https://www.verywellmind.com/dyslexia-definition-symptoms-causes-treatment-5525543
[3] Elias, M. (2025, March 26). *21 dyslexia statistics & facts: How common is it?* Discovery ABA. https://www.discoveryaba.com/statistics/dyslexia
[4] Yellow Busaba. (2025, October 1). *Dyslexia statistics & facts.* YellowBusABA. https://www.yellowbusaba.com/post/dyslexia-statistics-facts
[5] Witts, C. (2023, October 4). *Dyslexia and reading comprehension: Exploring challenges and strategies.* Forbrain. https://www.forbrain.com/dyslexia-children/reading-comprehension/
[6] Sunil, A. B., et al. (2023). Dyslexia: An invisible disability or different ability. *Indian Psychiatry Journal, 32*(Suppl 1), S72–S75. https://doi.org/10.4103/ipj.ipj_196_23
[7] Ambitions ABA. (2025, February 25). *73 dyslexia statistics and facts: How many people have dyslexia?* Ambitions ABA. https://www.ambitionsaba.com/resources/dyslexia-statistics
[8] Broggi, M., Ready, R. E., & Moore, D. L. (2019). Screening for reading disability in university students with phonological processing and working memory tasks. *Dyslexia, 25*(3), 256–266. https://doi.org/10.1002/dys.1625
[9] Kutest Kids. (2025, January 10). *Dyslexia statistics & facts.* Kutest Kids. https://www.kutestkids.com/blog/dyslexia-statistics-facts
[10] Wery, J. J., & Diliberto, J. A. (2017). The effect of a specialized dyslexia font, OpenDyslexic, on reading rate and accuracy. *Annals of Dyslexia, 67*(2), 114–127. https://doi.org/10.1007/s11881-016-0127-1
[11] Glassman, S. (2021, October 16). *The best assistive technology for dyslexics.* WIRED. https://www.wired.com/story/the-best-assistive-technology-dyslexia/
[12] Javourey-Drevet, L., et al. (2022). Simplification of literary and scientific texts to improve reading fluency and comprehension in beginning readers of French. *Applied Psycholinguistics, 43*(2), 485–512. https://doi.org/10.1017/S014271642100062X
[13] Murphy, S. (2020). *Dyslexia in the writing center: Multimodal strategies. Peer Review, 4.* https://thepeerreview-iwca.org/issues/issue-4-0/dyslexia-in-the-writing-center-multimodal-strategies/
[14] International Dyslexia Association. (n.d.). *Dyslexia basics.* https://dyslexiaida.org/dyslexia-basics/
[15] Franco, D. D. (2024, September 30). *Dyslexia in the workplace: Supporting employees with dyslexia.* Meditopia. https://meditopia.com/en/forwork/articles/dyslexia
[16] World Health Organization. (n.d.). *International Day of Persons with Disabilities.* https://www.who.int/campaigns/international-day-of-persons-with-disabilities
[17] HubSpot. (n.d.). *The HubSpot website blog: Tips, guides, and best practices for building websites.* HubSpot. https://blog.hubspot.com/website
[18] Smith, J. (2025). *PyMuPDF* (Version 1.26.4). PyPI. https://pypi.org/project/PyMuPDF/
[19] Shinyama, Y. (2025, September 15). *pdfminer.six* (Version 2025.9.15). PyPI. https://pypi.org/project/pdfminer.six/
[20] Bispo, N. (2024, September 27). *A guide to optical character recognition (OCR) with Tesseract.* Unstract. https://unstract.com/blog/guide-to-optical-character-recognition-with-tesseract-ocr/
[21] Iron Software. (2025, August 4). *Tesseract vs Microsoft OCR: OCR features comparison.* Iron Software. https://ironsoftware.com/csharp/ocr/blog/compare-to-other-components/tesseract-vs-microsoft-ocr-comparison/
[22] Folio3. (2024, September 27). *Tesseract OCR: Understanding its features, applications, and limitations.* Folio3. https://www.folio3.ai/blog/tesseract-ocr/
[23] Python Software Foundation. (2025). *python-docx* (Version 1.2.0). https://python-docx.readthedocs.io/en/latest/
[24] Roug, S. (2020). *odfpy* (Version 1.4.1). PyPI. https://pypi.org/project/odfpy/
[25] SciVision. (2023, October 5). *scivision/office-headless.* GitHub. https://github.com/scivision/office-headless
[26] Nomidl. (2023, February 16). *Pre-processing text data: Cleaning and normalization.* Nomidl. https://www.nomidl.com/natural-language-processing/pre-processing-text-data-cleaning-and-normalization/
[27] Yeatman, J. D. (2016, February). *What is the role of the visual system in reading and dyslexia?* International Dyslexia Association.

https://dyslexiaida.org/what-is-the-role-of-the-visual-system-in-reading-and-dyslexia/

[28] Bertoni, S., et al. (2019). Is excessive visual crowding causally linked to developmental dyslexia? *Neuropsychologia, 130,* 107–117. https://doi.org/10.1016/j.neuropsychologia.2019.04.018

[29] International Dyslexia Association. (n.d.). *Structured literacy: Effective instruction for students with dyslexia and related reading difficulties.* https://dyslexiaida.org/structured-literacy-effective-instruction-for-students-with-dyslexia-and-related-reading-difficulties/

[30] Zorzi, M., et al. (2012). Extra-large letter spacing improves reading in dyslexia. *Proceedings of the National Academy of Sciences, 109*(28), 11455–11459. https://doi.org/10.1073/pnas.1205566109

[31] GeeksforGeeks. (2025, July 23). *Multimodal retrieval augmented generation (multimodal RAG).* GeeksforGeeks. https://www.geeksforgeeks.org/artificial-intelligence/multimodal-retrieval-augmented-generation-multimodal-rag/

[32] Keelor, J. L., et al. (2020). Text-to-speech technology: Enhancing reading comprehension for students with reading difficulty. *Assistive Technology Outcomes and Benefits, 14*(2), 19–35. https://www.atia.org/wp-content/uploads/2020/06/ATOB-V14-A2-Keelor_etal.pdf