A Customizable AI-Powered Automatic Text Simplification Tool for Supporting In-Situ Text Comprehension

Nazmun Nahar Khanom*
Computer Science
Tulane University
New Orleans, LA, Louisiana, USA
nkhanom@tulane.edu

Aaron Gershkovich*
Tulane University
New Orleans, Louisiana, USA
Agershkovich@tulane.edu

Oliver Alonzo
DePaul University
Chicago, Illinois, USA
oliver.alonzo@depaul.edu

Saad Hassan
Department of Computer Science
Tulane University
New Orleans, Louisiana, USA
saadhassan@tulane.edu

Abstract

People with disabilities represent linguistically diverse communities. For example, among Deaf and Hard of Hearing (DHH) people, many of whom use sign language as their primary language, there is significant variation in written language literacy, highlighting that some might benefit from reading comprehension support tools. Prior research has demonstrated the benefits of lexical and syntactic approaches to Automatic Text Simplification for DHH readers and explored design considerations. Building on this work, we present a fully automatic, GPT-based text comprehension tool that provides in-situ reading support. The tool, released with this demo paper, is easily customizable and adaptable to support a range of disability communities and literacy levels. We present usage scenarios to spark conversations around broader applicability, personalization needs, and future studies comparing in-situ reading support to chatbot-style GPT interfaces.

CCS Concepts

• **Human-centered computing** → *Accessibility technologies*; **Empirical studies in accessibility**; **Accessibility systems and tools**.

Keywords

Automatic Text Simplification, Reading Assistance, Chatbots, AI, AI Reading Tools, GPT, Deaf and Hard-of-hearing, DHH, Reading

ACM Reference Format:
Nazmun Nahar Khanom, Aaron Gershkovich, Oliver Alonzo, and Saad Hassan. 2025. A Customizable AI-Powered Automatic Text Simplification Tool for Supporting In-Situ Text Comprehension. In *The 27th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '25), October 26–29, 2025, Denver, CO, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3663547.3759713

*Both authors contributed equally to this research.

ASSETS '25, Denver, CO, USA

ACM ISBN 979-8-4007-0676-9/25/10
https://doi.org/10.1145/3663547.3759713

1 Introduction and Related Work

Automatic Text Simplification (ATS) encompasses a range of natural language processing (NLP) techniques aimed at reducing the linguistic complexity of text while preserving its meaning [2]. Over the past two decades, ATS has shown promise in supporting a wide range of reader populations, including people with dyslexia [30], aphasia [24], and autism [39], as well as Deaf and Hard of Hearing (DHH) people [4, 20], second-language learners [28], and children [29, 35]. Across these groups, unique challenges with decoding complex vocabulary and sentence structures can impact comprehension and engagement with written content in educational, professional, and everyday contexts [5, 13].

For DHH readers—many of whom use sign language as their primary language—there is significant variation in English literacy skills. In the U.S., over 17% of DHH adults have low literacy, with some studies reporting that some high school graduates may read at a fourth-grade level [36]. Prior research has demonstrated the benefits of both `lexical` (word-level) [9] and `syntactic` (sentence-level) [11] simplification strategies, often leading to improved comprehension and reading confidence for DHH readers. However, these studies have typically relied on Wizard-of-Oz [10] or semi-automatic research prototypes [7], with limited exploration of fully automated, real-time systems that adapt to users' specific needs and contexts.

Research on accessible technology design emphasizes the importance of user control, visual presentation, and interaction modality [14, 19]. In web-based reading support technologies, features such as pop-up word explanations, in-place sentence rewrites, and user-invoked simplifications ensure that the reading support technology is useful without being intrusive. One recent work by Alonzo et al. [7] introduced a semi-automatic prototype[1] powered by lexical and syntactic models fine-tuned on specialized corpora such as the BiSECT dataset [22]. With the advent of powerful and widely

[1]Source code for the original automatic text simplification prototype: https://github.com/oliveralonzo/ats-prototype

available large language models (LLMs)—such as OpenAI's Generative Pre-trained Transformers (GPT) [1]—fully automatic and easily integrated reading support tools are now feasible.

In this demonstration paper, we present and release a fully automatic, AI-powered text comprehension tool that offers customizable, in-situ reading support. Our system builds on the design of the previously released research prototype [7] by integrating GPT-4 to deliver real-time simplification support. We release the codebase as supplementary material with this submission and can be accessed on GitHub: https://github.com/TITHI-KHAN/ATS-GPT.

To demonstrate the customizability and extensibility of this automatic tool, we introduce a feature that tailors simplification output based on user-specified literacy levels, and we document how this customization can be implemented in the codebase. We conclude with example use cases and a discussion of future research directions, such as expanding support to additional communities and comparing in-situ interfaces with chatbot-style alternatives.

2 Automatic Text Simplification Tool

We built on top of the prototype developed by Alonzo et al. [7], a semi-automatic, customizable reading assistance tool designed as a research instrument to explore the design space of reading support tools through structured usability testing with DHH participants. Implemented as a web browser extension using HTML, CSS, and JavaScript, the tool identifies sentences and makes API calls to a local server to retrieve simplified versions from a pre-populated database. It also highlights complex words and sentences, with available replacements, to enable interactive simplification. Users can request `lexical` (word-level), `syntactic` (sentence-level), or `hybrid` (a combination of both) simplifications. The tool was designed to explore the design space of the tool. It thus allows users to select options for various design parameters including the amount of text to modify at a time, whether the complex and simplified texts are highlighted, and the location and duration of simplifications. More details about the design and implementation can be found in the original paper [7].

Since the original prototype was designed as a research instrument for controlled studies, it made sense to pre-generate complex word identifications and sentence simplifications. In our system, we replaced the API calls to a local server with OpenAI's GPT API [25]. We chose the GPT API due to its high adaptability, which allows for straightforward customization through prompt design or even the integration of additional data within the prompt. At the time of implementation, we used GPT-4, though this can easily be modified, e.g., via the "Playground" section of the API platform.

Some research shows that LLMs like GPT-3.5 and GPT-4 can produce significantly better simplification outcomes when guided by carefully crafted prompts [16, 21]. For example, including explicit instructions (e.g., "*replace complex words with simpler synonyms*," "*split long sentences into shorter ones*") and structured example pairs (e.g., "*Complex: ... Simple: ...*) improves meaning preservation and fluency [12, 21]. Kew et al. [21] found that detailed prompts yielded higher simplicity and preservation scores than generic ones. With well-designed instructions and few-shot examples, GPT-based systems have even matched or outperformed specialized ATS models on standard benchmarks [12, 37]. This literature informed the construction of our prompts. However, some studies also note that even advanced models like GPT-4 occasionally struggle with lexical paraphrasing—failing to simplify specific terms or jargon—which underscores the need for more contextual information or additional finetuning [38]. We leave the experimentation with different prompt performances to future work.

Upon activating the prototype, text from the current web interface is sent to a Flask-based API connected to GPT. We chose to send the entire text and use a single prompt to retrieve all three forms of simplifications supported by our system, as the latency of individual API calls would not justify making separate requests for each user selection or word/sentence to simplify. A single prompt is issued for the entire page to perform four tasks: (1) identify complex words and generate synonyms, (2) generate a `lexical` simplification (word-level replacements), (3) produce a `syntactic` simplification (sentence restructuring), and (4) create a `hybrid` simplification that combines both strategies. The system returns a structured JSON output with four keys: *lexical*, *syntactic*, *hybrid*, and *words* (a mapping of complex to simpler terms). Basic error handling is included. Figure 1 illustrates the system workflow for browser-based automatic text simplification. A sample prompt and output format within a code snippet are also provided in Appendix A. Both the prompts and output structure are easily customizable.

3 How Can Researchers Make Use of this Tool?

3.1 Adding New Customization Features

The customization of the tool enabled exploration of design space of the prototype was explored with DHH adults using a structured usability-testing method [7]. Even after integration with the GPT API, the tool remains highly customizable. Thanks to this integration, however, researchers may now easily add new features or modify existing ones within the codebase. To demonstrate this, we added a feature that allows users to adapt the output to different literacy levels. Users can select a grade level from `Elementary`, `High School`, or `College`, based on their reading ability. This functionality was implemented by creating three prompt templates tailored to each literacy level for identifying complex words and generating simplifications. The underlying prompts included additional textual cues based on the best practices listed in section 2, which can be modified. For example, the `Elementary` version uses simplified vocabulary and appends clarifying phrases, such as "*short, easy to read sentences*" and "*words an elementary student would know*" to the prompts. The `High School` version balances clarity with light academic phrasing, appending short phrases to the prompts like "*familiar to a high school reader*" and "*conversational language.*" The `College` version allows some complex words to remain in order to maintain "*semantic integrity*" and "*domain-specific vocabulary.*" Figure 2 illustrates the automatic text simplification tool with all the parameter settings (including this newly added feature).

Since the underlying model does not rely on a specific training corpus, researchers can experiment with different G PT prompt strategies or allow users to define their own. In addition, they could incorporate corpora collected from specific communities to further tailor the prompts and outputs, such as the recent dataset on word complexity from DHH annotators [8].

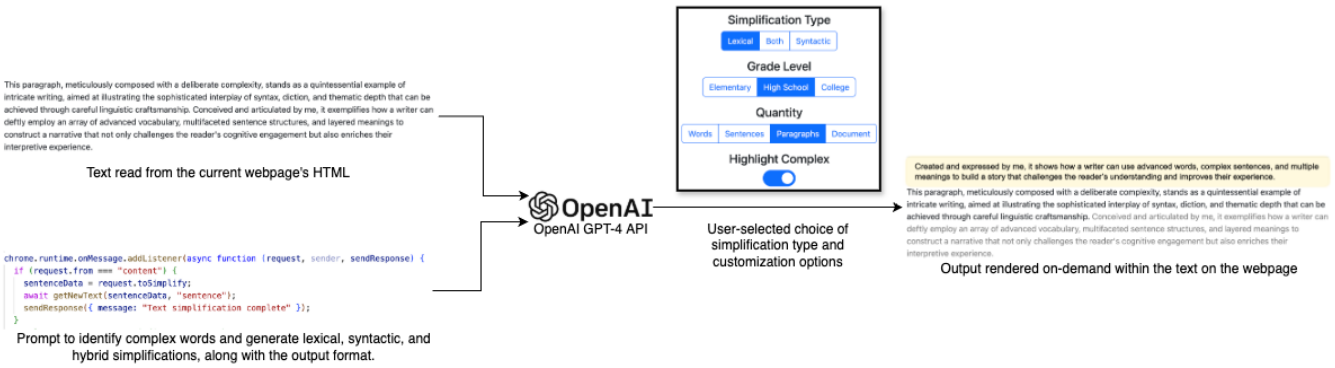

Figure 1: Schematic illustrating the system design with its components. Text is read from the webpage and sent along with simplification prompts to the GPT-4 API. The output is stored in a JSON file and rendered on-demand based on the format requested by the user.

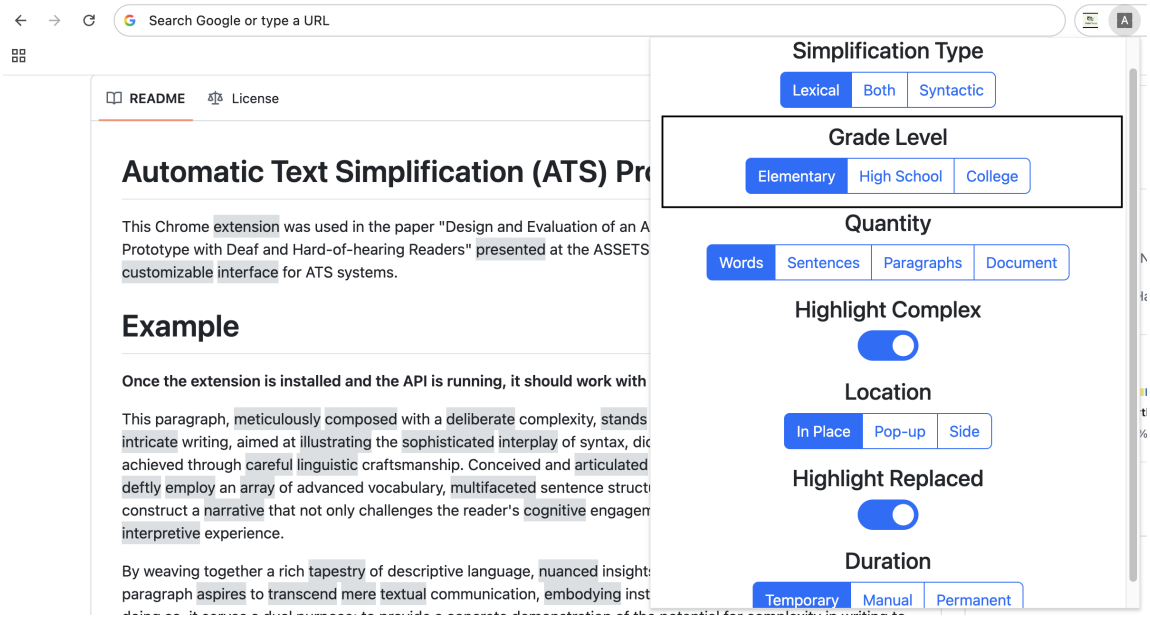

Figure 2: Interface of the web extension with the new customization feature to modify prompts used for identifying complex words and generating simplifications based on different grade levels.

3.2 Research with Other Communities

While the text simplification tool was designed with DHH readers in mind [6], this demonstration introduces a prompt-based approach with GPT-4 integration that offers a platform for the researchers to explore new customization strategies, compare interface paradigms, and adapt the tool for other communities. For example, people with dyslexia may benefit from sentence-level rewrites that reduce cognitive load [30], while second-language learners might prefer inline definitions or vocabulary scaffolding [27]. Readers with aphasia or autism may also benefit from syntactic simplifications that minimize ambiguity and cognitive demand [15, 23]. Because the system allows users or researchers to tailor prompt strategies and simplification levels, it can be adapted to support different populations and opens opportunities for participatory customization.

A slightly more time-consuming modification would be adjusting the typographic parameters and visual parameters, e.g., (color and line spacing), which are known to improve readability for different communities [18, 31–33]. We hope that our demo will spark conversations among researchers working with different communities regarding customization.

Our modular codebase separates user settings, prompt construction, and interface rendering into distinct components. Simplification is driven by prompts that are generated based on user-selected parameters and sent through an API call. This architecture could potentially enable the researchers to experiment with the tool as a reference or guideline for different target audiences such as non-native speakers [26] or people with cognitive disabilities [34].

3.3 In-situ vs. Chatbot-style Interfaces

Our system can also be used to investigate the distinct benefits and drawbacks of in-situ versus chatbot-style interfaces (popularized by leading AI chatbots, e.g., ChatGPT, Gemini) for reading support [1]. In-situ reading support embeds assistance directly into the reading environment and offers benefits such as highlighting complex words, providing on-demand synonyms or rephrasings, and maintaining the user's reading flow [3, 4, 7, 17]. In some contexts, it may be preferred because it preserves context, supports autonomy, and allows more granular, on-demand ATS [9, 10]. However, in-situ designs must avoid visual clutter and ensure high accuracy, as errors are immediately visible and can undermine trust [11, 27]. In-situ changes may also add reading time when the interface does not support efficient comparison between the original and simplified texts [7].

In contrast, chatbot-style interfaces offer personalized, conversational assistance that can adapt explanations over multiple "conversational turns" and provide deeper clarification, which can be especially valuable for users engaging with technical or unfamiliar material [1, 2]. Yet, they may disrupt reading flow, require more interactions for comparison, and require users to formulate queries, which can be an added barrier for those with low English literacy. They may also introduce risks of AI inaccuracy or hallucination [1, 22].

By evaluating how people with disabilities interact with each design paradigm, researchers can help illuminate how design choices impact readability, comprehension, user control, and trust in AI-driven reading support technologies.

4 Conclusion

We release a fully automatic GPT-based ATS system for in-situ reading support and demonstrate how it can be customized for users with different English literacy levels. We hope to spark conversations with researchers working with diverse reading disability groups and those exploring empirical questions, such as comparing in-situ and chatbot-style support.

Acknowledgments

This material is partially based on work supported by the Louisiana Board of Regents Research Competitiveness Subprogram (RCS) award (093A-24). The authors acknowledge the use of OpenAI's ChatGPT and Google's Gemini as editorial tools for refining the clarity of the text.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Suha S. Al-Thanyyan and Aqil M. Azmi. 2021. Automated Text Simplification: A Survey. *ACM Comput. Surv.* 54, 2, Article 43 (March 2021), 36 pages. doi:10.1145/3442695

[3] Oliver Alonzo. 2022. The use of automatic text simplification to provide reading assistance to deaf and hard-of-hearing individuals in computing fields. *SIGACCESS Access. Comput.* 132, Article 3 (March 2022), 1 pages. doi:10.1145/3523265.3523268

[4] Oliver Alonzo, Lisa Elliot, Becca Dingman, and Matt Huenerfauth. 2020. Reading experiences and interest in reading-assistance tools among deaf and hard-of-hearing computing professionals. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, New York, NY, USA.

[5] Oliver Alonzo and Saad Hassan. 2025. A Review of 25 Years of Human-Computer Interaction Research on Reading Support Technologies for People with Disabilities Published in the ACM Digital Library. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility* (Denver, CO, USA) *(ASSETS '25)*. Association for Computing Machinery, New York, NY, USA, 21 pages. doi:10.1145/3663547.3746355

[6] Oliver Alonzo, Sooyeon Lee, Akhter Al Amin, Mounica Maddela, Wei Xu, and Matt Huenerfauth. 2024. Design and Evaluation of an Automatic Text Simplification Prototype with Deaf and Hard-of-hearing Readers. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility* (St. John's, NL, Canada) *(ASSETS '24)*. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3663548.3675645

[7] Oliver Alonzo, Sooyeon Lee, Akhter Al Amin, Mounica Maddela, Wei Xu, and Matt Huenerfauth. 2024. Design and evaluation of an automatic text simplification prototype with deaf and hard-of-hearing readers. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, New York, NY, USA, 1–18.

[8] Oliver Alonzo, Sooyeon Lee, Mounica Maddela, Wei Xu, and Matt Huenerfauth. 2022. A Dataset of Word-Complexity Judgements from Deaf and Hard-of-Hearing Adults for Text Simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, Sanja Štajner, Horacio Saggion, Daniel Ferrés, Matthew Shardlow, Kim Cheng Sheang, Kai North, Marcos Zampieri, and Wei Xu (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Virtual), 119–124. doi:10.18653/v1/2022.tsar-1.11

[9] Oliver Alonzo, Matthew Seita, Abraham Glasser, and Matt Huenerfauth. 2020. Automatic Text Simplification Tools for Deaf and Hard of Hearing Adults: Benefits of Lexical Simplification and Providing Users with Autonomy. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13.

[10] Oliver Alonzo, Jessica Trussell, Becca Dingman, and Matt Huenerfauth. 2021. Comparison of Methods for Evaluating Complexity of Simplified Texts among Deaf and Hard-of-Hearing Adults at Different Literacy Levels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA.

[11] Oliver Alonzo, Jessica Trussell, Matthew Watkins, Sooyeon Lee, and Matt Huenerfauth. 2022. Methods for Evaluating the Fluency of Automatically Simplified Texts with Deaf and Hard-of-Hearing Adults at Various Literacy Levels. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA.

[12] Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. ASSET: A Dataset for Tuning and Evaluation of Sentence Simplification Models with Multiple Rewriting Transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 4668–4679. doi:10.18653/v1/2020.acl-main.424

[13] Alex Atcheson, Omar Khan, Brian Siemann, Anika Jain, and Karrie Karahalios. 2025. "I'd Never Actually Realized How Big An Impact It Had Until Now": Perspectives of University Students with Disabilities on Generative Artificial Intelligence. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 42, 22 pages. doi:10.1145/3706598.3714121

[14] Jeffrey P. Bigham, Richard E. Ladner, and Yevgen Borodin. 2011. The design of human-powered access technology. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility* (Dundee, Scotland, UK) *(ASSETS '11)*. Association for Computing Machinery, New York, NY, USA, 3–10. doi:10.1145/2049536.2049540

[15] Siobhan Devlin and Gary Unthank. 2006. Helping aphasic people process online information. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. ACM, New York, NY, USA.

[16] Yutao Feng, Jipeng Qiang, Yun Li, Yunhao Yuan, and Yi Zhu. 2023. Sentence simplification via large language models. *arXiv preprint arXiv:2302.11957* (2023).

[17] Steven M. Goodman, Erin Buehler, Patrick Clary, Andy Coenen, Aaron Donsbach, Tiffanie N. Horne, Michal Lahav, Robert MacDonald, Rain Breaw Michaels, Ajit Narayanan, Mahima Pushkarna, Joel Riley, Alex Santana, Lei Shi, Rachel Sweeney, Phil Weaver, Ann Yuan, and Meredith Ringel Morris. 2022. LaMPost: Design and Evaluation of an AI-assisted Email Writing Prototype for Adults with Dyslexia. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility* (Athens, Greece) *(ASSETS '22)*. Association for Computing Machinery, New York, NY, USA, Article 24, 18 pages. doi:10.1145/3517428.3544819

[18] Peter Gregor and Alan F. Newell. 2000. An empirical investigation of ways in which some of the problems encountered by some dyslexics may be alleviated using computer techniques. In *Proceedings of the Fourth International ACM Conference on Assistive Technologies* (Arlington, Virginia, USA) *(Assets '00)*. Association for Computing Machinery, New York, NY, USA, 85–91. doi:10.1145/354324.354347

[19] Julia Himmelsbach, Markus Garschall, Sebastian Egger, Susanne Steffek, and Manfred Tscheligi. 2015. Enabling accessibility through multimodality? interaction modality choices of older adults. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia* (Linz, Austria) *(MUM '15)*. Association for Computing Machinery, New York, NY, USA, 195–199. doi:10.1145/2836041.2836060

[20] Shuxu Huffman, Si Chen, Kelly Avery Mack, Haotian Su, Qi Wang, and Raja Kushalnagar. 2025. "We do use it, but not how hearing people think": How the Deaf and Hard of Hearing Community Uses Large Language Model Tools. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 33, 9 pages. doi:10.1145/3706599.3719785

[21] Tannon Kew, Alison Chi, Laura Vásquez-Rodríguez, Sweta Agrawal, Dennis Aumiller, Fernando Alva-Manchego, and Matthew Shardlow. 2023. BLESS: Benchmarking Large Language Models on Sentence Simplification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 13291–13309. doi:10.18653/v1/2023.emnlp-main.821

[22] Joongwon Kim, Mounica Maddela, Reno Kriz, Wei Xu, and Chris Callison-Burch. 2021. BiSECT: Learning to split and rephrase sentences with bitexts. *arXiv preprint arXiv:2109.05006* (2021).

[23] Thomas Langford, Alex Leff, and Daniela Romano. 2021. IReadMore: A reading therapy app co-designed by people with aphasia and Alexia. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, New York, NY, USA.

[24] Xiaojuan Ma. 2010. An online multimedia language assistant for people with aphasia and other language barriers. *SIGACCESS Access. Comput.* 96 (Jan. 2010), 46–49. doi:10.1145/1731849.1731858

[25] OpenAI. 2025. OpenAI GPT API. https://platform.openai.com/docs/models

[26] Gustavo Henrique Paetzold. 2016. *Lexical simplification for non-native english speakers.* Ph. D. Dissertation. University of Sheffield.

[27] Gustavo H. Paetzold and Lucia Specia. 2017. A survey on lexical simplification. *J. Artif. Int. Res.* 60, 1 (Sept. 2017), 549–593.

[28] Sarah E Petersen and Mari Ostendorf. 2007. Text simplification for language learners: a corpus analysis.. In *SLaTE*. 69–72.

[29] Nancy Rasche, John Pourcho, Shuang Wei, Cheryl Zhenyu Qian, and Victor Yingjie Chen. 2013. Literacy LABELS: emergent literacy application design for children with autism. In *ACM SIGGRAPH 2013 Posters*. ACM, New York, NY, USA.

[30] Luz Rello. 2012. DysWebxia: a model to improve accessibility of the textual web for dyslexic users. *SIGACCESS Access. Comput.* 102 (Jan. 2012), 41–44. doi:10.1145/2140446.2140455

[31] Luz Rello and Ricardo Baeza-Yates. 2013. Good fonts for dyslexia. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility* (Bellevue, Washington) *(ASSETS '13)*. Association for Computing Machinery, New York, NY, USA, Article 14, 8 pages. doi:10.1145/2513383.2513447

[32] Luz Rello and Ricardo Baeza-Yates. 2016. The Effect of Font Type on Screen Readability by People with Dyslexia. *ACM Trans. Access. Comput.* 8, 4, Article 15 (May 2016), 33 pages. doi:10.1145/2897736

[33] Luz Rello and Jeffrey P. Bigham. 2017. Good Background Colors for Readers: A Study of People with and without Dyslexia. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility* (Baltimore, Maryland, USA) *(ASSETS '17)*. Association for Computing Machinery, New York, NY, USA, 72–80. doi:10.1145/3132525.3132546

[34] Sabina Sieghart, Björn Rohles, and Ann Bessemans. 2024. Empowering Independence Through Design: Investigating Standard Digital Design Patterns For Easy-to-Read Users.. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 911, 18 pages. doi:10.1145/3613904.3641911

[35] Anastasia Smirnova, Kyu beom Chun, Wil Louis Rothman, and Siyona Sarma. 2025. Text Simplification for Children: Evaluating LLMs vis-à-vis Human Experts. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 512, 10 pages. doi:10.1145/3706599.3719889

[36] Carol Bloomquist Traxler. 2000. The Stanford Achievement Test: National norming and performance standards for deaf and hard-of-hearing students. *Journal of deaf studies and deaf education* 5, 4 (2000), 337–348.

[37] Xuanxin Wu and Yuki Arase. 2024. An in-depth evaluation of gpt-4 in sentence simplification with error-based human assessment. *arXiv preprint arXiv:2403.04963* (2024).

[38] Xuanxin Wu and Yuki Arase. 2024. An In-depth Evaluation of GPT-4 in Sentence Simplification with Error-based Human Assessment. *arXiv preprint arXiv:2403.04963* (2024).

[39] Victoria Yaneva, Irina Temnikova, and Ruslan Mitkov. 2015. Accessible texts for autism: An eye-tracking study. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility - ASSETS '15*. ACM Press, New York, New York, USA.

A Code Snippet with Prompt

Listing 1: A code snippet showing one of the example prompts used in our codebase for text simplification using GPT and receiving the output.

```
def generate_simplifications(self, text):
    try:
        instruction = f"""
        For the given sentence: "{text}", perform the
             following tasks:
        1. Identify all complex or uncommon words and
             list simpler synonyms. Focus on words that a
              general reader or non-native speaker might
             find difficult, and ensure the suggested
             synonyms preserve the original meaning.

        2. Rewrite using simpler vocabulary. Replace
             complex or rare terms with more common
             synonyms while keeping the meaning unchanged
             . Simplify as many words as possible,
             ensuring the sentence remains grammatically
             correct and semantically accurate.

        3. Paraphrase in simpler, more straightforward
             language. Use clear, everyday wording
             without omitting important details. The
             meaning should remain exactly the same,
             expressed in accessible terms.

        4. Rewrite with simpler syntax. Break down long
             or complex sentence structures into shorter,
              clearer ones while preserving the original
             meaning.

        5. Simplify both vocabulary and sentence
             structure. Replace difficult words with
             easier synonyms and split up complex
             constructions if needed. Ensure the result
             is fluent, easy to read, and faithful to the
              original meaning.

        Format:
        "{text}": {
            "lexical": "...",
            "words": {
                "complex_word1": "simple_word1",
                "complex_word2": "simple_word2"
            },
            "syntactic": "...",
            "hybrid": "..."
        }
        """
        response = client.chat.completions.create(
            model="gpt-4",
            messages=[
                {"role": "system", "content": "Simplify␣
                     complex␣sentences."},
                {"role": "user", "content": instruction}
            ]
        )
        simplified_text = response.choices[0].message.
             content
        return self.clean_json_string(simplified_text)
    except Exception as e:
        print(f"Error␣while␣calling␣OpenAI␣API:␣{e}")
        return None
```