Chair for Computer Science 10 (Media Computing and Human-Computer Interaction)

RWTH AACHEN UNIVERSITY

MagicTextreader: Investigating an Interactive AI System for Text Accessibility

Master's Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Ernst Maximilian Münker

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Ulrik Schroeder

Registration date: 17.12.2024
Submission date: 17.06.2025

Zentrales Prüfungsamt/Central Examination Office

RWTH AACHEN UNIVERSITY

Eidesstattliche Versicherung
Declaration of Academic Integrity

Münker, Ernst Maximilian

Name, Vorname/Last Name, First Name

370833

Matrikelnummer (freiwillige Angabe)
Student ID Number (optional)

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/ Masterarbeit* mit dem Titel
I hereby declare under penalty of perjury that I have completed the present paper/bachelor's thesis/master's thesis* entitled

MagicTextreader: Investigating an Interactive AI System for Text Accessibility /

MagicTextreader: Untersuchung eines interaktiven KI-Systems für Textzugänglichkeit

selbstständig und ohne unzulässige fremde Hilfe (insbes. akademisches Ghostwriting) erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt; dies umfasst insbesondere auch Software und Dienste zur Sprach-, Text- und Medienproduktion. Ich erkläre, dass für den Fall, dass die Arbeit in unterschiedlichen Formen eingereicht wird (z.B. elektronisch, gedruckt, geplottet, auf einem Datenträger) alle eingereichten Versionen vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.
independently and without unauthorized assistance from third parties (in particular academic ghostwriting. I have not used any other sources or aids than those indicated; this includes in particular software and services for language, text, and media production. In the event that the work is submitted in different formats (e.g. electronically, printed, plotted, on a data carrier), I declare that all the submitted versions are fully identical. I have not previously submitted this work, either in the same or a similar form to an examination body.

Zülpich, den 17.06.25

Ort, Datum/City, Date

Unterschrift/Signature

*Nichtzutreffendes bitte streichen/Please delete as appropriate

Belehrung:
Official Notification:

§ 156 StGB: Falsche Versicherung an Eides Statt
Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.
§ 156 StGB (German Criminal Code): False Unsworn Declarations
Whosoever before a public authority competent to administer unsworn declarations (including Declarations of Academic Integrity) falsely submits such a declaration or falsely testifies while referring to such a declaration shall be liable to imprisonment for a term not exceeding three years or to a fine.
§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt
(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.
(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.
§ 161 StGB (German Criminal Code): False Unsworn Declarations Due to Negligence
(1) If an individual commits one of the offenses listed in §§ 154 to 156 due to negligence, they are liable to imprisonment for a term not exceeding one year or to a fine.
(2) The offender shall be exempt from liability if they correct their false testimony in time. The provisions of § 158 (2) and (3) shall apply accordingly.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:
I have read and understood the above official notification:

Zülpich, den 17.06.25

Ort, Datum/City, Date

Unterschrift/Signature

Contents

List of Figures and Tables

Abstract

The rise of digital reading has introduced new ways to engage with written content, complementing rather than replacing printed books. Digital platforms have improved technical accessibility through features like magnification and contrast, yet the text itself often remains a barrier. Many adults struggle with literacy, syntactic complexity, and dense concepts, as highlighted by OECD data. Accessibility is also influenced by other factors such as tone, style, or personalization. At the same time, large language models (LLMs) offer powerful capabilities to adapt text for diverse audiences. However, current e-reading tools do not support AI-based text transformation, and little research has explored how such functionality might impact digital text accessibility.

This thesis addresses this gap with *MagicTextreader*, an AI-enhanced e-reader that enables real-time text transformation. The system provides modular building blocks—three types defined by their control mechanisms: toggle, choice, and slider—through which various transformations can be defined, such as complexity, tone, style, or personalization adaptations. Users can also define custom transformations via natural language interaction. Additionally, the prototype is embedded within an accompanying platform that lays the foundation for a research toolkit to support further studies on AI-assisted reading.

A first evaluation of the prototype was conducted through a mixed methods study (N=11) using non-fiction content. Participants applied predefined and custom transformations, with data collected through survey, observation and qualitative user interviews. Findings indicate that transformations—especially for complexity, tone, and custom adaptations via natural language—enhanced perceived understanding, reading flow, and motivation. However, the system also introduced barriers, including confusion and concerns about information integrity due to limited transparency. The study yields design implications not only relevant for the prototype but also for other interactive systems that integrate LLMs, indicating that while AI-driven text transformation can enhance accessibility, it must be paired with transparency and trust mechanisms.

Überblick

Das Aufkommen digitalen Lesens hat zu neuen Wegen geführt, um mit Texten zu interagieren – als Ergänzung, nicht als Ersatz gedruckter Bücher. Digitale Plattformen verbessern die Zugänglichkeit durch Funktionen wie Vergrößerung oder Kontraste, doch der Text selbst bleibt oft eine Barriere – viele Erwachsene kämpfen laut OECD-Daten mit Literalität, komplexer Syntax und hoher Informationsdichte. Zugänglichkeit wird zudem von weiteren Faktoren wie Tonalität, Stil oder Personalisierung beeinflusst. Gleichzeitig bieten Large Language Models (LLMs) vielversprechende Möglichkeiten, Texte für verschiedene Zielgruppen anzupassen. Dennoch unterstützen bestehende E-Reading-Tools keine KI-gestützte Texttransformation, und es gibt wenig Forschung zu deren Auswirkungen auf die Textzugänglichkeit.

Diese Masterarbeit adressiert diese Lücke mit *MagicTextreader*, einem KI-unterstützten E-Reader, der Echtzeit-Texttransformation ermöglicht. Das System stellt modulare Bausteine bereit – drei Typen, definiert durch ihre Steuermechanismen: Toggle, Choice und Slider –, mit denen sich Transformationen wie die Anpassung von Komplexität, Tonalität, Stil, Format oder Personalisierung definieren lassen. Eigene Transformationen können zusätzlich über Interaktion in natürlicher Sprache erstellt werden. Der Prototyp ist Teil einer Plattform, die als Grundlage eines Research Toolkits für weiterführende Studien dienen kann.

Der Prototyp wurde erstmals in einer Studie mit Sachtexten und kombiniertem Methodenansatz (N=11) evaluiert, in der vordefinierte und eigene Transformationen genutzt werden konnten. Die Ergebnisse deuten an, dass insbesondere Anpassungen in Komplexität, Tonalität sowie individuelle das wahrgenommene Textverständnis, den Lesefluss und die Lesemotivation verbessern konnten. Gleichzeitig traten auch Barrieren auf, darunter Verwirrung sowie Bedenken hinsichtlich der Informationsintegrität aufgrund begrenzter Transparenz. Die Studie liefert Design-Implikationen auch für andere interaktive Systeme mit LLMs und deutet an, dass KI-gestützte Texttransformation Zugänglichkeit fördern kann – jedoch nur in Kombination mit Mechanismen für Transparenz und Vertrauen.

Acknowledgments

First, I would like to thank Prof. Dr. Jan Borchers and Prof. Dr. Ulrik Schroeder for examining my thesis.

Many thanks to the Media Computing Group for the opportunity to work on this thesis, and especially to Sarah Sahabi for the dedicated and motivating supervision, as well as the feedback which helped me a lot throughout the process.

A big thank you also to my family and friends—thank you for your support, advice, and encouragement.

Finally, I would like to thank all participants in the study for their time and effort. Your contributions made this thesis possible.

Conventions

Throughout this thesis we use the following conventions:

- The thesis is written in American English.
- The first person is used in the plural form.
- Unidentified third persons are referred to neutrally or in the plural form.

Short excursuses are set off in colored boxes.

EXCURSUS:
Excursuses are set off in orange boxes.

Where appropriate, paragraphs are summarized by one or two sentences that are positioned at the margin of the page.

This is a summary of a paragraph.

Citations after a period (e.g., ". [Ren, 2022]") indicate support for the entire paragraph, unless specific statements are attributed to other sources. In Chapter 2 "Background", paragraphs compiled from multiple sources are marked explicitly, e.g., *Based on:* [Ren, 2022].

Citation rules per paragraph

Chapter 1

Introduction

"The medium is the message"

—Marshall McLuhan

The digital revolution has fundamentally transformed how we consume written information [Pae, 2020]. This transformation traces back to pioneering efforts in the early 1970s [Lebert, 2009].

1.1 The Digital Reading Revolution

Since Michael Hart launched Project Gutenberg in 1971, creating the first digital book library, the evolution of digital reading has progressed from a niche technological experiment to a mainstream method of content consumption [Lebert, 2009]. This transformation accelerated dramatically with Amazon's introduction of the Kindle in 2007, which marked a turning point in mass adoption of digital reading [Ren, 2022].

Historical evolution of digital reading from 1971 to present

Today, digital reading has become an integral part of daily life, as in the United States alone, the percentage of people reading e-books has nearly doubled from 17% in 2011 to 30% in 2021 [Faverio and Perrin, 2022]. Similar trends are

Quantitative evidence of digital reading adoption visible in other markets, with Germany's e-book revenue share growing from 0.5% to 6.1% between 2010 and 2023 [Statista Research Department, 2022].

1.2 Digital Text Accessibility Advantages

Revolutionary impact on text distribution

The transition from print to digital text formats has introduced unprecedented opportunities for improving text accessibility and distribution [McNaught and Alexander, 2014]. While traditional printed books offer limited adaptability, digital platforms have revolutionized how readers can interact with and consume text content, fundamentally transforming the accessibility landscape [McNaught and Alexander, 2014].

Technical features enhancing accessibility

Based on McNaught and Alexander [2014], digital platforms provide crucial technical accessibility features that significantly improve reading experiences. They argue that, when properly formatted, e-books can offer extensive customization options, including text magnification with reflow capability that maintains readable line lengths and improves reading speed. Also color and contrast customization options benefit both visually impaired and dyslexic readers, as well as those reading in extreme lighting conditions.

1.3 The Accessibility Gap

OECD literacy statistics reveal comprehension challenges

According to Štajner [2021], despite these technological accessibility improvements, a significant gap persists in content accessibility. While digital formats address visual and physical barriers, the inherent properties of the text itself—including vocabulary, syntax, conceptual density, and information structure—often remain inaccessible. According to the OECD's Adult Literacy Report, approximately 16.7% of adults across 24 countries have literacy below Level 2, requiring lexical simplification, while about 50% struggle with syntactic complexity [Štajner, 2021]. Even more, around 89.4% of the population finds it difficult to comprehend conceptually dense texts [Štajner, 2021].

Scientific content accessibility challenges

This accessibility gap is particularly evident in scientific and technical content. Non-experts often avoid authoritative sources due to complex language and specialized terminology [Ermakova et al., 2021], instead relying on potentially less reliable sources that prioritize commercial or political interests over accuracy. Research shows this avoidance could be prevented by providing content in different styles—for instance, Michielutte et al. [1992] demonstrate that using narrative styles and visual aids significantly improves comprehension and engagement across different reading levels.

Personalization and individual differences influence accessibility or reading process

Beyond content complexity, even subtle linguistic personalization can impact comprehension and engagement. Dutke et al. [2016] demonstrate that simply changing generic references to second-person possessive pronouns (e.g., replacing "the eye" with "your eye" in an anatomy text) measurably improves reader transfer performance and engagement. In general, reader demographics and factors including motivation, background knowledge, education level or reading strategies affect the reading process [Fulcher, 1997].

1.4 The AI Revolution in Text Processing

Evolution of language models

As Wang et al. [2024] describe in their survey, the landscape of text processing has been dramatically transformed by the evolution of artificial intelligence, progressing from language models based on pure statistics through neural approaches to, finally, large language models (LLMs). According to their survey, these advanced systems demonstrate unprecedented capabilities in text comprehension and generation, fundamentally changing how we approach text transformation. The integration of deep learning has enabled systems to manage ambiguity, contextual meaning, and complex syntactic structures without relying on explicit rules [Johri et al., 2021].

Current capabilities in text adaptation

Modern language models exhibit adaptability in tasks requiring minimal supervision, as they demonstrate proficiency in personalized content generation, capable of tailoring outputs based on specific user contexts and preferences [Meguellati et al., 2024]. This capability extends to complex tasks such as transforming scientific content, where these systems can generate multiple variations of text adapted for diverse audiences [Kim et al., 2024]. These capabilities have also, for example, already been investigated for text simplification purposes, with a recent paper presented at CHI24 comparing machine-generated simplified summaries with those written by experts and concluding that they can perform comparably [August et al., 2024]. On the other hand, a known limitation of large language models is their tendency to generate hallucinations—outputs that are linguistically plausible but factually incorrect [Huang et al., 2025].

1.5 Research Gap and Opportunity

Research gap in integrated solutions

Despite the convergence of increasing e-reader adoption trends and advances in AI text transformation technologies, a systematic literature search utilizing keyword combinations such as "interactive, ebook, e-book, llm reader, ai, artificial intelligence, e-reader, text transformations, adaptive text systems" revealed a gap for integrated solutions that incorporate AI-based text transformation capabilities into e-readers.

Research opportunity

This might present an opportunity to investigate how integrated solutions embedding interactive text transformation capabilities into e-readers could impact text accessibility. The implementation of individual components in existing research - from AI-powered document augmentation [Lo et al., 2023] to specialized reading aids [Li et al., 2025] - suggests the technical feasibility of such investigation.

1.6 Proposed Solution

This thesis addresses the identified research gap through the development and investigation of an AI-enhanced text reader prototype and also contributes a foundation for an accompanying research toolkit for further investigation.

The prototype extends conventional e-reading functionality by incorporating AI-driven text transformation features that allow users to dynamically modify the presentation texts. To achieve this, the prototype provides modular building blocks—three types defined by their control mechanism: toggle, choice, and slider—through which various transformations can be defined such as complexity, tone, style or personalization, allowing the adjustment of the text across various dimensions. Users can also define custom transformations via natural language interaction.

Prototype enables user-controlled AI text transformation

Each of these dimensions represents what we define as a Text Accessibility Dimension:

TEXT ACCESSIBILITY DIMENSION:
A dimension is a characteristic of text that can influence how accessible it is to different readers. Each characteristic that leads to different versions of a text along an axis (including binary variations) is considered its own text accessibility dimension.

Definition: *Text Accessibility Dimension*

Examples include complexity (simple to advanced vocabulary) or also individual personalization, where each person has their own dimension describing how personalized the text is for that specific individual. This concept builds upon the definition of Text Accessibility in Section 2.1.

For a first evaluation of the prototype, a session-based mixed methods study explores how users interact with the AI-enhanced text reader during non-fiction reading. The study combines quantitative and qualitative approaches, in a setting that approximates natural reading conditions, in order to maintain an ethnographic orientation. The aim is to examine how the AI-enhanced text reader affects text ac-

First steps investigating prototype's impact on accessibility and usability cessibility, gather initial insights into user interaction patterns, and uncover preliminary design implications and future research directions. The contributions of this work include:

- A prototype implementation of an AI-enhanced text reader
- A foundation for an accompanying research toolkit to investigate AI-enhanced text readers
- Initial insights about the prototype's impact on text accessibility, design implications, and future research potential

1.7 Research Questions

The study addresses three primary research questions:

RQ1: In what ways does an AI-enhanced text reader affect text accessibility for users?

RQ2: What design implications emerge from user interactions with an AI-enhanced text reader?

RQ3: What research directions and application contexts show promise for AI-enhanced text readers?

1.8 Thesis Structure

The remainder of this thesis is structured as follows.

Chapter 2 "Background" introduces the concept of text accessibility, outlines key influencing factors, and presents the context of digital reading along with the EPUB format. Building on this, Chapter 3 "Related Work" reviews research on large language model-based systems for text adaptation and accessibility, as well as natural language interfaces, and discusses relevant work on text simplification, and data collection methods.

Chapter 4 "Design and Implementation of the MagicTextreader" describes the design and implementation of the MagicTextreader, following a structured process from system goals and requirements to key design decisions and the final implementation. Chapter 5 "Study and Results" presents a first mixed-methods study conducted to evaluate the prototype in order to address the research questions. It outlines the study design and reports findings in a descriptive manner based on questionnaires, interviews, observations, and interaction logs.

The findings are then discussed in Chapter 6 "Discussion", which relates them to the research questions using data triangulation and reflects on limitations. Finally, Chapter 7 "Summary and Outlook" summarizes the main contributions of this work and outlines directions for future research.

Chapter 2

Background

This chapter provides background information on text accessibility and digital reading. It begins with a definition of text accessibility—serving as a basis for assessing the impact of the MagicTextreader in the study—followed by an overview of various influencing factors. The subsequent sections describe the context of digital reading and introduce the EPUB format to support understanding of the content presented in Chapter 4 “Design and Implementation of the MagicTextreader”.

2.1 Text Accessibility

Accessibility has both everyday meanings (approachable, attainable, available) and specific technical applications across different contexts [Iwarsson and Ståhl, 2003]. In this thesis, text accessibility is understood as an umbrella term and defined as follows:

Text accessibility encompasses factors enabling or hindering text attainment

TEXT ACCESSIBILITY:
Text Accessibility encompasses the diverse factors that enable or hinder individuals’ ability to approach, attain a text or make it available. Based on: [Iwarsson and Ståhl, 2003].

Definition:
Text Accessibility

Building on this definition, in the context of text, accessibility involves multiple dimensions including linguistic structure, syntax, vocabulary, contextual structure, purpose, audience, layout, conceptual complexity, visual elements, stylistic features, tone, textual organization, and reader-writer relationships through use of pronouns, tense, and voice [Fulcher, 1997; Petrova, 2016; Zavattaro et al., 2015]. These dimensions are broadly categorized into intrinsic factors (inherent to the text itself, such as linguistic complexity and content structure) and extrinsic factors (related to presentation and format, such as font size, color, and layout). Research shows that text accessibility barriers affect various population segments differently, creating challenges for effective information delivery across demographics and content types [Štajner, 2021]. The following sections focus primarily on the intrinsic factors that influence text accessibility rather than formatting or display considerations.

2.1.1 Factors Influencing Text Accessibility

Each of the following factors can represent one or more text accessibility dimensions.

Text complexity creates barriers

Complexity According to OECD data, 16.7% of adults across 24 countries have literacy below Level 2 (requiring lexical simplification), while about 50% struggle with syntactic complexity, and 89.4% find conceptually dense texts difficult to comprehend [Štajner, 2021]. Scientific texts create higher cognitive demands due to complex syntax and specialized terminology [Petrova, 2016]. Lexical complexity particularly affects dyslexic children and poor readers, though simplifications can improve reading speed without compromising comprehension [Gala and Ziegler, 2016]. Non-experts often avoid authoritative scientific sources due to complex language, instead choosing potentially less reliable sources that prioritize engagement [Ermakova et al., 2021; Michielutte et al., 1992].

Text cohesion reduces cognitive demands

Text Cohesion and Structure The logical connection between ideas (cohesion) significantly influences comprehension, especially for readers unfamiliar with the subject.

Highly cohesive texts reduce cognitive demands by explicitly connecting ideas, particularly helping readers with limited subject knowledge [Ozuru et al., 2009]. Cohesive texts that personalize content through elaborations, specific headers, and thematic sentences increase recall of key information [Ozuru et al., 2009].

Style and Tone The reader-writer relationship, established through pronouns, tense, and voice, influences how readers connect with information [Fulcher, 1997]. Changing generic references to second-person possessive pronouns (e.g., replacing "the eye" with "your eye") improves transfer performance and reader engagement [Dutke et al., 2016]. Positive, personalized language fosters higher participation and trust compared to neutral or negative content [Zavattaro et al., 2015]. Communication tone affects audience reception, with stylistic features playing an important role in text accessibility [Fulcher, 1997; Zavattaro et al., 2015]. Another example is health education literature that is written on a level that makes it inaccessible to a large proportion of the population; studies show that using narrative text style rather than rigid bullet-point formats significantly improves comprehension, especially for individuals with lower reading skills, while maintaining interest across all literacy levels [Michielutte et al., 1992].

Personalization affects audience reception

Reader Demographics and Preferences Reader factors affecting the reading process include motivation, background knowledge, interest, education level, and reading strategies [Fulcher, 1997]. Studies indicate readers with minimal subject knowledge benefit most from low-complexity summaries, while those with greater familiarity prefer more complex content that preserves nuanced information [August et al., 2024]. High-familiarity readers tend to skip sections of low-complexity summaries, while simplified text may inadvertently increase readers' overconfidence [August et al., 2024]. Age can also influence preferences, with older adults preferring proper grammar, polite tone, clear instructions, and avoidance of abbreviations or excessive punctuation, while younger audiences more readily accept casual language and visual elements [Kuerbis et al., 2017], findings that come from a study examining mobile messaging preferences. Older adults respond better to "you"

Reader characteristics affect text interaction

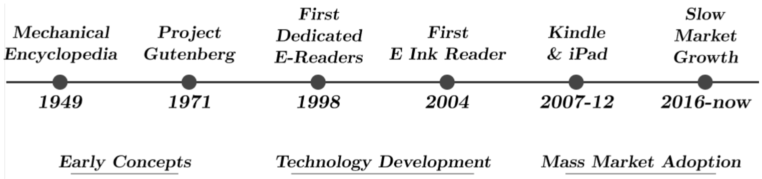

Figure 2.1: Timeline of Digital Reading Evolution, synthesized from key milestones drawn from various sources discussed in Section 2.2.1.

statements and dislike textese (e.g., "u r gr8"), reflecting cognitive processing differences where complex formatting can reduce readability for older populations [Kuerbis et al., 2017].

2.2 Digital Reading

2.2.1 Digital Reading Evolution

Digital reading technologies originated in 1949, preceding Project Gutenberg by over two decades.

Digital reading traces back to Robles' 1949 "Mechanical Encyclopedia" [Ludovico, 2018], predating the commonly referenced Project Gutenberg initiative of 1971 [Lebert, 2009], as shown in Figure 2.1. While experiments emerged in the interim (including early hypertext experiments in the 60s that allowed non-linear linking of digital text, with first working stand alone systems like HyperCard in the 80s, and Michael Joyce's hypertext fiction "afternoon" published in the 90s), the first dedicated e-readers appeared in 1998 with the Rocket eBook and SoftBook. This evolution continued when Sony's Librie, released in 2004 as the first commercially available E Ink reader, featured paper-like display characteristics that reduce eye strain compared to light-emitting screens and several other advantages like longer battery life. *Based on:* [Pilato, 2004; Bai et al., 2014; Bolter, 2001].

Amazon's 2007 Kindle launch transformed the market and catalyzed mainstream adoption.

The market transformed during the technology development phase with Amazon's 2007 Kindle launch, propelling digital reading into mainstream adoption (see Figure 2.1).

This moment saw e-book revenue in the United States increase from $869 million in 2010 to $2.07 billion by 2011, with e-books reaching 20.8% of revenue and 23.8% of unit sales in trade book publishing by 2013. *Based on:* [Wischenbart, 2013; Gilbert, 2015].

Amazon's dominance was fueled by its $9.99 flat pricing strategy for e-books, regardless of the title's print price or publisher costs. This approach disrupted the traditional book business, threatening brick-and-mortar bookstores' viability and reducing publishers' profitability. The pricing model ended through regulatory intervention and publishers' collective actions, shifting the competitive landscape of the digital publishing economy. *Based on:* [Ren, 2022].

Amazon's $9.99 pricing strategy disrupted traditional publishing economics and distribution models.

This disruption triggered competitive responses across the publishing ecosystem. Apple entered the market with iPad and iBooks in 2010, capturing approximately 10% of the e-book market by 2012. Barnes & Noble countered with its NOOK platform, growing from 2% market share in August 2011 to 7% by August 2012. Traditional publishers and booksellers developed their own platforms to challenge Amazon's dominance, exemplified by Germany's Tolino alliance formed in 2013 by major book chains including Thalia, Weltbild-Hugendubel, with Bertelsmann's Club and Deutsche Telekom as technology partners. Their pricing of €99 compared to Amazon's €129 for comparable devices demonstrated the competitive dynamic within the maturing market. *Based on:* [Wischenbart, 2013; Gilbert, 2015].

Market competition intensified with Apple, Barnes & Noble, and publisher-led platforms entering the e-book sector.

The rise of smartphones and tablets accelerated e-book adoption by expanding reading capabilities beyond dedicated e-readers. This diversification of reading devices contributed to the growth phase depicted in the timeline (Figure 2.1), with e-book market penetration reaching 20% in the USA and 10% in Germany by 2013. However, despite initial triple-digit growth rates, market expansion began to stabilize by 2013, entering the "Slow Market Growth" phase identified in the timeline. *Based on:* [Wischenbart, 2013; Gilbert, 2015; Chang and Kong, 2012].

Mobile device proliferation expanded e-book access beyond dedicated e-readers.

By 2025, the global e-book market has reached revenue

E-book market consolidation resulted in platform-dominated ecosystems with limited interoperability.

projected at $14.92 billion and approximately one billion readers worldwide. The market has consolidated around a few dominant platforms, with Amazon maintaining approximately 70% of e-book sales in the US and also in other larger book markets, for example Germany, where 68% of consumers report Amazon as their e-book source. This consolidation has created closed ecosystems where major platforms employ proprietary formats and business models. Despite the growth of e-books, they remain a small segment of the overall book market. According to 2024 market data, e-books account for only 16% of the total book market share, while audiobooks represent 8.6%. *Based on:* [Statista, 2025; Gilbert, 2015; Statista, 2023; Benhamou, 2015; Statista, 2025]

2.2.2 E-book Formats

E-book formats evolved from 1998 OEBPS, later EPUB.

The following presents the formats that are mainly used by current major e-book providers or were relevant during the mass market adoption phase (based on Section 2.2.1). All current formats specifically developed for reading e-books evolved from the Open eBook Publication Structure (OEBPS) developed in 1998, which later became the EPUB standard. Most e-book formats (except PDF) descend from OEBPS or EPUB 2.0.1, sharing XML and CSS foundations with integrated media support. The development of these specialized e-Reader formats was intended, among other things, to address the limitations that PDFs have in the digital reading context. *Based on:* [Bläsi and Rothlauf, 2013].

PDF preserves appearance and lacks reflowability.

PDF is an established document format with applications beyond e-books. It is standardized by the International Organization for Standardization (ISO) and adopted by governments in over 75 countries as their documentation format of choice. With billions of documents in circulation, PDF is the printing industry's required format for professional jobs. While PDF excels at preserving exact document appearance, its fixed layout creates usability problems on small screens where text cannot reflow to fit the display size. Nevertheless, PDF is widely used as an e-book format and supported by all major e-book platforms, including Amazon Kindle, Apple Books, and Google Play Books. *Based on:* [Rosenthol, 2013; Schwarz et al., 2018].

Amazon's Proprietary Formats (AZW, AZW3, KF8, MOBI, KFX, KPF) function exclusively within the Kindle ecosystem launched in 2007. Amazon is phasing out the MOBI format (completed March 2025) in favor of EPUB, DOCX, and KPF. These formats descend from OEBPS modified with XHTML, JavaScript, and Frames. *Based on:* [Bläsi and Rothlauf, 2013; Publishing, 2025; Scott and Orlikowski, 2022].

Amazon formats limited to Kindle. MOBI discontinued by 2025.

Apple Books primarily supports EPUB with proprietary extensions developed to address early EPUB limitations. Currently, EPUB is the only format supported for general book distribution on Apple Books. *Based on:* [Bläsi and Rothlauf, 2013; Inc., 2025].

Apple Books uses EPUB with proprietary extensions.

Barnes & Noble's Nook platform developed fixed layout extensions to EPUB while creating page images from PDF files, balancing format standardization with platform-specific optimizations [Bläsi and Rothlauf, 2013].

Nook combines EPUB extensions with PDF-based images.

EPUB has emerged as the industry standard format across all major platforms. Its primary advantage is reflowable text that adapts to different screen sizes and orientations. EPUB exists in several variants including flowing EPUB for reflowable content and fixed layout EPUB for precise positioning. It is the only format accepted by Apple Books, one of two primary formats for Google Play Books, and now accepted by Amazon's publishing platform. Despite increasing convergence around EPUB, platform-specific restrictions remain, particularly in Amazon's ecosystem which prohibits transferring purchases to non-Amazon devices. Section 2.2.3 will examine EPUB in greater detail as it forms the foundation for this thesis project. *Based on:* [Bläsi and Rothlauf, 2013; Publishing, 2025; Google, 2025; Inc., 2025].

EPUB: industry standard with reflowable text, platform restrictions persist.

2.2.3 Electronic Publication (EPUB) Format

Unless otherwise explicitly stated, the content of this section is based on the books *On the Interoperability of eBook Formats* by Bläsi and Rothlauf [2013], *EPUB 3 Best Practices* by Garrish and Gylling [2013], and the W3C specification [W3C, 2025].

EPUB maps most physical book components

EPUB is an open and free container format [Çelikbaş, 2011] (see Section 2.2.4) that supports most standard book structural elements in a systematized format. The specification enables implementation of covers, title pages, copyright information, tables of contents, prefaces, chapter divisions, sections, images, tables, footnotes, indexes, glossaries, bibliographies, and other conventional book components through standardized markup.

Additional functionality beyond print books

Furthermore, EPUB extends conventional book functionality through the following features which also address accessibility needs:

1. **Layout Options:**
   - Reflowable layouts that dynamically adapt to different screen dimensions and user preferences
   - Fixed layouts that maintain precise positioning when required (like PDF)
2. **Customizable parameters** that allow readers to adjust presentation according to their needs and preferences, allowing customization of fonts, text size, line spacing, margins, color schemes etc.
3. **Enhanced Media Integration:**
   - Media overlays for synchronized text-audio presentation
   - Embedded multimedia support (audio, video)
   - MathML for mathematical content rendering
4. **Navigation Enhancements:**

- Semantic structure tagging for improved content organization
- Page navigation references corresponding to print editions
- Skippable and escapable content markers for better reading / narration flow

5. **Accessibility metadata** that provides standardized descriptors of a publication's accessibility characteristics, enabling discovery of content based on specific accessibility needs and capabilities.
6. **WCAG compliance support**: The World Wide Web Consortium [World Wide Web Consortium (W3C), 2025] define in their EPUB Accessibility requirements [Publishing@W3C Publishing Maintenance Working Group, 2024] how to apply the Web Content Accessibility Guidelines standard [World Wide Web Consortium (W3C), 2025] to EPUB publications. It provides specific techniques and conformance requirements for making digital books perceivable, operable, understandable, and robust for all users, including those using assistive technologies.

2.2.4 EPUB Container Structure

Unless otherwise explicitly stated, the content of this section is based on the books *What is EPUB 3* by Garrish [2011], *EPUB 3 Best Practices* by Garrish and Gylling [2013], and the W3C specification [W3C, 2025].

The EPUB (Electronic Publication) format, now at version 3.3, is a container format for e-books that packages several web technologies. It encapsulates XHTML, CSS, XML, SVG, media formats (PNG, JPEG, MP3, AAC, etc.), font technologies (TrueType, OpenType, WOFF, etc.), and supporting technologies like JavaScript within a ZIP-based container, which together constitute an e-book. EPUB follows a defined structure with five main components:

EPUB is a container format bundling XHTML, CSS, XML, SVG, and other web technologies that together constitute an e-book.

1. A **META-INF** directory containing configuration files and the first place an epub reader looks for the cen-

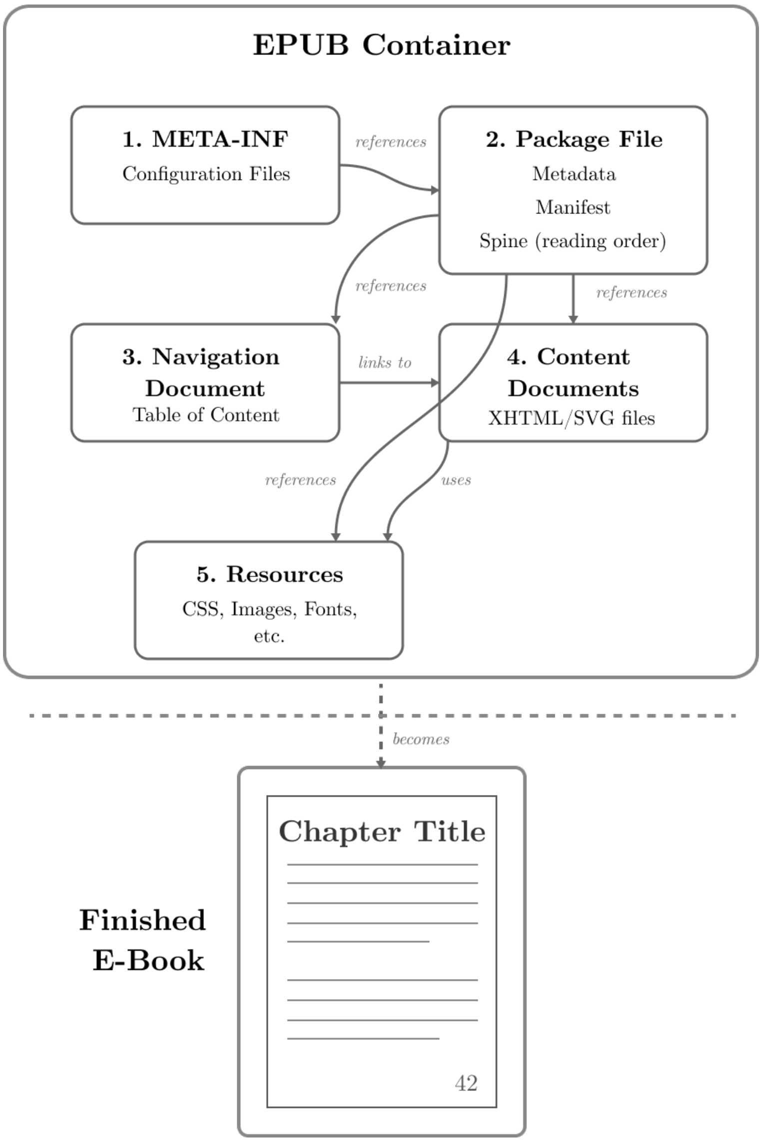

Figure 2.2: EPUB 3.3 [W3C, 2025] Container Structure

tral configuration file **container.xml**, which then explicitly references the next central configuration file of the epub: the package file (2)

2. The **package file** (`.opf`) that serves as the central manifest and organizational hub of the publication, explicitly referencing:

    - A navigation document (3) in its manifest section
    - All content documents (4) and resources (5) in its manifest section
    - The default reading order of content documents in its spine section

3. A **navigation document** provides the table of contents with references to content documents
4. **Content documents** (XHTML/SVG files) that form the actual content of the e-book
5. **Resources** (CSS, images, fonts, etc.) that are referenced by/embedded in content documents

Each component plays a specific role in the EPUB ecosystem, with explicit references creating the structured relationship necessary for e-readers to properly render the publication.

Chapter 3

Related Work

This chapter presents prior work on systems that leverage large language models (LLMs) for text adaptation and accessibility, as well as on natural language interfaces. It includes research on prompt engineering and key challenges in LLM-based systems, including hallucinations, latency, non-determinism, and information integrity. The chapter also outlines studies on the accessibility benefits of e-books, developments in natural language processing and text simplification systems, and research on survey and logging methods relevant to the research toolkit.

3.1 E-Book Accessibility Advantages

McNaught and Alexander [2014] examined the accessibility features of e-books compared to traditional print materials. Their research identified that properly formatted e-books offer extensive customization features unavailable in physical texts, including text magnification with reflow capability that maintains readable line lengths and improves reading speed. They documented how color and contrast customization options benefit both visually impaired and dyslexic readers, addressing limitations inherent to static printed materials. Additionally, their work highlighted the integration of text-to-speech technology that allows for au-

Digital texts enable customization impossible in print: text reflow, contrast options, and audio conversion dio conversion using high-quality, natural-sounding voices across multiple languages, providing alternatives not possible with physical books.

3.2 Natural Language Processing

Natural Language Processing (NLP) is fundamental to this thesis as it provides the technological foundation for the text transformation capabilities of the MagicTextreader, enabling the system to process, understand, and modify text to meet diverse user needs.

NLP allows machines to understand and create natural text

Chopra et al. [2013] characterize NLP as a branch of artificial intelligence and language studies that help computers work with human language. They state that NLP enables machines to understand and generate natural language text.

Evolution from rules to statistical methods

NLP began with rule-based systems in the 1950s, where developers created explicit language rules Johri et al. [2021]. In the 1980s, a fundamental reorientation happened toward statistical NLP approaches that replaced complex hand-crafted rules with data-driven methods like Hidden Markov Models (HMMs), probabilistic Context-Free Grammars, or Support Vector Machines (SVMs) [Nadkarni et al., 2011].

Word2Vec shifted focus to neural networks

Subsequently, another shift to neural network-based approaches happened in the 2010s, accelerated when Mikolov et al. [2013] created Word2Vec, a simple neural network that learns how words relate to each other, which worked well for many NLP tasks, as highlighted in the survey by Zhao et al. [2023].

Evolution to transformers for effective sentence processing

Tan et al. [2020] and Young et al. [2018] describe in their reviews the progression of neural architectures for effective sentence processing in NLP. According to their reviews, Sutskever et al. [2014] gave a neural network architecture (built with recurrent neural networks (RNNs) and specifically Long Short-Term Memory (LSTM) networks) to transform one text sequence into another (like English sentences into French sentences). Then by the attention mechanism from Bahdanau et al. [2014] that allowed models to focus on relevant text segments and finally by Vaswani et al. [2017]'s transformer architecture, which enabled neural networks to process entire sentences at once rather than word by word sequentially, while still maintaining focus on relevant text segments.

This led to transformer-based models such as BERT [Devlin et al., 2019], which, according to Rogers et al. [2021], achieved state-of-the-art results across numerous benchmarks and was subsequently adopted as a standard baseline in many of them.

Transformers set new baseline

According to the survey by Zhao et al. [2023], Large Language Models (LLMs) like GPT [Brown et al., 2020] that are also built on the transformer architecture represent the current state of NLP development, showing new capabilities such as in-context learning, instruction following, and step-by-step reasoning not found in earlier, smaller models.

LLMs are current state-of-art with new capabilities

3.3 Text Simplification Systems

Building upon the NLP advancements discussed previously, text simplification systems represent a specific application area that is particularly relevant to this thesis, as they align with one of the key dimensions through which the MagicTextreader can transform text. According to Štajner [2021], early research on text transformation systems was mainly dedicated to automatic text simplification systems. Furthermore, since the late 1990s, automatic text simplification has emerged as a natural language processing method with potential to improve text accessibility for people with reading or cognitive disabilities. Meanwhile, Shardlow [2014] notes that the first practical uses focused on aviation manuals, where Boeing created grammar and style checkers to help writers produce simplified English documentation for their commercial aircraft manuals following the ASD-STE100 [Aerospace and Defence Industries Association of Europe (ASD), 2021] standard.

Early text simplification systems emerged from accessibility research and industry standards

Rule-based approaches dominated early text simplification systems despite their limitations

Shardlow [2014] explain that early lexical simplification worked by replacing complex words with simpler synonyms, often using basic word frequency measurements to find suitable replacements. They remark that this process sometimes changed or lost the original meaning, which was one reason these rule-based approaches had major limitations in scaling up and adapting to different fields and languages. Štajner [2021] describes how the field gradually grew to include various simplification techniques, such as word replacement, sentence structure reorganization, and adding explanations. Despite this progress, these rule-based systems needed extensive manual effort, which limited how practical they were in real-world applications according to Shardlow [2014] and Štajner [2021]

The field evolved from rule-based to machine learning approaches, with research growing quickly since the 1990s

According to Štajner [2021], the field has witnessed a shift from rule-based systems to data-driven approaches. They document that research in English text simplification has evolved through three distinct phases: rule-based systems (until 2010), data-driven supervised machine learning (2010-2014) where algorithms learn from labeled examples and rely on human-engineered features to identify complex text patterns, and neural text simplification systems (2015 onwards) which use neural networks that learn complex language patterns from large amounts of data with less manual engineering. Further, a crucial task for these data-driven approaches is finding high-quality parallel datasets of paraphrases that can serve as training data. They also mention that research on text simplification has both evolved methodologically and increased in volume over time, with few publications using terms like 'text simplification,' 'syntactic simplification,' 'sentence simplification,' or 'lexical simplification' in the 1990s, compared to over 1,000 publications annually by 2019 on Google Scholar, including more than 60 publications containing these keywords in their titles.

First data-driven systems treated simplification as monolingual translation using Wikipedia parallel corpora

For example, Zhu et al. [2010] proposed a data-driven system that trained a tree-based model using paired sentences from regular and simple Wikipedia, treating text simplification as translation within the same language. Similarly, Wubben et al. [2012] also treated simplification as a translation problem but developed a phrase-based machine translation approach (which breaks sentences into meaningful chunks or 'phrases' and uses statistical models learned from parallel texts to determine the most likely translations for each chunk and their proper order [Osborne, 2010]), learning simplification rules directly from parallel Wikipedia texts without relying on syntactic trees, which allowed their system to capture a wider range of simplification operations while maintaining grammaticality.

From then on, neural text simplification systems became according to Štajner [2021] more common. Nisioi et al. [2017] applied the attention mechanism to text simplification, implementing sequence-to-sequence models with attention for simplifying text, which their extensive human evaluation showed could significantly outperform the best phrase-based and syntax-based machine translation approaches when ranked with appropriate metrics. Later, Zhang and Lapata [2017] built upon the attention mechanism by using reinforcement learning to balance simplicity, fluency, and meaning preservation.

Neural simplification based on attention mechanisms outperformed traditional machine translation approaches

Hybrid approaches emerged like Maddela et al. [2020] who proposed a system that uses rule-based methods for splitting and deletion operations, paired with neural paraphrasing, enabling better content preservation while allowing users to control various aspects of readability, addressing limitations of end-to-end models that primarily focus on deletion operations.

Hybrid approaches emerged combining rule-based methods with neural techniques

Martin et al. [2019] developed transformer-based systems allowing users to control specific aspects of the simplified text, such as length and word complexity. Martin et al. [2020] introduced MUSS, a language-agnostic framework that leverages transformer models with controllable generation and large-scale paraphrase mining from web data to perform sentence simplification without requiring labeled parallel simplification data.

Transformer-based approaches enabled controllable simplification and language-agnostic frameworks

More recent approaches use large language models and are discussed in the next section.

3.4 Large Language Models

Building upon the NLP evolution described in section 3.2, Large Language Models (LLMs) represent, according to Zhao et al. [2023], the state of the art in NLP with diverse capabilities. This section examines their development, technical aspects, and applications in text transformation systems, with a focus on their relevance to the MagicTextreader. LLMs form the technological foundation of the MagicTextreader's text transformation capabilities.

3.4.1 Technical Foundations

LLMs' scale enables unique capabilities beyond smaller models

Large Language Models (LLMs) are neural network systems that can understand and create human language [Chang et al., 2024]. A key factor is their scale—according to Wei et al. [2022], this larger scale gives LLMs special abilities that smaller language models simply do not have.

Transformer architecture with self-attention forms LLMs' core

These models are latest of the advancements presented in Section 3.2 and the core of these models is according to Chang et al. [2024] and Zhao et al. [2023] mainly the parallelizable Transformer architecture with self-attention mechanism that was introduced by Vaswani et al. [2017].

Transformers generate text through probabilistic next-word prediction

Following Vaswani et al. [2017], on an abstract and simplified level, Transformers generate text by determining the most probable next word based on an existing text. They repeat this process continuously. After adding a new word, they recalculate again which word is most probable to follow next, based on the now extended text. This is how a coherent text sequence is created word by word.

Next-word prediction enables solving complex tasks

According to Raffel et al. [2020]; Brown et al. [2020]; Wei et al. [2022], the ability to determine the most probable next word in various contexts enables transformers to solve complex tasks. Following them, for logical problems, the correct solution often corresponds to the most probable text continuation, provided the model was trained with correct examples.

This text generation approach is not new in principle – earlier sequence-to-sequence models based on recurrent neural networks (see Section 3.2), like [Bahdanau et al., 2014] showed in their translation approach, also generated text by sequentially predicting the most probable next token based on previous context.

Self-Attention allows parallel processing and tracking long dependencies

However, according to Vaswani et al. [2017]; Chang et al. [2024], the difference between recurrent neural networks and the Transformer architecture lies in the processing method: While RNNs must process texts sequentially and word by word, a Transformer can process a text in parallel. According to them, the Self-Attention mechanism, the core element of this architecture, determines which other words are relevant for each word in the text. Thus, a word at the beginning of a sentence can directly influence the interpretation of a word at the end (and it can even keep track of long dependencies like words that influence each other but are far away from each other) – without going through a chain of calculations. According to them, this connection between all words allows the model to capture relationships such as references, dependencies, and thematic connections more effectively.

Early Transformers trained for specific purposes

Early language models based on the Transformer architecture are typically trained for specific purposes with corresponding training examples [Chang et al., 2024].

LLMs' pre-training on massive datasets enables emergent capabilities

Large language models, however, follow a different training approach. They undergo pre-training on massive text collections, as proposed first by Radford et al. [2018]. Their learning objective involves predicting the next word in a sequence, as in Vaswani et al. [2017], or filling in masked words within text, as in Devlin et al. [2019]. As these models scale in size, new capabilities emerge at specific thresholds, without showing signs until that critical point is reached [Wei et al., 2022]. These capabilities are referred to as emergent capabilities, and they enable large models to solve a wide range of tasks across various benchmarks that smaller models cannot [Wei et al., 2022].

Fine-tuning aligns models with preferences

Following this initial pre-training phase, despite their emergent capabilities, these models are still further fine-tuned - adapted through additional training to perform specific applications aligned with human preferences and requirements [Ouyang et al., 2022].

3.4.2 Prompt Engineering

Prompt Engineering guides LLM text generation

As discussed earlier in this section, LLMs generate text by predicting the most probable continuation of a given input based on prior training and fine-tuning. According to Sahoo et al. [2024], this input is referred to as the prompt, and the systematic design of such prompts to achieve desired outcomes is known as prompt engineering.

Shot terminology categorizes example inclusion approaches

One aspect of prompt engineering is whether to include examples of similar tasks and their solutions in a prompt, and this can be expressed using the shot terminology systematized by [Brown et al., 2020]. According to them, zero-shot prompting includes no examples in the prompt—only natural language instructions. One-shot prompting incorporates a single example before the actual task, and few-shot prompting contains multiple examples. They also refer to this as in-context learning, as the model "learns" the task from the context (i.e., the existing/provided text).

Increased examples improve model accuracy

Brown et al. [2020] compared zero-shot, one-shot, and few-shot prompting methods on their GPT-3 model across various tasks, including closed-book question answering, translation, common sense reasoning, reading comprehension, and arithmetic. They observed that accuracy generally improved as the number of example shots increased, demonstrating that the model effectively learns from examples.

Chain-of-thought prompting enhances reasoning tasks

Wei et al. [2022] introduce chain-of-thought prompting, showing that guiding models to reason step-by-step using examples improves accuracy on complex tasks. They demonstrate that this technique especially increases arithmetic reasoning performance and argue that it expands the set of tasks language models can perform successfully. Although potentially not directly relevant for this work, it should be considered when designing effective prompts.

White et al. [2023] introduce "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT," documenting patterns like Persona (instructing the LLM to adopt a specialist role), Reflection (directing the LLM to explain its reasoning process), Fact Check List (commanding the LLM to list claims requiring verification), Template (telling the LLM to format output according to a specific structure), and Cognitive Verifier (ordering the LLM to break complex questions into sub-questions before answering). This catalog of prompt instructions might serve as a good foundation to formulate a prompt for a specific transformation task.

Prompt Pattern Catalog systematizes effective prompting techniques

Zhou et al. [2022] show that LLMs themselves can be used to craft/optimize a prompt. OpenAI [2025] provide a meta prompt to generate a prompt from a given task description, and Anthropic (author of Claude) even provides an interactive tool to help create or refine existing prompts leveraging their own model [Anthropic, 2025].

LLMs optimize their own prompts

Providers of LLMs like OpenAI (ChatGPT), Anthropic, Google, Meta, etc. have their own guidelines [OpenAI, 2025; Anthropic, 2025; Google DeepMind, 2025; Meta AI, 2025] on how to engineer prompts. Most of them mention concepts already listed here, such as providing examples in the prompt, as well as additional considerations like advocating for precise instructions and mixing natural language with markup languages like XML [Bray et al., 2008], JSON [Bray, 2017], or Markdown [CommonMark Community, 2020] to separate parts from each other and introduce hierarchy, among others. One common technique used by these providers is to structure a prompt as a sequence of messages, allowing the language model to generate the next message in the sequence. Most providers offer APIs that handle message separation and delimitation internally.

Providers of LLMs offer guidelines

Li et al. [2023] show that incorporating emotional stimuli (derived from psychological theories) into prompts (such as appending "This is very important to my career") significantly improves performance. Their approach increased scores by 115% on Big-Bench [bench authors, 2023] and by 8% on instruction induction [Honovich et al., 2022], both LLM benchmarks that contain diverse tasks.

Emotional stimuli in prompts boost performance

3.4.3 LLM Hallucinations

According to Huang et al. [2025], a limitation of LLMs is their tendency to generate hallucinations—outputs that are linguistically plausible but factually not correct. They group these into factuality and faithfulness hallucinations. They state that factuality hallucinations occur when a model provides factually incorrect information, for example, in response to a question. Further, faithfulness hallucinations are more relevant in text transformation tasks, where the output fails to accurately reflect a given instruction or context. These are further divided by them into three subtypes: instruction inconsistency, where the model does not follow the instruction (e.g., it fails to apply a requested transformation); context inconsistency, where it ignores or misuses contextual information (e.g., it generates a transformed version that sounds fluent but is not based on a given page); and logical inconsistency, where it produces errors in reasoning.

3.4.4 Response Time of LLMs

Length of generated text influences response time the most

Response time is an important factor in interactive systems [Shneiderman, 1984], and thus the response time of an LLM becomes relevant for an interactive system using LLMs like the MagicTextreader. According to Narayanan et al. [2023], the reponse time is generally influenced by the model size, underlying hardware, and optimizations. Further they state, the response time of an LLM depends on the length of the prompt and the length of the generated text. They show a roughly piecewise linear relationship between these factors and response time. From their evaluation, it can be estimated that the effect of output length on response time is approximately 250 times greater per token than that of the input, highlighting the dominant role of the generated text length. To provide users with useful feedback during transformations, estimating response time becomes relevant. Since it largely depends on the length of the generated text, predicting this length might offer an entry point. Zheng et al. [2023] demonstrate that LLMs themselves can be used to predict the expected length of their output given a prompt.

3.4.5 Non-Determinism of LLMs

Most LLMs generate different output text when given the same input prompt multiple times, as their generation process involves elements of random sampling to maintain fluent and varied responses. Non-determinism persists even under settings designed to enforce consistency, due to system-level factors beyond user control. [Atil et al., 2025; Holtzman et al., 2020]

3.5 Information Integrity Challenges in LLM-Based Systems

As LLMs produce hallucinations (see Section 3.4.3), the issue of information integrity arises for interactive systems that present LLM-processed information to users. Information integrity is a concern for businesses, governments, society, and individuals, as false information can lead to decisions that are not in the best interest of individuals or organizations, with consequences that can cost the economy billions of dollars [Geisler et al., 2003; Lewandowsky et al., 2012; Kuklinski et al., 2000]. This concern is particularly relevant given that knowledge is durable and persists without declining with age [Marsh et al., 2016]. The repetition of false or misleading information increases the likelihood that people will believe it [Pillai and Fazio, 2021]. The impact of false information from LLMs, specifically ChatGPT, has become a research focus, with a Google Scholar search for "chatgpt false information impact" yielding over 67,000 results as of May 2025. Amaro et al. [2023] investigated ChatGPT's fake information impact through a user study and found a statistically significant difference in satisfaction and trust between users who discovered early that ChatGPT produced false information compared to those who discovered this later during their interaction.

LLM hallucinations threaten decision-making integrity

Late error discovery reduces user trust and satisfaction

3.6 Language Models for Text Adaptation and Accessibility

3.6.1 Prompting Engineering for Text Transformation

Augmented zero-shot learning enables diverse style transfers

Reif et al. [2021] propose a prompting method called "augmented zero-shot learning" to transform text stylistically across multiple dimensions, such as altering sentiment, increasing descriptiveness, incorporating metaphors, and adding comic or melodramatic elements, among others. Their approach provides diverse examples of different rewrites (each demonstrating different style changes) so that the prompt does not necessarily contain an example of the specifically requested style transfer but rather examples of other style transfers. They conducted human evaluations comparing the "style strength" of their method's outputs against human-rewritten "ground truths." Their results indicate that this method produces rewrites that humans evaluate as comparable to human-written ones in terms of style strength.

Multi-step prompting transfers author style with minimal samples

Patel et al. [2022] propose a multi-step prompting technique to transfer an author's style using just a few samples. They demonstrate this by transferring styles between different Reddit users and their posts. Their process involves three steps: First, they convert a small sample (N=15) from each author to neutral text using zero-shot prompting. Second, they have an LLM describe each sample's style using comma-separated adjective lists. Finally, they perform style transfer through few-shot prompting by providing examples of transformations between neutral text and original samples (described by the adjective lists), then having the LLM complete a neutral text from another author in the described style. They evaluate their approach against an authorship identification model and demonstrate that it successfully fools the model over 50% of the time, with texts being rated more likely to be in the target author's style than the source author's style.

LLMs outperform state of the art in sentence simplification tasks

Feng et al. [2023] tested LLMs for sentence simplification using very basic prompts in zero-shot, one-shot, and two-shot settings. Their results showed that one-shot prompting performed best and exceeded recent state-of-the-art systems. They found that: (1) GPT-3.5 and ChatGPT better removed unnecessary information while adding helpful content; (2) ChatGPT performed well on Portuguese and Spanish tasks, showing strong multilingual abilities; and (3) Human evaluators rated LLM simplifications similar to human-written ones.

3.6.2 Methods for Controllable Text Generation with LLMs

Controllable text generation balances constraints with quality

Liang et al. [2024] examines controllable text generation (CTG) for large language models. CTG allows to generate text that meets requirements while maintaining quality. The authors present CTG as an expanding research area with various methods for controlling text attributes and content. Their main contribution is organizing CTG into two key dimensions: how well the text follows control conditions and how it maintains quality. They also divide CTG approaches into content control and attribute control categories, and review implementation methods during both training (of the LLM itself) and inference phases (in the prompt when using the model). This organized overview provides valuable insights for understanding text transformation techniques that can be applied to accessibility applications.

Classification-guided incremental style transfer outperforms direct prompting

Luo et al. [2023] describe an approach that leverages LLMs for style transfer by prompting an LLM to classify the style rather than directly prompting it to transform text and then performing a discrete search to modify sentences incrementally until the desired style is achieved. They also compare this method against zero-shot prompting with direct transformation instructions, demonstrating that their incremental classification-guided approach achieves better results.

3.6.3 LMs/LLMs for Accessibility-Focused Text Transformations

Medical text simplification requires audience-specific approaches

Makhmutova et al. [2024] created a multi-modal medical text simplification dataset consisting of triplets: original text, human-written simplified text, and a ChatGPT-simplified version generated using a basic zero-shot prompt. Their evaluation concludes that while ChatGPT generally transforms texts to be quite understandable, it is not yet reliable enough for medical text simplification. They specifically advocate for target group-specific simplifications, noting the challenge of determining which terms require simplification and which do not depends on the audience.

GPT-4 simplifications preferred despite lower guideline adherence; study notes importance of involvement

Uricchio et al. [2024] conducted two exploratory studies examining ChatGPT's knowledge of Easy to Read (E2R) guidelines and text simplification capabilities according to these guidelines. GPT-4 demonstrated significant guideline knowledge when directly questioned. Using zero-shot prompting that positioned the model as an accessibility expert with detailed guidelines, they found GPT-3.5 could only simplify 1/7 texts according to the E2R guidelines, while GPT-4 managed 4/7, both underperforming compared to simplifications manually crafted by instructed students. In their evaluation, however, participants with intellectual disabilities mostly preferred GPT-4's simplifications over manually crafted ones, despite GPT-4's simplifications following the guidelines less. In their discussion, they acknowledge this result by citing the Inclusion Europe Association's statement [Inclusion Europe, 2018] that "people with intellectual disabilities know best what is good for them," recognizing that what works best in practice may not always align with prescribed guidelines and that the users for whom the text is intended should be able to take part in this process.

LLMs enable adaptive complexity interfaces for personalized reading

August et al. [2024] compare language complexity summaries written by experts versus those generated by language models through three studies. They find that LLM-generated lower complexity summaries can match the quality of expert-written ones. They conclude that the appropriate complexity level depends on the audience and advocate for using LLMs to improve science communication. They specifically recommend that interface designers create interactive and adaptive reading interfaces that can generate summaries tailored to readers' needs, allowing users to select different complexity versions and using brief surveys to determine optimal adaptations.

Narrative perspective shifting offers potential accessibility implications

Chen [2022] study the transformation of narrative perspective as a text modification task. Their approach employs neural architecture based on the BERT language model to identify and alter entity references. Chen [2022] note that this perspective-shifting capability alters the reading experience and could be applicable across various text genres including fiction, educational materials, self-help resources, and self-diagnostic tools, potentially offering implications for text accessibility.

3.6.4 Interactive LLM Systems for Reading Enhancement or Text Generation

Interactive prototypes address PDF reading experience

Lo et al. [2023] investigate whether advances in AI and HCI can enhance reading interfaces for PDFs. They identify five key challenges readers face with research papers: discovery of relevant literature, efficient skimming as publication volumes grow, comprehension of dense technical content, synthesis across multiple papers, and accessibility limitations of static PDF formats. The authors investigate ten interactive prototypes that address these challenges, including systems that highlight citations to recently viewed papers, color-code and make navigable different discourse facets (e.g., objectives, methods, results), provide popup explanations for technical definitions, and offer on-demand LLM-based summarization and question-answering for complex sections.

Kindle LLM integration enhances reading continuity

Rex [2024] developed a minimalist interface for Mixtral, an LLM by Mistral AI [Mistral AI Team, 2023], optimized specifically for Kindle devices. This implementation allows readers to easily look up information using the LLM without needing to switch devices during reading sessions, demonstrating a reader system that integrates an LLM for better accessibility even while requiring users to switch to the Kindle built-in browser.

Interactive generation improves text customization

Faltings et al. [2023] propose an interactive text generation architecture that simulates an interactive user during training to steer a model step by step to edit text until it satisfies specific criteria. They argue that while most LLMs can generate high-quality text, the typical zero-shot prompts used for generation are often underspecified, making it difficult for models to satisfy user needs without additional information. Their architecture employs both an LLM and a token edit model that incrementally substitutes parts of existing text. This interactive approach, which enables users to steer text transformation toward a desired version, represents a consideration for designing systems like the MagicTextreader.

3.6.5 Related LLM Applications in Educational or Content System Contexts

LLM-based emoji translation boosts word guessability

Klein et al. [2024] use few-shot prompting with GPT-4 to translate text into an emoji-based version. They then examine whether these emoji translations support word prediction in the original text. In a controlled study, they find that, with the help of the emoji translation, the guessability of masked words increases by 55%. The authors suggest that emoji translations could support applications such as reading acquisition for children or language learning for adults, as emojis provide sufficient semantic coverage to represent many words.

LLM-based reading companions improved comprehension and engagement

Chen and Leitch [2024] examine the use of LLMs as reading companions to support the learning of students. In a semester-long study, students using Claude's LLM [Anthropic, 2023] showed improved comprehension and engagement compared to a control group. The authors also warn about risks such as overreliance. They emphasize the need for responsible design to support learning and protect student well-being.

Zhiyuli et al. [2023] use LLMs to rate books and generate summaries, rather than transforming e-books. They let the model rate books for a general or specific audience using zero-shot (providing only title and author) and few-shot approaches (adding tables of ratings for other books by the same author). Comparing against ratings from Goodreads [Goodreads, 2025], and Douban [Douban, 2025], they found their approach outperformed established methods, with accuracy improving when transitioning from zero-shot to few-shot. In their summary evaluation, human annotators blindly rated summaries generated by GPT-3.5 [OpenAI, 2025] and Wenxin (Baidu's LLM) [Wang et al., 2021]. Wenxin-generated summaries outperformed both GPT-3.5 and expert-written ones from Douban, indicating GPT-3.5's limitations for this task. Their methodology for building the evaluation dataset is notable, as they selected books with the highest number of comments or ratings.

LLMs generate accurate book ratings and summaries

Sajja et al. [2024] introduce a platform that uses LLMs as teaching assistants to provide an interactive framework for personalized learning. They evaluate how AI can be integrated into existing learning platforms, which educational content is most appropriate for adaptation, how different learning styles can be supported, and what challenges arise when building such systems. Their findings demonstrate that LLM-generated quizzes, flashcards, and assessment tests effectively deliver personalized learning experiences. They identify extracting structured data from PDFs as a key challenge in this process. Their approach to content adaptation via LLMs provides relevant insights for text transformation systems such as the MagicTextreader.

LLMs enable interactive personalized learning experiences

3.7 Natural Language Interaction Systems

Natural language interfaces (NLIs) allow users to interact with systems via human language, either through speech or text [White and O'Connor, 2022]. They include chatbots, voice assistants, and interactive voice response systems [White and O'Connor, 2022]. According to Følstad and Brandtzæg [2017], these interfaces are increasingly seen as

NLIs shift HCI focus from GUIs to dialogue central to HCI, shifting the design focus from graphical user interfaces to conversation and dialogue.

The W3C [White and O'Connor, 2022] highlights that NLIs support multiple input and output modalities, making them particularly relevant for accessibility, as users with different abilities can choose their preferred mode of interaction.

Chatbots enable goal-driven interaction

The rise of chatbots exemplifies this shift as they act as natural language interfaces to data and services, enabling goal-oriented interaction in both text and voice formats [Følstad and Brandtzæg, 2017]. Motivational studies show that users value chatbots for productivity, entertainment, and curiosity [Brandtzaeg and Følstad, 2017].

NLIs offer flexible, intuitive, and context-aware interaction

Hendrix [1982] list several advantages of natural language interfaces. These include their flexibility (usable across a wide range of use cases), the minimal training required for users, and the ease of remembering natural language compared to interfaces that require structured commands. They also mention that natural language can be faster than using menus or writing queries, and it allows follow-up questions that build on the context of previous dialogue.

Effective NLI design incorporates social and conversational intelligence

Designing for NLIs also requires attention to social dynamics. Chaves and Gerosa [Chaves and Gerosa, 2021] identify key social characteristics for chatbot design, such as conversational intelligence (e.g., turn-taking), social intelligence (e.g., empathy), and personification (e.g., tone and identity).

NLIs increasingly rely on LLM-based architectures

As natural language processing evolves toward LLMs (see Section 3.2), this trend is also reflected in the development of natural language interfaces and chatbots, which increasingly rely on LLM-backed systems. This observation is supported by a brief personal review of recent research on Google Scholar, where most recent NLI-related publications are based on LLM technologies.

3.8 Survey and Logging Methods in HCI

Survey research and event logging are established methods in HCI and ethnographic studies [Müller et al., 2014] and relevant to this thesis as the foundation of the research toolkit must provide these methods to support the mixed-methods study conducted.

3.8.1 Surveys

Survey research is often used in combination with other research methods and in HCI, surveys can follow or precede qualitative studies such as ethnographic research to quantify observed insights, or to identify general patterns for further exploration [Müller et al., 2014].

Müller et al. [2014] note that in HCI, surveys are a valuable method for gathering information on user behavior, preferences, demographics, task success, motivations, and experiences with technology.

Question types in surveys can, according to Müller et al. [2014], be open-ended or closed-ended. Further, open-ended questions allow respondents to provide answers in their own words as free text, which is useful when possible answers are not well defined. Closed-ended questions provide a predefined set of options and are suitable for measuring frequencies, attitudes, or comparisons through formats such as single-choice, multiple-choice, rating, or ranking scales [Müller et al., 2014].

Biases in survey responses are, according to Müller et al. [2014], a central concern and list the following biases. Satisficing occurs when respondents provide minimal-effort answers. Acquiescence bias refers to a general tendency to agree with statements regardless of content. Response order and question order biases may also influence how answers are selected based on the position or sequence of items in the survey. Social desirability bias leads participants to respond in ways they believe will be viewed positively.

3.8.2 Event Logging and Interaction Tracking

Event logging complements ethnographic observation

According to Crabtree et al. [2006], ethnographic studies often focus on understanding user behavior in a certain context. Further, in this setting, event logging and interaction tracking are used to complement traditional observational methods.

Crabtree et al. [2006] examine the use of system-generated event recordings in ethnographic studies of ubiquitous computing. They describe how these recordings are used alongside video data to support the analysis of interaction that is distributed across devices and locations. Their work focuses on the development and use of 'record and replay' tools to support the reconstruction of user activity.

Process Mining Handbook defines structured formats for analyzing event sequences

The Process Mining Handbook [Van Der Aalst and Carmona, 2022] introduces standard formats for event logs. Each event typically includes at least a case identifier (to associate the event with a specific user or session), an activity label (to describe the type of action performed), and a timestamp (to determine the order and timing of events) but can also include many other attributes to enhance context. This structure supports the analysis of sequences of events in interaction data.

Chapter 4

Design and Implementation of the MagicTextreader

This chapter presents the design and implementation of the *MagicTextreader* system, progressing from the underlying motivations and system overview through system goals, scope definition, requirements analysis, key design decisions, and finally the technical implementation.

4.1 Motivation

Before introducing the system, this section restates the underlying motivation. The challenges and opportunities identified in Chapter 2 "Background" and 3 "Related Work" reveal three key findings that inform the design and implementation:

- Text accessibility is widely recognized as a challenge, with studies indicating that a large portion of readers struggle with complex or specialized language. Additionally, research suggests that readers vary in their cognitive processing styles and reading preferences.

- The capabilities of LLMs present opportunities for transforming texts to address text accessibility needs, as well as personalizing reading experiences.
- Despite these accessibility challenges and the emerging capabilities of LLMs, current e-readers address only limited accessibility needs (e.g., font size adjustment, screen brightness), while existing research demonstrates isolated applications of LLMs for text transformation without investigating or integrating these capabilities into comprehensive, user-controlled reading systems.

To address this gap, the *MagicTextreader* system was developed as both a functional prototype and research toolkit for investigating the prototype in use.

4.2 System Overview

MagicTextreader enables modular, user-driven AI text adaptations

The *MagicTextreader* system provides a prototype of an AI-enhanced e-reader that allows users to apply modular text transformations along various text accessibility dimensions, categorized by their control types: toggles (for binary changes), choices (for discrete options), and sliders (for continuous adjustments between two reference points). Using these modular transformations, many types of accessibility dimensions can be addressed—such as complexity, tone, style, format, and personalization. Additionally, a built-in chat interface allows users to define their own transformations via natural language, which are then mapped to one of the three control types.

Prototype is embedded in platform as research toolkit

An accompanying evaluation platform positions the system as a research toolkit, supporting participant management, pre- and post-reading questionnaires, logging user event data, and enabling both initial investigation and future research into AI-enhanced reading technologies.

The following sections explain the system's core mechanism in detail, establish the design goals, and define the scope and boundaries.

4.2.1 Core Mechanism

The core mechanism of the *MagicTextreader* enables readers to transform text in e-books according to their preferences at the point of need.

For example, a reader may come across a section with specialized terminology or complex explanations while reading a non-fiction book. In such a case, the reader could access transformation controls that offer various parameters for text modification. One example might be a complexity control with levels like "Original", "Intermediate", and "Simple". A reader without domain-specific knowledge could select "Intermediate" and the system is designed to transform and present the passage in a more accessible language. Figure 4.1 illustrates an exemplary user interaction with the mechanism.

Example: Complex text is simplified on demand via complexity control

Exemplary illustration of user interaction with the core mechanism of MagicTextreader

4.2.2 System Goals

The following high-level goals were formulated in order to arrive at a suitable implementation. These goals provide the foundation for deriving system requirements and guiding subsequent design decisions that shape the implementation.

- **E-Reader Functionality:** Provide the core features from an e-reader (see Section 2.2.2)
- **Transformation Capability:** Enable readers to apply and customize predefined text transformations along various text accessibility dimensions within the reading interface
- **Custom Transformation Creation:** Allow users to define transformations through natural language interaction via chat interface.

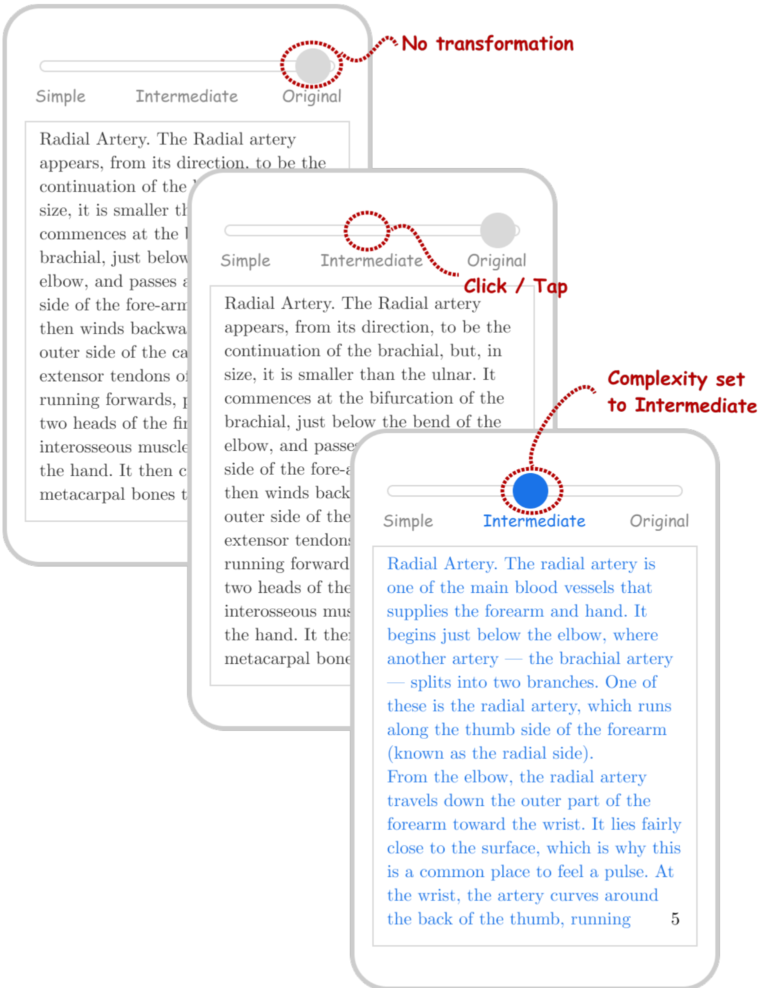

Figure 4.1: Exemplary illustration of user interaction with the core mechanism of MagicTextreader: The figure shows three interface layers from background to foreground, demonstrating how a user adjusts the complexity of a medically dense text. The background displays an e-book reader with the original excerpt from Gray's Anatomy [Gray et al., 1901], describing the radial artery. A complexity slider at the top offers the options 'Simple', 'Intermediate', and 'Original'. The user selects 'Intermediate' (middle layer), resulting in a transformed version of the text shown in the foreground. The rewritten text is generated using GPT-4o [OpenAI, 2024] with a zero-shot prompt, tailored to suit an average adult reader.

- **Acceptable User Interface:** Ensure the interface design does not introduce accessibility barriers that would confound investigation of transformation-related accessibility impacts.
- **Research Support:** Support data collection and investigation of the prototype in use.

4.2.3 Scope and Boundaries

The following defines the scope and boundaries of the prototype and research toolkit.

Supported Content The system processes e-books in EPUB format. EPUB was chosen because it is an open standard supported by all major platforms, most e-books are available in this format, and it already addresses accessibility needs (see Section 2.2.3 and 2.2.2). Although not exclusively limited to non-fiction, the reader is designed with processing non-fiction books in mind, as the succeeding study and investigation of text accessibility is restricted to non-fiction.

Limited to EPUB format

Reader Accessibility Features Standard visual accessibility features such as page reflow, text size and font adjustments, and appearance settings (e.g., dark and light mode) are supported. In contrast, highlighting, hyperlinks, and the table of contents were intentionally left out. This is based on the transformation architecture (Section 4.4.4) and the quality requirements for performance (QR1), consistency (QR2), and resource efficiency (QR4). Highlighting was omitted, as participants of the succeeding study will not access the text again after reading. Hyperlinks were excluded because navigation could lead to sections not yet transformed. A table of contents was not included, as the succeeding study provides only short, self-contained excerpts.

Supports visual aids and omits navigation, highlighting, hyperlinks

Modalities The system focuses on text-to-text transformations without extending to multimedia conversions.

Supports only text-to-text

Interaction Models The prototype explores multiple in-

Supports adjustable and user-defined text transformations teraction paradigms for transformations: pre-defined text transformations that can be adjusted as well as the creation of transformations using natural language instruction. The customization of transformations is offered since readers have different needs, personalization has been proposed as a potential factor influencing accessibility, with the view that users should be involved in defining what works best for them (see Section 3.6.3).

Uses external LLMs via prompt engineering for transformation

External LLM Dependency The system leverages external LLMs for text processing, without attempting to develop or refine language models. It relies on prompt engineering to guide transformations — an approach consistent with methods adopted in related work (Section 3.6) and chosen to prioritize the interaction and transformation experience over model development. As a result, an active internet connection is required.

Research support tools are included

Research Support Besides the prototype, research tools are provided including data collection mechanisms, survey tools, and usage control for a first evaluation and a foundation for a research toolkit.

Designed for digitally literate users

Target Audience The system targets users with basic digital literacy who are familiar with common digital interfaces such as web browsers, mobile applications, and standard interaction patterns. Language support is limited to English and German.

4.3 Requirements

Goal-based requirements in accordance with rapid prototyping

To support the development of the prototype, the requirements were defined in a goal-oriented and deductive manner, rather than through iterative refinement. This approach is based on Goal-Oriented Requirements Engineering, which emphasizes deriving system behavior from defined goals [Van Lamsweerde, 2001]. It is also aligned with the approach described in Constructive Design Research, where prototypes are used to explore design ideas and inform later investigations [Koskinen et al., 2013]. Additionally, this approach draws inspiration from rapid prototyping methodologies, where the prototype itself can help to refine or reveal requirements through actual use [Hartson and Smith, 1991]. The grouping into functional and quality requirements was informed by common concerns in HCI and software quality models such as ISO/IEC 25010 [International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), 2023].

4.3.1 Functional Requirements (FR)

Functional requirements define what the system must do to fulfill its purpose, including core capabilities, features, and behaviors.

1. **Digital Reading** — The system must allow users to read digital books with basic navigation features and provide common accessibility features prevalent in typical e-book readers and relevant for the succeeding study.

2. **Text Transformation** — The system must provide text transformation capabilities that modify text content while preserving original meaning.

3. **Pre-defined & Custom Transformations** — The system must offer pre-defined transformations and enable users to create and update custom transformations.

4. **Transformation Status Feedback** — The system must provide feedback on transformation status and progress.

5. **Transformation Controls** — The system must provide controls appropriate to different types of transformation.

6. **Research Support** — The system must support research activities by collecting usage data, facilitating user surveys, and enabling analysis of reader behavior and transformation effects.

7. **Multilingual Support** — The system must support multiple languages for interface, content, and transformations.

8. **Authentication** — The system must provide authentication for participants and administrators.

9. **Administration** — The system must provide administrative capabilities for managing system components and research data.

4.3.2 Quality Requirements (QR)

Quality requirements define how well the system must perform its functions, including performance, usability, reliability, and other quality attributes.

1. **Performance** — The system must transform text with minimal perceived latency, allowing users to apply and discard transformations with near real-time feedback.

2. **Consistency** — The system must transform text in a predictable and uniform manner across all parts of a document. Transformations should maintain coherence between different segments of text, preserve the logical flow of content, and produce identical results when the same settings are applied. This ensures a seamless reading experience both within a single session and across different sessions.

3. **Usability** — The interface must be intuitive for applying, adjusting, and managing transformations without extensive training.

4. **Resource Efficiency** — The system must be efficient in its use of external API resources.

5. **Cross-platform Compatibility** — The reader must function across different devices and browsers with consistent behavior.

6. **Coherence** — Text transformations must preserve the original meaning and intent of the content while applying the requested changes.

7. **Reading Flow** — The system must maintain reading flow when applying transformations with minimal disruption.

8. **Data Protection** — The system must protect user data and provide appropriate access controls for the research context.

4.4 Design Decisions

This section outlines the key design decisions that shaped the implementation of the *MagicTextreader*. Each decision addresses specific requirements and includes a rationale explaining the reasoning behind the choice.

4.4.1 System Architecture

The fundamental design decision was to implement a dual-purpose system that serves both as a functional prototype of an AI-enhanced reader and as a research toolkit. This directly addresses the core requirements of creating a working digital reader (FR1) and supporting research activities (FR6).

Architecture consists of reader and research platform

The MagicTextreader prototype thus consists of two integrated parts: first the MagicTextreader itself, which provides the interactive reading and transformation functionality, and second the surrounding platform, which supports evaluation in the succeeding user study.

Web-based client-server design

Both parts are implemented as a web-based system using a client-server architecture. This approach was selected to leverage epub.js (Sec. 4.4.2), which provides EPUB rendering capabilities through web technologies, to ensure cross-platform compatibility (QR5) across devices and take advantage of the well-suited nature of web technologies for processing XML-based content [Zisman, 2000] (like EPUB).

Client handles EPUB rendering, provides controls, and surveys

The client (frontend) runs in the user's browser. It renders EPUB content (Sec. 4.4.2), provides transformation controls (Sec. 4.4.8), supports custom transformations (Sec. 4.4.10), displays questionnaires (Sec. 4.4.13), and presents a book library (FR1).

Server manages data, OpenAI API, and study administration

The server (backend) handles data persistence and processing. It stores e-books and transformation definitions, manages participants and research data (Sec. 4.4.15 and 4.4.16), logs interactions (Sec. 4.4.12)), and communicates with the OpenAI API (Sec. 4.4.5). This separation of concerns centralizes data management and simplifies study administration while maintaining responsive performance (QR1).

4.4.2 EPUB Reader

epub.js enables EPUB reading in web environment

To support reading e-books in the EPUB format (scope limitation) epub.js [FuturePress, 2025] was selected. Specifically, the system employs React Reader [Sletten, 2017], a React-based wrapper around epub.js, which provides reading functionality within a web environment.

Requirements Addressed: FR1, QR6

Chosen for open-source, lightweight, and customizability

Rationale: Several open-source EPUB reader toolkits were evaluated for their suitability in rendering and customizing EPUB content. These include Readium (JavaScript for web, Swift for iOS, Kotlin for Android) [Foundation, 2025], FolioReaderKit (Swift for iOS) [Team, 2025], EpubReader (.NET for Windows) [Team, 2025], epub.js (JavaScript for web), Thorium Reader (Electron for Windows, macOS, Linux) [EDRLab, 2025], Bibi (JavaScript for web) [Matsushima, 2025], Calibre (Python for Windows, macOS, Linux, and web-based) [Goyal, 2025], and Bookworm (Vala for Linux) [Das, 2025]. Although all considered tools support EPUB 2 and 3 formats and common accessibility features, epub.js was selected for the prototype due to its lightweight structure, direct access to the DOM, and suitable interfaces for implementing transformation features. Its web-based design also facilitates cross-platform compatibility. In addition, epub.js is well-documented and widely used, with community engagement reflected in metrics such as over 6,700 GitHub stars, 204 watchers, and more than 1,100 forks.

4.4.3 Content Preprocessing System

Books are preprocessed and divided into XML-comment separated segments, each approximately 2000 characters in length, to support granular transformation and efficient processing.

Books are pre-segmented for transformation<

Requirements Addressed: FR1, QR1

Rationale: A fixed, preprocessing-based strategy was chosen instead of dynamic segmentation based on user navigation for several key reasons. The text is parsed once hierarchically, with segment boundaries aligning with sentence endings and XML tags. This creates coherent chunks and eliminates the need for repeated parsing during book navigation. Using consistent numbered segment identifiers makes referencing, caching, and logging more straightforward. Fixed segmentation also generally reduces complexity by avoiding the need for a dynamic segmentation strategy. XML comment delimiters were selected for their non-intrusive integration with the content structure.

Fixed preprocessing segments text for coherence and efficiency

The chunk size of 2000 characters was chosen based on an estimate of typical e-book page content. At first, reference material from an Amazon Kindle Paperwhite advertisement, showing a page that contains approximately 850 characters, was used as an initial orientation [Amazon, 2024]. The chunk size was increased to accommodate variations in layout and font size, but remained within reasonable limits, as smaller chunks can be processed more efficiently by large language models (LLMs).

Chunk size set for typical e-book page and LLM efficiency

4.4.4 Transformation Architecture

Segment-based architecture processes content with context awareness

The system employs a segment-based transformation architecture that processes content with context awareness from neighboring segments, using a sequential approach that prioritizes the reader's current position. As introduced in Section 4.4.3, books are preprocessed and divided into fixed segments that can be used for transformation.

Each time a page is opened or turned while transformations are active, the architecture executes the following steps:

1. **Determine Current Segment:** Identify the lowest-numbered segment that contributes content to the visible page. This becomes the *current segment*.
2. **Mark Surrounding Segments for Transformation:** Mark the current segment, its immediate left neighbor, and its immediate right neighbor for transformation—only if they have not already been transformed.
3. **Find Reference Segments:** For each segment marked for transformation, check its left and right neighbors. If a neighbor is not also marked and has already been transformed, include it as a reference segment.
4. **Transform Segments:** Transform all marked segments using the reference segments for context. The transformed versions replace the original content on the page.
5. **Mark Upcoming Segments for Buffering:** Mark the next two segments after the current one for buffered transformation—again, only if they have not yet been transformed.
6. **Find Reference Segments** For each buffered segment, check its neighbors and include already-transformed ones (not also marked) as reference segments.
7. **Transform Buffered Segments:** Transform the buffered segments in the background using their reference context. Store them for seamless forward navigation.

Requirements Addressed: FR2, QR1, QR2, QR6, QR7

Rationale:

The architecture centers transformation around the user's current reading position to ensure relevance and responsiveness (**QR1**, **QR2**, **QR7**). Including neighboring segments in the transformation context preserves local coherence and reduces textual discontinuity (**FR2**, **QR6**, **QR7**). The decision to transform the current segment along with its immediate left and right neighbors ensures that the user's current page is fully transformed, and that a likely next page (when navigating forward) is already available without delay, as in most conditions a single segment exceeds the user's typical page size. Reference segments ensure smooth transitions between previously and newly transformed segments (**QR2**, **QR6**). The buffering strategy preloads likely future segments to reduce even further latency during forward navigation (**QR1**, **QR7**).

Architecture focuses on current and neighboring segments for coherence and responsiveness

4.4.5 LLM Integration Strategy

The system integrates large language models from OpenAI via their API service. These models are used for executing text transformations as well as for enabling natural language interaction to create new transformations. The system uses the `gpt-4o-mini` model for transformation tasks and the `gpt-4o` model for chatbot-based interaction.

OpenAI LLMs for transformations and interaction

Requirements Addressed: FR2, QR1, QR4, QR8

Rationale: LLMs from several providers were considered, including those listed in chapter 3. Model rankings from the Chatbot Arena benchmark [Chiang et al., 2024] were used to compare performance. OpenAI is selected based on the following considerations:

Chosen for performance, security, and integration features

- Different general-purpose models are available which vary in size, latency, fine-tuning and cost, including several ranked among the top 20.

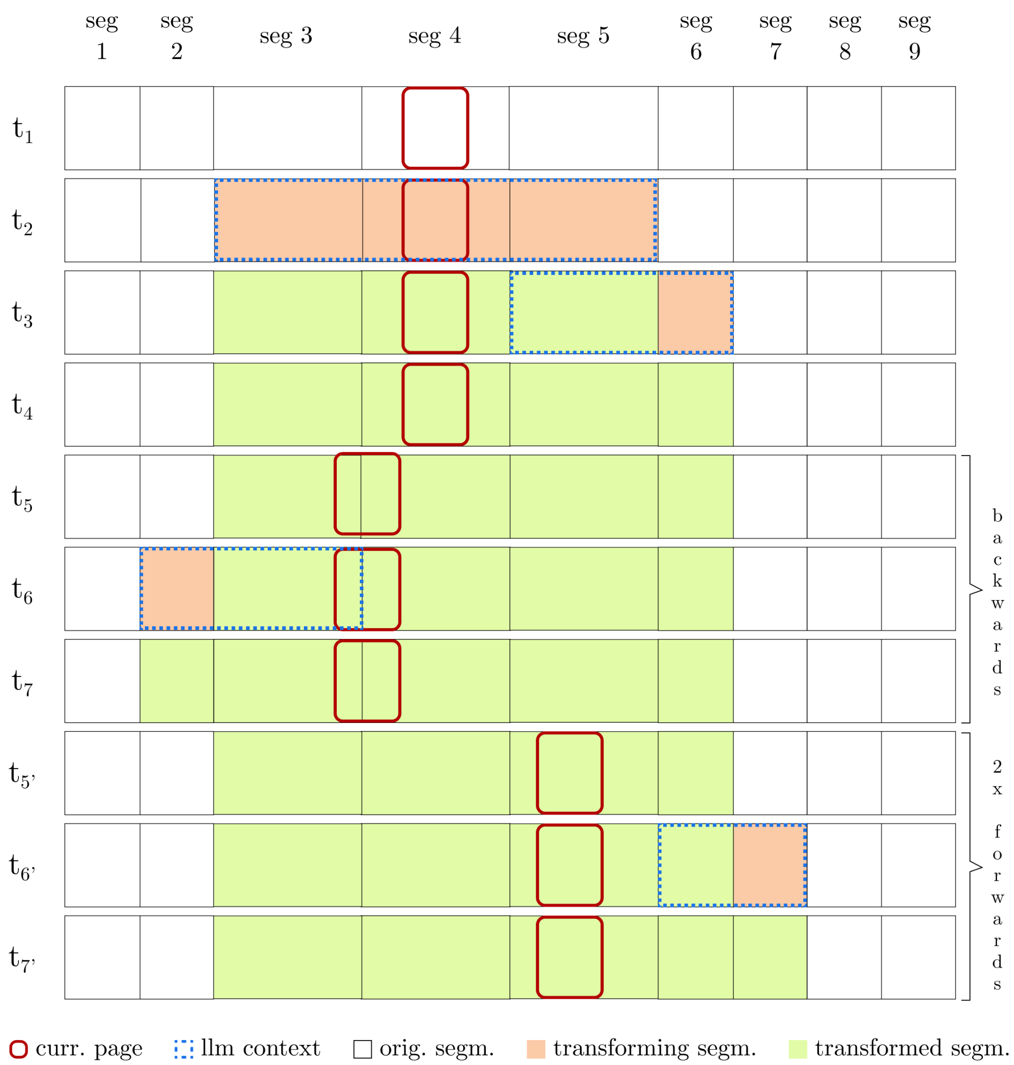

Figure 4.2: Visualization of the segment-based transformation architecture across time steps. The x-axis represents the first segments of some book, while the y-axis shows a timeline corresponding to user interactions. Between t1 and t2, the user activates a transformation after navigating to one of the first pages. The figure further illustrates how the system responds to backward navigation (t5-t7) and forward navigation (t5′–t7′) following the initial transformation process.

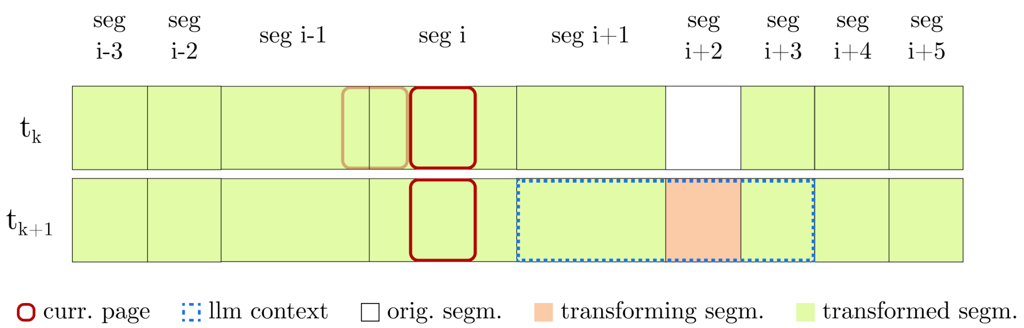

Figure 4.3: Visualization of the segment-based transformation architecture, showing a scenario where the buffering mechanism marks segment i+2 for transformation after the user navigates into segment i. It is illustrated how both neighboring segments, already transformed, are used as context. Such a situation may arise when the user navigates between segment chains while no transformations were active in between.

- Access provided by encrypted API, which removes the need for local model deployment, simplifies integration and does not store prompts or generated content after processing (QR8)
- A comprehensive prompt engineering framework is available with features like the Assistants API [OpenAI, 2024].
- Exact usage records are provided, enabling a cost and usage monitoring system (see Sec. 4.4.14) (QR4)

OpenAI's general-purpose models are preferred over those fine-tuned for logical reasoning or code generation, as the system focuses on text transformation and natural language interaction. To determine the most suitable models for each use case, balance is struck between response time and the models' accuracy and capabilities.

At the time of integration (December 2024), `gpt-4o` is OpenAI's highest-ranked general-purpose model based on the benchmark. Due to the importance of response time in interactive systems, smaller models such as `gpt-4o-mini` and `gpt-3.5-turbo` were also evaluated.

Evaluated response time for decision support

A small response time test was conducted to assess model suitability. In this test, models were instructed to replace every second word in a given text with the token *blabla*. Two task variants were tested:

- **3-to-3 Task:** The model receives three input segments (each approximately 2000 characters) and transforms all three. This simulates the standard transformation step (Section 4.4.4).
- **2-to-1 Task:** The model receives two segments but transforms only the first. The second segment is provided as context. This simulates a buffer transformation step.

The segments were sampled from Gray's Anatomy [Gray et al., 1901]. Details on implementation and prompt formulation are provided in the Appendix A. The results are shown in figure 4.4.

gpt-3.5-turbo fast but unreliable

`gpt-3.5-turbo` produces the fastest responses but fails to apply the transformation correctly in two out of eight test runs. Due to limited reliability, this model is not used.

gpt-4o-mini chosen for accurate, low-latency transforming

`gpt-4o` and `gpt-4o-mini` completed all test cases correctly. Based on lower latency, `gpt-4o-mini` is selected for text transformation tasks (QR1). Measured response times are approximately 30 seconds for transforming three segments and 12 seconds for one segment with one additional segment as context. When switching transformations on, usually only two segments need to be transformed before the transformed version can be made visible to the user. Thus, the average response time after enabling transformations can be estimated at approximately 20 seconds, as response time behaves roughly linearly with the length of the generated text (see Section 3.4.4).

gpt-4o used for chatbot interaction with higher quality

`gpt-4o` is used for chatbot interaction, where smaller output lengths is expected and generation quality can be preferred.

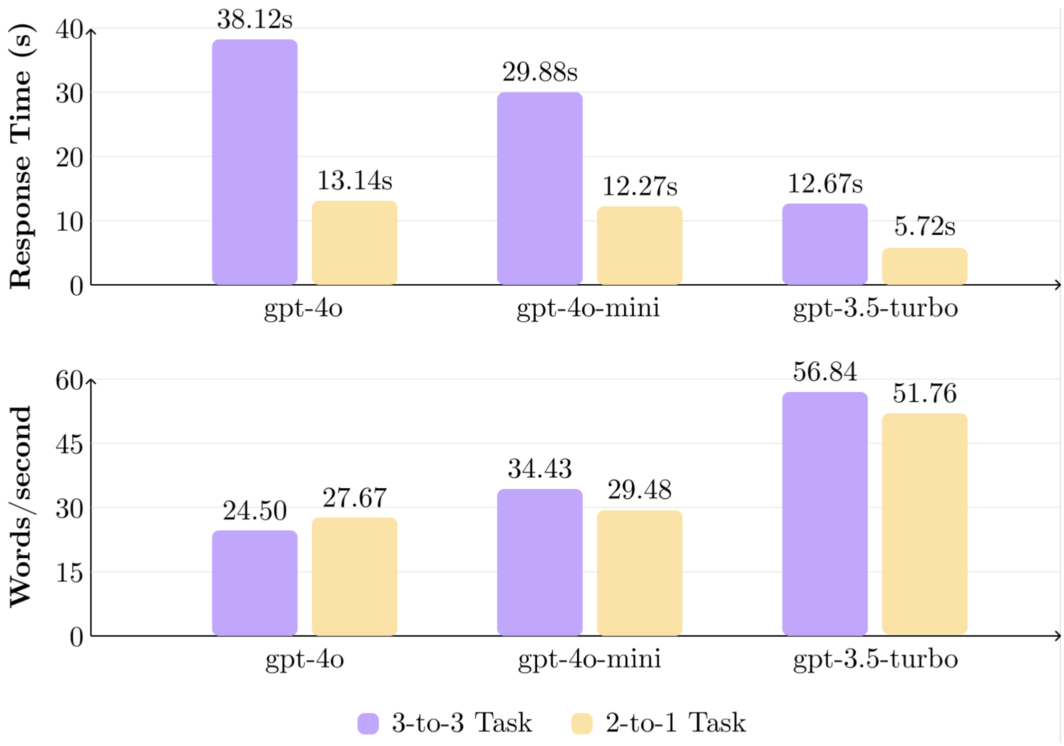

Figure 4.4: Comparison of the performance of the GPT-4o and GPT-4o-mini [OpenAI, 2024] and GPT-3.5 [OpenAI, 2023] models on the "Every 2nd blabla" text transformation task. In this task, the model is instructed to replace every second word in the input text with the token *blabla*. Two task variants are evaluated. In the 3-to-3 task, the model receives three input segments (each approximately 2000 characters in length) and is required to transform all three. In the 2-to-1 task, the model receives two input segments, but only the first segment is to be transformed; the second segment is provided as context only and is not included in the output. The first bar chart shows the average response time (N = 4) of each model for both task variants, while the second bar chart presents the corresponding processing speed in words per second.

4.4.6 Prompt Engineering Framework

The system implements few-shot prompting with examples for text transformations, and zero-shot prompting for natural language interactions related to the creation of text transformations. No model training or fine-tuning is performed to optimize the models for either the transformations or the natural language interaction.

OpenAI completions API enables structured prompt-based transformations

For text transformations, the system uses OpenAI's completions API [OpenAI, 2024]. The API allows to structure the prompt as message list consisting of a system message and a user message. The system message defines the transformation task and includes: the transformation objective, constraints or warnings, input format explanation, task description, and example input-output pairs to support few-shot learning of the output format. It concludes with a list of transformation preferences that specify how the model should transform the input segments. The user message contains a list of xml-separated segments to be transformed or used as context. Figure 4.5 illustrates this structure.

OpenAI's assistant API allows to define functions ensuring LLM output follows schema

For natural language transformation creation, the system uses OpenAI's assistants API [OpenAI, 2024] to formulate a prompt that includes two components: instructions and a function definition. The instruction block defines the assistant's role, outlines supported transformation types, and specifies the goal of assisting the user in formulating a new transformation. The function block defines `createTransformation()`, a referenceable function that returns a structured transformation definition aligned with the system's internal schema. This allows the model to produce actionable transformation specifications. Figure 4.6 shows the static prompt structure; the dynamically appended message history from user interaction, as well as the initial system message prompting the user to define a transformation, are omitted from the figure.

Requirements Addressed: FR2, FR3, QR6

Prompt engineering selected over fine-tuning given scope constraints and proven effectiveness

Rationale: Prompt engineering is applied instead of fine-tuning, as training models lies outside the system's scope and related work outlines similar tasks where only prompt-based methods were used successfully (QR6). Few-shot prompting is used for segment transformation to ensure well-formatted output, which is required for further processing. Zero-shot prompting is used for natural language interaction, where strict format adherence is not necessary and otherwise correct output is ensured through the function reference and definition. The prompt structure follows techniques from Section 3.4.2, especially following provider guidelines, such as using markup formats, clearly

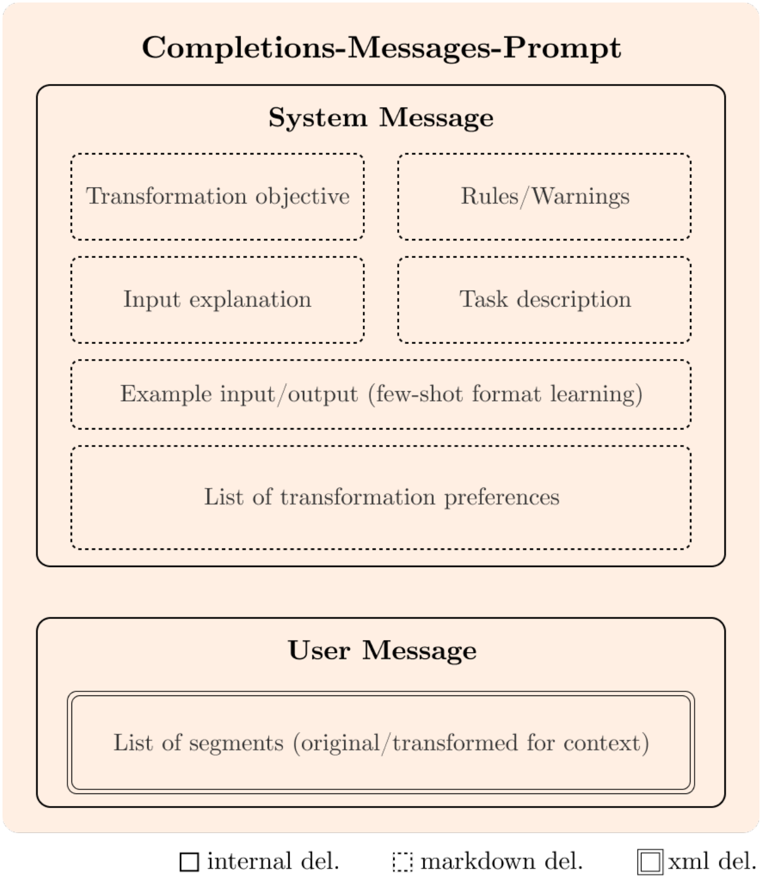

Figure 4.5: Illustration of the prompt structure used by MagicTextreader to transform e-book segments.

defined task goals and was further optimized using the model itself.

4.4.7 Transformation Cache System

Transformed segments are cached locally, keyed by the book, section, and transformation configuration. Accordingly, when a user requests a transformation of a segment for a specific book and section that has already been processed with the same configuration, the system retrieves the cached segment instead of reprocessing it.

Caching system stores transformed segments locally

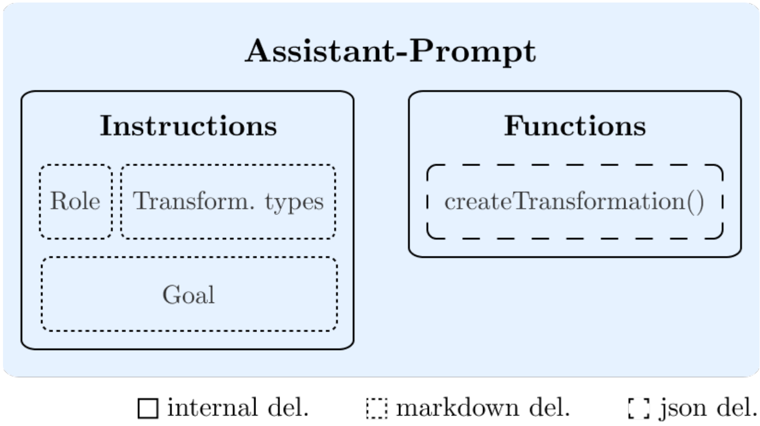

Figure 4.6: Illustration of the prompt structure used to enable natural language interaction for creating new text transformations in MagicTextreader.

Requirements Addressed: QR1, QR2, QR4, QR8

Caching addresses LLM non-determinism while improving consistency, speed, and cost-efficiency

Rationale: The cache system addresses the non-deterministic nature of LLM outputs (see Section 3.4.5) by caching transformed segments. This ensures that users receive consistent results when revisiting previously transformed content (QR2). By retrieving already-processed segments instead of reprocessing them, the system reduces response time in these cases (QR1) and lowers the number of API calls made to the LLM provider, thereby decreasing costs (QR4). While a shared backend-based caching mechanism could reduce processing effort even further—by enabling reuse of identical transformations across users—this is not implemented to keep the architecture simple for the research prototype. Moreover, most e-readers operate as local applications.

4.4.8 Modular Transformations

Modular transformation system enables flexible configuration and combination of components

The system supports the definition of text transformations in a modular fashion, meaning they are independent, user- or system-defined components that can be individually

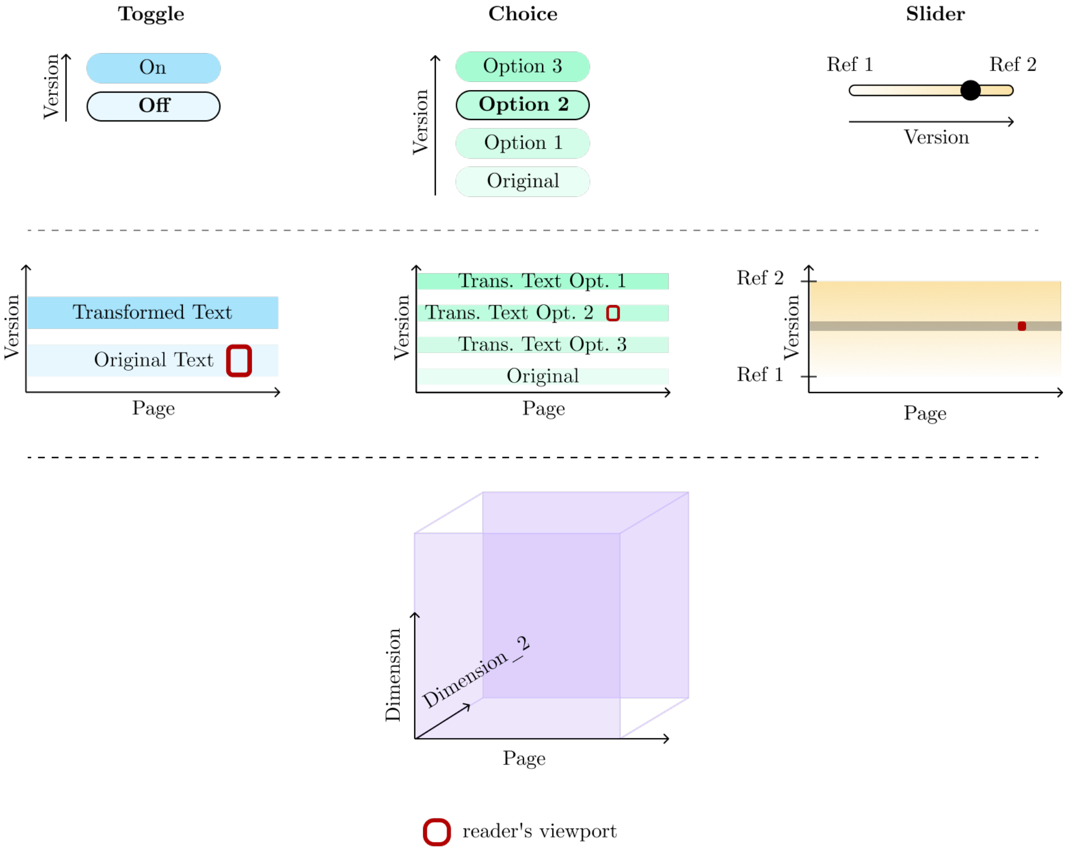

Figure 4.7: Illustration of three transformation control types mapped to a conceptual space (inspired by Card et al. [1991]). In this space, the pages of a book are arranged as a horizontal text stream along the x-axis, while different versions of each page—representing variations along a single text accessibility dimension—are mapped vertically along the y-axis. (1) A *Toggle Control* maps its two states (e.g., on/off) to two vertically stacked text streams; switching the toggle selects one stream. (2) A *Choice Control* maps each selectable option along the control's y-axis to a corresponding text stream, also arranged vertically. (3) A *Slider Control* maps its continuous position along the x-axis to intermediate versions of a text, spanning between two reference versions arranged from bottom to top along the y-axis. A 3D representation at the bottom illustrates how multiple text accessibility dimensions can be combined in a multi-dimensional space: the x-axis indicates page position, while the y- and z-axes each represent one text accessibility dimension. Each coordinate represents a unique combination of transformations. This concept extends to n-dimensions as more transformations are applied in combination.

configured, selectively shown or hidden, and applied either separately or in combination.

It distinguishes between three types of transformations that differ in how options are structured and how users interact with them.

Toggle transformations provide binary on/off functionality for single isolated changes

Toggle transformations operate along a binary text accessibility dimension and can either be activated or deactivated. Each represents a single, isolated transformation whose presence or absence defines the complete range of states. Conceptually, this type corresponds to a nominal variable with two categories. An example is a translation function that toggles the text between its original language and German.

Choice transformations offer selection among multiple distinct unordered options

Choice transformations operate along a multi-valued text accessibility dimension and offer a selection among multiple distinct options, which may represent different styles, categories, or formats. The options are logically grouped but do not follow a sequential order. Intermediate values between them are not meaningful, and the transformation can be associated with a nominal scale. An example is a text-style transformation where users can choose between journalistic, neutral, or scientific formulations.

Slider transformations enable continuous adjustment along ordered reference points

Slider transformations operate along a continuous text accessibility dimension and allow for adjustment along a continuous or ordered scale. The options are sequentially related, and intermediate values carry semantic meaning. This type of transformation is associated with ordinal or interval-level variables. An example is a complexity transformation where the text can be adapted progressively from simple to complex language.

Transformations defined by names and natural language instructions with interpolation for sliders

Each transformation is defined by a name and a description of how the transformation is applied. For toggle transformations, this description is a single natural language instruction associated with the active state. For choice transformations, each selectable option is defined by a name and a corresponding instruction that specifies the transformation when that option is selected. Slider transformations are defined by a name and a set of ordered reference options, each of which includes a name and a natural language instruction. Instructions for intermediate states are constructed by including the instructions of the two nearest reference options in the prompt, each annotated with a percentage weight indicating its contribution, with the weights summing to 100% based on the slider's position.

When multiple transformations are applied, the prompt instructs the language model to satisfy all associated instructions simultaneously. This does not significantly influence the response time, as it mainly depends on the output length (see Section 3.4.4), and the combined instructions only increase the input length minimally.

Requirements Addressed: FR2, FR3,FR5, FR6, QR3

Rationale: A modular transformation system is chosen to enable reusability and support both predefined and user-defined transformations through a unified mechanism (FR2, FR3). It also ensures adaptability for the research study (QR3, FR6).

Three transformation control types address different text accessibility dimensions based on their properties

The three transformation control types are selected to reflect different types of text accessibility dimensions based on their properties and the text characteristics discussed in Sections 2 and 3 (FR5). Choice transformations are used for multi-valued dimensions such as style or tone adjustments, where distinct options can be grouped under a nominal scale. Toggle transformations operate on binary dimensions with two discrete states, also corresponding to a nominal scale. Slider transformations support continuous dimensions such as complexity adjustment, representing transformations along an ordinal or interval scale.

Control types follow natural mapping principles

The control types are mapped according to principles of natural mapping [Norman, 2013] and incorporate attributes of direct manipulation interfaces [Shneiderman et al., 2016]. The conceptual structure is inspired by the design space model from Card et al. [1991], as illustrated in Figure 4.7 (QR3).

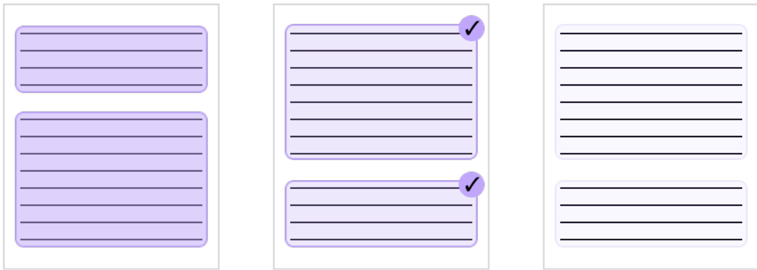

Figure 4.8: Illustration of the visual feedback system: The figure shows three instances of the same e-book page, each overlaid with visual indicators in a different state—during transformation (loading), immediately after the transformation is applied (ready), and during the fading of the overlay (fading). It also highlights how paragraph size and layout can visibly change as a result of text size differences.

4.4.9 Visual Feedback System

Visual feedback system indicates transformation status through paragraph overlay state

To provide immediate feedback during transformation processes, the reader interface overlays each paragraph with a semi-transparent box that visually indicates the current state. Three states are distinguished: a loading state while a transformation is in progress, a ready state immediately after the transformation is applied, and a fading state where the overlay gradually disappears. These cues help users understand the system's status and observe changes in paragraph layout caused by text size differences.

Requirements Addressed: FR4, QR4, QR3, QR7

Prevents interface confusion during processing and highlights layout changes

Rationale: The visual feedback system provides immediate UI feedback while transformations are in progress (FR4, QR4). As the interface is locked during processing, these visual indicators help users understand that the system is working. Additionally, the overlays remain briefly after transformations complete and fade out gradually, helping users track how paragraph sizes and layout may have changed due to text size differences (QR3, QR7).

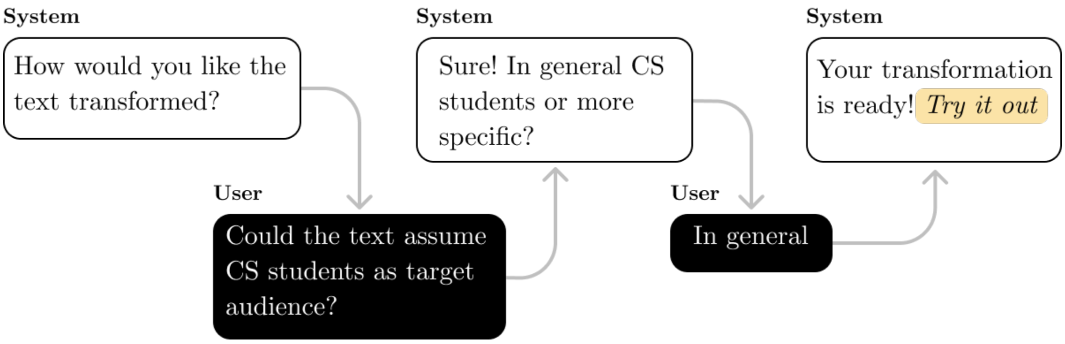

Figure 4.9: An illustration of an exemplary user interaction with the chat-based interface to create and modify transformations.

4.4.10 Custom Transformation Interface

A chat-based interface enables users to create and modify transformations using natural language. The interaction is guided and includes immediate testing capabilities. In an initial message, users are asked how they would like the text to be transformed. After a brief exchange to clarify the requirements, the system generates an appropriate transformation. The user can then adjust and apply this transformation. Figure 4.9 shows an exemplary user interaction with the interface.

Chat interface allows users to create custom transformations through natural language

Requirements Addressed: FR3, QR3, QR6

Rationale: Natural language interaction is used to specify custom transformations, as it allows users to express their intentions in their own language and offers several potential advantages, such as enabling the system to ask contextual follow-up questions (see Section 3.7) (QR3). The resulting descriptions of the desired transformation are particularly well suited for use with LLMs, as these are also controlled through prompts in natural language. However, the system does not let users write transformation prompts directly; instead, it lets the LLM generate them based on the information gathered during the interaction. This follows findings from related work showing that LLMs can engineer their own prompts and ensures that the generated prompts follow predefined, tested formats (QR6). Further, transformations are not applied directly but are instead in-

Natural language enables intuitive transformation specification while LLM generates structured prompts tegrated into the modular transformation system to allow users to adjust parameters later via the control types defined in Section 4.4.8, with one type automatically assigned (in consultation with the user) to the created transformation.

4.4.11 Multilingual Support

System supports internationalization with English and German localizations provided

The system is designed for internationalization and supports multiple languages across all components, including the interface, content, chat, and text transformations. Localizations are provided for English and German.

Requirements Addressed: FR7

Multi-language support enables broader accessibility testing across linguistic contexts

Rationale: Multi-language support broadens the potential user base and ensures that participation in the subsequent study is not limited to speakers of a single language and enables investigation of the system's impact on text accessibility in different linguistic contexts (FR7). English and German were selected for localization, as it is expected that participants in the subsequent study will be fluent in either or both of these languages.

4.4.12 Data Collection Framework

Event tracking framework captures user interactions

The system records user interactions using an event tracking framework. Tracked events include book navigation, transformation usage, and chat interactions. Each event includes the following components:

- **type:** Specifies the event category (e.g., navigation, transformation start, transformation complete).
- **participant, book, transformation:** Identify the user involved, the book being read, and the transformation applied.
- **timestamp:** Records the time at which the event occurred.

- **user properties, event properties:** Store additional contextual information, such as user characteristics or details specific to the event.

Requirements Addressed: FR6

Rationale: The event data supports the analysis conducted in the subsequent study (FR6). As discussed in Section 3.8.2, event tracking complements traditional observational methods in ethnographic research. The selected attributes follow conventions from process mining literature (see Section 3.8.2): identifiers such as participant and book group events into cases, the event type provides an activity label, and the timestamp determines event order and timing.

Event data structured following process mining conventions supports ethnographic analysis<

4.4.13 Survey System

The system integrates configurable pre- and post-interaction questionnaires. These are accessible within the platform and managed through the administrative interface. Surveys support open-ended questions via free text input and closed-ended questions using single-choice, multiple-choice (with optional randomization), and scale-based inputs with arbitrary labels.

Configurable questionnaires support various question types for user research

Requirements Addressed: FR6

Rationale: Surveys are integrated into the platform to support the research study without requiring a context switch to external tools (FR6). The supported question types align with those described in Section 3.8.1, enabling both qualitative and quantitative data collection. Randomization of response options is included to mitigate response order bias (see Section 3.8.1).

Integrated surveys eliminate context switching while preventing response order bias

4.4.14 Resource Management System

The system includes a usage monitoring and limit mech-

Multi-level cost monitoring included

anism to control API-related costs at the participant, transformation, and global levels. Each call to the OpenAI API is logged along with the associated cost and a reference to the participant, transformation, or natural language interaction that triggered the call. Each user is assigned an individual cost limit, and the system maintains an overall global cost limit. If either limit is reached, the system disables further API calls, thereby enforcing cost constraints.

Requirements Addressed: QR4

API usage limits prevent excessive costs and ensure fair resource allocation

Rationale: The system limits API usage to prevent excessive costs resulting from unintended behavior, such as repeated calls caused by system errors or loops, which is particularly relevant in a prototype context. It also ensures that no single participant can consume a disproportionate share of the available API budget, preserving resources for other users/participants (QR4).

4.4.15 Authentication System

A token-based authentication system controls participant access, while session-based authentication is used for administrative access related to research management.

Requirements Addressed: FR8, FR9, QR4, QR8

Authentication restricts access due to API costs and prevents interference

Rationale: Authentication is required to restrict participant access, as the number of users must be limited due to API usage costs (QR4, QR8). Administrative access is separated to allow researchers to configure the study, define transformations, manage surveys, and review collected data without unauthorized interference (FR9, QR8). Token-based access provides a simple mechanism for participants, while session-based authentication ensures continuous access for administrative tasks.

4.4.16 Administration Framework

Administrative interface supports research study configuration and setup

An administrative interface allows researchers to manage all system components, including transformations, books, participants, and collected data. Through the interface, predefined transformations can be created and modified, books can be uploaded and described, participant access can be managed through token-based invitations, and questionnaires can be configured and reviewed. The interface also provides access to event data and allows definition and monitoring of usage limits.

Requirements Addressed: FR1, FR3, FR6, FR8, FR9, QR4, QR8

Rationale: The administration interface is required to define the book library used in the study (FR1), provide a set of predefined transformations (FR3), manage participant access (FR8), and configure and review questionnaire and event data (FR6). It also enables the definition and enforcement of usage limits to control API-related costs (QR4) and ensures that study-related data can be accessed and modified only through a protected interface (FR9, QR8).

Fulfills core functional requirements for study management

4.5 Implementation

This section details the concrete realization of the design decisions outlined in Section 4.4. While the System Overview (Section 4.2) and Design Decisions established the architectural foundation and rationale for the MagicTextreader, the implementation translates these conceptual choices into software components. The focus lies on critical implementation aspects not covered in the design phase: technology integration strategies, core algorithms, prompt assembly, and testing strategies.

4.5.1 Technology Stack

The system's architecture design decision (see Section 4.4.1) established a web-based client-server architecture, which is consequently reflected in the technology stack. In addition to prior design decisions dictating certain technologies, the selected technologies were chosen to support either rapid prototyping, fulfill specific requirements, or comply with common development standards.

- **Frontend:** React, TypeScript, Next.js, epub.js, Tailwind CSS, shadcn/ui
- **Backend:** Django, Django REST Framework, OpenAI Python API
- **Database:** SQLite, Web Storage API

React-based frontend stack enables component architecture and rapid UI development

Frontend React [Meta Platforms, Inc., 2024] was selected for its component-based structure and efficient state handling and popularity [Zammetti, 2020], suitable for integrating epub.js, which provides direct DOM access needed for overlay rendering (Section 4.4.9). TypeScript [Microsoft Corporation, 2025] adds static type checking which might improve code quality [Bogner and Merkel, 2022]. Next.js [Vercel, 2016] was chosen for its file-based routing, Tailwind CSS [Tailwind Labs, 2025] and shadcn/ui [shadcn, 2025] provide reusable components and styling for rapid UI development ([Gerchev, 2022]), including support for the interface controls in Section 4.4.8 and chat elements (Section 4.4.10).

Django chosen for rapid backend development thanks to built-in features and package infrastructure

Backend Python Django [Foundation, 2025] was chosen for its rapid development capabilities [Ghimire, 2020] and built-in support for different authentication methods (Section 4.4.15) and administration (Section 4.4.16). Django REST Framework enables REST-compliant API development[Hawke, 2011; Rodríguez et al., 2016]. Integration with OpenAI's models is achieved using the official Python library, as required by Section 4.4.5. In general, Python as programming language offers a development environment that is well suited for AI tasks due to its versatility and extensive library support [Türkmen et al., 2024].

Database SQLite [Hipp, 2025] is used for prototyping due to its simple setup and lack of administrative overhead [Consortium, 2025].

4.5.2 Components

Figure 4.10 illustrates the system architecture, including frontend views, backend services, persistent models, and external integrations. This section first outlines the structural composition, followed by a walkthrough of component interactions during a typical user session.

Frontend Structure The client application includes five top-level components:

- **Consent Screen**: Validates invitation codes and collects participant consent.
- **Pre-Survey**: Loads and submits a pre-interaction questionnaire.
- **Book Library**: Displays available EPUB books.
- **Reader**, (see Section 4.5.6 and Figure 4.11 for an illustration) subdivided into:
  - **EPUB-Reader**: Renders EPUB content and manages navigation.
  - **Toolbar**: Displays transformation controls (toggle, choice, slider), accessibility settings, and a loader.
  - **Overlay**: Visual indicator during transformations (see Section 4.5.7).
  - **Chat**: Supports creation of custom transformations via natural language (see Section 4.5.9).
- **Post-Survey**: A follow-up questionnaire shown after reading.

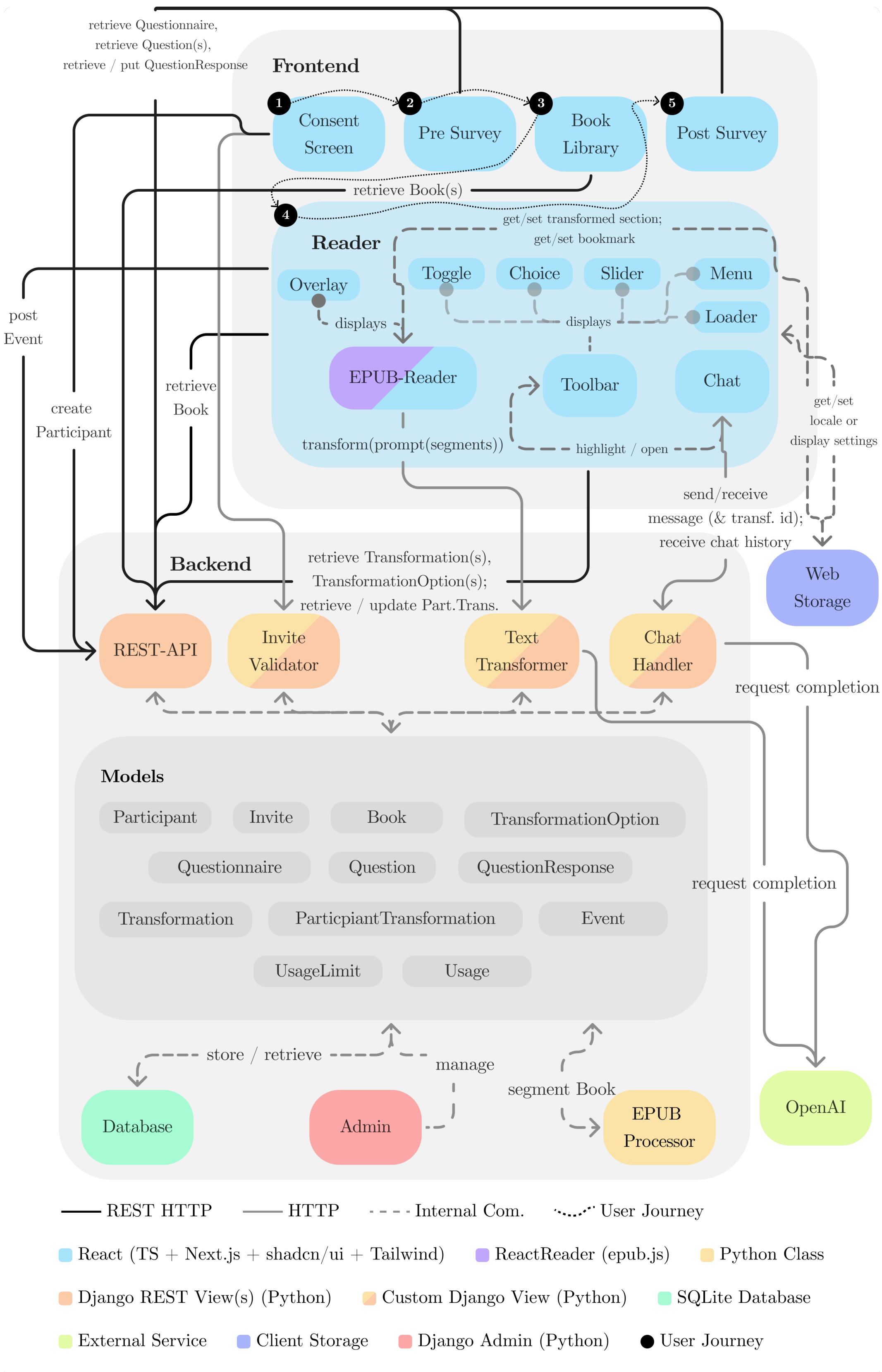

Figure 4.10: Component-based system architecture of the MagicTextReader platform, showing the frontend, backend, data models, and external integrations. The diagram illustrates both external and internal communication flows, key services (e.g., text transformation, chat), the complete user journey from consent to post-survey, model relationships, and the technologies used.

Backend Structure The server-side includes the following:

- **REST API**: Provides access to all shared models (e.g., books, transformations, surveys).
- **InviteValidator**: For validating participant codes.
- **TextTransformer**: Transforms text segments via OpenAI (see Section 4.5.4)
- **ChatHandler**: Generates responses to user input and creates/updates transformations (see Section 4.5.9).
- **EPUB Processor**: Segments EPUBs upon upload (see Section 4.5.5).
- **Admin**: Allows management of shared models and participant/user access.
- **Database**: Stores participants, responses, events, transformations.

User Journey A typical participant session begins at the Consent Screen (1), where the invite code embedded in the URL is validated via the InviteValidator. If accepted and the participant consents, a new `Participant` is created through the REST API. The user is then forwarded to the Pre-Survey (2), where the questionnaire and existing responses are retrieved, and new or updated answers are submitted to the REST API, and upon completion, the user is redirected to the Book Library (3).

Participant session progresses from consent form through survey to book library

From the library, a selected book opens the Reader (4). The EPUB is rendered using the embedded EPUB-Reader, which also restores the last reading position from Web Storage. The system queries available transformations and the active configuration from the backend. If transformations are enabled, the Toolbar displays appropriate controls—toggles, choice menus, or sliders—depending on the type. For each section, the system checks whether a transformed version exists in the

Book selection opens reader with transformation controls and starts transformation architecture

Web Storage. If not, a request is sent to the backend's TextTransformer endpoint, which builds a prompt and queries OpenAI's API. While waiting for the response, a loading Overlay is shown and subsequently, the transformed text is injected into the EPUB-Reader and chached in Web Storage.

Chat interface enables natural language transformation definition and direct application

When the user navigates or switches transformation settings, the system checks for already-transformed segments or requests new ones accordingly. At any point, the user can open the Chat to describe a new transformation in natural language. The message is sent to the ChatHandler, which constructs a prompt and uses OpenAI's API to generate a response. If a transformation definition is returned, the user is offered to apply it directly. If confirmed, the Toolbar is updated and the transformation becomes active.

Session concludes with post-survey while system tracks events and manages resources<

At the end of the session, the user proceeds to the Post Survey 5, which mirrors the pre-survey but presents different questions. Throughout the session, key events (e.g., navigation, toggles, chat activity) are logged via posting `Events`. Usage of transformation and chat endpoints is tracked and rate-limited. Uploaded EPUBs are segmented by the EPUB processor before becoming available, and the Admin allows researchers to manage all content, including books, participants, and study settings.

Additional components follow standard architecture patterns with REST API integration

Simple Components The following components are not discussed in detail: Consent Screen, Pre-Survey, Post-Survey (for the survey system—see Section4.4.13), and Book Library. Each of these frontend components is implemented using `shadcn/ui` and built on the shared stack (React, TypeScript, Next.js, Tailwind CSS), interfacing with the backend solely via the REST API to retrieve, submit, or update the corresponding data models. The Admin component leverages Django's built-in admin interface [Django Software Foundation, 2025] and provides model-based access for content and participant management. The InviteValidator is a minimal Django view that checks the validity of invitation codes without exposing invite strings.

Language Support To fulfill the multilingual requirements outlined in Section 4.4.11, the frontend supports both English and German through locale files. At the beginning of the session, users are prompted to select a language via a menu. The selected preference is stored using the Web Storage and applied throughout the session to all interface components.

Frontend supports English and German through locale files

4.5.3 Modular Transformations

To implement the modular transformation system described in Section 4.4.8, the backend defines two central data models shared with the frontend: `Transformation` and `TransformationOption`. These models encode the available transformation types, user interface metadata, and transformation-specific parameters.

A `Transformation` includes:

- **Name** and **Description** (for UI explanation)
- **Panel type**: *choice* or *slider*
- **Visibility**: *public* or *creator*
- **Is active**, **order**, and **thread_id**
- **Slider configuration** (if applicable): **min_value**, **max_value**, **step**, and **default_value**
- **Metadata**: creator and usage statistics

Each `TransformationOption` is linked to its parent transformation and defines:

- **Label** and **Value** (tick position or categorical identifier)
- **Instruction**: the transformation instruction passed to the LLM
- **Order**: for UI rendering

Transformation controls instantiated according to the control system specification

Transformations—identified by their control types such as toggles, choices, and sliders—are instantiated according to the specification in Section 4.4.8. Toggle transformations are implemented as a constrained form of choice transformations—defined by a single *On* option—with their inactive state corresponding to the absence of an associated selection. Slider-based transformations are implemented such that users select values along absolute references (see Figure 4.11), with a dedicated reset button to revert to the original. Otherwise, providing a reference point for the original would require LLM-based assessment of the original text, possibly introducing additional errors that could cause user confusion, prioritizing system simplicity and reliability.

Sliders use absolute references with reset buttons to avoid LLM-based assessments

ParticipantTransformation model manages user-specific transformation activation and configuration<

Activation and configuration of transformations at the user/participant level are managed via the `ParticipantTransformation` model. This model links a participant, book, and transformation and a selected option (or slider value). A transformation is considered active for a participant if such an association exists. The toolbar retrieves these mappings through the REST API, and the TextTransformer endpoint automatically resolves active configurations for each request.

4.5.4 Text Transformer Endpoint

The TextTransformer endpoint processes transformation requests originating from the EPUB Reader. The endpoint accepts `POST` requests and performs three main operations:

- Parses the incoming request and identifies the segments to be transformed
- Constructs a transformation prompt based on participant preferences
- Extracts and returns the transformed segments from the language model response

Request Format The request must include the `book_id` as a query parameter and a JSON body with the following structure:

Listing 4.1: TextTransformer POST request format

```
{
  "current": <number>,
  "segments": {
    "<segment_id>": {
      "status": "<string>",
      "content": "<string>"
    }
  }
}
```

Only segments with status `"original"` are submitted for transformation. Transformed segments already present in the cache are ignored.

Prompt Construction A prompt is constructed in accordance with the framework described in Section 4.4.6. The system message begins with a static instruction template (see Appendix C), followed by a dynamically constructed list of active transformation instructions.

Transformation preferences are derived from the participant's `ParticipantTransformation` relations (as introduced in Section 4.5.3). If a discrete option is selected (as in toggle or choice transformations), its instruction is included. For slider-based transformations, the two closest reference options are identified and their instructions are combined using weighted annotations that reflect the slider position, summing to 100%.

Discrete transformations use direct instructions while sliders use weighted instruction combinations

Example:

Listing 4.2: Example instruction list in system prompt

```
30% The transformed text must use scholarly-level
    language.
70% The transformed text must use intermediate-level
    language.
The transformed text must be in German.
```

These instructions are appended line-by-line to the system message. The user message is then constructed by serializing the input segments into a comma-separated XML structure, matching the internal representation used by the EPUB Processor.

Regex post-processing for segment extraction

Segment Reconstruction The assembled prompt is sent to OpenAI's Completion API [OpenAI, 2024] using the official Python library [OpenAI, 2025]. The response is post-processed using regular expressions to extract the transformed segments, even in cases where the model deviates slightly from the expected structure. Only segments that were originally marked as `"original"` are updated and returned in the JSON response to the EPUB Reader.

4.5.5 EPUB Processor

EPUB Processor segments uploaded books using recursive algorithm preserving text boundaries

Upon each EPUB upload, the Django backend invokes the EPUB Processor to segment content (Section 4.4.3) using a recursive algorithm that preserves sentence and word boundaries. The EPUB archive is first extracted to a temporary directory via Python's `zipfile`. Content documents are then iterated using the `EbookLib` package [Kovář, n.d.] (see Section 2.2.4 for structure), and segmentation is applied to each.

Segmentation Algorithm

- **Parse Document:** XML structure is parsed using `BeautifulSoup4` [Richardson, n.d.].
- **Start at `<body>`:** Segmentation begins at the `<body>` element and proceeds recursively.
- **Recursive Logic:**
  - **Text nodes:**
    * Split at sentence boundaries into segments <=2000 characters.

    * If a sentence exceeds the limit, further split at word boundaries.
  - **Element nodes:**
    * If total content fits within limit, treat as a single segment.
    * Otherwise, recurse into child nodes.

Segment Annotation Segments are marked within the XML tree using custom comments:

```
<!-- MagicTextreaderPart id="X" status="orig" -->
...<!-- // MagicTextreaderPart -->
```

Here, `X` denotes a sequential ID. The annotated trees are held in memory, then written to the temporary directory and repackaged into a new EPUB archive.

4.5.6 Reader

The EPUB-Reader implements the transformation architecture introduced in Section 4.4.4 and the cache system described in Section 4.4.7. It is based on the `epub.js` [FuturePress, 2025] library and its React wrapper `ReactReader` [Sletten, 2017], which provides a component-based interface for rendering EPUB-files within an `iframe`. The transformation system is realized by leveraging properties and callbacks exposed by `ReactReader`:

EPUB-Reader implements transformation architecture using epub.js library and ReactReader wrapper

- `getRendition`: Provides access to the `rendition` object once the reader is initialized. An event listener is attached to the `rendered` event, which fires whenever a new section is displayed. Initially, the original HTML content of the current section is extracted by accessing the `iframe` DOM via `document.querySelector` [MDN contributors, 2025]. The HTML is saved in a temporary variable to serve as the baseline for comparison and restoration. If an active transformation exists, the system queries Web Storage for a cached version of the transformed section corresponding to the current transformation setup, participant, and book. If found, the transformed HTML is injected directly into the `iframe`'s DOM. As resource links are temporarily generated by *ReactReader* across sessions, all resource references (e.g., images, fonts) in the transformed HTML are synchronized with the original version before injection.

- `locationChanged`: This callback is triggered on navigation events (and initially) and receives an EPUB Canonical Fragment Identifier (`epubcfi`) [Sorotokin et al., 2011], which serves as a pointer to the current position in the EPUB. Using this reference, a `Range` object [MDN contributors, 2023] is created to extract the corresponding subtree from the section's DOM. Segment identification is performed by locating the first custom comment marker (see Section 4.5.5), from which the segment ID is extracted using regular expressions. Transformed segments are inserted by replacing the relevant portion of the HTML string (also using regular expressions), and the updated HTML is re-injected into the `iframe`. The (partly) transformed section is then saved to Web Storage.

- `theme`: This property applies styling preferences such as font size, font family, and color mode to the `ReactReader` interface [Sletten, 2017]. These preferences are set by the user and stored persistently using Web Storage to ensure consistency across sessions.

- `location` and `url`: These properties are used to load the EPUB file and restore the participant's last reading position, respectively.

Toolbar provides transformation controls, display settings, and chat access interface

The Toolbar complements the `EPUB-Reader` by offering an interactive interface for transformation selection, display customization and access to the Chat (see Section 4.5.9). It is composed of transformation-specific UI elements— Toggles, Choices, and Sliders—each corresponding to a different type of transformation as defined in Section 4.4.8. They provide visual feedback on their current activation status, and any reconfiguration must be confirmed via a checkmark button to apply the updated transformation settings. Additional elements include a Menu for adjusting font size, font family, and theme. These preferences are stored locally using Web Storage and applied via the `theme` attribute of `ReactReader`. The Toolbar also includes an option to close the book and displays a Loader animation while transformations are being processed. All UI components are implemented using `shadcn/ui` and built on the shared frontend stack (React, TypeScript, Next.js, Tailwind CSS). Communication with the backend is handled exclusively via the REST API to retrieve, submit, or update the corresponding data models.

4.5.7 Overlay

To support the visual feedback system (see Section 4.4.9), the interface displays overlays on paragraphs (that are displayed by the EPUB-Reader) containing segments undergoing transformation. Since `epub.js` renders EPUB content as HTML within an iframe, the system traverses the iframe's DOM to locate the segments using annotations inserted during preprocessing (see Section 4.5.5), identifies those currently visible in the viewport, and renders overlays aligned with the corresponding paragraphs to match their size and indicate each segment's transformation state (see Figure 4.11).

Overlays are rendered on visible segments to indicate transformation status

4.5.8 Events

The frontend records user interactions in accordance with the structure defined in Section 4.4.12 and transmits them to the REST API. The following events are tracked, grouped by component. Additional context data sent with each event is indicated in parentheses.

Frontend tracks user interactions via REST API

- **EPUB-Reader:** Open book (configuration), Close book, View section (section_id), Turn page

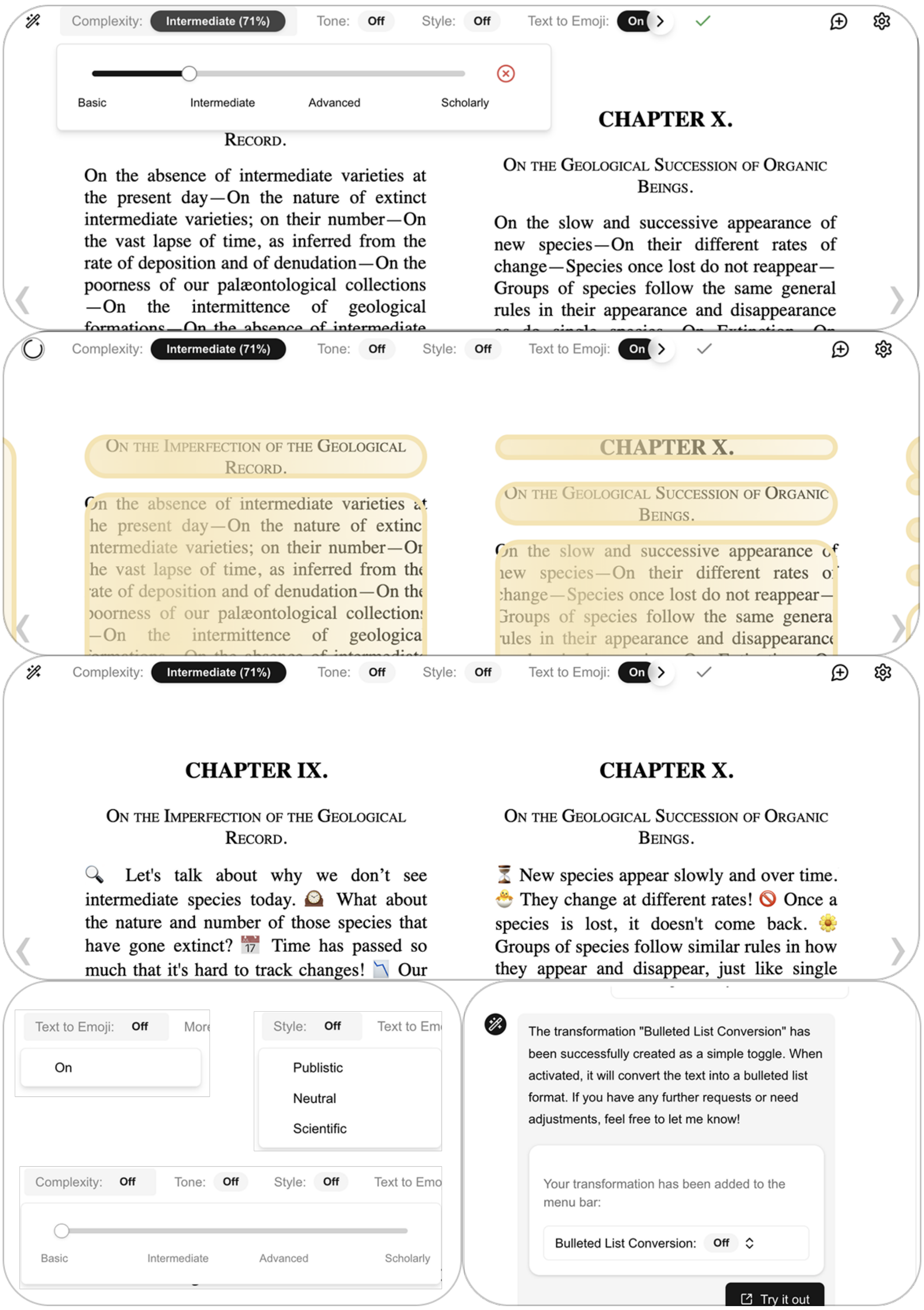

Figure 4.11: Screenshots of the MagicTextreader prototype. The top three sections show the Reader: initially with no transformation applied but with a complexity slider in use and a Text-to-Emoji transformation option enabled; then after applying transformations, displaying a pulsing overlay and a loading indicator; and finally, showing the transformed text. The displayed book page is from On the Origin of Species by Means of Natural Selection by Charles Darwin [Darwin, 1861]. The bottom left section presents all control types views (Toggle, Choice and Slider), while the bottom right shows the chat interface used to create transformations.

(direction), Request transformation (transformation_config), Complete transformation (transformation_config)

- **Chat:** Send Message (message, history), Received message (message, history), Create / Update transformation (transformation_id), "Try it out" (transformation_id), Add transformation, Close chat
- **Toolbar:** Change transformation, Enable transformation, Edit transformation, Adjust font size (font_size), Change theme (dark/light), Change font family

4.5.9 Chat

The Chat provides an interface (see Figure 4.11 for an illustration) for creating and updating transformations via a conversational assistant in accordance with Section 4.4.10. Users get access chat either by clicking the *New Chat* button or by editing an existing transformation, in which case the corresponding `thread_id` is passed to the view for context.

Chat interface enables conversational creation and editing of transformations

The component presents a minimal chat interface (that displays chat messages and indicates the creation or update of transformations) and performs the following two operations:

- Requests a chat history from the backend
- Sends new messages to and retrieves responses from the ChatHandler

The ChatHandler is implemented as a custom Django endpoint that accepts HTTP-GET and POST-requests and communicates with OpenAI's Assistant [OpenAI, 2024] and Thread [OpenAI, 2024] APIs via their Python library [OpenAI, 2025]. It handles two HTTP methods:

- **GET:** If a `thread_id` is provided, the backend fetches the full message history for that thread from OpenAI. If no `thread_id` is provided, it creates a new thread and initializes it with a system message: *"Hi! I'll help you create a new text transformation. Do you have any preferences in mind?"*

- **POST:** Upon receiving a new user message, the backend adds it to the corresponding thread and starts a new assistant *run*. The assistant, configured according to Section 4.4.12 with an instruction and function definition (see Appendix B), generates a response to the thread. If the assistant issues a function call (e.g., to create or update a transformation), the backend executes it and updates/creates the corresponding transformation models as described in Section 4.5.3. A confirmation message is then added to the thread to complete the function call cycle. Finally, all new assistant messages are returned to the chat component

API usage and costs are tracked per participant. Response generation is blocked at cost limits, but errors are not reported, as this is unlikely in the study to happen.

4.5.10 Testing Strategies and Development Process

Mock transformations enable cost-free testing and development of transformation logic

Mock Transformations To enable implementation and interface testing without causing API costs, the TextTransformer was initially configured to apply mock transformations. A first placeholder algorithm programmatically replaced every second word in a text segment with *Blabla* and introduced artificial delays matching realistic response times (see Section 4.4.5). Another mock transformation was used that replaced the content of each segment with a chain of current timestamps, where the total number of inserted timestamps could be scaled to a configurable percentage of the original segment's length. This allowed visual verification of transformation logic and end-to-end testing of the frontend without causing costs, including evaluation of how the reader handles transformations that alter the length of the original segments.

Hard-coded limits protect early OpenAI API integration from uncontrolled usage

Controlled API Integration Early integration with the

OpenAI API was protected by hard-coded backend limits, restricting the number of requests per minute and per hour to prevent uncontrolled usage. For prompt construction and debugging, the same *Blabla* transformation (in this case applied via an LLM instruction) was reused to validate injection and formatting logic before switching to real transformations.

Reproducibility via Temperature Control During development, requests to the OpenAI Completions API [OpenAI, 2024] were made using a low `temperature` parameter to reduce output randomness. While full determinism cannot be guaranteed 3.4.5, this strategy enabled more reproducible testing outcomes and simplified debugging of transformation behaviors.

Low temperature parameter improves testing reproducibility and debugging

Chapter 5

Study and Results

This chapter presents a first study conducted to investigate the following research questions:

RQ1: In what ways does an AI-enhanced text reader affect text accessibility for users?

RQ2: What design implications emerge from user interactions with an AI-enhanced text reader?

RQ3: What research directions and application contexts show promise for AI-enhanced text readers?

In the following, the chapter is structured into two main parts: the study design and the study results. The first part describes the overall setup, including methodological choices, materials, procedures, and ethical considerations. The second part presents the findings based on questionnaire data, user observations, interviews, and interaction logs.

5.1 Study Design

5.1.1 Methodology

Mixed methods study design enables first comprehensive evaluation

Given the limited existing research on integrated text transformation tools, a session-based mixed methods study—combining both quantitative and qualitative approaches—was conducted to investigate the impact on text accessibility and user interaction patterns with the MagicTextreader. This approach allows for a more comprehensive understanding of complex research problems than the use of a single method alone. As noted by Palinkas et al. [2019], "the use of quantitative and qualitative approaches in combination provides a better understanding of research problems than does either approach alone," a position supported by earlier works [Creswell and Clark, 2017; Morse, 2016; Locke, 2002]. While an ethnographic approach with participants using the prototype on their own and on demand was initially considered, preliminary experiments revealed OpenAI API instability and lack of proper error management within the prototype, making the system too unpredictable for unobserved use.

Technical instability prevented unsupervised ethnographic study approach

Sessions designed to approximate natural reading conditions while encouraging exploration

Central to the study is an interaction phase in which participants engage with the system in a session-based setting designed to approximate natural reading conditions. While limited in duration and incentivized for exploration, the setup still aims to support situated use and leans toward an ethnographic orientation and exploration.

The choice of methods—questionnaires, observation, and semi-structured interviews—was inspired by established approaches to toolkit evaluation Ledo et al. [2018], though the toolkit itself was not the main object of investigation, but rather its specific application instance.

No experimental manipulation of variables

As the study was exploratory and ethnographically oriented, no experimental conditions were applied, and no variables were manipulated.

5.1.2 Setup

Environment The study is conducted in individual remote sessions. Participants are invited to video calls, which remain open throughout the session. In the call, the study is introduced, instructions are given, and the concluding interview is conducted. Participants receive an invite link to access the MagicTextreader platform on their own device. Supported devices include desktop computers, laptops, and tablets; smartphones are not supported due to limited interface scalability.

Remote study sessions conducted via video calls with device-specific platform access

Sessions Each session lasts approximately 60-90 minutes and follows a fixed sequence: introduction, pre-interaction questionnaire, system walkthrough, free interaction phase, post-interaction questionnaire, and a concluding semi-structured interview. For procedural details, see Section 5.1.7.

Structured 60-90 minute protocol with questionnaires and interview

Materials The study materials include a selection of non-fiction books (see Section 5.1.4), a set of predefined transformation options available in the interface (see Section 5.1.5), and structured questionnaires administered before and after the interaction (see Section 5.1.6).

Books, Pre-Defined Transformations and Questionnaires

Data Collection Data is collected through multiple sources. Interaction events are automatically logged by the reader interface (see Section 4.5.8). Participants complete questionnaires before and after the interaction (see Sections 5.1.6 and 5.1.6). In addition, observation notes are taken during the session, and qualitative data is gathered through a short interview at the end (see Section 5.1.8).

Event logs, questionnaires, observations, and interviews

5.1.3 Participant Recruitment

Participants were recruited through convenience sampling using personal networks, including friends, acquaintances,

Convenience sampling recruited adult participants fluent in English or German family members, and individuals from the university campus. No specific inclusion or exclusion criteria were applied beyond the requirement that participants were adults (18 years or older) and fluent in either English or German. While text simplification is particularly relevant for specific groups such as individuals with dyslexia, as outlined in Section 2.1.1, the section also shows that text accessibility is a broader issue affecting a wide range of readers and aspects of a text. For this reason, no further specific recruitment criteria is applied.

5.1.4 Book Selection

Non-Fiction Scope and Rationale

Non-fiction chosen as transformations preserve informational purpose

This study uses only extracts from non-fiction books, as they present information or facts [Cambridge Dictionary, 2025], allowing for an investigation into how transformations impact accessibility without undermining the text's core purpose. In contrast, fiction relies on how it is written—the author's intent, tone, and form are central to its function [Currie, 1985]—making such transformations likely to distort its meaning and intent.

Selection Process

Five diverse non-fiction books selected to minimize bias and ensure variety

To minimize bias through specific book choices, five books were selected based on a combination of factors: popularity (based on highest rank on Best Book Lists [Goodreads, 2024]), topic diversity, availability in both English and German, and a wide publication year range (from 1945 to 2022). These criteria aim to capture a broad range of non-fiction styles and content areas while keeping the number of books manageable for feasibility.

Book Categories

From each of the following categories, one book was selected:

- Religion
- Finance
- Philosophy
- Technology
- History

Each participant can select one of these book extracts via the integrated book library, with access restricted to the duration of the study.

Following the approach by Zhiyuli et al. [2023] of prompting LLMs to generate book blurbs using only titles and instructions (see Section 3.6.5), GPT-4o [OpenAI, 2024] was used to generate blurbs for the library books to support participants in their selection process.

Legal considerations

Only small book extracts (less than 15% of the works) were used, made available unchanged and exclusively during the study sessions. Accessibility-focused transformations occurred locally (on participants' devices) and on demand, with efforts made to preserve information integrity. Their outcomes were also accessible only during the sessions and could not be downloaded.

5.1.5 Pre-Defined Transformations

The reader was set up with five predefined text transformations, each representing a different text accessibility dimension, which are available to all participants. These transformations were chosen based on and inspired by factors influencing text accessibility, as discussed in Section 2.1.1. Participants can select and apply these transformations via

Setup includes five predefined transformations based on text accessibility factors the interface. For each option, the exact transformation instruction used in the LLM prompt is provided in **Appendix F** and was crafted using the prompt engineering techniques outlined in Section 3.4.2.. The following list presents each transformation along with its type, effect, available options or range, and the rationale (based on Section 2.1.1) for its inclusion.

Complexity
: **Type:** Slider (0–100, step size: 5)
Effect: Adjusts the complexity of the text to match the reader's proficiency level.
Range: Basic → Intermediate → Advanced → Scholarly
Rationale: Selected to address the need for text simplification.

Tone
: **Type:** Choice
Effect: Adapts the text to sound more formal, casual, convincing, or humorous.
Options: Formal, Conversational, Persuasive, Humorous
Rationale: Selected based on research suggesting that tone influences how readers connect with information.

Style
: **Type:** Choice
Effect: Reformulates the text to reflect different writing cultures: publicistic, neutral, or scientific.
Options: Publicistic, Neutral, Scientific
Rationale: Selected because studies show that certain styles can enhance comprehension.

More Personal
: **Type:** Slider (0–100, step size: 25)
Effect: Influences how personally involved the narrator appears in the text.
Range: Neutral → Slightly → Moderately → Very → Extremely
Rationale: Selected based on a study suggesting that a more personal tone can improve transfer performance.

Text to Emoji

Type: Toggle
Effect: Enriches the text with relevant emojis.
Option: On
Rationale: Selected to reflect preferences among younger audiences for visual elements and varying messaging styles.

5.1.6 Questionnaires

A pre- and post-interaction questionnaire was developed to support and complement qualitative observations with quantitative data. Questions include both open- and closed-ended formats. The design (following [Müller et al., 2014]) aims to minimize bias and avoid broad, leading, or double-barreled formulations and the System Usability Scale [Brooke et al., 1996] is included for comparability. When directly asking about preferences, the questionnaires use Likert-scale items [Likert, 1932] to capture graded responses.

Questionnaire design follows established survey methodology with SUS for usability assessment

Pre-Interaction

The pre-interaction questionnaire collects structured information before participants use the reader. It covers five main sections, each addressing factors that we assume—or for which there are indications—that they may influence how participants experience text transformations:

1. **Demographics** – Basic background information such as education, occupation, age, and gender. These data are relevant for generalizability, as characteristics like gender can affect results [Offenwanger et al., 2021], and different groups benefit differently from interventions (see Section 2.1.1).

2. **Reading Habits and Preferences** – Questions about how often participants read, whether they prefer digital or print, how long they read, and how they ap-

proach understanding texts. These habits may influence how participants experience text transformations, especially since digital reading accounts for only 16% of the market (see Section 2.2.1), suggesting a preference for print.

3. **Content and Subject Matter** – Types of texts and topics the participants usually read, and their reasons for reading. This might help to understand how familiar participants are with non-fiction texts, which may affect their responses to the interventions.
4. **Technical Experience and Preferences** – Familiarity with digital reading tools, file formats, and preferred device features. This allows analysis of whether users with different digital skills use the tool differently or face usability challenges.
5. **Accessibility Requirements** – Possible visual or cognitive difficulties related to reading. These are considered because complexity and related factors can especially impact groups such as dyslexic readers or those with lower reading skills (see Section 2.1.1).

The complete version of the questionnaire can be found in **Appendix D**.

Post-Interaction Questionnaire Design

The post-interaction questionnaire is used to collect structured feedback after participants interact with the reader. It is designed to quantify user experience across reading, usability, and performance, and to help explain or support qualitative observations—such as issues with speed or system behaviour. It is divided into five sections, three of which are specifically tailored to gather quantified data addressing the study's research questions:

1. **Reading Impact (RQ1)** – Measures how transformations affected reading speed, understanding, and experience of reading non-fiction texts.

2. **Design and Usability (RQ2)** – Focuses on how usable and understandable the transformation system was, including layout and interaction types.

3. **Technical Performance** – Captures whether users experienced any technical problems or delays while using the system.

4. **Future Usage and Improvements (RQ3)** – Asks how likely users are to use such a tool in the future and what could be improved.

5. **System Usability Scale (SUS)** – Standardised set of items to evaluate general usability of the system [Brooke et al., 1996].

The full version of the post-interaction questionnaire is included in **Appendix E**.

5.1.7 Study Protocol

The study follows the following procedure and consists of several consecutive phases.

Study explained and consent obtained before proceeding with session

Introduction and Consent The session begins with a short introduction. The participant is informed that the study investigates an ePub reader with additional features. The goal is to explore how the tool is used in the context of non-fiction reading. It is explained that data is collected through questionnaires, interaction logging, and a short interview at the end. The participant gives consent via a digital form before continuing and additionally fills out a consent form that is explained and provided (see Appendix G for the consent form), either digitally or by printing, signing, and sending back a scanned or photographed copy.

Pre-Interaction Questionnaire After consent is given, the participant completes the pre-interaction questionnaire introduced before in Section 5.1.6.

Reader interface introduced with book selection and basic navigation instruction

System Introduction and Task Explanation The interface of the reader is introduced. The participant is prompted to choose one non-fiction book from a small collection, selecting one that best fits their personal interest (see Section 5.1.4). Basic navigation is explained, including how to turn pages using on-screen buttons or keyboard arrows.

Predefined transformations introduced with usage guidelines and prototype limitations

The predefined transformation options are introduced (see Section 5.1.5). It is noted that the transformations are powered by AI and may take a few seconds to process. During this time, a yellow overlay indicates the loading state, and it is advised not to switch pages until the transformation is complete. Additionally, it is mentioned that the system is a prototype and occasional delays are expected or errors can occur.

The participant is also shown how to create custom transformations using a chatbot interface. As an example, a transformation is created that rewrites paragraphs as bullet points.

The participant is then informed about four small tasks that should be completed during the session:

- Apply at least two predefined transformations.
- Create and apply at least one custom transformation.
- Read with a transformation enabled for at least five minutes.
- Fully read one book excerpt, which is later used in comprehension questions.

Free interaction phase allows 3̃0-minute exploration with observation and task guidance

Interaction Phase The participant has about 30 minutes of free interaction. During this time, the tasks should be completed, but otherwise, the participant is free to explore the system as desired, including switching between books. The session is continuously observed, and a written protocol is kept to document notable behaviors, comments, and reactions. Participants are encouraged to think aloud if they feel comfortable. If questions arise, they can be addressed directly. Occasional reminders are given if tasks appear to be forgotten or incomplete.

Post-Interaction Questionnaire and Interview After the interaction phase, the participant completes the post-interaction questionnaire introduced before in Section 5.1.6. This is followed by a short semi-structured interview to gather additional qualitative feedback (described in Section 5.1.8).

5.1.8 Semi-structured Interview

Each session concludes with a short semi-structured interview aimed at gathering qualitative insights into participants' reading experience, interactions with the interface, and views on the tool's potential.

Sessions conclude with semi-structured interviews exploring accessibility and interface experiences

The interview follows a flexible format and is guided by prepared prompts aligned with the study's three research questions. Specifically, the questions cover four thematic domains: the impact of the transformations on text accessibility, interaction with the interface, perceived future potential of the tool, and any additional user reflections.

Responses are documented through written notes taken during the conversation.

5.1.9 Ethics

The study follows ethical standards regarding informed consent, data privacy, and protection. Participants were informed about the study's purpose, procedures, and their rights before providing consent.

Privacy

Participation was voluntary, and participants could withdraw at any time without providing a reason. Data access is limited strictly to authorized members of the research team.

Data Protection

Secure storage, encryption, anonymization, and scheduled deletion is ensured

All collected data were stored on secure servers at RWTH Aachen University and were accessible only to involved staff. Data transmission occurred exclusively via encrypted connections. Personally identifiable information was anonymized as much as possible without compromising the study's objectives. All data are retained only for the duration necessary to complete the analysis and are permanently deleted thereafter.

5.1.10 Data Analysis Approach

Python scripts process collected data generating descriptive statistics and summaries

Python scripts were developed to analyze questionnaire and event data, collecting all recorded data and applying statistical processing including calculation of averages, minimum and maximum values, and differences. The scripts output .md or .csv files containing all relevant data, which are then logically grouped and summarized. Questionnaire responses in English and German were consolidated. Survey and interaction log data are presented descriptively without empirical analysis.

Qualitative findings are thematically organized into coherent patterns and themes

Findings from observations and semi-structured interviews are thematically grouped following Braun and Clarke [2006] and Braun and Clarke [2023]. This thematic analysis organizes qualitative data into coherent patterns and themes. Within groups, findings are ordered by frequency of mention without implying significance or relevance. In rare cases, findings are included in multiple categories when relevant to both groups. Participant remarks and responses in German were translated to English for presentation.

5.2 Results

Eleven participants (N = 11) completed the study, including pre- and post-interaction questionnaires, prototype interaction with observation, and semi-structured interviews.

5.2.1 Pre-Interaction Questionnaire

Demographics Most participants held a Bachelor's or Master's degree. Their daily activities included research, part-time work, full-time work, and study. The majority were aged between 25 and 34. The gender distribution consisted of six male and five female participants. Four participants chose English as interface language for the reader and platform, reading English book extracts, with one participant also providing remarks and conducting the semi-structured interview in English. The remaining participants used German.

Participants comprised educated adults aged 25-34 with varied backgrounds and language preferences

Reading Habits and Preferences Preferences for reading format were roughly evenly split between digital and analog media. Participants reported reading non-fiction texts regularly, with most sessions lasting 30–60 minutes and weekly reading time under 10 hours. German was the primary language, with English also used frequently. Articles and non-fiction books were commonly read. Reading occurred across diverse domains, particularly academic and scientific contexts, and was mainly motivated by personal interest and academic requirements. Comprehension was considered important, and common aids included highlighting, summarization, and visual diagrams. Strategies for complex texts included re-reading and looking up terms.

Participants show diverse reading habits with mixed digital-analog preferences and comprehension strategies

Technical Experience and Preferences Participants reported high comfort with digital reading tools. PDF readers were used by nearly all, followed by Kindle and Apple Books. Preferred features included search, highlighting,

Participants show digital reading comfort preferring PDF format and display customization. PDF was the most commonly read format.

Devices and Display Laptops and smartphones were the most frequently used devices. Larger screens were generally preferred. Most participants favored light mode and medium font sizes.

Accessibility Requirements None of the participants reported reading-related conditions.

5.2.2 Observations and Interview Responses

User Reactions to Text Transformations

Expressed strong satisfaction with successful transformations

Positive Reactions Nine participants exhibited strong positive emotions when custom transformations functioned as expected and also nine participants expressed in general satisfaction with transformation outcomes, with comments such as "I felt very good about it, I thought I would like to use it in everyday life", "Wow, that really hits the mark.", "I find it very interesting to read it that way" and "the expectations of my own transformation were very fulfilled." One participant noted that the transformation provided "a nice opportunity to read the text again from other perspectives."

Reported improved reading flow

Eight participants reported improved reading flow following transformations. The reading experience was described as positively influenced, with comments including "I noticed a slight simplification. The reading flow felt better and the grammar seemed somewhat simplified" and "it's easier to read now." One participant specifically mentioned finding "it much more pleasant to read the book with transformations; I didn't like the original writing style."

Vocal engagement

Six participants showed vocal engagement and laughter during reading sessions. Users found the transformed versions entertaining and amusing, with reactions such as "it sounds funny compared to the normal version: the educated language is amusing" and "the humorous tone makes it significantly more accessible." The transformed content was described as "very amusing" by some users.

Two participants explicitly indicated preference for transformed versions over the original text, with one stating they found "the formal style significantly better than the original version."

Negative Reactions Seven participants stated dislike of specific transformations. The "More Personal" transformation was particularly criticized, with comments such as "what didn't appeal to me was the personal aspect" and "very personal doesn't hit the mark well." Emoji usage was frequently rejected, described as "rather inappropriate" and "only for children." Some participants noted that certain transformations "lost the meaning" and "changed the taste" of the original text.

Disliked "More Personal" and emoji transformations

Six participants reported confusion about transformation differences. Users expressed difficulty distinguishing between various versions, stating "I couldn't determine the difference between different versions" and noted that "sometimes there seem to be inconsistencies between segments." The lack of clear differentiation between transformation levels was a recurring concern.

Confusion about transformation differences and inconsistencies

Six participants expressed disappointment with transformation outcomes. Some transformations were perceived as making "the text not make sense" or becoming "more confusing." One participant mentioned that with overly simplified complexity, they "would have put the book away."

Disappointed with outcomes

One participant noted the inadequacy of certain transformations, describing an informative transformation as sounding "more like a bullet-point summary" rather than the desired outcome.

Transformation inadequacy

Mixed Reactions Four participants expressed ambivalence about the text modification concept. Some users recognized that text transformation represents "a huge intervention in the book" while simultaneously showing interest in the functionality.

Improved reading flow and comprehension with transformations

Behavioral Changes Nine participants noted improved reading flow when using transformations. Comments included "then it was very fluid to read" and "using this app made it faster that I understand things." Users reported that "with that tool I got some parts better and was understanding and going forward."

Six participants exhibited vocal reading behaviors during transformation use.

Increased reading motivation through humorous transformations

Five participants reported increased motivation to read. Users expressed being "curious about the further content of the book" and noted that "humorous was surprisingly useful for reading something you don't feel like reading; it helped motivate you to continue reading." The humorous transformation was particularly effective in creating "a bit more anticipation" while reading.

No change in reading strategy

Four participants reported no change in reading strategy despite using transformations. Some maintained that their strategy "was not changed" and expressed concerns about trusting transformations for important texts.

Reduced need for repetition and no skipping of difficult passages

Three participants reported reduced need for repetition of sentences and no skipping of difficult passages. Users noted that "normally I would skip difficult passages or read them multiple times; like that I don't need that and would just read through."

Reduced need for reading aids

Two participants stated reduced need for reading aids, with one mentioning that "with simpler language it stays in the head better and therefore changed my reading strategy."

Interface Design Issues

Core Usability Issues Ten participants perceived slider values as ambiguous. Users expressed confusion with comments such as "I don't understand what the percentage values mean. It must be much clearer what the options mean" and "I don't understand what the reference points mean."

Slider values ambiguous

Six participants showed repeated confusion about transformed or original status. Users frequently questioned whether they were viewing the original or transformed version, with comments like "is this already the transformed version" and "when I didn't know what was the transformation and what was the original, then I was very uncertain."

Confusion about status

Five participants lost their reading position after transformation. Users noted that "you don't know exactly where the reading position is after the transformation, usually it jumps by a page" and expressed desire to "remember where I was."

Loss of reading position

Five participants experienced unexpected layout reflows after transformation. Users questioned "am I on the same page now" and suggested that "it should be scaled directly so that it matches page to page, even if empty areas arise."

Confusion caused by unexpected layout reflow

Five participants showed confusion with chat interface input formats and intention expression. Users found it difficult to "properly convey" their transformation requests through the chat interface.

Confusion with chat interface input formats and intention

Four participants tested extreme values after confusion about small changes. When subtle modifications were not apparent, users moved sliders to maximum positions "to see if anything changes."

Confusion about small changes

Two participants experienced confusion from excessive simultaneous transformations. Users suggested that "one should limit the number of transformations that are active at the same time" and noted they "didn't notice that they were still on."

Confusion from simultaneous transformations

Two participants asked again how to apply transformations or open the chat; these functions had been demonstrated earlier to all participants, including them.

Transformation delays disrupt reading flow

Technical Performance Issues Five participants found that transformation delays disrupted reading flow. One mentioned that if the system is "not instant, then I wouldn't use it. That would annoy me."

Information Architecture Issues Five participants suggested starting with broad slider levels accompanied by examples, followed by an option for fine-tuning adjustments later.

Four participants requested better control over multiple active transformations. Users wanted to "be able to determine the order in which multiple transformations are applied one after another."

Requested visual markers for position mapping

Two participants requested visual markers of reading position or clearer mapping between versions. Users desired "to set a marker at my current reading position that I can find in the transformed version" or maintain "the layout exactly like the original."

Toolbar perceived as intuitive

Interface Effectiveness Seven participants found the toolbar intuitive but assumed immediate application and suggested it should disappear. Users treated the interface "like quick access" and found the "checkmark too much." Some suggested that "I would prefer if it were hidden again so you can concentrate fully on reading."

Choice controls perceived intuitive and reliable

Six participants found choice controls intuitive and reliable. Users preferred "clicking rather than sliding" and noted that "with choice it's easier to understand." The discrete options provided clearer expectations compared to continuous sliders.

Chat interface familiar through ChatGPT

Six participants found the chat interface familiar and compared its use with ChatGPT. Users were "already used to it like with ChatGPT," making the interaction pattern recognizable.

Five participants found chat-based creation cognitively demanding but appreciated the results. Users noted that "you have to give very precise instructions" and found "creating custom transformation most difficult compared to the others." However, they appreciated that the system "accomplished it very well" despite the difficulty in formulating requests.

Chat creation cognitively demanding but results appreciated

Four participants perceived toggle controls as trustworthy, finding the binary state safer. Users preferred "on/off toggle" especially "for those you created yourself" and found the "best generation experience" with this control type.

Toggles perceived as trustworthy

User Mental Models and Expectations

MENTAL MODELS:
Mental models are the internal representations people form to understand how a system or object works [Norman, 2013].

Definition: *Mental Models*

System-Aligned Mental Models Seven participants correctly assumed that changing transformations always used the original as the starting point.

Five participants correctly understood how the visual feedback overlay symbolized loading and transitions between versions. Users interpreted the overlay as an indicator of text processing.

Three participants used the visual feedback overlay as orientation for how the text changes. Users noted that "the boxes got smaller, so I think the text became smaller."

System-Misaligned Mental Models Seven participants

Expected baseline as reference

wanted to slide relative to the original or understand the classification of the original text. Users questioned "what is the baseline and how is the original text classified" and "where was the original on the slider."

Expected local transformations

Seven participants expected local changes while the system worked globally. Users expressed desire to "transform only certain areas" and complained that "you can't look up or transform individual things." Many wanted to "mark a specific place and transform it rather than everything at once."

Assumed bigger slider changes

Six participants assumed bigger changes and showed confusion about small changes on sliders. Users noted that "with the slider, the distances weren't necessarily even or too close together in the result" and "with smaller changes I don't notice a difference."

Five participants assumed transformations were quicker to switch or reversible. Users expected more immediate and flexible switching between transformation states.

Five participants expected deeper personalization but experienced surface adaptation. Users anticipated more profound changes, stating "too little 'extremely personal,' would have expected more".

Expected immediate transformation creation

Five participants assumed they would directly receive a transformation after the first chat message and later change if necessary through trial and error. Users expected that "immediately after my message a transformation is created and I can try it out and then adjust it."

Formulated page-specific transformation descriptions

Four participants formulated transformation descriptions for the current book and page rather than general transformations. Users commented on specific text characteristics like "the text lacks humor" or "the sentences here are too long."

Assumed quick-control

Two participants assumed transformations would be applied instantly, interpreting the interface as resembling a quick-access menu. The visual design led them to expect immediate results and made the setup process feel unnecessary.

Missing Functionality Expectations Seven participants wanted to look up terms and ask the AI for explanations. Users requested functionality to "ask what the term means" and have the AI "explain terms" within the reading context.

Transparency and Feedback Needs Seven participants wanted to know what changed in terms of style, content, structure, and degree of modification. Users could not "exactly name what changed" and suggested "it would be good to know the exact transformation instruction also for the predefined options." Clear communication about modifications was consistently requested.

Wanted transparency about transformation changes

Five participants requested hover comparison or toggle to original text. Users wanted "quick switching" functionality and the ability to "quickly look at the original text with one click." This was considered "very important for trust" especially in scientific work.

Four participants desired clearer feedback about scope and intensity of transformations. Users questioned whether settings like "if I set 'light,' does it still have to be lyrical" and requested "much more transparent" information with "better reference points."

Three participants needed clearer visual cues or animations for transitions. Users suggested "it would be cool if you could see exactly how it rewrites itself with live animation" and requested clearer indication "that the new version is now coming in."

Usage Context and Preferences

Text Type Preferences Eight participants identified educational content as appropriate for transformation. Users mentioned "specialized books, books with scientific terms or complex content" and noted particular value "for children and people who don't enjoy reading."

Seven participants stated willingness to use transformations with non-fiction texts. Users expressed interest in applying the system to "factual texts," "scientific non-fiction texts," "papers, scientific work" and "scientific publications."

Fiction use explicitly rejected

Six participants explicitly rejected fiction use. Users emphasized that "for fiction and novels I find it completely unsuitable because you would falsify the art" and "the author's style should remain." The consensus was that "for fiction, storytelling is already engaging and you want to know how the author describes things."

Two participants mentioned foreign language texts as a type of content they would be interested in using the system for.

One participant mentioned newspaper articles as particularly suitable for transformation.

One participant saw potential for fiction use, but only in the form of supportive features such as reminders or aids to help follow the storyline or track characters.

Complexity described as most useful/helpful

Transformation Preferences and Reactions Ten participants described complexity as the most useful transformation. Users stated they "would most likely adjust complexity for efficiency because then I understand faster" and found "complexity most helpful." The simplification aspect was consistently valued for improving comprehension speed. Three participants also remarked that they tried complexity first since it was the first option in the toolbar.

Tone adjustment created engagement

Seven participants identified tone adjustment interest, especially humorous transformations, and found them engaging. Users appreciated the "nice, uplifting way to still get this information" and wanted "to get a mood from the factual text." Humorous transformations were noted as "more engaging to read."

Custom chat transformations valued

Seven participants found custom transformations through the chat interface very useful. Users considered "creating a custom transformation that adapts exactly to my situation very exciting" and found "custom transformations fun."

Four participants ignored playful types after brief exploration. Users dismissed emoji-based transformations as "gimmicky" and two noted they "didn't prefer humorous tone" after initial testing.

Use Motivations Six participants mentioned personal situational adaptation as a key motivation. Users expressed interest in being able to "describe their situation or goal and then have entire books adapted to their own situation" and appreciated reading from different perspectives, such as "from the university perspective."

Situational adaptation

Five participants cited understanding improvement as a primary goal. Users believed "other perspectives help" and that transformations aid "text comprehension" by making "sentences smaller."

Improvement of understanding

Four participants stated faster reading as an objective. Users aimed to "absorb knowledge as efficiently as possible" and reported "getting through the text faster" with transformations.

Faster reading

Two participants mentioned fluent reading and efficiency improvements as desired outcomes. Users set goals to "achieve higher reading flow" and maximize knowledge absorption efficiency.

Fluent reading and efficiency

One participant mentioned skill building through complexity increase, noting interest in transformations "that make the text somewhat more difficult" for practice purposes.

Skill building through complexity

Feature Requests and Enhancement Suggestions Seven participants requested chat functionality with the book, including "conversation with the author about the book," explanations of terms, and dialogue capabilities with the content.

A conversation with the author

Five participants requested text summaries and person-

Personalized summaries

alized summaries. Users wanted "summaries" and envisioned "reading the book top-down instead of bottom-up through summaries." Personalized summaries "like a letter from the author to myself" were particularly desired.

Advanced settings

Four participants requested improved speed and saved settings. Users wanted "presets where you can save multiple transformations as default settings" and "save standard settings." Faster processing was consistently mentioned as important.

Semantic filtering or search

Three participants requested filter or semantic search functionality. Users wanted to "pre-filter the book for my situation" and emphasized that "a search function that can also search semantically is most important."

One participant requested free text fields to define transformations, providing more direct input methods.

One participant requested paragraph key points preview, suggesting that "a core statement of the paragraph is presented in advance."

One participant requested image generation for paragraphs to support visualization of content.

Trust and Concern Expressions

Need to verify original vs transformed

Information Integrity Concerns Nine participants demonstrated need to verify original versus transformed content. Users emphasized that "you always have to validate whether the transformation is as you would like it" and showed consistent checking behavior.

Showed loss of trust

Five participants showed loss of trust in transformed content. Users became "uncertain whether the text is still correct" and "wouldn't trust it with important texts that information might be lost." Trust concerns affected willingness to use transformations for critical reading.

Emphasized critical importance of no information loss

Five participants emphasized the critical importance of no information loss. Users stressed that "it's important to me that no facts were changed" and worried about "things being added" to the original text. Maintaining factual accuracy was a primary concern.

Four participants stated that trust dependency would affect non-fiction use. Users noted they "have to believe what it writes" and acknowledged that "trust wouldn't be quite there" for important texts.

Two participants noted that adjective additions felt like meaning shifts. Users observed that "adjectives were added that change the text in meaning" and felt transformations "lost the meaning" and "changed the taste" of the original.

Skill Development Concerns One participant expressed fear of losing technical language skills, worrying that "their own language use would be lost" through consistent use of simplified transformations.

General Observations

Overall Impressions Seven participants showed general amazement with the reader functionality. Users found it "cool" and were impressed "that it worked so well." The interface was described as "very clean" and "very intuitive to use." Users noted the experience as "interesting and somehow new" and expressed surprise "that you could already lay the styles so accurately."

Overall amazement

Five participants recognized original text complexity issues, noting that texts like those used in the study already had manageable complexity levels, yet transformations still provided value.

Four participants noted that experiences were text-dependent and transformation-dependent. Users observed that results varied "depending on the transformation, whether you can read it faster/slower" and "whether you trusted the whole thing depended on the transformation."

Results vary by transformation

Overlay misalignment, page reflows, transformation failures, display bugs

Prototype Technical Issues In three cases, the feedback overlay sometimes did not align accurately with the corresponding paragraphs and appeared offset. One participant observed unexpected page reflows when the window was resized after the reader had loaded. In another case, during a demonstration of a custom bullet point transformation, the transformation did not visibly apply. For one book extract changing to dark mode or adjusting font size/family did not work properly.

5.2.3 Post-Interaction Questionnaire

The post-interaction questionnaire collected structured feedback, including responses using a 5-point scale (1 = strongly negative, 5 = strongly positive).

Positive feedback on reading speed and comprehension

Reading Impact 82% of participants indicated that reading with transformations felt faster or much faster, and 90% found the text easier or much easier to understand. Reading flow was also rated by 90% as smooth or much smoother. Long passages were seen as manageable or more manageable by 60%, and 54% rated finding key information in the text as easier or much easier. In multiple-choice responses, participants most often selected complex terms, dense content, and sentence structure as areas where the transformations were helpful.

Interface usability positive, combined transformations somewhat predictable

Design and Usability All participants reported that the available transformations were easy or very easy to find, and 82% found it easy or very easy to identify the right one. 82% rated the transformation controls as appropriate or very appropriate, and all found the menu layout effective or very effective. On/Off transformations were most frequently chosen as the most natural to use; 90% said they were satisfied or very satisfied with the control. Only 36% rated using multiple transformations at the same time as smooth or very smooth (all others were indifferent). 90% of participants found the transformation effects predictable or somewhat predictable and also found the changes clear or very clear. Additionally, 90% found creating custom transformations via the chatbot intuitive or very intuitive.

Technical Performance Some participants reported technical issues, including slow system response and transformation errors. 55% agreed or strongly agreed that transformations were applied quickly, 36% were indifferent, and 11% strongly disagreed. 45% agreed that multiple transformations were applied smoothly, 45% were indifferent, and 9% disagreed.

Technical issues and moderate performance ratings

Future Usage and Improvements Participants indicated interest in using the tool for reading scientific texts, textbooks, non-fiction works, and foreign-language material. Reported challenges included identifying suitable transformations, managing several at once, and creating new ones. Suggestions for improvement included adding summarization features and support for unfamiliar terms. 73% said they would use or definitely use the tool in the future, with the rest saying "maybe" Similarly, 73% said it compares better or much better to other reading tools they have used, while the others rated it about the same.

Positive interest despite challenges, suggestions for improvement

System Usability Scale Score The system received a SUS score of 87, which Bangor et al. [2008] classify as 'excellent,' within the acceptable range and top quartile based on a decade of SUS data. Figure 6.1 shows the score in comparison to a benchmark of iPhone and iPad apps from Kortum and Sorber [2015].

5.2.4 Interaction Log Data

Book Selection, Reading Patterns, and Transformation Latency All book extracts were selected at least once. The extract from the finance category was chosen most frequently, followed by religion, technology, and the remaining categories. Four out of eleven participants viewed two different books, while the rest interacted with only one.

Finance extracts most popular, 35-minute average reading time, 27-second transformation latency

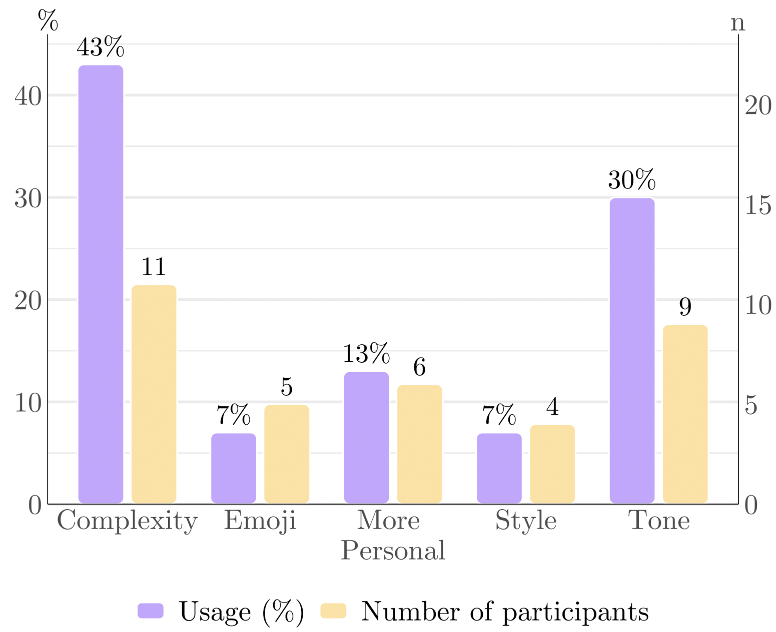

Figure 5.1: Distribution of predefined transformation usage by dimension. The chart illustrates the percentage share (purple) indicating how frequently each transformation appeared in participant requests that included predefined transformations, and the number of distinct participants who applied each transformation (yellow).

Participants spent an average of 35 minutes reading (range: 19–63 minutes), with an average of 50 seconds spent per page. The average number of page turns was 51 (range: 15–79). The average latency for transformations to take effect was 27 seconds, with a minimum of 12.4 seconds and a maximum of 40 seconds.

Transformation Usage Participants utilized all predefined transformations to varying extents (see Figure 5.1). Complexity was most frequently used, followed by Tone, More Personal, Emoji, and Style transformations. On average, participants used 5 different transformations (range: 3–7), including custom.

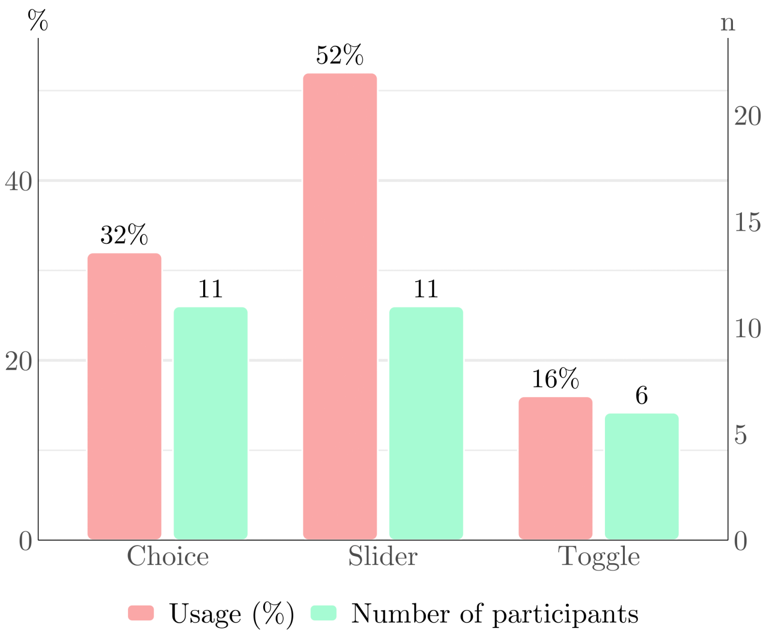

Figure 5.2: Distribution of transformation usage by control type. The chart shows the percentage share (red) indicating how frequently each transformation type appeared in participant requests, and the number of participants who applied each transformation type (green).

Transformation Control Types and Value Preferences Slider-type transformations accounted for the majority of usage, followed by choice-type and toggle-type transformations (see Figure 5.2).

Among choice-type transformations, most options were selected with similar frequency, except "Humorous," which was explored by 9 participants.

For predefined sliders (*Complexity* and *More Personal*), 35% of participants started with extreme values. For custom sliders, this occurred in 83% of cases.

Custom Transformations Participants created 19 custom transformations, with a near-even distribution across control types: toggle transformations were the most frequent

(7), followed by sliders (6) and choices (6). These can be grouped into the following categories:

- **Simplification and accessibility (6):** reducing sentence complexity, adjusting vocabulary, or adapting text for younger readers.
- **Stylistic and tonal changes (5):** academic, lyrical, journalistic, or humorous expression.
- **Perspective shifts (5):** adapting the text to reflect a specific context or to align with the perspective of a particular persona, such as emphasizing financial implications or providing historical framing.
- **Structural and visual formatting (3):** formatting text as lists or visualizing numerical information.

On average, participants sent five chat messages to create a custom transformation. The minimum number of messages sent was two, and the maximum was fourteen. This suggests an average of approximately three messages were required to define a transformation.

Most transformations used in isolation

Use of Single vs. Combined Transformations Six participants combined multiple transformations during interaction, which accounted for approximately 24% of the total transformation usage. In 76% of cases, transformations were used individually. Simultaneous activation of two transformations occurred in 15% of cases, three in 3%, four in 4%, and five or more in 2%. The most common transformation pairs were Complexity + Tone (14%), Tone + More Personal (10.5%), and Complexity + More Personal (10.5%). Across all combined usages, Complexity was involved in 43.8% of pairs, Tone in 42.1%, and More Personal in 24.5%.

Chapter 6

Discussion

The following chapter discusses the results of the preceding study in relation to the research questions. Additionally, the possible engagement of analog readers, prototype issues and limitations are discussed. For this, data triangulation is employed across surveys, interaction logs, observations, and interviews, following established methodological triangulation principles [Jick, 1979; Denzin, 1978].

6.1 Accessibility Impact (RQ1)

As introduced in Section 2.1, Text Accessibility is described in this thesis as an umbrella term encompassing the diverse factors that enable or hinder an individual's ability to approach, attain, or access a text. In the following it is discussed, which observations or subjective expressions might be assigned either to the "Enable" or "Hinder" category. For the latter, an attempt is also made to determine whether these barriers can be resolved.

6.1.1 Enable

Direct Effects

Subjective improvements in comprehension, reading flow and motivation

Direct effects indicate positive effects but only as subjective measures. Participants reported that custom transformations made it easier and simpler to understand the text. They also reported improved reading flow through simplification or tonal change to "humorous" and showed preference for transformed versions. Several participants articulated an increased motivation to read the book in a humorous version and noted that transformations removed the need for reading aids or skipping of difficult content. Many participants stated as a motivation to use the system that it improves "understanding".

Ratings confim subjective improvements

In the post-questionnaire, 82% of participants indicated that reading with transformations felt faster or much faster, 90% found the text easier to understand, and 90% rated the reading flow as smooth or much smoother. Long passages were seen as manageable or more manageable by 60% of participants, and 54% found key information easier to locate.

Indirect Effects

Positive emotional responses suggest engagement benefits but lack clear accessibility validation

These could indicate an "enable effect" but could also be neutral and have no impact. Participants showed positive emotion to read texts from different perspectives and displayed laughter, vocal engagement and amusement during reading. They also remarked that transformations made reading faster. However, while emotions, laughter, and vocal engagement might help individuals to engage with a text, it remains unclear whether this translates to improved text accessibility, and whether when somebody states "it made me faster" this also let the participant attain the text the same way as when they would read slower.

6.1.2 Hinder

Resolvable Barriers

Several barriers were identified that could potentially be resolved through better design or technical improvements. Confusion between versions and perceptions of oversimplification—where some participants found the transformed text too easy and said they would not read it—could both be mitigated through better communication and design (see Section 6.2). Participants who were not satisfied with the outcome as it differed from their expectations could benefit from improvement of LLM models. Some participants reported that text became harder to understand, which might be addressed through LLM improvement or also better design.

Version confusion, deviating outcomes and oversimplification barriers

All other issues discussed in Section 6.2 including mismatched mental models or lack of transparency—likely contributed to confusion and may have hindered attainment of the text. However, the section also outlines potential ways to address or mitigate these problems.

All issues discussed in Design Implications may hinder text attainment

Unresolvable Barriers

Certain barriers appear to be unresolvable. Trust concerns and the resulting need to verify original versus transformed content represent fundamental challenges. While a good transparency mechanism might mitigate these concerns, it does not eliminate the need for verification itself, and this overhead remains a barrier. Additionally, meaning shifts resulting from LLM hallucinations can be reduced with better models but not entirely removed (see explanation in Section 6.2.13).

Trust concerns and LLM hallucinations represent fundamental barriers

6.2 Implications for Design (RQ2)

This discussion aims to provide implications that should

Design implications offer interpretive guidance while acknowledging multiple valid conclusions not be taken as absolute, but rather as one possible interpretation; readers are encouraged to draw their own conclusions and models from the findings (following Dourish [2006]). Likewise, the example adaptations illustrate possibilities based on the current prototype and are not intended as definitive solutions. Both the implications and prototype adaptations were concluded by addressing triangulated study findings through core usability principles, as outlined by Norman [2013]—including affordances, signifiers, feedback, constraints, mapping, mental models, and visibility of system status. Many of these implications may also be relevant to the design of other interactive systems that integrate LLMs.

In the following, With "Adaptation" it is always referred to exemplary prototype adaptations that demonstrate how the implication could be addressed. Implications progress from communication and interface foundations, through usability and user control considerations, to trust-building and advanced interaction capabilities, concluding with technical constraints.

6.2.1 Communicate

Indicate Activity Status

Indicate activity status clearly to users. *Reason:* Participants assume instant transformation but were able to understand that the system is processing. *Adaptation:* See "Convey Background Processing" below.

Indicate Version Status

Indicate version status (original/transformed) very clearly. *Reason:* Participants were confused whether original or transformed version was shown. *Adaptation:* Indicate within the content directly the status as the toolbar alone was not clear enough.

Clear Interface Elements

Interface clarity prevents user confusion

Avoid interface elements or a setup of interface elements that cannot clearly communicate what they mean, manipulate or how they instruct the AI, and avoid options that are too close to each other. *Reason:* Toggles and Choice Control were found intuitive, reliable and perceived trustworthy. Sliders that could not communicate in the way they were setup what they do caused confusion and ambiguity. As a result, participants also tended to try extreme values (backed by observations and event data). *Adaptation:* See next two sections.

Default to Broad Changes, Allow Refinement

Reason: Participants sometimes could not see a difference between different slider settings and were confused. As broad manipulation is set up as default, the change of state is easier communicated and fine adjustment should still be allowed later. *Adaptation:* For the slider control have broad steps as default and some way to additionally adjust finer when needed.

Reference to Current State

Users want to adjust relative to the current state

Give users a reference to current state or allow relative manipulation. *Reason:* Many participants asked where the "original" text is on a slider or wanted to change the text relative to the original, indicating that the implementation decision (see Section 4.5.3) to avoid LLM-based assessment of original text created a usability gap. *Adaptation:* The prototype adaptation could either classify the original text using the LLM and indicate its position on a slider or choice control, or rework the controls to communicate relative adjustments—reflected in the LLM prompt as instructions like "Transform the text to be more or less...".

Aim for Simplicity

Aim for simplicity and decompose complex controls. *Reason:* Participants were confused when a slider included multiple reference points with different meanings, which also led to options being so close together that they became difficult to distinguish. *Adaptation:* Decompose sliders with multiple reference points to multiple different transformations.

6.2.2 Show Transitions

Use some form of transition to indicate how the text reflows or changes. *Reason:* Participants used the visual overlay as orientation for how the text changes in its shape. *Adaptation:* Conceptually not necessary for the current prototype.

6.2.3 Imitate

Familiar interface improves usability

Imitate familiar reader designs, features, and chatbot interfaces. *Reason:* Participants found the chat interface easy to use, likely due to familiarity with services like ChatGPT. The SUS scores suggest the overall interface was perceived as acceptable—comparable to or better than top-rated iPad or iPhone apps (see Figure 6.1). While the sample size was small (N=11), it is generally easier to detect usability issues than to confirm strong usability, so these results should be interpreted with caution [Lewis, 1996; Bangor et al., 2008]. Post-survey responses also revealed varied preferences in visual settings and highlighted a need for standard reader features and controls. *Adaptation:* Integrate additional common reader functionalities, such as term lookup.

Comparison of MagicTextreader's SUS Score with Benchmark Application Distributions

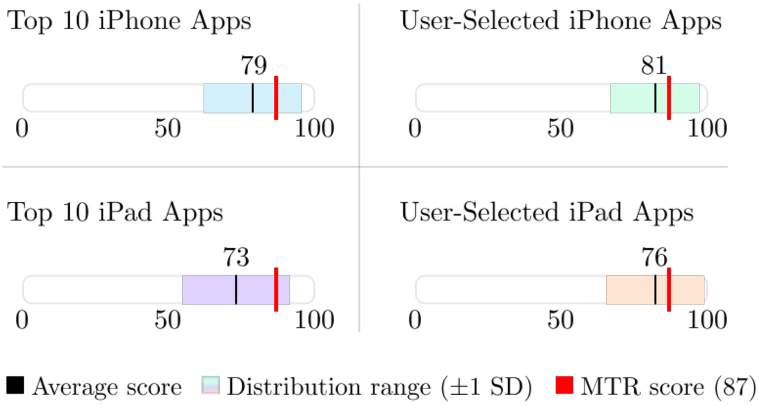

Figure 6.1: MagicTextreader's (MTR) SUS score [Brooke et al., 1996] compared to benchmark distributions from Kortum and Sorber [2015] survey of iPhone and iPad app usability ratings, showing performance relative to top-rated and user-selected applications.

6.2.4 Avoid Bias in All Areas of a Research Toolkit

Order bias affects interface element usage beyond surveys alone

Biases—such as the order of interface elements—are not limited to surveys (see Section 3.8.1). *Reason:* Interaction logs showed that the complexity transformation was used most frequently. In interviews, participants reported starting with it simply because it was the first option presented. *Adaptation:* Display transformation options in a randomized order for each participant to reduce order bias in a study setup.

6.2.5 Reduce Cognitive Load

Infer User Intent

Infer what users are trying to express and adapt accordingly. *Reason:* Participants often found it cognitively demanding to articulate their intent clearly. *Adaptation:* No immediate changes needed—participants were often positively surprised by the outcomes, even when they criticized their own input.

Suggest & Demonstrate

Provide suggestions and demonstrate interaction patterns. *Reason:* Participants found it cognitively demanding to formulate requests in the chat. *Adaptation:* Offer example prompts and interaction suggestions to guide and inspire users.

Support a Trial-and-Error Approach

Reason: Participants expected immediate results when interacting with the chat and found it cumbersome to answer multiple follow-up questions. *Adaptation:* Generate a transformation based on the initial user input without requiring excessive detail. Take user decisions by default, while allowing revisions if needed.

Respect Situated Intent, Enable Generalization

Transformation requests are rather contextually than abstractly formulated

Allow users to formulate requests based on the immediate context, rather than expecting abstract or general instructions from the outset. *Reason:* Participants expressed their transformation wishes in the chat in relation to the currently open book content—reflecting a context-driven mental model. *Adaptation:* The prototype already supported such situated input implicitly; this behavior should be communicated explicitly. Additionally, the system could offer a shortcut to generalize these transformations if users wish to apply them more broadly.

6.2.6 Enforce Simplicity & Logical Use

Restrict to Simple Uses

Limit simultaneous transformations for clarity

Restrict functionality to simple and logical use cases. *Reason:* Applying too many transformations simultaneously can lead to illogical or unclear outcomes. Participants occasionally failed to notice that multiple transformations were active. *Adaptation:* Limit users to two simultaneous transformations. If they attempt to exceed this limit, prompt them to disable one or switch to expert mode. Event data shows that two transformations cover 91% of use cases.

Ensure Logical Order

Ensure a logical order in the application of transformations. *Reason:* The sequence in which transformations are applied can affect the outcome, and some users had differing expectations based on the order. *Adaptation:* Prompt the LLM to determine a reasonable order automatically, or allow manual control in expert mode (see below).

Provide Expert Mode for Advanced Users

Reason: While restrictions help prevent confusion, some users may want full control—event data shows cases of five or more transformations used simultaneously. *Adaptation:* Include an expert panel that allows users to define the order of transformations and unlock an unlimited number of transformations.

6.2.7 Put Users in Control

Enable on-demand use by allowing users to choose which parts of the text should be transformed. *Reason:* Some participants expected to select specific areas for transformation, or expressed a desire to transform only certain parts of the text rather than the entire content. *Adaptation:* Provide the option to manually select a text segment, triggering a contextual popup that offers to apply the transformation only to the selected portion.

6.2.8 Make Use of Natural Language Interaction

Allow users to formulate their requests in natural language. *Reason:* Participants expressed strong satisfaction and emotional engagement when interacting in natural language and using their own transformations. This approach supports personalization and aligns well with the way LLMs process input (see Section 3.4); additionally, user-generated transformations can be further refined (see Section 3.4.2). *Adaptation:* Not necessary for the current prototype.

6.2.9 Convey "Background" Processing

Interface should communicate background AI processing to manage user expectations

Make it clear that AI-driven transformations occur in the background, and avoid reinforcing the mental model of a quick-access menu. This can be complemented by offering the ability to save transformation settings for later use. *Reason:* Participants often expected immediate application of transformations, assuming quick access or instant results—despite LLM latency (see Section 3.4.4). Additionally, many participants requested the option to save their transformation settings for reuse. *Adaptation:* Require users to intentionally open a dedicated configuration menu, reinforcing the idea of setup rather than instant toggles. Within the menu, offer a small preview of the transformation. Once the user presses "Apply," clearly indicate that background processing has started and that they will be notified upon completion, preserving reading flow. When the transformation is ready, inform the user and, upon confirmation, use a gradual transition to help them track changes and maintain their reading position.

6.2.10 Build Trust Through Transparency

Atomic Control Mechanism

Links transformed sentences to original sources for verification

Enable an atomic control mechanism that lets users trace each transformed sentence back to its original source sen-

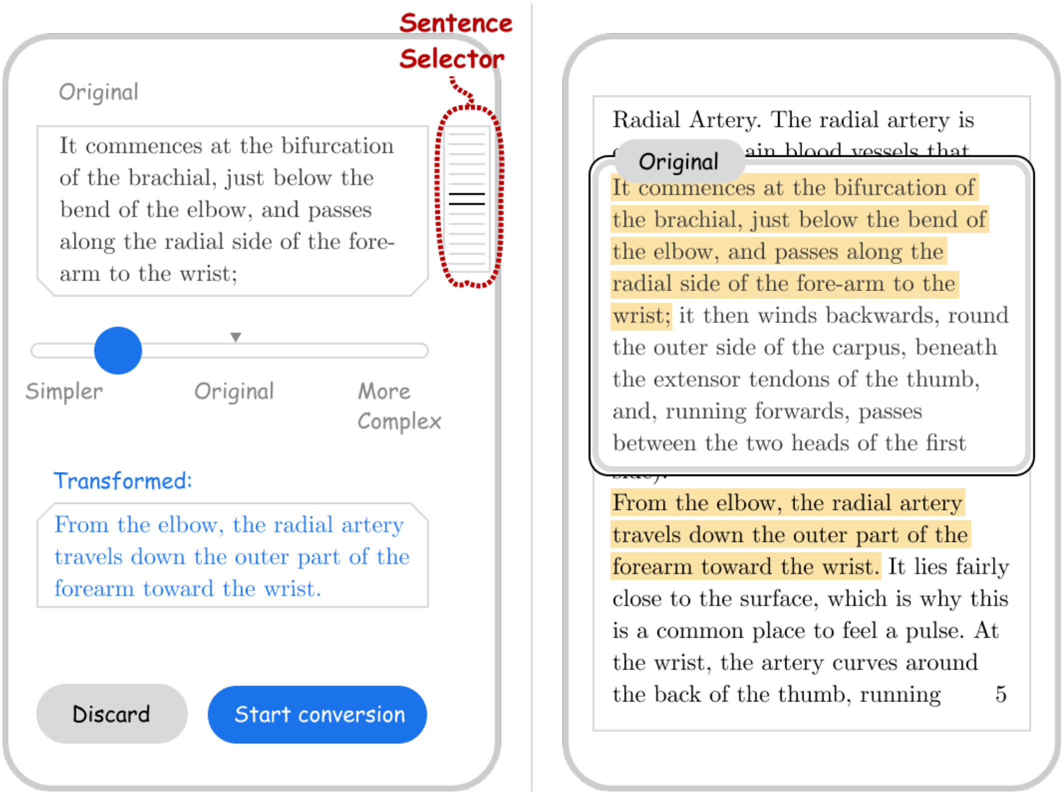

Figure 6.2: Left: Shows an exemplary prototype adaptation that enables quick exploration through a sentence-level transformation preview. One sentence from the current page is selected—based on visibility or LLM selection—and displayed in both original and transformed form. Users can explore other sentences using the Sentence Selector, adjust complexity relative to the original, and quickly assess the effect before applying it to the full text. Background processing is conveyed by labeling the apply button "Start conversion". The LLM can even be given more text context before and after the sentence for the transformation, as for low response time primarily the output length is important (see Section 3.4.4). When a user then decides to apply the transformation, the previewed sentence is enforced to become part of the full transformed version of the text, so it can already serve as a reference for how the reading position might have changed. While the transformation loads, the reader can continue reading the original version. Once the new version is ready, they are notified and prompted to confirm its application. Upon confirmation, a gradual transition may be used to help visualize how the text reflows or maps. Right: Illustrates a prototype adaptation enabling an atomic control mechanism. When a transformed sentence is selected, a contextual popup displays its original counterpart, showing the corresponding sentence highlighted while maintaining access to surrounding content. The popup is scrollable, repositionable, and resizable to avoid occlusion. When the user flips pages while the comparison popup is open, it automatically reappears: showing the corresponding original version of the last sentence on the new page when flipping back, or the first sentence on the new page when flipping forward. (Excerpts from Gray et al. [1901].)

tence(s). *Reason:* Participants expressed a loss of trust in the transformed content and emphasized the importance of preserving information integrity. Many indicated they would only use the tool if they had a way to verify that no information was lost. *Adaptation:* Assign (invisible) labels to all sentences in the book's XML. Instruct the LLM to propagate these labels to the corresponding transformed sentences, supporting one-to-many and many-to-one mappings. Use these labels to link transformed sentences to their source(s), allowing users to view the original sentence(s) on demand.

Side-by-Side View

Side-by-side comparison view enables overall verification

Provide a side-by-side view for comparing the original and transformed versions of the text. *Reason:* Participants frequently engaged in comparison behavior and expressed the need to verify how the transformed text differed from the original. Similar to the atomic control mechanism, this feature could help rebuild trust in the transformed content by allowing easy verification. It may also reduce concerns about potential information loss. *Adaptation:* The screen could be split down the middle, with each side hosting an independent reader view (original on one side, transformed on the other). Synchronization between the views—such as scroll position or highlighted content—should be automatically performed. Alternatively, a sliding mechanism could be introduced to switch between versions, potentially using a vertical (up-/down) gesture to avoid conflict with horizontal page navigation, which aligns with the conceptual space illustrated in Figure 4.7.

Quick Exploration

Preview mechanism reduces uncertainty about transformation effects through sample outputs

Enable quick exploration through small example outputs. *Reason:* Participants often struggled to understand what a transformation or option would do. Due to the unpredictable nature of LLMs (see Section 3.4.3 and Section 3.4.5), and the latency introduced by longer outputs (see Section 3.4.4), it can be difficult to explore or interpret options effectively. *Adaptation:* When a user adjusts transformation settings, show a small popup that previews the effect on a single sentence—ideally one from the current page. The surrounding context can still be included, as latency mainly depends on the number of words generated (see Section 3.4.4). This preview could also be integrated directly into the chat interface, where new transformations immediately produce visible effects.

Make Transformation Instructions Visible

Display the exact transformation instructions in a clear and readable format. *Reason:* Participants were often unsure how specific options affected the transformation. Presenting the underlying instructions in an understandable way may reduce this uncertainty. *Adaptation:* Show a concise, human-readable version of the core instruction—e.g., as an overlay or tooltip—so users can easily see what guidance the LLM will follow.

Show Combination Model

Visual pipeline display clarifies transformation combination and sequencing

Display the active transformation model and the number of transformations currently applied. *Reason:* Some participants were unsure how transformations were combined or in what order they were applied, leading to different expectations about the outcome. Others were unaware that multiple transformations were active at the same time. *Adaptation:* Building on the idea of making exact transformation instructions visible, the overlay could be enhanced to show a pipeline as a flow diagram, clearly illustrating the sequence and combination of applied transformations.

Introduce Orientation Cues

Orientation cues help users maintain reading position during transformations

Provide cues that help users maintain orientation and reidentify their position in the text. *Reason:* Participants often lost their reading position and were confused by jumps or reflows, especially after transformations. Some relied on jumping footnotes as informal anchors. The flexible reflow layout of EPUBs, while generally beneficial (see Section 2.2.3), becomes a drawback when orientation cues are missing. *Adaptation:* Introduce subtle, consistent markers that appear in both the original and transformed versions—such as small reference numbers combining paragraph and page counts—to help users track their location and regain context after changes.

6.2.11 Alert

Encourage critical review as early error discovery has impact on trust and satisfaction

Encourage users to critically assess the outcomes of transformations. *Reason:* LLMs tend to hallucinate (see Section 3.4.3). While many participants expressed concerns about information integrity and trust in the transformed text, others did not explicitly consider this—even when suggesting use cases in professional or contexts where accuracy is crucial (see Section 3.5). As reported in the same section, users who discover inaccuracies early tend to report significantly higher trust and satisfaction than those who encounter them later. *Adaptation:* Display a warning the first time a transformation is applied or when users return to the reader after a break. Additionally, clearly and persistently indicate whether the content is original or transformed like discussed before.

6.2.12 Allow Context/Resource-Aware Free LLM Interaction

Context-aware chat enables book interaction and addresses many participant requests

Provide users with an optional general chat feature that automatically includes relevant context—such as the current book, reading position, or selected segments—or allows them to easily select context manually. Additional information, like tool usage instructions or an FAQ, can be appended to the context, addressing issues raised by participants who still had questions (e.g., how to apply transformations) even after a demonstration. This also accounts for uncertainty about how comfortable users would have been without direct explanation. *Reason:* Several participant requests—such as chatting with the book, top-down summary-based exploration, or term explanations (the latter even being expected)—can be supported with minimal additional interface complexity or development effort. *Adaptation:* Alongside the transformation chat, offer an additional or unified chat interface where the LLM receives the current book section or segment as context. Alternatively, implement dynamic context loading via function calls, enabling the LLM to fetch relevant content on demand and delegate accordingly. (Providing the entire book as input is avoided due to LLM response time constraints; see Section 3.4.4.)

6.2.13 Handle Constraints Pragmatically

Improve Perceived Speed

Perceived speed can be improved through prioritization

Improve the perceived speed by processing the smallest and most relevant chunks independently, or by dynamically adjusting chunk size based on real conditions. *Reason:* Participants consistently highlighted speed as crucial. However, they also accepted some initial delay as part of the workflow—especially since event data showed they spent an average of 50 seconds per page, which is longer than the typical LLM response time. Notably, LLM response time depends primarily on output length (see Section 3.4.4). *Adaptation:* Reprocess the EPUB dynamically, based on the current screen size and font, as soon as the reader opens the view. Choose segment sizes that closely match what is actually visible on the screen. Rather than processing the current, left, and right segments together, begin with only the currently visible segment to prevent layout reflow. Then, in the background, independently process the right segment (anticipating forward navigation), followed by the left segment.

Accept LLM Limitations

Accept some error in LLM systems; focus on mitigation

Acknowledge hallucinations as an inherent limitation of LLMs, and avoid attempting to eliminate them at all costs. *Reason:* Due to their non-determinism, it is inherently difficult to predict or guarantee what output an LLM will produce for a given instruction (see Section 3.4.5). During the development of the MagicTextReader, several new LLMs were released that showed improved performance on hallucination benchmarks, as well as speed gains through smaller and faster models (see Chatbot Arena benchmark [Chiang et al., 2024]). Investing heavily in countering hallucinations at all costs may be rendered obsolete by such ongoing model improvements. Instead, rely on mechanisms described under "Alert and Transparency" to mitigate risks—these may become less critical over time, but are unlikely to become entirely unnecessary. As no algorithm can solve the Halting Problem [Turing et al., 1936], a perfectly reliable AI—which is ultimately just an algorithm running on a Turing-complete system—cannot exist for all possible inputs without also solving that problem. Therefore, some level of error or uncertainty will always remain. *Adaptation:* Plan for future replacement of LLM models with more capable and efficient alternatives as they become available.

6.3 Future Research Directions, Application Context (RQ3)

To guide future research and potential application contexts, the following section discusses possible focal points: the types of texts studies could target, the kinds of transformations to prioritize, and suggested next steps for gaining deeper insights.

6.3.1 Suitable Text Types for Transformation (As Indicated by Participants)

Future research could further explore the use of text transformation in non-fiction contexts—including scientific publications, educational content, foreign language texts, and newspaper articles—as these were commonly mentioned by participants. Non-fiction presents facts or information [Cambridge Dictionary, 2025], making it a promising area for transformation without compromising the core intent of the text.

Non-fiction texts show promise for transformation applications

In contrast, applying transformations to fiction should be approached with caution. Around half of the participants explicitly rejected such use, arguing that it would distort the artistic nature of fiction. They emphasized the importance of preserving the author's writing style, which they saw as integral to the work itself. One participant did acknowledge potential value in using transformations as supportive aids—such as helping readers follow the storyline or track characters—but not for altering the narrative itself. Given that fiction heavily depends on tone, form, and authorial intent [Currie, 1985], transformations risk undermining its meaning.

Fiction transformations require caution due to artistic integrity concerns

6.3.2 Promising Transformation Types (As Indicated by Participants)

Future work may further explore transformations related to complexity, tone, and personalization, as participants expressed a range of preferences and reactions that suggest promising directions.

Predefined Transformations

Complexity The complexity transformation appeared to have a positive impact on reading for several participants. It was rated as the most useful transformation, frequently mentioned as a learning or skill-building opportunity, and

Complexity transformation showed strongest positive impact and highest usage generated the highest number of transformation requests (according to event data). However, as noted in the design implications, order bias may have influenced this outcome. This area also aligns with existing research on text simplification (see Section 3.3).

Tone increased engagement and showed strong responses

Tone The humorous tone transformation also showed potential benefits. It appeared to improve reading flow and elicited strong responses such as laughing, reading aloud, and increased engagement. It was the second most used transformation, though its position in the toolbar may also have influenced this. Participants described it as exciting and particularly helpful for making otherwise boring content more enjoyable.

Other Transformations Transformations involving emojis, stylistic changes, or making the text more personal generally received neutral or negative feedback. Participants either showed no preference or actively rejected these options.

Custom Transformations (Created via Chat)

Custom transformations caused strong positive emotional responses and promise diverse personalization dimensions

Custom transformations created through chat interactions were associated with strong positive emotional responses. Most participants found them useful, interesting, and appreciated the sense of personalization. Many envisioned use cases involving personal or situational adaptation—such as tailoring or filtering an entire book to fit their individual context. Event data reflected a wide range of personalization dimensions explored by participants.

6.3.3 Suggested Next Steps

Refining and Evaluating Prototype Adaptations

Trust and content integrity issues require further verification and benchmarking

Further evaluation of the discussed design implications and prototype adaptations is recommended, particularly since issues of trust and content integrity were central in participant feedback and are not yet sufficiently addressed. Incorporating a benchmark for hallucinations in transformations within the current setup could support risk estimation, as, according to Renn [1989], gaining or maintaining trust is one of the most frequently mentioned objectives of risk communication.

Field Studies with the Toolkit

Future studies using the improved toolkit—potentially beyond HCI-focused research and involving domains concerned with text accessibility—could include long-term, ethnographic investigations. These might surface different insights than controlled lab studies, particularly around how interactions evolve over time or how novelty effects and learning curves are overcome [Kjærup et al., 2021].

Assessing Impact on Text Accessibility

To better understand the tool's actual impact on accessibility, studies should examine the trade-off between the added overhead of comparison or transparency features and the benefits of transformation (e.g., changes in complexity, tone, or customization) relative to the original text. This could involve controlled comprehension tests, assessments of reading speed and flow, and comprehension rate analysis. Currently, findings are indicative rather than conclusive, and more rigorous evidence is needed to determine significance.

Future studies should measure accessibility benefits against transformation overhead costs

Target Text Types and Audiences for Evaluation

Recommended text domains for further evaluation include those mentioned by participants (e.g., general audience texts and educational material for children), as well as specialized but often inaccessible domains such as health and

Future evaluation should expand text domains and include diverse participant groups legal literature [Curtotti and McCreath, 2013; Michielutte et al., 1992].

Participant samples should inlcude a range of user groups, including the general population, individuals with reading difficulties (who may particularly benefit—see Section 2.1.1), non-readers (who may engage more through tone or personalization features), and children (given the relevance of educational use and the influence of age and education level on accessibility).

6.4 Engagement of Analog Readers

Analog readers showed interest despite digital reading aversion

Participants who indicated a preference for analog reading in the pre-interaction questionnaire still expressed interest in using the MagicTextreader in the future, reflected in the post-interaction responses and also in the semi structured interview. This may suggest a potential to engage users who are not yet familiar with or inclined toward digital reading.

6.5 Prototype Issues

High-specificity inline CSS in books overrides reader styling

During the study, the following prototype issues occurred. For one book extract changing to dark mode or adjusting font size/family did not work properly. A brief investigation showed that some books use inline styles or high-specificity CSS (see W3C standard [Etemad and Jr., 2023]) directly in their XML. These styles override the reader's default styles, which have lower specificity and no !important flag. Style attributes must be applied with higher CSS specificity or marked with `!important` when needed.

Window resizing breaks feedback overlay requiring recalculation

The feedback overlay seems offset after window size changes as the prototype does not properly resize and reposition the paragraph overlays after the window is resized, especially during transition. Window resizes must be tracked and considered in the calculation process and recalculated always when necessary.

API instability forced shift from ethnographic approach to controlled sessions

Before the presented study was conducted, preliminary experiments revealed significant API instability with the OpenAI service, including frequent connection failures and API-related errors that rendered the system nearly unusable. These technical issues led to the decision against pursuing a full ethnographic approach where participants would use the prototype independently, as the system proved too unpredictable for unobserved sessions. An earlier attempt at the session-based study with a small number of participants also failed due to these frequent connection failures and API-related errors, leading to the results being discarded. For future ethnographic setups, implementing robust error handling mechanisms would be essential to manage service interruptions gracefully. It may also be valuable to collect data from the OpenAI platform status page [OpenAI, 2025] to analyze patterns in reliability across weekdays. Automated latency tests could further help infer usage load and performance fluctuations throughout the day. As an alternative, one might consider using a local LLM to eliminate dependency on external APIs. However, running large models locally requires substantial hardware resources [Naveed et al., 2023], which may constrain to smaller models—introducing a tradeoff between reliability and the speed or quality of output [Kang et al., 2025].

LLM self-checking could enable automatic error detection and reprocessing of transformations

The problem of improperly applied transformations could be addressed by leveraging the LLM to self-check its own output—a method previously proposed by Miao et al. [2023] for validating its chain of thought after generation. After a transformation is applied, the LLM could be prompted to review the result and identify any errors. If issues are detected, the system could automatically trigger a reprocessing step based on the error report.

6.6 Limitations

6.6.1 Study Limitations and Generalizability

Homogeneous participant sample limits generalizability to broader user populations<

The generalizability of the findings is limited by the participant sample. The group was relatively homogeneous, with most participants aged between 25 and 34, holding higher education degrees, and having strong familiarity with digital reading tools. This profile does not reflect the full diversity of potential users, particularly in terms of age, educational background, and digital literacy. Additionally, none of the participants reported reading-related conditions, and all participants were already regular readers. As a result, no conclusions can be drawn about the tool's impact on individuals who currently avoid or struggle with reading.

Study environment limitations were balanced by reading patterns and high usability ratings

The controlled study environment may have influenced participant behavior. However, some bias could be mitigated: all book excerpts were selected at least once across participants. Session time averaged 35 minutes, matching participants' typical reading durations (30–60 minutes) as preferences from the post-interaction show, though actual reading time was lower due to prototype interaction. Transformation delays were shorter than average page viewing time, thus only disrupted flow when transformations changed. Participants reported few technical issues and gave the tool usability ratings comparable to top iPhone/iPad apps, indicating good general usability among this participant group (though these results should not be easily generalized beyond the study sample).

6.6.2 Prototype Limitations

PDF conversion enables compatibility

Though the reader can only process EPUB and not PDF, which is widely used for document sharing [Marinai et al., 2011], it can be converted to EPUB even when this can be difficult if the PDF does not mainly contain text [Marinai et al., 2011].

API costs negligible for research but significant for real-world deployment

API costs were not really relevant as per participant on average only 0.07$ of costs occurred for 35 min of use. As participants mostly gave less than 10 hours per week as average reading time in worst case that would accumulate to 63$ per year.

6.6.3 Toolkit Limitations

The toolkit requires hosting on a web server and some technical expertise to set up, which may pose a barrier for non-technical users. It lacks built-in tools for analyzing questionnaire responses and event data—Python scripts were used for this in the current study, which proved cumbersome. Additionally, the localization setup involves many steps.

Toolkit requires technical expertise and lacks built-in analysis tools

Chapter 7

Summary and Outlook

The last chapter presents a summary and an overview of the contributions of this work. The thesis concludes with an outlook on future work that builds on the findings.

7.1 Summary

In this thesis, our primary objective was to investigate a system that integrates AI-powered text transformation capabilities—specifically using LLMs—with a focus on its impact on text accessibility.

To explore this, we developed a research prototype called *MagicTextreader*. The system allows users to apply modular text transformations along various text accessibility dimensions, categorized by their control types: toggles (for binary changes), choices (for discrete options), and sliders (for continuous adjustments between two reference points). Using these modular transformations, many text accessibility dimensions can be addressed—such as complexity, tone, style, format, and personalization. Additionally, a built-in chat interface allows users to define their own transformations via natural language, which are then mapped to one of the three control types.

MagicTextreader prototype was developed that enables modular text transformations through three control types

Prototype uses React frontend and Django backend and also serves as research tookit

The prototype was implemented using a client-server architecture, with a React and `epub.js` frontend for rendering and interaction, and a Django-based backend responsible for data management and communication with the LLM that performs the transformations. *MagicTextreader* is accompanied by a surrounding platform that supports study administration, including session setup, pre- and post-interaction questionnaires, and event logging—making it suitable both as a functional prototype and a research toolkit.

Mixed methods study (N=11) evaluates accessibility impact and design implications

To evaluate the system, we conducted an initial session-based mixed methods study (N=11) focused on non-fiction texts. Participants explored five predefined transformations—complexity, style, tone, more-personal adjustment, and emoji enhancement—and were encouraged to create their own via the chat interface. The study aimed to assess the system's impact on text accessibility, uncover design implications, and identify directions for future research.

Findings indicate transformations may improve accessibility but also introduce barriers

Findings show that certain transformations—especially complexity and tone adjustments, as well as user-created transformations—subjectively improved understanding, reading flow, and motivation for several participants. However, the prototype also introduced new accessibility barriers, particularly confusion due to unclear communication and trust or information integrity concerns, which in turn revealed a lack of transparency mechanisms.

Design implications were discussed that may also be relevant for other LLM-integrated interactive systems

Based on these insights, comprehensive design implications were discussed that address all resolvable barriers identified in the study and conclude that these implications may also benefit other interactive systems integrating LLM capabilities. Finally, several promising directions for future research emerged (see in the outlook below)—particularly focused on non-fiction use cases—while fiction was explicitly rejected as a transformation target by about half of the participants.

7.2 Outlook

The outlook can be derived directly from the research directions discussed in Chapter 6 “Discussion”. Prototype improvements should address the tension between possible accessibility gains from text transformations and trust barriers that create new accessibility challenges. Enhanced transparency mechanisms, integrated hallucination benchmarks, and clearer communication could mitigate these barriers and thus enable more the attainment of texts than hinder.

Prototype should be adapted according to design implications

Regarding transformation types, simplification, tone adjustment, and custom-created transformations represent promising directions for evaluation in non-fiction contexts. These transformations showed indications of impact on reading comprehension, engagement, and personalization based on participant feedback and usage data. Additionally, non-fiction might represent the target domain for future research, with participants supporting applications in scientific publications, educational content, foreign language texts, and newspaper articles. Domain-specific applications in medicine or law may prove valuable, as these fields remain inaccessible to much of the population [Curtotti and McCreath, 2013; Michielutte et al., 1992].

Simplifications, tone adjustments and personalized ones are most promising dimensions

Non-fiction or domain-specific applications may prove valuable

From a methodological perspective, improving the research toolkit or hosting it on a web server to enable researchers without technical skills to conduct studies might be a crucial step for advancing text accessibility research. Furthermore, future research could benefit from longitudinal field studies that extend beyond laboratory settings toward ethnographic approaches. These studies could provide insights into how user interactions evolve over time and how learning curves affect real-world reading contexts [Kjærup et al., 2021].

A full ethnographic replication of the study might reveal additional patterns

To assess accessibility impacts more rigorously, controlled comprehension tests, reading speed measures, and comprehension rate analyses are necessary to move from indicative observations toward robust evidence.

Empirical studies are necessary to evaluate impact

Additionally, broadening participant diversity may be a key research priority. Including general populations, individuals with reading difficulties, non-readers, and children could reveal varying degrees of accessibility impact. The limited sample size in this study (N=11) restricts generalizability and indicates the need for larger, more diverse samples.

Lastly, broader accessibility of text transformation functionality could benefit from addressing platform ecosystem integration. While individual reader-based solutions currently face limitations in providing users with access to e-books themselves due to platform restrictions [Bittar, 2014], the ongoing integration of LLMs into mobile operating systems [Wu et al., 2024] suggests that text transformation capabilities may eventually become native features of digital reading platforms.

Appendix A

Benchmarking OpenAI models

Listing A.1: Benchmarking OpenAI models by measuring response times for a basic token substitution task. It runs two variants: (1) a 3-to-3 task where three input segments are fully transformed, and (2) a 2-to-1 task where only one of two segments is transformed while the other serves as context. Input texts are sampled from *Gray's Anatomy* [Gray et al., 1901].

```
import time
from statistics import mean
from openai import OpenAI

client = OpenAI(api_key="---")
trials = 4

original_text_3 = """
<part segment="1">
THE Veins are the vessels which serve to return the
    blood from the capillaries of the different parts
    of the body to the heart. They consist of two
    distinct sets of vessels, the pulmonary and
    systemic. ... (shortened for appedix) ... </part>
<part segment="2">
and this communication exists between the larger
    trunks as well as between the smaller branches.
    Thus, in the cavity of the cranium, and between
    the veins of the
... (shortened for appedix) ...
</part>
```

```
<part segment="3">
The superficial veins usually have thicker coats than
    the deep veins, and the veins of the lower limb
    are thicker than those of the upper.
... (shortened for appedix) ...
</part>
"""

original_text_2 = """
<part segment="1">
THE Veins are the vessels which serve to return the
... (shortened for appedix) ...
<part segment="2">
and this communication exists between the larger
    trunks as well as between the smaller branches. T
... (shortened for appedix) ...
</part>
"""

# prompt = f"Transform all parts of the following
    text entirely such that every second word is
    reliably and consistently replaced with the word
    'blabla', starting from the second word and
    ensure that no part of the original text is left
    unprocessed, and maintain the original
    punctuation and structure as much as possible.
    DON'T truncate or leave out anything:
    \n\n\"{original_text_3}\""
prompt = f"Transform␣and␣return␣only␣part/segment␣1␣
    of␣the␣following␣text␣such␣that␣every␣second␣word␣
    is␣reliably␣and␣consistently␣replaced␣with␣the␣
    word␣'blabla',␣starting␣from␣the␣second␣word.␣
    DON'T␣truncate␣or␣leave␣out␣anything:␣␣
    \n\n\"{original_text_2}\""

# Models to benchmark
models = [
    "gpt-4o-2024-11-20",
    "gpt-4o-mini",
    "gpt-3.5-turbo"
]

# ==== BENCHMARKING ====
results = []

for model in models:
    latencies = []
    rates = []
```

```
    for _ in range(trials):
        start_time = time.time()
        response = client.chat.completions.create(
            model=model,
            messages=[{"role": "user", "content":
                prompt}]
        )
        end_time = time.time()

        latency = end_time - start_time

        output_text =
            response.choices[0].message.content
        word_count = len(output_text.split())
        rate = word_count / latency

        latencies.append(latency)
        rates.append(rate)

    results.append({
        "Model": model,
        "Trials": trials,
        "Avg Latency (s)": round(mean(latencies), 3),
        "Avg Rate (words/s)": round(mean(rates), 3)
    })

# ==== LATEX OUTPUT ====
print("\nLaTeX Table Output:\n")
print(r"\begin{table}[h!]")
print(r"\centering")
print(r"\begin{tabular}{|l|c|c|c|}")
print(r"\hline")
print(r"\textbf{Model} & \textbf{Trials} &
    \textbf{Avg Latency (s)} & \textbf{Avg Rate
    (words/s)} \\")
print(r"\hline")
for result in results:
    print(f"{result['Model']} & {result['Trials']} &
        {result['Avg Latency (s)']} & {result['Avg
        Rate (words/s)']} \\\\")
print(r"\hline")
print(r"\end{tabular}")
print(r"\caption{Performance on text simplification
    for an average adult}")
print(r"\label{tab:text_simplification_benchmark}")
print(r"\end{table}")
```

Appendix B

Chat Assistant Instruction & Function Definition

Listing B.1: Base instruction for the natural language chat. The instruction was derived following techniques from Section 3.4.2, especially following provider guidelines, such as using markup formats, clearly defined task goals and was further optimized using the model (gpt-4o [OpenAI, 2024]) itself.

```
You are a system designed to help users create and
    configure text transformation. For example, a
    text transformation could be something that
    changes the style, tone or complexity of the text.

A transformation consists of a name, description,
    panel type, icon, activity status, and various
    configuration options. Each transformation has
    one of two panel types:

1. **Slider**: Used when the transformation has a
    range of values, defined by minimum and maximum
    limits, step increments, and a default value.
    Keep the number of steps as reasonable as
    possible and try to use dividers of 100 as step
    size.
2. **Choice**: Used when the transformation has
    predefined states or can simply be toggled
```

```
    on/off. In the latter case the transformation
    only as the single option on.

The most important artefacts are the transformation
    options as they describe with their
    transformation_prompt-field the way how to
    transform the text.

Your task is to gather all necessary information
    through a conversation and dynamically decide the
    appropriate panel type based on user input. Then,
    you must construct the corresponding
    transformation and options.

When designing the transformations:
- For **sliders**, the slider value is used to
    determine the two closest options and their
    transformation_prompts are used to transform the
    text.
- For **choices**, the options and their
    transformation prompts are directly used to
    transform the text. If it's␣a␣simple␣toggle,␣add␣
    a␣single␣"On"␣option.

Once␣the␣information␣is␣collected,␣call␣the␣
    `create_transformation`␣function␣with␣the␣
    gathered␣data␣to␣create␣the␣transformation␣and␣
    its␣options.
```

Listing B.2: Function Definition for the natural language chat. The function defintion was derived following techniques from Section 3.4.2, especially following provider guidelines, such as using markup formats, clearly defined task goals and was further optimized using the model (gpt-4o [OpenAI, 2024]) itself.

```
{
   "type":"function",
   "function":{
      "name":"create_transformation",
      "description":"Create␣a␣transformation␣with␣its␣
          configuration␣and␣options␣based␣on␣user␣
          input.␣Each␣chat␣can␣only␣have␣one␣
          transformation␣-␣feel␣free␣to␣call␣this␣
          function␣again␣anytime␣to␣update␣its␣
          configuration␣and␣options␣with␣new␣values.",
      "strict":true,
      "parameters":{
```

```
"type":"object",
"properties":{
  "name":{
    "type":"string",
    "description":"The␣name␣of␣the␣
        transformation."
  },
  "panel_type":{
    "type":"string",
    "enum":[
      "choice",
      "slider"
    ],
    "description":"The␣type␣of␣panel␣used␣
        for␣the␣transformation:␣'choice'␣
        for␣selections,␣'slider'␣for␣
        adjustable␣ranges."
  },
  "options":{
    "type":"array",
    "description":"Options␣relevant␣for␣
        both␣'choice'␣and␣'slider'␣panel␣
        types.␣For␣sliders,␣these␣describe␣
        the␣closest␣values␣and␣there␣must␣
        be␣one␣option␣per␣step.",
    "items":{
      "type":"object",
      "properties":{
        "label":{
          "type":"string",
          "description":"Label␣for␣the␣
              option.␣In␣case␣of␣'slider'␣
              panel␣types,␣it␣should␣be␣
              rather␣short."
        },
        "value":{
          "type":"number",
          "description":"Value␣associated␣
              with␣the␣option.␣Must␣be␣a␣
              number␣between␣min_value␣
              and␣max_value␣(inclusive)␣
              if␣panel_type␣is␣slider.␣-1␣
              otherwise!"
        },
        "transformation_prompt":{
          "type":"string",
          "description":"This␣string␣is␣
              used␣as␣an␣instruction␣for␣
              the␣transformation.␣It␣
```

```
                    should␣have␣the␣form␣'The␣
                    transformed␣text␣must...'"
               },
               "order":{
                  "type":"integer",
                  "description":"The␣order␣of␣the␣
                     option␣within␣the␣
                     transformation."
               }
            },
            "required":[
               "label",
               "value",
               "transformation_prompt",
               "order"
            ],
            "additionalProperties":false
         }
      },
      "slider_config":{
         "type":"object",
         "description":"Configuration␣for␣
            sliders,␣applicable␣only␣if␣
            panel_type␣is␣'slider'.",
         "properties":{
            "min_value":{
               "type":"number",
               "description":"Minimum␣value␣of␣
                  the␣slider."
            },
            "max_value":{
               "type":"number",
               "description":"Maximum␣value␣of␣
                  the␣slider."
            },
            "step":{
               "type":"number",
               "description":"Increment␣step␣for␣
                  the␣slider."
            },
            "default_value":{
               "type":"number",
               "description":"Default␣value␣of␣
                  the␣slider."
            }
         },
         "required":[
            "min_value",
            "max_value",
```

```
            "step",
            "default_value"
          ],
          "additionalProperties":false
        }
      },
      "required":[
        "name",
        "panel_type",
        "options",
        "slider_config"
      ],
      "additionalProperties":false
    }
  }
}
```

Appendix C

Transformation Instruction

Listing C.1: Transformation Instruction (System Message Infix) – The instruction was derived following techniques from Section 3.4.2, especially following provider guidelines, such as using markup formats, clearly defined task goals and was further optimized using the larger model gpt-4o [OpenAI, 2024] instead of the used model gpt-4o-mini [OpenAI, 2024].

```
Objective: Transform the provided HTML-text strictly
    according to the specified Transformation
    Preferences.

Key Rules:
1. **Retain Exact HTML Structure**: Do not modify,
    correct, or complete any HTML elements,
    attributes, or structure, even if tags appear
    mismatched, incomplete, or non-standard.
2. **Follow Preferences Exactly**: Implement the
    transformations only as described in the
    specified preferences. Do not infer or apply
    additional changes outside of the preferences.
3. **Preserve Content Integrity**: Ensure all
    non-transformed content remains untouched,
    including whitespace and character encoding.
4. **Handle Ambiguities**: If a preference or the
    HTML text is ambiguous, output a warning note
    while preserving the HTML unchanged.
```

```
5. **Output Format**: The output should consist
    solely of the transformed HTML-text, without any
    added explanations or notes.

---

### Input Format:
The input HTML contains sections wrapped in special
    comments with the following structure:
<!-- MagicTextreaderPart id="X" status="A" -->
<content to transform>
<!-- // MagicTextreaderPart -->

---

### Task:
1. Transform only the parts where `status="orig"`.
2. Skip parts marked with `status="trans"` and leave
    them out of the output.
3. Return the modified HTML for transformed parts
    only, in the same commented structure:
<!-- MagicTextreaderPart id="X" status="trans" -->
<transformed content>
<!-- // MagicTextreaderPart -->

---

#### Example Input:
<!-- MagicTextreaderPart id="1" status="orig" -->
<....>
<!-- // MagicTextreaderPart -->
<!-- MagicTextreaderPart id="2" status="trans" -->
<....>
<!-- // MagicTextreaderPart -->
<!-- MagicTextreaderPart id="3" status="orig" -->
<....>
<!-- // MagicTextreaderPart -->

---

### Example Output:
<!-- MagicTextreaderPart id="1" status="trans" -->
<....>
<!-- // MagicTextreaderPart -->
<!-- MagicTextreaderPart id="3" status="trans" -->
<....>
<!-- // MagicTextreaderPart -->
```

```
So do not include parts marked with ‘status="trans"‘
    in the output.

---

### Transformation Preferences:
For parts of the HTML marked with ‘status="orig"‘,
    apply the following transformations:
**Preserve HTML Structure**: Do not modify the HTML
    tags or structure. Only change the textual
    content within the tags.
```

Appendix D

Pre-Interaction Questionnaire

The following questionnaire was used to collect structured data before participants interacted with the prototype. The design (following [Müller et al., 2014]) aims to minimize bias and avoid broad, leading, or double-barreled formulations. When directly asking about preferences, the questionnaire uses Likert-scale items [Likert, 1932] to capture graded responses.

Whenever "Other" was provided as a response option, an additional text input field was included to allow participants to specify their answer.

D.1 Demographics

1. What is your highest level of education?
   (single choice)

   - Middle school
   - High school
   - Some college
   - Bachelor's
   - Master's
   - Doctorate
   - Diploma
   - Other

2. What is your primary daily activity?
   (multiple choice)

   - Student
   - Employed full-time
   - Employed part-time
   - Self-employed
   - Researcher/Academic
   - Not currently employed
   - Retired
   - Other

3. Field/Area of study or work (optional):
   (text)

4. Age range
   (single choice)

   - 18–24
   - 25–34
   - 35–44
   - 45–54
   - 55+

5. Gender
   (single choice)

   - Female
   - Male
   - Other

D.2 Reading Habits and Preferences

D.2.1 General Reading Behavior

1. What is your preferred format for reading?
   (single choice)

   - Digital (e.g., e-books, PDFs, online articles)

- Analog (printed books)
- No preference

2. How often do you read non-fiction texts?
   (single choice)

   - Daily
   - Several times per week
   - Weekly
   - Monthly
   - Rarely
   - Never

3. How many hours per week do you typically spend reading?
   (single choice)

   - 0–5
   - 6–10
   - 11–20
   - 20+

4. How long are your typical reading sessions?
   (single choice)

   - <30 minutes
   - 30–60 minutes
   - 1–2 hours
   - 2+ hours

D.2.2 Language and Comprehension

1. What is your native language?
   (text)

2. What languages do you regularly read in?
   (text)

D.3 Content and Subject Matter

1. What type of non-fiction do you usually read?
   (multiple choice)

   - None
   - Article
   - Non-fiction book
   - Other

2. In which fields do you most frequently read non-fiction?
   (multiple choice)

   - Academic research
   - Professional development
   - Technical documentation
   - Science/Technology
   - Business/Economics
   - History/Politics
   - Philosophy/Theory
   - Self-improvement
   - Other

3. What are your primary reasons for reading?
   (multiple choice)

   - Recommendations
   - Required reading
   - Personal interest
   - Current events
   - Career development
   - Academic requirements
   - Personal growth
   - Other

Reading Comprehension Importance and Strategies

1. How important is reading comprehension in your daily work/study?
   (scale 1–5)

   - 1 – Not important

- 3 – Moderately important
- 5 – Very important

2. What aids help you best understand complex texts?
   (multiple choice)

   - Highlighting key points
   - Summarization
   - Dictionary definitions
   - Simpler language versions
   - Translation to other languages
   - Visual aids/diagrams
   - None of these
   - Other

3. What is your preferred reading level for non-fiction texts?
   (single choice)

   - Simple/Basic
   - Intermediate
   - Advanced/Technical
   - Depends on the topic

Handling Complex Text

1. How do you typically handle text that is too complex?
   (multiple choice)

   - Look up terms
   - Re-read multiple times
   - Skip difficult sections
   - Seek simpler versions
   - Ask others for help
   - Give up reading
   - Other

D.4 Technical Experience and Preferences

D.4.1 Tools and Formats

1. Rate your comfort level with digital reading tools.
   (scale 1–5)

   - 1 – Very uncomfortable
   - 3 – Neutral
   - 5 – Very comfortable

2. Which digital reading tools have you used before?
   (multiple choice)

   - Kindle
   - PDF readers
   - Apple Books
   - Google Play Books
   - Calibre
   - Academic paper readers
   - Other

3. What file formats do you commonly read?
   (multiple choice)

   - PDF
   - EPUB
   - HTML
   - Word
   - Plain text
   - Other

D.4.2 Features and Preferences

1. What features do you value most in digital reading tools?
   (multiple choice)

- Search functionality
- Highlighting
- Note-taking
- Customizable display
- Dictionary integration
- Cross-device sync
- Sharing capabilities
- Other

D.4.3 Device and Display

1. What devices do you use for reading?
   (multiple choice)

   - Smartphone
   - Tablet
   - Laptop
   - Desktop
   - E-reader
   - Other

2. What screen size do you prefer for reading?
   (single choice)

   - Small (phone)
   - Medium (tablet)
   - Large (laptop/desktop)

3. Do you usually read on dark or light mode?
   (single choice)

   - Light
   - Dark
   - System default
   - Varies

4. Preferred font size range?
   (single choice)

   - Small

- Medium
- Large
- Extra large

D.5 Accessibility Requirements

1. Do you have any visual or reading-related conditions?
   (multiple choice)

   - Dyslexia
   - Visual impairment
   - Color blindness
   - ADHS
   - Other
   - None

Appendix E

Post-Interaction Questionnaire

The following questionnaire was used to collect structured data after participants interacted with the prototype. The design (following [Müller et al., 2014]) aims to minimize bias and avoid broad, leading, or double-barreled formulations and the System Usability Scale [Brooke et al., 1996] is included for comparability. When directly asking about preferences, the questionnaire uses Likert-scale items [Likert, 1932] to capture graded responses.

Whenever "Other" was provided as a response option, an additional text input field was included to allow participants to specify their answer.

E.1 Reading Impact (RQ1)

1. Reading with transformations made me read:
   (scale: 1–5)

    - 1 = Much slower
    - 3 = No difference
    - 5 = Much faster

2. Using transformations made understanding the text:
   (scale: 1–5)

   - 1 = Much harder
   - 3 = No difference
   - 5 = Much easier

3. With transformations, finding key information was:
   (scale: 1–5)

   - 1 = Much harder
   - 3 = No difference
   - 5 = Much easier

4. With transformations, long passages felt:
   (scale: 1–5)

   - 1 = More overwhelming
   - 3 = No difference
   - 5 = More manageable

5. The transformations affected my reading flow:
   (scale: 1–5)

   - 1 = Very disruptively
   - 3 = No effect
   - 5 = Very smoothly

6. Which aspects of the text became more accessible?
   (multiple choice)

   - Technical terms
   - Complex sentences
   - Text structure
   - Main arguments
   - Supporting details
   - Nothing became more accessible
   - Other

7. The transformations helped me most with:
   (multiple choice)

- Understanding complex terms
- Following arguments
- Maintaining focus
- Processing dense information
- Connecting ideas
- No noticeable help
- Other

8. Using the transformations made me feel:
   (multiple choice)

   - More confident
   - Less overwhelmed
   - More engaged
   - More confused
   - More frustrated
   - No different than usual
   - Other

E.2 Design and Usability (RQ2)

1. Finding the right transformation was:
   (scale: 1–5)

   - 1 = Very difficult
   - 3 = Neutral
   - 5 = Very easy

2. Finding available transformations was:
   (scale: 1–5)

   - 1 = Very difficult
   - 3 = Neutral
   - 5 = Very easy

3. The transformation controls (on/off, choice, slider) were appropriate for their purpose:
   (scale: 1–5)

   - 1 = Strongly disagree
   - 5 = Strongly agree

4. The menu bar layout was effective for accessing transformations:
   (scale: 1–5)

   - 1 = Strongly disagree
   - 5 = Strongly agree

5. I would prefer transformations to be organized as:
   (single choice)

   - A toolbar at the top
   - A sidebar
   - Floating bubbles near text
   - A collapsible panel
   - No preference
   - Other

6. When a transformation changed the text, this change was:
   (scale: 1–5)

   - 1 = Very hard to notice
   - 3 = Neutral
   - 5 = Very clear to see

7. Getting comfortable with the transformations took:
   (scale: 1–5)

   - 1 = Very long
   - 3 = Moderate time
   - 5 = Very quick

8. When multiple transformations were active, using the tool was:
   (scale: 1–5)

   - 1 = Very confusing
   - 3 = Neutral

- 5 = Very clear

9. The transformation effects were:
   (scale: 1–5)

   - 1 = Very unpredictable
   - 3 = Somewhat predictable
   - 5 = Very predictable

10. The transformation that felt most natural to use was:
    (single choice)

    - On/Off transformations
    - Choice transformations
    - Slider transformations
    - None felt natural

11. Rate your satisfaction with Toggle (On/Off) transformations:
    (scale: 1–5)

    - 1 = Very unsatisfied
    - 3 = Neutral
    - 5 = Very satisfied

12. Rate your satisfaction with Choice transformations:
    (scale: 1–5)

    - 1 = Very unsatisfied
    - 3 = Neutral
    - 5 = Very satisfied

13. Rate your satisfaction with Slider transformations:
    (scale: 1–5)

    - 1 = Very unsatisfied
    - 3 = Neutral
    - 5 = Very satisfied

14. Creating custom transformations through chat was:
(scale: 1–5)
    - 1 = Very confusing
    - 3 = Neutral
    - 5 = Very intuitive

E.3 Technical Performance

1. Did you experience any technical issues?
(multiple choice)
    - Slow responses
    - Transformation errors
    - Interface glitches
    - Loading problems
    - Other
2. Transformations applied quickly when enabled:
(scale: 1–5)
    - 1 = Strongly disagree
    - 5 = Strongly agree
3. Multiple transformations performed smoothly:
(scale: 1–5)
    - 1 = Strongly disagree
    - 5 = Strongly agree

E.4 Future Usage and Improvements (RQ3)

1. For what types of content would you most likely use this tool?
(multiple choice)

- Scientific and technical books
- Textbooks and learning materials
- Non-fiction and guidebooks
- Fiction (narrative literature)
- Books in foreign languages (to improve comprehension)
- Other

2. What aspects were most difficult to learn?
   (multiple choice)

   - Finding appropriate transformations
   - Understanding settings
   - Creating custom transformations
   - Managing multiple transformations
   - Other

3. What additional transformation types would be useful?
   (text)

4. Would you use this tool for future reading?
   (scale: 1–5)

   - 1 = Definitely not
   - 3 = Maybe
   - 5 = Definitely yes

5. How does this compare to other reading tools you've used?
   (scale: 1–5)

   - 1 = Much worse
   - 3 = About the same
   - 5 = Much better

E.5 System Usability Scale

1. I think that I would like to use this system frequently

2. I found the system unnecessarily complex
3. I thought the system was easy to use
4. I think that I would need the support of a technical person to be able to use this system
5. I found the various functions in this system were well integrated
6. I thought there was too much inconsistency in this system
7. I would imagine that most people would learn to use this system very quickly
8. I found the system very cumbersome to use
9. I felt very confident using the system
10. I needed to learn a lot of things before I could get going with this system
    (all items: scale 1–5)

    - 1 = Strongly disagree
    - 5 = Strongly agree

Appendix F

Detailed Instructions for Transformation Options

The instructions for the different transformation options were developed to complete the infix of the transformation instruction provided in Appendix C and align with prompt engineering techniques from Section 3.4.2, especially following provider guidelines, such as using markup formats, clearly defined task goals and was further optimized using the larger model gpt-4o [OpenAI, 2024] instead of the used model gpt-4o-mini [OpenAI, 2024].

Complexity (slider):

- **Basic** – The transformed text must be simple and easy to understand for a young teenager (around 12–14 years old), using short sentences, common words, and avoiding technical terms.
- **Intermediate** – The transformed text must be clear and accessible to an average adult without specialized knowledge, using straightforward sentences and avoiding overly technical language.
- **Advanced** – The transformed text must be precise and articulate, suitable for an educated au-

dience, with varied sentence structures and appropriate use of technical terms.

- **Scholarly** – The transformed text must be sophisticated and nuanced, suitable for academic or professional contexts, with complex sentence structures and advanced vocabulary.

Tone (choice):

- **Formal** – The transformed text must use a professional and polished tone, adhering to formal language conventions, with precise vocabulary and a respectful, neutral demeanor.
- **Conversational** – The transformed text must use a friendly and approachable tone, mimicking natural spoken language with informal expressions and a personal touch.
- **Persuasive** – The transformed text must use a compelling and assertive tone, designed to convince the audience through logical reasoning, emotional appeals, and motivational language.
- **Humorous** – The transformed text must use a lighthearted and entertaining tone, incorporating humor, wit, and playful language to engage the audience.

Style (choice):

- **Publistic** – The transformed text must adopt a vibrant and engaging style, similar to that of opinion pieces or journalistic articles, using rhetorical devices and expressive language to captivate the reader.
- **Neutral** – The transformed text must maintain a balanced and impartial style, using objective language that conveys information clearly and without bias.
- **Scientific** – The transformed text must adopt a precise and formal style, suitable for academic or technical writing, with clear logic, evidence-based statements, and domain-specific terminology.

More Personal (slider):

- **Neutral** – The transformed text must remain formal and impersonal.
- **Slightly** – The transformed text must include slight elements of personal touch.
- **Moderately** – The transformed text must be moderately personal, blending professional and personal tone.
- **Very** – The transformed text must be predominantly personal, with strong personal engagement.
- **Extremely** – The transformed text must be thoroughly personal, almost resembling a friendly conversation.

Text to Emoji (toggle):

- **On** – The transformed text must include many context-relevant emojis to enhance meaning.

Appendix G

Consent Form

Einverständniserklärung für Masterarbeitsstudie

Evaluation eines KI-gestützten E-Readers mit Echtzeit-Texttransformation.
Studienleitung: ………..………..………..………..………..………..………..………..……….
………..………..………..………..………..………..………..………..……….

Ziel der Studie: Das Ziel der Studie ist es, zu verstehen, wie Nutzende das evaluierte System verwenden, um Herausforderungen und Stärken in der Interaktion zu identifizieren und Auswirkungen auf die Zugänglichkeit von Texten zu untersuchen.

Ablauf: Im Verlauf der Studie werden Sie mit dem zu evaluierenden System interagieren, gestellte Aufgaben lösen sowie zwei Fragebogen ausfüllen. Weiterhin wird ein semi-strukturiertes Interview geführt, um Ihre Eindrücke und Beobachtungen während der Interaktion besser zu verstehen und zu besprechen.

Risiken/Beschwerden: Es könnte sein, dass Sie die Teilnahme an der Studie ermüdet. Sie werden mehrere Gelegenheiten haben, sich zu erholen; zusätzliche Pausen sind ebenfalls möglich. Es sind keine weiteren Risiken im Zusammenhang mit der Studie bekannt. Sollten die Aufgaben oder das Interview anstrengend für Sie sein, kann die Bearbeitung zu jedem Zeitpunkt pausiert oder abgebrochen werden.

Nutzen: Die Resultate der Studie werden zur Verbesserung des evaluierten Systems genutzt. Darüber hinaus sollen sie Erkenntnisse darüber liefern, inwiefern das System die Zugänglichkeit von Texten beeinflusst und gegebenenfalls Perspektiven für zukünftige Forschung eröffnen.

Erhobene Daten: Im Rahmen der Studie werden Daten durch zwei Fragebögen vor und nach der Interaktion erhoben. Der von Ihnen verwendete Prototyp zeichnet automatisch Ereignisdaten auf, beispielsweise zur Navigation, zu Klicks oder anderen Interaktionen. Während der Nutzung werden Sie beobachtet; dabei können Reaktionen und Rückfragen dokumentiert werden. Auch das anschließende freie Interview wird protokolliert.

Vertraulichkeit: Alle Informationen, die während der Studienphase gesammelt werden, werden streng vertraulich behandelt. Die gesammelten Daten werden auf Servern der RWTH Aachen University gespeichert und sind nur durch Bedienstete der RWTH Aachen University einzusehen.
Bitte beachten Sie, dass bei der Interaktion mit dem Chatbot im Rahmen der Studie Ihre Nachrichten an die Server von OpenAI übermittelt werden, um eine passende Antwort zu generieren. Die Verarbeitung erfolgt gemäß deren geltenden Datenschutzbestimmungen. Weiterhin kann zu jeder Zeit eine Löschung der protokollierten & erhobenen Daten durch eine formlose E-Mail an ………..………..………..………..………. eingefordert werden. Zusätzlich kann natürlich während der Studie die Angabe von Daten oder die Beantwortung von Fragen verweigert werden. Die personenbezogenen Daten werden maximal bis zum 30.06.2025 gespeichert.

☐ Ich habe die Hinweise auf diesem Formular gelesen, verstanden und bin damit einverstanden.

Name der teilnehmenden Person	Unterschrift der teilnehmenden Person	Datum
	Studienleitung	Datum

Wenn Sie Fragen zu dieser Studie haben, wenden Sie sich bitte an ………..………..……… per Email:
………..………..………..………..……….

Figure G.1: Study Consent Form

Bibliography

[1] Aerospace and Defence Industries Association of Europe (ASD). *ASD-STE100: Simplified Technical English, Specification ASD-STE100*. ASD - European Aerospace and Defence Industries Association, issue 8 edition, 2021. Available at: `https://www.asd-ste100.org/`.

[2] Ilaria Amaro, Paola Barra, Attilio Della Greca, Rita Francese, and Cesare Tucci. Believe in artificial intelligence? A user study on the ChatGPT's fake information impact. *IEEE Transactions on Computational Social Systems*, 2023.

[3] Amazon. Kindle Paperwhite (2024). `https://www.amazon.de/kindle-paperwhite-2024/dp/B0CFP6F89F`, 2024. Accessed: 2025-01-14.

[4] Anthropic. Claude: An AI assistant by Anthropic. `https://claude.ai`, 2023. Accessed: 2025-06-05.

[5] Anthropic. Prompt Engineering Overview, 2025. URL `https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/overview`. Accessed: 2025-04-19.

[6] Berk Atil, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu, Lixinyu Xu, and Breck Baldwin. Non-Determinism of "Deterministic" LLM Settings, 2025. URL `https://arxiv.org/abs/2408.04667`.

[7] Tal August, Kyle Lo, Noah A. Smith, and Katharina Reinecke. Know Your Audience: The benefits and pitfalls of generating plain language summaries beyond the "general" audience. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery. `doi.org/10.1145/3613904.3642289`.

[8] Tal August, Kyle Lo, Noah A Smith, and Katharina Reinecke. Know Your Audience: The benefits and pitfalls of generating plain language summaries beyond the" general" audience. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–26, 2024.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[10] Peng Fei Bai, Robert Andrew Hayes, Mingliang Jin, Lingling Shui, Zi Chuan Yi, Li Wang, Xiao Zhang, and Guofu Zhou. Review of paper-like display technologies (invited review). *Progress in electromagnetics research*, 147:95–116, 2014.

[11] Aaron Bangor, Philip T Kortum, and James T Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24 (6):574–594, 2008.

[12] BIG bench authors. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

[13] Francoise Benhamou. Fair use and fair competition for digitized cultural goods: the case of eBooks. *Journal of cultural economics*, 39:123–131, 2015.

[14] Ana Carolina Bittar. Unlocking the Gates of Alexandria: DRM, Competition and Access to E-Books. *Competition and Access to E-Books (July 25, 2014)*, 2014.

[15] Christoph Bläsi and Franz Rothlauf. *On the interoperability of eBook formats*. European and International Booksellers Federation Brussels, 2013.

[16] Justus Bogner and Manuel Merkel. To type or not to type?: a systematic comparison of the software quality of JavaScript and typescript applications on GitHub. In *Proceedings of the 19th International Conference on Mining Software Repositories*, MSR '22, page 658–669. ACM, May 2022. `doi.org/10.1145/3524842.3528454`.

[17] Jay David Bolter. *Writing space: Computers, hypertext, and the remediation of print*. Routledge, 2001.

[18] Petter Bae Brandtzaeg and Asbjørn Følstad. Why people use chatbots. In *Internet Science: 4th International Conference, INSCI 2017, Thessaloniki, Greece, November 22-24, 2017, Proceedings 4*, pages 377–392. Springer, 2017.

[19] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[20] Virginia Braun and Victoria Clarke. Toward good practice in thematic analysis: Avoiding common problems and be (com) ing a knowing researcher. *International journal of transgender health*, 24(1):1–6, 2023.

[21] Tim Bray. The JavaScript Object Notation (JSON) Data Interchange Format. RFC 8259, December 2017. URL `https://www.rfc-editor.org/info/rfc8259`.

[22] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible Markup Language (XML) 1.0 (Fifth Edition). `https://www.w3.org/TR/xml/`, 2008. W3C Recommendation.

[23] John Brooke and others. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[24] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and others. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[25] Cambridge Dictionary. non-fiction, 2025. URL `https://dictionary.cambridge.org/dictionary/english-german/non-fiction?q=Non-fiction`. Accessed: 2025-05-29.

[26] Stuart K Card, Jock D Mackinlay, and George G Robertson. A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems (TOIS)*, 9(2):99–122, 1991.

[27] Zeki Çelikbaş. EPUB use in digital libraries: Developing an online epub creator application. *Digital Publishing and Mobile Technologies*, page 120, 2011.

[28] Yong-Ho Chang and Byoung-Hun Kong. The emergence and ensuing typology of global ebook platform-The case study on Google eBook, Amazon Kindle, Apple iBooks Store. *Journal of the Korea Academia-Industrial cooperation Society*, 13(8):3389–3404, 2012.

[29] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and others. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45, 2024.

[30] Ana Paula Chaves and Marco Aurelio Gerosa. How should my chatbot interact? A survey on social characteristics in human–chatbot interaction design. *International Journal of Human–Computer Interaction*, 37(8):729–758, 2021.

[31] Celia Chen and Alex Leitch. LLMs as Academic Reading Companions: Extending HCI Through Synthetic Personae. *arXiv preprint arXiv:2403.19506*, 2024.

[32] Mike Chen. *Changing the Narrative Perspective: A New Language Processing Task and Machine Learning Approaches*. Ohio University, 2022.

[33] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, and others. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*, 2024.

[34] Abhimanyu Chopra, Abhinav Prashar, and Chandresh Sain. Natural language processing. *International journal of technology enhancements and emerging engineering research*, 1(4):131–134, 2013.

[35] CommonMark Community. CommonMark Specification. `https://spec.commonmark.org/`, 2020. Standardized Markdown Specification.

[36] SQLite Consortium. Appropriate Uses For SQLite, 2025. URL `https://www.sqlite.org/whentouse.html`. Accessed: 2025-05-23.

[37] Andy Crabtree, Steve Benford, Chris Greenhalgh, Paul Tennent, Matthew Chalmers, and Barry Brown. Supporting ethnographic studies of ubiquitous computing in the wild. In *Proceedings of the 6th conference on Designing Interactive systems*, pages 60–69, 2006.

[38] John W Creswell and Vicki L Plano Clark. *Designing and conducting mixed methods research*. Sage publications, 2017.

[39] Gregory Currie. What is fiction? *The Journal of Aesthetics and Art Criticism*, 43 (4):385–392, 1985.

[40] Michael Curtotti and Eric McCreath. A right to access implies a right to know: An open online platform for research on the readability of law. *J. Open Access L.*, 1:1, 2013.

[41] Charles Darwin. *On the origin of species by means of natural selection, or, the preservation of favoured races in the struggle for life*. J. Murray, 1861.

[42] Siddhartha Das. Bookworm - A simple ebook reader for Elementary OS. `https://github.com/babluboy/bookworm`, 2025. Accessed: 2025-05-14.

[43] N.K. Denzin. *The Research Act: A Theoretical Introduction to Sociological Methods*. McGraw-Hill, New York, NY, 1978.

[44] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[45] Django Software Foundation. *The Django Admin Site*. Django Software Foundation, 2025. URL `https://docs.djangoproject.com/en/5.2/ref/contrib/admin/`. Version 5.2.

[46] Douban. Douban Books, 2025. `https://book.douban.com/` (accessed on 01.04.2025).

[47] Paul Dourish. Implications for design. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 541–550, 2006.

[48] Stephan Dutke, Anna Christina Grefe, and Claudia Leopold. Learning from scientific texts: Personalizing the text increases transfer performance and task involvement. *European Journal of Psychology of Education*, 31:499–513, 2016.

[49] EDRLab. Thorium Reader - A cross-platform desktop reading app. `https://github.com/edrlab/thorium-reader`, 2025. Accessed: 2025-05-14.

[50] Liana Ermakova, Patrice Bellot, Pavel Braslavski, Jaap Kamps, Josiane Mothe, Diana Nurbakova, Irina Ovchinnikova, and Eric San-Juan. Text simplification for scientific information access: CLEF 2021 simpletext workshop. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*, pages 583–592. Springer, 2021.

[51] Elika J. Etemad and Tab Atkins Jr. Selectors Level 4. `https://www.w3.org/TR/selectors-4/`, 2023. W3C Candidate Recommendation.

[52] Felix Faltings, Michel Galley, Baolin Peng, Kianté Brantley, Weixin Cai, Yizhe Zhang, Jianfeng Gao, and Bill Dolan. Interactive text generation. *arXiv preprint arXiv:2303.00908*, 2023.

[53] Michelle Faverio and Andrew Perrin. Three-in-ten Americans now read e-books, 2022. URL `https://www.pewresearch.org/short-reads/2022/01/06/three-in-ten-americans-now-read-e-books/`. Accessed: 2025-01-15.

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

[55] Asbjørn Følstad and Petter Bae Brandtzæg. Chatbots and the new world of HCI. *interactions*, 24(4):38–42, 2017.

[56] Django Software Foundation. Django: The Web Framework for Perfectionists with Deadlines, 2025. URL `https://www.djangoproject.com/`. Accessed: 2025-05-23.

[57] Readium Foundation. Readium - Open-source EPUB Reading System Components. `https://github.com/readium`, 2025. Accessed: 2025-05-14.

[58] Glenn Fulcher. Text difficulty and accessibility: Reading formulae and expert judgement. *System*, 25(4):497–513, 1997. `doi.org/10.1016/S0346-251X(97)00048-1`.

[59] FuturePress. epub.js - JavaScript library to render EPUB documents in the browser. `https://github.com/futurepress/epub.js/`, 2025. Accessed: 2025-05-14.

[60] Núria Gala and Johannes Ziegler. Reducing lexical complexity as a tool to increase text accessibility for children with dyslexia. In Dominique Brunato, Felice Dell'Orletta, Giulia Venturi, Thomas François, and Philippe Blache, editors, *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 59–66, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[61] Matt Garrish. *What is EPUB 3?* " O'Reilly Media, Inc.", 2011.

[62] Matt Garrish and Markus Gylling. *EPUB 3 best practices*. " O'Reilly Media, Inc.", 2013.

[63] E. Geisler, P. Prabhaker, and M. Nayar. Information integrity: an emerging field and the state of knowledge. In *PICMET '03: Portland International Conference on Management of Engineering and Technology Technology Management for Reshaping the World, 2003.*, pages 217–221, 2003. `doi.org/10.1109/PICMET.2003.1222797`.

[64] Ivaylo Gerchev. *Tailwind CSS*. SitePoint Pty Ltd, 2022.

[65] Devndra Ghimire. Comparative study on Python web frameworks: Flask and Django, 2020.

[66] Richard J Gilbert. E-books: A tale of digital disruption. *Journal of Economic Perspectives*, 29(3):165–184, 2015.

[67] Goodreads. Best Book Lists, 2024. URL `https://www.goodreads.com/list/tag/best`. Accessed: 2025-06-03.

[68] Goodreads. Goodreads: Book Reviews, Recommendations, and Discussion, 2025. `https://www.goodreads.com/` (accessed on 01.04.2025).

[69] Google. *Book file guidelines - Google Play Books Partner Center Help*, 2025. URL `https://support.google.com/books/partner/answer/3424254?hl=en`. Accessed: 2025-04-02.

[70] Google DeepMind. Introduction to Prompt Design, 2025. URL `https://ai.google.dev/gemini-api/docs/prompting-intro`. Accessed: 2025-04-19.

[71] Kovid Goyal. Calibre - The official source code repository for the calibre ebook manager. https://github.com/kovidgoyal/calibre, 2025. Accessed: 2025-05-14.

[72] Henry Gray, Thomas Pickering Pick, Robert Howden, and John Chalmers Da Costa. *Anatomy, descriptive and surgical*. Bounty Books New York, 1901.

[73] H Rex Hartson and Eric C Smith. Rapid prototyping in human-computer interface development. *Interacting with computers*, 3(1):51–91, 1991.

[74] Sandro Hawke. REST. https://www.w3.org/2001/sw/wiki/REST, June 2011. Semantic Web Standards Wiki, World Wide Web Consortium.

[75] Gary G Hendrix. Natural-language interface. *American Journal of Computational Linguistics*, 8(2):56–61, 1982.

[76] D. Richard Hipp. SQLite Home Page, 2025. URL https://www.sqlite.org/. Accessed: 2025-05-23.

[77] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration, 2020. URL https://arxiv.org/abs/1904.09751.

[78] Or Honovich, Uri Shaham, Samuel R Bowman, and Omer Levy. Instruction induction: From few examples to natural language task descriptions. *arXiv preprint arXiv:2205.10782*, 2022.

[79] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and others. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

[80] Apple Inc. *Apple Books Formatting and Content Guidelines*, 2025. URL https://help.apple.com/itc/applebooksstoreformatting/en.lproj/static.html. Accessed: 2025-04-02.

[81] Inclusion Europe. Do Not Write for Us Without Us: Involving People with Intellectual Disabilities in the Writing of Texts that Are Easy to Read and Understand. https://sid.usal.es/idocs/F8/FDO23139/write_for_us.pdf, November 2018. With the support of the European Commission.

[82] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). ISO/IEC 25010:2023 — Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model. https://www.iso.org/standard/78176.html, 2023. Accessed: 2025-05-09.

[83] Susanne Iwarsson and Agneta Ståhl. Accessibility, usability and universal design—positioning and definition of concepts describing person-environment relationships. *Disability and rehabilitation*, 25(2):57–66, 2003.

[84] Todd D Jick. Mixing qualitative and quantitative methods: Triangulation in action. *Administrative science quarterly*, 24(4):602–611, 1979.

[85] Prashant Johri, Sunil K Khatri, Ahmad T Al-Taani, Munish Sabharwal, Shakhzod Suvanov, and Avneesh Kumar. Natural language processing: History, evolution, application, and future work. In *Proceedings of 3rd International Conference on Computing Informatics and Networks: ICCIN 2020*, pages 365–375. Springer, 2021.

[86] Hao Kang, Qingru Zhang, Han Cai, Weiyuan Xu, Tushar Krishna, Yilun Du, and Tsachy Weissman. Win Fast or Lose Slow: Balancing Speed and Accuracy in Latency-Sensitive Decisions of LLMs. *arXiv preprint arXiv:2505.19481*, 2025.

[87] Taewook Kim, Dhruv Agarwal, Jordan Ackerman, and Manaswi Saha. Steering AI-Driven Personalization of Scientific Text for General Audiences. *arXiv preprint arXiv:2411.09969*, 2024.

[88] Maria Kjærup, Mikael B Skov, Peter Axel Nielsen, Jesper Kjeldskov, Jens Gerken, and Harald Reiterer. *Longitudinal studies in HCI research: a review of CHI publications from 1982–2019*. Springer, 2021.

[89] Lars Henning Klein, Roland Aydin, and Robert West. Emojinize: enriching any text with emoji translations. *arXiv preprint arXiv:2403.03857*, 2024.

[90] Philip Kortum and Mary Sorber. Measuring the usability of mobile applications for phones and tablets. *International Journal of Human-Computer Interaction*, 31(8):518–529, 2015.

[91] Ilpo Koskinen, John Zimmerman, Thomas Binder, Johan Redstrom, and Stephan Wensveen. Design research through practice: From the lab, field, and showroom. *IEEE Transactions on Professional Communication*, 56(3):262–263, 2013.

[92] Alen Kovář. EbookLib: Python E-book library for EPUB2/EPUB3. `https://pypi.org/project/EbookLib/`, n.d. Accessed: 2025-05-25.

[93] Alexis Kuerbis, Katherine van Stolk-Cooke, and Frederick Muench. An exploratory study of mobile messaging preferences by age: Middle-aged and older adults compared to younger adults. *Journal of rehabilitation and assistive technologies engineering*, 4:2055668317733257, 2017.

[94] James H Kuklinski, Paul J Quirk, Jennifer Jerit, David Schwieder, and Robert F Rich. Misinformation and the currency of democratic citizenship. *The Journal of Politics*, 62(3):790–816, 2000.

[95] Marie Lebert. A short history of ebooks, 2009.

[96] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. Evaluation strategies for HCI toolkit research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2018.

[97] Stephan Lewandowsky, Ullrich KH Ecker, Colleen M Seifert, Norbert Schwarz, and John Cook. Misinformation and its correction: Continued influence and successful debiasing. *Psychological science in the public interest*, 13 (3):106–131, 2012.

[98] James R Lewis. Binomial confidence intervals for small sample usability studies. In *Adv. Appl. Ergon. Proc. 1st Int. Conf. Appl. Ergon*, volume 732, 1996.

[99] Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. Large language models understand and can be enhanced by emotional stimuli. *arXiv preprint arXiv:2307.11760*, 2023.

[100] Zilin Li, Shaofei Shen, and Zhilong Xie. LLM-Powered Reading Aid for Visually Impaired Online Learners. In Tim Schlippe, Eric C. K. Cheng, and Tianchong Wang, editors, *Artificial Intelligence in Education Technologies: New Development and Innovative Practices*, pages 281–295, Singapore, 2025. Springer Nature Singapore.

[101] Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and others. Controllable text generation for large language models: A survey. *arXiv preprint arXiv:2408.12599*, 2024.

[102] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[103] Kyle Lo, Joseph Chee Chang, Andrew Head, Jonathan Bragg, Amy X. Zhang, Cassidy Trier, Chloe Anastasiades, Tal August, Russell Authur, Danielle Bragg, Erin Bransom, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Yen-Sung Chen, Evie Yu-Yen Cheng, Yvonne Chou, Doug Downey, Rob Evans, Raymond Fok, Fangzhou Hu, Regan Huff, Dongyeop Kang, Tae Soo Kim, Rodney Kinney, Aniket Kittur, Hyeonsu Kang, Egor Klevak, Bailey Kuehl, Michael Langan, Matt Latzke, Jaron Lochner, Kelsey MacMillan, Eric Marsh, Tyler Murray, Aakanksha Naik, Ngoc-Uyen Nguyen, Srishti Palani, Soya Park, Caroline Paulic, Napol Rachatasumrit, Smita Rao, Paul Sayre, Zejiang Shen, Pao Siangliulue, Luca Soldaini, Huy Tran, Madeleine van Zuylen, Lucy Lu Wang, Christopher Wilhelm, Caroline Wu, Jiangjiang Yang, Angele Zamarron, Marti A. Hearst, and Daniel S. Weld.

The Semantic Reader Project: Augmenting Scholarly Documents through AI-Powered Interactive Reading Interfaces, 2023. URL `https://arxiv.org/abs/2303.14334`.

[104] Kyle Lo, Joseph Chee Chang, Andrew Head, Jonathan Bragg, Amy X Zhang, Cassidy Trier, Chloe Anastasiades, Tal August, Russell Authur, Danielle Bragg, and others. The semantic reader project: Augmenting scholarly documents through ai-powered interactive reading interfaces. *arXiv preprint arXiv:2303.14334*, 2023.

[105] Karen Locke. Qualitative research and evaluation methods. *Organizational Research Methods*, 5(3):299, 2002.

[106] Alessandro Ludovico. An e-publishing archaeology. *Editorial Coordinators: Rufus Adebayo, Ismail Farouk, Steve Jones, Maleshoane Rapeane-Mathonsi*, page 299, 2018.

[107] Guoqing Luo, Yu Tong Han, Lili Mou, and Mauajama Firdaus. Prompt-based editing for text style transfer. *arXiv preprint arXiv:2301.11997*, 2023.

[108] Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. Controllable text simplification with explicit paraphrasing. *arXiv preprint arXiv:2010.11004*, 2020.

[109] Liliya Makhmutova, Giancarlo D Salton, Fernando Perez-Tellez, and Robert J Ross. Automated Medical Text Simplification for Enhanced Patient Access. In *BIOSTEC (2)*, pages 208–218, 2024.

[110] Simone Marinai, Emanuele Marino, and Giovanni Soda. Conversion of PDF books in ePub format. In *2011 International Conference on Document Analysis and Recognition*, pages 478–482. IEEE, 2011.

[111] Elizabeth J Marsh, Allison D Cantor, and Nadia M Brashier. Believing that humans swallow spiders in their sleep: False beliefs as side effects of the processes that support accurate knowledge. In *Psychology of learning and motivation*, volume 64, pages 93–132. Elsevier, 2016.

[112] Louis Martin, Benoît Sagot, Eric de la Clergerie, and Antoine Bordes. Controllable sentence simplification. *arXiv preprint arXiv:1910.02677*, 2019.

[113] Louis Martin, Angela Fan, Eric De La Clergerie, Antoine Bordes, and Benoît Sagot. MUSS: Multilingual unsupervised sentence simplification by mining paraphrases. *arXiv preprint arXiv:2005.00352*, 2020.

[114] Satoru Matsushima. Bibi - EPUB Reader on your website. `https://github.com/satorumurmur/bibi`, 2025. Accessed: 2025-05-14.

[115] Alistair McNaught and Huw Alexander. Ebooks and accessibility. *Ebooks in education: Realising the vision*, pages 35–49, 2014.

[116] MDN contributors. Range - Web APIs | MDN, 2023. URL `https://developer.mozilla.org/en-US/docs/Web/API/Range`. Accessed: 2025-05-27.

[117] MDN contributors. Document: querySelector() method - Web APIs | MDN, 2025. URL `https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelector`. Accessed: 2025-05-27.

[118] Elyas Meguellati, Lei Han, Abraham Bernstein, Shazia Sadiq, and Gianluca Demartini. How Good are LLMs in Generating Personalized Advertisements? In *Companion Proceedings of the ACM on Web Conference 2024*, pages 826–829, 2024.

[119] Meta AI. Prompting - How-to Guides, 2025. URL `https://www.llama.com/docs/how-to-guides/prompting/`. Accessed: 2025-04-19.

[120] Meta Platforms, Inc. *React – The library for web and native user interfaces*. Meta, 2024. `https://react.dev/`.

[121] Ning Miao, Yee Whye Teh, and Tom Rainforth. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. *arXiv preprint arXiv:2308.00436*, 2023.

[122] Robert Michielutte, Judy Bahnson, Mark B Dignan, and Elissa M Schroeder. The use of illustrations and narrative text style to improve readability of a health education brochure. *Journal of Cancer Education*, 7(3):251–260, 1992.

[123] Microsoft Corporation. *TypeScript – JavaScript With Syntax for Types*. Microsoft, 2025. `https://www.typescriptlang.org/`.

[124] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[125] Mistral AI Team. Mixtral of Experts, 2023. URL `https://mistral.ai/news/mixtral-of-experts`. Accessed: 2025-04-24.

[126] Janice M Morse. *Mixed method design: Principles and procedures*. Routledge, 2016.

[127] Hendrik Müller, Aaron Sedley, and Elizabeth Ferrall-Nunge. Survey research in HCI. *Ways of Knowing in HCI*, pages 229–266, 2014.

[128] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.

[129] Deepak Narayanan, Keshav Santhanam, Peter Henderson, Rishi Bommasani, Tony Lee, and Percy S Liang. Cheaply estimating inference efficiency metrics for autoregressive transformer models. *Advances in Neural Information Processing Systems*, 36:66518–66538, 2023.

[130] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.

[131] Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu. Exploring neural text simplification models. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 85–91, 2017.

[132] Don Norman. *The Design of Everyday Things*. Basic Books, revised and expanded edition, 2013.

[133] Anna Offenwanger, Alan John Milligan, Minsuk Chang, Julia Bullard, and Dongwook Yoon. Diagnosing bias in the gender representation of HCI research participants: how it happens and where we are. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2021.

[134] OpenAI. GPT-3.5-turbo. OpenAI API documentation, November 2023. URL `https://platform.openai.com/docs/models/gpt-3.5-turbo`. Accessed 2025-06-15.

[135] OpenAI. GPT-4o System Card. *arXiv preprint arXiv:2410.21276*, 2024.

[136] OpenAI. OpenAI API Reference - Assistants, 2024. URL `https://platform.openai.com/docs/api-reference/assistants`. Accessed: 2025-05-18.

[137] OpenAI. OpenAI API Reference - Completions, 2024. URL `https://platform.openai.com/docs/api-reference/completions`. Accessed: 2025-05-18.

[138] OpenAI. OpenAI API Reference - Threads, 2024. URL `https://platform.openai.com/docs/api-reference/threads`. Accessed: 2025-05-18.

[139] OpenAI. GPT-3.5 Technical Report, 2025. `https://platform.openai.com/docs/models/gpt-3-5` (accessed on 01.04.2025).

[140] OpenAI. OpenAI Python Library, 2025. URL `https://github.com/openai/openai-python`. Accessed: 2025-05-23.

[141] OpenAI. OpenAI Platform Status, 2025. URL `https://status.openai.com/`. Accessed June 11, 2025.

[142] OpenAI. Text Generation and Prompting, 2025. URL `https://platform.openai.com/docs/guides/text?api-mode=responses`. Accessed: 2025-04-19.

[143] Miles Osborne. Statistical Machine Translation., 2010.

[144] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and others. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[145] Yasuhiro Ozuru, Kyle Dempsey, and Danielle S McNamara. Prior knowledge, reading skill, and text cohesion in the comprehension of science texts. *Learning and instruction*, 19(3):228–242, 2009.

[146] Hye K. Pae. *The Impact of Digital Text*, pages 209–217. Springer International Publishing, Cham, 2020. ISBN 978-3-030-55152-0. `doi.org/10.1007/978-3-030-55152-0_11`.

[147] Lawrence A Palinkas, Sapna J Mendon, and Alison B Hamilton. Innovations in mixed methods evaluations. *Annual review of public health*, 40(1):423–442, 2019.

[148] Ajay Patel, Nicholas Andrews, and Chris Callison-Burch. Low-Resource Authorship Style Transfer: Can Non-Famous Authors Be Imitated? *arXiv preprint arXiv:2212.08986*, 2022.

[149] Tatiana Petrova. Eye movements in reading the texts of different functional styles: evidence from Russian. In *Intelligent Decision Technologies 2016: Proceedings of the 8th KES International Conference on Intelligent Decision Technologies (KES-IDT 2016)–Part II*, pages 285–298. Springer, 2016.

[150] Fabrizio Pilato. Sony LIBRIe–The first ever e-Ink e-Book reader. *Mobile Magazine [online]*, 25, 2004.

[151] Raunak M Pillai and Lisa K Fazio. The effects of repeating false and misleading information on belief. *Wiley Interdisciplinary Reviews: Cognitive Science*, 12 (6):e1573, 2021.

[152] Amazon Kindle Direct Publishing. *What file formats are supported for eBook manuscripts?*, 2025. URL `https://kdp.amazon.com/en_US/help/topic/G200634390`. Accessed: 2025-04-02.

[153] Publishing@W3C Publishing Maintenance Working Group. EPUB Accessibility 1.1: Content Conformance Requirements for EPUB Publications. `https://www.w3.org/TR/epub-a11y-11/`, September 2024. W3C Recommendation, last updated September 2024; accessed 17 June 2025.

[154] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and others. Improving language understanding by generative pre-training, 2018.

[155] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[156] Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*, 2021.

[157] Xiang Ren. Understanding the digital publishing economy: From eBook disruption to platform ecosystem. *The SAGE Handbook of the Digital Media Economy. London: SAGE*, pages 301–324, 2022.

[158] Ortwin Renn. Trust and credibility in risk communication, 1989.

[159] Anders Rex. Kindllm: An LLM Chat Web App Optimized for Kindle. `https://github.com/andersrex/kindllm`, 2024. Accessed: 2025-04-24.

[160] Leonard Richardson. beautifulsoup4: Screen-scraping library. `https://pypi.org/project/beautifulsoup4/`, n.d. Accessed: 2025-05-25.

[161] Carlos Rodríguez, Marcos Baez, Florian Daniel, Fabio Casati, Juan Carlos Trabucco, Luigi Canali, and Gianraffaele Percannella. REST APIs: A large-scale analysis of compliance with principles and best practices. In *Web Engineering: 16th International Conference, ICWE 2016, Lugano, Switzerland, June 6-9, 2016. Proceedings 16*, pages 21–39. Springer, 2016.

[162] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the association for computational linguistics*, 8:842–866, 2021.

[163] Leonard Rosenthol. *Developing with PDF: Dive Into the Portable Document Format*. " O'Reilly Media, Inc.", 2013.

[164] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

[165] Ramteja Sajja, Yusuf Sermet, Muhammed Cikmaz, David Cwiertny, and Ibrahim Demir. Artificial intelligence-enabled intelligent assistant for personalized and adaptive learning in higher education. *Information*, 15(10):596, 2024.

[166] Thorsten Schwarz, Sachin Rajgopal, and Rainer Stiefelhagen. Accessible EPUB: making EPUB 3 documents universal accessible. In *Computers Helping People with Special Needs: 16th International Conference, ICCHP 2018, Linz, Austria, July 11-13, 2018, Proceedings, Part I 16*, pages 85–92. Springer, 2018.

[167] Susan Scott and Wanda Orlikowski. The digital undertow: How the corollary effects of digital transformation affect industry standards. *Information Systems Research*, 33(1):311–336, 2022.

[168] shadcn. *shadcn/ui – A Set of Beautifully-Designed, Accessible Components and a Code Distribution Platform*. shadcn, 2025. `https://ui.shadcn.com/`.

[169] Matthew Shardlow. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70, 2014.

[170] Ben Shneiderman. Response time and display rate in human performance with computers. *ACM Computing Surveys (CSUR)*, 16(3):265–285, 1984.

[171] Ben Shneiderman, Catherine Plaisant, Max Cohen, Steven Jacobs, Niklas Elmqvist, and Nicholas Diakopoulos. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson, 6 edition, 2016.

[172] Gerhard Sletten. React Reader: An ePub-reader for React, powered by Epub.js. `https://github.com/gerhardsletten/react-reader`, 2017. Accessed: 2025-05-14.

[173] Peter Sorotokin, Garth Conboy, Brady Duga, John Rivlin, Don Beaver, Kevin Ballard, Alastair Fettes, and Daniel Weck. EPUB Canonical Fragment Identifier (epubcfi) Specification. Technical report, International Digital Publishing Forum, October 2011. URL `https://idpf.org/epub/linking/cfi/epub-cfi-20111011.html`. Recommended Specification, 11 October 2011.

[174] Sanja Štajner. Automatic text simplification for social good: Progress and challenges. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2637–2652, 2021.

[175] Statista. Amazons Kindle-Shop ist Deutschlands E-Book-Favorit. `https://de.statista.com/infografik/31047/meistgenutzte-anbieter-fuer-ebooks/`, 2023. Zugriff am 31. März 2025.

[176] Statista. Books - Worldwide. `https://www.statista.com/outlook/amo/media/books/worldwide`, 2025. Accessed: 2025-03-31.

[177] Statista. eBooks - Worldwide. `https://www.statista.com/outlook/dmo/digital-media/epublishing/ebooks/worldwide`, 2025. Accessed: 2025-03-31.

[178] Statista Research Department. Umsatzanteil von E-Books im Buchmarkt bis 2021, 2022. URL https://de.statista.com/statistik/daten/studie/303339/umfrage/umsatzanteil-von-e-books-im-buchmarkt/. Accessed: 2025-01-15.

[179] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[180] Tailwind Labs. *Tailwind CSS – A Utility-First CSS Framework for Rapid UI Development*. Tailwind Labs, 2025. https://tailwindcss.com/.

[181] Zhixing Tan, Shuo Wang, Zonghan Yang, Gang Chen, Xuancheng Huang, Maosong Sun, and Yang Liu. Neural machine translation: A review of methods, resources, and tools. *AI Open*, 1:5–21, 2020.

[182] FolioReader Team. FolioReaderKit - A Swift ePub reader and parser framework for iOS. https://folioreader.github.io/FolioReaderKit/, 2025. Accessed: 2025-05-14.

[183] VersOne Team. EpubReader - .NET library to read EPUB files. https://github.com/vers-one/EpubReader, 2025. Accessed: 2025-05-14.

[184] Alan Mathison Turing and others. On computable numbers, with an application to the Entscheidungsproblem. *J. of Math*, 58(345-363):5, 1936.

[185] G Türkmen, A Sezen, and G Şengül. Comparative Analysis of Programming Languages Utilized in Artificial Intelligence Applications: Features, Performance, and Suitability. *International Journal of Computational and Experimental Science and Engineering*, 10(3):461–469, 2024.

[186] Tiberio Uricchio, Silvia Ceccacci, Ilaria D'Angelo, Noemi Del Bianco, and Catia Giaconi. Investigating OpenAI's ChatGPT Capabilities to Improve Accessibility of Textual Information: An Explorative Study. In *International Conference on Human-Computer Interaction*, pages 326–336. Springer, 2024.

[187] Wil MP Van Der Aalst and Josep Carmona. *Process mining handbook*. Springer Nature, 2022.

[188] Axel Van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *Proceedings fifth ieee international symposium on requirements engineering*, pages 249–262. IEEE, 2001.

[189] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[190] Vercel. Next.js – The React Framework. https://nextjs.org, 2016. Accessed: 2025-05-23.

[191] W3C. EPUB 3.4 Overview. https://w3c.github.io/epub-specs/epub34/overview/, 2025. Accessed: 2025-04-03.

[192] Shuohuan Wang, Yu Sun, Yang Xiang, Zhihua Wu, Siyu Ding, Weibao Gong, Shikun Feng, Junyuan Shang, Yanbin Zhao, Chao Pang, Jiaxiang Liu, Xuyi Chen, Yuxiang Lu, Weixin Liu, Xi Wang, Yangfan Bai, Qiuliang Chen, Li Zhao, Shiyong Li, Peng Sun, Dianhai Yu, Yanjun Ma, Hao Tian, Hua Wu, Tian Wu, Wei Zeng, Ge Li, Wen Gao, and Haifeng Wang. ERNIE 3.0 Titan: Exploring Larger-scale Knowledge Enhanced Pre-training for Language Understanding and Generation, 2021. URL https://arxiv.org/abs/2112.12731.

[193] Zichong Wang, Zhibo Chu, Thang Viet Doan, Shiwen Ni, Min Yang, and Wenbin Zhang. History, development, and principles of large language models: an introductory survey. *AI and Ethics*, pages 1–17, 2024.

[194] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and others. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[195] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and others. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[196] Jason White and Joshue O'Connor. Natural Language Interface Accessibility User Requirements. https://www.w3.org/TR/naur/, September 2022. URL https://www.w3.org/TR/naur/. W3C Group Draft Note, 3 September 2022.

[197] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*, 2023.

[198] Rüdiger Wischenbart. *The global eBook market: current conditions & future projections*. " O'Reilly Media, Inc.", 2013.

[199] World Wide Web Consortium (W3C). World Wide Web Consortium – Making the web work for everyone. https://www.w3.org/, 2025. Accessed: 17 June 2025.

[200] World Wide Web Consortium (W3C). Web Content Accessibility Guidelines (WCAG). https://www.w3.org/WAI/standards-guidelines/wcag/, 2025. Last accessed: 17 June 2025.

[201] Liangxuan Wu, Yanjie Zhao, Chao Wang, Tianming Liu, and Haoyu Wang. A First Look at LLM-powered Smartphones. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering Workshops*, pages 208–217, 2024.

[202] Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, 2012.

[203] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[204] Frank Zammetti. Modern Full-Stack Development. *Using TypeScript, React, Node. js, Webpack, and Docker/F. Zammetti.–New York: Apress*, 395:17, 2020.

[205] Staci M Zavattaro, P Edward French, and Somya D Mohanty. A sentiment analysis of US local government tweets: The connection between tone and citizen involvement. *Government information quarterly*, 32(3):333–341, 2015.

[206] Xingxing Zhang and Mirella Lapata. Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931*, 2017.

[207] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and others. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

[208] Zangwei Zheng, Xiaozhe Ren, Fuzhao Xue, Yang Luo, Xin Jiang, and Yang You. Response length perception and sequence scheduling: An llm-empowered llm inference pipeline. *Advances in Neural Information Processing Systems*, 36:65517–65530, 2023.

[209] Aakas Zhiyuli, Yanfang Chen, Xuan Zhang, and Xun Liang. Bookgpt: A general framework for book recommendation empowered by large language model. *arXiv preprint arXiv:2305.15673*, 2023.

[210] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2022.

[211] Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, 2010.

[212] ANDREA Zisman. An overview of XML. *Computing and Control Engineering*, 11(4):165–167, 2000.

Index

Typeset June 17, 2025