An AI-Resilient Text Rendering Technique for Reading and Skimming Documents

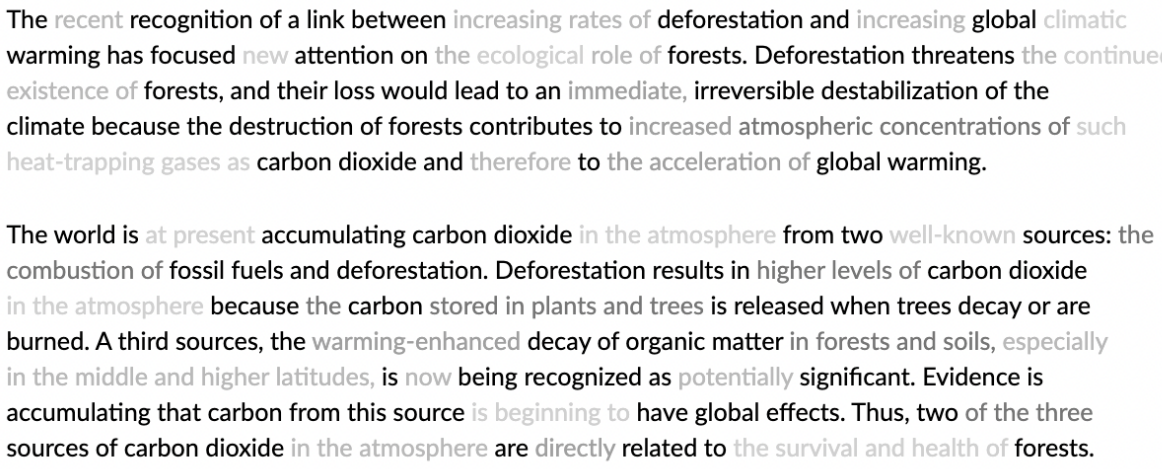
Ziwei Gu
ziweigu@g.harvard.edu
Harvard University
Boston, Massachusetts, USA

Ian Arawjo
ian.arawjo@gmail.com
Harvard University
Boston, Massachusetts, USA

Kenneth Li
ke_li@g.harvard.edu
Harvard University
Boston, Massachusetts, USA

Jonathan K. Kummerfeld
jonathan.kummerfeld@sydney.edu.au
University of Sydney
Sydney, New South Wales, Australia

Elena L. Glassman
glassman@seas.harvard.edu
Harvard University
Boston, Massachusetts, USA



The recent recognition of a link between increasing rates of deforestation and increasing global climatic warming has focused new attention on the ecological role of forests. Deforestation threatens the continued existence of forests, and their loss would lead to an immediate, irreversible destabilization of the climate because the destruction of forests contributes to increased atmospheric concentrations of such heat-trapping gases as carbon dioxide and therefore to the acceleration of global warming.

The world is at present accumulating carbon dioxide in the atmosphere from two well-known sources: the combustion of fossil fuels and deforestation. Deforestation results in higher levels of carbon dioxide in the atmosphere because the carbon stored in plants and trees is released when trees decay or are burned. A third sources, the warming-enhanced decay of organic matter in forests and soils, especially in the middle and higher latitudes, is now being recognized as potentially significant. Evidence is accumulating that carbon from this source is beginning to have global effects. Thus, two of the three sources of carbon dioxide in the atmosphere are directly related to the survival and health of forests.

Figure 1: The Grammar-Preserving Text Saliency Modulation (GP-TSM) rendering technique applied to two paragraphs from a passage from a GRE reading comprehension test. The lighter the text color that each word is rendered in, the earlier it was cut in the backend's recursive sentence compression process. The darkest subset of text can be read as grammatical sentences that preserve as much of the semantic value of the original document as possible, and every successive level of lighter text can be added to these darkest sentences—adding detail without breaking grammaticality.

ABSTRACT

Readers find text difficult to consume for many reasons. Summarization can address some of these difficulties, but introduce others, such as omitting, misrepresenting, or hallucinating information, which can be hard for a reader to notice. One approach to addressing this problem is to instead modify how the original text is rendered to make important information more salient. We introduce Grammar-Preserving Text Saliency Modulation (GP-TSM), a text rendering method with a novel means of identifying what to de-emphasize. Specifically, GP-TSM uses a recursive sentence compression method to identify successive levels of detail beyond the core meaning of a passage, which are de-emphasized by rendering words in successively lighter but still legible gray text. In a lab study (n=18), participants preferred GP-TSM over pre-existing word-level text rendering methods and were able to answer GRE reading comprehension questions more efficiently.

CCS CONCEPTS

• **Human-centered computing → Empirical studies in HCI**.

KEYWORDS

text visualization, human-AI interaction, natural language processing

ACM Reference Format:
Ziwei Gu, Ian Arawjo, Kenneth Li, Jonathan K. Kummerfeld, and Elena L. Glassman. 2024. An AI-Resilient Text Rendering Technique for Reading and Skimming Documents. In *Proceedings of the CHI Conference on Human*

CHI '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0330-0/24/05…$15.00
https://doi.org/10.1145/3613904.3642699

Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA. ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3613904.3642699

1 INTRODUCTION

Readers can find text difficult to consume for a variety of reasons related to the author(s)' choices and the readers' skills and context. First and foremost, there may be a large volume of text relative to the time and attention the reader is willing or able to set aside to read it. In addition, sentences may be long, have ambiguous parses, and/or have a complex structure, e.g., 'garden-path sentences' [113]. The subset of the language used by the author(s) may not have a high degree of overlap with the reader's sight vocabulary [25]. Finally, the reader may be still learning how to read in that language, or the reader may have cognitive differences or conditions that impede reading.

Automated text summarization techniques, including but not limited to crowd-powered systems [10], prompting large language models (LLMs) [105], and other AI technologies, can address a subset of these difficulties, i.e., the resulting text may be shorter, with simpler sentence structures and fewer unusual words [62]. However, unless there is information within the original document that is truly redundant, the result is a lossy representation of the original document, regardless of whether the process is abstractive[1] or extractive[2].

Specifically, automated summarization methods can introduce multiple types of errors: "crimes" of omission, hallucination, and misrepresentation. These methods may judge some details as insufficiently relevant and omit them when they are actually crucial to the reader, given the reader's particular knowledge, context, preferences, values, and task. Some methods may introduce false or irrelevant information that is not derived from the original text, often referred to as hallucinations in the context of generative AI [46, 67, 109].[3] And when summarizers paraphrase or choose what subset of the original text to preserve in the summary, they risk shifting the resulting meaning further away from the original text than the reader would accept, given their goals and context, *if they knew* (i.e., misrepresentation). No single summary can be perfect for every reader because each reader can have their own context, tasks, and tolerances. While personalised summarization has been studied for over a decade [98], it still relies on a coarse characterizations of users [95] and there are key aspects of a reader's interests, context, and task that are unobservable.

Instances of these errors in AI-generated summaries are impossible for readers to notice unless they also read the entire original document. Errors of hallucination tend to look plausible at a glance, errors of omission leave nothing to be noticed in the summary itself,[4] and errors of misrepresentation will only be noticed if they conflict with the readers existing knowledge. Recovering from these AI errors is hard because readers have to (1) first *notice* AI choices that may or may not reflect one of these errors, and (2) have *enough context to judge* whether or not the AI choices reflect any of these errors; these are pre-requisites to the previously proposed human-AI interaction design guidelines [5], i.e., "support efficient dismissal" and "support efficient correction." We call an interface that supports users in noticing, judging, and recovering from AI errors like these *AI-resilient.*

One potentially AI-resilient alternative approach to automated summarization is to instead modify the visual attributes of the original text to support faster reading of the original document (skimming). We propose and evaluate such an approach, which we call Grammar-Preserving Text Saliency Modulation (GP-TSM). Its novelty comes from the computational method used to determine which words in the original text to de-emphasize—and by how much. Specifically, GP-TSM uses a recursive sentence compression method to identify successive levels of detail beyond the core meaning of a passage, which are de-emphasized by rendering words with successively less opacity, e.g., lighter and lighter but still legible gray text when black text is on a white background. The lighter the text color that each word is rendered in, the earlier it was cut in the backend's recursive sentence compression process. We describe the approach as "grammar-preserving" because each subset of each sentence—at any minimum level of opacity the reader chooses to read—remains grammatical, which supports a more natural flow of reading. A formative study in which GP-TSM was semi-automated (with a human in the loop) validated the value it would provide if fully automated.

Prior text rendering methods have computed a variety of functions over words and sentences within a document (from unigram frequency [11, 13] to neural-network-based semantic similarity [35, 100]) and reified the results of that computation into a variety of visual attribute modifications including font attributes [11, 13, 73, 82, 89] and background color [35, 92, 100]. In particular, two previously published ideas that were presented *without evaluation* proposed helping readers skim by reifying unigram frequency: Brath and Banissi [13] use font weight to do so, and Biedert et al. [11] use opactiy. In our final user study, we compare font opacity modulated by GP-TSM to font opacity modulated by unigram frequency (as a control condition we call Word-Frequency Text Saliency Modulation or WF-TSM).

A within-subjects user study (N=18) demonstrates that the final design of GP-TSM not only helps readers complete non-trivial (GRE) reading comprehension tasks more efficiently, it is also strongly preferred over font opacity modulated by unigram frequency (WF-TSM). In summary, we contribute:

- The design and implementation of GP-TSM, a recursive sentence-compression-based text rendering method that supports reading and skimming
- A formative within-subjects user study that demonstrates the value of GP-TSM's text rendering strategy—using a semi-automated sentence-compression backend

[1] A method of summary generation where the system creates a condensed version of the source text using novel sentences. It rephrases the original content to produce a coherent summary, potentially introducing new words and structures.

[2] A method of summary generation where the system selects and extracts whole sentences or fragments directly from the source text to construct the summary. It does not modify the original content but rather curates important segments to form the summary.

[3] The term confabulation from psychology, described as "honest lying"[36], may be a more accurate analogy than hallucination, but the latter is more popular among computer scientists when referring to generative AI.

[4] This interface challenge is analogous to how users cannot recognize the false positives of spam detection algorithms just by looking at their inbox, because the decisions are made silently, leaving no trace in the inbox itself; users have to explicitly look at every email in their spam folder to exhaustively find false positives.

- A summative within-subjects user study that (1) demonstrates the benefits of the fully automated GP-TSM relative to prior text rendering methods and (2) collects evidence that GP-TSM's preservation of grammar at every level of successively grayed text is key.

2 RELATED WORK

Made possible by the capabilities of large language models (LLMs), GP-TSM is an extension of a range of prior work on text summarization and text rendering intended for reading, skimming, and information retrieval support. In this section, we seek to contextualize GP-TSM within the broader narrative of natural language processing (NLP) and the foundational role of earlier research on text rendering and reading and skimming support systems.

2.1 Reading and Skimming

Reading natural language documents can require non-trivial mental effort [44, 45, 75]. In particular, long, complicated sentences can be hard to understand [113]. Studies from the American Press Institute [27] show that when a document's average sentence length is 14 words, readers understand more than 90% of what they are reading. At 43 words, comprehension drops to less than 10%. But long complex sentences remain a common occurrence. For example, recent education research articles written in English had an average sentence length of 24.7 words [23], similar to news text and biomedical text, which average 24.8 and 24.5 words per sentence respectively in standard corpora [53].

Given the cognitive effort reading requires, readers frequently resort to skimming, which is a rapid, selective, and non-linear form of reading [2]. Eye tracking studies [30, 74] validate that such behavior is extremely common. However, multiple studies have suggested a significant trade-off between reading speed and comprehension [65, 66, 76, 87]. In addition, skimming, a skill that takes time to learn and employ effectively, requires strategy and attention [29]. In particular, when skimming in unfamiliar contexts, readers tend to struggle to stay focused, miss key information, and lack confidence in their understanding [28, 60, 102]. Studies have shown that the comprehension of important and unimportant information from a text was equally degraded by an increase in reading rate [16, 31, 50].

2.2 Text Summarization Methods

Text summarization can be either extractive or abstractive. Extractive summarization selects a set of text segments from the original document(s) and combines the segments to form a summary. Note that in these approaches, the summary is entirely composed of verbatim content, i.e., words have been removed but none have been added. The earliest extractive systems selected a set of sentences [58]. More recent work has also compressed and/or merged the sentences that are selected [57]. A range of modeling [26, 43, 47, 59, 96] and learning [72, 91, 107, 110] methods have been explored.

One drawback of the extractive approach is that it can be difficult or impossible to *concisely* capture meaning while only using verbatim content; this is in contrast to abstractive summarization, which generates novel sentences to capture the essence of the content [3, 106, 111]. Abstractive summarization can be more flexible, concise, and human-readable. Historically, extractive summarization was more successful in terms of accuracy and coherence, but recent improvements in natural language generation using LLMs has made abstractive summarization effective and popular [93].

For our specific application, only extractive summarization is suitable. Our goal is to modulate the saliency of words in the *original text* so that users can easily bypass certain words during skimming while maintaining an uninterrupted reading flow. This goal aligns with a specific family of extractive summarization known as sentence compression, or compressive summarization. While traditional extractive summarization predominantly involves selecting whole sentences, compressive summarization aims to select the shortest subsequence of words *within* a sentence that yields an informative and grammatical sentence [64]. This framework allows for a more concise representation of the original content while retaining the essence of its meaning. Various techniques have been developed within this framework [7, 22, 33, 51, 68, 97, 108]. *Notably, our approach introduces a novel feature—a recursive process that generates multiple nested levels of compression, in which information is captured at varying degrees of detail.* In contrast, while there has recently been some work on generating a set of summaries that vary in detail, it has been abstractive, with content at each level that does not overlap [112].

A range of technologies have been applied to summarization, from traditional NLP techniques [3, 8, 34, 69] to large language models (LLMs) [103–105] and even crowd-powered methodologies [10]. The recent improvements in LLMs has significantly increased the quality of summarization methods, including compressive summarization. The summaries they produce are better in terms of coherence, grammaticality, and coverage of critical content [32, 94].

There are a variety of systems that employ summarization within their enhanced reading environments. Specifically, many systems add abstractive and/or extractive summaries to give the reader additional, shorter, possibly simpler text that augments the original content. For example, Paper Plain [6] uses AI models to generate abstractive summaries of each section of a medical paper which is intended to make the science literature more approachable to healthcare consumers. Marvista [19], a human-AI collaborative reading tool, employs an extractive strategy in the "before reading" phase and automatically chooses a summative subset of text for users based on their time budget and questions they want to answer. Marvista then uses AI-generated abstractive summaries to help readers review and recall important information from articles in the "after reading" phase. NEWSLENS [54] describes a quote-extraction based summary using entity extraction and dependency trees to complement news headlines and represent potentially important details from the rest of the article.

However, regardless of the method used, both abstractive and extractive summarization can introduce significant changes in meaning, e.g., through misinterpretation of the input or unintended meanings of the output [3, 109]. To mitigate this lossy nature of summarization methods, our approach supports reading and skimming by adjusting the way text is rendered, keeping all text available to the user to enable recovery from AI errors, or AI decisions that do not suit their needs.

2.3 Text Rendering Modulation Methods

Extensive research has been conducted on text rendering methods to enhance readability, with a significant focus on font attributes. Prior studies have demonstrated that reading performance can improve when using a font that is individually optimal for a reader [9, 18, 24]. However, there is no universally ideal font suitable for all readers, and a reader's optimal font may not always align with their preferred choice [89, 90]. Building upon this line of research, a machine-learning-based model named FontMART has been developed to predict the font that enables the fastest reading speed for an individual [14]. Our work aims to complement this research in readability by focusing on making key information more *salient*, thereby facilitating both focused reading and efficient skimming.

Modulating text saliency is a widely studied aspect of textual information representation. This technique modifies the visual attributes of text to promote words of interest and guide readers' attention, making pertinent information more perceptible and thereby enhancing comprehension and the user experience [12, 42]. We adopt the term "saliency" based on its definition (a "bottom-up, stimulus-driven perceptual quality which makes some items stand out from their neighbors") [42], and its use in augmented reality [85, 88], computer vision [17, 55], and cognitive science [37, 56].

A range of visual strategies have been introduced to promote text saliency in digital reading environments. Brath's *Visual Encoding Pipeline Extended for Text* [12] describes how different types of textual data, including the literal text itself, can be mapped to visual attributes, and then drawn as marks in a layout. Brath and Banissi [13] describe the varied use of typography, or font attributes such as bold, italic, font size and case, on an individual word level to convey information in a way that is intended to facilitate skimming and text analysis. Unlike GP-TSM, multiple visual attributes of text were simultaneously varied rather than just text opacity and no controlled studies were presented that evaluate their effectiveness. Biedert et al. [11] propose a prototype called QuickSkim that assigns a lower text opacity value to non-content words like articles and conjunctions if skimming is detected from eye tracking. Stoffel et al. [82] focus on the size of individual words and create thumbnails by retaining a readable font size of interesting text while shrinking less interesting text. Strobelt et al. [83] surveyed and tested the effectiveness of nine common text highlighting techniques, including various font attributes such as font color, font style and font weight, both individually and in combination. However, their studies involved tasks such as visual interference examination and visual conjunctive search, while our studies focus on reading and skimming. Similarly, Parra et al. [73] explores multiple types of encoding of information on documents, including font size, font luminance, and background color lumination/saturation, but for a different purpose: visualizing neural attention directly on text. Additionally, Shimabukuro et al. [79] explored character-level omissions to dynamically create abbreviations of text and Chevalier et al. [20] explored animation techniques to enhance navigating the revision history of textual documents.

Color as a tool for emphasis and differentiation has been employed in multiple systems [35, 100]. For example, Scim [35] introduced a color-coded system to label different types of (entire) sentences in scientific articles. HiText [100] uses sentence highlighting at various saliencies, where the saliency of the highlight corresponds to the position of the sentence in a list ranked from most to least predicted importance. Semantize [92] conveys multiple predicted attributes, e.g., polarity of emotion and grade level, of specific sentences or paragraphs of a document by rendering them with different visual attributes, e.g., modulating the background color according to predicted polarity of emotion and modulating font size and spacing according to predicted grade level.

GP-TSM is solidly within this existing text rendering modulation tradition. GP-TSM's primary contribution is in its method for computing *what* to de-emphasize. Our approach, recursive sentence compression, enables the visual distinction of *multiple nested grammatical subsets of sentences*. As a result, our novel method of computation also presents a new *scope* of visual attribute modulation.

There are multiple existing computational methods for determining text saliency. Both Brath and Banissi [13] and Biedert et al. [11] weight words based on English language word frequency or document-level unigram frequency. These word-frequency-based methods are optimized for highlighting unique words. Unlike GP-TSM, that does not take into account the core meaning of a text document or the relationship between words within the same sentence. For example, HiText [100] uses a neural-network based model to rank whole sentences according to their predicted importance to the overall document's meaning, while GP-TSM uses an LLM-based model to perform sentence compression recursively until any additional word removals would change the overall meaning more than a threshold amount (among other considerations described in Section 3.4).

Many systems use text rendering modulation methods, highlights, and/or annotations to visualize their analyses of text inplace. For example, Scim [35] helps readers skim scientific articles by highlighting sentences about different key aspects of a paper using different colors, with a density configurable by readers. VarifocalReader [52] automatically annotates and highlights text segments in the detail layer and uses the opacity of the annotation highlight to indicate the confidence value from its support-vector-machine-based active learning component. TextViewer [21], built for literary scholars, renders text with colored underlines to denote text that has been tagged, with the saturation value of the underline corresponding to the absolute value of the tag weight. GP-TSM is versatile and easily integrable, making it suitable for use in these and other contexts, much like the other text rendering methods discussed above.

3 GP-TSM

Building upon the insights gleaned from our review of the challenges of reading and skimming, existing text summarization methods, and existing text rendering modulation methods, this section describes the design and implementation of our proposed text rendering method: GP-TSM. We provide a comprehensive overview of our design goals, the design space, and our design process, followed by an explanation of GP-TSM and its implementation.

3.1 Design Goals

We aspired to design a text rendering interface that alleviates some of the cognitive demands of reading, skimming, or performing information retrieval on natural language documents—particularly those with long, complicated sentences—without compromising the integrity of the original content. From our design explorations, we decided that an effective interface toward that objective should have the following requirements:

(1) **Remain faithful to the original text**. The system should not automatically reword or add new words or phrases to the original text. It should preserve the original text, while rendering it in a way that aids reading, skimming, or information retrieval. This principle of preserving the integrity of the original content is also a primary design goal of a previously developed tool, Doccurate [84], which was developed for healthcare, a domain where precise wording is critical.

(2) **Integrate seamlessly into existing reading experiences**. The system should complement and not interfere with the existing digital reading workflow that people are used to. It should provide all the functionalities in the same view, minimizing the overhead of mode and context switching. This principle also guided HiText [100]; they called this goal "ergonomic unobtrusiveness."

(3) **Support reading at multiple levels of detail**. The system should help users navigate the full complexity of a text, shifting focus seamlessly between different levels of semantic coverage, or granularity [70, 111], from the big picture to the fine details. It should allow users to decide how much detail they want to read and, in case they want a closer read, enable them to do so without requiring any extra action on the user's part, e.g., pressing a button to reveal more detail. Visualizing all levels of detail also means users do not have to guess whether they care enough about what could be hidden in order to decide whether to perform an action to reveal lower levels of detail.

(4) **Support skimming without interrupting flow.** The system should improve skimming of text while minimizing the impact on the user's natural reading flow. In particular, as much as possible, it should avoid presenting users with salient text that is unparsable as a coherent thought, i.e., the system should present a complete sentence rather than a phrase or sentence fragment.

(5) **Be resilient to AI errors by enabling the reader to (a) notice, (b) have enough context to judge, and (c) easily recover from, automated decisions they disagree with.** If the system makes an (automated) judgement call that is inappropriate given the reader's values, preferences, knowledge, context, or task, the reader should be able to recognize that without taking any additional action beyond looking at the interface itself, and proceed without being negatively affected by it. This design goal adds the critical observation that *noticing* an AI's choice and *having enough context to accurately judge* an AI's choice as deserving of preservation or dismissal *are pre-requisites to the previously proposed human-AI interaction design guidelines [5] "support efficient dismissal" and "support efficient correction."* For example, hidden details critical to a reader but judged insufficiently important to keep by an AI cannot be noticed because they are hidden, unless the user is lucky enough to take an action to reveal them and discover that they disagreed with the AI's judgement.

3.2 Design Process

The design of GP-TSM is the result of numerous iterations. To thoroughly explore the design space of skimming support interfaces, we started off by delineating a diverse set of key dimensions and the potential options for each (as detailed in Table 1). To ground our explorations, we constructed prototypes within a browser application, each encompassing different combinations of candidate text attributes, text attribute modulation scopes, interaction techniques, computation scopes, and methods of computing what will be rendered with those text attributes. This approach helped us explore key points within the design space without necessarily implementing all possible feature combinations. In the rest of this section, we describe the process that led to GP-TSM's final features and pivotal design choices, using language consistent with Brath's textbook on textual visualizations [12].

3.2.1 Text Attribute to Modify. Similar to QuickSkim [11], we chose to modulate opacity—which, for black text on a white background, is equivalent to choosing font colors on a gray scale. We chose opactiy over alternatives like background color or stylistic indicators such as italics, typefaces, and underlines, because of our interest in minimising visual distraction. Modulating opacity allows for a graded emphasis on text without disrupting the visual cohesion of the paragraph, offering a smooth reading experience. Since it is a continuous feature, it can be modulated to varying degrees to differentiate multiple levels of detail.

We also found opacity modulation to be a generally intuitive mapping of meaning for users. For example, with black text on a white background, lighter text that has less contrast with the background denotes detail, while darker text signifies criticality. Regardless of the text and background colors, modulating opacity allows the text containing the details to have less and less contrast with the background behind it—"fading away" or moving back "into the background." We tried altering other font attributes, such as font hue or width, but found that their meaning was less clear to participants in early pilot studies.

To fulfill our design goals, we ensure that even the least opaque text is still legible, i.e, consistent with guidelines on contrast ratios provided by WCAG (Web Content Accessibility Guidelines) [15], so that words that the computational method deems to be details are not hidden. (If they were hidden, they would be unnoticeable by a reader who needed them given their context.) Nevertheless, our design may still pose challenges for certain groups of people, which we discuss further in the Discussion section.

Deciding to use opacity instead of similar attributes, like font weight or bolding, was difficult. Bold text inherently demands attention, drawing the reader's eye immediately to those words or phrases [12]. While this is effective for emphasizing certain sections, it is contradictory to our goal of de-emphasizing or suggesting skippability. our approach of fading out text provides a more subtle

Text Attribute	Scope of Attribute Modification	Interaction Technique	Scope of Computation	Extractive Summarization Method
highlight opacity	character	clicking (a button)	sentence	TF-IDF
highlight color	word	clicking (a carousel)	**paragraph**	constituency tree analysis
font opacity	phrase	dragging a slider handle	document	dependency tree analysis
font hue	*(novel)* **nested grammatical subset(s) of the sentence**	gesturing (pinch to zoom)	corpus	linear programming
font size	entire sentence	pressing keys on a keyboard		latent semantic analysis
font weight		scrolling with a mouse		autoencoder
font width		swiping on a touchscreen		**large language model**
oblique		**toggling rendering on/off**		
typeface				
underline				
case				
background color				

Table 1: The design space we explored for interfaces that support reading, skimming, and/or information retrieval, including 5 main parameters and alternative values for each parameter. The space is influenced by Brath [12]'s purpose-agnostic *Visualization Encoding Pipeline Extended for Text.* The bolded items describe the final design of GP-TSM.

indication of detail level without aggressively diverting the reader's focus. Bold can also be visually overpowering and might create visual fatigue over extended reading periods, especially in documents with frequent emphasis changes [48]. In contrast, our less obtrusive fading method enables a more balanced and potentially smooth reading flow.

3.2.2 Scope of Attribute Modification. Our chosen scope of attribution modification could be defined as *word*, but words or even phrases, unlike prior work, are not considered as independent units of analysis. Instead, we consider *grammatical subset(s) of each sentence*, which reveal as many levels of successively smaller detail as the method of computation identifies during its recursive sentence compression process. This is aligned with the design goal of supporting skimming without interrupting flow; one can skim at any minimum level of opacity (skipping over, if one chooses, the words between, which are faded even more) and still be reading coherent sentences that preserve as much of the semantic meaning of the original as possible.

3.2.3 Interaction techniques, if any. Our choice to focus on desktop computers instead of mobile devices was influenced by studies that found improved memory and better performance when people read on desktop computers compared with mobile devices [4, 80]. We conjecture that desktop environments may offer a more conducive setting for extensive reading, particularly for longer and more complex documents, since a larger screen displays more information in a single view without requiring frequent scrolling or zooming. Therefore, we concentrated solely on desktop interactions involving the mouse and keyboard, setting aside interaction alternatives such as swiping and gesturing.

After implementing and piloting a variety of desktop interactive techniques, including sliders, carousels, and amouse scrolling mechanisms for transitioning between hiding/revealing different levels of information granularity, our final design of GP-TSM can just be turned on and off, by keyboard or mouse. Our rationale for this is two-fold: rooted in both our design goal of ensuring seamless integration into existing reading workflows and allowing readers to notice and recover from automated decisions they disagree with. Hiding information interferes with the latter goal, and preliminary studies indicated that the act of choosing levels interfered with the former goal as well. These preliminary studies, which included a mouse-scrolling feature, suggested that such interactive elements could inadvertently disrupt reading, diverting user attention from the primary task of comprehension. We also observed that other interaction methods could overcomplicate the system, which intimidated some users due to the steeper learning curve. By designing a system that automatically determines and displays text saliency without demanding active user adjustments, we aim to reduce the cognitive load required for reading, an already demanding task.

3.2.4 Scope of Computation. We consider entire paragraphs when determining which units of text to modify. This choice, over finer grained (e.g., a sentence) or coarser grained (e.g., a document) alternatives, was motivated by our formative empirical observations when prototyping at each level. When considering each sentence in isolation, relatively little text within the sentence was de-emphasized. This was because we attempted to constrain the core meaning of the un-faded text to be very close to the original text—in this case, the sentence itself.

However, many paragraphs have an overarching single topic, especially in certain kinds of writing like non-fiction. Using the paragraph as the scope of computation provides more leeway to de-emphasize parts of sentences (and sometimes, eventually even entire sentences) while still not straying too far from the overall core meaning of the paragraph. In other words, a typical paragraph is large enough to yield significant amounts of text for de-emphasis, but small enough to have a single coherent theme that is the focus of summarization.

Choosing something larger than paragraphs, such as entire documents, poses the challenge of the computational method making even larger choices about what to de-emphasize that a given reader might disagree with; in other words, it would be deciding on a larger, more noticeable scale which set of ideas are most critical to retain, un-faded. GP-TSM—no matter what scale of decisions an AI is making—allows readers to notice and, without taking any additional action, recover from differences of 'opinion' between the user and the AI, but GP-TSM does not prevent annoyance. The larger the scale that the AI is “getting it wrong” (for the user), the more likely a reader may turn GP-TSM off altogether.

3.2.5 Extractive summarization method. There are many methods of extractive summarization. We started by exploring classical syntax-based methods, but found that parser errors and the limited flexibility of pruning parse trees led to output that was ungrammatical and/or missing key words. Specifically, we tried running a dependency parser and shortening the sentence by removing subtrees based on depth and dependency type, similar to Filappova et al. [34]. This frequently removed important contextual information such as key adjectives and noun phrases. Most existing extractive summarization methods also failed to achieve the desired results as they could not be used to generate our multi-level recursive extractive summary. These observations led us to explore the potential of an LLM-based approach, given their recent improvements [38, 78, 86].

3.3 Overview of the GP-TSM System

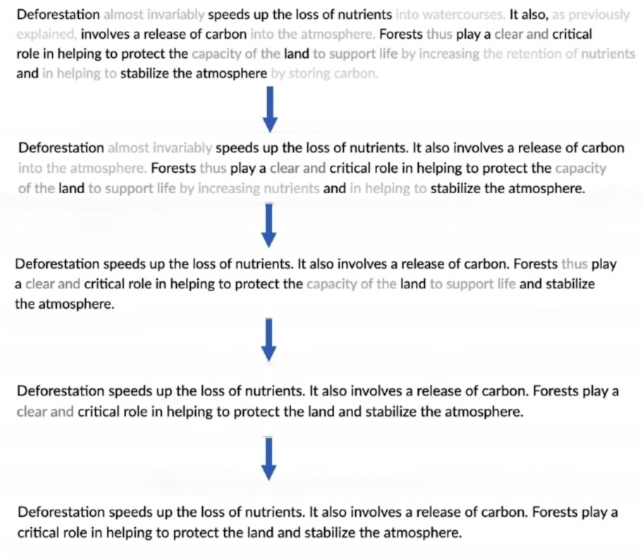
The GP-TSM visualization re-renders plain text at multiple levels of opacity; these levels reveal multiple successive recursive levels of grammatically correct detail within each paragraph. The levels are determined by successively shortening the passage across multiple rounds (Figure 2). Words deleted in the first round are deemed the *least* important, and therefore given the least opacity; words deleted in the second round are deemed slightly more meaningful and appear more opaque; and so on. Words that are never removed remain in full color. In other words, the GP-TSM visualization operates like this: some text, not entirely relevant to the core meaning of a sentence, appears lighter than relatively more important text. When a sentence is artificially long and complicated and full of irrelevant continuations and phrases that add little to the overall meaning, the opacity of certain words and phrases is reduced based on the outcome of successive rounds of shortening. When a sentence is simple, words remain salient.[5]

3.4 Algorithmic Workflow

Producing the GP-TSM visualization for a given passage is nontrivial because it involves ensuring that every level of extraction remains both grammatical and sufficiently close to the core meaning of a passage, for some designer-set threshold and notion of closeness. Our approach is powered by a large language model (LLM). Specifically, we prompt OpenAI’s GPT4 with a single paragraph at a time and ask it to:

“Delete spans of words or phrases from the following paragraph that don’t contribute much to its meaning, but keep readability:
{paragraph}
Please do not add any new words or change words, only delete words.”

Though an LLM-based approach seemed fairly successful in our pilots and within the studies reported, we reflect on the inherent limitations of our choice of using an AI tool, especially its nondeterminism, in the Discussion section.



Deforestation almost invariably speeds up the loss of nutrients into watercourses. It also, as previously explained, involves a release of carbon into the atmosphere. Forests thus play a clear and critical role in helping to protect the capacity of the land to support life by increasing the retention of nutrients and in helping to stabilize the atmosphere by storing carbon.

Deforestation almost invariably speeds up the loss of nutrients. It also involves a release of carbon into the atmosphere. Forests thus play a clear and critical role in helping to protect the capacity of the land to support life by increasing nutrients and in helping to stabilize the atmosphere.

Deforestation speeds up the loss of nutrients. It also involves a release of carbon. Forests thus play a clear and critical role in helping to protect the capacity of the land to support life and stabilize the atmosphere.

Deforestation speeds up the loss of nutrients. It also involves a release of carbon. Forests play a clear and critical role in helping to protect the land and stabilize the atmosphere.

Deforestation speeds up the loss of nutrients. It also involves a release of carbon. Forests play a critical role in helping to protect the land and stabilize the atmosphere.

Figure 2: An illustration of how a paragraph shortens with each round of extraction. Each level stays grammatical after shortening. The increasingly faded text at each level before the final most concise extractive summary show what will be removed at each level; the most faded text at the top level was removed first. What is rendered at the top level in this figure is the only rendering of this process that readers see.

While leveraging an LLM is the computational method behind our recursive sentence compression approach, simply asking an LLM to do this is insufficient on its own for three reasons: (1) sometimes it adds or changes words, (2) the quality of the output varies, and (3) it only provides one set of words to de-emphasize. Our approach incorporates solutions to each of these:

Undoing LLM-inserted words and substitutions. We use a SequenceMatcher[6] to identify words that the LLM has added or changed. These represent rewrites and are hence not allowed as they would mean the user is no longer seeing the original paragraph. We replace substitutions with the original words from the paragraph, and remove insertions; the result is the *post-reversion* LLM response.

Improving output quality. Whenever the LLM generates a shortened paragraph, it may fall short of fulfilling its prompt, e.g., by removing words that lead to grammatical errors; only adding or substituting words; or removing words in a way that changes the meaning of the text too significantly. We address this by prompting the LLM with the same paragraph multiple times (8 times in our case). Empirically, we observed that usually at least one of the eight paragraphs produced was sufficiently high quality for GP-TSM to continue.

To automatically identify the highest quality response, we composed a custom heuristic evaluator. This heuristic evaluation assesses response quality based on a combination of four scores: semantic fidelity, response length, paraphrasing frequency, and

[5] This fading of text, from “In other words...” onwards, was generated directly from our LLM-based method described in Section 3.3 Implementation.

[6] https://docs.python.org/3/library/difflib.html

grammatical correctness. The semantic fidelity score is the similarity between the *original* (pre-summarization) paragraph and the shortened paragraph, calculated using the cosine similarity of their respective embeddings produced by Sentence Transformers [77]. The length score measures how closely the response's length aligns with a preset optimal length, which, based on prototyping, was set to 85% of the previous level's length. The paraphrasing metric quantifies the inverse of detected insertions and substitutions (as determined by a SequenceMatcher)—before such insertions and substitutions were automatically reverted. The grammaticality score involves re-prompting GPT4 to evaluate the syntax of the response after reversion, on a crude scale: 0 for 'bad grammar', 0.5 for 'moderately grammatical,' and 1 for 'grammatically correct.'[7] All four scores are scaled to range from 0 to 1. These scores are combined, via averaging, to produce an overall quality measure of each individual post-reversion LLM-shortened paragraph. Finally, we select the highest scoring option, discard the rest, and proceed to the next level, which takes as input the highest-scoring LLM output that was just chosen.

Identifying multiple levels of relevance. For each paragraph in the given passage, we run multiple rounds of LLM-powered extractive paragraph summarization—each on the results of the previous round—to identify multiple levels of criticality within each paragraph. In each round, we use the methods described above to (a) request 8 responses from GPT4, (b) resolve word addition and substitution, and (c) select the best option using the evaluator. In the first round, the input is the entire paragraph. In subsequent rounds, the input is the best output from the previous round.

This recursive extractive summarization process stops when the LLM "refuses" to cut any words from the summary chosen for the "deepest" level reached so far. We chose this stopping criterion after observing that the LLM will often return the paragraph unchanged if it cannot find additional words to delete, and that this is a better stopping criterion than any other heuristic we experimented with because it is sensitive to the complexity of the original paragraph. More complex paragraphs can accommodate more recursive levels of summarization, while simpler paragraphs may have very few words that can be cut and still maintain grammaticality. Our recursive process stops when no words are deleted in any of the eight summarized paragraphs generated by the LLM.

4 USER STUDIES

We evaluated GP-TSM in two studies—a preliminary user study of the effectiveness of the visualization given a partially automated backend and a summative user study that measures the impact of GP-TSM when fully automated. In every user study, every interface being tested was referred to by an arbitrarily assigned color, e.g., "reader-green" or "reader-blue", a strategy that has been used in prior work, e.g., [81].

4.1 Preliminary User Study

4.1.1 Overview. To understand whether the GP-TSM visualization we propose improves reading comprehension and the reading experience, we first conducted a preliminary user study with 18 participants involving a semi-automated human-supervised version of GP-TSM. In this phase of our work, our aim was to gauge the efficacy of modulating text opacity over nested grammatical subsets of sentences while setting aside concerns about the quality of the backend. In other words, we wanted to verify that grammar-preserving text saliency modulation actually helps, if the eventual fully-automated AI backend is able to perform as well as the human-in-the-loop (partially automated) AI backend we used in this study.

Our decision to employ such a partially automated approach stems from emerging practices in prototyping AI and NLP systems [99, 101], which argue that, given the significant effort and time required to verify output quality of a production-ready AI-powered system, Wizard of Oz-like techniques that employ human-verified AI outputs should be used first before deciding whether to implement the actual AI system.

Our preliminary study evaluates the exact same visualization as the eventual fully automated GP-TSM system, but instead of automatically choosing the best response from GPT4, a human inspector picked the response they believed was best, e.g., had the least rephrasing, and then manually reverted any rephrases in the chosen response. When necessary, the human inspector also edited the response to fix ungrammaticality.

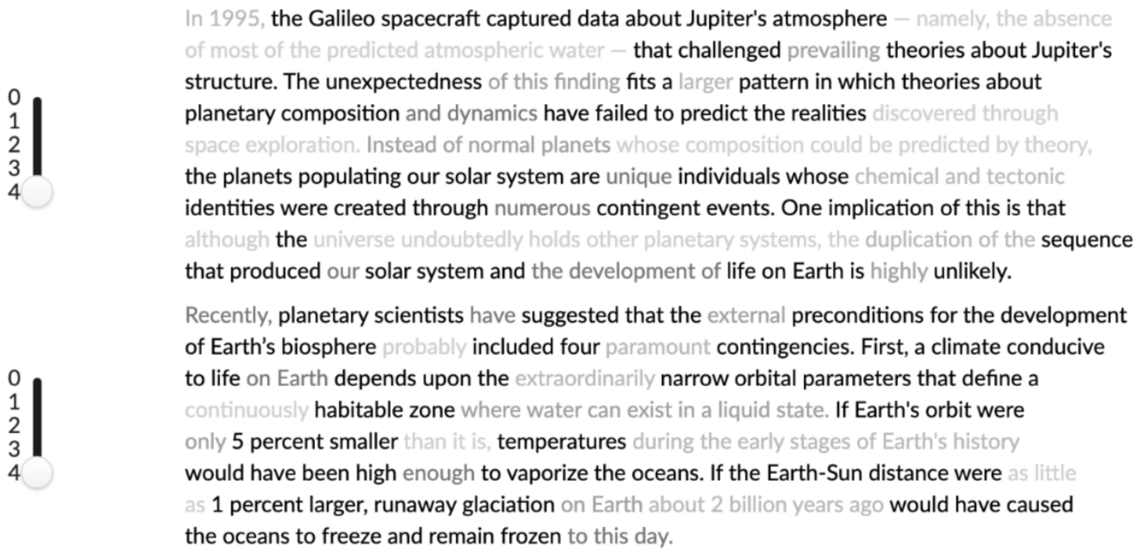
One additional interactive variant was included as an extra condition for comparison in this preliminary study, which was not included in the final fully automated GP-TSM evaluated in the second study. In this variant, a slider or mousewheel affordance can be used to hide text below a certain level of opacity. It, however, partially violates the design goals: even though it is trivial to reveal hidden levels of detail by moving the slider, unless the slider to set to its lowest setting, which is equivalent to the static (and final) version of GP-TSM, it is not possible for the reader to notice and recover from automated decisions they disagree, unless they remember what was hidden.

We were interested primarily in the following questions:

- *How does the GP-TSM visualization affect people in reading and skimming?*
- *What is the user experience like when using the GP-TSM visualization for reading and skimming?*
- *What kinds of value, if any, does interactive granularity control provide for readers?*

In summary, to study these questions, we designed a within-subjects design with three conditions: HITL-GP-TSM (Human-in-the-Loop GP-TSM), HITL-GP-TSM-Interactive (Human-in-the-Loop GP-TSM with interactive granularity), and Control. HITL-GP-TSM is our partially-automated GP-TSM visualization, with only a simple toggle to turn it on and off; HITL-GP-TSM-Interactive added interactive granularity to HITL-GP-TSM; and Control was simply presenting the original plain text. All conditions used the same font and font size (Lato, 14pt). Figure 3 presents a screenshot of the HITL-GP-TSM-Interactive condition.

[7] The prompt we used was: *"Score the following paragraph by how grammatical it is. {paragraph}*
Answer A for grammatically correct, B for moderately grammatical, and C for bad grammar. Only respond with one letter." An A was mapped to a 1, a B was mapped to 0.5, and a C was mapped to 0.



In 1995, the Galileo spacecraft captured data about Jupiter's atmosphere — namely, the absence of most of the predicted atmospheric water — that challenged prevailing theories about Jupiter's structure. The unexpectedness of this finding fits a larger pattern in which theories about planetary composition and dynamics have failed to predict the realities discovered through space exploration. Instead of normal planets whose composition could be predicted by theory, the planets populating our solar system are unique individuals whose chemical and tectonic identities were created through numerous contingent events. One implication of this is that although the universe undoubtedly holds other planetary systems, the duplication of the sequence that produced our solar system and the development of life on Earth is highly unlikely.

Recently, planetary scientists have suggested that the external preconditions for the development of Earth's biosphere probably included four paramount contingencies. First, a climate conducive to life on Earth depends upon the extraordinarily narrow orbital parameters that define a continuously habitable zone where water can exist in a liquid state. If Earth's orbit were only 5 percent smaller than it is, temperatures during the early stages of Earth's history would have been high enough to vaporize the oceans. If the Earth-Sun distance were as little as 1 percent larger, runaway glaciation on Earth about 2 billion years ago would have caused the oceans to freeze and remain frozen to this day.

Figure 3: Screenshot of the HITL-GP-TSM-Interactive interface in the preliminary user study. The (static) HITL-GP-TSM interface is exactly the same, but without the sliders or the responsiveness to mouse scrolling to hide segments of text below a certain level of opacity.

4.1.2 Procedure, Participants, Conditions, and Measures. We recruited 18 participants (8 female and 10 male; 8 between 19-24 years of age and 10 between 25-34 years of age) from university mailing lists at an R1 university in North America. Participants received a $20 Amazon gift card as compensation. Our screen criteria was: *"Participants need to be fluent in English and over 18 years of age."* Participants' self-reported English reading proficiency was relatively high (asked to rate proficiency out of ten, with ten highest: M=8.38 (SD=1.37).

Our study was split into the following parts: informed consent, three sequential reading tasks, and a final survey. Prior to starting each task, each participant went through a short walk-through of the task and affordances of the assigned interface condition. Each participant was given up to 10 minutes to complete each reading task, and asked to complete the task as fast as they could to the best of their ability. The entire study took about 60 minutes.

Each reading task was completed in a separate interface condition (HITL-GP-TSM, HITL-GP-TSM-Interactive, or Control). Participants encountered each interface and each reading task exactly once, and both the reading task order and condition order was counterbalanced across participants. Specifically, we performed a partial counterbalancing of passages to conditions that ensured each passage appeared the same number of times in each condition, and in each condition in each position. Were there any substantial differences in difficulty between passages, this counterbalancing reduces the effect such a difference may have, however we only sampled half of an entire counterbalancing set, which is why subsequent analysis described in the results uses a mixed effects model. We refer to passages as R1, R2, and R3.

We chose Graduate Record Examinations (GRE) passages and reading comprehension questions[8] as our tasks, specifically the 'Long Passages' subsections of the GRE Verbal Reasoning section, each with exactly four questions. They are a relatively standardized measure of reading comprehension; they are specifically designed to require close reading, measure participants' understanding of the text, are standardized to have similar difficulty, and all questions count equally towards the final score [61]. Notably, the three selected passages are of comparable length, with word counts of 472, 446, and 444, respectively.

After each reading task, participants completed a questionnaire to record their reflections on their experience and perceived difficulty of the task in the assigned condition. Questions included an overall rating of the interface and NASA TLX survey questions, and two questions about self-rated task performance. The HITL-GP-TSM-Interactive and HITL-GP-TSM conditions had four additional questions about the visualization; and HITL-GP-TSM-Interactive had another two additional questions about the interactive granularity. After finishing the reading tasks, participants were asked to fill out a post-study survey to indicate their preferences across all three conditions and provide further qualitative feedback. Post-study surveys are provided in Appendix A.

4.1.3 Results. We analyzed reading task results with a three-factor (repeated measures) ANOVA mixed effects model; specifically, investigating each dependent variable on fixed factors *Condition*, *Passage*, and *Order* (the position of the task, first second or third in the sequence) and any interaction effects among these factors, controlling for the random factor of *Participant*. Satterthwaite's

[8] All the passages and questions we used are from publicly available GRE Practice Tests provided by the Educational Testing Service (ETS).

method was used to estimate denominator degrees of freedom. Pairwise comparison with Tukey's HSD (with α=0.05) was conducted between each of the three conditions and three passages. Hereafter, we refer to these methods as ANOVA and Tukey's test, respectively.

ANOVA analysis shows a significant main effect of Condition on reading comprehension scores (p=.03, $F_{2,11.3}$=4.81). Using Tukey's test, we found that, compared to participants using Control, participants with access to interactive granularity (HITL-GP-TSM-Interactive) scored significantly better on reading comprehension questions(p=0.047)—answering approximately three fourths of an additional question correctly out of four. Participants in HITL-GP-TSM were not far behind, though the difference was not significant—they answered approximately half an additional answer correctly out of four, relative to participants in Control. This can be seen in Figure 4 and alternatively in a different type of encoding in Figure 8 (in Appendix B).

ANOVA analysis also found a significant main effect of Condition on time spent completing each reading task (p=.022, $F_{2,10.6}$=5.52). Participants using HITL-GP-TSM completed their reading comprehension questions in only 7.9min (SD=1.9min) on average, which Tukey's test shows was significantly faster (p=0.029) than participants in Control, which completed their reading comprehension questions, on average, 1.4 minutes later, at 9.3min (SD=1.2min). Other tests do not reach significance.

Participants using HITL-GP-TSM-Interactive were not significantly faster than Control, but this may have been due to effects that would ultimately fade with additional use if this were deployed in the wild, i.e., some participants spent some of their time playing with the interactive elements, "trying out different widgets" and "figuring out what exactly the mouse and sliders do."

Participants generally expressed a preference for the 2 GP-TSM conditions over Control. ANOVA analysis shows a significant main effect of Condition on participants' answers to the questions in Table 2, which were asked after each reading task. Tukey's test shows that HITL-GP-TSM received significantly more positive ratings from participants than Control in 5 out of the 9 questions that were asked in both conditions, for *overall experience, how mentally demanding the task was in that condition, how hurried or rushed they felt,, ability to recognize key points in the passage* and *ability to recognize how key points were supported by additional detail.* HITL-GP-TSM-Interactive only received significantly more positive ratings from participants compared to Control on the question about *overall experience.*

These preliminary results verify the usability and helpfulness of the GP-TSM visualization in supporting reading comprehension, suggesting that it would be worthwhile to implement and evaluate a fully automated version of GP-TSM.

The benefits of interactive granularity were less clear cut. While the HITL-GP-TSM-Interactive condition also results in significantly better performances and reading experience than Control, the difference between HITL-GP-TSM-Interactive and HITL-GP-TSM is not significant. Moreover, only HITL-GP-TSM results in both significantly faster task completion and significantly lower perceived difficulty. Therefore, we decided not to carry the feature of interactive granularity forward into the next stage of development.

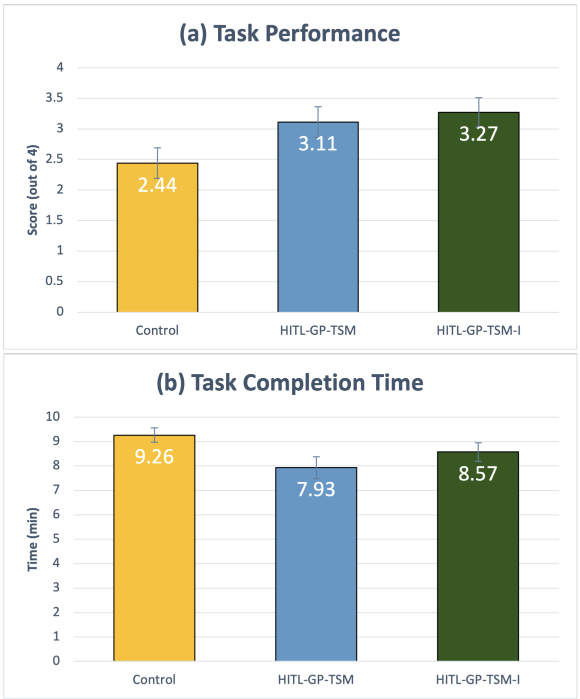


Figure 4: In the preliminary user study, HITL-GP-TSM-Interactive resulted in significantly better performance on the reading comprehension task than Control—on the order of nearly an entire reading comprehension question out of a total of 4, though participants in the HITL-GP-TSM condition were not far behind. In the HITL-GP-TSM condition, participants completed their reading comprehension tasks significantly faster than when using the Control. The error bars represent standard error.

4.2 Main User Study of the fully automated GP-TSM

4.2.1 Overview. After implementing a fully automated version of GP-TSM, described in Section 3.4, we conducted a user study with a separate set of 18 participants to evaluate the efficacy of the fully automated static GP-TSM, using a very similar study format. This time, we were interested primarily in the following questions:

- *How does the fully automated GP-TSM affect reading comprehension? Reading experience?*
- *How does GP-TSM compare to the nearest previously published text saliency modulation method for reading and skimming?*
- *Is the rendering of multiple nested grammatical subsets of sentences resulting from recursive extractive summarization intelligible to users?*
- *What is the impact of preserving the grammaticality of each nested subset of each sentence in GP-TSM provide on users, relative to a version of GP-TSM that does not preserve grammaticality?*

To answer these questions, we modified the preliminary study design in the following ways:

Question Statements	GP-TSM	GP-TSM-I	Control
How would you rate your overall experience in this interface?	5.61 (1.24)*	5.44 (1.25)*	4.06 (1.21)
How mentally demanding was the task? ***[Lower is better (LIB)]***	4.33 (1.37)*	4.56 (1.46)	5.5 (1.29)
How physically demanding was the task? *(LIB)*	1.94 (1.35)	2.28 (1.41)	2.67 (1.71)
How hurried or rushed was the pace of the task? *(LIB)*	2.72 (1.45)*	3.17 (1.62)	4.28 (1.67)
How successful do you think you were in accomplishing the task?	5.11 (1.41)	5.33 (1.14)	4.56 (1.50)
How hard did you have to work to accomplish your level of performance? *(LIB)*	3.94 (1.21)	4.17 (1.20)	4.94 (1.43)
How insecure, discouraged, irritated, stressed, and annoyed were you during the task? *(LIB)*	2.67 (1.50)	3.06 (1.55)	3.78 (1.90)
I could recognize the key points in the passage.	6.11 (1.02)*	5.89 (1.08)	5.0 (1.37)
I could recognize how the key points are supported by additional detail in the passage.	5.89 (1.13)*	5.5 (1.29)	4.78 (1.56)
The system's choice of what to gray out and what to keep at full font weight made sense to me.	5.61 (1.42)	5.78 (1.31)	N/A
I think I know why certain words were lighter than others.	5.67 (1.24)	5.83 (1.25)	N/A
I found it helpful that certain words were lighter than others.	5.44 (1.20)	5.67 (1.28)	N/A
The different levels of gray helped me see the relationships between different parts of sentences.	5.28 (1.23)	5.22 (1.35)	N/A

Table 2: Statistics of scores in the survey after each reading task. For brevity, we use GP-TSM for HITL-GP-TSM and GP-TSM-I for HITL-GP-TSM-Interactive. Participants were asked to rate their agreement with statements related to their reading experience on a 7-point Likert scale from "Strongly Disagree" (1) to "Strongly Agree" (7). The questions 2 through 7 (and their scales) were adapted from the NASA Task Load Index [39]. *"LIB"* stands for *"Lower is better."* Statistics in column 2, 3, and 4 are presented in the form of mean (standard deviation). ANOVA analysis shows a significant main effect of Condition on participants' answers. Statistically significant ($p < 0.05$) differences compared with Control through Tukey's HSD tests are marked with a *. For the last four statements, which concern the text opacity visualization and thus do not apply to the control condition, significance was calculated based on just the two remaining experimental conditions.

First, the interactive granularity condition was replaced with WF-TSM, which we identified as the nearest previously published text saliency modulation method for reading and skimming. As in [11], WF-TSM modulates font opacity, but is based on unigram frequency [11, 13]. In other words, words that appear less frequently are rendered more opaque in WF-TSM, and more frequent words are less opaque. It is worth noting that the percentage of words that are less than fully opaque in WF-TSM is comparable to that in the GP-TSM condition, so any effects we observe are not due to how many words are grayed out, but *which* words are grayed out.

Second, we added a second study component to the end, in which users experience and are asked to reflect on reading the same passage in two different conditions: GP-TSM and a new control, NGP-TSM. NGP-TSM is GP-TSM with grammaticality constraints removed from both places within its workflow: the LLM prompt and the LLM response evaluator. Specifically, the phrase *but keep readability* in the GP-TSM prompt was replaced with *Don't worry about grammar*,[9] and the grammaticality score was removed from the evaluation heuristic, and hence un-enforced. In other words, GP-TSM enforces grammaticality at every minimum level of opacity and NGP-TSM does not; asking participants to compare them helps us answer our research question about the criticality of grammaticality enforcment to the success of GP-TSM.

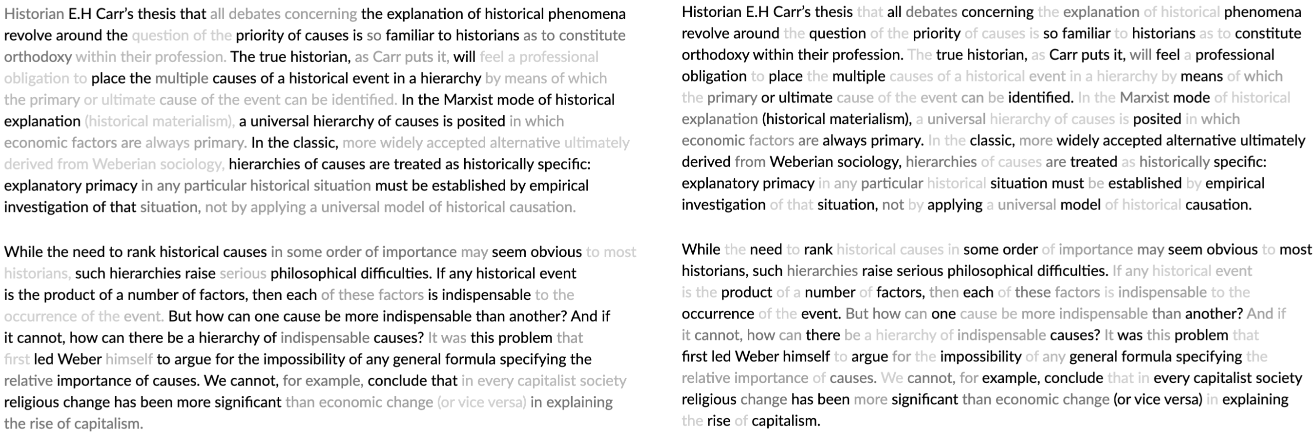
Specifically, this means that after participants finished all the reading tasks and the post-all-reading-tasks survey that were present in both the preliminary user study and this user study, we asked them to participate in a 5-minute survey where we presented them with a view of the same passage rendered twice, side by side, once with GP-TSM and once with NGP-TSM. An example is included in Appendix C. We counterbalanced the presentation order of the two passages to ensure that each appeared on the left and right sides an equal number of times. We then inquired if participants could discern any differences between the two and, if so, to specify those differences. Additionally, we sought their preference between the two visualizations. This part of the study was exploratory and preliminary, meant to give us an indication of the value grammar preservation adds to our system.

All other aspects, including the Control condition, the chosen reading passages and the counterbalancing of conditions, passages, and their respective pairings, remained consistent with the preliminary study. Figure 5 presents screenshots of the GP-TSM and WF-TSM conditions in the main user study, each displaying the same passage.

While exact timing information was not recorded, the fully automated GP-TSM took approximately 2-3 minutes to compute and render the GRE Long Passages texts for each reading task. This time did not affect participants' task time because the results were cached.

4.2.2 Participants. We recruited a separate set of 18 participants (7 self-identified as female, 10 as male, and 1 as non-binary; 7 were

[9] The modified, non-grammar preserving extractive summarization prompt, in its entirety, was: *"Delete spans of words or phrases from the following paragraph that don't contribute much to its meaning. Don't worry about grammar:*
{paragraph}
Please do not add any new words or change words, only delete words."



Historian E.H Carr's thesis that all debates concerning the explanation of historical phenomena revolve around the question of the priority of causes is so familiar to historians as to constitute orthodoxy within their profession. The true historian, as Carr puts it, will feel a professional obligation to place the multiple causes of a historical event in a hierarchy by means of which the primary or ultimate cause of the event can be identified. In the Marxist mode of historical explanation (historical materialism), a universal hierarchy of causes is posited in which economic factors are always primary. In the classic, more widely accepted alternative ultimately derived from Weberian sociology, hierarchies of causes are treated as historically specific: explanatory primacy in any particular historical situation must be established by empirical investigation of that situation, not by applying a universal model of historical causation.

While the need to rank historical causes in some order of importance may seem obvious to most historians, such hierarchies raise serious philosophical difficulties. If any historical event is the product of a number of factors, then each of these factors is indispensable to the occurrence of the event. But how can one cause be more indispensable than another? And if it cannot, how can there be a hierarchy of indispensable causes? It was this problem that first led Weber himself to argue for the impossibility of any general formula specifying the relative importance of causes. We cannot, for example, conclude that in every capitalist society religious change has been more significant than economic change (or vice versa) in explaining the rise of capitalism.

Historian E.H Carr's thesis that all debates concerning the explanation of historical phenomena revolve around the question of the priority of causes is so familiar to historians as to constitute orthodoxy within their profession. The true historian, as Carr puts it, will feel a professional obligation to place the multiple causes of a historical event in a hierarchy by means of which the primary or ultimate cause of the event can be identified. In the Marxist mode of historical explanation (historical materialism), a universal hierarchy of causes is posited in which economic factors are always primary. In the classic, more widely accepted alternative ultimately derived from Weberian sociology, hierarchies of causes are treated as historically specific: explanatory primacy in any particular historical situation must be established by empirical investigation of that situation, not by applying a universal model of historical causation.

While the need to rank historical causes in some order of importance may seem obvious to most historians, such hierarchies raise serious philosophical difficulties. If any historical event is the product of a number of factors, then each of these factors is indispensable to the occurrence of the event. But how can one cause be more indispensable than another? And if it cannot, how can there be a hierarchy of indispensable causes? It was this problem that first led Weber himself to argue for the impossibility of any general formula specifying the relative importance of causes. We cannot, for example, conclude that in every capitalist society religious change has been more significant than economic change (or vice versa) in explaining the rise of capitalism.

Figure 5: Screenshots of the GP-TSM (left) and WF-TSM (right) interfaces in the main user study.

between 18-24 years of age, 10 between 25-34 years of age, and 1 between 35-44 years of age) from university mailing lists at an R1 university in North America. None of them previously participated in the preliminary user study. Participants received a $25 Amazon gift card as compensation. Participants' self-reported English reading proficiency was relatively high, when asked to rate proficiency out of ten, with ten highest: M=8.59 (SD=1.71).

4.2.3 *Quantitative Results.* We used the same statistical analysis process as in the preliminary study (Sec. 4.1.3) to analyze the reading task results and Likert survey questions.

Overall, we find that participants performed significantly better when using GP-TSM compared to Control, and when using Control compared to WF-TSM. Participants also completed tasks significantly faster when using GP-TSM, compared to both WF-TSM and Control (Figure 6 and alternatively Figure 9). Specifically, ANOVA analysis shows a significant main effect of Condition on reading comprehension scores (p=.02, $F_{2,11.3}$=4.79), and Tukey's test shows that participants earned significantly higher scores when using GP-TSM compared to participants using Control (p=.021). Tukey's test also shows that WF-TSM was significantly worse than Control (p=.045).

ANOVA analysis shows a significant main effect of Condition on task times while achieving these reading comprehension scores (p=.0026, $F_{2,10.6}$=6.72). Tukey's test shows that, when using GP-TSM, it took participants, on average 8min 7s to complete the reading task (SD=1min 36s), which was, on average, about a minute faster than when using the Control (M=9.25min, SD=1min 4s)(p=0.019). Participants using WF-TSM were slightly slower than the Control by 15 seconds on average (M=9.5min, SD=49s), though that difference was not significant. The difference between GP-TSM and WF-TSM, however, was still significant (p=0.003).

Table 3 shows participants answers to questions asked immediately after each reading task, with some exceptions when the questions are irrelevant in a given condition. ANOVA analysis shows a significant main effect of Condition on their answers. Overall, participants generally expressed preference for GP-TSM over WF-TSM and Control. Tukey's test shows that GP-TSM received significantly more positive ratings from participants than Control in 4 out of the 9 questions that were asked in both conditions (p<0.05), for *overall experience, how mentally demanding the task was in that condition, how hard they had to work in that condition,* and *recognizing key points in the passage.* GP-TSM also received significantly more positive ratings from participants than WF-TSM in 6 out of the 9 questions that were asked in both conditions (p<0.05). These questions included all of the same questions that were significant for the GP-TSM-Control comparison and additionally included

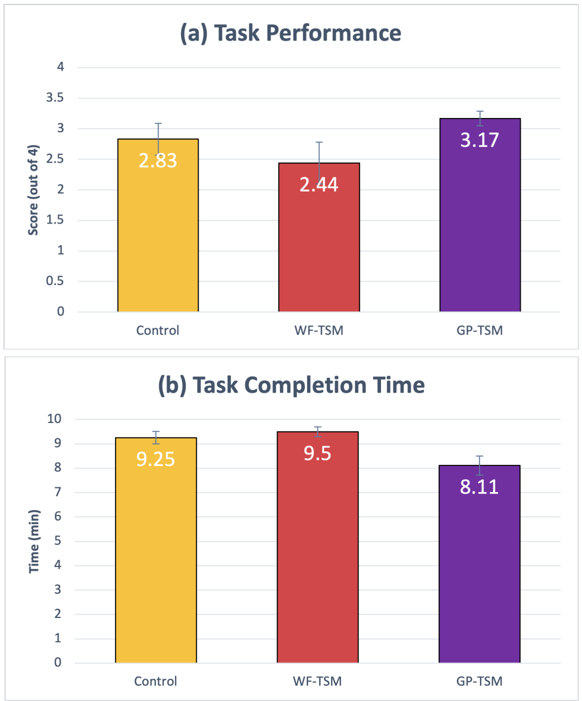


Figure 6: Participants performed significantly better and significantly faster in the reading comprehension task when using GP-TSM compared to the control conditions in the user study. The error bars represent standard error.

Question Statements	**GP-TSM**	**WF-TSM**	**Control**
How would you rate your overall experience in this interface?	5.52 (1.45)*†	4.1 (1.19)	4.35 (1.32)
How mentally demanding was the task? ***[Lower is better (LIB)]***	3.98 (1.46)*†	5.21 (1.48)	5.13 (1.3)
How physically demanding was the task? *(LIB)*	1.91 (1.34)	1.98 (1.54)	1.85 (1.62)
How hurried or rushed was the pace of the task? *(LIB)*	3.15 (1.88)	4.49 (1.91)	4.35 (1.72)
How successful do you think you were in accomplishing the task?	5.09 (1.37)*	3.87 (1.56)	4.61 (1.48)
How hard did you have to work to accomplish your level of performance? *(LIB)*	3.65 (1.16)*†	4.97 (1.52)	4.78 (1.43)
How insecure, discouraged, irritated, stressed, and annoyed were you during the task? *(LIB)*	2.32 (1.38)*	4.57 (2.1)	2.29 (1.31)
I could recognize the key points in the passage.	6.16 (1.04)*†	5.12 (1.11)	5.23 (1.06)
I could recognize how the key points are supported by additional detail in the passage.	5.9 (1.09)	4.98 (1.25)	5.38 (1.46)
The system's choice of what to gray out and what to keep at full font weight made sense to me.	5.91 (1.67)*	2.38 (1.44)	N/A
I think I know why certain words were lighter than others.	5.36 (1.14)*	4.83 (1.21)	N/A
I found it helpful that certain words were lighter than others.	5.67 (1.81)*	3.12 (1.33)	N/A
The different levels of gray helped me see the relationships between different parts of sentences.	5.19 (1.22)*	2.41 (1.25)	N/A

Table 3: Statistics of scores in the survey after each reading task. Participants were asked to rate their agreement with statements related to their reading experience on a 7-point Likert scale from "Strongly Disagree" (1) to "Strongly Agree" (7). Questions 2 through 7 (and their scales) were adapted from the NASA Task Load Index [39]. *"LIB"* stands for *"Lower is better."* Statistics in column 2, 3, and 4 are presented in the form of mean (standard deviation). ANOVA analysis shows a significant main effect of Condition on participants' answers. Statistically significant (p < 0.05) differences compared with WF-TSM and Control are marked with * and †, respectively. For the last four statements, which concern the text opacity visualization and thus do not apply to the control condition, significance was calculated based on the two experimental conditions.

how successful they thought they were and *how insecure, discouraged, etc. they felt.* Finally, in the questions which were only asked in the GP-TSM and WF-TSM conditions because they asked specifically about opacity modulation which was not present in Control, GP-TSM received significantly better Likert scale ratings than WF-TSM for all 4 questions (last 4 rows of Table 3), which were all about the *intelligibility of why certain words were less salient* and their *helpfulness*, especially for *seeing the relationships between different parts within sentences.*

After experiencing all the conditions, participants were asked to rate their agreement on a 7-point scale (7 being the highest) with the following statement for each condition: *"I would like to use [Condition] to read online text of interest to me in the future"*. GP-TSM received a mean agreement score of 5.8 (SD=1.8) while Control received a mean agreement score of 4.7 (SD=1.3) and WF-TSM received a mean agreement score of 3.3 (SD=2). This difference was significant ($p<0.05$) between GP-TSM and both Control and WF-TSM using additional pairwise unpaired t-tests.

After experiencing all the conditions, participants were also asked to directly rank conditions. Participants expressed a strong preference for GP-TSM over WF-TSM and Control (Figure 7).

4.2.4 Qualitative Feedback. Overall, participants were positive about GP-TSM and its functionality. Below, we group participants' responses to the survey questions[10] around a set of themes that were frequently mentioned.

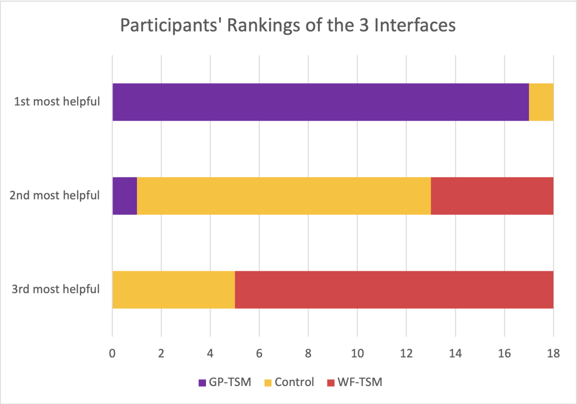


Figure 7: Participants' rankings of the 3 conditions in terms of their helpfulness for reading comprehension tasks

Improved Reading Efficiency. Twelve out of 18 participants appreciated how text saliency was modulated on the GP-TSM interface, noting that it facilitated more efficient reading by letting them skip words but still grasp the gist of the passage (P1, P2, P3, P5, P6, P7, P9, P12, P14, P15, P16, P18). For example, P2 wrote, *"My reading became faster and less congested [using GP-TSM] because I could easily skip over and ignore words that were grayed out if I simply wanted to get the main idea of the passage."* Further, P1 commented on how easy it was for *"the structural logic of the passage to be absorbed"* given GP-TSM's multi-level visualization.

[10] What did you like about the interface?
What did you not like about the interface?
What do you wish the interface had?

As for the mechanism by which reading efficiency was improved, participants provided interesting reflections on how GP-TSM shaped their reading experience. It seemed that GP-TSM facilitated a two-step reading process where participants *"first skimmed the black words for a quick grasp of the gist and then went back to certain sections to read more details"* (P17). For instance, P6 observed, *"The interface almost allowed me to read the passage in two ways. One way was to read each word, regardless of color like normal. But, alternatively, if I just read the black text I got the crux of the argument with none of the additional filler."* P10 reflected on their question answering procedure specifically: *"I was better able to search back through the text to find key words or ideas that related to the questions I was trying to find answers for. I felt that I could continuously read the bold words and they formed understandable sentences."*

Sensible Visualization. Participants highlighted that the graying was *"pretty consistent and reasonable"* to them (P9), with a lot of comments on how they agreed with what the system chose to gray (P2, P7, P12, P13, P16, P17), though P14 complained about the fact that sometimes *"certain transition words are in gray,"* which *"prevents [P14] from picking up the transition logic between two elements in a sentence, or between two sentences."*

Participants also commented on the text visualization itself, pointing out that the graying was *"natural"* (P11) and *"not too much drama but provided just the right amount of contrast for engaging reading"* (P14). Overall, participants found that the system's design helped them focus on key points and read more efficiently (P1, P3, P7, P8, P16, P17). Some mentioned they were excited about using GP-TSM in the future (P11: *"I would want to use [GP-TSM] to read my history readings."*), possibly as a Chrome extension.

Explanation Needed. Despite a broad preference for the GP-TSM interface, participants also offered several suggestions for improvement. A recurrent theme was the desire for more explicit guidance, which reveals that not all participants quickly and intuitively grokked what the levels of gray meant. For example, P2 suggested that *"a prompt that explained why some words were grayed out could have been helpful."* Similarly, P12 noted that *"it would have helped to have a tutorial to understand why some text was more gray."*

Readability. Others offered legibility-related suggestions and requests for customization about the visual attribute being modified. For instance, P6 suggested using *"entirely different colors like RGB"* instead of shades of gray. P16 complained that *"the lightest gray text was a bit difficult to read; I wish it could be a little darker."* There are also font-size-related legibility concerns. For example, P12 thought *"the text was too small and close together,"* and P14 wanted to see *"an A+/A- icon by the side,"* so they could enlarge the font.

Interactivity. P2 and P15 suggested additional features to interact with the grayed-out words. P15 proposed a slider to allow users to *"customize the degrees of graying"* and P2 wished they could *"put away the grayed out words"* entirely so they could focus on reading the words in black (a feature previously supported by HITL-GP-TSM-Interactive).

4.2.5 GP-TSM vs. NGP-TSM: the Comparison Interview. As described in Sec. 4.2, after completing all the reading tasks and reflecting on the three conditions of GP-TSM, WF-TSM, and Control, all the participants looked at the same passage shown twice, side by side, rendered with GP-TSM and NGP-TSM—one enforcing grammaticality at every minimum level of opacity and the other not. Fifteen (15) out of all 18 participants perceived a difference between the grammar-preserving and non-grammar-preserving renderings, although some could not specify exactly what the difference was. For instance, P2 noticed, *"[GP-TSM] seems to gray out longer chunks of text, while [NGP-TSM] grays out a lot of single words."* P11 mentioned, *"In [GP-TSM], the transition is much more natural. In [NGP-TSM], honestly, I don't understand why certain words are in gray."* P17 reported, *"I actually didn't feel that much a difference, but I seemed to have an easier time reading in [GP-TSM]."*

Half (9) of the 18 participants successfully identified, to varying degrees, that the key difference was in grammaticality. While some only sensed the difference, others were able to articulate specifically the grammatical errors in the NGP-TSM case. For example, P9 successfully observed, *"[NGP-TSM] grays out many articles, prepositions, and other determiners, while [GP-TSM] doesn't."* Similarly, P18 elaborated, *"[NGP-TSM] grays out a lot of 'the', 'a' and 'to', which is a little bit annoying to me. Those words may not carry much meaning, but they are still important to the structure of sentences."*

As for user preferences, all 15 who perceived the difference between the two interfaces preferred GP-TSM to NGP-TSM because they felt it enabled them to achieve better comprehension and higher reading efficiency. For instance, P13 said, *"I like [GP-TSM] better. It just makes more sense to me. When I skipped the gray parts I still understood everything."* P16 explained, *"I prefer [GP-TSM] because I can completely skip words in gray here but [in NGP-TSM] I still have to read some of the gray text to understand what is going on."* In summary, this part of the study provides evidence that GP-TSM's preservation of grammar at every level is key to the observed improvement in reading efficiency and user preference.

5 DISCUSSION

These user studies demonstrate the benefits of Grammar-Preserving Text Saliency Modulation (GP-TSM) for English reading comprehension. Participants responded positively to the chosen visual text attribute to be modified, i.e., text opacity, and especially strongly to the strategy by which text opacity was modulated, i.e., nested grammatical subsets of sentences that revealed layers of detail around the core of each sentence. One participant had reservations about the lack of predicted importance that the backend recursive extractive summarization process often assigned to transition words; this is evidence that the design goal concerning participants' ability to notice and recover in situations when they disagree with an automated judgement has been fulfilled, i.e., "AI-resilience". In spite of that keen observation, which was possible due to GP-TSM's design, there was a general consensus that GP-TSM's choices about which sections to gray out were superior to WF-TSM, which is the nearest alternative method of text saliency modulation in the literature. Notably, the impact of the grammaticality enforced within GP-TSM's backend workflow was clearly perceived by most users; it garnered attention and praise in interviews when participants could see

GP-TSM side by side with its non-grammaticality-enforcing twin, NGP-TSM. Multiple measures suggest that GP-TSM enhances reading efficiency, overall user experience, and reduces the perceived difficulty of reading.

By far, our most compelling quantitative evidence are the gains in performance and decreases in task time when using GP-TSM, compared to the controls. Participant performance on standardized test questions is less subjective than self-reported efficacy, which can be affected by social and cognitive biases, such as the lab setting, wanting to please the researchers or guessing the hypotheses. When the relative gains in efficiency are considered alongside findings from post-task surveys and qualitative feedback, there is strong evidence that GP-TSM, as a visualization tool, supports faster and improved reading comprehension for English readers.

The beauty of the GP-TSM technique lies in its simplicity: at its core, all GP-TSM does is change the visual saliency of words by adjusting their opacity. This preserves the integrity of the original text and minimizes "ergonomic obtrusiveness" [100] while providing readers with a form of "contextual cuing" to arm them with "incidental knowledge about global context", which they can harness to better assign visual attention and memory when reading [40]. By showing multiple levels of detail at once with successively less opacity, GP-TSM empowers readers to freely choose their level of engagement with the material. By preserving grammaticality at each level, GP-TSM supports a coherent reading and skimming experience. The evidence from our user studies indicates that all our design goals were fulfilled.

Reflecting on the number of levels of opacity and their visual distinction, we encountered a tradeoff. We aimed for the least significant level to remain legible to ensure no loss of information, while also enabling clear differentiation among levels to allow readers to select and consistently engage with the level they consider the most suitable. However, the perceptibility of the differences among levels becomes challenging in complex sentences with many levels. Furthermore, according to Stevens's power law, people perceive changes in gray scale not linearly, but rather by a factor of approximately 0.5 [71]. For instance, a threefold increase in opacity might only be perceived as 1.5 times more significant, further complicating the differentiation of levels. This issue is reflected in feedback: some participants struggled to read the lightest gray text, while others had some difficulty discerning the various levels and understanding how they elucidate the relationships between different parts of sentences.

Reflecting on the user experience, an intriguing transformation in reading patterns emerged from the feedback. Many participants pointed out, in one way or another, a two-step reading process that GP-TSM interface seems to promote. Initially, readers focused on the darker, more salient text to grasp the primary narrative or theme of the passage. This 'overview' phrase of reading gave them a framework or scaffold of the content. Subsequently, they revisited the passage to delve into the grayed-out sections, filling in details where the questions were asked or their interest was piqued. This sequence resonates with efficient content absorption strategies highlighted in speed reading literature, where readers first capture the gist and then delve deeper [1, 63]. The interface, therefore, may inadvertently facilitate this structured, layered reading approach, which might explain the improvement in reading efficiency and comprehension.

6 LIMITATIONS AND FUTURE WORK

Reflecting on our technical approach, opting for an LLM-based backend enhanced the quality of the extractive summaries, but sacrificed speed and transparency. The black-box nature of LLMs reduces the transparency of the decisions they make, and their complexity slows down the system, potentially impacting future deployment of GP-TSM to real-time reading scenarios. In particular, our choice of GPT-4 might limit the potential applications of our system due to data privacy concerns [49]; future work targeted at sensitive data should consider other open-source models that respect data privacy. In addition, the inherent non-determinism of LLMs can lead to variations in outputs for similar inputs, adding another layer of unpredictability in the LLM's responses. Although we partially mitigate this by requesting multiple responses and picking the best one algorithmically, our heuristic-based approach is not foolproof and may occasionally miss the most contextually relevant or coherent response. Despite these drawbacks, we still stick to an LLM-based approach because our primary focus at this stage remains optimizing the accuracy and relevance of text saliency modulation, which is currently best produced by LLM-based recursive extractive summarization at the paragraph level. As LLMs continue their current trend of advancement, we expect GP-TSM to continue to improve in quality and speed, making it increasingly feasible to use—as one participant explicitly requested—as a Chrome extension.

Beyond the challenges posed by LLMs, our study also faces several other limitations. First, the limited sample size and sampling procedure could have skewed our conclusions due to a lack of diversity in participant background. Future evaluations of GP-TSM should actively include a wider array of participants, such as younger or older age groups, users with varying educational backgrounds, and individuals from different cultural and linguistic contexts. These groups may encounter distinct challenges or exhibit different interaction patterns with GP-TSM: age-related differences in technology adoption and comprehension skills, cultural variations in text interpretation, and educational disparities in reading abilities could all significantly impact the effectiveness of GP-TSM. Expanding our understanding of these diverse user experiences is critical to a comprehensive understanding of the utility of GP-TSM across a broader spectrum of users. Moreover, recent work [41] has identified needs of those with cognitive impairments, as well as possible directions for text tools to support them, such as helping readers *prioritize* what to read. The evidence collected so far indicates that GP-TSM may fulfill that need, but future evaluations of GP-TSM should engage participants from that specific group to determine if GP-TSM offers advantages for that community.

Second, our measure of reading comprehension relied upon long passages from the GRE test, and how well GP-TSM generalizes to other text styles and formats is yet unknown. This raises questions about the adaptability of GP-TSM across various genres and complexities of text, such as technical manuals, legal documents, or everyday communication. Further, although our user study empirically evaluates the usability and usefulness of GP-TSM, we rely solely on participants' accounts of their interactions to understand *how* they used GP-TSM, which could be subject to bias. Follow-up work could use eye-tracking studies to provide insights into how GP-TSM shapes users' reading and skimming patterns. Finally, while we adhere to the guidelines provided by WCAG (Web Content Accessibility Guidelines) on contrast ratios of text, we acknowledge that modulating font opacity can make text less legible, and thus less accessible, especially to those with visual impairments.

Finally, we believe it is important to continue to explore the design space of *AI-resilient* interfaces. Our understanding is that GP-TSM is AI-resilient because, given that none of the original text is removed or rearranged, the errors of omission, hallucination, and misrepresentation instead show up as automated text attribute choices the reader disagrees with, and these automated choices are noticeable, presented with all the necessary context for the reader to judge because: (1) Text attribute changes are always visible in the interface (i.e., no automated choice results in something hidden and therefore difficult to notice). (2) The reader is still looking at the original text so they have all the context they need to choose for themselves whether they agree with each automated choice or not (and what it implies about the text, e.g., whether that segment of text is particularly important or not). Generalizing this notion of AI-resiliency to additional tasks and domains is, we believe, important and exciting future work.

ACKNOWLEDGMENTS

We appreciate the engagement of all our participants. This material is based upon work supported by the National Science Foundation under Grant No. IIS-2107391.

REFERENCES

[1] Mahmoud Sulaiman Hamad Bani Abdelrahman and Muwafaq Saleem Bsharah. 2014. The Effect of Speed Reading Strategies on Developing Reading Comprehension among the 2nd Secondary Students in English Language. *English Language Teaching* 7, 6 (2014), 168–174.

[2] Denise E Agosto. 2002. Bounded rationality and satisficing in young people's Web-based decision making. *Journal of the American society for Information Science and Technology* 53, 1 (2002), 16–27.

[3] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268* (2017).

[4] Mesfer Alrizq, Sara Mehmood, Naeem Ahmed Mahoto, Ali Alqahtani, Mohammed Hamdi, Abdullah Alghamdi, and Asadullah Shaikh. 2021. Analysis of Skim Reading on Desktop versus Mobile Screen. *Applied Sciences* 11, 16 (2021), 7398.

[5] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–13.

[6] Tal August, Lucy Lu Wang, Jonathan Bragg, Marti A Hearst, Andrew Head, and Kyle Lo. 2022. Paper plain: Making medical research papers approachable to healthcare consumers with natural language processing. *arXiv preprint arXiv:2203.00130* (2022).

[7] Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 481–490.

[8] Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly Learning to Extract and Compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 481–490. https://aclanthology.org/P11-1049

[9] Michael Bernard, Chia Hui Liao, and Melissa Mills. 2001. The effects of font type and size on the legibility and reading time of online text by older adults. In *CHI'01 extended abstracts on Human factors in computing systems*. 175–176.

[10] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. 313–322.

[11] Ralf Biedert, Georg Buscher, Sven Schwarz, Jörn Hees, and Andreas Dengel. 2010. Text 2.0. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI EA '10)*. Association for Computing Machinery, New York, NY, USA, 4003–4008. https://doi.org/10.1145/1753846.1754093

[12] Richard Brath. 2020. *Visualizing with text*. CRC Press.

[13] Richard Brath and Ebad Banissi. 2014. Using font attributes in knowledge maps and information retrieval. In *CEUR Workshop Proceedings*, Vol. 1311. CEUR Workshop Proceedings, 23–30.

[14] Tianyuan Cai, Shaun Wallace, Tina Rezvanian, Jonathan Dobres, Bernard Kerr, Samuel Berlow, Jeff Huang, Ben D Sawyer, and Zoya Bylinskii. 2022. Personalized Font Recommendations: Combining ML and Typographic Guidelines to Optimize Readability. In *Designing Interactive Systems Conference*. 1–25.

[15] Ben Caldwell, Michael Cooper, Loretta Guarino Reid, Gregg Vanderheiden, Wendy Chisholm, John Slatin, and Jason White. 2008. Web content accessibility guidelines (WCAG) 2.0. *WWW Consortium (W3C)* 290 (2008), 1–34.

[16] Ronald P Carver. 1984. Rauding theory predictions of amount comprehended under different purposes and speed reading conditions. *Reading Research Quarterly* (1984), 205–218.

[17] Antonio Cedillo-Hernandez, Manuel Cedillo-Hernandez, Mariko Nakano Miyatake, and Hector Perez Meana. 2018. A spatiotemporal saliency-modulated JND profile applied to video watermarking. *Journal of Visual Communication and Image Representation* 52 (2018), 106–117.

[18] Maneerut Chatrangsan and Helen Petrie. 2019. The effect of typeface and font size on reading text on a tablet computer for older and younger people. In *Proceedings of the 16th International Web for All Conference*. 1–10.

[19] Xiang'Anthony' Chen, Chien-Sheng Wu, Tong Niu, Wenhao Liu, and Caiming Xiong. 2022. Marvista: A Human-AI Collaborative Reading Tool. *arXiv preprint arXiv:2207.08401* (2022).

[20] Fanny Chevalier, Pierre Dragicevic, Anastasia Bezerianos, and Jean-Daniel Fekete. 2010. Using text animated transitions to support navigation in document histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 683–692.

[21] Michael Correll, Michael Witmore, and Michael Gleicher. 2011. Exploring collections of tagged text for literary scholarship. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 731–740.

[22] Shrey Desai, Jiacheng Xu, and Greg Durrett. 2020. Compressive summarization with plausibility and salience modeling. *arXiv preprint arXiv:2010.07886* (2020).

[23] T Deveci. 2019. Sentence length in education research articles: A comparison between Anglophone and Turkish authors. *The Linguistics Journal* 14, 1 (2019), 73–100.

[24] Jonathan Dobres, Nadine Chahine, Bryan Reimer, David Gould, Bruce Mehler, and Joseph F Coughlin. 2016. Utilising psychophysical techniques to investigate the effects of age, typeface design, size and display polarity on glance legibility. *Ergonomics* 59, 10 (2016), 1377–1391.

[25] Edward W Dolch. 1936. A basic sight vocabulary. *The Elementary School Journal* 36, 6 (1936), 456–460.

[26] Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. BanditSum: Extractive Summarization as a Contextual Bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 3739–3748. https://doi.org/10.18653/v1/D18-1409

[27] William H DuBay. 2004. The principles of readability. *Online Submission* (2004).

[28] Geoffrey B Duggan and Stephen J Payne. 2006. How much do we understand when skim reading?. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*. 730–735.

[29] Geoffrey B Duggan and Stephen J Payne. 2009. Text skimming: The process and effectiveness of foraging through text under time pressure. *Journal of experimental psychology: Applied* 15, 3 (2009), 228.

[30] Geoffrey B Duggan and Stephen J Payne. 2011. Skim reading by satisficing: evidence from eye tracking. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1141–1150.

[31] Mary Dyson and Mark Haselgrove. 2000. The effects of reading speed and reading patterns on the understanding of text read from screen. *Journal of research in reading* 23, 2 (2000), 210–223.

[32] Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *arXiv preprint arXiv:2304.01746* (2023).

[33] Katja Filippova, Enrique Alfonseca, Carlos A Colmenares, Łukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 conference on empirical methods in natural language processing*. 360–368.

[34] Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*. 25–32.

[35] Raymond Fok, Hita Kambhamettu, Luca Soldaini, Jonathan Bragg, Kyle Lo, Marti Hearst, Andrew Head, and Daniel S Weld. 2023. Scim: Intelligent Skimming

Support for Scientific Papers. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. 476–490.

[36] Asaf Gilboa and Morris Moscovitch. 2002. The cognitive neuroscience of confabulation: A review and a model. *Handbook of memory disorders* 2 (2002), 315–342.

[37] Hadi Hadizadeh. 2016. A saliency-modulated just-noticeable-distortion model with non-linear saliency modulation functions. *Pattern Recognition Letters* 84 (2016), 49–55.

[38] Thilo Hagendorff and Sarah Fabi. 2023. Human-Like Intuitive Behavior and Reasoning Biases Emerged in Language Models–and Disappeared in GPT-4. *arXiv preprint arXiv:2306.07622* (2023).

[39] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage publications Sage CA: Los Angeles, CA, 904–908.

[40] Christopher Healey and James Enns. 2011. Attention and visual memory in visualization and computer graphics. *IEEE transactions on visualization and computer graphics* 18, 7 (2011), 1170–1188.

[41] Hendrik Heuer and Elena L. Glassman. 2023. Accessible Text Tools: Where They Are Needed & What They Should Look Like. In *Extended abstracts of the 2023 CHI conference on human factors in computing systems*. 1–7. https://doi.org/10.1145/3544549.3585749

[42] Laurent Itti. 2007. Visual salience. *Scholarpedia* 2, 9 (2007), 3327.

[43] Aishwarya Jadhav and Vaibhav Rajan. 2018. Extractive Summarization with SWAP-NET: Sentences and Words from Alternating Pointer Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 142–151. https://doi.org/10.18653/v1/P18-1014

[44] Stefan Jänicke, Thomas Efer, Marco Büchler, and Gerik Scheuermann. 2014. Designing close and distant reading visualizations for text re-use. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics*. Springer, 153–171.

[45] Stefan Jänicke, Greta Franzini, Muhammad Faisal Cheema, and Gerik Scheuermann. 2015. On Close and Distant Reading in Digital Humanities: A Survey and Future Challenges. *EuroVis (STARs)* 2015 (2015), 83–103.

[46] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *Comput. Surveys* 55, 12 (2023), 1–38.

[47] Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. Neural Extractive Summarization with Hierarchical Attentive Heterogeneous Graph Network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 3622–3631. https://doi.org/10.18653/v1/2020.emnlp-main.295

[48] Elizabeth Keyes. 1993. Typography, color, and information structure. *Technical communication* (1993), 638–654.

[49] Sunder Ali Khowaja, Parus Khuwaja, and Kapal Dev. 2023. ChatGPT Needs SPADE (Sustainability, PrivAcy, Digital divide, and Ethics) Evaluation: A Review. *arXiv preprint arXiv:2305.03123* (2023).

[50] Dina Kiwan, Ayesha Ahmed, and Alastair Pollitt. 2000. *The effects of time-induced stress on making inferences in text comprehension*. Ph. D. Dissertation. University of Bristol.

[51] Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. *arXiv preprint arXiv:1604.03357* (2016).

[52] Steffen Koch, Markus John, Michael Wörner, Andreas Müller, and Thomas Ertl. 2014. VarifocalReader—in-depth visual analysis of large text documents. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 1723–1732.

[53] Jonathan K. Kummerfeld, Jessika Roesner, Tim Dawborn, James Haggerty, James R. Curran, and Stephen Clark. 2010. Faster Parsing by Supertagger Adaptation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Uppsala, Sweden, 345–355. https://aclanthology.org/P10-1036

[54] Philippe Laban, John Canny, and Marti Hearst. 2020. A framework for a text-centric user interface for navigating complex news stories. (2020).

[55] Jia Li, Yonghong Tian, and Tiejun Huang. 2014. Visual saliency with statistical priors. *International journal of computer vision* 107 (2014), 239–253.

[56] Zhaoping Li. 2002. A saliency map in primary visual cortex. *Trends in cognitive sciences* 6, 1 (2002), 9–16.

[57] Chin-Yew Lin. 2003. Improving Summarization Performance by Sentence Compression — A Pilot Study. In *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages*. Association for Computational Linguistics, Sapporo, Japan, 1–8. https://doi.org/10.3115/1118935.1118936

[58] H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development* (April 1958), 159–165.

[59] Ling Luo, Xiang Ao, Yan Song, Feiyang Pan, Min Yang, and Qing He. 2019. Reading Like HER: Human Reading Inspired Extractive Summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3033–3043. https://doi.org/10.18653/v1/D19-1300

[60] Anne Mangen and Adriaan Van der Weel. 2016. The evolution of reading in the age of digitisation: an integrative framework for reading research. *Literacy* 50, 3 (2016), 116–124.

[61] Manhattan Prep. 2011. *Reading comprehension & essays*. Manhattan Prep, New York, NY, USA.

[62] Inderjeet Mani. 2001. *Automatic summarization*. Vol. 3. John Benjamins Publishing.

[63] Elyza Martiarini. 2013. THE EFFECTS OF SPEED READING METHOD UPON STUDENTS'READING COMPREHENSION. *Deiksis* 5, 02 (2013), 89–105.

[64] André FT Martins and Noah A Smith. 2009. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*. 1–9.

[65] Michael E Masson. 1982. Cognitive processes in skimming stories. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 8, 5 (1982), 400.

[66] Michael EJ Masson. 1983. Conceptual processing of text during skimming and rapid sequential reading. *Memory & cognition* 11, 3 (1983), 262–274.

[67] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661* (2020).

[68] Afonso Mendes, Shashi Narayan, Sebastiao Miranda, Zita Marinho, André FT Martins, and Shay B Cohen. 2019. Jointly extracting and compressing documents with summary state representations. *arXiv preprint arXiv:1904.02020* (2019).

[69] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.

[70] Rutu Mulkar-Mehta, Jerry R Hobbs, and Eduard Hovy. 2011. Granularity in natural language discourse. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*.

[71] Tamara Munzner. 2014. *Visualization analysis and design*. CRC press.

[72] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1747–1759. https://doi.org/10.18653/v1/N18-1158

[73] D Parra, H Valdivieso, A Carvallo, G Rada, K Verbert, and T Schreck. 2019. Analyzing the design space for visualizing neural attention in text classification. In *Proc. ieee vis workshop on vis x ai: 2nd workshop on visualization for ai explainability (visxai)*.

[74] K Pernice, K Whitenton, J Nielsen, et al. 2014. How People Read Online: The Eyetracking Evidence. *Fremont, USA: Nielsen Norman Group* (2014).

[75] Keith Rayner, Alexander Pollatsek, Jane Ashby, and Charles Clifton Jr. 2012. *Psychology of reading*. Psychology Press.

[76] Keith Rayner, Elizabeth R Schotter, Michael EJ Masson, Mary C Potter, and Rebecca Treiman. 2016. So much to read, so little time: How do we read, and can speed reading help? *Psychological Science in the Public Interest* 17, 1 (2016), 4–34.

[77] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[78] Nicholas Riccardi and Rutvik H Desai. 2023. The Two Word Test: A Semantic Benchmark for Large Language Models. *arXiv preprint arXiv:2306.04610* (2023).

[79] Mariana Shimabukuro and Christopher Collins. 2017. Abbreviating text labels on demand. (2017).

[80] Sujan Shrestha. 2007. Mobile web browsing: usability study. In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*. 187–194.

[81] Nikhil Singh, Guillermo Bernal, Daria Savchenko, and Elena L Glassman. 2022. Where to hide a stolen elephant: Leaps in creative writing with multimodal machine intelligence. *ACM Transactions on Computer-Human Interaction* (2022).

[82] Andreas Stoffel, Hendrik Strobelt, Oliver Deussen, and Daniel A Keim. 2012. Document thumbnails with variable text scaling. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1165–1173.

[83] Hendrik Strobelt, Daniela Oelke, Bum Chul Kwon, Tobias Schreck, and Hanspeter Pfister. 2015. Guidelines for effective usage of text highlighting techniques. *IEEE transactions on visualization and computer graphics* 22, 1 (2015), 489–498.

[84] Nicole Sultanum, Devin Singh, Michael Brudno, and Fanny Chevalier. 2018. Doccurate: A curation-based approach for clinical text visualization. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 142–151.

[85] Jonathan Sutton, Tobias Langlotz, Alexander Plopski, Stefanie Zollmann, Yuta Itoh, and Holger Regenbrecht. 2022. Look over there! investigating saliency modulation for visual guidance with augmented reality glasses. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–15.

[86] Zhengwei Tao, Zhi Jin, Xiaoying Bai, Haiyan Zhao, Yanlin Feng, Jia Li, and Wenpeng Hu. 2023. EvEval: A Comprehensive Evaluation of Event Semantics for Large Language Models. *arXiv preprint arXiv:2305.15268* (2023).

[87] Craig S Tashman and W Keith Edwards. 2011. Active reading and its discontents: the situations, problems and ideas of readers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2927–2936.

[88] Eduardo E Veas, Erick Mendez, Steven K Feiner, and Dieter Schmalstieg. 2011. Directing attention and influencing memory with visual saliency modulation. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1471–1480.

[89] Shaun Wallace, Zoya Bylinskii, Jonathan Dobres, Bernard Kerr, Sam Berlow, Rick Treitman, Nirmal Kumawat, Kathleen Arpin, Dave B Miller, Jeff Huang, et al. 2022. Towards individuated reading experiences: Different fonts increase reading speed for different individuals. *ACM Transactions on Computer-Human Interaction (TOCHI)* 29, 4 (2022), 1–56.

[90] Shaun Wallace, Rick Treitman, Jeff Huang, Ben D Sawyer, and Zoya Bylinskii. 2020. Accelerating adult readers with typeface: a study of individual preferences and effectiveness. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–9.

[91] Hong Wang, Xin Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Self-Supervised Learning for Contextualized Extractive Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2221–2227. https://doi.org/10.18653/v1/P19-1214

[92] Alan J Wecker, Joel Lanir, Osnat Mokryn, Einat Minkov, and Tsvi Kuflik. 2014. Semantize: Visualizing the sentiment of individual document. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*. 385–386.

[93] Adhika Pramita Widyassari, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko, Abdul Syukur, Affandy Affandy, et al. 2022. Review of automatic text summarization techniques & methods. *Journal of King Saud University-Computer and Information Sciences* 34, 4 (2022), 1029–1046.

[94] Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark. *arXiv preprint arXiv:2303.13648* (2023).

[95] Hongyan Xu, Hongtao Liu, Zhepeng Lv, Qing Yang, and Wenjun Wang. 2023. Pre-trained Personalized Review Summarization with Effective Salience Estimation. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, Toronto, Canada, 10743–10754. https://doi.org/10.18653/v1/2023.findings-acl.684

[96] Han Xu, Eric Martin, and Ashesh Mahidadia. 2015. Extractive Summarisation Based on Keyword Profile and Language Model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, 123–132. https://doi.org/10.3115/v1/N15-1013

[97] Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. *arXiv preprint arXiv:1902.00863* (2019).

[98] Rui Yan, Jian-Yun Nie, and Xiaoming Li. 2011. Summarize What You Are Interested In: An Optimization Framework for Interactive Personalized Summarization. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Edinburgh, Scotland, UK., 1342–1351. https://aclanthology.org/D11-1124

[99] Qian Yang, Justin Cranshaw, Saleema Amershi, Shamsi T Iqbal, and Jaime Teevan. 2019. Sketching nlp: A case study of exploring the right things to design with language intelligence. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[100] Qian Yang, Gerard de Melo, Yong Cheng, and Sen Wang. 2017. HiText: Text reading with dynamic salience marking. In *Proceedings of the 26th International Conference on World Wide Web Companion*. 311–319.

[101] Qian Yang, Yuexing Hao, Kexin Quan, Stephen Yang, Yiran Zhao, Volodymyr Kuleshov, and Fei Wang. 2023. Harnessing biomedical literature to calibrate clinicians' trust in AI decision support systems. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.

[102] Ji Soo Yi. 2014. Qndreview: Read 100 chi papers in 7 hours. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 805–814.

[103] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. Extractive summarization via chatgpt for faithful summary generation. *arXiv preprint arXiv:2304.04193* (2023).

[104] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. SummIt: Iterative Text Summarization via ChatGPT. *arXiv preprint arXiv:2305.14835* (2023).

[105] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848* (2023).

[106] Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931* (2017).

[107] Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural Latent Extractive Document Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 779–784. https://doi.org/10.18653/v1/D18-1088

[108] Yang Zhao, Zhiyuan Luo, and Akiko Aizawa. 2018. A language model based evaluator for sentence compression. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 170–175.

[109] Zheng Zhao, Shay B Cohen, and Bonnie Webber. 2020. Reducing quantity hallucinations in abstractive summarization. *arXiv preprint arXiv:2009.13312* (2020).

[110] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive Summarization as Text Matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 6197–6208. https://doi.org/10.18653/v1/2020.acl-main.552

[111] Ming Zhong, Yang Liu, Suyu Ge, Yuning Mao, Yizhu Jiao, Xingxing Zhang, Yichong Xu, Chenguang Zhu, Michael Zeng, and Jiawei Han. 2022. Unsupervised Multi-Granularity Summarization. *arXiv preprint arXiv:2201.12502* (2022).

[112] Ming Zhong, Yang Liu, Suyu Ge, Yuning Mao, Yizhu Jiao, Xingxing Zhang, Yichong Xu, Chenguang Zhu, Michael Zeng, and Jiawei Han. 2022. Unsupervised Multi-Granularity Summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 4980–4995. https://doi.org/10.18653/v1/2022.findings-emnlp.366

[113] Richard P Zipoli Jr. 2017. Unraveling difficult sentences: Strategies to support reading comprehension. *Intervention in School and Clinic* 52, 4 (2017), 218–227.

A POST-STUDY SURVEY

Question Statements
Please rank the 3 interfaces from most to least helpful for answering the reading questions.
What did you like most about the interface you found the most helpful? [open-ended]
Are there any features missing that you'd like to see in the interface you found the most helpful? [open-ended]
I would like to use Reader-Blue to read online text of interest to me in the future.
I would like to use Reader-Green to read online text of interest to me in the future.
I would like to use Reader-Red to read online text of interest to me in the future.

Table 4: Questions in the post-study survey. The last three ask participants to rate their agreement with them on a 7-point Likert scale from "Strongly Disagree" (a score of 1) to "Strongly Agree" (a score of 7).

B SCATTER PLOTS OF USER STUDY RESULTS

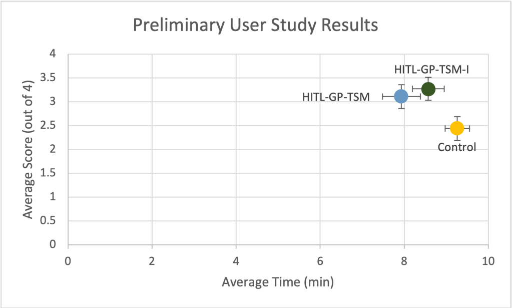


Figure 8: A scatter plot of the preliminary user study results showing average time and performance of the 3 study conditions, with error bars representing standard error.

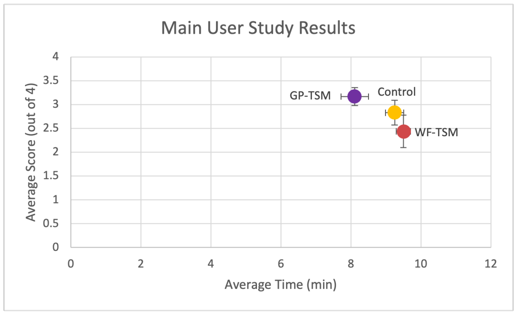
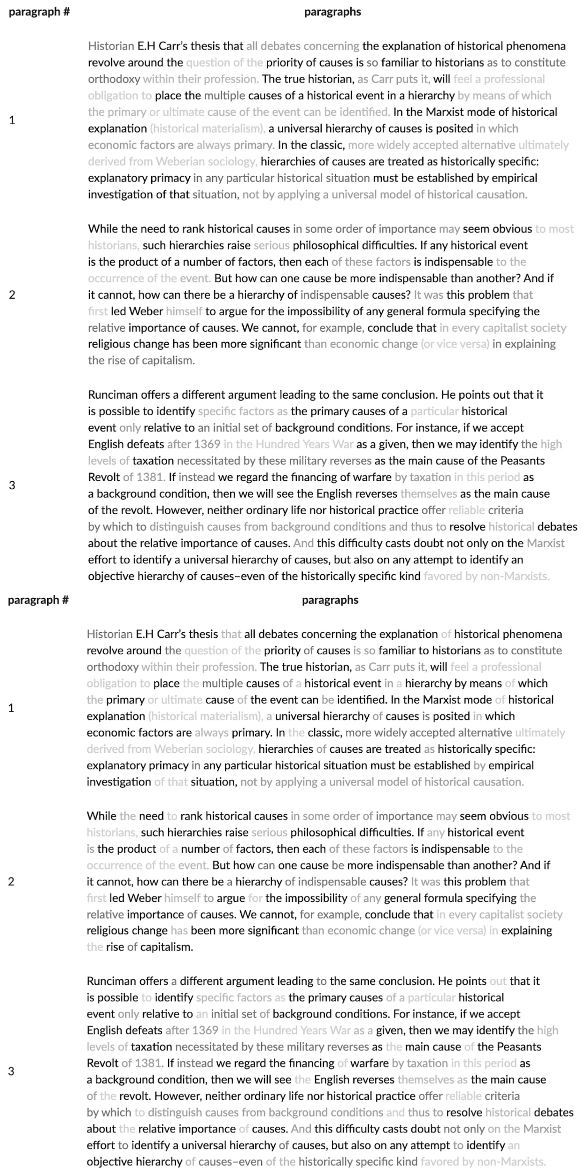


Figure 9: A scatter plot of the main user study results showing average time and performance of the 3 study conditions, with error bars representing standard error.

C GP-TSM VS. NGP-TSM IN COMPARISON

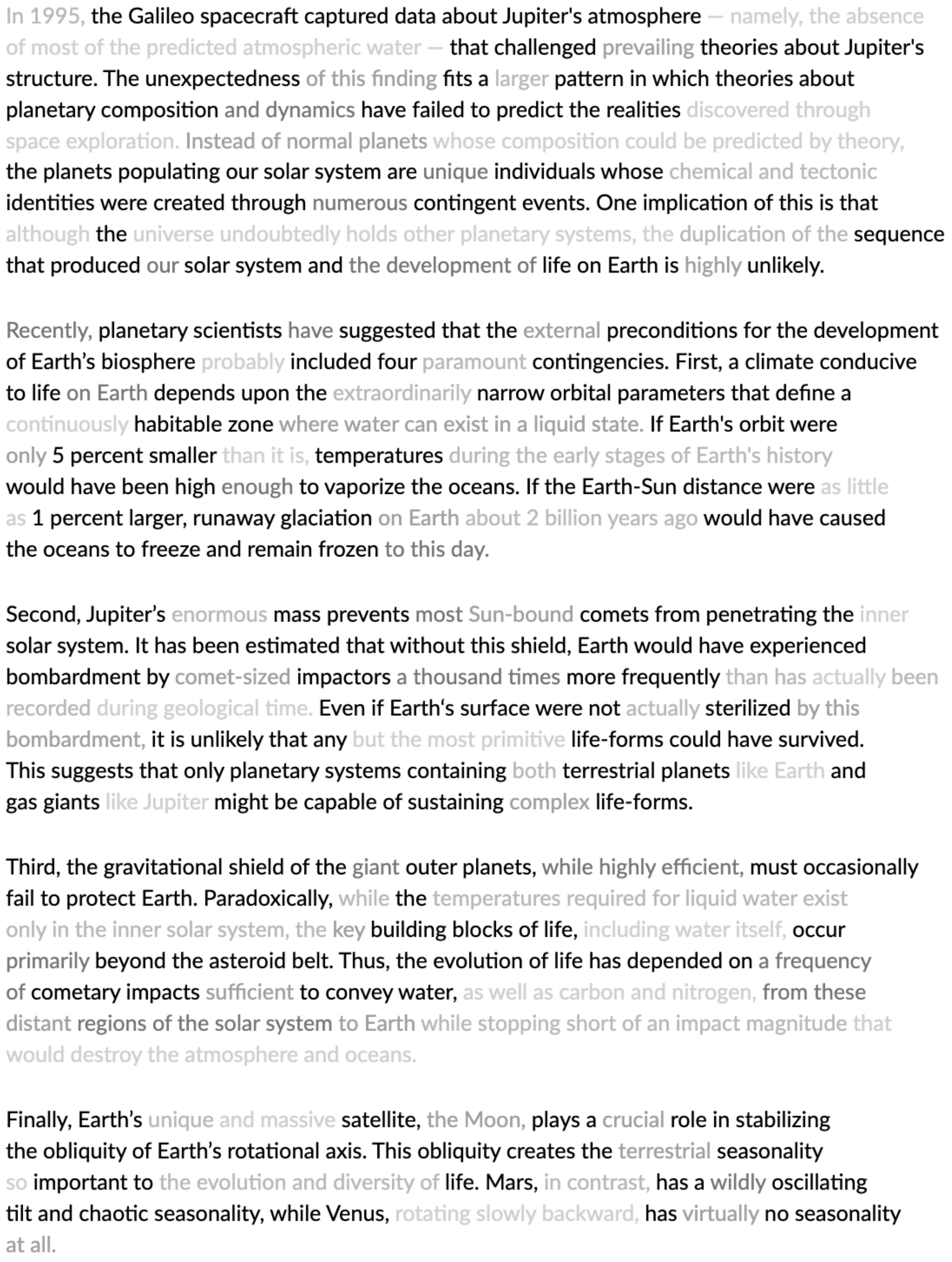


paragraph #	paragraphs
1	Historian E.H Carr's thesis that all debates concerning the explanation of historical phenomena revolve around the question of the priority of causes is so familiar to historians as to constitute orthodoxy within their profession. The true historian, as Carr puts it, will feel a professional obligation to place the multiple causes of a historical event in a hierarchy by means of which the primary or ultimate cause of the event can be identified. In the Marxist mode of historical explanation (historical materialism), a universal hierarchy of causes is posited in which economic factors are always primary. In the classic, more widely accepted alternative ultimately derived from Weberian sociology, hierarchies of causes are treated as historically specific: explanatory primacy in any particular historical situation must be established by empirical investigation of that situation, not by applying a universal model of historical causation.
2	While the need to rank historical causes in some order of importance may seem obvious to most historians, such hierarchies raise serious philosophical difficulties. If any historical event is the product of a number of factors, then each of these factors is indispensable to the occurrence of the event. But how can one cause be more indispensable than another? And if it cannot, how can there be a hierarchy of indispensable causes? It was this problem that first led Weber himself to argue for the impossibility of any general formula specifying the relative importance of causes. We cannot, for example, conclude that in every capitalist society religious change has been more significant than economic change (or vice versa) in explaining the rise of capitalism.
3	Runciman offers a different argument leading to the same conclusion. He points out that it is possible to identify specific factors as the primary causes of a particular historical event only relative to an initial set of background conditions. For instance, if we accept English defeats after 1369 in the Hundred Years War as a given, then we may identify the high levels of taxation necessitated by these military reverses as the main cause of the Peasants Revolt of 1381. If instead we regard the financing of warfare by taxation in this period as a background condition, then we will see the English reverses themselves as the main cause of the revolt. However, neither ordinary life nor historical practice offer reliable criteria by which to distinguish causes from background conditions and thus to resolve historical debates about the relative importance of causes. And this difficulty casts doubt not only on the Marxist effort to identify a universal hierarchy of causes, but also on any attempt to identify an objective hierarchy of causes–even of the historically specific kind favored by non-Marxists.

paragraph #	paragraphs
1	Historian E.H Carr's thesis that all debates concerning the explanation of historical phenomena revolve around the question of the priority of causes is so familiar to historians as to constitute orthodoxy within their profession. The true historian, as Carr puts it, will feel a professional obligation to place the multiple causes of a historical event in a hierarchy by means of which the primary or ultimate cause of the event can be identified. In the Marxist mode of historical explanation (historical materialism), a universal hierarchy of causes is posited in which economic factors are always primary. In the classic, more widely accepted alternative ultimately derived from Weberian sociology, hierarchies of causes are treated as historically specific: explanatory primacy in any particular historical situation must be established by empirical investigation of that situation, not by applying a universal model of historical causation.
2	While the need to rank historical causes in some order of importance may seem obvious to most historians, such hierarchies raise serious philosophical difficulties. If any historical event is the product of a number of factors, then each of these factors is indispensable to the occurrence of the event. But how can one cause be more indispensable than another? And if it cannot, how can there be a hierarchy of indispensable causes? It was this problem that first led Weber himself to argue for the impossibility of any general formula specifying the relative importance of causes. We cannot, for example, conclude that in every capitalist society religious change has been more significant than economic change (or vice versa) in explaining the rise of capitalism.
3	Runciman offers a different argument leading to the same conclusion. He points out that it is possible to identify specific factors as the primary causes of a particular historical event only relative to an initial set of background conditions. For instance, if we accept English defeats after 1369 in the Hundred Years War as a given, then we may identify the high levels of taxation necessitated by these military reverses as the main cause of the Peasants Revolt of 1381. If instead we regard the financing of warfare by taxation in this period as a background condition, then we will see the English reverses themselves as the main cause of the revolt. However, neither ordinary life nor historical practice offer reliable criteria by which to distinguish causes from background conditions and thus to resolve historical debates about the relative importance of causes. And this difficulty casts doubt not only on the Marxist effort to identify a universal hierarchy of causes, but also on any attempt to identify an objective hierarchy of causes–even of the historically specific kind favored by non-Marxists.

Figure 10: Side-by-side view of GP-TSM (above) and NGP-TSM (below) shown to a participant in one of the comparison interviews

D MORE EXAMPLES OF GP-TSM



In 1995, the Galileo spacecraft captured data about Jupiter's atmosphere — namely, the absence of most of the predicted atmospheric water — that challenged prevailing theories about Jupiter's structure. The unexpectedness of this finding fits a larger pattern in which theories about planetary composition and dynamics have failed to predict the realities discovered through space exploration. Instead of normal planets whose composition could be predicted by theory, the planets populating our solar system are unique individuals whose chemical and tectonic identities were created through numerous contingent events. One implication of this is that although the universe undoubtedly holds other planetary systems, the duplication of the sequence that produced our solar system and the development of life on Earth is highly unlikely.

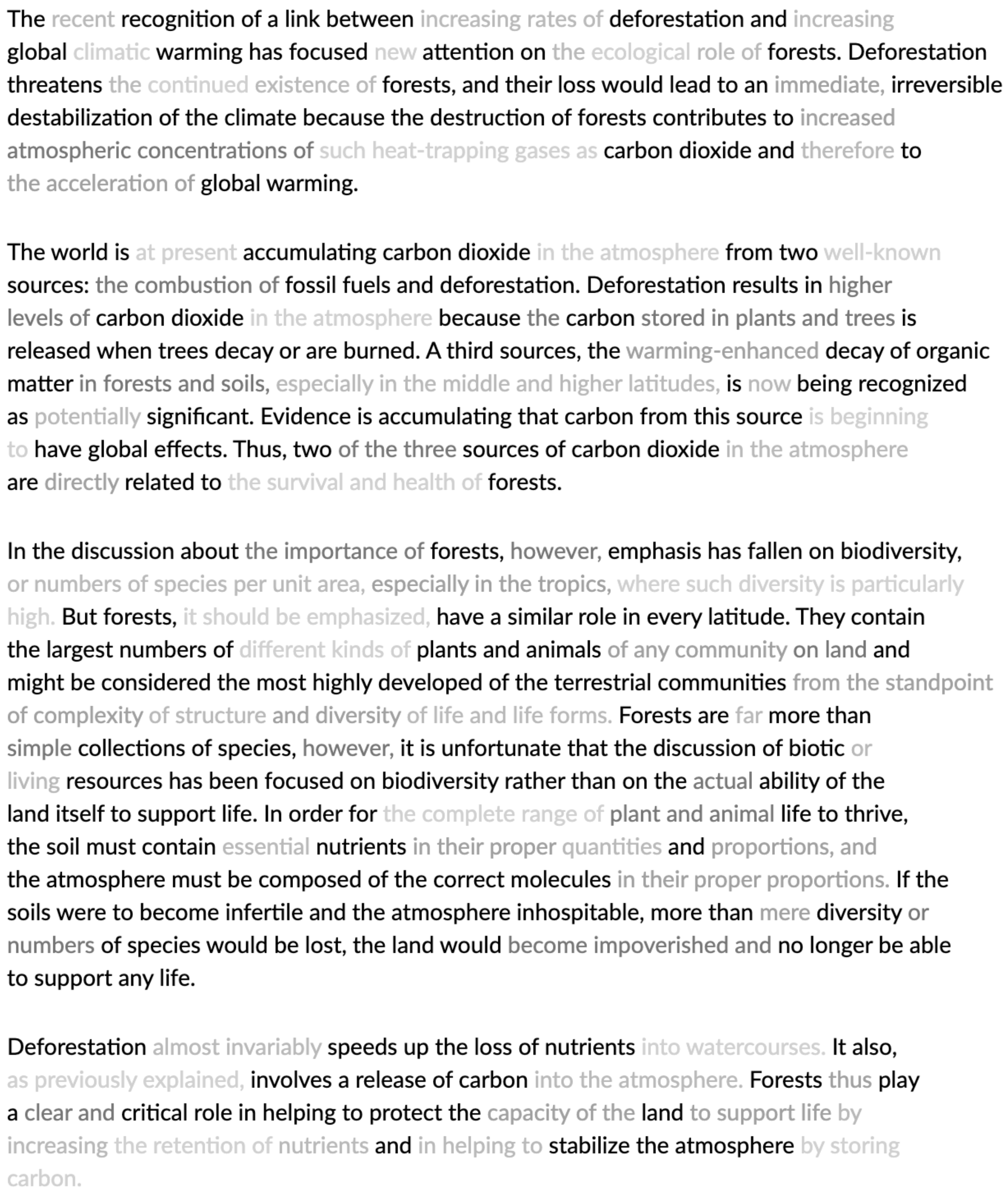
Recently, planetary scientists have suggested that the external preconditions for the development of Earth’s biosphere probably included four paramount contingencies. First, a climate conducive to life on Earth depends upon the extraordinarily narrow orbital parameters that define a continuously habitable zone where water can exist in a liquid state. If Earth's orbit were only 5 percent smaller than it is, temperatures during the early stages of Earth's history would have been high enough to vaporize the oceans. If the Earth-Sun distance were as little as 1 percent larger, runaway glaciation on Earth about 2 billion years ago would have caused the oceans to freeze and remain frozen to this day.

Second, Jupiter’s enormous mass prevents most Sun-bound comets from penetrating the inner solar system. It has been estimated that without this shield, Earth would have experienced bombardment by comet-sized impactors a thousand times more frequently than has actually been recorded during geological time. Even if Earth‘s surface were not actually sterilized by this bombardment, it is unlikely that any but the most primitive life-forms could have survived. This suggests that only planetary systems containing both terrestrial planets like Earth and gas giants like Jupiter might be capable of sustaining complex life-forms.

Third, the gravitational shield of the giant outer planets, while highly efficient, must occasionally fail to protect Earth. Paradoxically, while the temperatures required for liquid water exist only in the inner solar system, the key building blocks of life, including water itself, occur primarily beyond the asteroid belt. Thus, the evolution of life has depended on a frequency of cometary impacts sufficient to convey water, as well as carbon and nitrogen, from these distant regions of the solar system to Earth while stopping short of an impact magnitude that would destroy the atmosphere and oceans.

Finally, Earth’s unique and massive satellite, the Moon, plays a crucial role in stabilizing the obliquity of Earth’s rotational axis. This obliquity creates the terrestrial seasonality so important to the evolution and diversity of life. Mars, in contrast, has a wildly oscillating tilt and chaotic seasonality, while Venus, rotating slowly backward, has virtually no seasonality at all.

Figure 11: GRE Passage 1 rendered using GP-TSM, as an additional example of how GP-TSM works.



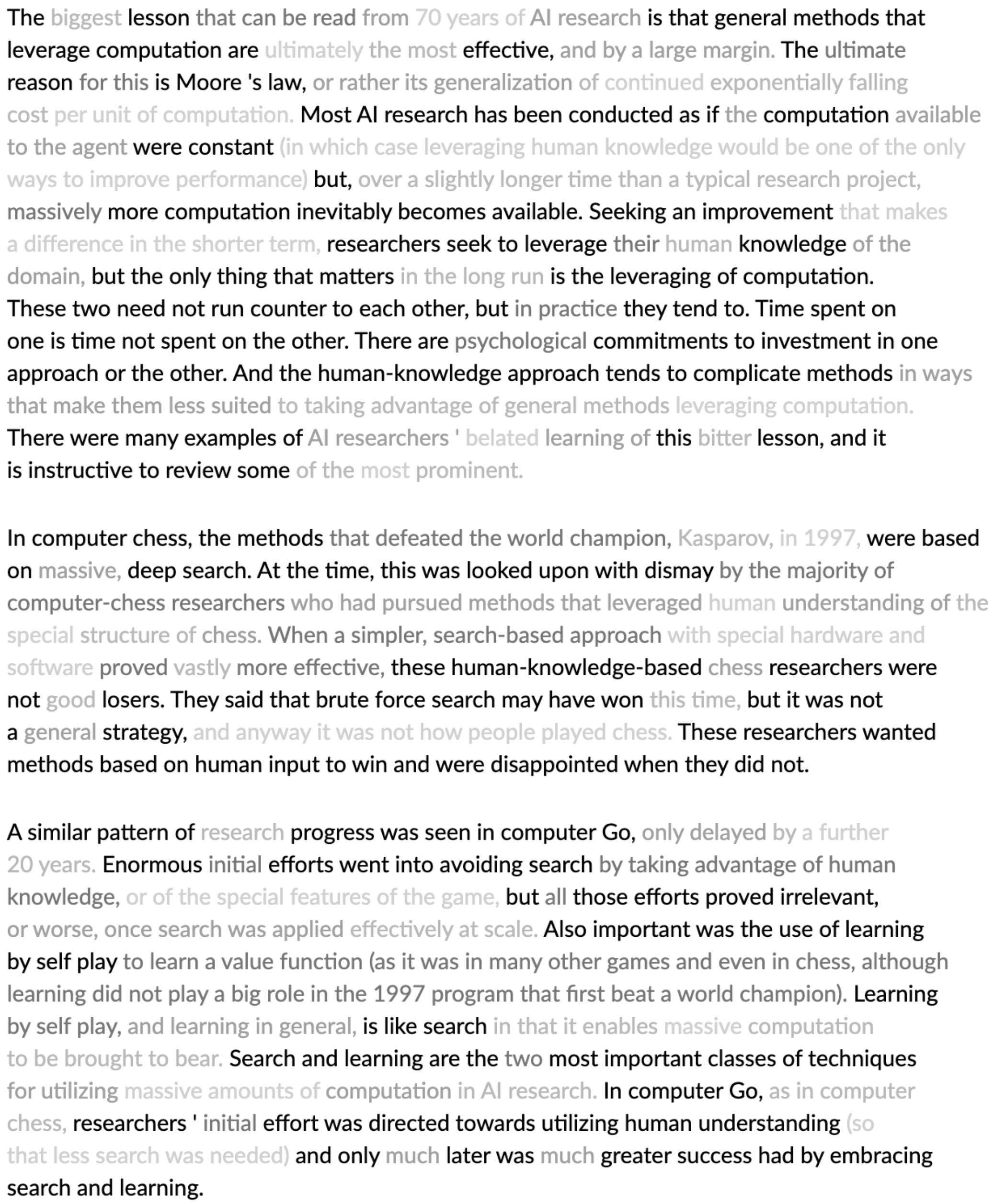
The recent recognition of a link between increasing rates of deforestation and increasing global climatic warming has focused new attention on the ecological role of forests. Deforestation threatens the continued existence of forests, and their loss would lead to an immediate, irreversible destabilization of the climate because the destruction of forests contributes to increased atmospheric concentrations of such heat-trapping gases as carbon dioxide and therefore to the acceleration of global warming.

The world is at present accumulating carbon dioxide in the atmosphere from two well-known sources: the combustion of fossil fuels and deforestation. Deforestation results in higher levels of carbon dioxide in the atmosphere because the carbon stored in plants and trees is released when trees decay or are burned. A third sources, the warming-enhanced decay of organic matter in forests and soils, especially in the middle and higher latitudes, is now being recognized as potentially significant. Evidence is accumulating that carbon from this source is beginning to have global effects. Thus, two of the three sources of carbon dioxide in the atmosphere are directly related to the survival and health of forests.

In the discussion about the importance of forests, however, emphasis has fallen on biodiversity, or numbers of species per unit area, especially in the tropics, where such diversity is particularly high. But forests, it should be emphasized, have a similar role in every latitude. They contain the largest numbers of different kinds of plants and animals of any community on land and might be considered the most highly developed of the terrestrial communities from the standpoint of complexity of structure and diversity of life and life forms. Forests are far more than simple collections of species, however, it is unfortunate that the discussion of biotic or living resources has been focused on biodiversity rather than on the actual ability of the land itself to support life. In order for the complete range of plant and animal life to thrive, the soil must contain essential nutrients in their proper quantities and proportions, and the atmosphere must be composed of the correct molecules in their proper proportions. If the soils were to become infertile and the atmosphere inhospitable, more than mere diversity or numbers of species would be lost, the land would become impoverished and no longer be able to support any life.

Deforestation almost invariably speeds up the loss of nutrients into watercourses. It also, as previously explained, involves a release of carbon into the atmosphere. Forests thus play a clear and critical role in helping to protect the capacity of the land to support life by increasing the retention of nutrients and in helping to stabilize the atmosphere by storing carbon.

Figure 12: GRE Passage 2 rendered using GP-TSM, as an additional example of how GP-TSM works.



The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. The ultimate reason for this is Moore 's law, or rather its generalization of continued exponentially falling cost per unit of computation. Most AI research has been conducted as if the computation available to the agent were constant (in which case leveraging human knowledge would be one of the only ways to improve performance) but, over a slightly longer time than a typical research project, massively more computation inevitably becomes available. Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation. These two need not run counter to each other, but in practice they tend to. Time spent on one is time not spent on the other. There are psychological commitments to investment in one approach or the other. And the human-knowledge approach tends to complicate methods in ways that make them less suited to taking advantage of general methods leveraging computation. There were many examples of AI researchers ' belated learning of this bitter lesson, and it is instructive to review some of the most prominent.

In computer chess, the methods that defeated the world champion, Kasparov, in 1997, were based on massive, deep search. At the time, this was looked upon with dismay by the majority of computer-chess researchers who had pursued methods that leveraged human understanding of the special structure of chess. When a simpler, search-based approach with special hardware and software proved vastly more effective, these human-knowledge-based chess researchers were not good losers. They said that brute force search may have won this time, but it was not a general strategy, and anyway it was not how people played chess. These researchers wanted methods based on human input to win and were disappointed when they did not.

A similar pattern of research progress was seen in computer Go, only delayed by a further 20 years. Enormous initial efforts went into avoiding search by taking advantage of human knowledge, or of the special features of the game, but all those efforts proved irrelevant, or worse, once search was applied effectively at scale. Also important was the use of learning by self play to learn a value function (as it was in many other games and even in chess, although learning did not play a big role in the 1997 program that first beat a world champion). Learning by self play, and learning in general, is like search in that it enables massive computation to be brought to bear. Search and learning are the two most important classes of techniques for utilizing massive amounts of computation in AI research. In computer Go, as in computer chess, researchers ' initial effort was directed towards utilizing human understanding (so that less search was needed) and only much later was much greater success had by embracing search and learning.

Figure 13: GRE Passage 3 rendered using GP-TSM, as an additional example of how GP-TSM works.