TextVision: A more efficient way to work with research

Melis Aslan[1,2], Maximilian Bosse[1,2], Daniel Ehlers[1,2], Marlon Hinz[1,2], Philipp Olschewski[1,2], Jannik Podszun[1,2], Elias Scharlach[1,2], Leon Selzer[1,2], Yukun Wu[1,2], Aliki Anagnostopoulou[1,2] and Daniel Sonntag[1,2]

[1]*German Research Center for Artificial Intelligence (DFKI), 26129 Oldenburg, Germany*
[2]*Carl von Ossietzky University of Oldenburg, 26129 Oldenburg, Germany*

Abstract

Large language models remain constrained by the limitations of current user interfaces and interaction paradigms, which hinder their ability to process complex, multimodal information beyond simple text input and output. Our proposed interface, TextVision, aims to address this limitation by enhancing how researchers interact with AI, providing a wide range of functionalities for analyzing, editing, creating new documents, and facilitating collaboration. TextVision advances state-of-the-art human-AI interaction through improved usability and novel interaction techniques, enhancing scientific research and development workflows. As a result, the user can access integrated tools, including a text editor, a PDF viewer, and an AI assistant in a chatbot format. The AI assistant can provide answers based on user input and is context-aware. This output can be enhanced using the built-in prompt designing tool to create efficient, AI-optimized prompts. Users can also select between the latest proprietary LLMs and fine-tuned open-source models tailored for specific tasks.

Keywords

Large Language Models, Human-AI Interaction, Interactive User Interfaces, Prompt Optimization, Prompt Engineering, Scientific Research

1. Introduction

Despite the growing adoption of large language models (LLMs) in diverse applications, their full potential remains unrealized, primarily due to inherent limitations in user interfaces and interaction paradigms. Identifying suitable input prompts for these models is a daunting challenge, often requiring extensive experimentation and domain-specific expertise. Many applications are restricted to basic text-based inputs and outputs, failing to adequately address real-world information processing tasks' complex, multimodal requirements, mainly when dealing with intricate document formats such as PDFs.

Joint Proceedings of the ACM IUI Workshops 2025, March 24-27, 2025, Cagliari, Italy
melis.aslan@uni-oldenburg.de (M. Aslan); maximilian.bosse@uni-oldenburg.de (M. Bosse); daniel.ehlers@uni-oldenburg.de (D. Ehlers); marlon.hinz@uni-oldenburg.de (M. Hinz); philipp.olschewski@uni-oldenburg.de (P. Olschewski); jannik.podszun@uni-oldenburg.de (J. Podszun); elias.scharlach@uni-oldenburg.de (E. Scharlach); leon.selzer@uni-oldenburg.de (L. Selzer); yukun.wu@uni-oldenburg.de (Y. Wu); aliki.anagnostopoulou@dfki.de (A. Anagnostopoulou); daniel.sonntag@dfki.de (D. Sonntag)
© 2025 Copyright © 2025 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

We propose TextVision, an interface that connects the capabilities of LLMs and intelligent user interfaces (IUIs) through a user-friendly platform [1]. TextVision provides users with a dedicated workspace for storing their state of the editor, chat, and prompts. Users can create and evaluate their prompts within this workspace before utilizing them within the application at a later point. The main goal of TextVision is to enhance human-AI interaction, focusing on aiding researchers in working with academic papers. By identifying essential fundamentals of prompt engineering and human-computer interaction (HCI), TextVision aims to develop a new approach to IUIs. This is achieved by leveraging LLMs within an interactive PDF viewer and editor, offering a transformative method to work with multiple papers and documents to create new ones. Furthermore, our system incorporates an AI-assisted knowledge management component, which leverages context-aware capabilities to provide users with relevant information and support throughout their tasks. This module is built upon a composite architecture that integrates proprietary and open-source variants of LLMs, including those specifically fine-tuned for scientific research applications [2].

2. Related work

For document editing and processing, prior LLM-based approaches have primarily focused on tasks such as streamlining workflows [3, 4] and facilitating research work [5, 6]. [7] propose using LLMs as evaluators, showcasing their potential for unbiased and scalable assessment in various contexts. [8] emphasize leveraging user interfaces as writing tools that facilitate reflection on the user's writing process, enhancing engagement and productivity. Similarly, feedback mechanisms are explored for principle generation, enabling iterative refinement of ideas and practices [9]. Additional insights include suggestions for improving peer reviews through structured support systems powered by AI [10] and the development of multimodal systems for co-creating scientific alternative text, enhancing accessibility and inclusivity in scholarly communication [11]. Finally, [12] explore various ways AI can support scientific reading, such as discovery, efficiency, comprehension, synthesis, and accessibility.

However, current AI tools rely on generic prompts, often neglecting the specific requirements of the use case. TextVision challenges the current state of the art by enabling users to design, refine, and save custom prompt shortcuts, utilizing the potential of AI systems and facilitating the interaction between the user and LLMs, which is crucial, especially with non-AI experts [13]. Our work emphasizes the integration of prompt editing to facilitate the creation of effective prompts. The frameworks and tools most closely related to our approach are [14] and [15]: The former enables users to generate improved prompts to enhance LLM output quality without requiring programming expertise, while the latter focuses on advancing human-AI interaction by applying direct manipulation principles. TextVision synthesizes these principles, offering a prompting function and additional features specifically designed to support and streamline scientific work, thereby advancing human-AI interaction in this domain.

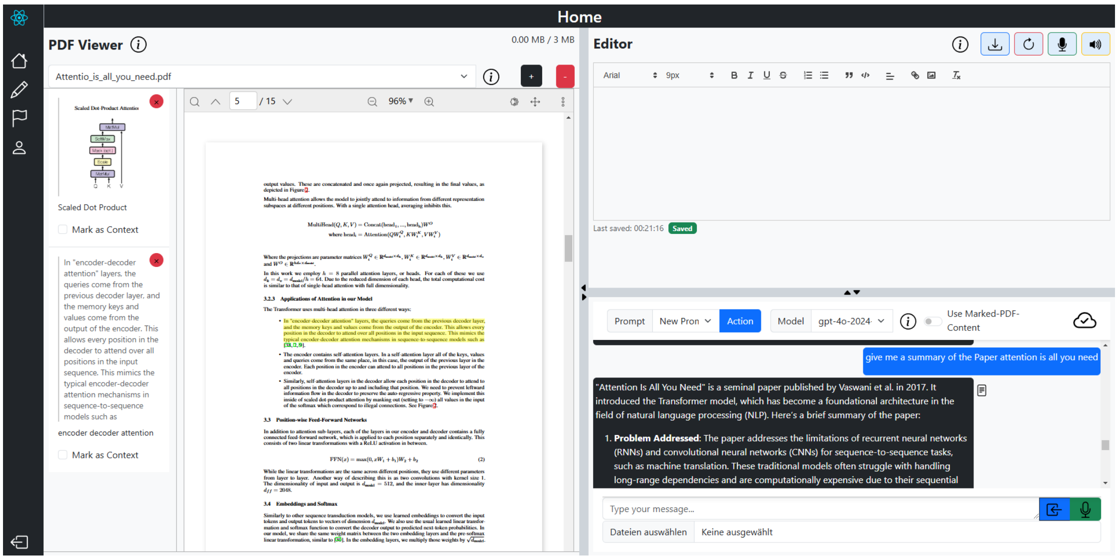

Figure 1: Overview of the TextVision homepage. The interface is divided vertically: the left section contains the PDF viewer for document interaction, including text highlighting and note creation, while the right section is split horizontally into the editor for document editing and the chat interface for interacting with the AI assistant.

3. TextVision: Structure and Workflow

3.1. The interface

The TextVision platform is organized into three discrete pages. The primary page, designated as the project center, enables users to create and participate in workspaces, which are dedicated containers for specific projects. Each workspace comprises two subsidiary pages: the homepage and the `PromptDesigner` (Section 3.3.2). This modular design is informed by evidence-based design decisions, which have been rigorously tested to optimize user experience [16].

The homepage's content is divided vertically in the middle. The PDF-viewer, located on the left, transfers, displays, and interacts with PDF files (shown in Figure 1). TextVision allows the upload of multiple PDFs. However, only one can be displayed simultaneously. A drop-down menu with the names of all uploaded files is shown at the top of the PDF-viewer to facilitate easy interchange of the projected PDF. The user can interact with the PDF by marking text or dragging the mouse to mark an area. Both actions result in a pop-up that contains a text area and three action buttons. The first option is to ask the LLM directly. Here, a combination of the text input of the text area and the marked content is used as a prompt. The second option is the creation of a note: in this case, the content in the PDF that has been marked remains marked, and a note is generated that displays the content and includes the text input as a description. The generated notes are displayed in a sidebar on the left side of the PDF. Each note can be designated as a context, signifying that it will be utilized as supplementary material for a prompt within the chat. Moreover, to enhance navigation, upon clicking on a note in the sidebar, the note's contents in the PDF will be scrolled into focus. This function is functional across multiple

PDFs, and if the content of the clicked note is not present in the current PDF, the PDF will switch to the file containing the note and scroll to the content. The last option is to cancel the marking action, which results in everything written in the text area being dismissed, closing the pop-up, and not highlighting the marked content.

As seen in Figure 1, the homepage's right side is divided horizontally in the middle. The editor, located in the top half of the page, incorporates "what you see is what you get" (WYSIWYG) style [17]. WYSIWYG refers to an editor or software interface in which the content displayed during the editing process resembles its final appearance in the published or printed form. In the context of editors, a WYSIWYG interface allows users to create and modify documents visually without needing to understand or write underlying code or markup languages. This approach enhances usability by providing immediate visual feedback, making the editing process more intuitive and efficient [18].

The chat is situated at the bottom of the right-hand side, serving as an interface for the LLM. It is divided into three distinct sections: the *header*, *chat history*, and *chat input*. Its header displays the status of the connection, allows for the selection of the marked notes as context, determines the response generation model, and selects the saved prompts from the PromptDesigner. The chat history displays the prompt request from the user and the responses in chronological order. Each LLM response has a button that integrates its content into the editor. In the chat input, the user can make prompts and add images. Consider the scenario where users require assistance crafting well-defined input queries (prompts) or seek inspiration to initiate their work. To facilitate effective task management, TextVision incorporates an adaptive prompt generation module that analyzes both the user-inputted text and the contents of the currently loaded PDF. This module generates contextually relevant prompt suggestions, which are then presented above the input field for the user's consideration.

3.2. Features and use cases

The TextVision platform is designed to address various real-world scenarios where users require streamlined and efficient tools for handling complex document workflows. One prominent use case centers on research, a context in which TextVision can significantly enhance the productivity and effectiveness of users engaged in e. g. scientific or technical investigations. The following scenarios, based on challenges often faced in research, illustrate how and why the system is used in the research domain.

3.2.1. Reviewing and Summarizing Research Papers

A common challenge for researchers is reviewing large numbers of academic papers, extracting critical insights while managing time constraints. TextVision provides a streamlined solution to this problem by allowing users to upload and interact with documents directly within the platform.

- **Scenario:** A researcher uploads a set of PDF articles related to a specific topic. Using the integrated PDF viewer, they can browse the documents while leveraging the AI assistant to highlight key sections such as abstracts, methodology descriptions, or results.

- **Interaction:** The user selects a specific passage, such as a complex paragraph in the discussion section, and requests the AI to generate a simplified summary or explanation. Instead of manually extracting and inputting the passage into the LLM, the assistant can directly reference the marked paragraph, leveraging an integrated context-aware retrieval mechanism akin to DirectGPT[15].
- **Outcome:** This can reduce cognitive load, accelerate comprehension, and allow the researcher to focus on synthesizing insights rather than trying to understand dense text.

3.2.2. Designing and refining research prompts

In research, effective querying is essential for obtaining accurate and relevant information. TextVision's prompting concept (Section 3.3) builds upon [14] by assisting researchers in finding, creating, and optimizing prompts for specific research tasks.

- **Scenario:** A researcher is conducting a systematic review on the impact of climate change on agricultural productivity. They need to extract specific data points from multiple research papers. As they interact with the platform, they receive prompt recommendations based on their current input and the content of the uploaded documents.
- **Interaction:** As the user begins typing a query in the chat, the Prompt Recommendations suggest tailored prompts that are contextually relevant to their current research focus. If the user accepts a suggestion, they can use it immediately. Alternatively, they can refine the prompt in the `PromptDesigner` window to ensure it aligns with their needs.
- **Outcome:** This process streamlines the research workflow by helping researchers create more accurate queries with less effort. The ability to reuse and optimize queries ensures that researchers can maintain consistency and efficiency across different phases of their work. These features give researchers more control over the information retrieval process, which can improve the speed and quality of their research.

3.2.3. Cross-referencing and contextualizing information

Researchers frequently encounter situations where they must cross-reference information from multiple sources or require context to understand specific findings better. TextVision facilitates this process by enabling the ability to switch between viewing different documents and integrating context-aware AI assistance.

- **Scenario:** While working on a literature review, a user identifies two papers with differing conclusions about a similar hypothesis. By uploading both documents, they can use TextVision to cross-reference specific sections and query the AI-Assistant for contextual explanations or summaries of key differences.
- **Interaction:** The user highlights relevant paragraphs in each paper and prompts the AI assistant to compare methodologies or results, providing a synthesized analysis.
- **Outcome:** This can help the researcher uncover insights and enhance their ability to draw informed conclusions for their work.

3.3. Prompting Concept of TextVision

The prompting concept of TextVision is designed to optimize how users interact with the platform's AI-powered assistant by facilitating more effective communication between the user and the system. The prompting concept is divided into two core components: the *prompt recommendations* and the dedicated `PromptDesigner` window.

3.3.1. Prompt recommendations

The first element of the TextVision prompting concept is the *prompt recommendation* system, which plays a crucial role in assisting users by offering real-time suggestions as they interact with the chat functionality. This system dynamically analyses the user's input in the chat field, providing tailored prompts after a brief delay once the user stops typing. These recommendations are generated based on several context factors: the immediate input the user has entered into the chat field, the history of the ongoing conversation, and any relevant content from the PDF document being viewed. This ensures the suggestions are highly contextual and aligned with the user's current needs. By providing these recommendations, TextVision reduces cognitive load, as users do not need to repeatedly think of specific prompt formulations or manually navigate between different documents or sections of the interface. In this case, the platform does most of the work, anticipating the following steps based on the available context. This should result in a smoother interaction, allowing users to focus more on their work and less on formulating precise queries. It should also reduce the time spent making decisions, as users can select from a list of pre-generated prompts closely aligned with their task or document.

3.3.2. `PromptDesigner`

The second relevant feature in TextVision's prompting concept is the `PromptDesigner` window, which serves as a dedicated space for users to create, optimize, and save their own prompts. While the prompt recommendation system addresses real-time suggestions, the `PromptDesigner` allows for an intentional and customized prompt creation. This window enables users to design prompts tailored to their needs or workflow. Whether a user needs highly specialized prompt templates for certain document types, specific interactions with the AI, or wants to create a set of frequently used prompts for efficiency, the `PromptDesigner` offers an intuitive interface to accomplish this. Users can save these personalized prompts for later use, ensuring they don't need to enter or recreate them repeatedly. Once created and saved, these optimized prompts are seamlessly integrated into the main TextVision interface. Users can easily access and use their saved prompts by selecting them from a combobox in the chat section. This further integrates the platform's functionality by combining the prompt recommendations' dynamic nature with the user-created prompts' flexibility.
Figure 2 illustrates the GUI of the `PromptDesigner`. To utilize the feature in TextVision, users begin by creating a new prompt entry. This involves providing a name, a brief description, and the actual prompt they wish to optimize. Once the basic details are set, the user selects the model to evaluate the prompt. In addition to selecting a model, the user must choose a prompting framework to assess the prompt. The framework uses a star rating system, with five stars representing the highest quality. This gives users clear feedback on how well their prompt

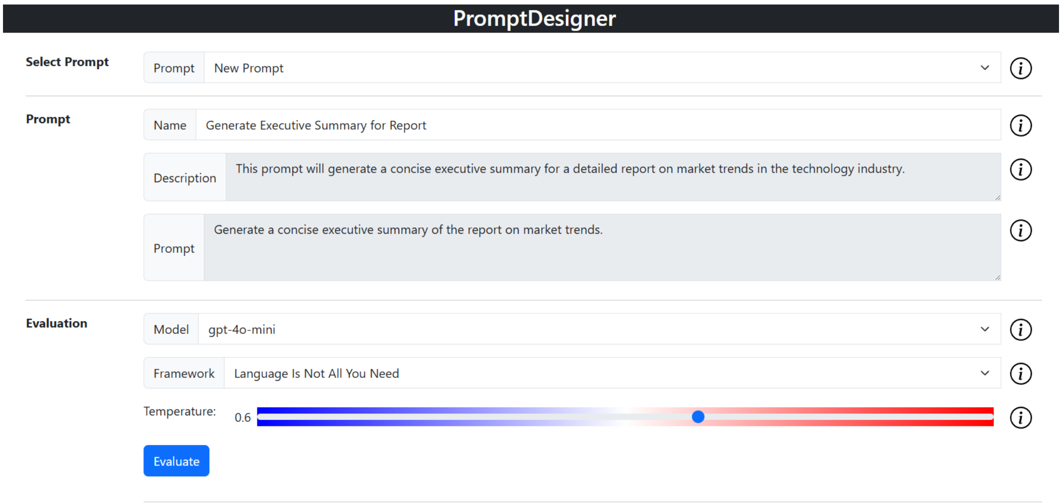

Figure 2: Graphical user interface (GUI) of the `PromptDesigner`. Users can create, edit, and evaluate prompts by specifying a name, description, and prompt text. Additionally, model selection, framework evaluation, and temperature adjustments allow for prompt customization and optimization.

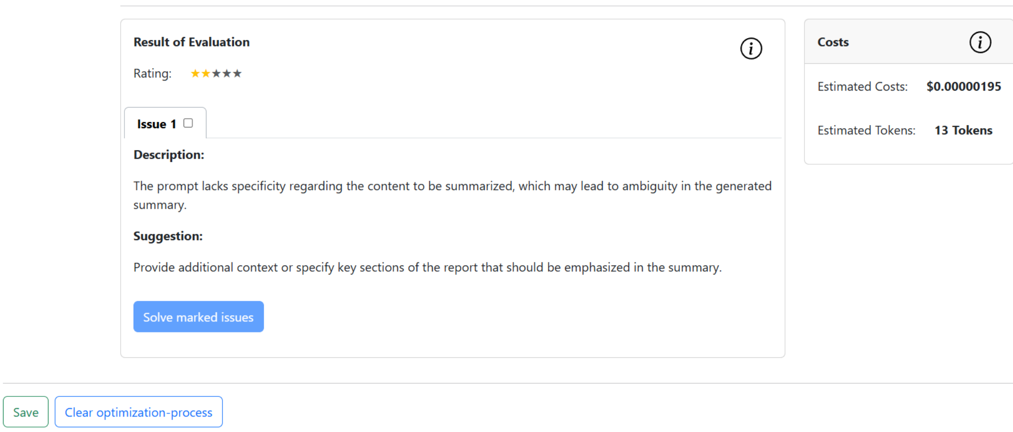

Figure 3: Evaluation results in the `PromptDesigner`. After evaluating a prompt, users receive a star rating and a detailed analysis of potential issues with corresponding suggestions for improvement. Selected issues can be resolved automatically to generate an optimized prompt.

is structured according to the selected framework. Optionally, a temperature bar allows the user to adjust the model's creativity level, with the default set at 0.6. This setting influences how deterministic or varied the generated responses will be.

After evaluation, the user receives a star rating and a list of prompt issues (Figure 3), each with a description and suggestion to resolve it. Users select issues via checkboxes and click"solve" to generate an optimized prompt, which can be saved for future use in the chat. This workflow ensures prompts are refined, enhancing interaction quality and task efficiency.

3.3.3. TextVision's framework integration

TextVision implements a dual approach to prompt engineering: firstly by integrating established frameworks, and secondly by providing its novel, optimised framework. For the integration of existing frameworks, the system includes four well-established prompting guides: the Prompt Pattern Catalogue [19], which provides systematic patterns for improving ChatGPT interactions; PEARL [20], which focuses on planning and executing actions across long documents; Principled Instructions [21], which provides fundamental guidelines for questioning LLMs; and Advanced Prompt Design Methods [22], which offers comprehensive prompt engineering techniques.

In addition to these established frameworks, TextVision is introducing its framework, which combines and streamlines practical approaches from several sources [23, 24, 19, 21, 25]. This framework uniquely integrates core concepts such as the ROMANE approach[1] to structured role-based prompting, chain-of-thought reasoning [25] and systematic prompt patterns while focusing on practical effectiveness.

The framework integration is implemented through a flexible PDF-based approach that dynamically allows the system to load and apply these prompting methodologies. Users can evaluate and optimise their prompts against any of the integrated frameworks or TextVision's custom framework, with the system providing detailed feedback and suggestions for improvement based on the guidelines of the chosen framework. TextVision's custom framework emphasises role-based architecture, chain of reasoning processes and pattern-based design principles [19, 21]. The framework incorporates quality assessment mechanisms with multiple severity levels for consistent, timely evaluation and optimization while maintaining flexibility for different use cases.

4. Architecture

TextVision's implementation follows a modular architecture that integrates AI capabilities while maintaining extensibility and reliability. The system is built around three core components: the AI backend for LLM integration, a prompt management system and a comprehensive file processing pipeline with retrieval-augmented generation (RAG) capabilities. This architecture ensures efficient handling of various AI models, prompt optimization, and document processing while maintaining a clear separation of concerns and high maintainability (see Figure 4).

4.1. Integration of LLMs

TextVision utilizes an abstract factory pattern via the `AbstractAIBackend` class, supporting various LLM providers while ensuring stability. Current models include OpenAI's GPT-4, GPT-4 Turbo, GPT-4o [26] - as of now, the most recent and powerful OpenAI LLM -, GPT-4o-mini, and O1 series for text processing and reasoning [27], with Whisper V2 [28] for speech and DALL-E 3 for images. The architecture's `AbstractAIBackend` interface supports dynamic model switching, concurrent implementations, standardized prompting, RAG

[1] https://www.janeggers.tech/eeblog/2023/besser%2Dprompten%2Dgib%2Dder%2Dki%2Dgut%2Dstrukturierte%2Dromane%2Ddann%2Dgibt%2Dsie%2Ddir%2Dauch%2Ddie%2Drichtigen%2Dantworten/

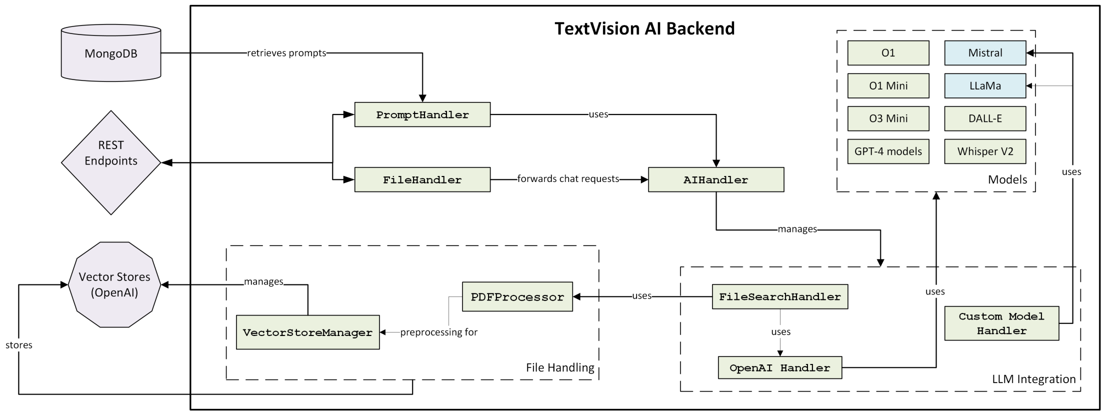

Figure 4: Modular architecture of the TextVision AI backend. The system is composed of three core components: a scalable AI backend for LLM integration, a prompt management system for dynamic prompt optimization, and a file processing pipeline with RAG capabilities.

via `FileSearchHandler`, efficient vector management, and multimodal orchestration. Extensibility is a key strength, enabling integration of self-hosted models like Mistral 7B [29] and potential support for larger models like LLaMA 3B or 70B [30]. These two models are fine-tuned on the `scientific_lay_summarisation - PLOS - normalized` dataset [31], loaded from Huggingface, containing 24,773 article-summary pairs for training, which is designed explicitly for creating lay summaries of scientific literature, aligning with TextVision's scientific approach [32]. TextVision prepares interfaces for emerging models like Claude and Gemini via a standardized API, ensuring consistent context handling across multimodal inputs. It adapts to AI advancements with minimal disruption and targets level 1 combined multimodality by processing voice, text, and images simultaneously [33].

4.2. Prompt management system

TextVision's prompt management system, via the `PromptHandler` class, streamlines prompt handling and optimization through a dynamic, framework-based architecture. It focuses on system prompt management, framework integration, and prompt evaluation/optimization. Predefined system prompt templates ensure consistent communication with LLMs across use cases like evaluation, issue resolution, and suggestion generation. Framework integration employs a PDF-based approach, enabling users to dynamically load prompting frameworks by processing their documentation through the `PDFProcessor` component. Extracted guidelines inform prompt evaluation and optimization. The optimization pipeline tracks iteration counts, scores, and improvements. It refines prompts until a score of 9.0+ is reached, improvements drop below 0.5 points, or three iterations are completed. Issues are categorized by severity (HIGH, MEDIUM, LOW), with detailed compliance feedback for precise and framework-aligned prompt refinement.

4.3. File processing and RAG integration

TextVision's file processing and RAG system, implemented via `FileSearchHandler` and `VectorStoreManager`, enables efficient document handling, storage, and context-aware responses using vector embeddings and retrieval mechanisms [34]. The `PDFProcessor` class encodes/decodes documents between binary and base64 formats for seamless frontend-backend data transmission and converts PDFs to markdown using pymupdf4llm[2]. It validates format compliance and size (up to 512MB), supporting PDFs, markdown, and Word documents while preserving structure and metadata. `VectorStoreManager` underpins the RAG system by managing vector stores for semantic search with configurable TTL settings for efficient resource use. It automates document chunking, embedding generation, and vector store maintenance. `FileSearchHandler` employs an event-driven architecture for real-time and batch search operations, ensuring efficient retrieval and citation tracking across multiple documents. Dynamic context window management optimizes response generation, leveraging OpenAI's vector store capabilities. The modular design supports scalability, maintainability, and future RAG enhancements, ensuring reliable document-based knowledge processing and retrieval [34].

5. Planned user study

We plan to conduct a user study to determine whether the TextVision interface improves document creation efficiency, usability, and overall user experience. We aim to recruit 20 participants for a balanced distribution of 10 people per skill group. Participants will be classified into two groups, low-level and high-level, based on their previous experience and familiarity with AI tools. After participating in the study, the users must complete two questionnaires, the System Usability Scale (SUS) [35] and NASA-TLX [36], and a semi-structured final interview.

6. Conclusion

TextVision merges LLM capabilities with user-centric interfaces via a modular architecture for an intuitive text processing experience. Its dynamic prompt recommendation adapts in real time, simplifying prompt creation, while the `PromptDesigner` enables personalized templates. Multimodal input and RAG techniques support seamless document editing, review, and generation in a unified interface. We will evaluate efficacy and usability through a user study with quantitative and qualitative data collection, and address integration into larger interactive deep learning projects [37, 38].

Acknowledgements

This work was funded by the German Federal Ministry of Education and Research (BMBF) under grant numbers 01IW23002 (No-IDLE) and 01IW24006 (NoIDLEChatGPT), as well as by the Endowed Chair of Applied AI at the University of Oldenburg.

[2]https://pypi.org/project/pymupdf4llm/

References

[1] D. Sonntag, Intelligent user interfaces - a tutorial, arXiv (Cornell University) (2017). URL: https://arxiv.org/abs/1702.05250. doi:10.48550/arxiv.1702.05250.

[2] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, in: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022, OpenReview.net, 2022. URL: https://openreview.net/forum?id=nZeVKeeFYf9.

[3] P. Mathur, A. Siu, V. Manjunatha, T. Sun, DocPilot: Copilot for automating PDF edit workflows in documents, in: Y. Cao, Y. Feng, D. Xiong (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 232–246. URL: https://aclanthology.org/2024.acl-demos.22/. doi:10.18653/v1/2024.acl-demos.22.

[4] G. Lin, T. Feng, P. Han, G. Liu, J. You, Arxiv copilot: A self-evolving and efficient LLM system for personalized academic assistance, in: D. I. Hernandez Farias, T. Hope, M. Li (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 122–130. URL: https://aclanthology.org/2024.emnlp-demo.13/. doi:10.18653/v1/2024.emnlp-demo.13.

[5] S. Wu, H. Shen, D. S. Weld, J. Heer, M. T. Ribeiro, Scattershot: Interactive in-context example curation for text transformation, in: Proceedings of the 28th International Conference on Intelligent User Interfaces, IUI 2023, Sydney, NSW, Australia, March 27-31, 2023, ACM, 2023, pp. 353–367. URL: https://doi.org/10.1145/3581641.3584059. doi:10.1145/3581641.3584059.

[6] R. Fok, H. Kambhamettu, L. Soldaini, J. Bragg, K. Lo, M. A. Hearst, A. Head, D. S. Weld, Scim: Intelligent skimming support for scientific papers, in: Proceedings of the 28th International Conference on Intelligent User Interfaces, IUI 2023, Sydney, NSW, Australia, March 27-31, 2023, ACM, 2023, pp. 476–490. URL: https://doi.org/10.1145/3581641.3584034. doi:10.1145/3581641.3584034.

[7] M. Desmond, Z. Ashktorab, Q. Pan, C. Dugan, J. M. Johnson, Evalullm: Llm assisted evaluation of generative outputs, in: Companion Proceedings of the 29th International Conference on Intelligent User Interfaces, IUI '24 Companion, Association for Computing Machinery, New York, NY, USA, 2024, p. 30–32. URL: https://doi.org/10.1145/3640544.3645216. doi:10.1145/3640544.3645216.

[8] J. Kim, R. C. Flanagan, N. E. Haviland, Z. Sun, S. N. Yakubu, E. A. Maru, K. C. Arnold, Towards full authorship with AI: supporting revision with ai-generated views 3660 (2024). URL: https://ceur-ws.org/Vol-3660/paper17.pdf.

[9] S. Petridis, B. D. Wedin, J. Wexler, M. Pushkarna, A. Donsbach, N. Goyal, C. J. Cai, M. Terry, Constitutionmaker: Interactively critiquing large language models by converting feedback into principles, in: Proceedings of the 29th International Conference on Intelligent User Interfaces, IUI 2024, Greenville, SC, USA, March 18-21, 2024, ACM, 2024, pp. 853–868. URL: https://doi.org/10.1145/3640543.3645144. doi:10.1145/3640543.3645144.

[10] S. P. Neshaei, R. Rietsche, X. Su, T. Wambsganss, Enhancing peer review with ai-powered

suggestion generation assistance: Investigating the design dynamics, in: Proceedings of the 29th International Conference on Intelligent User Interfaces, IUI '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 88–102. URL: https://doi.org/10.1145/3640543.3645169. doi:10.1145/3640543.3645169.

[11] N. Singh, L. L. Wang, J. Bragg, Figura11y: Ai assistance for writing scientific alt text, in: Proceedings of the 29th International Conference on Intelligent User Interfaces, IUI '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 886–906. URL: https://doi.org/10.1145/3640543.3645212. doi:10.1145/3640543.3645212.

[12] K. Lo, J. C. Chang, A. Head, J. Bragg, A. X. Zhang, C. Trier, C. Anastasiades, T. August, R. Authur, D. Bragg, E. Bransom, I. Cachola, S. Candra, Y. Chandrasekhar, Y. Chen, E. Y. Cheng, Y. Chou, D. Downey, R. Evans, R. Fok, F. Hu, R. Huff, D. Kang, T. S. Kim, R. Kinney, A. Kittur, H. B. Kang, E. Klevak, B. Kuehl, M. Langan, M. Latzke, J. Lochner, K. MacMillan, E. Marsh, T. Murray, A. Naik, N. Nguyen, S. Palani, S. Park, C. Paulic, N. Rachatasumrit, S. Rao, P. Sayre, Z. Shen, P. Siangliulue, L. Soldaini, H. Tran, M. van Zuylen, L. L. Wang, C. Wilhelm, C. Wu, J. Yang, A. Zamarron, M. A. Hearst, D. S. Weld, The semantic reader project, Commun. ACM 67 (2024) 50–61. URL: https://doi.org/10.1145/3659096. doi:10.1145/3659096.

[13] J. D. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, Q. Yang, Why johnny can't prompt: How non-ai experts try (and fail) to design LLM prompts, in: A. Schmidt, K. Väänänen, T. Goyal, P. O. Kristensson, A. Peters, S. Mueller, J. R. Williamson, M. L. Wilson (Eds.), Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI 2023, Hamburg, Germany, April 23-28, 2023, ACM, 2023, pp. 437:1–437:21. URL: https://doi.org/10.1145/3544548.3581388. doi:10.1145/3544548.3581388.

[14] S. Sucik, D. Skala, A. Svec, P. Hraska, M. Suppa, Prompterator: Iterate efficiently towards more effective prompts, in: Y. Feng, E. Lefever (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023 - System Demonstrations, Singapore, December 6-10, 2023, Association for Computational Linguistics, 2023, pp. 471–478. URL: https://doi.org/10.18653/v1/2023.emnlp-demo.43. doi:10.18653/V1/2023.EMNLP-DEMO.43.

[15] D. Masson, S. Malacria, G. Casiez, D. Vogel, DirectGPT: A direct manipulation interface to interact with large language models (2024) 975:1–975:16. URL: https://doi.org/10.1145/3613904.3642462. doi:10.1145/3613904.3642462.

[16] A. Ejaz, D. S. A. Ali, M. Y. Ejaz, D. F. A. Siddiqui, Graphic user interface design principles for designing augmented reality applications, International Journal of Advanced Computer Science and Applications 10 (2019). URL: http://dx.doi.org/10.14569/IJACSA.2019.0100228. doi:10.14569/IJACSA.2019.0100228.

[17] D. C. Smith, C. Irby, R. Kimball, E. Harslem, The star user interface: An overview, in: Proceedings of the June 7-10, 1982, national computer conference, 1982, pp. 515–528.

[18] B. Shneiderman, M. Cohen, S. Jacobs, C. Plaisant, N. Diakopoulos, N. Elmqvist, Designing the User Interface Strategies for Effective Human-Computer Interaction, Global Edition, Pearson Deutschland, 2017, pp. 215–219. URL: https://elibrary.pearson.de/book/99.150005/9781292153926.

[19] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, D. C. Schmidt, A prompt pattern catalog to enhance prompt engineering with ChatGPT,

arXiv preprint, 2023. URL: https://arxiv.org/abs/2302.11382. arXiv:2302.11382.
[20] S. Sun, Y. Liu, S. Wang, D. Iter, C. Zhu, M. Iyyer, PEARL: Prompting Large Language Models to Plan and Execute Actions over long documents, 2024. URL: https://aclanthology.org/2024.eacl-long.29.
[21] S. M. Bsharat, A. Myrzakhan, Z. Shen, Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4, arXiv preprint, 2024. URL: https://arxiv.org/abs/2312.16171. arXiv:2312.16171.
[22] X. Amatriain, Prompt design and engineering: Introduction and advanced methods, arXiv preprint, 2024. URL: https://arxiv.org/abs/2401.14423. arXiv:2401.14423.
[23] X. Liu, J. Wang, J. Sun, X. Yuan, G. Dong, P. Di, W. Wang, D. Wang, Prompting frameworks for large language models: A survey, arXiv preprint, 2024. URL: https://arxiv.org/abs/2311.12785. arXiv:2311.12785.
[24] Y. Liu, G. Deng, Z. Xu, Y. Li, Y. Zheng, Y. Zhang, L. Zhao, T. Zhang, K. Wang, Y. Liu, Jailbreaking chatgpt via prompt engineering: An empirical study, 2024. URL: https://arxiv.org/abs/2305.13860. arXiv:2305.13860.
[25] L. Reynolds, K. McDonell, Prompt programming for large language models: Beyond the few-shot paradigm, 2021. URL: https://doi.org/10.1145/3411763.3451760. doi:10.1145/3411763.3451760.
[26] L. B. Hensel, N. Yongsatianchot, P. Torshizi, E. Minucci, S. Marsella, Large language models in textual analysis for gesture selection, in: Proceedings of the 25th International Conference on Multimodal Interaction, ICMI '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 378–387. URL: https://doi.org/10.1145/3577190.3614158. doi:10.1145/3577190.3614158.
[27] OpenAI, GPT-4 technical report, CoRR abs/2303.08774 (2023). URL: https://doi.org/10.48550/arXiv.2303.08774. doi:10.48550/ARXIV.2303.08774. arXiv:2303.08774.
[28] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, I. Sutskever, Robust speech recognition via large-scale weak supervision, in: A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, J. Scarlett (Eds.), International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, volume 202 of *Proceedings of Machine Learning Research*, PMLR, 2023, pp. 28492–28518. URL: https://proceedings.mlr.press/v202/radford23a.html.
[29] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, CoRR abs/2310.06825 (2023). URL: https://doi.org/10.48550/arXiv.2310.06825. doi:10.48550/ARXIV.2310.06825. arXiv:2310.06825.
[30] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Rozière, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen,

H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. M. Kloumann, I. Misra, I. Evtimov, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, et al., The llama 3 herd of models, CoRR abs/2407.21783 (2024). URL: https://doi.org/10.48550/arXiv.2407.21783. doi:10.48550/ARXIV.2407.21783. arXiv:2407.21783.

[31] T. Goldsack, Z. Zhang, C. Lin, C. Scarton, Making science simple: Corpora for the lay summarisation of scientific literature, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, Association for Computational Linguistics, 2022, pp. 10589–10604. URL: https://doi.org/10.18653/v1/2022.emnlp-main.724. doi:10.18653/V1/2022.EMNLP-MAIN.724.

[32] Pszemraj, Scientific lay summarisation - plos - normalized dataset, Hugging Face Datasets, 2024. URL: https://huggingface.co/datasets/pszemraj/scientific_lay_summarisation-plos-norm.

[33] A. K. Sinha, C. Kulkarni, A. Olwal, Levels of multimodal interaction, in: Companion Proceedings of the 26th International Conference on Multimodal Interaction, ICMI Companion '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 51–55. URL: https://doi.org/10.1145/3686215.3690153. doi:10.1145/3686215.3690153.

[34] OpenAI, Assistants file search beta documentation, OpenAI Platform Documentation, 2024. URL: https://platform.openai.com/docs/assistants/tools/file-search.

[35] J. Brooke, SUS-A quick and dirty usability scale, Usability evaluation in industry 189 (1996) 4–7. URL: https://www.taylorfrancis.com/chapters/edit/10.1201/9781498710411-35/sus-quick-dirty-usability-scale-john-brooke, publisher: London, England.

[36] S. G. Hart, Nasa-Task Load Index (NASA-TLX); 20 Years Later, Proceedings of the Human Factors and Ergonomics Society Annual Meeting 50 (2006) 904–908. URL: https://journals.sagepub.com/doi/10.1177/154193120605000909. doi:10.1177/154193120605000909.

[37] D. Sonntag, T. Gouvea, M. Barz, A. Anagnostopoulou, S. Liang, S.-J. Bittner, F. Scheurer, The Interactive Deep Learning Enterprise (No-IDLE) meets ChatGPT, Technical Report, German Research Center for AI, 2024.

[38] D. Sonntag, M. Barz, T. Gouvea, A look under the hood of the Interactive Deep Learning Enterprise (No-IDLE), Technical Report, German Research Center for AI, 2024.