FlowFrame: A New Scrollytelling Format for Journal Articles

Thrisna Singh
University of Pennsylvania
Philadelphia, Pennsylvania, USA
thrisna@seas.upenn.edu

Abstract

While the insights presented from research articles represent the peak of current academic progress, these articles can be difficult to understand due to the dense text in them. This dense text can increase the cognitive overload of a reader due to the language, amount of information being presented, and the separation of figures, tables, and text. While there has been a standard format for academic journal articles for years, a new format for web articles is emerging that utilizes the inherent attributes of computers and hypertext to reimagine the storytelling experience. In the new scrollytelling format, based on the user interaction of scrolling, different and specific visual and textual elements of an article are shown to the user. In this paper, I present a new tool, FlowFrame, that seeks to apply the new scrollytelling format to presenting the information of an academic journal article. FlowFrame attempts to reduce the split-attention effect and the amount of cognitive load that appears when reading a standard academic journal article. We first define the problem addressed and explore related works in the field. Then, we present a system design of our solution FlowFrame and discuss the implications of this work. Finally, we discuss the limitations of the current tool and future directions.

CCS Concepts

• **Human-Computer Interaction → Interaction design**.

Keywords

Scrollytelling, Cognitive Load, Split-attention effect, User Interface Design, Information Visualization, Interactive Reading Tools, FlowFrame

ACM Reference Format:
Thrisna Singh. 2024. FlowFrame: A New Scrollytelling Format for Journal Articles. In *Proceedings of Interactive Reading (CIS 7000 '24).* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Academic journal articles follow a specific format full of dense text interspersed with figures and tables. This dense text is characterized by academic language and lots of prose, making the text difficult to engage with and comprehend. This structure, while a staple in the field of academia, can create barriers to understanding due to both the dense text of articles and the inaccessibility of important figures and tables. The dense and large amount of text can increase the cognitive load of a reader as they try to understand the text. The placement of figures and tables often leads to readers needing to flip between pages to reference these visuals, increasing the split-attention effect exhibited while reading journal articles.

I seek in this paper to address the problem of overwhelm when reading academic journal articles. Specifically, as journal articles are often overwhelming due to the dense and sheer amount of text and the separation of figures and text, leading to the split-attention effect and cognitive overload, we observe that it is more difficult for readers to absorb the information of the journal articles effectively. To address this, I created FlowFrame, which converts academic journal articles into a scrolling format, displaying one paragraph at a time with its associated visual. This approach aims to reduce cognitive load, enhance readability, and minimize the split-attention effect by ensuring that readers have immediate access to both text and visuals in a clean and unified format.

The problem of reading dense text without associated visuals always present has been studied for decades. In 1982, 55 experiments were conducted to indicate the positive effects of illustrations on reading [6]. Building on this study, Butcher investigated the effect on comprehension by testing readers understanding of the circulatory system based off of text-only reading, text and simple diagrams, and text and complex diagrams [3]. Based on the amount of research conducted on adding multimedia to text, it is clear that this is an important field to examine with important problems present. As no one has addressed this problem with respect to academic journal articles, I attempt to do so in this paper.

To address the problem of separation of text and visuals and the amount of overwhelming text in a journal article, in this paper, I make the following contributions: First, I examine related work and the problem statement. Then, I discuss the system design of my solution, FlowFrame, and the implications of it. Finally, I conclude by discussing the current limitations of FlowFrame as well as future work to be done.

2 Background and Related Work

In this section, we provide background on cognitive load theory and the split-attention effect, both of which I attempt to lower when reading journal articles with FlowFrame compared to the standard format for journal articles. Then, I discuss related work in the field and how the ideas of FlowFrame go beyond this work.

2.1 Cognitive Load Theory

Cognitive Load Theory is a really important concept in modern psychology. It establishes the assumption that human beings have a limited capacity for their working memory and that the more cognitive activities being conducted, the more restraints we place

CIS 7000 '24, December 12, 2024, Philadelphia, PA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

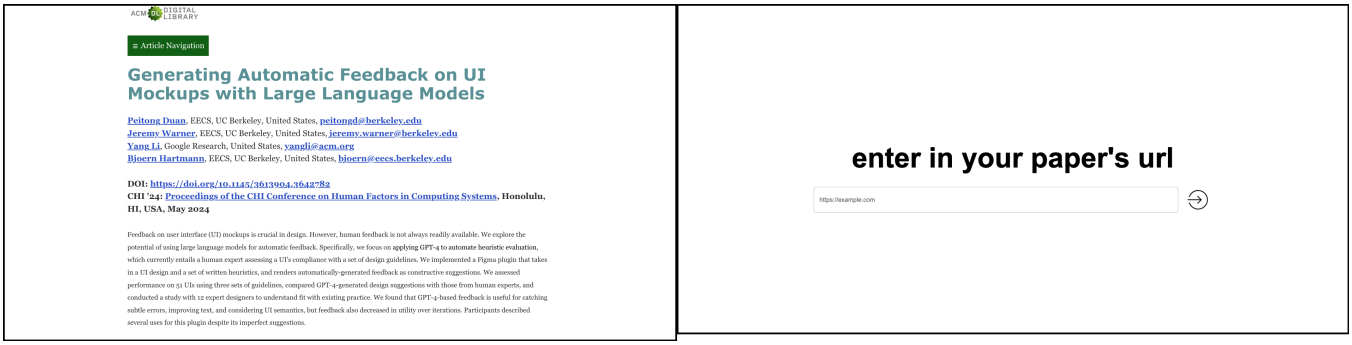

Figure 1: Box A to the left showcases an example HTML-formatted journal article that can be inputted into FlowFrame for conversion to a scrollytelling format. Box B to the right showcases the opening screen of the tool where users can enter in the URL of a journal article.

on this capacity [4]. Due to the limited capacity of working memory, we want to optimize the cognitive activities being conducted. Specifically, the main concern of this theory deals with how people allocate the limited cognitive resources they have when conducting problem-solving and learning activities [4].

2.2 Split-attention effect

The split-attention effect specifically occurs when people must split their attention between different information sources and then mentally integrate these various information sources [4]. For instance, this can occur when learners must split their attention between text and figures. The mental integration that occurs due to synthesizing the various information together uses cognitive resources and thus increases cognitive activities and a person's cognitive load. The physical integration of information sources, such as placing figures and text together, can reduce a reader's cognitive load and improve learning of the material. In experiments conducted by Chandler and Sweller, it was demonstrated that the integration physically of text and figures increased comprehension (as determined by test question performance) and decreased the amount of time taken to process information [4].

2.3 Work Done Thus Far

There has been a lot of research done on combining text and visuals, as well as on the scrolling-telling article format. We discuss some of those in this section.

Butcher conducted a study to test comprehension of the circulatory system using differently formatted text, specifically the following cases: text-only, text with complex diagrams, and text with simple diagrams. She introduced material about the circulatory system a section at a time and allowed readers to click through the material using arrows. After testing comprehension via the creation of inference questions, drawings of one's understanding, and more, she found that out of all the cases, readers gain the most comprehension with simple diagrams and text. [3]. We expand upon Butcher's work of associating a small amount of text with a diagram in this paper by expanding this to the use case of understanding and improving comprehension of academic journal articles rather than school material.

A journal article by Sriram Karthik Badam, Zhicheng Liu, and Niklas Elmqvist also attempts to couple text and visuals, but this time through the creation of Elastic Documents. This specific approach only focuses on data and tables. It allows users to view text and its associated tables. While this tool also provides data visualization, such as via charts, to showcase different formats of the data [2], my work takes the concept of showing tables alongside text but expands this to also include images.

While we do not use Idyll in this paper, the concept of FlowFrame takes inspiration from interactive web articles published using Idyll. Idyll is a language that allows authors to create interactive articles. This includes scrollytelling news articles as well as articles with other, previously JavaScript-based components. Idyll presents an overall language that authors then have to employ in order to create the scrollytelling format that they want to showcase to readers [5]. I take inspiration from the types of articles that can be created by Idyll, but automate the process by automatically taking text and images and converting to a scrolling-telling format without rewriting it in Idyll.

3 Problem Overview

As stated earlier, through the creation of FlowFrame, I attempt to address the problem of overwhelm while reading academic journal articles by reducing the split-attention effect and cognitive load that occur while reading. Specifically, I attempt to answer the following research questions through the creation of FlowFrame:

RQ1. How does a scrolling-telling format with synchronized text and visuals impact comprehension, cognitive load, engagement, and the split-attention effect?

RQ2. Does aligning visuals with corresponding text improve retention, recall, and the reader's ability to connect concepts while reducing the split-attention effect?

In the rest of this article, we discuss the system design of FlowFrame and how this design is led with these research questions in mind.

Figure 2: Box A to the left showcases the initial title screen shown after an article has been processed. Box B in the middle showcases the abstract screen shown upon scrolling down from the title screen. Box C to the right showcases the standard layout of content for the majority of an article.

4 System Design

In this section, we will discuss the technical design of FlowFrame. FlowFrame is a Flask application. Users must use Google Chrome to run the application, and they must have ChromeDriver installed.

4.1 Translating URL To HTML

FlowFrame begins with a URL input screen, shown in Figure 1 (Box 2). The tool accepts the URLs of HTML versions of academic journal articles. An example of one of these articles can be seen in Figure 1 (Box 1). This article can be found at https://dl.acm.org/doi/fullHtml/10.1145/3613904.3642782. Once a user clicks the arrow on the screen or presses Enter on the keyboard, FlowFrame will automatically run a Python script that uses ChromeDriver to save the HTML contents of the inputted URL, as well as any images on the webpage.

4.2 Translating HTML To JSON

FlowFrame then takes the generated HTML and images and creates a few JSON files to store the contents of both by running a Python script.

First, the contents regarding the title of the paper, the author(s) information, conference information, DOI linkage, keywords, CSS concepts, and ACM Reference Format are extracted from the HTML of the journal article and saved to a title.json file.

Next, the id, label, and citation of all references listed in the journal article are saved to a citations.json file.

Last, but not least, the core contents of the paper are saved into a content.json file. This file stores each section (as well as any subsections and subsubsections of that section) and all visual elements of the journal article. Visual elements include any figures or tables. For each visual element, information regarding the element type, id, file path, and caption is saved. For each section/subsection/subsubsection, the id, title, paragraphs and any lists in those bodies of text are saved. For any piece of text, such as a paragraph or a caption, an array of individual sentences are saved, as well as the associated visual for each paragraph. We determine associated visuals for a paragraph using the following rules:

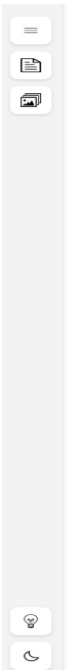

Figure 3: This showcases the menu bar for FlowFrame. From the top to the bottom, the buttons are for the following: Navigation Panel, References Panel, Diagrams Panel, Highlights, and Dark Mode.

(1) If a paragraph explicitly mentions a figure or table (or multiple ones) by name, the associated visual will be the figure or table with the most mentions in that paragraph.
(2) Else, the associated visual for a paragraph will be the figure or table that was last shown in the original text prior to the current paragraph. I base this off the assumption that figures and tables appear in a journal article in order of relevance to the surrounding text.

Translating the URL to HTML and then translating the HTML to JSON takes on average 13 seconds to complete.

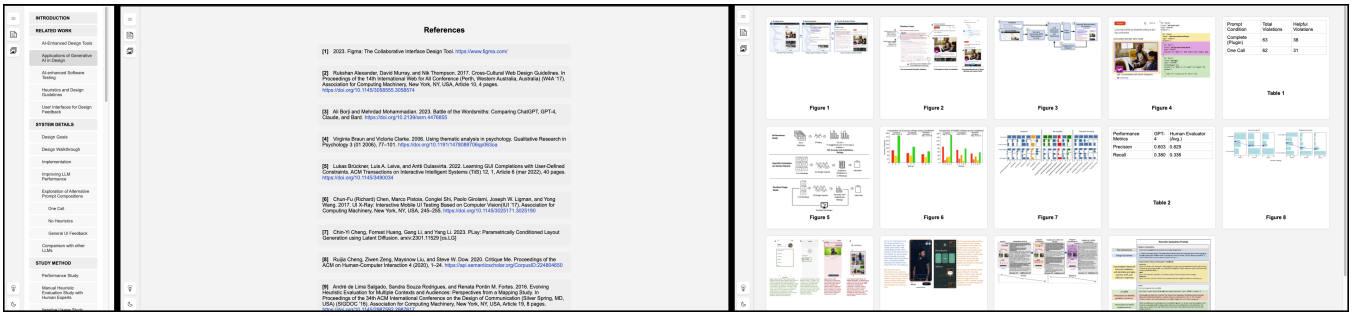

Figure 4: Box A to the left showcases the navigation panel for FlowFrame. Box B in the middle showcases the opened-up References Panel. Box C to the right showcases the Diagrams Panel, containing all figures and tables for the paper.

4.3 UI Implementation

In this section, we will showcase all of the different UI features of FlowFrame.

4.3.1 ScrollyTelling Format. After the URL has been translated to HTML and the HTML has been translated to a JSON format, the JSONs are then used to showcase the content in a scrollytelling format. First, all of the contents of the title.json file are shown to the reader, as shown in Figure 2 (Box A). When the user scrolls down, the screen updates to showcase the abstract of the article in the center of the screen. This can be seen in Figure 2 (Box B). Another downward scroll showcases the main content layout, discussed more in detail in a later section. Users can scroll both forward and backward between different "screens" of text and their associated visuals. This allows users to go forward to the next paragraph of the journal article or backwards to the previous paragraph of the article. Only one paragraph is shown to the user at a time. If an associated visual is the same between two consecutive paragraphs, then it is only the text that updates. This ensures a smooth transition of content. Users can also scroll faster if they want to scroll past multiple screens at a time.

4.3.2 Content Layout. The main content layout for a section can be seen in Figure 2 (Box C). I aimed for this to have a clean, minimal look to reduce overwhelm and cognitive overload. All content is centered in the middle of the screen with a white background. The title of the section can be seen at the top. On the left, one paragraph of text is shown at a time. If this paragraph is part of a subsection or a subsubsection, the corresponding titles for these are shown on top of the paragraph. On the right, the figure or table associated with the paragraph is shown. The caption of the figure or table is included below it. If users want to expand the table/figure, they can click a tiny magnifying glass in the upper right-hand corner of the image. This will showcase a popup that showcases a larger version of the table/figure, as well as the caption for the associated visual.

4.3.3 Menu Bar. On the left-hand side, there is a vertical menu bar with five buttons for the following:

- Navigation Panel
- References Panel
- Diagram Panel
- Highlights
- Dark Mode

The menu bar can be seen in Figure 3.

4.3.4 Navigation Panel. As shown in Figure 4 (Box A), the Navigation Panel showcases each section, subsection, and subsubsection of the journal article. It allows users to jump to a specific portion of the article. This gives readers the flexibility to jump to a specific section without needing to scroll all the way there. This is useful for readers who are coming back to reread a specific section or reference that section.

4.3.5 References Panel. The References Panel, after clicking the second button in Figure 3, slides open to reveal all citations from the article (Figure 4 (Box B)). Instead of scrolling down to the bottom of the journal article to see the entire list of citations, this panel provides access to that list of citations at any point during the reading journey.

4.3.6 Diagram Panel. The Diagram Panel slides open after clicking the third button on the menu bar (Figure 3). This panel, shown in Figure 4 (Box C) showcases each figure or table contained in the journal article in an item box, as well as the name of that figure or table. If a user were to click a figure or table, a popup would expand to showcase that visual in a larger view. The popup would also contain the caption for the visual.

The Diagram Panel can be accessed at any point during the reading experience. This means that users can get a global perspective of all visuals within the article at any given time.

4.3.7 Inline Popups. In the paragraphs of content shown to users (Figure 2), some text may be colored blue. This indicates that users can click that text to receive a popup with more information.

There are two potential inline popups that a user may come across in FlowFrame. For every reference to a citation, normally shown as a number(s) in brackets, in a paragraph, users can click the reference number to receive a popup showcasing that specific citation (Figure 5 (Box A)). The reference number corresponds to the same reference number for that citation shown in the References Panel. Users can click the x in the upper right-hand corner or outside of the popup to return to the content that they were originally reading. If a figure is explicitly mentioned in a paragraph as well, users can click the figure name to receive a popup showcasing that visual and its caption (Figure 5 (Box B)). This is account for paragraphs that mention multiple figures, as I only show one associated visual next to the paragraph shown to the user.

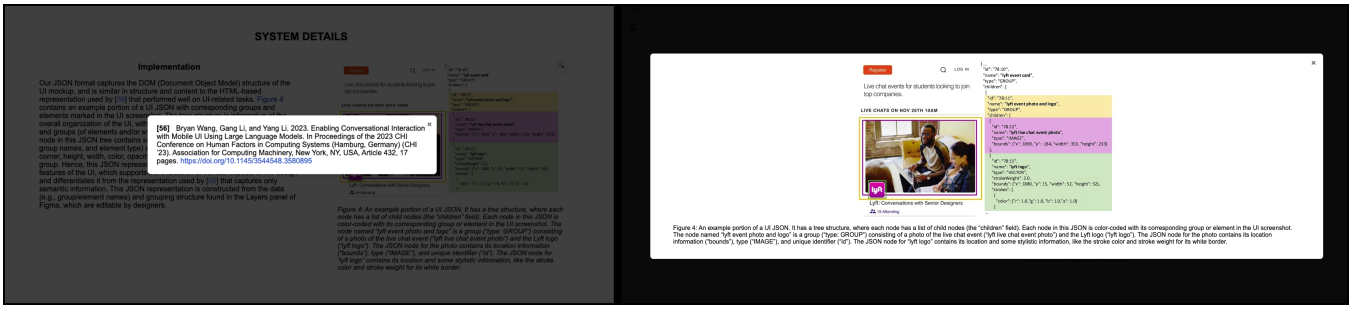

Figure 5: Box A to the left showcases a popup that appears when clicking inline citations. Box B on the right showcases a popup that appears while clicking inline references to a figure or by clicking the magnifying glass of the associated visual or by clicking a specific figure or table in the Diagrams Panel.

4.3.8 Dark Mode. The last button in the menu bar (Figure 3) adds a preference and convenience feature for users. It will translate FlowFrame to a "dark mode," where the coloring behind all features turns darker, as shown in Figure 6.

4.3.9 Highlighting. On the menu bar (Figure 3), there is a button containing a lightbulb. If users click this button, the highlighting feature contained within FlowFrame is turned on. Clicking on the button again toggles the feature off.

This highlighting feature highlights similar sentences between the text paragraph shown on the screen and the caption of the associated visual of that paragraph. If a user hovers over a sentence in either the paragraph or the caption and an associated sentence exists, both sentences will be highlighted. I felt that showcasing all highlights to the user at once was overwhelming for some paragraphs, as some paragraphs are completely about a specific figure, thus leading to a lot of highlights occurring between the paragraph and caption. Thus, users must hover over a sentence to see the corresponding highlight.

Some paragraphs may have no highlights, as that paragraph does not explicitly talk about something mentioned in the caption of the associated visual. The lightbulb button in the menu bar can show the state of highlights on the current screen. The state of the button updates as users scroll. As shown in Figure 7, the lightbulb button can have the following states:

- If the button has a white background without any glow around the button, the highlighting feature is not on.
- If the button has a yellow background, the highlighting feature is on, and there are highlights on the current screen that can be accessed by hovering over sentences.
- If the button has a white background and a yellow glow around the button, the highlight feature is on, but there are no highlights on the current screen.

To determine if two sentences are similar, I use a cosine similarity function to assess the similarity in their words and structure.

5 Discussion

FlowFrame provides a sleek, clean design that does two things: combine visual and textual elements to reduce any split-attention

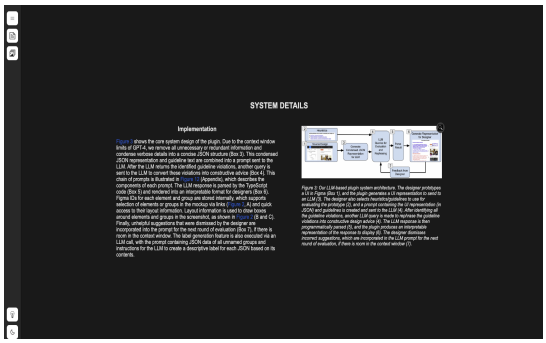

Figure 6: This showcases how the tool appears when using Dark Mode.

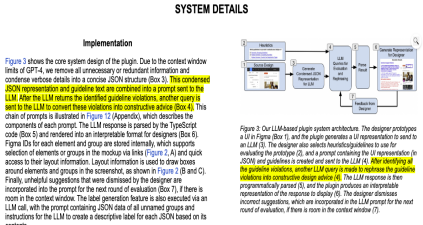

Figure 7: This showcases FlowFrame's highlighting tool.

effects caused by switching contexts within previous journal article design and reduce the amount of dense text shown at once to the user. Each feature of the tool acts to reduce the amount of cognitive activity required while reading and learning the material within a journal article and thus should increase overall comprehension of the article.

Feedback from users demoing the tool liked the sleek look of the tool, with content material being in the center of a white screen. This sleek look acts to minimize content and distractions. One comment though stated that the text did appear to have very close

spacing and overall usability would improve if this was increased. As the main content screen showcases an associated visual and a paragraph at a time, this combination of material for the main content should increase comprehension due to the lowering of the split-attention effect and lowering of cognitive load from that effect. The scrolling method will improve engagement as users physically have to interact throughout the reading of the material.

It is still up in the air if this constant engagement is productive to user comprehension or tiring. As such, the inclusion of a navigation panel allows for multiple outlets for users to engage with the content of the journal article. This gives users flexibility while using FlowFrame on how they want to navigate information.

It does seem like the combination of visual elements and text on the main content frame for FlowFrame does increase the ability of users to make connections between figures and text. The highlighting tool especially helps this cause, as does the automatic association of visual elements with the text shown. Users do not have to spend time finding the related diagram and thus can save cognitive resources while reading. As well, users do not have to spend time finding the specific sentences that are related to portions of the caption. While this is a benefit of FlowFrame, more research needs to be conducted to see if this format of showcasing journal articles improves overall retention and recall of information.

The reduction of material showcased to the user should also reduce reader overwhelm when approaching a journal article, as only a small portion of material is shown at a time to users. We do make sure that readers have the resources needed to gain the extra information that they may like, though, without having all of that information constantly being presented to them. This can be seen through the inline popups that FlowFrame employs that allow a user to see information regarding a specific citation or a figure upon a click. The requirement of hovering over text to showcase highlights also acts to reduce overwhelm. Users still get the information they need, such as knowing if there are highlights on a screen via the status of the highlights button, but it is not in their face at all times.

The citations panel and diagrams panel group important information together that can be accessed at any point during reading. Users do not have to scroll to the bottom of the article to see all citations, but rather can click a button and see all at once. In addition, users do not need to hop through all pages of a journal article to find all diagrams but can view all of them at once due to immediate access to the diagrams panel. This panel offers an overarching visual overview of the article. This is unique and cannot be found in a regular journal article or in other tools. This can be useful for visual learners who might understand the text better by looking first at all of the diagrams and tables present.

Overall, the tool appears to reduce the split-attention effect and reduce cognitive overload by placing text and diagrams together. It is novel in the fact that it is able to turn any HTML journal article URL into a scrolling-telling format and that it changes the format of content of a journal article to showcase associated text and diagrams always next to each other.

6 Limitations

Despite the tool addressing the research questions posed and adhiring to the design goals of this paper, there are still some limitations of FlowFrame that need to be discussed.

First of all, the highlighting feature contained within FlowFrame uses a cosine simalarity function to determine how similar sentences in a text paragraph are to sentences in the caption of that paragraph's associated visual. This function determines similarity, and therefore if two sentences (one from the caption and one from the text paragraph) should be highlighted together based on how similar the words and structure of the sentences are, rather than if the meaning of each content is similar. While an LLM may have hallucinations sometimes, it would be more accurate in determining similarity between the true meaning of two sentences. It should also be noted that the use of an LLM to benefit the accuracy of this feature would increase costs of the tool.

Another limitation of FlowFrame can be seen in the fact that there was no proper usability study conducted to see how effective of a tool FlowFrame is. Without user feedback, we do not know if the tool is truly useful for helping the reading experience of academic journal articles. A qualitative and quantitative study would help us to determine what portions of the tool are useful and which can be improved. For instance, through a qualitative study, we could determine if users enjoy the scrollytelling format proposed in this tool or if they find it annoying to constantly have to scroll through the article. This would help us shape how future iterations of the tool look, such as adding in an automatic scrolling feature.

Finally, one last limitation is that some articles either do not have a lot of figures or tables included or they do not showcase these figures till later in the article. For instance, some authors may make the choice to not include any figures or tables until after the Introduction. FlowFrame would then not have any associated visual for the paragraphs in these sections, reducing the effectiveness of FlowFrame.

7 Future Work

While FlowFrame has the potential to benefit readers of academic journal articles right now, there are still a lot of features and paths that can be employed to make the tool more robust.

Inclusion of PDF Formats. For instance, as mentioned previously, one of the main limitations of this tool is that it is only able to translate an HTML version of the article into a scrollytelling format. While articles are becoming more available in these formats, there are still a lot of articles from the past and ones that are being deployed today that are not available in the HTML format. Thus, the next iteration of FlowFrame should be expanded to handle PDF formats as this will make the tool more accessible to users.

Inclusion of Image Segmentation. Currently, there is emerging research in the field of image segmentation. Image segmentation allows for the isolation of different regions within figures and tables [1]. We can combine these tools with augmented reading interfaces

to connect text to specific portions of an image. Specifically, Alyssa Wang is conducting novel research in this field to create interfaces that create connections between sentences in the main text of a journal article, sentences within a figure's caption, and specific segments of that figure [7]. Future iterations of FlowFrame would seek to incorporate Wang's research into the tool, as this would allow for more specific connections between a text paragraph shown in FlowFrame and its associated visual and its caption. Specifically, it would expand FlowFrame's functionality to also showcase connections from within a figure/table to a user.

Transition to a Holistic Tool. Rather than simply being a tool to help users read content, FlowFrame can be expanded to be a more holistic reading comprehension tool for readers of academic journal articles. One way that this can be done would be to add annotation features to FlowFrame. For instance, in the future, we can incorporate note-taking capabilities and user-based highlighting into the tool. In addition to this, we can expand the tool to allow users to maintain a shared history of articles that have been read. Users would be able to switch between different journal articles and see the scrollytelling format of each one. They would also have access to their annotations of those articles and be able to refer back to their personal highlights and notes.

Adding Keyboard Shortcuts. At the moment, FlowFrame allows for navigation within an article via scrolling or by using the navigation menu to jump to a specific section. We can make the tool more accessible by allowing readers to use keyboard shortcuts to navigate through the reading. For instance, readers can use the up or down arrow navigation keys to scroll up or down within the article. This can be expanded further to allow for keyboard shortcuts throughout the applications, such as clicking a specific key to showcase the diagrams panel.

8 Conclusion

To conclude, FlowFrame is a great first step in reducing the overwhelm caused by readers trying to understand journal articles. It successfully takes in the URL of an HTML-formatted journal article and translates the contents of that article to a scrollytelling format that showcases one paragraph of text at a time, alongside its associated visual and the caption of that visual. It incorporates features, such as highlights between captions and sentence paragraphs and diagram panels, to provide information to the reader in a concise manner. The layout of the tool acts to reduce the split-attention effect present in the normal format of journal articles and thus also reduces the cognitive load of a reader.

References

[1] Meta AI. 2024. SAM-2: Segment Anything in Images and Videos. *Meta AI Research* (2024). https://ai.meta.com/research/publications/sam-2-segment-anything-in-images-and-videos/ Accessed: 2024-12-07.

[2] Sriram Karthik Badam, Zhicheng Liu, and Niklas Elmqvist. 2019. Elastic Documents: Coupling Text and Tables through Contextual Visualizations for Enhanced Document Reading. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 661–671. https://doi.org/10.1109/TVCG.2018.2865119

[3] Kirsten R. Butcher. 2006. Learning from text with diagrams: Promoting mental model development and inference generation. *Journal of Educational Psychology* 98, 1 (2006), 182.

[4] Paul Chandler and John Sweller. 1992. The split-attention effect as a factor in the design of instruction. *British Journal of Educational Psychology* 62, 2 (1992), 233–246.

[5] Matthew Conlen and Jeffrey Heer. 2018. Idyll: A Markup Language for Authoring and Publishing Interactive Articles on the Web. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 977–989. https://doi.org/10.1145/3242587.3242600

[6] W.H. Levie and R. Lentz. 1982. Effects of text illustrations: A review of research. *ECTJ* 30 (1982), 195–232. https://doi.org/10.1007/BF02765184

[7] Alyssa Wang. 2024. interactive-figures. https://github.com/ahwang16/interactive-figures. Accessed: 2024-12-07, Published: 2024-06-16.