Wigglite: Low-cost Information Collection and Triage

Michael Xieyang Liu
Carnegie Mellon University
xieyangl@cs.cmu.edu

Andrew Kuznetsov
Carnegie Mellon University
adkuznet@cs.cmu.edu

Yongsung Kim
Carnegie Mellon University
yongsung@cmu.edu

Joseph Chee Chang
Allen Institute for AI
josephc@allenai.org

Aniket Kittur
Carnegie Mellon University
nkittur@cs.cmu.edu

Brad A. Myers
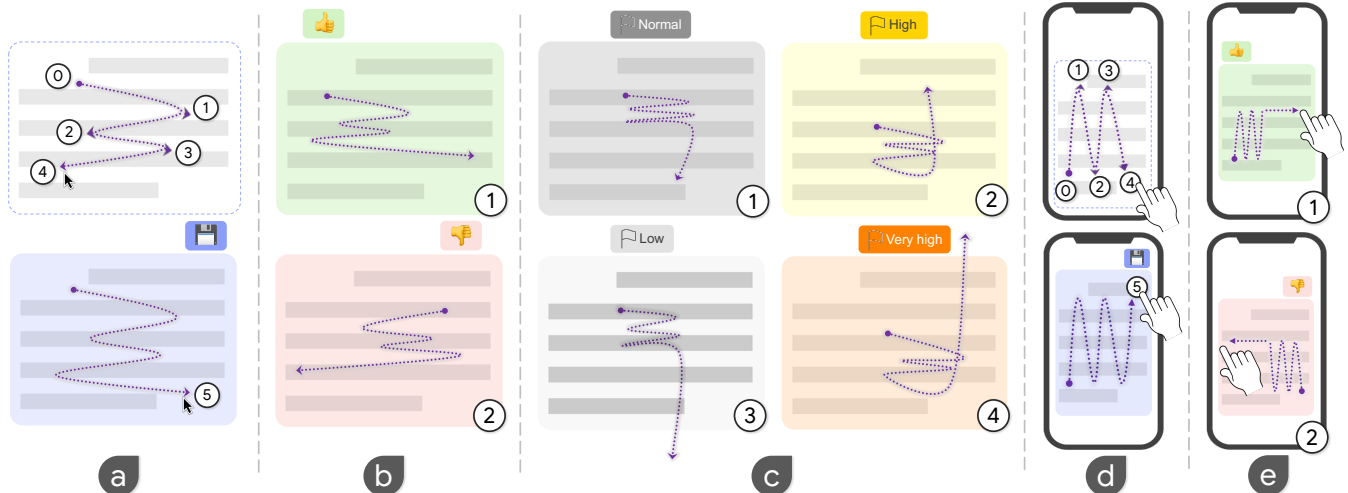Carnegie Mellon University
bam@cs.cmu.edu

Figure 1: We introduce the "wiggling" technique: rapid back-and-forth movements of a mouse pointer on desktop (a) or a finger on mobile devices (d) that do not require any clicking to perform, yet are sufficiently accurate to select the desired content, while at the same time supporting an optional and natural encoding of valence rating (positive to negative) (on desktop: b1-2, on mobile: e1-2) or classification of priority (to facilitate triage) (c1-4) by ending the wiggle with a swipe in different directions.

ABSTRACT

Consumers conducting comparison shopping, researchers making sense of competitive space, and developers looking for code snippets online all face the challenge of capturing the information they find for later use without interrupting their current flow. In addition, during many learning and exploration tasks, people need to externalize their mental context, such as estimating how urgent a topic is to follow up on, or rating a piece of evidence as a "pro" or "con," which helps scaffold subsequent deeper exploration. However, current approaches incur a high cost, often requiring users to select, copy, context switch, paste, and annotate information in a separate document without offering specific affordances that capture their mental context. In this work, we explore a new interaction technique called "wiggling," which can be used to fluidly collect, organize, and rate information during early sensemaking stages with a single gesture. Wiggling involves rapid back-and-forth movements
of a pointer or up-and-down scrolling on a smartphone, which can indicate the information to be collected and its valence, using a single, light-weight gesture that does not interfere with other interactions that are already available. Through implementation and user evaluation, we found that wiggling helped participants accurately collect information and encode their mental context with a 58% reduction in operational cost while being 24% faster compared to a common baseline.

CCS CONCEPTS

• **Human-centered computing → Interactive systems and tools**.

KEYWORDS

Sensemaking, Mental models, Interaction technique

1 INTRODUCTION

From consumers researching products, to patients making sense of medical diagnoses, to developers looking for solutions to programming problems, people spend a significant amount of time on the internet discovering and researching different options, prioritizing which to explore next, and learning about the different trade-offs that make them more or less suitable for their personal goals [16–18, 52, 63, 64]. For example, a YouTuber seeking to upgrade her vlogging setup may learn about many different camera options from various online sites. As she discovers them, she implicitly prioritizes which are the most likely candidates she wants to investigate first, looking for video samples and technical reviews online that speak either positively or negatively about those cameras. Similarly, a patient might keep track of different treatment options and reports on positive or negative outcomes; or a developer might go through multiple Stack Overflow and blog posts to collect possible solutions and code snippets relevant to their programming problem, noting trade-offs about each along the way.

While the number of options, their likely importance, and evidence about their suitability can quickly exceed the limits of working memory, the high friction of externalizing this mental context means that people often still keep all this information in their heads [15, 38, 63, 69, 85]. Despite the multiple tools and methods that people use to capture information, such as copying and pasting relevant texts and links into a notes app or email [8], taking screenshots and photos [84], or using a web clipper [27], collecting web content and encoding a user's mental context about it remains a cognitively and physically demanding process involving many different components: just the collection component itself involves deciding what and how much to collect, specifying the boundaries of the selection, copying it, switching context to the target application tab or window, transferring the information into the application where it will be stored [31], causing frequent interruptions to the users' main flow of reading and understanding the actual web content [38, 52, 69], especially on mobile devices [15, 36]. In addition, components such as prioritizing options by importance result in additional overhead to move or mark their expected utility, which can change as users discover new options or old assumptions become obsolete. When further investigating each option, to keep track of evidence about its suitability, a user further needs to copy and paste each piece of evidence (e.g., text or images from a review or link to a video) and annotate it with how positive or negative it is relative to the user's goals.

Beyond the cognitive and physical overhead of collecting content and encoding context, prior work suggests that for learning and exploration tasks, people are often uncertain about which information will eventually turn out to be relevant and useful, especially at the early stages when there are many unknown unknowns [6, 15, 32, 36]. This could further render people hesitant to exert effort to externalize their mental context if that effort might be later thrown away [33, 63]. One relevant example is Kittur's Clipper [51, 52], which proactively prompted people to specify the "valence" (rating of good or bad) of an option (e.g., a specific camera) measured on a particular dimension (e.g., autofocus capability) as they collected information. Even though this elicitation of the mental

model was done in situ and after much optimization of the interaction, it still required significant cognitive and physical effort and interruption, which prevented its widespread adoption. Other web clipping tools offer even less scaffolding for encoding mental context, typically only supporting a catch-all notes field that people rarely know how to take advantage of [23].

To summarize, we frame a fundamental sensemaking challenge for people trying to research and make decisions online as the high friction involved in capturing: (1) the content that they want to keep track of, which can range from a word, a phrase, an image, to a paragraph or multiple blocks of mixed multimedia content, (2) which option or topic that content corresponds to and its perceived priority for further investigation (which is called "triaging"), and (3) whether the evidence they find about that option or topic is positive or negative regarding its suitability for the user's goals (which is called "valence") [63].

Our vision in this work is to create a technique that reduces the friction for the transfer of a user's internal mental judgements while they are processing information into an external system that will capture those judgements and scaffold sensemaking and exploration. While it is a challenge for the cost of this transfer to be zero, we aim to reduce the overhead significantly by exploring a new class of gestures for this purpose based on "wiggling:" rapid back-and-forth movements of a pointer that do not require any clicking to perform, yet are sufficiently precise to accurately select the desired content, while at the same time supporting the optional and natural encoding of valence rating (positive to negative) and classification of priority (to facilitate triage) to the collected information (see Figure 1). In addition, this technique does not conflict with typical existing interactions (like selecting text or clicking on hyperlinks) and can be extended to other device form factors such as touchscreens and mobile. The rating and classification can be applied by ending the "wiggle" with a swipe in different directions (see Figure 1b,c,e).

We instantiate this class of wiggle-based gesture in an event-driven JavaScript library and a prototype system called Wigglite[1], which builds on top of an existing information and task management application called Skeema that already supports *clipping* and assigning *valence* to general web content as well as organizing them into *topics* with *priorities*. Wigglite consists of a Chrome extension and a mobile application, which enables users to capture and classify information fluidly while searching and browsing. To combat the issue of potentially collecting too much information, the system enables users to easily filter and sort the collected information based on the encoding that the users applied at collection time (or later).

In a lab evaluation with participants, we found that using Wigglite to collect and triage information incurs 58% less overhead cost to perform without sacrificing operational accuracy. In addition, participants generally preferred the wiggling techniques over Skeema alone due to its easiness and naturalness to perform as well as its ability to encode their mental contexts in an organic way.

The primary contributions described in this paper include:

- A novel class of wiggle-based gestures that are cognitively and physically lightweight to perform to collect information, and can simultaneously encode aspects of users' mental context,

---

[1]Wigglite stands for **W**iggling for **I**nformation **G**athering and **G**enerating **L**ightweight **I**mpressions for **T**riage and **E**ncoding.

- A prototype event-driven JavaScript library that implements such gestures and runs in web browsers,
- Wigglite, a prototype system that takes advantage of the wiggle-based gestures to enable information capturing and classification during sensemaking that works on both desktop and mobile devices,
- A lab evaluation that offers empirical insights into the usability, usefulness, and effectiveness of the Wigglite system.

2 RELATED WORK

2.1 Making Sense of Online Information

This work builds on theories of sensemaking as defined as developing a mental model of an information space in service of a user's goals [25, 53, 79]. At a high level, the sensemaking process involves alternating between two phases: *foraging*, which involves people searching for and extracting information, often from various data sources; and *sensemaking*, the process of integrating the amassed information to form a schema or representation to interpret the space [74]. In the following two sections, we present a brief review of prior studies and tools that are pertinent to these two phases.

2.2 Capturing Information

It has been reported that the foraging phase is where people spend the majority of time during a sensemaking process [9, 14, 68, 73]. Therefore, there have been many research and commercial tools to try to help people better capture information during this phase. Some focused on keeping track of entire webpages or documents, such as SenseMaker [4] and browser bookmarks and reading lists; while others enabled users to capture finer-grain units within a web document, such as Hunter Gather [81], Clipper [52], and Google Notebook [34]. However, there is usually a high cost associated with these clipping mechanisms, from carefully maneuvering the cursor to specify the selection boundaries to frequent context switches between the content being read and the note-taking applications.

Prior work has also explored various ways to speed up the collection process. On the one hand, multiple approaches have been proposed to make *selecting* desired content faster by offering pre-defined selection boundaries. For example, systems like Entity Quick Click and Citrine [7, 47, 82] employ techniques like named-entity recognition [67] to pre-process and highlight semantically meaningful entities in a document and allow users to collect and annotate relevant information with a single click. However, no matter how subtle the highlighting is, this could lead to distractions to a user's cognition process while comprehending the actual web content. Wigglite, in contrast, leverages the organic boundaries that are readily defined on web content (e.g., individual words, phrases, as well as block level HTML tags, such as <div></div>, <p></p>, <img />, etc.) [15], and only shows the boundaries when a wiggle is activated, which minimizes distractions to the reading experience.

On the other hand, research and commercial products have explored *collecting* information on behalf of users as they search and browse the web. The history view in most modern browsers offers a linear depiction of a user's activities on a page level. Unfortunately, this format proves to provide little cue for helping people recall the information they have seen [57, 95], and few people reported using the history feature [3, 11, 48, 49, 89]. Works such as Thresher

[39] and Dontcheva et al.'s web summarization tool [26] let users create and curate patterns and templates of information that they want to collect through examples, and then automatically collect that information from pages that users visit in the future. However, sensemaking is a dynamic process [74], especially during the foraging phase [52], and, unlike Wigglite, these template-based approaches lack the necessary flexibility for users to capture whatever they want on the fly. Our recent system called Crystalline [65] explored having a system automatically collect potentially important information by leveraging natural language processing (NLP) techniques to understand the content users are browsing as well as signals from their browsing behavior, such as cursor positions [44], clicks [37], and dwell time on a page [21] to implicitly indicate the user's interest. However, Crystalline is specifically designed and tuned for the domain of programming where web content is usually regularly positioned and formatted [43, 63]. It is unclear how this approach would scale to general web content.

2.3 Rating and Organizing Information

Prior work has introduced various ways to incorporate information classification into the foraging phase. For example, Clipper [52], Unakite [63], and Adamite [41] all prompt the user to optionally categorize an information clip after it has just been captured. Spar.tag.us [40] enables users to associate custom tags with individual paragraphs. ForSense [75] leverages natural language processing to automatically cluster information clips based on themes and topics. Wigglite draws from and builds upon this prior work while focusing on selecting, clipping, rating, and classifying the information all in one gesture.

There have also been a number of research tools developed to support in-depth organizing and structuring, such as the WebBook and WebForager by Card et al. [13], which use a book metaphor to find, collect, and manage web pages and information, Webcutter, which collects and presents URL collections in tree, star, and fisheye views [66], SenseMaker [4] for evolving collections of information, and Mesh [16], and our Unakite [63, 64] which build comparison tables of various options and criteria. However, in the current work, we focus on helping users organize information into topics, a quick yet sufficiently expressive and more commonly-used structure [23], especially during the early stages of sensemaking.

2.4 Recognizing and Using Gestures

The wiggle gesture we use in Wigglite (as shown in Figure 1) has a similar form to a scratch-out gesture in some previous systems used for undo [93], edit [77], or delete. Wiggling has also been used by some window managers, for example, Microsoft Windows 7 in 2009 introduced "Aero Shake" [86] where grabbing the title bar with the mouse and shaking the window left and right minimizes all other windows, or restores them. However, these gestures all require that the mouse button first be depressed, while our approach, on the contrary, is specifically designed to work when with none of the mouse buttons are depressed. In addition, macOS has an accessibility feature that supports shaking the mouse to make the size of the pointer much larger to help locate the pointer [83]. Importantly, our testing shows that those features do not interfere with our browser-based implementation of wiggle-based gestures.
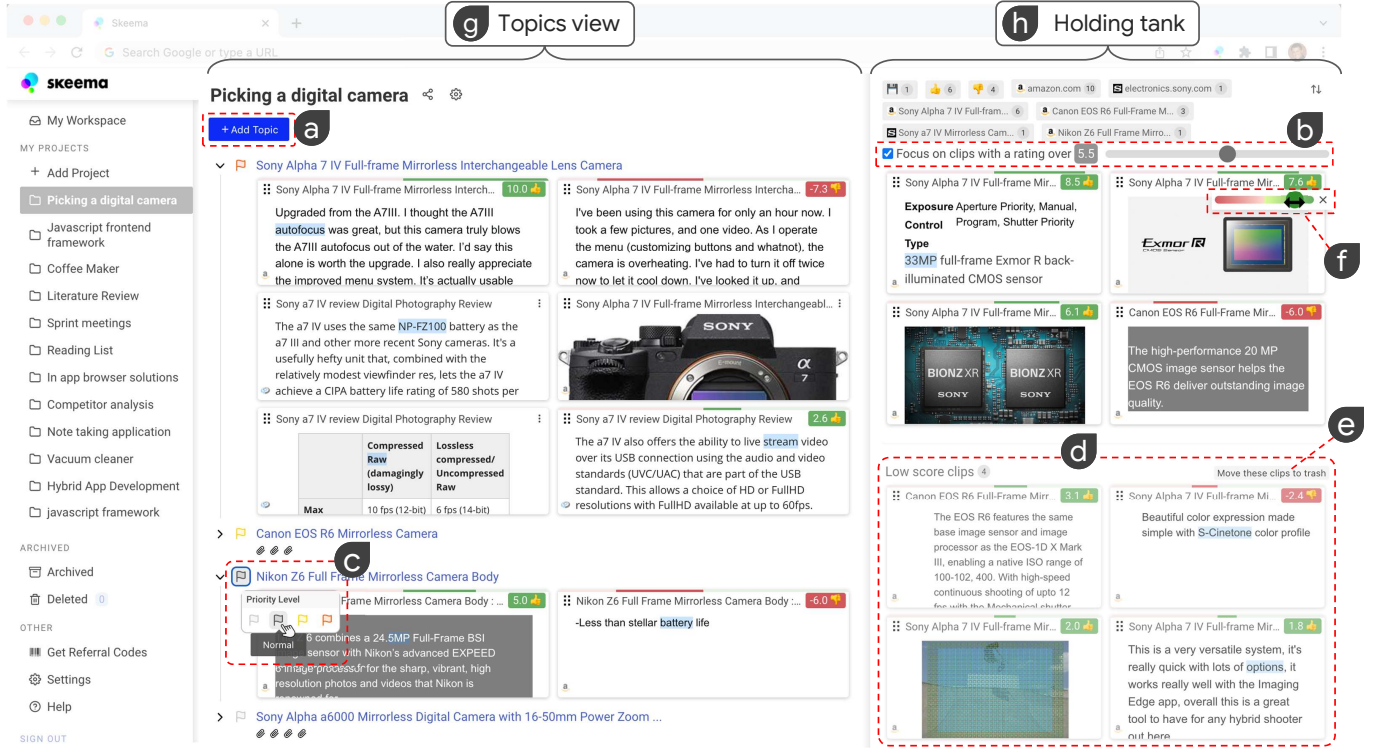
Figure 2: Wigglite's UI built on top of SKEEMA. On the left is the topics view (g) where users can create a topic (a) as well as change its perceived priority (c). On the right is the holding tank (h) that holds the collected information, in which users can filter out information with a lower rating using the slider (b). As a result, clips with rating scores lower than the set threshold would be automatically grouped together at the end and grayed out (d), and users can easily archive or put them in trash by clicking a button (e). In addition, users can manually adjust the valence rating of an information clip (f).

Over the years, many complex gesture *recognizers* have been developed, such as the Rubine recognizer [77], which extracts multiple features from a trajectory and uses a linear classifier for recognition. However, these parametric recognizers are difficult to control with respect to the variances in gestures to be supported. Another approach is template-based gesture recognition, such as the $1 recognizer [94] and the Protractor recognizer [62], which compare new trajectories to the pre-defined gesture templates, and is more lightweight without sacrificing too much accuracy. However, these recognizers can be both time and resource intensive, especially on mobile devices where the computing power and resources are usually limited. In our work, we built a heuristics-based ad-hoc recognizer (see section 4), allowing the system to perform real-time eager recognition [78] without impacting the performance of other UI activities on both desktop and mobile devices. In addition, building on prior evidence that people can accurately perform swipes to as many as eight different directions [12, 54], we support ending the wiggle gesture with a directional swipe to further classify the collected information or encode people's mental context in situ.

3 BACKGROUND AND DESIGN GOALS

In this work, we explore using "wiggling" to select, clip, classify, as well as rate a piece of content in a single gesture during sensemaking, minimizing the interruption to people's main activities of reading and comprehending the content. To ground our research,

we build on an existing information and task management system called SKEEMA. First, we briefly describe SKEEMA and its features related to the context of this work. Then, we discuss the design goals and processes for the wiggling gesture for the new Wigglite system.

3.1 The SKEEMA system

SKEEMA is a Chrome browser extension designed to support people's need to collect and organize information and manage their tabs during online sensemaking. Different from general web clippers that typically only support saving entire pages of web content into an individual note within a notebook [27], SKEEMA enables people to save an arbitrary amount of web content as information clips (Figure 4c) into a holding tank (Figure 2h), and later organize them into topics in the topics view (Figure 2g). For clipping, SKEEMA offers two methods:

- **Clipping text**: Users can select any arbitrary content in the usual way using the cursor and click the clipping button that pops up to collect the selected texts (see the upper part of Figure 6 in the appendix).
- **Clipping screenshot**: Users can use the screenshot feature to drag out a bounding box to save the desired content (see the lower part of Figure 6 in the appendix).

To help users express whether a piece of evidence that they collected is positive or negative with regard to their own goal, SKEEMA allows users to add a valence rating from -10 to +10, with negative

values indicating a "con" and denoted by a "thumbs-down" emoji and positive values indicating a "pro" and denoted by a "thumbs-up" emoji (see Figure 4c1).

Skeema allows users to organize information into thematically related topics in the topics view (Figure 2g). To achieve that, users need to manually create a topic (Figure 2a), enter a name, and drag the desired information cards from the holding tank and drop it into the topic. Users can also set priority to a topic to indicate its perceived utility and how much they want to follow up on it, which defaults to be "Normal", but can also be set to "Low", "High", or "Very high" (Figure 2c).

Although Skeema has the support for collecting finer-grain content (which research has shown to be the unit of information that people usually think in and work with during sensemaking [69, 80]), there is still a high cost in specifying the collection boundary and adding ratings and priorities to the collected information and topics (which users would have to switch to the Skeema tab to do). In addition, clipping text in Skeema loses the text's original CSS styling, which might be helpful for quicker recognition later on [63], and Skeema does not gracefully support collecting consecutive blocks of mixed content (e.g., consumer review text of a camera followed up some sample photos, such as shown in Figure 6b).

3.2 Design Goals for Low-cost Information Capturing and Triaging

Guided by prior work and well as the limitations of Skeema discussed above, we set out to provide an interaction that could simultaneously reduce the cognitive and physical costs of *capturing* information while providing natural extensions to easily and optionally *encode* aspects of users' mental context during sensemaking. We hypothesize that such an effective interaction should have the following characteristics:

(1) **Accuracy**: It needs to be accurate and precise enough to lock onto the content the users intend to collect.
(2) **Efficiency**: It should be quick and low-effort to perform, and minimize interruptions to the main activities that users are performing, such as learning and active reading.
(3) **Expressiveness**: It should be extendable to provide natural and intuitive affordances for users to express aspects of their mental context at the moment. In the scope of this work, we would like to have wiggling support encoding valence ratings as well as topic priorities.
(4) **Integration**: It should be a complement to and not interfere with the existing interactions that users already use, such as using the pointer to select text and pictures or click on links.

Below, we present a brief overview of the iterative design exploration leading to the current wiggle-based interactions.

3.3 Iterative Design Exploration

To begin our exploration, we took a desktop-first approach and brainstormed various interactions that would address these four design goals. To ground our explorations, we also prototyped these candidate interactions using JavaScript in a browser, which is where a large portion of the reading and collecting happens [15, 36]. Like previous approaches, collecting the desired content, including text

and/or images, can be broken down into two main phases: *(a) identifying the desired target* and *(b) triggering the collection.*

One of the interactions we first explored was simply clicking on the desired content (or in the gutter to the left or right) to capture it into the system, similar to existing interactions supported by some text editors such as Microsoft Word. Although straightforward, this interferes with existing selection methods, and would require users to first enter a "grabber" mode, possibly through a special hotkey combination, which violates both design goals (2) and (4). Next, we experimented with hovering the pointer over the target content and keeping it still for a period of time in order to trigger a collection. This has the benefit of not interfering with existing interaction methods as there is no clicking required, satisfying goal (4). However, research has shown that when heavily engaged in active reading and sensemaking tasks, people often need to select and save information frequently within short time intervals [15, 90], and waiting for a noticeable amount of time will add an inherent cost to every collection operation a user wants to perform and therefore is likely to interrupt the user's main activity, violating design goal (2).

Next, we experimented with using non-click gestures (satisfying goal (4)) performed on the desired target to trigger the selection, since gestures are considered intuitive to perform and widely used in both commercial and academic systems [55, 56, 61, 78, 88]. One of the promising ideas was to use the mouse pointer to sketch out a certain shape over the desired target to trigger a collection. In addition, by varying the shape, it could theoretically support encoding different aspects of users' mental model, such as sketching a "+" for marking it as a "pro" and "-" as a "con" [94], supporting design goal (3). However, similar to using keyboard shortcuts, it is hard for users to learn and memorize the different shapes without special affordances [56, 96]. Furthermore, making sure one sketches out the correct gesture may require non-trivial physical as well as cognitive demand, violating goal (2), and even so, these shapes can have a high false recognition rate, violating goal (1).

We then experimented with gestures that do not require special training or practice in order to perform accurately. One that worked particularly well is wiggling the mouse pointer, i.e., making small ballistic back-and-forth movements, on top of the desired collection target (Figure 1a). Here, the choice of a target could be determined from the average or starting location of the mouse pointer during the gesture, and the user continues to perform the same back-and-forth motion until reaching a certain threshold to trigger the collection. Indeed, prior work has suggested that people naturally use the mouse pointer to guide their attention while reading [38], or even unconsciously have the pointer follow their eye gaze [44], so the pointer could be readily available to initiate a wiggle in place.

This has some additional benefits, such as it seemed natural and intuitive like scratching off something [77, 93], it can be activated without clicking, which can be both cognitively and physically costly [52], and is robust against false positives since only a very specific motion pattern could trigger a collection. Furthermore, it can be chained with optional operations such as swiping in different directions that not only are consistent with the wiggling gesture itself but also intuitively map to users' mental context (such as swiping left/right for negative/positive and up/down for various levels of importance, and even leveraging the amount of distance traveled of a swipe to encode a continuous value).

Since there is no mouse pointer on mobile devices such as smartphones, and using fingers to move left and right in browsers triggers page navigation back and forth, whereas up and down is used for scrolling, we decided to take advantage of these small up-and-down scroll events, since they are not currently in use by any existing interactions. Therefore the wiggling counterpart on mobile devices became using the finger to quickly scroll up-and-down while the finger is over the desired collection target (Figure 1d).

4 THE WIGGLITE SYSTEM

4.1 Wiggle-based Gestures

For desktop computers with a traditional mouse, trackpad or trackball input device, the wiggle interaction consists of the following stages, as illustrated in Figure 1a,b,c:

(1) **Acquiring the collection target**: To initiate, users move their mouse pointer onto the target content that they would like to collect (Figure 1a0) and initiate the wiggling movement specified in the steps below. Wigglite uses an always-on wiggle gesture recognizer to automatically detect the start of a wiggling gesture. This avoids the requirement of an explicit signal like a keyboard key or mouse down event, which might conflict with other actions, and has the benefit of combining activating and performing the gesture together into a single step, therefore reducing the starting cost of using the interaction technique.

(2) **Wiggle**: To collect the target content, users simply move the mouse pointer left and right approximately inside the target content. To indicate that the system is looking to detect the wiggling gesture, it will display a small "tail" (e.g., Figure 3c2) that follows the pointer on the screen, and replaces the regular pointer with a special one containing the SKEEMA icon. Wigglite also adds a dotted blue border to the target content to provide feedback about what content will be collected, and the blue color grows in shade as users perform more lateral mouse movements (Figure 1a1-4). This is analogous to half-pressing the shutter button to engage the auto-focus system to lock onto a subject when taking photos with a camera. To assist with collecting fine grain targets, ranging from a word to a block (e.g., a paragraph, an image), Wigglite allows users to vary the average size of their wiggling to indicate the target that they would like to collect: if the average size of the last five lateral movements of a pointer is less than 65 pixels (a threshold empirically tuned that worked well in our pilot testing and user study, but implemented as a customizable parameter that individuals can tune based on their situations), Wigglite will select the word that is covered at the center of the wiggling paths; while larger lateral movements will select a block-level content (details discussed in section 4.3.2). In addition, users can abort the collection process by simply stopping wiggling the mouse pointer before there are sufficient back-and-forth movements.

(3) **Collection**: As soon as users make at least five back and forth motions (optimized for the amount of physical effort required and the number of false positive detection through pilot testing, but is also implemented as a parameter that can be customized by individuals in practice, details discussed in section 4.3.1), the system will commit to the collection, and gives the target a

darker blue background showing that a wiggle has been successfully activated (as shown in Figure 1a5). If users want to collect multiple blocks of content, they can just naturally continue to wiggle over other desired content after this activation. Or, they can stop wiggling. However, if users have selected the wrong target, an undo button appears, which can be clicked to cancel the collection (Figure 3e1).

(4) **Extension**: Instead of just stopping the wiggle motion after collection, users can leverage the last wiggle movement and turn it into a "swipe", either horizontally to the right or left to encode a positive or negative valence rating (as shown in Figure 1b1,b2), or vertically down or up to specify a topic and priority for that topic (as shown in Figure 1c1-4). Feedback for the extension uses different colors for the background of the target content to provide visual salience (details discussed in section 4.2).

Similarly, on a mobile device with touch screens:

(1) **Acquiring collection target**: To initiate, a user's finger touches the target content that should be selected.

(2) **Wiggle**: To collect the target block, the user keeps the finger on the screen and starts making small up-and-down scrolling movements. Similar to the desktop scenario, the system adds a dotted blue border to the target content to provide feedback that the wiggling is being detected (Figure 1d0-4). Note that due to the limitations of the large size of the finger with respect to an individual word [15] as well as the unique use cases of mobile devices (e.g., quickly consuming and collecting blocks of information on the go [46, 91]), Wigglite for mobile only supports selecting block-level content such as paragraphs or images.

(3) **Collection**: As soon as the user makes at least five up-and-down motions, the system will commit to the collection by giving the target a darker blue background (Figure 1d5). Now, the user can stop wiggling and lift the finger from the screen. Similar to the desktop version, an undo button pops up that lets the user cancel the collection in case of an error. Note that due to the limited screen real estate that typical mobile devices afford, additional blocks of content will have to be first scrolled into view for users to then capture them, which would make the interaction less fluid. Therefore, collecting multiple blocks of content is currently not supported by Wigglite on mobile.

(4) **Extension**: Instead of stopping the wiggle motion after collection, users can end the wiggle with a horizontal swipe to the left or right to achieve similar encoding capabilities described for the desktop version. After the system detects the wiggle, it turns off other actions until the finger is lifted, so the swipes do not perform their normal actions. (But the normal swipes, scrolling, and other interactions still work normally when not preceded by a wiggle.) Currently, since Wigglite already uses the vertical dimension for detecting wiggling movement on a mobile device, and large cross-screen vertical movements are difficult to perform, especially when holding and interacting with a single hand, we opted not to make a mobile equivalent of encoding topic priorities.

4.2 An Overview of The Wigglite System

Wigglite enables users to collect and triage web content via wiggling. First of all, after a regular wiggle with no extension (Figure
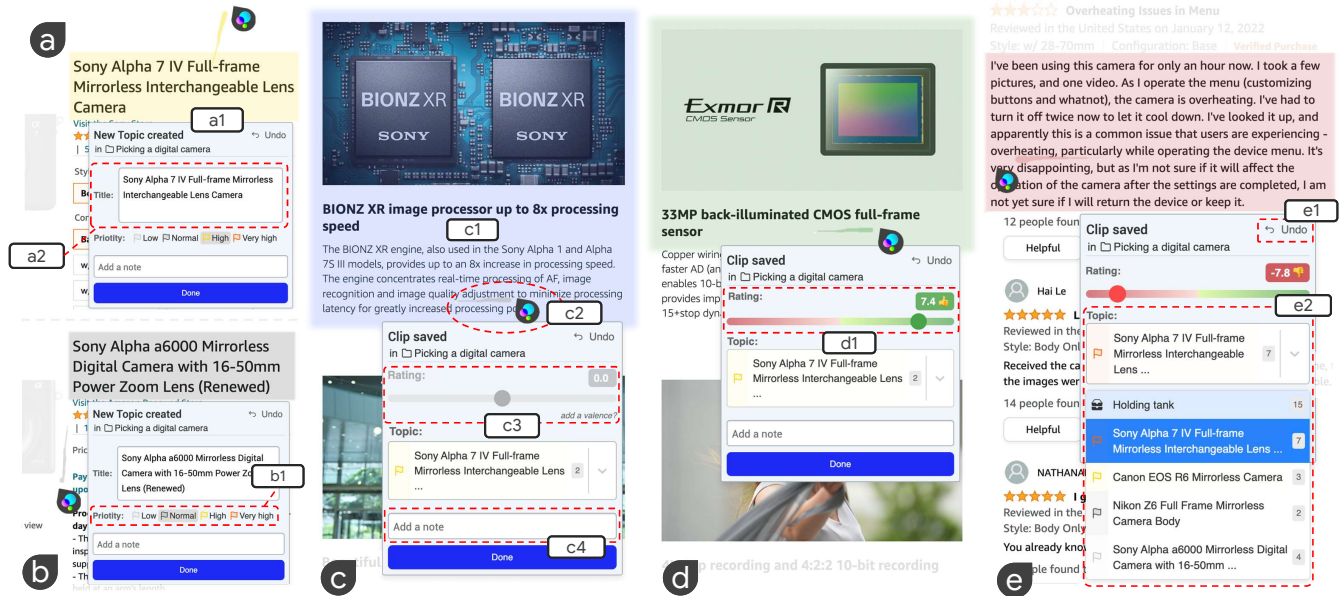
Figure 3: Using wiggling to collect information as well as encode priorities and valence ratings. Specifically, as shown in (c), users can wiggle (c2) over the desired content (c1) to collect it into the information holding tank (Figure 4c). A popup dialog will be presented near the just collected content to allow users to optionally add a valence rating (c3), pick a topic that the content should go into (e2), add notes (c4), as well as undo the collection (e1). In addition to regular collection, users can also end the wiggle with a swipe right to encode a positive rating (d) or left to encode a negative rating (e), which can also be changed in the popup dialog (d1). Furthermore, by ending a wiggle with a swipe up (a) or down (b), users can create a new topic with different priorities (b1), and can change the title of the topic directly in the popup dialog (a1).

3c), Wigglite presents a popup dialog (augmenting the original Skeema popup) directly near the collected content to indicate success. In addition to Skeema's notes field (Figure 3c4), users can attach a valence rating (Figure 3c3) and pick the topic that this piece of information should be organized in (Figure 3e2), as opposed to post-hoc organization using drag and drop as required by Skeema. By default, it goes into the last topic the user picked or the holding tank if none was picked initially. Unlike Skeema where information was saved in pure text format or an inflexible screenshot with limited resolution, Wigglite leverages the technique introduced in [63, 65] to preserve and subsequently show the content with its original CSS styling, including the rich, interactive multimedia objects supported by HTML, like links and images. This makes the content more understandable and useful, and also helps users quickly recognize a particular piece of information among many others by its appearance [63].

Of course, a more fluid way to encode user judgements than what was described above is to leverage the natural extension of the wiggle gesture discussed in the previous section: to encode a valence rating in addition to collecting a piece of content, users can end a wiggle with a horizontal "swipe", either to the right to indicate positive rating (or "pro", characterized by a green-ish color that the background of the target content turns into, and a thumbs-up icon, as shown in Figure 3d), or the left for negative rating (or "con", characterized by a red-ish color that the background of the collected block turns into, and a thumbs-down icon, as shown in Figure 3e). Optionally, users can also turn on real-time visualizations of "how much" they swiped to the left or right to encode a rating score representing the degree of positivity or negativity, and can adjust

that value in the popup dialog (Figure 3d1) or from the information card (Figure 2f). Under the hood, Wigglite calculates this score as the horizontal distance the pointer traveled leftward or rightward from the average wiggle center divided by the available distance the pointer could theoretically travel until it reaches either edge of the browser window. This score is then scaled to be in the range of -10 to 10 to match with the existing values provided by Skeema.

Alternatively, to directly create a topic and encode it with a priority from wiggling, users can either end the wiggle with a swipe up (encoding "high", characterized by a yellow-ish color that the background of the target content turns into, as shown in Figure 3a) or down (encoding "normal", characterized by a gray-ish color that the background of the target content turns into, as shown in Figure 3b). Optionally, if the user swipes all the way up or down to the edge of the browser window, Wigglite will additionally encode two more levels of priorities, "urgent" and "low", indicated by a bright orange and a muted gray color (Figure 3b1), which can be adjusted in the popup dialog (Figure 3b1) as well as in the topics view (Figure 2c). In this case, the content will instead be used as the default *title* of the newly created topic (which users can change in the popup dialog directly as shown in Figure 3a2 or later in the topics view).

To help users better manage the information that they have gathered in the holding tank, Wigglite offers several additional features on top of the original Skeema system. First, it enables users to sort the information cards by various criteria, such as in the order of valence ratings or in temporal order (Figure 4b). Second, it offers category filters (Figure 4a) automatically generated based on the encodings that users provided using wiggling (or edited later) and the provenance of information (where it was captured from). Users can

Figure 4: Wigglite's information holding tank shown both on desktop and on mobile, which houses content that users collected through wiggling in the form of information cards (c). In addition, on desktop, users can apply different filters (a) and sorting mechanisms (b) to the information cards.

quickly toggle those on or off to filter the collected information. For example, in Figure 4b, the information with a "positive rating" or "negative rating" and collected from "amazon.com" was filtered and shown, as indicated by the dark gray background of the corresponding filters (if none of the filters are enabled, all the information cards will be shown). Third, users can quickly filter out information with a lower rating (e.g., indicating that it was less impactful to a user's overall goal and decision making) by adjusting the threshold using the "Focus on clips with a rating over `threshold`" slider shown in Figure 2f. As a result, clips with rating scores lower than the set threshold would be automatically grouped together at the end and grayed out (Figure 2d), and users can easily archive or put them into the trash in a batch by clicking the "Move these clips to trash" button (Figure 2e). These organizational features further help users reduce clutter in the holding tank, and provide a scaffold for them to start dragging and dropping clips into their respective topics.

Due to the limited screen size and use cases of a mobile device, we chose to only let users view the clips along with their valence in the holding tank (Figure 4c).

4.3 Design and Implementation Considerations

In this section, we discuss important design and implementation considerations made through prototyping Wigglite with JavaScript in a browser to achieve the design goals specified in section 3.2.

4.3.1 Recognizing a wiggle gesture. For accurately recognizing the wiggle pattern, we explored several options. One way is to use an off-the-shelf gesture recognizer such as the $1 [94] or the Protractor [62] recognizer. Although these recognizers may be lightweight and easy to customize, they are fundamentally designed to recognize distinguishable shapes such as circles, arrows, or stars, while the path of our wiggle gesture does not conform to a particular shape that is easily recognizable (and we argue that it should not conform to any particular shape, the sketching of which would increase the cognitive and physical demand). A second option we investigated was to build a custom computer vision based wiggle recognizer using transfer learning from lightweight image classification models such as MobileNets [42]. Though these ML-based models improved the recognition accuracy in our internal testing, they incurred a

noticeable amount of delay due to browser resource limitations (and limitations in network communication speed when hosted remotely). This made it difficult for the system to perform eager recognition [78] (recognizing the gesture as soon as it is unambiguous rather than waiting for the mouse to stop moving), which is needed to provide real-time feedback to the user on their progress.

To address these issues, we discovered that a common pattern in all of the wiggle paths that users generated with a mouse or trackpad during pilot testing share the characteristic that there were at least five (hence the activation threshold mentioned in section 4.1) distinguishable back and forth motions in the horizontal direction, but inconsistent vertical direction movements. Similarly, on smartphones, wiggling using a finger triggers at least five consecutive up and down scroll movements in the vertical direction but inconsistent horizontal direction movements. Therefore, we hypothesized that only leveraging motion data in the principle dimension (horizontal on desktop, and vertical on mobile) would be sufficient for a custom-built recognizer to differentiate intentional wiggles from other kinds of motions by a cursor or finger.

Based on our implementation using JavaScript in the browser, we found that it successfully supports real-time eager recognition with no noticeable impact on any other activities that a user performs in a browser. Specifically, the system starts logging all mouse movement coordinates (or scroll movement coordinates on mobile devices) as soon as any mouse (or scroll) movement is detected, but still passes the movement events through to the rest of the DOM tree elements so that regular behavior would still work in case there is no wiggle. In the meantime, the system checks to see if the number of reversal of directions in the movement data in the principle direction exceeds the activation threshold, in which case a "wiggle" will be registered by the system. After activation, the system will additionally look for a possible subsequent wide horizontal or vertical swipe movement (for creating topics with priority or encoding valence to the collected information) without passing those events through to avoid unintentional interactions with other UI elements on the screen. As soon as the mouse stops moving, or the user aborts the wiggle motion before reaching the activation threshold, the system will clear the tracking data to prepare for the next possible wiggle event.

4.3.2 Target Acquisition. In order to correctly lock onto the desired content without ambiguity, we explored two approaches that we applied in concert in Wigglite. The first approach is to constrain the system to only be able to select certain targets that are usually large enough to contain a wiggling path and semantically complete. For example, one could limit the system to only engage wiggle collections on block-level semantic elements [1], such as `<div>`, `<p>`, `<h1>`–`<h6>`, `<li>`, `<img>`, `<table>`, etc. This way, the system will ignore inline elements that are usually nested within or between a block-level element. This approach, though sufficient in a prototype application, does rely on website authors to organize content with semantically appropriate HTML tags.

The second approach is to introduce a lightweight disambiguation algorithm that detects the target from the mouse pointer's motion data in case the previous one did not work, especially for a small `<span>` or an individual word. To achieve this, we chose to take advantage of the pointer path coordinates (both X and Y)

in the last five lateral mouse pointer movements, and choose the target content covered by the most points on the path. Specifically, we used the same re-sampling and linear interpolation technique introduced in the $1 gesture recognizer [94] to sample the points on a wiggle path to mitigate variances caused by different pointer movement speeds as well as the frequency at which a browser dispatches mouse movement events.

On mobile devices, since the vertical wiggling gesture triggers the browser's scrolling events, the target moves with and stays underneath the finger at all times. Therefore, we simply find the target under the initial touch position.

When Wigglite is unable to find a target (e.g., when there is no HTML element underneath where the mouse pointer or the finger resides) using the methods described above, it does not trigger a wiggle activation (and also not the aforementioned set of visualizations), even if a "wiggle action" was detected. This was an intentional design choice to further avoid false positives as well as to minimize the chances of causing distractions to the user.

4.3.3 Integration with existing interactions. Notice that the wiggling interaction does not interfere with common active reading interactions, such as moving the mouse pointer around to guide attention, regular vertical scrolling or horizontal swiping (which are mapped to backward and forward actions in both Android and iOS browsers) [70, 85]. In addition, wiggling can co-exist with conventional precise content selection that are initiated with mouse clicks or press-and-drag-and-release on desktops or long taps or edge taps on mobile devices [20, 76]. Furthermore, unlike prior work that leverages pressure-sensitive touch screens to activate a special selection mode [15], the wiggling interaction does not require special hardware support, and can work with any kind of pointing device or touch screen.

4.4 Implementation Notes

We implemented the wiggling technique as an event-driven Java-Script library that can be easily integrated into any website and browser extension. Once imported, the library will dispatch `wiggle`-related events once it detects them. Developers can then subscribe to these events in the applications that they are developing. All the styles mentioned above are designed to be easily adjusted through predefined CSS classes. The library itself is written in approximately 1,100 lines of JavaScript and TypeScript code.

The Wigglite browser extension is implemented in HTML, Type-Script, and CSS and uses the React JavaScript library [28] for building UI components. It uses Google Firebase for backend functions, database, and user authentication. In addition, the extension is implemented using the now standardized Web Extensions APIs [71] so that it would work on all major browsers, including Google Chrome, Microsoft Edge, Mozilla Firefox, Apple Safari, etc. However, we primarily targeted Google Chrome and Microsoft Edge to minimize testing efforts during development.

The Wigglite mobile application is implemented using the Angular JavaScript library [35], the Ionic Framework [45] and works on both iOS and Android operating systems. Due to the limitations that none of the current major mobile browsers have the necessary support for developing extensions, Wigglite implements its own browser using the `InAppBrowser` plugin from the open-source

Apache Cordova platform [2] to inject into webpages the JavaScript library that implements wiggling as well as custom JavaScript code for logging and communicating with the Firebase backend.

5 USER EVALUATION

We conducted an initial lab study to evaluate the usability and usefulness of Wigglite in helping people collect information as well as encode aspects of their mental context while doing so. Specifically, we aimed to address the following research questions:

- **RQ1 [Accuracy]**: Are wiggle-based interactions sufficiently accurate to help users collect what they want?
- **RQ2 [Efficiency]**: Are wiggle-based interactions sufficiently low-friction to perform without interrupting the primary reading and sensemaking activities?
- **RQ3 [Expressiveness]**: Are the proposed extensions of marking priorities and valence useful in helping people encode their mental contexts?
- **RQ4 [Integration]**: Do wiggle-based interactions interfere with existing interactions that people are already using?

5.1 Participants

We recruited 12 participants (6 male, 6 female; 3 students, 3 software engineers, 2 UX designers, 1 UX researcher, 1 medical doctor, 1 administrative staff member, and 1 entrepreneur) aged 21-38 years old (mean age = 28.5, *SD* = 4.5) through emails and social media. Participants were required to be 18 or older and fluent in English. All participants reported experience reading and making sense of large amounts of information online for either professional or personal purposes on a daily basis, and had tried or were using commercially available web clipping and organization tools and systems, such as the Evernote Clipper, OneNote, or Notion.

5.2 Study Methodology

The study was a within-subjects design with each participant engaging in two tasks, one using Wigglite with SKEEMA in the experimental condition, and the other just using SKEEMA in the control condition, counterbalanced for order. For our control condition, SKEEMA provided the affordances of a web clipping tool, which would provide a more conservative and matched baseline than no tool support. Specifically, our control condition enabled participants to capture text through a popup button (Figure 6a1) to save highlighted text and a screenshot clipper instead of the wiggle interaction. After saving the information, participants could set the priority of topics and the valence of information in the workspace view (Figure 2), versus being able to encode them as a continuation of the wiggle in Wigglite.

For each task, participants were presented with a product category they needed to research, and a set of three Amazon pages from which they were required to collect information. Participants were instructed to read through the provided webpages, collect information, and organize the information clips into topics, such as by different options or different criteria in which the options should be evaluated. They were required to at least collect 10 information clips as well as create a minimum of 3 topics with priority for each task. Participants had 15 minutes to complete the task, but could inform the experimenter to move on if they finished early.

The two tasks were:

- *(A)* Choosing a digital mirrorless camera: participants were told to imagine that they were to purchase a new mirrorless camera to take photos of their spouse and young kids on their weekend road trips.
- *(B)* Buying a vacuum cleaner: participants were told to imagine that they were to buy a new vacuum cleaner in preparation for moving into a new house with a newborn baby and their two pets.

In order to minimize differences between tasks and participant decision making, we provided a fixed set of web pages per task, each with approximately eight screens of content. As described in the results, the two tasks took approximately the same amount of time for participants to finish, and were counterbalanced in order and randomized across conditions.

Each study session started by obtaining consent and having participants fill out a demographic survey. Participants were then given a 10-minute guided tutorial showcasing the various features of Wigglite as well as the baseline system, and a 10-minute free-form practice session to familiarize themselves with the features of both systems. At the end of the study, participants completed a survey and engaged in a semi-structured interview about their experience with the tool. The interview focused on participants' perceptions of using the wiggle-based interactions. The questions probed the perceived effectiveness of wiggling, their current practices around collecting information, and scenarios where they thought wiggling would be useful and how they would modify it to be more useful. The interviews were audio-recorded and transcribed, after which qualitative coding and thematic analysis [19] were performed.

Each study session took approximately 60 minutes to complete, using a designated MacBook Pro computer with Google Chrome and Wigglite installed as well as a Logitech MX Master 2S mouse. All sessions were carried out in person, with the participants and the experimenter appropriately masked following COVID-19 mitigation guidelines. All participants were compensated $15 for their time. The study was approved by our institution's IRB.

6 RESULTS

All participants were able to complete each task within the specified 15 minute time limit. Below, we compile together both quantitative and qualitative evidence to evaluate Wigglite with respect to our four design goals and research questions.

6.1 RQ1 [Accuracy]

First, evaluate if the wiggling gestures are accurate enough to help users collect and express what they want. Specifically, we looked for cases where: (1) participants hit the undo button to dismiss an incorrect wiggle activation and redo the wiggling due to Wigglite picking up the wrong target content, which turned out to be on average 0.67 (SD = 0.65) times per person per task, and only accounted for 1.48% of the 45.16 (SD = 8.82) total wiggle actions participants on average performed per task; (2) participants had to use the popup dialog to immediately edit the valence or topic priority because Wigglite picked the wrong swipe direction, which turned out to be 0; (3) participants had to redo the wiggling gesture because the previous one they performed did not activate at all,

which turned out to be on average 0.92 (SD = 0.67) times per person per task, and only accounted for 2.01% of the total wiggle actions participants on average per task.

This evidence suggests that the wiggling technique provided by the current Wigglite system is sufficiently accurate and robust, at least with ample amount of training and practice. It would be interesting for future work to explore how it performs in the wild, potentially without much upfront practice, and examine whether and how people's wiggling accuracy and performance evolve over time.

6.2 RQ2 [Efficiency]

Second, we are interested in understanding if Wigglite creates a more fluid experience when collecting and triaging information with less interruption compared to the baseline condition. For this comparison, we opted to measure two key metrics: the *overhead cost* of using a tool to collect and triage information, and the total amount of *time* it took for participants to finish each task. For the Wigglite condition, we calculate the overhead cost as the portion of the time participants spent on directly interacting with Wigglite (performing wiggling gestures, interacting with the popup dialog if necessary, filtering the information clips, organizing them in the workspace view, etc.) out of the total time they used for a task (vs. reading and comprehending the web pages) [63, 65]. Similarly, in the baseline condition, the overhead cost accounts for situations where participants use the highlighting or screenshot feature to collect information, organize them in the workspace view, etc.

We conducted a two-way repeated measures ANOVA to examine the within-subject effects of condition (Wigglite vs. baseline) and task (A vs. B) on overhead cost. There was a statistically significant effect of condition ($F(1, 20) = 40.7$, $p < 0.001$) such that the overhead cost was significantly lower (58% lower, as shown in Table 1 and Figure 5a) in the Wigglite condition (Mean = 13.4%, SD = 0.06) than in the baseline condition (Mean = 31.7%, SD = 0.08). There was no significant effect of task ($F(1, 20) = 0.46$, $p = 0.51$). In addition, a two-way repeated measures ANOVA was conducted to examine the within-subject effects of condition (Wigglite vs. baseline) and task (A vs. B) on task completion time. There was a statistically significant effect of condition ($F(1, 20) = 20.8$, $p < 0.001$) such that participants completed tasks significantly faster (23.6% faster, as shown in Table 1 and Figure 5b) with Wigglite (Mean = 536.8 seconds, SD = 67.0 seconds) than in the baseline condition (Mean = 702.8 seconds, SD = 102.9 seconds). Again, there was no significant effect of task ($F(1, 20) = 0.77$, $p = 0.38$).

As the condition had a statistically significant impact on both the overhead cost as well as the task completion time (with faster completion and lower overhead cost in Wigglite conditions), Wigglite indeed helped participants reduce the overhead costs of collecting and triaging information and speed up their sensemaking process overall, even though the majority of their time was necessarily spent reading and understanding the material in both conditions.

Furthermore, in the post-study interview, participants overall appreciated the increased efficiency afforded by Wigglite, especially using the wiggling gestures. Many (9/12) mentioned that the perceived workload to collect information that they have encountered was minimal, saying that "*It felt like I didn't do anything to get those snippets into the system*" (P3), and was fluid enough that it did not

	Condition	Overhead cost	Time (seconds)	*n* of clips collected	*n* of topics created using wiggling	*n* of topics created separately in the workspace view	Total *n* of topics created
Task A	Baseline	33.0% (8.60%)	713.7 (76.0)	21.0 (7.03)	N/A	4.17 (1.17)	4.17 (1.17)
	Wigglite	14.0% (7.89%)	558.7 (76.5)	38.3 (5.28)	7.50 (1.05)	0.50 (0.84)	8.00 (1.89)
Task B	Baseline	30.40% (7.31%)	692.0 (131.4)	19.5 (6.81)	N/A	4.67 (0.52)	4.67 (0.52)
	Wigglite	12.8% (2.74%)	515.7 (54.3)	37.3 (8.64)	7.17 (1.17)	0.50 (1.22)	7.67 (2.39)
Average	Baseline	31.7% (7.73%)	702.8 (102.9)	20.3 (6.68)	N/A	4.42 (0.90)	4.42 (0.90)
	Wigglite	13.4% (5.67%)	536.8 (67.0)	37.8 (6.85)	7.33 (1.07)	0.50 (1.00)	7.83 (2.07)

Table 1: Statistics for various performance measures in the user study. Standard deviations are included in the parentheses.
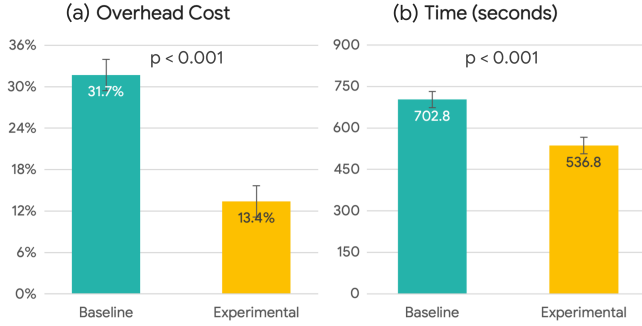

Figure 5: Using Wigglite incurred significantly less overhead cost (a) and helped participants finished the tasks significantly faster (b) when compared to the baseline condition in the user study.

interrupt their flow of reading the task pages, such as "*I just wiggle and move on, in fact, when I am wiggling on something, my eyes are already onto the next paragraph, no more stopping to do the regular clipping thing any more*" (P11). Together with the quantitative evidence above, Wigglite did offer a more fluid experience when collecting and rating information with less interruption.

6.3 RQ3 [Expressiveness]

Third, we are also interested to know to what extent Wigglite induces changes in people's behavior, especially given the natural extension that wiggling affords to encode priorities and valence.

As shown in Table 1, participants collected significantly more information using wiggling (on average 37.8 clips, SD = 6.85) than when using the conventional selecting or screenshot workflow (on average 20.3 clips, SD = 6.68) ($p < 0.01$), despite spending less time on the tasks. Among the collected information clips using wiggling, 75.3% of them were encoded with either a positive or negative valence. Similarly, participants created significantly more topics using Wigglite (on average 7.83 topics, SD = 2.07) than in the control condition (on average 4.42 topics, SD = 0.90) ($p < 0.01$), where topics were required to be created separately in the workspace view. It is also worth noting that using wiggling to create topics (7.33 times, SD = 1.07) almost eliminated the need to separately (0.50 times, SD = 1.00) create topics (granted that most participants did at least edit the title of the topics in the popup dialog or in the workspace view to make them more succinct and easier to read).

This evidence suggests that participants indeed were able to use Wigglite to externalize the perceived utility of a particular piece of information as well as their mental judgements of how it aligned with their goals in situ.

Furthermore, in the post-study interviews, some (4/12) participants reflected that Wigglite would enable them to express their perceived utility in a way that is also useful for subsequent sorting and ranking. For example, P5 mentioned that "*I really enjoyed the threading [creating topics with priorities] feature, being able to say something is important or extra important on the spot would help me stay on top of my todo list.*" However, perhaps due to the limited scale of the lab study, we did not observe significant differences in the types of information participants used as topics–most of them are about the different options as well as some criteria to evaluate a product. Future and potentially larger-scale investigations are required to understand the types of information users collect using a lightweight gesture like wiggling versus using conventional capturing methods.

6.4 RQ4 [Integration]

Last but not least, we would like to understand if the wiggle gesture would interfere with participants' normal behaviors during web browsing, such as unconsciously using the mouse pointer to guide their attention [44], clicking [37], or scrolling (false positives). To measure this, we looked for cases where participants hit the undo button to dismiss a wiggle activation due to Wigglite had wrongly recognized some regular mouse movements as a wiggle, which turned out to be 0 across the board. This provides evidence that the wiggling gestures added by Wigglite do not interfere with the existing interactions and user behaviors.

6.5 Other Subjective Feedback

In the survey, participants reported (in 7-point Likert scales) that they thought the interactions with Wigglite were understandable and clear (Mean = 6.25, SD = 0.45), Wigglite was easy to learn (Mean = 6.42, SD = 0.67), and they enjoyed Wigglite's features (Mean = 6.25, SD = 0.62). In addition, compared to the baseline condition (Mean = 5.75, SD = 0.62), they thought using Wigglite (Mean = 6.17, SD = 0.39) would help make their information collection and triaging processes more efficient and effectively ($p = 0.017$), and would recommend Wigglite (Mean = 6.33, SD = 0.49) over the baseline version of Wigglite (Mean = 5.92, SD = 0.29) to friends and colleagues ($p = 0.007$), both differences were statistically significant under paired t-tests. Details of the survey questions and scores are presented in Table 2.

In addition, some participants reflected on the playfulness and attractiveness of the wiggle interactions and how it encouraged them to collect information compared to what they normally have to go through. For example, P8 said: "*It's fun, you know? I didn't quite believe it at the beginning, but it actually made grabbing stuff so much fun*", and P1 suggested that "*somehow with this, I don't think going through something that I'm not familiar with would be*

Question Statements	Wigglite condition	baseline condition
I would consider my interactions with the tool to be understandable and clear.	6.25 (0.45)	6.17 (0.72)
I would consider it easy for me to learn how to use this tool.	6.42 (0.67)	6.33 (0.49)
I enjoyed the features provided by the tool.	6.25 (0.62)	6.08 (0.67)
Using this tool would help make my information collection and triaging processes more efficient and effective.	6.17 (0.39)*	5.75 (0.62)*
If possible, I would recommend the tool to my friends and colleagues.	6.33 (0.49)*	5.83 (0.39)*

Table 2: Statistics of scores in the post-tasks survey. Participants were asked to rate their agreement with statements related to their experience interacting with Wigglite and the baseline on a 7-point Likert scale from "Strongly Disagree" (a score of 1) to "Strongly Agree" (a score of 7). Statistics in column 2 and 3 are presented in the form of mean (standard deviation). Statistically significant differences ($p < 0.05$) through paired t-tests are marked with an *. The survey questions and scales were adapted from a validated SUS scale [58].

as daunting as it used to be". Four of the participants even went on to ask when Wigglite will be released publicly so that they could use it for their own work and personal tasks, and wondered if they could customize the system, such as by "*writing some sort of plugin, like the one I wrote for Obsidian [72], to map the different directional swipes to what I want depending on the situations that I'm in*" (P11).

7 LIMITATIONS

One potential limitation to wiggling is the suitability of its rapid back-and-forth movements to user populations with motor impairments or advanced age, for example, users with hand tremors. There are several ways in which wiggling might be more suitable than expected or relatively easily adapted to such populations. First, since wiggling uses the initial mouse location as its selection anchor, a user can take their time adjusting to arrive at the correct area (which would still require less accuracy than traditional highlighting). Once there, they could initiate selection without clicking, which could address mouse slip while clicking, a common problem with advanced age or motor impairment [29, 30, 87]. If issues with tremor lead to lower accuracy, one approach that might be investigated is smoothing mouse movement using generative models trained on a user's individual behavior (e.g., [92]). More generally, additional research is needed to understand the suitability of wiggling across a variety of user capabilities, contexts, and devices [59, 60].

Our lab study had several limitations. Given the short amount of training and practice time, some participants might not have been able to fully familiarize themselves with the wiggle-based collection and triaging techniques offered by Wigglite. The study tasks and topics might not be the ones that participants typically encounter, and therefore they may not have sufficient motivation or background context as in real life. However, we attempted to mitigate these risks through carefully preparing the study setup: (1) we chose the training and real study tasks based on actual product comparison topics that people are faced with; (2) we had participants practice using Wigglite as well as its baseline version for each condition simulating what they needed to do for the real tasks, and (3) we provided participants with ample amount of background information to help them get prepared. We would like to further address these limitations in the future by having participants use Wigglite for their own work and personal tasks and projects, which would presumably fuel them with the necessary motivation and context and engage with Wigglite in a more organic way.

While sensemaking in various domains might exhibit different characteristics and therefore lead to different information foraging behavior patterns, we chose both the study tasks to be in the domain of comparison shopping to at least make sure that the tasks are roughly of equal difficulty. In addition, product comparison shopping embodies many of the common sensemaking properties and needs that people have, for example, it is information dense so that users would potentially have to read and process lots of information and collect quite a few items, and users would often have to interpret the information based on their own goals and context, so that there is a need for them to externalize their mental context alongside the collected information. Nevertheless, we would like to address this limitation by evaluating Wigglite in a variety of domains where sensemaking usually occurs, such as students conducting literature reviews, patients researching medical diagnoses, and programmers learning unfamiliar APIs.

There was also a risk of participants already being familiar with a topic, such as an expert photographer doing task (A). However, in the post-study interviews, we confirmed that none reported having extensive experience or expertise in any of the task topics.

Finally, due to the limited set of capabilities of the Wigglite mobile application and the similarity of features with its desktop counterpart, we did not evaluate the mobile app in our lab study, and therefore could not directly compare the wiggle-based collection and triaging techniques for sensemaking to a baseline. Informally in our pilot testing, using wiggling to collect information and optionally encode a positive or negative valence was much faster and more convenient than any common information capturing methods that people currently use on mobile devices, such as copying and pasting text and taking screenshots or photos [84]. Nevertheless, we would like to evaluate the wiggle-based techniques and Wigglite for mobile in a formal lab study in the future.

8 FUTURE WORK

An essential goal of this work is to explore ways to enable people to focus on reading and comprehending actual content rather than splitting attention on the mechanics of collecting information as well as externalizing their mental context. However, prior research [5, 22, 50] has suggested that there is a higher likelihood for people to recall and trust the information if they consciously spend time collecting and synthesizing it with the existing information. This raises an interesting tension and trade-off between pursuing low-cost interactions for information capturing and triaging

versus consciously collecting and synthesizing the encountered information — future research would be required to examine the long-term effect of using lightweight systems like Wigglite on people's learning outcomes as well as decision results in various kinds of sensemaking scenarios.

Research on activity-based computing [10, 24] has suggested the benefits of granting users access to their information repository as well as the ability to perform tasks across multiple devices. While the current implementation of Wigglite mobile application does enable users to collect information and triage it with valence as well as to review their collected information, extending it to support more complex operations such as creating and curating topics could be an interesting direction for future work. We anticipate two challenges: (1) how to reasonably leverage the limited screen real estate to design user interfaces and interactions that feel native to a mobile device while also being functionally lightweight and intuitive enough to perform, and (2) exploring the realistic role of mobile devices in the sensemaking ecosystem [46, 91], i.e., striking a balance between pursuing feature parity with the Wigglite desktop counterpart and designing to specifically support practical use cases (e.g., reviewing and triaging information during a commute).

Though we extended the wiggle gesture to support encoding the valence and priority of information, we envision a larger design space where different aspects and properties of the wiggling movement could be mapped to various functions to increase its expressiveness and utility. For example, the *speed* of each movement, the *duration* of the total movement, or the *size* of the gesture (currently mapped to target size selection) are all continuous measures that may intuitively map to various qualities that could be used in interactions, such as uncertainty or confidence towards some content. Building on what we mentioned in section 6.5 and 7, future work could investigate: (1) ways to support customizing the mapping of the different aspects of a wiggle according to users' preferences and sensemaking scenarios, and (2) intelligently learning and adapting to users' needs and habits over time, such as re-calibrating the recognition software to account for individuals' cognitive and physical conditions [87].

Last but not least, we envision a future where the wiggling technique as a new class of interaction can be extended and applied to tasks and applications other than sensemaking. On the one hand, a consistent application scenario envisioned by participants involved using wiggling for repeated selection, extraction, and annotation of various types of data as part of a data processing task, such as aggregating recipes online before going shopping, or saving specific torque numbers of fasteners while working on a car. On the other hand, wiggling might also be used for many other system-level behaviors as it does not conflict with most traditional selections or gestures. For example, wiggling with popup or cross through menus could be applied for quickly modifying device settings on the fly, such as screen zoom or brightness. Additionally, wiggling could serve as an initial activation for a much more expressive set of gestures, or even summon an intelligent agent (such as a voice assistant) that can perform a complex action on the specified item.

9 CONCLUSION

This work explored a new interaction technique called "wiggling," which can be used to fluidly collect, organize, and rate information during early sensemaking stages with a single gesture. Nowadays, people face the challenge of capturing the information they find for later use as well as externalizing their mental judgement about its valence and priority during many online learning and exploration tasks, such as conducting comparison shopping, researching medical diagnoses, and searching for code snippets. While current select-copy-paste-based approaches incur a high cost when doing so, the wiggling technique afforded by the Wigglite system discussed in this paper only involves light-weight back-and-forth movements of a pointer or up-and-down scrolling on a smartphone, which can indicate the information to be collected and its valence in a single gesture, and does not interfere with other available system interactions. In addition, we envision an extensive design space where the wiggling interaction can be adapted to support a wide range of custom actions beyond sensemaking in the future.

ACKNOWLEDGMENTS

REFERENCES

[1] [n.d.]. Block-level elements - HTML: HyperText Markup Language | MDN. https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements

[2] Apache. [n.d.]. Apache Cordova. https://cordova.apache.org/

[3] Anne Aula, Natalie Jhaveri, and Mika Käki. 2005. Information search and re-access strategies of experienced web users. In *Proceedings of the 14th international conference on World Wide Web (WWW '05)*. Association for Computing Machinery, New York, NY, USA, 583–592. https://doi.org/10.1145/1060745.1060831

[4] Michelle Q. Wang Baldonado and Terry Winograd. 1997. SenseMaker: An Information-exploration Interface Supporting the Contextual Evolution of a User's Interests. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '97)*. ACM, New York, NY, USA, 11–18. https://doi.org/10.1145/258549.258563

[5] Marcia J Bates. 1989. The design of browsing and berrypicking techniques for the online search interface. *Online review* (1989). Publisher: MCB UP Ltd.

[6] David Bawden, Clive Holtham, and Nigel Courtney. 1999. Perspectives on information overload. *Aslib Proceedings* 51, 8 (Jan. 1999), 249–255. https://doi.org/10.1108/EUM0000000006984 Publisher: MCB UP Ltd.

[7] Eric A. Bier, Edward W. Ishak, and Ed Chi. 2006. Entity quick click: rapid text copying based on automatic entity extraction. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. Association for Computing Machinery, New York, NY, USA, 562–567. https://doi.org/10.1145/1125451.1125570

[8] Horatiu Bota, Paul N. Bennett, Ahmed Hassan Awadallah, and Susan T. Dumais. 2017. Self-Es: The Role of Emails-to-Self in Personal Information Management. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval (CHIIR '17)*. Association for Computing Machinery, New York, NY, USA, 205–214. https://doi.org/10.1145/3020165.3020189

[9] Joel Brandt, Philip J. Guo, Joel Lewenstein, Mira Dontcheva, and Scott R. Klemmer. 2009. Two Studies of Opportunistic Programming: Interleaving Web Foraging, Learning, and Writing Code. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1589–1598.

https://doi.org/10.1145/1518701.1518944 event-place: Boston, MA, USA.

[10] Frederik Brudy, Christian Holz, Roman Rädle, Chi-Jui Wu, Steven Houben, Clemens Nylandsted Klokmose, and Nicolai Marquardt. 2019. Cross-Device Taxonomy: Survey, Opportunities and Challenges of Interactions Spanning Across Multiple Devices. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–28. https://doi.org/10.1145/3290605.3300792

[11] Michael D. Byrne, Bonnie E. John, Neil S. Wehrle, and David C. Crow. 1999. The tangled Web we wove: a taskonomy of WWW use. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '99)*. Association for Computing Machinery, New York, NY, USA, 544–551. https://doi.org/10.1145/302979.303154

[12] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. 1988. An empirical comparison of pie vs. linear menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '88)*. Association for Computing Machinery, New York, NY, USA, 95–100. https://doi.org/10.1145/57167.57182

[13] Stuart K. Card, George G. Robertson, and William York. 1996. The WebBook and the Web Forager: Video Use Scenarios for a World-Wide Web Information Workspace. In *Conference Companion on Human Factors in Computing Systems (CHI '96)*. ACM, New York, NY, USA, 416–417. https://doi.org/10.1145/257089.257407

[14] Nicholas J. Cepeda, Harold Pashler, Edward Vul, John T. Wixted, and Doug Rohrer. 2006. Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological Bulletin* 132, 3 (May 2006), 354–380. https://doi.org/10.1037/0033-2909.132.3.354

[15] Joseph Chee Chang, Nathan Hahn, and Aniket Kittur. 2016. Supporting Mobile Sensemaking Through Intentionally Uncertain Highlighting. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 61–68. https://doi.org/10.1145/2984511.2984538

[16] Joseph Chee Chang, Nathan Hahn, and Aniket Kittur. 2020. Mesh: Scaffolding Comparison Tables for Online Decision Making. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 391–405. https://doi.org/10.1145/3379337.3415865

[17] Joseph Chee Chang, Nathan Hahn, Adam Perer, and Aniket Kittur. 2019. SearchLens: composing and capturing complex user interests for exploratory search. In *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19)*. Association for Computing Machinery, Marina del Ray, California, 498–509. https://doi.org/10.1145/3301275.3302321

[18] Joseph Chee Chang, Yongsung Kim, Victor Miller, Michael Xieyang Liu, Brad A Myers, and Aniket Kittur. 2021. Tabs.do: Task-Centric Browser Tab Management. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 663–676. https://doi.org/10.1145/3472749.3474777

[19] Kathy Charmaz. 2006. *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis*. SAGE. Google-Books-ID: 2ThdBAAAQBAJ.

[20] Chen Chen, Simon T. Perrault, Shengdong Zhao, and Wei Tsang Ooi. 2014. BezelCopy: an efficient cross-application copy-paste technique for touchscreen smartphones. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*. Association for Computing Machinery, New York, NY, USA, 185–192. https://doi.org/10.1145/2598153.2598162

[21] Mark Claypool, Phong Le, Makoto Wased, and David Brown. 2001. Implicit interest indicators. In *Proceedings of the 6th international conference on Intelligent user interfaces (IUI '01)*. Association for Computing Machinery, New York, NY, USA, 33–40. https://doi.org/10.1145/359784.359836

[22] National Research Council and others. 2000. *How people learn: Brain, mind, experience, and school: Expanded edition*. National Academies Press.

[23] Anita Crescenzi, Yuan Li, Yinglong Zhang, and Rob Capra. 2019. Towards Better Support for Exploratory Search through an Investigation of Notes-to-self and Notes-to-share. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 1093–1096. https://doi.org/10.1145/3331184.3331309

[24] David Dearman and Jeffery S. Pierce. 2008. It's on my other computer! computing with multiple devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. Association for Computing Machinery, New York, NY, USA, 767–776. https://doi.org/10.1145/1357054.1357177

[25] Brenda Dervin. 1983. An overview of sense-making research concepts, methods, and results to date. (1983). http://www.worldcat.org/title/overview-of-sense-making-research-concepts-methods-and-results-to-date/oclc/733067203

[26] Mira Dontcheva, Steven M. Drucker, Geraldine Wade, David Salesin, and Michael F. Cohen. 2006. Summarizing Personal Web Browsing Sessions. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 115–124. https://doi.org/10.1145/1166253.1166273

[27] Evernote. [n.d.]. Best Note Taking App - Organize Your Notes with Evernote. https://evernote.com

[28] Facebook. 2018. React - A JavaScript library for building user interfaces. https://reactjs.org/

[29] Mingming Fan, Zhen Li, and Franklin Mingzhe Li. 2020. Eyelid Gestures on Mobile Devices for People with Motor Impairments. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3373625.3416987

[30] Mingming Fan, Zhen Li, and Franklin Mingzhe Li. 2021. Eyelid gestures for people with motor impairments. *Commun. ACM* 65, 1 (Dec. 2021), 108–115. https://doi.org/10.1145/3498367

[31] Guillaume Faure, Olivier Chapuis, and Nicolas Roussel. 2009. Power tools for copying and moving: useful stuff for your desktop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. Association for Computing Machinery, New York, NY, USA, 1675–1678. https://doi.org/10.1145/1518701.1518958

[32] Kristie Fisher, Scott Counts, and Aniket Kittur. 2012. Distributed Sensemaking: Improving Sensemaking by Leveraging the Efforts of Previous Users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 247–256. https://doi.org/10.1145/2207676.2207711

[33] Shane Frederick, George Loewenstein, and Ted O'Donoghue. 2002. Time Discounting and Time Preference: A Critical Review. *Journal of Economic Literature* 40, 2 (June 2002), 351–401. https://doi.org/10.1257/002205102320161311

[34] Google. 2012. Google Notebook. https://www.google.com/googlenotebook/faq.html

[35] Google. 2019. Angular - One Framework. Mobile & Desktop. https://angular.io/

[36] Nathan Hahn, Joseph Chee Chang, and Aniket Kittur. 2018. Bento Browser: Complex Mobile Search Without Tabs. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, Montreal QC, Canada, 251:1–251:12. https://doi.org/10.1145/3173574.3173825

[37] Yoshinori Hijikata. 2004. Implicit user profiling for on demand relevance feedback. In *Proceedings of the 9th international conference on Intelligent user interfaces (IUI '04)*. Association for Computing Machinery, New York, NY, USA, 198–205. https://doi.org/10.1145/964442.964480

[38] Ken Hinckley, Xiaojun Bi, Michel Pahud, and Bill Buxton. 2012. Informal Information Gathering Techniques for Active Reading. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1893–1896. https://doi.org/10.1145/2207676.2208327 event-place: Austin, Texas, USA.

[39] Andrew Hogue and David Karger. 2005. Thresher: automating the unwrapping of semantic content from the World Wide Web. In *Proceedings of the 14th international conference on World Wide Web (WWW '05)*. Association for Computing Machinery, New York, NY, USA, 86–95. https://doi.org/10.1145/1060745.1060762

[40] Lichan Hong, Ed H. Chi, Raluca Budiu, Peter Pirolli, and Les Nelson. 2008. SparTag.us: a low cost tagging system for foraging of web content. In *Proceedings of the working conference on Advanced visual interfaces (AVI '08)*. Association for Computing Machinery, New York, NY, USA, 65–72. https://doi.org/10.1145/1385569.1385582

[41] Amber Horvath, Michael Xieyang Liu, River Hendriksen, Connor Shannon, Emma Paterson, Kazi Jawad, Andrew Macvean, and Brad A Myers. 2022. Understanding How Programmers Can Use Annotations on Documentation. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3491102.3502095

[42] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861 [cs]* (April 2017). http://arxiv.org/abs/1704.04861 arXiv: 1704.04861.

[43] Jane Hsieh, Michael Xieyang Liu, Brad A. Myers, and Aniket Kittur. 2018. An Exploratory Study of Web Foraging to Understand and Support Programming Decisions. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 305–306. https://doi.org/10.1109/VLHCC.2018.8506517 ISSN: 1943-6092.

[44] Jeff Huang, Ryen W. White, Georg Buscher, and Kuansan Wang. 2012. Improving searcher models using mouse cursor activity. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval (SIGIR '12)*. Association for Computing Machinery, New York, NY, USA, 195–204. https://doi.org/10.1145/2348283.2348313

[45] Ionic. [n.d.]. Cross-Platform Mobile App Development. https://ionicframework.com/

[46] Shamsi T. Iqbal, Jaime Teevan, Dan Liebling, and Anne Loomis Thompson. 2018. Multitasking with Play Write, a Mobile Microproductivity Writing Tool. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. Association for Computing Machinery, New York, NY, USA, 411–422. https://doi.org/10.1145/3242587.3242611

[47] Zachary Ives, Craig Knoblock, Steve Minton, Marie Jacob, Partha Talukdar, Rattapoom Tuchinda, Jose Luis Ambite, Maria Muslea, and Cenk Gazen. 2009. Interactive Data Integration through Smart Copy & Paste. *arXiv:0909.1769 [cs]* (Sept.

2009). http://arxiv.org/abs/0909.1769 arXiv: 0909.1769.

[48] William Jones, Harry Bruce, and Susan Dumais. 2001. Keeping found things found on the web. In *Proceedings of the tenth international conference on Information and knowledge management (CIKM '01)*. Association for Computing Machinery, New York, NY, USA, 119–126. https://doi.org/10.1145/502585.502607

[49] William Jones, Susan Dumais, and Harry Bruce. 2002. Once found, what then? A study of "keeping" behaviors in the personal use of Web information. *Proceedings of the American Society for Information Science and Technology* 39, 1 (2002), 391–402. https://doi.org/10.1002/meet.1450390143 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/meet.1450390143.

[50] Andruid Kerne, Andrew M Webb, Steven M Smith, Rhema Linder, Nic Lupfer, Yin Qu, Jon Moeller, and Sashikanth Damaraju. 2014. Using metrics of curation to evaluate information-based ideation. *ACM Transactions on Computer-Human Interaction (ToCHI)* 21, 3 (2014), 1–48. Publisher: ACM New York, NY, USA.

[51] Aniket Kittur, Andrew M. Peters, Abdigani Diriye, and Michael Bove. 2014. Standing on the Schemas of Giants: Socially Augmented Information Foraging. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. ACM, New York, NY, USA, 999–1010. https://doi.org/10.1145/2531602.2531644

[52] Aniket Kittur, Andrew M. Peters, Abdigani Diriye, Trupti Telang, and Michael R. Bove. 2013. Costs and Benefits of Structured Information Foraging. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2989–2998. https://doi.org/10.1145/2470654.2481415

[53] G. Klein, B. Moon, and R. R. Hoffman. 2006. Making Sense of Sensemaking 1: Alternative Perspectives. *IEEE Intelligent Systems* 21, 4 (July 2006), 70–73. https://doi.org/10.1109/MIS.2006.75

[54] Gordon Kurtenbach and William Buxton. 1993. The limits of expert performance using hierarchic marking menus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93)*. Association for Computing Machinery, New York, NY, USA, 482–487. https://doi.org/10.1145/169059.169426

[55] Edward Lank and Eric Saund. 2005. Sloppy selection: Providing an accurate interpretation of imprecise selection gestures. *Computers & Graphics* 29, 4 (Aug. 2005), 490–500. https://doi.org/10.1016/j.cag.2005.05.003

[56] Huy Viet Le, Sven Mayer, Maximilian Weiß, Jonas Vogelsang, Henrike Weingärtner, and Niels Henze. 2020. Shortcut Gestures for Mobile Text Editing on Fully Touch Sensitive Smartphones. *ACM Transactions on Computer-Human Interaction* 27, 5 (Aug. 2020), 33:1–33:38. https://doi.org/10.1145/3396233

[57] Scott LeeTiernan, Shelly Farnham, and Lili Cheng. 2003. Two methods for auto-organizing personal web history. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. Association for Computing Machinery, New York, NY, USA, 814–815. https://doi.org/10.1145/765891.766008

[58] James R. Lewis. 2018. The System Usability Scale: Past, Present, and Future. *International Journal of Human–Computer Interaction* 34, 7 (July 2018), 577–590. https://doi.org/10.1080/10447318.2018.1455307 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10447318.2018.1455307.

[59] Franklin Mingzhe Li, Di Laura Chen, Mingming Fan, and Khai N. Truong. 2021. "I Choose Assistive Devices That Save My Face": A Study on Perceptions of Accessibility and Assistive Technology Use Conducted in China. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3411764.3445321

[60] Franklin Mingzhe Li, Michael Xieyang Liu, Yang Zhang, and Patrick Carrington. 2022. Freedom to Choose: Understanding Input Modality Preferences of People with Upper-body Motor Impairments for Activities of Daily Living. https://doi.org/10.1145/3517428.3544814 arXiv:2207.04344 [cs].

[61] Mingzhe Li, Mingming Fan, and Khai N. Truong. 2017. BrailleSketch: A Gesture-based Text Input Method for People with Visual Impairments. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '17)*. Association for Computing Machinery, New York, NY, USA, 12–21. https://doi.org/10.1145/3132525.3132528

[62] Yang Li. 2010. Protractor: a fast and accurate gesture recognizer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. Association for Computing Machinery, New York, NY, USA, 2169–2172. https://doi.org/10.1145/1753326.1753654

[63] Michael Xieyang Liu, Jane Hsieh, Nathan Hahn, Angelina Zhou, Emily Deng, Shaun Burley, Cynthia Taylor, Aniket Kittur, and Brad A. Myers. 2019. Unakite: Scaffolding Developers' Decision-Making Using the Web. In *Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. ACM, New Orleans, LA, USA, 67–80. https://doi.org/10.1145/3332165.3347908 event-place: New Orleans, LA, USA.

[64] Michael Xieyang Liu, Aniket Kittur, and Brad A. Myers. 2021. To Reuse or Not To Reuse? A Framework and System for Evaluating Summarized Knowledge. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (April 2021), 166:1–166:35. https://doi.org/10.1145/3449240

[65] Michael Xieyang Liu, Aniket Kittur, and Brad A. Myers. 2022. Crystalline: Lowering the Cost for Developers to Collect and Organize Information for Decision Making. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing*

Systems (CHI '22). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3491102.3501968 event-place: New Orleans, LA, USA.

[66] Yoelle S. Maarek, Michal Jacovi, Menachem Shtalhaim, Sigalit Ur, Dror Zernik, Israel Z. Ben-Shaul, Yoelle S. Maarek, Michal Jacovi, Menachem Shtalhaim, Sigalit Ur, Dror Zernik, and Israel Z. Ben-Shaul. 1997. WebCutter: a system for dynamic and tailorable site mapping. *Computer Networks and ISDN Systems* 29, 8-13 (Sept. 1997), 1269–1279. https://doi.org/10.1016/S0169-7552(97)00050-0

[67] Alireza Mansouri, Lilly Suriani Affendey, and Ali Mamat. 2008. Named entity recognition approaches. *International Journal of Computer Science and Network Security* 8, 2 (2008), 339–344. Publisher: Citeseer.

[68] Gary Marchionini. 1995. *Information Seeking in Electronic Environments*. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511626388

[69] Catherine C. Marshall and Sara Bly. 2005. Saving and Using Encountered Information: Implications for Electronic Periodicals. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 111–120. https://doi.org/10.1145/1054972.1054989 event-place: Portland, Oregon, USA.

[70] M. R. Morris, A. J. B. Brush, and B. R. Meyers. 2007. Reading Revisited: Evaluating the Usability of Digital Display Surfaces for Active Reading Tasks. In *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07)*. 79–86. https://doi.org/10.1109/TABLETOP.2007.12

[71] Mozilla. 2022. Browser Extensions | MDN. https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions

[72] Obsidian. 2022. Obsidian. https://obsidian.md/

[73] Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological Review* 106, 4 (1999), 643–675. https://doi.org/10.1037/0033-295X.106.4.643 Place: US Publisher: American Psychological Association.

[74] Peter Pirolli and Stuart Card. 2005. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In *Proceedings of International Conference on Intelligence Analysis*. http://www.phibetaiota.net/wp-content/uploads/2014/12/Sensemaking-Process-Pirolli-and-Card.pdf

[75] Napol Rachatasumrit, Gonzalo Ramos, Jina Suh, Rachel Ng, and Christopher Meek. 2021. ForSense: Accelerating Online Research Through Sensemaking Integration and Machine Research Support. In *26th International Conference on Intelligent User Interfaces (IUI '21)*. Association for Computing Machinery, New York, NY, USA, 608–618. https://doi.org/10.1145/3397481.3450649

[76] Volker Roth and Thea Turner. 2009. Bezel swipe: conflict-free scrolling and multiple selection on mobile touch screen devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. Association for Computing Machinery, New York, NY, USA, 1523–1526. https://doi.org/10.1145/1518701.1518933

[77] Dean Rubine. 1991. Specifying gestures by example. *ACM SIGGRAPH Computer Graphics* 25, 4 (July 1991), 329–337. https://doi.org/10.1145/127719.122753

[78] Dean Rubine. 1992. Combining gestures and direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92)*. Association for Computing Machinery, New York, NY, USA, 659–660. https://doi.org/10.1145/142750.143072

[79] Daniel M. Russell, Mark J. Stefik, Peter Pirolli, and Stuart K. Card. 1993. The Cost Structure of Sensemaking. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93)*. ACM, New York, NY, USA, 269–276. https://doi.org/10.1145/169059.169209

[80] M. C. Schraefel and Yuxiang Zhu. 2001. Interaction design for Web-based, within-page collection making and management. In *Proceedings of the 12th ACM conference on Hypertext and Hypermedia (HYPERTEXT '01)*. Association for Computing Machinery, New York, NY, USA, 125. https://doi.org/10.1145/504216.504247

[81] M. C. schraefel, Yuxiang Zhu, David Modjeska, Daniel Wigdor, and Shengdong Zhao. 2002. Hunter Gatherer: Interaction Support for the Creation and Management of Within-web-page Collections. In *Proceedings of the 11th International Conference on World Wide Web (WWW '02)*. ACM, New York, NY, USA, 172–181. https://doi.org/10.1145/511446.511469

[82] Jeffrey Stylos, Brad A. Myers, and Andrew Faulring. 2004. Citrine: providing intelligent copy-and-paste. In *Proceedings of the 17th annual ACM symposium on User interface software and technology (UIST '04)*. Association for Computing Machinery, New York, NY, USA, 185–188. https://doi.org/10.1145/1029632.1029665

[83] Apple Support. 2022. Make the pointer easier to see on Mac. https://support.apple.com/guide/mac-help/make-the-pointer-easier-to-see-mchlp2920/12.0/mac/12.0

[84] Amanda Swearngin, Shamsi Iqbal, Victor Poznanski, Mark Encarnación, Paul N. Bennett, and Jaime Teevan. 2021. Scraps: Enabling Mobile Capture, Contextualization, and Use of Document Resources. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3411764.3445185

[85] 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 (CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2927–2936. https://doi.org/10.1145/1978942.1979376

[86] Paul Thurrott and Rafael Rivera. 2009. *Windows 7 Secrets*. John Wiley & Sons. Google-Books-ID: 1EXt2U84WZ0C.

[87] Shari Trewin, Simeon Keates, and Karyn Moffatt. 2006. Developing steady clicks: a method of cursor assistance for people with motor impairments. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility (Assets '06)*. Association for Computing Machinery, New York, NY, USA, 26–33. https://doi.org/10.1145/1168987.1168993

[88] Radu-Daniel Vatavu and Ovidiu-Ciprian Ungurean. 2019. Stroke-Gesture Input for People with Motor Impairments: Empirical Results & Research Roadmap. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300445

[89] Harald Weinreich, Hartmut Obendorf, Eelco Herder, and Matthias Mayer. 2006. Off the Beaten Tracks: Exploring Three Aspects of Web Navigation. In *Proceedings of the 15th International Conference on World Wide Web (WWW '06)*. ACM, New York, NY, USA, 133–142. https://doi.org/10.1145/1135777.1135802

[90] Steve Whittaker. 2011. Personal information management: From information consumption to curation. *Annual Review of Information Science and Technology* 45, 1 (2011), 1–62. https://doi.org/10.1002/aris.2011.1440450108

[91] Alex C. Williams, Harmanpreet Kaur, Shamsi Iqbal, Ryen W. White, Jaime Teevan, and Adam Fourney. 2019. Mercury: Empowering Programmers' Mobile Work Practices with Microproductivity. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 81–94. https://doi.org/10.1145/3332165.3347932

[92] Parker Williams, Jeffrey Jenkins, and Joseph Valacich. 2016. Real-Time Hand Tremor Detection via Mouse Cursor Movements for Improved Human-Computer Interactions: An Exploratory Study. *SIGHCI 2016 Proceedings* (Dec. 2016). https://aisel.aisnet.org/sighci2016/10

[93] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. 2009. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. Association for Computing Machinery, New York, NY, USA, 1083–1092. https://doi.org/10.1145/1518701.1518866

[94] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology (UIST '07)*. Association for Computing Machinery, New York, NY, USA, 159–168. https://doi.org/10.1145/1294211.1294238

[95] Sungjoon Steve Won, Jing Jin, and Jason I. Hong. 2009. Contextual web history: using visual and contextual cues to improve web browser history. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1457–1466. https://doi.org/10.1145/1518701.1518922

[96] Jingjie Zheng, Blaine Lewis, Jeff Avery, and Daniel Vogel. 2018. FingerArc and FingerChord: Supporting Novice to Expert Transitions with Guided Finger-Aware Shortcuts. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. Association for Computing Machinery, New York, NY, USA, 347–363. https://doi.org/10.1145/3242587.3242589

APPENDIX

Skeema Content Clipping

For clipping, Skeema offers two methods:

- **Clipping text**: First, use the cursor to select the desired content (Figure 6a in the appendix) in the conventional way, then click the clipping button (containing the Skeema chrome extension icon, see Figure 6a1) that popups to collect the selected texts into an information card in the holding tank. After that, there will be a popup dialog (Figure 6c) on the upper right corner of the screen to indicate success as well as allow users to externalize their mental context such as valence judgement into the notes field associated with the collected content.

- **Clipping screenshot**: To collect images and other types of non-text content, users can use the screenshot feature: they need to click the screenshot button (Figure 6b1) on right edge of the browser window to enter the screenshot mode, press and hold down the mouse left button to drag out a bounding box around the desired content, and release the left button. Then the screenshot will be saved into the holding tank as an image, followed by the popup dialog (Figure 6c).
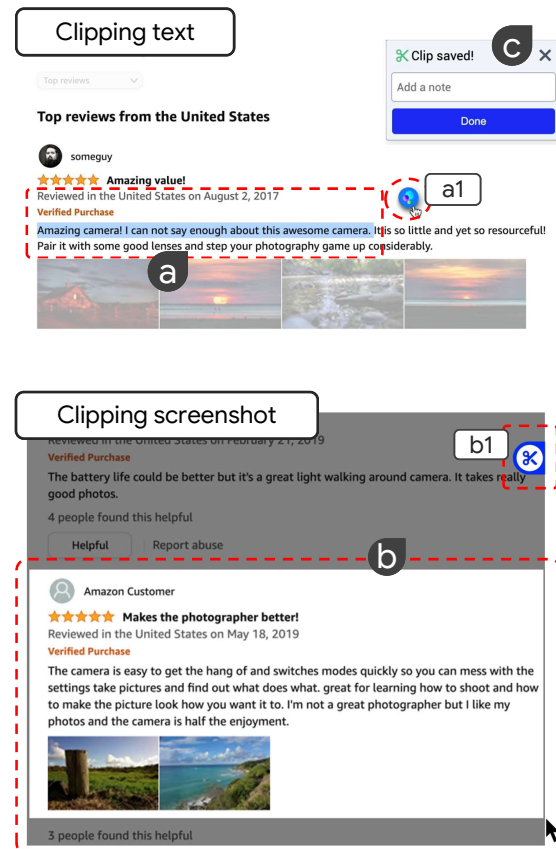

Figure 6: Two types of information clipping mechanisms that Skeema supports: clipping text (top) and clipping screenshot (bottom).