RubySlippers: Supporting Content-based Voice Navigation for How-to Videos

Minsuk Chang*
School of Computing, KAIST
Naver AI LAB
minsuk@kaist.ac.kr

Mina Huh
School of Computing, KAIST
minarainbow@kaist.ac.kr

Juho Kim
School of Computing, KAIST
juhokim@kaist.ac.kr

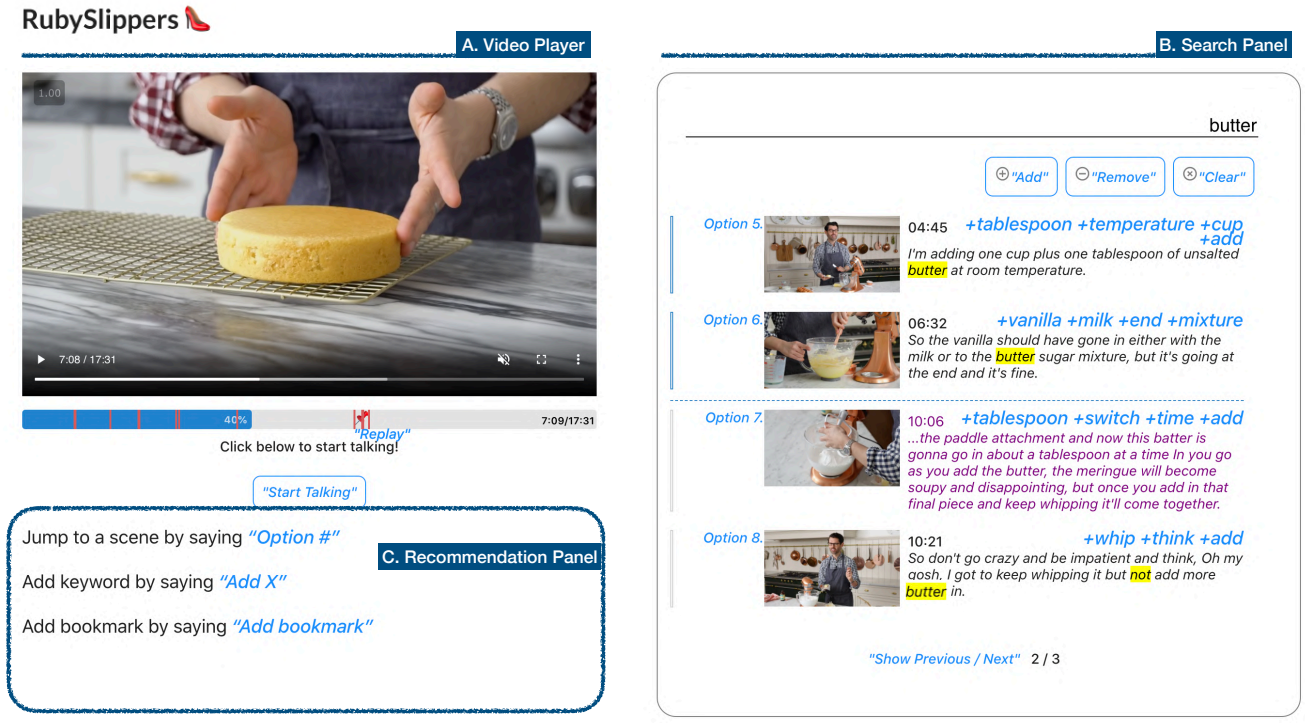


Figure 1: RubySlippers is a multi-modal interface supporting voice navigation with three main components: (A) Video player and timeline. (B) Search panel where keywords-based search results are shown. (C) Recommendation panel provides suggestions of search keywords and available navigation commands at each interaction interval.

ABSTRACT

Directly manipulating the timeline, such as scrubbing for thumbnails, is the standard way of controlling how-to videos. However, when how-to videos involve physical activities, people inconveniently alternate between controlling the video and performing the tasks. Adopting a voice user interface allows people to control the video with voice while performing the tasks with hands. However, naively translating timeline manipulation into voice user interfaces (VUI) results in temporal referencing (e.g. "rewind 20 seconds"), which requires a different mental model for navigation and thereby limiting users' ability to peek into the content. We present RubySlippers, a system that supports efficient content-based voice navigation through keyword-based queries. Our computational pipeline automatically detects referenceable elements in the video, and finds the video segmentation that minimizes the number of needed navigational commands. Our evaluation (N=12) shows that participants could perform three representative navigation tasks with fewer commands and less frustration using RubySlippers than the conventional voice-enabled video interface.

*This work was done when this author was at KAIST

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445131

CCS CONCEPTS

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

KEYWORDS

Voice User Interface, Video Navigation, How-to Videos

ACM Reference Format:
Minsuk Chang, Mina Huh, and Juho Kim. 2021. RubySlippers: Supporting Content-based Voice Navigation for How-to Videos. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3411764.3445131

1 INTRODUCTION

Learning from how-to videos involves complex navigational scenarios [5]. For example, people often revisit previously watched segments to clarify misunderstandings, skip seemingly familiar contents, or jump to the later parts to see the result and prepare for future steps. A common strategy people use in traditional video interfaces with mouse and keyboard is "content-based referencing"—for example, peeking the video's content either by scrubbing or hovering on the timeline for thumbnails or performing sequential jumps to move the playback positions.

However, how-to videos for popular tasks, such as cooking, makeup, and home-improvements, require manipulations with physical objects and involve physical activities. As a result, viewers need to use both hands to control the video and carry out the task at hand. This incurs costly context switches and heavy cognitive load while tracking the progress of both the video and the task. Ideally, voice user interfaces like Alexa or Google Assistant can provide an opportunity to separate the two activities - controlling of video using voice and applying the instruction with two hands.

The most straightforward and standard method of supporting voice interaction for video interfaces is to directly translate the timeline manipulation into voice commands, such as navigating the video with commands like "skip 20 seconds" or "go to three minutes and 15 seconds". However, this "temporal referencing" strategy requires a different mental model for navigation than directly manipulating the timeline because it limits the users' ability to peek into the content.

Researchers have analyzed how people currently navigate how-to videos when only remote-control like voice commands are available, and how people want an ideal voice navigation system to be designed with a wizard-of-oz study. The most informative and decisive design criteria inferred are that people idealize the strategies they use in daily conversations with other people [5].

Through a formative study with a research probe that followed the guidelines (i.e. supporting conversational strategies), we have identified four challenges in supporting "content-based referencing" in voice user interfaces for how-to videos. First, users want to use succinct keyword-based queries instead of conversational commands to alleviate the burden of constructing sentences Second, users cannot recall the exact vocabulary used in the video. Third, users have difficulties with remembering the available commands. Finally, unlike timeline interactions, voice inputs suffer from speech recognition errors and speech parsing delays.

To overcome these challenges, we present RubySlippers, a prototype system that supports voice-controlled temporal referencing and content-based referencing through keyword-based queries. Our computational pipeline automatically detects referenceable elements in the video and finds the video segmentation that minimizes the number of needed navigational commands. RubySlippers also suggests commands and keywords contextually to inform the user about both available commands and potential candidate target scenes.

In a within-subjects study with 12 participants, we asked participants to carry out a series of representative navigational tasks, focusing on evaluating the effectiveness and benefits of content-based referencing strategies. Participants found the keyword-based queries—our main feature for supporting content-based referencing—in RubySlippers useful and convenient for navigation. They were also able to effectively mix content-based referencing and temporal referencing using RubySlippers to fit the needs of their navigational task.

This paper makes the following main contributions:

- Results of analysis comparing temporal based referencing and content-based referencing techniques in voice video navigation. Specifically, challenges with content-based referencing in voice user interfaces.
- RubySlippers, a prototype video interface which supports both temporal referencing and content-based referencing with voice
- The computational pipeline that segments a how-to video into units that effectively support keyword based interaction techniques for navigation
- Results from the controlled study showing the value of keyword based references in supporting efficient voice navigation for how-to videos

2 RELATED WORK

This work expands prior research on video navigation techniques, interaction techniques for tutorial videos, and designing voice user interfaces.

2.1 Video Navigation Interaction

Swift [23] and Swifter [25] improved scrubbing interfaces by presenting pre-cached thumbnails on the timeline. Together with the video timeline, video thumbnails [36] are commonly used to provide viewers with a condensed and straightforward preview of the video contents, facilitating the searching and browsing experiences. To support users to easily spot previously watched videos, Hajri et al. [1] proposed personal browsing history visualizations. Crockford et al. [8] found that VCR-like control sets, consisting of low-level pause/play operations, both enhanced and limited users' browsing capabilities, and that users employ different playback speeds for different content. Dragicevic et al. [10] found that the direct manipulation of video content (via dragging interactions) is more suitable than the direct manipulation of a timeline interface for visual content search tasks. A study on video browsing strategies reported that in-video object identification and video understanding tasks

require different cognitive processes [9]. Object identification requires localized attention, whereas video understanding requires global attention.

Our work builds upon video navigation research by exploring efficient methods of implementing the advantages of temporal and content-based navigation techniques into voice interactions while focusing on how-to videos.

2.2 Interacting with How-to Videos

For software tutorials, Nguyen et al. [30] found that users complete tasks more effectively by interacting with the software through direct manipulation of the tutorial video than using conventional video players. Pause-and-play [34] detected important events in the video and linked them with corresponding events in the target application for software tutorials. FollowUs [21] captured video demonstrations of users as they perform a tutorial so that subsequent users can use the original tutorial, or choose from a library of captured community demonstrations of each tutorial step. Similarly, Wang et al. [40] showed that at-scale analysis of community-generated videos and command logs can provide workflow recommendations and tutorials for complex software. Also, SceneSkim [32] demonstrated how to parse transcripts and summaries for video snippet search. VideoDigests [33] presented an interface for authoring videos that make content-based transcript search techniques work well.

Specific to educational videos, LectureScape [16] utilized large scale user interaction traces to augment the timeline with meaningful navigation points. ToolScape [15] utilized storyboard summaries and an interactive timeline to enable learners to quickly scan, filter, and review multiple videos without having to play them from the beginning to the end. Localized word cloud summarizing the scenes in a MOOC video [41] has decreased navigation time. Smart Jump [44] is a system that suggests the best position for a jump-back. The authors' analysis of the navigation data revealed that more than half of the jump-backs are due to the "bad" positions of the previous jump-backs. Specific to referencing behaviors, commenting on video sharing platforms showed that people use temporal location as the main means of anchor [42].

In this research we build upon this rich line of augmenting interactions with how-to vidoes, but specifically focusing on how to augment voice interaction techniques with respect to different navigation scenarios.

2.3 Troubles with Voice Interfaces and Common Repair Strategies

Most voice user interfaces adapt a conversational agent like Amazon's Alexa, Apple's Siri or Google Assistant. While previous work has shown the effectiveness of voice interaction in assisting user tasks such as image editing [22] and parsing images [6], most interactions for video and audio control are primarily basic content playback controls [4].

Commonly reported user problems in using voice interactions are discoverability of available commands [7] and balancing the trade-off between expressiveness and efficiency [26]. Instead of controlling the video with voice, researchers have explored efficient methods for controlling the software with voice. For example, displaying available voice commands when the user hovers the tools in the image editing software [37], and vocal shortcuts which are short spoken phrases to control interfaces [17] have been shown to be effective. However, these methods are specifically designed to only work for software-related tasks in which the voice commands operate the software, making them difficult to apply for how-to videos that involve physical tasks.

For voice user interfaces, common recovery strategies are hyperarticulation and rephrasing [28], both of which usually do not lead to a different outcome. Although the experiment was done with chatbots and not with voice assistants, providing options and explanations as a means of repairing a broken conversation was generally favored by users [3].

There have been many guideline level suggestions for how to design voice interactions. For example, guiding users to learn what verbal commands can execute VUI actions and what actions are supported to accomplish desired tasks with the system are important [29]. Also, allowing users to recognize and recover from errors is just as important as preventing user errors, and flexibility and efficiency of use is needed [27].

One of the most relevant building blocks of our approach is the analysis of navigation behavior using voice. Instead of watching these how-to videos passively, viewers actively control the video to pause, replay, and skip forward or backwards while following along the video instructions. Based on these interaction needs, previous work has proposed usage of conversational interfaces for navigation of how-to videos for physical tasks [5].

Voice interfaces for navigating how-to videos remain underexplored, and no concrete VUI specifically designed to support how-to video navigation has been introduced hitherto. We provide interpretations of the design recommendations specifically for navigating how-to videos with voice interfaces, and demonstrate how they can be realized with a prototype implementation.

3 FORMATIVE STUDY

In this research, we characterize two navigation strategies, temporal referencing and content-based referencing. Temporal referencing is when the anchor of navigation in the user mental model is the time. For example, in a typical GUI-based video player, viewer uses temporal referencing by clicking on the timeline when the viewer knows that's exactly the timestamp of the targeted scene. For voice user interfaces, the voice commands like "skip 20 seconds" support temporal referencing.

For content-based referencing, the anchor for navigation is the content. For example, viewers use content-based referencing when they examine the thumbnail or moves around playback position of the video to navigate to the target scene. Translating this to voice user interfaces, users must be able to issue voice commands that describe the content of scenes like "go to the part where the chef dices tomatoes". Although for the latter, we are yet to see a voice-driven system design that supports this effectively.

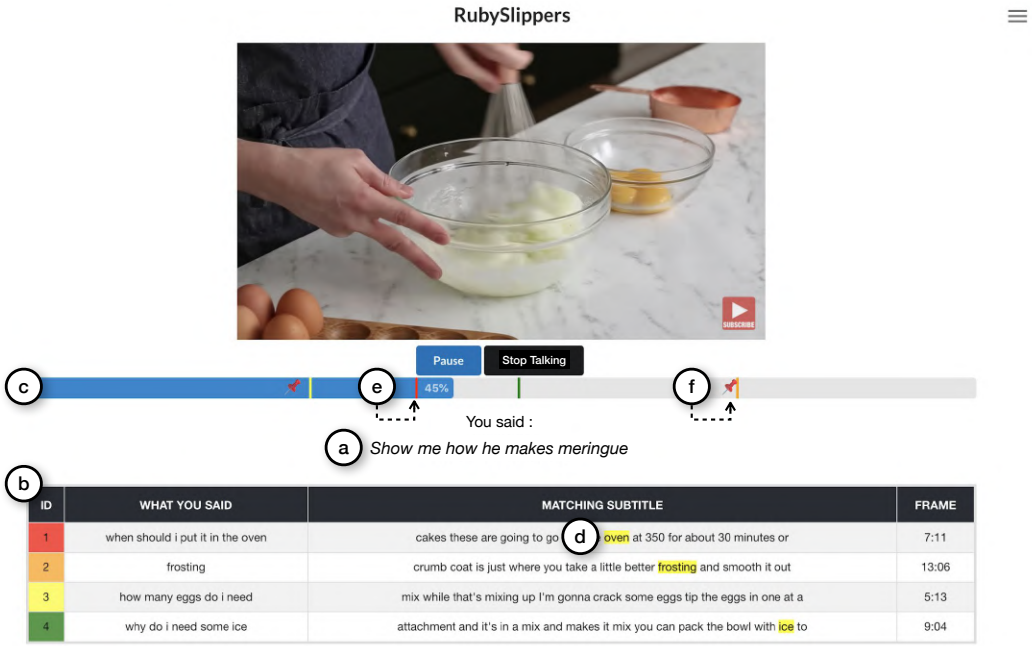


Figure 2: Our research probe supports temporal referencing and content-based referencing through basic speech recognition. Main features of research probe: (a) Real-time transcript of the user is shown. (b) The history table stores all previous queries, matching subtitle and the timestamps and highlights the word with the most influence to the resulting target scene. (c) On the progress bar, the scenes resulting from previous navigations are marked (e) which users can easily reissue with shortcut. (f) Users can also manually add bookmarks.

3.1 Research Probe

To understand the advantages and disadvantages of the two referencing strategies in voice user interfaces, we conducted a formative study with 12 participants.

As the apparatus of the formative study, we built a voice-enabled video player that supports both temporal referencing and content-based referencing (Figure 2). Using the player, users can play, pause, fast-forward, and rewind by specifying the location or the interval for temporal referencing. For content-based referencing, users can describe the target scene they are looking for in a conversational manner. The system parses the users' voice query and calculate the similarity against each sentence in the transcript for the best match. We used word-mover distance [20] based algorithm for matching. While the method used not being the state of the art technology, we judged the accuracy this method provided was enough to understand the advantages and challenges of content-based navigation at this stage. Considering the frustration from the speech recognition errors, which was well reported in [28], we asked our participants to look beyond the speech recognition errors. The research probe also stores all previous queries and the matched results in a table (Figure 2.(b)) as a bookmark, which users can simply refer to by their ID to reissue the same navigation query, like "go to 2" for second item in the bookmark.

3.2 Study Procedure

We recruited participants with an online community advertisement. The criteria for invitation were the prior exposure to video tutorials and basic English proficiency.

We conducted a counter-balanced within-subjects study, where participants could only use one of the two strategies to perform five common navigating tasks in how-to videos: navigation to a scene with specific object usage, navigation to all scenes where a specific object appears, navigation to a scene with factual information, and multiple video comparison. To evaluate the efficiency and efficacy of each strategy, we measured both task completion time and cognitive load with 10-point scale NASA-TLX [13].

At the end of the study, the participants were given the freedom to explore mixing the two strategies. We also conducted semi-structured interviews to gather qualitative feedback for a deeper understanding.

To give a preview of the system, we gave a brief tutorial session on how to use each feature. To familiarze and build trust in the system's ability, we also provided some example "working cases", like "play", "paust", "stop" for temportal referencing, and "show me where she bought potatoes", "How many liters of water does she use for the plant" for content-based referencing. There were 3 sessions altogether and 3 domains of how-to videos were chosen: baking, packing and planting. In the first two sessions, participants were restricted to use only one navigation scheme where in the last session they were allowed to used both. In each session, 5 questions

from 3 different types were asked to the participants. The first type was visual search, which meant that they were asked to navigate to the frame. Both single target and multi target questions were asked. The second type was video question and answering. Here we asked both short answers and long answer questions. In the third type, multi-video search, participants were asked to watch 5 different videos of the same domain and answer the questions which asked about the trend of watched videos. To complement the performance analysis with a qualitative understanding of participants' experience, we included semi-structured interviews. Each interview session took about 15 minutes asking about their how they felt about using the system.

3.3 Results

The difference in cognitive load between the two strategies was not significant. People reported a high task load in both the temporal and content-based referencing Table 1. However, the finding we learned is that the sources of task load for each strategies are different.

For temporal referencing, participants said it is tedious and laborious to jump around because in this condition they had to specify the exact temporal location of the scene that they want to watch, and remembering exactly when a certain event happened or having to make multiple corrections to reach the moment in the video is very tiring. They also felt pressured.

For content-based referencing, participants' stress mostly came from system failures in understanding their utterances. Participants were thinking too much about which words they should pick when formulating the query sentences, because either they could not remember the vocabulary or because they wanted to be efficient and find one magic word to include that makes the hit. From the interviews, participants reported issuing a voice command in "conversational" form is burdensome, and would rather use a combination of discrete keywords. For example when P5 tried "From which shop did she buy the bags?", the system did not populate the scene P5 wanted. In the interview, P5 said "I had to try hard not to include words that are less necessary, but buying and bags ARE necessary. It's so stressful to come up with a *correct* sentence, and repeat long sentences over and over." Participants felt like there is one correct sentence that will take them to the scene they want, and it suddenly became a guessing game for them that they did not want to play.

From the interview feedback, we deducted the problem of content-based voice navigation into the problem of how to help users find the minimum set of keywords that describe the scene they are looking for. Combining both study results and the interview findings, we have identified the following user challenges in efficiently navigating how-to videos difficult:

C1. Difficulty in referring to objects and actions that appear multiple times across the video
C2. Difficulty in precisely recalling the exact vocabulary due to divided attention
C3. Difficulty in remembering what the available commands are and how to execute them
C4. Inconvenience caused by the time delay from parsing and speech processing

The first challenge is that the same objects and actions appear multiple times throughout the video, and the more important they are, the more frequently they appear. This directly conflicted with what participants wanted to do. Participants wanted use as fewest words as possible when referencing. We observed that especially to minimize parsing errors, participants tried to use shorter and shorter sentences when they were experiencing system failures. However, because the objects and actions appear in multiple places across the video, participants needed to construct longer sentences in order to narrow down, which caused more parsing and recognition errors.

The second challenge is that participants have difficulty with recalling the exact words used in the video because the attention of the user is divided into performing the task and formulating the query. While participants noted recall of the words as easier than recall of the timestamps, it is still challenging especially when equipped with little background knowledge. For example, when participants were presented with an image of "a carry-on"—the precise term used in the video— and were asked to "find at which shop the person in the video bought this?", they tried different words like bags, baggage, luggage and suitcase in their query sentences. The system could not find the correct scene. Also, P2 first searched for the word "sugar" to find out how much was needed. When 17 results showed up, P2 tried with the query "spoon of sugar" but got 0 matching result. Then, P2 tried with the query "cup of sugar" and got 10 results. P2 failed in narrowing down the search, and had to examine all the options to find the answer.

The third challenge is that users do not know what commands are available nor how to execute them. Users are frustrated when they forgot how to initiate commands or update them when the initial command failed. Participants repeatedly asked how to talk to the system, and whether they can see the list of commands next to them all the time.

The fourth challenge is that voice interactions take more time, because they have parsing delays whereas direct manipulation of the timeline, which most users are already accustomed to, does not. The first two challenges are cause by the characteristics of a video tutorial and the latter two are commonly reported challenges in voice interfaces.

The first two challenges were uniquely identified through our study, where the latter two are well-reported in previous research in voice interaction usability.

3.4 Design Goals

Based on the analysis of the interview and suggestions from the participants, we identified three design goals for tools to support content-based voice navigation for how-to videos. The design goals individually address three key user tasks in voice based video navigation, which are initiating a command (D3), referencing (D1), and revising the command (D2).

D1. Provide support for efficient content-based referencing using keywords rather than full sentences.
D2. Provide support for effective query updates.
D3. Provide support for informing users about executable commands and potential navigation.

	Mental	Physical	Temporal	Performance	Effort	Frustration	$\overline{X}$
Temporal Referencing	5.67	4.67	6.25	5.58	6.83	6.67	5.56
Content-based Referencing	6.07	4.21	5.86	5.29	6.36	5.86	5.61

Table 1: Cognitive load measured with NASA-TLX for 12 participants. There aren't any significant differences in cognitive load between the two referencing strategies.

4 RUBYSLIPPERS

With the three design goals in mind, we present RubySlippers (Figure 1), a voice enabled video interface that allows users to use both temporal referencing and content-based referencing. Below, we walk through two scenarios illustrating some of the advantages of using RubySlippers when navigating how-to videos, and subsequently describe the features that enable content-based voice navigation. We then also describe the computational pipeline that powers RubySlippers.

4.1 Scenario 1

Dorothy loves to cook at home, but is a novice at baking. She wants to make a birthday cake for a friend with the help from a video tutorial online. She decided to use RubySlippers to avoid touching the computer with hands covered in flour. For the first couple of minutes, she easily followed the instructions using pauses and by changing playback speeds with voice. However, when the chef in the video put the vinegar into the mixture, she couldn't remember how much vinegar was needed. As preparation of the ingredient was in the earlier part of the video, she talked to the system "Vinegar" and could easily find the scene where the Vinegar is being added on the search panel. Dorothy had to just say "option one" to navigate to the part.

While Dorothy was busy whipping the cream, the video kept playing and moved on to a few minutes later. After a couple of failed attempts to guess the original location with the command "Go back 30 seconds", she talked to the system "Cream". However, RubySlippers displayed more than ten scenes where the word "cream" was mentioned. Instead of peeking into all the options, she simply added "whip" by saying "add whip" and came back to the original point.

4.2 Scenario 2

Glinda, a friend of Dorothy, is throwing a birthday party tonight. She is preparing for a party makeup and selects a how-to video of her style. While it is her first time using RubySlippers. After the lip makeup, she wanted to skip the step of blushing cheek and watch how to do contouring. When she said "Contour" to the system, it responded with a list of synonyms appearing in the video which were "Bronzer, Outline, Brown, Shadow, Darken". So she replaced her query with "Bronzer" and could quickly reach the target scene.

Glinda was following the step of applying the glitter on her eyes. While she was applying it to her right eyelid, the video—edited to avoid redundancy—fast-forwarded the same process with the left eye and moved on to the next step. After she re-visited the same scene multiple times to finish the left eye, RubySlippers automatically added a "Replay" mark, reducing the burden of Glinda to repeat the query. When she did a couple of more jump-back by saying "Replay", a loop was created which repeated the same step with no input until she escaped.

4.3 Keyword-based Querying

To address D1, RubySlippers supports keyword-based queries for users to describe parts of the video they would like to navigate to. These keywords are pre-populated using an NLP pipeline which we later describe in the 4.7. RubySlippers returns the list of scenes resulting from the keyword-based search below the search bar Figure 3.(b). The corresponding locations on the timeline are marked with vertical orange lines (Figure 3.(b)). The search keyword is highlighted in the transcript corresponding to the scene.

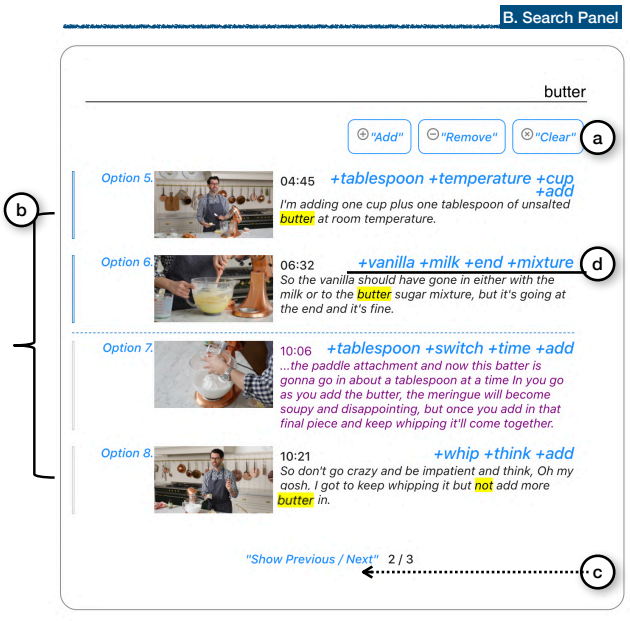


Figure 3: RubySlippers Search Panel: In the search panel, users can search and choose among the option scenes. (a) Users can update the current query by adding, replacing, or removing keywords. (b) The search result is shown in chronological order. Each item has a visual thumbnail, timestamp, transcript, and keyword suggestions for further query specification. (c) Users can also browse search result pages with voice commands. (d) Keywords that help users narrow down the search result are shown.

4.4 Updating Queries with Keyword Composition

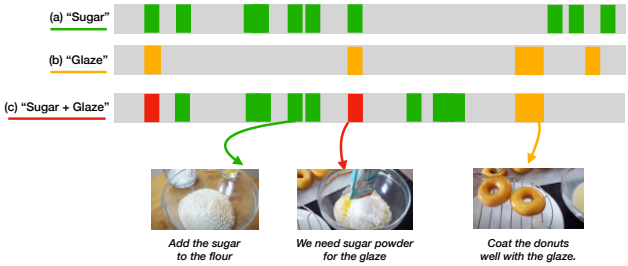


Figure 4: An illustration of how query updating with keyword composition works. Parts containing the keyword "Sugar" are marked in green, parts containing the keyword "Glaze" are marked in orange. The part that contains the composition to two keywords "Sugar" and "Glaze" are marked in red.

To address D2, users can update their previously issued query by adding, replacing or removing keywords (D2). (Figure 3.(a)) RubySlippers assists this query update by informing the users which keywords can narrow down their search result when added to the current query (Figure 3.(d)). For each search item in the list, keywords that are likely to uniquely describe the item while also reducing the number of search results when added are shown. A visual illustration of how query updating works is shown in Figure 4. Also, there is a vertical bar in the search results which indicates where the current video playback is. The visited scenes are visually distinguished with different color so that the user can quickly understand which options are unseen.

4.5 Command and Keyword Suggestion

To address D3, RubySlippers displays available commands or example keywords for initiating the navigation (Figure 5). RubySlippers takes both the current user state and the previous interaction into account to make recommendations. For example, the system shows a word cloud [38] for initial seed (Figure 5.(a)) and displays available commands like "undo" to recover after a navigation has been made. It also suggests semantically similar words if the input keyword from the user does not appear in the video (Figure 5.(e)) and helps users to make quick search by informing how to narrow down the number of options when there are too many (Figure 5.(c)).

4.6 Automated Bookmarks

RubySlippers creates an automated bookmark for frequently visited scenes. Users involved in physical tasks frequently pause to control the pace of the task and to make sure the task progress is aligned with the video's progress [5]. So users often need to visit the same spot in the video multiple times. With the automated bookmarks, users can revisit a previously visited scene without issuing the same command repeatedly. Users can avoid recalling their previous

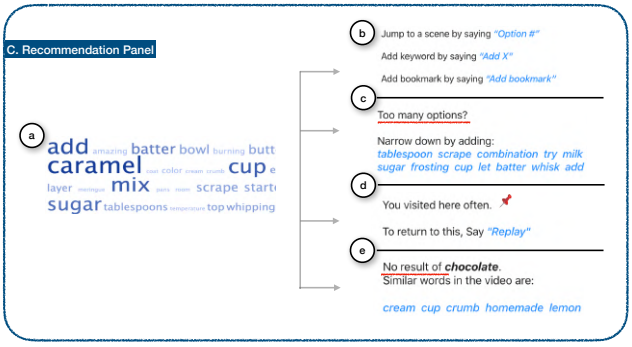


Figure 5: RubySlippers Recommendation: The recommendation panel provides suggestions which adaptively change at each interaction interval. (a) The system displays a word cloud for initial seed. It serves as a keyword summary that can help users recognize and remember the main events happened in each video. (b, d) When the user starts the navigation, it shows available commands so that users can smoothly connect to the next interaction. (c, e) Keywords that can be added to the current query are suggested in support of users narrowing down or fix the search.

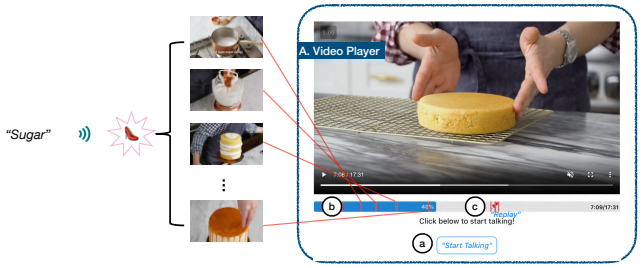


Figure 6: RubySlippers Video Player: The video player consists of the how-to video and the timeline bar showing the progress of the video, and the speech recognition status is shown below. (a) To start speaking to RubySlippers, users first must turn on the speech recognition by clicking the "Start Talking" button. After the recognition is on, real-time transcript of the user is shown. (b) On the timeline, the timestamps of the search result items are marked with vertical red lines. (c) Bookmarks for frequently visited scenes are automatically created and marked with "replay".

queries, which is cumbersome and difficult with voice, but rely on recognition [31].

When a same referenceable unit is visited more than two times, RubySlippers automatically adds a "Replay" mark on the timeline (Figure 6.(c)), reducing the burden of users to repeat the same query. In Figure 5.(d), RubySlippers informs users of the creation of bookmark and how to use it. When there are more than one bookmarks,

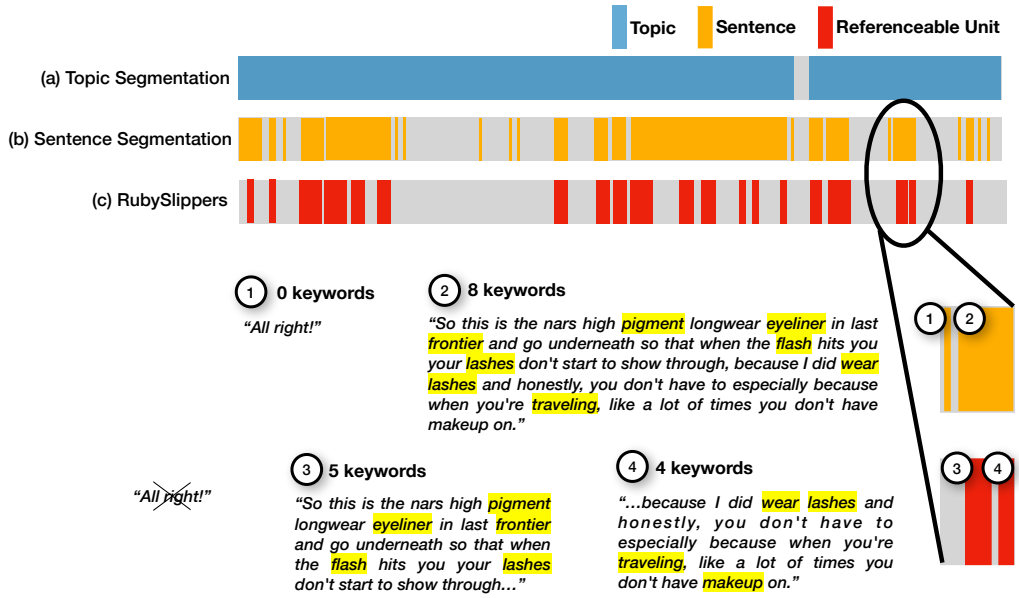


Figure 7: Our pipeline segments the transcript into units where the number of keywords are balanced. This pipeline is designed to make keyword-based queries efficient, in that more fine-grained searches are possible, and narrowing down the search with adding keywords is faster.

each is specified with the bookmark number to which users can refer to distinguish one from another.

4.7 Computational Pipeline

The computational pipeline that powers RubySlippers segments the transcript into units of that contain referenceable keywords to support keyword based querying and query updating. We highlight its three components: 1) video segmentation, 2) option population, and 3) keyword suggestion.

First, the pipeline pre-processes the transcript of the video and runs a part-of-speech tagger to pre-populate proper nouns, nouns, and verbs which correspond to objects and actions. The intuition is that they likely correspond to objects and actions, which the users can reference in navigation. Both the observations from our formative study and the prior work in cognitive psychology [43] show that people organize knowledge structures about an event around object and action as units.

We then split the video transcript into units which users can refer to with keywords and are in lengths containing no more than five keywords, making it easier to be understood at a glance. First, we use sentence-level segmentation using punctuation marks.

To recover from punctuation errors in the transcript, we run the BiRNN punctuator [39] over transcripts and further fix the punctuation marks. Two of the authors examined the resulting transcripts for corrected punctuation to further enhance the validity. Since the text in the transcripts in these how-to videos are informal and colloquial, the sentences are often incomplete sentences and grammatically incorrect. Therefore, off-the-shelf sentence segmentation techniques yield segments that are longer than typical sentences we expect. Thus, we use dependency relations to further split those that have more than 50 tokens. We run a dependency parser to find "conjunction words", and split long segments by using these conjunction words as delimiters.

Then we take each of the "clauses", count the number of "keywords" in it and split this long "clauses" into "referenceable scenes" where each scene contains at least two "keywords" but no more than five keywords.

To summarize, our resulting "referenceable scenes" may or may not by full sentences, and the segments might even be in the middle of a "sentence". The output of our method is illustrated in Figure 7.

For populating the search result, we use exact keyword matching. This means only the ones that containing maximum number of keywords in the query are populated and returned as search results. For populating the additional keyword suggestions in each scene for assisting the query update, we select and show the ones that significantly decrease the number of option scenes. If the query results have more than 12 scenes, adding one of these keyword suggestions will drop the number of scenes below 12. If the query results contain less than 12 scenes, then adding one of these keyword suggestions will drop the number of scenes below 4.

5 EVALUATION AND RESULTS

We evaluated the effectiveness and task load of RubySlippers through a lab study. Speficically, goals of our evaluations were (1) to assss the feasbility of keyword-based querying and updating as means of supporting "content-based navigation", (2) to assess the effect in task load the addition of "content-based referencing" to "temporal referencing", and (3) to gain feedback on the effectiveness

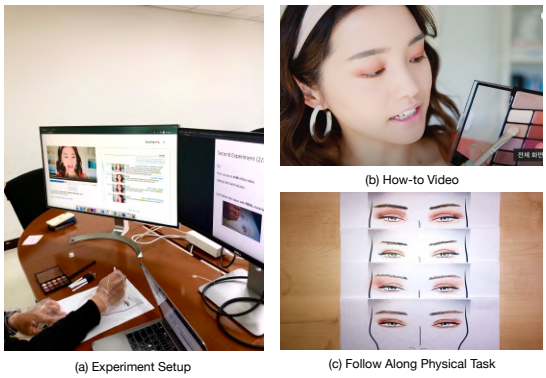
of RubySlippers in the following three representative navigational tasks.

(1) single target navigation for information seeking
(2) multi target navigation for information seeking
(3) following along the video while applying the instructions to the task at hand

We conducted a counterbalanced within-subjects study between, navigational support (temporal vs temporal + content-based), video domain type, and the order.

5.1 Participants

We recruited 12 participants (9 male, 3 female, mean age 22.75, stdev=2.45, max=27, min=19) through an online community posting. People who participated in the formative study were excluded from this recruitment. Each study session was 80 minutes long, and the participants were paid 20,000 KRW (~USD 17).



(a) Experiment Setup (b) How-to Video (c) Follow Along Physical Task

Figure 8: Participants were instructed to use only voice to control the video during tasks.

5.2 Study Procedure

We followed the safety guideline [12] to ensure safety during COVID-19. When participants arrived at the experiment location, they were asked to measure their body temperature and wear a mask. They were asked to sanitize their hands and wear gloves throughout the whole study process. They were asked to bring their own headset. All participants' body temperatures were within the normal temperature range. During the experiment, all windows were open and air conditioner was turned on to keep the room ventilated. We sanitized the room after each session.

After the safety check, participants were given a tutorial of the RubySlippers interface, and they were asked to go through a practice session to familiarize with the interface. Then the participants were asked to complete navigation tasks in two sessions. In each session they are assigned different domain of how-to video - one of baking or makeup - and different navigation condition - either temporal reference only or both temporal and content-based approach. Participants were instructed to use only voice to control the video during tasks. During the experiment, two different monitors were shown to the user (8.(a)): one showing the system, RubySlippers and the other displaying the task at the moment. We ran semi-structured interviews after the session to gain a deeper understanding of the strengths and weaknesses of RubySlippers and the decisions about specific referencing strategies participants employed, or interesting usage patterns. We recorded participants' screens and audio during each session upon their consent.

We selected four single-person video tutorials on YouTube shown in Table 3. The criteria for selecting videos include duration of the video, amount of information given verbally, and the availability of manually added captions. There were two sessions per participant and two domains of how-to videos were chosen: baking and makeup. For each domain we selected two videos. One how-to-bake video (Video 1-1 of Table 3) was used only for the practice session. Video 1-2 was used when the condition of the session was baking and both Video 2-1 and 2-2 were used for makeup condition. We chose two makeup videos instead of one because while all four videos are in almost the same length, the amount of transcript of the instructor varied such that the amount of textual information in two makeup tutorials were almost equivalent to that in one baking tutorial.

In each session, three different tasks were given to the participants. The first task type was a multi-target navigation task, in which participants were required to navigate to multiple scenes to find all the answers. For example, participants were asked to find answers to the question "how many and what type of brushes are used in the video?". Second task type was a single-target navigation task, in which we asked information about a frequently appearing object in the video. For example, participants were asked to find and explain what the chef does to expand the dough, while "dough" appears 16 times in the video. In the third task type, physical task, participants were asked to follow along some part of the video (about 90 seconds) with real ingredient and tools prepared by the researchers. For example, they were asked to apply eye makeup on the printed face on a sheet of paper. We prepared ingredients and tools needed to follow the cooking videos, and also makeup supplies needed to follow the makeup video. Participants were wearing plastic gloves at all times.

5.3 Results and Findings

Participants were experienced with baking while most were novices with makeup. As a result, the evaluation of RubySlippers revealed that content-based navigation not only helps experienced users but also novices. We summarize the quantitative results and present main findings with respect to the three design goals, usage patterns, and usability and usefulness of RubySlippers.

5.3.1 Quantitative Results. The results of the time taken for task completion and the number of interaction measurement in median are depicted in Table 4 and Table 5. A Wilcoxon Signed-Rank test was conducted to evaluate the effect of type of reference on the time taken and number of interaction made to complete the task. For all tests, an alpha level of 0.05 was used. For the number of interactions, we counted the number of command invocations participants made. We counted repeated invocations of the same command due to recognition failure as one invocation.

Participants used less number of interactions in TB+CB than in the baseline TB condition. For the multi-target search task, the average number of interactions were 8.75 (SD = 3.93) and 15.33 (SD = 6.33) for TB+CB and TB respectively. The pairwise difference was

Participant Number	Age	Gender	Prior Experience Watching How-to Videos		
			Makeup	Baking	Others
P1	19	F	YES	YES	Home Workout
P2	20	M	NO	NO	Arts and Crafts
P3	23	M	NO	NO	Programming
P4	25	M	NO	NO	X
P5	24	M	NO	YES	Arduino
P6	24	M	NO	YES	Piano
P7	27	M	NO	YES	PC Assembly
P8	23	F	YES	YES	Dance
P9	20	M	NO	NO	Guitar
P10	24	F	YES	YES	X
P11	24	M	NO	NO	Juggling
P12	20	M	NO	NO	Piano, Programming

Table 2: Background Information of Study Participants

Video ID	Title (Duration)	Domain	Creator	URL
Video 1-1	Amazing Caramel Cake(1051s)	Baking	Preppy Kitchen	[18]
Video 1-2	Amazing Hot Cross Buns(1083s)	Baking	Preppy Kitchen	[19]
Video 2-1	Drugstore Makeup Tutorial(926s)	Makeup	Jenn Im	[14]
Video 2-2	Passport Photo Makeup(936s)	Makeup	Roxette Arisa	[2]

Table 3: Four how-to videos used in the evaluation study.

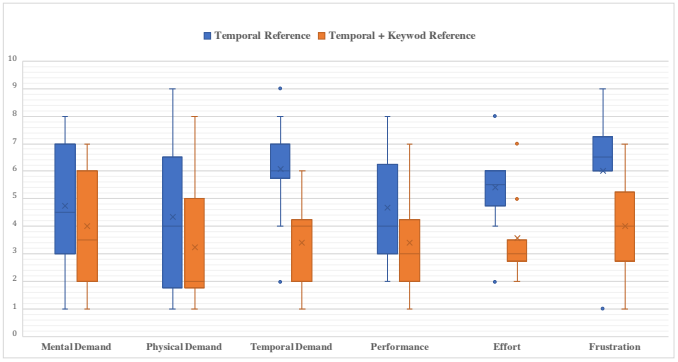


Figure 9: Boxplot of median NASA-TLX scores for the 2 reference conditions (0 = low, 10 = high)

significant (W = 5, p <0.005). There was no significant difference for remaining two tasks. For the frequently appearing object search task, the average number of interactions were 10.33 (SD = 4.72) and 10.42 (SD = 5.74) for TB+CB and TB respectively. For the follow-along physical task, the average number of interactions were 11.92 (SD = 5.73) and 14.25 (SD = 6.66) for TB+CB and TB respectively.

We did not observe any significant difference in time taken for task completion between TB+CB and the baseline TB condition. For the multi-target search task, the average number of interactions were 320.42 (SD = 119.67) and 282.92 (SD = 145.36) for TB+CB and TB respectively. For the frequently appearing object search task, the average number of interactions were 277.50 (SD = 175.09) and 305.42 (SD = 91.29) for TB+CB and TB respectively. For the follow-along physical task, the average number of interactions were 473.33 (SD = 180.97) and 457.08 (SD = 176.05) for TB+CB and TB respectively.

Figure 9 gives the box plot of the NASA-TLX scores using a median. We saw a statistically significant drop in temporal demand, effort, and frustration in the "temporal and content"condition. While not as significant, effort has also decreased in the "temporal and content"condition.

5.3.2 Qualitative Feedback. The feedback from participants was overall positive, with all participants agreeing that RubySlippers is useful for navigation. After having experienced both conditions, participants appreciated the ability to concisely express their intended navigation target using keywords. P11 said *"I could definitely see how it would be helpful. Especially having experienced content-based referencing first and then when I had to use temporal referencing only, I kept feeling the urge to use content-based referencing."*

D1. support for efficient content-based referencing using keywords. Participants found that content-based referencing is more robust in information seeking tasks. Participants in temporal referencing only condition had to rely mostly on visual cues, but missed many relevant timings and information while skipping around and fast-forwarding. Five participants had explicitly said that they're not sure if they had found all the information they were asked to find. On the other hand, participants in "temporal and content"condition were confident that found all the answers, and that they trust the system more. Specifically, P4 said, *"I can focus more on tasks not having to keep memory of timestamps nor the overall order of some independent events. I can use less commands to find what I want, and it leads to less delays! Apart from the (speech processing) delays and recognition errors I love the idea!"*

D2. support for effective query update. Participants found the idea of using composition of keywords for updating queries convenient, and that it helped them understand the content better. P6 had noted, *"It's useful to be able to search the combination of multiple keywords - not just for narrowing down the search but I could see how to objects are used together."* For example, when butter and cream are used together salt is also being used. While it was enough to use butter and cream to find the relevant information, participants also learned salt is another important ingredient used in the step. Also,

Video ID	Task Type	Time to Complete (sec)	
		Temporal	Temporal+Content
Video 1-2	1-a. Multi-Target Search	230	310
	1-b. Frequently Appearing Object Search	240	185
	1-c. Follow Along Physical Task	545	525
Video 2-1	2-a. Multi-Target Search	270	295
	2-b. Follow Along Physical Task	365	400
Video 2-2	2-c. Frequently Appearing Object Search	335	230

Table 4: Median time-to-completion in seconds per video per navigation task

Video ID	Task Type	Number of Interactions	
		Temporal	Temporal+Content
Video 1-2	1-a. Multi-Target Search	12	6
	1-b. Frequently Appearing Object Search	5	7.5
	1-c. Follow Along Physical Task	18.5	16
Video 2-1	2-a. Multi-Target Search	15	11.5
	2-b. Follow Along Physical Task	14	14
Video 2-2	2-c. Frequently Appearing Object Search	11	7.5
Total	Temporal: $(\overline{X} = 13.33, max = 29, min = 4, stdev = 6.44)$		
	Temporal + Content: $(\overline{X} = 10.33, max = 22, min = 2, stdev = 4.89)$		

Table 5: Median number of command invocations per video per navigation task and the summary of each condition

the participants found the search result items shown on the search panel on the right as structural dividers, and found it useful for navigation. P10 had said, *"It's just like video chapters (on description or command on YouTube) but in micro levels. I like it as it's simple and work as a shortcut."* The search results are the parts of the video that queried objects and actions appear. Participants found these to be meaningful markers that helped them understand whole task procedure of the entire video.

D3. support for informing users about executable commands and potential navigation. Participants also found the ambient help of displaying the available commands and potential keyword suggestion useful. P8 said, *"I sometimes said keywords that doesn't exist in the video, but I was able to quickly recover by looking at the keyword suggestion. It was super useful."* Similarly, P5 noted, *"Even when I cannot recall or do not know the exact name of the object, I searched for an action related to that object or a co-occurring object. Looking at the keyword suggestions in the options list, I could figure out the name and go to that scene!"*

Application to other types of how-to videos When asked what other types of how-to videos they think RubySlippers would be useful for, participants showed excitement and provided many interesting ideas. Many participants said long videos that they do not have the patience to watch would benefit from RubySlippers even if it is not a tutorial video.

Two participants mentioned how RubySlippers would be useful for lecture videos. P4 noted, *"I want to use this system for watching lecture videos when my hands are tied from note-taking. When the formula taught in earlier part of the video is used later and I cannot remember the exact equation, I can easily navigate backward and return."*

Participants noted that videos that have less clear structure would also benefit from RubySlippers. P1 said, *"Unlike baking or makeup which have strict orders of steps, it's hard to guess the time of random-order videos (home workout). So in those, it will be more useful!"*

Usage Patterns. Four participants used content-based referencing for navigating to a general area and temporal referencing for refinement when they were looking for an exact moment in the video. This is quite effective because the user is narrowing down the search space using keywords, and then pin pointing the exact scene with specific timestamp.

Also, participants used groups of results that are closed together on the timeline as a unit of navigation. For example, in the make-up video, a number of brushes appear. When searching for a specific brush in the video, participants used one word query "brush", and hypothesized the resulting scenes that are closer together are likely using the same brush, so they would examine one scene from each group to find the brush they waned.

The most interesting and unexpected finding is that P12 found content-based referencing helpful for understanding and learning the tutorial content. P12 had said, *"It would also help me to memorize the content of the video - Keyword-based referencing helps me to remember keywords and key concepts of the video much more than temporal-referencing."* This is particularly interesting, and warrants further investigation and future work, because we might be seeing effects of "self-explanation" [35] as a byproduct of efficient interaction design for voice navigation.

6 DISCUSSION, LIMITATIONS, AND FUTURE WORK

We discuss findings, generalizability, and possible limitations of this work.

6.1 Speech Recognition Errors

In our studies, participants experienced that the speech recognition errors and parsing delays are still a major roadblock in using VUI. This was expected [28]. In this work, we aimed to show how content-based navigation using keywords can address the invisible nature of VUI. Therefore, during the analysis, we counted repeated utterances due to speech recognition failure as one utterance in measuring the number of interactions. This was to focus on counting users' attempts and updates of navigation by ruling out the recognition failure, a relatively inconsistent variable among users, while measuring time-to-completion to reflect the effect of speech recognition errors and parsing delays.

Acknowledging the limitation of the speech recognition technology, some participants have suggested alternatives like gaze or gesture for controlling the video.

6.2 User Confidence and Trust in Content-based Referencing

Participants reported they felt more confident when navigating and trusted the query results more when using content-based referencing than when using temporal referencing. We hypothesize this is a result of our pipeline using exact keyword matching: the results are binary, either the keyword is in the search result or it isn't. Scenes shown in RubySlippers are guaranteed to contain the keyword, which helps participants build a clear mental model of what to expect from a query result.

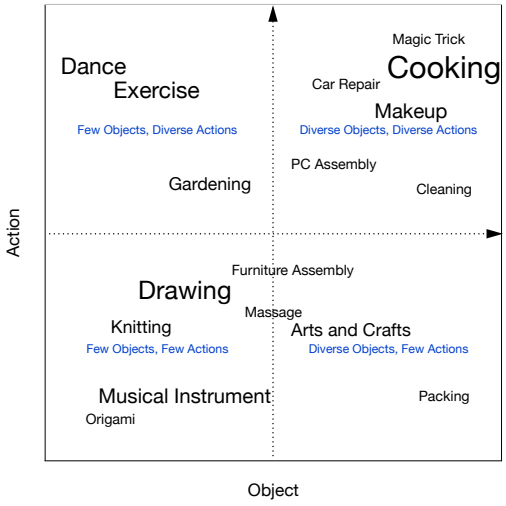


Figure 10: Analysis of most searched how-to video domains with respect to a variety of objects and actions (size logarithmically proportional to popularity)

6.3 Supporting the Broader Landscape of How-to Videos

Objects and actions that appear in the videos are essential ingredients of keyword based navigation. To see how our approach can generalize to other types of how-to videos, we performed a breadth-first random sampling of the popular how-to video domains in order to classify them according to a variety and scale of objects and actions in the video.

We explored top three hundred most viewed video tutorials when searched with the query "how to step by step" on YouTube and distilled 17 different domains of how-to video involving physical activity. For each domain of how-to videos, we examined three videos with similar length (around 10 minutes). We counted and averaged the number of objects and actions appearing in these videos while regarding body parts as objects.

We classified the styles of how-to videos with respect to the two dimensions: object and action (Figure 10). Our keyword-based approach works best with domains located in the first quadrant. For videos that have lots of keywords and lots of actions, users gain confidence about being able to make this random access anchoring around the keywords.

However, for videos in the third quadrant that have less objects and less actions still remain unexplored with the challenge of extracting referenceable term from the transcript. For example, in the video tutorial for origami, the words "corner, edge, fold, crease" are repeatedly used throughout the video, thus making the visual comparison inevitable.

With taking visual content into account, one possible line of future work is to investigate how adopting object detection and optical character recognition can expand the set of referenceable items in the absence of verbal descriptions. For example, when the person in the video shows an important item and refers to it using pronouns like "this", and has the keyword as graphic element on the scene for dramatic effect, our current pipeline cannot detect the keyword despite its importance. With the advancement of computer vision algorithms, we expect visually computed components to be promising additions.

6.4 Leveraging Interaction at Scale

There are two dimensions in which we can consider scalability. One of them is facilitating multiple users' interaction on each video. Participants have suggested that they would be interested in seeing other viewers' querying history. Leveraging interaction traces like keywords and navigation patterns opens up opportunities for further advancing navigation interaction and successful examples in other domains include LectureScape [16] and Patina [24].

The other dimension is hyper-personalization for each user. Once personal history accumulates, the system can infer what types of queries this specific user initiates, and possibly aim to understand the user struggle. Similar approaches have been explored in understanding the usability issues of software [11].

6.5 Transferring to Voice Assistants

The user scenario that involves how-to videos by design includes a visual display. We leveraged this fact, and were able to design effective visual pointers that guided users to navigate how-to videos.

However, we noticed one participant who had some familiarity with the task at hand successfully navigating without looking at the screen, but only verbally issuing commands. This was particularly interesting because it allows us to understand how to generalize the design of referencing strategies so it would be applicable to voice assistants without screens.

It would be a meaningful extension of this research to take a deeper look at what micro interactions are possible without the screen, and what the conditions are for successfully navigating a tutorial without visual display.

6.6 Is Mouse Interaction the Holy Grail?

The focus of our study was to design and implement voice-based content-based referencing for video interface. We did not compare RubySlippers against direct manipulation methods like mouse or keyboard as they correspond to different user scenarios and environments. However, whether using temporal referencing and content-based referencing in voice user interfaces is as efficient as interacting directly with the timeline using the mouse is still an interesting question. If not, how do we design voice interactions so that they are as efficient? Perhaps can voice interfaces be more efficient than direct control if natural language processing and speech processing techniques advance? They are meaningful interaction challenges, and we hope this research will catalyze future work in designing the next voice interfaces.

7 CONCLUSION

This paper presents RubySlippers, a voice-based navigation system for how-to videos. RubySlippers supports efficient content-based voice navigation through keyword-based queries. Our user study demonstrates that RubySlippers provides efficient, stress-free navigation for how-to videos in voice user interfaces.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2020R1C1C1007587).

REFERENCES

[1] Abir Al-Hajri, Gregor Miller, Matthew Fong, and Sidney S Fels. 2014. Visualization of personal history for video navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1187–1196.

[2] Roxette Arisa. 2020. MAKEUP FOR PASSPORT PHOTOS/ID PICTURES *no flashback, smooth skin* | Roxette Arisa. https://www.youtube.com/watch?v=9qoDdXFwBdo

[3] Zahra Ashktorab, Mohit Jain, Q Vera Liao, and Justin D Weisz. 2019. Resilient chatbots: repair strategy preferences for conversational breakdowns. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[4] Morteza Behrooz, Sarah Mennicken, Jennifer Thom, Rohit Kumar, and Henriette Cramer. 2019. Augmenting Music Listening Experiences on Voice Assistants.. In *ISMIR*. 303–310.

[5] Minsuk Chang, Anh Truong, Oliver Wang, Maneesh Agrawala, and Juho Kim. 2019. How to design voice based navigation for how-to videos. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.

[6] Ming-Ming Cheng, Shuai Zheng, Wen-Yan Lin, Vibhav Vineet, Paul Sturgess, Nigel Crook, Niloy J Mitra, and Philip Torr. 2014. ImageSpirit: Verbal guided image parsing. *ACM Transactions on Graphics (TOG)* 34, 1 (2014), 1–11.

[7] Eric Corbett and Astrid Weber. 2016. What can I say? Addressing user experience challenges of a mobile voice user interface for accessibility. In *Proceedings of the 18th international conference on human-computer interaction with mobile devices and services*. 72–82.

[8] Chris Crockford and Harry Agius. 2006. An Empirical Investigation into User Navigation of Digital Video Using the VCR-like Control Set. (2006), 340–355.

[9] Wei Ding and Gary Marchionini. 1998. *A study on video browsing strategies*. Technical Report.

[10] Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. 2008. Video browsing by direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 237–246.

[11] Adam Fourney, Richard Mann, and Michael Terry. 2011. Characterizing the usability of interactive applications through query log analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1817–1826.

[12] HM Government. [n.d.]. Working safely during COVID-19 in labs and research facilities. https://assets.publishing.service.gov.uk/media/5eb9752086650c2799a57ac5/working-safely-during-covid-19-labs-research-facilities-200910.pdf.

[13] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[14] Jenn Im. 2018. Everyday Drugstore Makeup Tutorial. https://www.youtube.com/watch?v=09HfTthoGEw

[15] Juho Kim. 2013. Toolscape: enhancing the learning experience of how-to videos. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 2707–2712.

[16] Juho Kim, Philip J Guo, Carrie J Cai, Shang-Wen Li, Krzysztof Z Gajos, and Robert C Miller. 2014. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 563–572.

[17] Yea-Seul Kim, Mira Dontcheva, Eytan Adar, and Jessica Hullman. 2019. Vocal shortcuts for creative experts. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[18] Preppy Kitchen. 2019. Amazing Caramel Cake Recipe. https://www.youtube.com/watch?v=CHbrXX23cto

[19] Preppy Kitchen. 2020. Amazing Hot Cross Buns Recipe. https://www.youtube.com/watch?v=XCf2zZ-_Swo

[20] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From Word Embeddings to Document Distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* (Lille, France) *(ICML'15)*. JMLR.org, 957–966.

[21] Benjamin Lafreniere, Tovi Grossman, and George Fitzmaurice. 2013. Community enhanced tutorials: improving tutorials with multiple demonstrations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1779–1788.

[22] Gierad P Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, Jason Linder, and Eytan Adar. 2013. Pixeltone: A multimodal interface for image editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2185–2194.

[23] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2012. Swift: Reducing the Effects of Latency in Online Video Scrubbing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 637–646. https://doi.org/10.1145/2207676.2207766

[24] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2013. Patina: Dynamic heatmaps for visualizing application usage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3227–3236.

[25] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2013. Swifter: Improved Online Video Scrubbing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 1159–1168. https://doi.org/10.1145/2470654.2466149

[26] Sarah McRoberts, Joshua Wissbroecker, Ruotong Wang, and F Maxwell Harper. 2019. Exploring Interactions with Voice-Controlled TV. *arXiv preprint arXiv:1905.05851* (2019).

[27] Christine Murad, Cosmin Munteanu, Leigh Clark, and Benjamin R Cowan. 2018. Design guidelines for hands-free speech interaction. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. 269–276.

[28] Chelsea Myers, Anushay Furqan, Jessica Nebolsky, Karina Caro, and Jichen Zhu. 2018. Patterns for how users overcome obstacles in voice user interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–7.

[29] Chelsea M Myers. 2019. Adaptive suggestions to increase learnability for voice user interfaces. In *Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion*. 159–160.

[30] Cuong Nguyen and Feng Liu. 2015. Making Software Tutorial Video Responsive. In *CHI*.

[31] Jakob Nielsen. [n.d.]. 10 Usability Heuristics for User Interface Design.[Online] 1995.

[32] Amy Pavel, Dan B Goldman, Björn Hartmann, and Maneesh Agrawala. 2015. Sceneskim: Searching and browsing movies using synchronized captions, scripts

and plot summaries. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 181–190.

[33] Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala. 2014. Video Digests: A Browsable, Skimmable Format for Informational Lecture Videos. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. Association for Computing Machinery, New York, NY, USA, 573–582. https://doi.org/10.1145/2642918.2647400

[34] Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. 2011. Pause-and-play: Automatically Linking Screencast Video Tutorials with Applications. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. ACM, New York, NY, USA, 135–144. https://doi.org/10.1145/2047196.2047213

[35] Marguerite Roy and Michelene TH Chi. 2005. The self-explanation principle in multimedia learning. *The Cambridge handbook of multimedia learning* (2005), 271–286.

[36] Yale Song, Miriam Redi, Jordi Vallmitjana, and Alejandro Jaimes. 2016. To click or not to click: Automatic selection of beautiful thumbnails from videos. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 659–668.

[37] Arjun Srinivasan, Mira Dontcheva, Eytan Adar, and Seth Walker. 2019. Discovering natural language commands in multimodal interfaces. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 661–672.

[38] Daniel Steinbock. [n.d.]. TagCrowd. https://tagcrowd.com.

[39] Ottokar Tilk and Tanel Alumäe. 2016. Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration. In *Interspeech 2016*.

[40] Xu Wang, Benjamin J. Lafreniere, and Tovi Grossman. 2018. Leveraging Community-Generated Videos and Command Logs to Classify and Recommend Software Workflows. In *CHI*.

[41] Kuldeep Yadav, Kundan Shrivastava, S Mohana Prasad, Harish Arsikere, Sonal Patil, Ranjeet Kumar, and Om Deshmukh. 2015. Content-driven multi-modal techniques for non-linear video navigation. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*. 333–344.

[42] Matin Yarmand, Dongwook Yoon, Samuel Dodson, Ido Roll, and Sidney S Fels. 2019. " Can you believe [1: 21]?!" Content and Time-Based Reference Patterns in Video Comments. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[43] Jeffrey Zacks, Barbara Tversky, and Gowri Iyer. 2001. Perceiving, remembering, and communicating structure in events. *Journal of experimental psychology. General* 130 (04 2001), 29–58. https://doi.org/10.1037//0096-3445.130.1.29

[44] Han Zhang, Maosong Sun, Xiaochen Wang, Zhengyang Song, Jie Tang, and Jimeng Sun. 2017. Smart jump: Automated navigation suggestion for videos in moocs. In *Proceedings of the 26th international conference on world wide web companion*. 331–339.