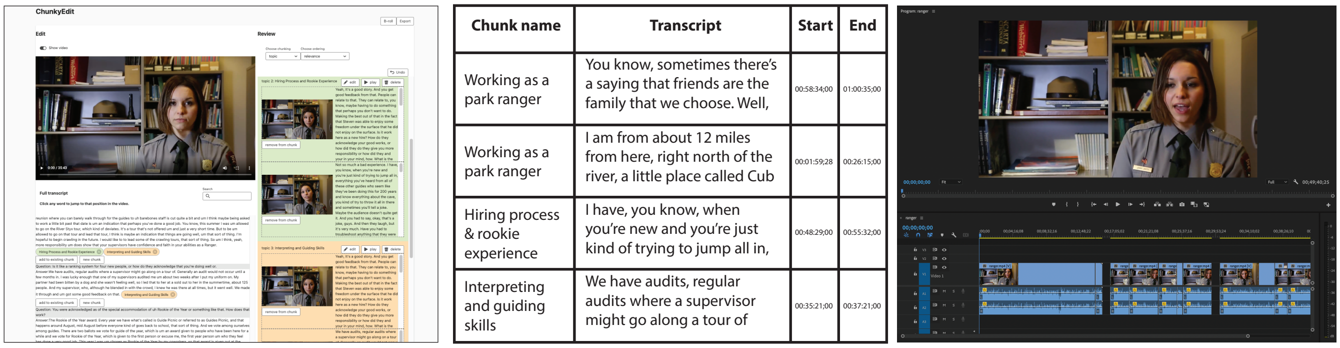
ChunkyEdit: Text-first video interview editing via chunking

Mackenzie Leake
leake@adobe.com
Adobe Research
San Francisco, California, USA

Wilmot Li
wilmotli@adobe.com
Adobe Research
Seattle, Washington, USA



Chunk name	Transcript	Start	End
Working as a park ranger	You know, sometimes there's a saying that friends are the family that we choose. Well,	00:58:34;00	01:00:35;00
Working as a park ranger	I am from about 12 miles from here, right north of the river, a little place called Cub	00:01:59;28	00:26:15;00
Hiring process & rookie experience	I have, you know, when you're new and you're just kind of trying to jump all in,	00:48:29;00	00:55:32;00
Interpreting and guiding skills	We have audits, regular audits where a supervisor might go along a tour of	00:35:21;00	00:37:21;00

ChunkyEdit UI — Paper edit — Video stringout

Figure 1: ChunkyEdit takes a video interview as input. It then transcribes the video and groups the transcript into "chunks" based on similar questions or themes in the interviewee's answers. Users can explore these chunks and define their own in the ChunkyEdit UI (left) before exporting a "Paper edit" (center) or a "Video stringout" (right), two common intermediate formats.

ABSTRACT

The early stages of video editing present many cognitively demanding tasks that require editors to remember and structure large amounts of video. In our formative work we learned that editors break down the editing process into smaller parts by labeling and organizing footage around central themes. Using current video editing tools, this process is slow and largely manual. We present a system called ChunkyEdit for helping editors group video interview clips into thematically coherent chunks, which can then be exported to existing video editing tools and composed into an edited narrative. By focusing on this intermediate step, we leverage computation to do tedious organizational tasks, while preserving the editor's ability to control the primary storytelling decisions. We explore four different topic modeling approaches to creating video chunks. We then evaluate our tool with eight professional video editors to learn how a chunking-based approach could be incorporated into video editing workflows.

CCS CONCEPTS

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

KEYWORDS

Video editing, Chunking, Topic modeling applications

ACM Reference Format:
Mackenzie Leake and Wilmot Li. 2024. ChunkyEdit: Text-first video interview editing via chunking. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3613904.3642667

1 INTRODUCTION

In the days of film editing, the cost of acquiring videos was far higher, and therefore production teams were more selective in what they captured. As digital cameras have increased the scale of captured video, the organizational demands of video editing have also increased. In order to produce a compelling, coherent narrative, editors must watch hours of input footage and identify the most relevant parts, which involves remembering the content, tone, and quality of the inputs and the state of the current edit.

Building deep familiarity with the content through watching and re-watching the videos can inform the writing and shaping of the narrative, but simply keeping all of these moving parts in one's mind is a nearly impossible task [53]. The psychology literature tells us that there is a limit to how much information people can store in their working memory [45, 47, 48, 63]. One strategy for extending human memory is through the technique of *chunking*, which involves organizing raw information into logical groups that reduce the number of distinct entities humans need to consider [43, 60]. This strategy has been explored across many different types of applications, including visual memory [45], verbal recall [6], and games, such as chess [13, 26]. Our formative work indicates that video editors also apply a form of chunking in the early stages of editing (Fig. 2). They use textual representations, such as shot

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

ACM ISBN 979-8-4007-0330-0/24/05…$15.00
https://doi.org/10.1145/3613904.3642667

lists or transcripts, to identify relevant sections of video footage. Then, as they make further editing decisions about the narrative structure, they group clips according to themes and arrange these sections in an intermediate text-only format called a **paper edit**. Editors also group thematically-related sequences of clips into a single timeline to create a **stringout** that they can watch and share with collaborators for feedback. The paper edit and the stringout organize the raw footage into candidate "chunks" that help editors assemble a **rough cut** — i.e., a draft of the final edited film — without having to keep every meaningful moment of footage in their minds.

Producing a paper edit and stringout and translating these intermediate representations into a rough cut are time-consuming tasks with existing editing tools. For scripted content, editors have to repeatedly play different video takes, or versions of the recording, to pick the best performances that fit together to maintain a consistent tone. For unscripted content, editors have to review large amounts of footage to identify emergent themes and collect all of the relevant clips. Existing tools do not represent footage at the granularity of "chunks," so editors have to manually scrub, playback, trim, and label captured footage to create chunked representations (i.e., paper edits and stringouts).

Our main insight leverages the observation that editors manage the large space of potential editing decisions by focusing on grouping clips thematically. Inspired by the related notion of "chunking" from the cognitive science literature, we call these thematic groups of clips "chunks." We focus specifically on video interviews: a particular type of video that is quite common across news, documentaries, and even some scripted films, and we incorporate an understanding of the way interviews are written, filmed, and edited into the design of our tool. While the content of video interviews varies widely, the structure of an interview follows natural conversation. We introduce *ChunkyEdit*, a tool that directly supports chunking to facilitate early stage editing tasks by leveraging advances in content understanding. Our work makes four main contributions:

(1) An automatic video segmentation and labeling pipeline for video interviews
(2) Prompt engineering and algorithmic approaches for producing video chunks
(3) An interface that supports chunk-based editing
(4) An evaluation of how chunk-based editing may be utilized by video editors

We demonstrate the effectiveness of our system by generating video chunks for 12 interview videos (4-81 mins., 6-33 pairs of questions and answers) and collecting feedback from 8 professional video editors about how ChunkyEdit could fit into their existing workflows. Our work indicates that there are several potential benefits to focusing on chunks. For example, chunking matches what many experienced editors already do and helps them focus on high priority parts of the videos. It can also facilitate the feedback and review process with other stakeholders, such as clients and producers, in the early stages of video editing. Throughout the rest of this paper, we motivate the design of ChunkyEdit through our review of the video production literature and interviews with editors, describe the techniques that identify relevant chunks within video interviews, and discuss how editors would incorporate ChunkyEdit into their video editing process.

2 INTERVIEW EDITING BACKGROUND

To deepen our understanding of the interview editing process, we conducted three one-hour formative interviews with professional editors (FE1, FE2, FE3) and reviewed three video editing books [7, 21, 23] recommended to us by these editors. Each of our interviewees has least 8 years of video editing experience producing interviews. FE1 works on television documentaries, involving interviews with on and off-camera interviewers, FE2 produces employee profile feature videos for a large company, and FE3 creates profiles of academic researchers at a university. Despite the differences in the types of interviews the editors work on, we found common themes, which we discuss below, in the challenges of assembling video sequences and getting feedback on intermediate versions.

2.1 Conducting an interview

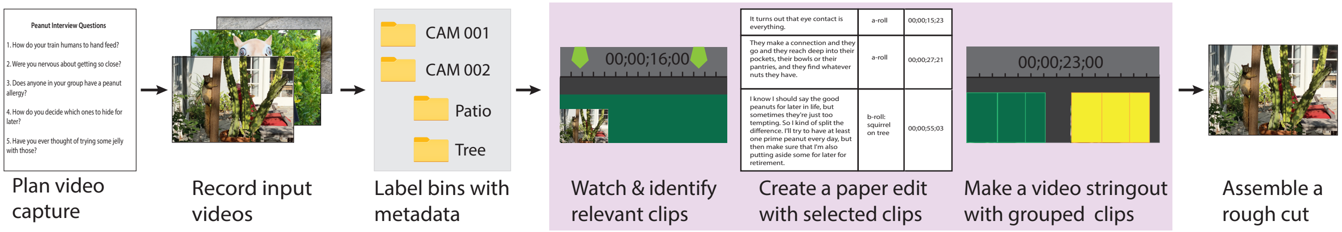
In a typical video interview, there is one **interviewer** and at least one **interviewee** and at least one camera recording the conversation. Even if the interviewer will not appear on camera in the final edit, their voice asking the questions is recorded. All editors (FE1-FE3) said that typically the interview questions are written ahead of time. However, the interviewer will sometimes choose to change the question order during the recording in order to make the conversation flow more naturally. The interviewer may also reiterate a question to give the interviewee a chance to rephrase their answer to make it easier to understand in the later edit (e.g., if the interviewee cuts off their answer mid-sentence or phrases something awkwardly) [7]. Interviewers may also ask follow up questions to get the interviewee to elaborate upon an interesting point. Repeated questions may appear at any point in the recording, but follow up questions typically appear immediately after one another. A good interviewer thinks about the final output edit when writing and verbalizing the questions during the interview.

While in some cases the editor is involved in the interview question writing and filming, it is more common for the editor to receive videos from other members of the production team. Current video editing software does not organize videos by question, though this is a common task for editors to do manually, which is particularly challenging if they were not present for the interview recording. If we take into account this structure during the editing process, then we can help editors know where to focus their attention. Based on this structure, we can incorporate the following design guidelines into a tool to support editing interviews:

DG1 **Organize interview by question**: Make it easy to go through the video by questions.
DG2 **Group repeated questions**: Allow editors to easily consider answers to reiterated questions together.
DG3 **Group follow up questions**: Group follow up questions with related questions to find similar content.

2.2 Creating a paper edit

Once an interview has been recorded, an editor begins the task of editing the content into a watchable form. Sheila Curran Bernard, an award-winning filmmaker, states that there are two reasons to edit an interview: *"to focus information for placement in the best possible location in your film's story and to shorten it"* [7]. Shortening



Peanut Interview Questions

1. How do your train humans to hand feed?
2. Were you nervous about getting so close?
3. Does anyone in your group have a peanut allergy?
4. How do you decide which ones to hide for later?
5. Have you ever thought of trying some jelly with those?

It turns out that eye contact is everything.	a-roll	00;00;15;23
They make a connection and they go and they reach deep into their pockets, their bowls or their pantries, and they find whatever nuts they have.	a-roll	00;00;27;21
I know I should say the good peanuts for later in life, but sometimes they're just too tempting. So I kind of split the difference. I'll try to have at least one prime peanut every day, but then make sure that I'm also putting aside some for later for retirement.	b-roll: squirrel on tree	00;00;55;03

Figure 2: The editing process begins far ahead of placing videos on a timeline. A video creator first plans for video capture, often producing artifacts like a shot list or a set of interview questions. Next, they record their videos and import these into video editing software, manually grouping clips into folders called bins based on low-level metadata, such as the camera or recording location. Then, they start the process of breaking down the larger video editing task into individual editing decisions (purple region) by watching and identifying relevant clips and creating intermediate editing formats, such as a paper edit and a video stringout. These are critical organizational steps toward assembling a full edited sequence for a rough cut.

is a particularly important part of the process, as *"a person will talk to you for 10 minutes, an hour, or maybe two or three hours, and you'll end up using only a few bites at most" [7]*. Many editors, including FE1, produce what is called a **paper edit**, which involves using transcripts to make content selections. The paper edit typically includes a description of the visuals that will appear, a transcript selection, and some timing information from the videos. Producing a paper edit can be an important part of helping the editor discover what content is within the videos and carve out the structure of the edit. Editors typically create paper edits in document editing tools that are separate from the video editing software. The importance of creating paper edits leads to the following design goal:

DG4 **Help editors generate a paper edit**: Make it easy to gather and label selected parts of an interview.

2.3 Building video sequences

Professor of video production Donald Diefenbach highlights one of the main challenges of the early stages of editing: managing the large scale of the input footage: *"The permutations of constructing the order and duration of shots in a program are virtually infinite, and there is no single best way to build a visual program"* [21]. In practice, most early editing passes involve working at the line or at minimum the sentence level. Bernard says, *"Don't make the editor crazy by cutting out every third word and expecting her to construct a sentence or a paragraph out of the bits and pieces"* [7].

To help manage the large volume of input footage, editors typically focus on generating **sequences**, which are distinct sub-parts of the overall edit. FE2 and FE3 said that they typically build these sequences in order, grouping clips thematically as they work through the footage in its original order. FE1, however, typically starts with the part of the interview that is most interesting or central to the story and then continues building out thematic sequences, using the transcript to find related parts in various sections of the input videos (Fig. 3). Editors will sometimes color code these different sequences or make some notes in a document to describe the themes. However, building sequences is hard with existing tools because editors have to manually identify key themes and then find all of the relevant clips for each theme. The labeling strategies editors use lead to the following design goals:

DG5 **Group clips thematically**: Help editors identify and label themes in videos.
DG6 **Identify central parts of the interview**: Help editors focus on the main topics in interviews.
DG7 **Support line level cuts**: Help editors work at a coarse level of granularity in early edits.

2.4 Creating shareable stringouts

An important part of the editing process is eliciting feedback from fellow members of the production team. These "screenings" can be highly informal, such as having a fellow editor stop by the editing room as an editor is working, or formal events with test audiences. All of our editors, FE1-FE3, mentioned the challenges of knowing what to show to best reflect their progress and give a sense of what the final edit will look like. One common strategy for early stage screenings with other members of the production team is showing several potential sequences that will appear in the final edit. Bernard suggests including "moments that affect you in some way, whether emotionally or intellectually. Look for scenes and sequences that can play on their own and interview bites that seem strong and clear" [7]. FE3 echoed this sentiment, often showing highly emotional and surprising aspects of the future film during the early screening sessions. This editor typically collects these sequences along a single video editing timeline in what is called a **stringout** and plays these different sections to get feedback. The stringout helps editors make progress toward a **rough cut**. Bernard defines a rough cut as *"a draft of your film that is significantly longer than the final show will be. But your general story and structure are in place, and you have some, if not all, of your elements on hand. The rough cut stage is often the best time to reassess major issues of story and structure and experiment with alternatives; this becomes more difficult as the film is fine tuned"* [7]. Karen Everett, a filmmaker and documentary editing consultant, advises filmmakers to *"establish your film's storytelling grammar"* through the use of placeholders for any content that is missing or temporary in the screening version and providing a transcript for attendees so they can follow along and make notes [23]. Therefore, in screenings editors will sometimes add titles and temporary b-roll (i.e., video and images that help illustrate the narrative but are not the main video) or graphics to edited chunks to give a flavor of what the final

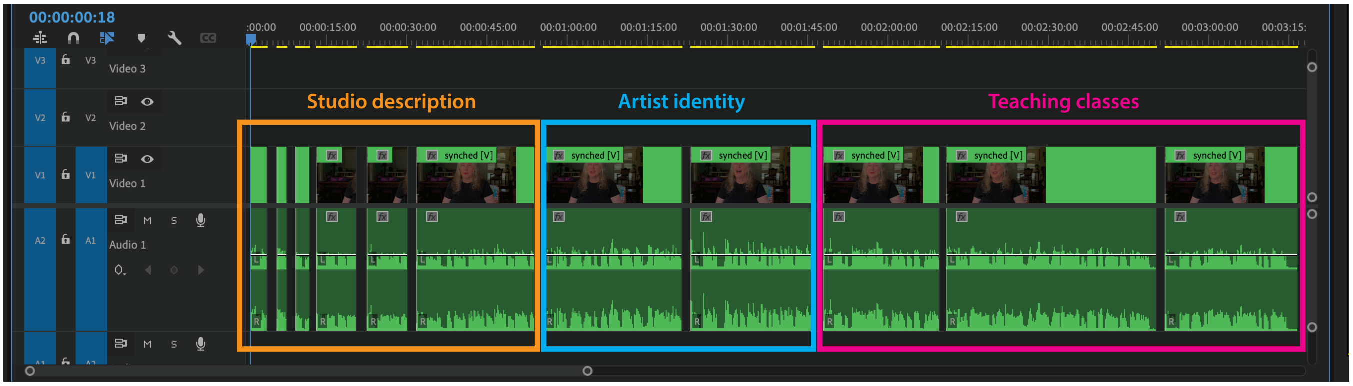


Figure 3: FE1 shared an image of their Adobe Premiere Pro timeline for a recent project. They arranged clips, grouped thematically, back-to-back on a video timeline, to produce a stringout.

stylistic elements will be. Based on this information about creating shareable content, we find that the following guidelines would be helpful for editors:

DG8 **Produce stringouts**: Produce intermediate formats that can easily be shared during a screening.
DG9 **Provide placeholders for content**: Make it easy for editors to add temporary content, such as b-roll.

Based on these insights from our review of interview editing books and our conversations with editors, we built ChunkyEdit to facilitate the early parts of the editing process.

3 RELATED WORK

Our work builds upon prior work that uses computation to help people more effectively and efficiently edit videos across a range of domains. We first review related tools and techniques for editing and reviewing video footage and discuss prior work in other applications of topic modeling, which is the main computational technique ChunkyEdit uses for producing video chunks. Then, to connect video editing and the notion of chunking from the cognitive science literature, we provide an overview of chunking.

3.1 Tools for video editing and review

Prior work has introduced a wide range of techniques for automating the editing of videos across different domains, including unscripted meetings and social gatherings [4, 64], advertising videos [16], video montages, [67], narrated videos [65], physical demonstration and tutorial videos [14, 15], scripted conversational scenes [37], crowdsourced concert videos [59], and horror movie trailers [62]. While some of these computational video editing tools (e.g., [5, 50, 55]) take a data-driven approach to making editing decisions, others manually encode heuristics specific to their application domain to make editing decisions. For example, Leake et al., 2017 [37] use cinematographic idioms, such as showing the visible speaker, to guide the editing of conversations; Arev et al., 2014 [4] use saliency to guide shot selection in social videos; Girgensohn et al., 2000 [24] use manually encoded rules for shot length, camera motion, and brightness to select suitable clips; Huber et al., 2019 use keywords to guide the selection of b-roll video in narrative videos [31]; Davis, 2003 [19], Schofield et al., 2015 [59], and Kim et al., 2015 [33] use video templates to guide shot capture and selection. Many of these tools, including Silver [10], SceneSkim [51], and QuickCut [65], utilize transcript-based interfaces for connecting the content within spoken audio in the video. Like these other tools, ChunkyEdit encodes domain-specific rules into the editing process and uses a transcript-based approach to editing. However, it focuses specifically on interviews, which are unscripted dialogue-based videos that appear in a wide variety of output video formats. By supporting exploration of thematically related questions and answers in the transcript, ChunkyEdit provides specialized support for interview editors beyond what general-purpose timeline or text-based video editing tools provide.

While many tools focus on helping an editor assemble video sequences, several systems have focused on evaluating input videos automatically for common issues, such as visual continuity [54] and stability [28]. Video Lens [40] helps users visualize and explore collections of videos and video metadata. Others have focused on more social and collaborative aspects of the video making process, such as building interfaces to support video review and feedback [46, 52]. Likewise, several commercial tools, such as Frame.io [1], focus on collaborative video editing and review. While ChunkyEdit shares a similar goal of improving the intermediate stages of the video editing process and facilitating video review, it does so through the creation of labeled video chunks, rather than a complete edit.

3.2 Human-in-the-loop topic modeling

Topic modeling, which involves identifying common themes within text documents, has been applied to texts across a wide range of domains and remains an active area of natural language processing (NLP) research [18]. While topic modeling is often used to categorize large collections of static documents, it has also been used in closer-to-real-time scenarios, such as taking notes and organizing meeting agendas [11] and annotating and visualizing social media data [17]. Most topic modeling methods are fully automatic, and this can create situations in which the resulting topic clusters are hard for people to understand and label [12, 44]. Human-in-the-loop and interactive topic modeling allow users to iterate on the set of chosen topics, which can make the resulting text groups and their labels easier to understand [29, 30, 32, 38, 61]. Recent work has explored using large language models, such as ChatGPT, to help domain experts label clusters [57]. ChunkyEdit is inspired by these combinations of automatic and human-in-the loop topic modeling approaches. It presents users with a range of topic modeling criteria based on the interview questions and answers and allows users to adjust the names and composition of topic chunks.

3.3 Chunking

The notion of chunking has been studied in the psychology and cognitive science community in the context of a wide range of applications, such as problem solving [20, 36] and perception and memory tasks [26, 43]. Chunking describes how people break down a stream of information into manageable size units of information and use this structure to process information effectively [26]. A chunk can be described as *"a collection of elements having strong associations with one another, but weak associations with elements within other chunks"* [26]. Chunking can be 'goal-oriented,' involving a conscious effort to break down a problem or 'perceptual,' involving a more automatic process [26]. Prior work in the HCI community has explored how to identify cognitive chunks from user interface usage patterns [58] and how to use the idea of chunking to bridge the gap between novice and expert user patterns in user interfaces [9]. To our knowledge, chunking has not been studied in the context of video editing, though it has been explored in a number of other similar communication and creative tasks, such as teaching communication skills [8], learning sequences of facts [34], remembering symbolic drawings [22], and free-hand sketching [39]. In this work we explore how incorporating chunking into the video editing process can facilitate the production and sharing of intermediate video editing steps. Our approach of breaking down editing into smaller decisions and helping editors structure their thinking is inspired by the notion of chunking in the psychology literature, and we use the term "video chunks" to describe the collections of thematically grouped video clips.

4 UI OVERVIEW

The goal of ChunkyEdit is to identify a set of thematic chunks within an interview video and allow editors to iterate on these selections as they work toward a rough cut. The ChunkyEdit UI (Fig. 4) has two main panels for editing and reviewing and then allows the user to export a paper edit or stringout for further editing.

4.1 Editing panel

Users begin by uploading their video, which is automatically transcribed. In the main Editing Panel (Fig. 4a) users see a large video player, which allows them to watch the input interview video. They can optionally toggle the show video button to off (Fig. 4b) to suppress the video if they want to focus on producing a text or audio edit without being distracted by the video playback. Below the player is the the full interview transcript (Fig. 4c). The interview transcript is automatically divided into questions, highlighted in gray, and answers, which appear in white (Fig. 4d). The user can click any of the words within the transcript to jump to the corresponding part of the video. Colored labels indicate an answer's membership in that automatically labeled chunk based on the current chunking strategy (Fig. 4e). Users can remove a particular chunk label by clicking the 'x' button within the colored label. Users can manually add a particular answer to an existing chunk label by clicking the 'add to existing chunk' button. They can also initiate a new chunk by clicking the 'new chunk' button next to an answer. They will then see a text box that allows them to enter the name of that chunk. The editing panel also has a search bar that allows users to search the content of the transcript for relevant sections (Fig. 4f). After searching, an 'add topic chunk' button appears, which allows users to create a new chunk with the name of the search term as the label and the most relevant interview answers as initial members of the chunk.

4.2 Review panel

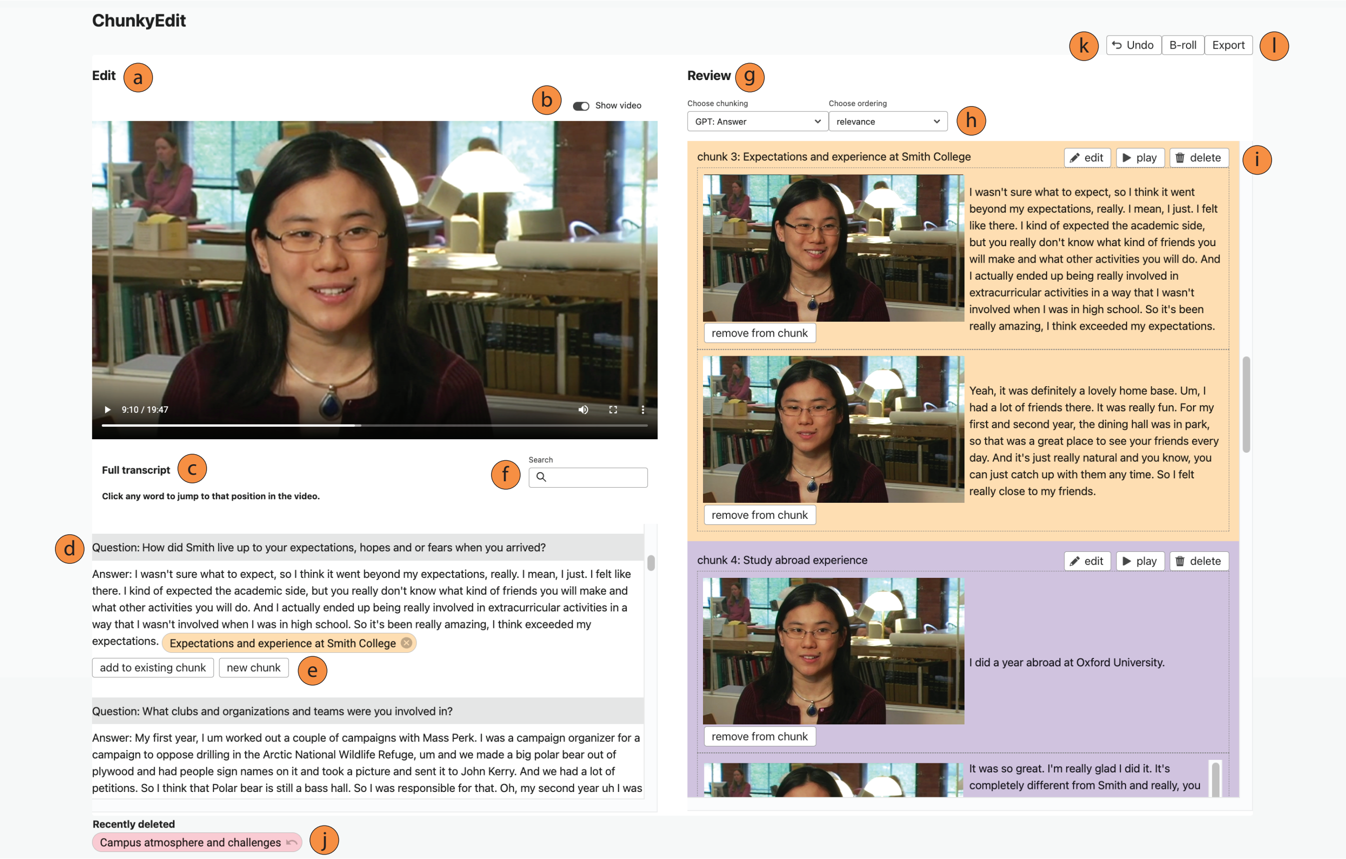
In the Review Panel (Fig. 4g) users can use the chunking dropdown to select whether to chunk by question or answer (please see Sec. 5.2 for more details about these methods). Using the ordering dropdown, they can also select the ordering within the chunks, either by relevance or their original ordering within the interview recording (Fig. 4h). Each of the chunks for the chosen parameters appears highlighted in a different color (Fig. 4i). Users can click the 'play chunk' button to play a particular chunk in the main viewer in the Edit Panel. They can remove a particular answer from the chunk by clicking the 'remove from chunk' button or delete the entire chunk by clicking the trash can icon. They can also delete entire chunks (Fig. 4j) if the editor does not think they would be useful to the final edit. While deleting a chunk removes it from the review panel, the chunk remains available should the user want to revisit it later.

4.3 Placeholder B-roll

In order to support the ability of users to place temporary b-roll videos or images, ChunkyEdit provides a panel for uploading and labeling these assets (Fig. 4k). Users can add a placeholder image or upload their own media file. Users are asked to provide a brief description of what topic the video or image is related to, which the system uses to suggest placement within an existing chunk.

4.4 Export

After finishing the chunking process in the tool, users can export their chunks (Fig. 4l) in two different types of outputs that support different video editing workflows and potential collaboration methods. The paper edit produces a formatted PDF document with the chunks, their transcript segments, and the corresponding timecode (Fig. 1, middle). The EDL output generates an editing decision list file for the stringout that places the corresponding clips in each chunk on a timeline with a two-second gap between each chunk. Any chunks that the user deleted in the UI are placed at the end of the timeline so that they can be retrieved easily if the user decides to add them back. This EDL file can be opened in many common video editing programs, such as Adobe Premiere Pro, Avid Media Composer, and Final Cut Pro (Fig. 1, right). Users can then rearrange the chunks on the timeline, extend or trim clips, and make any additional edits as normal in their video editing program.



Question: How did Smith live up to your expectations, hopes and or fears when you arrived?

Answer: I wasn't sure what to expect, so I think it went beyond my expectations, really. I mean, I just. I felt like there. I kind of expected the academic side, but you really don't know what kind of friends you will make and what other activities you will do. And I actually ended up being really involved in extracurricular activities in a way that I wasn't involved when I was in high school. So it's been really amazing, I think exceeded my expectations.

Question: What clubs and organizations and teams were you involved in?

Answer: My first year, I um worked out a couple of campaigns with Mass Perk. I was a campaign organizer for a campaign to oppose drilling in the Arctic National Wildlife Refuge, um and we made a big polar bear out of plywood and had people sign names on it and took a picture and sent it to John Kerry. And we had a lot of petitions. So I think that Polar bear is still a bass hall. So I was responsible for that. Oh, my second year uh I was

chunk 3: Expectations and experience at Smith College

I wasn't sure what to expect, so I think it went beyond my expectations, really. I mean, I just. I felt like there. I kind of expected the academic side, but you really don't know what kind of friends you will make and what other activities you will do. And I actually ended up being really involved in extracurricular activities in a way that I wasn't involved when I was in high school. So it's been really amazing, I think exceeded my expectations.

Yeah, it was definitely a lovely home base. Um, I had a lot of friends there. It was really fun. For my first and second year, the dining hall was in park, so that was a great place to see your friends every day. And it's just really natural and you know, you can just catch up with them any time. So I felt really close to my friends.

chunk 4: Study abroad experience

I did a year abroad at Oxford University.

It was so great. I'm really glad I did it. It's completely different from Smith and really, you

Figure 4: The ChunkyEdit UI has two main panels: the Editing Panel (a) and the Review Panel (g). The Editing Panel displays the video (b) and the full transcript (c), which comprises question and answer pairs (d). It also shows the chunk labels (e) and allows for adding new chunks, adjusting chunk membership, and searching the transcript (f). Chunks are displayed, each in a different color. Users can view different chunking and ordering methods (h), and play, edit, or delete chunks (i). Deleted chunks (j) remain available for re-use. Users can try out temporary b-roll (k) and export a paper edit or stringout (l).

5 IMPLEMENTATION

ChunkyEdit creates and presents video chunks through a web app to support the design goals discussed in Sec. 2.

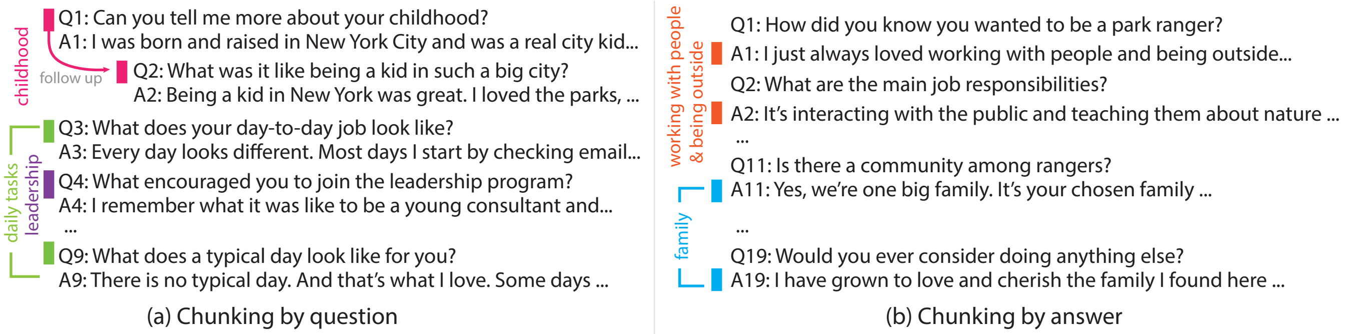
5.1 Pre-processing & Segmentation

We first obtain a verbatim transcript of the text spoken by the interviewer(s) and interviewee(s) using the Speechmatics API [2]. We obtain word-level timings and speaker diaritization results, which assign speaker labels (e.g., 'speaker 1,' 'speaker 2,' etc.) to each word. This speaker assignment is based entirely on the audio difference between the speakers, as often in interviews the interviewer does not appear on camera and only their voice is captured.

Early in the editing process, editors typically work at a relatively coarse level of granularity (Design goal 7)(for more information, see Sec. 2.3). For our initial chunk generation, we split clips according to each speaker's turn either asking or answering a question or making a statement (Design goals 1, 6, and 7). This typically corresponds to a question from the interviewer and an answer from the interviewee. While there can be additional people in these two roles, in this section we discuss our methods using the common two-person case for a single video. Questions and answers can range in length from a couple of words to several sentences. We automatically assign the speaker with the most words to be the interviewee, as typically the interviewee speaks longer than the interviewer. We call what the interviewer says a "question" regardless of the grammatical structure of the spoken line. Each of our chunking approaches described below automatically produces groups of **question-answer pairs**.

5.2 Chunking

We focus our efforts on two common strategies for organizing interview video footage: 1) grouping according to the interviewer's questions and 2) grouping by topics in the interviewee's answers. Question-based chunks are ordered based on their order within the input recording (Design goals 1, 2, and 3), and topic-based chunks are ordered according to their prominence within the entire interview transcript (Design goal 6). We provide an overview of these methods in Table 2, and in Sec. 6 we evaluate the trade-offs of these different chunking methods.


childhood
Q1: Can you tell me more about your childhood?
A1: I was born and raised in New York City and was a real city kid...
follow up
Q2: What was it like being a kid in such a big city?
A2: Being a kid in New York was great. I loved the parks, ...
daily tasks
leadership
Q3: What does your day-to-day job look like?
A3: Every day looks different. Most days I start by checking email...
Q4: What encouraged you to join the leadership program?
A4: I remember what it was like to be a young consultant and...
...
Q9: What does a typical day look like for you?
A9: There is no typical day. And that's what I love. Some days ...
working with people & being outside
Q1: How did you know you wanted to be a park ranger?
A1: I just always loved working with people and being outside...
Q2: What are the main job responsibilities?
A2: It's interacting with the public and teaching them about nature ...
...
Q11: Is there a community among rangers?
family
A11: Yes, we're one big family. It's your chosen family ...
...
Q19: Would you ever consider doing anything else?
A19: I have grown to love and cherish the family I found here ...

(a) Chunking by question

(b) Chunking by answer

Figure 5: ChunkyEdit supports grouping interview question-answer pairs by either similar questions (a) or answer themes (b). For question-based chunking (left), Question 2 (Q2) is a follow-up question to Q1 so it is grouped in the same chunk. The interviewer re-asks Q3 in Q9, which comes quite a bit later in the conversation. Despite this separation, ChunkyEdit groups these related questions. For answer-based chunking (right), A1 and A2 discuss similar themes of working with the public and being outside in nature so they are grouped in the same chunk. A11 and A19 both discuss finding family among the interviewee's coworkers and are also grouped, although these answers do not appear sequentially in the original recording.

5.2.1 *Chunking by question.*

An interviewer typically asks questions based upon a list of pre-written notes or questions. However, in practice interviewers often deviate from the written wording of these questions, change the order of the questions based on the natural flow of the conversation, and add, remove, or reword questions to get interviewees to elaborate on what they are saying (please see Sec. 2.1 for more detail about conducing an interview). Two common scenarios are follow-up questions (Design goal 3) and reiterated questions (Design goal 2).

Based on the questions voiced in the interview, we identify **follow-up questions**, which are questions that ask for additional detail about a previously discussed topic but may themselves lack enough detail or context to stand alone (e.g., a question may be "What was it like growing up in Montana?" and the follow-up may be "Was it cold?" Identifying follow-up questions is a relatively under-explored area of NLP research [35], but empirically we find that GPT-4 [49] does a good job at this task. Therefore, we use GPT-4 to identify follow-up questions using the following prompt:

> *You are a chatbot. You will answer whether the current question is a follow up to the previous question with a YES, NO, or UNCERTAIN response. The previous question is: <QUESTION>. The current question is: <QUESTION>.*

To capture reiterated questions, we compare the similarity of each question in the transcript using Sentence-BERT [56] embeddings and the built-in semantic similarity utility function. This returns a similarity score scaled 0 to 1. We group all question-answer pairs with questions with a similarity score above 0.5 into the same chunk and add all previously found follow-up questions regardless of similarity. We then prompt GPT-4 [49] to provide a summary of the questions in the same chunk, using the following prompt:

> *Paraphrase the following questions into a single question: [<QUESTION0>, <QUESTION1>. . .].*

This paraphrased question becomes the label for each chunk, which comprises the question-answer pairs associated with this label. Every question-answer pair is assigned to a single chunk. Because not every question is a follow-up question or similar to other questions, in some cases a chunk will only have one member whose label is the original question.

5.2.2 *Chunking by topic.*

We identify themes that emerge in the interviewee's answers by grouping text by topic in the transcript (Design goal 5). Grouping text by topic through topic modeling and text clustering is an active area of research within the NLP community [18]. Grouping or clustering parts of text documents by similarity is one task, and another, sometimes separate task, is providing labels for these groups of text. While there are many different approaches to topic modeling, the domain of editing video interviews poses a few particular challenges, including having relatively short texts (i.e., the number of words in a given interview is small relative to the large text corpora common in other NLP tasks). Given the relatively small size of each "document," i.e., interview transcript, we explored the following three methods of grouping interviewee answers by theme and providing semantically meaningful labels for each cluster:

GPT-4 clustering + labels (Answer 1): Our first approach to chunking by topic involves using GPT-4 [49] for both clustering and topic labeling. To obtain the topic labels and groups, we used the following prompt:

> *You are presented with an interview, which includes pairs, each consisting of a question and an answer. The collection is presented in JSON format with the name of the pair as $pair_x$ with x being the id of the pair. Your task is to find the main topics across the interview and the most similar pairs to each topic. Provide your response as a JSON object with the following schema: $topic :< topic >, pair_{ids} :< [idx] >$. The interview is: <INTERVIEW>. Please find at most 10 topics in the interview above. Pairs can be assigned to multiple topics. The topics should be in the form of a JSON as shown above.*

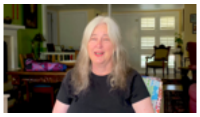
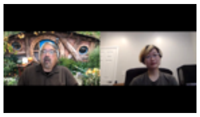
Video	Question	Answer 1: GPT-4	Answer 2: Hybrid	Answer 3: Keyword
CF	1) What clubs, organizations, and teams were you involved in, including any political or energy efficiency activism?	1) Choosing Smith College 2) Clubs, organizations, and activism	1) College experience and personal growth 2) Environmental activism and campus issues	1) Smith College 2) Campus
 LR	1) What is the balance between gathering and hoarding materials, and how many quilts have you made and given away?	1) Fabric hoarding 2) Quilting as an art	1) Quilting and sharing on Instagram 2) Quilting and fabric hoarding	1) Quilting skills 2) Sewing fabric
 RN	1) Can you discuss your experience at Stanford, including your background, the climate for kids of color, involvement in Asian-American organizations, and how your relationship with the community evolved over time?	1) Asian-American community and activities 2) Student organizations and unity	1) Asian-American identity and activism 2) Building Asian-American community	1) Silicon Valley 2) Asian-American community

Table 1: For three videos in our dataset (CF, LR, and RN), we show one example of the Question label and two label examples for the other three methods. While the topics of the labels are often similar across different methods, the labels vary in length and specificity. Question labels tend to be very long, while keyword labels are very short.

In this approach a single question-answer pair can be assigned to zero, one, or multiple chunks.

Sentence embedding clustering + GPT-4 labels (Answer 2): Our second approach uses a combination of text embedding, clustering, and GPT-4. First, we use Sentence-BERT [56] to embed the interview answers. Then we use UMAP [42] pre-processing to reduce the dimensionality of the embedded text data and HDBSCAN [41], a density-based clustering method, to perform clustering. For both algorithms we use the default parameters, except reducing *n_neighbors* from 15 to 2, to emphasize more local structure within the text embeddings by considering smaller neighborhoods when assigning clusters. Each resulting cluster has a numerical label, but no semantic label is provided. After filtering out the outlier cluster, which is indicated by a label of −1, we then use the following GPT-4 [49] prompt to label each of the resulting clusters:

> *You will be given a list of text strings clustered by topic.*
> *Your task is to provide 1-5 words to describe the topic.*
> *Here is the list: [<ANSWER0>,<ANSWER1>,. . .]*

Using this approach, every question-answer pair can be assigned to only a single labeled chunk or the group of outliers, which is not labeled or included in the final set of chunks.

KeyBERT clustering (Answer 3): Our third approach to topic modeling uses KeyBERT [27] to extract the most central topics within all of the answers first and then groups the answers according to their similarity to these main topics. First, we use KeyBERT to extract the top 10 most central n-grams, each containing 1-5 words. We then use the Sentence-BERT [56] semantic search function to find the top n most similar interview answers to each keyword to produce chunks above a similarity threshold of 0.2. Using this approach a single question-answer pair can be assigned to zero, one, or multiple chunks, but the number of chunks is fixed at 10. As this method is the only non-GPT method ChunkyEdit supports, the system defaults to this keyword-based approach if GPT-4 returns results that do not match our schema.

Method	Text	Cluster	Label	Assignment
Question	Question	GPT-4, SBERT	GPT-4	1
Answer 1: GPT-4	Answer	GPT-4	GPT-4	0, 1, 2+
Answer 2: Hybrid	Answer	UMAP, HDB-SCAN	GPT-4	0,1
Answer 3: Keyword	Answer	SBERT	KeyBERT	0, 1, 2+

Table 2: ChunkyEdit provides four different chunking methods, which each use different strategies for creating and labeling the chunks. The first (Question) focuses on grouping similar and follow up questions, and the other three (Answer 1-3) focus on common themes in the interviewee's answers. The choice of chunking strategy affects whether a question-answer pair can be assigned to 0, 1 or multiple chunks.

5.3 Temporary B-roll placement

ChunkyEdit provides a panel that allows users to attach temporary b-roll images or videos to particular chunks (Design goal 9). Users can select a solid gray placeholder image or upload their own media. The user provides a short text label for each image or video, and then ChunkyEdit finds potential chunks associated with that label. We use Sentence-BERT’s semantic search utility function to compare the user-provided text label and all of the question and answer pairs in the transcript. The temporary b-roll is assigned to the top-3 most similar chunks with at least a similarity score of 0.2, but the user can adjust this assignment in the UI.

6 RESULTS

We tested ChunkyEdit with 12 different video interviews (4-81 mins. in length, mean=43min.) from a range of publicly available video repositories from universities and public broadcasting stations (Table 3). These videos included oral histories and documentary interviews in a variety of formats, including live recordings and recorded Zoom calls.

Each of the different chunking methods can result in chunks of different sizes. The average number of chunks across the 12 videos varied from 2.92 for the Answer 2 method to 10 for the Answer 3 method (Table 3). For example, for video RN, the Answer 2 method produced 3 larger chunks, but the other three methods produced 9 or 10 smaller chunks. Qualitatively we observe that these labels for larger chunks are more vague (e.g., “building Asian-American community” for Answer 2 vs. “What are your most memorable experiences with ASA?” for the Question method). We see the greatest variance in the number of chunks for the Question method, where the total number of chunks for a particular video varied from 1 large chunk to 12 smaller chunks. Videos with fewer question chunks indicate that the interviewer was asking many similar questions or follow-up questions, while a larger number of chunks indicates that the interviewer asked about a broader range of topics. For example, in the LW example, the interviewer’s questions focused very narrowly on an instructor’s experience teaching a class and included many related and follow up questions (e.g., “What do you think are the most difficult aspects of teaching or discussing Aboriginal issues?” and “So when you were in one of those situations, what was your response at the time?”). Using the Question method, ChunkyEdit grouped all of these similar questions for LW into a single chunk (Table 3).

In addition to differences in the number of chunks and the average chunk size, there are also differences in the length and specificity of the chunk labels. While Answer 3 chunk labels tend to be short 1-5 word phrases with relatively little context (e.g., “campus” for CF), the Question labels are longer (e.g., “What clubs, organizations, and teams were you involved in, including political and energy-efficient activism?” for CF) (Table 1). Both the Answer 1 and Answer 2 methods typically provide label phrases that are a few words each (e.g., “Clubs, organizations, and activism” and “Environmental activism and campus issues”). Despite these differences in length, each of the labels provides a summary of the content within the chunk, and each may be appropriate for different projects. In our user evaluation (Sec. 7.1.4), we further discuss the number of chunks and the types of labels editors find useful.

6.1 Chunking evaluation

To evaluate the performance of our four interview chunking strategies, we focus on two questions: 1) how coherent is the dialogue within a chunk? and 2) how well does the assigned label describe the chunk? Evaluating the quality of clustering outputs computationally is known to be a complicated task with many trade-offs between different evaluation strategies [3]. These evaluation strategies typically focus on examining the intra-cluster and inter-cluster distances to assess the coherence of each cluster and the separation between different clusters. We evaluate our chunks using both inter- and intra- cluster distance metrics and human input.

6.1.1 *Distance-based evaluation.*

To evaluate chunk coherence using distance-based metrics, we embed all of the text using Sentence-BERT [56] and compute all intra-cluster and inter-cluster distances using Euclidean distance.

Cluster coherence: We calculate the Calinski-Harabasz (C-H) index, which compares the ratio of the inter-cluster distances to the intra-cluster distances. For this criteria, higher C-H scores indicate better clustering, as this corresponds to chunks that are more tightly defined. We find no significant differences across the four chunking methods due to high variance in the CH-indices across projects (Table 3) ($F = 1.04$, $p > 0.05$).

Cluster label similarity: To evaluate the quality of the chunk labels, we calculate the average similarity between the assigned label for the chunk and each of answers in the chunk using the SentenceBERT [56] semantic similarity utility function. Similarity scores range from 0 to 1, with scores closer to 1 reflecting greater similarity. Although the Question labels tend to be far longer than the other methods, particularly Answer 3 (Table 1), we find no significant difference in the average label similarity for the different methods (Table 3) ($F = 0.41$, $p > 0.05$).

6.1.2 *Human annotator evaluation.*

In addition to our distance-based evaluation, we also evaluated chunk coherence and label quality using human raters. We recruited 12 participants (6 female, 6 male, mean age = 40) on Prolific whose first language is English. We asked them each to rate on a scale of 1-7 the coherence of the text chunks and how well the chunk labels describe the chunks. For each of the two tasks, we presented each annotator with 12 examples, generated from three different randomly selected videos (CF, RN, LR) using our four different chunking methods (Table 1).

Overall the annotators rated the chunk coherence and label quality highly, particularly for the three non-keyword based methods. They indicated that Answer 1 and 2 produced the most coherent chunks (median=6), followed by Question (median=5), and Answer 3 (median=4). A Kruskal-Wallis test indicated significant differences in chunk coherence among these different methods ($H = 18.9$, $p < 0.01$), and a post-hoc Dunn’s test indicated there were significant differences between the Answer 1 and Answer 3 methods and the Answer 2 and Answer 3 methods and no significant differences between the other pairs of methods. Annotators also indicated that the Question and Answer 1 and 2 methods produced very relevant labels (median=6), while the Answer 3 method provided lower quality labels (median=3). A Kruskal-Wallis test found significant differences among these different topic modeling approaches

Video	# QA	Dur.	Question			Answer 1: GPT-4			Answer 2: Hybrid			Answer 3: Keyword		
	Pairs	(min)	*# Chunks*	*CH*	*Label*	*# Chunks*	*CH*	*Label*	*# Chunks*	*CH*	*Label*	*# Chunks*	*CH*	*Label*
CA	27	55	12	4.16	0.25	9	11.41	0.36	4	16.72	0.39	10	23.18	0.31
CB	23	58	5	4.37	0.31	10	368.77	0.37	2	14.71	0.35	10	18.76	0.46
CF	29	19	6	2.50	0.43	10	21.97	0.37	4	26.45	0.40	10	32.37	0.42
CH	9	25	1	SC	0.46	8	0.02	0.33	2	28.35	0.42	10	MC	0.36
ER	33	81	8	3.12	0.39	10	24.50	0.34	7	21.55	0.45	10	34.66	0.33
KH	21	67	10	8.18	0.39	10	23.67	0.39	3	9.37	0.52	10	9.44	0.43
KS	8	35	7	7.15	0.36	9	MC	0.28	2	7.16	0.46	10	MC	0.43
LR	11	10	7	4.82	0.45	10	4.57	0.57	2	39.26	0.52	10	0.41	0.55
LW	14	56	1	SC	0.60	10	0.90	0.47	2	11.88	0.55	10	2.07	0.58
MS	6	4	4	3.94	0.42	8	MC	0.50	1	MC	0.34	10	MC	0.36
RN	24	67	9	3.28	0.35	10	1.80	0.40	3	34.05	0.39	10	17.30	0.41
SF	18	38	1	SC	0.42	9	6.01	0.47	3	15.13	0.42	10	14.00	0.31
Avg.	19	43	5.92	4.61	0.40	9.42	46.36	0.40	2.92	20.42	0.44	10	16.91	0.41

Table 3: We used ChunkyEdit for 12 different projects with a range of topics and numbers of question-answer (QA) pairs. The number of chunks varied across the different methods for each project and resulted in varying levels of coherence, as indicated by the C-H scores. However, the average similarity between the labels and the clusters (*Label*) remained relatively consistent across the different methods. Any chunkings that resulted in a single chunk are denoted with *SC*, and any chunking methods that produced multi-chunk assignments are marked with *MC*.

($H = 20.6$, $p < 0.01$). A post-hoc Dunn's test indicated there were significant differences between the Answer 3 method and each of the other three methods and no significant differences between the pairs of other methods.

6.1.3 Evaluation conclusions. Our distance-based methods indicated no significant differences among the four methods in terms of chunk coherence and label relevance. However, human annotators found that while three of the methods performed similarly, the Answer 3 (keyword) method produced significantly worse results. The three higher performing methods all use GPT-4 to label–and in some cases–group the chunks, but they each result in different numbers of chunks and different labels. ChunkyEdit exposes all of these different chunking methods in the UI to allow users to determine which chunking works best for a particular video. In the next section, we discuss additional feedback from users about these chunking mechanisms in the context of the ChunkyEdit tool and video editing more broadly.

7 USER EVALUATION

In our user evaluation, we focused on learning more about how professional video editors organize and assemble their videos in the early stages of the editing process, collected feedback on the ChunkyEdit prototype, and learned about how chunk-based video editing tools could be used by editors.

Participants: We recruited 8 professional video editors on *UserInterviews.com* (Table 4). Participants (5 men, 3 women, aged 27-55) had 7-20 years of editing experience. Three are freelancers who run their own companies (E3, E4, E7), three work for a small company (E2, E6, E8), and two work for large media organizations (E1, E5). Participants E1 and E5-8 regularly use text-based video editing features (e.g., the text panel with automatic transcription in Premiere Pro). Our study protocol underwent an organizational approval process, and our participants were compensated $1 per minute.

Study format: All participants completed a one-on-one 60-90 minute virtual study session. We asked participants to discuss their current video production workflow for editing interviews. We then showed participants the ChunkyEdit interface and demonstrated the main features. Participants E1-E5 chose one or more videos to work with from our 12 sample videos (Table 3), and participants E6-E8 were each asked to provide their own input interview video (Table 5). E6 provided a 37-minute video about an adoption agency, E7 provided a 44-minute interview with a plant shop owner, and E8 provided a 9-minute interview with a music store employee. All participants then used ChunkyEdit to explore and organize the footage to produce a finished 2-5 minute interview profile. When they were done creating chunks, participants exported their chunks to an EDL file, opened them within an existing video editing program, and discussed how they would then proceed with their video editing process. E6-E8 were given an extra 30-minutes to further refine their ChunkyEdit stringout into an early rough cut in Premiere Pro.

7.1 User evaluation results

Participants noted that ChunkyEdit could be useful for a wide range of projects. E3 said, *"There was a documentary I was working on, and there was a lot of interviews. I think this type of interface would definitely have been helpful to me…It gives you a bird's eye view of what to cut out and what to leave in and how to make it flow better."* 7 out of 8 participants said that they would be likely to use a tool

ID	Experience				ChunkyEdit ratings					
	Years	Software	Company	Editing experience	Organized	Focus	Speed	Control	Share	Use
E1	15	Avid, FCP, PrPro	Large	Nonfiction & fiction films for non-profits & companies	3	5	4	2	3	5
E2	20	FCP, PrPro	Small	Senior editor; documentaries & behind-the-scenes videos	4	3	5	1	3	5
E3	7	Avid, PrPro	Free.	MFA in post-production; YouTube & Instagram videos	4	5	5	1	5	5
E4	12	Avid, FCP, PrPro	Free.	Edits short & long form documentaries	5	3	4	1	4	5
E5	20	Avid, FCP, PrPro	Large	Post-production supervisor & editor; celebrity interviews	4	4	2	1	5	3
E6*	15	Avid, FCP, PrPro	Small	3-7min profiles for non-profits	5	5	5	1	5	4
E7*	10	PrPro	Free.	Short fiction & non-fiction social interest films	3	4	5	1	2	4
E8*	20	FCP, PrPro	Small	Corporate profiles	5	4	5	2	5	5

Table 4: Participants were professional video editors (7-20 years of experience) working at various sized companies. They have a range of experiences editing different types of projects. Participants were asked to rate (on a scale of 1=not at all to 5=very much) how much they felt ChunkyEdit would help them feel more organized (Organized), focus on the relevant parts of an interview (Focus), speed up the editing process (Speed), reduce creative control (Control), and help share intermediate edits (Share). They were also asked to indicate how likely they would be to use ChunkyEdit in their editing workflow in the future (Use). *Note that E1-E5 edited with the 12 sample videos (Table 3), and E6-E8 provided their own videos (Table 5).

like ChunkyEdit if it were incorporated into an editing tool, such as Premiere Pro, Avid, or Final Cut Pro in the future (median = 5/5).

7.1.1 Organizing footage. Participants said that ChunkyEdit would be useful for helping to organize their videos (median = 4/5). E5 said that ChunkyEdit could be useful as both an organizational and an assembly tool: *"it's more useful than the transcript alone, because you get more use out of it with the tags and everything."* Several editors (E1-2, E4, E6-8) saw potential in using the tool directly in the ordering process to produce a stringout. E2 said that producing stringouts is helpful for communicating with clients: *"The client will ask for select stringouts… they're not gonna look at a transcript and they're not gonna watch the interview. So sometimes you're just playing selects for the clients to watch."*

7.1.2 Speeding up the editing process. 7 out of 8 editors indicated that ChunkyEdit could help greatly speed their rough cut production process (median = 5/5). E1 said that reducing the time to get to the main editing process is crucial because *"a lot of clients are very conscious of the amount of editing hours that go into a project… if we can already generate that raw stringout, then I'm already saving myself a few hours potentially at that point. That would definitely be really helpful at that point that I would be able to focus on the most important parts, whether it's something that I was selecting myself, or if somebody else was reviewing and selecting it."* Although E2 thought that ChunkyEdit could help speed the editing process, they said that they would likely still watch all of the input footage once to get a sense of the person's voice and tone of the overall piece. E2 said, *"This [ChunkyEdit] is definitely a good starting place. Otherwise I would be grouping it manually in the timeline, which seems more time consuming than just exporting this and bringing it into Premiere. Because even then, it's going to have to be edited, right? But at least it's kind of done the work of an AE [assistant editor] basically. I would probably trust this just as much as I would an assistant editor."* E5, who rated the potential speed up the lowest (2/5) and was the least likely to use ChunkyEdit in the future (3/5), said that they have a workflow they already like using that necessitates watching the videos multiple times. While participants saw value in the time savings ChunkyEdit could offer, they also indicated that it would not take away creative agency (median = 1/5, 1='not at all,' 5='very much'). E4 said, *"It's gonna really help familiarize yourself with the footage too. So I think the longer you kind of are working on that, the more it'll speed up the process. It wouldn't reduce the creative control."* E7 said, *"It's like getting to the same type of process that I already have, but just like in a quicker different way."*

7.1.3 Communicating editing decisions. Many participants saw value in the two currently supported export formats and also shared suggestions for additional export functionality that would be useful. E4 said that they thought that the paper edit produced by ChunkyEdit could be useful for communicating editing decisions with clients: *"This [The paper edit] seems like actually it would streamline the process for me a little bit, or at least put it in a way that I could send clients."* E1 said that this process of producing a stringout using an EDL could help producers working closely with editors: *"If I was providing all this footage to a producer and they don't really *have familiarity with an editing program, at least they would be able to play things out themselves."* While several participants said that they appreciated the universality of supporting the EDL format, E2 suggested also supporting XML formats, which work across editing programs and provide access to more internal structure within the editing program, such as video bins and markers.

7.1.4 Number of chunks. Several participants commented specifically on the appropriate number of chunks being selected. E4 found that the chunking and color coding functionality of ChunkyEdit closely approximated what they were already doing and indicated the tool was providing a reasonable number of topics for the examples shown: *"I think 2 to 10 is a pretty good ballpark. If I break it down too much then that doesn't seem to help at all. . . If I was creating a 5-minute corporate ad for some business, I would get it broken down probably into at least 4 to 5 different topics or even maybe I'd say 6. So I couldn't see myself going over 10."* E8 had experience editing a series of profile videos, taking 10-30 minute input video recordings and producing 2-minute outputs. They specifically noted that each of the four chunking methods produced a similar number of chunks to the number of topics they would consider for one of these profiles: *"I think there were always six or seven general topics that we covered and so that's what I was always looking for. . . [in ChunkyEdit] I think the number of clips is not overwhelming; it's not too short to where you feel like you don't have enough; I like the overall number per chunk and chunks."* The four different methods suggested 3-10 chunks for E8's provided video, and E8 chose to use the method that started with 7 chunks.

7.1.5 Chunk label quality. For different types of videos, editors chose different chunking methods. For the celebrity interviews E2 edits, each interviewee is asked similar questions, and they said that question-based chunking would be particularly useful. Both E7 and E8 were also particularly fond of the chunking by questions, as it closely resembled their current workflows. E7, who largely edits content from other sources and is only present for a small fraction of the interview recordings, said, *"It does in some ways mirror how I work with transcripts as is and in my paper edits."* Likewise, E8, who also mostly edits with unfamiliar content, said, *"The chunk by question – typically when you when you spit out your first transcript; I mean that's kind of what it looks like. So yeah, definitely that feels natural to me.* E6, however, preferred two of the chunking by answer methods (Answer 1 and Answer 2). They said, *"I feel like [methods] one and two gave me a higher level, a thematic view of the content, and that gives me a tool I didn't have when just looking at raw transcript."* E7, who also liked Answer 1 said, *"These categories are generally helpful and even just kind of in reminding myself what broad categories were in here."* E7 manually renamed three of the suggested chunks from Answer 1. However, they said that these manual labels were intended to provide notes to themselves about the placement of the clips within the sequence (e.g., "ending thoughts,") rather than the topic, *"I think the renaming was less about oh, this isn't accurate and more like I want to label this to put at the end."* This indicates that even at this chunking stage, the editor is considering future structuring and ordering decisions. While ChunkyEdit allows users to manually rename or add topics after generating chunks automatically, E2, E4, and E6 suggested the ability to upload a set of topics that the client requested at the start.

7.1.6 Comparison with standard text-based editing tools and transcript. In the study sessions E6-E8 labeled their own videos in ChunkyEdit and exported their stringouts to Premiere Pro for further editing to produce an early rough cut. Editors E6-E8 are all frequent users of the text panel in Premiere Pro, an automatic transcription tool that became commercially available in early 2023. All three editors gave feedback on the labeling methods and noted specific benefits of chunking beyond just the transcript (please see Appendix C for additional information). E6, who typically works with non-profits with small budgets, said that ChunkyEdit could help reduce costs by allowing these clients to identify relevant areas of the videos themselves, reducing the time the editor has to spend on the project: *"I think that's really cool that you can empower a non-editor to edit or at least to label and organize stuff and take all the things that they don't want and then let the editor edit.* E7, who views labeling the transcript as a central but tedious part of their process, said, *"I feel like I kind of chunk and categorize stuff just like this already. I'll make my own categories in a Google Doc and copy and paste quotes around. Having some suggested categories [in ChunkyEdit] could be helpful to review what's in an interview and give me a start because that feels daunting sometimes, particularly in an interview."* E6 and E7, who both generated long stringouts, mentioned liking having the same clip assigned to multiple chunks sometimes in order to see and hear how a particular clip would work in the context of different choices of surrounding clips. E8 had previously worked on a series of interviews with a larger team and said, *"[ChunkyEdit] would have saved an enormous amount of time at the company I worked for, had we been able to do this and make an EDL and just have the editor jump right on it. I did 60 of these videos for them, and this would have been so helpful."* E8 spent 15 minutes reviewing the chunks in the ChunkyEdit UI, opened ChunkyEdit output EDL in Premiere Pro, and then produced a 1.5-minute shareable rough cut in under 4 minutes of editing time (please see supplemental materials for the video).

8 DISCUSSION

Recent advances in speech-to-text and machine learning tools are providing many new opportunities to support text-based video editing and automated video editing tools across all levels of video editing, from novices to professionals. We made several decisions in the design of ChunkyEdit to support chunk-based editing by automating certain parts of the process while leaving other steps and decisions up to users. Many of the themes that emerged in our discussions with editors involved choosing what to automate and how to design future text-based video editing tools.

8.1 Balancing time and creative control

Editors frequently emphasize how much time editing takes and how much time pressure they are under from collaborators and clients. Editor Sheila Curran Bernard says that spending time is an important part of the editing process: *"Time is an increasingly rare commodity for filmmakers, especially during pre-production and editing. Yet time is what enables a film to have depth, in terms of research, themes, and layers of storytelling, it can enhance creativity"* [7]. Two of the editors in our user evaluation (E2, E5) mentioned that even with a tool like ChunkyEdit, they would still watch all of the videos at least once. Our participants repeated how important it was for them to find methods of working that make their process efficient but ultimately leave them with as much storytelling freedom as possible. As more automation becomes possible for certain aspects of video editing, an important question to consider is how to keep this freedom for creative professionals. ChunkyEdit provides one point on this spectrum from the manual, timeline-based editing that conventional video editing software supports to fully automated AI-based editing. Our evaluation indicates that professional editors appreciated this middle-ground between providing some automated organizational support, such as by grouping clips by related questions and identifying themes in the transcript, without producing full edits.

8.2 The future of text-based editing

While all of the editors in our study were used to working in some capacity with transcripts, they primarily use timeline-based editing interfaces. E1 and E6 suggested incorporating more awareness of the timing of the clips and the video timecode into the ChunkyEdit UI, while the other editors said that they appreciated working exclusively with the transcript and did not find the timecode relevant to editing in the more content-focused way that ChunkyEdit supports. Several participants, particularly those with experience with text-based editing, said that they saw the automatic clip labeling and grouping features in ChunkyEdit as being important next steps in text-based video editing tools. As editors tools continue to adopt more text-based features, new opportunities will emerge for designing video editing interactions that go beyond the timeline and put the emphasis on content and storytelling.

8.3 Supporting editors at all levels

We focused our evaluation on exploring how experienced editors could incorporate a chunk-based editing workflow into their process. While chunking in other domains has been widely explored as a mechanism experts use to remember information, teaching these strategies to novices has also been shown to be effective [8, 25]. Several editors mentioned the opportunity to use ChunkyEdit with clients or producers who may have less technical knowledge of editing. It is our hope that a chunk-based approach could potentially help less experienced editors organize video projects.

9 LIMITATIONS & FUTURE WORK

While ChunkyEdit facilitates early steps in editing, it has a few limitations and presents several opportunities for future work.

9.1 Editing granularity

ChunkyEdit treats question-answer pairs as the primary structure within an interview. Currently edits are only possible at the level of granularity of a single interviewee's turn (i.e., answer). There are trade-offs between working with a wider window of context around a particular line while helping editors get the video down to its desired length. While our evaluation indicates that the level of granularity of chunks is appropriate for the early stages of editing, for videos with exceptionally long and meandering interviewee responses, it may be helpful to work at a finer level of granularity, such as the sentence or phrase level. Two potential areas of future work that could address this include exploring multi-label topic modeling in the context of video editing and developing methods to help editors remove lines that are not central to the narrative.

9.2 Journalistic context & applications

Many of the applications we envision for ChunkyEdit involve editing informational videos that rely heavily on journalistic integrity. As with any video editing task that involves remixing content, there is inherent risk of losing appropriate context, intentionally or unintentionally. Bernard says, *"You must condense the interview material in a way that does not alter its initial meaning, no matter how 'close' it might be or how accurate it remains"* [7]. By supporting the ability for editors to focus on question and answer pairs in an interview, ChunkyEdit helps editors find common themes, while seeing them in the context of the longer conversation. While three of the four chunking strategies use GPT-4, if editors are concerned about sharing sensitive content with this tool, they can use the non-GPT (Answer 3: Keyword) method or manually select topic labels. Future work should further help editors make informed decisions about including relevant context and choosing a labeling method that is appropriate for the nature of the content.

9.3 Chunk quality

Topic modeling remains an imperfect approach that can lead to clusters that are difficult to label [12, 30, 44]. These models are highly driven by the right selection of parameters [61, 66], and our methods do not currently optimize any parameters. E4 and E8 specifically commented on the importance of finding the right number of chunks and indicated that ChunkyEdit did a good job of identifying the right number of groups to review. We provide simple ways for users to add or remove chunks manually, but future work could more carefully explore this trade-off between size and specificity of video chunks. Several participants mentioned that ChunkyEdit is most helpful for particularly long videos. E6 said, *"I think larger projects are gonna benefit a larger percentage, short projects less, but it would still benefit smaller projects because a smaller project probably has a shorter turn-around anyways, and this would shorten the turn-around even more."* Future work could further explore this relationship between the scale of the input content and the benefits of chunking to editors. In practice we found that editors explored the four available topic modeling approaches before choosing the one that best captured the themes they wanted to convey in the final edit. Future work could also consider how editors decide upon the best set of themes for a given project.

9.4 Supporting additional video genres

ChunkyEdit was designed around the turn-taking structure of interview-based videos and leverages this structure to identify thematic chunks within a single video. Considering how to support multiple videos and identifying themes across several interviews would provide opportunities to explore larger video collections, such as oral history archives. Additionally, E4 and E6 suggested providing multi-language support for allowing clustering of videos in different languages. Extending our approach to other types of videos could be possible for domains with regular structure, such as step-by-step instructional videos. An interesting challenge for future work is identifying and leveraging this structure, particularly in domains without dialogue, such as action-based or sports videos.

9.5 Chunking support for open-ended tasks

Chunking is a powerful cognitive process that people use to manage large amounts of data and make sense of procedural tasks [26]. In this work we discuss the potential use of this mechanism in tools to support video editing. We explored chunking by using text clustering and topic modeling to facilitate the task of making local video editing decisions. In future work we would like to study if and how this type of chunking can reduce the cognitive load of early editing decisions. In addition to video editing, there are likely many other creative workflows, such as animation, that involve sequential decision-making that could potentially benefit from a chunk-based approach that helps people break down a problem and organize their choices. There are many open challenges in identifying and supporting these complex, long-term decision-making processes.

10 CONCLUSION

As text-based video editing tools gain popularity, there are many opportunities to use the text to help editors make editing decisions more efficiently. In this paper we introduced a system for helping editors organize their footage by identifying the structure and topics that emerge within interview questions and answers. We found that taking this relatively modest approach to automation within a video editing process resonated with professional editors who are often under tight time pressure but value the craft of editing. It is our hope that tools like ChunkyEdit will help editors focus on these critical editing decisions that make editing an art form.

REFERENCES

[1] 2023. Frame.io. https://frame.io/

[2] 2023. Speechmatics. https://www.speechmatics.com/

[3] Ramiz M Aliguliyev. 2009. Performance evaluation of density-based clustering methods. *Information Sciences* 179, 20 (2009), 3583–3602.

[4] Ido Arev, Hyun Soo Park, Yaser Sheikh, Jessica Hodgins, and Ariel Shamir. 2014. Automatic editing of footage from multiple social cameras. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–11.

[5] Dawit Mureja Argaw, Fabian Caba Heilbron, Joon-Young Lee, Markus Woodson, and In So Kweon. 2022. The anatomy of video editing: a dataset and benchmark suite for AI-assisted video editing. In *European Conference on Computer Vision*. Springer, 201–218.

[6] Alan D Baddeley, Graham J Hitch, and Richard J Allen. 2009. Working memory and binding in sentence recall. *Journal of memory and Language* 61, 3 (2009), 438–456.

[7] Sheila Curran Bernard. 2004. Documentary storytelling for film and videomakers. (2004).

[8] Graham D Bodie, William G Powers, and Margaret Fitch-Hauser. 2006. Chunking, priming and active learning: Toward an innovative and blended approach to teaching communication-related skills. *Interactive learning environments* 14, 2 (2006), 119–135.

[9] William Buxton. 1995. Chunking and phrasing and the design of human-computer dialogues. In *Readings in Human–Computer Interaction*. Elsevier, 494–499.

[10] Juan Casares, A Chris Long, Brad A Myers, Rishi Bhatnagar, Scott M Stevens, Laura Dabbish, Dan Yocum, and Albert Corbett. 2002. Simplifying video editing using metadata. In *Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques*. 157–166.

[11] Senthil Chandrasegaran, Chris Bryan, Hidekazu Shidara, Tung-Yen Chuang, and Kwan-Liu Ma. 2019. TalkTraces: Real-time capture and visualization of verbal content in meetings. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[12] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-Graber, and David Blei. 2009. Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems* 22 (2009).

[13] William G Chase and Herbert A Simon. 1973. Perception in chess. *Cognitive psychology* 4, 1 (1973), 55–81.

[14] Peggy Chi, Nathan Frey, Katrina Panovich, and Irfan Essa. 2021. Automatic Instructional Video Creation from a Markdown-Formatted Tutorial. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 677–690.

[15] Peggy Chi, Joyce Liu, Jason Linder, Mira Dontcheva, Wilmot Li, and Bjoern Hartmann. 2013. Democut: generating concise instructional videos for physical demonstrations. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 141–150.

[16] Peggy Chi, Zheng Sun, Katrina Panovich, and Irfan Essa. 2020. Automatic video creation from a web page. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 279–292.

[17] Minsuk Choi, Sungbok Shin, Jinho Choi, Scott Langevin, Christopher Bethune, Philippe Horne, Nathan Kronenfeld, Ramakrishnan Kannan, Barry Drake, Haesun Park, et al. 2018. Topicontiles: Tile-based spatio-temporal event analytics via exclusive topic modeling on social media. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–11.

[18] Rob Churchill and Lisa Singh. 2022. The evolution of topic modeling. *Comput. Surveys* 54, 10s (2022), 1–35.

[19] Marc Davis. 2003. Editing out video editing. *IEEE multimedia* 10, 2 (2003), 54–64.

[20] Adriaan D De Groot. 2014. *Thought and choice in chess*. Vol. 4. Walter de Gruyter GmbH & Co KG.

[21] Donald L Diefenbach. 2009. *Video production techniques: Theory and practice from concept to screen*. Routledge.

[22] Dennis E Egan and Barry J Schwartz. 1979. Chunking in recall of symbolic drawings. *Memory & cognition* 7 (1979), 149–158.

[23] Karen Everett. 2021. Documentary Editing. (2021).

[24] Andreas Girgensohn, John Boreczky, Patrick Chiu, John Doherty, Jonathan Foote, Gene Golovchinsky, Shingo Uchihashi, and Lynn Wilcox. 2000. A semi-automatic approach to home video editing. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*. 81–89.

[25] Fernand Gobet. 2005. Chunking models of expertise: Implications for education. *Applied Cognitive Psychology* 19, 2 (2005), 183–204.

[26] Fernand Gobet, Peter CR Lane, Steve Croker, Peter CH Cheng, Gary Jones, Iain Oliver, and Julian M Pine. 2001. Chunking mechanisms in human learning. *Trends in cognitive sciences* 5, 6 (2001), 236–243.

[27] Maarten Grootendorst. 2020. KeyBERT: Minimal keyword extraction with BERT. https://doi.org/10.5281/zenodo.4461265

[28] Wilko Guilluy, Laurent Oudre, and Azeddine Beghdadi. 2021. Video stabilization: Overview, challenges and perspectives. *Signal Processing: Image Communication* 90 (2021), 116015.

[29] Enamul Hoque and Giuseppe Carenini. 2015. Convisit: Interactive topic modeling for exploring asynchronous online conversations. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*. 169–180.

[30] Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. 2014. Interactive topic modeling. *Machine learning* 95 (2014), 423–469.

[31] Bernd Huber, Hijung Valentina Shin, Bryan Russell, Oliver Wang, and Gautham J Mysore. 2019. B-script: Transcript-based b-roll video editing with recommendations. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.

[32] Hannah Kim, Barry Drake, Alex Endert, and Haesun Park. 2020. Architext: Interactive hierarchical topic modeling. *IEEE transactions on visualization and computer graphics* 27, 9 (2020), 3644–3655.

[33] Joy Kim, Mira Dontcheva, Wilmot Li, Michael S Bernstein, and Daniela Steinsapir. 2015. Motif: Supporting novice creativity through expert patterns. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 1211–1220.

[34] Iring Koch and Joachim Hoffmann. 2000. Patterns, chunks, and hierarchies in serial reaction-time tasks. *Psychological research* 63 (2000), 22–35.

[35] Souvik Kundu, Qian Lin, and Hwee Tou Ng. 2020. Learning to identify follow-up questions in conversational question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 959–968.

[36] John E Laird, Paul S Rosenbloom, and Allen Newell. 1984. Towards Chunking as a General Learning Mechanism.. In *AAAI*. Citeseer, 188–192.

[37] Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational video editing for dialogue-driven scenes. *ACM Trans. Graph.* 36, 4 (2017), 130–1.

[38] Tak Yeon Lee, Alison Smith, Kevin Seppi, Niklas Elmqvist, Jordan Boyd-Graber, and Leah Findlater. 2017. The human touch: How non-expert users perceive, interpret, and fix topic models. *International Journal of Human-Computer Studies* 105 (2017), 28–42.

[39] Xiaoyang Mao, Omar Galil, Quintcey Parrish, and Chiradeep Sen. 2020. Evidence of cognitive chunking in freehand sketching during design ideation. *Design Studies* 67 (2020), 1–26.

[40] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2014. Video lens: rapid playback and exploration of large video collections and associated metadata. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 541–550.

[41] Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software* 2, 11 (2017), 205.

[42] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *The Journal of Open Source Software* 3, 29 (2018), 861.

[43] George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review* 63, 2 (1956), 81.

[44] David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 conference on empirical methods in natural language processing*. 262–272.

[45] Matthew R Nassar, Julie C Helmers, and Michael J Frank. 2018. Chunking as a rational strategy for lossy data compression in visual working memory. *Psychological review* 125, 4 (2018), 486.

[46] Cuong Nguyen, Stephen DiVerdi, Aaron Hertzmann, and Feng Liu. 2017. CollaVR: collaborative in-headset review for VR video. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*. 267–277.

[47] Klaus Oberauer, Simon Farrell, Christopher Jarrold, and Stephan Lewandowsky. 2016. What limits working memory capacity? *Psychological bulletin* 142, 7 (2016), 758.

[48] Klaus Oberauer and Reinhold Kliegl. 2006. A formal model of capacity limits in working memory. *Journal of memory and language* 55, 4 (2006), 601–626.

[49] OpenAI. 2023. GPT-4. https://openai.com/gpt-4

[50] Alejandro Pardo, Fabian Caba, Juan León Alcázar, Ali K Thabet, and Bernard Ghanem. 2021. Learning to cut by watching movies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6858–6868.

[51] 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.

[52] Amy Pavel, Dan B Goldman, Björn Hartmann, and Maneesh Agrawala. 2016. VidCrit: video-based asynchronous video review. In *Proceedings of the 29th annual symposium on user interface software and technology*. 517–528.

[53] Karen Pearlman. 2017. Editing and cognition beyond continuity. *Projections* 11, 2 (2017), 67–86.

[54] Lyndsey Pickup and Andrew Zisserman. 2009. Automatic retrieval of visual continuity errors in movies. In *Proceedings of the ACM International Conference on Image and Video Retrieval*. 1–8.

[55] Sergey Podlesnyy. 2020. Towards data-driven automatic video editing. In *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery: Volume 1*. Springer, 361–368.

[56] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[57] Emil Rijcken, Floortje Scheepers, Kalliopi Zervanou, Marco Spruit, Pablo Mosteiro, and Uzay Kaymak. 2023. Towards Interpreting Topic Models with ChatGPT. In *The 20th World Congress of the International Fuzzy Systems Association*.

[58] Paulo J Santos and Albert N Badre. 1994. Automatic chunk detection in human-computer interaction. In *Proceedings of the workshop on Advanced visual interfaces*. 69–77.

[59] Guy Schofield, Tom Bartindale, and Peter Wright. 2015. Bootlegger: turning fans into film crew. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 767–776.

[60] Herbert A Simon. 1974. How Big Is a Chunk? By combining data from several experiments, a basic human memory unit can be identified and measured. *Science* 183, 4124 (1974), 482–488.

[61] Alison Smith, Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater. 2018. Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system. In *23rd International Conference on Intelligent User Interfaces*. 293–304.

[62] John R Smith, Dhiraj Joshi, Benoit Huet, Winston Hsu, and Jozef Cota. 2017. Harnessing ai for augmenting creativity: Application to movie trailer creation. In *Proceedings of the 25th ACM international conference on Multimedia*. 1799–1808.

[63] Mirko Thalmann, Alessandra S Souza, and Klaus Oberauer. 2019. How does chunking help working memory? *Journal of Experimental Psychology: Learning, Memory, and Cognition* 45, 1 (2019), 37.

[64] Anh Truong and Maneesh Agrawala. 2019. A Tool for Navigating and Editing 360 Video of Social Conversations into Shareable Highlights.. In *Graphics Interface*. 14–1.

[65] Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2016. Quickcut: An interactive tool for editing narrated video. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 497–507.

[66] Hanna Wallach, David Mimno, and Andrew McCallum. 2009. Rethinking LDA: Why priors matter. *Advances in neural information processing systems* 22 (2009).

[67] Miao Wang, Guo-Wei Yang, Shi-Min Hu, Shing-Tung Yau, Ariel Shamir, et al. 2019. Write-a-video: computational video montage from themed text. *ACM Trans. Graph.* 38, 6 (2019), 177–1.

A VIDEO EDITING GLOSSARY

- **Bins**: labeled folders used to organize videos and assets in a video editing tool
- **B-roll**: video and images that help illustrate the narrative but are not the main video
- **Paper edit**: a selection of transcript segments compiled into a text document. A paper edit is sometimes used in the early stages of editing to save selected content for use in the rough cut.
- **Rough cut**: an early stage output from the video editing process, typically with major story elements in place but lacking the final polishing elements, such as final narration, graphics, and effects
- **Screening**: an event in which collaborators or audience view intermediate or final edits and provide feedback
- **Sequence**: a set of clips assembled together on a video timeline
- **Stringout**: an intermediate editing output that displays selected video segments back-to-back on a timeline

B USER EVALUATION QUESTIONS

While much of the feedback from users came as they were exploring ChunkyEdit, at the end of the session, we asked the following open-ended questions:

(1) Which of the chunking methods present would be most helpful and why
(2) Which types of projects would this be most helpful for? Which types of projects would this be least helpful for?
(3) Please describe how this would change your workflow, if at all?
(4) Which features did you most enjoy? Which features did you least enjoy?
(5) Do you have suggestions for any additional features?

We also asked the following Likert-scale questions (on a scale of 1=not at all to 5=very much):

(1) To what extent do you feel ChunkyEdit would help you feel more organized?
(2) To what extent would the chunks help your production team find and focus on relevant parts of the interview?
(3) To what extent do you think this would have potential to speed up your editing process?
(4) To what extent do you feel this would reduce the amount of creative control you have over your editing process?
(5) To what extent do you feel this would help you in screenings or share intermediate edits with others?
(6) If this tool were included in an editing program like Adobe Premiere Pro, Avid Media Composer, or Final Cut Pro, how likely would you be to use it?

Additionally, we asked participants E6-E8 to rate the coherence of the clusters for each chunking method and the quality of the labels, in a similar fashion to the evaluation in Sec. 6.1.2, on a scale of 1=very incoherent/very low quality to 7=very coherent/very high quality for their own provided videos. We discuss these results in Appendix C.

ID	Video		Question		Answer 1: GPT-4		Answer 2: Hybrid		Answer 3: Keyword	
	Dur. (min)	Topic	Coh.	Label	Coh.	Label	Coh.	Label	Coh.	Label
E6	37	Non-profit volunteer	6	6	6	6	5	5	3	3
E7	44	Plant shop owner	6	4	5	5	4	2	2	2
E8	9	Music store employee	7	7	6	7	6	6	3	3

Table 5: Participants E6-E8 provided their own videos and rated the Coherence and label quality for the chunks produced by each of the four chunking methods (on a scale of 1-7). Overall participants preferred Question and Answer 1: GPT-4 to the other methods for their provided videos.

C USER FEEDBACK ON CHUNK QUALITY FOR THEIR OWN VIDEOS

In our user evaluation (Sec. 7) we asked E6-E8 to provide their own videos for the study. After completing the main study task of working toward a rough cut of a 2-5 minute profile video using any of the four provided chunking methods, they rated the coherence of the chunks and the quality of the chunk labels for each method (Table 5). While participants were free to use any of the chunking methods for the study task, all ultimately chose to use Answer 1: GPT-4.