

AQuA: Automated Question-Answering in Software Tutorial Videos with Visual Anchors

Saelyne Yang*
 saelyne@kaist.ac.kr
 Autodesk Research
 Toronto, Ontario, Canada

George Fitzmaurice
 george.fitzmaurice@autodesk.com
 Autodesk Research
 Toronto, Ontario, Canada

Jo Vermeulen
 jo.vermeulen@autodesk.com
 Autodesk Research
 Toronto, Ontario, Canada

Justin Matejka
 justin.matejka@autodesk.com
 Autodesk Research
 Toronto, Ontario, Canada

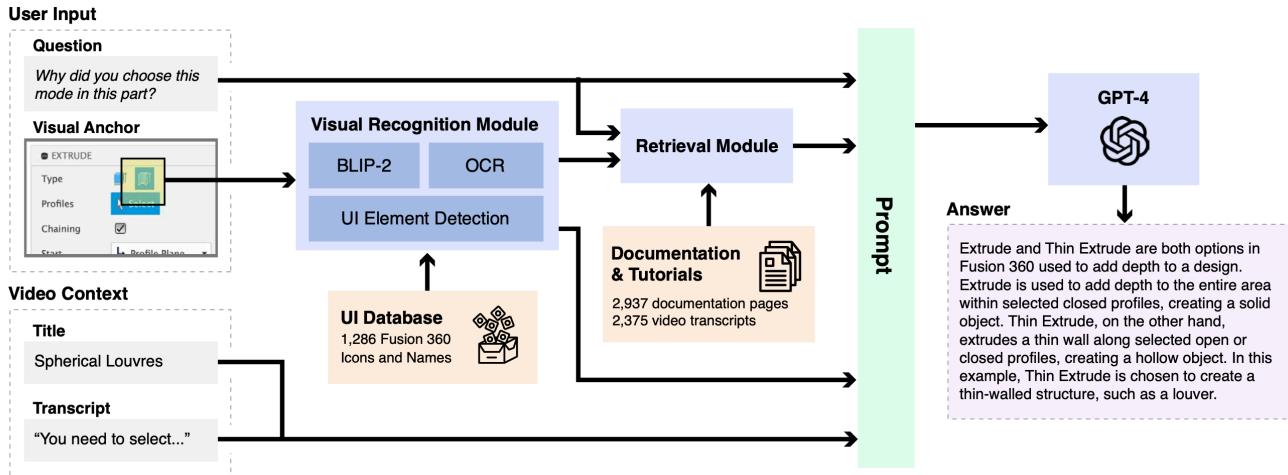


Figure 1: Overall architecture of our question-answer pipeline *AQuA*, which generates useful responses to questions made in software tutorial videos. Questions are accompanied by visual anchors, which are specific visual elements of interest in the video. The Visual Recognition Module generates a textual description of the visual anchor. Combining the description with the question, the Retrieval Module retrieves relevant articles to the queries. Resources in yellow boxes are software-specific materials (in this case, for Fusion 360). Along with these retrieved articles, the question text, and the visual anchor description, we include the title and relevant transcript sentences of the tutorial video and feed them into GPT-4 through crafted prompts.

ABSTRACT

Tutorial videos are a popular help source for learning feature-rich software. However, getting quick answers to questions about tutorial videos is difficult. We present an automated approach for responding to tutorial questions. By analyzing 633 questions found in 5,944 video comments, we identified different question types and observed that users frequently described parts of the video in questions. We then asked participants ($N=24$) to watch tutorial videos and ask questions while annotating the video with relevant

*Also with KAIST.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

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

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0330-0/24/05.

<https://doi.org/10.1145/3613904.3642752>

visual anchors. Most visual anchors referred to UI elements and the application workspace. Based on these insights, we built *AQuA*, a pipeline that generates useful answers to questions with visual anchors. We demonstrate this for Fusion 360, showing that we can recognize UI elements in visual anchors and generate answers using GPT-4 augmented with that visual information and software documentation. An evaluation study ($N=16$) demonstrates that our approach provides better answers than baseline methods.

CCS CONCEPTS

- Computing methodologies → Natural language generation;
- Human-centered computing → Interactive systems and tools.

KEYWORDS

tutorial videos, question answering, software learning, large language models, generative AI

ACM Reference Format:

Saelyne Yang, Jo Vermeulen, George Fitzmaurice, and Justin Matejka. 2024. *AQuA: Automated Question-Answering in Software Tutorial Videos with Visual Anchors*. 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, 19 pages. <https://doi.org/10.1145/3613904.3642752>

1 INTRODUCTION

Tutorial videos are a popular resource for people seeking to learn how to use feature-rich software applications [25, 44, 56]. These videos present workflows in great detail, with authors sharing their screens and often supplementing the workflow with verbal explanations. However, despite the abundant information they provide, viewers can face difficulties in understanding or following the content [8, 70]. To gain a better understanding of the material or clarify uncertain segments, viewers might rewind to a specific position and rewatch it or skip forward to anticipate the next steps [8, 73, 79]. To streamline this process, a number of systems have been proposed to help users navigate videos based on their current context and inquiry [7, 73].

However, there are instances where users still struggle to comprehend certain parts even after jumping around in the video, especially if the video doesn't address their specific queries. In such cases, they often leave questions in the comments section, requesting further explanations about specific parts of the video [54]. While timely answers to questions are crucial for effective learning from tutorials, obtaining answers from the community or the tutorial authors can take hours or days. In some instances, questions may even go unanswered. This delay in addressing questions can disrupt the learning process and discourage viewers from fully engaging with tutorial content.

To address the problem, we explored methods to automate the process of answering questions about tutorial videos. We first begin with an in-depth analysis of user question-asking behavior. To gain insights into this behavior, we collected a dataset of all 5,944 comments from the top 20 most popular video tutorials for Autodesk Fusion 360, a 3D Computer-Aided Design (CAD) software application. After identifying 663 questions in the comments, we further identified four main categories of questions: questions about the tutorial content ('Content'), questions regarding learners' personal settings or challenges in regard to the tutorial ('User'), questions concerning the video's meta-information ('Meta'), and questions not directly related to the content ('General').

We decided to focus on the first two categories due to their relatedness to the tutorial content (i.e., 'Content' and 'User'). A notable pattern that emerged in these categories was the tendency of users to reference parts of the video in their comments to provide temporal and spatial context. In particular, we noticed several cases of referring to the visual part of the video, which aligns with previous findings on general videos [74]. In contrast to this prior work [74], we observed this practice in software tutorial videos, in which it can be particularly evident since they feature visual demonstrations through screen sharing.

Inspired by these findings, our research delves further into the types of visual content that users reference in their questions. To explore this, we developed a system that allows users to ask questions by creating visual anchors, which are specific visual elements of

interest in the video that pertain to their questions. We conducted a study with 24 participants where they were asked to watch a tutorial video and formulate questions with relevant visual anchors. We selected four tutorial videos each for three software applications – Fusion 360, Photoshop, and Excel, resulting in a total of 12 tutorial videos. In this study, we collected 217 questions, each accompanied by one or more visual anchors relevant to the question.

Our analysis showed that the majority of visual anchors were related to specific user interface (UI) elements and the workspace of the software applications. Furthermore, nearly half of the questions required these visual anchors to supply essential contextual information. These findings underline the critical role of visual context in comprehending and responding to user queries in tutorial videos.

Based on our findings, we developed *AQuA*, a comprehensive pipeline to generate useful responses to questions that include visual anchors. Developed specifically for Fusion 360 as a case study, our pipeline identifies software UI elements in the visual anchors associated with questions and generates responses by leveraging the Large Language Model (LLM) GPT-4 [49], which is further enriched with specific knowledge about the software. We achieve this by drawing on official documentation and tutorial resources, which are generally available for most software applications.

We then evaluated our pipeline in a study with 16 Fusion 360 users. The results demonstrate that our pipeline produces more correct and helpful answers compared to baseline methods, and was the most favored. In the discussion, we outline design considerations for question-answering systems, providing insights into interactive and responsive learning experiences within the context of tutorial videos.

In summary, this paper presents the following main contributions:

- Two formative studies that uncover users' question-asking behavior in software tutorial videos.
- A comprehensive pipeline *AQuA*, which takes a novel multimodal approach with visual recognition and LLMs augmented with software-specific materials to generate answers to tutorial questions with visual anchors.
- An evaluation study that demonstrates the effectiveness of our pipeline in addressing user questions.

2 RELATED WORK

Our work presents an automated approach for answering questions asked on software tutorial videos that reference specific elements within the video. We discuss related work in the areas of software learning and UI understanding, video navigation and control, video question answering, and referencing techniques.

2.1 Software Learning, Tutorials, and UI Understanding

Software applications such as Adobe Photoshop and Autodesk AutoCAD provide rich functionality to accommodate users working on a wide range of tasks. However, it can be challenging to learn how to use such *feature-rich software*. Previous research has explored ways to simplify this learning process. For instance, Masson et al. [43] have focused on a "learning-by-doing" approach, introducing techniques that make users' trial-and-error cycle more meaningful.

However, while users can learn some aspects of the software by themselves, they may also encounter challenges and need to seek out additional help. A line of work has been conducted to support help-seeking behavior by facilitating searching for functionality, such as by using screenshots of software with Sikuli [75] or through multimodal input that refers to specific elements in the software in ReMap [16]. Other research has presented methods for offering contextual help, such as presenting web pages or videos in AmbientHelp [44], or specific segments within videos with RePlay [17]. Furthermore, researchers explored ways to better connect software users to their peers in the community. IP-QAT [45] enabled users to post questions directly within the software, while MicroMentor [24] facilitated getting help from experienced software users in real time.

Another way users learn about software is through tutorials. A body of work has focused on improving the usability of tutorials. Efforts have been made to enhance navigation to relevant parts of tutorial videos. Waken [5] recognizes and displays information about tools used in the tutorials, while other research such as ToolScape [26] and Fraser et al. [15] segment the videos into sections for easier navigation. Some systems have integrated the user's workflow into the video. Examples include SoftVideo [73], which provides real-time feedback on progress with tutorial content, Pause-and-Play [56] which controls video playback based on user progress, and Nguyen and Liu's work that allows users to learn directly from the video as if they are interacting with the software itself [47].

Beyond desktop applications, research has explored mobile and web applications, focusing on UI understanding for tasks like screen summarization and task automation. Using datasets like RICO [14] and WebUI [67], a number of approaches have leveraged view hierarchy information of screens. For instance, Screen2Vec [38] transforms UI screens into embeddings for tasks like screen retrieval, and Screen2Words [65] generates a summary of information that a UI screen contains. Combined with Large Language Models, Wang et al. [64] proposed an approach that enables conversational interaction with mobile UI. Recently, Spotlight [33] has been proposed, which does not require a view hierarchy but relies solely on visual screenshots to generate textual descriptions. In the realm of pixel understanding, Chen et al. [9] focus on detecting icon types, while Zhang et al. [76] focus on detecting UI element types, which has contributed to improved accessibility in mobile applications. In our work, we go beyond identifying the type of software UI elements in a single static image. We also identify the specific name of software UI elements in visual anchors that are captured and cropped in videos, by constructing a UI image database for a particular software application. Building on UI element understanding, our work aims to offer direct help while users learn from software tutorial videos by addressing their questions.

2.2 Video Navigation and Control

Users often encounter challenges when engaging with tutorial videos, struggling to comprehend or follow the content [8, 70]. In these situations, users may seek specific segments within the video to address challenges or resolve confusion [8, 73, 79]. To facilitate the process of locating needed segments in videos, several

researchers have proposed approaches to organize video content in a structured way [15, 26, 63, 71]. For example, Yang et al. [71] have demonstrated that displaying information types for each segment in how-to videos can enhance the search for answers within the video content. In efforts to enhance users' direct control over videos, various studies have explored how users can interact with videos conversationally [8, 39, 79]. For instance, Rubyslippers [7] allows content-based navigation of how-to videos using voice commands. These approaches empower users to pinpoint specific points of interest or points they need in their current context. However, there are still instances where users' queries or needs go beyond the information presented in the tutorial [30]. In our work, we address these cases where what users seek is not present in the video itself, by leveraging software-specific materials and the wealth of knowledge in Large Language Models.

2.3 Video Question Answering

Asking questions about the video content is a common user behavior [42, 54, 55]. Users often ask questions about parts of the video that need further explanation or request additional content [54]. Previous research in HCI has developed systems for question-answering in specialized domains such as programming [66], math [21], and children's general knowledge questions [31]. GVQA [62] and Kim et al. [28] have explored chart and graph comprehension through question-answering.

To address video question answering, the Computer Vision (CV) and Natural language processing (NLP) communities have introduced computational approaches and datasets. Among them, several benchmark datasets have been introduced that focus on how-to videos. For example, HowToVQA69M [69] contains question-answer pairs that are automatically generated from transcribed narrations. On the other hand, How2QA [37] and iVQA [69] collected questions and answers by presenting videos to crowd workers. In particular, TutorialVQA [13] and PsTuts-VQA [78] focus on *software tutorial videos*, collecting questions from crowd workers by presenting answer segments or having software experts craft questions. However, since these questions are artificially generated or automatically generated from transcripts, using these questions can be limiting when developing approaches to address questions from real-world users. Additionally, unlike these datasets, our approach goes beyond text-only queries by incorporating associated visual elements, which reflects how users would naturally ask questions.

In summary, our work takes a step further in assisting users with software tutorial videos by offering quick, automated and accurate responses to questions. Unlike previous video question-answering systems, our approach handles questions accompanied by visual elements, reflecting a common pattern in users' question-asking behavior on tutorial videos.

2.4 Video Referencing

Referencing specific audio or visual content within a video is a common practice during video interactions [59, 74]. Yarmand et al. have explored referencing patterns in user comments on videos, identifying types of references such as object, speech, and concept references [74]. While traditional video interfaces offer limited support for references (usually restricted to timestamps), this work

Category	Type	Definition	Example
Content	Concept	Asks about specific concepts explained in the video	<i>“Can you explain the difference between Press Pull and Extrude?”</i>
	Method	Seeks clarification about a particular action or process	<i>“How to import an existing path (e.g., from an SVG file or a Fusion sketch) into an electronics design?”</i>
	Reason	Asks the rationale behind certain instructions	<i>“Any reason for not using a construction line for the middle line?”</i>
	Alternative	Explores other ways to accomplish a task	<i>“Would the Intersect command work the same way for that?”</i>
User	Problem	Reports issues encountered while following the tutorials	<i>“Anyone else have issues with slow-downs when using several rectangular or circular arrays in sketches?”</i>
	Advice	Seeks personalized tips or guidance	<i>“I have a specific shape board where some components have to be located to fit in set openings. Advice?”</i>
Meta	Content	Asks about the topic or duration of the tutorial	<i>“What’s today’s topic? How long is this stream going on?”</i>
	Setting	Asks about the technical details of the tutorial production	<i>“What software are you using to screen-cast?”</i>
	Resource	Requests materials used in the tutorial	<i>“I was wondering where I can get the reference images?”</i>
General	Software	Asks about the software’s features	<i>“Is everything going to be migrated into Fusion in the future?”</i>
	Future Content	Suggests topics to be covered in upcoming tutorials	<i>“I hope in the future you will also make a real-time simulator in between circuit design and board design.”</i>
	Others	Asks about general topics	<i>“Is there a WhatsApp community for Fusion 360?”</i>

Table 1: Definition and examples of question categories and types derived from Formative Study 1. Minor grammar errors and typos in example comments are corrected.

suggests that the ability to easily refer to a part of a video enables a range of different applications, including enhanced engagement in live streams [12, 72].

The ability to refer to parts of a video fosters a clear understanding of what others are discussing and facilitates pinpointed feedback or areas of confusion. Mudslide [18] has shown that spatially contextualizing students’ confusion points on lecture slides can benefit both learners and instructors. HyperButton [27] has demonstrated that questions and answers anchored to specific frames can serve as valuable resources for future learners. As shown in Korero [11], this can also enhance mutual understanding among users and facilitate rich discussions about the video content. Video referencing can also enhance the learning experience. VideoSticker [6] allows users to extract specific objects from videos, helping learners take notes of the video content. Nguyen and Liu have introduced a tutorial video system where users can directly interact with videos by clicking on software elements in screencast videos [47]. In our research, we investigate video referencing behavior within the context of question-asking. This enables users to articulate their questions more effectively by making direct visual references to specific portions of the video.

3 FORMATIVE STUDY 1: SOFTWARE TUTORIAL VIDEO QUESTION ANALYSIS

To get insights into the requirements for the answer pipeline, we conducted two series of formative studies to understand users’ question-asking behavior in software tutorial videos. In our first study, we aimed to understand the types of questions users ask and identify the information required to provide answers. To achieve this, we gathered user comments from YouTube, focusing on Autodesk Fusion 360—a widely used and feature-rich 3D CAD software application—as our case study.

3.1 Method

We selected the top 20 popular archived live streams¹ from Fusion 360’s official YouTube channel for our analysis. These live streams are instructional videos that are created by the official channel with the purpose of explaining how to perform tasks or sharing tips with Fusion 360 learners. We chose archived live streams as opposed to non-live tutorial videos as the live streams allowed us to investigate question-asking behavior in a comprehensive way, considering both comments made during real-time viewing

¹www.youtube.com/@adskFusionstreams

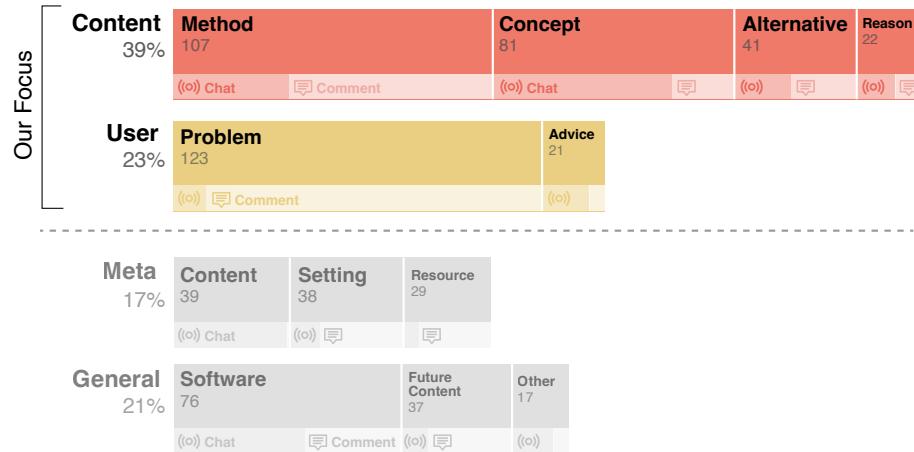


Figure 2: Categories and Types of questions identified from the analysis. Each row represents a category and each block represents a type. Under each block, the areas on the left and right represent live chat and comment data, respectively. Our focus is on *Content* and *User* questions, as these are vital for comprehending the tutorial and can often be answered without the involvement of the tutorial authors or software vendor.

where immediate help may be available, and comments made during asynchronous viewing after the live stream had ended, where *live* help is not accessible. We gathered a total of 5,944 messages, which included 3,905 live chat messages sent during the streams and 2,039 comments posted on the same archived videos. We used Chat Downloader [40] to collect live chat messages and used the YouTube Data API [20] to gather comment data.

We examined the collected comments to get a sense of the types of questions that were asked. We first filtered for comments that were (1) asking questions or making requests, (2) posted by viewers (not by the tutorial author or moderator), and (3) initial comments (not replies). This resulted in 633 questions out of the 5944 comments. Similar to Yarmand et al. [74], the lead author performed thematic coding of the set of 633 questions and iteratively discussed with the other authors to validate the codes and resolve any conflicts. After finalizing the codes, we grouped them into four main categories, reflecting the overarching themes of the questions.

3.2 Results

Table 1 provides definitions and examples for the 12 distinct types of questions we identified, which are divided into under four main categories:

- **Content (39.7%):** questions about the tutorial content presented in the tutorial.
- **User (22.6%):** questions about the viewer's settings or challenges in regard to the tutorial.
- **Meta (16.7%):** questions about the tutorial video's meta-information.
- **General (20.9%):** questions that are not directly related to the tutorial content.

The '**Content**' category encompasses questions related to the tutorial content presented in the tutorial, such as questions about concepts explained in the video (e.g., “*Can you explain the difference*

between Press Pull and Extrude?”) or questions that ask about the rationale behind certain instructions (e.g., “*Is there a reason for not using a construction line for the middle line?*”). ‘**User**’ focuses on questions about the viewer’s personal settings or challenges, such as questions reporting issues they encounter (e.g., “*That marking menu doesn’t seem to work. How do I fix it?*”) or those seeking tips or guidance (e.g., “*I have a specific shape board where some components have to be located to fit in set openings. Advice?*”). ‘**Meta**’ focuses on questions about the tutorial video’s meta-information, such as the technical details of the tutorial production (e.g., “*What software are you using to screencast?*”) or materials used in the tutorial (e.g., “*I was wondering where I can get the reference images?*”). Finally, **General** includes questions that are not directly related to the tutorial content, such as those asking about the software’s features (“*Is everything going to be migrated into Fusion in the future?*”).

Overall, ‘**Content**’ and ‘**User**’ questions are related to the tutorial content, seeking comprehension or practical help in understanding the tutorial. In contrast, ‘**Meta**’ and ‘**General**’ questions concern the meta-information or information unrelated to the tutorial’s core content. These inquiries typically require insights from either the tutorial author (e.g., providing material resources) or software developers (e.g., detailing new feature timelines).

3.3 Implications on the Answer Pipeline

In our exploration of automated methods to address questions, we specifically focus on ‘**Content**’ and ‘**User**’ questions, as these types of questions are often time-sensitive and crucial for enhancing comprehension and the learning experience with the tutorial content. Moreover, they can often be answered without the involvement of the tutorial authors or software vendors.

Since these types of questions have direct relevance to video content, a notable trend that emerged from our analysis was the frequent references to video in these questions, sometimes explicitly

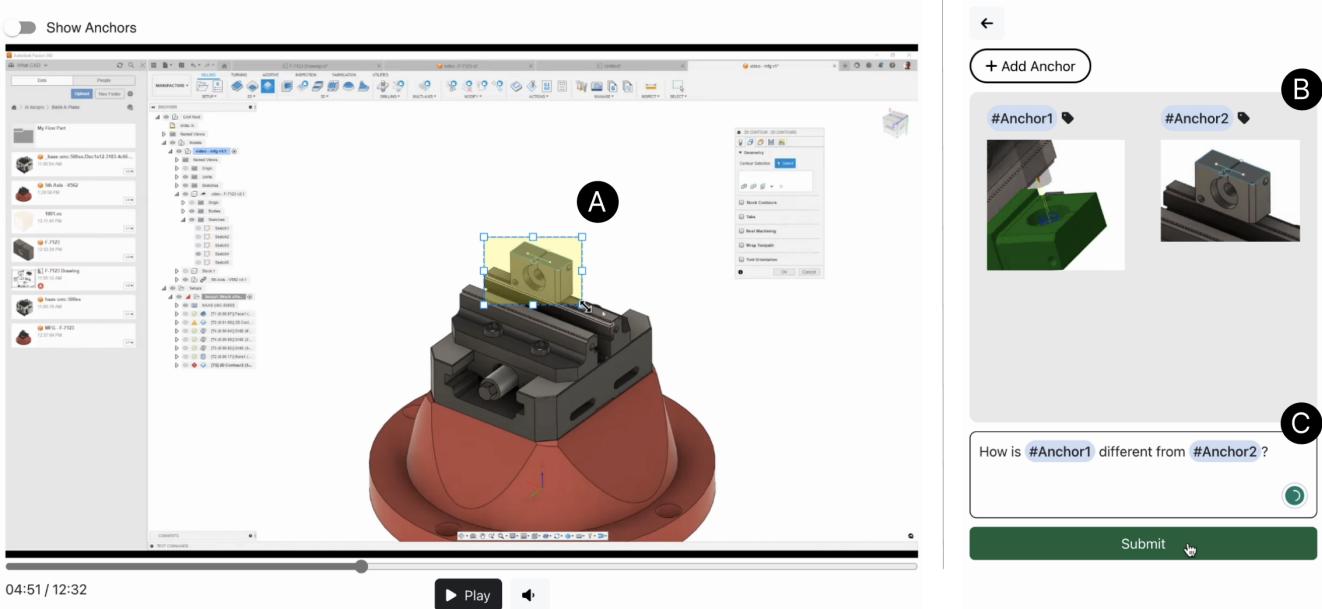


Figure 3: The system used for collecting questions with visual references. (A) Users can draw anchors on parts of the video they want to ask questions about, **(B)** which will be added to a temporary gallery. **(C)** Users can refer to each anchor in their questions.

citing timestamps, which echoes findings from prior research on referencing behavior in comments on a variety of videos [74]. It implies that our answer pipeline design should account for what in the video a question was about and the context of the tutorial when the question was posed. Furthermore, since these questions demand a deep understanding of the software, our pipeline should be able to provide accurate and software-specific answers.

Additionally, we observed that users often described visual elements of the video in their queries, detailing UI elements within the software (e.g., “[...] I do not have a ‘design’ option on the left hand drop down and I do not have a ‘constraints’ panel at the top [...]”). This behavior of visual references aligns with findings from earlier studies [41, 72, 74], which can be particularly prominent in software tutorial videos where the author conveys the workflow via screen sharing [47]. However, articulating visual objects in text can be challenging [29], and conventional video interfaces typically support only timestamp references in addition to text. This observation prompted our next study, where we aimed to explore the types of references people make when equipped with a tool that allows for visual references.

4 FORMATIVE STUDY 2: ANALYSIS OF QUESTIONS WITH VISUAL REFERENCES

To further investigate the visual referencing behavior in software tutorial videos, we conducted a second study to delve deeper into what people specifically refer to when mentioning specific parts of a video when asking questions. To accomplish this, we ran a data collection study where participants were instructed to watch a software tutorial video, ask questions, and annotate the video to

identify visual parts of the video relevant to their questions. Next, we describe the system we built, the study protocol, and the analysis of the results.

4.1 Data Collection System

We developed a web application that allows participants to ask questions by directly referring to a specific part of the video (Figure 3). Participants can draw an anchor on the video screen (Figure 3-A) that they want to ask a question about. These anchors are then saved to a temporary gallery (Figure 3-B), which allows participants to directly link to these anchors in their questions (Figure 3-C). Anchors can also be labeled with a hashtag (e.g., #palette) for easier reference. Clicking on an anchor will populate its label into the question box. Participants can include multiple anchors in a question if they wish to refer to different parts of the video. This can be particularly useful for questions that need to refer to an action that spans a longer segment of the video.

4.2 Study Design

4.2.1 Video Selection. To ensure we cover diverse types of feature-rich software, we selected tutorial videos for three different applications: Autodesk Fusion 360, Adobe Photoshop, and Microsoft Excel. We selected four videos for each application that satisfy the following criteria: (1) between 4 and 6 minutes in length, (2) published within the last three years, and (3) have more than 1,000 views to ensure the quality of the content. The authors manually verified the videos to ensure that the tutorial was high quality, relatively easy to follow, and well explained. This resulted in a total of 12 tutorial videos (Table 2).

Software	Video Topic	Video URL	Length	Remark
Fusion 360	Spherical Louvres	youtu.be/K0bKT5PmYx0	5:41	TO
	4 Ways to Export to DXF	youtu.be/f28TKYsqd6w	4:23	-
	Serrated Washer Using Loft To A Point	youtu.be/fM0AwDLq6_E	5:31	-
	Simulating Motion	youtu.be/pGFY-ZXm6G0	5:22	-
Photoshop	Poster Design	youtu.be/yFHfOlEVcxS	4:19	TO
	Soft & Dreamy Glow Effect	youtu.be/4YaQ5yHQDtG	4:32	-
	Using Animation	youtu.be/ugPYmEGIKxs	4:15	PiP, FS, TO
	Applying a Dual Lighting Effect	youtu.be/Q3sa4uraBkk	6:00	FS, TO
Excel	Creating Pivot Tables	youtu.be/NrUqtE7X05E	5:40	-
	Building a Power Query Function	youtu.be/UOCderIkdxM	5:01	-
	Handling Duplicates, Triplicates, Quadruplets	youtu.be/YsC6NYwHAnA	4:22	PiP, FS, TO
	CountIf and Pie Charts	youtu.be/8osaUuI-OUU	5:58	-

Table 2: Tutorial videos used in the question collection study. All videos are screencast tutorial videos. Lengths are in minutes:seconds. **PiP:** The talking head is displayed in picture-in-picture mode. **FS:** The talking head is shown in full screen, occasionally appearing in the video. **TO:** The video includes text overlays.

4.2.2 Participants. We recruited participants from software-specific community forums such as subreddits for these software applications (e.g., r/Fusion360/). We required participants to have at least some prior experience with the software to collect quality questions. To ensure this, they were asked to share details about their level of experience with the software, including how long they have been using it and the main tasks they typically perform with it. We initially recruited 33 participants, of which we excluded 9 after failing quality control measures (see Section 4.2.3), resulting in 24 participants who participated in the study (16 male, 6 female, 2 non-binary, mean age=30.2). The 24 participants were divided evenly over each software application, with 8 participants for each. Each of the 12 selected tutorial videos was assigned to two participants. As the study was done online, participants were expected to be able to access the web with their own desktop or laptop.

4.2.3 Task. The study was conducted in an asynchronous remote setting. After the researchers confirmed a participant's eligibility regarding their experience with the software, the participant received a URL to our system. The process began with an informed consent form, followed by a demographic survey. Participants then went through a brief tutorial detailing how to use our system to ask questions and make visual references. They were instructed to imagine themselves as someone watching this video to improve their skills and ask questions about the tutorial. Specifically, participants were asked to pose at least 10 questions, each accompanied by one or more visual anchors, which are visual elements in the video the question is about. They could redraw an anchor until they were satisfied with it. They could also rename the anchors and easily refer to them in the question by either clicking on the anchor or typing its name. At the end of the study, we asked for optional open-ended feedback about the overall study. As a quality control measure, we excluded participants with more than five non-question responses. These included those expressing appreciation for the tutorial video (e.g., "This is well explained") or making suggestions (e.g., "It would be better to specify a number here.") The study took around 40 minutes, and the participants were compensated with a \$30 USD gift card for their participation.

4.3 Results

From the study, we collected a total of 256 responses from 24 participants, each accompanied by at least one relevant visual anchor. We filtered out non-question responses, such as those expressing appreciation or making suggestions, leaving us with 217 questions in total. These questions were composed of 205 explicit questions and 12 questions which implicitly sought help (e.g., "#Anchor1 doesn't seem to work and I have no idea how to deal with it"). Below, we present the analysis of questions and associated visual anchors that we collected.

4.3.1 Questions and Visual Anchors. Each question was accompanied by one or more visual anchors. While most of the questions (91.7%) were associated with a single visual anchor, 18 out of 217 questions had two or more visual anchors for elaboration. Specifically, 16 questions used 2 anchors, and 2 questions used 3. These multiple anchors served various purposes, such as indicating the beginning and end of an action, posing questions about a 'before' and 'after' scenario, and suggesting alternative options to consider.

4.3.2 Visual Anchor Types. We first analyzed the types of visual anchors that the participants used in their questions. Two authors initially discussed the types of visual anchors based on their roles in the software. Following this, the lead author annotated each visual anchor according to its type and no ambiguous cases were encountered. Specifically, our analysis identified five distinct types of visual anchors:

- **UI Elements (52.7%):** Anchors on User Interface elements like tools, menus, and panels within the software.
- **Workspace (34.6%):** Anchors placed on the workspace of the software where the tutorial creator is performing the main task, such as the 3D CAD model in Fusion 360, the photo in Photoshop, and the grid of cells in Excel.
- **UI+Workspace (5.5%):** Anchors that capture both UI Elements and Workspace, including a full screenshot of the interface.

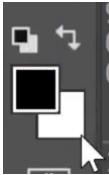
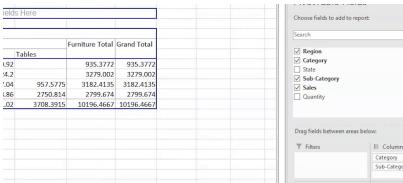
Type	Visual Anchor & Question
UI Elements	 <p>Can you explain more about what that icon is used for?</p>
Workspace	 <p>How do I determine my pattern spacing?</p>
UI+Workspace	 <p>Can you put a second field into the Row section as well as the Columns section?</p>
Annotation	 <p>What is the alternative command for this?</p>
Miscellaneous	 <p>Where did the calculator come from? Can you do this inside Fusion? How?</p>

Table 3: Types of visual anchors and associated questions collected in our study.

- **Annotation (3.8%):** Anchors attached to textual or graphical annotations that the tutorial creator has overlaid on the primary video footage.
- **Miscellaneous (3.4%):** Anchors that are not related to the software, such as those placed on external applications, video annotations (like pop-ups showing entered keyboard commands), or the face of the tutorial creator.

Most anchors were related to either the UI elements (52.7%) or the workspace (34.6%), with some capturing both (5.5%). Combining these three categories amounted to 92.8% of anchors. The questions associated with these anchors involved asking about the functionality of specific tools or about detailed methods of the workflow.

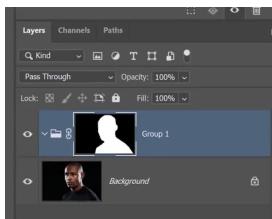
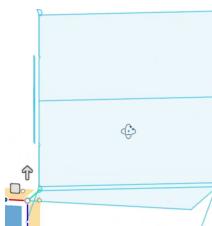
Role	Visual Anchor & Question
Necessary	 <p>Does this feature provide the same result as the pivot table template chosen in this video?</p>
Useful	 <p>Can you explain more about the layer mask option? What is it used for? and its uses?</p>
Irrelevant	 <p>What kind of file does it create that can't be opened by these programs?</p>

Table 4: Roles of visual anchors and associated questions collected in our study. Text in bold refers to the associated visual anchor.

Table 3 shows example questions and related visual anchors for each type.

4.3.3 Role of Visual Anchors. We then examined the level of involvement of visual anchors in the questions. We first found that the visual anchors could either be crucial for interpreting the questions (**Necessary**), merely add extra context (**Useful**), or not relevant to the question at all (**Irrelevant**):

- **Necessary (47.5%):** The question by itself is unclear or lacks context, and thereby visual anchors are required to fully comprehend the question.
- **Useful (49.3%):** The question is understandable without additional context, and visual anchors are relevant to the question's context.
- **Irrelevant (3.2%):** Visual anchors have no connection to the question.

The lead author annotated each visual anchor according to its role. Ambiguous cases (16 out of 235) were resolved through discussion with another author.

We identified that almost half of the questions (47.5%) required the accompanying visual anchors to be fully understood. These questions often used referential terms like “this” or “it”, which

directly pointed to the visual anchors for context. The remaining half (49.3%) mostly consisted of questions where the anchors played a supportive role. Only a small fraction of questions (3.2%) were found to have irrelevant anchors. These questions were mostly associated with what the tutorial author talked about when the visual anchor was drawn, or they were broad questions related to the general topic of the tutorial. Table 4 provides examples of questions and their corresponding visual anchors, categorized by the role each anchor played.

4.4 Implications on the Answer Pipeline

From the study, we could see that people mostly refer to software UI elements or the workspace when asking questions, and nearly half of the questions required these visual anchors to provide important contextual information. These findings highlight the importance of comprehending visual references associated with a question, particularly those related to software UI elements, which were found to be the majority of visual references.

4.5 Design Goals

From the two formative studies, we derived the following design goals of an answer pipeline for software tutorial videos. First, our answer pipeline should consider what was happening in the video when the question was asked (DG1). Second, it should be able to understand (multiple) visual references associated with a question, including software UI elements (DG2). Lastly, it should provide accurate and useful answers that reflect the software-specific knowledge (DG3).

- DG1: Consider the video context when the question was posed.
- DG2: Understand visual references associated with the question, including software UI elements.
- DG3: Provide accurate and useful software-specific information in answers.

5 AQuA: QUESTION-ANSWER PIPELINE

Based on our design goals (Section 4.5), we designed *AQuA*, a question-answer pipeline that generates useful responses to questions with visual anchors (Figure 1). We applied our pipeline to Fusion 360 to demonstrate the potential of this approach. While we tailored the pipeline to have specialized knowledge about Fusion 360, this approach is generalizable to other feature-rich software applications. *AQuA* will work for any software application for which tool or command names accompanied by icons or screenshots of corresponding UI elements, and a sufficiently large set of software documentation or existing tutorial materials are available. This would, for example, be the case for many other feature-rich software applications such as Adobe Photoshop or Illustrator, Autodesk AutoCAD or Maya, or Microsoft Word.

5.1 Overall Architecture of *AQuA*

Figure 1 illustrates the overall architecture of our question-answer pipeline. It takes the question text and visual anchor(s) as inputs. First, our Visual Recognition Module identifies the UI element in the visual anchor and generates a textual description of it (DG2,

Section 5.2) so that it can be easily provided to GPT-4 (which, at the time of writing, did not yet support multimodal prompts in its API). Then, we combine the visual description with the question text and search a database of software documentation and tutorial materials for articles relevant to the query (DG3, Section 5.3). Along with these retrieved articles, we include the video title and relevant transcript sentences to give context to the question (DG1, Section 5.4). All of these elements—question text, visual anchor descriptions, retrieved articles, and video context—are fed into GPT-4 through crafted prompts. These prompts instruct the model to provide answers to questions by specifying the components in the following order: Relevant articles; tutorial titles and transcripts; questions; and visual anchors. The prompts provided as input for GPT-4 can be found in Appendix A.3.

5.2 Visual Recognition Module

To identify visual anchors and generate textual descriptions, our pipeline includes a Visual Recognition Module (Figure 4) that is composed of Image Captioning, UI Element Detection, and Optical Character Recognition (OCR). Below, we explain these submodules.

5.2.1 Image Captioning. To obtain descriptions of general visual anchors such as parts of the application workspace, we use BLIP-2 [34], a visual-language model that shows high performance on zero-shot image captioning. BLIP-2 recognizes objects within an image, thereby constructing a well-defined image description that captures the core information. This can be particularly useful when the visual anchor contains objects that the tutorial author is working on (e.g., “*a gray steel washer on a white background*”) or generic objects that the author pulls up (e.g., “*a calculator with the number 360 on it*”).

5.2.2 UI Element Detection. Our formative study revealed that more than half of the visual anchors are related to UI elements. While visual-language models like BLIP-2 are excellent at describing real-life scenes (e.g., “*a couple with a dog on a leash on the beach at sunset*”) or general appearances of objects (e.g., “*a cylindrical object with a hole in it*”), they fall short in recognizing software-specific elements. For instance, for the visual anchor shown in Figure 4, BLIP-2 generates the description: *a blue cube with an arrow pointing up*, which, while not entirely wrong, is too generic and not helpful to identify the icon in the visual anchor as the Extrude tool in Fusion 360.

To more accurately recognize these UI elements, we created a UI element image database. This was done by crawling software help resources from the official Fusion 360 documentation [4]. We exploited the fact that the tool image and its name follow a specific pattern arranged in HTML unordered list items, with the tool name preceding the image. To extract the name and image, we accordingly parsed the HTML files based on this identified pattern. To further enrich our data, we ran Fusion 360 and extracted additional UI information by running a script to save all command icons. This approach yielded a total of 1,286 images along with their corresponding names—446 from the documentation and 840 from the software commands.

By leveraging this UI icon database that we created, our pipeline is able to identify UI elements within visual anchors by comparing

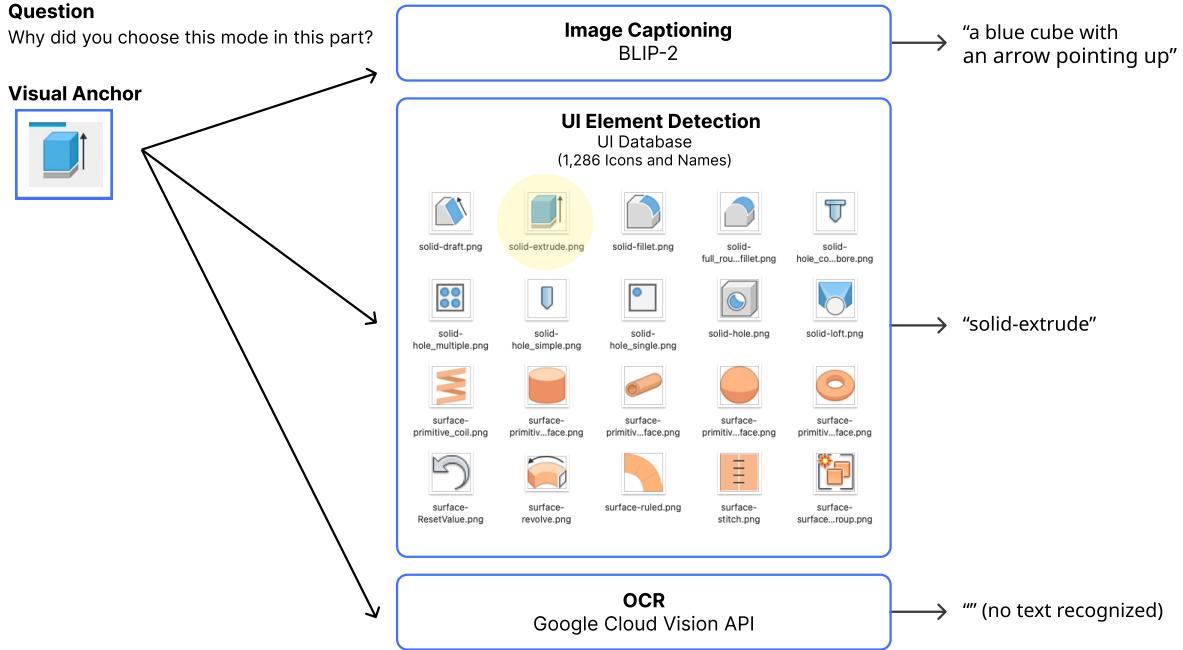


Figure 4: Our Visual Recognition Module is composed of Image Captioning, UI Element Detection, and Optical Character Recognition (OCR). We use BLIP-2 [34] to obtain a general description of the visual anchor in case it contains generic or workspace objects, and the Google Cloud Vision API [19] to detect any textual information in the anchor. For UI Element Detection, we first run UIED [68] to determine if there are multiple UI elements in the anchor. Then, we apply feature matching and template matching between each element in the anchor and those in the UI database. If the matching score exceeds a certain threshold, we retrieve the element's name.

them to the images in the database. If the dimensions of an image exceed a certain threshold (in our case, 100 pixels in both width and height), we first proceed with an initial UI element detection pass, as the image could contain multiple elements due to its size. We use UIED [68] to generate bounding boxes around each UI element within the image. Following this, we apply an image similarity algorithm to each bounding box to identify the corresponding image in our database. First, we use OpenCV's feature matching [51] to extract and compare visual features, thereby establishing matched features. We then select the top five candidate images based on the number of matched features. To refine our search, we apply template matching to the five candidate images using the Normalized Cross-Correlation Coefficient [52] to locate instances of a template image within a larger search image. If the highest template score exceeds 0.5, we deem it a successful match and retrieve the element's name.

5.2.3 OCR. Lastly, since UI elements often contain text [75], we employ Optical Character Recognition (OCR) to extract textual information using the Google Cloud Vision API [19]. This can be particularly useful when the visual anchor includes a menu or panel that lists the names of various functionalities.

We run the above three modules and combine their output to generate a final textual description for each visual anchor associated with a question. This description is provided as part of a prompt to GPT-4 when generating answers and is also used to retrieve relevant materials in the Retrieval Module (Section 5.3).

5.3 Retrieval Module

Large Language Models such as GPT-4 can suffer from hallucinations, which is when they generate false information [77]. To enhance the quality of responses and make GPT-4 generate answers specific to the software, we further enrich the pipeline with Retrieval Augmented Generation (RAG). RAG is an approach that combines retrieval and generation, producing specific and factual responses [32]. This is achieved by incorporating relevant information retrieved from a knowledge base when generating a response to a prompt. In our case, we constructed a knowledge base by gathering Fusion 360 articles. We used the same documentation source used in Section 5.2, which contains 2,937 HTML files. Additionally, we gathered 2,375 Fusion 360 tutorial videos from Autodesk Screencast [3, 22] and transcribed their audio using Amazon Transcribe [2]. We segmented each content into chunks, ensuring that each chunk did not exceed a certain length (i.e., 1,600 tokens). We then obtained embeddings for each chunk using OpenAI's text embedding model (`text-embedding-ada-002`) [50]. This resulted in a total of 5,635 *article chunks* (i.e., either documentation or tutorial transcripts) and their embeddings.

Using the same embedding model, our pipeline gets embeddings of the provided question text and visual anchor description. We then compare these embeddings with that of each article in the knowledge base we constructed, retrieving the top 50 articles based on the cosine similarity ranking between the embeddings. The top

Batch	Participant	Occupation	Experience (years)	Main Tasks
Batch 1	P1	CAD Engineer/Project Manager	4-5	Assemblies, Sheet Metal
	P2	3D Printer Technician	4	3D Modeling, Assemblies
	P3	Freelancer	4	Simulation, 3D Modeling, Data Management
	P4	Software Developer/Designer	4	3D Graphic Design
	P5	Retired	7	3D Modeling
	P6	Freelancer	1	Simulation, Data Management
	P7	Data Analyst	10	Simulation
	P8	Localization Program Manager	1	Design
Batch 2	P9	Consultant	7	3D Modeling
	P10	Software Developer	3	3D Graphic Design, Simulation
	P11	Mechanical Design Engineer	5	3D Modeling, Rendering, Manufacturing
	P12	Customer Service Representative	8	Parametric Modeling, Drawing, Renders
	P13	Routesetter	2-3	3D Modeling
	P14	Product Designer	4	3D Modeling, Product Design
	P15	Product Manager	8	Manufacturing, Design
	P16	Content Strategist	3	Demonstration

Table 5: Participant information including current occupation, number of years of experience, and the main tasks they perform with Fusion 360.

50 articles represent the top 1% among all articles, but the number of articles used by the pipeline typically ranges around the top 20, depending on their length. This is because we stop appending articles to the collection once adding another exceeds the input token limit of GPT-4 (i.e., 8,192 tokens in our case). The resulting collection of articles is then provided as part of a prompt to GPT-4 when generating answers.

5.4 Video Context

Finally, to provide the video context when the question was posed, we include both the title and a relevant segment of the transcript of the tutorial video about which the question was asked. We extract the two transcript sentences that are adjacent to the timestamp where the visual anchor was captured. Specifically, we locate the nearest sentence whose starting timestamp does not exceed the timestamp of when the visual anchor was captured, and then extract both that sentence and the preceding one. If there are multiple visual anchors, the two relevant transcript sentences for each visual anchor are concatenated. These title and relevant transcript sentences are provided as part of a prompt to GPT-4 when generating answers.

For demonstration and testing purposes, we transcribed the 12 videos we used in the second formative study (Section 4) using Whisper [57]. However, given that popular websites for tutorial videos, such as YouTube, already support downloading transcripts for many videos and continue to improve in accuracy, we imagine that in the future, transcripts with detailed timestamps could be easily provided as input to the pipeline in addition to the video.

6 EVALUATION

We evaluated our proposed question-answer pipeline with 69 questions from four Fusion 360 videos that were gathered during our formative study (Section 4). These questions consisted of 87% ‘Content’

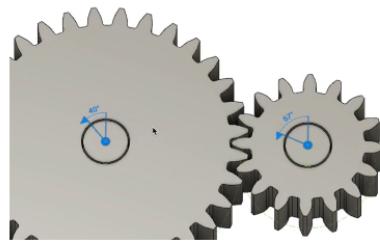
questions and 13% ‘User’ questions. We generated answers to these questions under three different conditions: (1) **Question-only**, where only the question text is provided to GPT-4; (2) **Question + Video**, where the question text together with the video title and transcript sentences are provided to GPT-4; and (3) **Full Pipeline (AQuA)**, in which also the visual anchor and relevant articles are provided to GPT-4. We treat Question-only and Question + Video as baseline methods, since the former could be seen as a plain GPT-4 while the latter could be considered a plain GPT-4 with a bit more information from the video. We instructed GPT-4 to generate answers in 50 words or less in all conditions. To reduce variability in GPT-4’s responses, we set the temperature parameter to 0, which minimizes randomness in the generated answers. We also used GPT-4-0613, a snapshot of GPT-4 that was available at the time of our study. The prompts used for each condition can be found in Appendix A.

6.1 Participants

We recruited participants with varying levels of experience in Fusion 360 (Table 5) to evaluate the answers generated under each condition. We initially recruited 21 participants, of which we excluded 5 after failing quality control checks (see Section 6.2), resulting in 16 (2 female, 14 male, mean age=34.5) who participated in the study. We recruited participants from a Fusion 360 subreddit (N=12) as well as people who develop the Fusion 360 software as part of their day job (N=4). We divided the 69 questions into two batches to accommodate for a reasonable study duration (30–60 minutes). The two batches contained 34 and 35 questions respectively, and we assigned 8 participants to each batch. Where appropriate, we discuss the results for each batch separately.

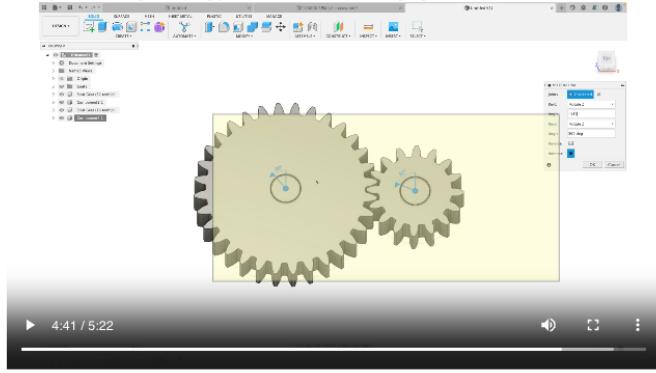
Question 27 / 35

is there a way they can rotate with the same speed



#Anchor1 (4:41)

You can refer to the video below if you need more context for the question.
Clicking on each image will bring you to the timestamp and highlight the anchor in the video.



Simulating Motion in Fusion 360 - Assemble Joints & Motion Links

Please select the answer you like the most by clicking on its ❤️ icon and rate each answer in terms of Correctness and Helpfulness.

❤️ Answer 1

Yes, you can make two gears rotate at the same speed in Fusion 360. This can be achieved by using the "Motion Link" feature under the assembly tab. You can set the ratio of rotation between the two gears to 1:1, meaning they will rotate at the same speed.

This answer is **correct**.This answer is **helpful**.

❤️ Answer 2

Yes, in Autodesk Fusion 360, you can control the speed of rotation by adjusting the settings in the "Orbit" tool or by using the "Animate" feature in the timeline for consistent rotation speed.

This answer is **correct**.This answer is **helpful**.

❤️ Answer 3

Yes, in Fusion 360 you can adjust the speed of rotation to match between different components. This can be done by setting the same motion parameters for the components you want to rotate at the same speed.

This answer is **correct**.This answer is **helpful**.

(optional) Share your reason for selecting that answer as your favorite

NEXT

Figure 5: The system used in our pipeline evaluation study. The participant can see the question, the video that the question was asked about at the right timestamp and with the visual anchor highlighted, and three generated answers in random order. They were asked to rate each answer in terms of its correctness and helpfulness on a scale of 1 to 7, and select their favorite answer among the three. Optionally, they could provide reasons for selecting their favorite answer.

6.2 Task and Measures

Participants were presented with each question, along with related visual anchors and the tutorial video that highlights the part where

the anchor was captured (Figure 5). The answers generated under the three conditions were randomly ordered for each question. Following the metrics of free-form question-answering in natural

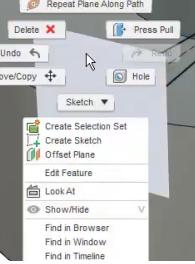
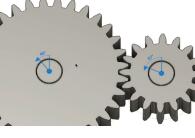
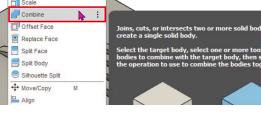
Question & Visual Anchor	Question-only	Question + Video	Full Pipeline (AQuA)
(1) How did you get this menu to appear? 	The menu in Autodesk Fusion 360 typically appears by either clicking on the relevant icon on the toolbar or by right-clicking to bring up a context menu. The exact method may vary depending on the specific menu you're referring to in the tutorial.	The menu in Autodesk Fusion 360 typically appears when you right-click on the workspace or when you select a specific tool from the toolbar at the top of the screen. In this case, the menu likely appeared when the instructor selected the "Create Sketch" tool.	The menu shown is the Marking Menu in Autodesk Fusion 360. It appears when you right-click anywhere in the canvas. This radial menu provides quick access to frequently used commands. To activate a command, drag the cursor in the direction of the command, then click anywhere in the highlighted wedge.
(2) Is there a way they can rotate with the same speed? 	Yes, in Autodesk Fusion 360, you can control the speed of rotation by adjusting the settings in the "Orbit" tool or by using the "Animate" feature in the timeline for consistent rotation speed.	Yes, in Fusion 360 you can adjust the speed of rotation to match between different components. This can be done by setting the same motion parameters for the components you want to rotate at the same speed.	Yes, you can make two gears rotate at the same speed in Fusion 360. This can be achieved by using the "Motion Link" feature under the assembly tab. You can set the ratio of rotation between the two gears to 1:1 , meaning they will rotate at the same speed.
(3) Why are we combining the bodies? 	Combining bodies in Fusion 360 helps to create a single, unified object. This is useful for simplifying the design process, reducing complexity, and ensuring that different parts fit together correctly in the final product.	Combining the bodies in Fusion 360 allows you to merge multiple bodies into a single body. This is useful for creating complex shapes and structures, and for simplifying your design process.	Combining bodies in Fusion 360 allows you to join, cut, or intersect multiple solid bodies to create a single solid body. This is useful for creating complex shapes or structures that are composed of multiple parts. It can also be used to cut out sections of a body using another body as a tool.

Table 6: Example answers generated from each condition. (1) Full Pipeline (AQuA) accurately identifies the exact menu name and provides relevant information, whereas Question + Video refers incorrectly to a menu by relying on the transcript. (2) Full Pipeline provides detailed instructions while Question-only generates inaccurate information. (3) Full Pipeline offers additional details about various operations, making it more informative than others. Note that typos in the questions have been corrected for clarity; however, they were not corrected when generating answers.

language generation [58], participants were asked to rate each answer in terms of its correctness (i.e., how accurate the answer is) and helpfulness (i.e., how well the answer addresses the question) on a 7-point Likert scale. Additionally, they were asked to choose their preferred answer among the three options and could optionally provide the rationale behind their choice. As a quality control measure, we excluded participants who had a preferred answer ranked lower than another option on both correctness and helpfulness. This resulted in 2 and 3 participants being excluded from Batch 1 and 2, respectively. At the end of the study, we asked for optional open-ended feedback about the overall answers and the study. For their participation in the 60-minute study, participants were compensated with a \$60 USD gift card.

6.3 Results

6.3.1 *Full Pipeline Favored Over Baseline Answer Generation Methods.* In selecting a preferred answer, the answer generated by Full Pipeline was favored most frequently (55.4%), in comparison to the Question-only (17.8%) and Question + Video condition (26.8%), as shown in Figure 7. The trend remained consistent for both Batch 1 and Batch 2 – 15.8%, 29.4%, and 54.8% for Batch 1 and 19.6%, 24.3%, and 56.1% for Batch 2, corresponding to Question-only, Question + Video, and Full Pipeline, respectively. In the open-ended feedback in which participants could provide feedback on why they selected Full Pipeline answers as their favorite, they noted that these answers were both accurate and specific (P4, P12, P14, P15, P16), provided more detail than others (P13, P14, P16), and followed the right sequence of addressing a question (P8, P16). For instance,

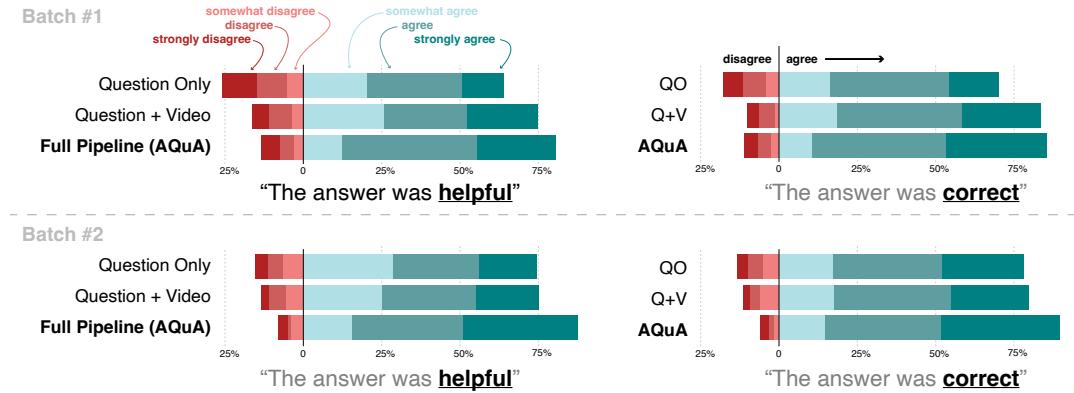


Figure 6: Distribution of Likert scale responses on Correctness and Helpfulness. Full Pipeline shows the highest correctness and helpfulness scores in both batches. Responses of "neither agree nor disagree" are omitted from the chart for clarity and readability.

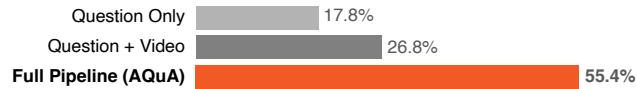


Figure 7: Results of the favorite answer selection. Answers generated from the Full Pipeline were selected as the favorite most often.

in the top example from Table 6, Question-only failed to comprehend the specific menu referred to in the question and Question + Video incorrectly inferred that it related to another tool based on the transcript instructions. However, Full Pipeline accurately identified the menu as the Fusion 360 *Marking Menu* and provided precise information about it. Additional examples of answers can be found in Table 6.

We also conducted a Friedman test to examine whether differences existed between the answer conditions. Since Batch 1 and Batch 2 were evaluated by separate sets of users, we performed the analysis independently for each batch. We observed statistically significant differences between answer conditions in both batches (Batch 1: $\chi^2 = 48.5$, $p < .001$ and $\chi^2 = 43.4$, $p < .001$, Batch 2: $\chi^2 = 32.5$, $p < .001$ and $\chi^2 = 62.8$, $p < .001$ for helpfulness and correctness, respectively). Subsequently, we conducted post-hoc analysis with a Nemenyi test to identify which groups accounted for these differences. For both batches, the answers generated by Full Pipeline were significantly more correct and helpful compared to the two other methods ($p = .009$ for Question + Video vs. Full Pipeline in Batch 1, $p = .001$ for the rest). Additionally, the correctness of Question + Video surpassed that of Question-only in Batch 1 ($p = .013$). Figure 6 displays the distribution of Likert scale responses for each condition across both batches.

6.3.2 When Did the Full Pipeline Fail? There were instances where Full Pipeline fell short in generating useful answers compared to the other conditions. It sometimes assembled unrelated information, making the answers unnecessarily complicated. For instance,

when asked about the nesting sketches feature in Fusion 360, P8 noted that the Full Pipeline's answer mixed the concepts of nesting sketches with nesting for manufacturing. Similarly, P13 pointed out that it was overly verbose and used complex language, making it difficult to understand. These observations suggest that retrieving the right amount of information is crucial for providing useful answers. Additionally, there were a few instances in which it generated incorrect information by not recognizing UI elements in visual anchors that were not present in the UI database. In that case, the pipeline had to rely on other available information, such as transcripts, which led to incorrect answers.

6.3.3 General Feedback on AI-generated Questions. Overall, participants felt that the answers offered fairly accurate information (P1, P2, P7, P14, P15). P2 remarked that these AI-generated answers have the potential to provide quick and clear responses, thereby eliminating the need to sift through lengthy videos, forum threads, or posts. Despite these strengths, areas for improvement were identified. P16 suggested enhancing the terminology and style used in the answers, following the guidelines used in the software. P15 recommended that answers be more direct and to the point. For instance, mentioning "Fusion 360" in an answer is unnecessary as it is already evident. We discuss design guidelines for question-answering systems in Section 7.1.

7 DISCUSSION AND FUTURE WORK

In this section, we discuss design considerations for question-answering systems, possible question-answer pipeline improvements, potential video question-answering interface designs built on top of our pipeline, and generalizability to other software and domains.

7.1 Design Considerations for Question-Answering Systems

Providing answers to users' questions as they learn and follow software tutorial videos is crucial. For instance, it can offer personalized explanations that go beyond one-size-fits-all tutorials, by addressing points of confusion unique to different learners. Additionally,

it can guide users on resolving current challenges they encounter while following tutorials, a common issue faced by learners [70]. We identified a number of factors that are important to consider in the design of automated question-answering systems from our evaluation study, which we discuss below.

7.1.1 The Right Tone. Maintaining an appropriate tone in answers is key. For instance, there was a case where a user noted a discrepancy between the provided instruction and the outcome after following it. They questioned what might have gone wrong, and P16 felt that the responses generated subtly laid blame on the user (e.g., “It’s possible that a step was missed or misunderstood”). However, P16 appreciated one answer that acknowledged the complexity of Fusion 360, implicitly allowing for the possibility that the instructions might not have been easy to follow. This led P16 to select that response as the preferred one, underscoring the importance of tone in responses. As a system assisting the user, it’s essential to avoid attributing blame and to provide help in a constructive manner.

7.1.2 Answer Length. LLMs like ChatGPT often generate lengthy answers, which might be informative but not always ideal for users seeking brief, straightforward information [23]. In our first formative study, we observed that human-generated answers typically consist of 1 to 2 sentences. Accordingly, we instructed GPT-4 to limit responses to 50 words or fewer in all three conditions. However, our evaluation study still revealed a divide within the word limit: some participants preferred detailed answers for practical instruction, while others sought concise responses, believing that excess details could obscure the main point. Thus, an ideal approach would need to consider the user’s preferences and prior knowledge when formulating responses. A strategy that might work for a diverse audience is offering expandable answers: beginning with a concise response and allowing users the option to see more details, or alternatively, asking the system to elaborate on its answer (see also Section 7.3.1).

7.1.3 Transparency about Uncertainty. LLMs are susceptible to generating false or misleading information, known as hallucinations [77]. In our evaluation study, a number of participants pointed out that an answer mentioned unrelated tools (P14) or unavailable operations (P2). While experienced users might identify such inaccuracies, new users could easily be misled. Therefore, it is crucial for LLMs to be transparent about their limitations and uncertainties. Recent work on explainable LLMs, such as the ability to cite specific evidence for claims [46], could be beneficial in this context. Additionally, the system could be designed to be more interactive. For example, if the system fails to comprehend the question or identify a visual anchor referenced in it, it should prompt the user for additional details, enabling more context-rich and accurate answers.

7.2 Improvements to the Question-Answer Pipeline

AQuA recognizes visual anchors, retrieves relevant software-specific materials, and includes tutorial video context to generate answers to a given question. This multimodal approach allows for a more comprehensive understanding of the user’s query and thereby generates accurate and helpful answers (Section 6). However, achieving this involves several components working together in the pipeline.

Since these modules run sequentially, there is a potential for latency issues. In our case, answer generation takes around 30 seconds once the offline models, such as BLIP-2 and GPT-4, are loaded. We believe future work could explore ways to further reduce latency, enabling nearly real-time question-answering.

Apart from the latency, there are a number of areas in which our pipeline could be further enhanced. First, the pipeline could be extended to identify the precise point of interest within a given visual anchor. Participants sometimes captured a larger area in their visual anchors than the specific subject of their question, and sometimes, the precise visual anchor could not be derived from the question text. Although visual anchors already provide a narrower scope compared to a full video frame, a further scope reduction could yield more accurate results. This specific point of interest could be inferred from the mouse pointer’s location while creating the visual anchor. Alternatively, the Visual Recognition Module could be integrated into an intuitive interface to allow users to select a specific visual anchor from a set of automatically recognized elements of interest in the video frame. Second, the pipeline’s robustness could be increased by incorporating additional resources. Relevant content from other online tutorials, Q&A forums, and previous user comments could be integrated to provide a more comprehensive answer. Third, to offer richer context for the tutorial video in question, the system could go beyond using the title and transcript for context. By employing a video captioning model specifically trained on screencast videos, such as the one by Li et al. [36], we could obtain a more nuanced understanding of the video’s content at the time the question was posed. Lastly, if we take into account learners’ progress on their software as in SoftVideo [73], we can offer more specific and detailed answers tailored to users’ individual levels of knowledge or current progress. Understanding users’ preferences and proficiency levels with the software can be especially useful when answering questions in the ‘User’ category.

7.3 Potential Interface Designs for Tutorial Video Systems

Our work opens up exciting opportunities for integrating the question-answering pipeline into a tutorial video system. Here, we discuss a number of interface ideas that we find particularly promising.

7.3.1 Conversational Question-Answering. Beyond single-turn question-answering, multi-turn conversations can facilitate deeper understanding and assistance. We envision that a promising interface design for our AQuA pipeline is a conversational, chat-based question-answering system similar to ChatGPT [48], as illustrated in Figure 8. This would allow users to ask follow-up questions or provide feedback on the answers they receive, which the system can incorporate into future answers. Moreover, the system could even proactively initiate conversations to monitor users’ progress and assess their comprehension, as in Shin et al. [60], thus delivering more personalized responses. The system could also include motivational phrases such as, “You’re asking great questions!” or “That’s correct, you’re a fast learner!” to further inspire and engage learners [1]. We believe these enhancements offer valuable opportunities for effective learning experiences with tutorial videos.

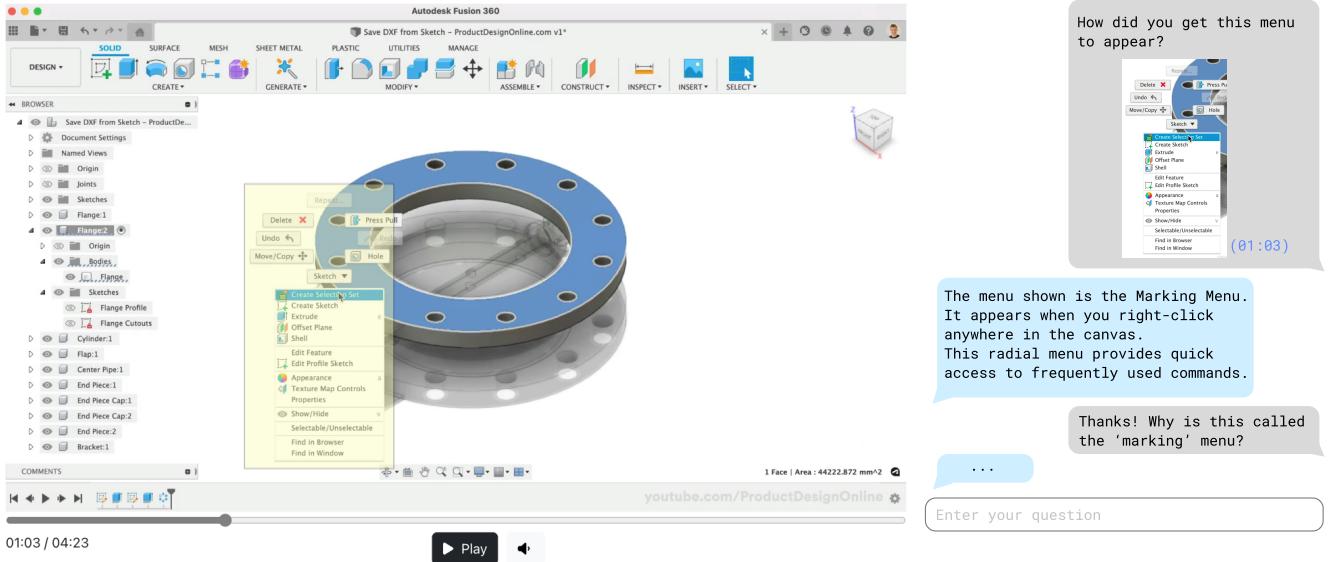


Figure 8: We envision that our pipeline could be leveraged in the future to develop a tutorial video system that supports conversational, chat-like question and answering. Learners could ask questions by referring to specific parts of the video. The system would then generate responses to these questions, while also allowing users to easily ask follow-up questions.

7.3.2 Support for Transcript Anchors. Our pipeline provides useful answers to queries that include visual anchors. An interesting interface extension could be to allow users to not only reference visual elements of interest, but also refer to parts of the audio transcript, which is another common type of reference in video [74]. Users could select or drag over parts of the transcript and ask questions about it (an example for Fusion 360 could be: “*What do you mean by ‘reference a construction plane’, and how is that done?*”). Allowing users to refer to both elements in the video and in the transcript can help them better articulate their questions.

7.3.3 Making Video Comments More Useful. Our approach to supporting questions and comments with visual anchors also opens up potential improvements to the interface design of (tutorial) video interfaces. Traditional video interfaces often separate video content from user comments, making it challenging to locate relevant discussions. With our approach, comments and automated answers can be organized based on the visual objects or components appearing in the video. For example, a user could visually select a tool of interest that is featured in the video to see related questions and comments about that tool. This could also be advantageous for tutorial authors by offering a quick overview of areas that generate the most questions, as demonstrated in Mudslide [18]. By efficiently reviewing questions from learners, authors can identify areas of confusion or topics that require more elaboration. These insights can serve as valuable feedback for authors when creating the next tutorial video. Furthermore, an interesting direction could be to simulate learners’ behavior, as explored in Generative Agents [53], and generate simulated questions even before publishing the video. This would enable authors to enhance their tutorial content by addressing potential points of clarification in advance.

7.4 Generalizability to Other Software

As discussed in Section 5, we believe our question-answer pipeline *AQuA* can easily generalize to other feature-rich software applications, such as Photoshop or AutoCAD. To adapt our approach to different software, only two components require replacement: (1) the *UI database*, encompassing software icons and names, and (2) *software articles*, such as documentation or tutorials. These resources are designed to recognize software UI elements in the visual anchor and provide software-specific information. In our demonstration with Fusion 360, we constructed these databases by crawling publicly available sources (details in Section 5). This implies the possibility of constructing similar databases for other software applications using their official documentation and publicly available tutorial resources. The remaining components would work the same, and by leveraging an off-the-shelf pre-trained LLM, we minimize the need for additional computational resources when adapting to other software applications. This approach makes our pipeline extensible and facilitates adapting the pipeline to various other software applications in the future.

7.5 Generalizability to Other Domains

It would be interesting to explore expanding the scope of our approach to other learning domains, such as instructional videos that teach physical skills, programming tutorials, or lecture videos. These videos, much like software tutorials, often convey information through both visual and verbal channels [10, 35, 61], which suggests the potential for questions with visual anchors. Given that LLMs and image captioning models are well-equipped with knowledge associated with everyday tasks and objects such as cooking and assembling furniture, it is conceivable that *AQuA*’s capabilities

could also extend to these domains, by using different resources (e.g., recipes and cookbooks instead of software documentation). For instance, users could anchor a question to a specific ingredient in a cooking video to inquire about its function and possible substitutes. The image captioning models are able to recognize ingredients, and by leveraging the rich knowledge encompassed by LLMs and enhanced by recipe-specific resources, the pipeline would likely be able to provide a comprehensive answer. On the other hand, for programming or lecture videos, we could rely more on OCR results as these videos often contain text-heavy content. Together with knowledge already embedded in LLMs and leveraging more specific materials such as textbooks, our pipeline could likely offer accurate answers. We believe that with some adjustments, our question-answering system with support for visual anchors could enable more contextual and comprehensive help systems across various domains.

8 CONCLUSION

We introduced an automated approach for answering questions in software tutorial videos. To achieve this, we conducted two formative studies to understand users' question-asking behavior. Focusing on questions related to the tutorial content, we discovered that users frequently refer to visual elements of the video, particularly focusing on UI components and the application workspace. Based on these insights, we developed *AQuA*, an LLM-based multimodal pipeline that generates useful answers to questions that include visual anchors, which are specific visual elements of interest in the tutorial video. Using software-specific resources such as software documentation and icons of tools, our pipeline identifies these visual anchors and generates answers tailored to the particular software. Our evaluation demonstrated that our approach yields more accurate and more helpful responses compared to baseline methods. Lastly, we discuss design considerations for question-answering systems and promising directions for future work, offering insights into the future of interactive and responsive learning experiences.

ACKNOWLEDGMENTS

We thank Amir H. Khasahmadi, Michael Chen, and Fraser Anderson from Autodesk Research for their help and feedback.

REFERENCES

- [1] Benjamin Alcott. 2017. Does Teacher Encouragement Influence Students' Educational Progress? A Propensity-Score Matching Analysis. *Research in Higher Education* 58, 7 (Jan 2017), 773–804. <https://doi.org/10.1007/s11162-017-9446-2>
- [2] Amazon. 2023. Amazon Transcribe. <https://aws.amazon.com/transcribe/>. Accessed: 2023-09-09.
- [3] Autodesk. 2023. Autodesk Screencast. <https://www.autodesk.com/support/technical/article/caas/tsarticles/ts/7IQzAbqSskV6l5Hm0ULMMp.html>. Accessed: 2023-09-10.
- [4] Autodesk. 2023. Fusion 360 Product Documentation. <https://help.autodesk.com/view/fusion360/ENU>. Accessed: 2023-09-10.
- [5] Nikola Banovic, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2012. Waken: Reverse Engineering Usage Information and Interface Structure from Software Videos. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) (*UIST '12*). Association for Computing Machinery, New York, NY, USA, 83–92. <https://doi.org/10.1145/2380116.2380129>
- [6] Yining Cao, Hariharan Subramonyam, and Eytan Adar. 2022. VideoSticker: A Tool for Active Viewing and Visual Note-Taking from Videos. In *27th International Conference on Intelligent User Interfaces* (Helsinki, Finland) (*IUI '22*). Association for Computing Machinery, New York, NY, USA, 672–690. <https://doi.org/10.1145/3490099.3511132>
- [7] Minsuk Chang, Mina Huh, and Juho Kim. 2021. RubySlippers: Supporting Content-Based Voice Navigation for How-to Videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (*CHI '21*). Association for Computing Machinery, New York, NY, USA, Article 97, 14 pages. <https://doi.org/10.1145/3411764.3445131>
- [8] Minsuk Chang, Anh Truong, Oliver Wang, Maneesh Agrawala, and Juho Kim. 2019. How to Design Voice Based Navigation for How-To Videos. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) (*CHI '19*). Association for Computing Machinery, New York, NY, USA, 1–11. <https://doi.org/10.1145/3290605.3300931>
- [9] Jieshan Chen, Amanda Swearngin, Jason Wu, Titus Barik, Jeffrey Nichols, and Xiaoyi Zhang. 2022. Towards Complete Icon Labeling in Mobile Applications. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (*CHI '22*). Association for Computing Machinery, New York, NY, USA, Article 387, 14 pages. <https://doi.org/10.1145/3491102.3502073>
- [10] Pei-Yu Chi, Joyce Liu, Jason Linder, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2013. DemoCut: Generating Concise Instructional Videos for Physical Demonstrations. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) (*UIST '13*). Association for Computing Machinery, New York, NY, USA, 141–150. <https://doi.org/10.1145/2501988.2502052>
- [11] Soon Hau Chua, Toni-Jan Keith Palma Monserrat, Dongwook Yoon, Juho Kim, and Shengdong Zhao. 2017. Korero: Facilitating Complex Referencing of Visual Materials in Asynchronous Discussion Interface. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article 34 (Dec 2017), 19 pages. <https://doi.org/10.1145/3134669>
- [12] John Joon Young Chung, Hijung Valentina Shin, Hajun Xia, Li-ye Wei, and Rubaiat Habib Kazi. 2021. Beyond Show of Hands: Engaging Viewers via Expressive and Scalable Visual Communication in Live Streaming. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (*CHI '21*). Association for Computing Machinery, New York, NY, USA, Article 109, 14 pages. <https://doi.org/10.1145/3411764.3445419>
- [13] Anthony Colas, Seokhwan Kim, Franck Dernoncourt, Siddhesh Gupte, Zhe Wang, and Doo Soon Kim. 2020. TutorialVQA: Question Answering Dataset for Tutorial Videos. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 5450–5455. <https://aclanthology.org/2020.lrec-1.670>
- [14] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A Mobile App Dataset for Building Data-Driven Design Applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) (*UIST '17*). Association for Computing Machinery, New York, NY, USA, 845–854. <https://doi.org/10.1145/3126594.3126651>
- [15] C. Ailie Fraser, Joy O. Kim, Hijung Valentina Shin, Joel Brandt, and Mira Dontcheva. 2020. Temporal Segmentation of Creative Live Streams. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3313831.3376437>
- [16] C. Ailie Fraser, Julia M. Markel, N. James Basa, Mira Dontcheva, and Scott Klemmer. 2019. ReMap: Multimodal Help-Seeking. In *Adjunct Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) (*UIST '19 Adjunct*). Association for Computing Machinery, New York, NY, USA, 96–98. <https://doi.org/10.1145/3332167.3356884>
- [17] C. Ailie Fraser, Tricia J. Ngoon, Mira Dontcheva, and Scott Klemmer. 2019. RePlay: Contextually Presenting Learning Videos Across Software Applications (*CHI '19*). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3290605.3300527>
- [18] Elena L. Glassman, Juho Kim, Andrés Monroy-Hernández, and Meredith Ringel Morris. 2015. Mudslide: A Spatially Anchored Census of Student Confusion for Online Lecture Videos. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) (*CHI '15*). Association for Computing Machinery, New York, NY, USA, 1555–1564. <https://doi.org/10.1145/2702123.2702304>
- [19] Google. 2023. Cloud Vision API. <https://cloud.google.com/vision/docs>. Accessed: 2023-09-10.
- [20] Google. 2023. YouTube Data API. <https://developers.google.com/youtube/v3>. Accessed: 2023-09-10.
- [21] Joshua Grossman, Zhiyuan Jerry Lin, Hao Sheng, Johnny Tian-Zheng Wei, Joseph Jay Williams, and Sharad Goel. 2019. MathBot: Transforming Online Resources for Learning Math into Conversational Interactions. <https://api.semanticscholar.org/CorpusID:236143850>
- [22] Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2010. Chronicle: Capture, Exploration, and Playback of Document Workflow Histories. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) (*UIST '10*). Association for Computing Machinery, New York, NY, USA, 143–152. <https://doi.org/10.1145/1866029.1866054>
- [23] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. *arXiv preprint arxiv:2301.07597*

- (2023).
- [24] Nikhita Joshi, Justin Matejka, Fraser Anderson, Tovi Grossman, and George Fitzmaurice. 2020. MicroMentor: Peer-to-Peer Software Help Sessions in Three Minutes or Less. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '20*). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376230>
- [25] Kimia Kiani, George Cui, Andrea Bunt, Joanna McGrenere, and Parmit K. Chilana. 2019. Beyond "One-Size-Fits-All": Understanding the Diversity in How Software Newcomers Discover and Make Use of Help Resources. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) (*CHI '19*). Association for Computing Machinery, New York, NY, USA, 1–14. <https://doi.org/10.1145/3290605.3300570>
- [26] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller, and Krzysztof Z. Gajos. 2014. Crowdsourcing Step-by-Step Information Extraction to Enhance Existing How-to Videos. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) (*CHI '14*). Association for Computing Machinery, New York, NY, USA, 4017–4026. <https://doi.org/10.1145/2556288.2556986>
- [27] Jeongyeon Kim, Junyong Park, and I-Hao Lu. 2021. HyperButton: In-Video Question Answering via Interactive Buttons and Hyperlinks. In *Asian CHI Symposium 2021* (Yokohama, Japan) (*Asian CHI Symposium 2021*). Association for Computing Machinery, New York, NY, USA, 48–52. <https://doi.org/10.1145/3429360.3468179>
- [28] Jiho Kim, Arjun Srinivasan, Nam Wook Kim, and Yea-Seul Kim. 2023. Exploring Chart Question Answering for Blind and Low Vision Users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 828, 15 pages. <https://doi.org/10.1145/3544548.3581532>
- [29] Tae Soo Kim, Seungsoo Kim, Yoonseon Choi, and Juho Kim. 2021. Winder: Linking Speech and Visual Objects to Support Communication in Asynchronous Collaboration. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (*CHI '21*). Association for Computing Machinery, New York, NY, USA, Article 453, 17 pages. <https://doi.org/10.1145/3411764.3445686>
- [30] Ben Lafreniere, Andrea Bunt, Matthew Lount, and Michael Terry. 2021. Understanding the Roles and Uses of Web Tutorials. *Proceedings of the International AAAI Conference on Web and Social Media* 7, 1 (Aug. 2021), 303–310. <https://doi.org/10.1609/icwsm.v7i1.14413>
- [31] Yoonjoo Lee, Tae Soo Kim, Sungdong Kim, Yohan Yun, and Juho Kim. 2023. DAPIE: Interactive Step-by-Step Explanatory Dialogues to Answer Children's Why and How Questions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 450, 22 pages. <https://doi.org/10.1145/3544548.3581369>
- [32] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) (*NIPS '20*). Curran Associates Inc., Red Hook, NY, USA, Article 793, 16 pages.
- [33] Gang Li and Yang Li. 2023. Spotlight: Mobile UI Understanding using Vision-Language Models with a Focus. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR'23)*.
- [34] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) (*ICML '23*). Article 814, 13 pages.
- [35] Jiannan Li, Mauricio Sousa, Karthik Mahadevan, Bryan Wang, Paula Akemi Aoyagui, Nicole Yu, Angela Yang, Ravin Balakrishnan, Anthony Tang, and Tovi Grossman. 2023. Stargazer: An Interactive Camera Robot for Capturing How-To Videos Based on Subtle Instructor Cues. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 800, 16 pages. <https://doi.org/10.1145/3544548.3580896>
- [36] Kunpeng Li, Chen Fang, Zhao Wen Wang, Seokhwan Kim, Hailin Jin, and Yun Fu. 2020. Screencast Tutorial Video Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [37] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. 2020. Hero: Hierarchical Encoder for Video+Language Omni-representation Pre-training. In *Conference on Empirical Methods in Natural Language Processing*. <https://api.semanticscholar.org/CorpusID:218470055>
- [38] Toby Jia-Jun Li, Lindsay Popowski, Tom Mitchell, and Brad A Myers. 2021. Screen2Vec: Semantic Embedding of GUI Screens and GUI Components. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (*CHI '21*). Association for Computing Machinery, New York, NY, USA, Article 578, 15 pages. <https://doi.org/10.1145/3411764.3445049>
- [39] Georgianna Lin, Jin Yi Li, Afsaneh Fazly, Vladimir Pavlovic, and Khai Truong. 2023. Identifying Multimodal Context Awareness Requirements for Supporting User Interaction with Procedural Videos. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 761, 17 pages. <https://doi.org/10.1145/3544548.3581006>
- [40] Joshua Lochner. 2023. Chat Downloader. <https://chat-downloader.readthedocs.io/en/latest/>. Accessed: 2023-09-10.
- [41] Zhicong Lu, Rubaiat Habib Kazi, Li-yi Wei, Mira Dontcheva, and Karrie Karahalios. 2021. StreamSketch: Exploring Multi-Modal Interactions in Creative Live Streams. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 58 (Apr 2021), 26 pages. <https://doi.org/10.1145/3449132>
- [42] Amy Madden, Ian Ruthven, and David McMenamy. 2013. A classification scheme for content analyses of YouTube video comments. *J. Documentation* 69 (2013), 693–714. <https://api.semanticscholar.org/CorpusID:206402932>
- [43] Damien Masson, Jo Vermeulen, George Fitzmaurice, and Justin Matejka. 2022. Supercharging Trial-and-Error for Learning Complex Software Applications. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (*CHI '22*). Association for Computing Machinery, New York, NY, USA, Article 381, 13 pages. <https://doi.org/10.1145/3491102.3501895>
- [44] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2011. Ambient Help. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) (*CHI '11*). Association for Computing Machinery, New York, NY, USA, 2751–2760. <https://doi.org/10.1145/1978942.1979349>
- [45] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2011. IP-QAT: In-Product Questions, Answers, & Tips. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) (*UIST '11*). Association for Computing Machinery, New York, NY, USA, 175–184. <https://doi.org/10.1145/2047196.2047218>
- [46] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaise, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. 2022. Teaching language models to support answers with verified quotes. *arXiv:2203.11147 [cs.CL]*
- [47] Cuong Nguyen and Feng Liu. 2015. Making Software Tutorial Video Responsive (*CHI '15*). Association for Computing Machinery, New York, NY, USA, 1565–1568. <https://doi.org/10.1145/2702123.2702209>
- [48] OpenAI. 2023. ChatGPT. <https://openai.com/blog/chatgpt>. Accessed: 2023-09-09.
- [49] OpenAI. 2023. GPT-4. <https://openai.com/gpt-4>. Accessed: 2023-09-09.
- [50] OpenAI. 2023. text-embedding-ada-002. <https://openai.com/blog/new-and-improved-embedding-model>. Accessed: 2023-09-09.
- [51] OpenCV. 2023. Feature Matching. https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html. Accessed: 2023-09-09.
- [52] OpenCV. 2023. Template Matching. https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html. Accessed: 2023-09-09.
- [53] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (*UIST '23*). Association for Computing Machinery, New York, NY, USA, Article 2, 22 pages. <https://doi.org/10.1145/3586183.3606763>
- [54] Elizabeth Poché, Nishant Jha, Grant Williams, Jazmine Staten, Miles Vesper, and Anas Mahmoud. 2017. Analyzing User Comments on YouTube Coding Tutorial Videos. In *2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC)*, 196–206. <https://doi.org/10.1109/ICPC.2017.26>
- [55] Rhitabrat Pokharel and Dixit Bhatta. 2021. Classifying YouTube Comments Based on Sentiment and Type of Sentence. *arXiv:2111.01908 [cs.IR]*
- [56] Suporn Pongnumkul, Mira Dontcheva, Wilmet Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. 2011. Pause-and-Play: Automatically Linking Screencast Video Tutorials with Applications (*UIST '11*). Association for Computing Machinery, New York, NY, USA, 135–144. <https://doi.org/10.1145/2047196.2047213>
- [57] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) (*ICML '23*). Article 1182, 27 pages.
- [58] Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2022. A Survey of Evaluation Metrics Used for NLP Systems. *ACM Comput. Surv.* 55, 2, Article 26 (Jan 2022), 39 pages. <https://doi.org/10.1145/3485766>
- [59] Peter Schultes, Verena Dorner, and Franz Lehner. 2013. Leave a Comment! An In-Depth Analysis of User Comments on YouTube. In *Wirtschaftsinformatik*. <https://api.semanticscholar.org/CorpusID:11966671>
- [60] Hyungyu Shin, Eun-Young Ko, Joseph Jay Williams, and Juho Kim. 2018. Understanding the Effect of In-Video Prompting on Learners and Instructors. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (*CHI '18*). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3173574.3173893>
- [61] Hijung Valentina Shin, Floraine Berthouzoz, Wilmet Li, and Frédéric Durand. 2015. Visual Transcripts: Lecture Notes from Blackboard-Style Lecture Videos. *ACM Trans. Graph.* 34, 6, Article 240 (Nov 2015), 10 pages. <https://doi.org/10.1145/2816795.2818123>
- [62] Sicheng Song, Junlong Chen, Chenhui Li, and Changbo Wang. 2023. GVQA: Learning to Answer Questions about Graphs with Visualizations via Knowledge Base. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing*

- Systems* (Hamburg, Germany) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 464, 16 pages. <https://doi.org/10.1145/3544548.3581067>
- [63] Anh Truong, Peggy Chi, David Salesin, Irfan Essa, and Maneesh Agrawala. 2021. Automatic Generation of Two-Level Hierarchical Tutorials from Instructional Makeup Videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 108, 16 pages. <https://doi.org/10.1145/3411764.3445721>
- [64] Bryan Wang, Gang Li, and Yang Li. 2023. Enabling Conversational Interaction with Mobile UI Using Large Language Models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 432, 17 pages. <https://doi.org/10.1145/3544548.3580895>
- [65] Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. 2021. Screen2Words: Automatic Mobile UI Summarization with Multimodal Learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) (*UIST '21*). Association for Computing Machinery, New York, NY, USA, 498–510. <https://doi.org/10.1145/3472749.3474765>
- [66] Rainer Winkler, Sebastian Hobert, Antti Salovaara, Matthias Söllner, and Jan Marco Leimeister. 2020. Sara, the Lecturer: Improving Learning in Online Education with a Scaffolding-Based Conversational Agent. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '20*). Association for Computing Machinery, New York, NY, USA, 1–14. <https://doi.org/10.1145/3313831.3376781>
- [67] Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P Bigham. 2023. WebUI: A Dataset for Enhancing Visual UI Understanding with Web Semantics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 286, 14 pages. <https://doi.org/10.1145/3544548.3581158>
- [68] Mulong Xie, Sidong Feng, Zhenchang Xing, Jieshan Chen, and Chunyang Chen. 2020. UIED: A Hybrid Tool for GUI Element Detection. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Virtual Event, USA) (*ESEC/FSE 2020*). Association for Computing Machinery, New York, NY, USA, 1655–1659. <https://doi.org/10.1145/3368089.3417940>
- [69] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2020. Just Ask: Learning to Answer Questions from Millions of Narrated Videos. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2020), 1666–1677. <https://api.semanticscholar.org/CorpusID:22723896>
- [70] Saelyne Yang and Juho Kim. 2020. What Makes It Hard for Users to Follow Software Tutorial Videos?. In *Proceedings of HCI Korea 2020*. The HCI Society of KOREA, South Korea, 531–536.
- [71] Saelyne Yang, Sangkyung Kwak, Juhoon Lee, and Juho Kim. 2023. Beyond Instructions: A Taxonomy of Information Types in How-to Videos. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 797, 21 pages. <https://doi.org/10.1145/3544548.3581126>
- [72] Saelyne Yang, Changyoon Lee, Hijung Valentina Shin, and Juho Kim. 2020. Snapshot-stream: Snapshot-Based Interaction in Live Streaming for Visual Art. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '20*). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3313831.3376390>
- [73] Saelyne Yang, Jisu Yim, Aitolkyun Baigutanova, Seoyoung Kim, Minsuk Chang, and Juho Kim. 2022. SoftVideo: Improving the Learning Experience of Software Tutorial Videos with Collective Interaction Data. In *27th International Conference on Intelligent User Interfaces* (Helsinki, Finland) (*IUI '22*). Association for Computing Machinery, New York, NY, USA, 646–660. <https://doi.org/10.1145/3490099.3511106>
- [74] Matin Yarmand, Dongwook Yoon, Samuel Dodson, Ido Roll, and Sidney S. Fels. 2019. "Can You Believe [1:21]!": Content and Time-Based Reference Patterns in Video Comments. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) (*CHI '19*). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300719>
- [75] Tom Yeh, Tsung-Hsiang Chang, and Robert C. Miller. 2009. Sikuli: Using GUI Screenshots for Search and Automation. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (Victoria, BC, Canada) (*UIST '09*). Association for Computing Machinery, New York, NY, USA, 183–192. <https://doi.org/10.1145/1622176.1622213>
- [76] Xiaoyi Zhang, Lilian de Greef, Amanda Swearngin, Samuel White, Kyle Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, Aaron Everitt, and Jeffrey P Bigham. 2021. Screen Recognition: Creating Accessibility Metadata for Mobile Applications from Pixels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (*CHI '21*). Association for Computing Machinery, New York, NY, USA, Article 275, 15 pages. <https://doi.org/10.1145/3411764.3445186>
- [77] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *arXiv preprint arXiv:2309.01219* (2023).
- [78] Wentian Zhao, Seokhwan Kim, Ning Xu, and Hailin Jin. 2021. Video Question Answering on Screencast Tutorials. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (Yokohama, Yokohama, Japan) (*IJCAI'20*). Article 148, 8 pages.
- [79] Yaxi Zhao, Razan Jaber, Donald McMillan, and Cosmin Munteanu. 2022. "Rewind to the Jiggling Meat Part": Understanding Voice Control of Instructional Videos in Everyday Tasks. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (*CHI '22*). Association for Computing Machinery, New York, NY, USA, Article 58, 11 pages. <https://doi.org/10.1145/3491102.3502036>

A PROMPTS USED IN QUESTION-ANSWER PIPELINE

A.1 (1) Question-Only

You need to answer questions about Autodesk Fusion 360 that people asked while watching a tutorial video. Please answer in 50 words or less.

Question: {question_text}

A.2 (2) Question and Video Context

You need to answer questions about Autodesk Fusion 360 that people asked while watching a tutorial video. Please answer in 50 words or less.

Tutorial: Title: {title}. Instructions: {transcript}

Question: {question_text}

A.3 (3) Our Full Pipeline

You need to answer questions about Autodesk Fusion 360 that people asked while watching a tutorial video. Please answer in 50 words or less. Each question is accompanied by relevant visual anchors, which are specific visual elements of interest in the video.

Use the below articles on the Fusion 360 software to answer the subsequent question. If the answer cannot be found in the articles, write ``I could not find an answer.''

Fusion 360 article section: {section 1}

Fusion 360 article section: {section 2}

...

Tutorial: Title: {title}. Instructions: {transcript}

Question: {question_text}

Visual Anchor:

{Anchor_label_1}: {blip}. It includes the Fusion 360 tools: {tool} and text: {ocr}.

{Anchor_label_2}: {blip}. It includes the Fusion 360 tools: {tool} and text: {ocr}.

...
