

TutoAI: A Cross-domain Framework for AI-assisted Mixed-media Tutorial Creation on Physical Tasks

Yuexi Chen*
ychen151@umd.edu
University of Maryland
College Park, Maryland, USA

Anh Truong
truong@adobe.com
Adobe Research
San Francisco, California, USA

Vlad I. Morariu
morariu@adobe.com
Adobe Research
College Park, Maryland, USA

Zhicheng Liu
leozcliu@umd.edu
University of Maryland
College Park, Maryland, USA

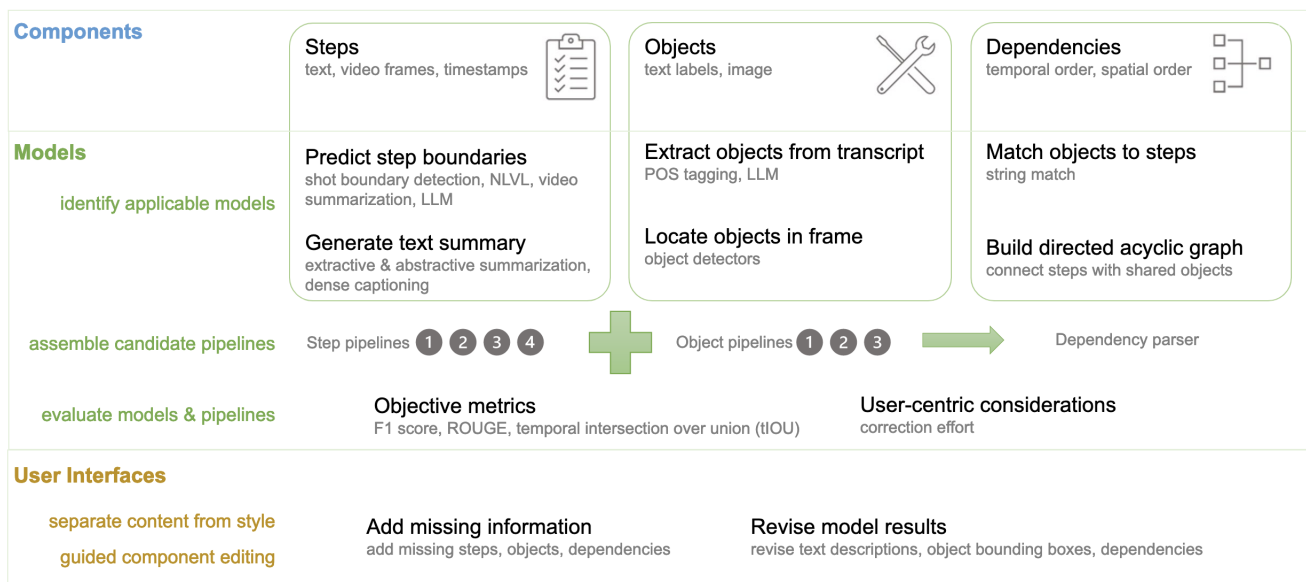


Figure 1: TutoAI is a framework for AI-assisted mixed-media tutorial creation. It has three levels: components, models, and user interfaces. After identifying components of common mixed-media tutorials, TutoAI assembles and evaluates relevant computational models to extract components. Then, it presents the results on a user interface for creators to review and edit.

ABSTRACT

Mixed-media tutorials, which integrate videos, images, text, and diagrams to teach procedural skills, offer more browsable alternatives than timeline-based videos. However, manually creating such tutorials is tedious, and existing automated solutions are often restricted to a particular domain. While AI models hold promise,

it is unclear how to effectively harness their powers, given the multi-modal data involved and the vast landscape of models. We present TutoAI, a cross-domain framework for AI-assisted mixed-media tutorial creation on physical tasks. First, we distill common tutorial components by surveying existing work; then, we present an approach to identify, assemble, and evaluate AI models for component extraction; finally, we propose guidelines for designing user interfaces (UI) that support tutorial creation based on AI-generated components. We show that TutoAI has achieved higher or similar quality compared to a baseline model in preliminary user studies.

*Part of the work done during an internship at Adobe



This work is licensed under a Creative Commons Attribution International 4.0 License.

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/10.1145/3613904.3642443>

CCS CONCEPTS

• **Human-centered computing** → Human-computer interaction (HCI); Interaction design.

KEYWORDS

Human-AI interaction, mixed-media tutorials, AI-assisted creation

ACM Reference Format:

Yuexi Chen, Vlad I. Morariu, Anh Truong, and Zhicheng Liu. 2024. TutoAI: A Cross-domain Framework for AI-assisted Mixed-media Tutorial Creation on Physical Tasks. 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, 17 pages. <https://doi.org/10.1145/3613904.3642443>

1 INTRODUCTION

Instructional videos are important sources for people to acquire new skills. However, the linear timeline-based video format provides limited overviews, with no explicit representation of the steps and their dependencies. Besides, navigating the timeline is tedious and imprecise. While users can fast-forward or replay videos, scrubbing the timeline might cause them to overlook vital information [66, 81].

Recent work has shown that mixed-media tutorials, which unify videos, images, text, and diagrams in an interactive user interface, offer more browsable alternatives. For example, YouTube Chapters [54] help navigate long-form videos: each chapter corresponds to a video segment with a short text description, a thumbnail, and a timestamp. Researchers have also proposed non-linear mixed-media tutorials for tasks such as applying makeup and cooking [49, 65, 74]. Such tutorials optimize user navigation by providing object details and organizing steps based on dependencies.

Although the benefits of mixed-media tutorials are confirmed, creating such tutorials from the original instructional videos remains challenging. Current approaches for authoring mixed-media tutorials are usually domain-specific, with both the tutorial components and extraction techniques tailored for each domain [14, 20, 49, 65]. While many have acknowledged the importance of generalization and argued how their approaches could apply to tutorials in other domains [32, 65, 67, 70], a cross-domain framework with shared vocabulary and reusable methodologies for mixed-media tutorial creation is still lacking. We believe such a framework will benefit the future development of mixed-media tutorial creation, as demonstrated in other research areas [7, 11].

Recent advances in AI, especially large language models (LLM) [9], have shown promise in content understanding and generation, and can potentially play a vital role in establishing a cross-domain framework. However, integrating AI with mixed-media tutorial creation is not straightforward. First, we have neither a vocabulary to describe the common components of mixed-media tutorials nor a systematic account of the roles of humans and AI in extracting such components. Second, a single component may have multi-modal appearances (e.g., cooking ingredients appearing in both the audio narration and video frames), and multiple machine learning (ML) models are applicable. Currently, there are no guidelines on how to assemble and evaluate ML models to obtain mixed-media tutorial components from original videos. Though the landscape of ML models changes over time, we believe there are general guidelines that could transcend specific models.

To address these challenges, we present TutoAI, the first cross-domain framework to integrate AI in creating mixed-media tutorials (Figure 1). We focus on instructional videos on physical tasks (e.g., cooking, hardware assembly) instead of concepts (e.g., lectures) or

digital artifacts (e.g., software usage, programming). The TutoAI framework has three levels: components, models, and user interfaces (UI). At the *component* level, we conduct a comprehensive survey to identify common components of mixed-media tutorials and analyze their representations. At the *model* level, we review ML methods to extract each component and present an approach to assemble and evaluate applicable ML models. At the *UI* level, we propose guidelines for building UIs that allow creators to review and edit AI-generated components and also implement an example interactive prototype.

We evaluate TutoAI in two ways. At the model level, we validate the performance of the assembled ML pipeline on a large set of cooking videos and a small set of diverse instructional videos. At the UI level, we evaluate the user-perceived component quality by conducting two studies with 24 general instructional video viewers and 2 YouTube creators. Our results show that TutoAI-generated components have higher or similar quality compared to a baseline model (YouTube Chapters [54]), and the TutoAI framework has the potential to be integrated into creators' workflow. In summary, we make the following contributions:

- A comprehensive survey for mixed-media tutorials and a taxonomy of mixed-media tutorial components.
- TutoAI, a cross-domain framework for AI-assisted mixed-media tutorial creation on physical tasks, including components, models, and UIs.
- Empirical evaluation of TutoAI framework in terms of model quality, user-perceived quality, and workflow integration.

2 RELATED WORK

2.1 Mixed-media tutorials

Mixed-media tutorials, though diverse in format, share commonalities in tutorial components and extraction methods.

Tutorial components. A common component is a step, usually a video segment, comprising a text description, a thumbnail, and a timestamp [20, 32, 53, 54, 70]. A step could range from a cooking procedure [12] to a software operation [20]. Another common component is objects, e.g., ingredients and equipment for cooking tutorials [44, 49, 74]. Besides steps and objects, some tutorials also organize steps based on dependencies, e.g., Truong et al. [65] grouped makeup video segments by facial parts in a two-level hierarchical format; Nawhal et al. [49] and Yang et al. [74] arranged cooking steps non-linearly by temporal and spatial dependencies. TutoAI, our proposed framework, has a *Components* level built upon components distilled from existing mixed-media tutorials.

Extraction methods. Tutorial component extraction from original videos could be manual, automatic, or mixed-initiative (detailed comparison in Appendix Table 2-4). Websites like WikiHow [71] and Allrecipes [44] depend on experts to draft tutorials; Crowdy [70] requires learners to identify subgoals and steps. In certain domains, automatic extraction methods are feasible. MixT [14] segments PhotoShop videos using software logs. Fraser et al. [20] implement a dynamic programming method to segment creative stream videos based on the transcript and software logs; Truong et al. [65] apply video shot detection and transcript segmentation methods for makeup videos. However, the above methods require

domain-specific data and may not apply to other domains. Mixed-initiative methods involve both human effort and computational techniques. Humans could provide input, e.g., ToolScape [32] gathers steps from crowdworkers and converges them through clustering algorithms. EverTutor [68] converts smartphone demonstrations by humans into interactive tutorials. Humans could also refine computational results, e.g., VideoWhiz [49] and RecipeDeck [12] both employ Part-of-Speech (POS) tagging to detect cooking actions and objects and then rely on annotators to refine the results. Video Digests [53] applies Bayesian topic segmentation to generate chapters in lecture videos, allowing users to improve upon them. The second level of TutoAI focuses on models, including an approach to identifying, evaluating, and assembling AI models to extract tutorial components. TutoAI also adopts a mixed-initiative approach, where humans refine computational results.

Cross-domain applicability is a goal in previous work on mixed-media tutorials. For example, Truong et al. suggest their segmentation algorithm for makeup videos could be adapted for cooking, DIY, and bartending [65]; Soloist [67] transforms instructional guitar videos into mixed-media tutorials, and the processing pipeline can be generalized to other instruments; Kim et al. show that the same annotation pattern combined with a clustering algorithm can process cross-domain instructional videos [32]; Crowdy [70] is a subgoal-based crowdsourcing annotation workflow.

TutoAI extends this line of work, aiming to create a general cross-domain framework for mixed-media tutorials. Unlike crowdsourcing annotation workflows, TutoAI relies on AI.

2.2 AI-assisted creation

AI has augmented human creativity, from generating visuals [45, 57] to crafting slogans and aiding scientific writing [9, 22]. However, AI outputs may be imperfect or misaligned with user intentions, necessitating human refinement. Researchers have built AI-assisted creation tools in multiple domains, e.g., Cococo [41] allows users to adjust the mood of AI-generated music notes. Morai Maker [25] is a game-level editor in which human and AI designers take turns to build a Super Mario Bros game. LaMPost [23] facilitates email writing for people with Dyslexia. Dang et al.’s text editor [16] supports writers to refine automatically generated paragraph summaries. Some tools focus on refinement instead of creation: e.g., refinement of topics returned by topic models [61]; repair of auto-extracted PDF tables [27]; refinement of medical images retrieved by ML models [10]. TutoAI also adopts an AI-assisted approach, supporting the creation of mixed-media tutorials with extensive refinement. Unlike previous work focusing on a single modality, TutoAI supports multi-modal mixed-media tutorial creation empowered by various ML models.

Providing guardrails for AI output is crucial. Previous research has proposed several principles for designing such mixed-initiative user interfaces [4, 28], such as “provide mechanisms for efficient agent-user collaboration to refine results” and “support efficient correction”. TutoAI adheres to these principles, and additionally shares design considerations for choosing ML methods across modalities.

2.3 Large language models (LLM) prompting

Large language models (LLM) [8, 9, 64], trained on internet-wide data, have demonstrated extraordinary potential in information

processing tasks such as text summarization. Users interact with LLMs by providing natural language descriptions of the task, also called *prompting* [55]. The most commonly used prompting technique is zero-shot prompting [5], which describes the task directly. There are also other prompting techniques, including few-shot prompting [42] and prompt chaining [73]. Researchers have applied zero-shot prompting to summarize various types of data, including news [24, 79], Reddit posts [75], meeting records [34] and stories [75]. Researchers have also applied LLMs to summarize video transcripts. Croitoru et al. [15] applied GPT-3 to summarize software tutorial video transcripts and then used the summary to detect key moments. LUSE [60] also uses zero-shot prompting to summarize tutorial video transcripts and generalize steps for a task across different videos. To evaluate the summarization quality of LLMs, researchers have used traditional metrics like ROUGE scores [37], which measures the number of overlapped n-grams in the reference and summarized text, as well as employed humans to examine different aspects of the output, including coverage [60], descriptiveness [60], coherence [79], faithfulness [79], relevance [79] and personal preferences [24].

TutoAI also relies on zero-shot prompting to summarize video transcripts. In addition to requesting a summary, TutoAI also asks an LLM to extract objects and timestamp information. Like Croitoru et al. [15] and LUSE [60], TutoAI uses the generated summary as input for other models. The difference is that their contributions are models that focus on a single task (e.g., detect video moments) and exclude humans from the loop, but TutoAI contributes an AI-assisted framework. As LLMs suffer from *hallucination* (plausible yet incorrect output) [80], involving human refinement is crucial for end users. Similar to previous research, we manually evaluated the output besides ROUGE scores.

3 TUTOAI OVERVIEW: AN AI-ASSISTED FRAMEWORK

The TutoAI framework aims to provide a cross-domain approach to AI-assisted creation of mixed-media tutorials on physical tasks. We expect the input to include an instructional video and its transcript. Our design goals, informed by the review of current mixed-media tutorials and ML methods, are:

- D1 **Support cross-domain tutorial creation:** Mixed-media tutorials are useful in diverse domains, and TutoAI should offer a generalized approach.
- D2 **Handle multi-modal data types:** The input instructional videos and the output mixed-media tutorials both contain multi-modal data. TutoAI should support multi-modality.
- D3 **Empower creators without information overload:** Given the multi-modalities in mixed-media tutorials and the vast landscape of ML models, TutoAI should present information to creators without overwhelming them.

3.1 Level 1: Components

As shown in Figure 1, TutoAI is built on three types of cross-domain components in mixed-media tutorials (D1): steps, objects, and dependencies (detailed in section 4). These components are multi-modal (D2), specifically:

- *Steps:* represented as text, images, video clips, and temporal metadata (timestamps)

- *Objects*: represented as text, images, and temporal metadata (appearance time in videos)
- *Dependencies*: encoded as hierarchical structures, diagrams, and links

The output mixed-media tutorials may include all or a subset of these components. For instance, YouTube Chapters [54] only utilize *steps*. For completeness (D2), we discuss all three component types in level 2 and level 3.

3.2 Level 2: Models

After identifying the components and their representations, we focus on methodologies to select and evaluate applicable ML models to obtain such components from instructional videos. Even though cutting-edge ML models change over time, the general approaches we suggest here transcend particular models (Section 5).

3.2.1 Identifying relevant models. The first task is identifying models capable of extracting information required for a component. We consider models that take visual or transcript data from the video as inputs (D2), and with outputs that match the desired component representations. For instance, if a step component requires text descriptions, then models that ingest video transcripts or frames, and output text descriptions are applicable.

3.2.2 Assembling models. After identifying relevant models, we assemble models into candidate pipelines based on input and output modalities. For example, if a step component requires text descriptions and timestamps, instead of finding a single model that generates both, we can assemble two different pipelines serving the same goal. In the first pipeline, one model generates text descriptions, and the other locates the descriptions in the video. Alternatively, we can assemble another pipeline where one model segments videos first and the other model generates text descriptions for each segment.

3.2.3 Evaluating models. After considering alternative ways to assemble models, we first find common benchmark metrics for model evaluation. Besides objective metrics, we also assess correction efforts for creators. For example, false positives (FPs) are deemed easier to fix than false negatives (FNs), as fixing FPs requires deletion, but fixing FN requires creation.

3.3 Level 3: User Interface (UI) design

AI-generated results are typically imperfect, requiring further refinement from humans. As shown in Figure 1, the UI should support creators to review and revise AI-generated results. To manage cognitive load (D3), the UI should display AI-generated results sequentially, allowing creators to focus on one aspect at a time, and the refined results could be input for subsequent stages, mitigating error propagation. Section 6 discusses UI design guidelines and presents an example implementation.

4 LEVEL 1: COMPONENTS IN MIXED-MEDIA TUTORIALS

To explore the design space of cross-domain mixed-media tutorials, we analyzed 13 mixed-media tutorials from three websites [44, 71, 76] and 10 research papers [12, 14, 20, 32, 39, 49, 53, 65, 68, 74], covering at least five domains including cooking, makeup,

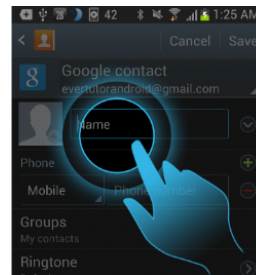
vehicle repair, software usage, and educational lectures. Though we focus on tutorials of physical tasks, we also borrow inspiration from other domains, e.g., lectures.

For each mixed-media tutorial, we annotated the informational units, such as ingredients in recipe tutorials, and visual representations. These units were then categorized into three types of components: step, object, and dependency. We also annotated extraction methods based on human roles (Appendix Table 2-4).

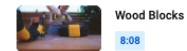
4.1 Step

Every tutorial comprises a sequence of steps, e.g., “duplicating a layer” in a PhotoShop tutorial [14]. These steps may be conveyed through text, images, and video clips. Among the 13 tutorials we studied, 12 used text descriptions, 10 featured images, and 7 included video clips. Auxiliary elements can enrich the primary media. Timestamps help locate the step in the original video, overlays emphasize parts of an image, and glyphs connect images or text. We found 5 out of 7 tutorials with video clips also provide timestamps; two tutorials have overlays on images, and one uses glyphs.

Figure 2 provides step examples in mixed-media tutorials. Specifically, Figure 2a shows a step in an interactive smartphone tutorial, marked by an overlay indicating the screen area to be clicked [68]; Figure 2b depicts an auto-generated YouTube Chapter for a DIY craft video featuring text, images, and video clips (with timestamps); Figure 2c illustrates a step in a cooking tutorial, where red and blue dots signify ingredients and actions, respectively [12]. Comprehensive details are in the Appendix.



(a) A step with an image and overlays in a smartphone tutorial [69]



(b) A step with text, images, video clips, and temporal metadata in a DIY craft tutorial [44]



(c) A step with text and glyph in a cooking tutorial [12]

Figure 2: Examples of steps in mixed-media tutorials (images used with permission)

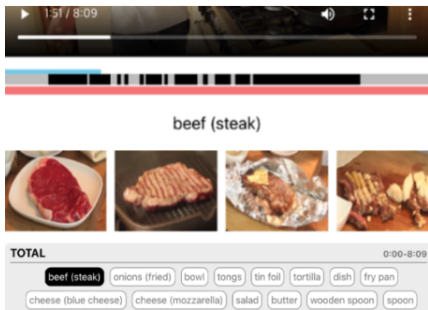
4.2 Object

Many mixed-media tutorials explicitly specify objects required for the task, such as ingredients and equipment in cooking tutorials [74], and UI widgets in software tutorials [20]. These objects can be represented through text, images, and timestamps marking their appearance in videos. In our dataset of 13 mixed-media tutorials, 7 explicitly included object components. While the remaining 6 tutorials contained objects implicitly in the instructions, they did not extract and represent these objects as individual components. All 7 tutorials with object components featured text descriptions, 3 incorporated object images, and 2 had appearance time in videos.

Things You'll Need

- Hose
- Roof cement
- Chisel
- Hammer

(a) Object components represented using text and interactive check-boxes in a roof repairing tutorial [6]



(b) Object components represented using text, images, and appearance time in a cooking tutorial [75]

Figure 3: Examples of objects in mixed-media tutorials (images used with permission).

Additionally, 3 offered interaction features, including checkboxes or clickable buttons that link objects with other components.

Figure 3 illustrates examples of objects in mixed-media tutorials. Figure 3a displays an object component from a roof repair tutorial on WikiHow [6], with interactive checkboxes to help users gather things needed; Figure 3b shows object buttons; clicking on an object button (e.g., “beef (steak)”) brings up video frames containing that object and the appearance time on the timeline [74]. All the examples are in the Appendix.

4.3 Dependency

Dependencies between steps are everywhere; they could be food processing order in recipe tutorials [12, 49, 74], concept prerequisites in lectures [39] and facial parts in makeup tutorials [65]. Dependencies may imply a different order than the one presented in the original instructional video. For example, in a cake recipe video, though the preparations of dry and wet ingredients are shown sequentially, they could be done in parallel [70]. In the TutoAI framework, we focus on physical tasks, where the dependencies between steps are the execution order. Of our collected 13 examples, 5 include dependencies explicitly. Among those 5, 4 utilize spatial layout to encode the dependency, 3 have links in the diagram.

Figure 4 shows dependency examples. Figure 4a shows groupings in a makeup tutorial where steps within each group are sequential but independent of other groups. Figure 4b maps out the dependencies of concepts in a lecture: orange nodes are already covered, and

FACE

22. I'm going to take my mineralized skinfinish in the ...

23. Forget you can use p star in the store at morphe in ...

24. I'm gonna take my favorite blush captivating by tarped.

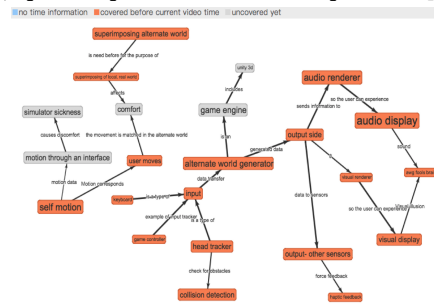
EYES

25. I'm going to use this browsing my benefit. And I'm ...

LIPS

26. I'm going to take lip land cream corset by Samantha. And ...

(a) Spatial dependencies in a makeup tutorial [66]



(b) Concept prerequisites in a lecture [39]



(c) Action dependencies in a cooking tutorial [75]

Figure 4: Dependency examples in mixed-media tutorials (images used with permission).

gray nodes are not. Figure 4c outlines cooking steps in different rows and columns: steps on the same row must be done sequentially, but steps on different rows could be done simultaneously; steps are also grouped by spatial dependencies (e.g., cutting board) in rectangles. All the examples are in the Appendix.

5 LEVEL 2: ASSEMBLE AND EVALUATE MODELS

We first review applicable models and candidate pipelines to extract mixed-media tutorial components. We then evaluate them on an annotated dataset of 347 cooking videos and finalize a pipeline.

Note that we only apply ML models to step and object extraction; for dependencies, we build a directed acyclic graph (DAG) based on the temporal order and shared objects between steps.

5.1 Applicable models and candidate pipelines

5.1.1 Step extraction. For the sake of completeness, we assume that a step component needs the following: a text description, the start and end timestamps in the video, and a representative video frame (thumbnail). As mentioned in section 3.2, we first identify relevant models:

- **Models for text descriptions.** We identified two types of models for generating text descriptions: text summarization and video dense captioning. Text summarization takes a chunk of text as input and shortens it while preserving the key information [17, 35, 46, 56]. Video dense captioning takes video frames and step timestamps as input and generates text descriptions for objects and their interactions within the step’s boundary [29, 69, 83].
- **Models for step timestamps.** We identified four model types for obtaining step timestamps: natural language video localization (NLVL), shot boundary detection, video summarization, and LLM prompting. NLVL localizes the start and end time of a step given a video and a step text description [21, 59, 77]. Shot boundary detection takes video frames as input, and returns candidate shot transition frames. Assuming that each shot represents a step, we can convert adjacent transition frame indices into the start and end timestamps [63, 82]. Video summarization condenses a long video by selecting and stitching together keyframes to form a shorter video [1, 26, 62, 84]. Similar to shot boundary detection, we can convert adjacent keyframe indices into step timestamps. We can also prompt LLMs to generate step timestamps if the input transcript contains word or sentence-level timestamps.
- **Models for step thumbnails.** We identified two types of models for selecting thumbnails: video summarization and shot boundary detection. As mentioned before, video summarization outputs representative keyframes. In addition to representative keyframes, shot boundary detection can filter dissimilar frames to get more thumbnail candidates.

To assemble pipelines that extract all the step information, we start with models that take video frames and transcripts as input and chain additional models based on the output. Figure 5 shows 4 candidate pipelines.

- **Pipeline 1: text summarization + NLVL + shot boundary detection.** As shown in Figure 5, pipeline (1) uses text summarization to extract step descriptions from the transcripts. Using step descriptions and the input video frames, it then leverages NLVL to obtain step timestamps. Lastly, it applies shot boundary detectors to derive thumbnails.
- **Pipeline 2: LLM + shot boundary detection.** Pipeline (2) uses LLM prompting to fetch both step descriptions and timestamps, followed by shot boundary detection to produce step thumbnails.
- **Pipeline 3: shot boundary detection + text summarization + shot boundary detection.** Pipeline (3) begins with

shot boundary detection to obtain step timestamps, followed by text summarization for text descriptions of each step, and concludes with another round of shot boundary detection for step thumbnails.

- **Pipeline 4: video summarization + video dense captioning.** Pipeline (4) employs video summarization to identify step thumbnails, and then obtains timestamps by converting adjacent keyframe indices into start and end timestamps. Given timestamps and video frames, dense captioning models generate step descriptions.

5.1.2 Object extraction. For the sake of completeness, we assume that an object component needs the following information: object names and an image containing the object’s bounding box. We have identified relevant models:

- **Models for object names.** We identified three types of models to extract object names: Part-of-Speech (POS) taggers, LLM prompting, and traditional object detectors. POS taggers take text as input, categorizing words’ roles in a sentence with grammatical properties such as nouns and verbs [2]. Obtaining object names from POS tagging results requires parsing nouns. LLMs can also be prompted to extract object names from text input. Traditional object detectors are trained on predefined object categories and, given input images, output detection names and bounding boxes [19, 38].
- **Models for object bounding boxes.** We identified two types of models to obtain object bounding boxes: traditional and open-vocabulary object detectors. As mentioned before, traditional object detectors take images as input and return bounding boxes as output. However, it can only recognize objects in the training dataset. Open-vocabulary object detectors take in both object names and images, and output bounding boxes for the object names [30, 36, 47].

After considering the relevant models, we assemble them into three candidate pipelines.

- **Pipeline 1: POS Taggers + Open-vocabulary detectors.** As shown in Figure 6, pipeline (1) uses POS taggers to identify object names from the video transcript. It then passes these names and video frames into open-vocabulary object detectors to localize the objects.
- **Pipeline 2: LLM + Open-vocabulary detector.** Pipeline (2) prompts an LLM to extract object names from the transcript and runs an open-vocabulary object detector.
- **Pipeline 3: traditional object detectors.** Pipeline (3) only uses traditional object detectors to obtain both the object names and bounding boxes.

5.2 Evaluation of applicable models and candidate pipelines

5.2.1 Overall evaluation approach and metrics. We evaluate models within the mentioned pipelines and discard any with subpar performance. Based on available source code and pre-trained models, we use at least one state-of-the-art (SoTA) implementation for each model type. While objective metrics are utilized, we also conduct manual inspections, especially when standard metrics fail to capture the error profiles. In the following subsections, we report the

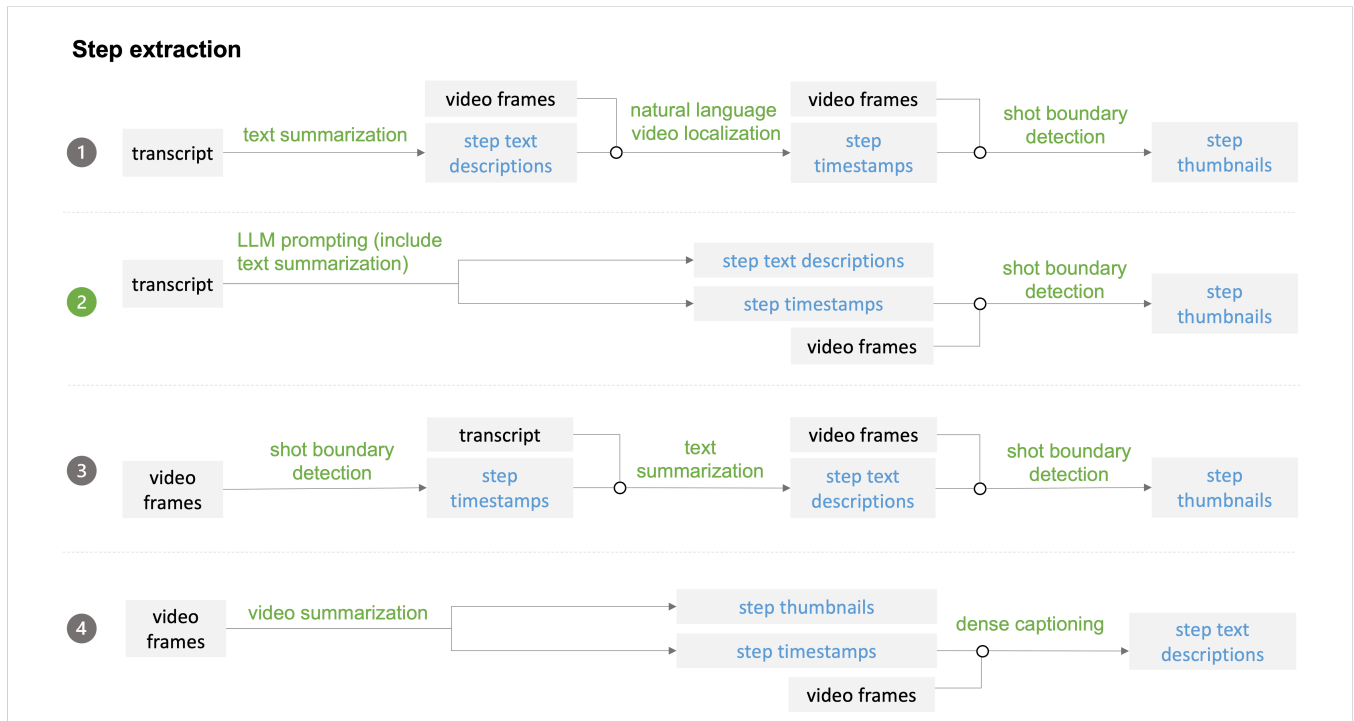


Figure 5: Four candidate pipelines for step extraction. Models are in green, and generated subcomponents are in blue. After evaluation, the chosen one is No.2.

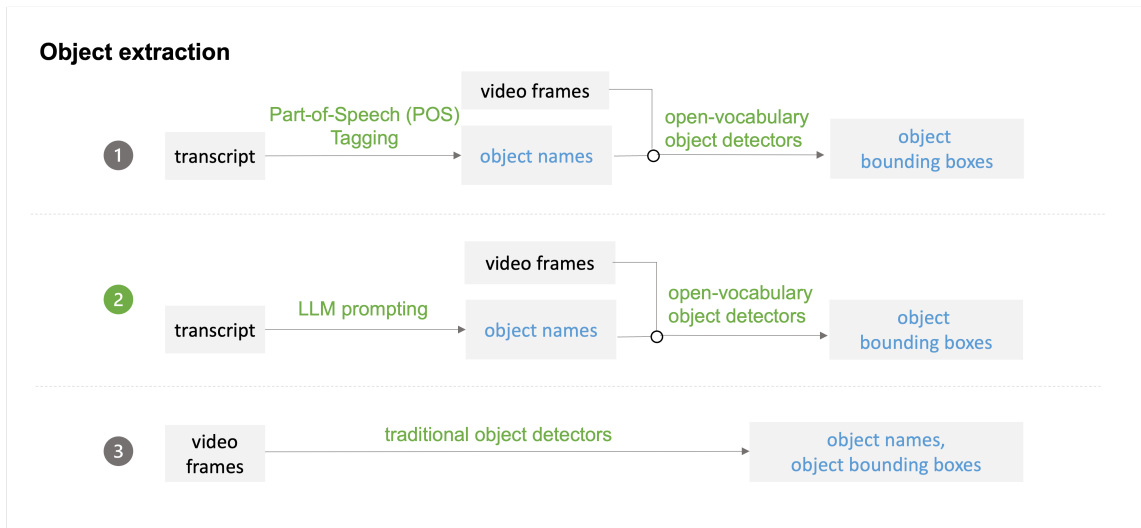


Figure 6: Three candidate pipelines for object extraction. Models are in green, and generated subcomponents are in blue. After evaluation, the chosen one is No.2.

main findings from the evaluation. Appendix A.1 provides detailed information about the evaluation dataset and results.

5.2.2 *Evaluation dataset.* We evaluated on the validation set of YouCook2 [82], containing 347 cooking videos with auto-generated

English transcripts. Each video has human-annotated objects, step descriptions, and start/end times.

5.2.3 *Step pipeline evaluation.*

Text descriptions: transcript summarization. Pipeline 1, 2, and 3 rely on text summarization to derive step text descriptions. We

assessed five methods, spanning both extractive (pulling key sentences from the source text, e.g., LexRank [46], TextRank [17]) and abstractive (rephrasing the original content, e.g., BART [35], T5 [56], GPT-3 [9]) methods. Among the five methods, GPT-3 leads by a large margin in ROUGE scores [37] (Appendix Table 5).

Traditional NLP metrics might not effectively gauge the quality of text generated by LLMs [40]. Through manual comparisons between GPT-generated descriptions and human annotations, we noted discrepancies that could affect ROUGE scores without necessarily compromising summarization quality. For instance:

- LLM identifies optional steps, e.g., put the salad in the fridge.
- LLM turns states into steps, e.g., from the statement “I’ve preheated my oven to 375 degrees”, it derived a step “Preheat oven to 375 degrees”.
- LLM includes more cooking details, e.g., temperature.

Given this, we decided to select LLM for text summarization.

Text descriptions: video dense captioning. Pipeline 4 relies on dense captioning to obtain text descriptions. We evaluated two video dense captioning methods: MT [83] and PDVC [69] and there are evident errors in object names and actions. For example, in the video “How to Make Fried Calamari | Hilah Cooking”¹, the human annotation is “drop the squid pieces into the oil”, but the dense captioning returns “add the chicken in a pot of water boil”. Consequently, we decided not to incorporate dense captioning models, leading to the removal of pipeline 4.

Step timestamps. In the remaining pipelines, we evaluated models to identify timestamps: NLVL method DORi [59] (Pipeline 1), LLM prompting (GPT-3 [9]) (Pipeline 2) and shot boundary detector ProcNets [82] (Pipeline 3).

For pipeline 1, we provided the video and ground truth step descriptions to DORi [59] to predict each step’s start and end time. After manual inspection, we found that the returned steps did not observe the order (e.g., step 3 is localized before step 2) and returned overlapping steps. Given the considerable editing effort required for such errors, and other NLVL models suffer from similar limitations, we eliminated Pipeline 1.

For pipeline 2, we applied LLM alone to predict the boundary timestamps. We sent a transcript and a prompt “*summarize the video transcripts in several steps and find the start and end time for each step*”. The transcript format is the same as the YouTube transcript, with each sentence beginning with a timestamp. Since this approach predicts both the step summaries and timestamps simultaneously, complicating quantitative evaluation without timestamping all 347 videos manually. We sampled 20 videos and conducted a qualitative evaluation, showing LLM returns ordered and non-overlapping steps, and the step descriptions and timestamps were reasonably matched with the ground truth.

For Pipeline 3, we employed ProcNets [82] to determine video shot boundaries. Relying solely on frame visuals, ProcNets scores each segment. We evaluated top-scored segments against the ground truth by computing the average temporal intersection over union (tIOU), however, given a low alignment (tIOU = 0.18), we didn’t proceed to generate text summarization for each step.

Therefore, we retained Pipeline 2 for extracting steps.

5.2.4 Object pipeline evaluation. As shown in Figure 6, individual model types include POS taggers (pipeline 1), LLM prompting (pipeline 2), open-vocabulary detectors (pipeline 1 and 2) and traditional object detectors (pipeline 3).

Object names. In Pipeline 1, we applied POS tagger Flair [3] to extract object names. For Pipeline 2, we prompted GPT-3 [9, 52] with the transcript and an instruction: “Identify the objects, ingredients, tools, equipment in this tutorial” and parsed objects from the response. In Pipeline 3, we employed a faster R-CNN [58] trained on the Visual Genome dataset [33]. Both POS taggers and GPT-3 outperformed visual detectors in identifying true positives. However, POS taggers often identified non-cooking objects, e.g., the chef’s necklace (Appendix Table 6). As such, we retained only Pipeline 2, leveraging LLM for object extraction.

Object bounding boxes. Considering the underwhelming results of traditional object detectors, we only evaluated open-vocabulary object detectors and eventually chose OWL-ViT [47] considering both performance and computational cost.

5.3 Final pipeline

We finalized our pipeline as shown in Figure 7, which includes Step pipeline 2 (Figure 5) and Object pipeline 2 (Figure 6). First, we extract steps from video transcripts by prompting LLM (here we use GPT-3.5 [50], assuming it has better performance than GPT-3): “Summarize the video transcripts in several steps and find start and end time for each step,” then we use a shot boundary detector [63] to pick thumbnails for each step. Next, to extract object components, we make a different prompt: “Find out what objects/ingredients/tools/equipment are required in this tutorial.” Then, we run an open-vocabulary detector [18] to identify the bounding boxes in video frames. Finally, we match object names to each step’s description via string match, then build dependencies between steps by the shared objects.

6 LEVEL 3: USER INTERFACES FOR MIXED-MEDIA TUTORIAL CREATION

6.1 Design considerations

Section 4 shows various mixed-media tutorial formats regarding visual representation, layout, and interactivity tailored to specific domains. Rather than advocating a one-size-fits-all format, we embrace the principle of *separating content from style*: mixed-media tutorial components are content that can be extracted, reviewed, and edited, with different styles (e.g., visual representations, layouts, and interactive behaviors) added later. We focus on enabling creators to inspect and modify content, assuming that a tool will auto-apply styles to the final tutorial. Thus, we propose the following UI design considerations to elevate the creator experience without information overload (D3).

- C1 Component-based creation.** The UI should break down the creation process into individual tasks based on the mixed-media tutorial components. The UI should sequence tasks so that the output from one task can provide context to help users perform subsequent tasks efficiently.

¹<https://www.youtube.com/watch?v=-k7trpuj3X8>

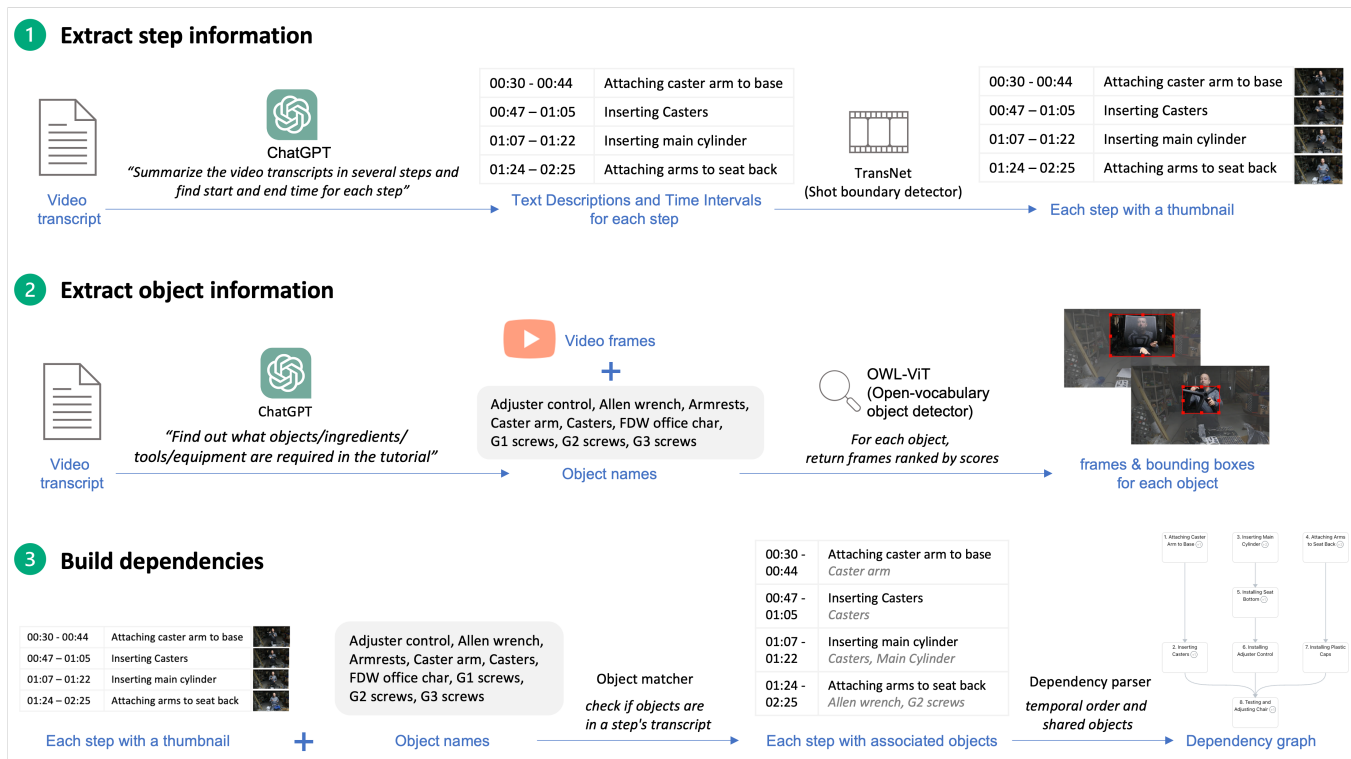


Figure 7: TutoAI’s machine learning pipelines to obtain objects and steps in instructional videos: 1. extract steps: ChatGPT processes the video transcript to produce text descriptions and time intervals for each step, then a shot boundary detector augments each step with a thumbnail; 2. extract objects: ChatGPT identifies the objects in the tutorial, then an open-vocabulary object detector returns the frames and bounding boxes of the objects; 3. build dependencies: an object matcher checks if objects are in a step’s transcript and produces a dependency graph.

C2 One modality at a time. To reduce context switching, when a component encompasses multiple modalities (i.e., text and images), the UI should break it down into subtasks. This will help simplify user interactions and avoid requiring users to operate across multiple modalities in a single task.

C3 Editable AI output. The UI should enable creators to keep, modify, or dismiss AI-generated results and add information missed by AI.

C4 Real-time edit preview. Upon editing, the UI should automatically reflect changes in the tutorial.

6.2 An example prototype

We reify these design guidelines into an example UI and use the video “How to make a seesaw for kids”¹ as input. In this implementation, we use a tutorial format depicted in Figure 8. The tutorial contains the following components: a video player and step boundary below it (Figure 8A), an object list (Figure 8B) over which users can hover to see an image of the selected objects (Figure 8C); step overviews, which consist of a text description, a representative thumbnail and objects for each step (Figure 8D); associated dependencies (Figure 8E), represented as arrows between steps, and the

buttons on the arrow show objects that connect steps. We chose this tutorial design for its comprehensive components without domain-specific assumptions.

The UI breaks up the creation process into five sequential tasks, each targeting a single tutorial component – steps, objects, or dependencies – in a single modality (C2). Creators can bypass any tasks and accept the default results if they deem the task unnecessary (C3). As they make changes, creators can preview the updates with the current modifications (C4) by the “view” button (Figure 9). Here is the workflow:

1) Identify steps. The UI shows the video and its transcript on the left, AI-generated steps with text descriptions and start/end timestamps on the right (Figure 9); creators can edit the text, add/delete steps, and update the time boundaries by dragging the range slider (C3).

2) Choose step thumbnails. The UI presents dissimilar candidate video frames. Creators can adjust the number of frames using a “show more/less” slider, and select a frame. (Appendix Figure 12). The thumbnails presented for a given step are bounded by the time boundaries identified for that step in task 1 (C1).

¹<https://www.youtube.com/watch?v=drDSY3ZZqnQ>, used with permission

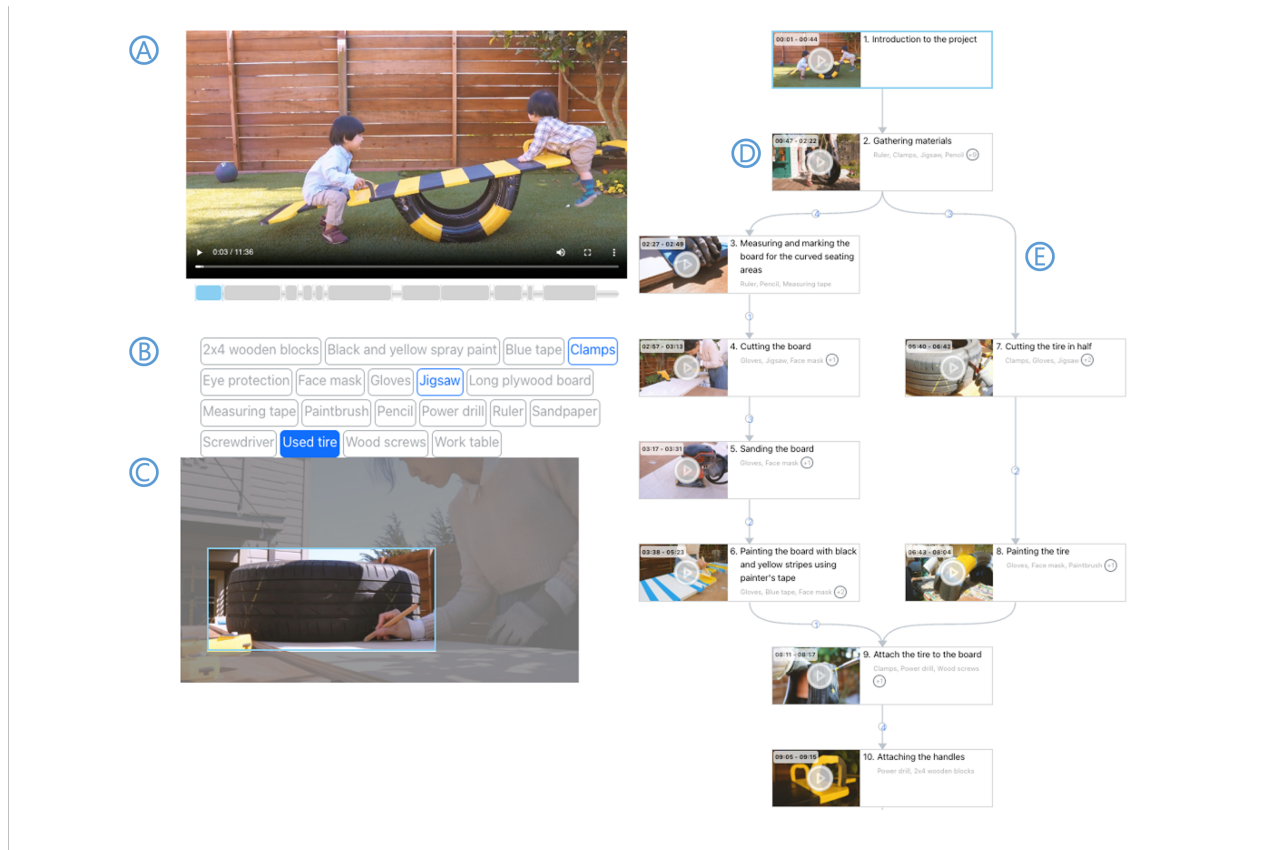


Figure 8: A mixed-media tutorial template on making a seesaw for kids: below the video player (A) is a list of required objects (B); hovering on the blue-bordered object will show the object’s image along with a bounding box (C); on the right is an overview of steps, (D) each step is a video clip with start and end time, text descriptions and associated objects. (E) The arrows between the steps indicate the dependencies.

3) Select objects. The UI suggests an object list required for the tutorial and associates the objects with the steps (Appendix Figure 13). Creators can modify objects and change their step associations (C3).

4) Crop objects. Creators can choose a representative image for each object (Appendix Figure 14). The UI shows a list of objects refined by users in task 3 (C1) and presents candidate frames with probable object bounding boxes, which creators can adjust (C3).

5) Build dependencies. The final task is to build dependencies (Appendix Figure 15). The UI displays a node-link diagram of dependencies based on shared objects between the steps, as identified in task 3 (C1). Creators can add/delete links via drag and drop (C3).

7 TUTOAI FRAMEWORK EVALUATION: MODEL

To demonstrate our pipeline’s generality, we evaluated it on a small yet diverse dataset.

7.1 Dataset

Inspired by the object-action quadrant for instructional videos [13], we considered the following diversity dimension of instructional

videos: creator, task, video duration, number of steps, number of objects. The content creator dimension allows us to capture variations over editing styles such as instructional or conversational narration, concise versus verbose steps, use of music fillers, etc. As a result, we collected a dataset of 20 videos (Table 1) across four domains: cooking, crafting, makeup, and repair. Each video within a domain focused on a different task (e.g., fixing an iPhone vs. fixing a hole in the wall for repairs) and was made by a different creator. We manually annotated the 1) objects and 2) step boundary timestamps and used these as ground truths. We assessed our pipeline on object extraction and timestamp prediction.

7.2 Object extraction results

We compare object extraction results with the ground truth using the F1 score, computed as:

$$F1(o_{ours}, o_{gt}) = \frac{2|o_{ours} \cap o_{gt}|}{|o_{ours}| + |o_{gt}|}$$

where o_{ours} is the set predicted by our pipeline and o_{gt} is the ground truth, and $|o|$ denotes the number of objects in the set. As shown in Table 1 column 8 (“F1”), our object extraction F1 scores fall between

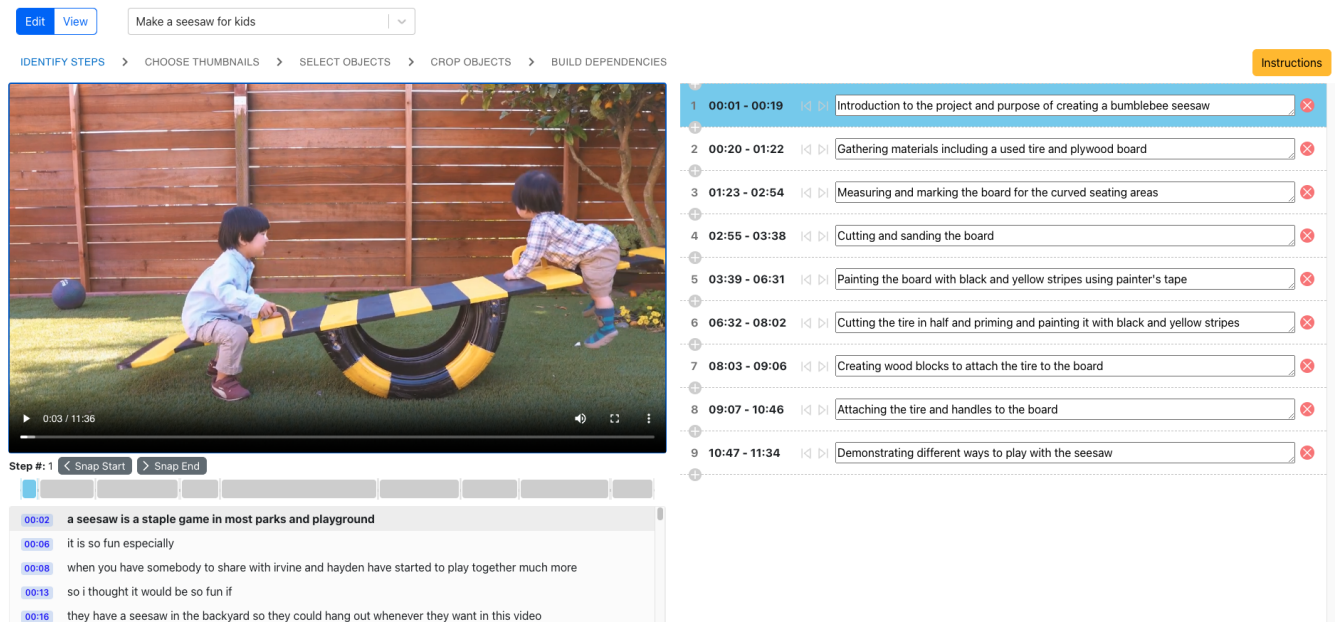


Figure 9: Identify steps. This task aims to break down the video into several steps and provide text descriptions and time boundaries for each step. On the left is a video player and its transcript (“Make a seesaw for kids”); on the right are the AI-generated steps.

0.56 to 1, with an average of 0.88, indicating great performance across domains. False negatives often resulted from objects not explicitly referenced in the transcript.

7.3 Step boundaries

Our pipeline outputs a sequence of steps, including text descriptions and start and end timestamps. On average, it yields 1.3 false negative steps and 0.25 false positive steps per video (Table 1 column 11 “# False Neg.” and column 12 “# False Pos.”). The low false negative and false positive rates suggest that our pipeline does a good job of extracting steps. Introduction and conclusion segments accounted for most false negative steps, and false positive steps were incorrectly inferred from verbose narrations. We then used F1 score to assess predicted timestamps against the ground truth. For false negative steps, we set t_{ours} to $[0, 0]$ to signify that this step did not appear. Aggregate F1 scores ranged from 0.22 to 0.95, averaging 0.59 (Table 1 column 13 “Avg. F1”). In general, we found that our pipeline performed better on the step localization task for shorter tutorials and tutorials with more concise steps. Certain video editing decisions, such as using non-speech fillers between steps, showing step execution before verbally describing it, and describing steps out of order, also negatively impacted localization. Our aggregate F1 score suggests reasonable alignment between predicted step boundaries and ground truth with room for improvement, which can be achieved via more sophisticated prompt engineering.

8 TUTOAI FRAMEWORK EVALUATION - UI

To evaluate the quality of AI-extracted components perceived by users and the tutorial creation experience, we conducted two preliminary user studies to understand 1) if the TutoAI framework generates higher-quality mixed-media tutorial components than a baseline method before editing, 2) if the TutoAI framework generates mixed-media tutorials that are more useful for consumers than a baseline method after editing, and 3) the potential of integrating TutoAI into creators’ existing workflow.

8.1 Study design rationales

We identify both instructional video consumers and influencers who make instructional videos as potential users of our prototype. Video consumers who want to learn instructional content are motivated to interact with the mixed-media tutorials and can benefit from tutorial creation. For example, Kim et al. find that when students contributed to creating subgoal-based tutorials, they became more attentive to learning [70]; popular video platforms also support video consumers to create video clips (e.g., YouTube’s “create clip”²) and mixed-media notes (e.g., Coursera’s “save note”³). Therefore, we recruited participants who frequently watch instructional videos for study 1. Several participants also disclosed that they had created mixed-media tutorials before, confirming our assumption. For study 2, we recruited two YouTube creators who regularly publish instructional videos.

²<https://support.google.com/youtube/answer/10332730>

³<https://blog.coursera.org/ready-for-retention-presenting-a-unified-note-taking-experience/>

Video ID	Domain	Duration (minutes)	Objects					Steps				
			Ours # Obj.	GT # Obj.	False Neg.	False Pos.	F1	Ours # Steps	GT # Steps	# False Neg.	False Pos.	Avg. F1
36FOyZ26ld0	cooking	0:24	10	10	0	0	1	4	5	1	0	0.95
j4UVB6MPsKw	cooking	5:27	16	16	3	3	0.81	6	6	0	0	0.80
BAp1AXn82Pg	cooking	7:32	20	23	3	0	0.93	8	9	1	0	0.72
Y-Y9CXGRJPU	cooking	13:50	24	26	3	1	0.92	9	12	3	3	0.34
L0Gu2KDCS6o	cooking	15:10	17	19	2	0	0.94	9	12	3	0	0.22
zQ8gThfBDqU	crafting	3:40	12	14	2	0	0.92	11	11	0	0	0.69
OUMfV1D0_RQ	crafting	4:58	8	6	1	3	0.71	9	9	0	0	0.72
SX4DCFDKMzc	crafting	7:48	13	18	6	1	0.77	13	13	0	0	0.65
DU4DiGeLr6Y	crafting	10:21	5	5	0	0	1	6	7	1	0	0.74
VKZI7X-UIe8	crafting	18:55	17	19	2	0	0.94	7	8	1	0	0.52
Ls969BmW1kw	makeup	5:00	13	13	3	3	0.77	9	12	3	0	0.57
skZ-nUB_b00	makeup	5:26	10	12	2	0	0.91	13	13	0	0	0.70
QmPiBCu5_ME	makeup	7:49	16	18	2	0	0.94	10	12	2	0	0.71
gkkmHizG2As	makeup	13:10	8	9	1	0	0.94	6	6	0	0	0.69
9f7zmCSzG9E	makeup	13:26	25	25	2	2	0.92	8	11	3	0	0.42
lj7YK1IIRUM	repair	2:23	16	16	0	0	1	14	15	1	0	0.81
ZWlq_fWRrZI	repair	4:09	9	7	1	3	0.75	7	9	2	0	0.39
B4iWwUzxFWA	repair	4:17	5	13	8	0	0.56	4	6	2	0	0.61
p55lnFCorQ4	repair	9:57	11	9	1	3	0.8	12	15	3	2	0.31
b-GLL-Vsu9s	repair	11:38	11	12	1	0	0.96	10	10	0	0	0.33

Table 1: Pipeline evaluation on ground truth. We annotate ground truth for 20 instructional videos from 4 different domains and test the object extraction and step boundary detection components of our pipeline on these videos. Our pipeline performs object extraction very well (average F1 = 0.88) across domains. Our steps boundary detection performs relatively well on at least one video in each domain (F1 = 0.59).

In both studies, we used auto-generated YouTube Chapters [54] as the baseline. Although TutoAI was inspired by previous works, these tutorials were either generated automatically using a domain-specific approach [14, 20, 31, 65, 68] or manually without AI assistance [39, 74]. Mixed-initiative approaches [12, 32, 49, 53] do not provide comparable creation experience like TutoAI. We thus determined that YouTube Chapters [54] is the most reasonable baseline since they also support cross-domain generation of steps.

8.2 Study 1: general users

8.2.1 Recruitment: we recruited 24 participants (female: 10, male: 13, non-binary: 1) who regularly watch instructional videos on YouTube (several times a week: 7, several times a month: 14, several times a year: 3). They watch instructional videos in various domains: cooking (19), home projects (15), software & programming (15), sports & fitness (13), electronics (9), beauty (6), and animals & pets (3). 12 participants have used the YouTube Chapter feature. Though not prolific YouTube creators, five participants have created video tutorials: for a mobile app (P1), cooking (P5), robots (P11), design tools (P19), and Android development (P20).

8.2.2 Instructional videos: we chose two instructional videos on YouTube: office chair assembly⁴ and strawberry blueberry shortcakes⁵. We randomly split the participants into two groups: A (office

chair assembly, video length: 5 minutes 18 seconds) and B (strawberry blueberry shortcakes, video length: 7 minutes 32 seconds). Participants’ median familiarity with the video topic was 2.5 and 3.0, respectively (1: not familiar at all, 5: extremely familiar).

8.2.3 Procedures: First, we briefly introduced the concept of mixed-media tutorials and editing features of the UI, then, participants followed a step-by-step instruction to reproduce a Kung Pao chicken⁶ mixed-media tutorial created by TutoAI as a warm-up. Then, the participants were asked to create a mixed-media tutorial for the assigned video and think aloud. Next, participants completed a survey and provided open-ended feedback. Each session was remotely conducted over Zoom and lasted about 1 hour. Each participant received a \$20 Amazon gift card. The study was approved by the Institutional Review Board (IRB) Committee.

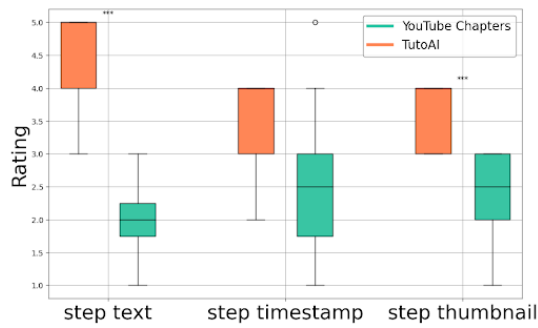
8.2.4 Findings: We observed that participants applied different strategies to create mixed-media tutorials. Some participants watched the entire video first, some watched each step’s video clip based on the AI-generated results first, and some did not watch the video but read the transcript instead.

Quality of AI-generated results. We asked the participants to rate the quality of components generated by TutoAI *before editing* and YouTube auto-generated Chapters on a five-point Likert scale, where 1 means “the quality is so low that the author needs to start from scratch”, and 5 means “the quality is so high that

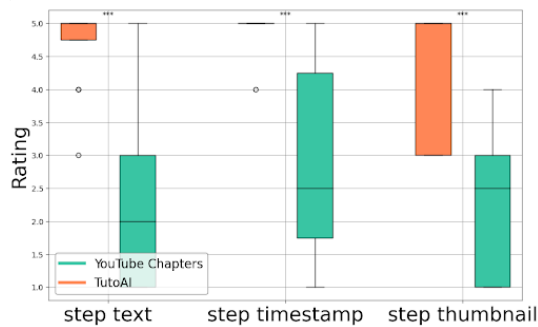
⁴<https://youtu.be/OEIDupReh8Q>

⁵<https://youtu.be/BAp1AXn82Pg>

⁶from the YouCook2 dataset: <https://youtu.be/ntiGX3X-spA>



(a) **Quality before editing, TutoAI vs. YouTube Chapters, text: 4.6 ± 0.65 vs. 2.0 ± 0.71 ($p=0.009$); timestamps: 3.5 ± 0.65 vs. 2.5 ± 1.19 ($p=0.075$); thumbnails: 3.6 ± 0.49 vs. 2.3 ± 0.75 ($p=0.021$)**



(b) **Usefulness after editing, TutoAI vs. YouTube Chapters, text: 4.7 ± 0.62 vs. 2.3 ± 1.25 ($p=0.003$), timestamps: 4.9 ± 0.28 vs. 2.8 ± 1.52 ($p=0.021$), thumbnails: 4.3 ± 0.94 vs. 2.2 ± 1.16 ($p=0.015$)**

Figure 10: Component quality of group A: office chair assembly. Before editing (left), after editing (right).

the author barely needs to do anything”. YouTube Chapters only generates timestamps, thumbnails, and text descriptions for each step. We conducted a Wilcoxon Signed-Rank test with a Bonferroni correction, and found TutoAI generated higher quality results than YouTube chapters in 2/3 comparisons in group A (Figure 10a): TutoAI vs. YouTube Chapters, text: 4.6 ± 0.65 vs. 2.0 ± 0.71 ($p=0.009$); timestamps: 3.5 ± 0.65 vs. 2.5 ± 1.19 ($p=0.075$); thumbnails: 3.6 ± 0.49 vs. 2.3 ± 0.75 ($p=0.021$). For group B, the benefits of TutoAI are not statistically significant (Appendix Figure 17 (a)): TutoAI vs. YouTube Chapters, text: 4.4 ± 0.64 vs. 3.6 ± 1.04 ($p=0.138$); timestamps: 3.3 ± 1.25 vs. 3.0 ± 1.0 ($p=1.000$); thumbnails: 3.4 ± 0.76 vs. 2.4 ± 1.38 ($p=0.138$). Other scores of TutoAI components are in Appendix Figure 18.

Perceived Usefulness of Tutorial Components. We asked participants to rate each component’s usefulness for tutorial consumers *after editing*, where 1 refers to “I don’t think consumers will benefit from this component,” and 5 refers to “I’m confident that consumers will benefit from this component.” We conducted a Wilcoxon Signed-Rank test with a Bonferroni correction, and found TutoAI results more useful than YouTube Chapters in 3/3 comparisons in group A (Figure 10b). Specifically, TutoAI vs. YouTube Chapters, text: 4.7 ± 0.62 vs. 2.3 ± 1.25 ($p=0.003$), timestamps: 4.9 ± 0.28 vs. 2.8 ± 1.52

($p=0.021$), thumbnails: 4.3 ± 0.94 vs. 2.2 ± 1.16 ($p=0.015$). For group B, the benefits of TutoAI are not statistically significant. TutoAI vs. YouTube Chapters, text: 4.8 ± 0.43 vs. 3.8 ± 1.16 ($p=0.063$), timestamps: 4.8 ± 0.37 vs. 4.0 ± 1.22 ($p=0.192$), thumbnails: 4.0 ± 0.91 vs. 2.6 ± 1.50 ($p=0.153$). Other scores of TutoAI components are in the Appendix Figure 19.

TutoAI vs. YouTube Chapters. Although TutoAI has received higher scores than YouTube Chapters in both videos in the user study, the statistical results are insignificant for the strawberry blueberry shortcake video. We looked into the user study recordings and found that since text descriptions of YouTube Chapters are very short (“Strawberry topping” and “Chantilly cream”), the participants deem them to be helpful as long as they contain important keywords. In comparison, the step descriptions generated by TutoAI are “Preparing the strawberries for the topping” and “Preparing the Chantilly cream using an air disc container”. Although TutoAI provided more details, the participants believe the essential keywords have been captured by YouTube Chapters. On the other hand, the YouTube Chapters for the office chair assembly video missed most keywords, e.g., “Base Assembly”, and were deemed less useful than TutoAI-generated text descriptions: “Attaching Caster Arm to Base”. To more conclusively demonstrate the superiority of the fine-grained text descriptions generated by TutoAI, we need more experiment data involving more instructional videos.

Dependencies and other components. Many participants (17/24) found the dependency diagram useful (rated 4 or 5), e.g., P12 said “The flow charts were amazing...if I didn’t want to watch the video, I could just see the steps...I am getting a visual representation of the whole video.” While some expressed confusion, P4 said “dependency diagram was a bit tricky to understand.” Besides existing components, participants also brainstormed new tutorial components, e.g., 3D object augmentation/more camera angles (P11).

Application Scenarios. The participants shared situations where they would like to have a mixed-media tutorial, e.g., build a pet snake vivarium (P5) and collaborative software development (P8). Some participants also mentioned situations where they would like to create a mixed-media tutorial to refresh their memory, e.g., P9 said “I make quilts, and I have to look up a lot of tutorials for how to finish the quilt because you only do it once every time.”

8.3 Study 2: YouTubers

8.3.1 Preparation: we recruited two YouTube creators (E1 and E2) who regularly publish instructional videos. For each YouTuber, we picked several of their videos with auto-generated YouTube Chapters. We ran our ML pipeline on the video: “bike rack installation”⁷ (E1) and “how to make a seesaw for kids”⁸ (E2) and loaded the results into TutoAI UI. During the study, we briefly introduced mixed-media tutorials and asked them to complete a step-by-step warm-up task to get familiar with the UI. Then, they created a mixed-media tutorial for the video and provided oral feedback along the way. Each participant received a \$50 Amazon gift card.

8.3.2 Findings: we asked them about the impression of AI-generated results and workflows in creating instructional videos.

⁷<https://youtu.be/5nHD0vy9R5g>, used with permission

⁸<https://youtu.be/drDSY3ZZqnQ>, used with permission

TutoAI vs. YouTube auto-generated Chapters. Both YouTubers spoke highly of the TutoAI-generated results, e.g., when asked about the quality of steps, E1 said *“I’d say probably about a 4 (out of 5). There were a few things I changed, but for the most part, it was a good starting point.”* When shown the auto-generated YouTube Chapters, E1 gave them a 2.5 to 3: *“the first few are getting the breaks pretty good, but they lost some of the steps that your software captured”*. E2 believed it needs a redo completely: *“I won’t be able to use any of this...“Wood blocks” is just the name of the material, not something meaningful for the viewers to imagine”*. The author-created steps are in Appendix Figure 16.

Attitudes towards dependencies. E1 expressed enthusiasm in applying dependency diagrams: *“I really like the dependency diagram, especially for a procedural how-to video...it helps them understand... when you might need to skip a step or there might be a branch...”*. E2 saw the dependency diagram has better use in cooking videos, *“for example, cooking...you can do many things at the same time. But for my (DIY) tutorial, it kind of depends on one flow.”*

Incorporate TutoAI into existing workflow. We asked both E1 and E2 to share their thoughts on incorporating TutoAI into their workflow. E1 said *“I think this is a great tool... I don’t know that it would necessarily save me time just creating chapters. It’s a different animal because this is giving me the ability to do a lot more, especially creating the flow charts, which I really like... viewers would get a lot out of this as opposed to just a regular chapter”*. E2 recounted that in the past, she spent about 1 hour writing down steps and time boundaries of a 10-min video she created (6 times of the original video length), and to her relief, with the help of TutoAI, it only took her 17.5-minutes to finalize steps and time boundaries for an 11.5-minute video (1.5 times of the original video length).

9 DISCUSSION

We have proposed TutoAI, the first cross-domain framework for AI-assisted mixed-media tutorial creation. TutoAI extends earlier efforts in generalizing tutorial creation beyond a single domain [32, 65, 70]. It adopts a holistic approach by distilling common tutorial components from existing work, presenting methodologies to identify, evaluate, and assemble AI models to extract components, and introducing a guided workflow for users to inspect and modify extraction results. In this section, we reflect on the lessons learned from our exploration and discuss the broader implications.

9.1 Selecting models and constructing pipelines

We demonstrated how to identify, evaluate, and assemble computational models into integrated pipelines to extract tutorial components. Given the rapid advancement in AI, we acknowledge that the pipeline we select may not sustain peak performance. For example, multi-modal LLMs are equipped with vision capabilities [43, 51, 78], and dense video captioning models may improve rapidly by benefiting from large-scale pre-trained models [85]. Despite technological advances, our work provides enduring insights that transcend the specific models. We propose the following guidelines for future endeavors that incorporate AI into tutorial creation:

- **Adopt a multi-modal perspective:** Models across different modalities could achieve similar goals, e.g., object detectors

based on video frames and LLM prompting based on transcripts can both identify object names, and each has its SoTA models. By assembling multiple pipelines with the same objective, we can explore the solution space more comprehensively without premature commitment.

- **Leverage strong models for cross-modal enhancement:** Currently, an LLM perform the best at extracting object names. Starting with the best results in one modality, we can minimize errors in other modalities, e.g., object names extracted by an LLM will guide open-vocabulary object detectors to localize objects. Future research should keep monitoring SoTA methods in different modalities.
- **Focus on user-centric model selection:** While each ML problem has standard metrics for evaluation, higher scores do not equate to better user experience. Though comparing models across modalities may not be straightforward due to distinct metrics, a potential universal metric could be the user’s effort required to refine the output. For example, an NLVL model DORi [59] returns higher tIOU (temporal intersection over union) than ProcNets [82] in video segmentation, but DORi does not observe the order of steps, leading to overlapping and reverse-ordered steps, which require additional user edits. To avoid overwhelming users, we eventually dropped the model.

9.2 Designing AI-Assisted user workflows

We believe it is important to tailor the design of mixed-media tutorial formats for different use cases. The tutorial format in our prototype shown in Figure 8 serves only as an example interface. The following guidelines can inform future efforts to design AI-assisted tutorial creation workflows.

- **Simplify tutorial creation by guiding and constraining user actions:** The sequential editing workflow in TutoAI is structured and domain-agnostic, following the Wizard interface design pattern [72]. One potential benefit of this approach is that the complex task of tutorial creation is transformed into a sequence of understandable stages, where the relationships between the stages are implicitly captured. Users can thus focus on individual tasks without worrying about how to structure the overall workflow. The UI should also ensure the results satisfy implicit constraints (e.g., the intervals of two steps should not overlap).
- **Separate content from style:** While mixed-media tutorials are available in diverse formats, TutoAI underscores the value of separating content from style. In our prototype, the user workflow focuses on extracting accurate component information; the visual representations and interactivity of the components in the tutorial are automatically applied to the extraction results. This general approach is adaptable to any mixed-media tutorial with a predefined format. Our prototype offers multiple formats for a customized consumer experience, including a list-based view of steps and a dependency diagram (Appendix Figure 16). Future tools can provide more flexibility in formatting tutorials, yet the principle of separating content from style remains valid.

- **Support graceful degradation:** The performance of ML models can be uncertain and unpredictable. Even though the overall performance of our pipeline is reasonable, it may be disappointing in some cases. Therefore, it is important to design a UI that supports tutorial creation when AI-powered component extraction fails. To support such graceful degradation, users must be able to interpret the extraction results and make edits easily. To facilitate this, our UI is designed for low-effort error correction, e.g., users can adjust step boundaries with a range slider. In the worst case, where the extraction result is completely wrong, users can override the results and update the component manually.

9.3 Cross-domain generalization: tutorials, tools, and methodologies

TutoAI is motivated by previous work’s effort to generalize mixed-media tutorial creation beyond a single domain. Reflecting on our experience, we have identified multiple interpretations of cross-domain generalization:

- **CD1: Same tutorial format, diverse domains:** a tool for creating tutorials with the same format.
- **CD2: Same creation experience, diverse tutorial formats and domains:** a general-purpose tool for creating tutorials with diverse formats.
- **CD3: Same methodologies, diverse creation experiences, tutorial formats and domains:** a set of generalized methodologies to guide the design and development of tutorial creation tools; the tools can be general-purpose or domain-specific, supporting the creation of diverse tutorial formats

It is not our intention to advocate a one-size-fits-all tutorial format (CD1), as we have discussed in Section 6.1 and Section 9.2. We believe a general-purpose creation tool (CD2) can be useful, as exemplified by our prototype. Nevertheless, a general-purpose tool risks overlooking domain-specific nuances in terms of both components and ML pipelines. In TutoAI, we are not only trying to build a general-purpose tool (CD2) but also propose a set of generalized methodologies for tool builders (CD3). With advancements in AI, we demonstrate the feasibility of designing tutorial creation tools systematically. Our framework, encompassing three levels – components, models, and UIs – and the associated guidelines, is adaptable to various contexts. For example, to develop a tutorial creation tool for software instructional videos, we can standardize the components first (e.g., UI widgets, commands, data), then identify, evaluate, and assemble ML pipelines based on the guidelines outlined in Section 9.1. Though the component and model details may differ, the underlying approach remains the same.

9.4 Limitations and future work

Domain limitations. Though TutoAI is a cross-domain framework, it does not apply to all instructional videos. Chang et al. [13] classified instructional videos into a quadrant along an object-action coordinate system, distinguishing between “Diverse objects and diverse actions” (cooking, car repair, makeup, etc.), “diverse objects and few actions” (crafts and packing, etc.), “few objects and few actions” (drawing, musical instrument, etc.), and “few objects and

diverse actions” (dance, exercise, etc.). TutoAI focuses on physical tasks that involve diverse objects. For instructional videos with few objects or without concrete objects (e.g., lecture videos), TutoAI will have difficulty in constructing dependencies, as the dependency parser assumes steps share the same object depending on each other. Another related limitation is that if the same object was referred to differently, e.g., in the berry cake video, the creator uses “berries” to refer to both strawberries and blueberries in the late stage, and our method fails to detect the dependency between steps containing “berries” and “strawberries” (or “blueberries”). Future work could investigate identifying abstract items and more intelligent dependency parsing, especially dependencies between abstract concepts.

Representative frames selection. Currently, we use shot boundary detectors to present diverse frames as step thumbnail candidates, independent of text descriptions. In the future, thumbnail selection could leverage the text descriptions. e.g., multi-modal video summarization methods can extract representative frames and text summaries [26, 48] simultaneously, having the potential to return high-quality text-dependent representative frames.

Framework evaluation. We use user-perceived component quality as a proxy for learning effects, though the two may not be positively correlated. Further research is necessary to study if user rating of tutorials directly translates to better learning outcomes. Besides, the fact that users interacted with TutoAI but only looked at static YouTube Chapters’ screenshots may also cause bias in users’ ratings.

10 CONCLUSION

Transforming linear instructional videos into more browsable mixed-media tutorials will significantly elevate the learning experience, however, existing methods do not harness the full potential of the latest AI advances and are usually limited to specific domains. In response, we introduced TutoAI, a cross-domain framework for AI-assisted mixed-media tutorial creation. TutoAI provides a taxonomy for mixed-media tutorial components, a methodology to evaluate and select models for component extraction, and guidelines for UI implementation. Our empirical evaluation underscored the capability of TutoAI in extracting high-quality mixed-media tutorial components and helping authors create mixed-media tutorials. Moving forward, we believe the TutoAI framework will provide a strong foundation for future mixed-media tutorial development.

REFERENCES

- [1] Meta AI. 2022. *Video Summarization*. <https://paperswithcode.com/task/video-summarization>
- [2] Meta AI. 2022. *Video Summarization*. <https://paperswithcode.com/task/part-of-speech-tagging>
- [3] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)*, 54–59.
- [4] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*, 1–13.
- [5] Gregor Betz, Kyle Richardson, and Christian Voigt. 2021. Thinking aloud: Dynamic context generation improves zero-shot reasoning performance of gpt-2. *arXiv preprint arXiv:2103.13033* (2021).
- [6] David Bitan. 2022. *How to Repair a Leaking Roof*. <https://www.wikihow.com/Repair-a-Leaking-Roof>

- [7] Alan Blackwell and Thomas Green. 2003. Notational systems—the cognitive dimensions of notations framework. *HCI models, theories, and frameworks: toward an interdisciplinary science*. Morgan Kaufmann 234 (2003).
- [8] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).
- [9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
- [10] Carrie J Cai, Emily Reif, Narayan Hegde, Jason Hipp, Been Kim, Daniel Smilkov, Martin Wattenberg, Fernanda Viegas, Greg S Corrado, Martin C Stumpe, et al. 2019. Human-centered tools for coping with imperfect algorithms during medical decision-making. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–14.
- [11] Gaëlle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Laurent Bouillon, and Jean Vanderdonck. 2003. A unifying reference framework for multi-target user interfaces. *Interacting with computers* 15, 3 (2003), 289–308.
- [12] Minsuk Chang, Léonore V Guillaïn, Hyeungshik Jung, Vivian M Hare, Juho Kim, and Maneesh Agrawala. 2018. RecipeScape: An interactive tool for analyzing cooking instructions at scale. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–12.
- [13] 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*. 1–14.
- [14] Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2012. MixT: automatic generation of step-by-step mixed media tutorials. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 93–102.
- [15] Ioana Croitoru, Simion-Vlad Bogolin, Samuel Albanie, Yang Liu, Zhaowen Wang, Seunghyun Yoon, Franck Deroncourt, Hailin Jin, and Trung Bui. 2023. Moment Detection in Long Tutorial Videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2594–2604.
- [16] Hai Dang, Karim Benharrak, Florian Lehmann, and Daniel Buschek. 2022. Beyond Text Generation: Supporting Writers with Continuous Automatic Text Summaries. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–13.
- [17] Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* 22 (2004), 457–479.
- [18] Hugging Face. 2022. *Hugging Face Transformers: OWL-ViT*. Retrieved December 22, 2022 from https://huggingface.co/docs/transformers/model_doc/owlvit
- [19] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. 2021. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems* 34 (2021), 26183–26197.
- [20] 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*. 1–12.
- [21] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*. 5267–5275.
- [22] Katy Ilonka Gero, Vivian Liu, and Lydia Chilton. 2022. Sparks: Inspiration for science writing using language models. In *Designing Interactive Systems Conference*. 1002–1019.
- [23] Steven M Goodman, Erin Buehler, Patrick Clary, Andy Coenen, Aaron Donsbach, Tiffanie N Horne, Michal Lahav, Robert MacDonald, Rain Breaw Michaels, Ajit Narayanan, et al. 2022. LaMPoSt: Design and Evaluation of an AI-assisted Email Writing Prototype for Adults with Dyslexia. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–18.
- [24] Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356* (2022).
- [25] Matthew Guzdial, Nicholas Liao, Jonathan Chen, Shao-Yu Chen, Shukan Shah, Vishwa Shah, Joshua Reno, Gillian Smith, and Mark O Riedl. 2019. Friend, collaborator, student, manager: How design of an ai-driven game level editor affects creators. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.
- [26] Bo He, Jun Wang, Jieli Qiu, Trung Bui, Abhinav Shrivastava, and Zhaowen Wang. 2023. Align and Attend: Multimodal Summarization with Dual Contrastive Losses. *arXiv preprint arXiv:2303.07284* (2023).
- [27] Jane Hoffswell and Zhicheng Liu. 2019. Interactive repair of tables extracted from pdf documents on mobile devices. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [28] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 159–166.
- [29] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. 2016. Denscap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4565–4574.
- [30] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. MDETR-modulated detection for end-to-end multimodal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1780–1790.
- [31] Juho Kim, Philip J Guo, Carrie J Cai, Shang-Wen Li, Krzysztof Z Gajos, and Robert C Miller. 2014. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 563–572.
- [32] 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*. 4017–4026.
- [33] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision* 123, 1 (2017), 32–73.
- [34] Md Tahmid Rahman Laskar, Xue-Yong Fu, Cheng Chen, and Shashi Bhushan TN. 2023. Building Real-World Meeting Summarization Systems using Large Language Models: A Practical Perspective. *arXiv preprint arXiv:2310.19233* (2023).
- [35] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).
- [36] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiyu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. 2022. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10965–10975.
- [37] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.
- [38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.
- [39] Ching Liu, Juho Kim, and Hao-Chuan Wang. 2018. ConceptScape: Collaborative concept mapping for video learning. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–12.
- [40] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. GPTeval: NLG Evaluation using GPT-4 with Better Human Alignment. *arXiv preprint arXiv:2303.16634* (2023).
- [41] Ryan Louie, Andy Coenen, Cheng Zhi Huang, Michael Terry, and Carrie J Cai. 2020. Novice-AI music co-creation via AI-steering tools for deep generative models. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–13.
- [42] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786* (2021).
- [43] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. *arXiv preprint arXiv:2306.05424* (2023).
- [44] Dotdash Meredith. 2023. *Allrecipes*. <https://www.allrecipes.com/>
- [45] Midjourney. 2021. *Midjourney*. Retrieved December 19, 2022 from <https://www.midjourney.com/>
- [46] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.
- [47] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. 2022. Simple Open-Vocabulary Object Detection with Vision Transformers. *arXiv preprint arXiv:2205.06230* (2022).
- [48] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. 2021. Clip-it! language-guided video summarization. *Advances in Neural Information Processing Systems* 34 (2021), 13988–14000.
- [49] Megha Nawhal, Jacqueline B Lang, Greg Mori, and Parmit K Chilana. 2019. VideoWhiz: Non-Linear Interactive Overviews for Recipe Videos. In *Graphics Interface*. 15–1.
- [50] OpenAI. 2022. *Introducing ChatGPT*. <https://openai.com/blog/chatgpt>
- [51] OpenAI. 2023. *GPT-4V(ision) System Card*. <https://openai.com/research/gpt-4v-system-card>
- [52] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155* (2022).
- [53] Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala. 2014. Video digests: a browsable, skimmable format for informational lecture videos. In *UIST*, Vol. 10. Citeseer, 2642918–2647400.
- [54] Sarah Perez. 2020. *YouTube introduces Video Chapters to make it easier to navigate longer videos*. Retrieved October 18, 2022 from <https://techcrunch.com/2020/05/28/youtube-introduces-video-chapters->

- to-make-it-easier-to-navigate-through-longer-videos/?gucounter=1
- [55] Learn Prompting. 2023. *Your Guide to Communicating with Artificial Intelligence*. Retrieved November 14, 2023 from <https://learnprompting.org/>
- [56] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.
- [57] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*. PMLR, 8821–8831.
- [58] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).
- [59] Cristian Rodriguez-Opazo, Edison Marrese-Taylor, Basura Fernando, Hongdong Li, and Stephen Gould. 2021. DORI: discovering object relationships for moment localization of a natural language query in a video. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1079–1088.
- [60] Chuyi Shang, Emi Tran, Medhini Narasimhan, Sanjay Subramanian, Dan Klein, and Trevor Darrell. 2023. LUSE: Using LLMs for Unsupervised Step Extraction in Instructional Videos. <https://cveu.github.io/2023/papers/36.pdf> (2023).
- [61] Alison Smith, Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater. 2018. Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system. In *23rd International Conference on Intelligent User Interfaces*. 293–304.
- [62] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. 2015. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5179–5187.
- [63] Tomáš Souček and Jakub Lokoč. 2020. Transnet V2: an effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838* (2020).
- [64] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lambda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239* (2022).
- [65] 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*. 1–16.
- [66] Sylvaine Tuncer, Barry Brown, and Oskar Lindwall. 2020. On pause: How online instructional videos are used to achieve practical tasks. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [67] Bryan Wang, Meng Yu Yang, and Tovi Grossman. 2021. Soloist: Generating mixed-initiative tutorials from existing guitar instructional videos through audio processing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [68] Cheng-Yao Wang, Wei-Chen Chu, Hou-Ren Chen, Chun-Yen Hsu, and Mike Y Chen. 2014. Evertutor: Automatically creating interactive guided tutorials on smartphones by user demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 4027–4036.
- [69] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. 2021. End-to-end dense video captioning with parallel decoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6847–6857.
- [70] Sarah Weir, Juho Kim, Krzysztof Z Gajos, and Robert C Miller. 2015. Learnersourcing subgoal labels for how-to videos. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*. 405–416.
- [71] wikihow. 2023. *Welcome to wikiHow, the most trusted how-to site on the internet*. <https://www.wikihow.com>
- [72] Wikipedia contributors. 2023. Wizard (software) – Wikipedia, The Free Encyclopedia. [https://en.wikipedia.org/w/index.php?title=Wizard_\(software\)&oldid=1182151261](https://en.wikipedia.org/w/index.php?title=Wizard_(software)&oldid=1182151261) [Online; accessed 21-November-2023].
- [73] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–22.
- [74] Saelyne Yang, Sangkyung Kwak, Tae Soo Kim, and Juho Kim. 2022. Improving Video Interfaces by Presenting Informational Units of Videos. *CHI'22 Extended Abstracts. Association for Computing Machinery* (2022).
- [75] Xianjun Yang, Yan Li, Xinlu Zhang, Haifeng Chen, and Wei Cheng. 2023. Exploring the limits of chatgpt for query or aspect-based text summarization. *arXiv preprint arXiv:2302.08081* (2023).
- [76] YouTube. 2023. *YouTube*. <https://www.youtube.com/>
- [77] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. 2019. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1247–1257.
- [78] Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858* (2023).
- [79] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848* (2023).
- [80] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lema Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren’s Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *arXiv preprint arXiv:2309.01219* (2023).
- [81] 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 *CHI Conference on Human Factors in Computing Systems*. 1–11.
- [82] Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [83] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8739–8748.
- [84] Wencheng Zhu, Jiwen Lu, Jiahao Li, and Jie Zhou. 2020. Dsnet: A flexible detect-to-summarize network for video summarization. *IEEE Transactions on Image Processing* 30 (2020), 948–962.
- [85] Wanrong Zhu, Bo Pang, Ashish V Thapliyal, William Yang Wang, and Radu Soricut. 2022. End-to-end dense video captioning as sequence generation. *arXiv preprint arXiv:2204.08121* (2022).