ABSTRACT

Title of Dissertation: Bridging the Gulf of Evaluation in Human-AI Interaction for Knowledge Workers

Yuexi Chen
Doctor of Philosophy, 2025

Dissertation Directed by: Professor Zhicheng Liu
Department of Computer Science

For over half a century, user interfaces have served as the primary medium through which humans interact with software systems. To describe this interaction, researchers [1, 2] introduced a seven-stage action model encompassing goal formation, intention, action specification, execution, system response, interpretation, and evaluation. Central to this model are two critical challenges—referred to as the Gulf of Execution and the Gulf of Evaluation. The Gulf of Execution represents the gap between a user's goal and the means available to achieve it within the system, while the Gulf of Evaluation describes the gap between the system's perceived state and the user's goals.

The emergence of AI-powered interfaces has reshaped this interaction landscape. Unlike traditional deterministic systems, AI-powered interfaces often exhibit dynamic and unpredictable behaviors [3], prompting a re-examination of these Gulfs. On the one hand, the Gulf of Execution has narrowed: users can now articulate goals through natural language commands, leveraging large language model (LLM)-powered applications, rather than manually navigating complex

menus [4, 5]. On the other hand, the Gulf of Evaluation has widened: AI-generated outputs can be inaccurate or untrustworthy. For example, object detectors may incorrectly classify pedestrians on the road [6, 7], and LLMs have both intrinsic (contradicted by the source) and extrinsic (not supported by the source) hallucinations [8, 9].

The burden of evaluation largely falls on human knowledge workers, who apply knowledge to engage in non-routine problem solving and develop products and services [10]. Recent studies on knowledge workers have shown a wide adoption of AI and the practice of evaluating AI-generated results before use [11, 12]. Therefore, it's important to develop human-centered AI systems to bridge the Gulf of Evaluation for knowledge workers. To bridge the Gulf of Evaluation in the above two dimensions-inaccuracies and lack of trust-we must re-imagine human-centered AI systems beyond chatbots and design novel human-AI interactions. Therefore, the overarching research question of the thesis is: *How do we design human-centered AI systems to bridge the Gulf of Evaluation in human-AI interaction for knowledge workers?*

The thesis addresses this research question by introducing techniques to reduce inaccuracies and foster trust for representative knowledge workers. More specifically, to reduce inaccuracies, we developed TutoAI [13], a cross-domain framework for AI-assisted mixed-media tutorial creation on physical tasks. We present an approach to identifying, assembling, and evaluating AI models for creating mixed-media tutorials from instructional videos, along with an interface for creators to refine AI-generated components. To enhance trust, we developed an AI-assisted visual analytics tool called COALA [14] for a multilingual collaborative writing dataset. We contribute several interpretable techniques, including interactive clustering, textual pattern explanations and dedicated data visualizations to foster trust among communication researchers. To thoroughly evaluate machine learning models and build trust before deploying them in high-risk

applications, we developed SAFEGUARD AI for safety experts —a visual analytics tool powered by AI agents that reveals model inaccuracies and ensures regulatory compliance. Collectively, these systems highlight how human-centered techniques can effectively bridge the Gulf of Evaluation for diverse knowledge workers.

Bridging the Gulf of Evaluation in Human-AI Interaction for Knowledge Workers

by

Yuexi Chen

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2025

Advisory Committee:
Professor Zhicheng Liu, Chair/Advisor
Professor David Weintrop, Dean's representative
Professor Hal Daumé III, Departmental representative
Professor Fumeng Yang
Professor Ruohan Gao

Dedication

To all human knowledge workers.

“Le vent se lève. Il faut tenter de vivre.”

*—Paul Valéry, *Le Cimetière Marin**

Acknowledgments

Zeroth, I would like to thank the organizers of the HCIL Weekly Brown Bag Speaker Series. I initially attended those HCI talks just for the free pizza. But as the saying goes, if something's free, you're the product—those talks eventually converted me into an HCI researcher.

First, I'd like to thank my advisor, Prof. Leo Zhicheng Liu. I feel incredibly fortunate to have an advisor who consistently provides thoughtful feedback, is always responsive, and pays close attention to detail. Without his mentorship and support, I wouldn't have had the opportunity to intern at Adobe or work on so many interesting projects. I'd also like to thank my committee members: Prof. Hal Daumé III, Prof. David Weintrop, Prof. Fumeng Yang, and Prof. Ruohan Gao. I deeply appreciate their constructive feedback on my thesis. A special thanks to Hal, who retweeted my work to his tens of thousands of followers, and also to Fumeng, who provided detailed input on my most recent project, Safeguard AI. To my labmates in the Human-Data Interaction group—Kazi Tasnim Zinat, Chen Chen, Hannah Bako, Sneha Gathani, and Xiaoyu Liu—thank you for fostering a culture of open discussion and mutual support, whether about research or beyond.

Next, I want to thank my collaborators beyond UMD CS. I'm grateful for our collaborators in the iSchool—Prof. Ge Gao, Yimin Xiao and Naomi Yamashita from NTT—whose fresh perspectives helped shape the COALA project [14, 15, 16]. Thanks also to Prof. Niklas Elmqvist for his contributions to DocDancer [17, 18]. At Adobe Research, I'm grateful to Vlad Morariu,

Anh Truong, and Chris Tensmeyer. Despite COVID turning internships remote, the experience of interning at a leading industry lab was deeply enriching, both professionally and personally. I still remember the thrill of stepping into Adobe's World Headquarters in San Jose during intern week—it made me dream of building a company just as impactful one day. If Adobe gave me a glimpse into industry research, then my onsite internship at Bosch Research was fully immersive. Thank you to Jorge Ono Piazano, Jiajing Guo, and Vikram Mohanty for an unforgettable summer in Sunnyvale.

I want to thank the people who've shaped my life beyond research. Thank you to Gary, Adam, Max, and the Mokhtarzada brothers for showing me the startup world. I'd also like to thank a fellow Terp, Tim Sweeney, though we've never met. I was devastated after my first paper was rejected, and I stumbled upon his poetic reflections following the *Epic Games v. Apple* loss [19]. It reminded me that even billionaires face setbacks, yet they persevere. Thanks to my tennis buddies—David, Julio, Qiming, Tianrui—and my coaches at JTCC, Iosua and Pei, and the USTA teams I've joined (NetGirlz, Juan, Happy Farm, Racqueteers, Title XIs). Also, shoutout to my gym buddies, Zeying and Le—sports have provided consistent returns in dopamine, hedging against the volatility of research outcomes.

Last but not least, I want to thank my parents, who always prioritized my education. I still remember the first English dictionary they bought me—a tiny, 500-word children's dictionary. Two decades later, as an overeducated Sichuanese, I was often mistaken for a native English speaker. Finally, I would like to thank my partner, Yuchen. It's a stroke of luck to meet someone in college and share so many chapters together. As our favorite growth mindset book says [20], we won't *live* happily ever after—we'll *work* happily ever after.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

Since the term Artificial Intelligence (AI) was coined in 1956 [27], the field has undergone a sea-change. It began with rule-based systems, where intelligence was manually encoded using logic and expert knowledge. This was followed by the rise of statistical and deep learning models, which learn patterns from data [28, 29, 30]. Today, we are in the era of generative AI, powered by large-scale transformer architectures [31, 32, 33].

Despite the evolution of AI, humans remain in the loop. For example, AARON—a rule-based drawing robot developed by Harold Cohen [34]—produced autonomous artwork based on hand-coded heuristics, and human artists could intervene by refining its output or updating its rules [34]. Today, this collaboration has become more fluid. In text-to-image systems like DALL·E[35] and Midjourney[36], users can co-create visual content by providing natural language prompts and iteratively refining the results.

Back in the 1980s, researchers [1, 2] proposed a seven-stage action model to describe the general process in human-computer interaction, consisting of goal formation, intention, action specification, system response, execution, interpretation, and evaluation. Two critical challenges arise from this process: the Gulf of Execution and the Gulf of Evaluation. The Gulf of Execution refers to the distance between users' goals and the means to change the system state to achieve

those goals; the Gulf of Evaluation is the distance between the system's perceived state and the user's actual goals.

The Gulfs have evolved in human-centered AI systems. The Gulf of Execution narrows in conversational interfaces. For example, once a goal is formed, prompt engineering (crafting instructions to optimize the AI output) could bridge the Gulf of Execution [4, 5, 37]. However, the Gulf of Evaluation has widened: AI-generated results can be inaccurate or untrustworthy, making them difficult to interpret and evaluate.

Inaccuracies. Despite being trained on large-scale datasets, Large language models (LLM) [32, 38] may return erroneous results, and object detectors in autonomous driving systems may incorrectly classify objects on the road [7]. More importantly, recent trends have shown a shift from single models to compound AI systems [39], which consist of multiple models interacting with one another. Therefore, the inaccuracies in individual models could lead to amplified errors when building a compound AI system.

Lack of trust. Since AI models are inherently imperfect, as previously discussed, users may be reluctant to trust AI-generated results. For instance, lawyers who are aware that ChatGPT can hallucinate legal cases may be less likely to trust it for future legal writing [40, 41]. Beyond model inaccuracies, trust issues can also arise from other sources. Studies have shown that patients tend to trust human doctors more than AI systems, particularly when it comes to diagnosing high-risk diseases [42]. Another factor contributing to the lack of trust is the Process Gulf [43]: the difference between a human's process to achieve a task and the AI's process, e.g., a diffusion model generates images by starting from random noise, which is drastically different from human painters, or a clustering algorithm may create a mathematically sophisticated representation yet

hard to interpret [22].

Recent studies on knowledge workers show broad adoption of AI and the practice of evaluating AI-generated results before usage [11, 12], indicating the burden of evaluation largely falls on human knowledge workers, who apply knowledge to engage in non-routine problem solving and develop products and services [10]. To bridge the Gulf of Evaluation in the above two dimensions-inaccuracies and lack of trust-we must re-imagine human-centered AI systems beyond chatbots and design novel systems. Therefore, the overarching research question of the thesis is:

How do we design human-centered AI systems to bridge the Gulf of Evaluation in human-AI interaction for knowledge workers?

1.2 Methodology

There are diverse types of knowledge workers, and in this thesis, we selected three representative knowledge workers: content creators, communication researchers, and safety experts. We selected these groups not only because they each face unique challenges related to the Gulf of Evaluation, but also because of the availability of collaborators through our industry partnerships.

For example, in collaboration with Adobe, we developed TUTOAI [13], a mixed-media tutorial creation tool. For tutorial creators, the Gulf of Evaluation stems from the inaccuracies of machine learning (ML) models in generating mixed-media tutorials. In collaboration with communication researchers from both UMD and NTT, we developed COALA [14], an AI-assisted collaborative writing behavior analysis tool. For communication researchers, the Gulf of Evaluation is rooted in a lack of trust in the insights returned by ML models. In collaboration with

Bosch, we developed SAFEGUARD AI, an agent-powered model validation and regulation compliance tool designed for safety experts. In this case, the Gulf of Evaluation is due to both model inaccuracies and a lack of trust, as safety experts must thoroughly understand a model's limitations before deploying it in high-risk scenarios such as city driving.

Although each project targets a different dimension of the Gulf of Evaluation, they all address real-world applications and share common methodologies, including human-computer interaction, machine learning, data visualization, and user interface/user experience (UI/UX) design. Here is an overview of the research methods used in this thesis.

Requirement validation. The first stage is to validate the need to develop a new system. Common methods include user interviews and literature surveys. For example, in COALA, we interviewed two communication researchers about their data analysis goals, and surveyed existing data visualizations. We identified that existing systems do not satisfy their needs, prompting us to develop a novel visual analytics system.

UI/UX mockups. After confirming the users' needs, we must design the UI/UX. Before writing any code, we draw sketches in Microsoft PowerPoint and wireframes in prototyping tools like Figma [44]. For example, in TUTOAI, we devised multiple UI mockups and discussed them with several graphic designers at Adobe Inc.

Machine learning. After figuring out the features required in the system, we need to identify applicable machine learning modules based on the input. For example, in TUTOAI, we need to create mixed-media tutorials from instructional videos, so machine learning models that take videos as input are applicable. Since we do not focus on training machine learning models from scratch, we usually select pre-trained models based on public rankings [45, 46].

Software development. The next step is to integrate machine learning modules with an interactive user interface. If users only need to interact with the output, then we can store the output and load it to the front end, which could be built by libraries like React.js [47]. If complex data visualizations are involved, we need to use specialized libraries like D3.js [48]. If users need to interact with the machine learning models, we must also set up a backend, e.g., Flask [49].

User studies. After we finish the software development, we need to conduct user studies. First, we apply for Institutional Review Board (IRB) approval to ensure our study is not harmful. Then, we send out advertisements to recruit eligible users, prepare tutorials for the system we developed, and conduct the user study. After the study, besides quantitative surveys, we also have semi-structured interview questions to solicit more feedback. Optionally, we may conduct a statistical analysis of the user study results.

1.3 Contribution

My contributions lie in two aspects: novel systems that bridge the Gulf of Evaluation and findings from empirical studies.

Novel systems that bridge the Gulf of Evaluation. I contributed three novel human-centered AI systems, TutoAI, COALA, and Safeguard AI.

- TutoAI is a mixed-media tutorial creation tool developed for tutorial creators in collaboration with Adobe. The Gulf of Evaluation lies in the inaccuracies of machine learning (ML) models in generating mixed-media tutorials, and TutoAI reduces these inaccuracies through an ML pipeline and an interactive interface that allows users to refine AI-generated components.

- COALA is an AI-assisted collaborative writing behavior analysis tool developed for communication researchers in collaboration with NTT. The Gulf of Evaluation lies in a lack of trust in the insights returned by ML models, especially those involving pattern mining and clustering methods. COALA enhances trust through textual pattern explanations, arc diagrams, model confidence display, and an interface for interactive clustering.

- SAFEGUARD AI is an agent-powered model validation and regulation compliance tool developed for safety experts in collaboration with Bosch. The Gulf of Evaluation lies in both model inaccuracies and a lack of trust. SAFEGUARD AI includes an agent-based pipeline to auto-label datasets based on regulations, as well as a slice-discovery algorithm to identify scenarios where ML models do not perform well. This algorithm links model performance to regulations, providing an overview of risk analysis for deploying models in these scenarios. SAFEGUARD AI is also equipped with an interface for interactive regulation compliance analysis.

Findings from empirical studies. This thesis contributes two types of findings from empirical studies: usability evidence and implications for future system development.

- **Usability evidence.** Though we focus on the Gulf of Evaluation, good usability of our systems ensures that we maintain a narrow Gulf of Execution. For example, tutorial creators successfully create mixed-media tutorials using TUTOAI, and communication researchers use COALA to discover insights from the multilingual collaborative dataset easily.

- **Implications for future system development.** During our user studies, the insights could inform future system developers. For example, in the study of COALA, we found that

non-native English speakers frequently switch between translation and writing, resulting in frequent context changes. This finding suggests that future AI writing assistant developers introduce features that reduce context switches for non-native speakers. In TUTOAI, though the landscape of machine learning models changes rapidly, we will not be surprised if specific ML components used in our system become obsolete. However, we believe there are design guidelines that transcend particular models. We share design guidelines for evaluating and assembling different ML models to facilitate complex knowledge tasks, as well as design lessons we learned while working with specific user groups.

1.4 Organizations

CHAPTER 2 To bridge the inaccuracy dimension of the Gulf of Evaluation, we developed TUTOAI, a cross-domain framework for AI-assisted mixed-media tutorial creation on physical tasks. To reduce model inaccuracies, we present an approach for identifying, assembling, and evaluating AI models for creating mixed-media tutorials from instructional videos, along with an interface for creators to refine AI-generated components. We demonstrated that TutoAI has performed well in cross-domain mixed-media tutorials, as evidenced by user studies, thereby bridging the Gap of model inaccuracies.

CHAPTER 3 To address the lack of trust dimension in the Gulf of Evaluation, we collaborated with communication researchers to develop an AI-assisted visual analytics tool called COALA for a multilingual collaborative writing dataset. We identified several models that could be used to analyze the dataset. To enhance users' trust, we designed multiple data visualizations and provided explanations for the patterns they reveal. Additionally, we have developed an

interface that allows users to conduct interactive clustering.

CHAPTER 4 To deploy ML models into production, we must demonstrate regulation compliance. The Gulf of Evaluation lies in both model inaccuracies and a lack of trust. To thoroughly reveal model inaccuracies and enhance trust, SAFEGUARD AI includes an agent-based pipeline to auto-label datasets based on regulations, and a slice-discovery algorithm to identify scenarios where ML models do not perform well. We also contributed an interactive visual analytics tool for safety experts to investigate the regulation compliance of models.

CHAPTER 5 Collectively, these systems highlight how human-centered techniques can effectively bridge the Gulf of Evaluation for knowledge workers during human-AI interaction by reducing inaccuracies and enhancing trust. We also outline future directions, including extending support to a broader range of knowledge workers and developing new methods for supervising interactions among multiple AI agents.

Chapter 2: TutoAI

2.1 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 [50, 51].

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 [52] 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 [53, 54, 55]. 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 [53, 54, 56, 57]. While many have acknowledged the impor-

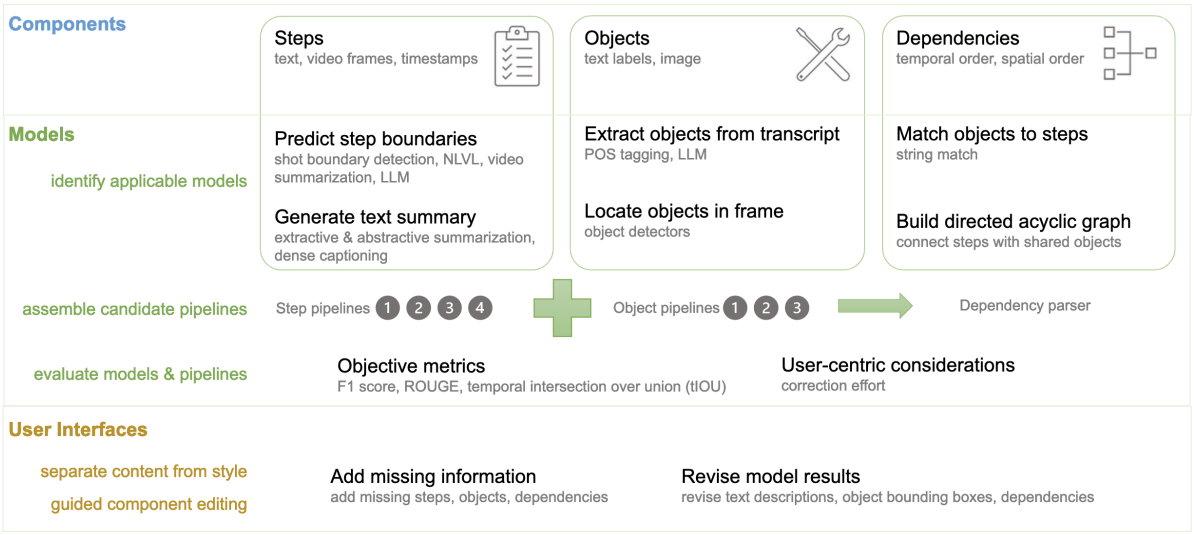


Figure 2.1: TutoAI is a framework for AI-assisted mixed-media tutorial creation. It has three levels: components, models, and user interfaces. After identifying components of common mixed-media tutorials, TutoAI assembles and evaluates relevant computational models to extract components. Then, it presents the results on a user interface for creators to review and edit.

tance of generalization and argued how their approaches could apply to tutorials in other domains [54, 58, 59, 60], 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 [61, 62].

Recent advances in AI, especially large language models (LLM) [32], 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 2.1). 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 [52]), 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.

2.2 Related work

2.2.1 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 [52, 56, 58, 59, 63]. A step could range from a cooking procedure [64] to a software operation [56]. Another common component is objects, e.g., ingredients and equipment for cooking tutorials [53, 55, 65]. Besides steps and objects, some tutorials also organize steps based on dependencies, e.g., Truong et al. [54] grouped makeup video segments by facial parts in a two-level hierarchical format; Nawhal et al. [53] and Yang et al. [55] 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 Wiki-

How [66] and Allrecipes [65] depend on experts to draft tutorials; Crowdy [59] requires learners to identify subgoals and steps. In certain domains, automatic extraction methods are feasible. MixT [57] segments PhotoShop videos using software logs. Fraser et al. [56] implement a dynamic programming method to segment creative stream videos based on the transcript and software logs; Truong et al. [54] 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 [58] gathers steps from crowdworkers and converges them through clustering algorithms. EverTutor [67] converts smartphone demonstrations by humans into interactive tutorials. Humans could also refine computational results, e.g., VideoWhiz [53] and RecipeDeck [64] both employ Part-of-Speech (POS) tagging to detect cooking actions and objects and then rely on annotators to refine the results. Video Digests [63] 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 [54]; Soloist [60] 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 [58]; Crowdy [59] 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.

2.2.2 AI-assisted creation

AI has augmented human creativity, from generating visuals [35, 36] to crafting slogans and aiding scientific writing [32, 68]. 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 [69] allows users to adjust the mood of AI-generated music notes. Morai Maker [70] is a game-level editor in which human and AI designers take turns to build a Super Mario Bros game. LaMPost [71] facilitates email writing for people with Dyslexia. Dang et al.'s text editor [72] 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 [73]; repair of auto-extracted PDF tables [74]; refinement of medical images retrieved by ML models [75]. 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 [3, 76], 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.

2.2.3 Large language models (LLM) prompting

Large language models (LLM) [32, 77, 78], trained on internet-scale 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* [79]. The most commonly used prompting technique is zero-shot prompting [80], which describes the task directly. There are also other prompting techniques, including few-shot prompting [81] and prompt chaining [5]. Researchers have applied zero-shot prompting to summarize various types of data, including news [82, 83], Reddit posts [84], meeting records [85] and stories [84]. Researchers have also applied LLMs to summarize video transcripts. Croitoru et al. [86] applied GPT-3 to summarize software tutorial video transcripts and then used the summary to detect key moments. LUSE [87] 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 [88], 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 [87], descriptivity [87], coherence [83], faithfulness [83], relevance [83] and personal preferences [82].

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. [86] and LUSE [87], 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) [89], involving

human refinement is crucial for end users. Similar to previous research, we manually evaluated the output besides ROUGE scores.

2.3 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.

2.3.1 Level 1: Components

As shown in Figure 2.1, 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 [52] only utilize *steps*. For completeness (**D2**), we discuss all three component types in level 2 and level 3.

2.3.2 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).

2.3.2.1 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.

2.3.2.2 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.

2.3.2.3 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 FNs requires creation.

2.3.3 Level 3: User Interface (UI) design

AI-generated results are typically imperfect, requiring further refinement from humans. As shown in Figure 2.1, 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 2.6 discusses UI design guidelines and presents an example implementation.

2.4 Level 1: Components in mixed-media tutorials

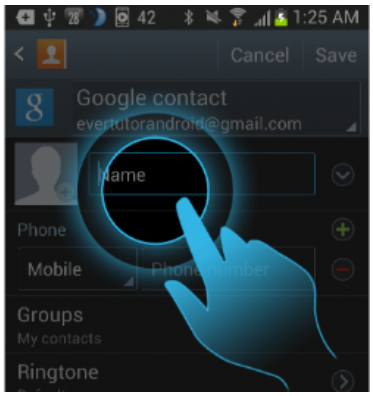
To explore the design space of cross-domain mixed-media tutorials, we analyzed 13 mixed-media tutorials from three websites [65, 66, 90] and 10 research papers [53, 54, 55, 56, 57, 58, 63, 64, 67, 91], covering at least five domains including cooking, makeup, vehicle repair, software usage, and educational 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).

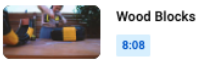
2.4.1 Step

Every tutorial comprises a sequence of steps, e.g., "duplicating a layer" in a PhotoShop tutorial [57]. 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 2.2 provides step examples in mixed-media tutorials. Specifically, Figure 2.2a shows a step in an interactive smartphone tutorial, marked by an overlay indicating the screen area to be clicked [67]; Figure 2.2b depicts an auto-generated YouTube Chapter for a DIY craft video fea-



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



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

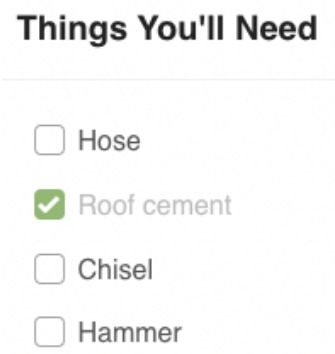
(c) A step with text and glyph in a cooking tutorial [12].

Figure 2.2: Examples of steps in mixed-media tutorials (images used with permission).

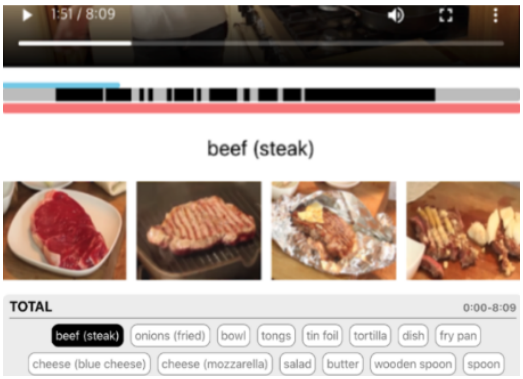
turing text, images, and video clips (with timestamps); Figure 2.2c illustrates a step in a cooking tutorial, where red and blue dots signify ingredients and actions, respectively [64]. Comprehensive details are in the Appendix.

2.4.2 Object

Many mixed-media tutorials explicitly specify objects required for the task, such as ingredients and equipment in cooking tutorials [55], and UI widgets in software tutorials [56]. 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. Additionally, 3 offered interaction features, including checkboxes or clickable buttons that link objects with other components.



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



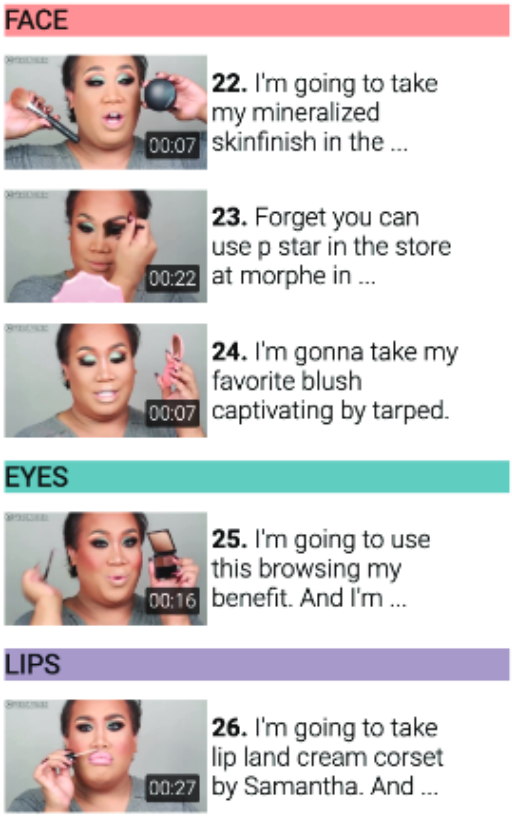
(b) Object components represented using text, images, and appearance time in a cooking tutorial [75]

Figure 2.3: Examples of objects in mixed-media tutorials (images used with permission).

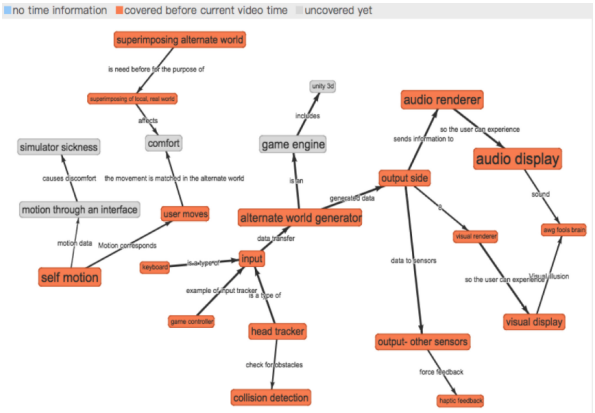
Figure 2.3 illustrates examples of objects in mixed-media tutorials. Figure 2.3a displays an object component from a roof repair tutorial on WikiHow [92], with interactive checkboxes to help users gather things needed; Figure 2.3b shows object buttons; clicking on an object button (e.g., "beef (steak)") brings up video frames containing that object and the appearance time on the timeline [55]. All the examples are in the Appendix.

2.4.3 Dependency

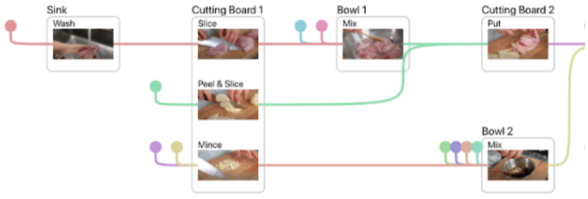
Dependencies between steps are everywhere; they could be food processing order in recipe tutorials [53, 55, 64], concept prerequisites in lectures [91] and facial parts in makeup tutori-



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



(b) Concept prerequisites in a lecture [39]



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

Figure 2.4: Dependency examples in mixed-media tutorials (images used with permission).

als [54]. 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 [59]. 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 2.4 shows dependency examples. Figure 2.4a shows groupings in a makeup tutorial where steps within each group are sequential but independent of other groups. Figure 2.4b maps out the dependencies of concepts in a lecture: orange nodes are already covered, and gray nodes are not. Figure 2.4c 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.

2.5 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.

2.5.1 Applicable models and candidate pipelines

2.5.1.1 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 [93, 94, 95, 96]. 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 [97, 98, 99].

- **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 [100, 101, 102]. 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 [103, 104]. Video summarization condenses a long video by selecting and stitching together keyframes to form a shorter video [105, 106, 107, 108]. 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.

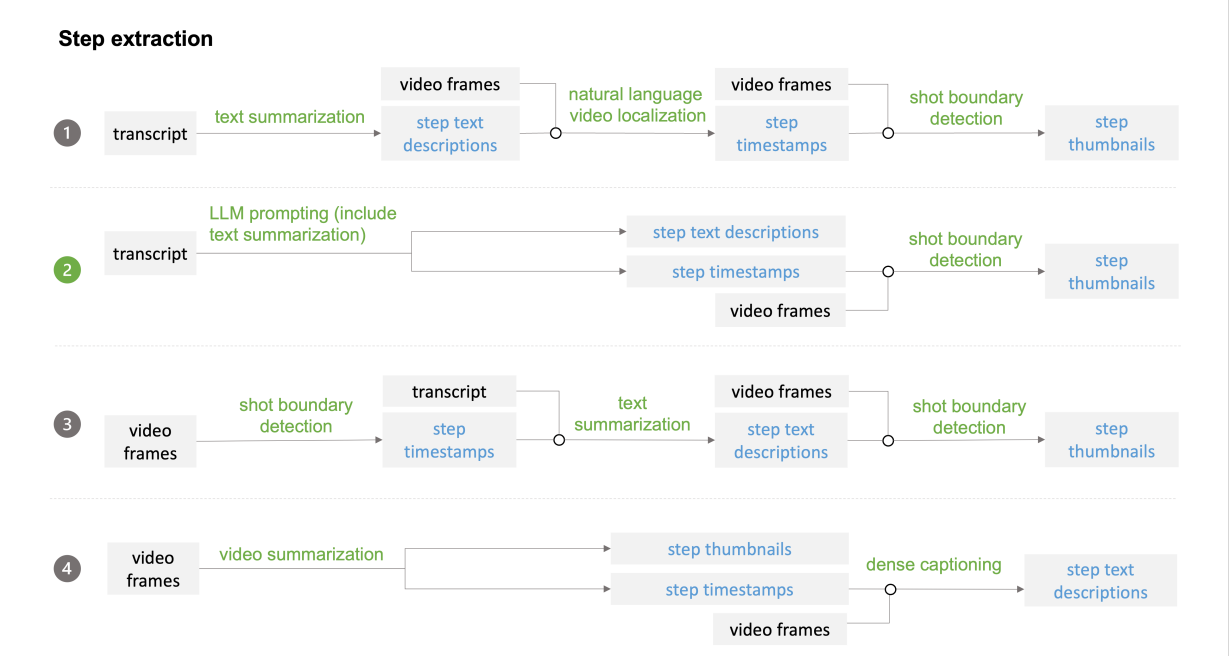


Figure 2.5: Four candidate pipelines for step extraction. Models are in green, and generated subcomponents are in blue. After evaluation, the chosen one is No.2.

- **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 2.5 shows 4 candidate pipelines.

- **Pipeline 1: text summarization + NLVL + shot boundary detection**. As shown in Figure 2.5, 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.

2.5.1.2 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 [109]. Obtaining object names from POS tagging results requires parsing nouns. LLMs can also be prompted to extract object names from text input. Tradi-

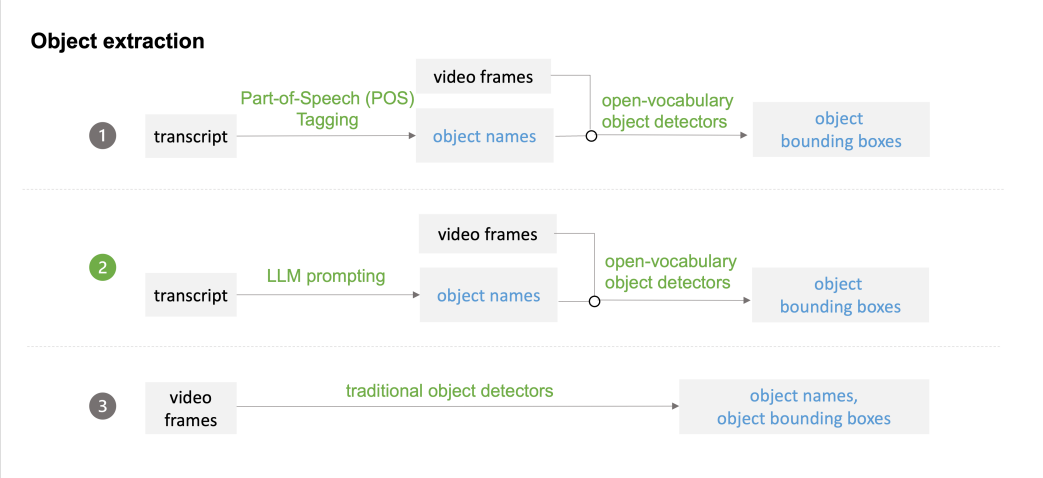


Figure 2.6: Three candidate pipelines for object extraction. Models are in green, and generated subcomponents are in blue. After evaluation, the chosen one is No.2.

tional object detectors are trained on predefined object categories and, given input images, output detection names and bounding boxes [110, 111].

- **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 [112, 113, 114].

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

- **Pipeline 1: POS Taggers + Open-vocabulary detectors**. As shown in Figure 2.6, 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.

2.5.2 Evaluation of applicable models and candidate pipelines

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

2.5.2.2 Evaluation dataset

We evaluated on the validation set of YouCook2 [104], containing 347 cooking videos with auto-generated English transcripts. Each video has human-annotated objects, step descriptions, and start/end times.

2.5.2.3 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 [94], TextRank [93]) and abstractive (rephrasing the original content, e.g., BART [95], T5 [96], GPT-3 [32]) methods. Among the five methods, GPT-3 leads by a large margin in ROUGE scores [88] (Appendix Table 5).

Traditional NLP metrics might not effectively gauge the quality of text generated by LLMs [115]. 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 [98] and PDVC [99] 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:

[1] https://www.youtube.com/watch?v=-k7trpuj3X8

NLVL method DORi [102] (Pipeline 1) , LLM prompting (GPT-3 [32]) (Pipeline 2) and shot boundary detector ProcNets [104] (Pipeline 3) .

For pipeline 1, we provided the video and ground truth step descriptions to DORi [102] 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 [104] 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.

2.5.2.4 Object pipeline evaluation

As shown in Figure 2.6, 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 [116] to extract object names. For Pipeline 2, we prompted GPT-3 [32, 38] 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 [117] trained on the Visual Genome dataset [21]. 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 [113] considering both performance and computational cost.

2.5.3 Final pipeline

We finalized our pipeline as shown in Figure 2.7, which includes Step pipeline 2 (Figure 2.5) and Object pipeline 2 (Figure 2.6) . First, we extract steps from video transcripts by prompting LLM (here we use GPT-3.5 [118], 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 [103] to pick thumbnails for each step. Next, to extract object

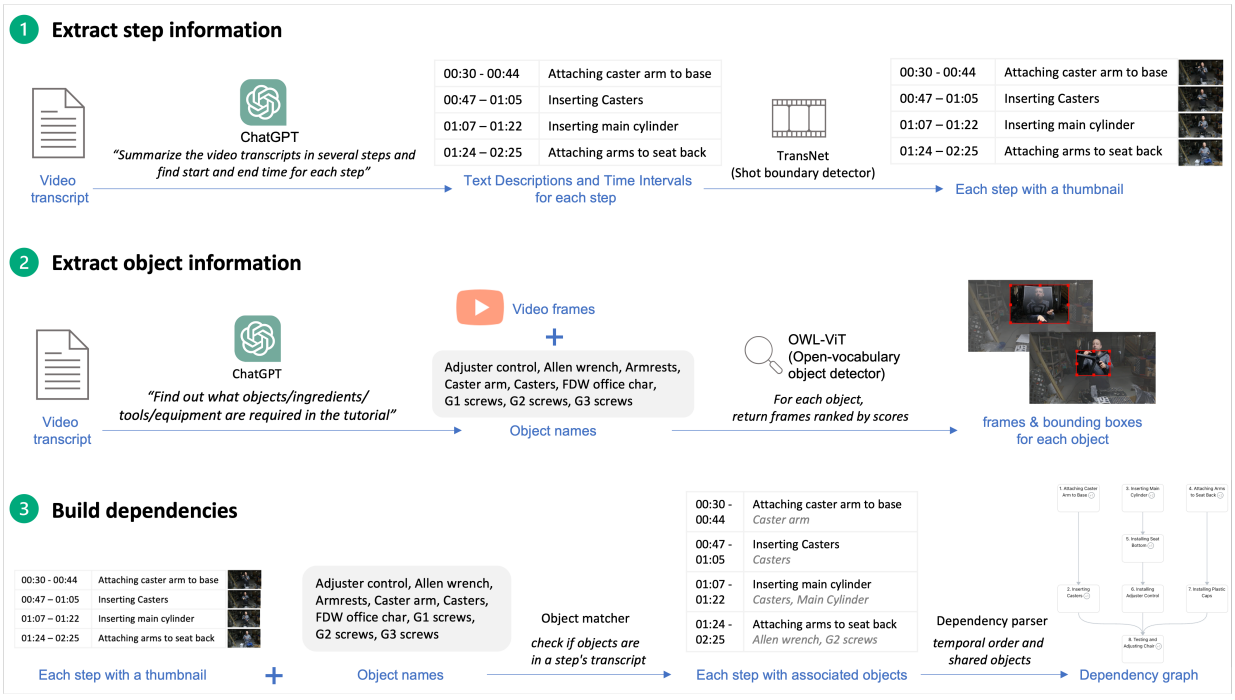


Figure 2.7: TutoAI's machine learning pipelines to obtain objects and steps in instructional videos: 1. extract steps: ChatGPT processes the video transcript to produce text descriptions and time intervals for each step, then a shot boundary detector augments each step with a thumbnail; 2. extract objects: 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.

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 [119] 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.

2.6 Level 3: User Interfaces for Mixed-Media Tutorial Creation

2.6.1 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.

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.

2.6.2 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 2.8. The tutorial contains the following components: a video player and step boundary below it (Figure 2.8A), an object list (Figure 2.8B) over which users can hover to see an image of the selected objects (Figure 2.8C); step overviews, which consist of a text description, a representative thumbnail and objects for each step (Figure 2.8D); associated dependencies (Figure 2.8E), 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

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

"view" button (Figure 2.9). Here is the workflow:

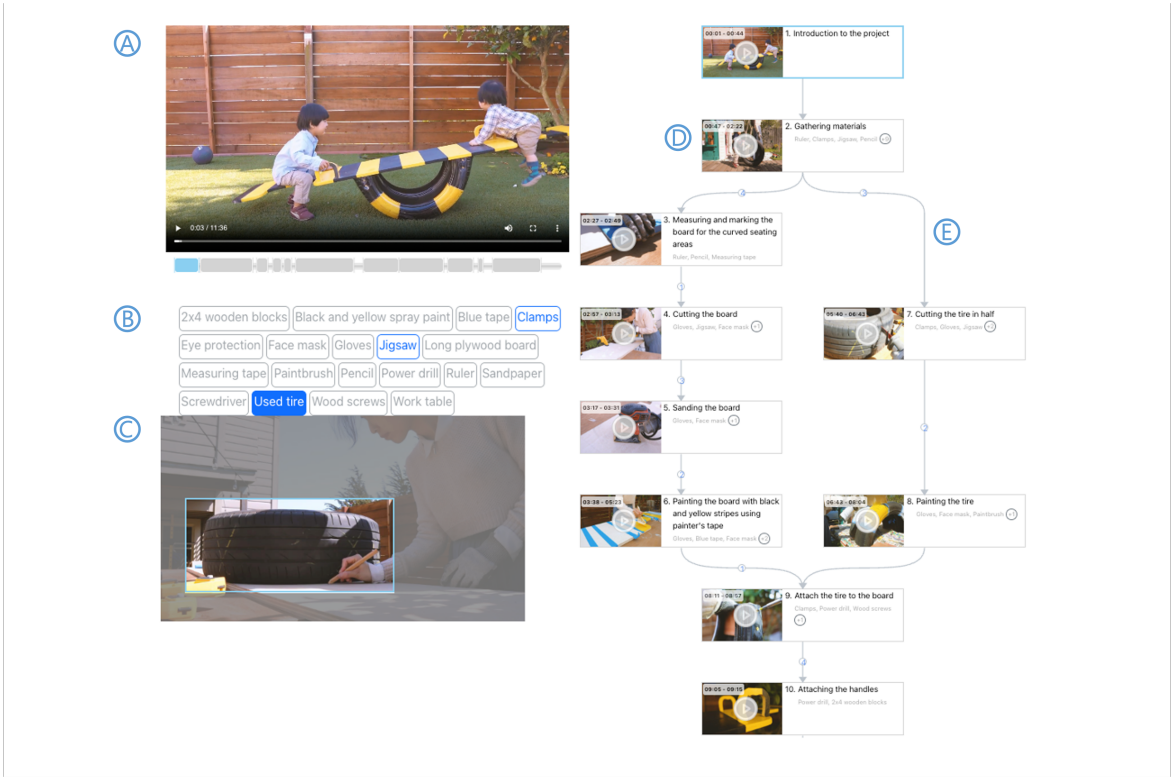


Figure 2.8: A mixed-media tutorial template on making a seesaw for kids: below the video player (A) is a list of required objects (B); hovering on the blue-bordered object will show the object's image along with a bounding box (C); on the right is an overview of steps, (D) each step is a video clip with start and end time, text descriptions and associated objects. (E) The arrows between the steps indicate the dependencies.

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 2.9); 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)**.

3) Select objects. The UI suggests an object list required for the tutorial and associates the

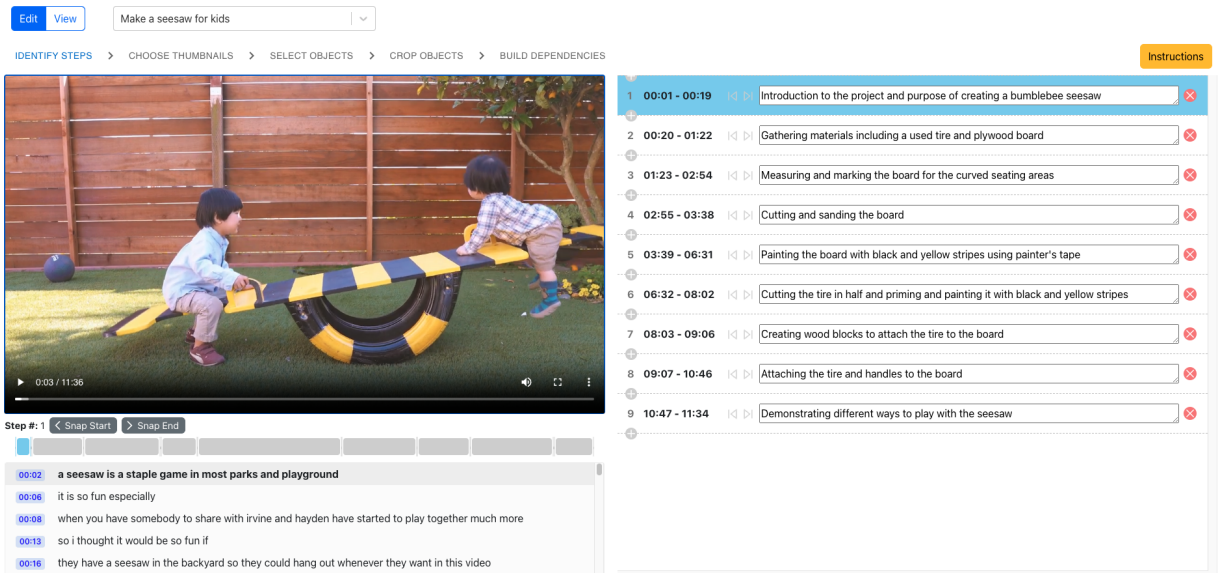


Figure 2.9: Identify steps. This task aims to break down the video into several steps and provide text descriptions and time boundaries for each step. On the left is a video player and its transcript ("Make a seesaw for kids"); on the right are the AI-generated steps.

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)**.

2.7 TutoAI framework evaluation - Model

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

2.7.1 Dataset

Inspired by the object-action quadrant for instructional videos [120], 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 2.1) 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.

2.7.2 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 2.1 column 8 ("F1"), our object extraction F1 scores fall between 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.

2.7.3 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 2.1 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 2.1 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.

2.8 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

			Objects					Steps				
Video ID	Domain	Duration (minutes)	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
lj7YK1lIRUM	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-GLI-Vsu9s	repair	11:38	11	12	1	0	0.96	10	10	0	0	0.33

Table 2.1: 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).

into creators' existing workflow.

2.8.1 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 [59]; 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.

In both studies, we used auto-generated YouTube Chapters [52] as the baseline. Although TutoAI was inspired by previous works, these tutorials were either generated automatically using a domain-specific approach [54, 56, 57, 67, 121] or manually without AI assistance [55, 91]. Mixed-initiative approaches [53, 58, 63, 64] do not provide comparable creation experience like TutoAI. We thus determined that YouTube Chapters [52] is the most reasonable baseline since they also support cross-domain generation of steps.

[2]https://support.google.com/youtube/answer/10332730
[3]https://blog.coursera.org/ready-for-retention-presenting-a-unified-note-taking-experience/

2.8.2 Study 1: general users

2.8.2.1 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).

2.8.2.2 Instructional videos:

We chose two instructional videos on YouTube: office chair assembly[4] and strawberry blueberry shortcakes[5]. 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).

2.8.2.3 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 chicken[6]

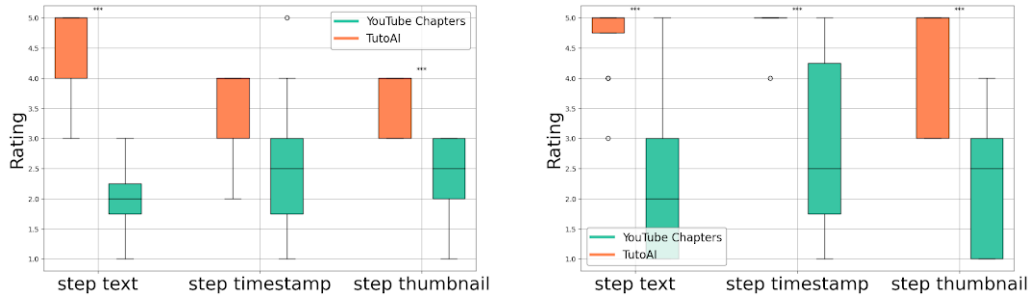
[4]https://youtu.be/OEIDupReh8Q
[5]https://youtu.be/BAp1AXn82Pg
[6]from the YouCook2 dataset: https://youtu.be/ntiGX3X-spA

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.

2.8.2.4 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 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 2.10a): TutoAI vs. YouTube Chapters, text: 4.6$\pm$0.65 vs. 2.0$\pm$0.71 (p=0.009); timestamps: 3.5$\pm$0.65 vs. 2.5$\pm$1.19 (p=0.075); thumbnails: 3.6$\pm$0.49 vs. 2.3$\pm$0.75 (p=0.021). For group B, the benefits of TutoAI are not statistically significant (Appendix Figure 17 (a)): TutoAI vs. YouTube Chapters, text: 4.4$\pm$0.64 vs. 3.6$\pm$1.04 (p=0.138); timestamps: 3.3$\pm$1.25 vs. 3.0$\pm$1.0 (p=1.000); thumbnails: 3.4$\pm$0.76 vs. 2.4$\pm$1.38 (p=0.138). Other scores of TutoAI components are in Appendix Figure



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

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 2.10b). 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."*.

2.8.3 Study 2: YouTubers

2.8.3.1 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.

2.8.3.2 Findings:

We asked them about the impression of AI-generated results and workflows in creating instructional videos.

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

[7]https://youtu.be/5nHD0vy9R5g, used with permission
[8]https://youtu.be/drDSY3ZZqnQ, used with permission

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).

2.9 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 [54, 58, 59]. 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.

2.9.1 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 [122, 123, 124], and dense video captioning models may improve rapidly by benefiting from large-scale pre-trained models [125]. 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 [102] returns higher tIOU (temporal intersection over union) than ProcNets [104] 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.

2.9.2 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 2.8 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 [126]. 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 for-

mats, 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.

2.9.3 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 2.6.1 and Section 2.9.2. 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 2.9.1. Though the component and model details may differ, the underlying approach remains the same.

2.9.4 Limitations and future work

Domain limitations. Though TutoAI is a cross-domain framework, it does not apply to all instructional videos. Chang et al. [120] 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 [107, 127] 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 ef-

fects, 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.

Chapter 3: COALA

3.1 Introduction

Non-native speakers (NNS) actively engage in collaborative writing across various contexts: international students write reports with classmates [128] and advisors [129]; Wikipedia contributors edit articles in different languages [130]; employees in multi-national corporations collaborate on project pages [131]. Previous research shows that when writing in a non-native language, NNS tend to produce shorter and less complex content compare to writing in their native language [132, 133, 134]. However, limited research has examined the collaborative writing processes involving NNS and how they write differently from native speakers (NS) [15]. Such knowledge will be insightful in enhancing collaboration outcomes and fostering inclusive team dynamics involving NNS.

To bridge this gap, in collaboration with communication researchers, we collected document history and screen recordings of 162 collaborative writing sessions from 27 teams. We are interested in *comparing* authors' behaviors across different linguistic backgrounds and writing stages. For example, what are the common behavior patterns of NS and NNS, respectively? How do they differ in the early and late stages of collaborative writing?

However, existing document visualization and analytics tools fall short of supporting our analysis goal. Visualizations of document versions have long been used to investigate the dy-

namics of collaborative writing. For example, History Flow [135] uses a Sankey diagram to encode content contributions from different Wikipedia authors across different versions; Time Curves [136] project different document versions on a 2D curve based on their similarity and temporal order. However, such visualizations focus on the document content instead of the writers' behaviors during the writing process, such as browsing the internet or using translation tools.

Event sequence visualization tools are better suited for behavioral data but still do not effectively meet our needs. For example, TipoVis [137] allows comparisons between two sequences at a time, while our analysis requires comparing behavior sequences across multiple authors and sessions. CoCo [138] supports comparing event sequences belonging to different cohorts, but focuses primarily on aggregated metrics such as frequencies. In our case, we need to identify behavioral differences between NS and NNS at multiple levels of granularity with meaningful qualitative descriptions.

Furthermore, to effectively analyze such complex behavioral datasets, it is important to combine visualizations with automated methods such as data mining and clustering. Previous research highlights challenges related to the interpretability and trustworthiness of automated methods [139, 140, 141, 142]. As communication research experts, our collaborators possess contextual knowledge about their dataset and multilingual communication in general but are not familiar with the mechanisms of automated methods. Before incorporating the model-generated results into their analysis, they must interpret and trust them.

In comparing the behaviors of NS and NNS in collaborative writing, we thus face two challenges: 1) the limitations of existing text and event sequence visual analytics approaches, and 2) the lack of interpretability and trust in automatic data analytics. To address these challenges, we worked closely with the communication experts to formulate data models and task require-

ments, iterated on visualization designs to assess their applicability and scalability, and identified factors that might hinder interpretation and trust. Based on these design iterations, we make the following contributions:

- COALA, a visual analytics tool for comparing native and non-native speakers' behaviors in collaborative writing. COALA displays the uncertainties of multiple clustering results and supports interactive refinement of clusters while leveraging large language models (LLM) to generate cluster summaries.
- Empirical validation of COALA through a focus group session (N=2+2) and individual study sessions with researchers in related fields (N=8), where the participants used COALA to analyze behavioral differences between NS and NNS.
- Design lessons for developing interpretable visual analytics in the context of communication research and findings that inform future AI-assisted collaborative writing tools and collaborative processes beyond writing.

3.2 Related work

3.2.1 Collaborative writing studies

Collaborative writing has been a topic of interest since the 1980s [143, 144, 145]. Early research focused on awareness and coordination during collaborative writing [146], common writing tasks, and the number of collaborators [147]. The rise of online collaborative writing tools like Google Docs, Microsoft Word, and Overleaf [148] has made collaborative writing a common practice. For instance, Olson et al. [149] analyzed collaborative writing patterns in 96

college assignments in Google Docs and found that balanced participation and leadership would result in higher writing quality [149]. Researchers also explored various aspects of collaboration, such as impression management [150], reluctance to write closely [151], preference over edits with explanations [152], differences of tasks across writing stages [153], and territorial behaviors [154].

Few studies focus on authors' off-document writing-related activities during collaborative writing, for example, navigating multiple applications (e.g., Google Docs and Adobe InDesign) [155] and coordinating writing tasks on Wikipedia discussion pages [156]. Collaborated with communication researchers, we focus on writing-related behaviors, including off-document behaviors like using a translator and browsing the internet, aiming to provide a new perspective on collaborative writing analysis.

3.2.2 Text visual analytics

Several visual analytics approaches have been designed to analyze the evolution of documents in collaborative writing. Itero [157] is a revision history analytics tool based on Google Docs that visualizes character insertion patterns and user contributions. History Flow [135] and DocuViz [158] encode each author's contribution as a colored vertical line, with the height of the line proportional to the content length. The flow-like visualization reveals the cooperation and conflict among co-authors by connecting the same line across different versions. Graphs are also widely used, where authors are represented as nodes, and edges could be disagreement [159] or revert actions [160]. Time Curves [136] is a timeline visualization based on points' similarity, which could visualize different document versions. Other visualizations include branch-

based visualizations [161], revision maps [162] and color-coded words by authorship [163, 164]. Compared to text visual analytics approaches, COALA focuses on sequences of writing-related behaviors.

3.2.3 Non-native speakers vs. native speakers

Compared to native speakers (NS), non-native speakers (NNS) usually produce shorter and less complicated content and have difficulty transferring writing strategies from their mother tongue [132, 133, 134]. Though NNS need more help in the expression aspects [165], they may still contribute to the ideation aspect [166]. Cheng et al. [128] found in a case study that NS students had more power in collaborative writing at the beginning, but the NNS student developed academic literacy along the way, and overall the group writing experience has improved. NNS' writing could also be improved by receiving direct edit feedback at early versions [129, 167], or exposure to well-written model text by NS [168]. Compared to these previous case studies, we have a larger collaborative writing dataset of NS-NNS with video-recorded author behaviors, poised to reveal more patterns beyond anecdotal evidence.

3.2.4 Event sequence analysis

There are numerous methods to analyze event sequences. Besides *visualizing* sequences, we categorize analysis methods based on tasks: *comparing*, *clustering* and *summarizing*.

Visualizing event sequences. The most straightforward visualization design for event sequences is to arrange the events on a timeline [169, 170]. When the number of sequences is large, flow-based visualizations could show the trend of bundled sequences. For example, Sankey diagrams

represent each event as a node, the length of the node and the thickness of the links between nodes encode event frequencies [171, 172]. Tree-based visualizations encode the frequency of events as the thickness of edges [173, 174]. Like tree-based visualizations, icicle plots encode events as stacked rectangles, ordered from top to bottom, usually colored by event categories [175, 176]. When subsequences are highly repetitive, matrix-based visualizations could show the transition trend clearly [177, 178].

Comparing event sequences. Multiple tools focus on comparison. CoCo compares two patient cohorts via statistical analysis with built-in metrics [138] distilled from domain expertise [179]. TipoVis compares event sequences of social and communicative behaviors by overlaying two sequences [137]. COQUITO [180] assists users in defining cohorts with temporal constraints and comparing sequences by overlapped branches. Directly linking event sub-sequences for comparison is also common [181, 182, 183].

Clustering event sequences. Several interactive tools are designed for clustering sequences. For example, Wang et al. [184] built an unsupervised interactive clustering system to analyze large-scale clickstream data. EventThread [185] clusters event sequences by latent stage categories. Gotz et al. [186] group event sequences by dynamic hierarchical dimension aggregation. Sequen-C [187] adopts an align-score-simplify strategy to cluster sequences. VASABI [188] clusters user profiles by topic modeling and uses multi-dimensional distributions to characterize each cluster.

Summarizing event sequences. Numerous methods have been developed to find an overview of a cluster of sequences. Sequence Synopsis [22] constructs a high-level overview of sequences by balancing the minimum description length (MDL) principle and the information loss. Core-Flow [174] extracts branching patterns in temporal event sequences. Frequence [172] is a visual

analytics tool built on a frequent pattern mining algorithm that handles multiple levels of details and concurrency. SentenTree [173] summarizes unstructured social media text in a tree structure.

COALA is also equipped with visualizing, comparing, clustering, and summarizing features. Compared to existing approaches, we focus on interpretation and trust by including multiple clustering methods and displaying uncertainties, supporting multi-level granularity of sequences, and leveraging large language models to generate more intuitive descriptions [32].

3.3 Study background

3.3.1 Background

We collaborated with two communication researchers from a public university in the US. One is a professor who has studied multilingual communication for more than a decade, and the other is a Ph.D. advisee of the professor who also has rich experience in multilingual communication. They collected a dataset of collaborative writing between native and non-native speakers and contacted us for suggestions in visual analytics.

We conducted longitudinal co-design sessions with our collaborators to understand the communication research analysis better. We met weekly or bi-weekly for 30 weeks. After they introduced the study background and data, we initially adapted existing text visualizations like Time Curves [136] and History Flow [135] (see Appendix). Though such text visualizations provide a glimpse of how authors contribute to the document, and how co-authors revise or delete each others' contribution, they do not address the research questions to compare authors' behaviors, so we designed dedicated features for analyzing authors' behavioral sequences. During the process, we showed them visualizations and interface mockups, incorporated the feedback into

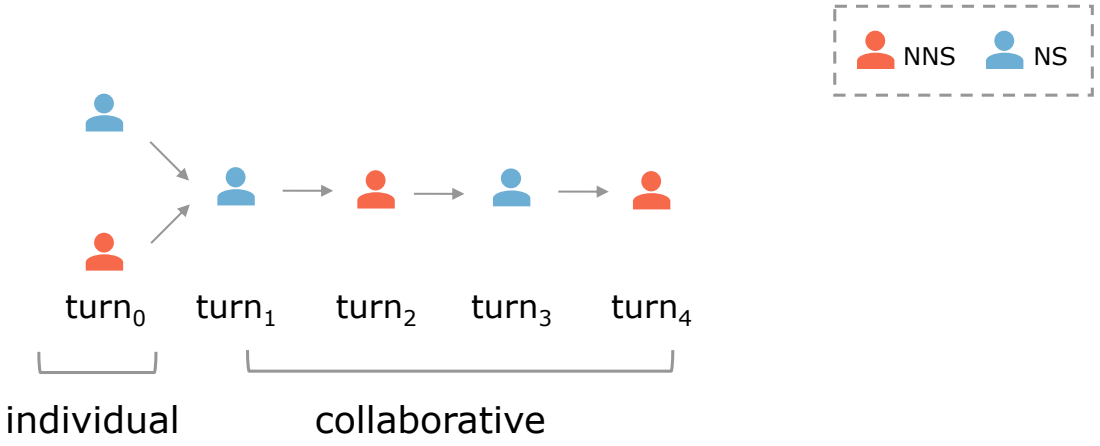


Figure 3.1: Turn-taking of NS (native speaker) and NNS (non-native speaker) of English in the multilingual collaborative writing study.

the next iteration, and finalized the visualization and interaction designs with them.

3.3.2 Data collection

Our collaborators recruited 29 native English speakers (NS) and 29 non-native English speakers (NNS) from an American university and a Japanese university for an online collaborative writing study. To ensure NNS are of similar English proficiency, all NNS are native Japanese speakers with *limited working proficiency* in English [189]. Participants are asked to act as if they were columnists for an English magazine who answer readers' questions about the role of technology in modern life. The topics include social media, remote learning, and digital privacy. To mitigate the influence of topic familiarity, one NS and one NNS are paired to form a writing group and are assigned a topic familiar to both authors. All participants are provided with preset Google accounts and links to blank Google Docs. NNS and NS of the same group do not know each other but are informed about co-authors' language proficiency. Participants are also asked to record their screen during writing. Participants are welcome to use any tools (e.g., search en-

gines, Google Translate, Grammarly) that they normally use during writing. Since the study was conducted before ChatGPT's release, no participant used generative AI tools.

Then, NNS and NS take turns writing an English essay jointly. The turn-taking setup is shown in Figure 3.1: first, each author writes independently (turn_0), ensuring they have actively thought about the task instead of being a free-rider. Then, NS review the write-ups of both authors from turn_0 and merges them into a single document (turn_1). During turn_1, NS can add/delete/edit any content. Next, NNS revise the joint document turn_2, followed by NS turn_3, and finally concluded by NNS turn_4. Participants are allowed up to 1 hour for each turn, and they could finish early if they have nothing more to contribute. After removing two teams that did not follow instructions, we have 27 remaining.

Our collaborators carefully designed the setup of the above experiment. First, though current online writing tools support simultaneous writing, turn-taking is still a popular workflow adopted by co-writers in practice, as they minimize the burden of syncing content mentally [190], protect authors' territoriality [154] and thus promote editing other co-authors' writing [191]. Second, NNS may face difficulties identifying opportunities for contribution in flexibly structured collaboration with NS. Previous research has introduced several techniques to interrupt the natural conversation flow and impose contribution opportunities of NNS, including artificial silent gaps [192] and a conversational agent [193]. Therefore, having a designated opportunity to ensure NNS contributes to collaborative writing is necessary. Besides, since it's one of the first studies on multilingual collaborative writing, researchers chose a simplified setup with only one NS and one NNS co-writer in a team to pinpoint the group dynamics easily, leaving more complex configurations for future research (e.g., NNS from different countries, unbalanced group settings where there are more NS than NNS or vice versa).

3.4 Data abstraction

Based on previous research, we introduce a general data model for the collaborative writing processes in our study.

Authors. In collaborative writing, there must be at least two authors. Let $A = \{a_1, a_2, ...\}$ be the set of authors. For each author a_i, the meta-information M_i is a set of the author's quantitative or qualitative attributes, e.g., author's ID, linguistic background, seniority and so on. In our study, there are two authors in a team, so $A = \{a_1, a_2\}$. Since we only care about the linguistic background of authors in this study, $M_1 = \{$*native-English-speaker*$\}$ and $M_2 = \{$*non-native-English-speaker*$\}$.

Events. Let E be the set of all collaborative writing events. The events could be on-document (E_{on}) and related ($E_{related}$), and $E = E_{on} \cup E_{related}$. Event $e_{on,i}$ is an edit made by authors, e.g., $e_{on,i}$ could include the author, editing types (add/delete), locations, and so on. Event $e_{related,i}$ is an event related to the collaborative writing process, but not limited to editing, e.g., leave a comment [194], make a post [156] and browse the internet. $e_{related,i}$ could include the event name, start and end time. We can automatically obtain E_{on} by comparing different document versions. To obtain $E_{related}$, one communication researcher manually coded events by watching the recordings, including the author, event type, and start and end time. After that, another communication researcher helped categorize events into higher levels. There are six types of events in total, and we highlight each event in different colors:

Writing: activities include "On Google Docs", "Using online editing tools", "Checking for creating citation".

Note-taking: activities include "Writing a note". The difference between "Writing" and "Note-taking" is that the note does not go into the writing, but serves as an auxiliary role.

Wordsmith-crosslingual: though the goal is to write an English essay, many NNS chose to write in Japanese and translate to English, or translate the write-up and read in Japanese. Such activities include "Using translators to read", "Using translators to write", "Searching for language-related information", "Checking a dictionary or thesaurus" and "Checking for meanings".

Wordsmith-English: some NS also seek help with expression, such activities include "Checking dictionary or thesaurus", "Searching for language-related information" and "Checking for meanings".

Active-search: authors may seek external evidence to build their argument. Activities include "Searching for online information".

Passive-search: After an author cites an external source in the collaborative write-up, the other author may check the content. Such activities include "Opening a URL to read information".

Table 3.1 shows summarized frequencies and duration of six high-level writing-related actions: "Writing" is the most frequent and time-consuming action, followed by "Wordsmith-crosslingual", "Active-search", "Passive-search", "Wordsmith-English" and "Note-taking".

Action	Duration		Frequency	
	NS	NNS	NS	NNS
Writing	52.5h (84.9%)	39.3h (61.9%)	706 (57.5%)	1548 (50.2%)
Passive-search	0.7h (1.1%)	0.1h (0.2%)	47 (3.8%)	22 (0.7%)
Active-search	8.1h (13.1%)	5.1h (8.1%)	425 (34.6%)	233 (7.6%)
Wordsmith-English	0.5h (0.8%)	0.0h (0.0%)	45 (3.7%)	1 (0.0%)
Wordsmith-crosslingual	0.1h (0.1%)	18.7h (29.6%)	4 (0.3%)	1277 (41.4%)
Note-taking	0.0h (0.0%)	0.1h (0.2%)	0 (0.0%)	4 (0.1%)

Table 3.1: Duration and frequencies of writing-related actions from video recordings

Turns and stages. In collaborative writing, authors take turns to write, which could be either sequential or parallel [149], depending on whether the timestamps of events overlap. In our study, as shown in Figure 3.1, $t_{0,ns}$ and $t_{0,nns}$ are turns where authors write individually, and t_1 to t_4 are collaborative turns. These turns are organized into writing stages; therefore, we have individual stages $t_{0,ns}$ for NS and $t_{0,nns}$ for NNS, and collaborative stages $\{t_{1,ns}, t_{3,ns}\}$ for NS and $\{t_{2,nns}, t_{4,nns}\}$ for NNS.

Document versions. Let V be the entire history versions of a single document, $V = \{v_1, v_2, ...\}$. Document v_{i+1} is the result of event E_i on v_i. Google Docs record word-level document history; however, our collaborators are not interested in such fine-grained analysis. Instead, they collected the document version at the end of each turn for each author $V_{end} = \{v_{0,ns}, v_{0,nns}, v_1, v_2, v_3, v_4\}$, then sampled three intermediate versions that reflected a person's writing progress during turn 2 to 4:

$$V_{inter} = \{v_{i,j} \mid i \in \{2,3,4\},\ j \in \{1,2,3\}\}.$$

In total, we have 15 versions: $V = V_{end} \cup V_{inter}$.

Sequences. The events happening in different turns from an author are grouped into a sequence based on the writing stages. As shown in Figure 3.1, we have four collections of sequences: $S_{NS,individual}$ are events in $turn_0$ by NS; $S_{NNS,individual}$ are events in $turn_0$ by NNS; $S_{NS,collaborative}$ are events in $turn_1$ and $turn_3$ concatenated; $S_{NNS,collaborative}$ are events in $turn_2$ and $turn_4$ concatenated. We chose to concatenate events in collaborative turns as suggested by communication researchers, as based on their experience, behaviors are similar across turns in the collaborative stage.

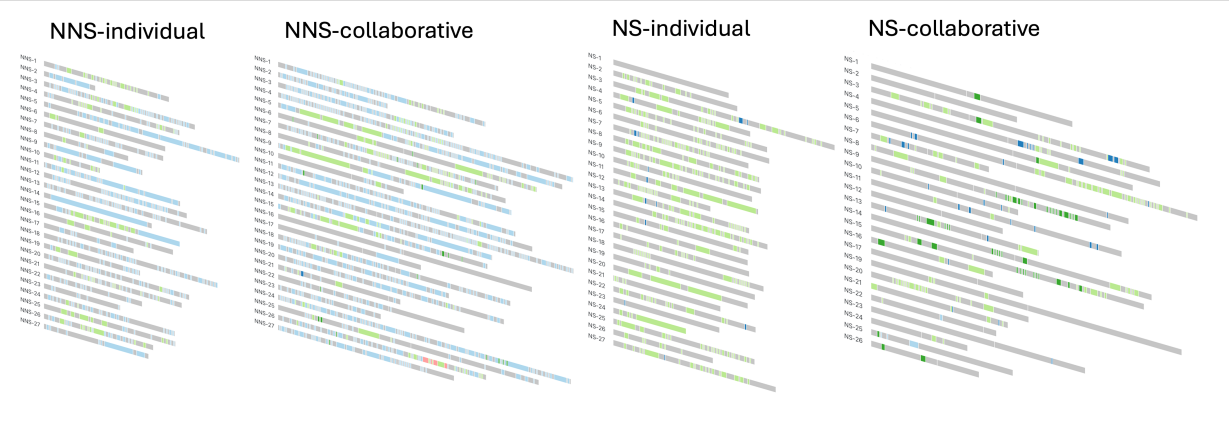


Figure 3.2: All sequences of non-native speakers' (NNS) and native speakers's (NS) behaviors during the individual and collaborative writing stage. The sequences in the collaborative writing stage are longer due to the concatenation of turns. The length of the rectangles indicates the duration of each event, and the color encodes the event types.

3.5 Methods

Communication researchers are interested in comparing authors' writing behaviors along two dimensions: writing stages (individual, collaborative), and author types (NS, NNS). For example, to answer the question "during the collaborative writing stage, how do NS and NNS writers' behaviors differ?", we need to compare two collections of sequences: $S_{NS,collaborative}$ and $S_{NNS,collaborative}$.

A primary challenge is that the sequences are highly heterogeneous, e.g., sequences in $S_{NNS,collaborative}$ range from 6 to 188 events, with an average of 75.6 and standard deviation of 47.0, making direct visual comparison impossible, as shown in Figure 3.2.

Therefore, we identified the following lower-level tasks to enable effective comparison:

- **T1: cluster similar sequences within each collection.** Besides automatic clustering methods, communication researchers should be able to incorporate their domain expertise into the clustering results.

- **T2: summarize each sequence cluster within each collection.** The summarization should provide rationales to help communication researchers interpret clusters.

In this section, we discuss computational methods to support task **T1** and **T2**, the interpretation challenges we faced in the design study, and our strategies to solve those challenges.

3.5.1 Computational methods

For T1, we decided to first automatically cluster sequences. Clustering algorithms are widely used to organize similar data points into groups [195]. Common methods include k-means [196], hierarchical clustering [197], DBSCAN [198], self-organizing maps [199] and so on. For T2, we reviewed the literature on visual summarization and data mining in event sequence analytics [200, 201, 202], and decided to use frequent patterns to summarize each cluster.

3.5.1.1 Method I: cluster and summarize sequences jointly

Our first approach is to cluster and summarize event sequences jointly. We chose Sequence Synopsis [22] because it is reported to produce higher quality visual summaries compared to other summarization methods [201]. Besides, sequential patterns are the most widely used summarization format [22, 172, 176, 203], compared to trees [174] and directed acyclic graphs (DAG) [173]. Sequence Synopsis clusters sequences and constructs a sequential pattern of each cluster by striking a balance between pattern conciseness and minimizing information loss from the original sequences. In our implementation, we can indirectly control the number of clusters K and the length of patterns by adjusting the weights of information loss and the number of patterns.

3.5.1.2 Method II: cluster first, summarize later

Our alternative approach is to cluster sequences first, and summarize each cluster. Here, we considered k-means and hierarchical clustering because these algorithms allow us to easily change the number of clusters. For each pair of sequences, we computed the Levenshtein distance (ignoring duration), which returns the minimal number of edits required to align the two sequences. Compared to other common distance metrics such as Euclidean distance, Levenshtein distance captures the order of the sequence and handles sequences of different lengths. Given K clusters, we also prefer nested results. For example, if two sequences are in the same cluster when $K = 4$, then we expected them to still be in the same cluster when $K = 3$. Therefore, we chose hierarchical clustering over k-means [204]. Since hierarchical clustering algorithms do not generate visual summaries for each cluster, we ran a commonly used maximal pattern mining algorithm VMSP [205] to extract patterns for each cluster. We set the minimum support to be 50%, i.e., the pattern has to be present in at least half of the sequences in the cluster. Different from Sequence Synopsis, which returns only one pattern for each cluster, VMSP returns multiple patterns, and we chose the longest one with the maximum support as the representative pattern.

Method I and method II differ in two key aspects: 1) the distance metric: method I does not rely on an explicit distance metric but learns how to cluster similar sequences and extract a representative pattern by balancing the pattern length and the information loss. In contrast, method II uses Levenshtein distance, which computes the edit distance between two sequences required for alignment; 2) the connection between patterns and clusters: for method I, the patterns and clusters are tightly coupled, each cluster is represented by a single pattern; for method II, clusters and patterns are loosely coupled, as pattern mining algorithms return multiple patterns for a clus-

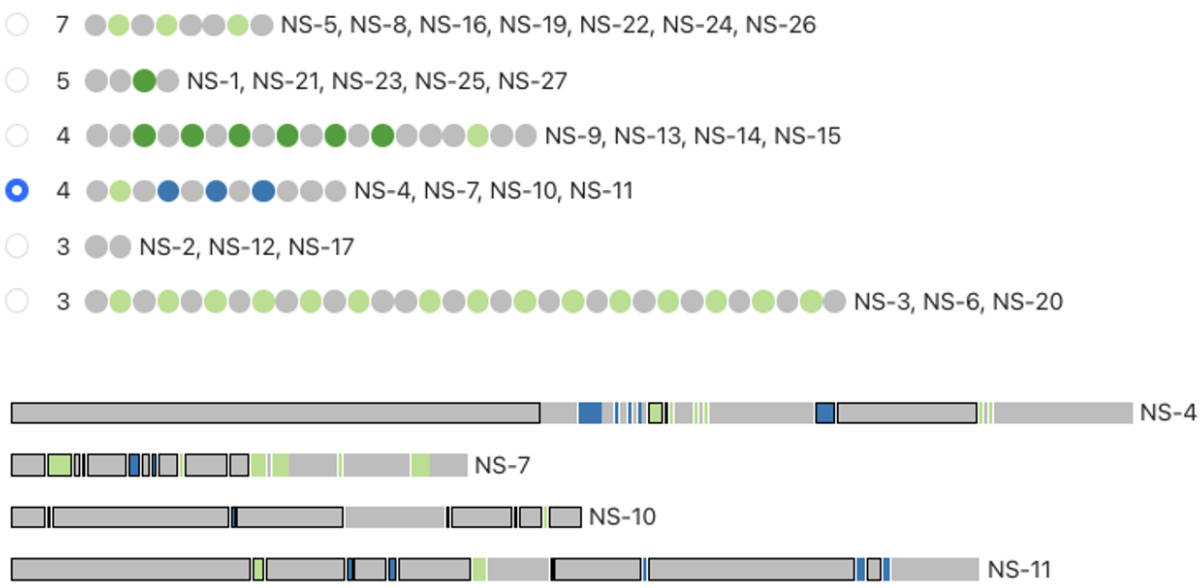


Figure 3.3: Six patterns returned by Sequence Synopsis [22] for native speakers during the collaborative writing stage. Each pattern is a sequence of circles, representing a visual overview of sequences belonging to the cluster. The original sequences of the currently selected pattern are displayed below. They are sequences of rectangles, with event duration encoded by length. Events matched to the pattern are outlined in black.

ter, it offers greater flexibility. By including both methods, COALA explores the computational space more thoroughly, provides multiple approaches to the clustering and summarization tasks, and thus brings different perspectives to the datasets.

3.5.2 Challenges in interpreting the computational results

We then designed an initial visualization to show the clustering and pattern mining results from these methods. Fig. 3.3 shows the result produced by Sequence Synopsis for NS' behaviors in the collaborative writing stage. There are six clusters, and for each cluster, Sequence Synopsis returns a representative pattern depicted as a sequence of circles. The number before each pattern is the cluster size, and the authors' IDs in the cluster are displayed after each pattern. To show the sequences in the cluster, users can click the radio button before each pattern (currently the

4th pattern is selected). In the raw event sequences displayed below the patterns, each event is a rectangle, where the color denotes the event type, and the length is the duration of the event. We highlight the event rectangles in black outlines if they are reflected in the pattern. We also implemented a similar interface for method II.

Visual summary. Though the patterns returned by Sequence Synopsis preserve the ordering of events in the sequences, the communication researchers expressed difficulty in interpreting such patterns and wonder whether it's possible to start with something more intuitive, for example, they suggest starting with one author's sequence and building clusters based on similar authors. Besides, it was unclear how each pattern differed from one another and how the algorithm performed the clustering. Take the sequences in Figure 3.3 for an example, the second and third cluster both feature a subsequence of Writing - Passive Search. However, it is unclear how these two clusters differ from each other. The same issue can be found in the first and the last clusters as well, where both clusters exhibit repetitive Writing - Active-Search subsequences.

Cluster membership. Since we have two sets of clustering results returned by Sequence Synopsis and hierarchical clustering, we let communication researchers switch to different results via a radio button. However, this caused great confusion. Even if we keep the number of clusters K the same for both methods, the clustering results returned by hierarchical clustering and Sequence Synopsis are not the same, and communication researchers are not sure which one to trust more. Besides, there are no explanations for why the sequences are grouped into each cluster or the differences between the clusters.

To summarize, the communication researchers encountered the following interpretation challenges:

- **C1:** two clustering methods' output differ.
- **C2:** lack of explanations of why the sequences are assigned to a particular cluster.
- **C3:** the extracted patterns are not easily interpretable.

3.5.3 Strategies to address the interpretation challenges

To address the above challenges, we devised several strategies, including ensemble and interactive clustering (**C1**), visualizing sequence-level information for clustering rationales (**C2**), and using large language models (LLM) to generate text summaries (**C3**).

3.5.3.1 Support ensemble and interactive clustering (C1)

To address **C1**, we decided to show the consensus and discrepancies in the results produced by different methods, inspired by the idea behind ensemble clustering [206].

We obtained multiple clustering results with varying numbers of clusters, ranging from 2 to N, where N represents the maximum number of patterns identified by Sequence Synopsis. We use Sequence Synopsis as a constraint because hierarchical clustering can produce as many clusters as the data points. Then we evaluate the clustering results of both Sequence Synopsis and hierarchical clustering using the same number of clusters. Given two clustering results A and B, and a cluster number K ($2 <= K <= N$), we try to match each cluster in A to a cluster in B in a way that maximizes the total overlap between cluster members. We used the Hungarian algorithm [207] to obtain the assignments with the most overlap. Then, we only keep assignments when the intersection size is larger than 1 (it is trivial to have a cluster of only one sequence). Therefore, the size of consensus clusters is usually smaller than K. For unclustered sequences,

we treat them as singletons.

Figure 3.4 shows a revised version of the visualization, where the sequences enclosed within a rectangle box have cluster assignments confirmed by both SequenceSynopsis and hierarchical clustering methods. In contrast, sequences without a consensus (i.e., NNS-2, NNS-3) are not enclosed, indicating they are singletons. Singletons are expected since the two methods leverage different computational techniques. Singletons remind communication researchers to give additional consideration to these authors, as computational methods differ in their cluster memberships. To help users assign singletons to clusters and revise existing cluster memberships, we provide a slider to adjust the number of clusters and support manually rearranging authors across clusters via drag-and-drop. Furthermore, to assist analysts in finding similar authors based on an author of interest, we also implemented a recommendation feature: we calculated the similarity between two sequences by summing over their sequence-level Levenshtein distance with the Levenshtein distance between their Sequence Synopsis [22] cluster patterns, and recommended top five to users.

3.5.3.2 Visualize sequence-level information for clustering rationales (C2)

To help users understand why sequences are clustered (**C2**) and assess the similarities and differences between sequences within and across clusters, we devised two solutions: one focuses on local information of individual sequences, while the other focuses on the global context:

- **Local information:** for individual sequences, we improve the visualization design to support users in comparing pairs of sequences visually.
- **Global context:** for all sequences, we reveal their pairwise distances to support users in

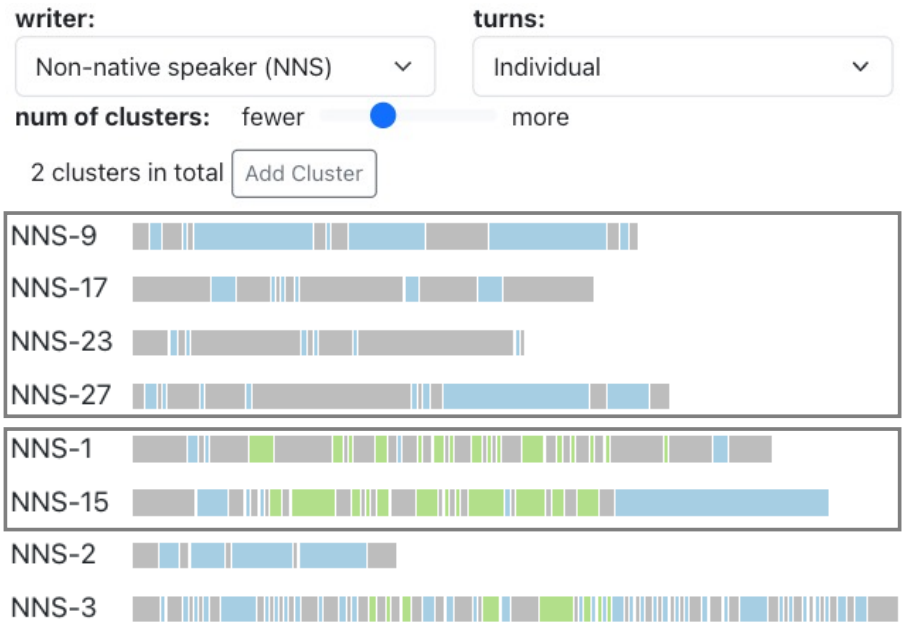


Figure 3.4: Consensus of clusters: sequences assigned to the same cluster by both Sequence Synopsis and hierarchical clustering are in the box; other sequences are outside of the box. Users can also change the number of clusters by dragging the slider, or manually add new clusters.

understanding the overall distribution of sequences in terms of pairwise similarity.

Visualizing local information of individual sequences. As shown in Figure 3.4, our earlier visualization design of an individual sequence presents the event sequences as it is, where colors encode the event types and the rectangle length encodes the duration. Such a design preserves all the information in the raw data, and communication researchers quickly conclude that NNS usually exhibit much more fragmented workflows than NS, frequently alternating between writing and other events. In contrast, NS usually allocates large chunks of uninterrupted time dedicated to writing. However, it is hard to generate additional insights. Therefore, we considered several design variants for visualizing individual sequences: trees, transition matrices, and arc diagrams.

- *Variant II: tree.* As communication researchers are interested in comparing NS and NNS'

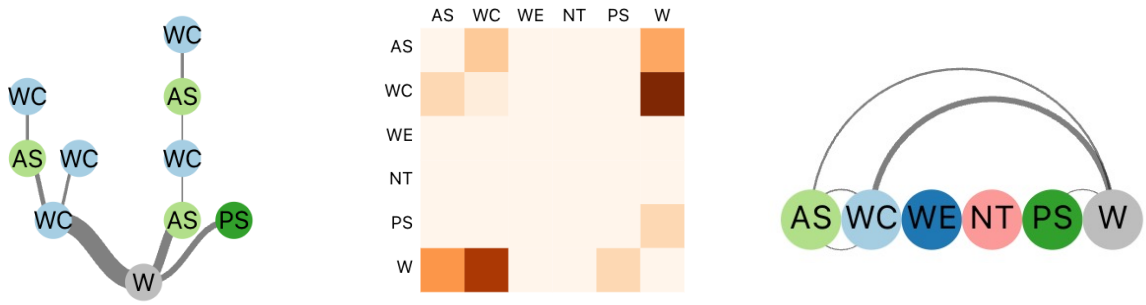


Figure 3.5: Design variants: tree, transition matrix, and arc diagram (final design) to visualize sequence information for author NNS-15.

events before Writing, we extracted all unique subsequences ending with Writing. As depicted in Fig. 3.5, each tree node represents an event, and each edge denotes a transition, with the edge's thickness reflecting the frequency. While communication researchers appreciated the completeness of the tree visualization, they noted some redundancies, e.g., excessively extended tree branches like "WC - AS - WC - AS".

- *Variant III: transition matrix.* To mitigate the issue of redundant sequences, we adopted a transition matrix, where each row/column is an event, and each transition starts from the row and ends at the column. The cell's darkness signifies the normalized transition frequency. As illustrated in Fig. 3.5, it's easy to identify common event pairs, e.g., from Writing (W) to Active-Search (AS) and vice versa. However, the matrix is sparse as frequent transitions concentrate on a few cells, resulting in underutilized space.

- *Variant IV: arc diagram.* For the final design, we chose an arc diagram. We borrowed the event node design from the tree visualization, but instead of arranging them in a tree structure, we put all the nodes on the same row and connected them with arcs of different thicknesses, indicating the transition frequencies. The communication researchers like

the simplicity and compactness of the design, as it is easy to spot which events precede Writing. Besides, unlike the transition matrix, where it is challenging to eliminate empty cells, we can easily conceal unwanted arcs. For example, we removed outgoing edges from Writing, as we are interested in events that precede each writing action instead of after.

Global pairwise distance of sequences. After improving the visualization for individual sequences, we helped users compare sequences globally via pairwise distances. For hierarchical clustering, we computed the Levenshtein distance between sequences and normalized it by the sequence length. Since Sequence Synopsis does not return distance scores, we used the Levenshtein distance between patterns as a proxy. Therefore, if sequences belong to the same cluster according to Sequence Synopsis, their distance is 0. We plot it on a 2D scatterplot, with each sequence represented as a dot, and the sequence of interest is placed at the origin.

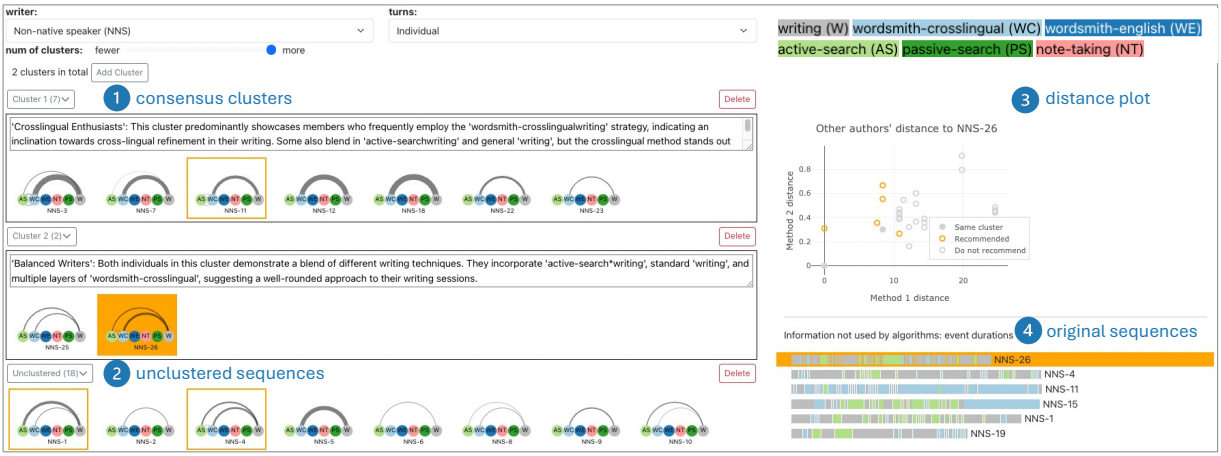


Figure 3.6: The clustering panel of COALA. ① shows consensus clusters of authors returned by Sequence Synopsis [22] and hierarchical clustering; ②: unclustered sequences. When users select an author, its background turns orange, and recommended authors are highlighted in orange outlines. On the right, a 2D scatterplot (③) shows the current author's behavior distance between other authors. ④ shows detailed sequence information.

3.5.3.3 Use LLMs to generate text summaries (C3)

To address **C3** (extracted patterns are not interpretable), we drew inspiration from the analysis process of communication researchers. We observed that they described clusters in natural language; for example, they described the visualization in Figure 3.4 as "*The first cluster mostly shows writing and wordsmith-crosslingual, so the NNS participants here spent most of their time figuring out how to produce writing in English through Japanese. The second cluster features more active search during the writing session even though participants still performed crosslingual language editing at the beginning and end of their sessions. These NNS participants followed a different writing path - they segmented the content (active information search in the middle) and the editing (wordsmith at the beginning and end) aspects of the writing and focused on one task at a time.*"

Recently, large language models (LLM) have shown great promise in data analysis [32, 38, 208] and LLM is found to have a high degree of agreement with human coders in thematic analysis [209]. Therefore, we leveraged LLM to generate text descriptions. We first tried capturing a screenshot of each cluster of arc diagrams (Figure 3.5) and prompted GPT-4V [210] with an explanation of the color encodings to describe each cluster. For example, we used the following prompt: "*The figure contains several event sequences, each showing an author's writing-related behaviors. There are six types of events: Active-Search, Wordsmith-Crosslingual, Wordsmith-English, Note-Taking, Passive-Search, and Writing. Each colored node is an event type, and you can find the event type in the colored legend. The arc thickness is the transition frequency of two events. Please name this cluster and provide a brief description.*"

However, GPT-4V sometimes ignored faint arcs between two nodes. To ensure GPT cap-

tures all transitions, including rare ones, we provided the transition data in JSON format, which was used to generate the arc diagram. Each entry contains the source event, the destination event, and the normalized frequencies. For example, source: Wordsmith-Crosslingual, destination: Active-Search, frequency: 0.25. Communication researchers found explanations returned by GPT fascinating and intuitive, and adopted some descriptions in their analysis. For example, GPT-4V calls one cluster "versatile writers" and explains that the cluster balances events between Wordsmith-Crosslingual and Active-Search, which suggests writers are comfortable with both actions without a dominant one.

Our prompting strategy to generate clusters could be generalized for other sequence datasets, and could be a plug-in for many existing visual analytics systems. We also found in later user study sessions that users borrow words from LLM-generated summaries during the analysis, especially for users new to the dataset.

3.6 COALA

Integrating all these strategies, we built COALA, a visual analytics tool to **co**mp**a**re col**la**borative writing behaviors of native and non-native English authors. It has two tabs: one for inspecting and modifying the clustering results (Figure 3.6) and the other for comparing clustering results for authors of interest (Figure 3.7).

The first tab (Figure 3.6) supports refining sequence clusters and summaries. After selecting authors and writing stages in the dropdown menu, it displays consensus clusters of sequences by Sequence Synopsis and hierarchical clustering (①) and unclustered sequences (②). Users can drag the slider to change the number of clusters, add or delete clusters, revise cluster de-

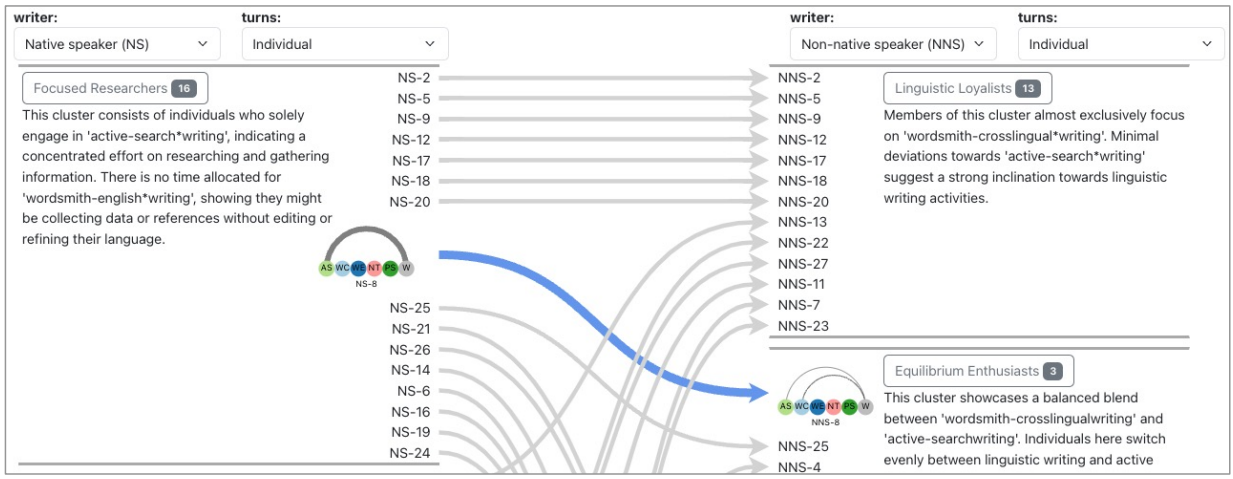


Figure 3.7: The comparison panel of COALA. It shows a two-panel layout: each has its dropdown menu for users to configure the author and writing stages. An arrow connects authors from the same team on both panels. Users can click an arrow to directly compare the two sides' arc diagrams.

scriptions, and drag and drop arc diagrams directly across clusters. To facilitate the refinement process, when users select an author's arc diagram, its background turns orange, and recommended similar authors outside its cluster are highlighted in orange outlines. Currently, an author in the second cluster (orange background) is selected, and several authors outside the cluster are recommended. On the right, a 2D scatterplot (③) shows the selected author's sequence distance between other authors according to Sequence Synopsis (method 1) and hierarchical clustering (method 2), and users can hover over each dot to see the sequence ID and distance. Users can also hover over ④ to inspect the event types and durations.

The second tab (Figure 3.7) supports directly comparing collections of sequences and individual sequences. It shows a two-panel layout; each panel has its dropdown menu for users to select collections of sequences. Currently, we select NS-individual on the left panel and NNS-individual on the right panel. An arrow connects authors of the same team. Authors are organized by clusters in Figure 3.6 and sorted by minimizing edge crossings. Once the user clicks an ar-

row, arc diagrams will appear on both panels. For example, now it shows that NS-8 transitions frequently between Active-Search and Writing, while NNS-8 has a balanced Active-Search and Wordsmith-Crosslingual activities before Writing, highlighting NNS' effort in information gathering and also translation. For cluster-level comparisons, since the arrows are sorted, users can observe the flow directions, similar to a Sankey diagram.

3.7 Validation: user studies

The validity of our design is rooted in the user-centered design process reported in the previous sections. We also organized two user studies to evaluate COALA. The first, a focus group session with two existing and two new communication researchers, examined the usage of COALA in real-world setting; the second, individual sessions with 8 graduate students, evaluates the effectiveness of COALA for general users who work more independently.

3.7.1 User study setup

3.7.1.1 Focus group (N=2+2)

We organized a focus group session with the two existing communication researchers along with two of their colleagues. All four participants belong to the same research group. The two new researchers are familiar with the multilingual communication research but are unfamiliar with the dataset details, nor have they seen COALA before. Though our collaborators are informed about the methods and individual visualizations, they also have not used COALA. Therefore, we conducted a 1-hour study session with all communication researchers (N=2+2). The goal of the group study is to use COALA in a realistic work setting to make sense of a dataset. In the

beginning, we introduced the dataset and analytic tasks (**T1** and **T2**), and then we gave a tutorial on how to use the tool. We deployed COALA online, and each researcher accessed the tool on their laptop. They were encouraged to think aloud and discuss their findings during the session. We took notes during the session, and after the group session, we also invited them to write down their findings in a shared document.

3.7.1.2 Individual user studies (N=8)

Besides the focus group, we also expanded the evaluation scope by recruiting eight graduate students with related research backgrounds (communication: 2, second-language education: 2, CSCW: 2, political science: 1, natural language processing: 1). All participants have first-hand collaborative writing experiences and are interested in understanding collaborative writing behaviors. Among them, four are NS, and four are NNS. To better evaluate the usability and design effectiveness of COALA, we set up individual user study sessions with them.

The procedure is similar: first, participants read a background introduction of the dataset (we prepared a shortened and simplified version of section 3.3 and section 3.4). We then demonstrated how to use COALA to cluster similar authors and compare their writing behaviors. Participants were asked to complete two tasks: analyzing authors' behaviors across two dimensions—writing stages (individual vs. collaborative) and author types (NS vs. NNS). Participants can either write their findings in COALA or describe them verbally. Participants are given one hour to complete the tasks. Participants were asked to think aloud during the study, and after they completed the tasks, we conducted a brief interview to ask them questions about the effectiveness of COALA and the quality of model-generated results. We chose not to collect the ratings quan-

titatively due to the limited sample size and the lack of a baseline tool for comparison. Instead, we focus on gathering more qualitative feedback. All except one participant completed the tasks on time, and every participant was paid with a $20 Amazon gift card. The study is IRB-approved.

3.7.2 Effectiveness of COALA

3.7.2.1 Focus group (N=2+2). COALA facilitate discussions

All expert users had no difficulty in using COALA, and they were surprised to see that they focused on different aspects of the dataset during the analysis. For instance, one focused on the durations in the event sequence, and the other focused on the event transitions. This divergence triggered experts' discussion of why one aspect of collaborative writing was important. Ultimately, they concluded that COALA offered multiple angles to analyze the dataset they would otherwise miss due to their intuitions.

3.7.2.2 Individual user studies (N=8). COALA helps uncover insights

Among the seven participants who completed the study, three found the arc diagram design more effective, two preferred the sequence diagram, and two considered both effective. Participants prefer the arc diagram mostly because it summarizes the event sequences. For example, P8 appreciated how the arc diagram *"compresses information"*. In contrast, participants preferred the sequence diagram because it encoded the duration information (P5).

Regarding the model quality, five out of seven participants considered the initial clusters obtained by the consensus methods to be high quality and encouraged unbiased exploration. For example, P7 said, *"It's a really good starting point, because if you were to analyze this by*

yourself without any ideas...you can fall into typical stereotypes that...an AI model would not." Furthermore, six participants found the recommendation feature helpful, using it as a starting point for the analysis (P8), a voting mechanism (P6), and a verification method (P7).

Participants also made suggestions to improve model-generated results, including more details about how the recommendation methods work (P4), more fine-grained clusters to start with (P5), and a 3rd recommendation method based on time duration (P6).

Notably, none of the participants had analyzed multilingual collaborative writing datasets before. Despite their unfamiliarity with the dataset and COALA, most of them were able to familiarize themselves with COALA and complete the analysis tasks. By leveraging the model-generated results and data visualization designs, they uncovered insights similar to those of expert users and connected the findings with their own experience. This outcome highlights the general usability and design effectiveness of COALA.

3.7.3 Findings: collaborative writing patterns

Here we aggregated participants' findings by using COALA, including the dataset itself and their reflections on their own collaborative writing experience.

3.7.3.1 Individual stage patterns

NS and NNS are asked to write independently before collaborating in this experiment to avoid free-riders. Communication researchers observed distinct behaviors during the individual stage.

NS research extensively. Using our tool, communication researchers easily identified several

major clusters of NS. One cluster is characterized by dominant Active-Search - Writing behaviors, indicating they actively incorporated external information into their writing. Another cluster has more activities, besides AS - W, they also engaged in extensive paraphrasing activities: Wordsmith-English - Writing. Besides these two major clusters, there are also uncommon behaviors, e.g., one NS only displayed WE - W behavior without relying on external information; one NS went even further, solely engaged in writing, degenerating the arc diagram into a stick on the W node.

NNS encounter costly bilingual content production. Similarly, communication researchers identified several clusters of NNS. In one cluster, Wordsmith-Crosslingual - Writing dwarfed any other behaviors, implying NNS had significant usage of their native language to produce writing in English. Another cluster has more Active-Search - Writing behaviors, showing NNS also incorporated external information into their writing. Different from NS, several NNS also displayed AS - WC, meaning NNS also used their native language to transfer content from their content-related search to their writing output. Among the NNS who have engaged in both WC - W and AS - W, half are dominated by WC, and the rest are dominated by AS or are balanced. Notably, none of the NNS engaged with Wordsmith-English - Writing during the individual writing stage, highlighting the bilingual nature of NNS' writing process, as their limited English proficiency may have dissuaded them from writing directly in English. Compared to NS, many more NNS (n=10) only engaged in WC - W without relying on external information. This pattern hinted that NNS have dedicated much of their time to the costly process of writing in English - generating content in their native language first and then translating the content to English, either by themselves or by leveraging support from cross-lingual dictionaries and translation tools.

3.7.3.2 Collaborative stage patterns

Communication researchers found more diverse actions during collaborative stages, driven by co-authors' need to exchange information, refine text, and communicate with each other. For example, communication researchers observed multiple NS (n=8) and NNS (n=10) engaged in Passive-Search, indicating active information sharing. NS and NNS also display different distributions of behaviors in the collaborative stage.

NS shoulder more editing responsibilities. Transitions between AS-W and WE-W still have a strong presence, though the balance has shifted, with WE-W increases, and AS-W decreases. The shift towards editing indicates that NS is responsible for editing both parties' English expressions.

NNS gain more bandwidth for other tasks. Communication researchers witnessed an uptick in both WC-W and AS-W, suggesting NNS actively interpreting and incorporating NS-contributed content. In addition, more than a third of NNS engaged in PS-W. These observations suggested that when co-writers completed initial drafts and transitioned to a collaborative editing stage, NNS were partially relieved from the demanding task of producing English content. Thus, they had more bandwidth to perform other tasks, such as AS to enrich the joint writing further.

3.7.3.3 Echo with first-hand experience

In the individual user study sessions, besides the findings similar to the ones in the group study session, our participants also compared their findings through the lens of their collaborative writing experience. For example, P6, a native English speaker, noted that too many NS *"just*

spitball on the fly without actually doing any research" in this dataset, which aligned with his previous collaborative writing experience. In contrast, P2 (a non-native English speaker) observed that *"...native speakers become more cautious with their writing...when they are working with people, which is something that I wouldn't expect...if I am working with a non-native speaker of my language. I feel like I know more, so I would feel less need to actually check my writing."* P4 resonated with the Active-Search - Wordsmith-Crosslingual transition in the arc diagram: *"I was seeing myself do the same thing. I also like search for specific words in Korean, and then translated into English."*

Some participants also voiced their suggestions for best practices in collaborative writing. For example, P3, a native English speaker, pointed out that so much time spent on Wordsmith-Crosslingual is *"a waste of manpower, it makes more sense to focus on the ideation and don't care if it comes out ugly if the other person (NS) can fix it without having to do a lot of wordsmithing."*

3.7.4 Findings: participants' analysis strategies

We also studied participants' analysis strategies. For the group study, we analyzed the notes taken during the study and the document on which participants wrote findings. For the individual study, we analyzed the video recordings, including participants' screens and meeting transcripts.

3.7.4.1 Experts (N=2+2)

COALA is pre-populated with clusters returned by methods described in section 3.5. Initially, all communication researchers used those clusters to gain an overview of the dataset. They read the descriptions generated by GPT-4 and gradually started to refine clusters based on their

understanding.

For example, one researcher (E1) merged clusters based on the role of external resources in the writing processes: authors who use **Wordsmith-English** or **Wordsmith-Crosslingual** are classified as *"grammar-based"*, as they only need help with the expression, but not ideation; authors who leverage Active-Search and Passive-Search are classified as *"research-heavy"*, as such authors are still in the process of finding ideation.

On the opposite, another researcher (E2) broke down clusters into sub-clusters by examining the duration and transition frequencies of raw sequences carefully (Figure 3.6.④); for example, NS-20 and NS-22 were initially clustered into the same cluster by our algorithms as their behaviors are dominated by Active-Search, however, after examining the raw sequences, E2 found that NS-20 usually spent a long time on Active-Search before transitioning to Writing, different from NS-22, who rapidly transitioned between those two. Therefore, E2 created a new cluster called *"in the flow"*, and placed authors with less fragmented workflow into this cluster.

During the refinement, they also use the recommendations (orange border) to guide them in finding similar authors. After refinement, they moved to the comparison panel (Figure 3.7) and examined individual authors' behavior changes.

3.7.4.2 General users (N=8)

Similar to focus group participants, participants in individual sessions also started by understanding the pre-populated clusters and text descriptions.

Clustering strategies. We found participants in individual sessions follow similar high-level clustering strategies of E1 and E2. Similar to E1, four participants also clustered authors by

grouping multiple activities. For instance, P3 and P5 assigned authors performing more than two activities to a *"multi-tasking"* cluster; P3 assigned authors with AS or PS to the same cluster, as such behaviors implied that the authors incorporated external information.

Similar to E2, three participants refined the clusters with more fine-grained criteria. For example, P4 created two clusters for authors exhibiting dominant AS - W behaviors and labeled them as *"comfortable/uncomfortable with searching information in English"* based on the absence of AS - WC. P7 and P8 broke down clusters by considering the events' time durations, for example, P7 differentiated authors with WC - W behaviors by the event durations and labeled authors spent significant time on WC as *"long-term crosslingual enthusiasts"*. Similarly, P8 created several clusters with names like *"high/low active-search"* and *"strong/weak passive-search"* based on the durations.

Participants also interpreted the same behavior differently. For instance, for authors who wrote extensively without engaging in other activities, participants described them as *"have lots of knowledge and can pull citations from their memory"* (P4), *"stick to minimal approaches"* (P5), *"YOLO"* (P6) or *"confident"* (P7).

Reliance on the recommendation feature. Compared to E1 and E2, participants in the individual sessions rely more on the recommendation feature to discover similar authors. Some participants used it as a starting point to narrow the candidate pool; others used it to verify their assumptions. For example, P7 identified similar authors visually and then clicked the author of interest to see whether the recommendation confirmed her assumption. P6 employed the recommendation feature as a voting mechanism: when adding a new author to an existing group, P6 cross-checked the recommendation of multiple existing authors and only picked those endorsed

by multiple existing cluster members.

Similarly, after refining clusters, participants moved to the comparison panel to get an overview of the authors' behavior changes and clicked authors of interest for more fine-grained comparison.

3.8 Discussion

In this section, we reflect upon the lessons learned from understanding collaborative writing processes through visual analytics and the implications for AI-assisted collaborative writing tools.

3.8.1 Reflections on developing visual analytics tools

Information-seeking and visual analytics mantras have limitations in guiding tool design. When we started working with the communication researchers, we aimed to create an overview of authors' behaviors, following the visual information-seeking mantra "overview first, zoom and filter, then details-on-demand" [211]. Therefore, we chose Sequence Synopsis [22], which clusters sequences and generates an overview pattern for each cluster. However, as discussed in section 3.5.2, communication researchers found such overviews confusing. Instead, they asked whether it was possible to start with one author's sequence and manually build clusters based on similar authors. This request directly contradicted the mantra, and we realized that the introduction of computational models resulted in interpretation challenges, and an overview would only be meaningful if users could reliably interpret it. Since it is tedious to inspect and cluster individual sequences manually, we kept the automatic clustering results but addressed the interpretation challenges with ensemble and interactive clustering and large language model (LLM)-generated

text summaries. With the advances of AI, we expect more complex computational models to be incorporated into analysis. Although the visual analytics mantra ("Analyze first, show the important, zoom/filter, analyze further, details on demand") proposed by Keim et al. [212] tries to address the importance of analysis, what constitutes "the important" is highly dependent on the computational models and the tasks involved. Therefore, it is critical to emphasize the interpretability of model results and build a shared representation between analysts and models [213].

Visualizing ensemble clustering methods requires consolidating results. Clustering simplifies the analysis by grouping similar data points, but the variety of clustering methods and clustering results raised doubts among our collaborators about the output reliability. At first, we simply used radio buttons to toggle between methods, inadvertently presenting each method as interchangeable black boxes. However, having an agency over black boxes does not gain users' trust. Therefore, we visualized the uncertainty explicitly using a bounding box: sequences assigned to the same cluster by different methods were clustered, while discrepancies resulted in singletons. Besides, we also visualized the distance between different authors in a scatterplot to allow communication researchers to explore the similarity space intuitively. Our lessons show that mere juxtaposition is not enough when visualizing multiple models' results; instead, we should support analysts in interpreting multiple models' results without model-specific knowledge.

Caution needs to be exercised when updating model results based on user feedback. Our communication researchers think the automatically generated clusters are a useful starting point, yet still require revisions. Therefore, we implemented several user interactions, such as adjusting the number of clusters and rearranging cluster members. To minimize manual operations, we further suggested updating the recommendation interactively, e.g., if an author was moved from

one cluster to another, then both clusters' descriptions would change, and the distance function would be revised. However, our collaborators found this distracting, preferring a more stable interface and manual revision of AI-generated results. Though incorporating users' feedback can improve the quality of model results [214], it also requires more thoughtful inputs from users and imposes extra cognitive load to examine the updated interface [73, 215]. Therefore, future tools should carefully balance the benefits of refined models and the additional burden on users.

3.8.2 Design implications for AI-assisted collaborative writing tools

With the recent advances of large language models (LLM), researchers have proposed multiple AI-assisted writing tools [72, 216, 217, 218, 219, 220, 221, 222, 223], some are tailored for non-native speakers [16, 224, 225]. However, we have not yet seen AI-assisted writing tools targeting collaborative writing, potentially due to the lack of understanding of the dynamics of collaborative writing. Here we propose several potential features derived from our study.

Detect diverse contribution types. Traditional collaborative writing tools [135, 158] for tracking authors' contributions in a collaborative document usually rely on word count. In our initial exploration (Appendix), we also found that NS contributed more text, and our collaborators mentioned that sometimes NS complained that NNS wrote too little. However, according to the communication researchers, there are diverse contribution types that are not necessarily manifested in word counts. Tools that quantify multi-dimensional contributions will help teammates understand each other's contributions and foster team dynamics. For example, NNS could contribute to the overall writing by providing an idea, which was expanded and refined by NS. Other contributions, such as scaffolding and improving the document flow by reorganizing, are also equally

valuable. Automatically detecting such contributions requires tracking and deeply understanding the document, which may be potentially feasible by leveraging LLM.

Reduce context switching for NNS. NS and NNS spent about the same time on this lab study. However, the way they spent their time is notably different. NNS spent much more time transitioning between Wordsmith-Crosslingual and Writing, resulting in a more fragmented workflow. In contrast, NS usually dedicated extended time to writing, engaging in focused and uninterrupted work, as highlighted by Newport's concept of "deep work" [226]. To improve efficiency for NNS, we suggest that future collaborative writing tools introduce features to reduce multilingual context switching. For example, code editors like VSCode [227] support generating code by providing instructions in natural language, reducing the time for developers to search syntax of programming languages in external resources. Similarly, an AI-assisted editor could support NNS by providing high-level instructions in their native language and generating text in English.

Provide guardrails for machine translation. NNS is frequently involved in Wordsmith-Crosslingual during both individual and collaborative writing. However, machine translation is prone to information loss [228, 229], and NNS are usually unaware of it. For example, during an interview conducted by our collaborators, NS-1 mentioned that NNS-1 left a sentence, "*Why don't you stop comparing yourself to others and focus on yourself*?" Though NS-1 perceived an accusatory tone and wanted to change it, NS-1 eventually decided to respect NNS-1's writing and did not revise it. However, it was later revealed that NNS-1 originally wrote it in Japanese and the soft tone was lost in translation. Therefore, future collaborative writing tools could have built-in translators and monitor the tone before and after translation.

Summarize collaborators' activities. During the collaborative stage, we observed that authors

spent some time catching up on the latest changes in the document, for example, reading links provided by co-authors (Passive-Search). Though online collaborative writing tools like Google Docs have version history, they only reflect word-level change and do not summarize the semantic meanings of version changes. Therefore, future AI-assisted collaborative writing could support automatically summarizing co-authors' activities and facilitate the syncing up process.

3.8.3 Generalizability for understanding diverse collaborative processes

Though COALA is initially designed for a specific dataset that our collaborators have collected, COALA also has the potential to help analyze more generalized and diverse collaborative processes.

Diverse collaborative writing settings. Since our collaborators' study is one of the first studies on multilingual collaborative writing, they opted for a simplified setup: one NS and one NNS with clear turn-taking. In the future, communication researchers may conduct user studies with more complex scenarios, e.g., multiple NS and NNS, or NNS from different countries. Since COALA follows a generalized "cluster-summarize-compare" workflow, COALA can adapt to and process new sequences provided by researchers. Furthermore, the computational methods underpinning COALA are agnostic to event or author types, enabling researchers to introduce events and author types beyond those in Table 3.1. For example, if communication researchers want to study how people of varying Wikipedia editing experiences co-write an article, they could leverage a set of events recorded by Wikipedia, such as "create", "add", "delete", "revert". In this case, the factor of interest shifts from language proficiency to the authors' Wikipedia editing experience.

Collaborative processes beyond writing. While COALA focuses on analyzing multilingual collaborative writing, the design principles and computational techniques could extend to a wide range of collaborative processes that generate rich event sequences, such as visual information analysis, knowledge synthesis, and problem-solving. For instance, Isenberg et al. [230] examined how different teams behave in an information analysis task and derived a framework for such activities by analyzing temporal sequences. In another study, Isenberg et al. [231] recruited 15 pairs of participants to solve a problem by exploring 240 digital documents and identified eight collaboration styles. Similarly, Robinson et al. [232] recruited five pairs of geographers and disease biologists to complete a knowledge synthesis task, analyzing the participants' action frequency, time durations, and strategies. Recently, Yang et al. [233] studied how teams collaboratively engage and perform sensemaking tasks in an immersive environment. These studies highlight the importance of understanding the temporal sequences of participants' behaviors and identifying common patterns—an area where COALA's clustering and summarization features could provide significant benefits. Unlike previous methods which rely primarily on manual coding or frequency-based analysis, COALA offers a data-driven approach to interpret collaborative dynamics, and thus enhances both scalability and granularity in data analysis.

3.8.4 Limitations

One limitation of COALA is that it does not visualize text changes during collaborative writing. In the individual user studies, the only participant who struggled to complete the analysis tasks noted that having text alongside event sequences would make the analysis more concrete. While we explored adapting existing text visualizations to our dataset (see Appendix), our focus is

on novel approaches to support behavioral analysis. Future tools could seamlessly integrate both aspects to enhance analytical depth. Another limitation is that COALA's comparison feature is designed for dichotomous analysis, such as collaborative vs. individual or NS vs. NNS. For teams without clear binary distinctions, COALA only supports clustering but lacks dedicated comparison capabilities.

Chapter 4: SAFEGUARD AI

4.1 Introduction

Recently, AI has significantly transformed various aspects of society, from customer service chatbots to autonomous vehicles. To ensure the safe deployment of AI systems, regulators and AI communities from worldwide have proposed AI regulations [234, 235, 236], standards [237, 238, 239, 240, 241, 242, 243, 244, 245] and guidelines [246, 247, 248, 249]. For example, the EU AI Act [235], the first AI law that came into force in August 2024, categorizes AI systems into five risk levels: prohibited, high-risk, general purpose, limited-risk, and unregulated. The Act specifies the responsibilities of systems providers, particularly high-risk systems, such as those used in autonomous driving, credit scoring, and medical assistance, must demonstrate that the system complies with the mandatory requirements for trustworthy AI. This includes implementing quality and risk management systems to ensure compliance and mitigate risks.

One popular approach to demonstrate machine learning models' performance and risks is Model Cards [23]. As shown in Figure 4.1, a model card may include the model performance, training and evaluation dataset, and ethical considerations. However, according to a large-scale analysis of 32k model cards on HuggingFace [250], a leading open-source machine learning platform, the evaluation and limitations of the models have lower filled-out rates.

Besides, though model cards reflect models' performance, it does not directly link to regulations. For example, in self-driving domain, regulations are called ODDs (Operational Design Domains), which specify conditions that a self-driving car needs to test. Developers may vaguely know that the pedestrian detectors do not perform well on rainy days, but the regulation may have specified much more detailed conditions, including both weather types and road structures, and anecdotal evidence of the model performance does not satisfy the needs for regulation compliance. However, there are too many different combinations of different conditions, and thus it's infeasible to check all combinations manually.

A data subset depicted by a combination of metadata is called a "data slice", and many slice discovery algorithms have been developed to find divergent data slices (e.g., gender=female, age=40-50) given a performance metric [251, 252, 253, 254, 255]. However, training data often lacks regulatory metadata, e.g., training data for a pedestrian detector may include bounding boxes of humans, but not the road conditions and weather types. Furthermore, regulations are usually vague and may not explicitly mention all the conditions, therefore, it's also important to envision novel and realistic conditions based on the regulation document.

Recently, large language models are equipped with vision capabilities, and those vision-language models have shown promise in describing images, achieving high scores in benchmarks of vision tasks [256]. Besides, they also show promise in brainstorming [257, 258, 259]. For example, given a machine translation task, generative AI can brainstorm critical situations (e.g., immigrants use it to communicate with government officials) and potential harms (e.g., immigrants lose opportunities due to inaccurate translations) [257]. However, such envisioning tools are primarily designed to raise the awareness of model developers instead of quantitatively validating model performance.

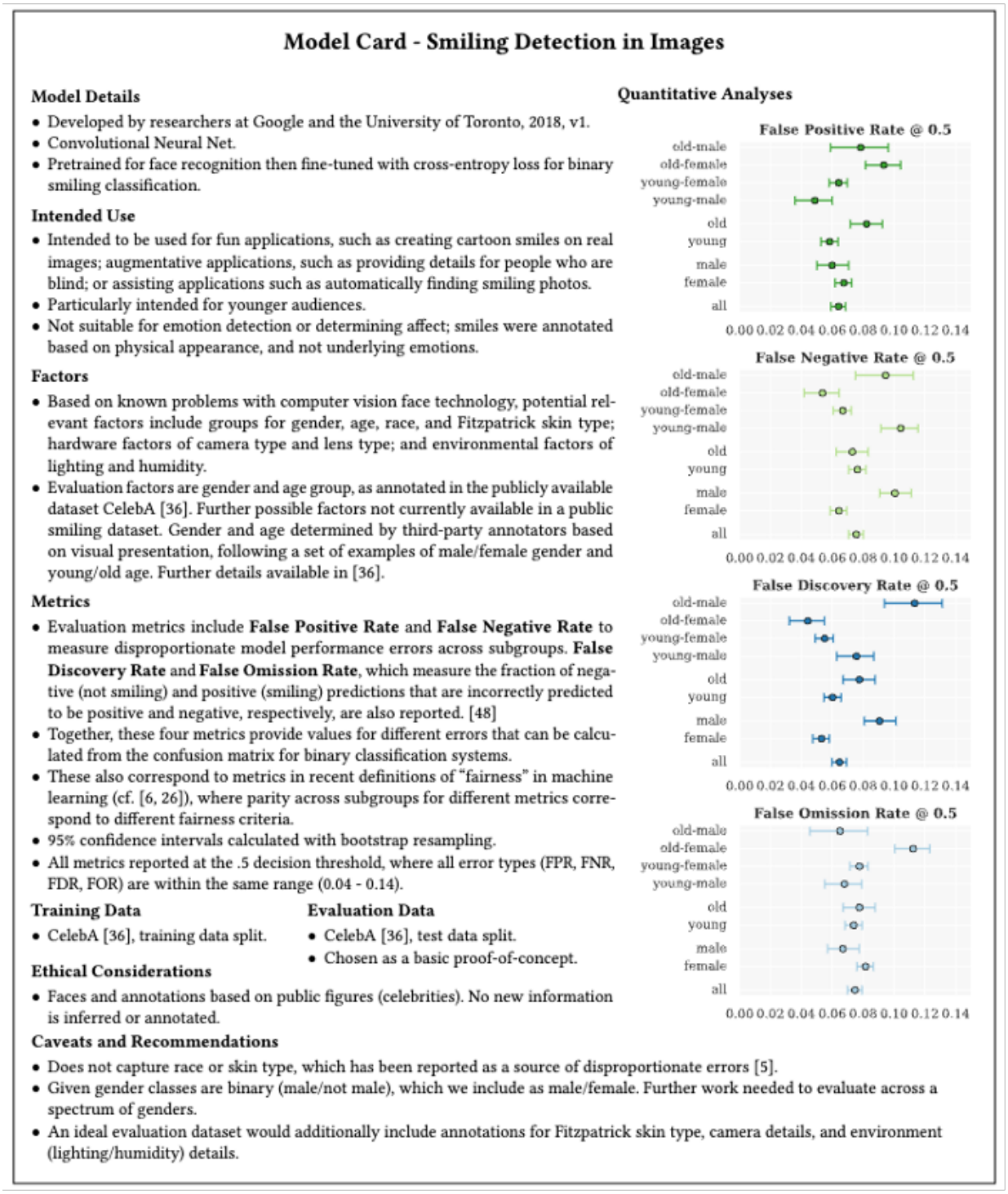


Figure 4.1: An example of model cards [23]

To bridge the gap between regulation compliance and model validation, we propose SAFEGUARD AI, which leverages generative AI agents to generate metadata for unlabeled datasets based on regulations, validates model performance and data coverage quantitatively, and envisions novel data slices not captured by the regulations. To summarize, our contributions include:

- A novel multi-agent workflow to validate machine learning models without relying on metadata.
- SAFEGUARD AI, a visual analytics system to help collect regulatory compliance evidence in terms of model performance and data coverage.
- Empirical results in two high-risk AI applications: a pedestrian detector of self-driving systems and a fire detector in fire alarm systems.

4.2 Related work

4.2.1 Regulation compliance

Recently, regulations and standards to ensure the safe and ethical use of AI are mounting. For example, EU AI Act [235], Exectutive Order on AI [260], NIST's AI Risk Management Framework [237] and International Organization for Standardization (ISO)'s AI-related standards in governance [245], risk management [240], AI lifecycle process [239], trustworthiness [241], bias [238], ethical and societal concerns [243], AI system management [242], and quality models [244]. Additionally, communities have proposed various responsible AI guidelines [246, 247, 248, 249, 261, 262].

However, demonstrating compliance with these regulations and guidelines remains a chal-

lenge. Traditional compliance software to monitor regulation compliance, such as SAP [263], Microsoft Purview Compliance Manager [264] and AWS compliance programs [265] are usually for company-level regulation compliance, instead of model-level.

4.2.2 Slice-based model validation

A common model-level validation approach are model card templates introduced by [266], including training process, datasets, and limitations. However, in practice, the evaluation and limitation sections in current model cards are usually undocumented [250], or only report model results in data subsets of a single category, e.g., gender==female. In practice, combinations of several metadata could lead to problematic data subsets, for example, commercial gender classification software may perform much worse on certain gender and skin types [267].

Manually checking all types of combinations of metadata is infeasible, and slice discovery algorithms are widely used to identify problematic data slices [251, 252, 253, 254, 268]. Some responsible AI toolkits also include slice discovery feature, for example, Microsoft's responsible AI toolbox [269], IBM AI Fairness 360 [270] and so on [271, 272].

However, the above methods rely heavily on metadata to characterize data slices. For unstructured high-dimensional data like videos and images, the labeling is challenging. Though researchers have proposed slice-finding methods to replace human-annotated metadata with proxies, such as attribution-weighted features of Convolutional Neural Networks (CNN) [273], the final hidden layer of neural networks [274], interpretable visual concepts [275, 276] and cross-modal embeddings [255, 277], those methods are not directly linked to regulation compliance.

SAFEGUARD AI is also a metadata-free slice-based method. The difference is that SAFE-

GUARD AI obtains metadata based on regulations through large language models, thereby bridging the gap between slice-based model validation and regulatory compliance.

4.2.3 Slice-based visualizations

Several methods have been proposed for visualizing data slices. SliceTeller [254] applies a matrix visualization based on UpSet [278], where each row is a data slice, and each column is an attribute of metadata, the presence of an attribute could be denoted as a dark circle in the cell. Divisi [279] uses a scatterplot, where each dot is a data slice, and their relative distances are based on t-SNE [280]. AttributionScanner [273] developed data slice mosaic, which projects data slices in 2D space by UMAP [281] and visualizes their convex hulls. Visual Auditor [282] visualizes the underperformed data slices characterized by at most two attributes in a co-occurrence matrix, where each row and column is an attribute, each cell is a circle, and the circle size denotes the dataset size and the color encodes the frequency. Different from previous visualizations that focus on the sets, SAFEGUARD AI uses a tree-based visualization to showcase the regulation structure, providing a more intuitive summary of regulation compliance.

4.2.4 Risk and harm envisioning

Envisioning potential risks and harms of technology requires imagination and creativity, and researchers usually lack the training and motivation to anticipate the unintended uses of their research [283]. Previous research has introduced card-based toolkits to guide the envisioning process; for example, Value Cards [284] helps practitioners understand the social impacts of machine learning by model cards and persona cards; Envisioning Cards [285] guide designers

through the value-sensitive design process based on four criteria: stakeholders, time, values, and pervasiveness.

With the advancement of large language models (LLMs), new frameworks have emerged to envision AI's potential harms. AHA (Anticipating Harms of AI) [258] is a framework that supports AI system creators, auditors, and decision-makers in foreseeing the potential harms of AI systems. Farsight [257] is an interactive tool that helps developers identify potential harms associated with AI applications and links envisioned harms with related news articles. Explore-Gen [259] uses LLMs to envision the uses and risks of AI and categorizes the risks by the EU AI Act.

Unlike the above-mentioned envisioning tools that focus on the harms and risks of generative AI, SAFEGUARD AI leverages generative AI to enhance the validation of traditional models, such as object detectors and classifiers, which are essential components in complex AI systems.

4.3 Formative interviews

To understand the current workflow in model validation and regulation compliance, we chose autonomous driving as an example, and conducted formative interviews with two experts who work for a multi-national company's autonomous driving division. Through the interview, we understand the type of regulations they work with, and the difficulty in conducting model validation on datasets without metadata.

Regulation. Ideally, autonomous vehicles should handle different road and weather conditions like human drivers. In reality, autonomous vehicles are set to operate under conditions defined by Operational Design Domains (ODD) [286]. If the conditions are beyond ODD, the vehicle should

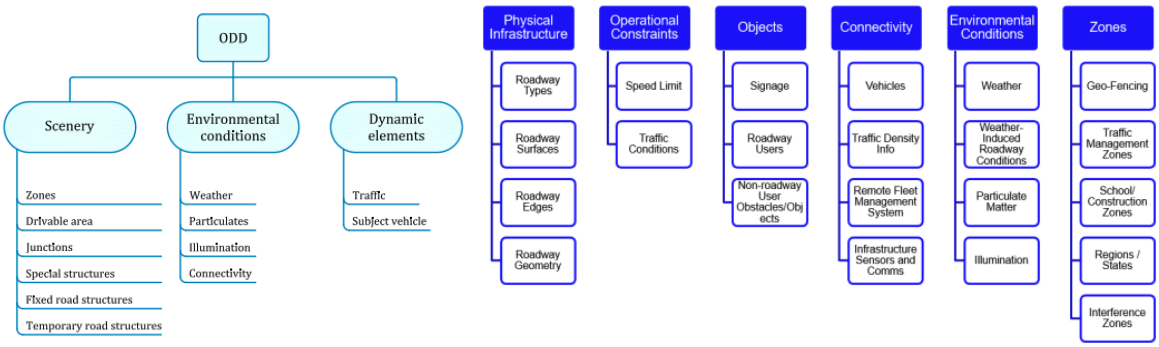


Figure 4.2: Examples of Operational Design Domains (ODD). Left: proposed by the British Standard Institute (BSI) [24], right: proposed by US Department of Transportation [25].

disengage from autonomous mode and be taken over by human drivers. Regulation agencies and car manufacturers have proposed various ODDs [24, 25, 287, 288, 289, 290]. Figure 4.2 shows two examples of ODD specifications. The left one is proposed by the British Standard Institute (BSI) [24], featuring three major categories: scenery, environmental conditions, and dynamic elements. Within each major category, there are also subcategories that car manufacturers need to test; for example, under the scenery branch, there are different zones, drivable areas, junctions, special structures, fixed road structures, and temporary road structures. The right one is proposed by the US Department of Transportation, including six categories: physical infrastructure, operational constraints, objects, connectivity, environmental conditions and zones. Similarly, within each category, there are also sub-categories. The combinations of different sub-categories are potential data slices used in model validation, for example, drivable area = ”divided single lane”; special structures = ”bridge”, weather = ”snowy”; traffic = ”moderate”. However, enumerating all possible combinations will result in combinatorial explosions, and it’s infeasible to examine all combinations.

Perception models. An autonomous driving system starts with perceiving its surroundings with sensors, including cameras, LiDAR, radar, and ultrasound. With the advancement of machine

learning, cameras are playing an increasingly important role, even replacing sensors of other modalities [291]. Visual detectors need to identify and localize objects from the camera feed; however, object detection failure is one of the major causes of autonomous vehicle accidents: for example, in several Tesla crashes, the object detector failed to detect trucks [292, 293]. According to the California Department of Motor Vehicles (DMV) [294], object detection failures are also reasons safety drivers took over autonomous vehicles. Based on our interviews with autonomous driving experts, such driving videos collected from test vehicles are usually not annotated, and the entire dataset could contain tens of thousands of driving videos.

Novel data slice discovery. Though an ODD has covered extensive conditions, it can not foresee all conditions in practice. For example, during our interview, experts mentioned that even the weather is clear, if an autonomous vehicle happens to follow a dump truck, the dust could cause problems for the visual object detector. Besides, the road structures are also evolving. For example, nowadays many parking lots have equipped with electric-vehicle (EV) charging stations and EV-only parking spots, which are not reflected in ODD documents, but has an impact on autonomous parking systems.

4.4 User tasks and system input

4.4.1 User tasks

We distilled the user tasks of SAFEGUARD AI by drawing inspiration from our interviews with domain experts in autonomous driving, and also previous slice-finding [254, 275, 282] and risk-envisioning tools [257].

T1. Get an overview of model performance against a regulation. Though existing slice

discovery methods support users to identify problematic data slices, such slices are characterized by metadata, not by regulations. Therefore, SAFEGUARD AI aims to present data slices based on regulations, thereby bridging the gap between model validation and regulatory compliance. Note that the overview should include both the performance and the data slice sizes, as we can only trust a data slice's performance when it has good support.

T2. Support novel data slice discovery. Validating models against known regulations and metadata is insufficient. First, the regulation can not enumerate all categories exhaustively, and safety experts must interpret it themselves; second, even if safety experts know which regulation label is important, there may be unknown factors that influence the models' performance. For instance, the experts do not know that "following a dump truck" could deteriorate model performance until they saw such video frames. Therefore, SAFEGUARD AI needs to support the discovery of novel data slices to better prepare safety experts for regulatory compliance and to discover new ways to improve the models.

4.4.2 System input

Here we introduce the system input required to achieve the above two tasks: regulatory labels, model predictions, and evaluation metrics.

Regulatory labels. We extracted regulation labels from external documents, particularly those issued by regulatory agencies, e.g., the *Operational Design Domain (ODD) taxonomy for an automated driving system (ADS)* issued by the British Standards Institution [24]. Those labels are organized as tree-like hierarchies, for instance, "Sunny" and "Cloudy" are grouped under the broader category "Environmental Conditions."(Fig 4.2).

However, regulatory documents only provide examples of labels under each tree branch. To enhance coverage, we also introduce *brainstormed labels*, for scenarios that are not explicitly listed in the regulations, such as "electric vehicle (EV) charging stations" or "following a dump truck."

Model predictions. In SAFEGUARD AI, we focus on object detection results on images. Therefore, for each image, we require a unique Image ID, a list of ground truth annotations (including bounding boxes and object names), and predictions (including bounding boxes, object names, and confidence scores). For each image, we calculate True Positives (correct detections), False Positives (incorrect detections), and False Negatives (missed detections) based on an Intersection over Union (IoU) threshold of 0.5.

Data slices. Once we obtained regulatory labels and model predictions, we ran DivExplorer [252] to get data slices. A data slice is a subset of the dataset that has divergent model performance compared to the rest of the data. Each slice is associated with the following attributes:

- Slice ID
- Slice name: typically a single label or a combination of labels (e.g., "Sunny + High-Traffic-Volume")
- Support: the size of the slice
- Average Precision (AP):

$$\text{AP} = \sum_{n=1}^{N} (r_n - r_{n-1})\, p_n \tag{4.1}$$

  where N = 11, r_n and r_{n-1} are the recall at points n and $n-1$ and p_n is the precision at point n.

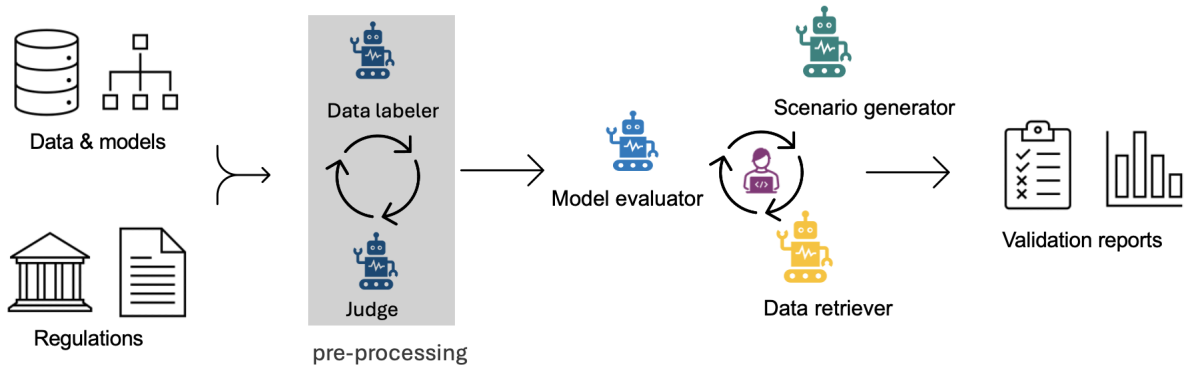


Figure 4.3: An overview of the multi-agent workflow of SAFEGUARD AI. A multi-agent system helps transform data, models, and regulations into validation reports.

It is important to note that different data slices can overlap. Some slices are subsets of others; for example, the slice "Sunny + High-Traffic-Volume" overlaps with the slice "Sunny + Line-Marker", and "Sunny + High-Traffic-Volume" is a subset of "Sunny."

4.5 Multi-agent workflows

To support **T1** and **T2**, we designed a multi-agent workflow involving several LLM-based agents as shown in Figure 4.3.

4.5.1 Data labeler and judge

In the absence of regulation-specific metadata, we implemented a novel two-stage annotation pipeline. The first stage employs an image-to-text generation prompt informed by methods in open-vocabulary image segmentation [295]. We evaluated multiple state-of-the-art vision-language models, including InternVL [296], Llama 3.2 [297], BLIP [298], CogVLM [299], and Llava [300], using the prompt: *"Please describe the image in detail, including all the visible objects."* We conducted systematic sampling of generated captions across multiple vision-language

models, evaluating them based on object completeness, scene understanding, and environmental condition recognition. We found that Llama 3.2 and InternVL 1.5 demonstrated superior performance, ultimately, Llama 3.2 was selected for implementation due to its convenient API.

The second stage extracts regulation-relevant labels via a structured prompt: *Read the image description carefully, and select applicable words from this list* {*input*}. *Image description:* {*captions*}. *Return your answer as a list of labels: [..., ...]"* This enables systematic extraction of domain-specific attributes; for instance, from a caption describing a steel bridge covered with snow, it will extract bridge (fixed road structures) and snowy (weather condition). For this extraction phase, we utilize GPT-4o, although direct image processing with this model was avoided due to cost considerations.

To validate annotations, we implement a judge agent using visual question answering techniques. This agent processes queries such as *"Does this image contain a bridge?"* to corroborate the extracted labels. However, the judge agent demonstrated variable performance in different images, suggesting that discriminative visual question answering presents greater challenges than generative caption production, potentially due to the need for more precise spatial and contextual reasoning.

4.5.2 Model evaluator and scenario generator

After obtaining regulation metadata from our annotation pipeline, we developed evaluator agents that extend beyond traditional slice discovery algorithms [252] by incorporating scenario criticality. While these algorithms identify performance divergence, they neglect domain-specific importance. For example, "crossroad" is usually considered more important than "straight road",

so a model that performs worse on the crossroad will be more concerning.

Our framework first isolates problematic slices through established discovery methods, then employs a dual-agent system to assess scenario criticality. These agents analyze regulations and scenarios through structured N-round discussions, producing both quantitative rankings and qualitative rationales. This approach yields a comprehensive assessment that balances statistical performance with regulatory significance.

The scenario generator subsequently analyzes evaluation results to hypothesize additional vulnerable conditions, prompting the data retriever to collect relevant examples. If the data retriever has successfully retrieved enough data, the model evaluator will aggregate the results of the new slice and present them.

4.5.3 Data retriever

The data retriever processes natural language queries from the scenario generator to identify relevant images. The initial evaluation revealed CLIP performance degradation with ambiguous queries. Therefore, we implemented a data retriever to decide whether it's necessary to refine the current query, and if so, generate a new query. Given the new query, we implemented two complementary embedding approaches: CLIP-based [301] text-to-image retrieval and text-to-text matching using GPT embeddings [302]. For text-based retrieval, we match query embeddings against caption embeddings. To address threshold selection challenges, we employ an adaptive percentile-based approach selecting the top $X\%$ of matches, allowing users to adjust precision-recall trade-offs through the interface. Comparative evaluation using precision@10 metrics demonstrates the effectiveness of our retrieval approach, as shown in Figure 4.4.

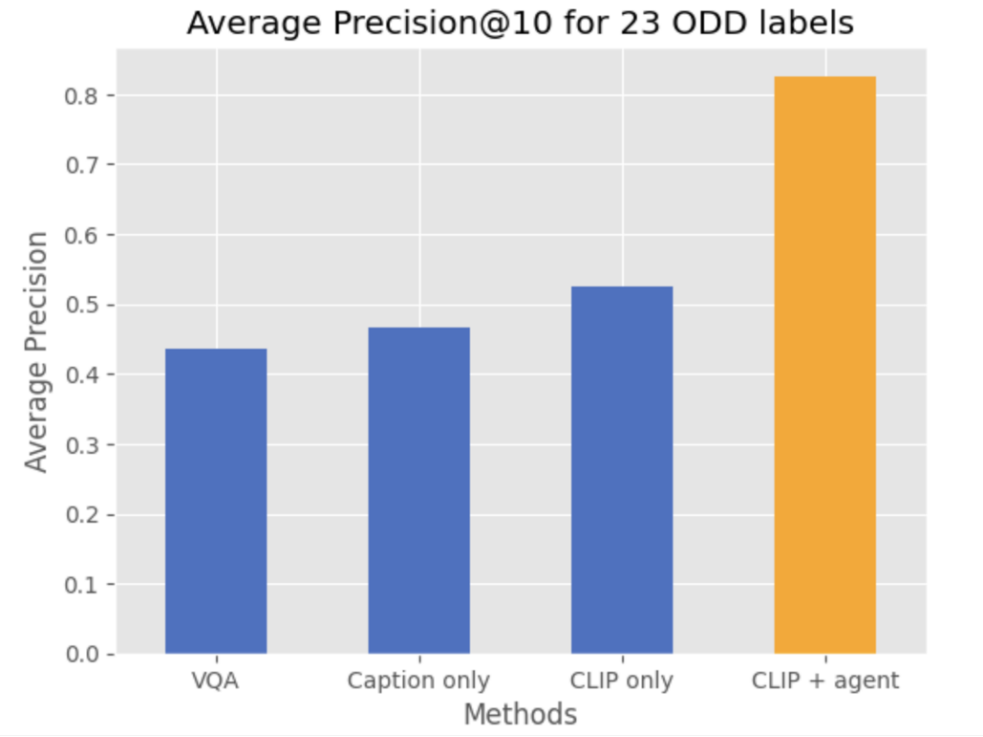


Figure 4.4: Average Precision@10 for 23 ODD labels on NuScenes [26], a large scale self-driving dataset. CLIP + Agent performs best.

4.6 Interface

We implemented a visual analytics system powered by the above multi-agent workflow. We implemented a visual analytics system powered by the multi-agent workflow described above. First, users can specify the regulation document, the object detection model, and the object of interest through dropdown menus. The system then presents an Average Precision (AP) vs. Support scatterplot (1), where each dot represents a slice. Since the scatterplot can become highly overlapped, we also provide a grid cell plot (2), where each cell corresponds to a slice and is color-coded by AP score: the lower the AP score, the darker the shade of red.

Because the grid cell view does not display support information, we implemented a slice view inspired by SliceTeller [254] (3). In this view, each row represents a slice, each column cor-

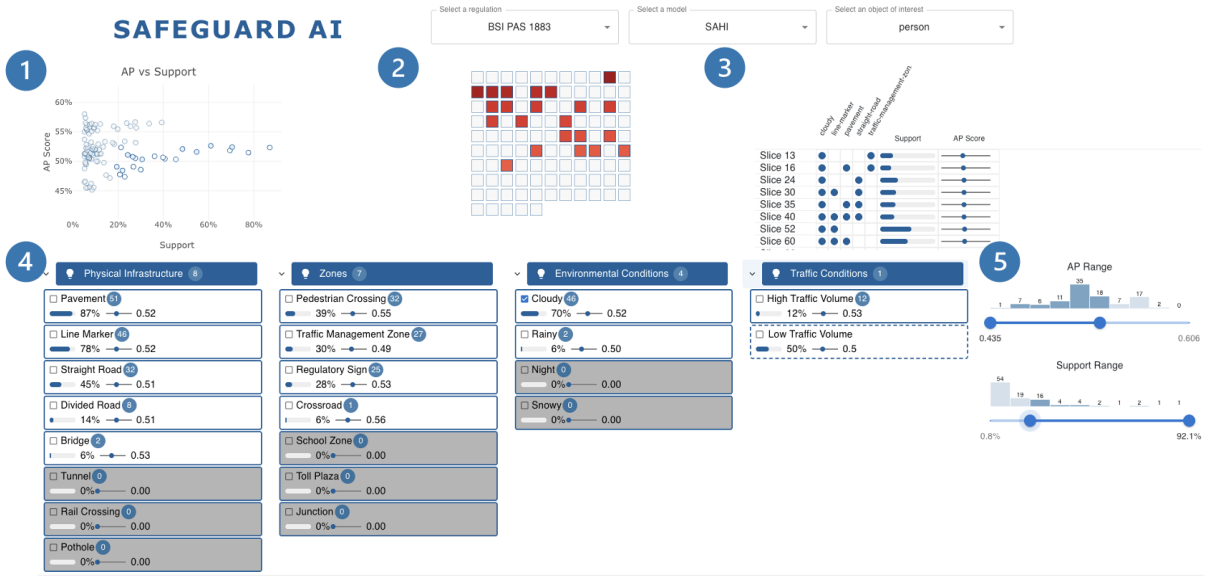


Figure 4.5: Safeguard AI User Interface. (1) Average Precision (AP) vs. support scatterplot, (2) grid-cell view, (3) slice view, (4) tree view, (5) Average Precision (AP) and support range sliders.

responds to a label, and a dot within the grid indicates the presence of that label. Two additional columns at the end show the support and AP scores for each slice.

We also provide a tree view (4) that mirrors the hierarchical structure of the regulation document. Unlike the slice view, the tree view aggregates performance metrics at the label level: the support and AP scores associated with each label are calculated by aggregating the corresponding metrics from all slices that include that label. Labels with insufficient data are grayed out. Additionally, each tree branch includes a light-bulb icon that supports brainstorming for new labels. For example, under the ”Traffic Conditions” branch, the label ”Low Traffic Volume” is highlighted with dashed lines to indicate it’s brainstormed by agents. Finally, users can configure AP and support thresholds (5) using range sliders to filter and focus on slices that meet specific criteria.

4.7 Preliminary evaluations

Since the safety experts needed to evaluate confidential model results within their organization, we conducted a preliminary evaluation using publicly available datasets and regulatory documents. This allowed us to assess the feasibility and effectiveness of the SAFEGUARD AI system without exposing proprietary data.

Preparation. For the regulatory framework, we used BSI-PAS 1883 [24], a widely recognized document released by the British Standards Institute that outlines Operational Design Domain (ODD) taxonomies for autonomous driving. As our dataset, we selected NuScenes [26], a large-scale urban driving dataset collected in Singapore and Boston. We filtered this dataset to include all front-camera images containing pedestrians, resulting in a subset of approximately 10,000 images. For the object detection model, we used SAHI [303, 304], a state-of-the-art detection framework optimized for identifying small objects like pedestrians. These model outputs were then processed using our multi-agent workflow, and the resulting predictions and metadata were loaded into the SAFEGUARD AI system.

Goals. The primary goal of this evaluation was to determine whether SAFEGUARD AI could fulfill the compliance analysis needs of safety experts. Specifically, we wanted to assess whether the system's features, workflows, and outputs would support regulatory validation tasks. If the evaluation proved successful, our collaborators could integrate their proprietary data into SAFEGUARD AI without requiring major modifications. To this end, we invited two safety experts who had participated in earlier formative interviews to provide feedback.

Procedures. Given the limited time and availability of safety experts, we opted for a guided

system walkthrough rather than a hands-on user study. The goal was to demonstrate core functionalities, elicit feedback, and explore whether the system's capabilities aligned with real-world compliance workflows. Each session followed the same structure. We introduced the public datasets and regulatory documents used in the demo. We then presented the SAFEGUARD AI interface. We demonstrated several key tasks: identifying underperforming data slices within specific regulation categories, adjusting sliders to filter slices by support or performance, and brainstorming new regulation labels using the UI. We held a 30-minute session with Expert 1 and a 60-minute session with Expert 2. After each demonstration, we asked them whether the current system supported their analysis needs. We also invited them to suggest additional features or improvements. Their feedback is summarized below:

SAFEGUARD AI effectively shows slice performance. Both experts were impressed by the tree-like diagram, which mirrored the structure of regulatory documents they work with, and by the cross-linked visualizations that allowed them to customize slices of interest. Compared to SliceTeller [254], which struggles with matrix sparsity issues, SAFEGUARD AI is more scalable by leveraging scatterplots and grid cell views to present a large number of slices.

Gap between data visualization and compliance reporting. Although SAFEGUARD AI provides effective interactive visualizations, it currently lacks textual explanations that could be directly incorporated into formal compliance reports. One expert suggested that, beyond visual analytics, SAFEGUARD AI should also generate actionable insights in textual form to streamline regulatory reporting workflows.

Expand the variety of supported metrics. In the current implementation, SAFEGUARD AI displays two metrics: support and AP, following conventions from previous work on data slice

analysis [254]. However, one expert noted that in her subdomain, additional metrics are often used, such as Car-to-Car Rear Stationary (CCRs)—a collision scenario where a moving vehicle strikes the rear of a stationary vehicle [305]. She suggested allowing users to define and add custom evaluation metrics.

Model performance with perturbations. Though SAFEGUARD AI shows the performance of images of different scenarios, one expert mentions that he's also interested in knowing how the change of conditions will influence a model's performance, e.g., if an image's environmental conditions changes from sunny to rainy, while keeping other surroundings the same. However, such analysis requires specialized data collection, which is not supported in our demo.

Chapter 5: CONCLUSION

5.1 Characteristics of Problems

This thesis describes three problems that different knowledge workers face during human-AI interaction. Those problems are unified by the shared dimensions of Gulf of Evaluation, and also the problem complexity and users' domain expertise.

Shared dimensions of Gulf of Evaluation. The commonality of the three problems lies in the Gulf of Evaluation: inaccuracies and lack of trust. In TUTOAI, tutorial creators aim to develop mixed-media tutorials for physical tasks that better meet the needs of their audience, as video tutorials are not easily browsable, while static text instructions can be too abstract. However, off-the-shelf machine learning models are not sufficient to automatically generate mixed-media tutorials, as they suffer from inaccuracies, and the pipelines of multiple models often result in compounding errors. In COALA, communication researchers seek to understand the collaborative writing behaviors of native and non-native English speakers through action sequences. While many AI-assisted event sequence analysis tools could cluster and mine patterns from such data, they often lack interpretability, which prevents researchers from trusting the results. In SAFEGUARD AI, safety experts need to determine whether they can trust AI models in high-risk scenarios such as autonomous driving. To do so, they must thoroughly evaluate model inaccuracies,

particularly performance issues in scenarios relevant to regulatory compliance.

Problem complexity and domain expertise. Beyond the shared challenges of inaccuracy and lack of trust, these users also face complex input sources and output formats that cannot be addressed by simply prompting an LLM or relying on a single machine learning module. For instance, as demonstrated in TUTOAI, although an LLM might be able to extract steps from a video transcript, it cannot extract visual objects from video frames or build interactive dependency diagrams; as shown in SAFEGUARD AI, although existing slice discovery algorithms could evaluate model performance, they are not aligned with required regulations.

Another reason these tasks cannot be easily addressed with automatic methods is that the users are domain experts. Tutorial creators, communication researchers, and safety experts each have their own contextual knowledge and professional judgment, which makes them less inclined to rely on AI-generated results blindly. As we observed in the user studies, tutorial creators in TUTOAI carefully refined AI-generated text descriptions; communication researchers in COALA override results generated by clustering algorithms.

5.2 Characteristics of Systems

To help bridge the Gulf of Evaluation for human-AI interaction for knowledge workers, I developed three systems: TUTOAI, COALA, and SAFEGUARD AI. Though those systems serve different purposes for different knowledge workers, they share common design guidelines and techniques.

Design guidelines. Although only TUTOAI explicitly proposed a set of design guidelines for developing human-AI co-creation systems, both COALA and SAFEGUARD AI were developed

following similar guidelines—particularly: "simplify complex creation by guiding user actions," "separate content from style," and "user-centric model selection."

Techniques to reduce inaccuracies. We developed several techniques aimed at reducing inaccuracies. For example, in TUTOAI, we assembled multiple ML models into pipelines and leveraged stronger models to guide weaker ones. While no machine learning model is perfect, the SAFEGUARD AI project demonstrates that we can still evaluate model performance thoroughly by analyzing data slices—and ensure safer deployment by restricting models to scenarios where their accuracy is deemed acceptable. While no machine learning model is perfect, the SAFEGUARD AI project demonstrates that we can still evaluate model performance thoroughly by analyzing data slices—and ensure safer deployment by restricting models to scenarios where their accuracy is deemed acceptable.

Techniques to foster trust. To foster trust, we incorporated several techniques across systems. In COALA, we augmented pattern mining results with textual explanations, redesigned visualizations to improve interpretability, and displayed consensus across multiple clustering methods. Additionally, interactive user interfaces that support exploratory data analysis help build user trust—an approach evident in both COALA and SAFEGUARD AI.

Interaction techniques. Although each system features a distinct user interface, they share several reusable interaction components. All begin with an overview interface: in TUTOAI, creators begin with a video player showing steps and timestamps; in COALA, communication researchers start with a cluster view of all authors; in SAFEGUARD AI, safety experts begin with a tree view of regulation labels. Additionally, all systems allow users to control the number of visible artifacts using sliders—for instance, the slider in TUTOAI that adjusts the number of

recommended frames, or the slider in SAFEGUARD AI that filters data slices.

Generalizability. While these systems are tailored for specific use cases, they can be adapted for related domains. For instance, beyond generating mixed-media tutorials, TUTOAI could serve as a video annotation or segmentation tool. Although COALA was originally developed for analyzing collaborative writing action sequences, it can be applied to other event sequences, as long as a clear taxonomy is available. Similarly, SAFEGUARD AI could be extended to analyze regulatory documents in other domains, provided those documents follow a hierarchical structure.

5.3 Future work

While this thesis involved diverse knowledge workers, they were all domain experts—for example, tutorial creators familiar with their topics, communication researchers with expertise in their datasets, and safety experts experienced with regulatory frameworks. However, we did not explore scenarios involving entry-level knowledge workers, such as individuals tasked with demonstrating regulatory compliance without prior experience. Prior research suggests that such people may adopt unexpected strategies when engaging with such tasks [306].

Another promising direction is to support monitoring dynamic multi-agent interactions. The systems in this thesis focused primarily on presenting AI-generated outputs to users for evaluation. Although we utilized multi-agent components like scenario generators, model evaluators, and data retrievers, our work emphasized the results rather than the interaction process between agents. With the increasing adoption of autonomous multi-agent systems [307, 308], future human-centered AI systems should aim to support knowledge workers not only in reviewing outputs but also in monitoring and steering the ongoing agent interactions [309].

APPENDIX - TUTOAI

Table 5.1: Summarized Objects in Mixed-media Tutorials by Human Roles

Human Role	**Topic**	**Source Count**
Create manually	General [66], Cooking [55, 65], Lecture [91]	4
No intervention	Software [56]	1
Refine computational results	Cooking [53, 64]	2

Table 5.2: The Roles of Human in Extracting Steps in Mixed-media Tutorials

Human Role	**Topic**	**Source Count**
Create manually	General [52, 65, 66], Cooking [55], Lecture [59]	5
No intervention	General [310], Software [56, 57], Makeup [54]	4
Provide input for computation	General [58], Software [67]	2
Refine computational results	Cooking [53, 64], Lecture [63]	3

Table 5.3: Summarized Dependencies in Mixed-media Tutorials

Human Role	**Topic and Relation**	**Source Count**
Create manually	Cooking: food processing order [53, 55], spatial relations [55]; Lecture: concept prerequisites [91]	4
No intervention	Makeup: spatial relations [54]	1
Refine/Input for computational methods	Cooking: food processing order [64]	1

(a) Successful case

(b) Failure case

(c) Successful case not in ground-truth

Figure 5.1: Image grounding examples returned by GPT-3 + OWL-ViT, an open-vocabulary detector. Green box: human annotation; red: returned by OWL-ViT. (a) GPT3: "fish sauce"; ground-truth: "sauce"; (b) GPT3: "salt"; ground-truth: "salt"; (c): GPT3: "2 pounds of chicken cutlets"; ground-truth: "chicken"; though the IOU is 0, it's a correct detection.

Model evaluation details

Dataset: YouCook2 [104] comprising 2000 untrimmed cooking videos with human annotations, averaging 5.27 minutes in length and containing 3-16 steps per video. Each step is annotated with the start time, end time, and text descriptions. The dataset splits are training (67%), validation (23%), and testing (10%). Only the training and validation sets have object annotations (bounding boxes and labels). Since some models were pre-trained on the training subset, we exclusively utilize the validation set. After filtering for auto-generated English transcripts, 347 videos remain. Auto-generated transcripts for each video were sourced from the YouTube API [311].

Extractive methods necessitate the parameter step count, K. For consistent benchmarking, we set K as the ground truth steps of each video. For LexRank [93] and TextRank [94], we used

Sumy's implementation [312]. For LLM prompting, we prompted GPT-3 with "summarize the recipe in K steps". For BART [313] and T5 [96], we used the HuggingFace [46] implementation of both methods with default parameters.

Method	Type	ROUGE-1	ROUGE-2	ROUGE-L
LexRank	extractive	0.25	0.06	0.21
TextRank	extractive	0.25	0.06	0.20
BART	abstractive	0.22	0.03	0.19
T5	abstractive	0.19	0.02	0.16
GPT-3	abstractive	**0.37**	**0.12**	**0.31**

Table 5.4: Average ROUGE (Recall-Oriented Understudy for Gisting Evaluation) F1 scores of different summarization methods.

ROUGUE-n scores measure the overlap of n-grams between generated and ground-truth summaries, and ROUGE-L is the Longest Common Subsequence (LCS)-based statistics.

For pipeline 2, we abstained from gauging the efficacy of shot boundary detection methods in extracting step thumbnails since no ground-truth thumbnails exist, and shot boundary detection yields multiple frame candidates.

For video dense captioning, we assume the ground-truth step timestamps are known. Since the goal of dense captioning is not summarization, but scene description, we do not compute ROUGE scores but manually inspect the results. After reviewing segment descriptions from a sample of 20 videos, errors are evident 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".

For POS taggers, we designated words labeled as NN, NNS, NNP, or NNPS [314] as nouns. For traditional object detector, we chose faster-r-CNN trained on Visual Genomes due to benefit

[1] https://www.youtube.com/watch?v=-k7trpuj3X8

from the large number of object categories. We down-sampled videos to one frame every 10 seconds and retained detected objects with confidence scores above 0.4, selecting the top 10 objects per frame.

Method	True positives	Label unavailable	Missing	False positives
Visual Detector [117]	2.8	2.9	4.2	43.6
POS tagging [116]	7.0	**1.1**	**1.5**	32.4
GPT-3 with prompt	**7.4**	**1.1**	**1.5**	**6.8**

Table 5.5: A quantitative comparison of object detection methods. On average, videos contain 9.6 ground-truth objects. Label unavailable: the object is not in the Visual Genome [21] dataset or is unmentioned in the transcript. Missing: fails to detect the object when the label is available. False positives: detections irrelevant to the cooking process.

Open-vocabulary detectors: we evaluated both OWL-ViT and MDETR [112]. For OWL-ViT, we provided OWL-ViT with object names extracted by GPT-3 from the transcript, among 3440 objects returned by GPT-3 that are also included in the human annotations, the mean IOU (Intersection over Union) of the ground truth bounding boxes and the predicted bounding boxes is 0.38. Examples of success and failure cases are shown in Figure 5.1. For MDETR [112], it has similar results, but the inference cost is much higher, therefore, we chose the HuggingFace implementation [119] of OWL-ViT [113].

Limitations of ML pipelines

We noticed two bottlenecks in our ML pipeline. One is the maximum number of tokens the text summarization method can take: Currently, we use GPT-3/3.5 API to process transcripts, which has a limit of 4096 tokens (a token is about 0.75 word) in a single round of conversation, e.g., both input and GPT-generated output. Empirically, that's a 10-15 min instructional video's transcript length and summarized steps. Fortunately, we see progress in this area, e.g., the newly

released GPT-4 supports at most 32768 tokens [315].

Another bottleneck is the open-vocabulary object detector. In the user studies, AI-generated bounding boxes received the lowest quality scores from participants. As the vision-language model is still an emerging research area, we expect the results to improve steadily in the future.

We also noticed the hallucination problems of LLM, e.g., it generates details like "4 eggs" and "all-purpose flour" when the transcript only mentions "eggs" and "flour". Other factors also influence step summarization quality, including automatic speech recognition (ASR) errors, shown in Table 5.7.

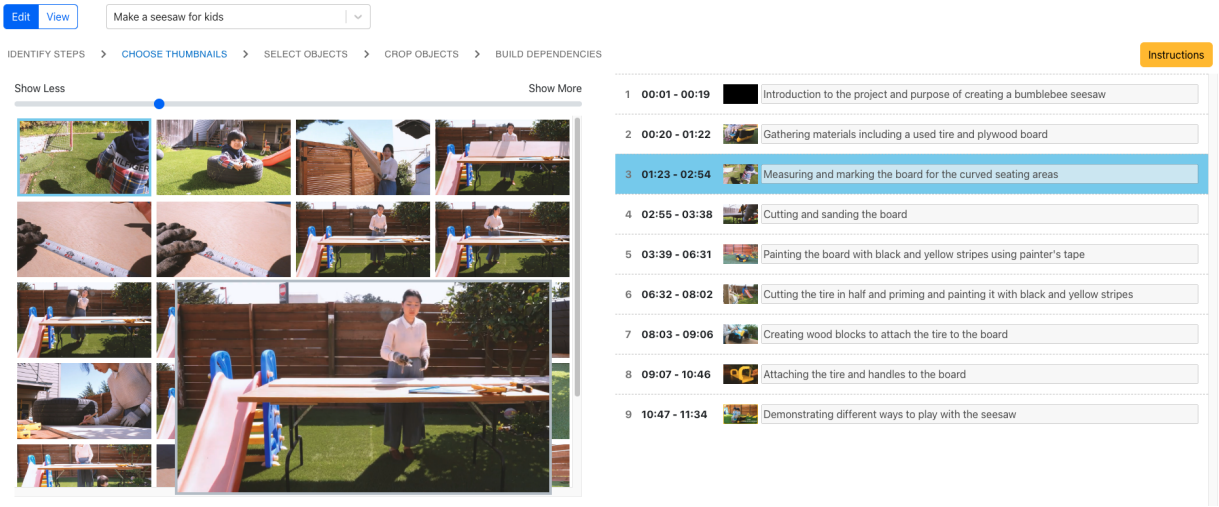


Figure 5.2: Choose thumbnails. The goal is to choose a representative image for each step. On the left are video frames selected by TutoAI. Hovering over a frame will show an enlarged version. Creators can control the number of displayed frames by dragging the slider toward "show more"/"show less." On the right are steps (now the editing is disabled).

Generality of TutoAI

We showed TutoAI's consistent performance in instructional videos across domains via user studies, including cooking, furniture assembly, craft, and vehicle. Unlike previous work which focus specifically on a single domain [53, 54], TutoAI has demonstrated its versatility

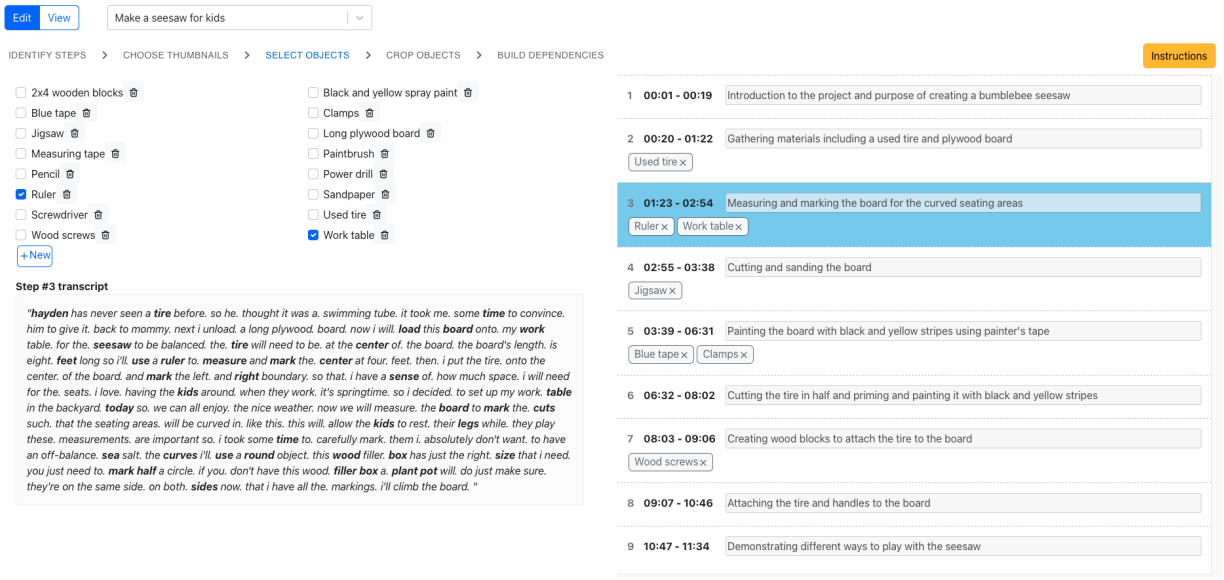


Figure 5.3: Select objects. The goal is to associate objects with each step to build a dependency between steps in later stages. Creators can add and delete objects in each step, add new objects, and delete objects for the entire video

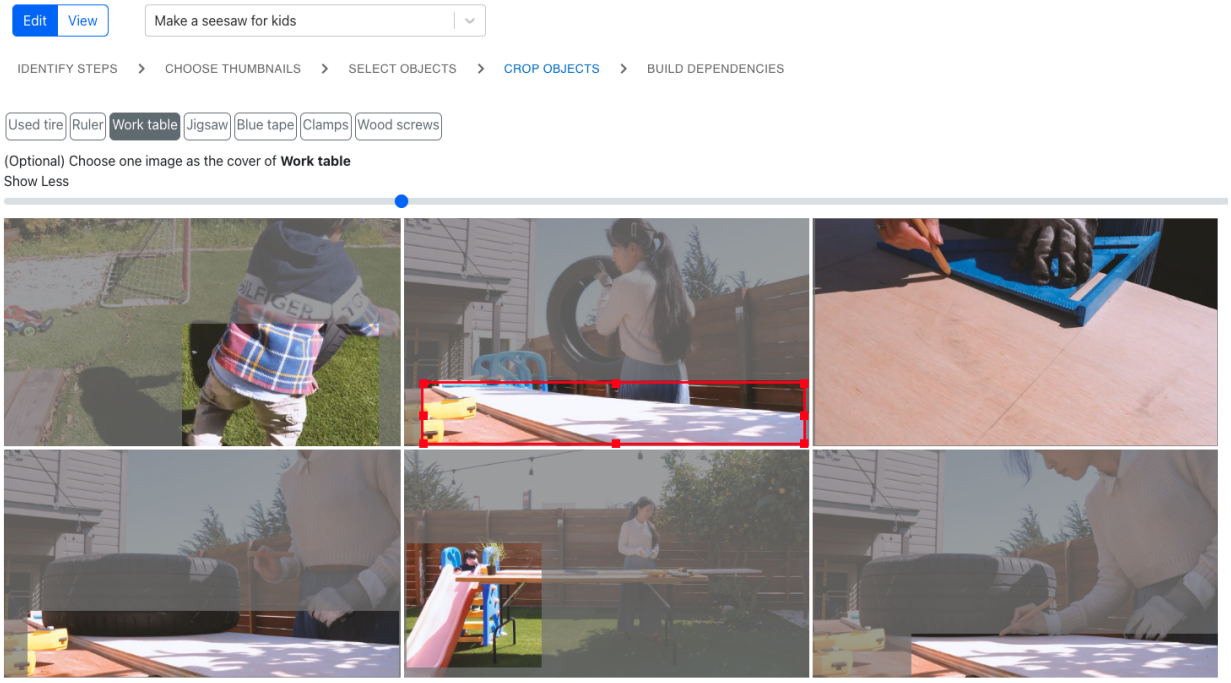


Figure 5.4: Crop objects. The goal is to provide object images for less common objects. Here, it shows recommended images for the "work table." Once an image is selected, creators can adjust the bounding boxes

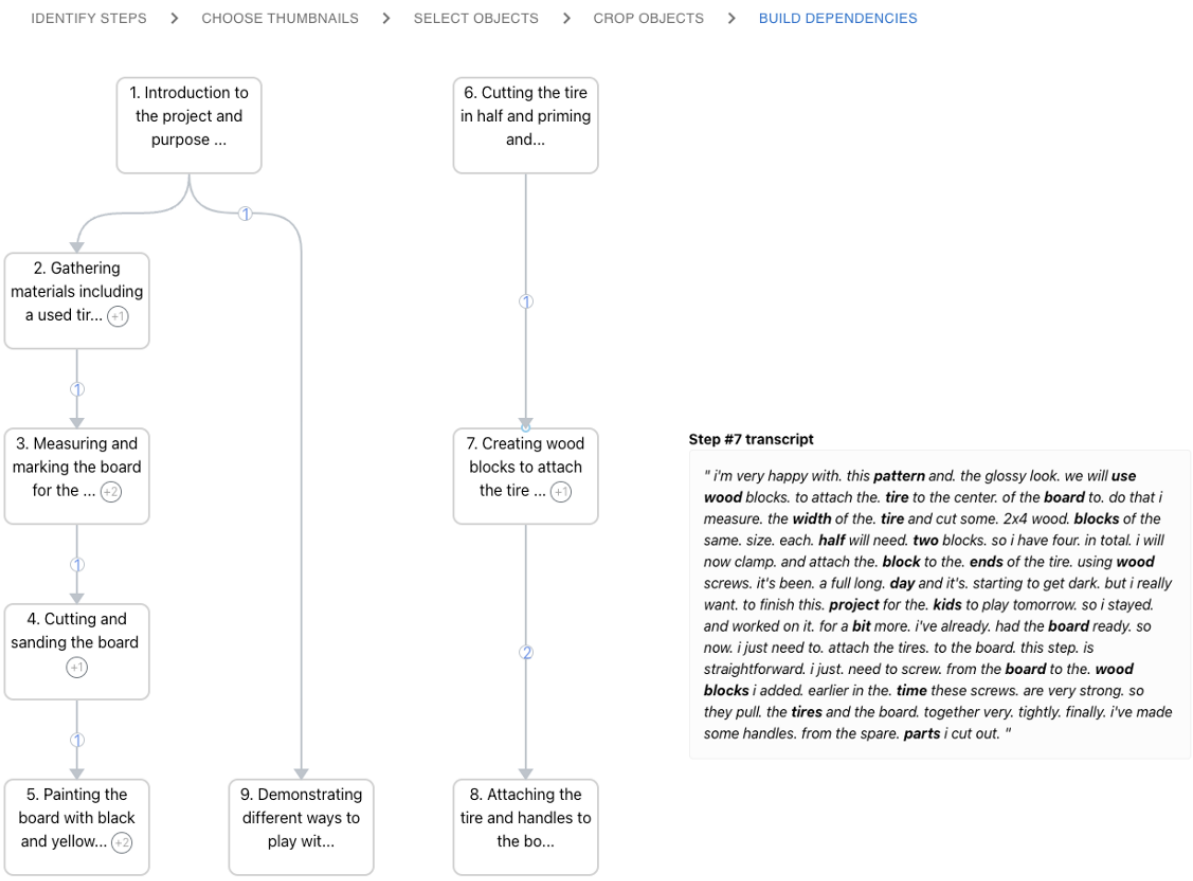


Figure 5.5: Build dependencies. The goal is to build dependencies between steps so consumers can easily skip and split tasks. To add new dependencies, creators start a new arrow from a step and connect the arrow to another. To delete a dependency, drag the arrow away from a step and release. To help creators recall the content of each step, hovering over a step will display its transcript at the bottom right.

empowered by LLMs and vision-language models.

Method	Error types	Examples	Video ID
Visual detector	label unavailable	“dough”	4K9h7ojJYkc
Visual detector	missing	“shrimp”	GXnzgRC3sd4
Visual detector	wrong	mistook “pan” for “bowl”	tGaAAI3aAUs
Visual detector	false positive	“necklace”	abfhnSaZFlA
POS Tagging/GPT-3	label unavailable	“wok”	eWBSMD3BiHM
POS Tagging	missing	“chickpea”	R5IAGR2SeaE
POS Tagging	wrong	“soy sauce”=¿ “soy”, “sauce”	ntiGX3X-spA
POS Tagging	false positive	“minutes”	tYg3lQ5aZv8
GPT-3	missing	“water”	jEo9VXYVrxs
GPT-3	wrong	“Cat cat spices”	luDzsPatsGw
GPT-3	false positive	“Clean hands”	7-FatJyHj_g

Table 5.6: Error examples in object detection methods

Error types	Examples	Video ID
ASR error	“...put off the plane” should be “put off the flame”	ikmPrpgWQ5M
object/action unmentioned	“here’s an egg put that in there” (didn’t mention the “bowl”)	TF1iWaX2-DM
video-text discrepancy	talk about animal welfare while chopping a cabbage	Z5bpo2sBsl8

Table 5.7: Other error types that influence text summary quality

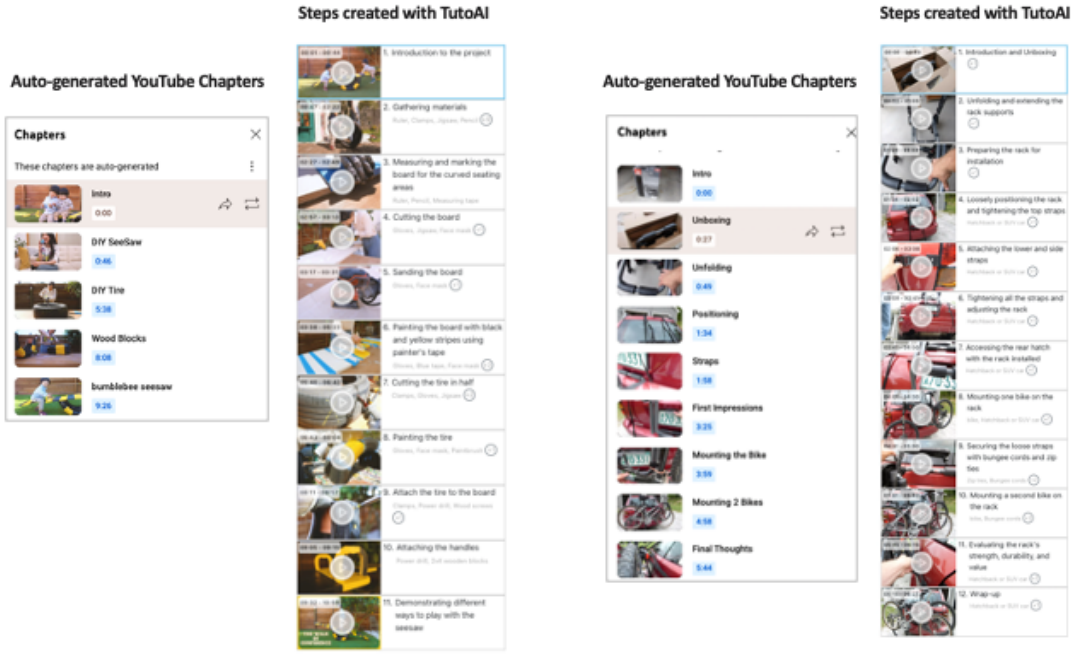
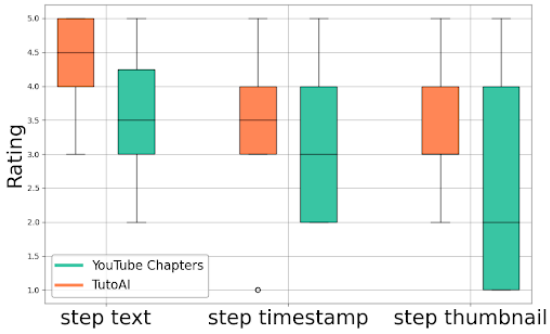
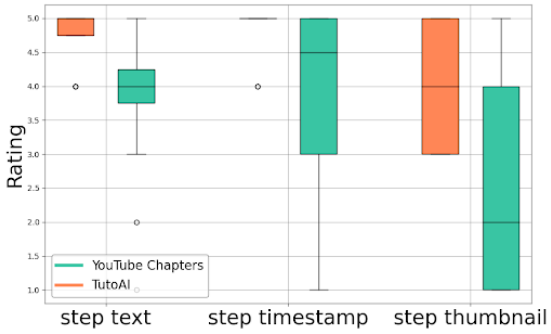


Figure 5.6: YouTube auto-generated chapters vs. TutoAI steps created by original authors



(a) Before editing: components quality comparison. 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)



(b) After editing: components usefulness comparison. 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)

Figure 5.7: Component quality of group B: strawberry blueberry shortcakes. Group B. Before editing (left), after editing (right)

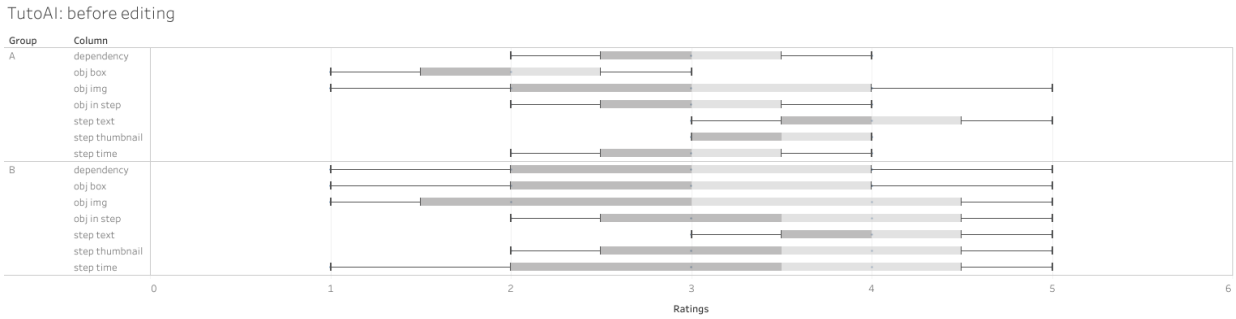


Figure 5.8: Before editing: TutoAI components quality. Group A: office chair assembly, Group B: strawberry blueberry shortcakes

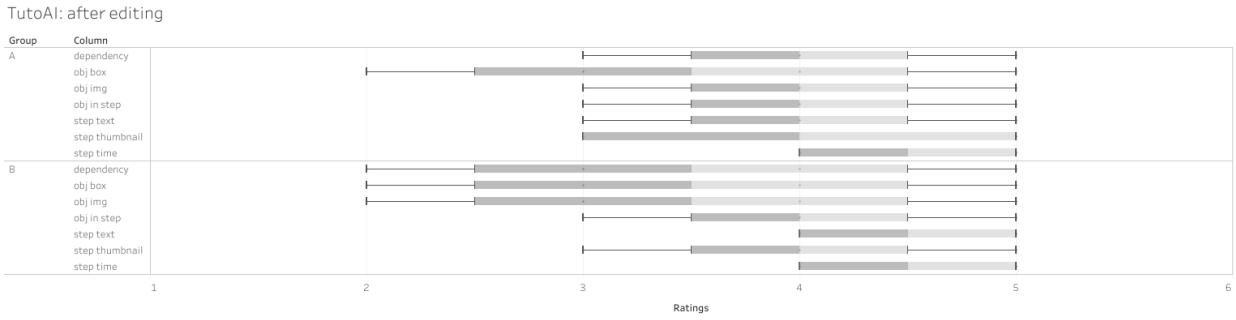


Figure 5.9: After editing: TutoAI components usefulness. Group A: office chair assembly, Group B: strawberry blueberry shortcakes

Table 5.8: Steps in mixed-media tutorials (images used with permission)

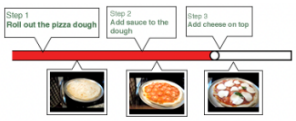
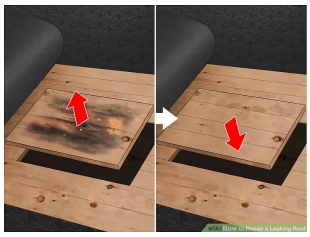
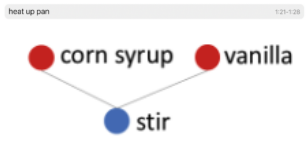
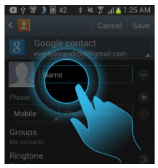
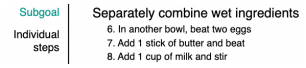
Topic	Source	Format	Human roles
General	ToolScape [58]		input for computational methods
	YouTube chapters [316]		create from scratch or NA
	WikiHow [92]		create from scratch
Cooking	videoWhiz [53]		refine computational results
	Yang et al [55]		create from scratch
	RecipeDeck [64]		refine computational results
Software	Fraser et al [56]		NA
	mixT [57]		NA
	EverTutor [67]		input for computational methods
Makeup	Truong et al [54]		NA
Lecture	Video Digests [63]		refine computational results
	Crowdy [59]		create from scratch

Table 5.9: Objects in mixed-media tutorials (images used with permission)

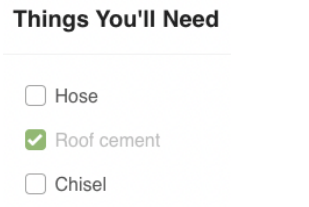
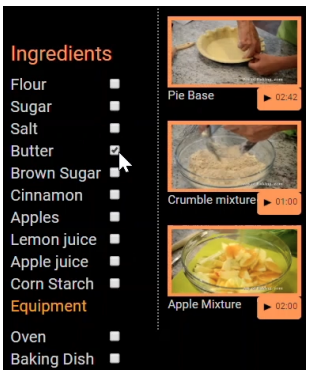
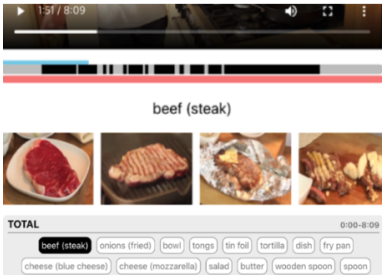
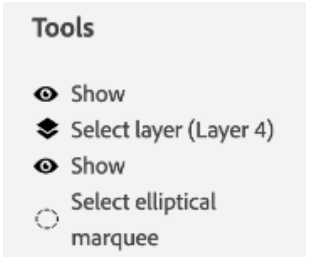
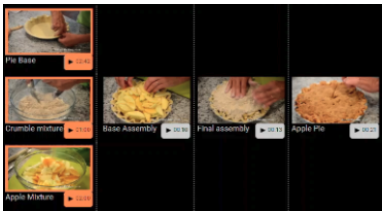
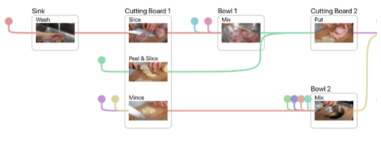
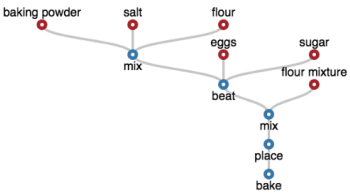
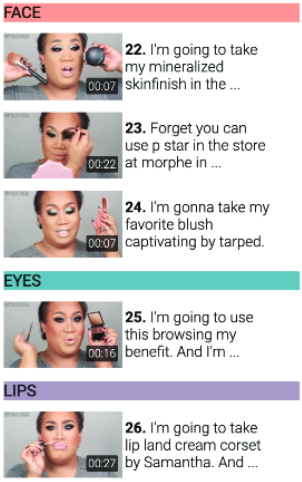
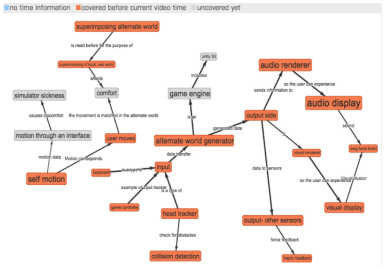
Topic	Source	Format	Human roles
General	WikiHow [66]		create from scratch
Cooking	videoWhiz [53]		refine computational results
	Yang et al. [55]		create from scratch
	RecipeDeck [64]	text list	refine computational results
Software Lecture	Fraser et al. [56]		NA
	ConceptScape [91]	text buttons	create from scratch

Table 5.10: Dependencies in mixed-media tutorials (images used with permission)

Topic	Source	Format	Relation	Human roles
Cooking	videoWhiz [53]		cooking order	create from scratch
	Yang et al. [55]		spatial relations/cooking order	create from scratch
	RecipeDeck [64]		cooking order	refine/input for computational methods
Makeup	Truong et al [54]		spatial relations	NA
Lecture	ConceptScape [91]		concept prerequisites	create from scratch

APPENDIX - COALA

Understanding document modification behaviors

In this section, we focus on understanding document modification behaviors E_{on}. Such behaviors could be inferred by comparing consecutive document versions via the Myers difference algorithm [317].

Task Analysis

In our interviews with communication researchers, there are two tasks about analyzing document modification behaviors:

T1: identify editing dynamics. Communication researchers hypothesize that NS and NNS contribute differently to the joint document and would like to know the difference reflected by the evolution of documents.

T2: analyze specific content. Communication researchers are interested in finding frequently co-edited content and seeing how NS and NNS edit them during collaborative writing.

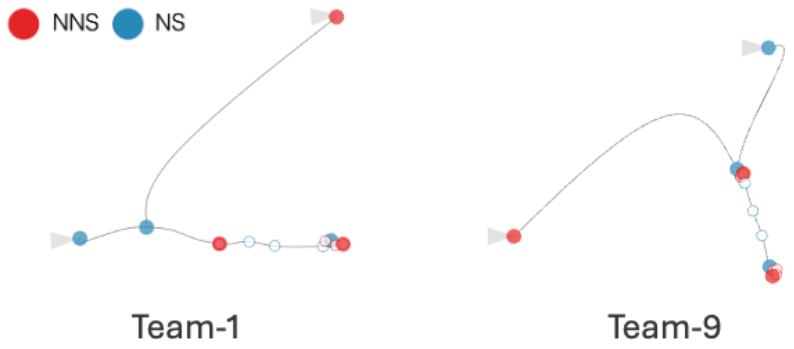


Figure 5.10: Branched time curves of team-1 and team-9. The gray arrows indicate the first versions written by NS and NNS. Big and filled dots are major versions, and small and hollow dots are intermediate versions.

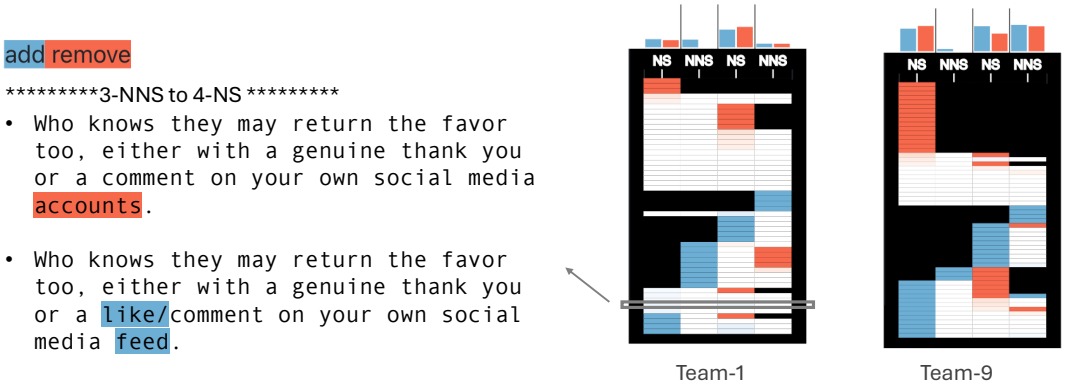


Figure 5.11: Sentence Flow of team-1 and team-9. Red: remove, blue: add, gray: edited by both authors.

Understanding editing dynamics by branched time curves

To tackle **T1**, we draw inspirations from the folded time curve proposed by Bach et al. [136], where each dot represents a document version, the order on the curve denotes the temporal information, and the pairwise distance depends on the text similarity. We proposed a modified version as depicted in Figure 5.10: since authors write individually before collaborating on the same document, we adopted a two-head start, indicated by the gray arrows, which refer to the initial draft written by NS and NNS. Then, we use red and blue to refer to NNS and NS' contributions, respectively. The communication researchers suggest highlighting the differences between intermediate document versions V_{inter} and end-of-the-day versions V_{end}, so we apply big filled dots for V_{end} and small hollow dots for V_{inter}.

The branched time curves give an overview of the document's evolution, especially the "V" shape part, which shows how the merged version differs from NNS and NS' initial draft. Inspired by the proximity of the merged version to NS' version, in a recently published paper at a premier social computing conference (not cited here for anonymity), our collaborators conducted a statistical significance test and found that the merged document's lexical distance was indeed significantly closer to NS' initial writing. Meanwhile, in subsequent versions, NS' edits always result in a larger lexical distance than NNS.

Tracking specific content with Sentence Flow

Though the time curves provide an overview on the document-level contributions from NS and NNS, the coarse granularity limits us from uncovering more content-specific insights (**T2**). Inspired by history flow [135, 158], we propose a revised version called Sentence Flow,

as depicted in Figure 5.11. The x-axis is the author; the y-axis are individual sentences in the document. The edits of a sentence is color-coded: white for no activity, blue and red for word-level add and remove, and black for deleting the entire sentence. The darkness of red and blue indicates the edit distance, the darker the larger. On top of the sentence flow, there are also bar charts quantify the total edit distance. If users click on a sentence, they can see the content and change logs.

Our collaborators identified two dominant editing patterns in sentence flow. One is within-author editing, where authors primarily revise their own sentences and rarely modify others', e.g., in Team 1, adjacent colored areas are uncommon, with only one instance shown in Figure 5.11, where the NS made minor modifications to NNS' sentences. The second pattern is between-author editing, as seen in Team 9 in Figure 5.11, where several sentences were consecutively revised by different authors, resulting in a more balanced distribution.

Though the modified versions of existing visualizations reveal document modification behaviors of NS and NNS, it does not answer questions about how content are generated in collaborative writing. To address these questions, we conduct and present an event sequence analysis of content generation behaviors in the next section.

Bibliography

[1] Edwin L Hutchins, James D Hollan, and Donald A Norman. Direct manipulation interfaces. *Human–computer interaction*, 1(4):311–338, 1985.

[2] Donald A Norman and Stephen W Draper. *User centered system design; new perspectives on human-computer interaction*. L. Erlbaum Associates Inc., 1986.

[3] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. Guidelines for human-ai interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–13, 2019.

[4] Ellen Jiang, Edwin Toh, Alejandra Molina, Kristen Olson, Claire Kayacik, Aaron Donsbach, Carrie J Cai, and Michael Terry. Discovering the syntax and strategies of natural language programming with generative language models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2022.

[5] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 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*, pages 1–22, 2022.

[6] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *European conference on computer vision*, pages 340–353. Springer, 2012.

[7] Ramesh Simhambhatla, Kevin Okiah, Shravan Kuchkula, and Robert Slater. Self-driving cars: Evaluation of deep learning techniques for object detection in different driving conditions. *SMU Data Science Review*, 2(1):23, 2019.

[8] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

[9] Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*, 2023.

[10] Brian D Janz, Jason A Colquitt, and Raymond A Noe. Knowledge worker team effectiveness: The role of autonomy, interdependence, team development, and contextual support variables. *Personnel psychology*, 50(4):877–904, 1997.

[11] Allison Woodruff, Renee Shelby, Patrick Gage Kelley, Steven Rousso-Schindler, Jamila Smith-Loud, and Lauren Wilcox. How knowledge workers think generative ai will (not) transform their industries. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–26, 2024.

[12] Alex Singla, Alexander Sukharevsky, Lareina Yee, Michael Chui, and Bryce Hall. The state of ai: Global survey — mckinsey, 3 2025. [Online; accessed 2025-04-24].

[13] Yuexi Chen, Vlad I Morariu, Anh Truong, and Zhicheng Liu. Tutoai: a cross-domain framework for ai-assisted mixed-media tutorial creation on physical tasks. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2024.

[14] Yuexi Chen, Yimin Xiao, Kazi Tasnim Zinat, Naomi Yamashita, Ge Gao, and Zhicheng Liu. Comparing native and non-native english speakers' behaviors in collaborative writing through visual analytics. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–16. ACM, April 2025.

[15] Yimin Xiao, Yuewen Chen, Naomi Yamashita, Yuexi Chen, Zhicheng Liu, and Ge Gao. (dis) placed contributions: Uncovering hidden hurdles to collaborative writing involving non-native speakers, native speakers, and ai-powered editing tools. *Proceedings of the ACM on Human-Computer Interaction*, 8(CSCW2):1–31, 2024.

[16] Yuexi Chen and Zhicheng Liu. Worddecipher: Enhancing digital workspace communication with explainable ai for non-native english speakers. In *Proceedings of the Third Workshop on Intelligent and Interactive Writing Assistants*, pages 7–10, 2024.

[17] Yuexi Chen, Zhicheng Liu, Christopher Tensmeyer, Niklas Elmqvist, and Vlad I Morariu. Docdancer: Authoring ultra-responsive documents with layout generation. In *2023 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 133–138. IEEE, 2023.

[18] Vlad Ion Morariu, CHEN Yuexi, Christopher Alan Tensmeyer, Zhicheng Liu, and Lars Niklas Emanuel Elmqvist. Responsive document authoring, March 5 2024. US Patent 11,922,110.

[19] Tim sweeney on x: "lost another court verdict, climbed another mountain. the world has come a long way since 2020 when this journey began, with much progress achieved by many people in many nations around the world. and onward we go! https://t.co/qlwhcbunys" / x, 9 2021. [Online; accessed 2025-04-22].

[20] Carol S Dweck. *Mindset: The new psychology of success*. Random house, 2006.

[21] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[22] Yuanzhe Chen, Panpan Xu, and Liu Ren. Sequence synopsis: Optimize visual summary of temporal event data. *IEEE transactions on visualization and computer graphics*, 24(1):45–55, 2017.

[23] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229, 2019.

[24] The British Standards Institution. Operational design domain (odd) taxonomy for an automated driving system (ads) – specifi cation. `https://www.bsigroup.com/globalassets/localfiles/en-gb/cav/pas1883.pdf`, 8 2020. (Accessed on 08/27/2024).

[25] Eric Thorn, Shawn C Kimmel, Michelle Chaka, Booz Allen Hamilton, et al. A framework for automated driving system testable cases and scenarios. Technical report, United States. Department of Transportation. National Highway Traffic Safety . . . , 2018.

[26] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[27] Artificial intelligence (ai) coined at dartmouth — dartmouth. [Online; accessed 2025-04-24].

[28] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

[29] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[30] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[32] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[33] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[34] Chris Garcia. Harold cohen and aaron—a 40-year collaboration. *Computer History Museum*, 23, 2016.

[35] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[36] Midjourney. Midjourney, 2021.

[37] JD Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2023.

[38] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[39] The shift from models to compound ai systems – the berkeley artificial intelligence research blog. `https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/`, 2 2024. (Accessed on 02/20/2024).

[40] Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D Manning, and Daniel E Ho. Hallucination-free? assessing the reliability of leading ai legal research tools. *Journal of Empirical Legal Studies*, 2024.

[41] Sara Merken. Ai 'hallucinations' in court papers spell trouble for lawyers — reuters, 2 2025. [Online; accessed 2025-04-28].

[42] Georgiana Juravle, Andriana Boudouraki, Miglena Terziyska, and Constantin Rezlescu. Trust in artificial intelligence for medical diagnoses. *Progress in brain research*, 253:263–282, 2020.

[43] Michael Terry, Chinmay Kulkarni, Martin Wattenberg, Lucas Dixon, and Meredith Ringel Morris. Ai alignment in the design of interactive ai: Specification alignment, process alignment, and evaluation support. *arXiv preprint arXiv:2311.00710*, 2023.

[44] Figma: The collaborative interface design tool. `https://www.figma.com/`, 2 2024. (Accessed on 02/20/2024).

[45] The latest in machine learning — papers with code. `https://paperswithcode.com/`, 2 2024. (Accessed on 02/20/2024).

[46] Hugging Face. Hugging face: The ai community building the future, 2023.

[47] React. `https://react.dev/`, 2 2024. (Accessed on 02/20/2024).

[48] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309, 2011.

[49] Welcome to flask — flask documentation (3.0.x). `https://flask.palletsprojects.com/en/3.0.x/`, 2 2024. (Accessed on 02/20/2024).

[50] Yaxi Zhao, Razan Jaber, Donald McMillan, and Cosmin Munteanu. "rewind to the jiggling meat part": Understanding voice control of instructional videos in everyday tasks. In *CHI Conference on Human Factors in Computing Systems*, pages 1–11, 2022.

[51] Sylvaine Tuncer, Barry Brown, and Oskar Lindwall. 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*, pages 1–12, 2020.

[52] Sarah Perez. Youtube introduces video chapters to make it easier to navigate longer videos, 2020.

[53] Megha Nawhal, Jacqueline B Lang, Greg Mori, and Parmit K Chilana. Videowhiz: Non-linear interactive overviews for recipe videos. In *Graphics Interface*, pages 15–1, 2019.

[54] Anh Truong, Peggy Chi, David Salesin, Irfan Essa, and Maneesh Agrawala. Automatic generation of two-level hierarchical tutorials from instructional makeup videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2021.

[55] Saelyne Yang, Sangkyung Kwak, Tae Soo Kim, and Juho Kim. Improving video interfaces by presenting informational units of videos. *CHI'22 Extended Abstracts. Association for Computing Machinery*, 2022.

[56] C Ailie Fraser, Joy O Kim, Hijung Valentina Shin, Joel Brandt, and Mira Dontcheva. Temporal segmentation of creative live streams. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[57] Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. Mixt: automatic generation of step-by-step mixed media tutorials. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 93–102, 2012.

[58] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J Guo, Robert C Miller, and Krzysztof Z Gajos. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 4017–4026, 2014.

[59] Sarah Weir, Juho Kim, Krzysztof Z Gajos, and Robert C Miller. Learnersourcing subgoal labels for how-to videos. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pages 405–416, 2015.

[60] Bryan Wang, Meng Yu Yang, and Tovi Grossman. 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*, pages 1–14, 2021.

[61] Alan Blackwell and Thomas Green. Notational systems–the cognitive dimensions of notations framework. *HCI models, theories, and frameworks: toward an interdisciplinary science. Morgan Kaufmann*, 234, 2003.

[62] Gaëlle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Laurent Bouillon, and Jean Vanderdonckt. A unifying reference framework for multi-target user interfaces. *Interacting with computers*, 15(3):289–308, 2003.

[63] Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala. Video digests: a browsable, skimmable format for informational lecture videos. In *UIST*, volume 10, pages 2642918–2647400. Citeseer, 2014.

[64] Minsuk Chang, Léonore V Guillain, Hyeungshik Jung, Vivian M Hare, Juho Kim, and Maneesh Agrawala. Recipescape: An interactive tool for analyzing cooking instructions at scale. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, 2018.

[65] Dotdash Meredith. Allrecipes, 2023.

[66] wikihow. Welcome to wikihow, the most trusted how-to site on the internet, 2023.

[67] Cheng-Yao Wang, Wei-Chen Chu, Hou-Ren Chen, Chun-Yen Hsu, and Mike Y Chen. Evertutor: Automatically creating interactive guided tutorials on smartphones by user demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 4027–4036, 2014.

[68] Katy Ilonka Gero, Vivian Liu, and Lydia Chilton. Sparks: Inspiration for science writing using language models. In *Designing Interactive Systems Conference*, pages 1002–1019, 2022.

[69] Ryan Louie, Andy Coenen, Cheng Zhi Huang, Michael Terry, and Carrie J Cai. 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*, pages 1–13, 2020.

[70] Matthew Guzdial, Nicholas Liao, Jonathan Chen, Shao-Yu Chen, Shukan Shah, Vishwa Shah, Joshua Reno, Gillian Smith, and Mark O Riedl. 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*, pages 1–13, 2019.

[71] 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. 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*, pages 1–18, 2022.

[72] Hai Dang, Karim Benharrak, Florian Lehmann, and Daniel Buschek. Beyond text generation: Supporting writers with continuous automatic text summaries. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pages 1–13, 2022.

[73] Alison Smith, Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater. 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*, pages 293–304, 2018.

[74] Jane Hoffswell and Zhicheng Liu. Interactive repair of tables extracted from pdf documents on mobile devices. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[75] 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. 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*, pages 1–14, 2019.

[76] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166, 1999.

[77] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[78] 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. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[79] Learn Prompting. Your guide to communicating with artificial intelligence, 2023.

[80] Gregor Betz, Kyle Richardson, and Christian Voigt. Thinking aloud: Dynamic context generation improves zero-shot reasoning performance of gpt-2. *arXiv preprint arXiv:2103.13033*, 2021.

[81] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021.

[82] Tanya Goyal, Junyi Jessy Li, and Greg Durrett. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*, 2022.

[83] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*, 2023.

[84] Xianjun Yang, Yan Li, Xinlu Zhang, Haifeng Chen, and Wei Cheng. Exploring the limits of chatgpt for query or aspect-based text summarization. *arXiv preprint arXiv:2302.08081*, 2023.

[85] Md Tahmid Rahman Laskar, Xue-Yong Fu, Cheng Chen, and Shashi Bhushan TN. Building real-world meeting summarization systems using large language models: A practical perspective. *arXiv preprint arXiv:2310.19233*, 2023.

[86] Ioana Croitoru, Simion-Vlad Bogolin, Samuel Albanie, Yang Liu, Zhaowen Wang, Seunghyun Yoon, Franck Dernoncourt, Hailin Jin, and Trung Bui. Moment detection in long tutorial videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2594–2604, 2023.

[87] Chuyi Shang, Emi Tran, Medhini Narasimhan, Sanjay Subramanian, Dan Klein, and Trevor Darrell. Luse: Using llms for unsupervised step extraction in instructional videos. *https://cveu.github.io/2023/papers/36.pdf*, 2023.

[88] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[89] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

[90] YouTube. Youtube, 2023.

[91] Ching Liu, Juho Kim, and Hao-Chuan Wang. Conceptscape: Collaborative concept mapping for video learning. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, 2018.

[92] David Bitan. How to repair a leaking roof, 2022.

[93] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

[94] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[95] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[96] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[97] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4565–4574, 2016.

[98] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8739–8748, 2018.

[99] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6847–6857, 2021.

[100] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275, 2017.

[101] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. 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*, pages 1247–1257, 2019.

[102] Cristian Rodriguez-Opazo, Edison Marrese-Taylor, Basura Fernando, Hongdong Li, and Stephen Gould. 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*, pages 1079–1088, 2021.

[103] Tomáš Souček and Jakub Lokoč. Transnet v2: an effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*, 2020.

[104] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[105] Meta AI. Video summarization, 2022.

[106] Wencheng Zhu, Jiwen Lu, Jiahao Li, and Jie Zhou. Dsnet: A flexible detect-to-summarize network for video summarization. *IEEE Transactions on Image Processing*, 30:948–962, 2020.

[107] Bo He, Jun Wang, Jielin Qiu, Trung Bui, Abhinav Shrivastava, and Zhaowen Wang. Align and attend: Multimodal summarization with dual contrastive losses. *arXiv preprint arXiv:2303.07284*, 2023.

[108] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5179–5187, 2015.

[109] Meta AI. Video summarization, 2022.

[110] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[111] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34:26183–26197, 2021.

[112] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021.

[113] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pages 728–755. Springer, 2022.

[114] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022.

[115] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

[116] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 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)*, pages 54–59, 2019.

[117] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[118] OpenAI. Introducing chatgpt, 2022.

[119] Hugging Face. Hugging face transformers: Owl-vit, 2022.

[120] Minsuk Chang, Mina Huh, and Juho Kim. Rubyslippers: Supporting content-based voice navigation for how-to videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2021.

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

[122] OpenAI. Gpt-4v(ision) system card, 2023.

[123] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[124] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

[125] Wanrong Zhu, Bo Pang, Ashish V Thapliyal, William Yang Wang, and Radu Soricut. End-to-end dense video captioning as sequence generation. *arXiv preprint arXiv:2204.08121*, 2022.

[126] Wikipedia contributors. Wizard (software) — Wikipedia, the free encyclopedia, 2023. [Online; accessed 21-November-2023].

[127] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. Clip-it! language-guided video summarization. *Advances in Neural Information Processing Systems*, 34:13988–14000, 2021.

[128] Rui Cheng. A non-native student's experience on collaborating with native peers in academic literacy development: A sociopolitical perspective. *Journal of English for Academic Purposes*, 12(1):12–22, 2013.

[129] Weipeng Yang, Yongyan Li, and Hui Li. Supervisor as coauthor in writing for publication: Evidence from a cohort of non-native english-speaking master of education students. *SN Social Sciences*, 1(2):44, 2021.

[130] Scott A Hale. Multilinguals and wikipedia editing. In *Proceedings of the 2014 ACM conference on Web science*, pages 99–108, 2014.

[131] Atlassian. Confluence — your remote-friendly team workspace — atlassian. `https://www.atlassian.com/software/confluence`, 9 2024. (Accessed on 09/09/2024).

[132] N Ann Chenoweth and John R Hayes. Fluency in writing: Generating text in l1 and l2. *Written communication*, 18(1):80–98, 2001.

[133] Mark Wolfersberger. L1 to l2 writing process and strategy transfer: A look at lower proficiency writers. *TESL-EJ*, 7(2):1–12, 2003.

[134] Kozue Uzawa and Alister Cumming. Writing strategies in japanese as a foreign language: Lowering or keeping up the standards. *Canadian Modern Language Review*, 46(1):178–194, 1989.

[135] Fernanda B Viégas, Martin Wattenberg, and Kushal Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 575–582, 2004.

[136] Benjamin Bach, Conglei Shi, Nicolas Heulot, Tara Madhyastha, Tom Grabowski, and Pierre Dragicevic. Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE transactions on visualization and computer graphics*, 22(1):559–568, 2015.

[137] Yi Han, Agata Rozga, Nevena Dimitrova, Gregory D Abowd, and John Stasko. Visual analysis of proximal temporal relationships of social and communicative behaviors. In *Computer Graphics Forum*, volume 34, pages 51–60. Wiley Online Library, 2015.

[138] Sana Malik, Fan Du, Megan Monroe, Eberechukwu Onukwugha, Catherine Plaisant, and Ben Shneiderman. Cohort comparison of event sequences with balanced integration of visual analytics and statistics. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pages 38–49, 2015.

[139] Jason Chuang, Daniel Ramage, Christopher Manning, and Jeffrey Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 443–452, 2012.

[140] Angelos Chatzimparmpas, Rafael Messias Martins, Ilir Jusufi, Kostiantyn Kucher, Fabrice Rossi, and Andreas Kerren. The state of the art in enhancing trust in machine learning models with the use of visualizations. In *Computer Graphics Forum*, volume 39, pages 713–756. Wiley Online Library, 2020.

[141] Paul Pu Liang, Yiwei Lyu, Gunjan Chhablani, Nihal Jain, Zihao Deng, Xingbo Wang, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multiviz: Towards user-centric visualizations and interpretations of multimodal models. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2023.

[142] Fumeng Yang, Zhuanyi Huang, Jean Scholtz, and Dustin L Arendt. How do visual explanations foster end users' appropriate trust in machine learning? In *Proceedings of the 25th international conference on intelligent user interfaces*, pages 189–201, 2020.

[143] Douglas C Engelbart. Collaboration support provisions in augment. In *Proceedings of the 1984 AFIPS Office Automation Conference (Los Angeles CA, 1984, 51-58 p.)*, 1984.

[144] Robert S Fish, Robert E Kraut, and Mary DP Leland. Quilt: A collaborative tool for cooperative writing. In *Proceedings of the ACM SIGOIS and IEEECS TC-OA 1988 conference on Office information systems*, pages 30–37, 1988.

[145] Ronald M Baecker, Dimitrios Nastos, Ilona R Posner, and Kelly L Mawby. The user-centered iterative design of collaborative writing software. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 399–405, 1993.

[146] Paul Dourish and Victoria Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 107–114, 1992.

[147] Hee-Cheol Kim and Kerstin Severinson Eklundh. Reviewing practices in collaborative writing. *Computer Supported Cooperative Work (CSCW)*, 10:247–259, 2001.

[148] Overleaf. Overleaf, online latex editor. `https://www.overleaf.com`, 9 2024. (Accessed on 09/10/2024).

[149] Judith S Olson, Dakuo Wang, Gary M Olson, and Jingwen Zhang. How people write together now: Beginning the investigation with advanced undergraduates in a project course. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(1):1–40, 2017.

[150] Jeremy Birnholtz, Stephanie Steinhardt, and Antonella Pavese. Write here, write now! an experimental study of group maintenance in collaborative writing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 961–970, 2013.

[151] Dakuo Wang, Haodan Tan, and Tun Lu. Why users do not want to write together when they are writing together: Users' rationales for today's collaborative writing practices. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):1–18, 2017.

[152] So Yeon Park and Sang Won Lee. Why "why"? the importance of communicating rationales for edits in collaborative writing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–25, 2023.

[153] Bahareh Sarrafzadeh, Sujay Kumar Jauhar, Michael Gamon, Edward Lank, and Ryen W White. Characterizing stage-aware writing assistance for collaborative document authoring. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3):1–29, 2021.

[154] Ida Larsen-Ledet and Henrik Korsgaard. Territorial functioning in collaborative writing: fragmented exchanges and common outcomes. *Computer Supported Cooperative Work (CSCW)*, 28:391–433, 2019.

[155] Ida Larsen-Ledet, Henrik Korsgaard, and Susanne Bødker. Collaborative writing across multiple artifact ecologies. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–14, 2020.

[156] Jodi Schneider, Alexandre Passant, and John G Breslin. Understanding and improving wikipedia article discussion spaces. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 808–813, 2011.

[157] Selen Türkay, Daniel Seaton, and Andrew M Ang. Itero: A revision history analytics tool for exploring writing behavior and reflection. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2018.

[158] Dakuo Wang, Judith S Olson, Jingwen Zhang, Trung Nguyen, and Gary M Olson. Docuviz: visualizing collaborative writing. In *Proceedings of the 33rd Annual ACM conference on human factors in computing systems*, pages 1865–1874, 2015.

[159] Fabian Flöck and Maribel Acosta. whovis: Visualizing editor interactions and dynamics in collaborative writing over time. In *Proceedings of the 24th International Conference on World Wide Web*, pages 191–194, 2015.

[160] Aniket Kittur, Bongwon Suh, Bryan A Pendleton, and Ed H Chi. He says, she says: conflict and coordination in wikipedia. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 453–462, 2007.

[161] Ignacio Perez-Messina, Claudio Gutierrez, and Eduardo Graells-Garrido. Organic visualization of document evolution. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, pages 497–501, 2018.

[162] Vilaythong Southavilay, Kalina Yacef, Peter Reimann, and Rafael A Calvo. Analysis of collaborative writing processes using revision maps and probabilistic topic models. In *Proceedings of the third international conference on learning analytics and knowledge*, pages 38–47, 2013.

[163] Fabian Flöck, David Laniado, Felix Stadthaus, and Maribel Acosta. Towards better visual tools for exploring wikipedia article development—the use case of" gamergate controversy". In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 9, pages 48–55, 2015.

[164] Johnny Torres, Sixto García, and Enrique Peláez. Visualizing authorship and contribution of collaborative writing in e-learning environments. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 324–328, 2019.

[165] Carol Severino, Jeffrey Swenson, and Jia Zhu. A comparison of online feedback requests by non-native english-speaking and native english-speaking writers. *The Writing Center Journal*, 29(1):106–129, 2009.

[166] Erving Goffman. *Forms of talk*. University of Pennsylvania Press, 1981.

[167] Khaled Karim and Hossein Nassaji. The revision and transfer effects of direct and indirect comprehensive corrective feedback on esl students' writing. *Language Teaching Research*, 24(4):519–539, 2020.

[168] Eun Young Kang. Using model texts as a form of feedback in l2 writing. *System*, 89:102196, 2020.

[169] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. Lifelines: visualizing personal histories. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 221–227, 1996.

[170] Milos Krstajic, Enrico Bertini, and Daniel Keim. Cloudlines: Compact display of event episodes in multiple time-series. *IEEE transactions on visualization and computer graphics*, 17(12):2432–2439, 2011.

[171] Krist Wongsuphasawat and David Gotz. Outflow: Visualizing patient flow by symptoms and outcome. In *IEEE VisWeek Workshop on Visual Analytics in Healthcare, Providence, Rhode Island, USA*, pages 25–28. American Medical Informatics Association, 2011.

[172] Adam Perer and Fei Wang. Frequence: Interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th international conference on Intelligent User Interfaces*, pages 153–162, 2014.

[173] Mengdie Hu, Krist Wongsuphasawat, and John Stasko. Visualizing social media content with sententree. *IEEE transactions on visualization and computer graphics*, 23(1):621–630, 2016.

[174] Zhicheng Liu, Bernard Kerr, Mira Dontcheva, Justin Grover, Matthew Hoffman, and Alan Wilson. Coreflow: Extracting and visualizing branching patterns from event sequences. In *Computer Graphics Forum*, volume 36, pages 527–538. Wiley Online Library, 2017.

[175] Krist Wongsuphasawat, John Alexis Guerra Gómez, Catherine Plaisant, Taowei David Wang, Meirav Taieb-Maimon, and Ben Shneiderman. Lifeflow: visualizing an overview of event sequences. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1747–1756, 2011.

[176] Zhicheng Liu, Yang Wang, Mira Dontcheva, Matthew Hoffman, Seth Walker, and Alan Wilson. Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. *IEEE transactions on visualization and computer graphics*, 23(1):321–330, 2016.

[177] Adam Perer and Jimeng Sun. Matrixflow: temporal network visual analytics to track symptom evolution during disease progression. In *AMIA annual symposium proceedings*, volume 2012, page 716. American Medical Informatics Association, 2012.

[178] Jian Zhao, Zhicheng Liu, Mira Dontcheva, Aaron Hertzmann, and Alan Wilson. Matrixwave: Visual comparison of event sequence data. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 259–268, 2015.

[179] Megan Monroe, Krist Wongsuphasawat, Catherine Plaisant, Ben Shneiderman, Jeff Millstein, and Sigfried Gold. Exploring point and interval event patterns: Display methods and interactive visual query. *University of Maryland Technical Report*, 80:134, 2012.

[180] Josua Krause, Adam Perer, and Harry Stavropoulos. Supporting iterative cohort construction with visual temporal queries. *IEEE transactions on visualization and computer graphics*, 22(1):91–100, 2015.

[181] Ji Qi, Vincent Bloemen, Shihan Wang, Jarke Van Wijk, and Huub Van De Wetering. Stbins: Visual tracking and comparison of multiple data sequences using temporal binning. *IEEE Transactions on visualization and computer graphics*, 26(1):1054–1063, 2019.

[182] Miriah Meyer, Tamara Munzner, and Hanspeter Pfister. Mizbee: a multiscale synteny browser. *IEEE transactions on visualization and computer graphics*, 15(6):897–904, 2009.

[183] Yuheng Zhao, Xinyu Wang, Chen Guo, Min Lu, and Siming Chen. Contextwing: Pairwise visual comparison for evolving sequential patterns of contexts in social media data streams. *Proceedings of the ACM on Human-Computer Interaction*, 7(CSCW1):1–31, 2023.

[184] Gang Wang, Xinyi Zhang, Shiliang Tang, Haitao Zheng, and Ben Y Zhao. Unsupervised clickstream clustering for user behavior analysis. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 225–236, 2016.

[185] Shunan Guo, Ke Xu, Rongwen Zhao, David Gotz, Hongyuan Zha, and Nan Cao. Eventthread: Visual summarization and stage analysis of event sequence data. *IEEE transactions on visualization and computer graphics*, 24(1):56–65, 2017.

[186] David Gotz, Jonathan Zhang, Wenyuan Wang, Joshua Shrestha, and David Borland. Visual analysis of high-dimensional event sequence data via dynamic hierarchical aggregation. *IEEE transactions on visualization and computer graphics*, 26(1):440–450, 2019.

[187] Jessica Magallanes, Tony Stone, Paul D Morris, Suzanne Mason, Steven Wood, and Maria-Cruz Villa-Uriol. Sequen-c: A multilevel overview of temporal event sequences. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):901–911, 2021.

[188] Phong H Nguyen, Rafael Henkin, Siming Chen, Natalia Andrienko, Gennady Andrienko, Olivier Thonnard, and Cagatay Turkay. Vasabi: Hierarchical user profiles for interactive visual user behaviour analytics. *IEEE transactions on visualization and computer graphics*, 26(1):77–86, 2019.

[189] Ilr scale - wikipedia. `https://en.wikipedia.org/wiki/ILR_scale`, 2 2024. (Accessed on 02/15/2024).

[190] Tom Boellstorff, Bonnie Nardi, Celia Pearce, and TL Taylor. Words with friends: Writing collaboratively online. *Interactions*, 20(5):58–61, 2013.

[191] Paul André, Robert E Kraut, and Aniket Kittur. Effects of simultaneous and sequential work structures on distributed collaborative interdependent tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 139–148, 2014.

[192] Naomi Yamashita, Andy Echenique, Toru Ishida, and Ari Hautasaari. Lost in transmittance: how transmission lag enhances and deteriorates multilingual collaboration. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 923–934, 2013.

[193] Xiaoyan Li, Naomi Yamashita, Wen Duan, Yoshinari Shirai, and Susan R Fussell. Improving non-native speakers' participation with an automatic agent in multilingual groups. *Proceedings of the ACM on Human-Computer Interaction*, 7(GROUP):1–28, 2023.

[194] Xuchao Zhang, Dheeraj Rajagopal, Michael Gamon, Sujay Kumar Jauhar, and Chang-Tien Lu. Modeling the relationship between user comments and edits in document revision. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5002–5011, 2019.

[195] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

[196] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

[197] Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[198] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[199] Jishang Wei, Zeqian Shen, Neel Sundaresan, and Kwan-Liu Ma. Visual cluster exploration of web clickstream data. In *2012 IEEE conference on visual analytics science and technology (VAST)*, pages 3–12. IEEE, 2012.

[200] Yi Guo, Shunan Guo, Zhuochen Jin, Smiti Kaul, David Gotz, and Nan Cao. Survey on visual analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):5091–5112, 2021.

[201] Kazi Tasnim Zinat, Jinhua Yang, Arjun Gandhi, Nistha Mitra, and Zhicheng Liu. A comparative evaluation of visual summarization techniques for event sequences. *arXiv preprint arXiv:2306.02489*, 2023.

[202] Kazi Tasnim Zinat, Saimadhav Naga Sakhamuri, Aaron Sun Chen, and Zhicheng Liu. A multi-level task framework for event sequence analysis. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[203] Peter J Polack Jr, Shang-Tse Chen, Minsuk Kahng, Kaya De Barbaro, Rahul Basole, Moushumi Sharmin, and Duen Horng Chau. Chronodes: Interactive multifocus exploration of event sequences. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(1):1–21, 2018.

[204] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Education India, 2016.

[205] Philippe Fournier-Viger, Cheng-Wei Wu, Antonio Gomariz, and Vincent S Tseng. Vmsp: Efficient vertical mining of maximal sequential patterns. In *Advances in Artificial Intelligence: 27th Canadian Conference on Artificial Intelligence, Canadian AI 2014, Montréal, QC, Canada, May 6-9, 2014. Proceedings 27*, pages 83–94. Springer, 2014.

[206] Sandro Vega-Pons and José Ruiz-Shulcloper. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372, 2011.

[207] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[208] Pingchuan Ma, Rui Ding, Shuai Wang, Shi Han, and Dongmei Zhang. Insightpilot: An llm-empowered automated data exploration system. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 346–352, 2023.

[209] Shih-Chieh Dai, Aiping Xiong, and Lun-Wei Ku. Llm-in-the-loop: Leveraging large language model for thematic analysis. *arXiv preprint arXiv:2310.15100*, 2023.

[210] Vision - openai api. https://platform.openai.com/docs/guides/vision. (Accessed on 02/06/2024).

[211] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *The craft of information visualization*, pages 364–371. Elsevier, 2003.

[212] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. *Visual analytics: Definition, process, and challenges*. Springer, 2008.

[213] Jeffrey Heer. Agency plus automation: Designing artificial intelligence into interactive systems. *Proceedings of the National Academy of Sciences*, 116(6):1844–1850, 2019.

[214] David Cohn, Rich Caruana, and Andrew McCallum. Semi-supervised clustering with user feedback. *Constrained clustering: advances in algorithms, theory, and applications*, 4(1):17–32, 2003.

[215] Hendrik Strobelt, Jambay Kinley, Robert Krueger, Johanna Beyer, Hanspeter Pfister, and Alexander M Rush. Genni: Human-ai collaboration for data-backed text generation. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1106–1116, 2021.

[216] Mina Lee, Katy Ilonka Gero, John Joon Young Chung, Simon Buckingham Shum, Vipul Raheja, Hua Shen, Subhashini Venugopalan, Thiemo Wambsganss, David Zhou, Emad A Alghamdi, et al. A design space for intelligent and interactive writing assistants. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–35, 2024.

[217] Zheng Zhang, Jie Gao, Ranjodh Singh Dhaliwal, and Toby Jia-Jun Li. Visar: A human-ai argumentative writing assistant with visual programming and rapid draft prototyping. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–30, 2023.

[218] Philippe Laban, Jesse Vig, Marti A Hearst, Caiming Xiong, and Chien-Sheng Wu. Beyond the chat: Executable and verifiable text-editing with llms. *arXiv preprint arXiv:2309.15337*, 2023.

[219] Hai Dang, Sven Goller, Florian Lehmann, and Daniel Buschek. Choice over control: How users write with large language models using diegetic and non-diegetic prompting. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2023.

[220] Mohi Reza, Nathan M Laundry, Ilya Musabirov, Peter Dushniku, Zhi Yuan "Michael" Yu, Kashish Mittal, Tovi Grossman, Michael Liut, Anastasia Kuzminykh, and Joseph Jay Williams. Abscribe: Rapid exploration & organization of multiple writing variations in human-ai co-writing tasks using large language models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2024.

[221] Lu Sun, Aaron Chan, Yun Seo Chang, and Steven P Dow. Reviewflow: Intelligent scaffolding to support academic peer reviewing. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, pages 120–137, 2024.

[222] Taewan Kim, Donghoon Shin, Young-Ho Kim, and Hwajung Hong. Diarymate: Understanding user perceptions and experience in human-ai collaboration for personal journaling. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2024.

[223] Mina Lee, Percy Liang, and Qian Yang. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–19, 2022.

[224] Takumi Ito, Naomi Yamashita, Tatsuki Kuribayashi, Masatoshi Hidaka, Jun Suzuki, Ge Gao, Jack Jamieson, and Kentaro Inui. Use of an ai-powered rewriting support software in context with other tools: A study of non-native english speakers. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–13, 2023.

[225] Yewon Kim, Mina Lee, Donghwi Kim, and Sung-Ju Lee. Towards explainable ai writing assistants for non-native english speakers. *arXiv preprint arXiv:2304.02625*, 2023.

[226] Cal Newport. *Deep work: Rules for focused success in a distracted world*. Hachette UK, 2016.

[227] Microsoft. Visual studio code - code editing. redefined. `https://code.visualstudio.com/`, 9 2024. (Accessed on 09/09/2024).

[228] Eva Vanmassenhove, Dimitar Shterionov, and Andy Way. Lost in translation: Loss and decay of linguistic richness in machine translation. *arXiv preprint arXiv:1906.12068*, 2019.

[229] Saif M Mohammad, Mohammad Salameh, and Svetlana Kiritchenko. How translation alters sentiment. *Journal of Artificial Intelligence Research*, 55:95–130, 2016.

[230] Petra Isenberg, Anthony Tang, and Sheelagh Carpendale. An exploratory study of visual information analysis. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1217–1226, 2008.

[231] Petra Isenberg, Danyel Fisher, Sharoda A Paul, Meredith Ringel Morris, Kori Inkpen, and Mary Czerwinski. Co-located collaborative visual analytics around a tabletop display. *IEEE Transactions on visualization and Computer Graphics*, 18(5):689–702, 2011.

[232] Anthony Christian Robinson. *Design for synthesis in geovisualization*. The Pennsylvania State University, 2008.

[233] Ying Yang, Tim Dwyer, Michael Wybrow, Benjamin Lee, Maxime Cordeil, Mark Billinghurst, and Bruce H Thomas. Towards immersive collaborative sensemaking. *Proceedings of the ACM on Human-Computer Interaction*, 6(ISS):722–746, 2022.

[234] Scott Wiener, Richard Roth, Susan Rubio, and Henry Stern. Bill text - sb-1047 safe and secure innovation for frontier artificial intelligence models act. `https://leginfo.legislature.ca.gov/faces/billNavClient.xhtml?bill_id=202320240SB1047`, 8 2024. (Accessed on 08/25/2024).

[235] European Parliament. Eu ai act: first regulation on artificial intelligence — topics — european parliament. https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence, 8 2024. (Accessed on 08/25/2024).

[236] The White House. Blueprint for an ai bill of rights — ostp — the white house. https://www.whitehouse.gov/ostp/ai-bill-of-rights/. (Accessed on 08/26/2024).

[237] National Institute of Standards and Technology. Ai risk management framework. https://www.nist.gov/itl/ai-risk-management-framework, 8 2024. (Accessed on 08/25/2024).

[238] Technical Committee : ISO/IEC JTC 1/SC 42. Iso/iec tr 24027:2021 - information technology — artificial intelligence (ai) — bias in ai systems and ai aided decision making. https://www.iso.org/standard/77607.html, 11 2021. (Accessed on 08/26/2024).

[239] Technical Committee : ISO/IEC JTC 1/SC 42. Iso/iec 5338:2023 - information technology — artificial intelligence — ai system life cycle processes. https://www.iso.org/standard/81118.html, 12 2023. (Accessed on 08/26/2024).

[240] Technical Committee : ISO/IEC JTC 1/SC 42. Iso/iec 23894:2023 - ai – guidance on risk management. https://www.iso.org/standard/77304.html, 2 2023. (Accessed on 08/26/2024).

[241] Technical Committee : ISO/IEC JTC 1/SC 42. Iso/iec tr 24028:2020 - information technology — artificial intelligence — overview of trustworthiness in artificial intelligence. https://www.iso.org/standard/77608.html, 5 2020. (Accessed on 08/26/2024).

[242] Technical Committee : ISO/IEC JTC 1/SC 42. Iso/iec 42001:2023 - ai management systems. https://www.iso.org/standard/81230.html, 12 2023. (Accessed on 08/26/2024).

[243] Technical Committee : ISO/IEC JTC 1/SC 42. Iso/iec tr 24368:2022 - information technology — artificial intelligence — overview of ethical and societal concerns. https://www.iso.org/standard/78507.html, 8 2022. (Accessed on 08/26/2024).

[244] Technical Committee : ISO/IEC JTC 1/SC 42. Iso/iec 25059:2023 - software engineering — systems and software quality requirements and evaluation (square) — quality model for ai systems. https://www.iso.org/standard/80655.html, 6 2023. (Accessed on 08/26/2024).

[245] Technical Committee : ISO/IEC JTC 1/SC 42. Iso/iec 38507:2022 - information technology — governance of it — governance implications of the use of artificial intelligence by organizations. https://www.iso.org/standard/56641.html, 4 2022. (Accessed on 08/26/2024).

[246] Defense Innovation Unit. Responsible ai guidelines. https://www.diu.mil/responsible-ai-guidelines#overview, 2022. (Accessed on 08/25/2024).

[247] Google AI. Google ai principles – google ai. https://ai.google/responsibility/principles/, 2024. (Accessed on 08/25/2024).

[248] Marios Constantinides, Edyta Bogucka, Daniele Quercia, Susanna Kallio, and Mohammad Tahaei. Rai guidelines: Method for generating responsible ai guidelines grounded in regulations and usable by (non-)technical roles. *Conference on Computer-Supported Cooperative Work & Social Computing; November 09–13; San José, Costa Rica booktitle: Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '24)*, 2024.

[249] Jessica Fjeld, Nele Achten, Hannah Hilligoss, Adam Nagy, and Madhulika Srikumar. Principled artificial intelligence: Mapping consensus in ethical and rights-based approaches to principles for ai. *Berkman Klein Center Research Publication*, (2020-1), 2020.

[250] Weixin Liang, Nazneen Rajani, Xinyu Yang, Ezinwanne Ozoani, Eric Wu, Yiqun Chen, Daniel Scott Smith, and James Zou. What's documented in ai? systematic analysis of 32k ai model cards. *arXiv preprint arXiv:2402.05160*, 2024.

[251] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. Automated data slicing for model validation: A big data-ai integration approach. *IEEE Transactions on Knowledge and Data Engineering*, 32(12):2284–2296, 2019.

[252] Eliana Pastor, Luca De Alfaro, and Elena Baralis. Looking for trouble: Analyzing classifier behavior via pattern divergence. In *Proceedings of the 2021 International Conference on Management of Data*, pages 1400–1412, 2021.

[253] Svetlana Sagadeeva and Matthias Boehm. Sliceline: Fast, linear-algebra-based slice finding for ml model debugging. In *Proceedings of the 2021 international conference on management of data*, pages 2290–2299, 2021.

[254] Xiaoyu Zhang, Jorge Piazentin Ono, Huan Song, Liang Gou, Kwan-Liu Ma, and Liu Ren. Sliceteller: A data slice-driven approach for machine learning model validation. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):842–852, 2022.

[255] Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. *arXiv preprint arXiv:2203.14960*, 2022.

[256] Scale — seal leaderboard: Visual-language understanding. https://scale.com/leaderboard/visual_language_understanding. [Online; accessed 2025-03-09].

[257] Zijie J Wang, Chinmay Kulkarni, Lauren Wilcox, Michael Terry, and Michael Madaio. Farsight: Fostering responsible ai awareness during ai application prototyping. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–40, 2024.

[258] Zana Buçinca, Chau Minh Pham, Maurice Jakesch, Marco Tulio Ribeiro, Alexandra Olteanu, and Saleema Amershi. Aha!: Facilitating ai impact assessment by generating examples of harms. *arXiv preprint arXiv:2306.03280*, 2023.

[259] Viviane Herdel, Sanja Šćepanović, Edyta Bogucka, and Daniele Quercia. Exploregen: Large language models for envisioning the uses and risks of ai technologies. *arXiv preprint arXiv:2407.12454*, 2024.

[260] Contributors to Wikimedia projects. Executive order 14110 - wikipedia, 11 2023. [Online; accessed 2025-03-10].

[261] Microsoft. Responsible ai principles and approach — microsoft ai. https://www.microsoft.com/en-us/ai/principles-and-approach, 8 2024. (Accessed on 08/26/2024).

[262] Nicholas Diakopoulos et al. Principles for accountable algorithms and a social impact statement for algorithms. https://www.fatml.org/resources/principles-for-accountable-algorithms, 8 2024. (Accessed on 08/04/2024).

[263] Governance, risk, and compliance (grc) software — sap. "https://www.sap.com/products/financial-management/grc.html". [Online; accessed 2025-03-10].

[264] Microsoft purview compliance manager — microsoft security. "https://www.microsoft.com/en-us/security/business/risk-management/microsoft-purview-compliance-manager". [Online; accessed 2025-03-10].

[265] Compliance programs - amazon web services (aws). [Online; accessed 2025-03-10].

[266] Hugging Face. Model cards. https://huggingface.co/docs/hub/en/model-cards, 8 2024. (Accessed on 08/25/2024).

[267] Untitled. [Online; accessed 2025-03-10].

[268] Ángel Alexander Cabrera, Will Epperson, Fred Hohman, Minsuk Kahng, Jamie Morgenstern, and Duen Horng Chau. Fairvis: Visual analytics for discovering intersectional bias in machine learning. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 46–56. IEEE, 2019.

[269] Github - microsoft/responsible-ai-toolbox. https://github.com/microsoft/responsible-ai-toolbox?tab=readme-ov-file, 6 2024. (Accessed on 06/09/2024).

[270] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4–1, 2019.

[271] Sérgio Jesus, Pedro Saleiro, Beatriz M Jorge, Rita P Ribeiro, João Gama, Pedro Bizarro, Rayid Ghani, et al. Aequitas flow: Streamlining fair ml experimentation. *arXiv preprint arXiv:2405.05809*, 2024.

[272] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in ai. *Microsoft, Tech. Rep. MSR-TR-2020-32*, 2020.

[273] Xiwei Xuan, Jorge Piazentin Ono, Liang Gou, Kwan-Liu Ma, and Liu Ren. Attributionscanner: A visual analytics system for metadata-free data-slicing based model validation. *arXiv preprint arXiv:2401.06462*, 2024.

[274] Greg d'Eon, Jason d'Eon, James R Wright, and Kevin Leyton-Brown. The spotlight: A general method for discovering systematic errors in deep learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1962–1981, 2022.

[275] Xiaoyu Zhang, Jorge H Piazentin Ono, Wenbin He, Liang Gou, Mrinmaya Sachan, Kwan-Liu Ma, and Liu Ren. Slicing, chatting, and refining: A concept-based approach for machine learning model validation with conceptslicer. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, pages 274–287, 2024.

[276] Jie Jeff Xu, Saahir Dhanani, Jorge Piazentin Ono, Wenbin He, Liu Ren, and Kexin Rong. Demonstration of vcr: A tabular data slicing approach to understanding object detection model performance. *VLDB 2024 Demo*, 2024.

[277] Eric Slyman, Minsuk Kahng, and Stefan Lee. Vlslice: Interactive vision-and-language slice discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15291–15301, 2023.

[278] Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister. UpSet: Visualization of intersecting sets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1983–1992, 2014.

[279] Venkatesh Sivaraman, Zexuan Li, and Adam Perer. Divisi: Interactive search and visualization for scalable exploratory subgroup analysis. *arXiv*, 2025. - do they support brainstorming? features beyond existing categories? - tabular dataset.

[280] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[281] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[282] David Munechika, Zijie J Wang, Jack Reidy, Josh Rubin, Krishna Gade, Krishnaram Kenthapadi, and Duen Horng Chau. Visual auditor: Interactive visualization for detection and summarization of model biases. In *2022 IEEE Visualization and Visual Analytics (VIS)*, pages 45–49. IEEE, 2022.

[283] Kimberly Do, Rock Yuren Pang, Jiachen Jiang, and Katharina Reinecke. "that's important, but...": How computer science researchers anticipate unintended consequences of their research innovations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2023.

[284] Hong Shen, Wesley H Deng, Aditi Chattopadhyay, Zhiwei Steven Wu, Xu Wang, and Haiyi Zhu. Value cards: An educational toolkit for teaching social impacts of machine learning through deliberation. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 850–861, 2021.

[285] Batya Friedman and David Hendry. The envisioning cards: a toolkit for catalyzing humanistic and technical imaginations. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1145–1148, 2012.

[286] Wikipedia authors. Operational design domain - wikipedia. `https://en.wikipedia.org/wiki/Operational_design_domain`, 2 2024. (Accessed on 08/27/2024).

[287] Guidelines for testing automated driving systems in canada. `https://tc.canada.ca/en/road-transportation/innovative-technologies/connected-automated-vehicles/guidelines-testing-automated-driving-systems-canada`, 8 2021. (Accessed on 08/27/2024).

[288] ASAM e.V. Asam openodd: Concept paper. `https://www.asam.net/index.php?eID=dumpFile&t=f&f=4544&token=1260ce1c4f0afdbe18261f7137c689b1d9c27576`, 2024. (Accessed on 08/27/2024).

[289] Automated Vehicle Safety Consortium et al. Avsc best practice for describing an operational design domain: Conceptual framework and lexicon. *SAE Industry Technologies Consortia*, 2020.

[290] AG Mercedes-Benz. Introducing drive pilot: An automated driving system for the highway, 2019.

[291] Tesla. Tesla vision update: Replacing ultrasonic sensors with tesla vision — tesla support. `https://www.tesla.com/support/transitioning-tesla-vision`, 4 2024. (Accessed on 08/27/2024).

[292] Khoa Lam. Incident 218: Tesla on autopilot crashed into flipped truck on taiwan highway. `https://incidentdatabase.ai/cite/218/`, 6 2020. (Accessed on 08/27/2024).

[293] Trisha Thadani, Rachel Lerman, Imogen Piper, Faiz Siddiqui, and Irfan Uraizee. Inside the final seconds of a deadly tesla autopilot crash - washington post. `https://www.washingtonpost.com/technology/interactive/2023/tesla-autopilot-crash-analysis/`, 10 2023. (Accessed on 08/27/2024).

[294] California DMV. Disengagement reports - california dmv. https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/disengagement-reports/, 2023. (Accessed on 08/27/2024).

[295] Xiaoqi Wang, Wenbin He, Xiwei Xuan, Clint Sebastian, Jorge Piazentin Ono, Xin Li, Sima Behpour, Thang Doan, Liang Gou, Han-Wei Shen, et al. Use: Universal segment embeddings for open-vocabulary image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4187–4196, 2024.

[296] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.

[297] Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. [Online; accessed 2025-03-10].

[298] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[299] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, et al. Cogvlm: Visual expert for pretrained language models. *Advances in Neural Information Processing Systems*, 37:121475–121499, 2024.

[300] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[301] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[302] Vector embeddings - openai api. [Online; accessed 2025-03-11].

[303] Fatih Cagatay Akyon, Cemil Cengiz, Sinan Onur Altinuc, Devrim Cavusoglu, Kadir Sahin, and Ogulcan Eryuksel. SAHI: A lightweight vision library for performing large scale object detection and instance segmentation, November 2021.

[304] Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing aided hyper inference and fine-tuning for small object detection. *2022 IEEE International Conference on Image Processing (ICIP)*, pages 966–970, 2022.

[305] Euro ncap — the european new car assessment programme. [Online; accessed 2025-04-26].

[306] Rohan Grover. Encoding privacy: Sociotechnical dynamics of data protection compliance work. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2024.

[307] Ai agents: Built to reason, plan, act — nvidia. [Online; accessed 2025-04-26].

[308] openai/openai-agents-python: A lightweight, powerful framework for multi-agent workflows. [Online; accessed 2025-04-26].

[309] Will Epperson, Gagan Bansal, Victor Dibia, Adam Fourney, Jack Gerrits, Erkang Zhu, and Saleema Amershi. Interactive debugging and steering of multi-agent ai systems. *arXiv preprint arXiv:2503.02068*, 2025.

[310] Matt G. Southern. Youtube begins adding chapters to videos automatically, 2021.

[311] pytube developers. pytube, 2023.

[312] Mišo Belica. sumy 0.11.0, 2023.

[313] facebook. facebook/bart-large-cnn, 2023.

[314] Penn Treebank Project. Alphabetical list of part-of-speech tags used in the penn treebank project, 2022.

[315] Microsoft. Learn how to work with the chatgpt and gpt-4 models (preview), 2023.

[316] fixitsamo. Diy how to fix a flat tire easy!, 2014.

[317] Eugene W Myers. An o (nd) difference algorithm and its variations. *Algorithmica*, 1(1-4):251–266, 1986.