Docent: Digital Operation-Centric Elicitation of Novice-friendly Tutorials

Yihao Zhu
zhuyh22@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Qinyi Zhou
zhouqy22@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

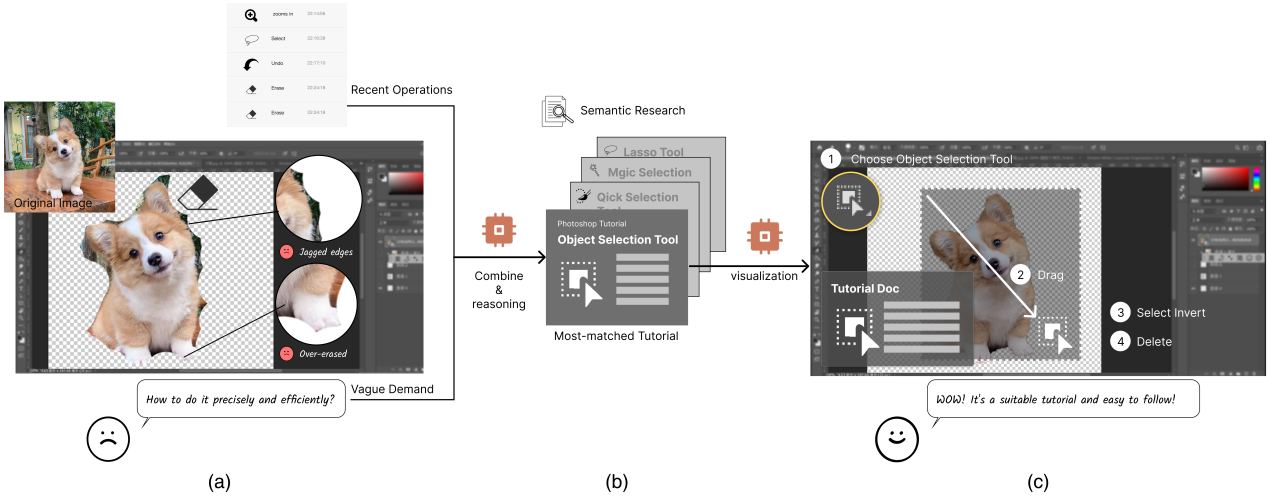

Figure 1: An example Docent use case in Photoshop. (a) A novice user intends to extract the puppy from its background using "Eraser" tool. (b) Docent utilizes the user's vague demand of seeking better practice and recent software operations to retrieve the most-related tutorial. (c) The user sees the in-situ tutorial visualized by Docent, and learns to use the "Object Selection" tool instead.

ABSTRACT

Nowadays, novice users often turn to digital tutorials for guidance in software. However, searching and utilizing the tutorial remains a challenge due to the request for proper problem articulation, extensive searches and mind-intensive follow-through. We introduce "Docent", a system designed to bridge this knowledge-seeking gap. Powered by Large Language Models (LLMs), Docent takes vague user input and recent digital operation contexts to reason, seek, and present the most relevant tutorials in-situ. We assume that Docent smooths the user experience and facilitates learning of the software.

CCS CONCEPTS

• **Software and its engineering** → **Documentation**; • **Human-centered computing** → **User interface toolkits**.

UIST '23 Adjunct, October 29–November 01, 2023, San Francisco, CA, USA

ACM ISBN 979-8-4007-0096-5/23/10.
https://doi.org/10.1145/3586182.3625121

KEYWORDS

Software Tutorials, Digital Operation, Large Language Model

ACM Reference Format:
Yihao Zhu and Qinyi Zhou. 2023. Docent: Digital Operation-Centric Elicitation of Novice-friendly Tutorials. In *The 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23 Adjunct), October 29–November 01, 2023, San Francisco, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3586182.3625121

1 MOTIVATION

Though software developers produce an abundance of online tutorials, novice users always fail to utilize these resources efficiently when encountering specific issues[6].

Imagine a novice Photoshop user attempting to remove a photo's background. Initially, the user opts to manually erase the dog's background using the "Eraser" tool, a method both imprecise and laborious. When seeking a more efficient method, formulating a search query becomes a challenge; the user struggles to translate their needs into precise search terms. This lead to prolonged keyword searches and browsing of entries. Moreover, once they locate the most-related tutorial, they're forced to switch between the tutorial and their task. All these factors contribute to an inefficient and tedious user experience.

We identify three major challenges in the tutorial-seeking process: **(a) Incompetence in articulating specific tutorial needs**, **(b) Difficulty in finding the most relevant tutorials**, **(c) Inefficiency in comprehending and implementing tutorial instructions**. Therefore, we introduce Docent, a novel system designed to mitigate these prevalent issues.

2 SYSTEM INTRODUCTION

As shown by Figure 1, when Docent is on, users can use the software as usual. When users encounter problems, they can pose ambiguous natural language questions. Just as if an expert of the software is sitting aside observing their screen, Docent will elicit the most relevant tutorial and display in-situ guidance on the software interface. This seamless experience is realized through three advancements in intention articulation, tutorial match-up, and tutorial integration.

Articulating search intention requires superior software mastery. This is because software masters can delineate the context and frame it in proper language[10, 13]. Previous work like Torta[8] and RePlay[4] captures UI events' names and interaction descriptions, presenting them for users to adopt in the expressions. However, these cues often fall short of describing the full context. Docent enhances this by recognizing ongoing semantic user activities using prolonged events series. Novice users can then supplement this context with imprecise natural language expressions. Docent amalgamates this user input with the context, and leverage LLM to reason about potential search intentions. With its extensive common knowledge possession and reasoning abilities[1], the generated intentions can be contextually accurate.

Moreover, matching these intents with appropriate tutorials is also challenging. Because the users have to conduct the exhaustive browse and evaluation through the vast array of candidate tutorials. Traditional searching methods rely on keyword matching, prompting some researchers to extract more metadata from tutorials to refine the matches[11, 12]. Docent, however, employs a semantic search approach. Pretrained with extensive language corpus, it can map any natural language expression in high-dimensional semantic space[7], enabling advanced comparison.

Lastly, comprehending and following an external tutorial can be taxing for users. Because switching between the tutorial and the software can cause huge cognitive load[2]. Docent builds upon established in-place tutorial display designs[3, 14] and we automate the integration to eliminate manual inputs. This improvement is realized by employing LLM to interpret the tutorial content and subsequently integrate highlighted hints on key UI elements.

3 TECHNICAL PLAN

To create a seamless tutorial search user experience, Docent adopts three modules, namely, **(a) Operation Monitor**, **(b) LLM-powered Matcher**, **(c) Tutorial Transformer**, as shown in Figure 2. Initially, the Operation Monitor captures and summarizes usage context. Then, the LLM-powered Matcher combines the context with the user's vague search input to infer possible intents. Using semantic search, this module finds the most-related tutorial. Finally the Tutorial Transformer converts the tutorial into in-situ guidance configuration and displays it on software interface.

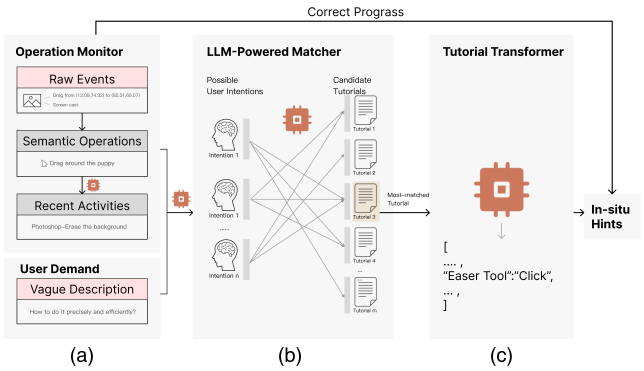

Figure 2: Docent technical flow. (a) The Operation Monitor captures raw computer events and transforms them into recent activity descriptions in three steps. (b) The LLM-Powered Matcher finds the best-aligned tutorial with inferred user intentions. (c) The Tutorial Transformer converts the retrieved tutorial into in-situ hints configuration

We implemented an Operation Monitor on Windows platform. The Monitor registers raw events like 'click' and 'drag' via Windows OS interaction API (e.g., user32.dll) and captured screencasts. Then, several related events are clustered to form a semantic operation using predefined rules. Finally, LLM is invoked to summarize the recent operations into a representation of current activity context.

Within the LLM-powered Matcher, we first prompt LLM to reason about the potential user inquiries based on the current activity context and the vague search expression. Given the variability in LLM outputs and the inherent ambiguity in discerning true intent, we concurrently instruct the LLM to generate a set of possible search expressions.

Following this, the Matcher utilizes the LLM to embed both the derived expressions and the tutorial metadata (including titles and primary descriptions) into high-dimensional vectors. The ideal tutorial is then selected by comparing the proximity between vectors representing user intention and tutorial content.

Once the optimal tutorial is sent to the Tutorial Transformer, the LLM interprets the tutorial and extracts guidance in a structured "Element-Action" format. These directives are then projected onto the software interface, highlighting interaction target and advance based on user's progress reported by the Operation Monitor.

4 DISCUSSION

We demonstrate the primary advantages of Docent in the following three aspects:

(1) It augments the learning curve of novice software users.
(2) It revitalizes and leverages existing tutorials.
(3) It is fully automated and requires no human intervention.

Additionally, we demonstrate that our system framework can efficiently address the challenges highlighted by previous works, such as tutorials in video form[4, 9, 11], cross-application tasks[4], and variations in sources[5], due to its general intelligence and operation system-level activity capture.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China under Grant No. 62132010, and by Beijing Key Lab of Networked Multimedia, the Institute for Guo Qiang, Tsinghua University, Institute for Artifcial Intelligence, Tsinghua University (THUAI), and by 2025 Key Technological Innovation Program of Ningbo City under Grant No. 2022Z080.

It is also supported by supported by Beijing Municipal Science and Technology Commission , Administrative Commission of Zhongguancun Science Park No.Z221100006722018.

REFERENCES

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, 1877–1901.

[2] Pei-Yu Chi, Bongshin Lee, and Steven M. Drucker. 2014. DemoWiz: re-performing software demonstrations for a live presentation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Toronto Ontario Canada, 1581–1590. https://doi.org/10.1145/2556288.2557254

[3] Pramod Chundury, Mehmet Adil Yalçin, Jonathan Crabtree, Anup Mahurkar, Lisa M Shulman, and Niklas Elmqvist. 2023. Contextual in situ help for visual data interfaces. *Information Visualization* 22, 1 (Jan. 2023), 69–84. https://doi.org/10.1177/14738716221120064

[4] C. Ailie Fraser, Tricia J. Ngoon, Mira Dontcheva, and Scott Klemmer. 2019. RePlay: Contextually Presenting Learning Videos Across Software Applications. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–13. https://doi.org/10.1145/3290605.3300527

[5] Ben Lafreniere, Andrea Bunt, Matthew Lount, and Michael Terry. [n. d.]. "Looks cool, I'll try this later!": Understanding the Faces and Uses of Online Tutorials. ([n. d.]).

[6] Tessa Lau, Clemens Drews, and Jeffrey Nichols. [n. d.]. Interpreting Written How-To Instructions. ([n. d.]).

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[8] Alok Mysore and Philip J. Guo. 2017. Torta: Generating Mixed-Media GUI and Command-Line App Tutorials Using Operating-System-Wide Activity Tracing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, Québec City QC Canada, 703–714. https://doi.org/10.1145/3126594.3126628

[9] Cuong Nguyen and Feng Liu. 2015. Making Software Tutorial Video Responsive. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Seoul Republic of Korea, 1565–1568. https://doi.org/10.1145/2702123.2702209

[10] Srishti Palani, Zijian Ding, Stephen MacNeil, and Steven P. Dow. 2021. The "Active Search" Hypothesis: How Search Strategies Relate to Creative Learning. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*. ACM, Canberra ACT Australia, 325–329. https://doi.org/10.1145/3406522.3446046

[11] Luca Ponzanelli, Gabriele Bavota, Andrea Mocci, Massimiliano Di Penta, Rocco Oliveto, Mir Hasan, Barbara Russo, Sonia Haiduc, and Michele Lanza. 2016. Too long; didn't watch!: extracting relevant fragments from software development video tutorials. In *Proceedings of the 38th International Conference on Software Engineering*. ACM, Austin Texas, 261–272. https://doi.org/10.1145/2884781.2884824

[12] Luca Ponzanelli, Gabriele Bavota, Andrea Mocci, Massimiliano Di Penta, Rocco Oliveto, Barbara Russo, Sonia Haiduc, and Michele Lanza. 2016. CodeTube: extracting relevant fragments from software development video tutorials. In *Proceedings of the 38th International Conference on Software Engineering Companion*. ACM, Austin Texas, 645–648. https://doi.org/10.1145/2889160.2889172

[13] Soo Young Rieh, Kevyn Collins-Thompson, Preben Hansen, and Hye-Jung Lee. 2016. Towards searching as a learning process: A review of current perspectives and future directions. *Journal of Information Science* 42, 1 (2016), 19–34.

[14] Mingyuan Zhong, Gang Li, Peggy Chi, and Yang Li. 2021. HelpViz: Automatic Generation of Contextual Visual Mobile Tutorials from Text-Based Instructions. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 1144–1153. https://doi.org/10.1145/3472749.3474812