BuildTogether: Reimagining AI Tools as Learning Partners

Manooshree Patel
manooshreepatel@berkeley.edu

1 Introduction

Educators (and students) around the world are concerned of the effect Artificial Intelligence (AI) is having on students. Students note that they feel they are "getting dumber" and losing a sense of pride over completed schoolwork—despite improved assessment outcomes [Goetze, 2025]. Feeling a loss of agency over completed work extends outside the classroom and the question of "*am I getting dumber?*" has plagued most users of AI-based tools at one point.

At the same time, AI-based tools, Large Language Models (LLMs) in particular, provide novel opportunities for human-machine collaboration that were not before possible. The conversational nature and impressive reasoning capabilities of LLMs have transformed the technology into a "do-it-all" tool. The question for engineers and educators then should become,"How can this technology most effectively be leveraged to promote user learning and agency?". The development of novel interaction modes between a user and an LLM are vital to effectively harnessing LLMs to not just be capable chatbots, but useful "thought partners" and tutors [Collins et al., 2024, Patel et al., 2025].

In this work, I focus on the issue of user-LLM collaboration when editing textual data. A report from Anthropic shows that one of the most common ways students use Claude is asking for "iterative refinement" on assignments [Anthropic, 2024]. When an English teacher provides feedback on a student's essay, they annotate the writing with blue circles and red lines. The key part of their strategy is not writing the generally most optimal sentences, but rather, on meeting the student where they are and uplifting the student's work [Underwood and Tregidgo, 2006]. In contrast, when a state-of-the-art LLM is prompted to provide edits, the model biases towards outputting the highest probability tokens. This results in model outputs that are far from the original user input, which is in many cases unnecessary. Users are left feeling as though the model either was unhelpful or that they must adapt this entirely new piece of work— completely devoid of their initial efforts.

This unnecessary rewriting of user's textual input presents two major implications for a user's learning and agency. Firstly, the model is no longer in the user's Zone of Proximal Development (ZPD) [Vygotsky and Cole, 1978], when the model completely reconstructs the user's input. Vygotsky defined the ZPD as the "distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving ... in collaboration with more capable peers". The AI-based tool takes on the role of the more capable peer, but by rewriting user input, is interfering with the user's cognitive development. Secondly, by unnecessarily completing the user's work for them, the user loses their epistemic agency, or their ability to control their own knowledge creation [Scardamalia et al., 2002]. When users lose control of their work, they no longer feel a sense of empowerment or agency when completing the task.

I propose a novel form of human-machine collaboration in which machines meet humans in the middle and both parties **build together**. Focusing on the use case of "iterative refinement", in this interaction paradigm, the AI tool:

1. makes the minimal necessary edits to the user's work
2. displays the introduced edits very clearly in comparison with the user's original work

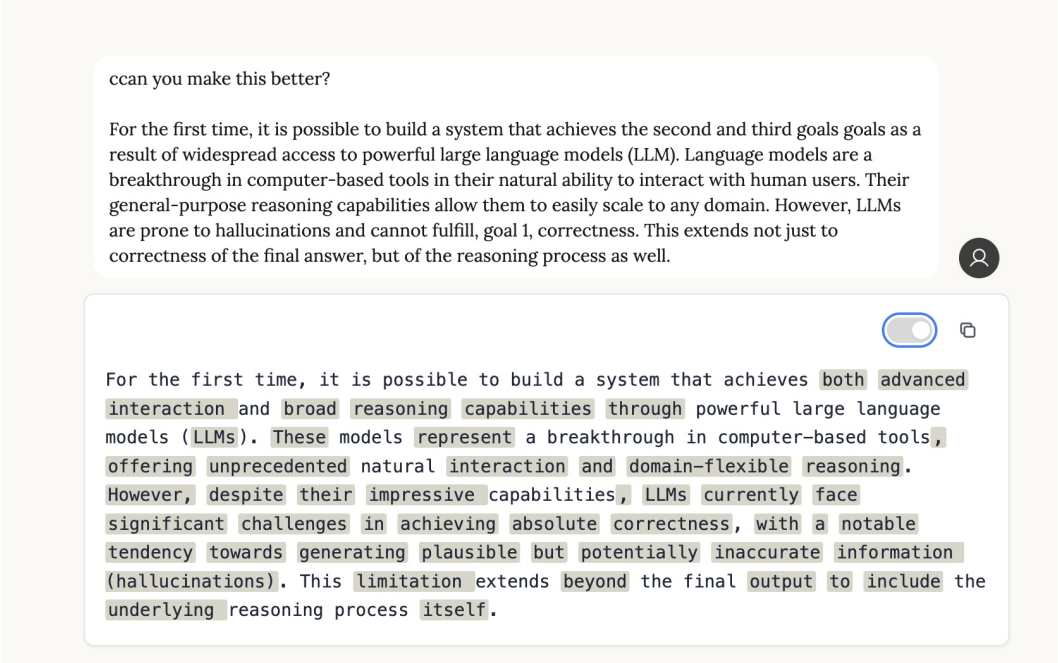

Figure 1: A screenshot of the edit feature. Claude makes minor changes to the user input, which are clearly highlighted. The user can easily toggle back and forth between their input and Claude's edits.

By making minimal edits and clearly showing how AI is merely enhancing the user's input, the user still feels in control of their work and that the work is definitively theirs (not the sole work of an AI tool). Intentionally, this workflow shouldn't slow down user productivity. The new interface gives users the ability to easily monitor contributions from the LLM, but doesn't hinder their collaboration with the LLM in any way. I develop a prototype of this feature in the mocked online Claude chat interface, shown in Figure 1. My code can be found at `https://github.com/manooshree/BuildTogether`. A video demo can be found at `https://www.youtube.com/watch?v=GzyrazMdqY&ab_channel=ManooshreePatel`.

2 Design

This feature design has two key components: 1) an API-level change in which Claude is instructed to make minimal edits 2) a user interface-level change in which a new component renders the difference in the user's inputs and Claude's updated version.

API-Level Changes. For the purposes of this prototype, I instruct Claude to make the minimal changes necessary to satisfy the user's query. In my testing, I found that prompt engineering Claude was effective. A more computationally expensive, but perhaps more robust, approach to invoke this behavior in Claude could be done in the model post-training phase. Specifically, in a reinforcement learning environment, a reward model could be designed to encourage minimal edits. The current implementation can handle all textual input ranging from essay paragraphs to code snippets.

User Interface Changes. A key challenge in this design was allowing the user to easily understand which changes were made by the model, without overwhelming the user with panels of text. I develop a React component which displays Claude's edits in clear highlights, overlaid on the user's input. The user can also easily toggle the component to view their original input, as shown in Figure 2. Highlighting text quickly communicates the location of all edits to the user. The decision to toggle between input and output, rather than displaying both at once, was intentional to reduce the amount of information on the screen.

The basis for this Claude interface mock-up is derived from this GitHub repository, which is under an MIT License. Colors were taken from the Claude chat interface. Previous design iterations can be

found in my Figma document. The online Claude chat tool was used both as a design study and also a developer tool in this project.

Figure 2: A feature use case in which a user uploads a buggy piece of code. The left panel displays the user's original input. The right panel displays Claude's edits, highlighted in gray. Additional commentary on the changes is displayed below the component.

Limitations. The current prototype has many limitations. My application cannot take in multimodal inputs or files. The output is not currently streamed, as has become customary in LLM chat applications. The model does not retain memory of previous messages in the chat. Most importantly, the Claude API is currently set to provide minimal revisions on user input. It does not perform well on other tasks. If this feature were to be implemented in the larger Claude application, user queries would need to be carefully rerouted to this feature.

Experiment Design Two levels of experiments could evaluate effectiveness of this feature. A qualitative study seen traditionally in education or human-computer interaction (HCI) research could be conducted on a small focus group of individuals (n = 20). At a high-level, users' proficiency in some task would be measured before and after their interaction with the feature to measure overall learning gains. A "think-aloud" protocol could be employed to elicit more fine-grained feedback on users' experiences as they use the tool. This smaller qualitative study would be especially useful in understanding users' affective experiences around agency and empowerment when using the feature. From a product perspective, important telemetry would include recording users' interactions with the toggle button, the copy button, and continued interactions with Claude's edited outputs in future messages. A successful feature launch would be characterized by high rates of interaction with the toggle and copy buttons, as well as continued engagement with Claude's edited outputs in subsequent messages. These metrics would suggest that users find the feature usable and valuable.

References

Catherine Goetze. The real reason why students are using ai to avoid learning. *Time*, April 2025. URL `https://time.com/7276807/why-students-using-ai-avoid-learning/`.

Katherine M Collins, Ilia Sucholutsky, Umang Bhatt, Kartik Chandra, Lionel Wong, Mina Lee, Cedegao E Zhang, Tan Zhi-Xuan, Mark Ho, Vikash Mansinghka, et al. Building machines that learn and think with people. *Nature human behaviour*, 8(10):1851–1863, 2024.

Manooshree Patel, Rayna Bhattacharyya, Thomas Lu, Arnav Mehta, Niels Voss, Narges Norouzi, and Gireeja Ranade. Leantutor: A formally-verified ai tutor for mathematical proofs. *arXiv preprint arXiv:2506.08321*, 2025.

Anthropic. Anthropic education report: How university students use claude. `https://www.anthropic.com/news/anthropic-education-report-how-university-students-use-claude`, 2024.

Jody S Underwood and Alyson P Tregidgo. Improving student writing through effective feedback: Best practices and recommendations. *Journal of Teaching Writing*, 22(2):73–98, 2006.

Lev Semenovich Vygotsky and Michael Cole. *Mind in society: Development of higher psychological processes*. Harvard university press, 1978.

Marlene Scardamalia et al. Collective cognitive responsibility for the advancement of knowledge. *Liberal education in a knowledge society*, 97:67–98, 2002.