

Feature Spec - Workshop your Writing with Bloom

Problem

Across all 3 of Courtland's conversations, people saw Bloom bot and they brought up *writing*. A big part of literacy is the ability to generate language as well as comprehend it.

Scaling writing instruction is hard and almost never done well. There's a strong asymmetric relationship between the amount of time students need to practice writing versus the amount of time teachers have to grade writing.

When Courtland was teaching, he'd give them many low-stakes opportunities to practice their writing. Every single Monday, he'd hand them a blue book and they'd write an essay by hand during the 40-min class period. These wouldn't be graded, the students would work together, and they'd talk about it over the course of that week. Every term, the students would have to polish one of those essays and bring it to class to be "workshopped".

Writing workshops are when students put their essays on the projector one-at-a-time for the whole class to read + discuss, and each student writes down feedback that they can then give to the student. It's a great feedback mechanism but the issue is it takes a *super long time* to get to this point. Students need to have a decent amount of experience writing, critiquing, etc before this is even a useful exercise.

Separately, the biggest academic integrity misalignment in private tutoring is essay writing. A similar asymmetric time relationship exists, where papers that require a lot of work are put off until the last minute, and tutors have other sessions to do. It's often the case that this is where tutors cross the line with providing help, because the tutors won't continue to have the student as a client if they don't get a good grade, which keeps the parents happy and the money coming through the door.

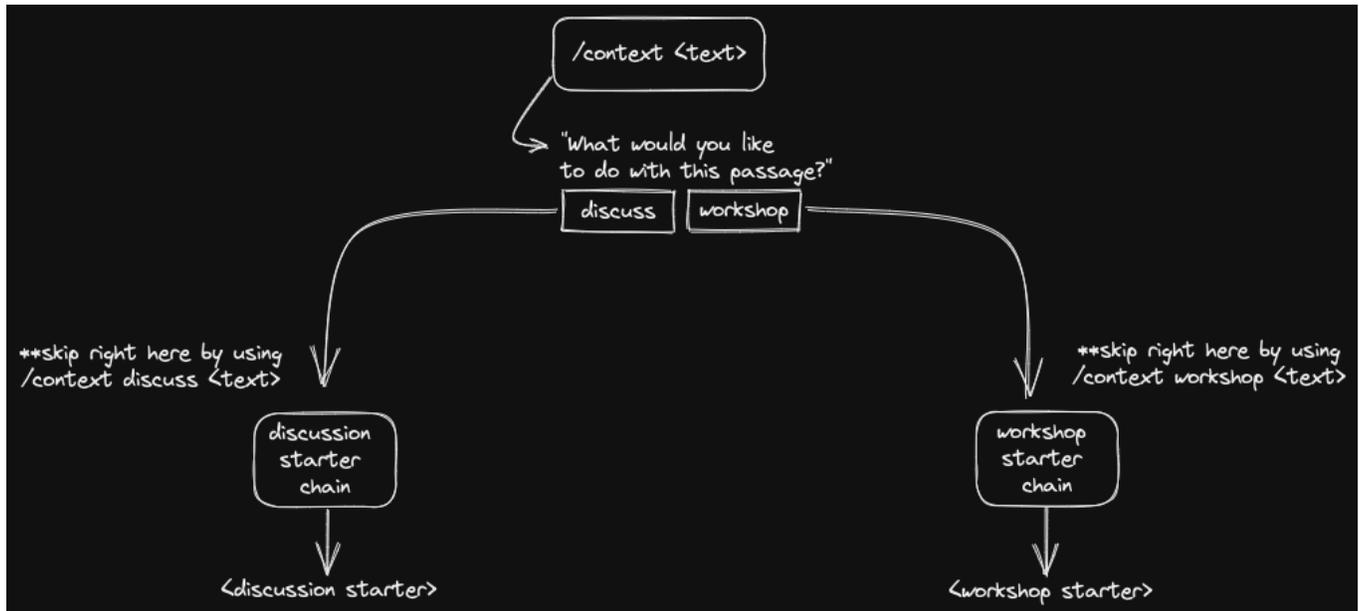
Kelly Bradley from the University of Kentucky said that her dream writing tool is something that would allow her grad students to workshop their abstracts before they brought them to her. Courtland said that so many kids just need help getting started writing and need someone to talk it out with.

Given the existing workflow of users bringing text to Bloom, a separate prompting workflow can be developed to elicit pedagogical thoughts on the dialogue surrounding a

passage of the student's writing, just like we do for dialogue surrounding a reading passage.

Feature Proposal

We're not changing a whole lot in order to incorporate this feature. Here's a high level diagram of the UX:



Users can still use the `/context` command and paste text, we will just no longer assume the text is something the student is reading and wants to discuss. We'll leverage the [buttons](#) functionality in Discord to allow the student to decide which task they'd like to do. If they are more familiar with the bot and know what they intend to do with the text they're bringing, they can skip directly to that task by saying `/context discuss` or `/context workshop`.

This will create the need for a separate chain for workshoping writing, but it remains consistent with the product paradigm decisions we've made thus far (which we're finding is a more robust paradigm than we might have thought):

1. Students bring a short-to-medium sized passage of text to the bot
2. The text gets injected into a few-shot example prompt template for the specified task
3. Pedagogical thoughts inform the dialogue responses

Re: 1, this allows us to avoid having to solve the problem of managing text embeddings of larger documents. Being able to ingest larger pieces of text would be desirable, but it also requires us to ensure we're selecting the relevant pieces of text to include in the context window every time. Inferring this from dialogue alone is very fuzzy, and would create

ample opportunity for the bot to fail. Constraint in this regard *might also be a feature* for the student, because it forces them to break down their reading/writing into much more specific pieces to discuss.

Re: 3, this has been our differentiating factor. Long term, our model's ability to predict solid pedagogical thought sequences that then inform its ensuing responses is going to be a strong moat. It's akin to having a strong Chain-of-Thought model. Our principles will inform how thoughts get revised by humans, and over time we'll have to develop a reputation for our bot just *being the best teacher*. So the fact that this writing workshopping idea fits nicely into this paradigm is a huge plus, and something we should continue to lean into.

Implementation

We'll likely introduce a new file or two into the `bot/` folder to separate out discussion and workshopping functions. The `core.py` file will need to be re-worked to generalize its functionality and no longer assume everything is discussing a passage.

You can only have one `on_message` event function (as well as `on_message_edit`, `context`, etc), so we need to remember what type of conversation is being had with any given channel id in order to consistently call the right chains. We can do this by adding an extra `type` variable to the `ConversationCache` class and updating the functions to query the type before proceeding.

We're also going to have a completely separate set of prompts (starter, thought + thought summary, response + response summary). This will likely imply a re-structure to the `prompts/` folder.

In order to follow convention, it would make sense to update the `examples.py` to be a command sub-group of `discuss`. Thus the new command would be `/context discuss <example name>`

This changes how the `load_chains()` and `load_memories()` functions work. Right now, they load everything assuming it's a reading discussion. I believe the solution is to make those functions more agnostic and send in specific things you want to load. Like we should switch it to `load_chain()` (singular) so it can construct a chain given a path to a `prompt_template` file.

Conclusion

Sama [said](#) these models are better reasoning engines than databases, so it follows to say these models should be better at giving writing feedback than writing essays. It should

be excellent at taking your writing and generating good reasoning about your work to inform dialogue responses.

Overall, this feature upgrade is forcing us to make the codebase more feature agnostic and further solidify our prompting paradigm. Should we have more ideas in the future that require a context-thought-response, adding it should only be a matter of generating the prompts.