Pilot + Evaluation Prep

Pilot [+1]

1. [+0.5] Present the pilot user with a brief statement of the scenario and task. Ask the pilot user to complete the task. Note: You might feel (very) nervous that something will break. That is OK. It's ok for the pilot user to break things as they test out your system. Be prepared to restart/recover your system when things break. Note what happened step by step. Include 0.5-1p of notes on one pilot user. Additionally, summarize in a few sentences: What happened? Why? What changes do you need to make to your system before the next pilot?

   Asked user (MS CS student) to use our pilot after explaining what is/how it works and come up with their own question. Started with “what is the meaning of life?”. This question achieved a relatively low complexity score (0.18). The response was short as well since this is an “easy” question to answer for an LLM. The user then played around with the slider, shifting the thinking limit from “automatic” to “deep.” Because this only changes the upper bound of response generation, the LLM response to the aforementioned question didn’t change. However, we think that due to how the settings on the slider are labeled (quick -> balanced -> thorough -> deep), users can be misled to think that a setting on the right end of the scale indicates the response will be longer. This is not the case. We probably need to rethink how the user chosen limit is handled and represented. The user also didn’t seem to pay much attention to the complexity score. If they did take note of it, they didn’t external comment on or interact with it in any particular way. Michael suggested changing the visualization of complexity score from a number to something more eye-catching. After suggesting the user try a more involved query, perhaps about their capstone, the user tried a query composed of multiple parts. Due to the volume of the response, we weren’t able to read it closely, but this more involved query received over double the previous query’s complexity score and received a much more lengthy response.

   Summary: We asked our user to try our system with queries they had to come up with. There was a misconception about what the settings on our slider meant since “deep” led them to believe that the response would be more involved, regardless of how simple or complex their query was, which isn’t the case. Our main takeaway from this user is that we should add some kind of FAQ or brief explanation to clarify how the thinking budget scaling works. Perhaps we should also rethink the slider as a UI component. In any case, the changes we will make before our next evaluation are:
   1. Revise the presentation of the metrics of complexity, thinking budget, etc.
   2. Provide FAQ/Info buttons that will help the user understand the definition of the terms

we present, as well as how to interact with the LLM. This is especially relevant to establishing that thinking budget is an upper limit, not a lower limit on the amount of 'thinking' done by the model.
3. Reorganize the UI to be more cohesive, making use of the sidebar.
4. Tune the heuristic to be more sensitive to use cases.
5. Take a look at the max_tokens definition of the queries we're sending (this parameter is defined as the thinking budget + response length token budget). This may allow us to provide a bound to the length of responses, potentially decreasing latency across this vector in a more apparent way than the thinking budget limit.

2. [+0.5] Involve another pilot user outside of the course. Include 0.5-1p of notes on this second pilot user. Summarize in a few sentences: What happened? Why? What changes do you need to make to your system before the next pilot?

   Asked Biology MS student from UCSD to use the pilot after explaining the context. The user started with "What are some important aspects to include in the introduction section of writing a thesis related to prokaryotic gene expression and regulation." The response was adequate, and the user noted that it was overall faster than Deepseek (compared thinking model latency). I explained that the thinking budget was an upper bound, not a lower one. Noticing the slider was set to "automatic" by default, the user tried the same query several times with pre-determined thinking budgets in order to compare the results. The results varied somewhat, but given the repeated nature of the query in the same thread, I stepped in to mention how the model was likely to repeat the first answer given that it had access to previous messages. The user then reloaded the page in between the tries and got marginally shorter answers for the quick response, and longer ones for the deep setting.

   I suggested trying a more complex query, at least one that might require more "thinking" rather than a generic answer. The user thought of an issue from their lab work: "I'm doing DNA extraction from a PCR and Gel electrophoresis product and when I send my samples in to get sequenced, they come back with noisy sequences. Given that the sequencing company is reliable, what should I make sure to do carefully during the purification procedure for better results." That query got a complexity score of 0.51, with decent but not impressive theories as for the origin of the issues. The user decided to add to the query "think very hard about this." The complexity score increased to 0.64 with this addition, and the actual results were substantially longer and more in depth than with the original prompt. This inspired me that we might manually modify the user's responses according to our "depth" levels in order to encourage Claude 3.7 to actually conform somewhat to the levels of thinking more effectively (trying to push the lower bound to match the thinking budget more often). Since the thinking budget is rarely reached for the average use case, even in the quickest setting, we might need to encourage levels of depth behind the scenes with just prompt engineering to improve overall user experience.

Before conducting an evaluation [+3]

1. [+0.5] Articulate 1-2 questions motivating the evaluation. In other words, what are the 1-2 things you want to prioritize learning through the evaluation?

   The question motivating the evaluation is the following:
   1. How well does heuristically defining a token limit for thinking models correlate with improving user satisfaction with their LLM-usage experience? E.g., by providing a balance between latency and response quality/transparency, reducing iteration count, etc.

2. [+0.5] What metrics will you use to answer the above research questions? Why are these metrics appropriate? What are the benefits and drawbacks of using these metrics? Requirements: You are required to conduct a mixed-methods study where you collect qualitative and quantitative data. In your response to this question, describe what kind of data (e.g., open-ended survey, interview, time, clicks, etc.) will be useful for answering your motivating questions.

   Our current plan, pending your feedback, is to perform A/B testing between the models we initially used during user research (ChatGPT and DeepSeek) and our project. In our comparison with ChatGPT, we will be able to find deltas in answer quality, transparency, and iteration count with respect to the user's perceptions. In our comparison with DeepSeek, we will be able to get a feel for the effectiveness of defining thinking budget caps w.r.t reducing the latency a user experiences when waiting for responses for their queries. Both of these comparisons will hopefully allow us to measure the validity of tuning thinking budget specifically to a user's query complexity.

   In regards to quantitative metrics:
   1. Response Latency per Question: how many seconds does it take for the tested model to finish their response, inclusive of thinking time (for reasoning models).

- Benefits: allows us a raw metric to measure if response latency is improved with a thinking budget cap, and will allow us to accurately capture differences between same queries on different models.
- Cons: only a useful comparison between identical questions, and may be conflated with server difficulties/ rate limits on models we're using.

2. Raw Number of Corrections / Clarifications required per query: how much follow-up does the user need to devote in order for the model to satisfactorily return a response?
- Benefits: allows us a metric to see the differences in response quality, e.g. how much smarter is one model than another? How much less does the user have to guide the LLM to an answer?
- Cons: very much can be conflated with weak user prompting behavior. May be difficult to extrapolate this metric between different users, simply because differing use cases/user behaviors make generalizing this metric difficult.

3. User Ratings on answer quality, latency time, etc.: how does the user assign a numerical value to their experience on the model, especially in regards to the user-focused metrics that we are testing (e.g. latency, iteration count, answer quality)?
- Benefits: assigns a quantitative rating to a user's qualitative experience with these models, which is otherwise difficult to ascertain just from interviews.
- Cons: as with all rating scales, suffers from user's hidden perceptions of what a rating 'defines' (e.g., a 4 for a user could be for a 'standard' experience, but a 3 from another user could be what they define as 'standard')

4. Token Limit vs. Tokens Used Ratio: how much of the assigned thinking budget is actually used in the generation of a response?
- Benefits: allows us to glean the strength of our heuristic, e.g. how consistently is the thinking budget being used up across responses? If the ratio is too low, then that means our thinking budget is doing nothing to limit the thinking of our model.
- Cons: not necessarily related to the user's experience with the model, moreso a validation of our heuristic.

5. Iteration Count: how many iterations does a user need to provide on their initial queries to 'solve' the problem presented in their workflow?

- Benefits: allows us a metric to see the differences in response quality, e.g. how much more trustworthy is one model vs. another, and how well can a model predict or extrapolate next steps?
- Cons: very much can be conflated with the user's assumptions or prompting behaviors, e.g. some users will assume iteration and do so naturally, while others will try to 'one-shot' the problem. May confound some comparisons.

In regards to qualitative data, we want to perform a structured interview in order do obtain responses to the following questions:
1. What was their attitude towards the detail, depth, and response length of the responses generated by our model vs. competitor models?
2. Did the amount of thinking they witnessed in our model accurately correlate to the level of transparency and depth required for their use case?
3. Etc.

Because we're performing an A/B test between our product and older models, we should be able to accurately glean the differences in user perception and user experience via the quantitative data we obtain as well as the user's 'think aloud' experience, which should inform us as to the effectiveness of heuristically guided thinking limits in both limiting the latency of thinking models and strengthening responses relative to 'over-thinking' or 'no-reasoning' models.

3. [+1] Specify a plan for recruiting participants.

- How will you contact participants (e.g., mailing lists, in-person, etc)?

We will reach out to the participants originally studied, as well as asking if they have any friends/peers that would be interested in participating in the evaluation.

- What are your inclusion/exclusion criteria for participants?

We're going to limit participants to our original target demographic: college students. We will also ask that users arrive with a workflow or task that is pertinent to their education/work and must be performed 'well', such that a thinking model wrapper would not be overly fluffy for their use cases.

- Will you include participants you interviewed for user research? Why or why not?

We will reach out to those people. Their prior experience with our user research will perhaps help their familiarity with reasoning models and we've pivoted/further developed our idea since then, so the info we'd gain from the people from our user research will likely be in the same realm as people who did not.

- Where will you perform the evaluation?

We will conduct this evaluation over Zoom on an individual basis. If initial participants prove that having them set up our model locally is untenable, then we will move to scheduling in-person.

- What data will you collect from participants? How will you inform them of this and obtain informed consent?

After the start of our introduction, we will proactively inform them of our collecting/measuring of the metrics referenced in (2.) above, as well as inform them that their verbal responses and reactions will be analyzed to glean qualitative insights about our product. We will inform them that their interview questions will be transcribed, if allowed. Furthermore, with respect to the assigned thinking budget vs. actual thinking length ratio, we will require copies of their responses and chat history, which we will ask ahead of time to collect.

4. **[**+1] Write out a step-by-step protocol for conducting each user evaluation. Getting on the same page is important for more easily conducting studies and analyzing data across participants. Your protocol should include: (1) a script of what you will say to each participant; (2) what behaviors/responses you expect from participants and how that may change the flow of the study, if at all; and (3) how you will transition between phases of the study (e.g., from a task to an interview).

The evaluation will be in two phases: the first being an A/B testing phase where the user will be participating in a Think/Aloud, filling out evaluation forms after each question asked to the model. The second phase will be a structured interview where their perceptions of user experience and model quality are directly inquired.

Prework:
Interviewer:
For each user, determine which non-project model (DeepSeek or ChatGPT) they will be using. Determine which model will come first in the evaluation with respect to the previous evaluations conducted; the intent of this is to prevent a potential confounder in user perception w.r.t model ordering.

Have the model evaluation form(s) on standby. Each page on the form asks the following questions:
1. Please provide the prompt of the question.
2. Please provide the model's thinking process, if applicable.
3. Please rate the answer quality of this response.

4. Please indicate your satisfaction with the latency you experienced with this response.

Interviewee:
If the interview is being conducted over Zoom, make sure you send the repository link to them ahead of time, and that they properly set up our model on their local device.

Request that the interviewee have a pertinent, relevant, and ideally somewhat-complex workflow that can leverage the power of more advanced large language models. These can be things such as homework, projects (coding or otherwise), or brainstorming.

Introduction [5 min]:
Thank them for participating in the evaluation. Vaguely introduce our project and its high-level intent. Introduce the task for evaluation: executing the identical workflow on the project model and the chosen non-project model. Detail exactly what they will do. E.g., start with identical prompts on each model. Execute the complete workflow on one model, before moving onto the next model. Assume no prior knowledge from interaction with the previous model.

Ask for their informed consent as to the data being collected: that is, the latency of their queries' responses, their user ratings, and transcripts of their query queries and query responses' thinking process (for evaluation of token used - token allocated ratio). (If consent is not provided w.r.t the thinking processes, settle for them providing a word count/character count. If consent is not provided w.r.t queries, simply label each question with a user number - query number; e.g. U0-Q0 for the first user's first query.)

"Hi, thank you for volunteering your time to evaluate our project! Our goal for this HCI project is to improve user satisfaction with their LLM experiences, and your experiences will inform us as to whether we've achieved that goal."

"The task we've set out for you today is to run an identical workflow on two models: one being our project, and the other being [ChatGPT/DeepSeek]. (Switch around the orders as intended) First, we're going to have you run your complete workflow on [ChatGPT/DeepSeek/Proj]. After you're done on the first model, we're then going to have you run your complete workflow on [ChatGPT/DeepSeek/Proj]. You're going to start off with the same prompt you first gave to the first model, and execute the same workflow assuming no prior knowledge from the responses of the first model. When you're going through this, we want you to pay special attention to how long you're waiting, how good the model's response is, and how well the response fulfills your goals."

"After each question and response, we're going to have you fill out a form that'll have you rate the quality of each response as well as your satisfaction with the response's delay. If the model is DeepSeek or our Model, we're going to have you provide the transcript of the thinking process so that we can obtain its word/char/token count for

further analysis. We are also going to be timing the delay of each response. If you have any issue providing any of this data, please let us know now and we can find workarounds."

"Feel free to think aloud as you work with these models. Indicate any opinions, positive or negative, that you have with your experience: all of this is valuable data for evaluation of your experience."

"After you complete your workflow on both of today's models, we're going to have a short interview concerning your experiences with the evaluation. If possible, we'd like to have a transcript of this interview. If you're uncomfortable with this, we are also willing to just take notes on your responses."

"If you have any questions, please feel free to ask them now."

If no questions, direct them to the first model being evaluated.

Evaluation [10-20 min]:
No set script for this. Make sure the user is executing the evaluation correctly, e.g. beginning with the same prompt on both models, executing an identical workflow one model at a time. Encourage them to vocalize their thought processes and indicate any points of confusion. Answer any questions they have w.r.t the models if they need clarification.

In terms of interviewer responsibilities:
1. For each question, time the delay between the question being submitted and the response being fully generated.
2. After each question/response, make sure the subject fills out a page on the form. Make sure the form corresponds to the correct model.

After they complete their full workflow on one model, direct them to the next model and the form corresponding to that model.

Potential behaviors:
Interviewee may express boredom/lose attention; this is not a bad thing (for the intents of data collection).

Interviewee may skim through the model(s') thought process/completely ignore it. This is completely fine as well; we want them to interact with these models as organically as possible.

Interviewee may pay 'too close' attention to the model, e.g. reading the output out loud due to the evaluation setting and the presence of an interviewer. If you sense that the interviewee is interacting inorganically with the model, feel free to encourage them to

relax/be less verbose.

Interviewee may be very quiet throughout the process. If you notice any behaviors you want elaboration or explanation on, feel free to query them as these behaviors pop up.

Structured Interview [5-10 min]:
“Now, we’re going to move on to an interview of your experiences with the evaluation today.”

“We’d first like to ask how you felt about the detail, depth, and response length of the responses generated by our model vs. the competitor models.

How did you feel about the quality of our model’s responses relative to [the other model]?

Did you feel that the detail and depth of these responses was appropriate for your needs?

Did you feel that the length of these responses, including the thinking processes, was appropriate/helpful for completing your chosen task?”

“In regards to the innovations of our model, you may have noticed metrics such as the “complexity level” of your questions. Did you notice any difference in thinking length or answer verbosity as this complexity level increased/decreased?”

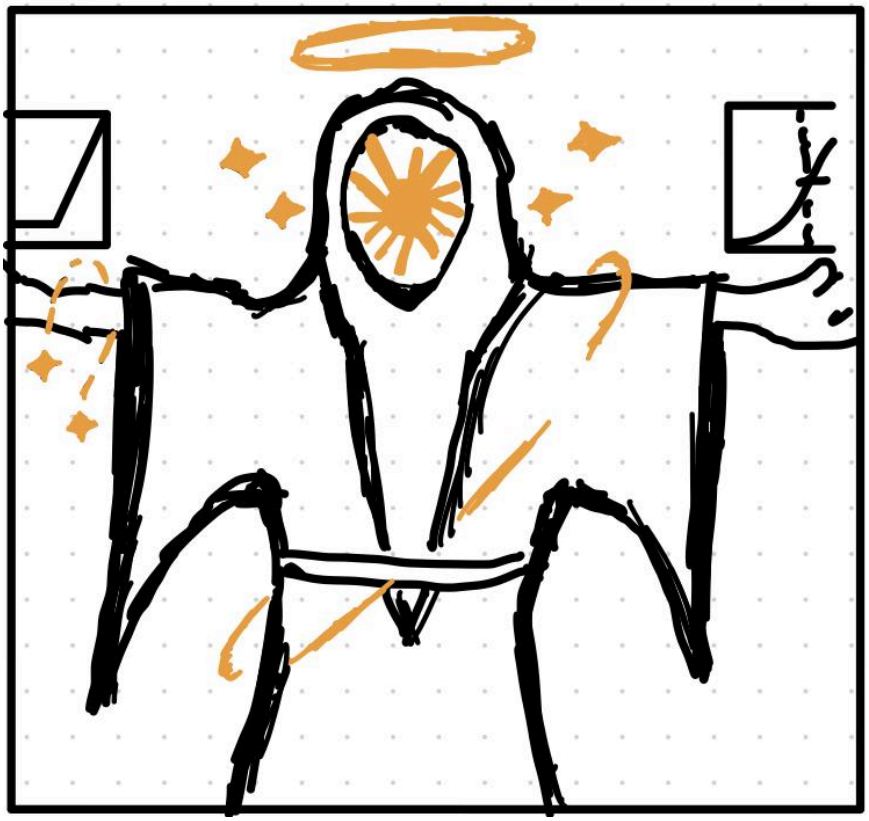
“Did you feel that the level of thought provided was appropriate and/or helpful for the completion of your workflow?”

[If DeepSeek]
“We had you try out DeepSeek, which is another example of a reasoning model that tends towards being very verbose in its thoughts. How would you compare the thinking you saw on DeepSeek versus the thinking you saw with our model?”

Post Interview:
On the Google Sheet corresponding to the answers on each Google Form, add the latency you measured to each of the interviewee’s responses. Paste the transcript of the interview somewhere that we can analyze later.

For fun [+0.5]

1. [+0.5] Name your system!
   Claude, the Sonnet, and the Wholly Spearmint (“CS W’s” == CS Dubz)
2. [+0.5] DEPTH: Design a logo for your system. Include a PNG in your repo. Add it to the README.



Did you use a generative AI tool for this assignment? If so, which tool(s) and how?

nah

How much time did you spend on this assignment

- as a group?

- individually?

Group: 6 hrs.

Ken: 1.5 hrs

Michael: 1.5 hrs
Matthew: 1.5 hrs