ELICITING HUMAN PREFERENCES WITH LANGUAGE MODELS

Belinda Z. Li[*] MIT CSAIL `bzl@mit.edu`
Alex Tamkin[*] Anthropic[†] `atamkin@cs.stanford.edu`
Noah D. Goodman Stanford `ngoodman@stanford.edu`
Jacob Andreas MIT CSAIL `jda@mit.edu`

ABSTRACT

Language models (LMs) can be directed to perform user- and context-dependent tasks by using labeled examples or natural language prompts. But selecting examples or writing prompts can be challenging—especially in tasks that require users to precisely articulate nebulous preferences or reason about complex edge cases. For such tasks, we introduce **generative active task elicitation (GATE)** for using *LMs themselves* to guide the task specification process. GATE is a learning framework in which models elicit and infer human preferences through free-form, language-based interaction with users. We identify prototypical challenges that users face when specifying preferences, and design three preference modeling tasks to study these challenges: content recommendation, moral reasoning, and email validation. In preregistered experiments, we show that LMs that learn to perform these tasks using GATE (by interactively querying users with open-ended questions) obtain preference specifications that are more informative than user-written prompts or examples. GATE matches existing task specification methods in the moral reasoning task, and significantly outperforms them in the content recommendation and email validation tasks. Users additionally report that interactive task elicitation requires less effort than prompting or example labeling and surfaces considerations that they did not anticipate on their own. Our findings suggest that LM-driven elicitation can be a powerful tool for aligning models to complex human preferences and values. [1]

1 INTRODUCTION

The complexity of human preferences makes them challenging to specify to machine learning systems. Consider the problem of designing a recommendation system for songs or news articles. First, system builders must develop a formal model of the potential factors influencing user preferences. Second, users must describe their preferences in a format that a learning algorithm can use to make future recommendations. Each of these steps requires mental effort and continual refinement by users and system builders. Until recently, the dominant approach in machine learning has specified preferences using *examples*: users first label a dataset with examples of the desired model behavior, then train a machine learning model on this dataset. This strategy has seen widespread use across diverse tasks, including image classification and question answering (Krizhevsky et al., 2012; Devlin et al., 2019). In more recent years, this paradigm has changed with the advent of *instruction following* methods (Brown et al., 2020)—by pre-training langauge models (LMs) on large-scale text corpora, it is possible to induce desired behaviors by conditioning only on natural language task specifications, in tasks as diverse as code generation and text summarization.

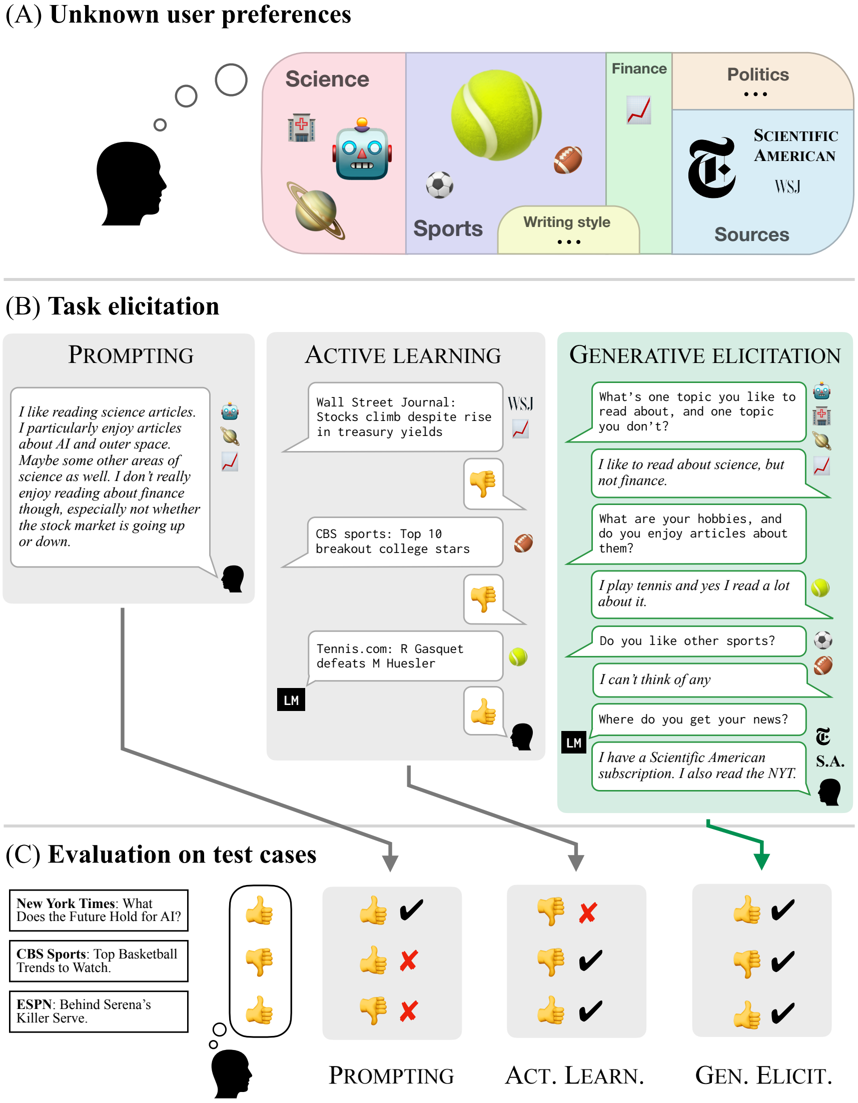
However, this progress has also accentuated the challenges described above: complex behaviors require an increasing amount of *prompt engineering* or *dataset design* to overcome the imprecision of natural language and prevent models from misunderstanding or misgeneralizing from spurious features of prompts or examples. For example, a user who says they enjoy reading tennis articles

---

[*] Equal contribution. Author order decided via coin flip.

[†] Work performed while at Stanford University.

[1] Code is ava/ilable at `https://github.com/alextamkin/generative-elicitation`



I like reading science articles. I particularly enjoy articles about AI and outer space. Maybe some other areas of science as well. I don't really enjoy reading about finance though, especially not whether the stock market is going up or down.

`Wall Street Journal: Stocks climb despite rise in treasury yields`

`CBS sports: Top 10 breakout college stars`

`Tennis.com: R Gasquet defeats M Huesler`

`What's one topic you like to read about, and one topic you don't?`

I like to read about science, but not finance.

`What are your hobbies, and do you enjoy articles about them?`

I play tennis and yes I read a lot about it.

`Do you like other sports?`

I can't think of any

`Where do you get your news?`

I have a Scientific American subscription. I also read the NYT.

New York Times: What Does the Future Hold for AI?

CBS Sports: Top Basketball Trends to Watch.

ESPN: Behind Serena's Killer Serve.

Figure 1: **Generative Active Task Elicitation (GATE) elicits user preferences through interactive, free-form questions, which can then be used in downstream decision-making.** Unlike non-interactive elicitation approaches (e.g., prompting), which rely entirely on the human to elucidate their preferences, generative elicitation is better able to probe nuances of human preferences. Unlike active learning approaches, generative elicitation can ask more generic, free-form questions. The three parts of this figure illustrate: **(A) Fuzzy user preferences:** A user wishes to translate their fuzzy preferences for how a task should be performed into a specification for a machine learning model. This is challenging because users lack perfect introspection, preferences can be difficult to specify in language, the specification needs to anticipate tricky real-world edge cases, and models may misgeneralize from provided examples or instructions. **(B) Task elicitation:** We consider various ways of eliciting these fuzzy preferences from users, including non-interactive prompting, active learning, and generative elicitation (GATE). **(C) Evaluation:** We evaluate methods on a held-out test set, scoring how well a language model predicted the true decisions made by the user.

could either be interested in the competitive tennis circuit or in improving their own serve. A few user-provided examples of tennis-related articles might fail to specify whether the user is interested in broader tennis content, such as tennis-themed satire. These challenges of *task ambiguity* (Finn et al., 2018; Tamkin et al., 2022a) loom large as models continue to be applied to more open-ended tasks and higher-stakes domains. Recent work (Don-Yehiya et al., 2024) has shown that in natural chatbot interactions, as many as 30% of the chats include explicit feedback, with the plurality of the feedback cases (35%) corresponding to when the user rephrased their last response, presumably due to the LM misinterpreting their initial query. Recent efforts to *align* LMs with human preferences have largely focused on training them to rank candidate outputs in accordance with fixed, pre-collected preference datasets (Ouyang et al., 2022; Rafailov et al., 2024; Hejna et al., 2024), or modeling preferences according to factors (Hu et al., 2023; Oh et al., 2024), but have overlooked the complexity of *eliciting* those preferences in the first place.

To address these challenges, we propose to use *LMs themselves* to help convert human preferences into automated decision-making systems. This paper describes **generative active task elicitation (GATE)**, a learning framework in which models elicit and infer user preferences through open-ended interaction. We describe several techniques for leveraging LMs to perform GATE—for example, by asking informative open-ended questions or generating edge cases for users to label.

Users face a number of different challenges when trying to specify or articulate their preferred solution to a task: they must identify the features of a solution that are relevant, evaluate tradeoffs in the relevant importance of these features, and reason about how they will interact in unexpected edge cases. To evaluate GATE and other specification methods, we design three preference modeling tasks to study these challenges: content recommendation, moral reasoning, and email validation. In pre-registered experiments, we find that LM-based task elicitation yields more equally or more accurate models than existing prompting or active learning techniques, while requiring comparable or less mental effort from users and surfacing novel considerations. More specifically, learning from LM-generated open-ended questions outperforms both prompting and active learning in the content recommendation and email specification tasks, and matches prompting in the moral reasoning task.

In summary, this paper introduces a new preference elicitation framework (GATE), a family of methods that perform GATE using pre-trained language models, and experimental evidence showing that these methods match or outperform existing prompting and labeling methods in modeling user preferences about language model behavior. Our results show that interactive, language-based task elicitation is a flexible and powerful tool for building personalized models, capable of overcoming many challenges inherent in prompt- and example-based methods.

2 ELICITING TASK PREFERENCES FROM HUMAN USERS

2.1 THE TASK ELICITATION FRAMEWORK

We study the problem of efficiently training a machine learning model to perform a task in accordance with a user's preferences. Throughout this paper, we use **task** to refer to a *context*- and *user*-dependent function $f : x \mapsto y$ that maps inputs x to outputs y. When building a personalized news article recommendation system, for example, x are articles and y are user preference scores for that article. Because different users may prefer different content, each user's individual preferences specify a distinct task: *content recommendation for Pat* and *content recommendation for Avery* are different tasks within the **domain** of content recommendation (Ziegler et al., 2020). To build such a model, we must collect some **task specification** from a human user (e.g., revealing what articles they are interested in). Current in-context and preference learning approaches admit a wide variety of specification types, including collections of labeled examples, natural language instructions, or combinations of the two.

What makes one type of specification preferable to another? Ideally, we would like specifications that are both (1) easy for humans to create and (2) informative to learners, enabling them to model human preferences accurately. Very abstractly, we seek a framework for gathering and learning from specifications that optimizes an objective:

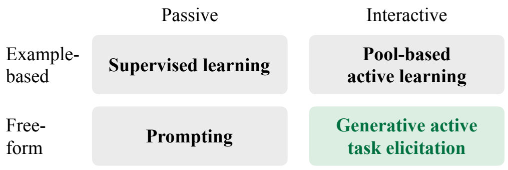
$$\alpha \cdot \text{human–predictor alignment} - \beta \cdot \text{specification cost} \tag{1}$$

where **human–predictor alignment** measures the extent to which model behaviors align with the human's desired behavior (e.g. user satisfaction with article recommendations), **specification cost** measures human effort in crafting the specification (e.g. time required for the user to specify their preferences about news articles), and α and β trade off between the two. Formally, let $\mathcal{H}_f$ denote a human user whose preferences are represented by a function f. We wish to design an **elicitation policy** $\mathcal{E}$ that interacts with $\mathcal{H}_f$ to produce a **task specification** s. This specification may then be input to a learning algorithm to produce a model $\hat{f}(s)$. Then, letting $C(\cdot)$ denote a scalar measure of specification cost, and $A(\cdot, \cdot)$ denote a measure of alignment distance between two predictors (lower is better), we wish to minimize (in expectation over the population of human users):

$$\mathbb{E}_{\mathcal{H}_f} \mathbb{E}_{s \sim \mathcal{E}(\mathcal{H}_f)} \left[\alpha \cdot C(s) + \beta \cdot A(f, \hat{f}(s)) \right] . \tag{2}$$

Here, C might measure the number of words the user typed to produce the specification s, while A might measure model–predictor agreement at the level of individual predictions from some population: $A(f, \hat{f}) = \mathbb{E}_x \| f(x) - \hat{f}(x) \|$. In general, appropriate definitions of C and A are domain-dependent; in this paper, our experiments compare the alignment of different predictors at a fixed cost. Evaluation of cost, alignment, and tradeoffs between them are discussed more in Section 5.

2.2 EXISTING PREFERENCE MODELLING PARADIGMS IN THE TASK ELICITATION FRAMEWORK



	Passive	Interactive
Example-based	**Supervised learning**	**Pool-based active learning**
Free-form	**Prompting**	**Generative active task elicitation**

Figure 2: Axes of variation in task elicitation.

Many existing approaches for preference learning can be described within the framework given in Figure 2. Understood as task elicitation procedures, existing approaches differ along two key axes: their level of *interactivity* and their level of *flexibility*. In interactive elicitation methods, queries can change depending on user responses (e.g., querying for the most useful information based on what is known thus far) while passive elicitation methods expect the user to provide specifications in a single shot. Example-based specification methods ask users to label a set of examples, while free-form elicitation approaches are less restrictive, allowing the user to provide a much wider range of inputs, including natural language instructions and explanations.

Supervised learning: passive, example-based In the most common supervised learning setup, the elicitation policy $\mathcal{E}$ simply instructs the human user $\mathcal{H}_f$ to generate a collection of labeled (input, output) pairs, after which $\hat{f}(s)$ is produced by fitting or fine-tuning a learned model using standard algorithms. This is an *example-based* process because the specification is provided via labeled examples and is *passive*, as the model does not interactively query the user to label additional data.

This category broadly covers in-context learning approaches that require users insert demonstrations in the context (Brown et al., 2020), as well as preference-learning approaches that ask users to rate model outputs on a scale (Ouyang et al., 2022), or select the best output in a pair of outputs (Rafailov et al., 2024).

Active learning: interactive, example-based In active learning, the elicitation policy is interactive. Users first assemble a fixed pool of unlabeled inputs x. Next, $\mathcal{E}$, selects from this pool an example whose label would be most informative. The user $\mathcal{H}_f$ provides a label for this example, then $\mathcal{E}$ selects the next-most-informative example, and so on (Cohn et al., 1994; Dagan & Engelson, 1995; Lewis & Gale, 1994; Settles, 2009). Finally, $\hat{f}(s)$ is trained as in supervised methods. Optimal experiment design methods (Emery & Nenarokomov, 1998) may be viewed as generalizations of this paradigm in which inputs x are generated rather than selected. *Interactive* processes enable the model to query for examples that may resolve uncertainty or ambiguity in the task specification (Tamkin et al., 2022b).

Recent approaches to in-context learning have explored various sampling techniques to determine which samples to insert in-context (Margatina et al., 2023a). Furthermore, preference learning approaches have explored *actively* selecting samples to label (Muldrew et al., 2024).

Prompting: passive, free-form Modern pre-trained models allow for specifying tasks in more flexible ways than simply labeling examples. For example, models can be conditioned with a *prompt* describing the user's intended task in natural language (Brown et al., 2020), or even a mix of language and image inputs (Alayrac et al., 2022). As with supervised learning, the labeling policy $\mathcal{E}$ here is simply an instruction to write a natural language task description (s), but the final predictor $\hat{f}(s)$ is produced by passing s to a pre-trained language model.

This category covers instruction-tuning techniques which improve LM's ability to follow free-form instructions in-context (Zhang et al., 2024), and preference learning techniques that learn from verbal feedback (Stephan et al., 2024).

3 GENERATIVE ACTIVE TASK ELICITATION

All of the methods above have important drawbacks: the burden typically falls upon the user to ensure that prompts or example sets are truly comprehensive specifications of their preferences, as any lack of clarity in the prompt could lead to task ambiguity (Tamkin et al., 2022a), resulting in undesired behavior during deployment. Resolving task ambiguity by crafting better prompts is challenging and time-consuming due to the difficulties of articulating nebulous personal preferences and anticipating edge cases that will emerge during deployment time.

However, one quadrant of Fig. 2 is not occupied by any of the aforementioned approaches: there is currently no method that leverages the flexibility of a free-form specification while using interaction to resolve uncertainty. We explore whether it is possible to combine the flexibility and richness of prompting-based specifications with the advantages of interactive methods such as active learning, by having a model interactively query users for these rich specifications. We term this family of methods **generative active task elicitation** (**GATE**).

3.1 METHODS FOR GATE

The effectiveness of language models (LMs) for understanding and producing free-form text suggests that they may be capable of eliciting and understanding user preferences. In this paper, we thus experiment with a family of GATE methods in which LMs serve as the backbone for both the elicitation policy $\mathcal{E}$ and the predictor $\hat{f}(s)$. See Figure 1 for examples. In particular, we implement the elicitation policy $\mathcal{E}$ by prompting an LM to ask the user questions while conditioning on the history of previous questions and answers. To make predictions $\hat{f}(s)$, an LM is prompted to predict a label conditioned on an input x and a complete elicitation transcript s provided as input. We experiment with several different information gathering policies, realized by simply prompting an LM to ask different kinds of questions:[2]

Generative active learning The LM generates examples for the user to label. This approach has the advantage of providing concrete scenarios to the user, including some they may not have considered *a priori*. For example, for content recommendation, the LM might generate the question: *Are you interested in the following article? The Art of Fusion Cuisine: Mixing Cultures and Flavors [...].*

Generative yes-or-no questions We restrict the LM to generating binary yes-or-no questions. This approach enables the model to elicit more abstract preferences while still being easy for the user to answer. For example, the model might probe a user's preferences by asking: *Do you enjoy reading articles about health and wellness?*

Generative open-ended questions The LM generates arbitrary questions requiring free-form natural language responses. This enables the LM to elicit the broadest and most abstract pieces of knowledge at the potential cost of being overly broad or challenging for the user to answer. For example, the LM might generate the question: *What hobbies or activities do you enjoy in your free time[...] and why do these hobbies or activities captivate you?*

---

[2] We also tried an *open-ended conversations* setting in early experiments, where we allowed users to freely converse with a LLM. However, we found that this freedom actually caused the LM to display some non-well-formed behavior, e.g. devolving into repetition, incoherence, or getting sidetracked by the most recent user response and losing track of the main task. Restricting the format to a particular type of question kept this behavior more at bay.

The user is not constrained in their response in any of the above settings; they are free to provide as much detail as they want. We present example elicitation transcripts for each policy in Figure 8.

4 EXPERIMENT SETUP

We consider preference learning problems in three different domains to evaluate our generative active task elicitation methods. A common feature of these domains is that they do not feature a single correct behavior that could be learned during LM pre-training; instead, models must elicit an individual human's preferences in order to make accurate predictions. We allow each human user ($\mathcal{H}$ in Eq. (2)) to interact open-endedly with an elicitation policy $\mathcal{E}$ for five minutes, resulting in a specification s. Next, humans and learned models $\hat{f}(s)$ independently label a set of held-out examples. Finally, we measure agreement between each human study participant and the LM conditioned on that user's specification s. See Figure 8 for examples of environments and dialogues.[3]

4.1 DOMAINS AND DATASETS

There are several distinct challenges that arise when users attempt to specify their preferred model behavior on a target task. In some cases, users may find it difficult to express tradeoffs between different features of model behavior (e.g. in the recommendation domain, the relative importance of a user's preference for short articles and articles about current events). In some cases, they may find it difficult to even articulate what the relevant features (e.g. length and topic) are, or to predict how they will correlate and interact in natural data. Preference modeling may also serve a variety of distinct functions in a larger automated decision-making system: understanding user's personal needs (e.g. for recommender systems), understanding their desired model behavior (e.g. for language model alignment), or probing the user's prior understanding of a task (e.g. as a prerequisite for teaching). We evaluate GATE in three domains designed to capture these various challenges:

Content Recommendation In this domain (which has served as a running example), individual users' preferences about the articles' content and style vary widely and may involve complex tradeoffs. After eliciting specifications from users, models' and users independently rate a set of articles (which are not available to users or learners during the elicitation phase). We constructed this evaluation set by collecting 16 popular online newspaper and magazine articles as test cases; ratings are based on the article's source website, title, and a one-paragraph description. Models are evaluated on their accuracy in identifying users' favorite articles from this set.

Moral Reasoning Moral preferences can be deeply personal and hard to articulate; indeed, even the features of moral scenarios that factor into humans' judgments can be difficult for them to verbalize. As a test-bed for eliciting moral values, we consider the question of when (if ever) it is ethical to steal a loaf of bread. After elicitation, users and models are presented with textual descriptions of scenarios in which someone wishes to steal a loaf of bread (e.g. if they are hungry, or if the bread was previously stolen from them) and asked to judge whether stealing the bread would be morally acceptable. Judgments are collected on 28 moral scenarios that were written manually by the authors as test cases. The authors designed these test cases by considering wide range of moral philosophies, including (but not limited to) deontology, utilitarianism, virtue ethics, animal welfare, etc. These test cases were designed in similar ways to moral vignettes in prior work (Jin et al., 2022), but shorter to ensure participants could annotate all scenarios in only a few minutes. Models are evaluated on their ability to predict human judgments about these scenarios.

Email Verification Lastly, we may wish to elicit user's degree of knowledge about a task in order to tailor teaching, or other communication, in accordance with their beliefs. Email verification is an example domain with many edge cases and requirements that a layperson may only have incomplete understanding of. For example, people may have differing prior expectations about what characters may be allowed in emails, how long they may be, etc. Here, we evaluate models' ability to predict user judgments about the well-formedness of emails. After elicitation, users and models are presented with a set of 54 test cases, and asked to label them as well-formed or malformed. The test cases were written by the authors based on the international standard outlined in IETF RFC

---

[3] The preregistration for our experiments and analyses can be found at: `https://osf.io/5v6nd/`.

6854, and carefully designed to probe for a diverse set of requirements, including special characters, character counts, foreign-language characters, subaddresses, in domains and local parts, etc. Models are evaluated on these test cases based on their ability to predict human well-formedness judgments.

A general note on evaluation: the test set in each domain is small by construction, to ensure that each human participant can fully label all test cases, and all results are comparable. We obtain a *complete* labeling of an evaluation set from *every one* of the 388 participants in our experiments, which in turn allows us to establish statistically significant differences between elicitation methods.

4.2 HUMAN INTERACTION

Human participants in these experiments were recruited from English-speaking users of Prolific. For the email validation task, we additionally recruited participants from several computer science programs at US universities. We recruited 20–30 participants for each domain-method pair (6 elicitation methods across 3 domains), for a total of 388 participants. We collected 1 transcript per participant. Participants were paid an average of $12/hr. Our experiments received IRB approval, and all participants consented to having their data used for our experiments. The breakdown of the number of participants allocated to each scenario and method can be found in Appendix B.1. Details of the user interface used in experiments may be found in Appendix B.2.

4.3 MODELING DETAILS

We use the GPT-4 model (`gpt-4-0613` snapshot; OpenAI, 2023) to both elicit user preferences (as an elicitation policy $\mathcal{E}$) and make predictions based on the elicited preferences (as a predictor $\hat{f}(s)$). We additionally run experiments on Mixtral, an open-source LM, in Appendix C.4. To elicit user preferences, we prompt GPT-4 with a domain description and the current interaction history, and instruct it to generate an informative but easy-to-answer edge case (for generative active learning) or question (for generative yes-or-no questions and generative open-ended questions). To make predictions, we prompt GPT-4 with the task specification s and a test sample x and ask it to generate a prediction for the test sample. The full text of the prompts can be found in Appendix A.

4.4 BASELINE METHODS

We compare GATE with several baseline approaches for specifying tasks. Here, the elicitation policy $\mathcal{E}$ is not parameterized by an LM, but constructed by the user or based on a pool of real examples.

Supervised learning We consider supervised learning as a baseline, as described in Section 2.2. We randomly present participants with questions from a large pool of examples and ask them to annotate up to the five-minute time limit. We study this approach exclusively in the content recommendation domain because pools of examples are not readily available in the other two domains. We use the Microsoft News Dataset (Wu et al., 2020) as our pool for this domain, a dataset of 160k news articles with descriptions. The license terms for research use of this dataset can be found at `https://github.com/msnews/MIND/blob/master/MSRLicense_Data.pdf`. We use the data consistent with the terms in “1) Use Rights”.

In-context active learning As a baseline active learning approach, we consider a pool-based active learning approach, as described in Section 2.2. For the elicitation policy, we use the diversity-based sampling approach of Margatina et al. (2023b); we first cluster the examples using a Sentence-BERT embedding model (Reimers & Gurevych, 2019) into 15 different clusters, then iteratively ask questions from each cluster in a round-robin fashion, up until the five-minute time limit.[4] This baseline is intended to capture the difficulty of selecting informative examples from a pool of unlabeled examples relative to generating informative examples from scratch. As with supervised learning, we study this approach exclusively in content recommendation.

---

[4] Margatina et al. (2023b) explored several different popular active learning sampling approaches for in-context learning (including random, uncertainty, and diversity sampling) and found little difference in empirical performance between them. We also ran exploratory model-model experiments in our domains and found no significant difference between these three sampling strategies. See details in Appendix D.

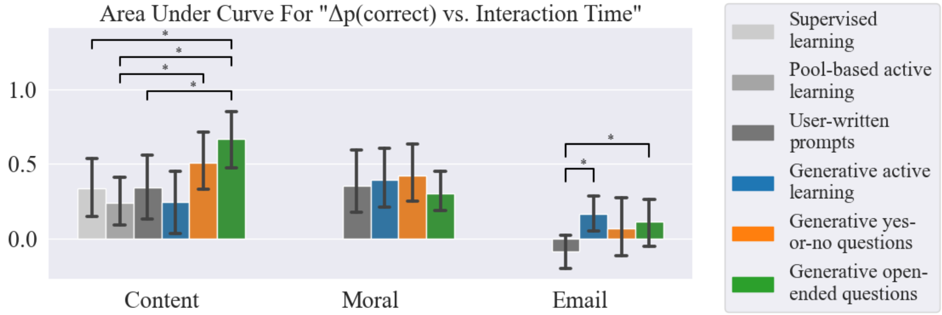


Figure 3: **Across three domains, our LM-prompting implementations of GATE are generally able to elicit human preferences beyond baseline supervised learning, active learning, or human-written prompts.** We measure the Area Under the "$\Delta p(correct)$ vs. Interaction time" Curve, which gives us a time-normalized metric for how well (and how quickly) each elicitation method is at aligning with human preferences. While GATE methods generally outperform the baseline methods as well as no interaction (represented by a $\Delta p(correct)$ of 0), we are only able to establish statistical significance between GATE and baselines in the content recommendation and email verification domains. * indicates a statistically significant difference ($p < 0.05$).

User-written prompts As a baseline that does not use interactive elicitation, we ask participants to write a short paragraph describing their preferences for the task for up to five minutes. We then use the text of this paragraph to prompt a LM to make decisions. This baseline is intended to capture the difficulty of specifying preferences in writing, both in terms of the effort it takes to write the paragraph and difficulty of writing a paragraph that fully specifies one's preferences.

4.5 EVALUATION AND METRICS

We measure how well models can predict the *probability* that users will answer questions a certain way, which we call $p(\text{correct})$. Specifically, we prompt the model with the interaction history s as a single test case, then ask the model to output a real-valued probability that a user would answer "yes" to the test case (e.g. the probability the user likes an article for content recommendation), which we call p_{LM}. This probability is outputted in token space as a number between 0.0 and 1.0, similar to past work (Branwen, 2020; Lin et al., 2022). The exact prompts we use for predicting probabilities can be found in Appendix A.2.

We define $p(\text{correct})$ as the probability the model assigns to the user-preferred answer. For example, if $p_{LM} = \texttt{0.8}$ for a given question, then $p(\text{correct})$ would be 0.8 if the user's answer were "yes" to the same question, and 0.2 if the user's answer was "no".

We use this metric instead of accuracy because we found modeling the uncertainty in (our estimate of) user's preferences was a more informative metric than predicting exact user decisions. In pilot experiments prompting the LM to predict binary yes/no decisions, we found this resulted in skewed predictions where the LM would predict one of 'yes' or 'no' for the entire test set, perhaps due to miscalibration of the model's implicit decision threshold. Furthermore, at the time of writing, token probabilities for GPT-4 were not available via the OpenAI API. That said, we also discuss and report a classification-based metric in Appendix C.2.

Given $p(\text{correct})$, we compute:

Area under the p(correct)-time curve We do not just care about the total information elicited, but about *how quickly* good information is elicited. That is to say, if two methods arrived at the same $p(\text{correct})$ at the end of five minutes, we want to reward the method that arrived a higher $p(\text{correct})$ faster. To do this, we compute the average change in $p(\text{correct})$ after *every minute* of human elicitation time (conditioning on the state of the transcript at that time). This produces a curve where the x-axis is time, and the y-axis is the average change in $p(\text{correct})$. By taking the

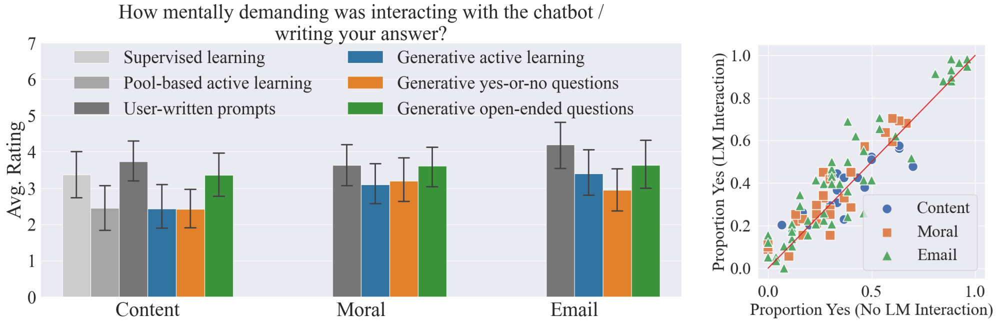


Figure 4: **Left: GATE methods are equally or less mentally demanding than other methods.** We plot the perceived mental demand across methods and domains (higher = greater mental demand). **Right: Language model elicitation does not shift human preferences.** We plot the proportion of participants who answered "yes" to each test question, comparing no LM interaction (user-written prompts) to LM interaction (GATE) elicitation. The red line is the $y = x$ curve, which serves as a guideline to see how well humans' no-LM interaction preferences align with their preferences post-LM interaction (if they align perfectly, the points should fall along this curve). We see that the points generally hover around this curve.

total area beneath this curve (AUC), we reward methods that arrive at higher $p(\text{correct})$ faster. We also plot the *final, average* $p(\text{correct})$'s after five minutes in Appendix C.3.

Rating of perceived effort across elicitation policies In addition to these performance-based metrics, we also ask users to rate how difficult they found the elicitation process to be. Specifically, we asked users "*How mentally demanding was writing your answer?*" in the non-interactive-elicitation setting, and "*How mentally demanding was interacting with the chatbot?*" in all elicitation settings (which include all other settings from Section 2.2). The "mentally demanding" wording was taken from the NASA TLX (Hart & Staveland, 1988). The question was assessed via a Likert scale from 1 (Very Little) to 7 (Very High). We also consider several additional questions to assess other usability tradeoffs. See Appendix E for the full list.

5 RESULTS

Evaluation results are shown in Figures 3 and 4. We plot the 95% confidence intervals of each setting as error bars, and perform permutation tests between each (baseline method, GATE method) pair to establish significance. Additional results, including sample conversations, can be found in Appendix C. Additional analyses can be found in Appendix D.3. Our results show that GATE methods...

...successfully elicit human preferences. GATE improves over no elicitation, where the model is prompted to make decisions before any user interaction. This is the case across all domains studied and all GATE methods (a positive score in Figure 3), with significance at the 0.05 level for all but the email domain, where only generative active learning was significant.

...are comparable to or better than other elicitation methods. The variant of GATE that asks open-ended questions matches (in the moral reasoning domain) or outperforms (in other domains) all other supervised learning, active learning, and prompting-based task elicitation methods. In the content recommendation and email domains, some other GATE variant also outperforms these baselines.

...are equally or less mentally demanding than user-written prompts. As shown in Figure 4 (left), users generally find interactive elicitation methods to be less mentally demanding, especially ones that involve labeling samples or answering yes/no questions, than non-interactive prompting.

We additionally run some of the settings using an open-source LM, Mixtral, which can be found in Appendix C.4. Our results show that Mixtral performs comparably to GPT-4, indicating that open-source models can be used in place of GPT-4 for GATE.

6 OTHER RELATED WORK

A fundamental challenge across many fields is how to obtain information about people's nebulous thoughts, preferences, and goals (Ericsson & Simon, 1980; Henderson et al., 1995; Christel & Kang, 1992; Zowghi & Coulin, 2005; Pacheco et al., 2018). Many works attempt to computationally describe or query human preferences, through bandits, Bayesian methods, inverse reinforcement learning, generative modeling, and more (Robbins, 1952; Yue et al., 2012; Chajewska et al., 2000; Emery & Nenarokomov, 1998; Ng et al., 2000; Hadfield-Menell et al., 2016; Mulla & Gharpure, 2023; Zhu & Bento, 2017). Most relevant to our work is active learning, which centers on how models can choose useful data points to learn from (Lewis & Catlett, 1994; Settles & Craven, 2008; Settles, 2009; Houlsby et al., 2011; Tamkin et al., 2022b). We extend this line of investigation to the generative setting, clarifying user intent by querying a user with *generated* examples and questions.

7 DISCUSSION AND CONCLUSION

We introduced the GATE framework to interactively elicit preferences from human users with free-form queries and answers. We presented initial evidence that LMs can successfully implement GATE to elicit human preferences (sometimes) more accurately and with less effort than supervised learning, active learning, or prompting-based approaches. There are many ways to expand on our implementation of GATE: Future work may explore more principled methods for elicitation, for example, integrating explicit notions of uncertainty. Second, larger models may be more capable elicitors: future work can explore scaling laws for elicitation. Finally, many real-world tasks such as software design and legal/medical decision-making present a richer set of constraints and edge cases. These applications thus offer a rich space of possible extensions of GATE.

LIMITATIONS

In this work, our exploration of GATE methods has been limited prompt-based approaches, and no explicit optimization of the objective in Equation (2). Future work can examine different ways of implementing free-form interactive querying, including approaches that might combine explicit optimization with the flexibility of language models.

In our human experiments (Section 5), we did not have the budget to survey a massive number of humans for human experiments. Thus, we were unable to establish statistical significance of GATE above baselines in certain domains. Furthermore, our sample of humans may be biased, as all of them speak English and are from the United States. This means that we have likely not captured the full spectrum of human preferences.

Finally, we would like note that our moral reasoning domain is very simplistic, and may be unable to capture all the nuances of human moral preference. This paper also does not endorse aligning to every potential human preference, understanding there are ethical risks to doing so. Overall, designers of public-facing systems that make decisions may wish to implement safeguards against allowing anyone to specify moral judgments.

ETHICAL CONSIDERATIONS

Our work presents several potential ethical benefits and risks. Examples of benefits include making it easier for software designers to incorporate nuanced user preferences, or empowering people with rare preferences or preferences that have historically not been considered when building software systems. In addition, improving the effort-performance ratio, especially by requiring less user typing, may help make language models more accessible to users with less time, familiarity with language models, or physical ability to use such systems. However, this direction carries risks as well. In particular, work on *thin slicing* (Ambady & Rosenthal, 1992) has demonstrated that small amounts of information about a user can sometimes be used to predict a broader range of personal characteristics, raising potential privacy considerations. The interactive nature of GATE also risks increasing *automation bias* (Goddard et al., 2012), where users place undue weight on a model's predictions. However, further work is necessary to establish if or when these risks are more significant for GATE than for prompting-based approaches to steering language models.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under grants IIS-2212310 and IIS-2331117, as well as an MIT Artificial Intelligence for Augmentation and Productivity Seed Grant. BZL is funded by an NDSEG Fellowship. We would like to thank Andi Peng and Leshem Choshen for feedback on earlier drafts of the paper.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: A visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Nalini Ambady and Robert Rosenthal. Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis. *Psychological bulletin*, 111(2):256, 1992.

Gwern Branwen. GPT-3 nonfiction — Calibration, 2020. URL `https://www.gwern.net/GPT-3-nonfiction#calibration`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Urszula Chajewska, Daphne Koller, and Ronald Parr. Making rational decisions using adaptive utility elicitation. In *AAAI/IAAI*, pp. 363–369, 2000.

Michael G Christel and Kyo C Kang. Issues in requirements elicitation, 1992.

David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Mach. Learn.*, 15(2):201–221, may 1994. ISSN 0885-6125. doi: 10.1023/A:1022673506211. URL `https://doi.org/10.1023/A:1022673506211`.

Ido Dagan and Sean P. Engelson. Committee-based sampling for training probabilistic classifiers. In *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*, ICML'95, pp. 150–157, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603778.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

Shachar Don-Yehiya, Leshem Choshen, and Omri Abend. Learning from naturally occurring feedback, 2024. URL `https://arxiv.org/abs/2407.10944`.

Ashley F Emery and Aleksey V Nenarokomov. Optimal experiment design. *Measurement Science and Technology*, 9(6):864, 1998.

K Anders Ericsson and Herbert A Simon. Verbal reports as data. *Psychological review*, 87(3):215, 1980.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *Advances in neural information processing systems*, 31, 2018.

Kate Goddard, Abdul Roudsari, and Jeremy C Wyatt. Automation bias: A systematic review of frequency, effect mediators, and mitigators. *Journal of the American Medical Informatics Association*, 19(1):121–127, 2012.

Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In Peter A. Hancock and Najmedin Meshkati (eds.), *Human Mental Workload*, volume 52 of *Advances in Psychology*, pp. 139–183. North-Holland, 1988. doi: https://doi.org/10.1016/S0166-4115(08)62386-9. URL `https://www.sciencedirect.com/science/article/pii/S0166411508623869`.

Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W. Bradley Knox, and Dorsa Sadigh. Contrastive preference learning: Learning from human feedback without rl, 2024. URL `https://arxiv.org/abs/2310.13639`.

Ron D Henderson, Mike C Smith, John Podd, and Hugo Varela-Alvarez. A comparison of the four prominent user-based methods for evaluating the usability of computer software. *Ergonomics*, 38 (10):2030–2044, 1995.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Yebowen Hu, Kaiqiang Song, Sangwoo Cho, Xiaoyang Wang, Hassan Foroosh, and Fei Liu. Decipherpref: Analyzing influential factors in human preference judgments via GPT-4, 2023. URL `https://arxiv.org/abs/2305.14702`.

Zhijing Jin, Sydney Levine, Fernando Gonzalez, Ojasv Kamal, Maarten Sap, Mrinmaya Sachan, Rada Mihalcea, Joshua Tenenbaum, and Bernhard Schölkopf. When to make exceptions: Exploring language models as accounts of human moral judgment. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL `https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

David Krueger, Tegan Maharaj, and Jan Leike. Hidden incentives for auto-induced distributional shift. *arXiv preprint arXiv:2009.09153*, 2020.

David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pp. 148–156. Elsevier, 1994.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pp. 3–12, Berlin, Heidelberg, 1994. Springer-Verlag. ISBN 038719889X.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*, 2022.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. Active learning principles for in-context learning with large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5011–5034, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.334. URL `https://aclanthology.org/2023.findings-emnlp.334`.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. Active learning principles for in-context learning with large language models, 2023b.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active preference learning for large language models, 2024. URL `https://arxiv.org/abs/2402.08114`.

Nikahat Mulla and Prachi Gharpure. Automatic question generation: A review of methodologies, datasets, evaluation metrics, and applications. *Progress in Artificial Intelligence*, 12(1):1–32, 2023.

Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, pp. 2, 2000.

Juhyun Oh, Eunsu Kim, Jiseon Kim, Wenda Xu, Inha Cha, William Yang Wang, and Alice Oh. Uncovering factor level preferences to improve human-model alignment, 2024. URL `https://arxiv.org/abs/2410.06965`.

OpenAI. Gpt-4 technical report, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf`.

Carla Pacheco, Ivan García, and Miryam Reyes. Requirements elicitation techniques: A systematic literature review based on the maturity of the techniques. *IET Software*, 12(4):365–378, 2018.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL `https://arxiv.org/abs/2305.18290`.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL `https://arxiv.org/abs/1908.10084`.

Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527 – 535, 1952.

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 1070–1079, 2008.

Moritz Stephan, Alexander Khazatsky, Eric Mitchell, Annie S Chen, Sheryl Hsu, Archit Sharma, and Chelsea Finn. RLVF: Learning from verbal feedback without overgeneralization, 2024. URL `https://arxiv.org/abs/2402.10893`.

Alex Tamkin, Kunal Handa, Avash Shrestha, and Noah Goodman. Task ambiguity in humans and language models. *arXiv preprint arXiv:2212.10711*, 2022a.

Alex Tamkin, Dat Nguyen, Salil Deshpande, Jesse Mu, and Noah Goodman. Active learning helps pretrained models learn the intended task. *Advances in Neural Information Processing Systems*, 35:28140–28153, 2022b.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3597–3606, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.331. URL `https://aclanthology.org/2020.acl-main.331`.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey, 2024. URL `https://arxiv.org/abs/2308.10792`.

Jia-Jie Zhu and José Bento. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*, 2017.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020.

Didar Zowghi and Chad Coulin. Requirements elicitation: A survey of techniques, approaches, and tools. *Engineering and managing software requirements*, pp. 19–46, 2005.

A GATE PROMPTS

A.1 ELICITATION PROMPTS

Below, we list the prompts we use for generating elicitation queries (to instantiate the elicitation policy $\mathcal{E}$) for the various GATE methods.

Generative active learning The prompt for generative active learning can be found below. Bracketed, italicized text are placeholders, e.g. for the elicitation transcript (so far). Color-coded text is domain-specific, e.g. the blue text represents text for the content recommendation domain, orange text represents text for the moral reasoning domain, and the green text represents text for the email validation domain.

Your task is to *[learn what topics a user is interested in reading online article about. People's interests are broad, so you should seek to understand their interests across many topics; in other words, go for breadth rather than depth. Do not assume a user has given a complete answer to any question, so make sure to keep probing different types of interests | learn under what conditions the user would believe it is or is not ethical to steal a loaf of bread | learn what rules a user believes a valid email address format must adhere to (e.g. for developing a regex format checker)]*.

Come up with a potential edge case to learn as much information as you can about what their desired behavior should be under different circumstances.
Make sure the edge case addresses different aspects of the system than the edge cases that have already been considered.

An example edge case is: *[Are you interested in the following article? Website Name: TechTalkDaily*
Title: 10 Must-Have Gadgets for 2023: A Comprehensive Guide
Description: Discover the top 10 gadgets that will revolutionize your daily life in 2023. From smart home devices to the latest wearables, we've got you covered with our comprehensive guide. | Situation: Is it ethical to steal a loaf of bread if you are starving to death and the bread would otherwise go to waste? | Should the following email be accepted? username@example.com]

Current cases:
[Elicitation transcript]

Generate the most informative edge case that, when answered, will reveal the most about the desired behavior beyond what has already been queried for above. Generate the edge case in the following format, and nothing else: "*[Are you interested in the following article? [edge case] | Situation: [edge case] | Should the following be accepted? [edge case]]*"

Generating Questions The prompt for generating both yes-or-no and open-ended questions can be found below. Once again, bracketed, italicized text are placeholders and color-coding indicates text for specific domains.

Your task is to *[learn what topics a user is interested in reading online article about. People's interests are broad, so you should seek to understand their interests across many topics; in other words, go for breadth rather than depth. Do not assume a user has given a complete answer to any question, so make sure to keep probing different types of interests | learn under what conditions the user would believe it is or is not ethical to steal a loaf of bread | learn what rules a user believes a valid email address format must adhere to (e.g. for developing a regex format checker)]*.

Previous questions:
[Elicitation transcript]

Generate the most informative *[yes/no question | open-ended question]* that, when answered, will reveal the most about the desired behavior beyond what has already been queried for above. Make sure your question addresses different aspects of the implementation than the questions that have already been asked. At the same time however, the question should be bite-sized, and not ask for too much at once. Phrase your question in a way that is understandable to non-expert humans; do not use any jargon without explanation. Generate the *[yes/no question | open-ended question]* and nothing else:

A.2 DECISION-MAKING PROMPTS

Given an elicitation transcript, we prompt the LM as follows to serve as a predictor $\hat{f}(s)$ to make decisions about each test case.

Content Recommendation The prompt for content recommendation can be found below. Bracketed, italicized text are placeholders, for the elicitation transcript and the test case sample.

> A user has a particular set of preferences over what articles they would like to read. They have specified their preferences below:
> *[Eliciation transcript]*
>
> Based on these preferences, would the user be interested in reading the following article? Answer with a probability between 0 and 1, where 0 means “definitely not interested” and 1 means “definitely interested”. Only output the probability and nothing else. If uncertain, make your best guess.
> *[Test case sample]*

Moral Reasoning The prompt for moral reasoning can be found below. Once again, bracketed, italicized text are placeholders.

> A user has a particular ethical code they follow. The following response(s) represent when this user would believe it is ethical to steal a loaf of bread.
> *[Eliciation transcript]*
>
> "Based on these response(s), would the user believe it is ethical to steal a loaf of bread in the following situation? Answer with a probability between 0 and 1, where 0 means “definitely not” and 1 means “definitely yes”. Only output the probability and nothing else. If uncertain, make your best guess.
> *[Test case sample]*

Email Validation The prompt for email validation can be found below. Once again, bracketed, italicized text are placeholders.

> A user has a particular format of emails that they believe to be valid. The following answer(s) represent this user’s preferences of whether these emails adhere to their desired format.
> *[Eliciation transcript]*
>
> Based on the user’s preferences, does the following email adhere to the user’s desired format? Answer with a probability between 0 and 1, where 0 means “definitely not” and 1 means “definitely yes”. Only output the probability and nothing else. If uncertain, make your best guess.
> *[Test case sample]*

B EXPERIMENTAL DETAILS

B.1 NUMBER OF PARTICIPANTS

The number of participants we recruited for our study, for each elicitation method and domain, can be found in the table below.

B.2 USER INTERFACE DETAILS

Details about the UI we built for our experiments can be found below. Recall that the human studies proceeded in two parts: elicitation, followed by decision-making.

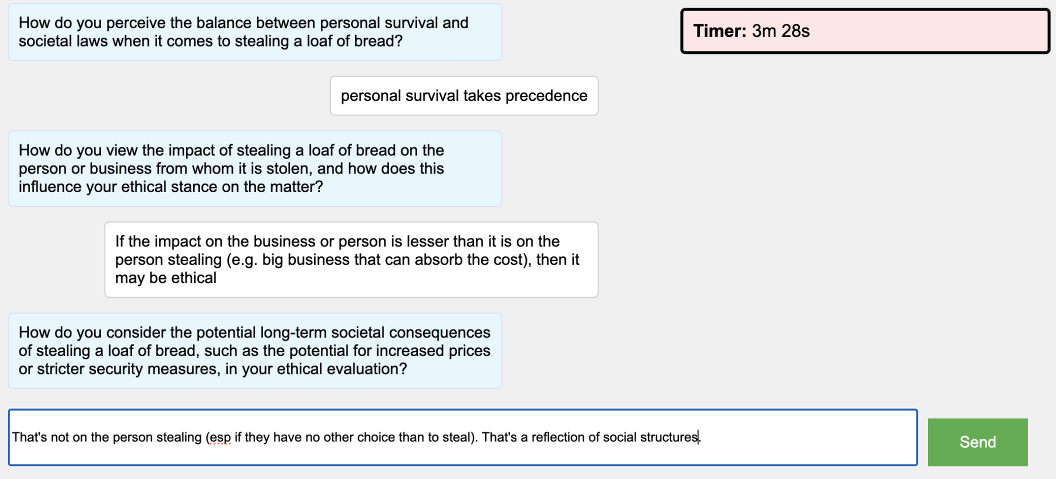
B.2.1 ELICITATION

For supervised learning, pool-based active learning, and the GATE methods, we had participants respond to a series of queries using the chatbot interface (Figure 5). For prompting, we had participants input a task description using the text-input interface (Figure 6).

The instructions for this phase can be found below.

	Content Recommendation	Moral Reasoning	Email Validation	Total
Supervised learning	30	-	-	30
Pool-based active learning	31	-	-	31
Prompting	30	30	26	86
Generative active learning	30	30	20	80
Generative yes-or-no questions	31	30	19	80
Generative open-ended questions	31	31	19	81
Total	183	121	84	388

Table 1: Breakdown of how many participants we recruit for each domain and elicitation method.



How do you perceive the balance between personal survival and societal laws when it comes to stealing a loaf of bread?

personal survival takes precedence

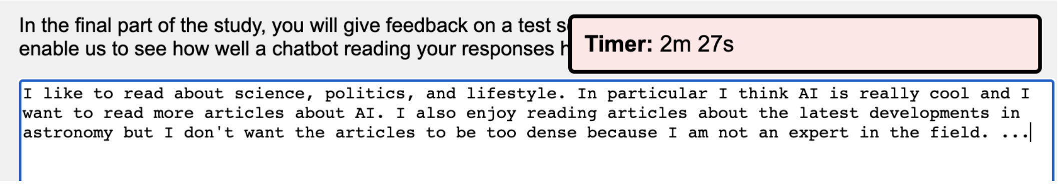
How do you view the impact of stealing a loaf of bread on the person or business from whom it is stolen, and how does this influence your ethical stance on the matter?

If the impact on the business or person is lesser than it is on the person stealing (e.g. big business that can absorb the cost), then it may be ethical

How do you consider the potential long-term societal consequences of stealing a loaf of bread, such as the potential for increased prices or stricter security measures, in your ethical evaluation?

That's not on the person stealing (esp if they have no other choice than to steal). That's a reflection of social structures

Figure 5: Chatbot UI built for elicitation phases of GATE methods, supervised learning, and pool-based active learning.



In the final part of the study, you will give feedback on a test s
enable us to see how well a chatbot reading your responses h

```
I like to read about science, politics, and lifestyle. In particular I think AI is really cool and I
want to read more articles about AI. I also enjoy reading articles about the latest developments in
astronomy but I don't want the articles to be too dense because I am not an expert in the field. ...
```

Figure 6: Text-input UI built for elicitation phase for prompting.

Content	We are testing a system for understanding people's interest in reading different kinds of online articles. For example, you might be interested in articles about some topics, but not about others.
Moral	We are testing a system for understanding people's fuzzy intuitions and preferences. In this experiment, we'll be capturing your moral intuitions about the act of stealing a loaf of bread, and whether there are certain cases where stealing may be morally permissible.
Email	We are testing a system for understanding people's fuzzy intuitions and preferences. In this activity, we're going to be looking at different strings of text and you'll be deciding if they look like they could be an email address or not. For example, most people would agree that "username@domain.com" looks like an email address, while "n12z5lFEN4" does not. However, the rules for what can be an email address can be very unusual, so what we're really interested in is your intuition on what an email address could look like. **Important:** We are not asking you to determine the rules for a *good* email address, or a *real (non-spam)* email address. We are simply asking about your intuition as to why certain strings look like email addresses and certain strings do not. **Tip:** in an email such as username@cs.stanford.edu, "username" is called the local-part of the email, while "cs.stanford.edu" is the domain. Furthermore, "cs" is a subdomain, and "edu" is a top-level domain.

Table 2: Domain-specific instructions presented to users for the elicitation phases.

Supervised Learning / Pool-based Active Learning We present users with the following instructions for both supervised learning and pool-based active learning. Bracketed, italicized text represent placeholders for domain-specific text. *[Domain instructions]* is a placeholder for the top-level instructions for each domain (see Table 2). Otherwise, blue text represents text for the content recommendation domain, orange text represents text for the moral reasoning domain, and green text represents text for the email validation domain.

[Domain instructions]

Try to answer in a way that accurately and comprehensively conveys your preferences, such that someone reading your responses can understand and make judgments as close to your own as possible. Feel free to respond naturally (you can use commas, short phrases, etc), and press [enter] to send your response. Note that the chatbot technology is imperfect, and you are free to avoid answering any questions that are overly broad or uncomfortable. When interacting with the chatbot, please avoid asking follow-up questions or engaging in open-ended dialogue as the chatbot is unable to respond to you.

Note: The chatbot will stop asking questions after 5 minutes, after which you can send your last response and you will be taken to the final part of the study.

In the final part of the study, you will give feedback on a test set of *[article headline and descriptions | moral situations | email addresses]*, which will enable us to see how well a chatbot reading your responses has learned *[what you like and dislike | your moral preferences | your email preferences]*.

Prompting We present users with the following instructions for prompting. Similar to above, bracketed, italicized text represent places where we insert domain-specific text.

[Domain instructions]

To the best of your ability, please explain all details about *[your preferences of what kinds of online articles you would like to read | your belief of when it is moral to steal a loaf of bread | your intuition of what makes email addresses look like email addresses]*, such that someone reading your responses can understand and make judgments as close to your own as possible. Try to be as detailed as possible. For example, if you were writing a regex that accepts only email-address-like strings, what might that regex look like? What are permissible / non-permissible symbols and characters, and in what positions?

Note: You will have up to 5 minutes to articulate your preferences. Please try to submit your response within that time. After you submit, you will be taken to the final part of the study.

In the final part of the study, you will give feedback on a test set of *[article headline and descriptions | moral situations | email addresses]*, which will enable us to see how well a chatbot reading your responses has learned *[what you like and dislike | your moral preferences | your email preferences]*.

GATE methods We present users with the following instructions for the three GATE methods (generative active learning, generative yes-or-no questions, generative open-ended questions). Once again, bracketed italicized text represent domain-specific text.

[Domain instructions]

This chatbot will ask you a series of questions about *[your preferences of what kinds of online articles you would like to read | your belief of when it is moral to steal a loaf of bread | your intuition of what makes email addresses look like email addresses]*. Try to answer in a way that accurately and comprehensively conveys your preferences, such that someone reading your responses can understand and make judgments as close to your own as possible. Feel free to respond naturally (you can use commas, short phrases, etc), and press [enter] to send your response. Note that the chatbot technology is imperfect, and you are free to avoid answering any questions that are overly broad or uncomfortable. When interacting with the chatbot, please avoid asking follow-up questions or engaging in open-ended dialogue as the chatbot is unable to respond to you.

Note: The chatbot will stop asking questions after 5 minutes, after which you can send your last response and you will be taken to the final part of the study.

In the final part of the study, you will give feedback on a test set of *[article headline and descriptions | moral situations | email addresses]*, which will enable us to see how well a chatbot reading your responses has learned *[what you like and dislike | your moral preferences | your email preferences]*.

B.2.2 DECISION-MAKING

For the decision-making phase, we presented users with the following instruction:

Content Recommendation

Please indicate whether you would like to read the following articles: “yes” if you would, “no” if you would not.

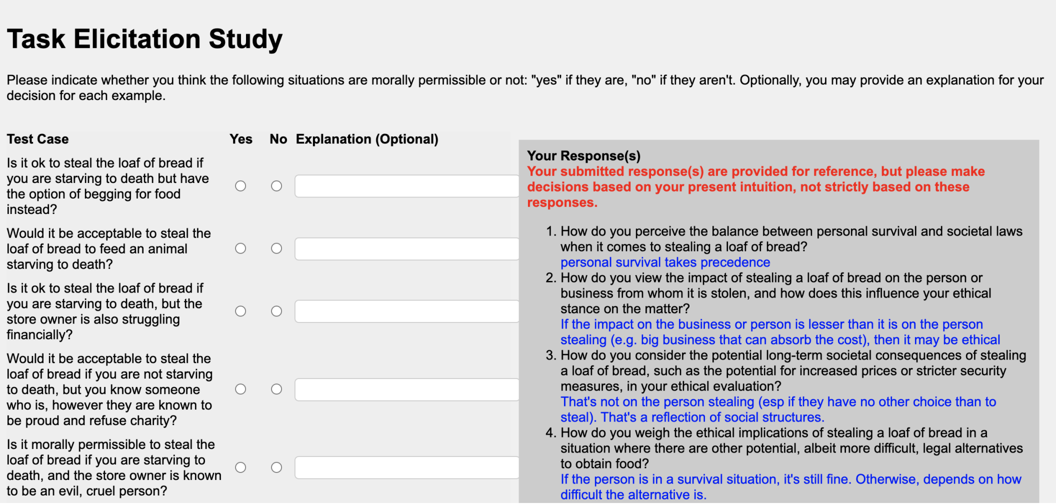
Moral Reasoning

Please indicate whether you think the following situations are morally permissible or not: “yes” if they are, “no” if they aren’t.

Email Validation

Please indicate whether you think the following strings look like reasonably well-formatted email addresses or not: “yes” if they do, “no” if they don’t.

Users are then presented with a list of test samples, and can use radio buttons to select whether each test-case sample is acceptable. See Figure 7.



Task Elicitation Study

Please indicate whether you think the following situations are morally permissible or not: "yes" if they are, "no" if they aren't. Optionally, you may provide an explanation for your decision for each example.

Test Case	Yes	No	Explanation (Optional)
Is it ok to steal the loaf of bread if you are starving to death but have the option of begging for food instead?	○	○	
Would it be acceptable to steal the loaf of bread to feed an animal starving to death?	○	○	
Is it ok to steal the loaf of bread if you are starving to death, but the store owner is also struggling financially?	○	○	
Would it be acceptable to steal the loaf of bread if you are not starving to death, but you know someone who is, however they are known to be proud and refuse charity?	○	○	
Is it morally permissible to steal the loaf of bread if you are starving to death, and the store owner is known to be an evil, cruel person?	○	○	

Your Response(s)

Your submitted response(s) are provided for reference, but please make decisions based on your present intuition, not strictly based on these responses.

1. How do you perceive the balance between personal survival and societal laws when it comes to stealing a loaf of bread?
   personal survival takes precedence
2. How do you view the impact of stealing a loaf of bread on the person or business from whom it is stolen, and how does this influence your ethical stance on the matter?
   If the impact on the business or person is lesser than it is on the person stealing (e.g. big business that can absorb the cost), then it may be ethical
3. How do you consider the potential long-term societal consequences of stealing a loaf of bread, such as the potential for increased prices or stricter security measures, in your ethical evaluation?
   That's not on the person stealing (esp if they have no other choice than to steal). That's a reflection of social structures.
4. How do you weigh the ethical implications of stealing a loaf of bread in a situation where there are other potential, albeit more difficult, legal alternatives to obtain food?
   If the person is in a survival situation, it's still fine. Otherwise, depends on how difficult the alternative is.

Figure 7: UI for the decision-making phase.

C ADDITIONAL RESULTS

C.1 SAMPLE TRANSCRIPTS

Sample transcripts of users interacting with the various generative active task elicitation methods can be found in Figure 8.

C.2 AUROC RESULTS

We measure AUROC over model-generated probabilities in addition to $\Delta p(\text{correct})$. Figure 9 is the analogous plot to Figure 3, but we measure the improvement in **AUROC** instead of $p(\text{correct})$, over interaction time, rewarding methods that achieve higher improvements in AUROC sooner.

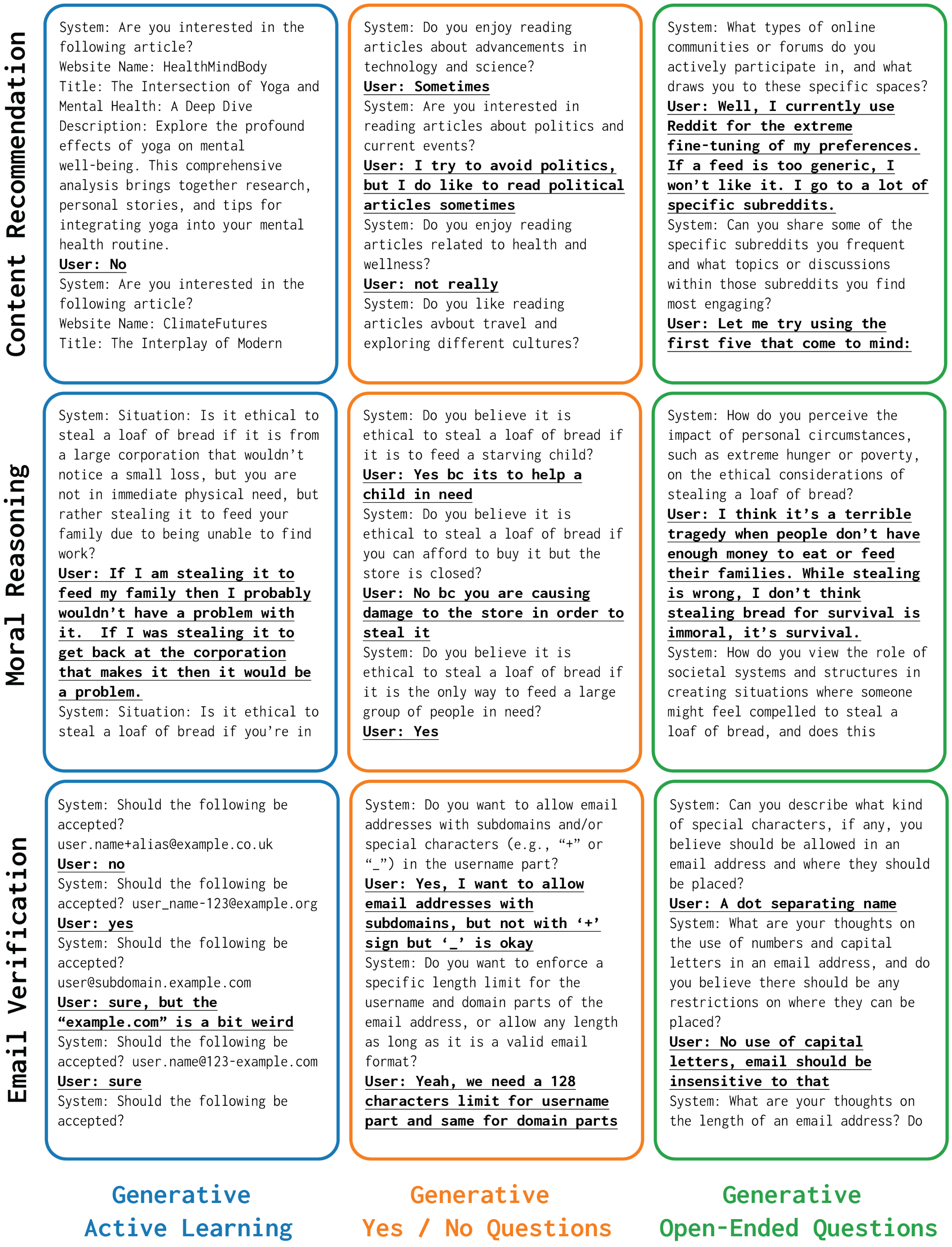
The general trends hold from Section 5: language models can elicit human preferences (beyond no interaction), and language model elicitation is comparable or better than other elicitation baselines. However, unlike the $p(\text{correct})$ metric, the AUROC metric is a simple classification-based metric. Due to potential miscalibration in LMs, making it difficult for them to output well-calibrated probabilities with the same threshold across questions, the overall improvements in this metric are lower (particularly for generative open-ended questions) and the variances are much higher. Thus, we see that it is harder to establish statistical significance using this metric.

C.3 FINAL $p(\text{CORRECT})$ RESULTS

In Figure 10, we plot the average final $p(\text{correct})$ after five minutes of interaction with each elicitation method. Overall, we find that GATE methods generally still outperform baselines in each setting, though the improvement is not always statistically significant with this metric. This indicates that GATE methods converge to a high performance quicker than baseline methods, then stagnate after a point, beyond which further interaction may not be worthwhile.

C.4 MIXTRAL RESULTS

To test the robustness of GATE to the choice of underlying language model (for elicitation and prediction), and particularly to test out how well open-source models perform relative to GPT-4, we try to replicate some of our experiments on Mixtral (`Mixtral-8x7B-Instruct-v0.1`), one of the state-of-the-art open-source models as of February 2024. We focus on the content recommendation setting as the one where GPT-4 models performed the best. Using the same prompts and overall setup as for GPT-4, we test Mixtral's ability to generate *open-ended* questions. We separately test


Content Recommendation
Moral Reasoning
Email Verification
Generative Active Learning
Generative Yes / No Questions
Generative Open-Ended Questions

System: Are you interested in the following article?
Website Name: HealthMindBody
Title: The Intersection of Yoga and Mental Health: A Deep Dive
Description: Explore the profound effects of yoga on mental well-being. This comprehensive analysis brings together research, personal stories, and tips for integrating yoga into your mental health routine.
User: No
System: Are you interested in the following article?
Website Name: ClimateFutures
Title: The Interplay of Modern

System: Do you enjoy reading articles about advancements in technology and science?
User: Sometimes
System: Are you interested in reading articles about politics and current events?
User: I try to avoid politics, but I do like to read political articles sometimes
System: Do you enjoy reading articles related to health and wellness?
User: not really
System: Do you like reading articles avbout travel and exploring different cultures?

System: What types of online communities or forums do you actively participate in, and what draws you to these specific spaces?
User: Well, I currently use Reddit for the extreme fine-tuning of my preferences. If a feed is too generic, I won't like it. I go to a lot of specific subreddits.
System: Can you share some of the specific subreddits you frequent and what topics or discussions within those subreddits you find most engaging?
User: Let me try using the first five that come to mind:

System: Situation: Is it ethical to steal a loaf of bread if it is from a large corporation that wouldn't notice a small loss, but you are not in immediate physical need, but rather stealing it to feed your family due to being unable to find work?
User: If I am stealing it to feed my family then I probably wouldn't have a problem with it. If I was stealing it to get back at the corporation that makes it then it would be a problem.
System: Situation: Is it ethical to steal a loaf of bread if you're in

System: Do you believe it is ethical to steal a loaf of bread if it is to feed a starving child?
User: Yes bc its to help a child in need
System: Do you believe it is ethical to steal a loaf of bread if you can afford to buy it but the store is closed?
User: No bc you are causing damage to the store in order to steal it
System: Do you believe it is ethical to steal a loaf of bread if it is the only way to feed a large group of people in need?
User: Yes

System: How do you perceive the impact of personal circumstances, such as extreme hunger or poverty, on the ethical considerations of stealing a loaf of bread?
User: I think it's a terrible tragedy when people don't have enough money to eat or feed their families. While stealing is wrong, I don't think stealing bread for survival is immoral, it's survival.
System: How do you view the role of societal systems and structures in creating situations where someone might feel compelled to steal a loaf of bread, and does this

System: Should the following be accepted?
user.name+alias@example.co.uk
User: no
System: Should the following be accepted? user_name-123@example.org
User: yes
System: Should the following be accepted?
user@subdomain.example.com
User: sure, but the "example.com" is a bit weird
System: Should the following be accepted? user.name@123-example.com
User: sure
System: Should the following be accepted?

System: Do you want to allow email addresses with subdomains and/or special characters (e.g., "+" or "_") in the username part?
User: Yes, I want to allow email addresses with subdomains, but not with '+' sign but '_' is okay
System: Do you want to enforce a specific length limit for the username and domain parts of the email address, or allow any length as long as it is a valid email format?
User: Yeah, we need a 128 characters limit for username part and same for domain parts

System: Can you describe what kind of special characters, if any, you believe should be allowed in an email address and where they should be placed?
User: A dot separating name
System: What are your thoughts on the use of numbers and capital letters in an email address, and do you believe there should be any restrictions on where they can be placed?
User: No use of capital letters, email should be insensitive to that
System: What are your thoughts on the length of an email address? Do

Figure 8: **Excerpts of real transcripts across the different domains and elicitation methods we investigate.** The `System` messages are generated by the language model, while the `User` messages are produced by human participants. Overall, the model is able to generate diverse and contextually-appropriate questions in each setting. See Sections 3.1 and 4.1 for more details on the domains and methods respectively.

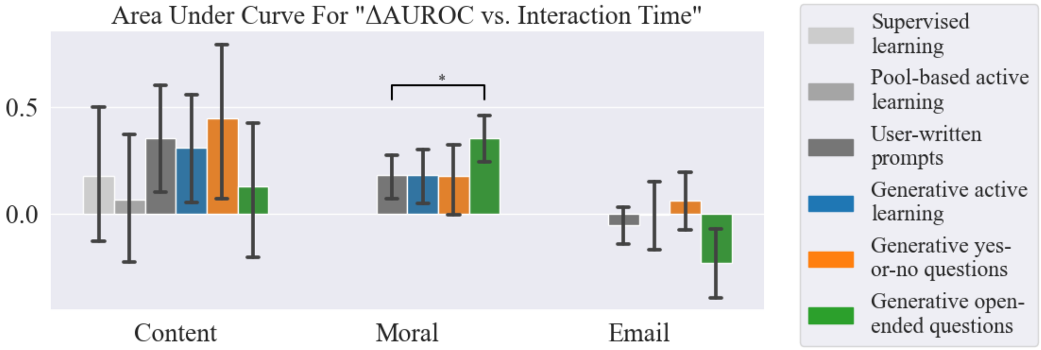


Figure 9: We plot the **Area Under the "ΔAUROC vs. Interaction time" Curve**, which gives us a metric for how well (and how quickly) each elicitation method is at aligning with human preferences. This plot is analogous to Figure 3, only we are using AUROC instead of $p(correct)$ for the alignment metric, which means that we are not measuring uncertainty. We see the same trends hold of GATE methods (generally) beating supervised learning, pool-based learning, and prompting approaches, while also beating no interaction ($\Delta\text{AUROC} = 0$) using this metric. However, we see generally smaller Δs over non-interaction using this metric, and higher variances, which make it harder to establish statistical significance.

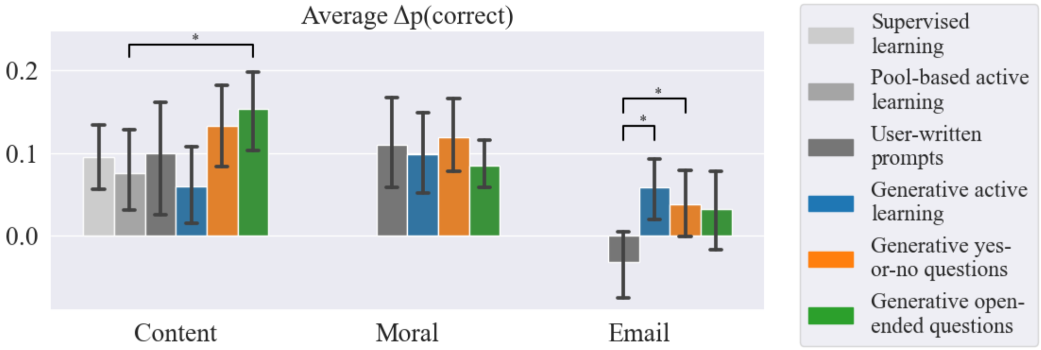


Figure 10: We plot the **average final** $p(\textbf{correct})$ after 5 minutes of interaction with each elicitation method. While we see the same general trends as in Figure 3, the improvements are not generally as statistically significant as when we were using AUROC, indicating that GATE methods converge to higher performance quicker than baseline methods.

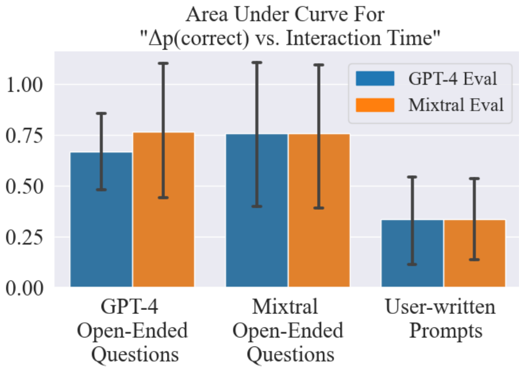


Figure 11: Effectiveness of Mixtral as an *elicitator* and a *predictor* in GATE.

Assistant: What hobbies or activities do you enjoy doing in your free time, and have you ever read any articles related to those interests?
User: I really love post apocalyptic fiction and I can't recall reading any articles in particular
Assistant: That's interesting! Have you ever explored any non-fiction articles or research related to post-apocalyptic scenarios, societal collapse, or emergency preparedness?
User: no, not really
Assistant: Have you ever come across any online articles or discussions about how post-apocalyptic fiction might reflect or influence real-world emergency planning or disaster response strategies?
User: no
Assistant: Have you ever looked into how post-apocalyptic fiction might inspire or impact emergency shelters, food storage, or water purification methods in real-world scenarios?
User: no, but it sound interesting
...[10 turns later]...
Assistant: Have you ever encountered any post-apocalyptic fiction that highlights unique or innovative approaches to education, healthcare, or governance in reimagined societies, and what intriguing ideas have you discovered that could potentially be applied to enhance resilience and adaptability in real-world communities?
User: can we talk about somethhing else? Let's chat about running. I love to run!
Assistant: That's great to hear about your love for running! Have you ever read any articles about how running or other physical activities can play a role in building resilience and adaptability during challenging times, either in a post-apocalyptic scenario or in real-world emergency preparedness?
User: Interesting, but no

out how effective Mixtral is as an elicitor and as a predictor. Thus we have six total settings, with three elicitation modes:

1. Mixtral yes-or-no questions
2. GPT-4 yes-or-no questions
3. User-written prompts

each of which are evaluated by two prediction models:

1. Mixtral predictor
2. GPT-4 predictor

Results are shown in Figure 11. We find that the Mixtral model performs comparably to GPT-4 as both an elicitor, and as a predictor, indicating that our results are robust across different choices of underlying language models. This is also a promising indication that open-source models can be used instead of GPT-4 for preference elicitation.

Qualitatively, we find that the open model likes to quickly hyperfixate on a topic once a user raises it as a topic of interest. An example can be found in Appendix C.4, where Mixtral, upon learning the user likes post-apocalyptic fiction on the first turn, pivots to only asking questions related to post-apocalyptic fiction.

C.5 GPT-4O PREDICTOR RESULTS

We run our collected transcripts using GPT-4o as a *predictor* (see Figure 12). We find generative open-ended questions still significantly outperforms all baselines in the content recommendation domain, though lags behind in the email validation domain. Qualitative analysis finds that GPT-4o is more sensitive to *noise* than GPT-4, having a higher proclivity to be affected by noisy labels in the transcript. For example, one user answered:

Assistant: Should an email address contain at least one period (.) after the @ symbol?
User: no

Then in the test cases answered:

Is “user@domain.edu" a valid email address?
User: yes

The noisy answer in the transcript caused GPT-4o to believe the user would answer “no” to the above test case, while GPT-4 predicted that the user would still answer “yes”.

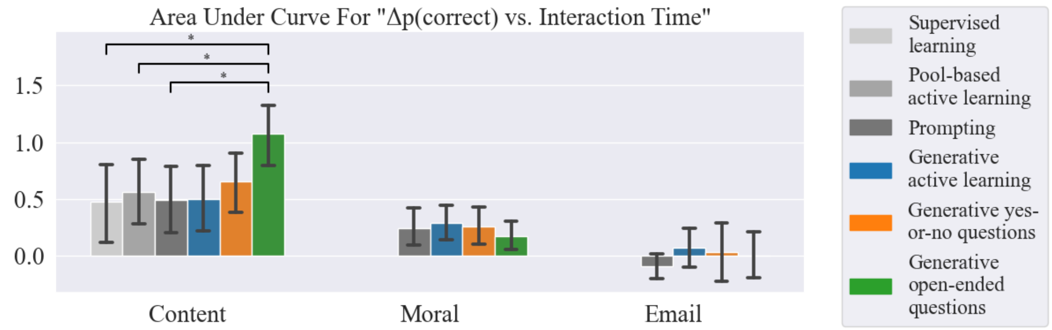


Figure 12: Using GPT-4o as a predictor on conversation transcripts elicited through GPT-4. We find the same general results as using GPT-4 is used as a predictor (Figure 3) on content recommendation and moral reasoning, only with weaker email validation results.

D MODEL-MODEL EXPERIMENTS

D.1 METHODS

We explore whether LMs can stand-in for human participants, enabling faster iteration loops and more research in this area. We generate various personas (specified in natural language) for each domain, and prompt LMs to respond to elicitation queries as their persona would.

For each domain, we construct a set of personas as follows:

Content Recommendation The personas are constructed by providing a brief biographical sketch of a hypothetical person, and were also constructed by the authors. A sample persona prompt is "*Education: Medical Doctorate. Occupation: Junior Surgeon at a regional hospital. Hobbies: Running marathons, traveling, and learning new languages.*"

Moral Reasoning We construct a variety of personas with a diverse array of moral perspectives, including Kantianism, Utilitarianism, and ethical egoism. A sample persona prompt is "*You subscribe to a Kantian code of ethics.*"

Email Validation Personas are instantiated by providing a regex to the model. The test cases are constructed by the authors. A sample persona prompt is "*You are validating that an email address adheres to a specific format (e.g. for designing a Python regex). The gold regex is . . .* `user@domain.co.co.co.co`"

We prompt as the LM as follows to answer questions according to their personas:

[Persona] Answer the question in the shortest way with minimal additional explanation.
[Question]

Furthermore, in the content recommendation domain, we implement three different selection strategies for pool-based active learning and explore their trade-offs, including random sampling (randomly selecting the next example to query), uncertainty-based sampling (selecting the example whose answer the LM is most uncertain about, i.e. the example with the highest-entropy),[5] and diversity sampling (described in Section 4.5).[6]

---

[5] Note that because GPT-4 does not return logits, we use a smaller GPT-3 `text-davinci-003` model to compute entropy over the answer distribution

[6] To avoid massive costs in uncertainty sampling, the pool was pre-filtered to a sensible size of a few hundred samples using diversity metrics. For comparability across methods, the same pre-filtered pool was used for all three sampling methods.

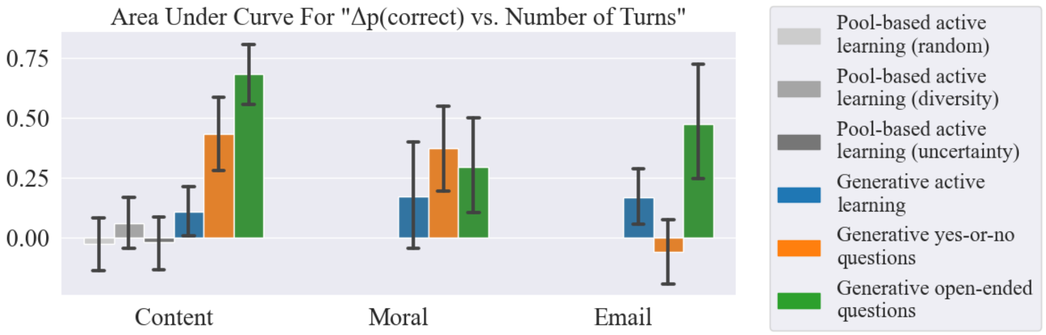


Figure 13: We plot the **Area Under the "$\Delta p(correct)$ vs. Number of Turns" Curve** for model-model experiments. This plot is analogous to Figure 3, only we are using LMs to simulate human users, and we are using number of turns as a proxy for interaction time. We see the same general trends as in Figure 3: GATE methods beat both no elicitation and pool-based active learning.

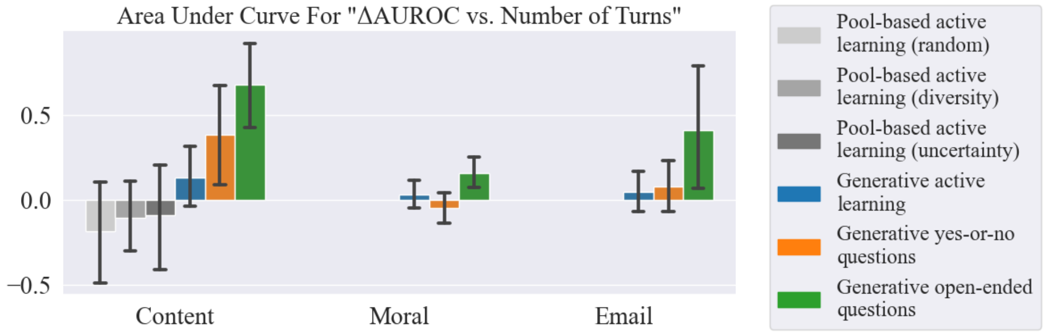


Figure 14: We plot the **Area Under the "ΔAUROC vs. Number of Turns" Curve** for model-model experiments. This plot is analogous to Figure 9, only we are using LMs to simulate human users, and we are using number of turns as a proxy for interaction time. We see the same general trends as in Figure 9: GATE methods beat both no elicitation and pool-based active learning.

D.2 RESULTS

Figures 13 and 14 shows results in each domain when we use a LM to simulate humans. Because human interaction times are unavailable for these experiments, we run interactive elicitation up to 5 turns, where we use number of turns as a proxy for human effort. Note that instead of measuring AUC of the "$\Delta p(\text{correct})$ vs. interaction time" curve, we instead measure AUC of the "$\Delta p(\text{correct})$ vs. number of turns" curve.

Can models be used to simulate human participants? In Figure 15, we plot the correlation between human experiment results and model-model experiment results for various elicitation methods. For both the human experiments and the model-model experiments, we compute the area under the "$\Delta p(\text{correct})$ vs. number of turns" curve, in addition to the average change in $p(\text{correct})$ *after 5 turns*.[7]

We find that on both metrics we evaluate, the model-model results generally correlate with human results in the content recommendation and email validation domains (methods that perform better in the model-model experiments generally also perform better in the human experiments), but not the

---

[7] Note that these metrics differ from we use to evaluate the human experiments in Section 4.5 – in particular by being turn-based instead of time-based – meaning we had to additionally compute these metrics on the human transcripts. This is necessary here because we must ensure that the model-model results and human results are measured along the same metric(s).

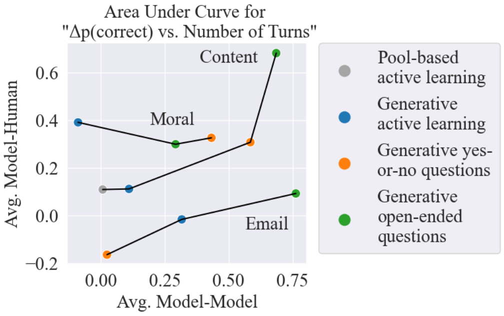


Figure 15: **Predictivity of model-model for model-human results.** We match up the *Area Under "*$\Delta p(correct)$ *vs. Number of Turns" Curve* metric for each elicitation method in each domain. We see that using the model to simulate human users is predictive of actual human results in the content and email domains, but not the moral domain.

moral reasoning domain. This could be for various reasons, including that the subtleties in human moral reasoning may be difficult to capture in a single persona prompt, and difficult to simulate even with our biggest LMs.

Which sampling strategy is the best for pool-based active learning? As seen in Figure 13, we experiment with three different pool-based active learning strategies (random, diversity-based, and uncertainty-based sampling), which perform comparably, with diversity sampling perhaps performing slightly better than the rest. This is in line with the findings from Margatina et al. (2023b). Thus, we use diversity sampling in our main human experiments.

D.3 ANALYSIS

Here, we present some additional analyses to better characterize the experiments.

How much variation there is in people's preferences? Elicitation is only helpful if there is variation in people's preferences; otherwise, a model could simply attain maximum performance by relying on its prior and ignoring the elicited information. To quantify how much variation there is in people's preferences, we compute the entropy in p(yes) for each question across participants. We find that many questions have high entropy while many others have little entropy, for an average entropy of 0.77 bits. Broadly, the results validate that our settings have significant variation in human preferences, enabling models to personalize themselves based on human preferences.

Does language model elicitation influence user preferences? Human preferences may shift when interacting with language models for a variety of reasons. For example, past work has studied *auto-induced distributional shift*, where machine learning models shift human behavior to be easier to predict (Krueger et al., 2020). To investigate whether this occurs in our experiments (or indeed if different elicitation methods induce different human preferences for any other reason), we compare the distribution of human labels on test samples from the three GATE methods with those from the user-written prompt experiments to see whether interacting with language models influences users' subsequent judgments. As seen in Figure 4 (right), we see no such effect.

What kinds of questions did the language models ask? We show a few examples of the language model questions in Figure 8. As the figure shows, these questions are complex and subtle, often building on the previous questions, representing a broad-based knowledge of the domain as well as possible nuances therein.

Why does prompting make things worse in the emails domain? In the emails domain in Figure 3, we observe that user-written preferences slightly decrease performance relative to a no-elicitation baseline. While it is possible this is an effect of noise, we also observe that some participants articulated preferences that were actually different from those they experienced when viewing email addresses. For example, one user wrote "*an email address should finish with .com or co.uk*" yet later decided that "*user@domain.edu*" was an acceptable email address. This indicates that users may not have a clear and comprehensive understanding of their own preferences, especially in more technical domains.

Can we automate evaluation? To probe whether evaluation could be automated, we conducted experiments where we *simulated* different human preferences using language models prompted with a diverse set of (automatically-generated) *personas*. These personas varied by domain, but generally contained information about a hypothetical person's preferences within that the domain. For example, in the content recommendation domain, we generated brief biographical sketches of hypothetical people, including their hobbies, interests, and careers, and conditioned GPT-4 on these biographical sketches to generate answers to queries. We found that model could simulate humans well in the content recommendation and email validation domains, but not in the moral reasoning domain. This suggests that while such personas may be a useful guide in some cases, they are not yet sophisticated enough to stand in for real human participants. See Appendix D for more details.

E HUMAN RATINGS OF USABILITY ACROSS ELICITATION POLICIES

E.1 METHODS

We ask users several questions to assess usability tradeoffs across elicitation policies. The following are the full list of questions, which we ask at different points in the experiment.

After elicitation but before seeing the test-cases:

1. How mentally demanding was interacting with the chatbot? (See discussion in Section 5)
2. To what extent did the chatbot raise issues or aspects about your preferences that you hadn't previously considered?
3. How comprehensively do you feel the chatbot's questions characterized your preferences about the task?

After seeing and labelling the test cases:

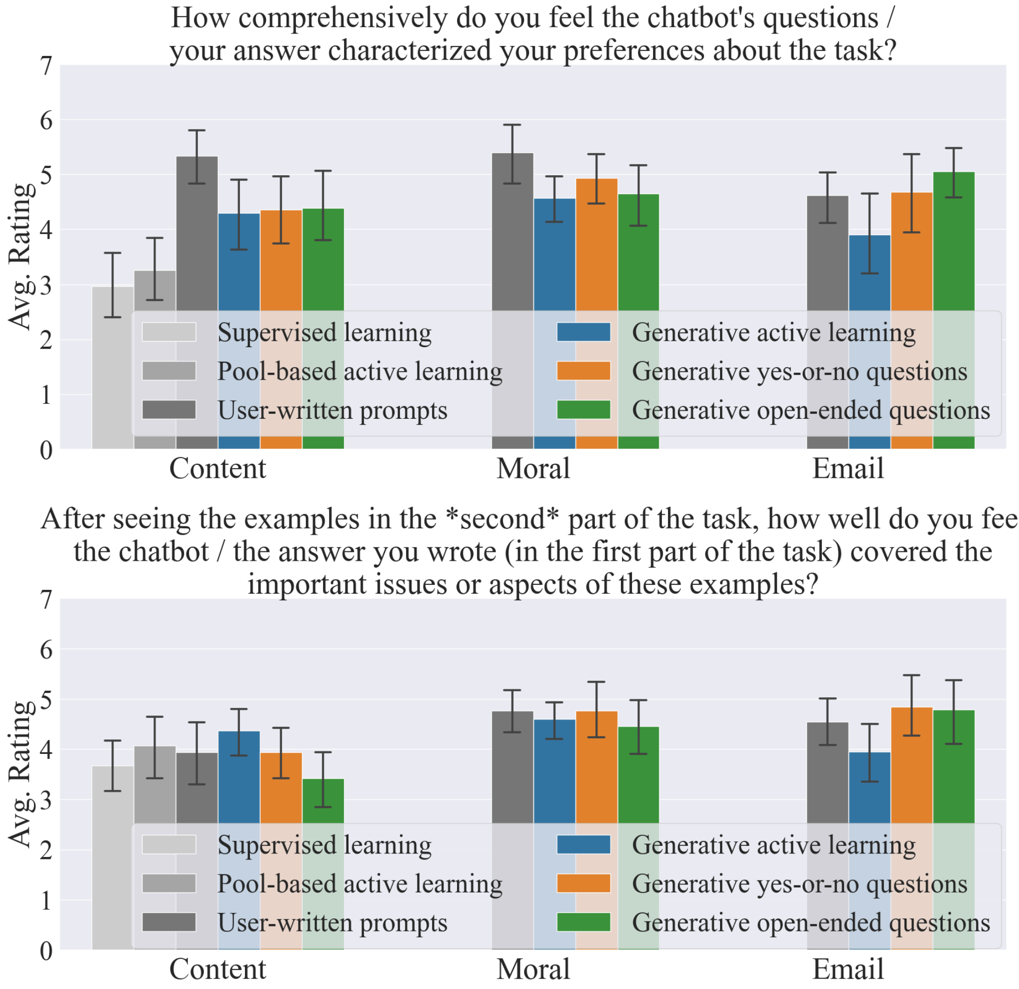
4. After seeing the examples in the second part of the task, how well do you feel the answer you wrote (in the first part of the task) covered the important issues or aspects of these examples?
5. When performing the second part of the task, to what extent did you refer back to your conversation history from the first part of the task?
6. How much experience have you had (if any) with interacting with language models (e.g. ChatGPT, GPT4, etc.)?
7. Do you have any other feedback about the task?

The last question was free response. All other questions were assessed via a Likert scale from 1 (Very Little/Poorly) to 7 (Very High/Well) with radio buttons.

E.2 RESULTS

The average ratings for the first question across each elicitation method and domain can be found in Figure 4. The average ratings for questions 2 – 5 are plotted in Figures 16 to 18.

From Fig. 16, we see that humans were on average overconfident on their ability to cover their preferences in prompts, particularly in the content recommendation and moral reasoning domains, reflected in the average rating of their perceived coverage dropping from an average of 5.3 to 3.9 (in the content recommendation domain) and an average of 5.4 to 4.8 (in the moral reasoning domain) after seeing the test cases. This indicates that humans are usually not aware of their mental limitations when writing prompts.


How comprehensively do you feel the chatbot's questions / your answer characterized your preferences about the task?

After seeing the examples in the *second* part of the task, how well do you feel the chatbot / the answer you wrote (in the first part of the task) covered the important issues or aspects of these examples?

Figure 16: Average perceived coverage of each elicitation method, before (above) and after (below) seeing the test cases. Higher indicates greater coverage.

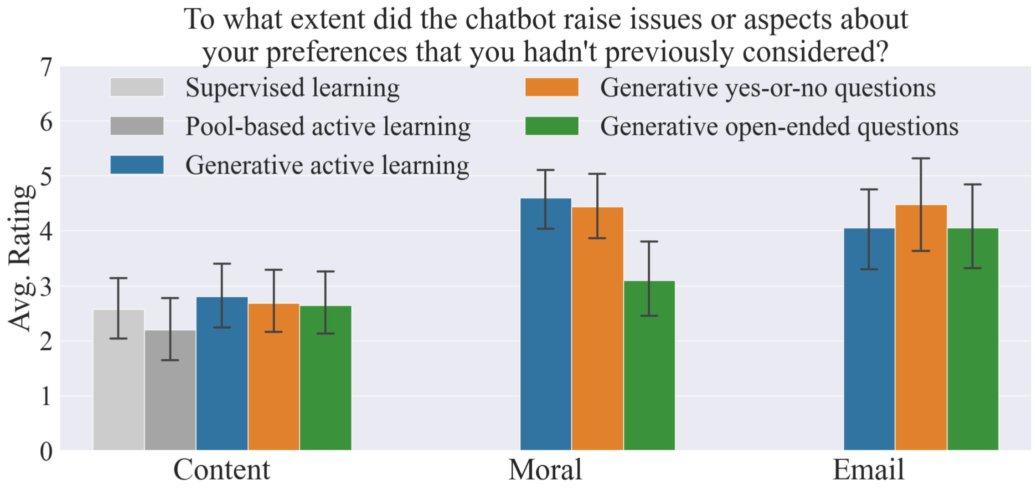


Figure 17: Extent participants perceived that each elicitation method drew out novel aspects of a domain that the user had not previously considered, averaged over each elicitation method. Higher indicates greater perceived novelty.

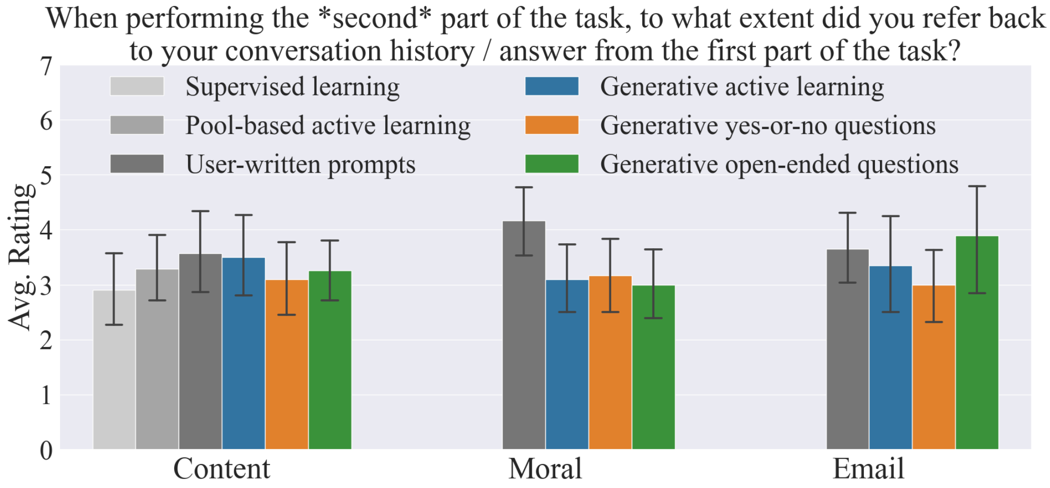


Figure 18: Extent participants referred back to the elicitation transcript when labelling test cases, averaged over each elicitation method. Higher indicates the user more heavily relied on the elicitation transcript.

From Figure 17, we see that the generative elicitation methods were on average able to surface more novel considerations in the moral reasoning and email validation domains than in the content recommendation domain, as they tend to have trickier and less intuitive edge cases.

Finally, from Figure 18, we see the extent to which users explicitly referred back to the elicitation history when making decisions on the test cases. This may influence how well-aligned the test case decisions are with the answers from the elicitation phase. When annotating test cases, we explicitly instruct participants *not* to follow the elicitation transcript if it does not align their intuition on a test sample (e.g. if the test sample surfaced a novel consideration not accounted for in the elicitation phase), though we were unable to validate how well participants followed this instruction.

F REPRODUCIBILITY

We will open-source all code used in creating GATE methods, constructing the user interface, and conducting the results and analysis. We will also release the pre-registration for our experiments. All prompts we used for querying GPT-4 (and Mixtral) in the decision-making and elicitation phases, and all instructions we presented to the user, can be found in the Appendix. In all cases, we queried GPT-4 (or Mixtral) with temperature 0 for replicability of experiments.

We also note that the model we mainly use is a closed-source model whose versions are periodically deprecated, which may hinder reproducibility. However, preliminary results with Mixtral indicate that open-source models are compatible with GATE and a promising avenue for future exploration.