Bayesian Preference Elicitation with Language Models

Anonymous ACL submission

Abstract

There is increasing interest in using language models (LMs) not only for answering queries but also for *gathering information* about the preferences of human users. This preference data can be used to fine-tune LMs via reward modeling or completing goal-oriented tasks. However, LMs have been shown to struggle with crucial aspects of preference learning: quantifying uncertainty, modeling mental states, and posing highly informative questions. These challenges have been addressed in other areas of machine learning, such as Bayesian Optimal Experimental Design (BOED), which focuses on designing informative queries within a well-defined feature space. But these methods, in turn, have historically been difficult to scale and apply to real-world problems (e.g. involving text and images), in which simply identifying the relevant features can be challenging. We introduce **OPEN** (**O**ptimal **P**reference **E**licitation with **N**atural language) a framework that uses BOED to guide the choice of informative questions and an LM to extract features and translate abstract BOED queries into natural language questions. By combining the flexibility of LMs with the precision of BOED, OPEN can optimize queries for informativity while remaining adaptable to real-world domains. Conducting user studies, OPEN outperforms existing LM- and BOED-based methods for preference elicitation.

1 Introduction

Understanding users' complex preferences and requirements is necessary for accurately and safely automating real-world tasks. Modeling preferences in a domain of interest requires both understanding *what features* of the domain are relevant to model, and *how important* these features are relative to each other (Lin et al., 2022; Lindner et al., 2022; Sadigh et al., 2017; Fürnkranz and Hüllermeier, 2012). For example, when building a content recommendation system, relevant features may be different article topics—e.g. science, politics, or celebrity culture. A user may decide whether or not to read an article based on the strength of their preferences along these different axes.

With the widespread use of preference data to align language models (LMs) with human users, there has been growing interest in using LMs themselves to elicit information about human preferences. Past work focusing on prompting LMs to ask questions about user preferences (Li et al., 2023; Lin et al., 2023; Piriyakulkij et al., 2023) has shown that LMs are capable of identifying decision-relevant features in domains spanning content recommendation, software engineering, and moral reasoning. Nonetheless, prompting offers limited control over LMs' information-gathering strategies and often fails to produce questions that are useful or informative.

By contrast, a long line of work on optimal experimental design, such as dueling bandits (Heckel et al., 2019) and Bayesian preference learning approaches (Foster et al., 2019; Gal et al., 2017; Houlsby et al., 2011), has developed methods to efficiently infer user preferences from a limited number of interactions. But these methods also have their limitations—they have primarily been applied to tasks with simple, highly-structured data where feature spaces are constrained and interactions are short. How can we get the best of both worlds? *Can we learn similarly sample-efficient models of user preferences in complex, open-ended tasks?*

In this paper, we introduce **O**ptimal **P**reference **E**licitation with **N**atural language **(OPEN)**—a framework that leverages the complementary aspects of LMs and Bayesian Optimal Experiment Design (BOED) methods (see Figure 1).

OPEN uses an LM to select the relevant features the user likely cares about (Figure 1, *steps 1, 2*), and a Bayesian model to select optimal pairwise

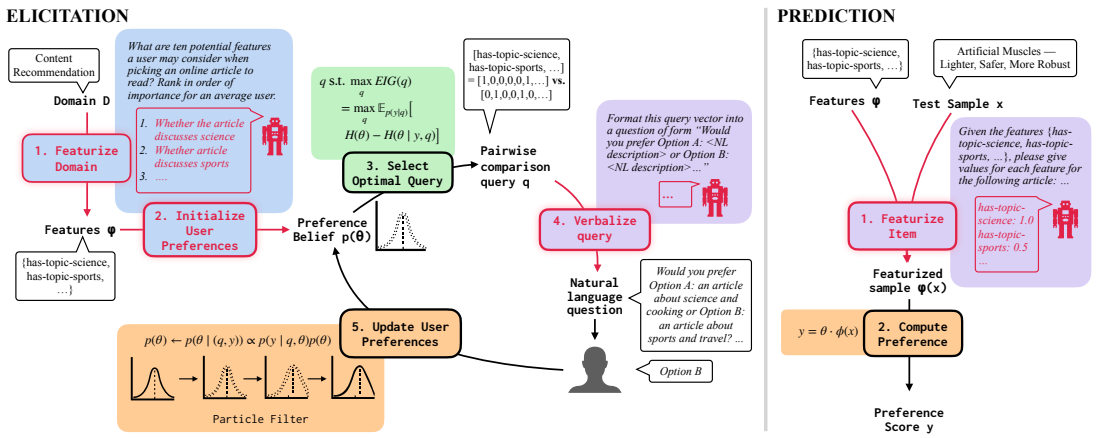

Figure 1: Overview of the OPEN framework. In red are the parts where we use a language model. During the *elicitation* stage, first, a domain D is featurized into feature ϕ with a language model, which also gives us a ranking of importance over features (which is used to initialized a prior $p(\theta)$ over user preferences). Based on the prior user preferences, the optimal pairwise comparison query q is computed, which is then verbalized using an LM into natural language. The user response is then taken to update the prior over beliefs. During the *prediction* stage, a LM is used to featurize a test sample according to the featurization ϕ derived from the elicitation stage, then a preference score is computed using the elicited preferences θ.

comparison queries (*step 3*), which are translated into natural language by a LM (*step 4*). In OPEN, we utilize a LM to provide feature coverage and a natural conversational interface with the user (*steps 1, 2, 4*), while leveraging a Bayesian model to track feature weightings and select informative questions (*steps 3, 5*). Using OPEN to elicit user preferences in a content recommendation domain, we find that it outperforms both LM- and BOED-based preference elicitation approaches.

2 Preliminaries & Background

We aim to actively model human preferences through iterative interactions with a user. Over the course of a conversation, we construct and update a model through a series of user queries, each designed such that they (1) **maximize information gain** of the users' preferences and (2) can reliably be answered by users and **elicit informative responses**.

2.1 Bayesian Optimal Experimental Design

We use Bayesian Optimal Experimental Design (BOED; Lindley, 1956, 1972; Rainforth et al., 2023) a mathematical abstraction for selecting optimal queries in an active learning setup, as the basis for our interaction framework. BOED is a common approach to preference learning and function estimation (Foster et al., 2019; Gal et al., 2017; Houlsby et al., 2011) and has been used across many different domains and disciplines (Cavagnaro et al., 2009; Dehideniya et al., 2018; Dushenko et al., 2020; Vanlier et al., 2012).

In BOED, the goal is to select an experimental design that maximizes the information gain for parameters of interest. We begin with a predictive model $p(y \mid q, \theta)$ which defines the relationship between an experiment q, an experimental outcome y, and the parameters of interest θ. The prior $p(\theta)$ describes an initial belief about the parameters.

We can formalize the information we gain about θ from each experiment as:

$$\mathrm{IG}(q, y) = H[\theta] - H[\theta \mid y, q], \tag{1}$$

where H is the Shannon entropy. However, because this notion of information gain relies on y, the outcome of the experiment, we cannot select the optimal experiment until after the outcome has been observed. To find the optimal design, we instead take the expectation of the information gain for the marginal distribution of y over all possible outcomes $p(y|q) = \mathbb{E}_{p(\theta)}[p(y|\theta, q)]$,[1] yielding:

$$\mathrm{EIG}(q) = \mathbb{E}_{p(y|q)}[\mathrm{IG}(q, y)]. \tag{2}$$

Thus, the optimal experiment can be defined as:

$$q^* = \mathrm{argmax}_q(\mathrm{EIG}(q)). \tag{3}$$

[1]This is equivalent to the mutual information between θ and y given q.

In our preference learning setting, we want to pick the pairwise comparison question (q) that will allow us to learn about the user's preferences (θ) as efficiently as possible. At each interaction, y is the user's response to q. BOED is uniquely powerful in our sequential setting–enabling us to, after each interaction, incorporate users' answers ad-hoc to greedily update our belief of the their preferences.

2.2 Modeling Human Preferences

Following previous preference learning literature (Lin et al., 2022; Lindner et al., 2022; Sadigh et al., 2017; Candeal-Haro and Induráin-Eraso, 1995), we model human preferences about examples x as linear in the features of x. We notate users' preferences as θ, the feature function as mapping from features to items as ϕ, and model pairwise judgements as the Bradley-Terry model (Bradley and Terry, 1952).

3 The OPEN Framework

Existing Bayesian and bandit-based preference learning methods assume that ϕ is known *a priori*, and that feature vectors can be straightforwardly compared (e.g. by comparing them directly, or with a templated procedure for turning features $\phi(x)$ into examples x). How should we apply these methods in domains like content recommendation where features themselves are complicated to extract and even more complicated to translate back into concrete instances? To operationalize BOED and other uncertainty-based preference learning approaches, we introduce OPEN: a domain-agnostic preference learning framework that incorporates LMs with BOED. OPEN makes it possible to extend BOED to complex domains—using LMs to identify environment-specific features, construct real-world examples from abstract feature representations, and interface with the user in—while maintaining BOED's principled approach to question-selection and decision-making.

At a high level, OPEN learns user preferences by:

1. **Featurization**: Using a description of the domain, D, the LM extracts a set of NL features which we use to define the feature function ϕ.

2. **Initializing User Preferences**: We prompt the LM to rank the NL features and set a prior over user preferences, $p(\theta)$, according to this ranking.

3. **Selecting an Optimal Question**: The Bayesian model samples all possible pairwise comparison questions and selects the maximally informative question, using Equation (3).

4. **Querying the User**: The LM translates q^* into a NL question for the user (this can also be thought of as ϕ^{-1}).

5. **Posterior Update**: Given the user's response to the question, we compute the Bayesian posterior $p(\theta \mid y, q)$.

6. **Prediction**: The Bayesian model predicts the user's response to each test case, using the Bradley–Terry model of human preferences.

Even for the linear preference model described in Section 2.2, computation of the posterior $p(\theta \mid y, q)$ (and thus computation of EIG(q)) is intractable. We implement a tractable approximation using a Bayesian Particle Filter (Gordon et al., 1993; Doucet and Johansen, 2008; Elfring et al., 2021). As each particle can be thought of as a plausible instantiation of user preferences θ, we refer to individual particles as **personas** $\mathbf{p}_i$.

Crucially, though, the OPEN does not necessitate the use of particle filtering or even BOED. Our framework fundamentally provides a domain-agnostic approach to active and principled preference learning: enabling any uncertainty-based preference-learning algorithm (multi-armed bandits, Bayesian active learning, etc.) to interface with a user in NL.

Below, we detail our implementation of the aforementioned components of OPEN.

3.1 Featurization

We prompt the LM with a general description of the domain of interest (e.g. content recommendation of news articles), and ask it to produce natural language descriptions of pertinent features in that domain, $\mathcal{F}$. For example, in the content recommendation domain, the LM outputs "*whether the article discusses science*" as a pertinent feature. This list of natural language features can then be converted to a function ϕ that transforms test samples to feature values by prompting another LM with $\mathcal{F}$ and a test sample x (see Section 3.6).

3.2 Initializing User Preferences

When prompting the LM for features in Section 3.1, we simultaneously ask it to *rank* features from *most*

to *least* important. These rankings are then used to initialize the prior for OPEN's belief of the user's preference, $p(\theta)$. Because we assume that θ is linear in the feature space, we model $p(\theta)$ as a Bayesian linear model with each feature weight parameterized by a normal distribution:

$$p(\theta_i) \sim \mathcal{N}(0, \sigma^2) \cdot w_i, \quad w_i = 1.2 - 0.12i \tag{4}$$

where σ^2 is the base variance for each feature before scaling, and w_i captures the relative importance of features from highest to lowest.

We construct our initial sample of N personas by sampling each element of each persona independently from $p(\theta_i)$.

3.3 Selecting the Optimal Question

To efficiently learn user preferences, we aim to select the pair of options that maximize the EIG regarding the user's preferences at each interaction step. This process is grounded in the BOED framework, as described earlier. Given a set of features, $\mathcal{F}$, we define the space of all possible pairwise comparison questions as:

$$\mathcal{Q} = \{(\mathbf{o}_a, \mathbf{o}_b) | \mathbf{o}_a, \mathbf{o}_b \in \{0,1\}^{|\mathcal{F}|}, \sum_{i=1}^{|\mathcal{F}|}(\mathbf{o}_a)_i = \sum_{j=1}^{|\mathcal{F}|}(\mathbf{o}_b)_j = K\},$$

where each option $\mathbf{o}_a$ and $\mathbf{o}_b$ is represented as a binary vector indicating the presence (1) or absence (0) of each feature in the comparison. In our experiments, we set $|\mathcal{F}| = 10$ and $K = 2$.[2] The goal is to select the question $(\mathbf{o}_a^*, \mathbf{o}_b^*) \in \mathcal{Q}$ that maximizes the EIG about the user's preferences θ.

Recalling Equation 3, the EIG for the optimal pairwise comparison $(\mathbf{o}_a^*, \mathbf{o}_b^*)$ with which to query the user is then given by:

$$(\mathbf{o}_a^*, \mathbf{o}_b^*) = \text{argmax}_{(\mathbf{o}_a, \mathbf{o}_b) \in \mathcal{Q}} \text{EIG}(\mathbf{o}_a, \mathbf{o}_b)$$

3.4 Querying the User: Mapping Pairwise Comparisons to NL

Given the optimal pairwise comparison, we prompt a LM to convert the feature vectors into a NL question for the user. This is equivalent to taken the inverse of the featurization function ϕ^{-1}. We include the set of NL features from the LM (Section 3.1) and the K present features. We prompt the LM with the feature set and pairwise comparison to develop a real-world example for each option ad-hoc. For example, in our content recommendation setting, this is an example article summary. We display these examples in the user interface alongside each option in the pairwise comparison question.

[2] From preliminary testing, we found that including ten total features ($|\mathcal{F}| = 10$) and two features ($K = 2$) in each option, best balanced users' mental effort with the informativity of the question.

3.5 Posterior Update

At each interaction step t, having observed the user's selection y_t between $(\mathbf{o}_a, \mathbf{o}_b)$, we update our belief about the user's preferences (θ) based on their response (y_t) to the pairwise comparison question (q_t) by computing the posterior:

$$p(\theta \mid y_t, q_t, \mathcal{H}) = \frac{p(y_t, q_t \mid \theta) \cdot p(\theta \mid \mathcal{H})}{p(y_t, q_t, \mathcal{H})}, \tag{5}$$

where $\mathcal{H} = \{(q, y)\}_{0 \cdots t-1}$ denotes the conversation history (prior sequence of questions and answers). We approximate the update in Equation (5) using our particle filter. At each step t, we reweight each particle $\mathbf{p}_i$ by

$$w_i = p(y_t | \mathbf{o}_a, \mathbf{o}_b, \mathbf{p}_i^\top) = \sigma(\mathbf{p}_i^\top \phi(\mathbf{o}_a) - \mathbf{p}_i^\top \phi(\mathbf{o}_b)). \tag{6}$$

Then, we fit a Gaussian to the distribution of $w_i \mathbf{p}_i$s. Finally, we resample new personas from this distribution. For a detailed overview of our algorithm, see Appendix B.

3.6 Prediction

After each user interaction, we evaluate our model via a set of test cases, which are also presented to the user at the end of the study. For our test cases, we use pairwise comparison questions (x_a, x_b), where each x_a, x_b are taken from real-world examples from the domain (e.g. for content recommendation, the headline and lede of a New York Times news article). For each test-case question, we select $y \in [x_a, x_b]$ according to:

$$\max_y \mathbb{E}_\theta \left[p(y \mid \theta, (x_a, x_b)) \right]$$

(where the expectation is taken over our distribution of $p(\theta)$ at that point in time). Following the Bradley-Terry model, we define preferences as

$$p(x_a \mid \theta, (x_a, x_b)) = \sigma(\theta \cdot \phi(x_a) - \theta \cdot \phi(x_b))$$
$$p(x_b \mid \theta, (x_a, x_b)) = 1 - p(x_a \mid \theta, (x_a, x_b))$$

To compute $\phi(x_a)$ and $\phi(x_b)$ (recall we were given a set of natural language descriptions of features

$\mathcal{F}$), we prompt a LM to give us concrete values for each features, based on the natural language feature descriptions and the test-set sample.

4 Experimental Setup

We use OPEN to investigate how incorporating LMs alongside explicit information-theoretic computations (via BOED) can improve existing preference-learning methodology and provide a novel NL interface with which to approach personalized machine learning.

We outline OPEN's experimental details, the three different baselines which we compare to, the evaluation framework, and the human participants' interface.

4.1 Hyperparameters

In the featurization step, we query the LM for $|\mathcal{F}| = 10$ features to define the domain. When sampling pairwise comparison questions, each option contains $K = 2$ features. We use GPT-4 as the LM for all of our experiments.

4.2 Baselines

We consider three baselines:

LM-only Open-Ended Questions Following Li et al. (2023)'s best-performing method in their content-recommendation setting, we prompt a LM to ask informative, open-ended questions, without any explicit optimization. The LM is provided the complete conversation history of the interaction before asking each subsequent question. The user is able to answer the question free-form.

LM-only Pairwise Comparison Questions We prompt a LM to ask an informative, pairwise comparison question, without any explicit optimization. The LM is provided the complete conversation history of the interaction and the feature set from Section 3.1 before asking each subsequent question. To answer the question, the user selects either 'Option A' or 'Option B' based on their preference.

User Self-Mapping Pairwise Comparisons to NL We provide the user the feature vectors of the optimal pairwise comparison question as determined from Section 3.3 as well as the NL descriptions of each feature. The user must use these descriptions to interpret the vector comparison query, then select either 'Option A' or 'Option B' based on their preference.

4.3 Human Experiment Details

We recruit native-English-speaking human participants from Prolific (Palan and Schitter, 2017). We recruited ∼40 people for each setting.[3] Mirroring Li et al. (2023), we conducted our prolific experiments in two steps: *(1) elicitation*: participants were asked to answer questions about their preferences for 5 minutes; *(2) prediction*: participants were then presented with the 15 pairwise comparison questions and asked which of the two options they preferred. We recruited new participants for each test across all experiments.

4.4 Domain: Content Recommendation

We consider the task of content recommendation– recommending news articles for a user to read. Each approach is evaluated based on its ability to predict which news articles the user would prefer on a set of 15 pairwise comparisons. Each comparison contains the lede and headline from New York Times articles, hand-collected by the authors. We select this task (1) because people's preferences have a large amount of variance and (2) to perform a direct comparison to Li et al. (2023)'s methodology. Further information about the articles is available in Appendix C.

4.5 Evaluation

We evaluate OPEN and the above baselines after each user interaction on a test set of 15 pairwise comparison questions, each comparing two different real-world news articles.

Methods We evaluate using two different methods: in **OPEN prediction**, we use the OPEN prediction method described in Section 3.6 to answer each test set question. In **LM prediction**, we prompt a LM to answer the test question, conditioned on the conversation history (up to some turn), similar to the evaluation approach taken by Li et al. (2023).

Metrics After each conversation turn, we evaluate the test-set accuracy at that point in time, using one of the two methods above. After the conversation has ended, we calculate the **Time-Integrated Delta Accuracy (TIDA)**, or the integral of the "Δaccuracy over time curve", where Δaccuracy is defined as the difference between test-set accuracy

[3] Additional details regarding the user studies are available in Appendix D

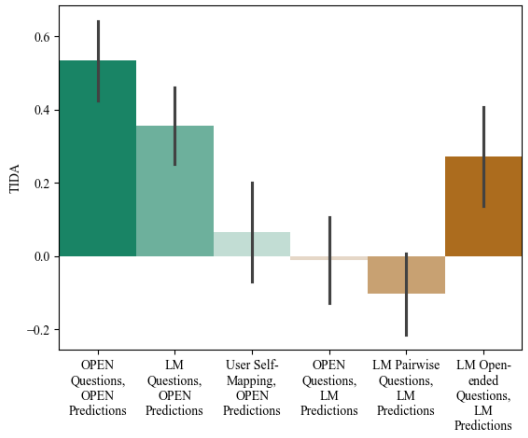

Figure 2: **Time-Integrated Delta Accuracy (TIDA) for each method**. We report the integral of the delta accuracy over time. OPEN improves over naive prompting-based approaches, all while improving transparency and reducing computational cost. Error bars are one standard error.

at the *current* interaction step and the initial interaction step. This metric rewards us for eliciting the user preferences quickly–even if two elicitation transcripts result in same Δ accuracy, the one that arrives at a higher Δ accuracy earlier in time is rewarded.

5 Results

We present the TIDA scores for OPEN against five different combinations of question generation and prediction in Figure 2 (refer to Sections 3 and 4.2 for question generation methods and Section 4.5 for prediction methods). Below, we discuss the main takeaways from our results.

OPEN is better at *making predictions aligned with* human preferences when compared to LM-only approaches. Comparing the OPEN *Questions,* OPEN *Predictions* bar to OPEN *Questions, LM Predictions* bar, we find that OPEN can make predictions that better align with human preferences than a prompted LM, from the same sequence of questions. This implies that LMs are still subpar at *in-context-learning* of human preferences from demonstrations, compared to Bayesian methods which explicitly keep track of the human's preference function; it underscores the importance of explicitly monitoring human preferences.

OPEN is better at *eliciting* human preferences compared to LM-only approaches. Comparing the OPEN *Questions,* OPEN *Predictions* bar to the comparable *LM Pairwise Questions,* OPEN *Predictions* bar, we find that OPEN *Pairwise Questions* outperforms *LM-only Pairwise Questions* for elicitation, indicating that when asking questions, there is value in tracking human preferences and querying based on explicit notions of uncertainty.

Of course, one benefit of LM elicitation approaches is that LMs are not constrained to asking only a certain type of question. Thus, as a topline, we elicit open-ended questions from LMs (*LM Open-ended Questions, LM Predictions*). Though open-ended questions far outperform pairwise questions (both ones selected by OPEN and selected by the LM) when evaluated with a LM, this improvement is overshadowed by the benefit of being able to use OPEN *prediction*, which is incompatible with LM-only elicitation methods. However, this suggests that future lines of work could explore eliciting open-ended questions compatible with OPEN predictions.

OPEN is better at eliciting human preferences compared to BOED-only approaches. When we remove the verbalization step (*User Self-mapping,* OPEN *Predictions)*, we find that users become significantly worse at accurately specifying their preferences. Thus, having a natural interface for interacting with the user, which we accomplish by verbalizing questions with a LM, is essential for being able to accurately and efficiently elicit user preferences.

6 Analysis

The Importance of Feature Weightings To understand the importance of *feature weightings* as opposed to absolute rankings of features, we asked users to order the set of NL features by importance. We tested if a linear or exponential weighting of their self-reported rankings were able to accurately predict their preferences on the test cases. We compared the user's absolute ranking to (1) the absolute rankings of OPEN by averaging OPEN's rankings across all personas at the last interaction step and (2) the performance of OPEN at the last interaction step with the learned feature weights.

Our findings, in Figure 4, indicate that the relative weighting of features is critical to OPEN's prediction, and that user's self-reported rankings are not a good indicator of their actual preferences. This suggests that methods that rely solely on human-written specifications, such as prompts,

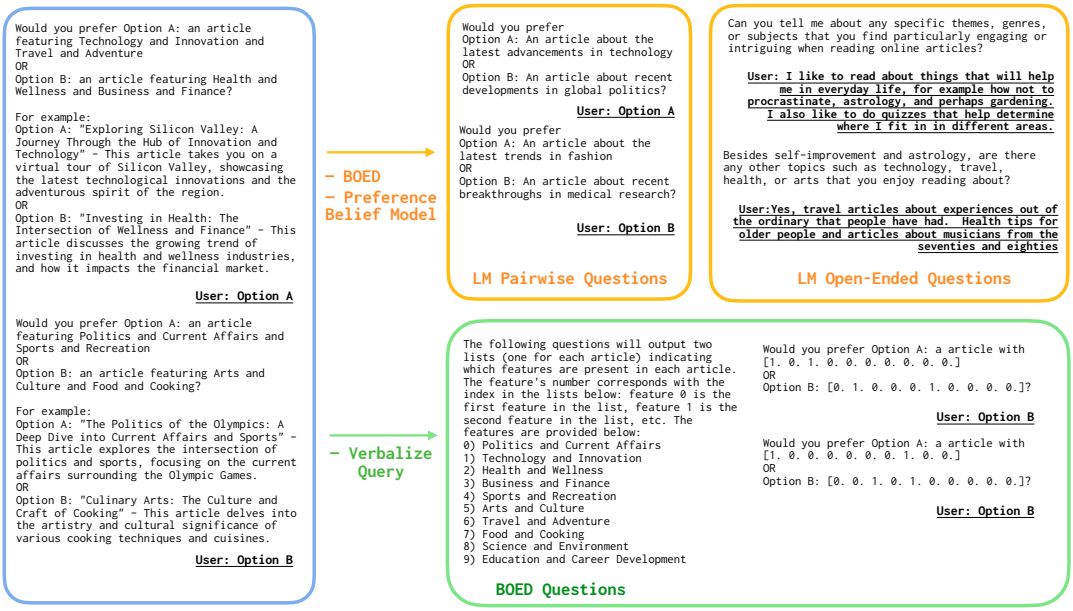

Figure 3: **Sample transcripts from OPEN vs. Baselines**

may not be as useful as methods that are able to keep track of the precise relative weights between features, as feature weights are typically much harder for users to specify than feature rankings.

Qualitative Analysis At the end of the study, we collected open-ended feedback from the users regarding their experience interacting with the system. Across the different elicitation methods, we found:

1. **LM's open-ended questions to be repetitive and overreliant on the LM's prior over user preferences**. For example one user noted: *"the chat bot did not seem to pick up on many parts of my input, as though it had its own 'agenda'."* Another user observed that the model repeated questions: *"the chatbot continued to ask the same exact question in the same way, it should have different variations and questions to dive deeper rather than repeating itself."* The transcripts from these users corroborate their observations.

2. **LM's pairwise questions (without an explicit feature set) failed to probe for *feature weighting***. A user commented that: *"I'm not sure what the purpose here is, but if it was to pinpoint my interests than they should have been comparing my previous choices together as well!"*

3. **Users struggled to self-map features to NL questions**. One user stated: *"I was thoroughly confused, however I think I figured it out and clicked accordingly."*

We also collected empirical feedback on participants' perceived effort across the different methods. We found that they reported consistent levels of mental demand across all methods (including the *LM Open-ended Questions*), indicating that OPEN questions were not more challenging for users than other settings. Further analysis and visualization of users' empirical feedback is included in Appendix D. Transcripts from each our settings can be found in Figure 3.

7 Related Work

Task Ambiguity and Underspecification Prior research has studied ambiguity in the context of language-guided tasks (Lake et al., 2019; Tamkin et al., 2023). In particular, in the context of prompt engineering (Brown et al., 2020), writing complete and unambiguous task specifications can be difficult (Li et al., 2023; 278, 1984). Futhermore, recent developments have encouraged fine-tuning approaches such as reinforcement learning with human feedback (RLHF) (Ziegler et al., 2019; Christiano et al., 2017) and direct preference optimization (DPO) (Rafailov et al., 2023) to better align LLM behaviors with humans' (underspeci-

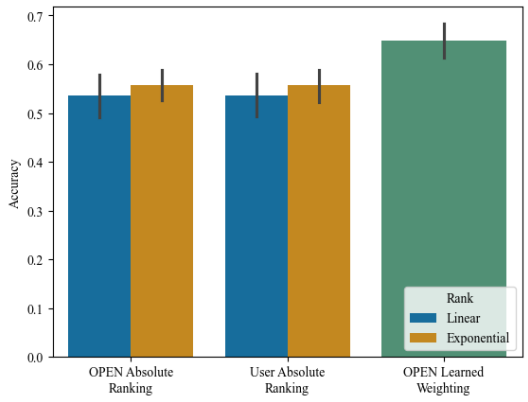

Figure 4: Accuracy for OPEN's absolute feature rankings, users' self reported absolute feature rankings, and OPEN's learned weights over features as described in Section 6. Our analysis indicates removing access to precise weights significantly hurts performance. Error bars are one standard error.

fied) preferences. These techniques, however, all assume access to existing human preference data. Our goal in this work is to examine how to elicit such data efficiently.

Classical Preference Learning Techniques Learning human preferences is a rich area of work that has spanned many fields over past few decades. Much work has gone into *optimally* querying for user preferences. Aside from Bayesian Optimal Experimental Design, which we use in this paper, approaches have included conjoint analysis (STERN, 1990; Arora and Huber, 2001; Kuhfeld et al., 1994), polyhedral methods (Toubia et al., 2007), multi-armed bandits (Lu et al., 2010), and dueling bandits (Heckel et al., 2019), etc. (Vayanos et al., 2021)

Active learning is an area of machine learning focused on interactively querying an expert to label new data points, often selected from a pool (Settles, 2010). Uncertainty-based active learning computes the next example to show the user based on explicit notions of uncertainty (Lewis and Catlett, 1994).

However, these approaches are often only applicable to simple domains with known feature spaces (or existing pools of examples), and may not be straightforward to derive for complex domains.

LM Preference Learning More recently, Li et al. (2023); Piriyakulkij et al. (2023) introduced approaches to active elicit user preferences with language models. They demonstrated LMs are able to actively elicit user preferences across a wide variety of domains beyond certain classical techniques like active learning. However, our work demonstrates that LMs are still subpar at tracking and using feature weightings to ask informative questions.

8 Discussion & Future Work

We presented OPEN, a domain-agnostic framework for combining Bayesian Optimal Experimental Design with LMs–using BOED to guide the choice of informative questions, and the LM to extract environment-relevant features and translate abstract feature queries into real NL questions. Beyond the improvements demonstrated by our findings, our system has two additional advantages over LM-only methods: (1) *we can improve the transparency of LMs* through extracting NL features. In our framework the NL features from the LM accompanied by an external uncertainity-driven model conditioned on the features, provide greater transparency than an LM. With OPEN, we can understand how important each feature is to the elicitation and prediction process. (2) *We can improve privacy while reducing computational costs*. By abstracting prediction to an external, linear system, we can model user behavior without the need for LM intervention. In OPEN, the Bayesian Particle Filter models each user's behavior separately while still incorporating the high-level, contextual information encoded in LMs. Furthermore, at test time, conducting inference on this linear system is orders of magnitude cheaper and yields better results than an LM.

OPEN's flexibility allows for the exploration of other, more intensive preference-learning approaches such as variational Bayesian methods (e.g. Foster et al., 2019) or multi-armed bandits (e.g. Lindner et al., 2022). We are also excited by the potential to further incorporate aspects of real-world learning environments. For example, future work could investigate using OPEN to develop an adaptive feature space–as the model's uncertainty increases, the model could query for and optimize new features that might better explain patterns within the data. Additionally, recent advancements in fine-tuning models from human feedback (Ouyang et al., 2022; Rafailov et al., 2023; Ethayarajh et al., 2024) found significant differences in data curation and model performance when changing the experimental design. OPEN enables future research to explore the impact of this parameter in an active preference elicitation environment.

Limitations

Although we provide the steps to build towards better active uncertainty-driven preference learning methods, our work has several limitations. In our study, OPEN was constrained to pairwise queries in a content recommendation setting. Also, we operated under a fixed feature space and with a closed-source LM. Future work could explore expanding along each of these axes: investigating other preference-learning domains, incorporating open-ended questions, expanding the feature space in conversation, and testing other LMs. Futhermore, future research should explore the impact of OPEN across additional real-world settings and population groups.

Ethical Considerations

Our work presents both ethical benefits and risks. Understanding user preferences in underspecified environments is crucial to avoiding real-world issues with AI systems such as spreading hate speech, generating illicit or copyrighted content, as well as perpetuating bias and stereotypes. Furthermore, as more systems are deployed without expert-supervision, ensuring that they can align themselves with users' values is crucial to ensuring safe and healthy interactions. However, in aligning with user preferences, it is also possible that systems may align with unwanted or dangerous tendencies. Developing guidelines for models' value-systems is crucial to ensuring the long-term success of human-AI interactions. Furthermore, working with preference data inherently presents risks as it can be used maliciously to manipulate individuals. It is necessary to ensure that preference data, when used by AI systems, is collected robustly and with supervision.

User Study We used the Prolific platform (Palan and Schitter, 2017) to conduct all of our human subject experiments. We paid workers in line with Prolific guidelines and solicited initial feedback via pilot studies to make the interactive experience more enjoyable for participants. Anecdotally, many participants expressed enjoying the study:

1. *"This was a very interesting exercise. Thank you for the opportunity."*
2. *"it was really interesting. good job."*
3. *"The task was straightforward and I liked that there were diverse topics to choose from. It felt like I had the chance to express my preferences on various subjects"*

Reproducibility Statement We provide thorough experimental details including the user interface and LM prompts in the Appendix. We will also release our codebase and anonymized user data on GitHub.

References

1984. Ieee guide for software requirements specifications. *IEEE Std 830-1984*, pages 1–26.

Neeraj Arora and Joel Huber. 2001. Improving Parameter Estimates and Model Prediction by Aggregate Customization in Choice Experiments. *Journal of Consumer Research*, 28(2):273–283. _eprint: https://academic.oup.com/jcr/article-pdf/28/2/273/17927222/28-2-273.pdf.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Juan Carlos Candeal-Haro and Esteban Induráin-Eraso. 1995. A note on linear utility. *Economic Theory*, 6(3):519–522.

Daniel Cavagnaro, Jay Myung, Mark Pitt, and Janne Kujala. 2009. Adaptive design optimization: A mutual information-based approach to model discrimination in cognitive science. *Neural computation*, 22:887–905.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Mahasen B. Dehideniya, Christopher C. Drovandi, and James M. McGree. 2018. Optimal bayesian design for discriminating between models with intractable likelihoods in epidemiology. *Computational Statistics Data Analysis*, 124:277–297.

A. Doucet and A. M. Johansen. 2008. A tutorial on particle filtering and smoothing: Fifteen years later.

Sergey Dushenko, Kapildeb Ambal, and Robert D. McMichael. 2020. Sequential bayesian experiment design for optically detected magnetic resonance of nitrogen-vacancy centers. *Phys. Rev. Appl.*, 14:054036.

Jos Elfring, Elena Torta, and René van de Molengraft. 2021. Particle filters: A hands-on tutorial. *Sensors*, 21(2).

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization.

Adam Foster, Martin Jankowiak, Eli Bingham, Paul Horsfall, Yee Whye Teh, Tom Rainforth, and Noah Goodman. 2019. Variational bayesian optimal experimental design.

Johannes Fürnkranz and Eyke Hüllermeier. 2012. *Preference Learning*, pages 2669–2672. Springer US, Boston, MA.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data.

N.J. Gordon, D.J. Salmond, and A.F.M. Smith. 1993. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proc. F Radar Signal Process. UK*, 140(2):107.

Reinhard Heckel, Nihar B. Shah, Kannan Ramchandran, and Martin J. Wainwright. 2019. Active ranking from pairwise comparisons and when parametric assumptions do not help. *The Annals of Statistics*, 47(6):3099 – 3126.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning.

Warren F. Kuhfeld, Randall D. Tobias, and Mark Garratt. 1994. Efficient experimental design with marketing research applications. *Journal of Marketing Research*, 31(4):545–557.

Brenden M. Lake, Tal Linzen, and Marco Baroni. 2019. Human few-shot learning of compositional instructions. In *Annual Meeting of the Cognitive Science Society*.

David D. Lewis and Jason Catlett. 1994. Heterogeneous Uncertainty Sampling for Supervised Learning. In William W. Cohen and Haym Hirsh, editors, *Machine Learning Proceedings 1994*, pages 148–156. Morgan Kaufmann, San Francisco (CA).

Belinda Z. Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. 2023. Eliciting human preferences with language models.

Jessy Lin, Daniel Fried, Dan Klein, and Anca Dragan. 2022. Inferring rewards from language in context.

Jessy Lin, Nicholas Tomlin, Jacob Andreas, and Jason Eisner. 2023. Decision-oriented dialogue for human-ai collaboration. *arXiv preprint arXiv:2305.20076*.

D. V. Lindley. 1956. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005.

D. V. Lindley. 1972. *Bayesian Statistics*. Society for Industrial and Applied Mathematics.

David Lindner, Sebastian Tschiatschek, Katja Hofmann, and Andreas Krause. 2022. Interactively learning preference constraints in linear bandits.

Tyler Lu, David Pal, and Martin Pal. 2010. Contextual multi-armed bandits. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 485–492, Chia Laguna Resort, Sardinia, Italy. PMLR.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Stefan Palan and Christian Schitter. 2017. Prolific.ac—a subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, 17:22–27.

Top Piriyakulkij, Volodymyr Kuleshov, and Kevin Ellis. 2023. Active preference inference using language models and probabilistic reasoning.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Tom Rainforth, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. 2023. Modern bayesian experimental design.

Dorsa Sadigh, Anca D. Dragan, S. Shankar Sastry, and Sanjit A. Seshia. 2017. Active preference-based learning of reward functions. In *Robotics: Science and Systems*.

Burr Settles. 2010. Active learning literature survey.

HAL STERN. 1990. A continuum of paired comparisons models. *Biometrika*, 77(2):265–273. _eprint: https://academic.oup.com/biomet/article-pdf/77/2/265/5618454/77-2-265.pdf.

Alex Tamkin, Kunal Handa, Avash Shrestha, and Noah Goodman. 2023. Task ambiguity in humans and language models. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Olivier Toubia, John Hauser, and Rosanna Garcia. 2007. Probabilistic polyhedral methods for adaptive choice-based conjoint analysis: Theory and application. *Marketing Science*, 26(5):596–610.

Joep Vanlier, Christian Tiemann, Peter Hilbers, and Natal van Riel. 2012. A bayesian approach to targeted experiment design. *Bioinformatics (Oxford, England)*, 28:1136–42.

Phebe Vayanos, Yingxiao Ye, Duncan McElfresh, John Dickerson, and Eric Rice. 2021. Robust active preference elicitation.

Daniel M. Ziegler, Nisan Stiennon, Jeff Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *ArXiv*, abs/1909.08593.

A Appendix

In the Appendix below, we provide additional details regarding our methodology, user studies, and LM prompts.

B Particle Filter Details

The detailed implementation of our particle filter algorithm is below:

For each persona $\mathbf{p}_i$ at step t:

1. We assume the Bradley-Terry model of human preferences (Bradley and Terry, 1952), where the probability of preferring Option A over Option B is directly proportional to the exponential of the difference in outcomes between Option A and Option B. For the sampled state of each persona, we **calculate the utility of each option**:

$$U(\mathbf{o}_{\{a,b\}}, \mathbf{p}_i) = \mathbf{p}_i \cdot \mathbf{o}_{\{a,b\}},$$

and **compute the preference probability** which indicates the probability that the persona prefers $\mathbf{o}_a$ over $\mathbf{o}_b$:

$$p(y_t|\mathbf{o}_a, \mathbf{o}_b, \mathbf{p}_t^i) = \sigma(U(\mathbf{o}_a, \mathbf{p}_i) - U(\mathbf{o}_b, \mathbf{p}_i)).$$

2. The **weight of each persona is updated** based on the observed preference, using the preference probability as a measure of how well the persona's preferences align with the user's selection:

3. **Compute the updated weight** based on the observed preference and expected utility:

$$\hat{w}_t^i = p(y_t|\mathbf{o}_a, \mathbf{o}_b, \mathbf{p}_t^i),$$

fit a Gaussian to the distribution, and **normalize the weights**:

$$w_t^i = \frac{\hat{w}_t^i}{\sum_{i=1}^N \hat{w}_t^i}$$

4. Finally, we **resample personas** via systematic resampling, selecting personas with a probability proportional to their updated weight. We reset the weights after resampling:

$$w_t^i = \frac{1}{N}$$

C Content Recommendation Evaluation Details

For evaluation in our content recommendation domain, we collected 30 articles–the article titles and lede–from the New York Times: (https://www.nytimes.com/). Articles topics ranged from "What to Cook This Week" to "'Barbie' vs. 'Oppenheimer'" to "Fighting to Govern Myanmar." Articles were randomly paired to create a persistent set of 15 pairwise comparison questions which were used throughout all of our evaluations.

D User Study

D.1 Prolific Survey

Screenshots of the Prolific Survey from the participants' viewpoint Prolific survey across different stages are provided in the following pages in Figures 5-9. We recruited English-speaking participants located in the United States. For each elicitation and prediction setting reported in 2, we recruited roughly 40 users. Specifically for each setting we had:

1. *OPEN Questions, OPEN Predictions*: 44 users
2. *LM Questions, OPEN Predictions*: 48 users
3. *User Self-Mapping, OPEN Predictions*: 39 users
4. *OPEN Questions, LM Predictions*: 44 users
5. *LM Pairwise Questions, LM Predictions:* 41 users
6. *LM Open-ended Questions, LM Predictions*: 39 users

E Analysis: User Feedback

Figure 10 visualizes the user feedback across multiple different axis. Each of the axis and their corresponding query for the user is below:

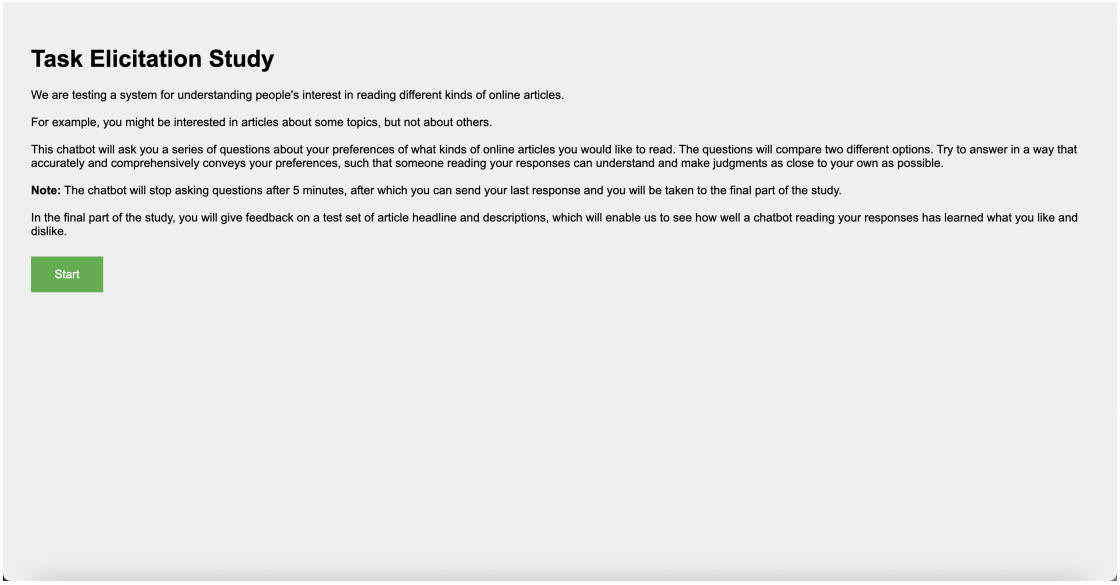

Figure 5: **Start screen**

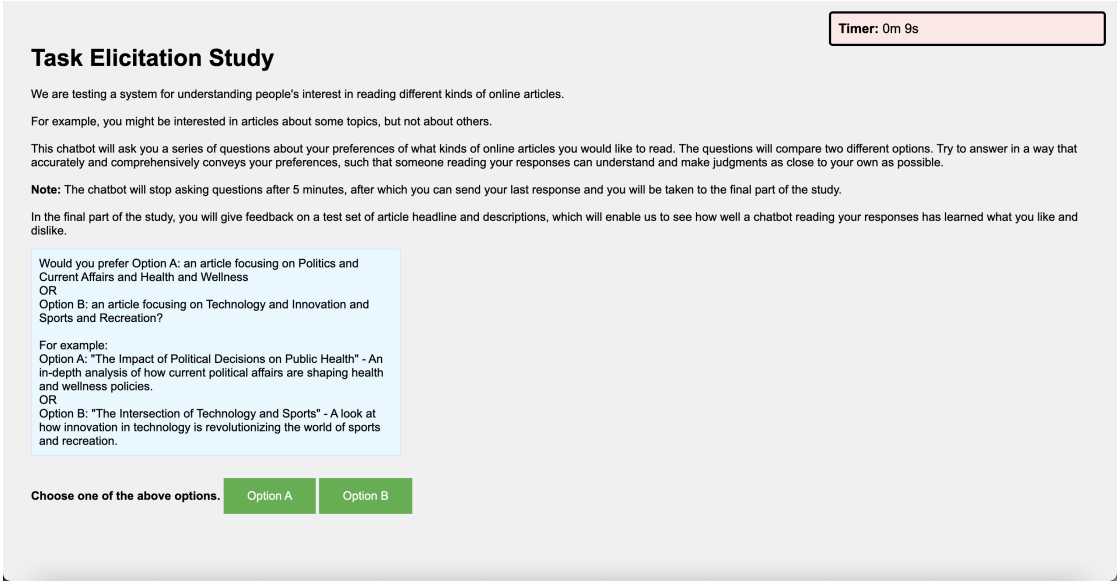

Figure 6: **Question-answering process**

Figure 7: **Collecting user feedback metrics**

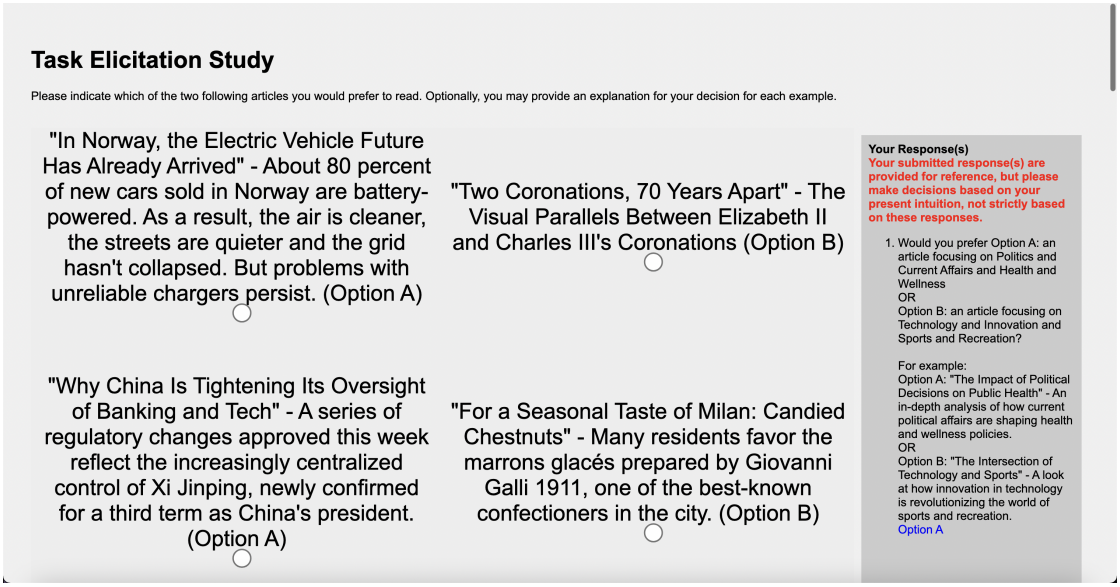

Figure 8: **Evaluation–user answering pairwise comparision test cases**

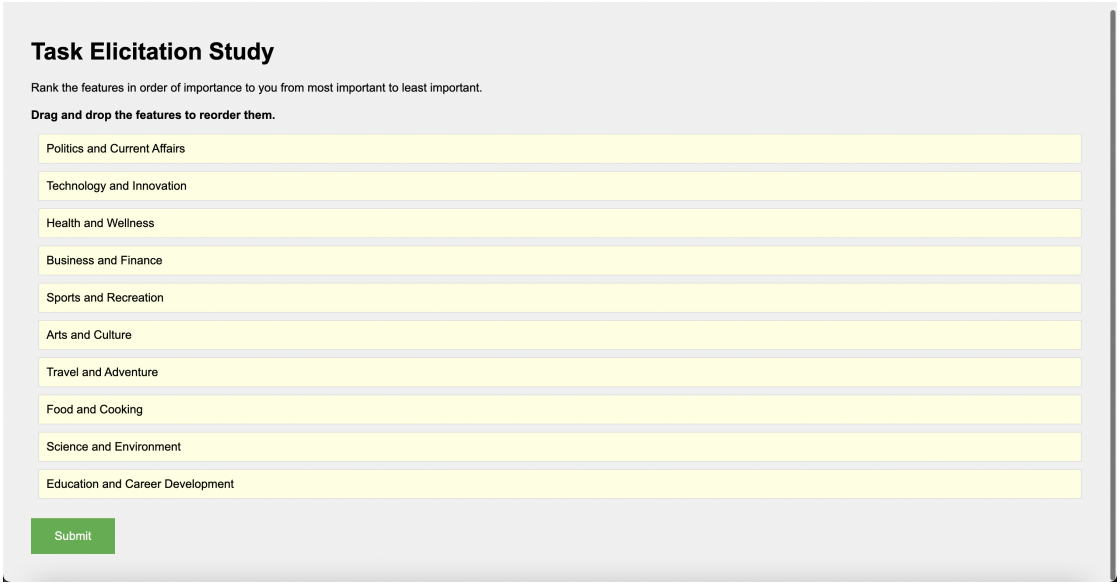

Figure 9: **User reorders NL features from most to least important**

1. feedback_challenge: *"How mentally demanding was interacting with the chatbot / writing your answer?"*
2. feedback_new_issues_interaction: *"To what extent did the chatbot raise issues or aspects about your preferences that you hadn't previously considered?"*
3. feedback_interaction_coverage_pretest: *"How comprehensively do you feel the chatbot's questions / your answer characterized your preferences about the task?"*
4. feedback_interaction_coverage_posttest: *"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?"*
5. feedback_testcase_use_history: *"When performing the *second* part of the task, to what extent did you refer back to your conversation history / answer from the first part of the task?"*
6. *feedback_lm_experience: "How much experience have you had (if any) with interacting with language models (e.g. ChatGPT, GPT4, etc.)?"*

As we can see, OPEN was not more challenging than the other methods across the different axes. Unsurprsingly, in the *User-Self Mapping* case, users went back to look at their conversation history much less frequently than other in other methods. Interestingly, users felt that LM Pairwise questions covered their preferences well pre-test, but this ranking dropped significantly post-test (and did not translate to a higher accuracy on the test set).

F LM Prompts

Below, we include the prompts used for the different LM-based tasks.

F.1 LM Open-ended Questions

```
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.\n\nPrevious questions: {
    interaction_history_formatted}.\n\
    nGenerate the most informative 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
    their preferences 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
```

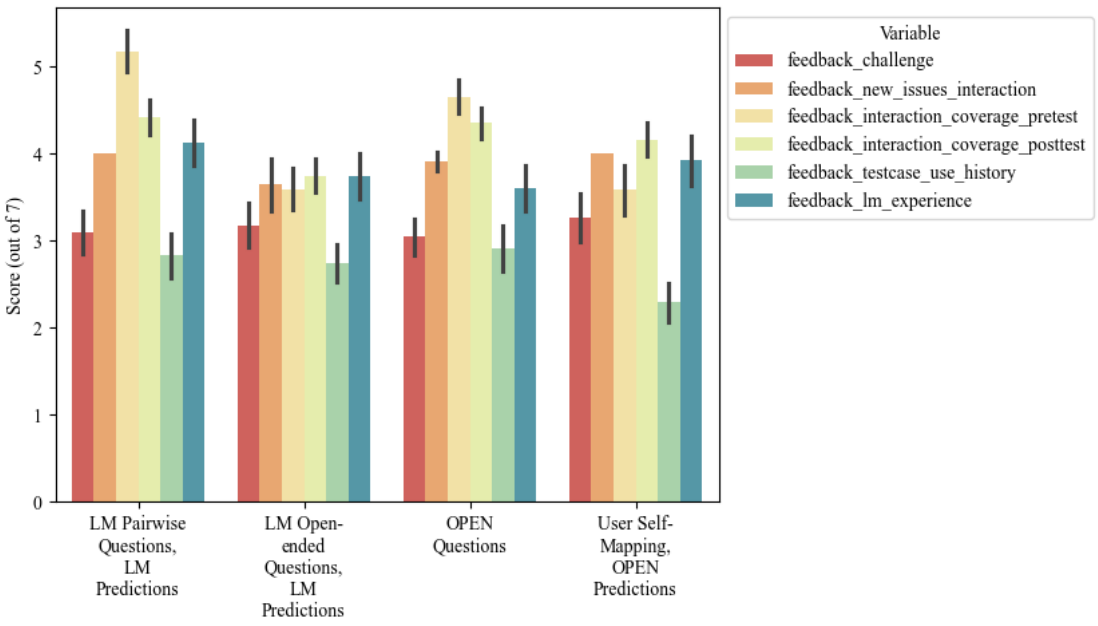

Figure 10

humans; do not use any jargon without explanation. Generate the question and nothing else.\n\nProvide your output in the format:\n\nQuestion: <question>

F.2 LM Pairwise Questions, LM Predictions

```
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.\n\nPrevious questions: {interaction_history_formatted}.\n\nGenerate the most informative pairwise comparison 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 their preferences 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 pairwise comparison question and nothing else.\n\nProvide your output in the format:\n\nWould you prefer\nOption A: <first article option>\nOR\nOption B: <second article option>?
```

F.3 LM Pairwise Question, OPEN Predictions

```
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.\n\nPrevious questions: {interaction_history_formatted}.\n\nThe following {num_features} features describe people's preferences for different kinds of articles: {features}.\n\nGenerate the most informative pairwise comparison question that, when answered, will reveal the most about the desired behavior beyond what has already been queried for above.\n\nMake sure your question addresses different aspects of their preferences 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.\n\nThen create two specific, real-world examples of news articles matching the description you provided. For each feature you use, make sure you match the wording of the feature verbatim to the list above. Provide your output in the format:\n\nWould you prefer Option A: an article with [features for option A]\nOR\nOption B: an article with [features for option B]?\n\nFor example: [Option A title and simple, one sentence description]\nOR\nOption B: [Option B title and simple,
```

```
one sentence description]
```

F.4 Extracting Environment Features

```
Your task is to learn what topics a user is
    interested in reading online articles
    about. Enumerate 10 binary topics (
    features) that may impact user's
    decisions when choosing which article to
     read. 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
    .\n\nOnly include the most important
    features in your list. Do not include
    features for which the user's preference
     would be obvious. Order the features
    from the most to least likely of
    interest.\n\nYour output should be in
    the following format:\n1) <first feature
    >\n2) <second feature>
```

F.5 Mapping BOED Pairwise Comparision to NL

```
Create two specific, real-world examples of
    news articles someone might be
    interested in reading based on the
    following question which juxtaposes two
    different news articles:\n\"Would you
    prefer Option A: an article with {
    pairwise_comparison_0}\nOR\nOption B: {
    pairwise_comparison_1}?\".\nThis
    question instantiates two articles based
     on their feature values. Features lie
    on a spectrum ranging from 0.0 to 1.0,
    where 0.0 corresponds to the absence of
    that feature and 1.0 indicates an
    extremely high presence of it.\n\nMake
    sure to maintain the relative difference
     between the two articles when
    generating the descriptions.\n\nProvide
    your output in the format:\nOption A: [
    Option A title and simple, one sentence
    description]\nOR\nOption B: [Option B
    title and simple, one sentence
    description]
```

F.6 LM Evaluation

```
Provide your best guess and the probability
    that it is correct (0.0 to 1.0) for the
    following question. Give ONLY the guess
    and probability, no other words or
    explanation. If you are unsure take your
     best guess (between Option A and Option
     B). For example:\n\nGuess: <most likely
     guess--either Option A or Option B--as
    short as possible; not a complete
    sentence!>\nProbability: <the
    probability between 0.0 and 1.0 that
    your guess is correct, without any extra
     commentary whatsoever; just the
    probability!>.\n\nA user has a
    particular set of preferences over what
    articles they would like to read. They
    have specified their preferences in a
    conversation below:\n{preferences}\n\
    nThe question is: Based on these
    preferences, which of the following two
    articles would the user prefer?\n\
    nOption A: {test_case_0}\nOR\nOption B:
    {test_case_1}
```