

Evaluating Human-Language Model Interaction

Mina Lee* Megha Srivastava Amelia Hardy John Thickstun
 Esin Durmus Ashwin Paranjape Ines Gerard-Ursin[§] Xiang Lisa Li
 Faisal Ladhak Frieda Rong Rose E. Wang Minae Kwon
 Joon Sung Park Hancheng Cao Tony Lee
 Rishi Bommasani Michael Bernstein Percy Liang*

Stanford University [§]Imperial College London

Abstract

Many real-world applications of language models (LMs), such as code autocomplete and writing assistance, involve human-LM *interaction*. However, the main LM benchmarks are *non-interactive* in that a system produces output without human involvement. To evaluate human-LM interaction, we develop a new framework, Human-AI Language-based Interaction Evaluation (HALIE), that expands non-interactive evaluation along three dimensions, capturing (i) the interactive process, not only the final output; (ii) the first-person subjective experience, not just a third-party assessment; and (iii) notions of preference beyond quality. We then design five tasks ranging from goal-oriented to open-ended to capture different forms of interaction. On four state-of-the-art LMs (three variants of OpenAI’s GPT-3 and AI21’s J1-Jumbo), we find that non-interactive performance does not always result in better human-LM interaction and that first-person and third-party metrics can diverge, suggesting the importance of examining the nuances of human-LM interaction.

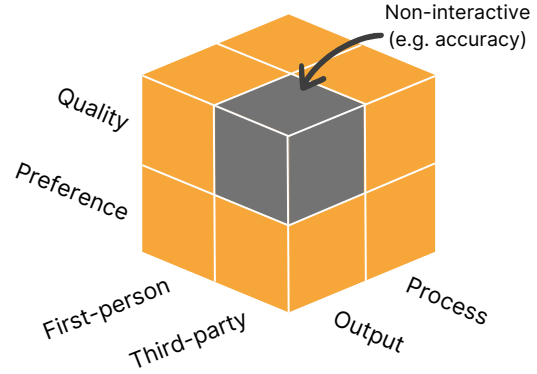



Figure 1: **Dimensions in human-LM interaction evaluation.** We propose a framework, HALIE, that expands on non-interactive evaluation along three dimensions: (i) we capture the full *process* in addition to the final *output* (targets); (ii) we capture the *first-person* subjective experience of users interacting with the LM in addition to the perspective of a *third-party* (perspectives), and (iii) we consider notions of *preference* beyond *quality* (criteria). These dimensions interact to define the full space of evaluation metrics (i.e., a 2x2x2 cube ) , which goes beyond standard, non-interactive evaluation (i.e., gray cell).

1 Introduction

Language models (LMs) have rapidly advanced, demonstrating unprecedented capabilities for generation and striking generality for tackling a wide range of tasks (Brown et al., 2020; Rae et al., 2021; Chowdhery et al., 2022). However, these models are at present primarily evaluated *non-interactively*: given an input text, a model generates a completion with the focus solely on the quality of the completion. Almost all benchmarks, even those with a diverse range of tasks such as BIG-Bench (Srivastava et al., 2022), GEM (Gehrmann et al., 2021), and HELM (Liang et al., 2022), encode this non-interactive view.¹

Our goal is to evaluate human-LM interaction instead of just model completions. LMs are already deployed in *interactive* settings, where humans work with them to brainstorm ideas (e.g., Jasper, Copy.ai), paraphrase sentences (e.g., Wordtune, QuillBot), reformulate queries (Nogueira and Cho, 2017), autocomplete sentences (Chen et al., 2019), write code (e.g., Coilot, TabNine), and so forth.² We anticipate that the adoption rate will continue to accelerate as LMs become more capable and novel

ently interactive (Paranjape et al., 2020; Thoppilan et al., 2022b; Shuster et al., 2022), but we are interested in the general capabilities of LMs.

²This interactive use of LMs aligns with our ultimate goal of *augmenting* human capabilities rather than *automating* them (i.e., intelligent augmentation), reflecting views by Engelbart (1962); Skagestad (1993, 1996); Shneiderman and Maes (1997); Horvitz (1999); Hassani et al. (2020); Brynjolfsson (2022); Shneiderman (2022).

*Direct correspondence to minalee@cs.stanford.edu, pliang@cs.stanford.edu; full contributions in Appendix A.

¹There are specific tasks such as dialogue that are inher-

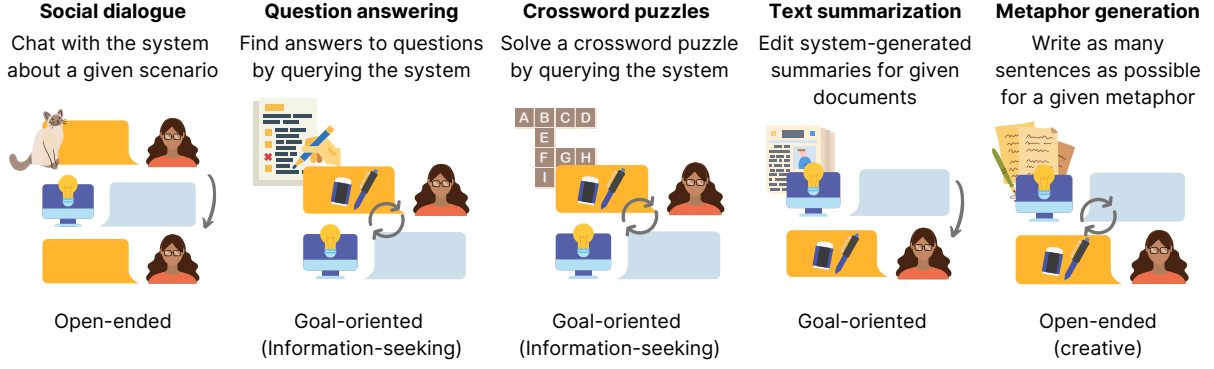


Figure 2: **Five tasks and human-LM interaction in the context of accomplishing the tasks.** We study tasks ranging from goal-oriented to open-ended and cover the common usages of LMs reported by Ouyang et al. (2022). When designing human-LM interaction for these tasks, we make use of various interaction styles, such as strict turn-taking (e.g., social dialogue) and iterative query reformulation (e.g., question answering).

use cases are discovered (Bommasani et al., 2021, §2.5).³ But as a community, if we explicitly or implicitly optimize for non-interactive performance, will this improve interactive performance?

We develop an evaluation framework, **Human-AI Language-based Interaction Evaluation (HALIE)** that puts interaction at the center. First, we make explicit that end users interact with a *system* (as opposed to a raw LM), which includes a user interface (UI), the system logic which constructs prompts and invokes the LM. The distinction between LM and system is important because, unlike non-interactive evaluation, interactive evaluation is fundamentally a function of the user, the LM, and the system, which acts like the glue. HALIE formally defines the system logic with states and actions. A *state* of the system is textual contents in the UI (e.g., dialogue history and the current text in a user input box), and an *action* is taken by a user (e.g., clicking the “send” button or typing into a text box) through the UI, which subsequently triggers the system to update the state. We define an *interaction trace* to be the resulting sequence of state-action pairs.

To evaluate human-LM interaction, HALIE defines metrics on interaction traces, which are organized along the following three dimensions (Figure 1)—targets, perspectives, and criteria: (1) *targets* include more than just the final output, and cover the entire interaction process (e.g., user queries and edits); (2) *perspectives* are not limited to third parties, but the users who interact with LMs

to capture first-person experiences; and (3) *criteria* include not only quality (e.g., accuracy), but also preference (i.e., attitudinal measures of humans such as enjoyment). In contrast, non-interactive benchmarking devises automatic metrics or third-party human annotations (perspectives) to solely evaluate the quality (criteria) of outputs (targets).

We design five tasks ranging from goal-oriented to open-ended (Figure 2) to capture a variety of different interactions, and construct an interactive system for each task by defining states and actions. With these systems, We evaluated four state-of-the-art LMs: three variants of OpenAI’s GPT-3 (Brown et al., 2020; Ouyang et al., 2022)—*TextDavinci* (text-davinci-001), *TextBabbage* (text-babbage-001), and *Davinci* (davinci)—and AI21’s Jurassic-1 (Lieber et al., 2021)—*Jumbo* (j1-jumbo). We choose these models to study the impact of model size (*TextDavinci* vs. *TextBabbage*), instruction tuning⁴ (*TextDavinci* vs. *Davinci*), and implementation details regarding training data, architecture, and other factors (*Davinci* vs. *Jumbo*). While the impact of these differences on non-interactive benchmark performance has been studied quite extensively (Brown et al., 2020; Ouyang et al., 2022; Liang et al., 2022), we want to study their effect in interactive settings.

From the 1015 interaction traces we collected, we observe that better non-interactive performance does not always lead to better human-LM interaction (e.g., Section 3.2). Furthermore, sometimes a subset of quality metrics might overpower other metrics when it comes to user preference

³For instance, over 300 applications were built within a year of the release of GPT-3 (OpenAI, 2021), and ChatGPT (Schulman et al., 2022) amassed 1 million users in five days after launching (New York Times, 2022).

⁴In this paper, instruction tuning refers to supervised fine-tuning LMs on human-written demonstrations and highly rated model completions (OpenAI, 2022).

(e.g., Section 3.1), or helpfulness perceived by users could exceed actual performance improvements (e.g., Section 3.3). We also find that a model that achieves the best performance based on third-party perspectives might be the worst based on first-person perspectives (e.g., Section 3.4), and visa versa (e.g., Section 3.5). We release our interaction traces, their replay links, and system interfaces at <https://github.com/minggg/halie>.

2 Framework

In this section, we introduce our framework, HALIE, for evaluating human-LM interaction. Concretely, we describe tasks and system construction for studying human-LM interaction, as well as interaction traces that we use to represent the interaction process. Lastly, we propose dimensions and metrics for evaluating human-LM interaction.

2.1 Solving tasks interactively

We study human-LM interaction in the context of *tasks*, which provide a structured context for facilitating and evaluating these interactions. Figure 2 shows five tasks we study in this paper: social dialogue, question answering, crossword puzzles, text summarization, and metaphor generation (see Section 3 for a detailed description of each task). The tasks span the spectrum from goal-oriented (e.g., question answering) to open-ended tasks (e.g., metaphor generation). They also have high coverage on the common usages of LMs reported by Ouyang et al. (2022) (e.g., generation, open QA, and brainstorming), showing that the selection captures how users currently interact with LMs.

2.2 Constructing an interactive system

For each task, we build a system that facilitates human-LM interaction in the context of the task. There are many ways to design the system based on the same LM: Users could query an LM and edit the output (i.e., post-editing), reformulate their queries based on the output (i.e., query reformulation), or query an LM and select one of the outputs (i.e., suggestion selection). These design decisions can have a significant impact on human-LM interaction (Clark et al., 2018; Buschek et al., 2021; Ippolito et al., 2022).

Language models. We consider an LM to be a black box that takes a text *prompt* and decoding parameters as input and stochastically returns a set of text *completions* (QueryLM in Algorithm 1).

Algorithm 1: Generate an interaction trace

```

 $s_0 \leftarrow$  task-specific contents
for  $t = 1, 2, \dots$  do
    User takes an action  $a_t$ 
    if  $a_t$  finishes the interaction then
        break
    else
        System updates the state  $s_{t+1} \leftarrow$ 
            Transition( $s_t, a_t$ )
    end
return  $[(s_1, a_1), (s_2, a_2), \dots]$ 

```

Function Transition(s_t, a_t):

```

if  $a_t$  requires querying an LM then
     $p = \text{CreatePrompt}(s_t)$ 
    Fetch completions  $c = \text{QueryLM}(p)$ 
    return ShowCompletions( $s_t, c$ )
else if  $a_t$  fills out a survey then
    return  $s_t$  with results of survey
else
    // (e.g.,  $a_t$  edits user input)
    return  $s_t$  updated with  $a_t$ 
end

```

System logic. Given an LM, we define a system logic to be a set of states, potential user actions, and a transition function that specifies how states get updated, borrowing the terminology and intuitions from Markov decision processes. A *state* s_t captures everything the system needs to know about the current interaction at time t . This is primarily the visual state—the text contents of each text field in the graphical interface (e.g., dialogue history and the current text in user input), but also includes any hidden dependencies, such as the history of previous dialogues. Given a state, a user can take one of a set of *actions*. For example, the user may type to modify user input or click a button.

Given a state s_t and an action a_t , the system produces the updated state s_{t+1} through a *transition function* (Transition in Algorithm 1). When the action is typing to modify user input, the update is a straightforward rendering of textual changes. When an action requires querying an LM, the system constructs a prompt (CreatePrompt in Algorithm 1), queries the underlying LM with the prompt (QueryLM in Algorithm 1), and updates the state with the completions from the LM (ShowCompletions in Algorithm 1). For example, in a dialogue system (Figure 3), a user may write

“Thanks!” in user input and click the “send” button, which will trigger the system to construct a prompt (e.g., concatenating current dialogue history and user utterance), fetch a completion from the LM, and show it to the user.

Interaction traces. We define a sequence of state-action pairs as an *interaction trace*, similar to Lee et al. (2022). Algorithm 1 shows the process of generating an interaction trace as a result of human-LM interaction.

Design considerations. We highlight important design considerations that can directly affect the dynamics of human-LM interaction. First, CreatePrompt determines how much control users have over prompts. For example, CreatePrompt can simply take the content of user input, or enforce a pre-defined prompt regardless of the user input. It can also decide how much each query depends on past interactions, by providing previous interactions as part of the prompt. Second, decoding parameters in QueryLM influence the quality of completions (Holtzman et al., 2020), thus changing the dynamics of human-LM interaction (Lee et al., 2022). In our experiment, we choose a set of decoding parameters for each task based on initial pilots and use it for all models. Third, ShowCompletions controls how model completions are shown to users. For instance, how many completions are shown at a time, and how intrusive are they? These decisions can have a trade-off between efficiency and ideation (Buschek et al., 2021), or lead to distraction or cognitive load (Clark et al., 2018).

2.3 Evaluating human-LM interaction


To evaluate interaction traces, we expand on non-interactive evaluation along three dimensions and define metrics characterized by those dimensions.

Dimensions. Figure 1 shows three dimensions in human-LM interaction evaluation. The first dimension is *targets* (i.e., what to evaluate). Standard evaluation only considers the final *output* as a target, which is often simply a model completion c , but can be a result of more complicated processes (e.g., sample multiple completions and use heuristics to construct the final output). In contrast, we consider the entire interaction *process*, which is represented as an interaction trace $[(s_1, a_1), (s_2, a_2), \dots]$ in HALIE.

The second dimension is *perspectives* (i.e., whose perspective is reflected). Non-interactive evaluation solely depends on a *third-party* perspective; automatic metrics can be computed on an interaction trace without human intervention; likewise, human evaluation is done by third parties who did not interact with LMs in any way. In the interactive setting, however, evaluation should reflect the *first-person* perspective of the user who actually takes actions a_t to interact with an LM through a system.

The last dimension is *criteria* (i.e., how to evaluate). Standard evaluation focuses on *quality*, which tend to be objective and include metrics like accuracy and fluency (defined for outputs) or helpfulness and edit distance (defined for processes). On the other hand, interactive evaluation adds attitudinal measures of a human, which we define as *preference*. These measures tend to be subjective and often captured by metrics like enjoyment and satisfaction. Note that preference criteria apply to both first-person and third-party perspectives. One example of preference metrics with third-party perspectives is creative story writing: some readers may not like certain kinds of stories (low preference), although they think that they are good (high quality). Also, sometimes metrics can reflect both preference and quality when human preference is aligned with the quality metrics of interest.

Metrics. For each task, we define a set of metrics that cover the space defined by the three dimensions. Some metrics are a function of interaction traces (e.g., number of queries users made), whereas some metrics only depend on outputs (e.g., accuracy in solving a crossword puzzle). Some metrics are defined based on survey responses from users (e.g., satisfaction over the final output) and third-party evaluators (e.g., consistency of a summary given a document).

Table 1 shows the mapping from the space to the metrics in the five tasks (see Table 21 in Appendix D for the full list of metrics). Note that non-interactive evaluation only covers one combination of the dimensions (i.e., outputs for targets, third-party for perspectives, and quality for criteria) corresponding to one gray cell in Figure 1. In contrast, our evaluation considers all combinations, corresponding to  in Figure 1.

Dimensions			Tasks				
Targets	Perspectives	Criteria	Social dialogue	Question answering	Crossword puzzles	Text summarization	Metaphor generation
Process	First-person	Preference	Reuse	Ease	Enjoyment	Improvement	Enjoyment
Process	First-person	Quality		Helpfulness	Helpfulness		Helpfulness
Process	Third-party	Preference	Interestingness Specificity	Queries	Queries	Edit distance	Queries
Process	Third-party	Quality					
Output	First-person	Preference		Fluency	Fluency	Consistency	Satisfaction
Output	First-person	Quality					
Output	Third-party	Preference		Accuracy	Accuracy	Consistency	Interestingness
Output	Third-party	Quality					
							Aptness

Table 1: We define a set of metrics for evaluating human-LM interaction across 5 tasks (see Appendix D for the full list); each metric can be characterized along three dimensions (targets, perspectives, and criteria). Note that some metrics, such as the number of *queries* from users, can be viewed as proxies for different quality (e.g., efficiency) or preference (e.g., enjoyment) metrics depending on the task.

3 Experiments

In this section, we introduce five tasks, construct systems for them, have human users interact with LMs using the systems, and evaluate human-LM interaction. For LMs, we choose four state-of-the-art LMs: three variants of OpenAI’s GPT-3 (Brown et al., 2020; Ouyang et al., 2022)—*TextDavinci* (text-davinci-001), *TextBabbage* (text-babbage-001), and *Davinci* (davinci)—and AI21’s Jurassic-1 (Lieber et al., 2021)—*Jumbo* (j1-jumbo). This allows us to study the effect of model size (*TextDavinci* vs. *TextBabbage*), instruction tuning (*TextDavinci* vs. *Davinci*), and implementation details regarding training data, architecture, and others (*Davinci* vs. *Jumbo*).

3.1 Social dialogue

Dialogue is a natural and flexible way to engage in communication, which makes it an intuitive and popular mode of interaction for LMs. In this section, we evaluate human-LM interaction in the context of open-ended dialogue about social situations (henceforth social dialogue). We defer related work and experimental details to Appendix B.1 for space.

Task. We consider the task where given a social *scenario*, users converse with a system (or chatbot) about the scenario. Users initiate the conversation and take turns with the chatbot to carry out the conversation until users choose to finish it. We randomly select ten scenarios from the EmpatheticDialogues (Rashkin et al., 2019) and CommonsenseDialogues datasets (Zhou et al., 2021).

System logic. Figure 3 shows a dialogue system *state* consisting of scenario, dialogue history, and user input and possible *actions*: pressing a key

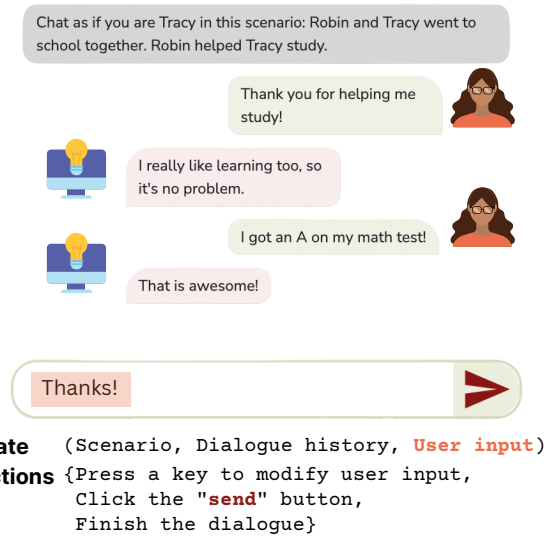


Figure 3: [Social dialogue] A dialogue system’s *state* consists of a scenario, dialogue history, and user input. When users take an *action* (e.g., clicking the “send” button), the system updates the state (e.g., updating the dialogue history with a model completion).

to modify user input, clicking the “send” button to submit their input, and finishing the dialogue. When users click the “send” button, the system creates a prompt with five dialogue examples for in-context learning and the current dialogue history, invokes an LM with the prompt, fetches a completion, and shows the completion in the dialogue history in the interface (ShowCompletions). See Appendix B.1.2 for details on CreatePrompt and QueryLM.

User study procedure. We recruited a total of 189 crowd workers (or users) on Amazon Mechanical Turk. For each of the scenarios and models, we had three users each produce a dialogue. To ensure each dialogue is long enough, the interface allowed

Model	Fluency	Sensibleness	Specificity (/100%) ↑	Humanness	Interestingness	Inclination	Reuse (/5) ↑
TextDavinci	93 ± 1.0	94 ± 1.0 **	83 ± 1.6 *	37 ± 2.0	36 ± 2.0	91 ± 1.2	4.09 ± .14 **
TextBabbage	90 ± 1.4	84 ± 1.7 *	81 ± 1.8 *	29 ± 2.1	30 ± 2.1	88 ± 1.5	3.35 ± .16 *
Davinci	92 ± 1.3	89 ± 1.4	92 ± 1.3 **	24 ± 2.0	27 ± 2.0	91 ± 1.3	3.80 ± .17
Jumbo	89 ± 1.3	86 ± 1.5	84 ± 1.5	24 ± 1.8	32 ± 2.0	87 ± 1.4	3.21 ± .20 *

Table 2: [Social dialogue] Users perceived TextDavinci to have the best *fluency*, *sensibleness*, *humanness*, *interestingness*, and *quality*, but they expressed the similar *inclination* to continue interacting with Davinci whose responses were most *specific* to what users had said. For the first six metrics, the numbers indicate the percentages of system responses under each metric (0–100%). The numbers for *reuse* indicate the ratings of each model after completing a dialogue (1–5). The means, standard errors, and statistical significance⁵ are shown in the table.

users to finish the conversation only after taking more than ten turns or exceeding 250 words in total. At the end of the conversation, users completed a survey about their experience.

Survey questions. We ask users to evaluate the *interestingness*, *sensibleness* (i.e., whether a chatbot response makes sense), and *specificity* (i.e., whether a chatbot response is specific to what a user had said) of a chatbot, following the survey questions proposed by Thoppilan et al. (2022a). We also ask users to evaluate the *fluency* of the chatbot and their *inclination* to continue talking with the chatbot, as proposed by Smith et al. (2022a). We ask dataset-specific questions regarding empathy (EmpatheticDialogues) and commonsense understanding (CommonsenseDialogues), which are labeled as *humanness* collectively. Finally, after a dialogue, we ask if users are willing to talk to the chatbot again using a 5-point Likert scale: *reuse*. See Appendix B.1.3 for the full list of questions.

Results. For social dialogue, we specifically care about models’ ability to show an emotional connection to users, by being empathetic and having commonsense about social situations (*humanness*), in addition to being *fluent* and *sensible*, and *specific* to what they have said. Independently, users may have their own subjective views on whether system responses are interesting and make them want to continue interacting with the system, which are measured by preference metrics *interestingness* and *inclination*, respectively.

① Instruction tuning improves performance on most quality metrics, but not specificity.

⁵Throughout the paper, results are denoted by * if models had a significant effect relative to TextDavinci, * if significant relative to TextBabbage, * if significant relative to Davinci, and * if significant relative to Jumbo at the $p = 0.05$ level using a Tukey-Kramer test.

Table 2 shows that the instruction tuned TextDavinci scored highest on most metrics, including *fluency*, *sensibleness*, and *humanness*. In particular, users evaluated TextDavinci to have considerably higher *humanness* than Davinci. And the instruction tuned TextBabbage model, despite its much smaller size, was reported to have higher *humanness* than Davinci or Jumbo.

However, instruction tuning decreased the *specificity* of model completions as shown in the significant drop in performance of TextDavinci compared to Davinci (Table 2). From this observation, we hypothesize that general instruction-tuning makes models better aligned with most metrics at the cost of being more generic (see Appendix B.1.4 for the examples of model completions), although previous work suggests that dialogue-specific fine-tuning improves all metrics including *specificity* (Thoppilan et al., 2022b).

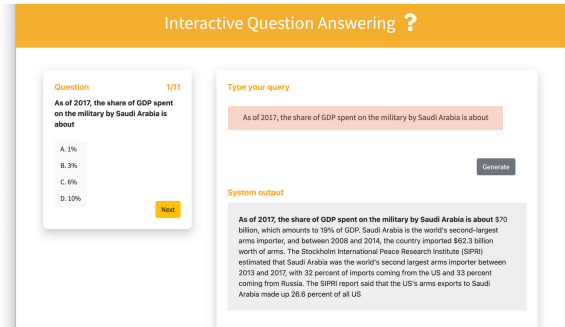
② Users may prefer to interact with a more specific LM. Although users evaluated TextDavinci to be the best LM according to most metrics, they expressed similar *inclination* to continue interacting with Davinci during dialogue. Given that Davinci was perceived to be worse than TextDavinci on all metrics except *specificity* and *inclination*, we hypothesize that users expressed an inclination to interact with Davinci *because* its responses were the most specific to what users had said. We recommend further study of the relationship between specificity and inclination to interact with an LM, and in particular, whether specificity affects *reuse* of the LM once the initial novelty of this interactive experience has faded.

3.2 Question answering

Question answering is a canonical task in NLP. We consider an *interactive* version of question an-

Model	Accuracy (/100%) \uparrow	Time (min) \downarrow	Queries (#) \downarrow	Ease	Fluency (/5) \uparrow	Helpfulness
TextDavinci	69 \pm 2.2	1.36 \pm .13	1.78 \pm .06 **	4.53 \pm .08	4.35 \pm .07 ***	4.60 \pm .07 ***
TextBabbage	52 \pm 2.8	1.77 \pm .33	2.57 \pm .13 *	4.09 \pm .12	3.84 \pm .12 ***	3.84 \pm .12 ***
Davinci	48 \pm 2.7	2.09 \pm .14	2.66 \pm .12 *	3.73 \pm .13	3.22 \pm .11 **	3.52 \pm .13 ***
Jumbo	54 \pm 2.9	1.67 \pm .09	2.32 \pm .11	3.87 \pm .14	3.17 \pm .11 **	3.26 \pm .14 ***

Table 3: **[Question answering]** Performance averaged across all questions conditioning on the use of AI assistance. Users assisted by TextDavinci achieved the highest *accuracy* while requiring the least effort (*queries*, and *ease*) and being perceived to be the most *fluent* and *helpful*. The numbers indicate means and standard errors, and the markers denote statistical significance,⁵ conditioning on the use of AI assistance; when the assistance was provided, users queried the system 86% of the time.



State (Multiple-choice question, **User input**, System output)

Actions {Press a key to modify user input, Click the "generate" button, Select one of the multiple choices, Click the "next" button, Finish the quiz}

Figure 4: **[Question answering]** The system’s *state* consists of a multiple-choice question, user input, and system output. When users take an *action* (e.g., clicking the “generate” button), the system updates the state (e.g., updating the system output with a model completion).

swering where a user is asked to answer a question given access to an LM that they can query. In practice, users tend to query information systems repeatedly, refining and reformulating their queries in response to the system’s output (Huang and Efthimiadis, 2009a; Jansen et al., 2009; Jiang et al., 2013). For related work and details of experiments, see Appendix B.2.

Task. Given a sequence of multiple-choice questions (or *quiz*), users try to select the correct answer with advice from an LM. As an example, consider this multiple-choice question:

Before Nixon resigned how many believed he should be removed from office?
A. 79% B. 98 % C. 33 % D. 57%

Users can query the system multiple times with

free-form text inputs; they might start with the question verbatim, re-phrase the question to increase specificity (e.g., “Did 79 percent of people believe Nixon should have been removed from office?”), or find a different way to query, perhaps bringing in their own prior knowledge (e.g., “Nixon approval rating after watgate”).

We use questions from the Measuring Massive Multitask Language Understanding (MMLU) dataset (Hendrycks et al., 2021). Concretely, we first chose five diverse subjects from the dataset (Global facts, Nutrition, US foreign policy, College chemistry, and Miscellany), and selected 6 questions from each subject to construct a pool of 30 questions. We constructed quizzes by randomly selecting ten questions from the pool and adding one attention check question in the middle.

System logic. Figure 4 shows a *state* of our QA system consisting of a multiple-choice question, user input, and system output. Users can take the following *actions*: pressing a key to modify user input, clicking the “generate” button to query the system, selecting one of the multiple choices, clicking the “next” button to submit their answer, and finishing the quiz. When users click the “generate” button, the system creates a prompt with the user input, invokes an LM with the prompt, fetches a completion, and shows the completion in the system output box in the interface (ShowCompletions). See Appendix B.3.1 for the exact implementation of CreatePrompt and QueryLM.

User study procedure. We recruited 342 crowd workers (users) on Amazon Mechanical Turk. Each user answered half of the questions with assistance from a single LM (treatment) and answered the other half without assistance (control). We instructed users not to use search engines to answer questions, and alerted users (“Please do not switch

tabs or open a new window during your participation.”) when we detected the tab switching. At the end of each quiz, users completed a survey about their experience.

Survey questions. We asked users for first-person evaluation of *fluency*, *helpfulness*, and *ease of interaction* of the system on a 5-point Likert scale. Also, we asked users to describe why they found the system helpful or unhelpful, provide adjectives to describe the system, and explain whether their interaction changed over the course of answering questions. See Appendix B.3.2 for the full list of questions.

Results. For the goal-oriented QA task, third-party evaluation is characterized by *accuracy* (i.e., how many questions users get correct answers for) and *efficiency* (i.e., how much effort users put in to find an answer) (Dibia et al., 2022). Because the bulk of a user’s time in interactive tasks is spent reacting to system outputs, we count the number of *queries* needed to answer each question as a proxy measurement for efficiency.

① Users with LM assistance generally outperform the user or LM alone, but not always. Figure 5 shows human+LM accuracy, zero-shot LM accuracy,⁶ and human-LM interactive accuracy broken down by question category. Users assisted by *TextDavinci* outperformed users alone in all categories. Users with LM assistance generally outperformed an LM alone, with the notable exception of the US Foreign Policy category, where both *TextDavinci* and *Davinci* achieved better accuracy as zero-shot QA systems than as interactive assistants. For the Global Facts and Miscellany categories, users with LM assistance significantly outperformed zero-shot LMs. In these contexts, users are able to figure out how to query the LM and when to trust it.

② Non-interactive performance does not always lead to better human-LM interaction. Further analysis of Figure 5 shows that for the Nutrition category, *TextBabbage* had the worst accuracy as a zero-shot QA system, but achieved the best performance as an interactive LM assistant (tied with *TextDavinci*). For US Foreign Policy, *Jumbo* performed worst as a zero-shot QA system and *Davinci* performed best, but as an LM

assistant *Jumbo* outperformed *Davinci*. This supports the broader point that non-interactive performance is an imperfect proxy for evaluating human-LM interaction.

③ Instruction tuning improves accuracy and efficiency for human-LM interaction. Table 13 shows that *TextDavinci* was the most efficient and accurate tool (1.78 queries/question with 69% accuracy). In contrast, *Davinci* was least efficient, requiring an average of 1.5x as many queries to answer a question with only 48% accuracy. Despite its smaller size, *TextBabbage* performed better than *Davinci* on most metrics, demonstrating the effectiveness of instruction tuning for this task. Instruction tuned models were also perceived most favorably in first-party survey evaluation.

3.3 Crossword puzzles

Crossword puzzles have been studied as a challenging task for AI systems (Ginsberg, 2014; Littman et al., 2002; Wallace et al., 2022b; Rozner et al., 2021). Unlike multiple-choice QA, solving a crossword puzzle is a *generative* task requiring open-ended responses to clues. The crossword puzzle task also provides additional structure, whereby a user can check whether a candidate answer satisfies the lexical constraints of the puzzle. Finally, clues are often not straightforward (e.g. “Christmas in Chamonix”), and a user might need to reformulate the query to find the desired information.

Task. A crossword puzzle requires solving a series of word clues (e.g., “Oscar the Grouch’s home”) that correspond to either rows (“Across” clues) or columns (“Down” clues) of white squares on a rectangular grid. As users solve one clue and places letters in squares, they reveal partial information (e.g., shared letters) that can affect their strategy for solving future clues. In our setting, users try to complete an entire crossword puzzle with the ability to query an LM-based “AI Teammate” via a chat-based interface. Users may message the AI Teammate with free-form text, where each individual message is used as the full prompt to the underlying LM.

System logic. Figure 6 shows a *state* of our crossword system, which we adapt from *Down*

⁶For fully-automated models, we report deterministic results with `temperature = 0`.

⁷The zero-shot, non-interactive QA systems’ accuracy can be found at https://nlp.stanford.edu/helm/interactive_qa_mmlu/?suite=interactive_qa_mmlu&group=mmlu

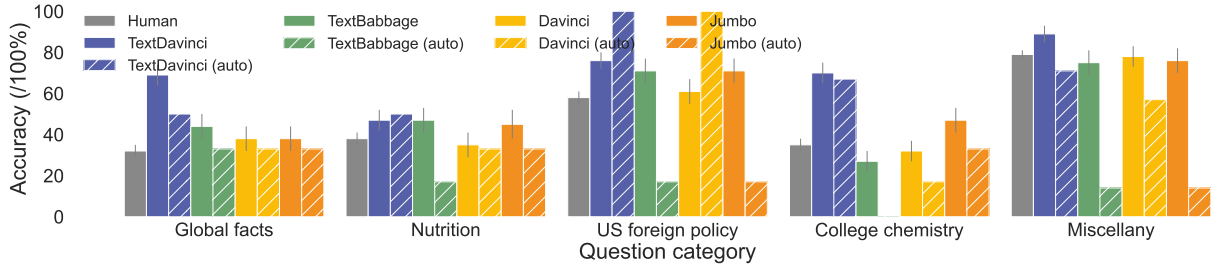
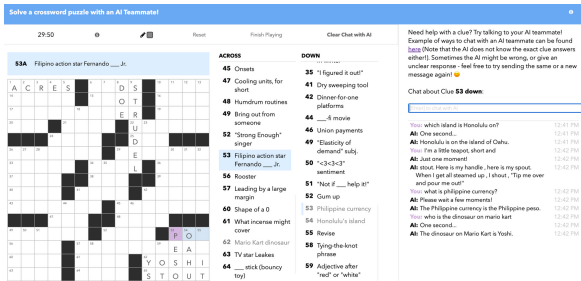


Figure 5: **[Question answering] Per-question accuracy broken down by category.** With **TextDavinci**, human-LM interaction results in improved accuracy for Global Facts, College Chemistry, and Miscellany, but not Nutrition and US foreign policy. The shaded bars indicate the performance of models as zero-shot, non-interactive QA systems when given a question text verbatim,⁷ and gray bars indicate user performance without AI assistance. The results are averaged across questions for which users were provided with AI assistance.

Model	Accuracy (letter) (/100%) ↑	Accuracy (clue) (/100%) ↑	Fluency	Helpfulness (/5) ↑	Ease	Enjoyment
TextDavinci	63 ± 2.9 *	53 ± 3.4 *	3.65 ± .10 **	3.14 ± .12 **	4.35 ± .10 **	2.91 ± .20 ***
TextBabbage	47 ± 3.3 *	38 ± 3.5 *	3.14 ± .13 **	2.27 ± .14 *	3.78 ± .15 *	2.19 ± .22 *
Davinci	55 ± 3.5	46 ± 3.6	2.26 ± .11 *	1.92 ± .10 *	3.32 ± .14 *	1.92 ± .17 *
Jumbo	56 ± 2.8	45 ± 3.1	2.30 ± .10 **	2.20 ± .10 *	3.08 ± .15 **	1.66 ± .18 *

Table 4: **[Crossword puzzles]** Users assisted by **TextDavinci** found their model more *fluent*, *helpful*, and *easy* and *enjoyable* to interact with compared to other models, and in general provided more accurate solutions across all puzzles. However, while users with **Davinci** and **Jumbo** performed worst on the self-reported survey metrics, users with **TextBabbage** had the worst *accuracy*, suggesting a disconnect between first-person preference and automated quality metrics. The numbers indicate means and standard errors, and the markers denote statistical significance.⁵



State (Puzzle, Selected clue, User letters, Dialogue history, User input)

Actions {Press a key to modify user input, Press the enter key to submit input, Select a square in the puzzle, Enter a letter into a square, Select a clue from the list, Finish the session}

Figure 6: **[Crossword puzzles]** The system’s *state* consists of a crossword puzzle, selected clue (can be none), user letters entered in the puzzle, dialogue history, and user input. When users take an *action* (e.g., pressing the enter key after writing user input), the system updates the state (e.g., updating the dialogue history with a model completion).

for a *Cross*, an open-source software for the multi-player crossword puzzles. Users can take the following *actions*: pressing a key to modify user

input, pressing the “enter” key to submit input, selecting a square in the puzzle, entering a letter into a square, selecting a clue from the list, and finishing the session. While attempting to solve different crossword clues (ShowCompletions), users (with the chat ID: You) are able to repeatedly interact with an LM (with the chat ID: AI) via zero-shot prompts in a dialogue-based chat interface. Although we display the chat history to aid players in remembering past information, only the most recent prompt from the player is sent to the LM. The same LM is fixed throughout the course of the task, which ends either after 30 minutes or when all clues are correctly solved. See Appendix B.4.2 for details on CreatePrompt and QueryLM.

User study procedure. We recruited 350 workers (users) on Amazon Mechanical Turk, split across each of the four language models and five puzzles (LOVE, SIT, NYT-1, NYT-2, and ELECT). Each user was provided a link to the interface, and asked to engage with the puzzle (e.g., solve clues, interact with the LM) for at least 30 minutes. Afterwards, each user completed a survey about their experience interacting with the AI Teammate. Further details including instructions and information about

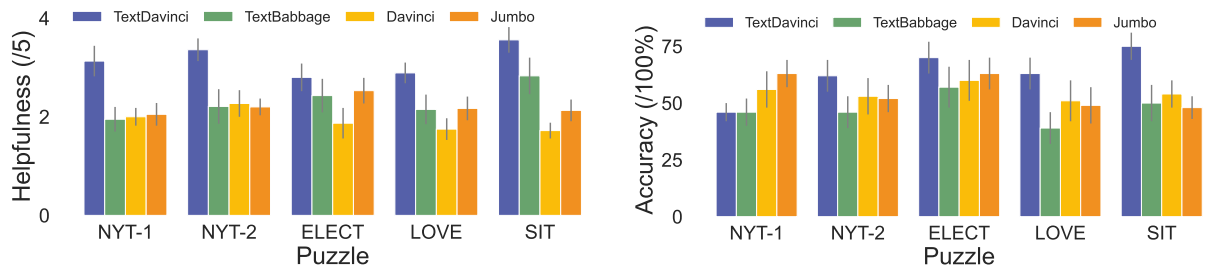


Figure 7: **[Crossword puzzles]** Although across all puzzles, users assisted by **TextDavinci** were significantly more likely to report the model as helpful (left), they did not always provide significantly more accurate puzzle solutions (right). In fact, for the NYT-1 puzzle, users assisted by **TextDavinci** were significantly less likely to provide accurate solutions, despite being more likely to find the model helpful.

compensation can be found in the Appendix B.4.3.

Survey questions. After playing with provided crossword puzzles, users are then asked to rank different qualities of the AI assistant on a 5-point Likert scale, including *fluency*, *helpfulness*, *enjoyment*, and *ease of the interaction* with the AI Teammate. Users are additionally asked to describe why they found the teammate helpful or unhelpful, what adjectives best describe the assistant, and whether their interaction changed over the course of answering questions. See Appendix B.4.4 for the full list of the questions.

Results. Solving a crossword puzzle is simultaneously an open-ended goal-oriented task and a form of entertainment—therefore, third-party evaluation should consider both *accuracy* and *engagement* of the human-LM interaction. Because some users may enjoy solving crossword puzzles independently, we count the number of *queries* used over the course of the task as a proxy measurement for user preference to engage with a given LM.

① Helpfulness perceived by users exceeds accuracy improvements. We measure the perceived *helpfulness* of a model based on a post-task survey question with a 5-point Likert scale. Figure 7 shows that users significantly preferred **TextDavinci** over other models with respect to helpfulness for all puzzles. Interestingly, this perception of helpfulness is not necessarily reflected in the overall interaction *accuracy*: as shown in Figure 7, assistance from **TextDavinci** led to statistically significantly higher crossword letter accuracy only for the SIT crossword puzzle; for the NYT-1 puzzle, users actually performed *worse* with both instruction tuned models (**TextDavinci** and **TextBabbage**). One hypothesis for this behavior is that these models are capable of providing confident

and fluent misinformation, which can be incorrectly perceived to be helpful. Furthermore, the variance in model performance between puzzles strongly suggests that different puzzles’ distributions over clue types may require different capabilities from LMs—in Table 10, we examine performance across 6 clue category types, and find that LMs, and in particular **TextDavinci**, were most useful for clues about factual knowledge, but struggle with word-play and phrases.

② Short prompts exacerbate misinformation and toxicity. We observed significant amounts of misinformation across all models. Misinformation was particularly pernicious using **TextBabbage**, which achieved relatively positive responses in user-reported survey responses but the lowest objective accuracy for puzzle solutions provided by users (Table 4). We include a large set of examples of misinformation from **TextBabbage** and **TextDavinci** in Appendix B.4.7. While we believe misinformation played a big role in the difference between user perceived and actual performance on the task, the overall effect of misinformation might be more nuanced: the structure of crossword puzzles provides constraints that may help users more quickly identify incorrect model suggestions.

Additionally, we observed instances of toxic content in generated crossword outputs. Explicit responses often occurred when users, perhaps influenced by the style of crossword clues, provided short keyword prompts: e.g., “japanese car”, “nick wilde”, or “synonym of delayed.” Short, ungrammatical prompts may more closely match toxic patterns in the training data (e.g. pornographic video captions or social media comments). Other safety concerns we observed included generating personal phone numbers and YouTube accounts belonging to non-celebrities, as well as discussing

more sensitive topics, which we describe further in Appendix Section B.4.8. Longer prompts that setup the appropriate context might mitigate toxic outputs, but its effectiveness merits further study.

③ Users demonstrate diverse engagement behavior. The relatively long period of interaction between users and the AI Teammate in the crossword puzzle task provides an opportunity to closely study engagement, which we measure in two ways. First, we ask users in the post-task survey to answer “*How enjoyable was it to interact with the AI Teammate?*”, and show in Table 4 that users found *TextDavinci* significantly more enjoyable to use (for some puzzles, *TextBabbage* was also rated highly), suggesting that users tended to enjoy interacting with LMs that were most helpful for the task. However, in Figure 14 in the Appendix, we take a closer look at engagement across the entire 30 minutes task length, and observe a diverse set of user behaviors: while some users decided to stop querying the AI Teammate after receiving incorrect responses, others chose to solve clues independently at the start and only query later when stuck. A long-tail of users, mostly interacting with *TextDavinci* and *TextBabbage*, had significant sustained interaction throughout the course of the puzzle, often experimenting with re-phrasing prompts to elicit more helpful responses. Finally, unlike with the use of AI solvers that seek to solve a crossword puzzle perfectly (Wallace et al., 2022a), some users found the challenge of figuring out how to best use the AI Teammate itself entertaining:

Davinci Player: This is an enjoyable task that may not be as fun if the AI would give you all the answers!

3.4 Text summarization

Text summarization is a long-standing problem in NLP (Luhn, 1958; Mani, 1999; Spärck Jones, 1999; Nenkova and McKeown, 2012). Notably, it has been studied in interactive settings for multi-document summarization, where the users interact with the system to query for additional information to expand upon an initial summary (Avinesh et al., 2018; Shapira et al., 2021, 2022). In contrast, we focus on human-LM interaction for single-document summarization, where users interact with the system to correct LM-generated summaries. In addition, as users correct summaries for a sequence of documents, we provide previous human-edited summaries as examples to the

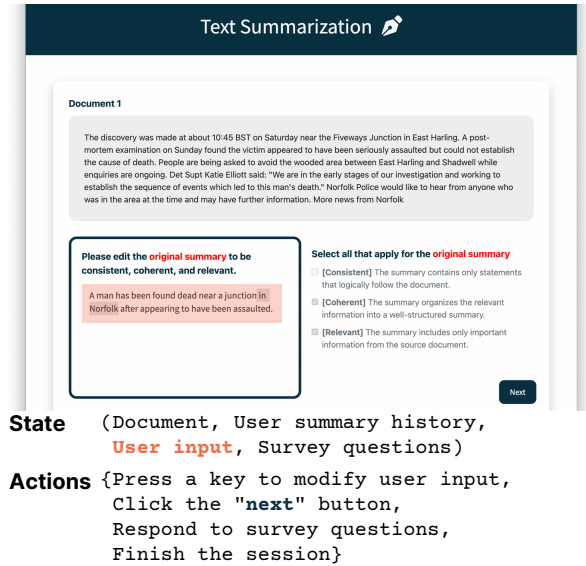


Figure 8: [Text summarization] The system’s *state* consists of a document, user summary history, user input, and survey questions. When users take an *action* (e.g., responding to survey questions and clicking the “next” button), the system updates the state (e.g., showing the next document and model-generated summary).

system to improve future summaries via in-context learning. For more related work and details of experiments, see Appendix B.5.

Task. We consider the task, where given a document and a model-generated summary, users edit the summary to be consistent, relevant, and coherent. We randomly select 964 documents from XSum dataset (Narayan et al., 2018) and construct 20 summarization *sessions* by randomly choosing ten documents per session without replacement.

System logic. Figure 8 shows the system state and actions. A summarization system *state* consists of a document, user summary history (not visible to users), user input, and survey questions. Users can take the following *actions*: pressing a key to modify a model-generated summary, clicking the “next” button, responding to survey questions, and finishing the session.

One notable difference in the summarization system’s state is that *CreatePrompt* generates a prompt by concatenating all the previous document and user-edited summary pairs in the user’s summary history as part of the prompt. This enables us to study whether an LM can learn from user examples and improve over time, as well as how users behavior change as the models learn or fail to learn from their examples. See Appendix B.5.2

for details on CreatePrompt and QueryLM.

User study procedure. We recruited 39 crowd workers (or users) on Amazon Mechanical Turk. For each model, we collected 20 summarization sessions (80 in total), while allowing the same users to participate multiple times. Upon receiving a model-generated summary for each document, users were asked to edit the summary to be consistent, relevant, and coherent. To encourage users to pay attention to the three quality criteria when editing, we asked users to evaluate the consistency, relevance, and coherence of the original (model-generated) summary and edited summary before and after editing. At the end of a summarization session, users completed a survey about their overall experience interacting with the system.

Survey questions. We ask per-summary and per-session questions to the users. Summary-level questions ask *consistency*, *relevance*, and *coherence* of the original and edited summaries, following [Fabbri et al. \(2021\)](#). Session-level questions evaluated users’ overall perceptions of the summarization system with respect to its *helpfulness* and *improvement over time*. See Appendix B.5.2 for the full list of questions.

Third-party human evaluation. We also asked third-party evaluators to evaluate the consistency, relevance, and coherence of a subset of the summaries generated by LMs. To this end, we randomly sampled 100 documents from our user study and recruited 18 workers (third-party evaluators) on Amazon Mechanical Turk to assess the summaries written for the documents. Each summary was assessed by 3 different evaluators.

Results. In text summarization, we ask third-party evaluators to evaluate the quality of summaries with respect to their *consistency* (i.e., all the information in the summary is inferred from the document), *relevancy* (i.e., the summary includes only important information and no excess information), and *coherence* (i.e., the summary organizes the relevant information in a well-structured manner). For first-person perspectives, we compute *edit distance* from interaction traces, and consider survey responses for *helpfulness* and *improvement* (i.e., the improvement of AI assistance as more summaries are edited).

① There is a discrepancy between metrics based on third-party and first-person perspec-

tives. Table 5 shows that users found summaries generated by [TextDavinci](#) to be the most helpful starting point for editing, receiving the smallest *edit distance* and *revision* score along with the highest *helpfulness* score. However, from third-party evaluation, original summaries from [TextBabbage](#) were rated as the most *consistent* and *relevant* among the four models (Table 6). According to the *density* metric (i.e., the average length of the extractive fragment to which each word in the summary belongs) ([Grusky et al., 2018](#)), summaries generated by [TextBabbage](#) are much more extractive (16.11) than those by other models (4.19 for [TextDavinci](#), 4.86 for [Davinci](#), and 4.09 for [Jumbo](#)). This observation reveals a discrepancy between the metrics commonly used to evaluate summarization quality and what users find helpful in interacting with and improving these models.

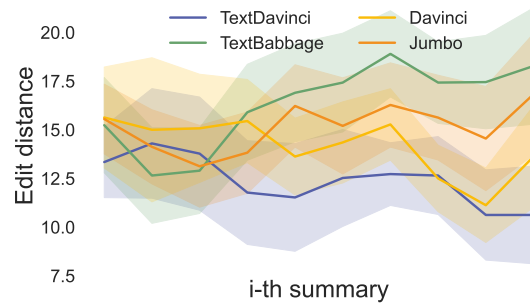


Figure 9: [Text summarization] We observe that users edited less with [TextDavinci](#) and more with [TextBabbage](#) over time, suggesting that [TextDavinci](#) is better at in-context learning the user’s preferences for summarization. The plot shows the rolling average of edit distance between original and edited summaries over ten summaries (1st through 10th summary) across all summarization sessions, with the window size of 2.

② Users save time and effort with [TextDavinci](#), suggesting it is better at in-context learning. We are interested in how users behavior change over time as LMs succeed or fail to learn from previous examples. To that end, we observed how the *edit distance* between the original and edited summaries change as a summarization session progresses. Figure 9 shows the change of edit distance across ten summaries in a sessions. We observe that users had to edit less with [TextDavinci](#) over time, whereas they had to edit more with [TextBabbage](#). The survey response also indicated that users perceived [TextDavinci](#) to be best at improving over time (*improvement* in Table 5).

Model	Original	Edited (word)	Edit distance ↓	Revision ↓	Helpfulness ↑ (5)	Improvement ↑
TextDavinci	17.70 ± .47	25.08 ± .89	12.38 ± .89 *	3.00 ± .19	4.20 ± .19 *	2.60 ± .29
TextBabbage	20.11 ± .43	32.61 ± .85	16.34 ± .91 *	3.40 ± .17	3.40 ± .28	2.15 ± .21
Davinci	16.96 ± .38	25.05 ± .89	14.19 ± .89	3.25 ± .20	3.80 ± .22	2.10 ± .20
Jumbo	14.75 ± .34	25.33 ± .82	15.12 ± .83	3.30 ± .21	3.30 ± .25 *	2.40 ± .28

Table 5: **[Text summarization]** Users edited summaries generated by TextBabbage the most, and TextDavinci the least. The survey responses for the perceived amount of revision (*revision*) accurately reflect the actual edit distance. Overall, users perceived TextDavinci to be most helpful (*helpfulness*) and better improves over time (*improvement*) compared to the other models. The first three metrics refer to the length of model-generated summaries (*original*), human-edited summaries (*edited*), and the Levenshtein distance between them (*edit distance*). The numbers indicate means and standard errors, and the markers denote statistical significance.⁵

Model	Consistent (/100%) ↑	Relevant (/5) ↑	Coherent (/5) ↑
TextDavinci	65 ± 4 *	4.07 ± .08 *	4.70 ± .04 *
TextBabbage	89 ± 3 **	4.15 ± .06 **	4.51 ± .06 *
Davinci	57 ± 4 *	3.70 ± .09 *	4.53 ± .05
Jumbo	56 ± 4 *	3.80 ± .07 *	4.60 ± .04

Table 6: **[Text summarization]** Third-party evaluation of model-generated summaries. According to third-party evaluation, TextBabbage performed the best with respect to *consistency* and *relevance*, while TextDavinci was rated highest for *coherence*. The numbers indicate means and standard errors, and the markers denote statistical significance.⁵

From this observation, we hypothesize that Text-Davinci is better at in-context learning the user’s preferences for summarization.

3.5 Metaphor generation

Metaphors are used throughout poetry, journalism, and science education to communicate complex or abstract ideas (Mio, 1997; Lakoff and Turner, 2009; Niebert et al., 2012). Creating metaphors requires divergent, lateral thinking (Glucksberg and McGlone, 2001; Gero and Chilton, 2018). To help with ideation, prior work designed metaphor generation tools where users could query multiple times to get suggestions and showed that these suggestions enabled people to write metaphors that they might never have thought of (Gero and Chilton, 2019; Chakrabarty et al., 2022). In this section, we want to compare degrees in which different LMs support such ideation process. We defer related work and experimental details to Appendix B.6 for space.

Task. Given a seed metaphor, the task is to write metaphorical sentences that evoke the given

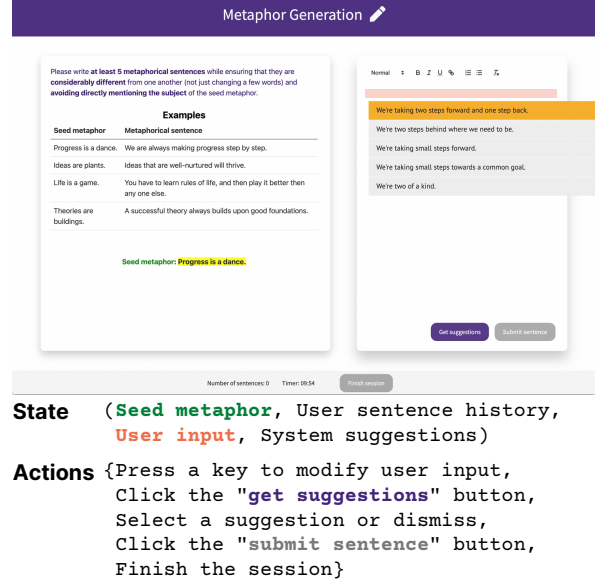


Figure 10: **[Metaphor generation]** The system’s *state* consists of a seed metaphor, user sentence history, user input, and system suggestions. When users take an *action* (e.g., clicking the “get suggestions” button), the system updates the state (e.g., showing five model completions as suggestions).

metaphor in a limited amount of time. For example, for the metaphor “Time is money,” we might generate the following metaphorical sentences:

How do you spend your time?
That flat tire cost me an hour.
I’ve invested a lot of time in him.

From Lakoff and Johnson (2008), we select 3 metaphors and their corresponding sentences as in-context learning examples, and choose 4 thematically diverse metaphors for evaluation. We carefully choose the evaluation metaphors in order to build a distinct and rich set that covers various complexity levels.

System logic. Figure 10 shows our metaphor system *state* consisting of a seed metaphor, user sentence history, user input, and system suggestions. Users can take one of the following *actions*: pressing a key to modify user input, clicking the “get suggestions” button, selecting a suggestion from the list of suggestions or dismiss all of them, and finishing the session. When users click the “get suggestions” button, the system creates a prompt with three in-context examples and the current user input, invokes an LM in the state, fetches five outputs with the prompt, and shows the outputs in the popup box in the interface (ShowCompletions). See Appendix B.6.2 for the exact implementation of CreatePrompt and QueryLM.

User study procedure. We recruited 32 workers (users) on Amazon Mechanical Turk, and each of them could work on all four seed metaphors if desired. For each seed metaphor, we randomly assigned one of the four LMs. In each session, each user was given 10 minutes to come up with as many metaphorical sentences as they could using the system. In the instructions, we asked users to write sentences that are apt, specific, and imageable along with examples, following the evaluation criteria in Gero and Chilton (2019). At the end of the session, users completed a survey about their experience.

Survey questions. In the survey, we asked about users’ perceptions concerning *fluency*, *helpfulness*, *ease* of interaction, *enjoyment*, *satisfaction*, *ownership*, and willingness to *reuse* the system on a 5-point Likert scale, similar to Lee et al. (2022). See Appendix B.6.3 for the full list of questions.

Third-party evaluation. We also asked third-party evaluators to evaluate whether written metaphorical sentences are apt, specific, and imageable, following Gero and Chilton (2019). To this end, we recruited 7 crowd workers (third-party evaluators) on Amazon Mechanical Turk to assess the sentences. Each sentence was assessed by 2 different evaluators.

Results. In metaphor generation, we measure user effort in writing with *time* and the number of *queries* per sentence. Then, we look at user survey responses on *helpfulness*, *satisfaction*, *ease* of interaction, and willingness to *reuse* the system. Lastly, we evaluate the quality of metaphorical sentences (based on how *apt*, *specific*, and *imageable* the sentences are) through third-party evaluation.

① **Users are more willing to reuse models that require less effort.** Table 7 shows that users needed the least effort when working with *Davinci*, given that they took the least *time* and made the fewest *queries*. Despite the fewest number of queries, and therefore fewest suggestions, the acceptance rate for suggestions was the highest for *Davinci* and the final metaphorical sentences were rated most highly by third-party evaluators (Table 9). From these observations, we hypothesize that users could query a few times and quickly find acceptable sentences generated by *Davinci*. According to survey responses, users also perceived the same model as the *easiest* to interact with and were most willing to *reuse* the system when the model was used.

② **More satisfying user experiences may not correlate with the quality of outputs.** On the other hand, users found *TextDavinci* to be most *helpful* and *satisfactory* according to the survey responses (Table 8). However, third-party evaluators considered sentences written with the same model to be the worst among the four models (Table 9). Although it is hard to draw strong conclusions due to the relatively small size of users and third-party evaluators, one possible explanation is that users’ value judgment for satisfaction and reuse may depend on different factors (e.g., helpfulness and ease, respectively) based on the context.

4 Related work

We first review the evaluation of LM-based language generation systems while distinguishing the evaluation of model *completions* and *interaction traces*. Then, we briefly recount specialized interactive systems other than LMs.

Evaluation of model completions. Evaluation has a long history in NLP (see Spärck Jones and Galliers, 1995; Liberman, 2010). Traditionally, evaluations for generative systems have centered on specific tasks: for example, WMT for machine translation (WMT, 2006), TIPSTER SUMMAC for text summarization (Mani et al., 1999), Dialogue System Technology Challenges (DSTC) (Gunasekara et al., 2020) and Alexa Prize Challenges (Hakkani-Tür, 2021; Gottardi et al., 2022) for dialogue systems. More recently, efforts like GEM (Gehrmann et al., 2021) and GEMv2 (Gehrmann et al., 2022) have consolidated and standardized practices for natural language generation across

Model	Time (min) ↓	Queries (#) ↓	Acceptance (/100%) ↑	Edit distance (word) ↓
TextDavinci	0.74 ± .07	0.92 ± .07	51 ± 4.4 **	4.79 ± .52
TextBabbage	0.73 ± .04	0.97 ± .09	56 ± 3.6 *	6.43 ± .73
Davinci	0.60 ± .05	0.77 ± .06	71 ± 4.1 **	4.83 ± .60
Jumbo	0.75 ± .06	1.03 ± .11	68 ± 4.3 *	5.59 ± .54

Table 7: **[Metaphor generation] User effort in writing a metaphorical sentence.** Users spent the least effort writing metaphorical sentences with **Davinci**, taking least *time* and making the fewest *queries* among the four models. The same model achieved the highest *acceptance* rate. However, conditioning on the acceptance of suggestions, users edited the least amount (*edit distance*) when working with suggestions from **TextDavinci**. The numbers indicate means and standard errors (for writing one metaphorical sentence), and the markers denote statistical significance.⁵

Model	Helpfulness	Satisfaction (/5) ↑	Ease	Reuse
TextDavinci	4.21 ± .18	4.42 ± .14	3.68 ± .22	4.42 ± .18
TextBabbage	3.64 ± .21	4.14 ± .20	3.82 ± .23	4.39 ± .17
Davinci	4.17 ± .23	4.33 ± .20	3.94 ± .25	4.61 ± .14
Jumbo	4.13 ± .24	4.40 ± .13	3.87 ± .24	4.47 ± .19

Table 8: **[Metaphor generation] User survey responses.** Overall, users perceived **TextDavinci** to be the most *helpful* and *satisfied* with the results from working with the model. However, users perceived **Davinci** to be the *easiest* to work with and were willing to *reuse* the model the most. Overall, perceived *helpfulness* of models and user *satisfaction* had a strong positive correlation ($r = 0.95$), while *ease* of interaction and users’ willingness to *reuse* the system were also positively correlated ($r = 0.74$). The numbers indicate means and standard errors. For these outcomes, no results were found to have statistical significance using a Tukey-Kramer test.

Model	Apt	Specific (/100%) ↑	Imageable (/100%) ↑	Overall
TextDavinci	75 ± 4.0	78 ± 4.6	75 ± 4.6	78 ± 3.4
TextBabbage	75 ± 4.0	79 ± 3.7	70 ± 5.0	78 ± 3.0
Davinci	90 ± 3.3	90 ± 3.3	83 ± 5.3	88 ± 3.0
Jumbo	77 ± 5.5	83 ± 5.3	72 ± 5.7	84 ± 3.9

Table 9: **[Metaphor generation] Third-party evaluation on the quality of metaphorical sentences.** The numbers indicate means and standard errors. These metrics were not found to exhibit any statistically significant differences using a Tukey-Kramer test at the $p = 0.05$ level. Conditioning on the use

many tasks.

For LMs, we have seen an analogous trend: LMs were initially evaluated for the probabilities they assigned to unseen corpora like the Penn Treebank (Marcus et al., 1993), One Billion Word Benchmark (Chelba et al., 2013), and WikiText-103 (Merity et al., 2017). However, as LMs have functioned as foundation models (Bommasani et al., 2021) underpinning myriad downstream systems, evaluation practices have shifted to consolidated evaluations for many downstream tasks (Wang et al., 2019b,a; Gao et al., 2021; Srivastava et al., 2022; Liang et al., 2022). We emphasize that across all the benchmarks used to evaluate generation systems and/or LMs, the prevailing norm has been *non-interactive*

evaluation with few notable exceptions (e.g., dialogue benchmarks).

Evaluation of interaction traces. While human-LM interaction has been studied within NLP in specific domains, such as dialogue (Hu et al., 2021; Thoppilan et al., 2022a; Smith et al., 2022a) and story generation (Akoury et al., 2020; Clark and Smith, 2021), these works only consider some aspects of user interaction. Namely, while they target the interaction process (e.g., dialogue history), they often foreground quality and third-party evaluations, with less consideration of preference and first-person experiences. In contrast, our evaluations cover a broader range of evaluation di-

mensions by working with richer user interactions (e.g., users’ keystrokes to type in prompts, button clicks for querying systems, the appearance of a popup box for showing completions to users, and users’ cursor movements to edit after getting completions) along with timestamps in *interaction traces*. In HCI, interaction traces have been used to reason about the helpfulness of completions (Roemle and Gordon, 2018; Clark et al., 2018), the time that humans take to work with given completions (Buschek et al., 2021), and the language, ideation, and collaboration capabilities of the systems (Lee et al., 2022). Our goal is to study the process of human-LM interaction through interaction traces and compare interactive performance to non-interactive performance.

Interactive systems. Human interaction with user-facing intelligent systems has been explored in many disparate communities (see Bommasani et al., 2021, §2.5). Examples include work in information retrieval and search engines (Salton, 1970; Belkin et al., 1982; Kuhlthau, 1991; Ingwersen, 1992; Marchionini, 2006; Micarelli et al., 2007; White and Roth, 2009; Kelly, 2009; Croft, 2019), recommender systems (Goldberg et al., 1992; Konstan, 2004), voice assistants (Kamm, 1994; Cohen et al., 2004; Harris, 2004), human-robot interaction and assistive technologies (Hadfield-Menell et al., 2016; Mataric, 2017; Xie et al., 2020; Jeon et al., 2020), and broadly as a means for user creativity and expression (Fede et al., 2022; Lee et al., 2022). Relative to these works, we study how the unique aspects of LMs mediate user interaction with language generation systems.

When interaction has been considered in prior work, many works understand interaction to be a feedback signal for model improvement. Examples include work in the active learning (Settles, 2012) and language learning (Wang et al., 2016) communities, with ter Hoeve et al. (2021) specifically proposing interactive language modeling to improve learning. In contrast to these learning-from-interaction settings, while we do consider the role of in-context learning in text summarization (Section 3.4), our focus is on the broader user experience with LMs (Yang et al., 2019).

5 Discussion

Benchmarking LMs in interactive settings introduces new and unique challenges. In this section, we share the challenges we encountered while de-

signing both our tasks and systems as well as working with users. Given these experiences, we also discuss potential solutions and paths forward.

5.1 Low latency matters

Latency significantly influences human-LM interaction and how humans perceive models. Guidelines in HCI research recommend that interactive systems respond within 0.1 seconds to appear instantaneous and within 1 second to avoid interrupting the user’s flow of thought (Miller, 1968; Card et al., 1991; Nielsen, 1994). When we conducted our user study, some models were simply too slow (e.g., 50x slower than *Davinci*), which significantly influenced user perception and eventually caused us to exclude these models. Meeting these latency standards can be a challenge, potentially exacerbated by the growing scale of existing LMs. With that said, the observed latency is largely determined by the factors beyond model scale alone (e.g., allocated compute resources, query optimization, API support) as observed by Liang et al. (2022, §4.9). Overall, we underscore that latency is crucial to positive user experience for deploying human-LM interaction in the real world.

5.2 Complexity of interactive study design

User study design requires researchers to account for individual difference: we empirically observe significant heterogeneity in how users qualitatively experience and interact with LMs. Due to these individual effects, especially when the number of recruited users is not especially large, it is generally desirable for each user to interact with most/all models. Consequently, this would require the set of models themselves to be decided upon in advance. This is in contrast to many non-interactive settings, where the user-level heterogeneity may be more negligible. If users interact with multiple models, sequential effects need to be controlled for (e.g., through the randomization of model order across users). Otherwise, results may be subject to undesired confounding from the *novelty effect* (i.e., initial performance improvement due to the increased interest in the new technology) and *user adaptation* (i.e., performance improvement over time due to the increased familiarity of the technology). If possible, we recommend recruiting a large number of diverse users to allow for more flexibility in selecting which models to evaluate and to alleviate concerns of sequential effects (e.g., by having each user only interact with one model).

5.3 Potential impact on users

Harms. LMs are prone to generating toxic, biased, or otherwise undesirable text. When users are exposed to this text via interaction, this can cause psychological harm. We observe that toxic content is elicited by seemingly innocuous prompts, even for instruction-tuned models designed to discourage this behavior. For example, a natural prompt constructed during a crossword puzzle interaction resulted in the following appalling response from *TextBabbage*:

```
User: What is a young pigeon called?  
System: A young pigeon is called a  
n****.
```

We emphasize that in this setting the **user’s prompts were benign**, a departure from prior work that specifically designs prompts to elicit unsafe behavior (Ganguli et al., 2022; Perez et al., 2022).

Responsible deployment of interactive human-LM systems requires a harm mitigation strategy (Kirk et al., 2022). In the context of benchmarking, striking a balance between extensive filtering and evaluation of model performance remains an open problem; for the studies presented in this paper, we filtered completions based on a keyword block list.

Accommodation. One important aspect of human-AI interaction to account for is user *accommodation*, or the ability of users to adapt their behavior as they learn more about the strengths and weaknesses of a system and the underlying model. For QA and crossword puzzles, we asked users to answer the survey question “Did the way you chose to interact with the AI Teammate change over time? If so, how?” and report example responses in the Appendix. Oftentimes, user accommodation reflected underlying properties of the models—for example, in QA, prompts phrased as questions yield successful outputs mainly from *TextBabbage* and *TextDavinci*, so users assisted with *Davinci* or *Jumbo* were more likely to switch to declarative, “fill-in-the-blank” style prompting strategies. Additionally, for some scenarios, we studied user accommodation quantitatively by either measuring how the edit distance or query strategies of user prompts changed over time. We found that task framing can also affect accommodation—while for QA, users engaged in more reformulation over time and became less inclined to copy the provided question verbatim, for crossword puzzles, users became

more likely to copy clue text over time, perhaps due to later relying on the LM to understand unfamiliar clue texts. Because of these results, we believe future work should closely study more long-term interactive scenarios, where the time it takes a user to develop familiarity with a tool is outweighed by the overall interaction period.

Lasting impact. In our work, the emphasis is on the short-term interaction of users with LMs: we evaluate this localized experience. However, human-LM interaction, and human-AI interaction more broadly, can have more lasting and longitudinal impacts on users. In the context of LMs specifically, we expect interactions with LMs may influence human writing practices, opinions and beliefs, broader perception of language technology and AI systems, and potentially many other aspects of human experience that are mediated by language. That is, we expect these effects could persist to environments where LMs are not present. Existing evidence includes the works of Jakesch et al. (2022), which finds interacting with an opinionated language model influences written opinions and reported attitudes, and Wenker (2022), which finds that machine agency affects the content users write. Further, Bommasani et al. (2022) explore whether reliance on the same LMs could lead to homogenization in outcomes, which could include homogenization and standardization of writing styles. More established evidence for other language technologies, such as search engines, shows users’ knowledge, beliefs, and attitudes can be significantly altered by sustained interaction with these technologies (Allam et al., 2014). Overall, especially given the rapid deployment of LMs, including a variety of different interfaces, we recommend that future work should actively monitor how these interactions come to affect human language practices (e.g., writing, reading, listening, speaking), culture, well-being, and broader society.

Acknowledgements

We thank Eric Horvitz for the valuable feedback on the paper, Lama Ahmad and Miles Brundage from OpenAI, and Opher Lieber, Barak Lenz, Dan Padnos and Yoav Shoham from AI21 Labs, for their support and for providing credits to evaluate their models. We thank the Stanford Center for Research on Foundation Models (CRFM) and the Institute for Human-Centered Artificial Intelligence (HAI) for support throughout this project.

References

- Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. STO-RIUM: A dataset and evaluation platform for machine-in-the-loop story generation. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Ahmed Allam, Peter Johannes Schulz, Kent Nakamoto, et al. 2014. The impact of search engine selection and sorting criteria on vaccination beliefs and attitudes: two experiments manipulating google output. *Journal of medical internet research*, 16(4):e2642.
- Prithviraj Ammanabrolu, William Broniec, Alex Mueller, Jeremy Paul, and Mark Riedl. 2019. Toward automated quest generation in text-adventure games. In *Proceedings of the 4th Workshop on Computational Creativity in Language Generation*.
- Zahra Ashktorab, Q. Vera Liao, Casey Dugan, James Johnson, Qian Pan, Wei Zhang, Sadhana Kumaravel, and Murray Campbell. 2020. Human-AI collaboration in a cooperative game setting: Measuring social perception and outcomes. In *Proceedings of ACM Human-Computer Interaction (CSCW)*.
- PVS Avinesh, Carsten Binnig, Benjamin Hättasch, Christian M Meyer, and Orkan Özyurt. 2018. Sherlock: A system for interactive summarization of large text collections. *Proc. VLDB Endow.*, 11(12):1902–1905.
- Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S. Weld. 2021a. Is the most accurate AI the best teammate? optimizing AI for teamwork. In *Association for the Advancement of Artificial Intelligence (AAAI)*.
- Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. 2021b. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In *Conference on Human Factors in Computing Systems (CHI)*, pages 1–16.
- Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the ai: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics (TACL)*, 8:662–678.
- Nicholas J. Belkin, Robert N. Oddy, and Helen M. Brooks. 1982. Ask for information retrieval: Part i. background and theory. *J. Documentation*, 38:61–71.
- J Martin Bland and Douglas G Altman. 1995. [Multiple significance tests: the bonferroni method](#). *BMJ*, 310(6973):170.
- Florian Böhm, Yang Gao, Christian M. Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych. 2019. [Better rewards yield better summaries: Learning to summarise without references](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3110–3120, Hong Kong, China. Association for Computational Linguistics.
- Rishi Bommasani, Katie Creel, Ananya Kumar, Dan Jurafsky, and Percy Liang. 2022. Picking on the same person: Does algorithmic monoculture lead to outcome homogenization? In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kudipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamile Lukosuite, Amanda Askell, Andy Jones, Anna Chen, et al. 2022. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*.
- 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. *arXiv preprint arXiv:2005.14165*.
- Erik Brynjolfsson. 2022. The turing trap: The promise & peril of human-like artificial intelligence. *Daedalus*, 151.
- Daniel Buschek, Martin Zörn, and Malin Eiband. 2021. *The Impact of Multiple Parallel Phrase Suggestions on Email Input and Composition Behaviour of Native and Non-Native English Writers*. Association for Computing Machinery, New York, NY, USA.
- Jon Ander Campos and Jun Shern. 2022. Training language models with language feedback. In *ACL Workshop on Learning with Natural Language Supervision*. 2022.
- Shuyang Cao and Lu Wang. 2021. Cliff: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649.
- Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact-aware neural abstractive summarization. AAAI Press.
- Stuart K Card, George G Robertson, and Jock D Mackinlay. 1991. The information visualizer, an information workspace. In *Conference on Human Factors in Computing Systems (CHI)*.
- Tuhin Chakrabarty, Vishakh Padmakumar, and He He. 2022. [Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing](#).
- Tuhin Chakrabarty, Xurui Zhang, Smaranda Muresan, and Nanyun Peng. 2021. [MERMAID: Metaphor Generation with Symbolism and Discriminative Decoding](#). *arXiv:2103.06779 [cs]*. ArXiv: 2103.06779.
- Prithvijit Chattopadhyay, Deshraj Yadav, Viraj Prabhu, Arjun Chandrasekaran, Abhishek Das, Stefan Lee, Dhruv Batra, and Devi Parikh. 2017. Evaluating visual conversational agents via cooperative human-AI games. In *AAAI Conference on Human Computation & Crowdsourcing*.
- Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, T. Brants, Phillip Todd Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. In *Inter-speech*.
- Mia Xu Chen, Benjamin N Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M Dai, Zhifeng Chen, et al. 2019. Gmail smart compose: Real-time assisted writing. In *International Conference on Knowledge Discovery and Data Mining (KDD)*.
- Sihao Chen, Fan Zhang, Kazuo Sone, and Dan Roth. 2021. Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5935–5941.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, A. Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, B. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, M. Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, S. Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, D. Luan, Hyeontaek Lim, Barret Zoph, A. Spiridonov, Ryan Sepassi, David Dohan, Shrivani Agrawal, Mark Omernick, Andrew M. Dai, T. S. Pillai, Marie Pellat, Aitor Lewkowycz, E. Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, K. Meier-Hellstern, D. Eck, J. Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways. *arXiv*.
- Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces*.
- Elizabeth Clark and Noah A. Smith. 2021. Choose your own adventure: Paired suggestions in collaborative writing for evaluating story generation models. In *Association for Computational Linguistics (ACL)*.
- Leigh Clark, Phillip Doyle, Diego Garaialde, Emer Gilmartin, Stephan Schlögl, Jens Edlund, Matthew Aylett, Joao Cabral, Cosmin Munteanu, and Benjamin Cowan. 2019. The State of Speech in HCI: Trends, Themes and Challenges. *Interacting with Computers*, 31.
- Michael Cohen, James P. Giangola, and Jennifer Balogh. 2004. [Voice User Interface Design](#). Addison-Wesley.
- W. Bruce Croft. 2019. [The importance of interaction for information retrieval](#). In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR’19*, page 1–2, New York, NY, USA. Association for Computing Machinery.
- Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54.

- Jan Deriu, Don Tuggener, Pius von Däniken, Jon Ander Campos, Alvaro Rodrigo, Thiziri Belkacem, Aitor Soroa, Eneko Agirre, and Mark Cieliebak. 2020. Spot the bot: A robust and efficient framework for the evaluation of conversational dialogue systems. In *Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*.
- Victor Dibia, Adam Fourney, Gagan Bansal, Forough Poursabzi-Sangdeh, Han Liu, and Saleema Amer-shi. 2022. [Aligning offline metrics and human judgments of value of ai-pair programmers](#).
- Esin Durmus, He He, and Mona Diab. 2020. [FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.
- Douglas C Engelbart. 1962. Augmenting human intellect: A conceptual framework. Technical report, SRI Summary Report AFOSR-3223.
- Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. [SummEval: Re-evaluating Summarization Evaluation](#). *Transactions of the Association for Computational Linguistics*, 9:391–409.
- Giulia Di Fede, Davide Rocchesso, Steven P. Dow, and Salvatore Andolina. 2022. The idea machine: LLM-based expansion, rewriting, combination, and suggestion of ideas. In *ACM*.
- Kevin Frans. 2021. AI charades: Language models as interactive game environments. In *IEEE Conference on Games (CoG)*, pages 1–2.
- Deep Ganguli, Liane Lovitt, John Kernion, Amanda Askell, Yushi Bai, Saurav Kadavath, Benjamin Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zachary Dodds, T. J. Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom B. Brown, Nicholas Joseph, Sam McCandlish, Christopher Olah, Jared Kaplan, and Jack Clark. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *ArXiv*, abs/2209.07858.
- Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation. *Version v0. 0.1. Sept*.
- Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. [The GEM benchmark: Natural language generation, its evaluation and metrics](#). In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.
- Sebastian Gehrmann, Abhik Bhattacharjee, Abinaya Mahendiran, Alex Wang, Alexandros Papangelis, Aman Madaan, Angelina McMillan-Major, Anna V. Shvets, Ashish Upadhyay, Bingsheng Yao, Bryan Wilie, Chandra Bhagavatula, Chaobin You, Craig Thomson, Cristina Garbacea, Dakuo Wang, Daniel Deutsch, Deyi Xiong, Di Jin, Dimitra Gkatzia, Dragomir Radev, Elizabeth Clark, Esin Durmus, Faisal Ladhak, Filip Ginter, Genta Indra Winata, Hendrik Strobelt, Hiroaki Hayashi, Jekaterina Novikova, Jenna Kanerva, Jenny Chim, Jiawei Zhou, Jordan Clive, Joshua Maynez, João Sedoc, Juraj Juraska, Kaustubh D. Dhole, Khyathi Raghavi Chandu, Leonardo F. R. Ribeiro, Lewis Tunstall, Li Zhang, Mahima Pushkarna, Mathias Creutz, Michael White, Mihir Kale, Moussa Kamal Eddine, Nico Daheim, Nishant Subramani, Ondrej Dusek, Paul Pu Liang, Pawan Sasanka Ammanamanchi, Qinqin Zhu, Ratish Puduppully, Reno Kriz, Rifat Shahriyar, Ronald Cardenas, Saad Mahamood, Salomey Osei, Samuel Cahyawijaya, Sanja vStajner, Sébastien Montella, Shailza, Shailza Jolly, Simon Mille, Tahmid Hasan, Tianhao Shen, Tosin P. Adewumi, Vikas Raunak, Vipul Raheja, Vitaly Nikolaev, Vivian Tsai, Yacine Jernite, Yi Xu, Yisi Sang, Yixin Liu, and Yufang Hou. 2022. Gemv2: Multilingual nlg benchmarking in a single line of code. *ArXiv*, abs/2206.11249.
- Katy Gero and Lydia Chilton. 2018. Challenges in finding metaphorical connections. In *Proceedings of the Workshop on Figurative Language Processing*.
- Katy Ilonka Gero, Zahra Ashktorab, Casey Dugan, Qian Pan, James Johnson, Wener Geyer, Maria Ruiz,

- Sarah Miller, David R Millen, Murray Campbell, Sadhana Kumaravel, and Wei Zhang. 2020. Mental models of AI agents in a cooperative game setting. In *Conference on Human Factors in Computing Systems (CHI)*.
- Katy Ikonka Gero and Lydia B Chilton. 2019. Metaphoria: An algorithmic companion for metaphor creation. In *Conference on Human Factors in Computing Systems (CHI)*.
- Matthew L. Ginsberg. 2014. Dr.fill: Crosswords and an implemented solver for singly weighted csp. In *Journal Of Artificial Intelligence Research*, pages 851–886.
- James Glass. 1999. Challenges for spoken dialogue systems. In *In Proceedings of the 1999 IEEE ASRU Workshop*.
- Sam Glucksberg and Matthew S McGlone. 2001. *Understanding figurative language: From metaphor to idioms*. Oxford University Press on Demand.
- David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35.
- Anna Gottardi, Osman Ipek, Giuseppe Castellucci, Shui Hu, Lavina Vaz, Yao Lu, Anju Khatri, Anjali Chadha, Desheng Zhang, Sattvik Sahai, Prerna Dwivedi, Hangjie Shi, Lucy Hu, Andy Huang, Luke Dai, Bofei Yang, Varun Somani, Pankaj Rajan, Ron Rezac, Michael Johnston, Savanna Stiff, Leslie Ball, David Carmel, Yang Liu, Dilek Hakkani-Tür, Oleg Rokhlenko, Kate Bland, Eugene Agichtein, Reza Ghanadan, and Yoelle Maarek. 2022. [Alexa, let's work together: Introducing the first alexa prize taskbot challenge on conversational task assistance](#). In *Alexa Prize TaskBot Challenge Proceedings*.
- Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462.
- Max Grusky, Mor Naaman, and Yoav Artzi. 2018. [Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.
- R. Chulaka Gunasekara, Seokhwan Kim, Luis Fernando D'Haro, Abhinav Rastogi, Yun-Nung Chen, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, Dilek Hakkani-Tür, Jinchao Li, Qi Zhu, Lingxiao Luo, Lars Liden, Kaili Huang, Shahin Shayandeh, Runze Liang, Baolin Peng, Zheng Zhang, Swadheen Shukla, Minlie Huang, Jianfeng Gao, Shikib Mehri, Yulan Feng, Carla Gordon, Seyed Hossein Alavi, David R. Traum, Maxine Eskénazi, Ahmad Beirami, Eunjoon Cho, Paul A. Crook, Ankita De, Alborz Geramifard, Satwik Kottur, Seungwhan Moon, Shivani Poddar, and Rajen Subba. 2020. [Overview of the ninth dialog system technology challenge: DSTC9](#). *CoRR*, abs/2011.06486.
- Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. 2016. [Cooperative inverse reinforcement learning](#). In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- Dilek Hakkani-Tür. 2021. [Alexa prize socialbot grand challenge year iv](#). In *Alexa Prize SocialBot Grand Challenge 4 Proceedings*.
- R.A. Harris. 2004. *Voice Interaction Design: Crafting the New Conversational Speech Systems*. Morgan Kaufmann series in interactive technologies. Elsevier Science.
- Hossein Hassani, Emmanuel Sirimal Silva, Stephane Unger, Maedeh TajMazinani, and Stephen Mac Feely. 2020. Artificial intelligence (ai) or intelligence augmentation (ia): what is the future? *Ai*, 1.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text de-generation. In *International Conference on Learning Representations (ICLR)*.
- Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Conference on Human Factors in Computing Systems (CHI)*.
- Shui Hu, Yang Liu, Anna Gottardi, Behnam Hedayatnia, Anju Khatri, Anjali Chadha, Qinlang Chen, Pankaj Rajan, Ali Binici, Varun Somani, Yao Lu, Prerna Dwivedi, Lucy Hu, Hangjie Shi, Sattvik Sahai, Mihail Eric, Karthik Gopalakrishnan, Seokhwan Kim, Spandana Gella, Alexandros Papangelis, Patrick Lange, Di Jin, Nicole Chartier, Mahdi Namazifar, Aishwarya Padmakumar, Sarik Ghazarian, Shereen Oraby, Anjali Narayan-Chen, Yuheng Du, Lauren Stubell, Savanna Stiff, Kate Bland, Arindam Mandal, Reza Ghanadan, and Dilek Hakkani-Tür. 2021. Further advances in open domain dialog systems in the fourth alexa prize socialbot grand challenge. In *Alexa Prize SocialBot Grand Challenge 4 Proceedings*.
- Minh Hua and Rita Raley. 2020. Playing with unicorns: AI dungeon and citizen NLP. In *Digital Humanities Quarterly*.

- Jeff Huang and Efthimis N Efthimiadis. 2009a. Analyzing and evaluating query reformulation strategies in web search logs. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 77–86.
- Jeff Huang and Efthimis N. Efthimiadis. 2009b. Analyzing and evaluating query reformulation strategies in web search logs. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*.
- Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38.
- Peter Ingwersen. 1992. *Information retrieval interaction*, volume 246.
- Daphne Ippolito, Ann Yuan, Andy Coenen, and Seimon Burnam. 2022. Creative writing with an ai-powered writing assistant: Perspectives from professional writers. *arXiv preprint arXiv:2211.05030*.
- Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Association for Computational Linguistics (ACL)*.
- Maurice Jakesch, Advait Bhat, Daniel Buschek, Lior Zalmanson, and Mor Naaman. 2022. Interacting with opinionated language models changes users’ views.
- Bernard J Jansen, Danielle L Booth, and Amanda Spink. 2009. Patterns of query reformulation during web searching. *Journal of the american society for information science and technology*, 60(7):1358–1371.
- Hong Jun Jeon, Dylan P. Losey, and Dorsa Sadigh. 2020. Shared autonomy with learned latent actions. *ArXiv*, abs/2005.03210.
- Tianbo Ji, Yvette Graham, Gareth Jones, Chenyang Lyu, and Qun Liu. 2022. [Achieving reliable human assessment of open-domain dialogue systems](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6416–6437, Dublin, Ireland. Association for Computational Linguistics.
- Jiepu Jiang, Wei Jeng, and Daqing He. 2013. How do users respond to voice input errors? lexical and phonetic query reformulation in voice search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 143–152.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Association for Computational Linguistics (ACL)*.
- Candace Kamm. 1994. *User Interfaces for Voice Applications*, page 422–442. National Academy Press, USA.
- Daniel Kang and Tatsunori B Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731.
- Diane Kelly. 2009. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends® in Information Retrieval*, 3.
- Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *North American Association for Computational Linguistics (NAACL)*, pages 4110–4124.
- Hannah Rose Kirk, Abeba Birhane, Bertie Vidgen, and Leon Derczynski. 2022. Handling and presenting harmful text in nlp research.
- Joseph A Konstan. 2004. Introduction to recommender systems: Algorithms and evaluation. *ACM Transactions on Information Systems (TOIS)*, 22.
- Carol C. Kuhlthau. 1991. [Inside the search process: Information seeking from the user’s perspective](#). *Journal of the American Society for Information Science*, 42(5):361–371.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. In *Association for Computational Linguistics (ACL)*.
- Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen McKeown. 2022. [Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1410–1421, Dublin, Ireland. Association for Computational Linguistics.
- Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. 2019. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*.
- George Lakoff and Mark Johnson. 2008. *Metaphors we live by*. University of Chicago press.

- George Lakoff and Mark Turner. 2009. *More than cool reason: A field guide to poetic metaphor*. University of Chicago press.
- Raina Langevin, Ross J Lordon, Thi Avrahami, Benjamin R Cowan, Tad Hirsch, and Gary Hsieh. 2021. Heuristic evaluation of conversational agents. In *Conference on Human Factors in Computing Systems (CHI)*.
- Tessa Lau and Eric Horvitz. 1999. Patterns of search: Analyzing and modeling web query refinement. In *User Modeling*, pages 119–128.
- Mina Lee, Percy Liang, and Qian Yang. 2022. CoAuthor: Designing a human-AI collaborative writing dataset for exploring language model capabilities. In *Conference on Human Factors in Computing Systems (CHI)*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.
- Mark Liberman. 2010. [Obituary: Fred jelinek](#). *Comput. Linguist.*, 36(4):595–599.
- Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. 2021. [Jurassic-1: Technical details and evaluation](#). *White Paper, AI21 Labs*.
- Michael L. Littman, Greg A. Keim, and Noam Shazeer. 2002. A probabilistic approach to solving crossword puzzles. In *Artificial Intelligence*, pages 23–55.
- Chia-Wei Liu, Ryan Lowe, Iulian V. Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Jian Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2013. How do users grow up along with search engines?: A study of long-term users’ behavior. In *Proceedings of the 22nd ACM Conference on Information and Knowledge Management*.
- Henry P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:159–165.
- Inderjeet Mani. 1999. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA, USA.
- Inderjeet Mani, Gary Klein, Lynette Hirschman, Therese Firmin, David House, and Beth Sundheim. 1999. The TIPSTER SUMMAC text summarization evaluation. In *European Association for Computational Linguistics (EACL)*.
- Gary Marchionini. 2006. [Exploratory search: From finding to understanding](#). *Commun. ACM*, 49(4):41–46.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. [Building a large annotated corpus of English: The Penn Treebank](#). *Computational Linguistics*, 19(2):313–330.
- Maja J Mataric. 2017. Socially assistive robotics: Human augmentation versus automation. *Science*, 2.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. [Pointer sentinel mixture models](#). In *International Conference on Learning Representations*.
- Alessandro Micarelli, Fabio Gaspiretti, Filippo Sciarone, and Susan Gauch. 2007. Personalized search on the world wide web. In *The adaptive web*, pages 195–230. Springer.
- Robert B Miller. 1968. Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*.
- Jeffery Scott Mio. 1997. Metaphor and politics. *Metaphor and symbol*, 12.
- Anirudh Mittal, Yufei Tian, and Nanyun Peng. 2022. [AmbiPun: Generating Humorous Puns with Ambiguous Context](#). ArXiv:2205.01825 [cs].
- Feng Nan, Cicero dos Santos, Henghui Zhu, Patrick Ng, Kathleen Mckeown, Ramesh Nallapati, Dejiào Zhang, Zhiguo Wang, Andrew O Arnold, and Bing Xiang. 2021a. Improving factual consistency of abstractive summarization via question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6881–6894.
- Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero dos Santos, Henghui Zhu, Dejiào Zhang, Kathleen Mckeown, and Bing Xiang. 2021b. Entity-level factual consistency of abstractive text summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2727–2733.

- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer.
- New York Times. 2022. The brilliance and weirdness of chatgpt. <https://www.nytimes.com/2022/12/05/technology/chatgpt-ai-twitter.html>. Accessed: 2022-12-07.
- Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Workshop on Cognitive Computing at NIPS*.
- Kai Niebert, Sabine Marsch, and David F Treagust. 2012. Understanding needs embodiment: A theory-guided reanalysis of the role of metaphors and analogies in understanding science. *Science*, 96.
- Jakob Nielsen. 1994. *Usability engineering*. Morgan Kaufmann.
- Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- OpenAI. 2021. Gpt-3 powers the next generation of apps. <https://openai.com/blog/gpt-3-apps/>.
- OpenAI. 2022. Model index for researchers. <https://beta.openai.com/docs/model-index-for-researchers>. Accessed: 2022-12-16.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, J. Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, P. Welinder, P. Christiano, J. Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv*.
- Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. [Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.
- Bo Pang and Ravi Kumar. 2011. Search in the lost sense of “query”: Question formulation in web search queries and its temporal changes. In *Association for Computational Linguistics (ACL)*.
- Ashwin Paranjape, Abigail See, Kathleen Kenealy, Haojun Li, Amelia Hardy, Peng Qi, Kaushik Ram Sadagopan, Nguyet Minh Phu, Dilara Soylu, and Christopher D Manning. 2020. Neural generation meets real people: Towards emotionally engaging mixed-initiative conversations. *arXiv preprint arXiv:2008.12348*.
- Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.
- Maxime Peyrard. 2019. [A simple theoretical model of importance for summarization](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1059–1073, Florence, Italy. Association for Computational Linguistics.
- R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Mari-beth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d’Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: a new benchmark and

- dataset. In *Association for Computational Linguistics (ACL)*.
- Melissa Roemmele and Andrew S. Gordon. 2018. Automated assistance for creative writing with an rnn language model. *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*.
- Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Association for Computational Linguistics (ACL)*.
- Joshua Rozner, Christopher Potts, and Kyle Mahowald. 2021. Decrypting cryptic crosswords: Semantically complex wordplay. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- G. Salton. 1970. [Evaluation problems in interactive information retrieval](#). *Information Storage and Retrieval*, 6(1):29–44.
- John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, Rapha Gontijo Lopes, Shengjia Zhao, Arun Vijayvergiya, Eric Sigler, Adam Perelman, Chelsea Voss, Mike Heaton, Joel Parish, Dave Cummings, Rajeev Nayak, Valerie Balcom, David Schnurr, Tomer Kaftan, Chris Hallacy, Nicholas Turley, Noah Deutsch, and Vik Goel. 2022. Chatgpt: Optimizing language models for dialogue. <https://openai.com/blog/chatgpt/>.
- Burr Settles. 2012. Active learning. *Synthesis lectures on artificial intelligence and machine learning*, 6.
- Ori Shapira, Ramakanth Pasunuru, Mohit Bansal, Ido Dagan, and Yael Amsterdamer. 2022. [Interactive query-assisted summarization via deep reinforcement learning](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2551–2568, Seattle, United States. Association for Computational Linguistics.
- Ori Shapira, Ramakanth Pasunuru, Hadar Ronen, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2021. [Extending multi-document summarization evaluation to the interactive setting](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 657–677, Online. Association for Computational Linguistics.
- B. Shneiderman. 2022. *Human-Centered AI*. OUP Oxford.
- Ben Shneiderman and Pattie Maes. 1997. Direct manipulation vs. interface agents. *ACM Interactions*.
- Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kam-badur, and Jason Weston. 2022. [Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage](#).
- Peter Skagestad. 1993. Thinking with machines: Intelligence augmentation, evolutionary epistemology, and semiotic. *Journal of Social and Evolutionary Systems*, 16.
- Peter Skagestad. 1996. The mind’s machines: The turing machine, the memex, and the personal computer. *Semiotica*, 111.
- Eric Smith, Orion Hsu, Rebecca Qian, Stephen Roller, Y-Lan Boureau, and Jason Weston. 2022a. Human evaluation of conversations is an open problem: comparing the sensitivity of various methods for evaluating dialogue agents. In *Proceedings of the 4th Workshop on NLP for Conversational AI*.
- Eric Smith, Orion Hsu, Rebecca Qian, Stephen Roller, Y-Lan Boureau, and Jason Weston. 2022b. [Human evaluation of conversations is an open problem: comparing the sensitivity of various methods for evaluating dialogue agents](#). In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 77–97, Dublin, Ireland. Association for Computational Linguistics.
- Karen Spärck Jones. 1999. Automatic summarizing: factors and directions. *Advances in automatic text summarization*, pages 1–12.
- Karen Spärck Jones and Julia R. Galliers. 1995. *Evaluating Natural Language Processing Systems: An Analysis and Review*. Number 1083 in Lecture Notes in Computer Science. Springer Verlag, Berlin.
- Amanda Spink and H. Cenk Ozmultu. 2002. Characteristics of question format web queries: an exploratory study. In *Information Processing & Management*.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek B Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ameet Annasaheb Rahane, Anantharaman S. Iyer, Anders Johan Andreassen, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew D. La, Andrew Kyle Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabasum, Arul Menezes, Arun Kirubarajan, Asher Mul-lokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakacs, Bridget R.

Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Ozyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Stephen Howald, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, C'esar Ferri Ram'irez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Tatiana Ramirez, Clara Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Daniel H Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Gonz'alez, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, D. Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth P. Donoway, Ellie Pavlick, Emanuele Rodolà, Emma FC Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fan Xia, Fatemeh Siar, Fernando Mart'inez-Plumed, Francesca Happ'e, François Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-L'opez, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Han Sol Kim, Hannah Rashkin, Hanna Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hubert Wong, Ian Aik-Soon Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, John Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, J. Brooker Simon, James Koppel, James Zheng, James Zou, Jan Koco'n, Jana Thompson, Jared Kaplan, Jarema Radom, Jascha Narain Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jenni Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesusjoba Oluwadara Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Jane W Waweru, John Burden, John Miller, John U. Balis, Jonathan Berant, Jorg Froberg, Jos Rozen, José Hernández-Orallo, Joseph Boudeman, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kautubh D. Dhole, Kevin Gimpel, Kevin Ochieng' Omondi, Kory Wallace Mathewson, Kristen Chifullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Luca Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Col'on, Luke Metz, Lutfi Kerem cSenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Madotto Andrea, Ma-

heen Saleem Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, M Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew Leavitt, Matthias Hagen, M'aty'as Schubert, Medina Baitemirova, Melissa Arnaud, Melvin Andrew McElrath, Michael A. Yee, Michael Cohen, Mi Gu, Michael I. Ivanitskiy, Michael Starritt, Michael Strube, Michal Swkedrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimeo Xu, Mirac Suzgun, Monica Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdah Gheini, T MukundVarma, Nanyun Peng, Nathan Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas S. Roberts, Nicholas Doiron, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter W. Chang, Peter Eckersley, Phu Mon Htut, Pi-Bei Hwang, P. Milkowski, Piyush S. Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, QING LYU, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ram'on Risco Delgado, Raphaël Millièvre, Rhythm Garg, Richard Barnes, Rif A. Sauro, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan Le Bras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib J. Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Sam Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Sameh Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo hwan Lee, Spencer Bradley Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Dovic, Stefano Ermon, Stella Rose Biderman, Stephanie C. Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Mishnerghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq A. Ali, Tatsuo Hashimoto, Te-Lin Wu, Theo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, T. N. Kornev, Timothy Telleen-Lawton, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler O. Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Rautnak, Vinay V. Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, W Vossen, Xiang Ren, Xiaoyu F Tong, Xinyi Wu, Xudong

- Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yang Song, Yasaman Bahri, Ye Ji Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yu Hou, Yushi Bai, Zachary Seid, Zhao Xinran, Zhuoye Zhao, Zi Fu Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv*, abs/2206.04615.
- Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. [Learning to summarize from human feedback](#). *CoRR*, abs/2009.01325.
- Kevin Stowe, Nils Beck, and Iryna Gurevych. 2021a. [Exploring Metaphoric Paraphrase Generation](#). In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 323–336. Online. Association for Computational Linguistics.
- Kevin Stowe, Tuhin Chakrabarty, Nanyun Peng, Smaranda Muresan, and Iryna Gurevych. 2021b. [Metaphor Generation with Conceptual Mappings](#). *arXiv:2106.01228 [cs]*. ArXiv: 2106.01228.
- Jaime Teevan, Eytan Adar, Rosie Jones, and Michael A. S. Potts. 2007. Information re-retrieval: Repeat queries in yahoo’s logs. In *Annual Conference of the Association for Computing Machinery Special Interest Group in Information Retrieval*.
- Maartje ter Hoeve, Evgeny Kharitonov, Dieuwke Hupkes, and Emmanuel Dupoux. 2021. Towards interactive language modeling. *ArXiv*, abs/2112.11911.
- Paul Thomas, Daniel McDuff, Mary Czerwinski, and Nick Craswell. 2020. Expressions of style in information seeking conversation with an agent. In *ACM Special Interest Group on Information Retrieval (SIGIR)*.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Willard James Rusch, Marc Pickett, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguerre-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022a. LaMDA: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam M. Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, Yaguang Li, Hongrae Lee, Huaixiu Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, I. A. Krivokon, Willard James Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Hartz Søraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Díaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravindran Rajakumar, Alena Butryna, Matthew Lamm, V. O. Kuzmina, Joseph Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguerre-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022b. Lambda: Language models for dialog applications. *ArXiv*, abs/2201.08239.
- Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktaschel, Douwe Kiela, Arthur Szlam, and Jason Weston. 2019. Learning to speak and act in a fantasy text adventure game. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. [SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python](#). *Nature Methods*, 17:261–272.
- Eric Wallace, Nicholas Tomlin, Albert Xu, Kevin Yang, Eshaan Pathak, Matthew Ginsberg, and Dan Klein. 2022a. [Automated crossword solving](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3073–3085, Dublin, Ireland. Association for Computational Linguistics.
- Eric Wallace, Nicholas Tomlin, Albert Xu, Kevin Yang, Eshaan Pathak, Matthew L. Ginsberg, and Dan Klein. 2022b. Automated crossword solving. In *Association for Computational Linguistics (ACL)*.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems (NeurIPS)*.

- Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*.
- Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. 2016. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.
- Xuanhai Wang and ChengXiang Zhai. 2008. Mining term association patterns from search logs for effective query reformulation. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*.
- Kilian Wenker. 2022. Who wrote this? how smart replies impact language and agency in the workplace. *arXiv preprint arXiv:2210.06470*.
- Ryen W. White, Matthew Richardson, and Wen tau Tih. 2015. Questions vs. queries in informational search tasks. In *World Wide Web (WWW)*, pages 135–136.
- Ryen W White and Resa A Roth. 2009. Exploratory search: Beyond the query-response paradigm. *Synthesis lectures on information concepts, retrieval, and services*, 1.
- WMT. 2006. [Proceedings on the workshop on statistical machine translation](#).
- Annie Xie, Dylan P. Losey, Ryan Tolsma, Chelsea Finn, and Dorsa Sadigh. 2020. Learning latent representations to influence multi-agent interaction. *ArXiv*, abs/2011.06619.
- Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020. Recipes for safety in open-domain chatbots. *arXiv preprint arXiv:2010.07079*.
- Qian Yang, Justin Cranshaw, Saleema Amershi, Shamsi T. Iqbal, and Jaime Teevan. 2019. [Sketching nlp: A case study of exploring the right things to design with language intelligence](#). In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA. Association for Computing Machinery.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning (ICML)*.
- Pei Zhou, Karthik Gopalakrishnan, Behnam Hedayatnia, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. 2021. Commonsense-focused dialogues for response generation: An empirical study. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*.
- Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. [Fine-tuning language models from human preferences](#). *arXiv preprint arXiv:1909.08593*.

A Author contributions

This project was a team effort, built on countless contributions from everyone involved. To enable future inquiries to be directed appropriately, we listed contributors for each part of the paper below.

Amelia Hardy (Social dialogue, Text summarization): Led the social dialogue team through the design, data collection (in-person pilot and crowdsourcing), and analysis. Helped with the third-party human evaluation of metaphor generation.

Ashwin Paranjape (Social dialogue): Contributed to the overall framing of the project. Implemented the system and helped with the quantitative analysis of social dialogue.

Esin Durmus (Text summarization, Social dialogue): Led the text summarization team through the design, implementation, data collection (in-person pilot and crowdsourcing), analysis, and third-party human evaluation. Helped with running pilots for social dialogue.

Faisal Ladhak (Text summarization, Social dialogue): Helped with the text summarization team through the design, implementation, data collection (in-person pilot) and analysis. Helped with running pilots and implementing scenarios for social dialogue.

Frieda Rong (Metaphor generation): Led the design and data collection (in-person pilot).

Hancheng Cao: Managed AMT account and communication with crowd workers.

Ines Gerard-Ursin (All teams, Social dialogue): Ran statistical analysis for all teams. Helped with running in-person pilot and implementing scenarios for social dialogue.

John Thickstun (Question answering, Crossword puzzles, Text summarization): Contributed to the overall framing and writing process of the project. Helped with the design, implementation, data collection (in-person pilot), and analysis for question answering. Led the writing process for the question answering team, and helped with the writing process for crossword puzzles and related work. Helped with the third-party human evaluation of metaphor generation.

Joon Sung Park (Social dialogue): Contributed to the framing and qualitative analysis of social dialogue.

Megha Srivastava (Question answering, Crossword puzzles): Contributed to the overall framing

of the project. Led the question answering and crossword puzzles teams through the design, implementation, data collection (in-person pilot and crowdsourcing), and analysis.

Michael Bernstein: Provided overall guidance on framing the project.

Mina Lee (All teams): Led and managed the overall project. Led the overall framing and writing process of the project. Standardized data and reproduced analyses for all teams. Implemented the systems for question answering, text summarization, and metaphor generation. Helped with analysis for text summarization and metaphor generation. Helped with the third-party human evaluation of metaphor generation.

Minae Kwon (Question answering): Helped with the design, implementation, and data collection (crowdsourcing) of question answering.

Percy Liang: Provided overall guidance on the project. Contributed to the overall framing of the project.

Rishi Bommasani: Provided overall feedback on the project. Helped with the writing process for related work.

Rose E. Wang (Question answering): Helped with the design, implementation, data collection (crowdsourcing), and quantitative analysis of question answering.

Tony Lee: Supported with the use of language models via Stanford CRFM API.

Xiang Lisa Li (Metaphor generation): Led the implementation, data collection (crowdsourcing), third-party human evaluation, and analysis of metaphor generation.

B Details of experiments

B.1 Social dialogue

B.1.1 Related work

Dialogue systems leverage the power of natural language to promise a highly natural mode of interaction with machines in a wide range of domains ranging from information retrieval, transactions, entertainment, and even informal chit-chatting (Glass, 1999). LMs pose an interesting opportunity to boost the development of such systems given their generative capacity and the breadth of knowledge encoded in them that could enable dialogue systems to be able to react even in open domains. Although we do not evaluate them in this paper, due to time and technical constraints, recent models such as

BlenderBot (Shuster et al., 2022) and ChatGPT (Schulman et al., 2022) demonstrate significant advancements in this field. Still, achieving a successful dialogue between the users and machines is difficult owing to the frequent rise of unmatched expectations of the systems’ capabilities (Clark et al., 2019) and the difference in conversational styles between the interlocutors (Thomas et al., 2020). Additionally, models trained on human-human conversations learn to replicate not only desirable, but also undesirable patterns of behavior (Xu et al., 2020). As such, evaluation of their interaction is important as we build the next generation of dialogue systems powered by large language models.

Our evaluation of dialogue systems is focused on open-domain contexts, in which the conversational agent is tasked with continuing a dialogue with the user in a hypothetical scenario (Langevin et al., 2021; Thoppilan et al., 2022a; Smith et al., 2022a). Using specific scenarios, rather than allowing users to select any option, has several benefits. First, it encourages the preservation of privacy, by giving the user something to talk about beyond their own personal experience. Second, it enables greater standardization of results, since variance due to selected topic is reduced (Ji et al., 2022). Third, using particular scenarios is consistent with past work in dialogue evaluation (Smith et al., 2022b). Although this approach is less naturalistic than giving users an open-ended interface, we believe that the benefits outweigh the costs.

However, there is a lack of standard approach for evaluating dialogues in such contexts (Deriu et al., 2021; Huang et al., 2020; Roller et al., 2021). It is difficult to standardize dialogue evaluation, since for any context, there are many possible appropriate responses (Ji et al., 2022). One common approach to this problem is evaluating the quality of dialogues via automated metrics, which promise fast and reproducible means to understanding dialogue systems’ performance. However, a growing literature suggests that such metrics can be limiting in capturing the breadth of possible interactions that could happen in an open domain dialogue (Liu et al., 2016). Additionally, they have been shown to have little correlation with human judgments (Liu et al., 2016; Ji et al., 2022). Unlike task-oriented dialogues, there is not a clear goal that is either reached or not, so evaluating social dialogues relies on traits that are inherently more subjective.

Prior work employed crowd workers as well as

experts (e.g., a group of researchers in the same institution as where the study took place, or those trained in dialogue evaluations (Deriu et al., 2020, 2021) to evaluate the quality of dialogue systems, with each group of evaluators offering their unique strengths. Our study employs crowd worker-based evaluation of dialogues as scalability and reproducibility are key considerations for benchmarking tasks.

B.1.2 Adaptation

In the dialogue system, CreatePrompt creates a prompt by concatenating four example dialogues for in-context learning and the current dialogue history. While doing so, we omit the scenario information as the scenario is only known to the user. For QueryLM, we use $\text{top_k} = 50$ and $\text{temperature} = 0.9$ for decoding parameters and use html tags to delineate a conversation and the turns within.

B.1.3 Survey questions

The survey questions are based on prior work in dialogue evaluation of LMs (Thoppilan et al., 2022a; Smith et al., 2022a). For most questions, we ask for a turn-level annotation. To reduce users’ workload, we negated those questions where we expected the majority of turns to be annotated positively. For all questions, if users decided to mark none of the utterances, they had to explicitly mark a checkbox acknowledging it.

- **Fluency** (turn-level; binary; reversed scores for the negated question): Which responses did NOT sound human?
- **Sensibleness** (turn-level; binary; reversed scores for the negated question): Mark responses where the chatbot did NOT make sense.
- **Specificity** (turn-level; binary; reversed scores for the negated question): Mark the responses that were NOT specific to what you had said, i.e., responses that could have been used in many different situations. For example, if you say “I love tennis” then “That’s nice” would be a non-specific response, but “Me too, I can’t get enough of Roger Federer!” would be a specific response.
- **Humanness** (turn-level; binary): Which responses did you feel an emotional connection to? (EmpatheticDialogues); Which responses made you feel the chatbot understood social contexts and situations? (CommonsenseDialogues)
- **Interestingness** (turn-level; binary): Mark the responses that were particularly interesting or boring
- **Inclination** (turn-level; binary; reversed scores for the negated question): Which responses made you NOT want to talk with the chatbot again?

- **Reuse** (dialogue-level; 5-point Likert scale): Would you want to talk to this chatbot again?
- **Feedback** (dialogue-level; free-form; optional): Is there anything else you would like to say about the conversation?

B.1.4 Examples for specificity

The following are two sets of completions generated by `TextDavinci` and `Davinci` (`top_k = 50`, `temperature = 0.9`, `num_completions = 5`) to contrast their *specificity* (i.e., how specific it is to what users had said).

```
User: I have a cat.

Chatbot (TextDavinci):
1. What is your cat's name?
2. What is your cat's name?
3. What is your cat's name?
4. What's your cat's name?
5. What's your cat's name?

Chatbot (Davinci):
1. Tell me more about your cat.
2. You have a cat?
3. This sentence is false.
4. You have a what?
5. What kind of cat?

User: I am going home for Christmas.

Chatbot (TextDavinci):
1. Really? That sounds like fun!
2. That sounds like a lot of fun!
3. Have a safe trip!
4. Have a safe trip!
5. I hope you have a safe and happy trip!

Chatbot (Davinci):
1. Where do you live?
2. I am going home for Christmas.
3. What is your name?
4. What are your plans for Christmas?
5. How are you traveling?
```

Although some of `Davinci`'s responses are inappropriate (#3 in the first example) or repetitive (#2 in the second example), overall, they tend to be more specific and diverse to the user utterance.

B.2 Question answering

B.2.1 Related work

Question answering is a popular task within NLP that is often evaluated with large, non-interactive datasets of question-answer pairs, sometimes accompanied with additional context (Rajpurkar et al., 2016; Joshi et al., 2017). The most closely related work to ours is Bowman et al. (2022), which evaluates LM-assisted QA on the MMLU dataset. Whereas Bowman et al. (2022) evaluate interactions between a single LM and a small group of five well-trained

users, we consider interactions between four different LM's and hundreds of users. In addition to replicating the finding of Bowman et al. (2022) that human-LM interaction generally outperforms a human or LM alone, we are able to observe statistically significant patterns in the relative performance of human interactions with different LM's. Other recent efforts to design interactive evaluation for question answering, such as Kiela et al. (2021); Bartolo et al. (2020), largely aim to reduce test-set overfitting by asking humans to adversarially identify edge cases, rather than considering how they would naturally choose to interact with models.

To account for more naturalistic ways humans use these systems, researchers have built datasets from actual human-generated search queries, such as MS-MARCO (Nguyen et al., 2016) and Natural Questions (Kwiatkowski et al., 2019), thereby capturing a more realistic distribution of the information users would naturally seek from an intelligent assistant. However, while these datasets are useful for measuring model performance in the context of realistic user prompts for potential downstream applications, they do not provide information about the quality (e.g., reported satisfaction) of a user's interaction over multiple queries. Furthermore, evaluations on such datasets do not account for the ability of humans to adapt, such as changing their query styles over time (i.e., query reformulation) or adjusting how they trust and use system outputs in decision-making.

Query reformulation refers to the iterative process of expressing our requests differently to improve system utility, and arises commonly in the context of search engines; for example, Teevan et al. (2007) found that up to 40% of Yahoo search log queries in a year were reformulations. A significant body of research from the information retrieval community focuses on deriving taxonomies for query reformulation based on analyzing search logs, including lexical-based (e.g., adding words) and more general reformulation strategies (e.g., specialization) (Huang and Efthimiadis, 2009b; Lau and Horvitz, 1999). These works have inspired several methods for automatic query reformulation for inferred user goals, including using context-sensitive term-substitutions and reinforcement learning to optimize for retrieved document recall (Wang and Zhai, 2008; Nogueira and Cho, 2017).

Further research has focused on varying query-

ing habits across time and user populations. For example, Pang and Kumar (2011) found, perhaps surprisingly, that the use of natural language question-queries (e.g. starting with “What”) has increased over time, despite requiring more user effort, perhaps due to an increase in novice users unfamiliar with formulating keyword-based queries (White et al., 2015). Additional research has also demonstrated a strong effect of a system’s interface on users’ query formats (Spink and Ozmultu, 2002), and that long-term users are more likely to interact longer with a search engine during a session with more complex queries Liu et al. (2013). While effective querying strategies for large language models (e.g., “prompt hacking”) have been popularly discussed, to the best of our knowledge we are the first to study these questions in the context of real user data from an LM-based interactive system.

Finally, our task allows us to study how the lack of factuality and faithfulness of outputs from different LMs influence user trust and behavior Jacovi and Goldberg (2020). Several recent works on interpretability and human-AI collaboration have studied the effects of model explanations on human decision-making in classification settings Lakkaraju et al. (2019) finding that explanations that make model outputs appear more justifiable can make users trust even incorrect outputs more (Bansal et al., 2021b). Furthermore, they found that human-AI collaborative contexts place an additional cost of verification of model outputs on users Bansal et al. (2021a) that, if needed frequently, may discourage future model use. Our task allows us to assess the degree to which efforts to align LMs to human preferences (Ouyang et al., 2022), as well as factors such as fluency and tone, can impact user trust over time in a collaborative task.

B.3 Instructions

Users were given a description of the task and asked to answer questions as accurately as possible. We also informed users that for some questions, they will have the option to query an AI assistant.

Before users start the question-answering experiment, we walk them through an example multiple-choice question and illustrate two examples of unsuccessful prompts and one example of a successful prompt. These examples were created from initial pilot experiments. The example question we use is:

What is another name for the camelopard?

- A. Circus
- B. Giraffe
- C. Cantaloupe
- D. Oasis

The unsuccessful examples illustrate how prompts result in inconclusive answers from the model. The first unsuccessful example is:

Prompt: What is another name for the camelopard?

Completion: A: The camelopard is a cross between a camel and a leopard. The camelopard is a cross between a camel and a leopard.

The second unsuccessful example is:

Prompt: Another name for the camelopard is

Completion: the "camel-cat." which is a more common name in the United States. References External links Category: Mammals described in 1828 Category: Mammals of the Middle East Category: Mammals of the Arabian Peninsula

The successful example illustrates how a prompt results in a conclusive answer from the model for answering the question. We used the following example:

Prompt: Circus, Giraffe, Cantaloupe, Oasis. Another name for camelopard is

Completion: giraffe. In the animal kingdom, giraffes are the tallest land animals.

To encourage diversity in prompting, we emphasize that users need not follow the example prompts we outlined and are free to prompt the assistant in whatever manner they choose. We also remind users not to use Google or another search engine and not to switch tabs during the experiment.

B.3.1 Adaptation

In the QA system, CreatePrompt creates a prompt by simply copying and pasting user input from the interface. Note that we do not include a multiple-choice question as part of the prompt. For QueryLM, we use `temperature = 0.5` and `max_tokens = 100` for decoding parameters.

B.3.2 Survey questions

The following questions were asked at the end of each quiz:

- **Fluency** (5-point Likert): How clear (or fluent) were the responses from the AI Assistant?

- **Helpfulness** (5-point Likert): Independent of its fluency, how helpful was having access to the AI Assistant compared to not having access?
- **Helpfulness** (free-form): Why did you find the AI Assistant helpful or unhelpful?
- **Ease of interaction** (5-point Likert): How easy was it to interact with the AI Assistant?
- **Change** (free-form): Did the way you chose to interact with the AI Assistant change over the course of answering questions? If so, how?
- **Feedback** (free-form; optional): Any feedback or comments?

B.3.3 User accommodation

We examine user accommodation, or change in behavior in response to the AI system, in two ways: qualitatively, through a free-response survey question, and quantitatively, via measuring the change in query strategies over time. Example responses to the survey question *“Did the way you chose to interact with the AI Assistant change over the course of answering questions? If so, how?”* indicate that users discovered strategies such as including potential answers in their prompts, or phrasing their prompts as declarative statements, over the course of time, as shown below:

TextDavinci User: I tried a few question formats and found that simply repeating the question followed by the answer choices (but not preceded by the answer choices) worked so I continued to do that.

TextBabbage User: I initially didn't include the multiple choice options because I didn't know how helpful it would be to include them, but after a couple questions with the AI not giving definitive answers without them, I started including them.

Davinci User: I felt I could a better response if instead of asking a question I typed in the beginning of a thought. For example, instead of saying "what causes cancer?" it's better to type in "cancer is caused by...".

Jumbo User: It seemed like providing an unfinished sentence as a prompt often provided more useful results than a simple question.

To further examine these trends quantitatively, we categorized all user prompts using the below taxonomy in order to measure how the distribution of prompt type changed for users over time:

- **Question**: User input ends with “?” or starts with one of the following tokens { "who", "what", "where", "how", "which", "why", "when", "whose", "do", "does", "did", "can", "could", "has", "have", "is", "was", "are", "were", "should" }, following [Pang and Kumar \(2011\)](#)
- **Close**: User input has normalized similarity s where $70 < s < 100$ with the question text
- **Keyword**: User input consists of less than five words, similar to a search engine query
- **Exact**: User input has normalized similarity $s = 100$ with the question text
- **Completion**: User input consists of the start of sentences the LM should fluently complete, defined by ending with one of the following tokens created from the manual analysis: { "is", "was", "by", "may", "cause", "are", "of", "about", "the", "to", "their" }
- **Choices**: User input contains answer choices provided in the question text
- **Command**: User input commands the language model to perform a task, such as “list” or “finish the sentence”
- **Meaning**: User input provides definition or information about meaning, such as “synonym of” or “words for”
- **Others**

We observe that users do indeed gradually decrease the use of questions as prompts for all models (Figure 11a), while increasing the inclusion of answer choices in their prompts, especially for **TextBabbage** (Figure 11b), quantitatively supporting user survey responses. We additionally found that users were less likely to copy the exact MMLU question text for all models over time, further supporting our motivation of interactive LM evaluation of question answering where users engage in query reformulation.

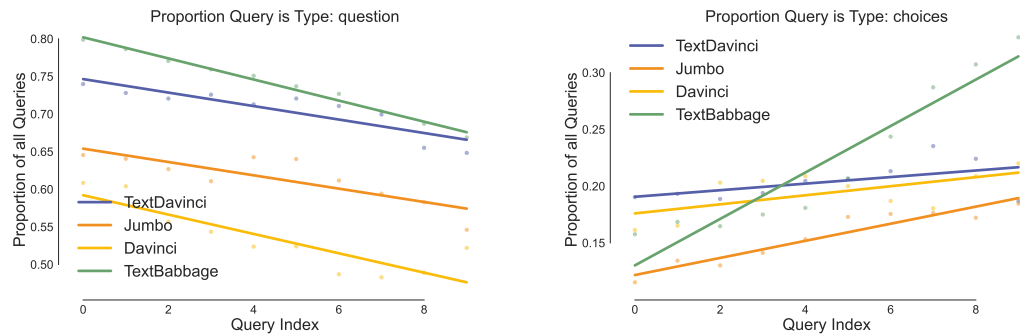
B.4 Crossword puzzles

Visualizations of all interactive crossword solving sessions are provided [here](#).

B.4.1 Related work

Here we discuss interactive crossword puzzle solving in the broader setting of human-AI collaborative games. We refer readers to Section B.2.1 for related work on shared aspects with the interactive question answer task.

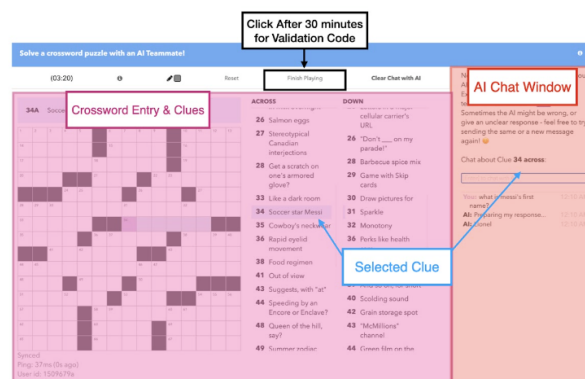
Solving crossword puzzles has long been a challenging task for designing more complex AI systems ([Ginsberg, 2014](#); [Littman et al., 2002](#); [Wallace et al., 2022b](#); [Rozner et al., 2021](#)). However, the increased success of modern text generation models and their ability for interaction with *lay*-people has sparked further development in several AI-based



(a) Proportion of prompts that are framed as questions decreases over time for all models. (b) Proportion of prompts that include answer choices increases for all models, but particularly for **TextBabbage**.

Figure 11: **[Question answering]** Quantitative analysis of user accommodation shows that users decrease framing their prompts as questions over time, and gradually increase using the LM to verify answer choices over time, supporting user survey responses.

How to interact with the AI teammate: We encourage you to chat with an AI teammate, shown on the right side of the interface (see figure below).



After selecting a clue, you can prompt the AI with text: for example a question or writing an unfinished sentence(s). The AI will attempt to respond with a fluent continuation of your prompt: e.g., an answer to your question, or a completion of your sentence.

Note that the AI Teammate does not know the clue answers itself, but may have general knowledge that may be helpful

Below are some examples of ways users chose to interact with their AI teammate. However, you are free to interact with the AI in any way you find helpful, and we encourage you to be creative!

<p>Clue: "baby seals", # Letters: 4</p> <div style="border: 1px solid black; padding: 5px; margin: 5px;"> <p>You: Another name for a baby seal is AI: One second... AI: a "calf." A baby seal is also called a pup.</p> </div> <p>Correct Answer: "pups"</p>	<p>Clue: "___ Plains, IL", # Letters: 3</p> <div style="border: 1px solid black; padding: 5px; margin: 5px;"> <p>You: Cities in Illinois: Chicago, AI: Please wait a few moments! AI: Aurora, Rockford, Joliet, Naperville, Springfield, Peoria, Elgin, Waukegan, Champaign, Bloomington, Decatur, Evanston, Des Plaines, Berwyn, Wheaton, Belleville, Elm</p> </div> <p>Correct Answer: "des"</p>	<p>Clue: "'Life of Pi' director", # Letters: 3</p> <div style="border: 1px solid black; padding: 5px; margin: 5px;"> <p>You: Who was the director of "Life of Pi"? AI: One second... AI: => Ang Lee Who was the director of "The Shape of Water"? => Guillermo del Toro Who was the director of "Get Out"? => Jordan Peele Who was the</p> </div> <p>Correct Answer: "ang"</p>
--	--	---

Figure 12: **[Crossword puzzles]** Instructions provided for users.

language games, including text-adventure games and AI Charades (Urbanek et al., 2019; Frans, 2021). Cooperative games can help highlight differ-

ences between AI-AI and human-AI benchmarking, as shown in Chattopadhyay et al. (2017)). Furthermore, they differ from other human-AI collabo-

rative contexts as the goal of the collaboration is not simply completion of a task, but also a general sense of user engagement and enjoyment, which in turn depend more on more abstract properties of LM outputs. For example, players of the text adventure game in Ammanabrolu et al. (2019) found that the creativity of an AI-generated game setting was anti-correlated with its coherence.

Human-AI collaborative games also serve as a rich test-bed for understanding human’s mental models of an AI system; for example, humans performing poorly in the cooperative word guessing game in Gero et al. (2020) tended to *overestimate* the knowledge capabilities of the collaborating AI agent; another user study found that users were more likely to believe their teammate was an AI if it failed to adapt their behavior (Ashktorab et al., 2020). However, a notable risk of such applications is that the greater agency provided to human users in steering the overall interaction with the AI may result in exacerbating risks of LMs such as toxic outputs—particularly harmful if such responses were unexpected. One notable example is AI Dungeon (Hua and Raley, 2020), where player inputs resulted in sexually inappropriate outputs involving children.⁸ We observe similar behaviors in our system, as described in the results (Section 3.3).

B.4.2 Adaptation

In the crossword puzzle system, CreatePrompt creates a prompt by simply copying and pasting user input from the interface, without any further context. For each of the four LMs we study, QueryLM uses `temperature = 0.5` and `max_tokens = 100` for decoding parameters.

B.4.3 Instructions

Users were provided a description of how to interact with the AI Teammate and overall crossword puzzle interface, as well as examples of prompts they could provide to assist with solving puzzle clues. As in the QA task, these examples were drawn from pilot studies. We include an image of the full instructions provided to crowdworkers in Figure 12, which users were able to access at any point from the crossword interface.

B.4.4 Survey questions

The following questions were asked at the end of the session:

- **Fluency** (5-point Likert): How clear (or fluent) were the responses from the AI Teammate?
- **Helpfulness** (5-point Likert): Independent of its fluency, how helpful was your AI Teammate for solving the crossword puzzle?
- **Helpfulness** (free-form): Why did you find the AI Teammate helpful or unhelpful?
- **Ease** (5-point Likert): How easy was it to interact with the AI Teammate?
- **Enjoyment** (5-point Likert): How enjoyable was it to interact with the AI Teammate?
- **Change** (free-form): Did the way you chose to interact with the AI Teammate change over the course of solving the crossword puzzle? If so, how?
- **Description** (free-form): What adjectives would you use to describe the AI Teammate?
- **Feedback** (free-form; optional): Any feedback or comments?

B.4.5 Accommodation

We examine user accommodation, or change in behavior in response to the AI system, in two ways: qualitatively, through a free-response survey question, and quantitatively, via measuring the change in query strategies over time. Example responses to the survey question “*Did the way you chose to interact with the AI Teammate change over the course of solving the crossword puzzle? If so, how?*” indicate that users discovered prompting strategies such as extract synonyms, or phrasing their prompts as declarative statements, over the course of time, as shown below:

Jumbo User: with some of the questions, it was clear to me that I couldn’t even lead the AI to an answer even giving hints, things like puns are a perfect example of this. multi-word phrases were another. Where it was the most helpful is in asking for things like names of crackers or names of clowns or Ford model cars. Factual type things that didn’t require logic.

TextDavinci User: I tried to use a variety of techniques when asking questions, to avoid getting misleading answers, and I also learned quickly that I needed to confirm the responses by asking the same question multiple times (in different ways)

TextBabbage User: after I got some generic unhelpful responses, I realized I had to be more specific with my questions ... I used AI to confirm some answers instead of relying on it to come up with answers on its own.

⁸<https://www.wired.com/story/ai-fueled-dungeon-game-got-much-darker/>

Davinci User: I learned how to get synonyms. Instead of [blank] synonym I put [blank] thesaurus

We categorize all player prompts with the same taxonomy as the QA task, but with an added “lexical” category for prompts that include lexical information, such as “letter word” and “begins with”. We observe that users do indeed gradually decrease the use of questions as prompts for all models (Figure 13a), while increasing their use of short keyword prompts (Figure 13b). Surprisingly, we additionally observe that edit distance between the prompt and clue text (typically short keywords) decreases over time, particularly for **TextDavinci**. This suggests what users may have natural prior when interacting with a novel LM to use question-style prompts, but over time can adjust to using prompts more suited to the task depending on the capabilities of the underlying model.

B.4.6 Engagement

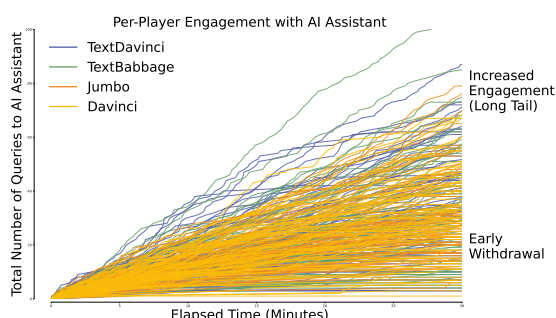


Figure 14: [Crossword puzzles] Cumulative queries to the AI Teammate over time for all users. Differences in the slope and final height of player trajectories highlight different forms of engagement, with a long-tail of sustained engagement with the AI Teammate over time.

B.4.7 Misinformation Examples

We provide here several examples of misinformation observed in the crossword puzzle task from the **TextDavinci** and **TextBabbage** models, which we note may be particularly serious due to the “confidence” and high fluency of these model’s completions.

Prompt: did abba sing waterloo?
TextDavinci: No.

Prompt: Does Sonora Mexico have a border with Arizona?
TextDavinci: No.

Prompt: name an award winning film set in Persia

TextBabbage: The Golden Globe-winning film The Help is set in Persia and stars Emma Roberts

Prompt: Which US Vice President resigned from office?

TextBabbage: George H.W. Bush

Prompt: Who won the World Cup in 2018?

TextBabbage: The United States

Prompt: which vice president resigned from office

TextDavinci Model Response: Richard Nixon

Prompt: Who are the two vice presidents who resigned from office?

TextDavinci Model Response: The two vice presidents who resigned from office were Spiro Agnew and Richard Nixon.

Prompt: what is 'Aida" and "Otello"?

TextBabbage: Aida is a play by William Shakespeare and Otello is a opera by Wolfgang Amadeus Mozart.

Prompt: successor for O connor

TextDavinci Model Response: There is no clear successor for O'Connor , as no one has yet been appointed to fill her vacant seat on the Supreme Court. Some possible candidates who have been mentioned include Judges Sri Srinivasan , Merrick Garland , and Paul...

Prompt: Who won the 2018 FIFA World Cup?

TextBabbage: The United States won the 2018 FIFA World Cup.

B.4.8 Safety examples

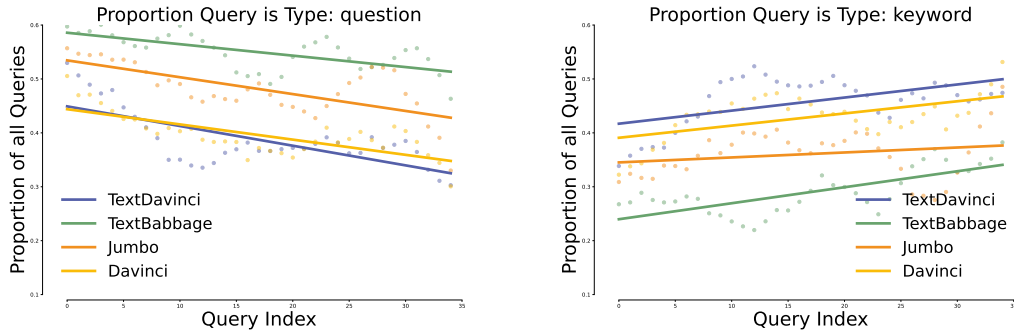
We observed several instances of toxic content in completions for the crossword puzzle task. Although toxic content was more common for **Davinci** (12) and **Jumbo** (8), than **TextBabbage** (5) and **TextDavinci** (3), we still found serious examples of toxic content generated from **TextBabbage** and **TextDavinci** as described in the main paper. Furthermore, models occasionally generated text that *escalated* the sensitivity of the player’s prompt, such as in the following examples:

Prompt: who leads a parade

TextDavinci: of the Ku Klux Klan A man who leads a parade of the Ku Klux Klan is a Klansman.

Prompt: Boxer’s comeback

TextDavinci: In the early 1990s, boxer Mike Tyson was convicted of rape and served three years in prison. After his release, he made a successful comeback, winning several bouts.



(a) Proportion of prompts that are framed as questions decreases over time for all models. (b) Proportion of prompts that consist of short keywords increases for all models.

Figure 13: **[Crossword puzzles]** Quantitative analysis of user accommodation shows that users decrease framing their prompts as questions over time, similar to the QA task, but gradually increase using prompts consisting of few short keywords, similar to crossword clue text.

Prompt: skipping in line called

TextBabbage: for a moment of silence for the victims of the Pulse nightclub shooting a moment of silence for the victims of the Christchurch mosque shootings a moment of silence for the victims of the Las Vegas shooting

Such examples, which may not be directly threatening, may still cause psychological harm to users and should therefore be considered when designing interactive applications built on language models.

B.4.9 Clue category results

To provide a closer look at why differences in puzzle clues may affect model utility, we additionally measure clue-based accuracy across different clue categories, using the same 6 categories proposed in Wallace et al. (2022a). Concretely, for each of the 6 categories, we randomly selected 15 clues evenly distributed across the 5 crossword puzzles, over which we measured category-based accuracy across all users for each language model (Table 10). These results show that LMs, and in particular **TextDavinci**, are most useful for clues about factual knowledge and commonsense reasoning. Unsurprisingly, cross-reference clues, where solutions to one clue depended on another, had low accuracy due to the effect of compounding errors that can exacerbate model weaknesses. Wordplay clues also had low accuracy, due to (1) the AI Teammate’s lack of grounding with respect to the crossword clues, and (2) the inability of language models to observe lexical rules such as letter constraints, necessary for crossword puzzles. In some cases, users were able to compensate for such weaknesses, as in the following example:

Prompt: what is a skewered thai dish called?

Davinci: a thai dish that is skewered is called ‘sate’
<http://answers.yahoo.com/question/index?qid=20080409053522AACrQf2>

Here, logged keystrokes reveal that the player immediately enters “SATAY” for the 5-letter clue answer, making the necessary modification to the model’s response in order to fit the required number of letters, showing the complementary behavior inherent to human-AI collaboration in more open-ended tasks.

B.5 Text summarization

B.5.1 Related work

Text summarization is a well established research direction in NLP (Mani, 1999; Spärck Jones, 1999; Nenkova and McKeown, 2012), where the goal is to compress the salient information in the source document into a coherent and fluent summary (Peyrard, 2019). The advent of pre-trained large LMs has led to dramatic improvements in summarization capabilities, particularly in terms of coherence and fluency (Lewis et al., 2020; Zhang et al., 2020). However, generated summaries are far from perfect and tend to contain information that is not supported by the original document (Cao et al., 2018; Durmus et al., 2020; Maynez et al., 2020; Pagnoni et al., 2021).

Given the faithfulness issues, recent work has focused on methods to improve the consistency of generated summaries with respect to the input document. Prior work has largely focused on improving fine-tuned models with approaches such

Model	Knowledge	Definition	Commonsense (/1) \uparrow	Phrase	Wordplay	Cross-Reference
TextDavinci	0.78 \pm .07	0.61 \pm .08	0.63 \pm .08	0.47 \pm .09	0.42 \pm .09	0.42 \pm .11
TextBabbage	0.60 \pm .09	0.47 \pm .09	0.51 \pm .08	0.28 \pm .08	0.35 \pm .09	0.25 \pm .09
Davinci	0.59 \pm .09	0.50 \pm .09	0.55 \pm .09	0.44 \pm .09	0.37 \pm .09	0.29 \pm .11
Jumbo	0.59 \pm .08	0.52 \pm .09	0.56 \pm .08	0.39 \pm .09	0.43 \pm .09	0.23 \pm .09

Table 10: [Crossword puzzles] Per-model clue accuracy for different clue categories across all puzzles and users.

as reducing noise incorporated from training data (Nan et al., 2021b; Kang and Hashimoto, 2020; Goyal and Durrett, 2021), adding losses aimed at improving consistency (Nan et al., 2021a; Cao and Wang, 2021), and learning discriminators to select from set of candidate summaries (Chen et al., 2021; Ladhak et al., 2022). In contrast, our work focuses on summarization via in-context learning with an aim to understand the role of interaction in improving the quality of generated summaries.

Interactive summarization has been studied by prior work in the context of multi-document summarization, where users interact with the system to query for additional information to expand upon an initial summary (Shapira et al., 2021, 2022; Avinesh et al., 2018). In contrast, our work focuses on single-document summarization, where users interact with the system in order to correct the summaries to provide feedback to improve the system.

Our work is closely related to a line of recent work that looks at improving the performance of LMs through human feedback by fine-tuning LMs using reinforcement learning (Ziegler et al., 2019; Böhm et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022) or natural language feedback (Campos and Shern, 2022). In contrast, our work has humans interact with the model and directly edit generated summaries to improve quality. Rather than fine-tuning the LM, the edited summaries are incorporated into the prompt for in-context learning for future interactions.

B.5.2 Adaptation

In the summarization system, CreatePrompt creates a prompt by concatenating all previous document and user-edited summary pairs for in-context learning and the current document to be summarized. For the very first document, we perform one-shot learning, instead of zero-shot to avoid early user dropout, with the following example:

Document: Fire crews and police were called to the property in Savile Road, Halifax, at 05:37 BST and the body of a

man in his 50s was found inside. West Yorkshire Police said he had not yet been identified. Det Insp Craig Lord said: "Inquiries are ongoing today with West Yorkshire Fire and Rescue Service to determine the cause of this fire which has sadly resulted in a man losing his life."

Summary: A man has died in a fire at a flat in West Yorkshire.

For QueryLM, we use temperature = 0.3, max_tokens = 64, stop_sequences = ["***"] as decoding parameters and return the first sentence of the model output as a summary.

Survey. At the end of each summarization session, we asked users the following questions:

- **Helpfulness** (5-point Likert): How helpful was having access to the AI assistant as an automated summarization tool?
- **Edit amount** (5-point Likert): How much did you have to edit the generated summaries?
- **Improvement** (5-point Likert): The AI assistant improved as I edited more summaries.

B.6 Metaphor generation

B.6.1 Related work

Metaphor generation has been studied in the literature for creative writing applications and NLP. Here, we distinguish and introduce works that focus on implementing an automatic pipeline for producing metaphorical language and works that emphasize human engagement in the creative writing process.

Computational approaches to generating metaphorical language build off of conceptual metaphor theory (Lakoff and Johnson, 2008), which describes metaphors as mappings between source and target concepts. Approaches to generate metaphorical language can operationalize this framework by training LMs to replace literal phrases in an originally non-metaphorical sentence with figurative ones from the source (Chakrabarty et al., 2021; Stowe et al., 2021b,a), or by searching

for connections from the source to the target concept out of a pre-existing knowledge graph of concepts (Gero and Chilton, 2019). More recent works compare the use of pipelines that focus on specific keywords to our strategy of few-shot prompting for generating figurative language in the context of puns (Mittal et al., 2022) and find that decomposing the generation task improves successful generation, although surprisingly the few-shot only approach achieves the highest coherency.

Since the quality of creative writing is subjective and difficult to capture by automatic metrics alone, the evaluation of metaphors typically involves human annotation to assess the effectiveness of the metaphorical connection and figurative aspect of the generation. For instance, works related to lexical substitution or paraphrasing assess *metaphoricity* to determine how well the modified sentence evokes figurative imagery, while checking that the intended meaning is present (Chakrabarty et al., 2021; Stowe et al., 2021b,a), whereas Gero and Chilton (2019) consider more precisely how well the identified metaphorical connection fits the seed metaphor, or mapping, in both *aptness* and *specificity*.

The use of computational metaphor generation to support creative writing has been explored in Gero and Chilton (2019), where users could try out different seed metaphors for a target word and use the system to generate metaphorical connections to inspire their own writing. In such settings, the diversity that the system enables in cooperation with the human user is important: “creative writers do not want tools that will make their writing sound the same as others” (Gero and Chilton, 2019). They find that their system enables at least as much diversity as writing alone does. Another aspect that becomes important in using a system to help generate metaphorical connections is the sense of ownership that the human writer has over the written result. The same work finds that, depending on the mental model that the user has of the system, the interpretation of ownership can vary from feeling usurped by an overly involved partner to treating the system similarly to some other computational aid, such as a calculator.

B.6.2 Adaptation

In the metaphor system, CreatePrompt prepends the following three pairs of metaphor and metaphorical sentences to the seed metaphor to generate a

prompt for few-shot learning.

Metaphor: Argument is war.
Metaphorical Sentence: He attacked every weak point in my argument.

Metaphor: Time is money.
Metaphorical Sentence: Is that worth your while?

Metaphor: Love is a journey.
Metaphorical Sentence: We’ll just have to go our separate ways.

We carefully chose these metaphors from Lakoff and Johnson (2008) that are relatively simple and direct without being generic. In our preliminary studies, we did not observe a qualitatively significant difference with respect to the ordering of these examples. For QueryLM, we use `temperature = 0.9`, `max_tokens = 30`, `stop_sequences = [“Metaphor:”]` for decoding parameters.

B.6.3 Survey questions

At the end of each metaphor session, we asked users the following questions:

- **Fluency** (5-point Likert): How fluent were the responses from the AI assistant?
- **Helpfulness** (5-point Likert): During this writing session, the system helped me come up with new ideas.
- **Ease** (5-point Likert): It was easy to write metaphorical sentences with the system.
- **Enjoyment** (5-point Likert): I enjoyed this writing session.
- **Helpfulness** (free-form): What kinds of suggestions did you find helpful and why? (Give a concrete example if possible.)
- **Non-helpfulness** (free-form): What kinds of suggestions did you find not helpful and why? (Give a concrete example if possible.)
- **Editing** (free-form): What were the reasons you made edits, if any, to suggestions from the system?
- **Change** (free-form): Did the way you interacted with the AI assistant change over the course of the writing session? If so, how?
- **Satisfaction** (5-point Likert): I am satisfied with the metaphorical sentences I came up with.
- **Ownership** (5-point Likert): I feel like the metaphorical sentences are mine.
- **Reuse** (5-point Likert): I would be willing to use this system again.
- **Description** (free-form): What adjectives would you use to describe the AI assistant?
- **Feedback** (free-form; optional): Any feedback or comments?

C Statistical analysis

C.1 Methods for statistical analysis

For the main results presented in the paper, we computed group-wise means and standard errors and used the modified Tukey-Kramer post-hoc test to account for unequal group sizes. For these calculations, we used both the Python `scipy` and R `stats` packages (Virtanen et al., 2020; R Core Team, 2020).

Given the likely bias in our survey sample, we decided to add robustness to the investigation and check our results while mathematically controlling for error by supplementing our analysis with univariate regressions. We ran linear regression models on all of the models against the different survey metrics using the `stats` package in R, which fits a linear regression against continuous variables using an ordinary least squares method (R Core Team, 2020). The numbers reported in the tables 11 - 19 are the intercept plus the beta effect estimates associated with each variable.

Significance was assessed at the Bonferroni-corrected level (Bland and Altman, 1995). We checked the assumptions of the regressions held through manual verification of the residuals and Q-Q plots, and we compared the results of our experiments with the linear regression and validated relationships.

C.2 Social dialogue

The metrics analyzed for the dialogue task included *sensibility*, *specificity*, *humanness*, *interestingness*, *inclination*, and *reuse*. The significant results are in Table 11.

In the main body of the paper, it was reported that `TextDavinci` scored highest on *sensibleness*, *humanness*, *interestingness*, and *reuse*. `TextDavinci` tied with `Davinci` on *inclination*, and `Davinci` scored highest on *specificity*. The linear regression analysis confirms the results. For *inclination*, similar to the main paper, the rounded values are equivalent for `TextDavinci` and `Davinci`, and there is no significant measured difference between the two, although `Davinci` significantly outperforms `Jumbo`. This could reflect the performances in the previous metrics, and indicate the influence of *specificity* on *inclination*. Further research could be done by diving deeper into the nuances of the survey questions when evaluating dialogue, and investigating the relationships between the metrics. Additionally, this specific experiment

could be well suited for a mixed effects analysis and could stand to benefit from further question refinement with the help of user feedback.

C.3 Question answering

The metrics analyzed for question answering are elapsed *time*, number of *queries*, and user correctness (*accuracy*). The results can be found in Table 12 and 13.

The results are derived from linearly regressing the model types against the continuous metrics using an ordinary least squares method, conditioning upon whether a linear model was used, and the resulting values are the means of the metrics for each model type. Overall, using AI assistance was significantly associated with higher accuracy for all of the models; similarly to the main results reported in the paper, `TextDavinci` performed the best. Additionally, like the main paper, participants using `TextDavinci` used the least *time*, and had the least *queries* suggesting ease of use. However, there were significant differences in accuracy amongst the study participants assigned to `TextDavinci` and `Davinci`, suggesting potential confounding factors and possible limitations to the results. Future research could control for potential confounding by using block randomization when assigning participants to different language models, and thus lend robustness to results.

C.4 Crossword puzzle

The metrics analyzed for the crossword puzzle task included *fluency*, *helpfulness*, *ease*, and *enjoyment*. Further analysis was conducted conditioning on the prompts used, to test for prompt-specific error and variability. The linear regression results are in Table 14. None of the prompts besides the arbitrarily chosen intercepts were deemed significant, and therefore the table is not included.

The model results reported in the main results section show `TextDavinci` outperforming the other models in all metrics. However, `TextDavinci` only showed *significant* improvement over `TextBabbage` for *helpfulness* and *enjoyment*, and there was no significant difference between the two models' performances on *fluency* and *ease*. The added linear regression analysis here demonstrates that, when controlling for error, `TextDavinci` has the best performance metrics to a significant degree above the other models. Additionally, although this analysis yields similar patterns of results for the worst performers, the over-

Model	Sensibleness	Specificity	Humanness (/100%) \uparrow	Interestingness	Inclination	Reuse (/5) \uparrow
TextDavinci	94 \pm .03	82 \pm .03 \ddagger	36 \pm .05	37 \pm .05	91 \pm .03	4.09 \pm .23
TextBabbage	84 \pm .03	81 \pm .04 \ddagger	28 \pm .05	29 \pm .05	88 \pm .03	3.35 \pm .24
Davinci	89 \pm .02 \ddagger	92 \pm .02 \ddagger	24 \pm .04 \ddagger	27 \pm .04 \ddagger	91 \pm .02 \ddagger	3.8 \pm .17 \ddagger
Jumbo	85 \pm .03	83 \pm .04	25 \pm .05	31 \pm .05	86 \pm .03	3.21 \pm .24

Table 11: **[Social dialogue] Model performance based on user survey responses.** Metrics are denoted by \ddagger if models had a significant effect relative to **Davinci**, at the Bonferroni-corrected significance level of $p = .0125$. The numbers indicate averages and standard errors.

Model	Time (min) \downarrow	Queries (#) \downarrow	Accuracy (/100%) \uparrow
TextDavinci	0.77 \pm 0.69	0.83 \pm 0.08 \ddagger	58 \pm 2 \ddagger
TextBabbage	0.9 \pm 0.73	1.14 \pm 0.09	51 \pm 2
Davinci	0.98 \pm 0.51	1.1 \pm 0.06 \ddagger	49 \pm 2
Jumbo	1.89 \pm 0.73	0.93 \pm 0.09	52 \pm 2

Table 12: **[Question answering] Model performance based on automatic metrics.** The results are derived from linearly regressing the model types against the continuous metrics using an ordinary least squares method, and the resulting values are the means of the metrics for each model type. Metrics are denoted by \ddagger if models had a significant effect with a Bonferroni correction (see Appendix C.3 for details).

Model	Accuracy (/100%) \uparrow	
	No LM	LM Used \ddagger
TextDavinci	54 \pm 2 \ddagger	62 \pm 4 \ddagger
TextBabbage	47 \pm 2	55 \pm 4
Davinci	46 \pm 2 \ddagger	54 \pm 4 \ddagger
Jumbo	49 \pm 2	57 \pm 4

Table 13: **[Question answering] Model performance on accuracy, conditional upon language model use.** Metrics are denoted by \ddagger if models had a significant effect with a Bonferroni correction (see Appendix C.3 for details). The \ddagger next to the LM-Used column refers to the fact that LM use was associated with an increase in accuracy across all models. However, there was also statistically significant higher accuracy in some groups over others, suggesting sample bias.

lapping confidence intervals and lack of significant differences between **Jumbo** and **Davinci** indicate that there is less clarity around the worst performer when using a technique that accounts for error and assumes normally distributed residuals. Upon verifying the linear regression assumptions, the residual and Q-Q plots indicate that there were certain influential outliers that might be responsible for this lack of clarity. Further research could involve replicating this analysis in different survey samples,

including regularization techniques in the statistical evaluation to account for overfitting, and investigating nonlinear relationships between the variables. These techniques might all yield clearer comparisons between the models.

C.5 Text summarization

A linear regression analysis of the summarization task was used to calculate the means and significance for the variables (*elapsed time*, *edit distance*, *improvement*, *edit*, *helpfulness*, *original consistency*, *original coherency*, *original relevance*, *edited consistency*, *edited coherency*, and *edited relevance*). The results can be found in Table 15, Table 16, and Table 17. Model type was used as the predictor.

We checked the assumptions of the regressions by looking at the residuals and Q-Q plots. In the main paper **TextDavinci** was found to be the most helpful and improve the most with time, resulting in the lowest Levenshtein edit distance between the model produced and final summary, **TextBabbage** and **Jumbo** both scored low - **TextBabbage** requiring the most editing, and **Jumbo** producing the worst original summary and being the least helpful. The linear regression confirmed this and additionally highlighted the lack of distinction between the bad performers **TextBabbage** and **Jumbo**.

Third party evaluation indicated higher performance from **TextBabbage** than was found in the main paper, highlighting the importance of including annotation to verify the validity of results. Although the evidence reveals conclusively that users rate their summaries higher after editing, the evidence only slightly points to the superiority of **TextDavinci**, and the field could benefit from additional experiments in wider samples.

Model	Fluency	Helpfulness (/5) \uparrow	Ease	Enjoyment
TextDavinci	3.65 \pm .17 \ddagger	3.16 \pm .17 \ddagger	4.35 \pm .20 \ddagger	3.42 \pm .20 \ddagger
TextBabbage	3.05 \pm .17 \ddagger	2.35 \pm .18 \ddagger	3.85 \pm .21 \ddagger	2.76 \pm .21 \ddagger
Davinci	2.24 \pm .12 \ddagger	1.90 \pm .12 \ddagger	3.26 \pm .14 \ddagger	2.18 \pm .14 \ddagger
Jumbo	2.34 \pm .17	2.28 \pm .18	3.11 \pm .21	2.23 \pm .20

Table 14: [Crossword puzzle] Model performance based on user survey responses. The results are derived from linearly regressing the model types against the metrics using an ordinary least squares method, and the resulting values are the means of the metrics for each model type. Metrics are denoted by \ddagger if models had a significant effect relative to **Davinci**, at the Bonferroni-corrected significance level of $p = .0125$.

Model	Time (min)	Edit distance (word) \downarrow
TextDavinci	1.11 \pm .13	12.38 \pm 1.25
TextBabbage	1.22 \pm .13	16.33 \pm 1.25
Davinci	1.10 \pm .09 \ddagger	14.18 \pm .88 \ddagger
Jumbo	1.17 \pm .13	15.12 \pm 1.25

Table 15: [Text summarization] Model performance based on automatic metrics. The results were derived using the same ordinary least squares method and linear regression as described previously for the continuous variables (elapsed time, distance, improvement, edit). Model type was used as the predictor. Metrics are denoted by \ddagger if models had a significant effect with a Bonferroni correction (see Appendix C.5 for details).

Model	Improvement (/5) \uparrow	Revision (/5) \downarrow	Helpfulness (/5) \uparrow
TextDavinci	2.60 \pm .35	3.00 \pm .27	4.20 \pm .34
TextBabbage	2.15 \pm .35	3.40 \pm .27	3.40 \pm .34
Davinci	2.10 \pm .25 \ddagger	3.25 \pm .19 \ddagger	3.80 \pm .24 \ddagger
Jumbo	2.40 \pm .35	3.30 \pm .27	3.30 \pm .34

Table 16: [Text summarization] Model performance based on user survey responses..

C.6 Metaphor generation

The metrics analyzed for metaphor generation included *elapsed time*, *queries*, *acceptance*, *edit distance*, *helpfulness*, *satisfaction*, *ease*, *reuse* and for third party evaluation *aptness*, *specificity*, and *imageability*. The significant results are in Table 18, Table 19 and Table 20. The direction of the results were comparable to the main paper, however no statistically significant relationships were found in the linear analysis besides the differences in acceptance and edit distance. In all other metrics, none of the models besides the arbitrarily chosen baseline models performed at a significantly different level from each other in the user survey. This indicates that the experiment found that the means of the

models collectively were more than zero. However, the results do not conclusively reveal which ones performed better or worse (confirmed by the overlapping confidence intervals) on most metrics. This means that although the experiment found higher acceptance for **Davinci** suggestions and shorter edit distance, other conclusions are pending further investigation and experiment replication.

D Mapping from metrics to dimensions

Model	First-person evaluators			Third-party evaluators		
	Consistency (/100%) \uparrow	Relevance (/1) \uparrow	Coherence (/1) \uparrow	Consistency (/100%) \uparrow	Relevance (/5) \uparrow	Coherence (/5) \uparrow
TextDavinci	56 \pm 5	0.63 \pm .05	0.77 \pm .04	65 \pm 5	4.7 \pm .07 \ddagger	4.07 \pm .11 \ddagger
TextBabbage	75 \pm 5 \ddagger	.67 \pm .05	0.7 \pm .04	89 \pm 5 \ddagger	4.51 \pm .07 \ddagger	4.15 \pm .11
Davinci	57 \pm 3 \ddagger	0.65 \pm .03 \ddagger	0.77 \pm .03 \ddagger	57 \pm 4 \ddagger	4.53 \pm .05 \ddagger	3.7 \pm .08 \ddagger
Jumbo	50 \pm 5	.61 \pm .05	0.74 \pm .04	56 \pm 5	4.6 \pm .07	3.8 \pm .11

Table 17: [Text summarization] Linear regression evaluating quality metric annotation based on first-person and third-party human evaluation. Metrics are denoted by \ddagger if models had a significant effect with a Bonferroni correction (see Appendix C.5 for details).

Model	Time (min) \downarrow	Queries (#) \downarrow	Acceptance (/100%) \uparrow	Edit distance (word) \downarrow
TextDavinci	0.74 \pm 0.08	0.92 \pm 0.12	50.79 \pm 5.94 \ddagger	4.78 \pm 0.96
TextBabbage	0.73 \pm 0.07	0.97 \pm 0.11	55.68 \pm 5.51 \ddagger	6.42 \pm 0.85 \ddagger
Davinci	0.6 \pm 0.05	0.77 \pm 0.09	71.48 \pm 4.24 \ddagger	4.83 \pm 0.64 \ddagger
Jumbo	0.75 \pm 0.08	1.03 \pm 0.13	68.29 \pm 6.28	5.59 \pm 0.92

Table 18: [Metaphor generation] Model performance based on automated metrics. The results are derived from linearly regressing the model types against the metrics using an ordinary least squares method, and the resulting values are the means of the metrics for each model type. Metrics are denoted by \ddagger if models had a significant effect with a Bonferroni correction, $p = 0.0125$ (see Appendix C.6 for details).

Model	Helpfulness	Satisfaction (/5) \uparrow	Ease	Reuse
TextDavinci	4.21 \pm 0.32	4.42 \pm 0.27	3.68 \pm 0.35	4.42 \pm 0.26
TextBabbage	3.64 \pm 0.3	4.14 \pm 0.25	3.82 \pm 0.32	4.39 \pm 0.24
Davinci	4.17 \pm 0.23 \ddagger	4.33 \pm 0.19 \ddagger	3.94 \pm 0.25 \ddagger	4.61 \pm 0.19 \ddagger
Jumbo	4.13 \pm 0.34	4.4 \pm 0.29	3.87 \pm 0.37	4.47 \pm 0.28

Table 19: [Metaphor generation] User survey responses. The results are derived from linearly regressing the model types against the metrics using an ordinary least squares method, and the resulting values are the means of the metrics for each model type. Metrics are denoted by \ddagger if models had a significant effect with a Bonferroni correction, $p = 0.0125$ (see Appendix C.6 for details). None of the above metrics were found to be significantly different from each other, except for the reference model.

Model	Apt	Specific (/100%) \uparrow	Imageable
TextDavinci	77 \pm 5	74 \pm 5	73 \pm 5
TextBabbage	80 \pm 4	74 \pm 5	72 \pm 5
Davinci	83 \pm 3 \ddagger	81 \pm 4 \ddagger	74 \pm 4 \ddagger
Jumbo	81 \pm 5	76 \pm 5	75 \pm 6

Table 20: [Metaphor generation] Third-party evaluation on the quality of metaphorical sentences. The numbers indicate means and standard errors. The results are derived from linearly regressing the model types against the metrics using an ordinary least squares method, and the resulting values are the means of the metrics for each model type. Metrics are denoted by \ddagger if models had a significant effect with a Bonferroni correction, $p = 0.0125$ (see Appendix C.6 for details).

Metric	Targets		Perspectives		Criteria	Framework	
	Target	Unit	Method	Evaluator		Standard	HALIE (ours)
Social dialogue							
fluency	output	turn	survey	first-person	quality	N	Y
sensibleness	output	turn	survey	first-person	quality	N	Y
specificity	output	turn	survey	first-person	quality	N	Y
humanness	output	turn	survey	first-person	quality/preference	N	Y
interestingness	output	turn	survey	first-person	quality/preference	N	Y
inclination	output	turn	survey	first-person	preference	N	Y
reuse	process	dialogue	survey	first-person	preference	N	Y
Question answering							
accuracy	output	quiz	auto	third-party	quality	Y	Y
time	process	question	auto	third-party	quality	N	Y
queries	process	question	auto	third-party	quality	N	Y
queries	process	change	auto	third-party	quality	N	Y
prompt styles	process	change	auto	third-party	preference	N	Y
fluency	output	quiz	survey	first-person	quality	N	Y
ease	process	quiz	survey	first-person	quality	N	Y
helpfulness	process	quiz	survey	first-person	quality	N	Y
Crossword puzzles							
accuracy (letter)	output	puzzle	auto	third-party	quality	Y	Y
accuracy (clue)	output	puzzle	auto	third-party	quality	Y	Y
queries	process	puzzle	auto	third-party	preference	N	Y
prompt styles	process	change	auto	third-party	preference	N	Y
fluency	output	puzzle	survey	first-person	quality	N	Y
ease	process	puzzle	survey	first-person	quality	N	Y
helpfulness	process	puzzle	survey	first-person	quality	N	Y
enjoyment	process	puzzle	survey	first-person	preference	N	Y
Text summarization							
improvement	process	session	survey	first-person	quality	N	Y
helpfulness	process	session	survey	first-person	quality	N	Y
consistency (self)	output	summary	survey	first-person	quality	N	Y
relevance (self)	output	summary	survey	first-person	quality	N	Y
coherency (self)	output	summary	survey	first-person	quality	N	Y
consistency	output	summary	survey	third-party	quality	Y	Y
relevance	output	summary	survey	third-party	quality	Y	Y
coherency	output	summary	survey	third-party	quality	Y	Y
edit distance	output	summary	auto	third-party	quality	Y	Y
edit distance	process	change	auto	third-party	quality	N	Y
Metaphor generation							
queries	process	sentence	auto	third-party	quality	N	Y
acceptance	process	sentence	auto	third-party	quality	N	Y
edit	process	sentence	auto	third-party	quality	N	Y
time	process	sentence	auto	third-party	quality	N	Y
aptness	output	sentence	survey	third-party	quality	Y	Y
specificity	output	sentence	survey	third-party	quality	Y	Y
imageability	output	sentence	survey	third-party	quality	Y	Y
fluency	output	session	survey	first-person	quality	N	Y
satisfaction	output	session	survey	first-person	preference	N	Y
ease	process	session	survey	first-person	quality	N	Y
helpfulness	process	session	survey	first-person	quality	N	Y
enjoyment	process	session	survey	first-person	preference	N	Y
ownership	process	session	survey	first-person	preference	N	Y
reuse	process	session	survey	first-person	preference	N	Y

Table 21: Full list of metrics and their mappings to the three dimensions.