Sparks: Inspiration for Science Writing using Language Models

Katy Ilonka Gero
katy@cs.columbia.edu
Columbia University
New York, New York, USA

Vivian Liu
vl2463@columbia.edu
Columbia University
New York, New York, USA

Lydia B. Chilton
chilton@cs.columbia.edu
Columbia University
New York, New York, USA

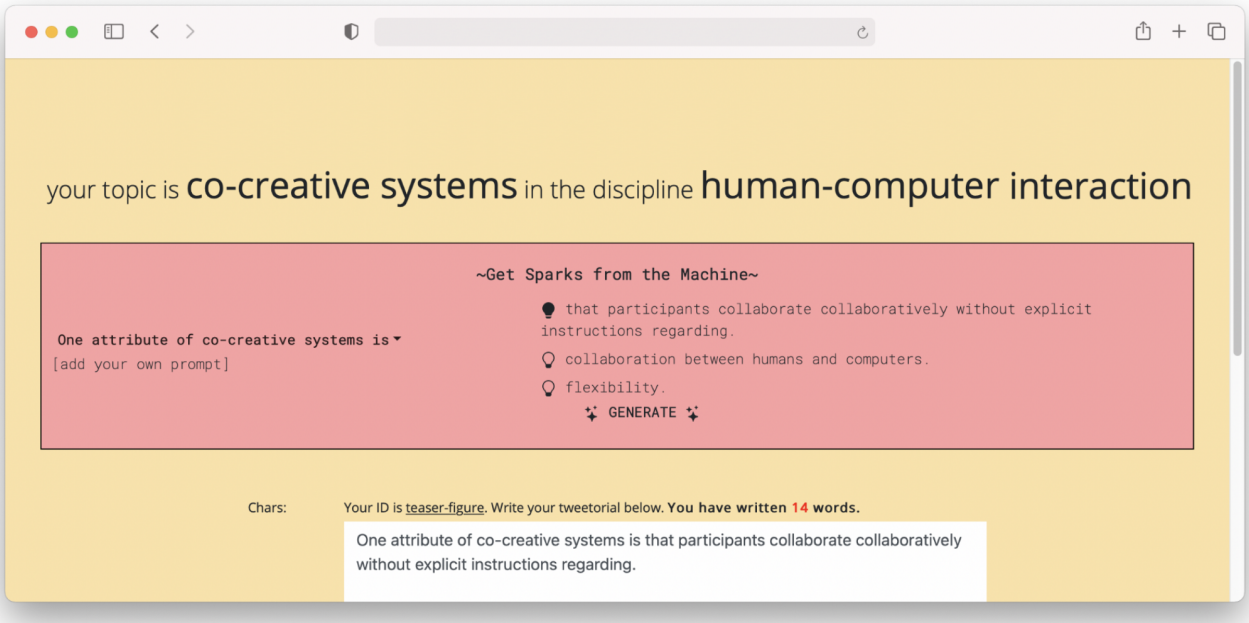

Figure 1: An example screenshot of our system for the topic 'co-creative systems' in the discipline 'human-computer interaction'. The system has generated three "sparks": sentences intended to inspire the participant when writing an explanation for their topic. The first spark has been marked as inspirational.

ABSTRACT

Large-scale language models are rapidly improving, performing well on a wide variety of tasks with little to no customization. In this work we investigate how language models can support science writing, a challenging writing task that is both open-ended and highly constrained. We present a system for generating "sparks", sentences related to a scientific concept intended to inspire writers. We find that our sparks are more coherent and diverse than a competitive language model baseline, and approach a human-written gold standard. We run a user study with 13 STEM graduate students writing on topics of their own selection and find three main use cases of sparks—*inspiration*, *translation*, and *perspective*—each of which correlates with a unique interaction pattern. We also find that while participants were more likely to select higher quality sparks, the average quality of sparks seen by a given participant did not correlate with their satisfaction with the tool. We end with a discussion about what impacts human satisfaction with AI support tools, considering participant attitudes towards influence, their openness to technology, as well as issues of plagiarism, trustworthiness, and bias in AI.

DIS '22, June 13–17, 2022, Virtual Event, Australia

ACM ISBN 978-1-4503-9358-4/22/06…$15.00
https://doi.org/10.1145/3532106.3533533

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; *Natural language interfaces*; • **Information systems** → *Language models.*

KEYWORDS

creativity support tools, writing support, co-creativity, science writing, natural language processing

ACM Reference Format:
Katy Ilonka Gero, Vivian Liu, and Lydia B. Chilton. 2022. Sparks: Inspiration for Science Writing using Language Models. In *Designing Interactive Systems Conference (DIS '22), June 13–17, 2022, Virtual Event, Australia.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3532106.3533533

1 INTRODUCTION

New developments in large-scale language models have produced models that are capable of generating coherent, convincing text in a wide variety of domains [1, 8, 55]. Their success has spurred improvements on many tasks, from classification, question answering, and summarization [8], to creative writing support [15]. These improvements demonstrate that language models have the potential to be powerful writing tools that can support writers in real-world, high-impact domains. Large-scale models are task agnostic, making them applicable to many tasks without requiring more training, and we believe such models are the future of AI technologies.

Despite their successes, language models continue to exhibit known problems, such as generic outputs [26], lack of diversity in their outputs [28], and factually false or contradictory information [36]. Additionally, there remain many unknowns about how this technology will interface with people in real-world writing tasks, such as what interactions best serve writers, how language models can best contribute to different writing forms [10], and how to mitigate the bias that language models encode [5].

In this work we study how language models can be applied to a real-world, high-impact writing task: science writing. This introduces challenges different to those in traditional creative writing tasks, such as writing stories and poetry, which tend to deal with common objects and relations. Science writing support requires a system to demonstrate proficiency within an area of expertise. We structure our work around the following research question:

> *RQ: How can language model outputs support writers in a creative but constrained writing task?*

As a test-bed, we use a science writing form called "tweetorials" [7]. Tweetorials are short, technical explanations of around 500 words written on Twitter for a general audience; they have a low-barrier to entry and are gaining popularity as a science writing medium [53]. We present a system that aims to inspire domain experts when writing tweetorials on a topic of their expertise. This system provides what we call "sparks": sentences intended to spark ideas in the writer. Our system generates sparks using a mid-sized language model (GPT-2 [43]) and a custom decoding method to encourage specific and diverse outputs.

We run two evaluations. In the first study, we compare the outputs from our custom decoding method to a competitive baseline as well as to a human-written gold standard, reporting on the diversity and coherence of all outputs. In our second study, we have 13 graduate students from five STEM disciplines write tweetorials with our system and report on how they thought about and made use of the sparks.

We make the following contributions:

- a system that generates "sparks" related to a scientific concept, including a custom decoding method for generating sparks from a pre-trained language model;
- an evaluation demonstrating that the sparks are more coherent and diverse than an off-the-shelf system, and approach a human gold standard; and
- a user study with 13 graduate students showing three main use cases of sparks and corresponding interaction patterns, as well as an analysis on how spark quality relates to participant satisfaction.

We end by discussing what might be driving user satisfaction in human-AI collaboration, how our results relate to concerns of plagiarism and bias in language models, and future directions for studying human-AI collaboration.

2 RELATED WORK

2.1 Natural Language Generation

A language model is any model that predicts the likelihood of a sequence of words. This can be used to generate text by giving the model a prefix and having it calculate the likelihood of each word in its vocabulary as the next word. This probability distribution can be used to select the next word, and thus generate text [29]. Language models are getting larger: they are being trained on more text and the models have more parameters [1, 8, 43]. It is useful to be able to take one language model and use it for many tasks, rather than having to train a new model for each task. Much recent work has been done on how to make the best use of these large language models, which have shown to be much more general purpose than previous ones [31], even showing promise in generating code [4].

It has been shown that a well-selected prefix, or 'prompt', can dramatically increase the performance of a language model on a specific task [45]. A resulting line of research has looked at how to search for or train prompts, and Li et. al. provide an excellent survey of this emerging field [35]. A useful distinction made in the survey is between discrete prompts, which are natural language prompts that read like normal text, and continuous prompts, which operate over the vector space of the language model. Continuous prompts have outperformed discrete prompts for GPT3 and BERT in some settings, suggesting that continuous prompts may produce better outcomes even if natural language prompts are more intuitive [37]. These results however are highly dependent upon many factors[1] [35]. Given that there is no clear leading method to find an optimal discrete or continuous prompt, we chose to hand-craft discrete prompts to be as intuitive as possible to the user, in the interest of establishing trust and promoting easier interaction.

Despite the successes of these models, problems remain. Language models tend to output repetitive and vague responses [26, 28]. They have no model of the truth; they are learning correlations from large amounts of text and thus are able to generate falsehoods. Finally, it has been well-documented that these models can generate offensive language, have distributional biases, and may copy text from the training data [5, 42]. In light of such issues, in this work we frame language model outputs as sources of inspiration for domain experts, rather than agents capable of completing a task independently or with minimal oversight.

[1] e.g. The directionality of the language model (unidirectional vs. bidirectional), the scale of the model, the method of selecting or training the prompt, the kind of training data utilized, and the type of downstream task.

2.2 Generative Writing Support

Technological writing support has a long history, but it has seen an increase in attention as language models have improved. Early work on language models for creative writing focused on activities such as storytelling [46] and metaphor writing [11]. While these tools proved helpful for writers, they were narrow in what they could provide. An early exploratory study found that auto-complete from a language model did not provide enough control for novelists [10]. More recent work has varied the ways in which technology can support the writer, for instance by providing description, plot points, or even asking questions, depending on the desires of the writer [3, 15]. Singh et al. [52] investigates a multimodal system for story writing, that includes language model suggestions, and discusses the *integrative leaps* writers make when incorporating suggestions. Lee et al. [33] presents a large dataset of how writers incorporate suggestions from GPT-3 in response to creative writing and argumentative writing prompts, quantifying measures like how many times suggestions with named entities were incorporated and how mutual turntaking was during the writing process.

Writing support designed explicitly for nonfiction writing tasks tends to be much more constrained, for instance Gmail's Smart Compose sentence completion and Smart Reply suggestions [12, 30]. Although these tools intend to suggest only text that the writer would have written in anyway, it has been shown that even these suggestions can change what people write [2]. Other work, on helping people craft responses to those in mental health crisis, focuses on providing writers feedback and suggested words, rather than complete phrases or sentences [41].

While language generation is a large field, few of its technologies are studied in the context of how they will be used by writers. For instance, although there is much work on automatic summarization [25, 61], there's less work on how the summaries might be used by people. Our work aims to study how text generated by language models might be used by writers in a science writing task. There is relation to a natural language generation task like summarization, because we are concerned with specific factual information (as opposed to commonsense knowledge) but we take a human centered approach where the language model provides suggestions, rather than a completed output.

2.3 Science Communication on Social Media

Science communication helps the public understand scientific contributions. It has been applied to vaccine misinformation [51], the COVID-19 pandemic [60], and climate change [24]. Traditionally, science communication took place through journals, conferences, articles, books, television and radio—places where peer review or editorial oversight was an implicit part of the publication process. However, the rise of digital networks and the ubiquity of social media presents opportunities for scientists to have direct channels to the public. Now any scientist can conduct science communication by posting about their work online [53], engaging in the 'Ask' communities on Reddit [22] or explaining a topic on Youtube [57].

This emerging trend, where the scientist can now partake in conversations outside of a gated process, reflects one of the many broad shifts away from traditional science communication. Scholars have reified this emerging form of communication as "post-normal science communication" [9]. Defining characteristics of post-normal science communication include a tolerance for subjectivity, an insertion of the self, the integration of advocacy, and call to actions. Despite these dramatic shifts, the original tenets of science communication such as storytelling, analogies, figures, and citations remain valuable, and storytelling in particular is a driving principle within our system. Our work engages with post-normal science communication by exploring how new technologies might help people partake in online science writing.

2.4 Expository and Narrative Theory

In studying how narratives are embedded in text, we turn to a rich body of literature about narratives and knowledge structure in semiotics and discourse theory. These domains inform our search for structures we could use to prompt language models. We looked at frameworks for both expository and narrative writing, because science writing is a hybrid of both. Specifically, we draw from the constructionist theory for narrative text, discourse theory for narrative text, and discourse theory for expository text. The constructionist framework of narratology states that all reading comprehension is "a search for meaning" [23]. Readers infer meaning as they build a mental model of why certain actions, events, and states are involved in a situation. The constructionist framework has a classification of inferences that we borrow for many of our prompt templates. Our prompts exemplify a subset of these classes such as case structure role assignment, causal antecedent, superordinate goals, and instantiation of a noun category. Concurrently, we examined expository text discourse theory for knowledge structures that would lend well to prompt templates. One framework for expository text introduced a taxonomy of methods (evaluation, explanation, occasion, and expansion) to enumerate different ways a writer can "influence the inference process of the reader" [54]. An alternative and popular framework from Meyer et. al. listed signal phrases that distinguish expository texts, such as 'specifically' or 'attributes of'. We chose to incorporate multiple signal phrases from Meyer's framework into our prompt templates [39].

3 FORMATIVE STUDY

In order to understand how a language model might best support the task of writing a tweetorial, we ran a formative study where participants were first given a technique for coming up with a compelling introduction, before being asked to write the first tweet of a tweetorial on a technical topic they were familiar with. Since the first tweet tends to set up the context and intention of the tweetorial [7] we expected this to be an effective and efficient way to understand what participants found difficult in the writing process, even when provided with writing strategies.[2]

[2]Initially we thought we could also run our final user study by asking participants to just write the first tweet, as we expected this to capture many of the creative aspects of tweetorial writing. However, a methodological finding was that writing the first tweet alone lacked some of the writing details we hoped to study as participants were not required to think through how they might actually continue from the structure they set up in the first tweet.

3.1 Methodology

We recruited 10 students from our institution's Computer Science department (6 women / 4 men; 7 undergraduates [no first years] / 3 PhD students). Participants went through a tutorial on how to write an engaging introduction on two example topics—recursion and virtual private networks—which included several examples and a step-by-step process for coming up with ideas. The tutorial was developed in consultation with a science writing instructor and presented the following process for writing an engaging first tweet: 1) brainstorm three concrete situations related to the topic, 2) turn each situation into a question for the reader, 3) select the most engaging question and revise.[3] The tutorial was intended to provide the participants with as much "unintelligent" support as possible, mimicking what would be taught in a graduate-level science writing class, such that we could identify where language models may be able to add benefit.

After the tutorial, participants were asked to select a topic from one of six Computer Science topics and write the first tweet for a tweetorial that would explain that topic.[4] Participants were asked to think aloud during the writing process and were not allowed to browse the web. Afterwards, they were asked a series of questions about their writing process in a semi-structured interview. The research team reviewed their writing with a science writing instructor. No formal coding was done, but general areas of success and areas for development were discussed.

3.2 Results

3.2.1 Participants reported that the task required creativity, and that it was difficult to come up with ideas. Although we never used the words 'creative' or 'creativity' when describing the task to participants, many participants reported that the task was difficult because it required creativity to come up with something that would engage the reader. Most participants said they don't typically do creative writing, so they found the task difficult and outside of their area of comfort. This supported our selection of tweetorials as a writing task, as we want to study a task that is both constrained and creative.

Participants found the tutorial helpful for a variety of reasons. Some liked seeing the examples, some appreciated a process to follow, and others found it comforting to see writing improve with brainstorming and revision. Several commented that the tutorial made the task look easy, but when they wrote about their own topic it was surprisingly difficult. 9 out of 10 participants said that making the topic interesting to a general audience was the most difficult part of the writing task. When pressed to be more specific, participants mentioned coming up with concrete examples/situations and creating an engaging question as hard tasks. Though this was influenced by the process the tutorial introduced, this confirmed that tutorials are not enough to fully support writers in this task.

3.2.2 Participants struggled to come up with ideas that created suspense. When reviewing what the participants had written, all the tweets mimicked the tone of the examples. However, the science writing instructor had critiques for all of them, and most of the critiques at the core were the same: the tweet lacked suspense. By this he meant, the tweet did not introduce a compelling problem or gap in the reader's understanding that would make the reader want to read more. Often this was because the example used wasn't particularly compelling or didn't reflect a real use case of the topic. Additionally, participants tended to repeat similar ideas to others who had selected the same topic.[5] Given that participants reported coming up with ideas difficult, it's likely that participants could have done better if given help with brainstorming.

We also noted that many of the tweets might be difficult to turn into tweetorials. For instance, some tweets engaged the reader with a question, but answering this question wouldn't require an explanation about the chosen topic. For this reason, in future studies we had participants write more than just the first tweet.

3.3 Design Goals

Based on our formative study, we developed two design goals:

(1) **Support writers with idea generation.** Given that language models have no model of truth, we want our system to come up with "sparks", intended to spark ideas in the writer, rather than having the system provide the ideas themselves. This aligns with prior work on creativity support tools, where users make use of system outputs as initial directions that are then interpreted and diverged from in the users' actual creation [20].[6]
(2) **Generate outputs that are coherent and diverse.** In order for writers to make use of outputs, even if they are not always perfectly accurate, they should be coherent: well-formed and generally reflecting accurate knowledge. Additionally, to support idea generation, outputs should also be diverse, such that writers have a variety of outputs to make use of.

4 SYSTEM DESIGN

4.1 Generating Sparks

4.1.1 Language model selection. To generate sparks we use GPT-2, an open source, mid-sized (1.5 billion parameters), transformer language model trained on 40GB of text from the web [43]. We use the huggingface implementation [58]. While larger open source models are available (though only to some),[7] we wanted to limit the size of the model we used as larger models are more expensive to run and take more time to generate text. Additionally, there have been many critiques of the super-large language models [5], and thus we wanted to use the smallest language model able to perform well for our use case. Anecdotally, we found that DistilGPT2, a 'distilled', smaller version of GPT-2 [48], was not able to produce coherent responses to our prompts. We experimented with fine-tuning GPT-2 on a small dataset of science writing, but found that

[3] The tutorials can be found at http://language-play.com/tech-tweets/tutorials.
[4] The topics were: hashing, sorting algorithms, Bayes theorem, HTTP, transistors, and Turning Machines. We selected these topics as ones that a) most computer science students should have learned in a formal setting, and b) could reasonably make for an interesting tweetorial.
[5] e.g. all the participants writing about HTTP used either Google or Twitter as their example, suggesting that people may converge on similar, easy to reach ideas.
[6] Additionally, this encourages the writer to feel more ownership over their final product, which has shown to be a concern in past work [41].
[7] For example, at the time this work was done, GPT-3 [8] was only accessible to those that had been granted access by OpenAI.

this made little difference, especially compared to modifying the decoding method or the prompts. For this reason most of our design effort focused on decoding and prompt engineering.

4.1.2 Decoding method. In addition to selecting a model, we had to design a decoding method—how to select the next token given the probability distribution the model outputs. There are several common ways of decoding from language models: greedy search, beam search, top-k sampling [26], and top-n sampling [18], to name a few. Different methods have different strengths and weaknesses. Greedy search, which selects the most likely word at each generation step, is rarely used for creative text generation as it tends to produce very generic responses (and rarely finds the most likely sequence of words). In contrast, beam search, which maintains a 'beam' of n possible outputs, can find more likely sequences and tends to produce high quality results [38]. When trying to generate multiple possible outputs for the same prompt, sampling methods, where words are sampled from the language model according to their likelihood, are often used. However this often decreases the coherence of the outputs, because very unlikely words can now be generated with some (albeit small) probability. For the purposes of having multiple unique sparks for our task, we designed a method that attempts to further increase the coherence of beam search while also increasing its ability to generate diverse outputs.

First, we modify the probability distribution using a normalized inverse word frequency, in order to increase the likelihood of infrequent words. Normalized inverse word frequency is often used in natural language generation to improve the specificity of outputs [32, 62], which is one method for increasing the overall quality of results. Here, we use normalized inverse word frequency purely during decoding as opposed to during training [21]. To calculate the word frequencies, we wanted a corpus that doesn't over-represent uncommon science words, like a science writing dataset might, but also reflects modern word usage. For these reasons, we use a corpus of Vox news articles that includes all articles published before March 2017.[8] Figure 2 shows an example of the probability distribution being modified. In this figure you can see that words like "governments", "Bitcoin", and "software" have increased weight, while words like "many", "both", and "all", are not modified.

Second, we use only the top 50 highest ranking tokens. This is sometimes called top-k sampling, as only the top k tokens are used [18]. However, since we're not using a sampling method, the effect of this is to ensure that the modified probability distribution doesn't introduce any incoherencies by dramatically increasing the rank of a token very far down in the original probability distributions. For example, Figure 2 shows that the probability of tokens related to 'cryptography' are dramatically increased; if this occurred when the token 'crypto' was ranked, say, 200th in the probability distribution, it may introduce incoherencies.

Third, we increase the diversity of outputs by forcing the first token of each output to be unique, but attempt to retain coherence by generating the rest of the tokens with beam search. While several more sophisticated methods have been proposed to increase diversity while retaining the coherence of beam search (e.g. [56]), in testing we found none were as effective as simply enforcing the first token to be unique.

[8]https://data.world/elenadata/vox-articles

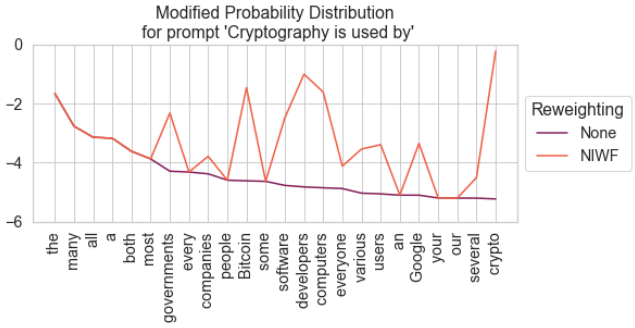

Figure 2: This graph shows how the likelihood of the 20 next most likely words given the prompt "cryptography is used by". The purple line shows the original probability distribution. The orange line shows the distribution after it has been rewieghted with normalized inverse word frequencies (NIWF). Words like "governments", "Bitcoin", "software", and "developers" have an increased probability, while words like "many", "both", and "all" are not modified.

Finally, in order to keep the sparks succinct and generating quickly, we only generate 10 tokens after the prompt and cut off the generation as soon as a sentence has been completed. We implement our decoding method using the huggingface transformers [58].[9] Step-by-step enumeration of the decoding process, and further development details, can be found in the Appendix.

4.1.3 Prompt design. We craft a 'prefix' prompt to pre-pend to any prompt used by a writer. Prefix prompts have been shown to greatly improve performance by providing the language model with appropriate context [45]. We found early on in development that simply providing the model with a technical topic was not enough—also providing a context area was necessary for it to appropriately interpret technical terms. For instance, if you use a prompt like "Natural language generation is used for", the model is likely to talk about linguistic research on languages, rather than computational methods. If instead you use the prompt, "Natural language generation, a topic in computer science, is used by" the results are much more likely to refer to computational language generation. Given this, we pre-pend all prompts with the following: "{topic} is an important topic in {context area}" where {topic} and {context area} are provided by the writer.

In hand-crafting our prompts, we wanted to make sure our prompts captured a range of relevant angles, so our system could flexibly work with any technical discipline. To do so, we synthesized work from expository and narrative theory into prompts capturing five categories: *expository, instantiation, goal, causal,* and *role*. Each category represented an angle that a writer might want to explore. All prompts can be seen in Table 1.

We manually developed these prompts according to established frameworks within narrative and expository theory that we referenced in our related work. Our prompts within the categories of *instantiation, goal, antecedent,* and *role* draw upon the constructionist framework of inferences, specifically the following categories:

[9]The repository with the code can be found at https://github.com/kgero/tech-tweets.

Table 1: Prompt templates designed for science writing task.

category	prompt
expository	One attribute of {topic} is Specifically, {topic} has qualities such as
instantiation	One application of {topic} in the real world is {topic} occurs in the real world when
goal	For instance, people use {topic} to {topic} is used for
causal	{topic} happen because For example, {topic} causes
role	{topic} is used by {topic} is studied by

case structure role assignment, causal antecedent, the presence of superordinate goals, and the instantiation of a noun category (respectively). Less formally, *instantiation* prompt templates suggest completions that instantiate where and in what ways topic X may occur in the real world. *Goals* prompt templates suggest completions that represent how topic X is used in the real world. *Causes* prompt templates suggest completions for how topic X might interact in cause and effect chains. *Roles* prompt templates cover entities involved with topic X. As tweetorials exhibit both elements of narrative and expository writing, we also borrowed signal phrases from Meyer's framework for expository text [39]—e.g. "specifically", "such as", "attribute"—and folded them within our prompt templates.

In testing we found that participants often wanted to 'follow up' on an output by entering in their own prompt. For this reason, we added the ability for writers to add their own prompts, though this prompt would also be pre-pended with our prefix.[10]

4.2 Interface

Figure 3 shows a screenshot of the system with its important features marked. The website consists of a single textbox for writing, and a 'prompt box' above it that allows writers to interact with the sparks. Writers can select a templated prompt from a dropdown menu, or type in their own prompt and add it to the dropdown list. When a prompt is selected, if they press 'GENERATE' the language model will generate a single spark. Writers can 'star' a spark by clicking on the lightbulb icon—this fills in the lightbulb and also pastes the spark into the textbox. If a writer selects a different prompt, the sparks already generated are preserved such that if they return to a previous prompt their generated sparks will be shown again.

The writing area textbox contains some features useful for the tweetorial writing task. The textbox is split into two sections with a line of dashes. Above the line is reserved for brainstorming and notes, a feature writers requested and found useful during pilot studies. Below the line is the text area for the tweetorial writing. A word count for the writer's tweetorial draft is displayed at the top of the textbox, and a character count for each tweet (separated by line breaks and two forward slashes) is displayed to the left. Figure 4 shows these features with an example from our user study.

The website is implemented using Python 3.7 and the Flask web framework.[11]

5 STUDY 1: SPARK QUALITY

We wanted to evaluate how well the sparks in isolation (i.e. not in a writing task) met our design goals of generating coherent and diverse sparks. We also wanted to test how well the sparks could support a wide range of topics, and if certain prompts supported some topics better than others. To do so, we compared the sparks generated by the custom decoding method to a baseline system, as well as a human-written gold standard.

We have three hypotheses:

- H1: The custom decoding produces more coherent and diverse outputs than a baseline system, but less coherent and diverse outputs than a human-written gold standard.
- H2: The custom decoding performs consistently across many different topics.
- H3: There is significant variance across output quality in topic+prompt combinations.

5.1 Methodology

We wanted to evaluate the quality of ideas for a variety of topics. We selected three disciplines that have a glossary of terms page on Wikipedia, and that have been demonstrated to be a rich discipline for science writing on social media.[12] These disciplines were computer science, environmental science, and biology. For each discipline we randomly sampled 10 topics from their glossary of terms page. See the appendix for the full list of topics studied.

5.1.1 Collecting a human-written gold standard. We wanted to collect human responses to our prompts to represent a gold standard or upper limit on the quality of ideas these prompts can generate. To do this, we recruited 2-3 PhD or senior undergraduate students in each discipline and had them complete the same prompts the language model did. These students acted like 'perfect' language models, with access to relevant expertise and a human-level understanding of how to write high quality sentences. Each student was paid $20/hour for as long as it took them to finish the task.

We explained to them that the purpose of the prompts was to generate ideas to support an expert writing about the topic for a general audience. Each student had to complete 5 prompts per topic in 3 different ways and was told to make the completions for a given prompt+topic combination maximally different (to encourage diversity). They were also instructed to ensure their completions were accurate, given their understanding of the topic, and that they could reference the web if they needed to check anything, as well as use web search results for inspiration. Finally, we explained that their ideas should be as concrete and specific as possible. Each

[10] One intriguing area of research is 'meta-prompting' [45] or 'chaining' [59], where the language model is used to generate the prefix for the next generation. While we found that this produced intriguing results for our use case, for example by having the model first produce a list of types of people who interact with a topic, and then putting those phrases into a downstream template, we thought it added too much complexity.

[11] A demo can be found at http://language-play.com/tech-tweets/enter-topic

[12] e.g. https://twitter.com/dannydiekroeger/status/1281100866871648256, https://twitter.com/GeneticJen/status/897153589193441281, and https://twitter.com/meehancrist/status/1197527975379505152

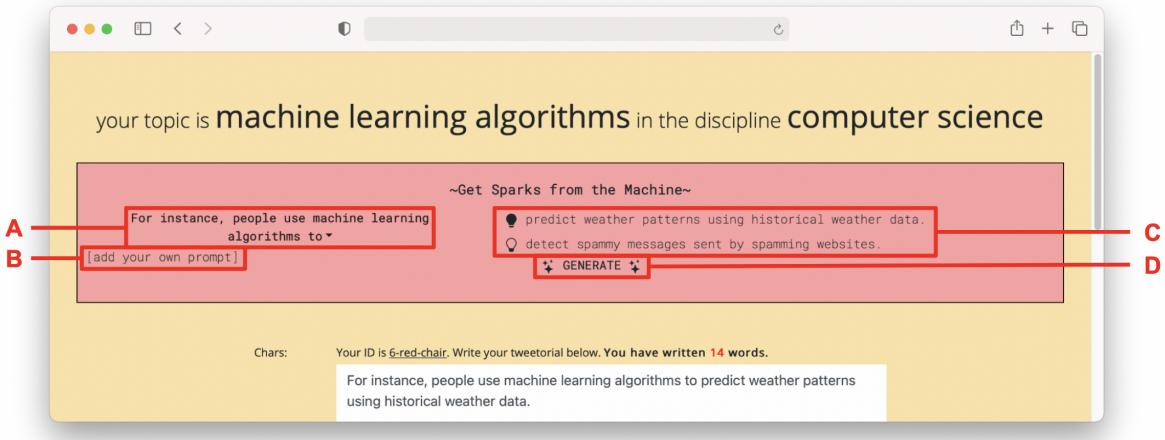

Figure 3: Example screenshot of our system that generates sparks. A: writers can select from 10 templates of prompts in a drop-down menu. B: writers can add their own prompt to the drop-down menu. C: sparks are generated with a lightbulb icon to the left; if writers click the lightbulb it will highlight and the spark is copied into the text area. D: writers can press the generate button in order to generate a new spark.

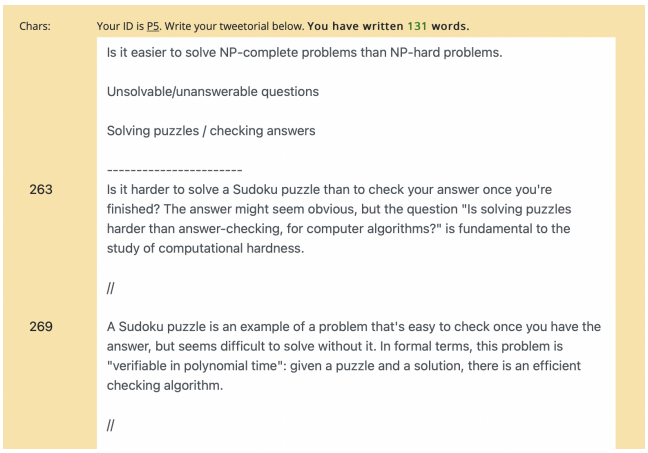

Figure 4: Screenshot of the text area from our user study. At the top is a word count, which counts only the words below the dashed line. Text above the dashed line is interpreted as brainstorming or notes. Participants can separate tweets with a double '//', and the character count for each tweet is shown to its left.

student completed 5 prompts for the 10 topics in their discipline, for a total of 5 x 10 x 3 = 150 completions per person. It took them on average 3.5 hours to come up with completions for all 10 topics in their discipline, and in the end we had 6 high quality completions per prompt+topic combination.

5.1.2 Baseline language model condition. We compare the custom decoding to a language model baseline: group beam search with hamming diversity penalty. This is a strong baseline that encourages diversity in the way Vijayakumar et al. [56] recommends, and can be implemented using arguments in the 'generate' function in the huggingface transformer library. Both the custom decoding and baseline model use the same underlying language model.

5.1.3 Measuring coherence and diversity. Coherence is notoriously difficult to measure automatically; measures like perplexity measure an output's likelihood under the model itself. For this reason we recruited domain experts to annotate outputs for coherence on a 0 - 4 scale, in line with knowledge graph evaluations [34]: 0 ("Doesn't make sense"), 1 ("Not true"), 2 ("Opinion/Don't know"), 3 ("Sometimes true"), and 4 ("Generally true").[13] For biology, we had 3 senior undergraduate students majoring in biology; for environmental science, we had 2 senior undergraduate students majoring in environmental science; for computer science, we had 2 PhD students from the computer science department.[14] Each discipline had 900 sentences to annotate (300 human generated, 300 from the baseline model, and 300 from the custom decoding). 250 randomly selected outputs from each discipline were annotated by two different annotators, and the Cohen's weighted kappa was calculated as: $\kappa = .54$ for biology, $\kappa = .51$ for environmental science, and $\kappa = .46$ for computer science. Given that the agreement was moderate, we had a single annotation for the remaining sentences. We also want to measure diversity, that is, for a set of outputs for a given prompt, how different are they from each other? Redundant or too similar outputs do not contribute new ideas to writers. We measure diversity with sentence embeddings specifically designed to elicit semantically meaningful cosine-similarities [44], by reporting the average distance between outputs within a given prompt. A higher average distance means that outputs are more dissimilar from each other, and therefore more diverse.

[13]This measures both coherence and cohesion, to lessen the load on annotators.
[14]The students could not have also participated in the generation portion.

Table 2: Example outputs from our three conditions for a single prompt+topic combination, and the average coherence (coh) and diversity (div) scores for each set of three outputs.

condition	coh	div	One attribute of source code is...
human	4	.38	it is typically written in a human-readable format. editability, so that programmers can easily change it to suit their needs. it is a description a computer program.
custom	4	.37	that it contains code written by humans. its modularity - code modules contain reusable code components. complexity.
baseline	2.6	.08	that it can be used as a source of information. that it can be used as a source of inspiration. its modularity.

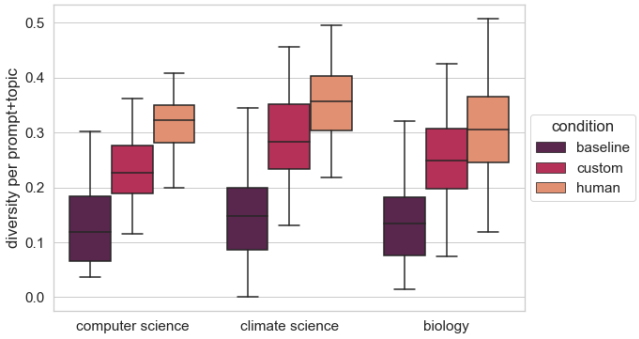

Figure 5: Distribution of diversity, split by discipline. Diversity is measured as the average sentence embedding distance per prompt+topic combination.

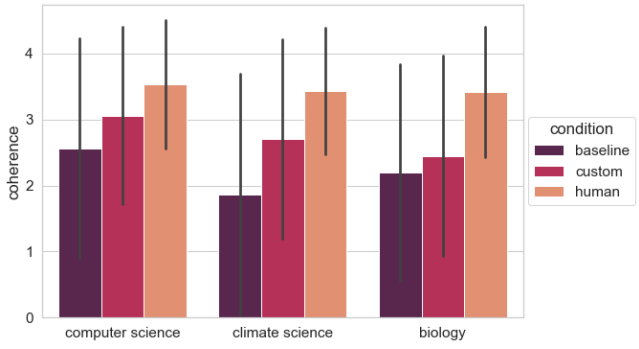

Figure 6: Mean coherence per prompt+topic combination with 95% confidence intervals, split by discipline. Each prompt completion was scored by a domain expert on a scale of 0 to 4.

5.2 Results

We confirm H1, finding that our system outputs are more coherent and diverse than the baseline, and approach a human-written gold standard. Figure 5 and Figure 6 show that the custom decoding method outperforms the baseline, but does not reach the performance of the human-written outputs. We perform two comparisons for each discipline—custom v. baseline and custom v. human—for a total of six null hypotheses per measure (diversity and coherence). For diversity, as we have normally distributed continuous data, we use two-tailed t-tests, and for coherence, as we have ordinal data, we use Mann-Whitney U tests. For each measure, we apply a Bonferroni correction ($m = 6$). We find a significant difference ($p < .001$) for all comparisons. Table 2 shows some example outputs from each conditions for a single prompt+topic. These examples demonstrate the quality of the human-written outputs: they are long, detailed, and diverse. Comparatively both language model methods are shorter, less specific, and more repetitive. However, the custom method improves the quality of the outputs.

It is important to acknowledge that the variation in both the diversity and coherence measures are quite large. This means that while on average the custom decoding is an improvement over the baseline, for any given prompt+topic combination the output could be very high quality or of a much lower quality. People using the system will not necessarily see this huge variation; they will only see the 10 or so model outputs that they generate.

We do not confirm H2, that the custom decoding performs consistently across many different topics. Figure 7 plots the average coherence for each topic with the black dots, and the coherence for each prompt+topic combination in the colored dots. From this we can see the variation in quality over the topics for the custom decoding method. For instance, the "computer security" outputs score an average of 3.7 in coherence, while "automata theory" outputs score 2.1. When looking at the human-created outputs, the quality is far more consistent, with no topics dropping below an average of 3 in coherence. This demonstrates that our system works well for some topics and less well for others. While we expected that our system would not perform as well as a human would, we did expect that the system would perform more consistently across topics. It is unclear why the language model performs significantly better on some topics, and given the way that these language models are trained it is difficult to inspect or even predict how well the model will perform on a given topic.

However, we do confirm H3, that output quality varies across topic+prompt combinations. Figure 7 shows that some prompt templates work better for some topics than others. In the human-written outputs, the variation is smaller, but still we see some range. For instance, let's look at the topic "dynein", the worst performing topic. The prompt "Dynein happens because" scores an almost 0 on the 0 to 4 coherence scale, while the prompt "One attribute of dynein is" scores a 3. Dynein is a family of proteins important in cell behavior. Owing to the nature of what dynein is, it makes sense that the system is more coherent on attributes of dynein, rather than why dynein "happens". However, it's notable that the human outputs scored 3 or above for all prompts for "dynein". Here is a human output about why dynein happens: "Dynein happens because organelles, such as the Golgi complex,

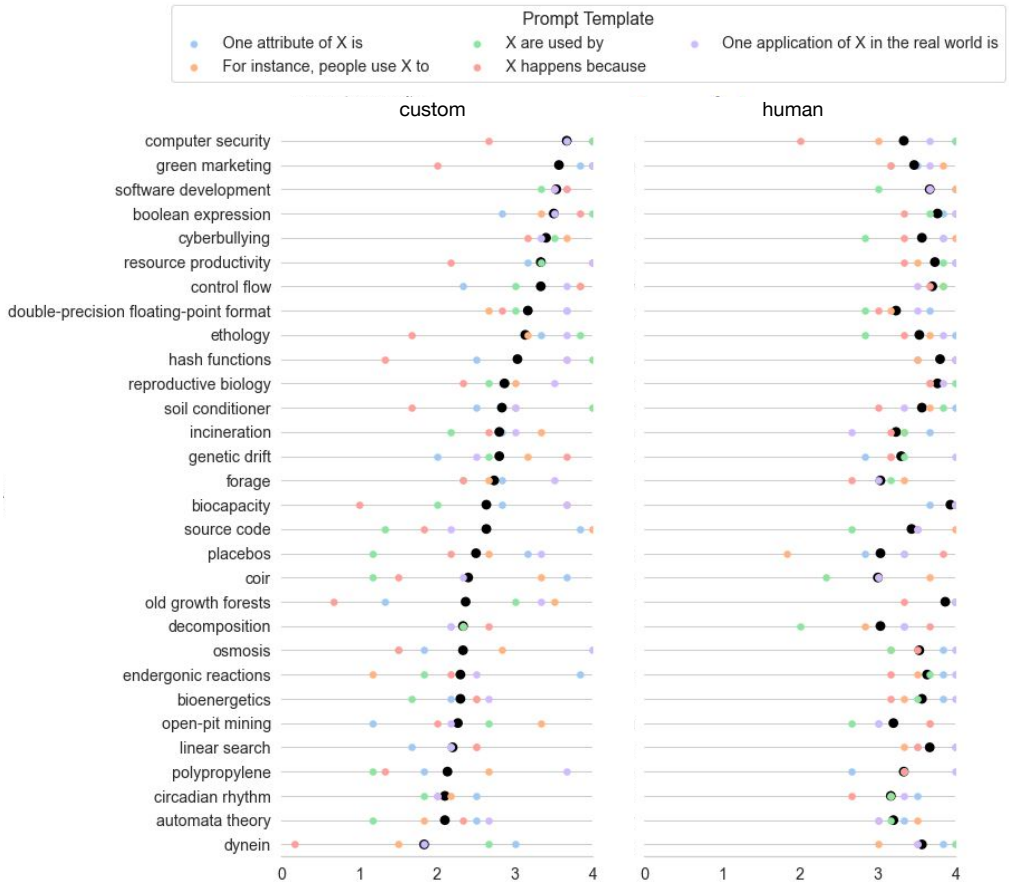

Figure 7: This graph shows the coherence per topic for the custom decoding and the human-created gold standard, where 0 is nonsensical or untrue and 4 is generally true. The black dot shows the average coherence of all responses for a given topic, while the colored dots show the average coherence for a given topic per prompt. Topics are ordered by average coherence in the custom decoding. This graph shows that some topics perform much better than others with custom decoding, while the human outputs are generally high quality regardless of topic. It also shows that within a topic there can be a large variation between prompt templates.

need to be positioned in cells." This sentence structure is a little convoluted, but it's clear that the human was able to compensate for the prompt and still write something coherent and meaningful. This highlights the importance of using a prompt that works well for the topic. Since we wanted to test our system with unseen topics, we ensure that participants can add their own prompts in case the template prompts don't work well for their topic.

6 STUDY 2: USER EVALUATION

The results of Study 1 confirmed that our custom decoding method outperforms a baseline system and approaches a human-written gold standard. In this study we sought to understand how writers make use of sparks when writing, and how spark quality relates to this usage. In particular, we pose the following research questions:

RQ1: In what ways do writers make use of language model outputs?
RQ2: What attributes of language model outputs, if any, correlate with writer usage and satisfaction?

We ran a single condition study intended to stress-test our system with a variety of unseen topics and collect rich data (both quantitative and qualitative) across participant action, perception, and cognition. While we did not have a baseline condition, we asked participants to compare their experience using sparks to their general writing process. This study was approved by the relevant IRB.

6.1 Methodology

6.1.1 Task. We evaluated how our system supported graduate students in writing tweetorials. Participants were asked to write approximately the first 100 words (or about five tweets) of their tweetorial.[15]

6.1.2 Participants. We use graduate students as they are eager to participate in science writing [27] and many tweetorials are already

[15]In pilot studies, participants felt intimidated by having to complete a draft within a specified period of time. By having them write the first 100 words, they were able to fully scope out their tweetorial without feeling pressured to produce a complete draft.

Table 3: Participant demographics. Low = once a year or so. Med = Once a month or so. High = once a week or so.

ID	Discipline	Science Writing (general / twitter)	Topic	Context Area
P1	Climate Science	Low / Low	rainfall variability	climate science
P2	Climate Science	Low / Never	predicting climate change	climate science
P3	Climate Science	Never / High	sea level change	geophysics
P4	Climate Science	Low / Low	glacier retreat over the holocene	paleoclimate
P5	Computer Science	Low / Never	computationally hard problems	computer science
P6	Computer Science	Never / Never	pseudorandomness	theoretical computer science
P7	Political Science	Med / Med	document embeddings	natural language processing
P8	Psychology	Never / Low	regulatory fit	psychology
P9	Psychology	Low / Low	motivated impression updating	social psychology
P10	Public Health	Low / Low	measurement of sexism	sociology
P11	Public Health	Never / Never	logistic regression	epidemiology
P12	Public Health	Low / Never	deprivation indices	public health
P13	Public Health	Med / Med	threat multiplier	environmental health

written by graduate students, demonstrating that this is a writing task our participants may conceivably want to engage in on their own. We recruited 13 STEM graduate students to write a tweetorial on a topic related to their research, while making use of the Sparks system.[16] Information about all participants can be found in Table 3.

6.1.3 Procedure. The study was run remotely via video chat and screen sharing. Participants were first asked to read an introduction to tweetorials, which explained what tweetorials are and walked through an example tweetorial. They were then introduced to the system and watched a short video that demonstrated the system's features and showed an example use case of the system. Participants could ask clarifying questions to the facilitator.[17] This portion typically took 10 - 15 minutes. At this point the participant was asked to pick a topic to write about, as well as provide a 'context area' that would give context to their topic and aid the system to correctly interpret their topic. Then they were given 20 minutes to interact with the system and complete the writing task. Mouse clicks and key presses while the participant interacted with the system were collected, as well as all sparks generated.

After this, the participant filled out a short survey, which included the Creativity Support Index [13], and partook in a semi-structured interview with the facilitator. During the interviews, participants were asked questions about the usefulness of the system and how their experience differed from their typical writing process. They were encouraged to review what they had written / the sparks they had seen to ground their responses. The survey and interview questions can be found in the appendix. The entire study took about an hour and participants were compensated $40 USD.

[16] In pilot studies we found that participants did not want to write about a provided topic. Even though topics were selected to be relevant and well-known in their disciplines, participants stated they did not feel comfortable (some said knowledgeable, some said motivated) explaining the provided topic. To encourage a realistic, self-motivated writing scenario, participants in this study were asked to pick their own topic. This had the additional benefit of stress-testing the system on a variety of topics unseen by those involved with the design.

[17] If participants asked to learn more about how the system worked, the facilitator said that it was an algorithm that could generate text in response to a prompt, and that they could discuss the system further after they completed the writing task.

6.1.4 Analysis. Participant interviews were transcribed and the authors performed a thematic analysis [6] on the transcripts. The analysis centered on: how sparks were helpful or unhelpful, how writing with the system compared to their normal writing process, and ownership concerns in response to writing with a machine. Relevant quotes were selected from the transcripts and collated in a shared document, where the authors iteratively discussed and collected the quotes into emergent themes. Finally, all sparks seen by participants were collected and annotated for common computer-generated text errors: 'Grammar and Usage', 'Redundant', and 'Incoherent' [17]. These annotations were done by graduate students. The coherence and diversity of sparks seen by each participant was measured as in Study 1.

6.2 Results

We structure this results section around our two research questions, and then report on how participants felt sparks compared to existing tools like web searches, and the issues of ownership and agency when writing with a computational aid. Participants came from across five STEM disciplines and selected a wide variety of technical topics to write about (see Table 3). We found that participant demographics did not correlate with any of our measures.

6.2.1 RQ1: In what ways do writers make use of language model outputs? Of our 13 participants, nine spoke in great detail about the ways in which sparks helped them. The remaining four reported that they did not find the sparks helpful. To answer our first research question we focus on the nine participants who found the system useful. In a later section of the analysis, we will analyze factors that may explain why four participants did not find the sparks helpful. Participants made use of sparks in three distinct ways: for inspiration, translation, and perspective. We talk about each of these in detail. Table 4 shows examples of the three main use cases participants reported, which we also discuss in the text below.

First, five of the participants reported on using sparks to provide them with inspiration. This was our intended use case

Table 4: Results of thematic analysis on reasons sparks were helpful. We report the three main use cases. Italics added by researchers to highlight where sparks influenced participant writing.

Use Case	Example Usage and Quote
inspiration	spark: People care about glacier retreat over the holocene because *glaciers affect sea level rise.* what participant wrote: ...Second, *the glaciers in South America have had an outsized impact on sea level rise.* xxx% of the current sea level rise has actually be attributed to the retreat of glaciers in South America! ... quote: "My specialty is very specific and technical. And it's often hard to figure out how to spin things in ways that feel relevant to people who don't study this. Sea level rise is something that people would find relevant."
translation	spark: In sociology, a deprivation index measures *societal conditions affecting individuals' abilities to obtain goods.* what participant wrote: ...relative deprivation experienced by individuals relative to others. It can be defined as *societal conditions affecting individuals' ability to obtain goods*, poverty levels relative to medium household income, among other definitions. ... quote:"Most of the time it [the system] was articulating the ideas that were already in my head in a way that's short and concise."
perspective	spark: One attribute of measurement of sexism is *that measuring sexism involves measuring attitudes towards men versus.* what participant wrote: The researchers in my study wanted to answer the question: "Does the level of sexism somewhere *impact that area's rate of gender-based violence?*" quote: "That was helpful because the research that I do around sexism is not concerned with people's attitudes, and instead concerned about things like incomes or legal rights or education levels. And so I wouldn't have even thought to talk about like sexism as it relates to people's attitudes."

of sparks, and we call this the 'inspiration' use case. These participants noted that the sparks provided good angles for discussing or introducing their topic. Table 4 shows how P4 used a spark about 'sea level rise' to make their topic 'glacier retreat over the holocene' more interesting to the average reader. Similarly, P2 noted that a spark about 'weather prediction models' was a useful entry point to their research on 'predicting climate change'. They said, "that's something within my field that the general public might be more familiar with than what I actually do." P7, writing on 'document embeddings', said, "[the system] definitely generated multiple [ideas] that I could have written different tweetorials about."

Second, six of the participants reported using sparks to help them with translation by providing detailed sentences to start with. We call this the 'translation' use case as participants reported that the sparks helped them 'translate' an amorphous idea in their head into a sentence.[18] Participants discussed the difficulty of writing technical definitions or including technical details, and remarked that although the sparks were often showing them information they already knew well, it was much faster and easier to draw on language from the sparks than to write a sentence from scratch. Table 4 shows how P12 used a spark to write a detailed sentence on 'deprivation indices'. They said "that would have probably taken me three sentences to write, then I'd have to spend time editing it down. This is a lot quicker." P7, writing on 'document embeddings', described the utility in this way: "[the sparks] do a really good job of compressing exactly the types of things that I would be going on Wikipedia or Google to get."

Third, three of the participants reported that the sparks showed them external perspectives. We call this the 'perspective' use case, as the sparks showed participants how their reader may be thinking about their topic. Table 4 shows how the sparks helped P10, who was writing about measuring sexism. She noted that many of the sparks talked about sexist attitudes and while that certainly is an aspect of measuring sexism, it isn't the aspect that she actually studies and therefore that might be an assumption that she will have to address in her tweetorial. P5, writing about 'computationally hard problems', noted that the sparks contained some technical words like 'NP-completeness', which made him reflect on whether or not someone who decided to read his tweetorial may already have some knowledge about his topic. Interestingly, participants discussed sparks that were factually wrong or incorrect in their interpretation of the topic as being useful because the sparks alerted them to misconceptions their readers may hold.

We wanted to investigate how these three use cases correlated with participants' actual interaction with the system. To do this, we labeled each participant with a single use case, where participants who mentioned more than one use case were labeled based on the use case they said was the most prominent or that they discussed the most. This resulted in four participants for 'inspiration', three for 'translation', and two for 'perspective'. (The remaining four participants said sparks were not helpful.) We then looked at writing timelines for each participant, noting when they interacted with sparks. Figure 8 shows participant timelines grouped by this categorization. **The 'translation' use case corresponds to much back and forth between writing and interacting with sparks, whereas the 'inspiration' and 'perspective' use cases correspond to longer stretches of independent writing.** We note that participants who said the sparks were not helpful had quite varied interaction patterns, suggesting that interaction pattern alone is not enough to determine utility of a writing support tool.

[18] We borrow the term 'translate' from the cognitive process model of writing [19]

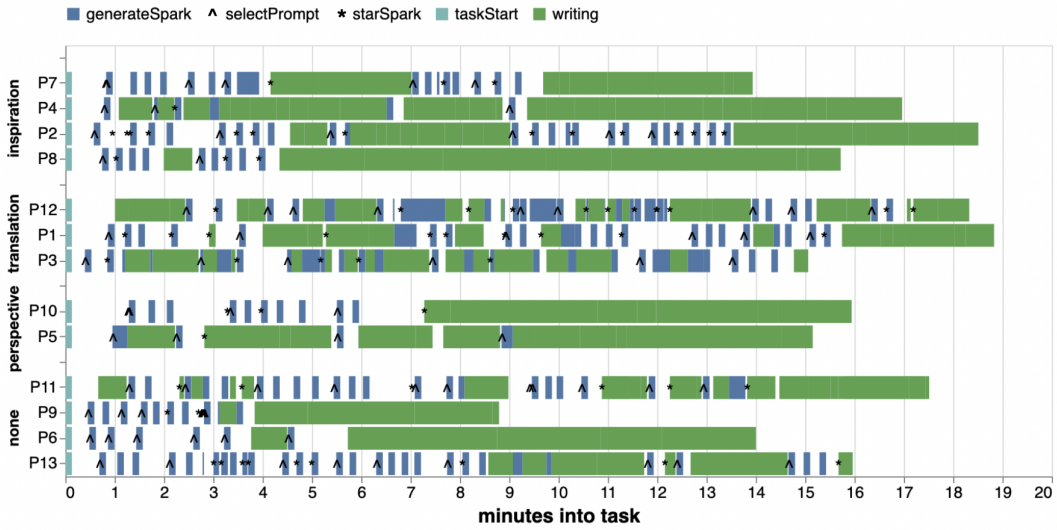

Figure 8: Timelines of all participants from the study, with time writing versus time generating sparks marked in different colors. Participants are grouped by their engagement pattern.

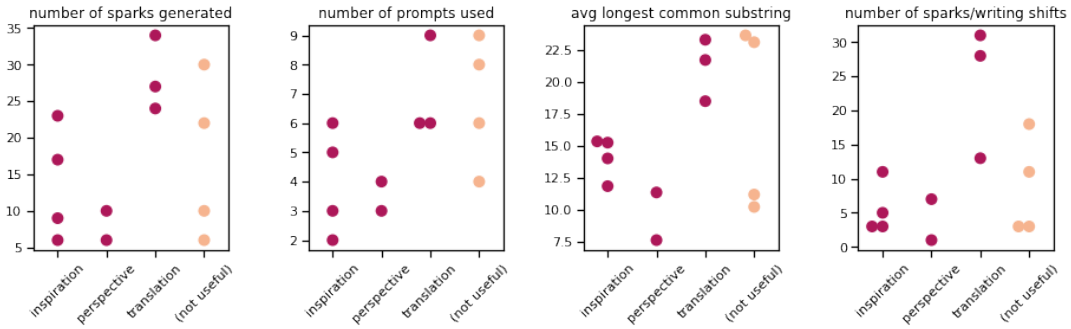

Figure 9: Four different measures of interaction, where participants are split by primary use case. Translation users generate more sparks, use more prompts, copy more from sparks, and shift between writing and generating sparks more than other users. There appear to be less differences between inspiration and perspective users.

In order to further examine how interaction patterns differ between use cases, we look at: 1) quantitative measures (the number of sparks generated), 2) temporal measures (the number of times a user swaps between generating sparks and writing), and 3) integrative measures (the average longest common substring between selected spark and what participant wrote). Figure 9 shows the results of these analyses. The 'translation' participants requested more sparks and used a higher variety of prompts to do so than others, suggesting that help with translation can occur more frequently throughout this setting of writing, or perhaps that the translation use case requires more sparks as part of its process. Interestingly, the number of starred sparks, as well as the percent of starred sparks (i.e. number of starred sparks divided by total sparks seen) is not noticeably different between the groups, suggesting that different use cases does not mean different levels of usefulness. We also see that 'translation' users moved back and forth between requesting sparks and writing more often than others; 'inspiration' and 'perspective' users tended to write for longer periods of time uninterrupted. Looking at how sparks were incorporated, 'translation' users tended to copy longer portions of sparks directly into their writing than 'inspiration' users. This analysis shows measurable interaction differences between the different use cases. In the next section, we analyze how the quality of sparks related to interaction patterns as well as participant satisfaction with the system.

6.2.2 RQ2: What attributes of language model outputs, if any, correlate with writer action and satisfaction? We look at the quality of individual sparks, as well as the aggregate quality seen per participant, and hold the following hypotheses:

- H1: Writers are more likely to star higher quality sparks.
  - H1a: Starred sparks have higher coherence than not-starred sparks.
  - H1b: Starred sparks have less errors than not-starred sparks.

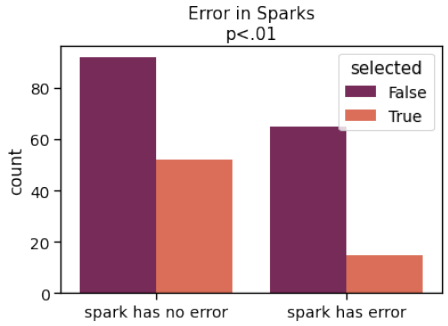

Figure 10: Sparks without any errors were significantly more likely to be selected ('starred') by participants than sparks with some kind of error.

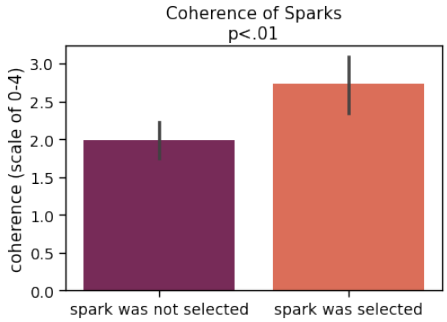

Figure 11: Sparks that were rated as being more coherent by expert annotators were significantly more likely to be selected ('starred') than sparks that were not selected.

- H2: Writers who see higher quality sparks are more likely to find the system useful.
  - H2a: Higher participant satisfaction is positively correlated with higher average spark coherence.
  - H2b: Higher participant satisfaction is positively correlated with higher spark diversity.
  - H2c: Higher participant satisfaction is negatively correlated with higher average error rate.

Of the 224 sparks seen by participants, 67 were starred, which amounts to 30% of sparks seen. Figure 10 looks at the error rate between sparks that were starred and those that were not. Due to sparsity in errors, we collate all the error categories, giving each spark a binary annotation of true or false for whether the spark contains any kind of error or not. Because of uneven sample sizes and the fact that we have a binary measure of error, we use a non-parametric test of proportions, the Fisher exact test, for significance. We find that sparks without errors are significantly more likely to be starred by participants ($p < .01$). Similarly, Figure 11 shows the results of the coherence annotation, looking at the coherence of sparks that where starred compared to those that were not. Because of uneven sample sizes, we use the Welch's t-test to test for significance, and we find that starred sparks have significantly higher coherence than those not starred ($p < .01$). **Thus we confirm H1: Writers are more likely to star higher quality sparks.**

To test our second set of hypotheses, that writers who see higher quality sparks are more likely to find the system useful, we look for correlations between our measures of spark quality (coherence, diversity, and error rate) and the results of the Creativity Support Index survey. We look at the individual creativity support measures (expressiveness, immersion, enjoyment, exploration, and results worth effort) as well as the aggregate measure, calculated as recommended by the creators of the index [13]. The aggregate measure nicely matches our interview data, where the four participants who reported that the system was not useful had the four lowest scores. We calculate the Pearson correlation coefficient and p-value to look for a linear relationship between variables and find no significant correlations. **We cannot confirm H2: Writers who see higher quality sparks are more likely to find the system useful.**

The interview data allows us to explore why spark quality may not correlate with usefulness. We can look at how participants who reported different levels of system usefulness responded to the same kind of error in different ways. Let's consider P10, P12, and P13. All are graduate students in the school of public health, and all commented that sometimes the sparks misinterpreted their topic. But P10 and P12 had some of the highest Creativity Support Index scores, and P13 had the lowest. P10 actually saw value in a spark we might consider low quality because it misinterpreted her topic but gave her additional perspectives she otherwise "would not have even thought to talk about". Describing the utility of sparks P10 said,

> There was a spark about measuring sexism by looking at people's attitudes towards women and men. And so that was helpful because the research that I do around sexism is not concerned with people's attitudes, and instead concerned about things like incomes, or legal rights or, education levels. And so I wouldn't have even thought to talk about like sexism as it relates to people's attitudes.

P12 thought of the spark as human error; they blamed low quality sparks on themselves. They also found the system very useful, and described seeing incorrect topic interpretations quite differently:

> It [the system] kept going to obesity. I think that's because of deprivation... So maybe I put the wrong field [context area]. Like, I could have said sociology instead of public health.

Whereas P13 described the same situation as a system error:

> I think it [the system] saw the word climate change and ... automatically went to the traditional climate change research about, sea level rising and stuff. And that wasn't at all what I was trying to write about.

These participants responded to the same error (misinterpreted topic) in different ways. **This suggests that other confounds, like participant attitudes, make spark quality insufficient as an explanation of perceived usefulness.**

Participants also responded differently to sparks which showed them things they already knew. Some participants found these sparks to be helpful. P1 writing about 'rainfall variability' and who had the highest Creativity Support Index score said, "I was impressed by the accuracy of most of them. [gives example spark]

That's awesome. Having that specificity in a concise sense was helpful, and more helpful than Wikipedia." On the other side, P9 writing about 'motivated impression updating' and who had the second lowest Creativity Support Index score, found these kinds of sparks to be useless: "I felt like it [the system] would be helpful to someone who doesn't know the topic. Not to someone who knows a lot about the topic."

6.2.3 How did the sparks compare to other resources like web searches? Participants tended to agree that the sparks were about as accurate as Googling, but they varied in whether the system was as useful. Figure 12 shows the results of our survey question about how sparks compared to what participants might find via Google. In the interviews participants were able to be more precise about how they perceived the differences. Some said that even though the sparks were not quite as good as Google, being able to stay in the context of writing and not be distracted by the results of a web search was more beneficial to them. Others found the sparks better than Google: P8, writing about 'regulatory fit', said that while Wikipedia is generally a good resource for older psychology concepts, it typically fails for more modern psychology research, whereas the sparks about her topic were correct some of the time.

But several participants mentioned simply feeling more 'in control' when using Google. P2 said, "It's probably just easier to navigate on Google because I'm more familiar with the phrasing and the patterns that will get me the results that I want." P9 said, "I feel like I have full control when I'm googling something over... where my brain wants to go and how I want to think of new ideas." P12 discussed trying different prompts and eventually giving up because they "could not get the prompts to give me that spark [I wanted]". These point to a potential learning curve of working with the sparks that participants were not able to overcome within their 20 minutes of writing.

6.2.4 Did participants have ownership concerns? Most participants had no ownership concerns about incorporating sparks. Several reported that because the system could never totally surprise them, they didn't feel like it had ownership over anything they wrote. Others said that since they are writing about public knowledge, it was unimportant where their ideas came from. One participant articulated that coming up with ideas is not the hardest part of science writing, but rather putting time and energy into building an audience and writing something engaging, so incorporating sparks would simply be one small part of a much larger endeavor that she took on. One participant compared the sparks to searching on Google (which they did all the time); another compared it to Grammarly (a grammar-checking service). One participant said that the sparks were simply elaborating on his own idea.

However, P9 talked about how he considered outreach and science writing to be part of his job as an academic, and thus any system that automated some aspect of this felt like it was taking over something that he found fundamental to his work. He said, "What this tool is accomplishing is an end in and of itself, right? Getting the opportunity to practice these things [idea generation for science writing] and organically generate them for myself is part of what it means to be an academic for me." P9 was also one of the participants who did not find the system useful.

While most participants had no ownership concerns, all participants expressed concerns about plagiarism. Several participants brought up that they were unsure exactly where the sparks were coming from, and they wanted to make sure that anything they took from the sparks was adequately changed, to alleviate any concerns about plagiarism. P2 described this as, "I think if I was using something like this, I would probably never use an entire sentence verbatim. Just because, if you don't know where it's pulling it from... I wouldn't want to run the risk of plagiarizing something accidentally even."

7 DISCUSSION

7.1 Why do some people find AI assistance more useful than others?

We found that there was no correlation between the average quality of sparks seen by a participant and how useful that participant found the system.[19] Several other studies of computational aids in a variety of domains have also found a high variation across participants in how useful a system is [10, 14, 20, 40]. In this section, we consider what else might be impacting perceived usefulness in human-AI collaboration.

The idea of an objective 'quality' of a system may be misleading. For example, presenting random words may seem like a reasonable baseline that more sophisticated systems can improve on. But randomness can be quite a strong baseline when it comes to creativity support. For instance, singer-songwriter David Bowie famously used random text generators when writing song lyrics; he used a tool called 'The Verbasizer', a computerized version of the cut-up technique which dates its history back to at least the Dadaists in the 1920s.[20] There exist today thriving communities—e.g. experimental writing, electronic literature—that draw upon surrealism and computation, and regularly makes use of randomness as a form of writing and/or writing support. For instance we can consider the contemporary work of John Cayley, Lillian-Yvonne Bertram, and Alison Parrish as grappling with the role of randomness in writing.

But what drives some writers to partake in this exploration? Probably since the beginning of time some creators have been seeking out inspiration in whatever form was available to them, and others have not. It might be that people's attitudes towards influence and inspiration has a large impact on their attitude toward a computational system, perhaps moreso than the quality of the system itself. This would explain why random suggestions can be seen as very useful by some, and factually correct generated sentences about a technical topic can be see as useless by others. Currently, it's unclear how people's openness to other kinds of influence, like random inspiration or ideas from a mentor or peer, relate to their openness to machine influence.

[19] This isn't to dismiss the impact of quality: within a participant, they preferred high quality sparks. And the proliferation of writing tools that make use of generated text [15, 45, 50] is likely due to the increased quality of generated text, and its correspondingly increased usefulness. But it seems like there are other confounding factors that complicate the relationship between system quality and perceived usefulness.

[20] Vice wrote an article about The Verbasizer in 2016: https://www.vice.com/en/article/xygxpn/the-verbasizer-was-david-bowies-1995-lyric-writing-mac-app and a modern version of the tool is available: https://verbasizer.com/

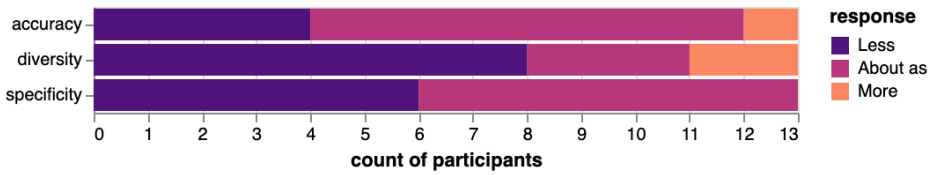

Figure 12: Most participants found the sparks about as or more accurate and specific than Google search. This was not true for diversity – most participants found the sparks to be less diverse than a Google search.

Expectations may also play a large role. One strength of large language models is writing sentences that we already kind of know—this is seen through the success of Gmail's Smart Compose, which seeks to only suggest extremely likely sentence completions [12]. But even that was divisive in our study, in which some participants appreciated sparks that detailed what they already knew (it helped them write concise, technical sentences more quickly and allowed them to stay in the context of writing) while others reported those kinds of sparks as useless. People may bring in expectations of the *kind* of help they are looking for, and dismiss anything that doesn't fit their model. Others may be more open, even looking for ways to find the system useful in the face of unexpected outputs.

Overall, we believe that participant attitudes are a major unknown factor when studying human-AI collaboration. Future work should investigate, or at least acknowledge, this confounding factor, as it complicates the seemingly simple question of 'how useful did you find the system'.

7.2 Providence and plagiarism as major writing concerns.

Most participants were worried about providence and plagiarism, bringing up these issues independent of any prompting from the interviewer. They didn't fully understand "where the sparks came from" and were worried that copying too much would be considered plagiarizing. This is not a concern we've seen reported in work on human-AI writing collaboration previously. Research on language models is attuned to if the model is copying from the data it is trained on [42], because that is viewed to be a sign of a low quality model and can result in data leakage from the training corpus. But even if we assume that the model is not copying from the source data, we may still need to ask the question of if it is okay—or even possible—to plagiarize from a language model.

Dehouche raises the ethical issue of plagiarizing from GPT3, stating that while language models have long been used for plagiarism *detection*, there needs to much more inquiry into plagiarizing *from* the model now that they can generate much more coherent, long-form text [16]. Dehouche argues that GPT3-generated text raises basic questions about authorship, because the author could be conceivably be the person who prompted and supervised text generation from the model, the computer scientists who developed and trained the model, the company offering access to the model, or the various anonymous authors whose text makes up the training data of the model.

In our setup participants would have struggled to plagiarize a whole tweetorial[21] yet they still raised these concerns. Historically plagiarism has assumed there is another scholar from which to steal words or ideas [47], but since authorship of text generated from language models is unclear, the issues of plagiarizing from them are unclear as well. Presumably the assumption of commercial writing support systems is that the writer is also the author of the generated text, thus removing any concern of plagiarism, but we saw that was not the assumption in our study. More work is needed to investigate this important question.

7.3 Bias in language models and the value of a biased perspective.

The bias of large language models is well-documented and a serious concern for anyone making use of this technology [5]. We selected the task of science writing as one where we expected there to be minimal issues of racism, sexism, and other kinds of prejudices brought up during the task. However, we still saw that the model was biased towards more prevalent topic associations. We saw this particularly in the case of sparks that misinterpreted a topic: these sparks were not wrong per se, but responded to the prompts with a viewpoint which sometimes differed from the participants' particular line of inquiry. Smaller language models trained on a more carefully curated dataset seem like a good solution to this problem, though it negates the utility of multi-purpose models.

Participants who reported these incorrect interpretations as useful introduce a novel use case of bias in language models more generally. If we acknowledge that language models are inherently biased based on their training data, we can start to envision how we might make use of that knowledge. For example, Schmitt and Buschek use chatbots as a way for story writers to develop characters, where writers progressively turn a bot into a specific character [49]. A biased language model is providing a specific perspective, and writers could make use of that perspective as a way to imagine their reader. Imagining your reader is an important and difficult part of writing [19]. What knowledge does your reader already have? Where will your reader get confused? When does your reader get bored? Great authors constantly consider these questions and adjust their writing accordingly.[22] Biased language models may be able to help writers model their reader, and help keep writers aware that any language model contains some kind of bias.

[21] though commercial systems like https://rytr.me/, https://researchai.co/ and https://www.sudowrite.com/ will happily spit out whole essays or stories

[22] Novelist George Saunders discusses this in an article for The Guardian: https://www.theguardian.com/books/2017/mar/04/what-writers-really-do-when-they-write

7.4 Limitations

Our system used a specific language model with a specific prompting method. Available language models are changing rapidly, as is the research on how to best make use of them. And while we picked our task to be representative of constrained and creative writing tasks, it differs greatly from other writing tasks people might be interested in like writing stories, academic papers, newspaper articles, or marketing copy.

Because we wanted our user study to closely mimic a realistic writing scenario, we had participants select their own topics. However, this introduced a large confounding factor, as different topics are more or less difficult to explain and make interesting, and different topics may elicit different levels of spark quality from the system, as seen in Study 1. One way we dealt with this confounding factor was by performing a single condition user study, as it didn't require us to control topics across conditions (and therefore across participants). This also allowed us to stress-test the system across many different, unseen topics. However, future work could benefit from comparative studies, either large-scale ones where participants can still pick their own topic but the size of the study minimizes topic as a factor, or smaller-scale ones where participants are assigned topics.

The small sample size of our study may have limited our ability to find significant correlations. Perhaps in larger studies we would find that the quality of system outputs *does* correlate with perceived usefulness. A hypothesis we hold, which would need to be tested, is that quality impacts perceived usefulness up to a point, after which increased quality has less impact than participant attitudes. We hope the results of our study inspire future work that can continue to explore how writers interact with language model outputs.

8 CONCLUSION

In this work we investigated how to use a large language model to support writers in the creative but constrained task of science writing. We developed a system that generates "sparks", sentences about a scientific concept intended to inspire writers. We found that our sparks were higher quality than a baseline system, and approached a human-written gold standard. In a user study with 13 STEM graduate students, we found that participants used the sparks in three main ways: as *inspiration* by providing ideas, to help with *translation* by providing detailed sentences, and by providing *perspectives* that helped them understand their reader. We also found that while participants preferred higher quality sparks, across participants average spark quality did not correlated with perceived usefulness. In the discussion we propose that participant attitudes towards writing may be influencing how they perceive system outputs, as well as discuss how designers might work with the inherent biases in large language models to develop more tools for writers.

ACKNOWLEDGMENTS

Thanks to Tim Requarth, who helped tremendously in the early days of this research thinking through what might actually be useful for science writing. This work was supported by the Brown Institute of Media Innovation, and NSF DGE - 1644869.

REFERENCES

[1] Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a Human-like Open-Domain Chatbot. *arXiv:2001.09977 [cs, stat]* (Feb. 2020). http://arxiv.org/abs/2001.09977 arXiv: 2001.09977.

[2] Kenneth C Arnold, Krysta Chauncey, and Krzysztof Z Gajos. [n.d.]. Sentiment Bias in Predictive Text Recommendations Results in Biased Writing. ([n. d.]), 8.

[3] Kenneth C Arnold, April M Volzer, and Noah G Madrid. [n.d.]. Generative Models can Help Writers without Writing for Them. ([n. d.]), 8.

[4] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program Synthesis with Large Language Models. *arXiv:2108.07732 [cs]* (Aug. 2021). http://arxiv.org/abs/2108.07732 arXiv: 2108.07732.

[5] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. ACM, Virtual Event Canada, 610–623. https://doi.org/10.1145/3442188.3445922

[6] Virginia Braun and Victoria Clarke. 2012. Thematic analysis. In *APA handbook of research methods in psychology, Vol 2: Research designs: Quantitative, qualitative, neuropsychological, and biological.*, Harris Cooper, Paul M. Camic, Debra L. Long, A. T. Panter, David Rindskopf, and Kenneth J. Sher (Eds.). American Psychological Association, Washington, 57–71. https://doi.org/10.1037/13620-004

[7] Anthony C. Breu. 2020. From Tweetstorm to Tweetorials: Threaded Tweets as a Tool for Medical Education and Knowledge Dissemination. *Seminars in Nephrology* 40, 3 (May 2020), 273–278. https://doi.org/10.1016/j.semnephrol.2020.04.005

[8] 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:2005.14165 [cs]* (July 2020). http://arxiv.org/abs/2005.14165 arXiv: 2005.14165.

[9] Michael Brüggemann, Ines Lörcher, and Stefanie Walter. 2020. Post-normal science communication: exploring the blurring boundaries of science and journalism. *Journal of Science Communication* 19, 03 (June 2020), A02. https://doi.org/10.22323/2.19030202

[10] Alex Calderwood, Vivian Qiu, Katy Ilonka Gero, and Lydia B Chilton. 2018. How Novelists Use Generative Language Models: An Exploratory User Study. In *23rd International Conference on Intelligent User Interfaces*. ACM.

[11] Tuhin Chakrabarty, Xurui Zhang, Smaranda Muresan, and Nanyun Peng. 2021. MERMAID: Metaphor Generation with Symbolism and Discriminative Decoding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 4250–4261. https://doi.org/10.18653/v1/2021.naacl-main.336

[12] Mia Xu Chen, Benjamin N. Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M. Dai, Zhifeng Chen, Timothy Sohn, and Yonghui Wu. 2019. Gmail Smart Compose: Real-Time Assisted Writing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, Anchorage AK USA, 2287–2295. https://doi.org/10.1145/3292500.3330723

[13] Erin Cherry and Celine Latulipe. 2014. Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. *ACM Transactions on Computer-Human Interaction* 21, 4 (Aug. 2014), 1–25. https://doi.org/10.1145/2617588

[14] 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*. ACM, Tokyo Japan, 329–340. https://doi.org/10.1145/3172944.3172983

[15] Andy Coenen, Luke Davis, Daphne Ippolito, Emily Reif, and Ann Yuan. 2021. Wordcraft: a Human-AI Collaborative Editor for Story Writing. *arXiv:2107.07430 [cs]* (July 2021). http://arxiv.org/abs/2107.07430 arXiv: 2107.07430.

[16] N Dehouche. 2021. Plagiarism in the age of massive Generative Pre-trained Transformers (GPT-3). *Ethics in Science and Environmental Politics* 21 (March 2021), 17–23. https://doi.org/10.3354/esep00195

[17] Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi. 2021. Scarecrow: A Framework for Scrutinizing Machine Text. *arXiv:2107.01294 [cs]* (July 2021). http://arxiv.org/abs/2107.01294 arXiv: 2107.01294.

[18] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. *arXiv:1805.04833 [cs]* (May 2018). http://arxiv.org/abs/1805.04833 arXiv: 1805.04833.

[19] Linda Flower and John R. Hayes. 1981. A Cognitive Process Theory of Writing. *College Composition and Communication* 32, 4 (Dec. 1981), 365. https://doi.org/10.2307/356600

[20] Katy Ilonka Gero and Lydia B. Chilton. 2019. Metaphoria: An Algorithmic Companion for Metaphor Creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–12. https://doi.org/10.1145/3290605.3300526

[21] Katy Ilonka Gero, Chris Kedzie, Savvas Petridis, and Lydia Chilton. 2021. Lightweight Decoding Strategies for Increasing Specificity. *arXiv preprint arXiv:2110.11850* (2021).

[22] Sarah A. Gilbert. 2020. "I run the world's largest historical outreach project and it's on a cesspool of a website." Moderating a Public Scholarship Site on Reddit: A Case Study of r/AskHistorians. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (May 2020), 1–27. https://doi.org/10.1145/3392822

[23] Arthur C. Graesser, Murray Singer, and Tom Trabasso. 1994. Constructing inferences during narrative text comprehension. *Psychological Review* 101, 3 (1994), 371–395. https://doi.org/10.1037/0033-295x.101.3.371

[24] P. Sol Hart and Erik C. Nisbet. 2012. Boomerang Effects in Science Communication: How Motivated Reasoning and Identity Cues Amplify Opinion Polarization About Climate Mitigation Policies. *Communication Research* 39, 6 (Dec. 2012), 701–723. https://doi.org/10.1177/0093650211416646

[25] Junxian He, Wojciech Kryściński, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2020. CTRLsum: Towards Generic Controllable Text Summarization. *arXiv:2012.04281 [cs]* (Dec. 2020). http://arxiv.org/abs/2012.04281 arXiv: 2012.04281.

[26] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. *arXiv:1904.09751 [cs]* (Feb. 2020). http://arxiv.org/abs/1904.09751 arXiv: 1904.09751.

[27] Emily L. Howell, Julia Nepper, Dominique Brossard, Michael A. Xenos, and Dietram A. Scheufele. 2019. Engagement present and future: Graduate student and faculty perceptions of social media and the role of the public in science engagement. *PLOS ONE* 14, 5 (May 2019), e0216274. https://doi.org/10.1371/journal.pone.0216274

[28] Daphne Ippolito, Reno Kriz, Maria Kustikova, João Sedoc, and Chris Callison-Burch. 2019. Comparison of Diverse Decoding Methods from Conditional Language Models. *arXiv:1906.06362 [cs]* (June 2019). http://arxiv.org/abs/1906.06362 arXiv: 1906.06362.

[29] Dan Jurafsky and James H. Martin. 2020. *Speech and Language Processing*.

[30] Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, Laszlo Lukacs, Marina Ganea, Peter Young, and Vivek Ramavajjala. 2016. Smart Reply: Automated Response Suggestion for Email. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, San Francisco California USA, 955–964. https://doi.org/10.1145/2939672.2939801

[31] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. *arXiv:1909.05858 [cs]* (Sept. 2019). http://arxiv.org/abs/1909.05858 arXiv: 1909.05858.

[32] Wei-Jen Ko, Greg Durrett, and Junyi Jessy Li. [n.d.]. Domain Agnostic Real-Valued Specificity Prediction. ([n. d.]), 8.

[33] Mina Lee, Percy Liang, and Qian Yang. 2022. CoAuthor: Designing a Human-AI Collaborative Writing Dataset for Exploring Language Model Capabilities. *CoRR* abs/2201.06796 (2022). arXiv:2201.06796 https://arxiv.org/abs/2201.06796

[34] Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense Knowledge Base Completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 1445–1455. https://doi.org/10.18653/v1/P16-1137

[35] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. *arXiv:2101.00190 [cs]* (Jan. 2021). http://arxiv.org/abs/2101.00190 arXiv: 2101.00190.

[36] Stephanie Lin, Jacob Hilton, and Owain Evans. [n.d.]. TruthfulQA: Measuring How Models Mimic Human Falsehoods. ([n. d.]), 13.

[37] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT Understands, Too. arXiv:2103.10385 [cs.CL]

[38] Clara Meister, Tim Vieira, and Ryan Cotterell. 2021. If beam search is the answer, what was the question? *arXiv:2010.02650 [cs]* (Jan. 2021). http://arxiv.org/abs/2010.02650 arXiv: 2010.02650.

[39] Bonnie J. F. Meyer and Melissa N. Ray. 2017. Structure strategy interventions: Increasing reading comprehension of expository text. *International Electronic Journal of Elementary Education* 4, 1 (Aug. 2017), 127–152. https://www.iejee.com/index.php/IEJEE/article/view/217

[40] Changhoon Oh, Jungwoo Song, Jinhan Choi, Seonghyeon Kim, Sungwoo Lee, and Bongwon Suh. 2018. I Lead, You Help but Only with Enough Details: Understanding User Experience of Co-Creation with Artificial Intelligence. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montreal QC Canada, 1–13. https://doi.org/10.1145/3173574.3174223

[41] Zhenhui Peng, Qingyu Guo, Ka Wing Tsang, and Xiaojuan Ma. 2020. Exploring the Effects of Technological Writing Assistance for Support Providers in Online Mental Health Community. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–15. https://doi.org/10.1145/3313831.3376695

[42] 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:2202.03286 [cs.CL]

[43] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. [n.d.]. Language Models are Unsupervised Multitask Learners. ([n. d.]), 24.

[44] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv:1908.10084 [cs]* (Aug. 2019). http://arxiv.org/abs/1908.10084 arXiv: 1908.10084.

[45] Laria Reynolds and Kyle McDonell. 2021. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. *arXiv:2102.07350 [cs]* (Feb. 2021). http://arxiv.org/abs/2102.07350 arXiv: 2102.07350.

[46] Melissa Roemmele. [n.d.]. Writing Stories with Help from Recurrent Neural Networks. ([n. d.]), 2.

[47] Ramin Sadeghi. 2019. The attitude of scholars has not changed towards plagiarism since the medieval period: Definition of plagiarism according to Shams-e-Qays, thirteenth-century Persian literary scientist. *Research Ethics* 15, 2 (April 2019), 1–3. https://doi.org/10.1177/1747016116654065

[48] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108 [cs]* (Feb. 2020). http://arxiv.org/abs/1910.01108 arXiv: 1910.01108.

[49] Oliver Schmitt and Daniel Buschek. 2021. CharacterChat: Supporting the Creation of Fictional Characters through Conversation and Progressive Manifestation with a Chatbot. In *Creativity and Cognition*. ACM, Virtual Event Italy, 1–10. https://doi.org/10.1145/3450741.3465253

[50] Hanieh Shakeri, Carman Neustaedter, and Steve DiPaola. 2021. *SAGA: Collaborative Storytelling with GPT-3*. Association for Computing Machinery, New York, NY, USA, 163–166. https://doi.org/10.1145/3462204.3481771

[51] Ashley Shelby and Karen Ernst. 2013. Story and science: How providers and parents can utilize storytelling to combat anti-vaccine misinformation. *Human Vaccines & Immunotherapeutics* 9, 8 (Aug. 2013), 1795–1801. https://doi.org/10.4161/hv.24828

[52] Nikhil Singh, Guillermo Bernal, Daria Savchenko, and Elena L. Glassman. 2022. Where to Hide a Stolen Elephant: Leaps in Creative Writing with Multimodal Machine Intelligence. *ACM Trans. Comput.-Hum. Interact.* (jan 2022). https://doi.org/10.1145/3511599 Just Accepted.

[53] Alice Soragni and Anirban Maitra. 2019. Of scientists and tweets. *Nature Reviews Cancer* 19, 9 (Sept. 2019), 479–480. https://doi.org/10.1038/s41568-019-0170-4

[54] Allen B. Tucker, Sergei Nirenburg, and Victor Raskin. 1986. Discourse and cohension in expository text. In *Proceedings of the 11th coference on Computational linguistics -*. Association for Computational Linguistics. https://doi.org/10.3115/991365.991419

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. [n.d.]. Attention is All you Need. ([n. d.]), 11.

[56] Ashwin K. Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models. *arXiv:1610.02424 [cs]* (Oct. 2018). http://arxiv.org/abs/1610.02424 arXiv: 1610.02424.

[57] Dustin J Welbourne and Will J Grant. [n.d.]. Science communication on YouTube: Factors that affect channel and video popularity. *Public Understanding of Science* ([n. d.]), 13.

[58] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[59] Tongshuang Wu, Michael Terry, and Carrie J. Cai. 2021. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. *arXiv:2110.01691 [cs]* (Oct. 2021). http://arxiv.org/abs/2110.01691 arXiv: 2110.01691.

[60] Kai-Cheng Yang, Francesco Pierri, Pik-Mai Hui, David Axelrod, Christopher Torres-Lugo, John Bryden, and Filippo Menczer. 2020. The COVID-19 Infodemic: Twitter versus Facebook. *arXiv:2012.09353 [cs]* (Dec. 2020). http://arxiv.org/abs/2012.09353 arXiv: 2012.09353.

[61] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *arXiv:1912.08777 [cs]* (July 2020). http://arxiv.org/abs/1912.08777 arXiv: 1912.08777.

[62] Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. [n.d.]. Learning to Control the Specificity in Neural Response Generation. ([n. d.]), 10.

A DETAILS OF SYSTEM DESIGN

A.1 Enumeration of Decoding Method

Let X be the prompt for the language model and Y be an output decoded from the language model given the prompt. Because we want multiple unique outputs from the same prompt, let Y^n be the n^{th} output decoded from the language model. Y^n is a sequence of tokens $(y_0^n, ..., y_i^n, ..., y_m^n)$; a partially decoded $Y_{0:i}^n$ would be $(y_0^n, ..., y_i^n)$.

At any given point in the generation process, let Z represent the set of all tokens in the vocabulary, such that any point we are selecting y_i from a probability distribution $P(Z|X + Y_{0:i-1})$.

Our decoding process is as follows:

(1) let y_0^n be the n^{th} most likely token in $P(Z|X)$
(2) while generating $Y_{1:m}^n$, select y_i^n from only the top 50 tokens in $P(Z|X + Y_{0:i-1})$
(3) while generating $Y_{1:m}^n$, modify $P(Z|X + Y_{0:i-1})$ with the normalized inverse word frequency (only top 50 tokens from unmodified distribution will be considered, as per step 2)
(4) perform beam search on the prefix $X + y_0^n$ with $k = 3$, selecting the top beam as the output

This decoding method was designed partially to be able to produce Y_n at any point, without having to generate $(Y^0, ..., Y^{n-1})$ or any $Y^{>n}$ outputs. This improves computation time, and while is common in sampling methods (where you can sample infinitely without requiring to know anything about previous samples) is not the case when using regular beam search to produce multiple outputs (where all n beams are generated at one time).

We also make use of two built-in huggingface functions for improving the quality of outputs. First, a small repetition penalty, modeled off of [31][23], where we set the repetition penalty to 1.2. Second, a blacklist that includes words that commonly derailed the output, like the word 'figure' which often resulted in an output like 'See figure 2 for more details'. Our blacklisted words were:

> one, two, three, four, five, six, seven, eight, nine, ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen, 'twenty, tens, hundreds, thousands, millions, Figure, figure, Fig, fig, Chapter, chapter

This decoding process was developed iteratively while testing the system with a variety of pilot users and test topics. We would regularly generate the top 10 responses to topics across computer science and biology to look for common failure points, like redundant responses, generic responses, incoherent responses, and factually false responses.

B DETAILS OF METHODS FOR STUDY 1

B.1 Full List of Topics Studied

- **Biology**: endergonic reactions, genetic drift, decomposition, dynein, circadian rhythm, placebos, ethology, osmosis, reproductive biology, bioenergetics.
  Topics randomly sampled from https://en.wikipedia.org/wiki/Glossary_of_biology.
- **Environmental science**: biocapacity, resource productivity, forage, polypropylene, open-pit mining, soil conditioner, incineration, green marketing, coir, old growth forests.
  Topics randomly sampled from https://en.wikipedia.org/wiki/Glossary_of_computer_science.
- **Computer science**: source code, automata theory, computer security, control flow, boolean expressions, double-precision floating-point format, linear search, software development, hash functions, cyberbullying.
  Topics randomly sampled from https://en.wikipedia.org/wiki/Glossary_of_environmental_science.

[23] And documented at https://huggingface.co/transformers/v4.6.0/internal/generation_utils.html#transformers.RepetitionPenaltyLogitsProcessor

C DETAILS OF METHODS FOR STUDY 2

C.1 Survey Questions

(1) What year of your graduate program are you in?
(2) What kind of graduate program are you in?
(3) What discipline do you study?
(4) How often do you write about technical topics for a general audience? e.g. blog posts, opinion articles, essays, etc.
(5) How often do you post on Twitter about technical topics?

C.2 Interview Questions

Questions about the task:

(1) Did you find any of the sparks helpful? If so, could you recall one spark that was helpful and explain in what way it helped? (Make sure to dig into how the spark related to what they eventually wrote. Ask them to point it out in what they wrote.)
(2) How do you think the sparks differed from what you would find on Wikipedia? How about Google search, or some other resource you use often?
(3) How did the existing prompts differ from your custom prompts?
(4) Could you recall one spark that wasn't helpful, and explain why?
(5) Were any of the sparks presented incorrect in some way? If so, what did you think of these?
(6) What made you decide to stop generating sparks?
(7) Did you have any concerns about ownership or agency?

Debriefing questions:

(1) Is there anything you'd like to share that I didn't ask about?
(2) Is there anything you'd like to know or ask me?