

# Sparks: Inspiration for Science Writing using Language Models

KATY ILONKA GERO, Columbia University, USA

VIVIAN LIU, Columbia University, USA

LYDIA B. CHILTON, Columbia University, USA

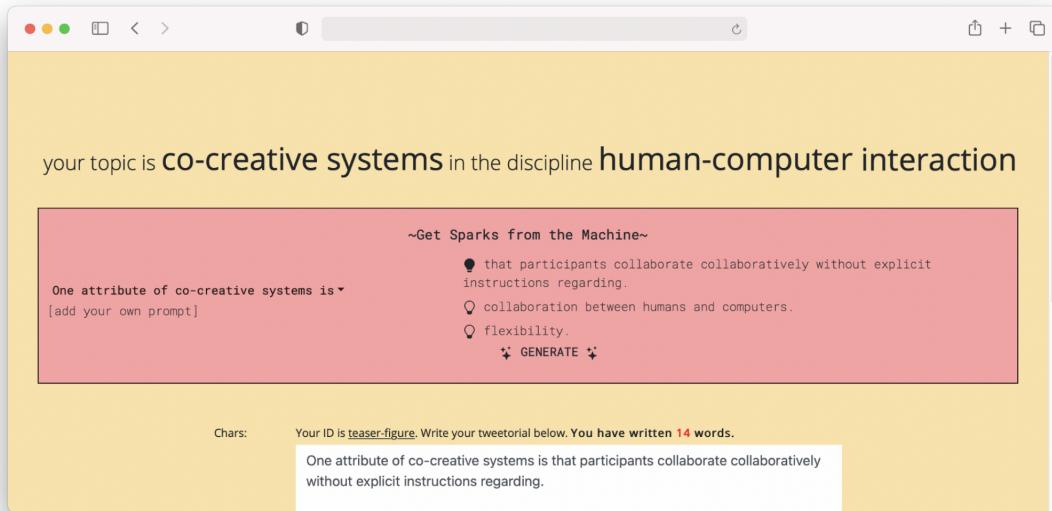


Fig. 1. An example screenshot of our system for the topic ‘co-creative systems’ in the discipline ‘human-computer interaction’. The system has generated three “sparks”: sentences intended to inspire the participant when writing an explanation for their topic. The first spark has been ‘starred’ as inspirational.

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-created gold standard. In a study with 13 PhD students writing on topics of their own selection, we find three main use cases of sparks: aiding with crafting detailed sentences, providing interesting angles to engage readers, and demonstrating common reader perspectives. We also report on the various reasons sparks were considered unhelpful, and discuss how we might improve language models as writing support tools.

---

Authors’ addresses: Katy Ilonka Gero, katy@cs.columbia.edu, Columbia University, New York, New York, USA; Vivian Liu, vivian@cs.columbia.edu, Columbia University, New York, New York, USA; Lydia B. Chilton, chilton@cs.columbia.edu, Columbia University, New York, New York, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

Manuscript submitted to ACM

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

Additional Key Words and Phrases: creativity support tools, writing support, co-creativity, science writing, natural language processing

#### **ACM Reference Format:**

Katy Ilonka Gero, Vivian Liu, and Lydia B. Chilton. 2021. Sparks: Inspiration for Science Writing using Language Models. 1, 1 (October 2021), 26 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 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 [3, 11, 50]. Their success has spurred improvements on many tasks, from classification to question answering to summarization [11], as well as creative writing support [17]. Language models have the potential to be powerful writing tools that can support writers in real-world, high-impact domains. These new 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 [35]. There remain many unknowns about how this technology will interface with people in real-world tasks, such as what interactions best serve writers, how language models can best contribute to different writing forms [13], and how to mitigate the bias that language models encode [8].

To that end, we study how language models can be applied to a real-world, high-impact writing task. In particular, we use a science writing form called “tweotorials” which explain technical concepts on Twitter for a general audience [10]. Tweotorials are short explanations of around 500 words which have a low-barrier to entry and are gaining popularity as a science writing medium [47]. Working on science writing requires a system to demonstrate proficiency within an area of expertise. This is much more difficult than traditional creative writing tasks, such as stories and poetry, which tend to deal with common objects and relations. Thus, we present the following research question:

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

In this paper, we present a system that aims to inspire domain experts when writing tweotorials 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 [39]) and a custom decoding method to encourage specific and diverse outputs. Additionally, we develop a set of 10 prompts based on narratology and expository theory that participants can use when interacting with the model.

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

We make the following contributions:

- a system that uses a language model to generate “sparks” related to a scientific concept;
- 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

- an exploratory study with 13 PhD students showing three main use cases of sparks.

We end by discussing how to best use language models in constrained writing tasks.

## 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 – either a sequence of words or a special ‘start of sequence’ word – and having it calculate the likelihood of all words in its vocabulary as the next word. This can be used to select the next word, and thus generate text. Language models are trained on a dataset of text that does not need to be annotated in any way, as the model can simply be trained on what the next word in a sentence will be [29].

Language models are getting larger: they are being trained on more text and the models have more parameters [3, 11, 39]. Much recent work has been 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 [6]. Additionally, 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.

It has been shown that a well-selected prefix, or ‘prompt’, can dramatically increase the performance of a language model on a specific task [41]. A resulting line of research has been automatically creating either natural language prompts or continuous vector prompts, to perform well on tasks [19, 34]. However, automatically learned prompts have yet to consistently outperform manually crafted prompts [19]. Additionally there have been very few studies on prompt selection for open ended, generative tasks [17].

Despite the successes of language models, problems remain. When considering the generation of text, language models tend to output repetitive and vague responses [26, 28]. Language models also have no model of the truth; they are learning correlations from large amounts of text. Thus they are able to produce text that includes falsehoods and offensive language [8].

### 2.2 Generative Writing Support

Technological writing support has a long history, but 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 [42] and metaphor writing [15]. While these tools proved helpful for writers, they were narrow in what they could provide. An exploratory study found that generic auto completion from a language model did not provide enough control for novelists [13]. More recent work on writing support for creative tasks 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 [5, 17].

Writing support for nonfiction writing tasks tends to be much more constrained, for instance as sentence completion [16]. A good example is Gmail’s smart reply function, which aims to suggest only text that the writer would have written in anyway [30], although it has been shown that even these suggestions can change what people write [4]. 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 [38].

While natural language generation is a large and growing 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, 57],

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's some relation to a natural language generation task like summarization, because we are concerned with real facts, 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 – consider how it has been applied to tackle vaccine misinformation [44], the COVID-19 pandemic [56], and climate change [24]. Traditionally, science communication took place through journals, conferences, articles, and books – places where peer review was an implicit part of the publication process. However, the rise of digital networks has made science establish a virtual presence through electronic journals and digital records. The ubiquity of social media further presented opportunities for scientists to have direct channels to the public. Now any scientist can conduct science communication online by posting about their work online [47], engaging in the ‘Ask’ communities on Reddit [22] or explaining something on Youtube [52]. Even PhD students or undergraduate researchers have the ability to disseminate their scientific knowledge at any time without depending on a venue or a publication.

This emerging trend, where the scientist can now partake in conversations outside of an implicitly gated, peer-review process, reflects one of the many broad shifts away from traditional science communication. Scholars of science communication have reified this emerging form of communication as “post-normal science communication” [12]. 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.

### 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 tweetorials are 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 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 from for many of our prompt templates. Our prompts exemplify a subset of these inference classes such as case structure role assignment, causal antecedent, the presence of 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 didactic methods (evaluation, explanation, occasion, and expansion) to enumerate the different conversational moves a writer can make to “influence the inference process of the reader” [48]. An alternative and popular framework was put forth by Meyer et. al., who enumerated 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 [37].

### 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 [10] we found this to be an effective and efficient way to understand what participants found difficult in the writing process, even when provided with writing strategies.

#### 3.1 Methodology

We recruited 10 students in Computer Science.<sup>1</sup> Participants were required to go through tutorials on how to write an engaging science writing 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. These tutorials focused on coming up with an intriguing question for the first tweet, and were developed in consultation with a science journalist. The process was: 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.<sup>2</sup> These tutorials were intended to provide the participants with as much “unintelligent” support as possible, such that we could identify where language models may be able to add benefit.

After the tutorials, 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.<sup>3</sup> Participants were asked to think aloud during the writing process. They were not allowed to browse the web. Afterwards, they were asked a series of questions about their writing process in a semi-structured interview.

After all participants had completed the study, the research team reviewed their writing with a science journalist. 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 didn't frame the task as a creative writing task, many participants described the task as 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 tutorials helpful, though for a variety of reasons. Some liked seeing the examples, others appreciated a process to follow, and still others found it comforting to see writing get better with brainstorming and revision. Several commented that the tutorials made the task look easy, but when they began to write about their own topic it was surprisingly difficult to come up with ideas.

Most participants (9 out of 10) 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 surely influenced by the process the tutorials introduced, this confirmed that tutorials are not enough to fully support writers in this task.

<sup>1</sup>6 women / 4 men; 7 undergraduates–no first years / 3 PhD students.

<sup>2</sup>Links to tutorials will be released after anonymous review.

<sup>3</sup>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 make for an interesting tweetorial.

**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 journalist 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. For instance, all the people writing about HTTP used either Google or Twitter as their example, suggesting that participants may converge on similar, easy to reach ideas.

Given that participants reported coming up with ideas difficult, it’s likely that participants could have done better if given help coming up with more ideas. Members of the research team also noted that many of the tweets written might be difficult to turn into full length tweetorials. For instance, if the question couldn’t really be answered with an explanation about their 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 for our system:

**3.3.1 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 [21]. Additionally, this also encourages the writer to feel more ownership over their final product, which has shown to be a concern in past work [38].

**3.3.2 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 [39]. We use the huggingface implementation [54]. While larger open source models are available, e.g. GPT-3 [11] or Megatron-LM [45], 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 [8], 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 [43], was not able to produce coherent responses to our prompts. We experimented with fine-tuning GPT-2 on a data set of science writing, but found that 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

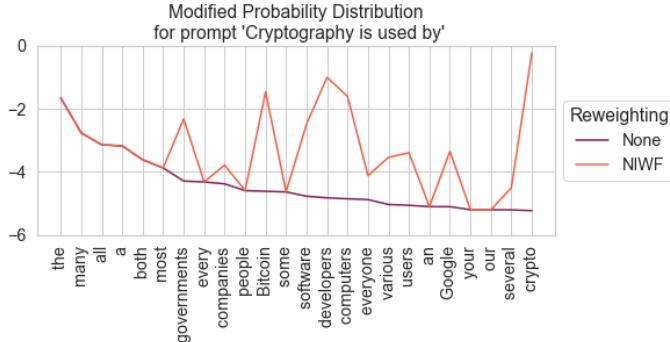


Fig. 2. This graph shows how the likelihood of the 20 next most likely words given the prompt "cryptography is used by". The orange line shows the distribution after it has been reweighted with normalized inverse word frequencies (NIWF). Words like "governments", "Bitcoin", "software", and "developers"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

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. Beam search tends to produce high quality results [36] but also tends to produce very similar results for the same prompt. Sampling methods can produce much more varied results, but at the cost of being less coherent. We designed a method that attempts to further increase the coherence of beam search while also increasing its diversity.

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, 58], which is one method for increasing the overall quality of results. To our knowledge this is the first work to use normalized inverse word frequency purely as a decoding method as opposed to during training. 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.<sup>4</sup> Figure 2 shows an example of the probability distribution being modified. In this figure you can see that words like "governments", "Bitcoin", "software", and "developers" have an 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, for example by dramatically increasing the rank of a token very far down in the original probability distributions.

Third, we increase the diversity of outputs by forcing the first token of each output to be unique, but attempt to retain coherence 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. [51]), in testing we found none were as effective as simply enforcing the first token to be unique.

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 [54].<sup>5</sup>

<sup>4</sup><https://data.world/elenadata/vox-articles>

<sup>5</sup>Link to code to be added after anonymous review.

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

**4.1.3 Prompt design.** Designing prompts for language models has become an active area of research, with many automatic methods being proposed [19, 34]. However, any automatic method requires at least some training data, and it's yet to be seen that automatically developed prompts can outperform hand-crafted prompts [19]. For these reasons, we hand-craft our prompts.

First 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 [41]. 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 were crafted based upon the constructionist framework of inferences, specifically the following categories: 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 [37] – e.g. “specifically”, “such as”, “attribute” – and folded them within our prompt templates.

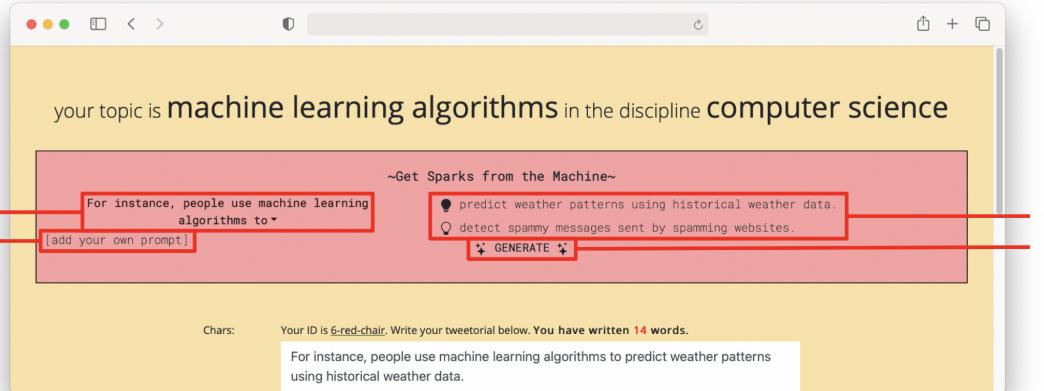


Fig. 3. Example screenshot of our system generates sparks. A: writers can select from 10 template of prompts in a drop-down menu. B: writers can add their own prompt to the drop-down menu. C: sparks are generated with a lightbulb icon to the left, if writers click the lightbulb it will highlight and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

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.<sup>6</sup>

#### 4.2 Interface

We design a website that takes in a writer’s topic and context area. 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 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.<sup>7</sup>

<sup>6</sup>One intriguing area of research is ‘meta-prompts’, where the language model is used to generate the prefix for the next generation [41]. 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.

<sup>7</sup>Link to demo to be released after anonymous review.

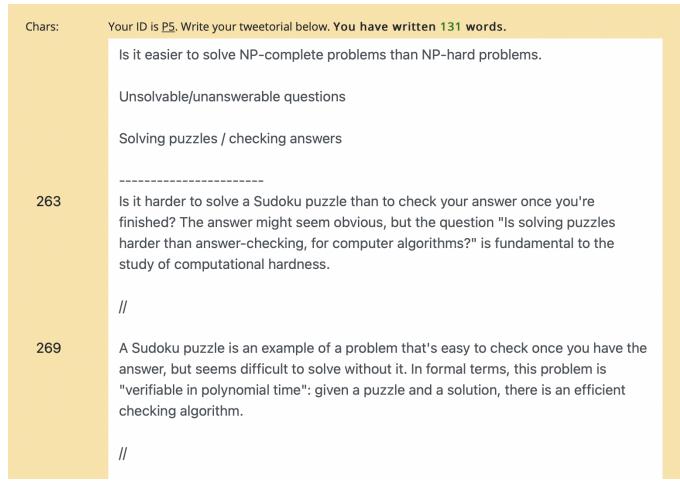


Fig. 4. Screenshot of the text area from our user study. At the top is a word count, which counts only the words below the dashed line. Text above the dashed line is interpreted as brainstorming or notes. Participants can separate tweets with a double ‘//’, and the character count for each tweet is shown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

## 5 STUDY 1: SPARK QUALITY

We wanted to evaluate the quality of the sparks generated by our system. In particular, we wanted to evaluate how well the sparks, in isolation (i.e. not in a writing task), met our design goals of coherent and diverse. 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-created 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-created gold standard.
- H2: The custom decoding performs consistently across many different topics.
- H3: Some prompts work better for some topics.

### 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.<sup>8</sup> 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-created 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. Each student was paid \$20/hour for as long as it took them to finish the task.

<sup>8</sup>e.g. <https://twitter.com/danniediekroeger/status/1281100866871648256>, <https://twitter.com/GeneticJen/status/897153589193441281>, and <https://twitter.com/meehanchrist/status/1197527975379505152>

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. 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 student completed 5 prompts for the 10 topics in their discipline, for a total of  $5 \times 10 \times 3 = 150$  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 [51] 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, especially without training data – measures like perplexity merely measure an output’s likelihood under the model itself. For this reason we recruited 10 domain experts to annotate outputs for coherence on a 0 - 4 scale, in line with knowledge graph evaluations [33]. 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.<sup>9</sup> 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 domain experts, and the Cohen’s weighted kappa was calculated as:  $\kappa = .54$  for biology,  $\kappa = .51$  for environmental science, and  $\kappa = .34$  for computer science. Given that the agreement was moderate, we had a single annotation for the remaining sentences.

We measure diversity with sentence embeddings [40], in particular we report the average distance between outputs within a given prompt. A higher average distance means that outputs are more dissimilar, and therefore more diverse.

## 5.2 Results

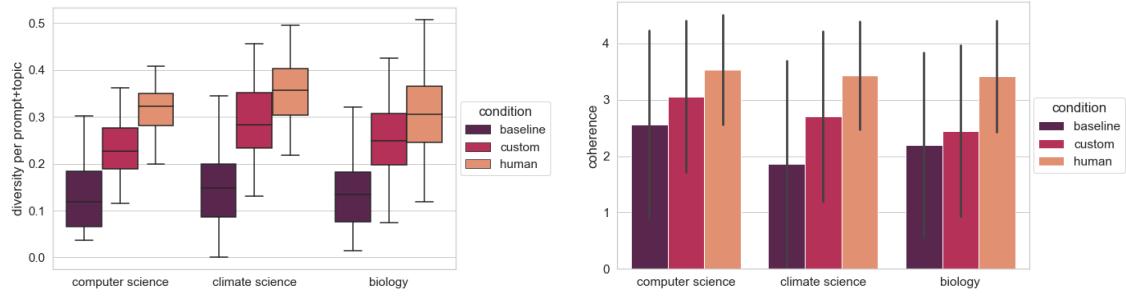
Overall, the baseline had low diversity and coherence across all disciplines, while the human-created outputs perform much better. Figure 5a and Figure 5b show that the custom decoding method outperforms the baseline, but does not reach the performance of the human-created outputs. For diversity, two-tailed t-tests show this to be a significant difference for all disciplines (computer science:  $p < .001$ , climate science  $p < .001$ , biology:  $p < .001$ ); for coherence, mann-whitney U tests show this to be a significant difference for all disciplines (computer science:  $p < .001$ , climate science  $p < .001$ , biology:  $p < .001$ ).

Table 2 shows some example outputs from each conditions for a single prompt+topic. These examples demonstrate the quality of the human generated outputs: they are long, detailed, and diverse. Comparatively both language model methods are shorter, less specific, and more repetitive. However, the custom method seems to improve the overall 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, and on average the

---

<sup>9</sup>The students could not have also participated in the generation portion.



(a) Distribution of diversity, split by discipline. Diversity is measured as the average sentence embedding distance per prompt+topic combination.

(b) Mean coherence per prompt+topic combination, split by discipline. Each prompt completion was scored by a domain expert on a scale of 0 to 4.

Fig. 5. Diversity and coherence measures across three test disciplines for three conditions: a baseline language model, a language model with the custom decoding, and a human-created gold standard. The custom decoding improves upon the baseline and approaches the human gold standard.

condition	coh	div	One attribute of source code is...
human created	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 decoding	4	.37	that it contains code written by humans. its modularity - code modules contain reusable code components. complexity.
baseline model	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.

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.

human-created outputs are better than the language model outputs, 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.

For this reason, we dig into the variation by topic and prompt. Figure 6 shows the average coherence per topic for the custom decoding method and the human-created outputs. It 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 that 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.

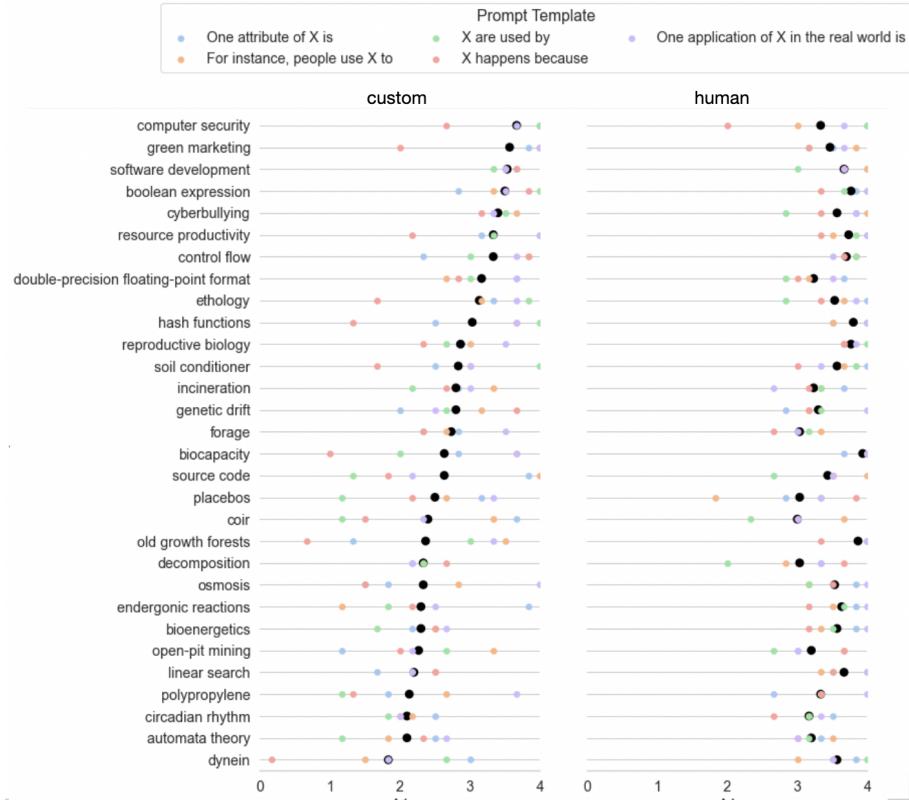


Fig. 6. This graph shows the coherence per topic for the custom decoding and the human-created gold standard, where 0 is nonsensical or untrue and 4 is generally true. The black dot shows the average coherence of all responses for a given topic, while the colored dots show the average coherence for a 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.

Figure 6 also shows that some prompt templates work better for some topics than others. In our system, the quality of outputs vary significantly with the prompt template. In the human generated 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. Given this, it makes sense that the system is more likely to produce coherent outputs on attributes, rather than why this family of proteins "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, need to be positioned in cells." Though this sentence structure is a little convoluted, it's clear that the human was able to compensate for the prompt and still write something coherent and meaningful.

Seeing the difference a prompt template can make 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 templated prompts don't work well for their topic.

## 6 STUDY 2: USER EVALUATION

We evaluated how our system supported PhD students in writing tweetorials. Tweetorials are short explanations of around 500 words which have a low-barrier to entry and are gaining popularity as a science writing medium [47]. We use PhD students as they are eager to participate in science writing [27] and many tweetorials are already written by PhD students, demonstrating that this a writing task our participants may conceivably want to engage in on their own. This study was approved by the relevant IRB.

### 6.1 Methodology

We recruited 13 participants, all students from five different STEM disciplines, to write tweetorials on a topic of their own choice, related to their area of study. By letting the participants pick their own topic, we ensured that they were writing within their area of expertise, and we were able to test our system on unseen topics.

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 when writing about 'machine learning algorithms'. Participants could ask clarifying questions to the facilitator. 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.

At this point the participant was asked to pick a topic to write about, as well as provide a 'context area' to aid the system in correctly interpreting their topic. Then they were given 15 - 20 minutes to interact with the system and write approximately the first 100 words of their tweetorial. 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 and partook in a semi-structured interview with the facilitator. The survey questions and the questions that structured the interview can be found in the appendix. The study took about an hour and participants were compensated \$40 USD for their time.

Participant interviews were transcribed and the authors performed a thematic analysis [9] on the interview transcripts. The analysis centered on three areas: how sparks were helpful, how sparks were unhelpful, 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 author discussed and collected the quotes into emergent themes.

### 6.2 Results

We report on participant demographics and topic selection in Table 3, as well as the prominent themes that emerged through our analysis in Table 4. Our thematic analysis covered three main areas: ways in which sparks were helpful, ways in which sparks were unhelpful, and how participants felt about incorporating sparks into their writing. For each area, we report themes with prevalence greater than 20%. Prevalence is measured by the number of participants who brought up that theme in their interview. In the case of ownership, there was a very high variability in responses, such that no one theme had over 20%. For this reason we report the reasons that people brought up in the text, but do not list them in the table. Behavioral data, such as time spent writing and how many sparks were starred, are also reported and timelines for each participants' activity can be found in Figure 7.

*6.2.1 Participant demographics and topic selection.* The 13 participants came from five STEM disciplines, with the most common disciplines being Climate Science and Public Health. All but one were doing PhD (the remaining doing a

Manuscript submitted to ACM

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

Table 4. Thematic analysis

Code	Prevalence	Example Quote
<i>reasons sparks were helpful</i>		
Crafted concise, detailed sentences.	54%	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.
Came up with ideas or angles.	46%	It [the system] reminded me, Oh, it's not just my application, there's these other people using the same technology, but working on other problems.
Showed reader perspectives.	31%	It [the system] reminded me that there might be a more common understanding of this thing that I'm writing about, that's different from the highly specific one I've been living in.
<i>reasons sparks were unhelpful</i>		
Incorrectly interpreted the topic.	38%	It [the spark] just wasn't helpful, but only because it was using the different sense of 'embedded'.
Inaccuracies.	23%	Some of the sparks said, like, logistic regressions are used to estimate relative risks, which is completely not true.
Not desired angle.	23%	Someone probably does really care about measuring sexist attitudes ... but it just isn't my focus.
Vagueness.	23%	I would say about 20% of them were just not specific enough to warrant talking about.

research Master's) and varied from their 2nd year to their 7th year in their program. Participants were asked how often they wrote about technical topics for a public audience, and how often they did so on Twitter. Most participants rarely or never did so, though a few did so on a monthly or even weekly basis.

Participants were asked to select a topic they understood well that was related to their research. The facilitator attempted to aid participants in selecting a topic that wasn't too broad, but also not too specific, but as the facilitator

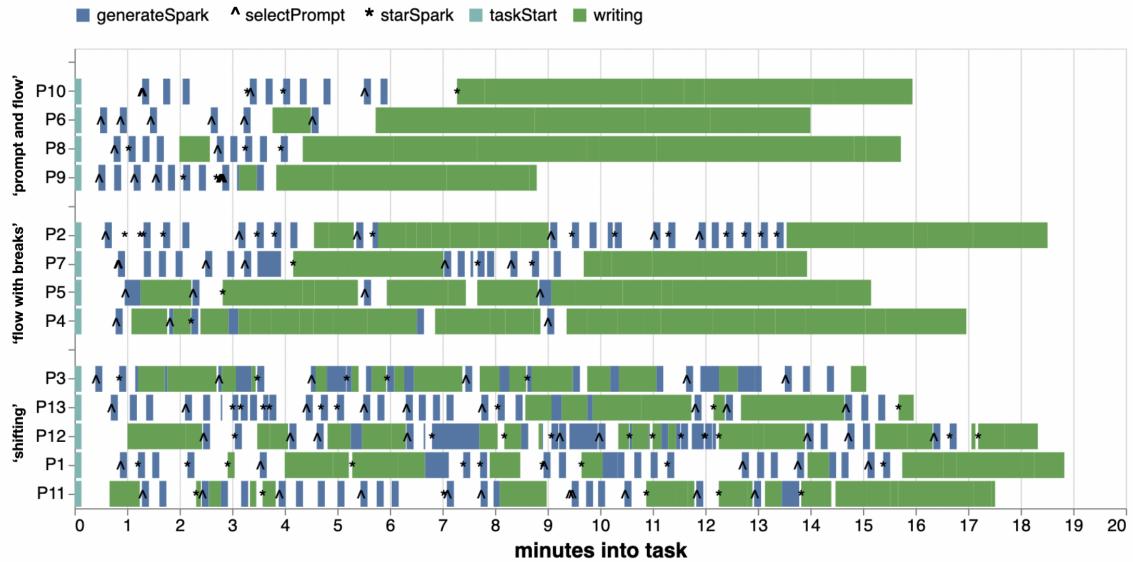


Fig. 7. Timelines of all participants from the study, with time writing versus time generating sparks marked in different colors. Participants are grouped by their engagement pattern.

did not necessarily have the same expertise as the participant this was at times difficult. Participants selected a wide range of topics, with no overlap. The full demographics and topic selection can be found in Table 3.

Given the diversity in the participants' topics, how well the system generated sparks on their topics, and how they articulated or responded to questions in the interview, there was a high variability in how participants felt about the system. For this reason, a prevalence of 50% or above is considered very high. This would mean that over 50% of participants independently responded in the same way to an open ended question, despite writing about a unique topic and seeing a unique response from the system.

**6.2.2 Participants had different engagement patterns when interacting with the system.** Overall we saw many different ways in which participants interacted with the system. Some participants generated and starred many sparks, while others generated only a handful and starred two or three. The average number of sparks generated was 17.2 (std=10) and the average number 'starred' was 5.1 (std=4.3). Additionally, it did not seem that generating or starring more sparks necessarily meant a participant found the system more helpful. Some participants who starred just two or three sparks talked at length in their interview about how useful the system was. Others who starred more than 10 complained that the system produced only vague or inaccurate outputs.

However, we did see themes of how people tended to interact with the system. Figure 7 shows all participants' task timeline, split into three archetypal patterns: 'prompt and flow', 'flow with breaks', and 'shifting'. We talk about each in detail:

**'Prompt and flow'** participants look at the sparks at the beginning, before they start writing, and then spend the rest of their time writing, never going back to generate more sparks. This kind of participant used ideas to jump start the writing, but once they were writing had no need to return to the sparks. Several participants described getting

some kind of inspiration from the sparks and then "flowing through an argument". Once these participants got into the 'flow' of writing, they didn't need any further support.

Although these participants don't generate more sparks, they may be using several sparks that they had 'starred' earlier. For instance, here are the first three tweets written by P8, where sentences or phrases inspired or highlighted:

Have you ever wondered why certain activities just feel right to you? Like you are able to do well and feel good in the process? Research on regulatory fit helps explain why certain activities feel right versus wrong... (1/n)

//

Regulatory fit happens because people can be driven by a range of different fundamental motives, and distinct activities "fit" each of these motives. For instance, people who are most motivated by their duties and obligations to others prefer activities that (2/n)

//

Involve carefully defending and protecting what they find important and methodically scrutinizing information. In contrast, people who are most motivated by hopes and aspirations prefer activities that involve eagerly making progress and engaging in curious exploration. (3/n)

Although they wrote all of this without generating more sparks, they did make use of two sparks – one for the first tweet ("Research on regulatory fit reveals that regulatory fit varies considerably across individuals depending upon factors.") and another for the second ("Regulatory fit happens because individuals behave differently under varying conditions."). So while they did not return to generating more sparks, they did return to the sparks they had previously starred.

'Flow with breaks' style participants are similar, but they pause in the middle of the writing and go back to sparks. This type of participant talked about returning to the sparks when they "would get stuck" or at some kind of inflection point in the writing process. P2 described this as, "there are a bunch of ideas that I need to kind of weave together before using the tool again, when I get more stuck." In both of these types of interactions, we see that the writer is using the sparks to jump-start their writing whenever something makes them pause, but when they are writing they are writing continuously, even if they are drawing on sparks they had earlier selected as potentially useful.

'Shifting' style participants are often moving between writing and generating sparks. They are constantly returning to the sparks for different details, or for help with crafting a sentence. They tended to star a lot of sparks and use a lot of custom prompts, as what they were looking for from the sparks changed as they proceeded through their writing, and a lot of language in their writing draws directly from a spark. For example, consider the start of P1's tweutorial on rainfall variability, where sentences or phrases inspired by sparks are highlighted:

Do you like coffee??

//

Want to understand how coffee and climate change are related?

//

Rainfall variability happens because rain falls differently depending on temperature and humidity.

//

Since temperature and humidity vary around the globe, we observe different rainfall patterns across the world.

//

For example, rainfall variability may cause extreme precipitation events leading to flooding or a lack of precipitation will lead to drought.

The last three tweets shown here are all taken directly from sparks, with minimal changes. It almost seems like collage, where the writer is mostly arranging language that came from the system.



(a) An example of how a participant in our study used a spark for crafting a detailed sentence. Highlighted text was inspired by the spark.  
 (b) An example of how a participant in our study used a spark to make their tweetorial more engaging to a general audience. Highlighted text was inspired by the spark.

This suggests there are different engagement patterns that people use when interacting with machine-generated suggestions. Although we expected participants would mostly use sparks for the beginning of their tweetorial, we found that participants used sparks at all points of their writing process. In the next sections, we report on the various ways in which participants made use of the sparks.

**6.2.3 Sparks helped participants craft concise and detailed sentences quickly.** Although we intended the system to inspire participants with new ideas, the most prevalent reason the participants cited for the spark being helpful was for crafting sentences. Many participants remarked that although the sparks were showing them information that they already knew well, it was much faster and easier to draw on language from the sparks than to write a sentence from scratch.

For instance, P12 said:

Most of the time [the system] was articulating the ideas that were already in my head in a way that's short and concise, which is useful. Like 'deprivation index measures the relative deprivation experienced by an individual relative to others,' that would have probably taken me like three sentences to write, then I'd have to spend time editing it down. And then yeah... this is a lot quicker.

Figure 8a shows another example in which P12 drew on a spark in order to write a clear and concise definition.

Several participants noted that they often go to Google or Wikipedia simply to get a well-written definition of a topic they understand well. This is something that the system was able to do for them without requiring a click away from the writing interface and incurring a change in context. P7 noted that it did a good job compressing what he would have looked for or found a Google search. P8 noted that all the sparks were similar to what she would have found on Wikipedia or via a Google search, but that they were "bite-sized" or "sound bite ready".

**6.2.4 Sparks reminded participants of other ideas or angles about their topic.** Several participants noted that the sparks provided good ideas or angles for discussing or introducing their topic. P2 noted that 'weather prediction models' were useful entry point to their research. They said, "that's something within my field that the general public might be more familiar with than what I actually do."

Figure 8b shows how P4 drew on a spark about the 'sea level rise' in order to make their topic of 'glacier retreat over the holocene' interesting to the reader. P4 said in their interview, "It's often hard to figure out how to spin things in ways that feel relevant to people who don't study this," and that the sparks helped her find ways to make her research

relevant. P7 said, “[the system] definitely generated multiple [ideas] that I could have written different tweotorials about.”

When asked if the sparks were giving them new ideas, many participants said that the system was helping them get to ideas they likely would have come up with themselves, but faster. For instance P4 said, “It was definitely faster. I think I would have gotten there, but it would have taken me longer.”

**6.2.5 Sparks encouraged participants to think about common reader perspectives.** Several participants noted that the sparks reminded them of how people reading their tweotorial might be interpreting their topic. For instance P10, who was writing about measuring sexism, noted that many of the sparks talked about sexist attitudes. She said the while that certainly is an aspect of measuring sexism, it isn’t the aspect that she actually studies. And so the sparks reminded her that people’s main assumptions when thinking about sexism is probably about attitudes and therefore that might be an assumption that she will have to address in her tweotorial.

P5, writing about computationally hard problems, noted that one of the sparks talked about NP-completeness. He said that while at first he thought this might be too technical, it then made him wonder if someone who was reading a tweotorial about computational hard problems might already know at least some of the keywords about this topic. In this way the sparks made him reflect on what knowledge his readers might already have.

**6.2.6 Sparks failed in different ways for different participants.** The ways in which participants found sparks to be unhelpful varied highly. The most common reason participants said sparks were unhelpful was that they incorrectly interpreted their topic. In this case, the sparks were not necessarily incorrect, but rather they reflected some alternate interpretation of their topic. For instance, P12, who was writing about deprivation indices, said that some of the sparks were about obesity. Obesity has little to do with deprivation indices, but they thought the algorithm may have been associating deprivation with nutrition. Similarly P8, who was writing about regulatory fit, commented on several sparks about government regulation, which is unrelated to her psychology topic, but she assumed the algorithm was simply free associating with the word ‘regulatory’.

Other reasons participants found the sparks to be unhelpful were factual inaccuracies, dealing with aspects of their topic that they were not trying to explain or that they did not study, and vague outputs. Participants also mentioned that some sparks were nonsensical, tautological, had too much jargon, or were simply “bizarre”.

Overall participants varied highly in how useful they found the sparks. Some participants found that the sparks were so low quality that they found the system completely unhelpful. Others said that even though some of the sparks were not helpful the ones that were helpful were so helpful they were unconcerned with a few that didn’t make sense or were off-topic.

**6.2.7 While most participants had no ownership concerns for science writing, a few felt apprehensive about accepting machine-generated ideas.** Most participants had no concerns about incorporating sparks into their writing. Several participants cited being very familiar with the material as a reason for not being concerned. 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 in the 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; another compared it to Grammarly (a grammar-checking service). One participant said that the sparks were simply elaborating on his own idea, and thus he still felt ownership of the material.

A few participants did have ownership concerns. One participant 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. Several participants noted that machine writing was simply the future, and while they may have some apprehensions, they knew they would have to get over it.

Across all participants, whether or not they had ownership concerns, they wanted to ensure that anything they wrote was in their own voice and that they did not plagiarize. 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." When asked if these edits would make the sparks less helpful, for instance if it meant it would take more time in order to edit what they chose to incorporate, most participants said that sparks would still overall speed up the writing process. But not everyone agreed; some participants thought the need to edit the sparks made them less helpful.

*6.2.8 Custom prompts allowed participants to iterate on how they interacted with the system.* Many participants noted that they were unsure how to get the best sparks out of the system. Several compared the system to Google but said it was easier to navigate Google because they already knew what kind of queries would get the best results. In the study though, they noted that they were unsure which prompts would produce the best sparks. Several participants even said that some of the sparks that were unhelpful may have been because their custom prompts didn't properly prime the system.

One way participants used the custom prompts was to get the system to better understand their topic. P13's discussed how the model didn't seem to correctly interpret his topic of "threat multiplier". You can see him iterating on custom prompts in an attempt to produce more accurate results:

- "threat multipliers are"
- "climate change is considered a threat multiplier because"
- "climate change is considered a threat multiplier for health disparities because"

Participants also used the prompts to generate more specific sparks. In the case of P7, the system seemed to understand his topic "document embeddings" well enough, but he was curious about specific aspects of this topic. You can see him looking for these in the following custom prompts:

- "Document embeddings help researchers"
- "Document embeddings help search engines"
- "Linguists can use document embeddings to"

Finally, others participants entered custom prompts that seemed to be used less for idea generation, and more as an auto complete. For instance, P12 entered several custom props designed to generate a definition:

- "The Townsend Index is"
- "In sociology, a deprivation index measures"
- "the variables in the Townsend Index are"

Overall, we saw many different uses of the custom prompts, and found them to be an important aspect of the system.

## 7 DISCUSSION

### 7.1 How to best use large language models in constrained writing tasks.

Creativity requires both convergent and divergent thinking [49]. We expected that the sparks would support participants with divergent thinking, by showing them how their topic might connect to things of interest to readers. And indeed many participants cited this as a main use case of the sparks. However, we found that many participants also found that sparks helped with convergent thinking. Sparks helped participants put the implicit, detailed, and often messy knowledge they had in their mind into concise sentences.

Existing work on large language models for storytelling tend to focus on divergent thinking. Language models provide writers with new ideas for plot points, character attributes, scene descriptions, and the like [1, 2]. However, what has typically been ignored is the convergent aspect of storytelling, for instance the tying together of plot points into a satisfactory ending. This is likely because of the difficulty language models have had in the past with staying on topic, though this may be changing as language models improve.

One of the reasons that the sparks may have been able to help participants with convergent thinking is because participants already knew what they were trying to explain. If we think of storytelling in response to a writing prompt as starting somewhere and perhaps not knowing the destination, we can think of writing tweotorials as knowing the destination but not where to start. In this case, participants were able to look at the sparks and recognize some of them as a convergence of the ideas already in their mind.

We sometimes saw this when participants would enter custom prompts into the system. They often knew the correct way to complete the prompt, but didn't necessarily know the best phrasing. For instance, P12 entered in the prompts "The Townsend Index is" and "the variables in the Townsend Index are" not because they didn't know what the Townsend Index is or what its variables are, but because they wanted the system to provide a well-crafted sentence for them.<sup>10</sup> In this instance we might think of the language model as performing convergent thinking for the writer and letting the writer recognize what is correct and well-phrased.

An area of future development we see is more explicitly using the language model to help writers with different aspects of the writing process, for instance by developing prompts intended to directly spark new ideas and other prompts to return concise definitions or summaries. We are seeing this in the storytelling space, where systems are being developed to prompt plot ideas or specific scene descriptions or focus on rewriting in a different style.

### 7.2 Three areas for technical development: better controls, increased breath, and graceful failing.

However, to support science writing we will need to improve the underlying technology. We see three main areas of technical development that may be fruitful.

The first is improving the controllability of language models. While it is clear that pre-trained language models can generate sentences that span the spectrums of specificity and complexity and take on a number of styles and opinions, it is less clear how to ensure that a language model generates the kind of outputs a writer wants. Prompt engineering seems like a promising avenue for this, especially as prompt engineering can be done in natural language. If prompt engineering is done with vectors, it will be completely opaque to users, but if we are able to control language models through natural language prompts alone, users will have some intuitive understanding as to how to control the language model and are more likely to be able to use this control themselves.

---

<sup>10</sup>The authors of this paper did not know what the Townsend Index is. For the curious, it is a measure of material deprivation levels in a population, and its variables are: unemployment, non-car ownership, non-home ownership, and housing overcrowding.

The second is increasing the breadth of a language model's knowledge. Again, it is clear that pre-trained language models contain much world knowledge, but it is also clear that this knowledge is not evenly distributed. There have been several calls to be more careful and explicit with the data that these pre-trained models are trained on [7, 8, 14]. If we better understood the training data, then we might better understand how much a model can be reasonably expected to know and can better augment the training data to increase a model's breadth.

Third, language models do not always fail gracefully or in a way that a writer might understand. In our study, when reflecting on sparks from a custom prompt that were tautological or redundant, P1 said, "maybe it's that either way you slice it 'atmospheric circulation patterns' is not the best way to start when you're talking to a general audience." In this case the participant was trying to make sense of why their custom prompt was producing low quality outputs – sentences that were syntactically correct but contained no real knowledge. They assumed that if the model was unable to generate a reasonable sentence then perhaps the topic was too specialized for her audience. But we are not sure why the system produced redundant results; perhaps the prompt wasn't engineered properly or something else went wrong. Studies could be done on a model's knowledge of the training data to see how a model responds differently when prompted with topics it has varying levels of knowledge of. Then we might be able to methodologically design methods that detect why model produce certain low quality outputs.

### **7.3 More experience with language models may improve results for users.**

When asked how the system compared to using a web search engine, many participants noted that they had so much experience with web search that they were more comfortable navigating it to find what they needed. In contrast, when using the system, they were unsure what kinds of prompts would produce the results they wanted. How users craft web search queries, and how they overall use web search engines for information retrieval, is a long and active area of study [46, 55]. For instance, advanced users query less frequently in a session, compose longer queries, and click on results further down on the list [53].

It's possible that some of the issues participants encountered, for instance outputs that incorrectly interpreted the topic or did not reflect participants desired angle, could be improved if participants had more experience and a better understanding of how to prompt language models. For example, communities interested in text-to-image generative models have been crowd-sourcing "tricks" (such as the appending of a phrase) that help steer generations to their liking. People are discovering workarounds by probing the system, in a way that is separate from the research on better prompt engineering. They are learning how to use the tool and its depth of understanding to their full advantage. No amount of prompt engineering will remove the need for user input, in whatever form that may be, just like no matter how much work is put into delivering better search results, users still have to learn how to craft good queries and build expertise and understanding of the system.

### **7.4 Limitations: the effects of personality, writing style, topic, and skill.**

Our user study was intended to be exploratory. The 13 participants came to the writing task with very different levels of experience in science writing, and very different topics for the system to respond to. Future work, and more developed systems, may be able to study these differences more carefully, perhaps with a larger set of participants, such that the impact of things like writing style or topic granularity may be able to be taken into consideration.

Additionally, it was clear that some participants were far more forgiving of system than others. For instance, some participants saw one or two inaccurate or redundant sparks and then considered that prompt or even the whole system to be a "dead end". Others were not phased by seeing sparks that didn't meet their needs. They even found ways to

re-purpose unhelpful sparks to be useful, for example by interpreting inaccurate sparks as an indication that their topic may have been too niche or as a signal for how readers might also misinterpret their topic. Work on how users develop mental models of AI systems has shown that some people are more likely to blame the system when something goes wrong, and others are more likely to blame themselves [20]. Future work may want to study if personality type is a predictor of how useful participants find these kinds of systems, such that it does not remain a confounding factor.

Finally, our system uses just one particular language model with one particular decoding method. Improving the generative abilities of language models is a large and growing area of work. We expect that their abilities will continue to improve, and this will obviously change the ways in which people react to them.

## 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-created gold standard. We also found in an exploratory study with 13 PhD students that participants used the sparks in many different ways: to craft detailed sentences, to get ideas for how to engage the reader, and to better understand common reader perspectives. Finally, we discussed how language models might be used as support tools for writing in the future, what areas of technical development we believe will be fruitful, and how users might learn to interact with language models in a writing context.

## REFERENCES

- [1] [n.d.]. AI Dungeon. <https://play.aidungeon.io/main/home>. Accessed: 2021-09-09.
- [2] [n.d.]. SudoWrite. <https://www.sudowrite.com/>. Accessed: 2021-09-09.
- [3] 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.
- [4] Kenneth C Arnold, Krysta Chauncey, and Krzysztof Z Gajos. [n.d.]. Sentiment Bias in Predictive Text Recommendations Results in Biased Writing. ([n. d.]), 8.
- [5] Kenneth C Arnold, April M Volzer, and Noah G Madrid. [n.d.]. Generative Models can Help Writers without Writing for Them. ([n. d.]), 8.
- [6] 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.
- [7] Jack Bandy and Nicholas Vincent. 2021. Addressing "Documentation Debt" in Machine Learning Research: A Retrospective Datasheet for BookCorpus. *arXiv:2105.05241 [cs]* (May 2021). <http://arxiv.org/abs/2105.05241> arXiv: 2105.05241.
- [8] 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>
- [9] 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>
- [10] 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>
- [11] 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.
- [12] 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>

- [13] 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.
- [14] Isaac Caswell, Julia Kreutzer, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Javier Ortiz Suárez, Iroro Orife, Kelechi Ogueji, Rubungo Andre Niyyongabo, Toan Q. Nguyen, Matthias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2021. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *arXiv:2103.12028 [cs]* (April 2021). <http://arxiv.org/abs/2103.12028> arXiv: 2103.12028.
- [15] 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.nacl-main.336>
- [16] 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>
- [17] 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.
- [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] Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners. *arXiv:2012.15723 [cs]* (June 2021). <http://arxiv.org/abs/2012.15723> arXiv: 2012.15723.
- [20] Katy Ilonka Gero, Zahra Ashktorab, Casey Dugan, Qian Pan, James Johnson, Werner Geyer, Maria Ruiz, Sarah Miller, David R. Millen, Murray Campbell, Sadhana Kumaravel, and Wei Zhang. 2020. Mental Models of AI Agents in a Cooperative Game Setting. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–12. <https://doi.org/10.1145/3313831.3376316>
- [21] 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>
- [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] 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>
- [34] 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.
- [35] Stephanie Lin, Jacob Hilton, and Owain Evans. [n.d.]. TruthfulQA: Measuring How Models Mimic Human Falsehoods. ([n. d.]), 13.

- [36] 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.
- [37] 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>
- [38] 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>
- [39] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. [n.d.]. Language Models are Unsupervised Multitask Learners. ([n. d.]), 24.
- [40] 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.
- [41] Larin 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.
- [42] Melissa Roemmele. [n.d.]. Writing Stories with Help from Recurrent Neural Networks. ([n. d.]), 2.
- [43] 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.
- [44] 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>
- [45] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2020. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. *arXiv:1909.08053 [cs]* (March 2020). <http://arxiv.org/abs/1909.08053> arXiv: 1909.08053.
- [46] Catherine L. Smith. 2017. Domain-independent search expertise: Gaining knowledge in query formulation through guided practice. *Journal of the Association for Information Science and Technology* 68, 6 (June 2017), 1462–1479. <https://doi.org/10.1002/asi.23776>
- [47] 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>
- [48] Allen B. Tucker, Sergei Nirenburg, and Victor Raskin. 1986. Discourse and cohesion in expository text. In *Proceedings of the 11th conference on Computational linguistics - Association for Computational Linguistics*. https://doi.org/10.3115/991365.991419
- [49] Barbara Tversky and Juliet Y. Chou. 2011. Creativity: Depth and Breadth. In *Design Creativity 2010*, Toshiharu Taura and Yukari Nagai (Eds.). Springer London, London, 209–214. [https://doi.org/10.1007/978-0-85729-224-7\\_27](https://doi.org/10.1007/978-0-85729-224-7_27)
- [50] 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.
- [51] Ashwin K. Vijayakumar, Michael Cogswell, Ramprasad 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.
- [52] 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.
- [53] Ryan W. White and Dan Morris. 2007. Investigating the querying and browsing behavior of advanced search engine users. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '07*. ACM Press, Amsterdam, The Netherlands, 255. <https://doi.org/10.1145/1277741.1277787>
- [54] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumont, Clement Delangue, Anthony Moi, Pierrick 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>
- [55] Takehiro Yamamoto, Yusuke Yamamoto, and Sumio Fujita. 2018. Exploring People’s Attitudes and Behaviors Toward Careful Information Seeking in Web Search. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, Torino Italy, 963–972. <https://doi.org/10.1145/3269206.3271799>
- [56] 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.
- [57] 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.
- [58] 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 METHODS FOR STUDY 1

### A.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](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](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](https://en.wikipedia.org/wiki/Glossary_of_environmental_science).

## B METHODS FOR STUDY 2

### B.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?

### B.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?