

Tweetorial Hooks: Generative AI Tools to Motivate Science on Social Media

Tao Long^{§✉}, Dorothy Zhang[¶], Grace Li[¶], Batool Taraif[¶], Samia Menon[✉],
Kynneddy Simone Smith[✉], Sitong Wang[✉], Katy Ilonka Gero[✉], Lydia B. Chilton^{§✉}

[✉]Columbia University, New York, NY [¶]Barnard College, New York, NY

[§]{long, chilton}@cs.columbia.edu

Abstract

Communicating science and technology is essential for the public to understand and engage in a rapidly changing world. Tweetorials are an emerging phenomenon where experts explain STEM topics on social media in creative and engaging ways. However, STEM experts struggle to write an engaging “hook” in the first tweet that captures the reader’s attention. We propose methods to use large language models (LLMs) to help users scaffold their process of writing a relatable hook for complex scientific topics. We demonstrate that LLMs can help writers find everyday experiences that are relatable and interesting to the public, avoid jargon, and spark curiosity. Our evaluation shows that the system reduces cognitive load and helps people write better hooks. Lastly, we discuss the importance of interactivity with LLMs to preserve the correctness, effectiveness, and authenticity of the writing.

Introduction

Communicating science and technology is important for the public to understand and engage in a rapidly changing world. Recently, a majority of the public learns about the world not from traditional publications, but from social media platforms (Shearer and Matsa 2018). *Tweetorials* are an emerging format for explaining complex scientific concepts on Twitter. They consist of a series of tweets that explain a technical concept in informal, narrative-driven ways (Breu 2019; Breu 2020). Whereas typical science writing is often formal, the norms of social media allow scientific conversations to take on a more personal style (Brüggemann, Lörcher, and Walter 2020), allowing for more creative forms of expression and engagement.

The most important part of a Tweetorial is the first tweet. This is often called a “hook” because it aims to hook the readers’ attention and spark their curiosity so they want to read more. Although there are many ways to do this, an analysis of Tweetorial hooks (Gero et al. 2021) has shown that a common pattern is to start with a specific, relatable experience that uses no jargon. However, the challenge is to find a common experience for technical topics that a general audience of readers will find engaging.

Many STEM experts want to write creative and engaging science-related content for the public, but are not trained to

do so. Their writing training is mainly for writing to peers — other experts who are familiar with the motivation for the work, who expect expert terminologies, and who know the context surrounding the science and the formal culture of academic writing (Aldous, An, and Jansen 2019). Such writing is typically (and purposefully) formulaic, and creative writing may even be discouraged in such contexts. Although there are many theories, examples, and books about public science communication, they lack mechanistic strategies proven to help people use them (Howell et al. 2019; McClain and Neeley 2014; Yeo 2015). Providing explicit support for informal science writing like Tweetorial hooks can better support experts in writing for the public.

We explore various ways for large language models (LLMs) to help people write engaging, creative hooks for computer science topics. We first explore how well LLMs can write hooks on their own by investigating three prompting strategies: instructions, instructions and examples, instructions, examples, and relatable experiences. We find that although adding examples and experiences in the prompt improves hooks, the LLMs still have much room for improvement. Then, we design an interactive system that scaffolds the process of writing hooks but allows users to accept, reject, or improve LLM suggestions at every step. In a user study with ten people proficient in their domain and familiar with Tweetorial hooks, we show this drastically improves their hooks and reduces cognitive load compared to writing without the system.

Related Work on LLMs and Writing Support

Advances in LLMs have resulted in machine abilities to complete prompts with rich knowledge, commonsense reasoning, and fluent language composition (Radford et al. 2019). Despite not being explicitly trained for specific tasks, these models possess impressive generative capabilities and can perform a diverse range of tasks. Moreover, providing just a few examples in the prompt itself can significantly enhance the quality of the model’s outputs (Brown et al. 2020).

LLMs show great promise for supporting creativity and writing tasks. They can help with story writing (Calderwood, Wardrip-Fruin, and Mateas 2022; Chung et al. 2022), brainstorming (Singh et al. 2022), and finding creative connections (Wang et al. 2023) as well as story angles from press releases (Petridis et al. 2023). They have been shown

to help with all three stages of the cognitive process model of writing (Flower and Hayes 1981): planning/ideation, translating/drafting, and reviewing (Gero, Liu, and Chilton 2022). Rather than executing these stages in a linear fashion, the writing process typically involves iterative use of these stages and requires writers to switch between their writing goals while keeping their audience in mind (Emig 1977). Because of this, writing can be taxing on both the writer’s short- and long-term memory, resulting in high cognitive demands (Hayes 1996). Thus, LLMs as a writing companion and support can benefit writers in reducing mental load.

Despite the successes of LLMs, problems remain. Language models tend to output repetitive and vague responses (Holtzman et al. 2020; Ippolito et al. 2019), particularly when a prompt is underspecified or too difficult to address. One approach to address this is to chain LLM prompts together (Wu, Terry, and Cai 2022): breaking down a problem into simpler and more explicit steps can make it easier for LLMs to complete. A bigger challenge is that language models have no model of truth. They learn correlations from large amounts of text, but they are not able to tell if the text they produce that includes falsehoods or offensive language (Bender et al. 2021). Thus, LLMs may best assist writers in producing higher-quality written outputs by providing support during the writing process instead of replacing the writer and writing on their own.

Headline writing is an established challenge in natural language processing. Fully automated systems have some successes at generating headlines (Bukhtiyarov and Gusev 2020), and some can even write ones in a “clickbait” style to hook readers (Jin et al. 2020; Xu et al. 2019). Although headlines do serve as hooks, traditional journalistic headlines have a different style than Tweeterial headlines. Tweeterial hooks are a little longer than headlines and use that space to start an engaging, relatable, and vivid personal anecdote. Thus the narrative, rather than the keywords, is the basis for engaging readers. This paper extends works on engaging readers with intriguing and meaningful content.

Background on Tweeterial Hooks

Tweeterials are a “collection of threaded tweets aimed at teaching users who engage with them” (Breu 2019). Across a wide range of topics from medicine, to climate science, to physics, to computer science, these tweets always introduce a technically complicated concept or answer a popular science question through informal, narrative-driven, and creative writing (Breu 2020; Gero, Liu, and Chilton 2022). Figure 1 shows the first, last, and some middle tweets that form the overall narrative. Hooks are the first tweet that grab readers’ attention and pull them into the narrative.

Previous work (Gero et al. 2021) has analyzed Tweeterial hooks and described attributes of high-quality ones: 1) a relatable and interesting example as a lead-in and 2) an intriguing question that is driving and specific that sparks readers’ curiosity. Relatable and specific content can take many forms. It can relate a topic to things in the news, refute a popular misconception, or take a common daily experience and help explain it. For the language to be relatable, the hook should not include jargon. Using unfamiliar technical terminology undermines the purpose of engaging the public (Bullock et al. 2019). Then, an intriguing question will be directly or implicitly proposed to the readers to help spark curiosity and draw them to the following thread. The unanswered question will connect the previous relatable example to the following threads of explanations. Thus, we establish a list of requirements (R) for a relatable and engaging Tweeterial hook:

- **R1 - Jargon-Free:** Does the hook avoid jargon or unexplained terminology so the general audience can understand it easily?
- **R2 - Specific and Relatable Example(s):** Does the hook include specific and relatable example(s) about the topic?
- **R3 - Sparks Curiosity:** Does the hook drive readers to continue reading and satisfy their curiosity?

Here are two examples of Tweeterial hooks for computer science topics that exhibit all these properties:

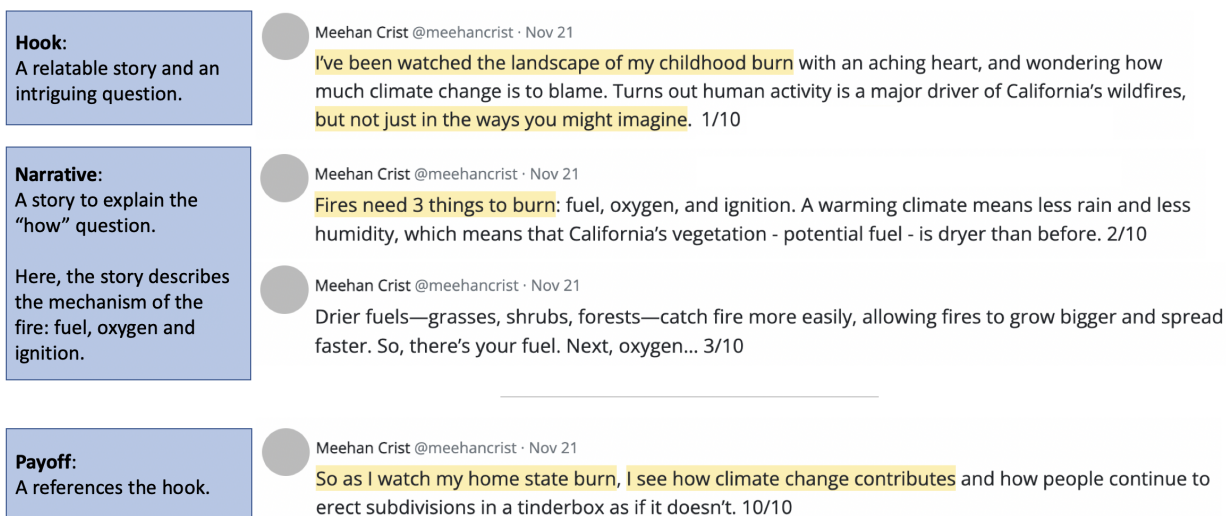


Figure 1: A Tweeterial about the California wildfires (Crist 2019) annotated for narrative structure. Yellow highlights indicate key phrases of the hook (including the relatable detail and the intriguing question), narrative, and payoff. More annotated Tweeterial examples can be found on our website: <http://language-play.com/tech-tweets/annotations>

- Virtual Private Network (VPN): “I once torrented Last Week Tonight — then my landlord got a complaint from Comcast! WTH? My friends never got caught. Ugh. So here are things I wished I had known about how to be sneaky on the internet:”
- Language Models: “My son relies on his Alexa to help with his math homework every single night. While I am concerned about his learning, I am interested in how Alexa understands what he is saying? Is it the same way that humans understand language? What is the difference? A thread on how language models helps with this:”

For each hook, the topic is motivated by an everyday experience. For VPNs, the experience is torrenting. For language models, the experience is Alexa. Each experience is told in a personal way (“then my landlord got a complaint from Comcast!”), with informal language (“With? My friends never got caught. Ugh.”). Often they have very specific details (“Last Week Tonight”). They don’t contain jargon—other than when mentioning the name of the topic towards the end of the hook. And they have a question or implied question that sparks curiosity and drives the reader to learn more (“how to be sneaky on the internet”). This is a lot to achieve in one tweet.

Studies on Tweetorial writing have shown that writing hooks is a key challenge for STEM experts (Gero et al. 2021). They are trained to write about their work in a formal tone for other experts, and it is difficult to go against that training. Also, they feel uncomfortable using subjective and informal language and avoid personal details, even though 80% of the Tweetorials have them.

In an exploratory study using LLMs to support Tweetorial writers, one of the major use cases was ideating concrete examples for the hook (Gero, Liu, and Chilton 2022). This indicates there is potential to help STEM experts write in informal styles. We build on this potential by studying LLMs’ potential to write hooks, then designing a workflow to scaffold writers’ hook writing process and using LLMs to suggest options for relatable experiences that are jargon-free and can spark curiosity.

Study 1: Prompt Engineering Study

We first investigate how well an LLM can write hooks without human intervention. Then, we compare the performance of three prompting strategies and use expert annotators to evaluate the outputs.

Participants and Procedures We identified 30 technical computer science topics that are important for a general audience to understand. We selected them randomly from the *Sideways Dictionary* (Jigsaw 2017) — a website for journalists to find accessible explanations for common technical terms. These terms included such as *Database*, *Browser Hijacking*, *Programming Language*, *Internet Service Provider*, and *Autocomplete* (See Appendix for the complete list).

The three prompting strategies (PS) we compared are:

- **PS1 (Instructions only)** is the most basic strategy which asks for a hook and provides simple instructions that the

hook should be jargon-free, include a relatable and specific example, and spark curiosity. This is the bare minimum needed to explain to the LLM the goal of a hook.

- **PS2 (Examples and Instructions)** has all the instructions from PS1, and adds five examples of good hooks we identified and collected inside the team. These hooks were taken from published Tweetorials and edited lightly for clarity. Adding examples is a known technique to help the LLM learn the “styles” that are difficult to describe or to phrase in specific instructions such as writing objective, writing structure, diction, and tone.
- **PS3 (Examples, Chained User Details, and Instructions)** is a three-stage pipeline that chains LLM prompts together (Wu, Terry, and Cai 2022), in addition to all the content from PS2. It first asks for the user’s topic to generate everyday examples, then common experiences, then a specific personal anecdote about this experience. LLM chaining is known to work well when instructions are complicated. It breaks down the problem into simpler steps and builds up to a complex output.

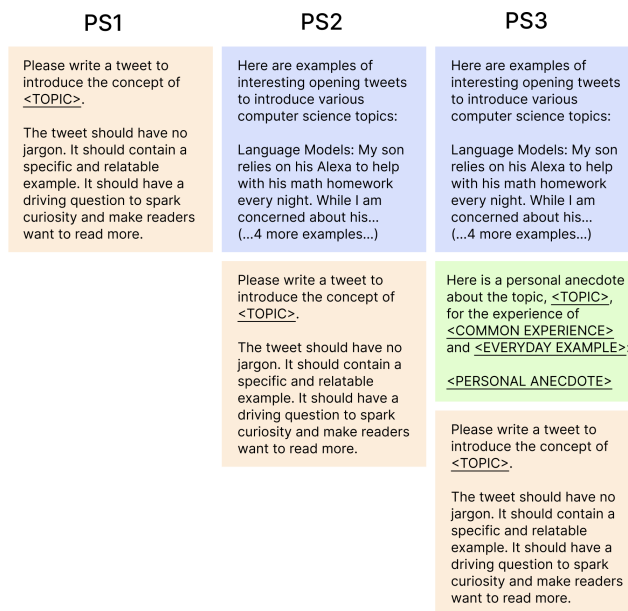


Figure 2: An illustration of the three prompting strategies

We used OpenAI’s GPT-3 API and its *text-davinci-003* model with the default settings for all parameters, as it was the most capable model available at the time of our study.

In this study, we investigate the following hypothesis:

Hypothesis #1: PS3 will attain the highest overall score and outperform both PS1 and PS2 across all three rubric categories. We believe that the prompt chaining will break down the complex hook writing task into simpler steps that LLMs will be better able to solve one at a time.

To evaluate the three prompting strategies, we hired three annotators with professional training in communication and writing to judge the hooks’ quality. Each annotator rated 270 hooks — 30 topics with three prompting strategies and three generations each. The annotators were paid \$20 per

hour and evaluated each hook on a 1 to 5 scale based on the criteria: whether it is jargon-free (R1), contains a relatable example (R2), and sparks curiosity (R3). They received a detailed annotation rubric with examples (See Appendix).

Results

Overall, the annotators had fair agreement on their assessment, with a Fleiss' kappa of 0.23.

According to our annotation results (See Figure 3), PS1 was the lowest-scoring strategy, with an average of 2.93 out of 5. PS2 and PS3 were only about half a point higher than PS1 at 3.49 and 3.47 out of 5, but about equal to each other. All three strategies performed pretty well at being jargon-free, even PS1. Seemingly, LLMs can follow the instruction to be jargon-free without examples. However, where PS1 struggled was in being relatable and sparking curiosity. Here, PS2 and PS3 performed 1 point better on relatability and almost 1 point better on curiosity. This indicated that the training examples in PS2 and PS3 did help LLMs “learn” how to write a more relatable hook with details.

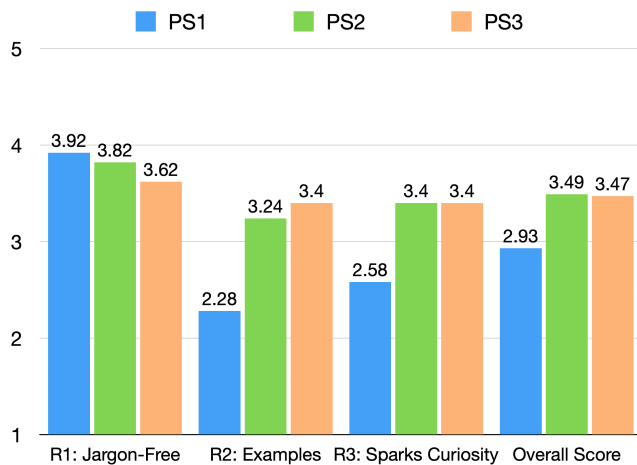


Figure 3: Average scores for each prompting strategy based on rubric performance

To answer our Hypothesis #1, **PS3 and PS2 were similarly good, and both were better than PS1.** Specifically, PS3 was only significantly better than PS1 for R2 (p-value < 0.001), R3 (p-value < 0.001), and the overall performance (p-value < 0.001). However, compared to PS2, PS3 performed similarly to PS2 in all categories. This was surprising because the average score of PS2 (SCORE - 3.49/5) left much room for improvement. We hoped the chaining in PS3 would improve the hook quality, but it did not.

One reason for PS3’s unideal performance was that, PS3 often included jargon and failed to be relatable, though PS3 provided more detailed experiences. For example, the lowest-scoring hook from PS3 on Table 4, we saw that, with a topic of Back End, it did not give out a more detailed experience than what PS2 usually had: “my recent experiences with Amazon Web Services’ Identity and Access Management feature...” It reflected a problem that PS3 often included details that were specific but not relatable and even contained jargon or unexplained terms like “Amazon Web

Services,” “Identity and Access Management,” and “bad end access.” Clearly, this experience was not relatable to general audiences, though it was detailed. Thus, for the lack of improvement from PS2 to PS3, we can see the lack of manual filtering of the specific. However, with humans in the loop, the process of picking better answers would help improve answers at every step and make the final results closer to the rubric. Thus, to understand whether human interventions help with the PS3, we conducted the following study.

Study 2: User Study

We conducted a user study to evaluate the effectiveness of our LLM-based Tweetorial solution for users with the need to communicate science to the general public.

System Description We built an interactive web application using HTML, Python, Javascript, Flask, and the GPT-3 API to help users write engaging hooks for technical topics. The interface scaffolded the process of writing a hook into steps and used GPT to generate suggestions that the users can regenerate, modify, or accept before going to the next stage. The system and the workflow can be seen in Figure 4:

- **Step 1. Everyday Examples of Topic:** Users input their topic, and the system generates five concrete everyday examples of that topic. For example, a user inputs the topic of “AJAX” from web programming, and the system generates five everyday examples such as “autocomplete in Google Search” and “loading new posts on Facebook without refreshing the page.” The user picks an everyday example that is factually correct and relatable.
- **Step 2. Common Experiences for an Everyday Example:** Given an everyday example (from the previous step, or edited), the system generates five common experiences people might have with that example. For example, a common experience the system suggested relating to “loading new posts on Facebook” is “Scrolling effortlessly for new content.” The user likes the relatable feeling of “scrolling effortlessly” but wanted a more vivid experience that would resonate even more with users. Inspired by the system, the user wrote: “Staying up late browsing social media.”
- **Step 3. Sample Personal Anecdote:** Given a common experience, the system generates three personal anecdotes and narratives. For example, the system generates three sample anecdotes that rephrase the common experience in a first-person view. The user liked the phrasing “just the other night, I found myself [scrolling Facebook]” - it aligned with their own experience and felt relatable. They didn’t like some of the dated language (“burning the midnight oil”), but they were willing to see a more specific version of the anecdote.
- **Step 4. More Specific Personal Anecdote:** Given a short personal anecdote, the system generates a new version with more specific details. Here, it made “Just the other night” more specific by saying “a quiet Friday night”. Here, the details weren’t correct and weren’t particularly engaging. Thus, they rewrote their own anecdote by drawing from their personal experience, with similar types of

specific language, but more succinct and authentic to their experience: “Yesterday, I was up until 3 am scrolling Facebook.”

- **Step 5. Sample Hook for a Specific Anecdote:** Given a specific personal anecdote, the system generates an example hook based on all previous inputs. For example, the user liked their own personal anecdote as a specific and relatable example (R2), but the system generated a good way to spark curiosity (R3): “*What’s the magic behind this continuous stream of posts?*” But the user adapted the language to be more emotionally heightened: “*Behind all the addiction algorithms, there’s a fundamental tech hack and it’s used on almost EVERY website to provide a smooth experience.*”
- **Step 6. Final Hook:** Users input their final hook into the text box. They can directly copy the LLM-generated hook from Step 5, they can adapt it (as seen in Figure 4), or write their own with inspiration from some of the ideas in previous steps. They click “Submit” when they are done.

Participants and Procedures We recruited ten participants from a local college student network and asked them to write Tweetorial hooks with and without our prototype in February 2023. The participants included six females and four males, with an average age of 20.1 years old. All ten users had expertise in computer science and familiarity with the particular topics we were asking them to write about. The study took around 1.5 hours, and they were paid \$30.

Before the study, participants first received a 10-minute introduction to Tweetorials and hooks. The introduction included explanations and examples of what constitutes a good hook. Then, they were asked to write hooks for six randomly chosen computer science topics from the list we used for the annotation study. The topics, in sequential order, were *Front End*, *Autocomplete*, *Programming Language*, *Net Neutrality*, *Application Programming Interface (API)*, and *Cyber-crime*. For the hook-writing tasks, we asked each of them to write on three topics using the system and three without the system. The participants were randomly assigned to two groups, each consisting of five. Group 1 wrote with the system for the first, third, and fifth topics, and without the system for the others. Group 2 followed the opposite order. This approach ensured a fair comparison by evenly distributing the system use across all topics and participants.

During each hook-writing task, we first provided the participants with the topics and informed them whether to use the system. Then, they had eight minutes to write a hook. During the session, users were informed that they could search for information online regardless of the conditions. After each hook writing task, we asked them to fill out a NASA Task Load Index (TLX) (Hart and Staveland 1988) questionnaire to understand their mental load and experiences quantitatively. After finishing all six writing tasks, we started a 25-minute semi-structured interview to learn more about their experiences and hook writing process.

In this study, we investigated the following hypothesis:

Hypothesis #2: Using the system reduces the mental load and increases the performance of writing hooks.

Tweetorial Hook Incubator ^ _ ^

Step 1: Please Input the Topic.

AJAX

Here are 5 everyday examples:

1. Loading new posts on Facebook without refreshing the page.
2. Autocomplete suggestions in Google Search.
3. Live weather updates on a website.
4. Loading new Tweets on Twitter without page reload.
5. Real-time Gmail inbox updates without refreshing the page.

Step 2: Generate Common Experiences.

Please enter your favorite everyday example here.

Loading new posts on Facebook without refreshing the page

Here are 5 common experiences:

1. Scrolling effortlessly for new content
2. Mixed feeling of excitement and surprise
3. Temporary lag in displaying new posts
4. Cycle of fresh and repetitive content
5. Unpredictable change in content arrangement

Step 3: Generate Personal Anecdotes.

Please enter your favorite common experience here.

Staying up late browsing social media, Scrolling effortlessly for new content

Here are 3 personal anecdotes:

1. Just the other night, I found myself burning the midnight oil scrolling through myriad social media posts, thanks to AJAX, the posts just kept appearing.
2. While losing track of time diving in the rabbit hole of social media late at night, I remember how AJAX ensures my feed continuous, never-ending.
3. I am always amazed at how the seamless and endless scrolling of social media feeds is due to AJAX. It indeed made late-night web surfing addictive!

Step 4: Generate a More Specific Anecdote.

Please enter your favorite personal anecdote here.

Just the other night, I found myself burning the midnight oil scrolling through myriad social media posts, thanks to AJAX, the posts just kept appearing.

Here is the specific version of the personal anecdote:

Very recently, on a quiet Friday night, I caught myself absorbed in the endless vortex of Facebook updates, staying up till the wee hours. The AJAX-backed mechanism ensured a ceaseless flow of new posts appearing on my feed.

Step 5: Let GPT Generate an Example Hook.

Please enter the specific personal anecdote here.

Yesterday, I was up until 3am scrolling Facebook. How did I not notice?

Here is the specific version of the personal anecdote:

It was 3 am & there I was, going through my Facebook feed without a pause. No loading screens, no waiting! It felt endless & got me thinking, *what's the tech magic behind this continuous stream of posts? So here we are, set to explore the wonders of 'AJAX' Stick around, it's going to be a fun ride! 1/*

Step 6: Write a Final Hook.

Please check all these info above and use them to write a final hook below:

Yesterday, I was up until 3am scrolling Facebook. How did I not notice? Behind all the addiction algorithms, there's a fundamental tech hack, and it's used on almost EVERY website to provide a smooth experience. Let me tell you about AJAX...

Figure 4: An example of how users write a Tweetorial hook about “AJAX” with our tool, Tweetorial Hook Incubator. Users have the flexibility to navigate through the workflow: they can follow it sequentially from top to bottom, start from the middle steps, return to previous steps, or restart the workflow. They can **accept the LLM outputs**, **adapt**, **use their own responses**, or regenerate.

Results

The TLX results are visualized in Figure 5 and Table 1. As we split participants into two groups for randomization, they had good internal consistencies within each group, with Cronbach’s Alphas of 0.78 and 0.85.

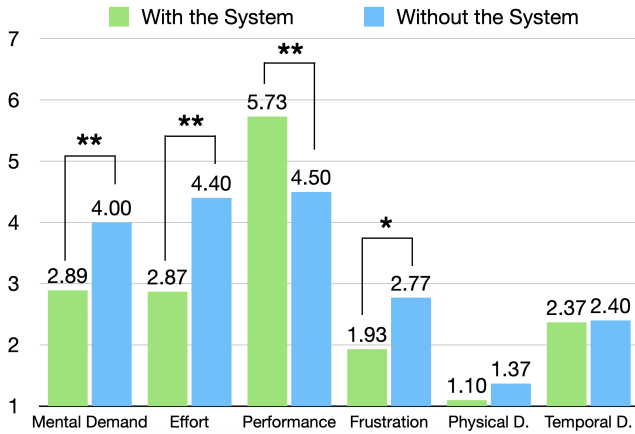


Figure 5: User study TLX results (** indicates statistical significance at the $p < .005$ level, * indicates statistical significance at the $p < .05$ level)

TLX Dimension	With System	Without	p-value
Mental Demand	2.87	4.00	0.004**
Effort	2.87	4.40	0.002**
Performance	5.73	4.50	0.001**
Frustration	1.93	2.77	0.02*
Physical Demand	1.10	1.37	0.08
Temporal Demand	2.37	2.40	0.598

Table 1: User study TLX results and p-values for Wilcoxon tests (** indicates statistical significance at the $p < .005$ level, * indicates statistical significance at the $p < .05$ level)

1. Less Mentally Demanding The TLX scores indicated that writing hooks was less mentally demanding with the system (SCORE - 2.87/7) than without it (SCORE - 4.00/7, p -value = 0.004). All ten users expressed that without the system, it was hard to find concrete and specific examples of abstract topics. Under that condition, many users did their own brainstorming, often trying to think of their own experiences with the topic and attempting to recall tangible details and emotions about it before they were able to start writing (P8). Five users said that even if they did come up with a few examples, it was challenging to narrow them down to one to fit the criteria: relevant, relatable, and interesting enough to make them keep reading (P1, P2, P5, P7, P8).

All ten users expressed the ease of using the system to help simplify language into digestible terms that more people can understand. P4 shared it is easier to brainstorm a lot of ideas, and it helped open horizons and applications, but they still ended up choosing one that resonated the most. P1, P2, P5, P7, P8, and P10 mentioned that the workflow was straightforward, easy, clear, and simple to use, easing mental burdens during the hook writing process. All ten users said they would use this tool in the future.

2. Less Effort The TLX scores indicated that writing hooks required less effort with the system (SCORE - 2.87/7) than without it (SCORE - 4.40/7, p -value = 0.002). Under the without system condition, seven users spent a lot of effort searching the Internet to find examples without much success. Even though there were some examples on Google, it was hard and time-consuming for users to find them. P2 and P5 shared that Google felt like an “*ocean of information*.” They had to spend a long time searching: skimming through the titles, avoiding getting technical information, and clicking on it to understand the material first and then adapting it to their own work. They needed to put down three to five search queries on Google to find the results they wanted. For example, P9 used “*net neutrality examples*” and “*net neutrality in simple terms*”; P2 used “*examples of APIs we use in our everyday lives*”, “*define programming language in a fun way*”, “*explain the term front end for a 5-year-old*” and “*what is the front end for dummies*”; P7 used “*examples of popularly used APIs*” and “*how to talk about programming languages in layman’s terms*.” Trying different terms took a long time and effort (P8) and often ended in failure (P1).

In contrast, P1, P2, P4, P6, and P7 all mentioned that the with-system experience was just effortless: “*easy to generate and regenerate*”, “*easy to find strong ideas*”, and easily “*reminded me of what I already knew*”. P8 shared that the writing workflow was seamless, enabling them to complete the hook writing process by following the steps without searching on Google. In total, eight of the ten participants finished the with-system writing process without Google.

3. Better Performance The TLX scores indicated that users achieved better performance writing hooks with the system (SCORE - 5.73/7) than without it (SCORE - 4.50/7, p -value = 0.001). Also, from the table 2, users felt more confident and satisfied with the results they obtained from the system when using LLMs, as they believed that the process involved fewer personal biases and LLMs had more knowledge about real common experiences. For instance, P8 mentioned that they believed the common experiences generated by LLMs were meant to be more familiar and relatable to the general public. In comparison, they reported concerns that the experiences they came up with on their own or from Google were not common enough and biased toward their personal background. Similarly, P2, P4, and P7 shared that they experienced these implicit biases and received fewer affirmations while trying to write a hook without LLMs, as they trusted LLMs more.

4. Users Edit LLM Hooks to Meet Requirements In Step 5 of the system, users were presented with a hook written by the LLM based on their responses to Steps 1-4. All ten users expressed that the LLM-generated hooks are good and useful, while six of them expressed the need to edit the LLM-generated hooks to make them more relatable and engaging. When asked to make a quick comparison between their edited version and the LLM-generated ones, all these six writers shared that their edits were necessary and helped elevate the quality of the hooks.

Responding to R1 (being jargon-free), P1, P8, and P10 shared that they still found jargon inside the LLM-generated

Topic	Without the System	With the System
Cybercrime	These days computers are a huge part of our lives- what illegal activities could be going on within our computers? In this thread, we will be exploring cybercrime, and what this could mean for our online safety. 1/	Have you ever received a call out of the blue from someone claiming to be from your bank, asking for your personal information? After this happened to me recently, I wondered what other kinds of cybercrime exist and how someone like me can protect themselves? Here’s what I found out:

Table 2: Collection of hooks generated in both the “with-system” and “without-system” conditions from the user study. Both examples are jargon-free (R1) and contain specific and relatable examples (R2, highlighted). Notably, the example from the “with-system” condition includes more specific details to resonate with users.

hooks. Thus, they removed the unexplained terminology or hard-to-understand acronyms. For example, P10 replaced the acronym of “ISPs” from the LLM-outputted hook with “Internet Service Providers.” They had concerns that the system might overlook requirements after chaining too much stuff. Also, they edited the hook for conciseness by cutting off extra questions and wordy introductions.

For R2 (including relatable and specific examples), several writers said that the LLM output felt robotic and rigid, thus making it less engaging (P1, P2, P5). For example, P1 mentioned that when they read the LLM-generated hook, they felt it would not interest readers. Also, P2 shared that the first sentences in many LLM-generated hooks felt like news headlines, which read like some emotionless statements. Thus, they edited the tone to become funnier and more personable. Also, P10 shared that they changed the time-related examples inside the hook as LLMs sometimes lacked updated information. Hence, they replaced the LLM output with a more recent example.

For R3 (sparking curiosity), P1 and P4 shared that they know what makes a tweet go viral and get clicks from their past Twitter experiences: using exaggeration, shock factors, and potentially misleading information. Then, P4 prepended “Apparently we’re gonna lose \$10.5 trillion to criminals over the internet by 2025. Isn’t that horrendous?” to the LLM-generated hook on cybercrime. They believed the addition of surprising data would attract readers more.

5. Users Edit LLM Hooks for Personal Style Users also edited the LLM-generated hooks to make them fit more according to their writing styles and favorite examples so they felt more connected and related to their hooks. For example, P10 shared that they wanted to use the exact syntax they used daily in this hook. So they changed a lot of word-level choices like from “Do you know what” to “Have you ever heard.” P10 also shared that they intentionally deleted words like “exactly” and split the two questions which were originally in one sentence into two separate short ones. From this, P10 shared that it made them feel that the hook sounded like themselves or their friends by referring to their usual language choices. Also, P1 and P10 edited all of the LLM-generated hooks when they reached Step 6, even though they stated they were already highly satisfied with them. They still expressed wanting to embed more of their styles inside the hook. P4 suggested that making these changes helped maintain their own voice, and P6 specifically added several hashtags and emojis as they liked them. According to P8, engaging in the final editing of the hooks helped them feel

a greater sense of agency and ownership over them. This was because they perceived the final product as being more original after undergoing the editing process. P8 specifically mentioned while editing, they shifted from the role of “creator” to the “first reader” of the hooks. By doing so, they gained a more objective and distant view of their writings.

Discussion and Future Works

In this paper, we demonstrate that LLMs can help contextualize technical information into relatable and engaging hooks. We scaffold the complex Tweakorial hook writing process by prompting LLMs for everyday examples, common experiences, and specific anecdotes. This scaffolding approach (MacNeil et al. 2021) helps STEM experts effectively communicate science to non-technical audiences. In the future, it is possible that similar tools could be built for other groups of experts, such as helping journalists reach younger audiences, helping medical professionals explain procedures to patients, or helping public service organizations spread messages to under-served communities.

However, LLMs are far from perfect and user interaction is essential to producing successful hooks. LLMs sometimes provide inaccurate examples for a topic and sometimes suggest experiences that a non-technical audience would not relate to, such as building a website or buying something on the dark web. Ultimately, the expert must decide whether the suggestions are correct and appropriate, and they cannot just “trust the machine.” Experts have the ability to judge whether the examples of the technology are correct (such as verifying that Spotify Wrapped does indeed use an API), but they might not understand non-technical audiences well enough to evaluate whether the suggested experiences resonate with them (such as being aware of a lawsuit between Oracle and Google). If an expert is unsure whether something would resonate with the public, they should ask members of their audience. One feature that could be built into such a system is to get human judgments from an online marketplace to provide audience feedback on demand.

Although LLMs have a wealth of information, they do contain biases and not all viewpoints are equally represented. For explaining science to the general public, the biases in the current LLMs like GPT-3 and GPT-4 are probably not problematic. However, if the intended audience were a more specific demographic, LLMs might not suggest examples and experiences that resonate with them. People of different ages, cultural backgrounds, education levels, language abilities, and geographic locations communicate very

differently. For example, an experience about using a laptop might not resonate with a low-income student who cannot afford a laptop and does all of their computing from a phone. Currently, LLMs mostly echo dominant perspectives, but it could be powerful to train LLMs to elevate the voices of non-dominant groups as a means to bridge the gap, better support the communities, and promote inclusivity.

In the study, users stated that it was important to them that their final hooks reflected their own personal style and creativity. This is in line with previous work on the social dynamics of AI co-creative systems (Gero, Long, and Chilton 2023) which has shown that when working with LLMs, writers care deeply about preserving their *intent* and the *authenticity* of their writing. To further enable this, some users suggested future versions of the system where writers can feed their hooks back to their system to “keep” their style for future generations, or add a “temperature” parameter to control the specificity of contextualized examples. These features can provide a range of agency when co-creating hooks with LLMs, thus aligning with the future vision of designing more user-focused interactive creativity support tools. These designs can empower users in their content creation process by fostering a sense of ownership and creative expression.

Conclusion

This paper explores integrating generative AI into the hook writing process for Tweetorials, a science communication method that motivates science through relatable examples and experiences. Our prompting engineering study suggests that including examples of good hooks in the prompt helped LLMs generate better hooks, but there is still a need for humans in the loop. To help experts write hooks, we built an LLM-based workflow that scaffolds the process: given a topic, the system suggests everyday examples of the topic, and the user can accept a machine suggestion, edit a machine suggestion, request more suggestions, or write their own. Based on the everyday example selected by the user, the system suggests common experiences. The user can again accept, edit, regenerate, or write their own. Based on the common experience selected, the system suggests a personal anecdote and can make the anecdote more specific while the user may edit these as well. Finally, the system produces an example hook that users can accept as is, or reference when finalizing their hook. Our user study shows this scaffolding greatly reduces the cognitive load of writing hooks. Also, as the outputs are editable at every stage, the hooks still convey the writer’s authentic style, voice, and experiences.

Author Contributions

TL finalized the prompt engineering works, built the system, led the annotation and user study, analyzed the results, and wrote the paper. DZ led the early prompt engineering and task understanding and contributed to the system and data collection. GL, BT, SM, and KS assisted with the early task understanding, data collection, and analysis. SW, KG, and LC provided overall guidance on the project, helped shape the two studies, and contributed to the writing.

Acknowledgments

This work has been supported by NSF-IIS-2129020 and NSF-EAR-2121649.

References

- [Aldous, An, and Jansen 2019] Aldous, K. K.; An, J.; and Jansen, B. J. 2019. The Challenges of Creating Engaging Content: Results from a Focus Group Study of a Popular News Media Organization. In Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, 1–6. Glasgow, Scotland, UK: Association for Computing Machinery.
- [Bender et al. 2021] Bender, E. M.; Gebru, T.; McMillan-Major, A.; and Shmitchell, S. 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, 610–623. Virtual Event Canada: Association for Computing Machinery.
- [Breu 2019] Breu, A. C. 2019. Why Is a Cow? Curiosity, Tweakorials, and the Return to Why. New England Journal of Medicine 381(12):1097–1098.
- [Breu 2020] Breu, A. C. 2020. From Tweetstorm to Tweakorials: Threaded Tweets as a Tool for Medical Education and Knowledge Dissemination. Seminars in Nephrology 40(3):273–278.
- [Brown et al. 2020] Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs]. arXiv: 2005.14165.
- [Brüggemann, Lörcher, and Walter 2020] Brüggemann, M.; Lörcher, I.; and Walter, S. 2020. Post-normal science communication: exploring the blurring boundaries of science and journalism. Journal of Science Communication 19(03):A02.
- [Bukhtiyarov and Gusev 2020] Bukhtiyarov, A., and Gusev, I. 2020. Advances of transformer-based models for news headline generation.
- [Bullock et al. 2019] Bullock, O. M.; Colón Amill, D.; Shulman, H. C.; and Dixon, G. N. 2019. Jargon as a barrier to effective science communication: Evidence from metacognition. Public Understanding of Science 28(7):845–853.
- [Calderwood, Wardrip-Fruin, and Mateas 2022] Calderwood, A.; Wardrip-Fruin, N.; and Mateas, M. 2022. Spinning coherent interactive fiction through foundation model prompts. In ICCC, 44–53. Association for Computational Creativity (ACC).
- [Chung et al. 2022] Chung, J. J. Y.; Kim, W.; Yoo, K. M.; Lee, H.; Adar, E.; and Chang, M. 2022. Talebrush: Sketching stories with generative pretrained language models. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, CHI ’22. New York, NY, USA: Association for Computing Machinery.

- [Crist 2019] Crist, M. 2019. “I’ve been watched the landscape of my childhood burn with an aching heart, and wondering how much climate change is to blame. Turns out human activity is a major driver of California’s wildfires, but not just in the ways you might imagine.”. <https://twitter.com/meehancrist/status/1197527975379505152>. [Online; accessed June-2022].
- [Emig 1977] Emig, J. 1977. Writing as a Mode of Learning. *College Composition and Communication* 28(2):122–128. Publisher: National Council of Teachers of English.
- [Flower and Hayes 1981] Flower, L., and Hayes, J. R. 1981. A Cognitive Process Theory of Writing. *College Composition and Communication* 32(4):365.
- [Gero et al. 2021] Gero, K. I.; Liu, V.; Huang, S.; Lee, J.; and Chilton, L. B. 2021. What makes tweetorials tick: How experts communicate complex topics on twitter. *Proc. ACM Hum.-Comput. Interact.* 5(CSCW2).
- [Gero, Liu, and Chilton 2022] Gero, K. I.; Liu, V.; and Chilton, L. 2022. Sparks: Inspiration for science writing using language models. In *Designing Interactive Systems Conference, DIS ’22, 1002–1019*. New York, NY, USA: Association for Computing Machinery.
- [Gero, Long, and Chilton 2023] Gero, K. I.; Long, T.; and Chilton, L. B. 2023. Social dynamics of AI support in creative writing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI ’23*. New York, NY, USA: Association for Computing Machinery.
- [Hart and Staveland 1988] Hart, S. G., and Staveland, L. E. 1988. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Human mental workload* 1(3):139–183.
- [Hayes 1996] Hayes, J. R. 1996. A new framework for understanding cognition and affect in writing. In *The Science of Writing: Theories, Methods, Individual Differences, and Applications*. Lawrence Erlbaum Associates.
- [Holtzman et al. 2020] Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2020. The Curious Case of Neural Text Degeneration. [arXiv:1904.09751 \[cs\]](https://arxiv.org/abs/1904.09751).
- [Howell et al. 2019] Howell, E. L.; Nepper, J.; Brossard, D.; Xenos, M. A.; and Scheufele, D. A. 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):e0216274.
- [Ippolito et al. 2019] Ippolito, D.; Kriz, R.; Kustikova, M.; Sedoc, J.; and Callison-Burch, C. 2019. Comparison of Diverse Decoding Methods from Conditional Language Models. [arXiv:1906.06362 \[cs\]](https://arxiv.org/abs/1906.06362). [arXiv: 1906.06362](https://arxiv.org/abs/1906.06362).
- [Jigsaw 2017] Jigsaw. 2017. Sideways dictionary. <https://sidewaysdictionary.com/#/>.
- [Jin et al. 2020] Jin, D.; Jin, Z.; Zhou, J. T.; Orii, L.; and Szolovits, P. 2020. Hooks in the headline: Learning to generate headlines with controlled styles. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 5082–5093*. Online: Association for Computational Linguistics.
- [MacNeil et al. 2021] MacNeil, S.; Ding, Z.; Quan, K.; Parashos, T. j.; Sun, Y.; and Dow, S. P. 2021. Framing creative work: Helping novices frame better problems through interactive scaffolding. In *Creativity and Cognition*. New York, NY, USA: Association for Computing Machinery.
- [McClain and Neeley 2014] McClain, C., and Neeley, L. 2014. A critical evaluation of science outreach via social media: Its role and impact on scientists. *F1000Research*.
- [Petridis et al. 2023] Petridis, S.; Diakopoulos, N.; Crowston, K.; Hansen, M.; Henderson, K.; Jastrzebski, S.; Nickerson, J. V.; and Chilton, L. B. 2023. Anglekindling: Supporting journalistic angle ideation with large language models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI ’23*. New York, NY, USA: Association for Computing Machinery.
- [Radford et al. 2019] Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8):9.
- [Shearer and Matsa 2018] Shearer, E., and Matsa, K. E. 2018. News Use Across Social Media Platforms 2018. *Pew Research Center*.
- [Singh et al. 2022] Singh, N.; Bernal, G.; Savchenko, D.; and Glassman, E. L. 2022. Where to hide a stolen elephant: Leaps in creative writing with multimodal machine intelligence. *ACM Trans. Comput.-Hum. Interact.* Just Accepted.
- [Wang et al. 2023] Wang, S.; Petridis, S.; Kwon, T.; Ma, X.; and Chilton, L. B. 2023. Popblends: Strategies for conceptual blending with large language models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI ’23*. New York, NY, USA: Association for Computing Machinery.
- [Wu, Terry, and Cai 2022] Wu, T.; Terry, M.; and Cai, C. J. 2022. AI Chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, CHI ’22*. New York, NY, USA: Association for Computing Machinery.
- [Xu et al. 2019] Xu, P.; Wu, C.-S.; Madotto, A.; and Fung, P. 2019. Clickbait? sensational headline generation with auto-tuned reinforcement learning.
- [Yeo 2015] Yeo, S. K. 2015. Public engagement with and communication of science in a web-2.0 media environment. *Washington, DC: The American Association for the Advancement of Science (AAAS)*.

Appendix

Due to the page limits, a high-resolution source appendix is linked here: <https://tinyurl.com/tweetappendix>

Score	1	2	3	4	5
<p>Jargon-Free:</p> <p>Does the hook avoid jargon or unexplained terminology so the general audience can understand it easily?</p>	<p>Very hard to understand; includes unexplained terminology and most readers cannot understand</p> <p>Example: Website takes forever to load? That's because domain name servers (DNS) are down or not responding. DNS servers are used to translate web addresses (like Google.com) into IP addresses that can be read and understood by computers. 1/</p> <p>Reason: "IP addresses", "translate web addresses", and "domain name servers" are all words most everyday people don't know. These words makes the meaning of this hook unclear.</p>	<p>Somewhat hard to understand; includes unexplained terminology but some readers can get the gist</p> <p>Example: Slow your internet has been while streaming your favorite show? What if Random Internet Congestion was actually caused by someone else? Sure might make you mad. So what's a DDoS attack, and how can you protect yourself? Read on to find out! 1/</p> <p>Reason: "Random Internet Congestion" and "DDoS attack" are words most everyday people don't know. The reader must dig deep into context clues to understand the hook.</p>	<p>Somewhat easy for most readers to understand; includes unexplained terminology but most readers can understand</p> <p>Example: With 2 Factor Authentication, you not only have to know the password to get into an account, you also need a second piece of information that only you should know. 1/</p> <p>Reason: This mentions an unfamiliar term first, but follows up with minimal explanation. "piece of information" is also unclear.</p>	<p>Somewhat easy for all readers to understand; includes unexplained terminology but all readers can understand</p> <p>Example: Have you ever had to submit your private medical information to your insurance? How do they keep it secure, yet allow more people to access it? Deidentification is a process that can make sure our information remains private – but how does it work? 1/</p> <p>Reason: "Deidentification" is a word most people don't know, but general context is provided so the reader can guess the meaning. However, a precise definition isn't given.</p>	<p>Very easy for all readers to understand; does not have any unexplained terminology</p> <p>Example: Have you ever forgotten your password? Gmail sends you a text message with a long number in it, and helps you stay safe! Let's read #2FactorAuthentication.</p> <p>Reason: "password", "Gmail", and "text message" are all terms people are familiar with. Meaning of hook is clear. "#2FA" is jargon but it only shows at the end and serves as an opening for the following tweets.</p>
<p>Specific & Reliable Example(s):</p> <p>Does the hook include a specific and reliable example(s) about the topic?</p>	<p>Has NO example</p> <p>Example: What are Domain Name Servers & why do they matter? Here's everything you need to know:</p> <p>Reason: There is no example.</p>	<p>Provides an example that is extremely not specific or reliable</p> <p>Example: Data security is important! #Deidentification strips personal info from data while preserving its structure and insights. Have you ever wondered how companies can analyze data without compromising your privacy? #DataPrivacy</p> <p>Reason: There is one example, "stripping personal info from data", but it is not specific or reliable.</p>	<p>Provides an example that is both somewhat specific & reliable</p> <p>Example 1: Have you ever upload a file to the internet and wondered how it gets from your computer to the site's server? Cloud Computing! Example 2: I recently witnessed the power of exploiting vulnerabilities first-hand when my team of hackers used a buffer overflow attack to gain access to an exposed Windows 2003 server. Here's an exploration of the world of hacking: 1/</p> <p>Reason: For example 1, "upload a file" is a reliable example but it is not specific. For example 2, "buffer overflow attack of a Windows 2003 server" is specific, but it's not reliable.</p>	<p>Provides an example that is specific & reliable for many readers, but not all</p> <p>Example: Have you ever been tempted to try to hack something? Recently, I had a friend try to access my math teacher's laptop in an attempt to improve his grade. But in the end, was it really worth it? Regardless of the outcome, I'm still curious as to why some people would risk so much to hack! Let's explore together. #hacking 1/</p> <p>Reason: The example is specific and reliable. But only some readers may connect with it, not all.</p>	<p>Provides an example that is specific & reliable for almost all readers</p> <p>Example: I once tormented Last Week Tonight -- then my landlord got a complaint from Comcast! Wth? My friends never got caught. Ugh. So here are things I wished I had known about how to be sneaky on the internet: 1/</p> <p>Reason: The example is intriguing and this experience is common and reliable to almost everyone. Also, "Last Week Tonight", "landlord", "Comcast", and "my friends" are detailed enough to make people feel vivid.</p>
<p>Sparks Curiosity:</p> <p>Does the hook give readers a specific and driving reason to keep reading to satisfy their curiosity?</p>	<p>The tweet does not generate curiosity for readers, OR it has a good question, but it is answered in detail.</p> <p>Example: With 2 Factor Authentication, you not only have to know the password to get into an account, you also need a second piece of information that only you should know. 1/</p> <p>Reason: This is a statement; there is no question. It also directly explains the term and doesn't prompt further questioning.</p>	<p>The tweet may generate mild curiosity for a small group of people.</p> <p>Example: When someone floods your website with too much traffic that it crashes, that's what's known as a #DDoSAttack. Have you ever had to deal with a similar scenario? What did you do? #CyberSecurity #Websecurity</p> <p>Reason: Only a select group may be curious about crashing a website through too much traffic. Hook doesn't present a specific or urgent question.</p>	<p>The tweet generates some curiosity for readers, OR it has a good question but provides too much of an answer</p> <p>Example 1: Have you ever uploaded a file to the internet and wondered how it gets from your computer to the site's server? Cloud Computing is the answer! Example 2: My son relies on his Alexa to help with his math homework every single night. While I am concerned about his learning, I am interested in how it works.</p> <p>Reason: These hooks have questions and examples to intrigue people, but they are not specific enough to make people feel very curious.</p>	<p>The tweet may generate curiosity for many readers, but not all</p> <p>Example: I just bought some things online, but how do I know the website I'm using is safe? Without HTTPS, anyone can intercept the data I send out...but how does HTTPS keep me protected when I'm online shopping? Here's what I've learned so far about online security: 1/</p> <p>Reason: The hook identifies general questions that many people might have, but does not have a specific question.</p>	<p>The tweet instills curiosity and makes you want to read more.</p> <p>Example 1: I once tormented Last Week Tonight -- then my landlord got a complaint from Comcast! Wth? My friends never got caught. Ugh. So here are things I wished I had known about how to be sneaky on the internet: Example 2: My son relies on his Alexa to help with his math homework every single night. While I am concerned about his learning, I am interested in how Alexa understands what he is saying? Is it the same way that humans understand language?</p> <p>Reason: These hooks provide the reader with a specific question that grabs their attention and makes them want to continue reading.</p>

Figure 6: A five-point scale rubric for annotators in the prompt engineering study

Ransomware	Database	Browser Hijacking	Machine Learning	API*	Patch	White Hat	Programming Language*	Trojan	Ad Blocking
Front End*	Peer-To-Peer	Net Neutrality*	Internet Service Provider	Tor	Black Hat	BitTorrent	Secure Socket Layer	Cybercrime*	Troll
Back End	Browser	Breach Notification	Access Control List	Dark Web	OAuth	Security Key	Autocomplete*	Backdoor	Spoofing

Table 3: 30 topic inputs for the prompt engineering study (*asterisked are the six topics randomly chosen for the user study)

Strategy	Lowest-scoring hook	Lowest-scoring hook with same topic as highest-scoring hook	Highest-scoring hook
PS1	<p>Application Programming Interface (API)</p> <p>A #programming interface that lets you calculate prices, map distances, and get stock quotes automatically? What else can an #API do and how can it help me?</p> <p>Average Score: 1.89</p>	<p>Browser Hijacking</p> <p>A hacker could take control of your online activity and webpages without you realizing it! Ever heard of browser hijacking? Have you taken the necessary steps to protect yourself? #browserhijacking #cybersecurity #hackerrisk</p> <p>Average Score: 2.89</p>	<p>Browser Hijacking</p> <p>Beware of browser hijacking! Ever gone to a website you didn't type in and been redirected somewhere else? That's browser hijacking! It can be used for malicious purposes - so how can you stay safe? #browserhijacking #security</p> <p>Average Score: 4.00</p>
PS2	<p>The Onion Router (TOR)</p> <p>Ever hear about the deep web and the black markets? Well, TOR is the gateway to them! What's required to risk taking a journey into the unknown? A thread on TOR and its implications:</p> <p>Average Score: 1.67</p>	<p>Internet Service Provider</p> <p>Have you ever wondered why you get more slow internet in your bedroom than at the dining table? What is it that your internet service provider does differently to affect your internet connection? Let's look into it: 1/</p> <p>Average Score: 3.44</p>	<p>Internet Service Provider</p> <p>Have you ever had an internet outage at the exact moment when you need to send an assignment or video chat with friends? Who's responsible for this? How are we connected to the internet? I'm about to explore what happens behind the scenes of the internet. Let's start by understanding the main characters involved - the Internet Service Providers. Ready?</p> <p>Average Score: 4.55</p>
PS3	<p>Back End</p> <p>Have you ever wanted to keep your data secure and away from prying eyes? You might not realize it, but controlling access to your accounts is just as important as data privacy. But how can you do this? My recent experience with Amazon Web Services' Identity and Access Management feature has shown me the power of properly managing back end access. Let's discover more! 1/</p> <p>Average Score: 2.11</p>	<p>Internet Service Provider</p> <p>Have you ever wondered why an Internet Service Provider (ISP) could make promises that never turn out to be true? I recently learned the hard-way when my streaming experience got slower even though I had chosen a top-tier plan from a big-name provider. Here's an exploration of how ISP's work and why you should take their promises with a grain of salt. #ISP #Technology #Internet #Promises 1/</p> <p>Average Score: 3.55</p>	<p>Internet Service Provider</p> <p>I used to think living in the suburbs would mean better internet connection. But my Verizon Fios service was so unreliable and slow that I felt like I was back in the dark ages! What can we do to better understand the nature of internet service providers, and how can they provide truly reliable service? Here's the story:</p> <p>Average Score: 4.78</p>

Table 4: Collection of good and bad hooks from the prompt engineering study