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

Tao Long, Dorothy Zhang, Grace Li, Batool Taraif, Samia Menon, Kynnedy Simone Smith, Sitong Wang, Katy Ilonka Gero, Lydia B. Chilton

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 scientists to engage in more personal style conversations to take on a more relatable, interesting, and 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 Neely 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, and creative writing. 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 (Caldwell, 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 Tweetorial headlines. Tweetorial 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 Tweetorial Hooks

Tweetorials 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 Tweetorial 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 Tweetorial hook:

- **R1 - Jargon-Free**: Does the hook avoid jargon or explained 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 Tweetorial hooks for computer science topics that exhibit all these properties:

- **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.”

![Figure 1: A Tweetorial 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 Tweetorial examples can be found on our website:](http://language-play.com/tech-tweets/annotations)
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 (“Why? 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 experiences, 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.

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.

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 PS2 and PS3 did help LLMs “learn” 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 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 Cybercrime. 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.
Step 1: Please input the topic.

**VPN**

Here are 5 everyday examples about that:
1. Connecting to a work office server from home
2. Using a public Wi-Fi connection in a restaurant or café
3. Accessing a video streaming service while traveling abroad
4. Keeping your web browsing anonymity and security
5. Securing data while using a shared or public network

Generate Everyday Examples

Step 2: Generate Common Experiences.

Please enter your favorite everyday example here.

**Receving data while using a shared or public network**

Generate Common Experiences

Then, the 5 common experiences about that are:
1. Data interception and snooping
2. Malware infection through malicious websites
3. Phishing attempts to steal sensitive information
4. Unauthorized access to computers, networks, data, and accounts
5. Unauthorized changes to settings and configurations

Step 3: Generate Personal Anecdotes.

Please enter your favorite common experience here.

**Receiving data while using a shared or public network**

Generate Personal Anecdote

Then, here are the 3 personal anecdotes about that:
1. Once, I was using a VPN to access my company documents when a suspicious email suddenly popped up in my inbox. I knew it wasn't associated with work, so I quickly shut down the VPN connection before opening the email.
2. Recently, I was connected to my company's VPN when I noticed usage activity on my browser. It seemed like someone was trying to access my private documents, so I quickly disconnected the VPN and reported the incident to my IT department.
3. While I was connected to my company's VPN, I received an alert that told me there was a phishing attempt on my secure network. As soon as I saw the warning, I immediately disconnected the VPN and notified my colleagues about the potential threat.

Step 4: Generate a More Specific Anecdote.

Please enter your favorite personal anecdote here.

**Receiving data while using a shared or public network**

Generate A Specific Anecdote

While I was connected to my company's VPN, I received an alert that told me there was a phishing attempt on my secure network. As soon as I saw the warning, I immediately disconnected the VPN and notified my colleagues about the potential threat...

Step 5: Let GPT Generate an Example Hook.

Please enter the specific personal anecdote here.

**Generating personal anecdote**

Generate An Example Hook

I also contacted our IT team to help us combat the phishing attempts. While I was logged into my company's secure network via the virtual private network (VPN), I received an alert notifying me of a suspicious phishing email that had been sent to one of our employees. Immediately, I disconnected the VPN and reached out to my colleagues to warn them of the potential threat. I then contacted our IT team to assist in gathering more information and developing a strategy to combat the attack.

Step 6: Write Yourself a Final Hook.

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

**VPN (Virtual Private Network)**

You have ever received a suspicious email that could have threatened your security. If you know the best way to protect yourself against these threats? I've got the answer - Virtual Private Networks (VPN)! Read on to see how they work and why they are important for online security. #EmailSecurity #VPN

Submit

Figure 4: An example of how users write a Tweetorial hook about VPN (Virtual Private Network) with our tool. They can follow the workflow from top to bottom or return to previous steps and start again. They can regenerate, modify, and accept the LLM outputs or use their own responses.

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) 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.

**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.

<table>
<thead>
<tr>
<th>TLX Dimension</th>
<th>With System</th>
<th>Without System</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mental Demand</td>
<td>2.87</td>
<td>4.00</td>
<td>0.004**</td>
</tr>
<tr>
<td>Effort</td>
<td>2.87</td>
<td>4.40</td>
<td>0.002**</td>
</tr>
<tr>
<td>Performance</td>
<td>5.73</td>
<td>4.50</td>
<td>0.001**</td>
</tr>
<tr>
<td>Frustration</td>
<td>1.93</td>
<td>2.77</td>
<td>0.02**</td>
</tr>
<tr>
<td>Physical Demand</td>
<td>1.10</td>
<td>1.37</td>
<td>0.08</td>
</tr>
<tr>
<td>Temporal Demand</td>
<td>2.37</td>
<td>2.40</td>
<td>0.598</td>
</tr>
</tbody>
</table>

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). 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 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 pretended “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

<table>
<thead>
<tr>
<th>Table 2: Collection of hooks generated under both the with-system and without-system conditions from the user study</th>
<th><strong>Without the System</strong></th>
<th><strong>With the System</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application Programming Interface (API)</strong></td>
<td>Have you heard about the huge Oracle and Google lawsuit but had no idea what it was about? What the hell even is an API, and why is it so important that they can’t be copyrighted? A guide for Supreme Court Justices (and you)</td>
<td>Ever wonder why your Spotify Wrapped is so fun? How do they know which artists and songs to highlight and recommend? Find out how Spotify’s developer tools can help analyze user listening history and trends to make tailored music content in this thread:</td>
</tr>
<tr>
<td><strong>Cybercrime</strong></td>
<td>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/</td>
<td>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:</td>
</tr>
</tbody>
</table>
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 Tweetorial 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 as 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.
Appendix

Due to the page limits, a high-resolution source appendix is linked here: [https://tinyurl.com/tweetappendix](https://tinyurl.com/tweetappendix)

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

<table>
<thead>
<tr>
<th>Score</th>
<th>Jargon-Free</th>
<th>Somewhat hard to understand, includes unexplained terminology and most readers cannot understand</th>
<th>Somewhat easy for most readers to understand; includes unexplained terminology but most readers can understand</th>
<th>Somewhat easy for all readers to understand; includes unexplained terminology but all readers can understand</th>
<th>Very easy for all readers to understand; does not have any unexplained terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Very hard to understand; includes unexplained terminology and most readers cannot understand</td>
<td>Example: Websites take forever to load! That's because domain names aren't DNS servers are down or not responding. DNS servers are used to translate addresses like Google.com into IP addresses that can be read and understood by computers.</td>
<td>Example: Slow your internet has been listed while uninstalling your favorite apps! What if Random Internet Congestion was actually caused by someone else? Sure might make you mad. What's a DDoS attack, and how can you prevent yourself? Read on to find out!</td>
<td>Example: Random Internet Congestion and DDoS attack are words no most everyday people don't know. The reader must dig deep into the book to understand the book's unexplained terminology.</td>
<td>Example: This question unanswerable term first, but follows up with minimal explanation. &quot;price of information&quot; is also unclear.</td>
</tr>
<tr>
<td>2</td>
<td>Little hard to understand, includes some unexplained terminology</td>
<td>Example: The data security is important! Data encryption and secure storage of data is a must to keep your information private. Have you ever won an interesting competition? Have you ever won an interesting competition? Have you ever won an interesting competition? Have you ever won an interesting competition?</td>
<td>Example: Data security is important! Data encryption and secure storage of data is a must to keep your information private.</td>
<td>Example: Reason: There is no example.</td>
<td>Example: Reason: This means unanswerable term first, but follows up with minimal explanation. &quot;price of information&quot; is also unclear.</td>
</tr>
<tr>
<td>3</td>
<td>Somewhat hard to understand, includes unexplained terminology</td>
<td>Example: What are Domain Name Server why do they matter? Here's everything you need to know.</td>
<td>Example: Security data is important! Data encryption and secure storage of data is a must to keep your information private.</td>
<td>Example: Does this mean unanswerable term first, but follows up with minimal explanation. &quot;price of information&quot; is also unclear.</td>
<td>Example: Provides an example that is somewhat specific &amp; reliable</td>
</tr>
<tr>
<td>4</td>
<td>Somewhat easy for most readers to understand; includes unexplained terminology but most readers can understand</td>
<td>Example: Data security is important! Data encryption and secure storage of data is a must to keep your information private. Have you ever won an interesting competition? Have you ever won an interesting competition? Have you ever won an interesting competition? Have you ever won an interesting competition?</td>
<td>Example: Reason: This provides an example that is somewhat specific &amp; reliable</td>
<td>Example: Provides an example that is somewhat specific &amp; reliable</td>
<td>Provides an example that is somewhat specific &amp; reliable</td>
</tr>
<tr>
<td>5</td>
<td>Very easy for all readers to understand; does not have any unexplained terminology</td>
<td>Example: Does this mean unanswerable term first, but follows up with minimal explanation. &quot;price of information&quot; is also unclear.</td>
<td>Example: Reason: This means unanswerable term first, but follows up with minimal explanation. &quot;price of information&quot; is also unclear.</td>
<td>Example: Provides an example that is somewhat specific &amp; reliable</td>
<td>Example: Provides an example that is somewhat specific &amp; reliable</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Lowest-scoring hook</th>
<th>Lowest-scoring hook with same topic as highest-scoring hook</th>
<th>Highest-scoring hook</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS1</td>
<td>Application Programming Interface (API)</td>
<td>Application Programming Interface (API)</td>
<td>Application Programming Interface (API)</td>
</tr>
<tr>
<td>PS2</td>
<td>The Onion Router (TOR)</td>
<td>The Onion Router (TOR)</td>
<td>The Onion Router (TOR)</td>
</tr>
<tr>
<td>PS3</td>
<td>Back End</td>
<td>Back End</td>
<td>Back End</td>
</tr>
</tbody>
</table>

| Table 4: Collection of good and bad hooks from the prompt engineering study |
|---------------------|---------------------------------|---------------------|---------------------|---------------------|---------------------|
| Ransomware          | Database                        | Browser Hijacking   | Machine Learning    | API*               | Trojan              |
| Front End*          | Peer-To-Foot*                   | Internet Service API| Tor                 | Patch White Hat    | Ad Blocking         |
| Back End            | Browser Notification            | Access Control List | Dark Web            | tIauth Security Key| Autocomplete*       |

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

| Table 4: Collection of good and bad hooks from the prompt engineering study |
|---------------------|---------------------------------|---------------------|---------------------|---------------------|---------------------|
| Ransomware          | Database                        | Browser Hijacking   | Machine Learning    | API*               | Trojan              |
| Front End*          | Peer-To-Foot*                   | Internet Service API| Tor                 | Patch White Hat    | Ad Blocking         |
| Back End            | Browser Notification            | Access Control List | Dark Web            | tIauth Security Key| Autocomplete*       |
References


[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].


[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...


