

GenAI and the Language Arts

Rebecca Ellis

“Technical writer head counts will be replaced with AI engineers.” These words from an esteemed colleague primed me for what was ahead: massive layoffs at the leading tech company I worked for as a vendor. And by the end of 2023, I was next.

In 2023, 262,242 employees were laid off in 1185 tech companies according to Layoffs.fyi and TechCrunch in a post-COVID restructuring frenzy. The majority of layoffs affected software engineers rather than technical writers, but tech writers are somewhat of a minority niche in IT.

My personal experience seems to be the strong indicator that the impact on tech writers was disproportionate and that my colleague’s doomsday warning is likely not far off base. I am a member of the Slack channel Write the Docs, a forum for technical writers and content engineers. Lately, it has been swarming with out-of-work tech writers promoting their portfolios and asking their peers for revisions.

In just the last quarter, I sent an average of 200 resumes to potential employers per month. I started in October to beat the holiday lull. Thanksgiving, Hanukkah, and Christmas came and went. While finding a job is normally an act of free will, of diligence and tenaciousness, I found myself wishing for a miracle with each passing holiday. A few more resumes were sent before I found myself unemployed by New Year’s Day.

On one hand, it is true that AI-augmented processes speed up many mundane processes, such as formatting in markdown, building nested indices, summarizing, or creating glossaries. These generative AI capabilities offload from technical writers some of the tedious tasks that often bog us down. In an ideal world where technical writers would truly act as humans in the loop, generative AI could save us time to focus our attention on quality content, proofreading, revision, learning and development. But in this hard world of the bottom line we are increasingly seen as replaceable.

So adaptability it is. Throughout my job search, I have been exploring the tech writer versus AI engineer dilemma motivated by the old adage “if you can’t beat ‘em, join ‘em.” I embarked on my own genAI experimentation journey, exploring how generative AI can be used to make certain documentation processes, like building topic lists and transforming plain text into markdown, easier. I am taking coursework on the OpenAI community forum in prompt engineering, LLMs, generative AI, and tooling with Langchain (a large language model interface).

I am learning that large language models, or LLMs, are the engines behind applications like ChatGPT. They are designed to be an AI interface with other software, code subroutines, and applications but developers interact with them using natural human language. LLMs are growing in their capacity beyond just interacting with text but can now interpret images and other types of content.

As I am reskilling and upskilling, I can't shoo away the nagging suspicion that if tech writers are in fact being replaced by AI engineers, the IT industry is committing a grave error. And the more I learn about how generative AI and LLMs work, the more I grow apprehensive about how the tech industry will truly be able to harness their power.

As Isa Fulford and Andrew Ng from OpenAI noted in their prompt engineering presentation, LLMs, like any other model, have limitations. They are trained on a lot of data but it is not trained on what it does NOT know (a contradiction that is impossible to fulfill):

“Even though the language model has been exposed to
a vast amount of knowledge during its training process, it has not perfectly
memorized the information it's seen, and so, it doesn't know the boundary
of its knowledge very well.”

This is a limitation of generative AI that leads to erroneous answers, or hallucinations. Developers are taking strides to improve these limitations, but LLM hallucinations still produce errors that necessitate human intervention. Experts in the language arts are needed to counteract these errors by crafting well-written AI prompts and reviewing outputs with an editor's eye for detail to verify the authenticity of language model responses.

To be good writers, and have the creativity to detect language model limitations, content developers need to be versed in the language arts – that is, the humanities. This humanities-trained expertise includes the abilities to fact check, to intuitively reason, and to discern the limits of a system – skills that will still be required by tech companies for a long time to come in order to safeguard quality in interacting with AI technologies. Tech companies, often with patchwork internal architectures, must maintain their ability to recognize where knowledge is lacking, and ensure that their technical documentation captures and fills these knowledge gaps, as well as reflects the current state and software versions included in their techstacks and infrastructures. Technical documentation is still playing catch-up in documenting onboarding processes to relatively new technologies that interface with AI.

For specific, one-off practical tasks ChatGPT, the web user interface of the OpenAI LLM, has proven quite helpful because it provides a user-friendly, completely codeless interface. Chat interfaces based on powerful LLMs provide an opportunity for tech writers and content developers to integrate better with web and backend developers.

With genAI applications like ChatGPT, however, there is often the assumption that the data used to train it is so vast that its response must be close to all-encompassing, neutral, and even could qualify as generic knowledge – akin to “just Google the answer.” However, the data used to train LLMs has been often shown to be biased.

This bias can be introduced when interacting with LLMs to train chat applications, a process which requires the exploration and definition of roles. LLM developers only interact with the assistant and user roles when developing genAI chat applications, but the system role has already been pre trained and is thus not transparent. So the question is, how (with what content or messages) are the system roles trained? To confront this problem, content experts trained in the language arts can be integrated earlier on in the development of such applications to help train them by feeding it with system messages, and thus help ensure quality, democratic development processes, and transparency. By retaining content experts rather than sacrificing them, tech companies can benefit from their linguistic training to interact with LLMs when developing genAI applications.

Role

```
messages =  
[  
  {"role": "system",  
   "content": "You are an assistant... "},  
  {"role": "user",  
   "content": "tell me a joke "},  
  {"role": "assistant",  
   "content": "Why did the chicken... "},  
  ...  
]
```

system

assistant



user

(Source: OPENAI:ChatGPT Prompt Engineering for Developers)

LLMs are more powerful than machine learning models. They can take structured and unstructured input and cumulatively learn, hence the term used for their applications, generative AI. What's more, LLMs can handle many different types of processes, each of which, in a traditional machine learning model, requires a separate algorithm and a separate model. LLMs thus prove the power of human language. Just as writers and poets have historically been the linguistic guardians of culture in all human societies, technical writers and content developers at tech companies, even in the AI age, should similarly serve as the linguistic stronghold between app developers and LLMs. Rather than being seen as superfluous and discardable, as trained

writers, editors, and critical thinkers, head counts for content roles should be retained, if not be increased.

Concretely, technical writers and documentation specialists have optimal skill sets for reviewing AI-generated topic lists to use for inferencing in applications, similar to more traditional technical writing tasks such as compiling glossaries, indices and knowledge bases. Content experts working in tandem with other developers via LLMs will ultimately improve and speed up development cycles. Technical writers can engineer generative AI prompts, help design chatbots for documentation sites, and review LLM extraction and summarization outputs for that can be used in other application development. Such tasks can be integrated with other development workflows more easily using docs-as-code in technical documentation.

Working with LLMs in this way can allow content writers and documentation specialists to be more deeply embedded with other developers by interacting more directly with software applications they are building. Tech writers and content developers can also be involved in training LLMs on customized data by providing humanities-trained insight into the authenticity, relevance and inclusiveness of data.

These are just some examples showing how vital it is for IT companies to retain experts who know how to write well. Since LLMs are more reliant on human language and increasingly less reliant on programming code to interact with developers, innovative tech companies must not only retain but promote language arts expertise if they are to be competitive and harness generative AI's potential.

As is often the case when we write about tech, *ethics* are mentioned last, and here as well. However, my intention in bringing up the rear with the most important question of all is for the purpose of emphasis. With AI disruption, content expertise is needed now more than ever to confront the societal and political implications facing generative AI: how are LLMs trained and in whose interest? Have the system roles behind LLMs and their generative AI applications been fed content that serves investors? Consumers? Authoritarian states? Is it to be deployed in war or for peace? Content experts – tech writers, documentation managers, information architects, content writers, UX designers – all have had exposure to the humanities with a strong foundation in the arts, philosophy, political and social sciences. This kind of expertise is required more than ever by tech companies to define and implement socially sustainable best practices in the use of LLMs and generative AI.