

How I Organize a Massive Library of AI Prompts

You'll write a good prompt for AI and want to reuse it.

Bryan Collins • 3 min read • 2025-07-07

<https://bryanjcollins.medium.com/how-i-organize-a-massive-library-of-ai-prompts-29706849a336>



Photo by [Ryunosuke Kikuno](#) on [Unsplash](#)

What's the best way to manage all your prompts?

Sure, you can reference them inside ChatGPT. But what happens when you want to switch from GPT to Claude or Gemini?

I'm going to share a couple of workflows I use for managing my AI prompts.

Sometimes I'll write a good prompt for GPT and get great output. But what happens when you have dozens of different prompts to manage?

Enter the concept of an AI prompt library.

Here's a prompt I created a few weeks ago that I've been using regularly. I told ChatGPT it's a copywriter who specializes in viral LinkedIn content. I explained I need help with hooks for my LinkedIn posts. This particular prompt consistently delivers good outputs.

Rather than rewriting this prompt every time, I could copy and paste it from GPT into another

LLM. But there's a better way.

To follow this approach, categorize and organize your prompts in a dedicated prompt library. And I use a prompt to do it.

I told ChatGPT my framework for categorizing prompts. I need a prompt name, version number, platform, use case, and the text plus notes. ChatGPT categorized the prompt for me.

Then I head over to a plain text app on my Mac. The specific app doesn't matter. You can do this in any writing application.

I use the hashtag AI prompt so I can quickly find it later. I give the plain text file a descriptive heading so I can locate it on my computer. Because I've used hashtags and context, I can quickly reference this prompt and see what it's good for.

This workflow, using plain text or markdown file,s works well if you're using prompts yourself. But what if you want version control, sharing with team members, or working online?

There's an app popular with developers called GitHub. Don't let that deter you. GitHub is easy to use and you can host all your prompts with version control. You don't even have to pay.

Once you've set up your GitHub account, create something called a gist. Click your icon in the top right, then head to your gists. These are snippets of code, in this case prompts. You store them inside GitHub and share with team members while managing version control and backing up your prompts.

Creating them is easy. Select create new gist, paste in your prompt, give it a description like "prompt for LinkedIn hooks," and add a descriptive file name.

Once you're happy, click create secret gist. Only you can access it, but you can share the link with others. You can leave comments like "works well in GPT but not Gemini."

If you want changes, select edit and modify the gist. This shows how version control works. GitHub shows what was removed, helping you build a library of prompts as gists.

Make sure you give each one a descriptive name so you can find and reference them later. This way you can use GitHub to manage prompts and test them across different LLMs.

I like this because I can select raw to see the raw version. I can use plain text, rich text, or markdown formatting. And I can revert changes quickly.

To manage your prompts effectively, begin by creating a plain text file on your computer with a descriptive heading.

Take it further by categorizing what the prompt is for and which LLMs it works well in.

Take it even further with GitHub gists to create snippets and handle version control. All these workflows are free.

Watch

If you need help writing prompts for ChatGPT, Gemini, and Claude, check out [Prompt Writing Studio](#).

Struggling to use AI without losing your voice? Get my pack of 101 Proven AI writing prompts. 🤖
Download it free [here](#).

Other mentions by Author

- bryanjcollins.medium.com | Written by Bryan Collins