From Keating to Coding (2nd Edition)
What Dead Poets Society Teaches Us About Prompt Engineering

In the 1989 film Dead Poets Society, Robin Williams delivers one of cinema's most enduring performances as John Keating, an English teacher who dares to treat language as more than mere academic formality. To Keating, words are weapons against conformity, vessels for purpose, and catalysts for change. The classroom becomes his battlefield, and language his chosen armament.

He stands on desks not for theatrics, but to invite perspective. He tears pages from textbooks not for rebellion's sake, but to liberate meaning from rigid tradition. He encourages his students to think, speak, and write with conviction. "No matter what anybody tells you," he says, "words and ideas can change the world."

That line—like much of the film—feels especially prophetic today because now, we are speaking not just to each other, but to machines. It turns out that our ability to shape those machines' outputs is still fundamentally dependent on the timeless lessons Keating tried to teach: word choice matters, syntax is power, and language—well-aimed—is revolutionary.

The Prompt is the New Command Line
We now live in an era of Generative Artificial Intelligence, where large language models (LLMs) like OpenAI's ChatGPT, Anthropic's Claude, and Google's Gemini can interpret human language and return text, code, images, and even decisions with uncanny coherence.

What used to require extensive programming now requires a sentence.

Want to generate a marketing plan?
Draft a legal memo?
Write Python code for a microservice?
Summarize research from four disparate sources?

Ask the right question, and an LLM can return exactly that—often in seconds. But therein lies the operative phrase: ask the right question. The model is only as useful as the prompt it's given. The difference between an LLM output that feels magical and one that feels mediocre often comes down to a handful of well-chosen words.

This isn't new. We've always known that language can move markets, win elections, start wars, and end them. But now, language moves models. And that subtle shift has opened up a new discipline: prompt engineering.
Prompt Engineering: The New Literacy
Prompt engineering is the ability to communicate with an inference model using precise, strategically structured natural language in order to achieve a desired output. It's not programming, but it is

programmatic. It doesn't require technical syntax—but it does demand logical clarity, linguistic precision, and domain awareness.

What we're witnessing is the dawn of a new kind of literacy—not just digital fluency, but semantic control. It's not enough to type in "generate a blog post" and hope for brilliance. A vague prompt yields vague results. A lazy query returns generic answers. And a poorly framed question can lead a model astray—even if its underlying intelligence is state-of-the-art.

To put it plainly: language is the new interface, and prompts are the new programming.

Prompt engineering is about more than getting a chatbot to cooperate. It's about shaping the behavior of sophisticated neural networks trained on billions of parameters and petabytes of data. It's about having a conversation with all of human knowledge—and getting back something you can actually use.

Keating, Concision, and Computational Semantics

Let's revisit that iconic Keating quote:

"Avoid using the word 'very' because it's lazy. A man is not very tired, he is exhausted. Don't use 'very sad,' use morose... Language was invented for one reason, boys—to woo women—and in that endeavor, laziness will not do."

Of course, Keating's quip about the evolutionary purpose of language is rhetorically playful. But behind the joke is a timeless insight: lazy language is ineffective language. Whether we're persuading a lover, a boardroom, or a transformer-based deep learning model, our success depends on the precision of our vocabulary and the intentionality of our expression.

A prompt like, "write a nice email" may result in bland pleasantries. But a prompt like, "write a warm yet assertive email to a client explaining a project delay while reinforcing trust in our delivery timeline" will yield a result that actually serves your objective. One is vague and emotional. The other is strategic and precise.

And just as Keating encouraged his students to embrace richer vocabulary, we must now consider the emotional and contextual weight of our words when prompting machines. "Welcoming" is not "excited." "Empathetic" is not "submissive." "Direct" is not "abrasive." The nuance matters—because the machine does not guess what you meant. It reflects what you asked.

Prompt Structures: A Brief Taxonomy

Beyond vocabulary and tone, prompt structure also influences model behavior. There are a few key types of prompt strategies that every

modern professional should understand:

1. Zero-Shot Prompts
These give the model no examples—just a direct instruction.
Example:

“Summarize this article in three bullet points.”

Useful for straightforward requests, but quality may vary depending on ambiguity.

2. Few-Shot Prompts
These include examples of desired output to guide the model.
Example:

“Here are three summaries of past articles. Now do the same for this one.”

Few-shot prompting improves reliability by establishing pattern recognition. It’s particularly helpful when formatting or tone must remain consistent.

3. Chain-of-Thought Prompts
These walk the model through a step-by-step reasoning process.
Example:

“Let’s first outline the problem, then define the stakeholders, then list possible solutions.”

This method dramatically improves the performance of LLMs on tasks requiring logical deduction or multistep synthesis.

Each structure is a different way of wielding language—not just to retrieve information, but to shape cognition. It’s the closest thing we have to a user manual for thinking machines.

The Lexicon of the Future is Human
There is a persistent misconception that as AI grows more advanced, it will eliminate the need for humans to “speak machine.” But the truth is the opposite. The more capable the machine becomes, the more vital it is that humans speak with intention.

The rise of LLMs does not diminish the value of language—it amplifies it. It makes linguistic skill a superpower. In an age where machines can write code, analyze policy, generate visuals, or simulate personalities, the individual who can articulate ideas clearly becomes the architect of possibility.

This reality isn’t just for developers. Whether you’re a CEO, a policy analyst, a high school teacher, or a transitioning veteran, prompt

engineering is now part of your toolkit. The best communicators will wield AI as a force multiplier—not because they understand the underlying math, but because they understand the weight of a well-formed sentence.

Conversations with Machines

Here's the irony: in learning to talk to machines, we may become better at talking to each other. Prompts require clarity. They force reflection. They demand empathy—not in the emotional sense, but in the design sense. You must anticipate ambiguity, reduce assumptions, and express yourself clearly.

In this way, prompting is not about telling AI what to do. It's about engaging in a kind of dialogue—a linguistic negotiation. And much like any relationship, the better you understand the other party's logic, the more effective your conversation becomes.

Keating wanted his students to "seize the day." But in a world of language models, the updated imperative might be: seize the syntax. Own your voice. Command your tools. Speak with precision—and let the machine meet your standard.

Final Thought: The Renaissance is Semantic

We are entering a renaissance not of art or industry, but of semantics. The next great technologists may not be software engineers but poets with precision. The ability to translate thought into language and language into action—across human and machine domains—is fast becoming the most valuable skill on the planet.

So, let us return to the lesson John Keating left us with when referencing "Oh me! Oh Life!" by Walt Whitman:

"That the powerful play goes on, and you may contribute a verse."

Today, that verse may take the form of a prompt, rendered in 300 tokens and parsed by a multi-billion-parameter model. But its origin remains the same: human imagination, channeled through language, toward something greater.

"And in that endeavor, laziness will not do."

Afterward

I first published this essay on LinkedIn earlier this year. Since then, my approach prompt engineering has evolved to some degree yet at its core, the essence of prompting has remained the same.

Among the most valuable changes in my approach: I now ask models to generate phased plans of action for extended tasks. This simple shift clarifies ambiguity, unveils dependencies, and allows me to trust the AI to carry out its work unsupervised.

While model preference will inevitably differ by workflow, I've found Anthropic's Claude Code, running natively in my terminal, delivers the most reliable performance for my needs.

Yet at the heart of every output, amid shifting platforms and processes, the essential lesson persists: quality begins with the prompt.

"Master the basics." It's a credo in Special Operations, and I am continually reminded that it rings true far beyond the original context—here, too, in the dialogue between human and machine.