WHITE PAPER FUTURE FICTION ACADEMY JANUARY 23, 2026

The Em Dash Problem

A Case Study in Tokens

The em dash became the great tell of AI prose. The reason isn't stylistic — it's structural. A short, shareable case study in how tokenization makes a single punctuation mark load-bearing, and what that reveals about prompt engineering.

ELIZABETH ANN WEST

PROMPT-ENGINEERING TOKENIZATION EM-DASH AI-AUTHORSHIP

The mark that gives the machine away

The em dash became the most notorious tell of AI-generated prose — the punctuation mark that makes a reader squint and think, *a machine wrote this.* Authors learned to hunt it down and delete it on sight; whole editing passes have been devoted to scrubbing it out. The advice hardened into reflex: avoid the em dash.

But almost nobody asks the more interesting question. *Why* does the model love the em dash in the first place? The answer is not stylistic. It is structural, and it lives one layer below the writing, in how a model chops your language into pieces. Understanding it does two things at once: it explains a mystery authors have lived with for years, and it demonstrates — in miniature — the entire argument that prompt engineering is a real, mechanical discipline rather than folklore.

Punctuation is a token

A language model does not read words; it reads **tokens** — the small units it breaks text into before it does anything else. A token can be a whole word, a fragment of a word, a single letter, or a punctuation mark. That last category is the key. Run a simple phrase through a tokenizer —

> *Elizabeth is Goddess Divine.*

— and it resolves into **five tokens**[1]: `Elizabeth`, `is`, `Goddess`, `Divine`, and `.`. The period is not decoration appended to the sentence. It is a token in its own right, weighed and selected exactly like any word.

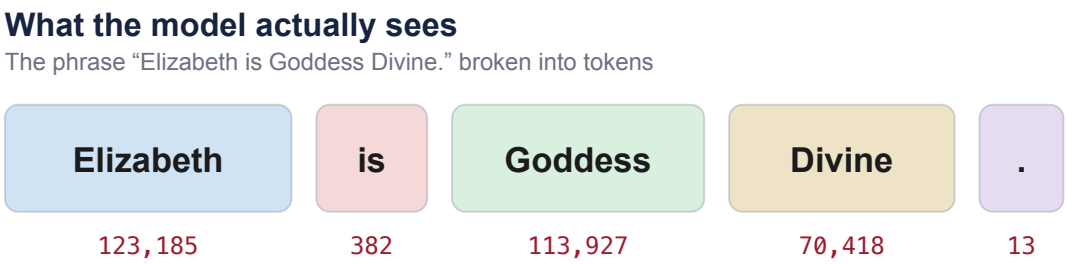

Figure: the OpenAI tokenizer splits the phrase into five tokens — the period included — each with its own vocabulary ID.

This is the whole secret of the em dash. **The em dash is a token too.** It is not a flourish the model adds after the fact; it is a unit the model can choose at any point, with its own learned probability of appearing, just like *fox* after "the quick brown." When a model reaches for the em dash, it is not being pretentious. It is doing precisely what it was built to do — selecting a high-probability token — and the em dash happens to be a token that appeared abundantly in its training text. That is why the em dash has been such a scourge[2]: it was never a style you needed to talk the model out of. It was a unit baked into the way the model sees the world.

A striking way to feel the scale of this: a single 1,000-word chapter required the model to make *at least* a thousand token decisions — and realistically closer to fifteen hundred[6], once you count words that split into multiple tokens plus all the punctuation. Fifteen hundred tiny choices, made in about two seconds, every one of them a token pulled from a probability distribution. The em dash is simply one choice the model keeps making.

The numbers behind the dash

Tokens are catalogued. Every token has an **ID** — a number marking which entry it is in a given model's vocabulary. And here the em dash tells on itself.

The em dash carries a different ID depending on the model class: it is token **2322**, **2345**, or **960**[3], depending on which generation of model you are talking to. The same mark, three different internal identities, because each model family was built with its own vocabulary (a point developed at length in the flagship paper). Some punctuation, by contrast, never moved at all. The period is token **13** across every generation tested. Why does the period hold still while words like *Goddess* leap thousands of positions between model versions? Because, as the source session puts it, *we didn't invent new punctuation.*[4] The set of marks is stable; the vocabulary of words and word-fragments churns.

The deeper trivia is almost comic. In the OpenAI vocabulary, the **exclamation point is token zero.**[5] The quotation mark is token one. The hashtag is token two. The most-used mark in the language — the humble period, the full stop — is *not* token zero, which is genuinely irritating if you think about it. As the session's verdict went: *the period should be zero. The full stop should be zero. It is the most used.* Instead the exclamation point got the honor, seemingly for no better reason than that it sits first on the keyboard. The lesson lands with a grin: *these people did not take extra linguistics classes. They took computer classes.* The vocabulary was numbered by engineers optimizing for the machine, not by anyone reasoning about how language is actually used — which is exactly the gap that makes prompt engineering, properly understood, a humanities discipline as much as a technical one.

Why this matters for your prompts

Two practical consequences follow, and they connect this small mark to the larger craft.

Deleting em dashes is treating a symptom. If the em dash is a high-probability token the model is structurally inclined to select, then scrubbing it in editing is endless whack-a-mole. The mark is a *signal* of how the model fills space, not a one-off mistake. Understanding why it appears lets you address the cause — the prose register that invites it — rather than chasing every instance after the fact.

"Just avoid it" is the old reflex, and the old reflex is outdated. Authors learned to never *name* the em dash in a prompt, on the theory that mentioning it only makes the model fixate. That belonged to an era when models could not act reliably on what-not-to-do instructions. As the companion paper *Reinforcement Logic* documents, that era is over. You can now constrain the em dash deliberately — pairing what is permitted against what is not — rather than tiptoeing around it and hoping. The em dash, in other words, is the perfect small test case for the entire shift this series describes: from superstition to mechanism, from avoidance to control.

The point in one sentence

The em dash is not a quirk of taste; it is a token — a numbered unit the model selects by probability — and once you can see it that way, you stop fighting the symptom and start steering the machine. That is the whole of prompt engineering, compressed into a single punctuation mark.

Token IDs cited here reflect the OpenAI tokenizer as demonstrated in the Future Fiction Academy session "Prompt Engineering IS Back!" and are model-family dependent. For the full treatment of tokens, embeddings, and model weights, see the flagship paper, Why a Single Word Changes Everything. *For constraining unwanted tokens deliberately, see* Reinforcement Logic.

Sources & Timestamps

All quotations and figures derive from the live Future Fiction Academy session *Prompt Engineering IS Back!* and its companion notes. Video timestamps below link to the exact moment; the companion page hosts the token tables and full worked examples.

1. Punctuation is a token: “Elizabeth is Goddess Divine.” = five tokens. — 00:36:56.438
2. “This is why the em dash has been a scourge.” — 00:37:07.760
3. Em dash token IDs by model class (2322 / 2345 / 960). See Notion companion page. — Notion companion page

4. Why the period stays token 13 across generations: “we didn’t invent new punctuation.” — Notion companion page
5. “The exclamation point is token zero… the period should be zero.” — 00:58:22.400
6. A 1,000-word chapter = ~1,500 token decisions in two seconds. — 00:59:30.000

Primary video: Prompt Engineering IS Back! — full session
Companion notes: Notion page