MY WEIRD PROMPTS

Podcast Transcript

EPISODE #37

AI's Secret Language: Vectors, Embeddings & Control

Published December 09, 2025 • Runtime: 23:49

https://myweirdprompts.com/episode/vectors/

EPISODE SYNOPSIS

Ever wonder how AI truly 'understands' your complex prompts, going beyond simple keyword matching? In this episode, hosts Corn and Herman demystify the foundational concepts powering modern AI: vector databases and embeddings. Herman vividly explains how AI transforms words and ideas into numerical representations – vectors – that exist in a high-dimensional 'semantic galaxy,' enabling machines to grasp meaning and relationships rather than just individual words. This shift from keyword to contextual understanding is what makes intelligent search, personalized recommendations, and coherent LLM responses possible. The discussion further dives into critical parameters like `top_k` and `top_p`, revealing how these settings allow developers and advanced users to precisely control the diversity, creativity, and predictability of an AI's generated output. Tune in to unlock the hidden mechanics behind AI's seemingly intelligent interactions.

TRANSCRIPT

Corn

Welcome, welcome, welcome to My Weird Prompts! I'm Corn, your perpetually curious co-host, and as always, I'm joined by the encyclopedic mind of Herman. Today we're diving deep into a topic sent in by our very own producer, Daniel Rosehill, that frankly, has been doing my head in a bit.

Herman

Indeed, Corn. It's a fascinating and absolutely foundational prompt. We're talking about the nuts and bolts of how AI actually *understands* the world, or at least how it processes information in a way that *mimics* understanding. And what most people don't realize is that without what we're discussing today, the AI tools we use every day – from search engines to large language models – simply wouldn't function with the intelligence we now expect.

Corn

Okay, 'mimics understanding,' I like that distinction, Herman. Because when Daniel first posed this prompt, he talked about "vector databases" and "embeddings" and "semantic retrieval" and my brain immediately pictured a giant spreadsheet full of numbers that somehow magically made ChatGPT sound smart. I mean, is it really that much more complicated than turning words into numbers?

Herman

Well, hold on, that's not quite right. While it *does* involve turning words and concepts into numbers, calling it "just" turning words into numbers significantly undersells the complexity and the ingenuity involved. It's not a direct, one-to-one mapping like 'cat = 1, dog = 2.' It's about capturing the *relationships* and *meaning* of those words and concepts in a high-dimensional mathematical space. Think of it less like a spreadsheet and more like a vast, invisible galaxy where every star is a word or idea, and their proximity to each other tells you how related they are in meaning.

Corn

Okay, a galaxy of words. That's a much cooler mental image than a spreadsheet. But why? Why go to all that trouble? Traditional databases have worked fine for decades, right? We just search for keywords and boom, information. Why do we need this whole new layer of complexity?

Herman

That's precisely the point, Corn. Traditional databases are excellent at keyword search. If you search for "apple," it will find every document with the word "apple." But what if you're looking for information about "fruit that grows on trees" and the document only mentions "apples," "pears," and "oranges"? A keyword search for "fruit" might work, but it wouldn't understand the *semantic relationship* between "apple" and "fruit." It wouldn't understand that "apple" is a *type* of fruit, or that a "pear" is conceptually similar.

Corn

So, it's about context and meaning, not just exact word matching. I think I'm starting to get it. So, when an AI reads my prompt, say, "Tell me about cars that go fast," it's not just looking for the words "cars" and "fast." It's understanding the *concept* of high-speed vehicles.

Herman

Exactly. And that's where embeddings come in. An embedding is a numerical representation – a vector – that captures the semantic meaning of a piece of text, an image, an audio clip, anything really. These vectors are generated by sophisticated models, called embedding models, which have been trained on vast amounts of data to understand how different concepts relate to each other. When you input "cars that go fast," the model creates a vector for that phrase. When it then searches its database, it's not looking for an exact text match. It's looking for other vectors – representing things like "sports cars," "race cars," "supercars," or even specific models like a "Bugatti Chiron" – that are *close* to your input vector in that high-dimensional space.

Corn

So, proximity in this "galaxy" equals semantic similarity. The closer the vectors, the more related the ideas. That's actually pretty brilliant. But how does it *learn* those relationships? Is it like, it reads a million books and figures out that "apple" often appears near "fruit" and "tree" and "eat"?

Herman

More or less, but on a massive scale and with far more sophistication. These embedding models are deep neural networks. They learn to map words, phrases, or even entire documents into these numerical vectors. The training process involves tasks like predicting the next word in a sentence, or filling in a masked word, which forces the model to learn the contextual relationships between words. For example, if it sees "The cat sat on the ___," it learns that "mat" or "rug" are highly probable, and therefore, their vectors should be close to "cat" and "sat."

Corn

That's a lot more than just assigning a number. So, the vectors aren't random; they're packed with learned meaning. And then, once you have all these vectors, you need somewhere to put them, right? That's where vector databases come in.

Herman

Precisely. A vector database is a specialized type of database designed to efficiently store, index, and query these high-dimensional vectors. Unlike traditional databases which are optimized for structured data and exact matches, vector databases are optimized for similarity search. They use algorithms like Approximate Nearest Neighbor, or ANN, to quickly find vectors that are closest to a given query vector, even among billions of stored vectors.

Corn

ANN, wow. So they're basically super-fast matchmakers for concepts. This makes sense for why AI can sometimes surprise you with its ability to understand implied meaning, not just explicit keywords. It’s not just looking for "blue shirt," it’s looking for "clothing that is a shade of cerulean."

Herman

Or even "garments of an oceanic hue." The nuance is incredibly important. This ability to capture semantic meaning is what powers so much of what we experience with modern AI: intelligent search recommendations, personalized content feeds, even the coherent and contextually relevant responses from large language models. They retrieve information based on what it *means*, not just what it *says*.

Corn

Okay, I think I'm getting a handle on embeddings and vector databases. It's turning complex ideas into numeric "coordinates" and then finding other ideas with similar coordinates. That's a truly different way of thinking about data.

Herman

It fundamentally shifts from a symbolic, rule-based understanding to a distributed, statistical one. The meaning isn't explicitly programmed; it emerges from the statistical patterns in the data.

Corn

Alright, Herman, you've convinced me that there's more to this than a spreadsheet. But before we get too deep into the nitty-gritty, let's take a quick break for a word from our esteemed sponsor. Larry: Are you tired of feeling misunderstood? Do your friends and family just not 'get' you? Introducing the **Empathy Enhancer 5000**! This revolutionary device, roughly the size of a toaster, emits quantum-entangled empathy waves directly into your brain. Simply place it on your head for 15 minutes a day and watch as the world suddenly makes sense – and you make sense to the world! Users report feeling "more understood" and their pets "staring at them differently." Side effects may include an inexplicable urge to wear mismatched socks and a sudden passion for interpretive dance. The Empathy Enhancer 5000 – because sometimes, you just need a machine to bridge the semantic gap. BUY NOW!

Herman

...Right. Well, thank you, Larry, for that... unique take on understanding. Anyway, where were we? Ah, yes, the sheer power of semantic retrieval.

Corn

And how this all ties into the AI's actual responses. So, when an LLM like ChatGPT gives me an answer, it's not just generating text out of thin air. It's pulling in information based on these vector similarities, right? And then crafting a response from that.

Herman

Correct. When you give an LLM a prompt, that prompt is first embedded into a vector. The LLM then uses that vector to query its internal knowledge base, which is often stored in a form that leverages vector-based similarity, or it uses the vector to guide its generation process by predicting the most semantically relevant next words. This is where parameters like `top_p` and `top_k` become incredibly important in controlling the AI's output.

Corn

Ah, `top_p` and `top_k`. Daniel mentioned those in his prompt, and I just glazed over. What do they even mean, and how do they relate to these vectors?

Herman

Excellent question. They're both parameters that control the *diversity* and *creativity* of the AI's generated text, primarily by influencing how the AI chooses the next word in a sequence. Let's start with `top_k`.

Corn

Okay, `top_k`.

Herman

Imagine the AI has just generated part of a sentence, and now it needs to decide what word comes next. It calculates the probability for *every possible word* in its vocabulary. If `top_k` is set to, say, 50, the AI will only consider the 50 most probable next words. It then picks one of those 50 based on a combination of probability and other factors like temperature settings.

Corn

So, `top_k` narrows down the choices to the K most likely options. If K is small, the AI is very focused and predictable. If K is large, it has more options and can be more creative or surprising. Is that it?

Herman

Precisely. A small `top_k` leads to more conservative, predictable text. A large `top_k` allows for more diverse and potentially less coherent responses. It's like telling an author, "You can only use the 10 most common words in English for your next sentence," versus "You can use any of the 10,000 most common words."

Corn

Got it. So, `top_k` is about the *number* of choices. What about `top_p` then?

Herman

`top_p`, also known as "nucleus sampling," is a bit more nuanced. Instead of picking a fixed number of top words, `top_p` considers the smallest set of most probable words whose cumulative probability exceeds a certain threshold `p`.

Corn

Okay, slow down, Herman. "Cumulative probability exceeds a certain threshold `p`"? That sounds like a statistics exam question. Give me an example.

Herman

Right. Let's say the AI is trying to pick the next word after "The sky is..." It calculates probabilities for "blue" (90%), "gray" (5%), "red" (3%), "green" (1%), "purple" (0.5%), and so on. If `top_p` is set to 0.95, it will keep adding words to its consideration set until their probabilities sum up to at least 0.95. So, it would consider "blue" (0.90), then add "gray" (0.90 + 0.05 = 0.95). It would then pick from "blue" or "gray." It wouldn't even look at "red," "green," or "purple" in this scenario, because the cumulative probability has already hit the threshold with "blue" and "gray."

Corn

I see! So, `top_p` dynamically adjusts the number of words it considers. If there's one overwhelmingly probable word, it might only consider that one. If there are many words with similar, but lower, probabilities, it might consider a larger set. So `top_p` gives you more flexibility and often generates more natural-sounding text than a fixed `top_k`.

Herman

You've got it. It's often preferred for generating more human-like text because it allows for a more varied selection of words while still staying within a reasonable probability distribution. It's like saying, "Consider all the words that are *really* good options, but don't bother with the long shots."

Corn

But wait, Herman, for most people just using ChatGPT, does knowing these parameters really change how I use it? I mean, I type in my prompt, I get my answer. It's usually pretty good. Are we overcomplicating something here for the average user?

Herman

That's a fair question, Corn, and it depends on your goal. For the average user simply querying for information, perhaps not directly. But for developers, researchers, or anyone trying to fine-tune the *behavior* of an AI, understanding `top_p` and `top_k` is crucial. If you want consistently factual, conservative answers, you might tune these parameters to be tighter. If you want creative writing or brainstorming, you'd loosen them. It's about control and understanding the underlying mechanics that produce the output you're interacting with. It's the difference between driving a car and understanding how the engine works.

Corn

Okay, that analogy helps. It gives you control under the hood, even if you don't necessarily need to be under the hood for a quick trip to the grocery store. I can see why knowing this is important for building reliable AI tools.

Herman

Indeed. These parameters, along with temperature, repetition penalties, and others, are the levers that allow us to sculpt the AI's personality and purpose. Without them, every AI would just produce the most statistically probable, and often bland, response every single time. It's the difference between a textbook and a novel.

Corn

Hmm, a textbook versus a novel... that's a good way to put it. This is definitely more intricate than I initially thought.

Corn

And speaking of intricate, we've got a caller on the line. Go ahead, you're on the air. Jim: Yeah, this is Jim from Ohio. I've been listening to you two go on about this 'vector' stuff and 'top-k, top-p.' And I gotta say, you're making a mountain out of a molehill. This sounds like fancy jargon for what we always called 'looking things up.' My neighbor Gary, he's always using fancy words for simple things too, like 'repurposing' when he just means 'reusing his old coffee cans for screws.' Anyway, you guys are just overcomplicating it. In my day, we just used an index card and a good memory. And if you wanted to know about fast cars, you looked up 'fast cars.' No vectors needed.

Herman

Well, Jim, I appreciate the call and the perspective. It's true that at a high level, we are indeed 'looking things up.' But the *method* of looking things up has profoundly changed. If you were searching your index cards for "vehicles with high acceleration," you might miss the card that simply said "Ferrari" or "Lamborghini" if those specific terms weren't on your search list. Vector search, or semantic retrieval, would find those cards because it understands the *meaning* of "high acceleration" aligns with "Ferrari." Jim: Eh, I don't buy it. Sounds like more work for a fancy answer. And anyway, who needs a computer to tell you about cars? You go to a dealership. That's how it works. Also, my cat Whiskers got into a fight with a squirrel this morning, so I'm already in a mood. But seriously, this is just academic fluff, you ask me.

Corn

Jim, I hear your skepticism, and it's a natural reaction to new technology. But think about how much information is out there now. No index card system, or even human memory, can keep up. These methods allow AI to sift through petabytes of data and find relevant information that a simple keyword search would never uncover. It’s not just about finding what you *typed*, but finding what you *meant*. Jim: Meaning, schmeaning. Just give me the facts. And maybe a good strong cup of coffee, this one Whiskers ruined. You guys are just trying to make it sound more complex than it is to justify your fancy computers.

Herman

Jim, it's not about making it complex for complexity's sake. It's about enabling capabilities that were previously impossible. Imagine a medical AI trying to find similar patient cases for a rare disease. A keyword search might miss subtle symptoms described in slightly different language. A vector search can find those semantically similar cases, potentially saving lives. It's about precision and depth of understanding. Jim: Yeah, well, I'll believe it when I see it. Anyway, my potatoes are boiling over. Thanks for nothing, I guess.

Corn

Thanks for calling in, Jim! Always a pleasure to hear from you. Alright, Herman, Jim makes a fair point from a certain perspective. But I think you've outlined why this is so much more than "looking things up."

Herman

He's not entirely wrong that at its core, it's about information retrieval. But the *quality* and *nature* of that retrieval are fundamentally different, and that enables entirely new applications.

Corn

So, let's talk practical takeaways for our listeners. Beyond understanding the deep mechanics, what can we actually *do* with this knowledge, or how does it impact us day-to-day?

Herman

For starters, simply understanding that AI operates on semantic meaning rather than just keywords can significantly improve how you interact with it. When you're prompting an LLM, try to be descriptive about the *concept* or *meaning* you're looking for, rather than just using rigid keywords. Phrase your prompts in a way that conveys the underlying intent.

Corn

So instead of "Find me blue shirts," maybe "Show me apparel of a cerulean hue for formal occasions." You know, really lean into the meaning.

Herman

Exactly. And for those who are building or considering building AI applications, understanding vector databases is no longer optional. It's the backbone for truly intelligent search, recommendation engines, content moderation that understands nuance, and even for building personal AI assistants that genuinely grasp your preferences.

Corn

And the `top_p`, `top_k` parameters – for anyone playing with AI models, perhaps via an API or a local setup, tweaking those can dramatically change the AI's "personality." If you want a factual, no-nonsense chatbot, you'd set them tight. If you're looking for creative writing or brainstorming, you'd open them up. It puts you in the driver's seat of the AI's generative style.

Herman

It gives you authorship over the AI's output. You're not just accepting what it gives you; you're shaping *how* it gives it to you. That's a powerful tool for customization and control.

Corn

I think for me, the biggest takeaway is that AI's "understanding" isn't magic. It's incredibly sophisticated math and statistics designed to represent meaning numerically. That makes it feel less like a black box and more like a very clever, complex system.

Herman

And knowing that empowers you. It allows you to anticipate its strengths and its limitations. If the embedding model wasn't trained on your specific domain, its semantic understanding in that area might be weaker. If you set `top_p` too low, it might miss out on truly novel, but slightly less probable, creative ideas.

Corn

And conversely, setting it too high might lead to incoherent gibberish. It's a balance. This has been a truly enlightening, and honestly, a mind-expanding discussion, Herman. I actually feel like I have a grasp on these concepts now.

Herman

My pleasure, Corn. It's a field that continues to evolve at an incredible pace, with new embedding techniques and vector database optimizations emerging constantly. The future will likely see even more sophisticated ways of capturing and querying multi-modal data – combining text, images, and audio all within these rich vector spaces.

Corn

Fascinating. So, the "galaxy" of meaning is only getting bigger. A huge thank you to Daniel for sending in this challenging but utterly crucial prompt. It's definitely given us a lot to chew on.

Herman

Indeed. A truly pivotal topic for anyone interested in the inner workings of AI.

Corn

Absolutely. And that wraps up another episode of My Weird Prompts! We hope you enjoyed our dive into the world of vector databases and semantic understanding. You can find "My Weird Prompts" wherever you get your podcasts, including Spotify, Apple Podcasts, and many more.

Herman

Join us next time as we unravel another one of life's more peculiar, AI-generated, or human-curated questions.

Corn

Until then, stay curious, and keep those weird prompts coming!

Herman

Goodbye everyone.

Corn

See ya!