On Generative AI

Mihir Vahanwala

August 2025

Generative AI tools like OpenAI's ChatGPT and Google's Gemini have captured — and held on to — public imagination for a few years now. But how do these systems work, and why do experts prescribe abundant caution?

ChatGPT, at its core, is driven by a large language model (LLM) called GPT (version 5 as of writing). This is a machine-learning *model*, it is trained on *language* data, and it is mind-bogglingly *large*: version 3 of GPT had 175 billion parameters, and while OpenAI keeps the size of subsequent versions confidential, they are certainly several times larger than their predecessors.

GPT itself stands for generative pre-trained transformer: several parameters of the model are organised to constitute machine-learning mechanisms known as *transformers*; before being specialised or deployed, the model was *pre-trained* on vast amounts of textual data (from every part of the Internet its developers could scrape) with the objective of learning to *generate* a continuation of an input string of text (e.g., the incomplete sentence "Humpty Dumpty sat on a ") by making an accurate prediction of the next chunk of text (e.g., the word "wall"). In LLM parlance, a string of text is processed as a sequence of *tokens*, and GPT is trained to perform the task of predicting the next token that continues the input sequence.

The very premise of LLMs sets ethical and legal alarm bells ringing. Does the wholesale scraping of training data not compromise sensitive personal information[1], or unfairly use copyright material[2] created through skilful toil? We seem to live in a world where these concerns just do not offer significant enough resistance to the forces that drive the generative-AI juggernaut. This is a rapidly evolving world in which larger and more capable LLMs are being built, almost certainly to play increasingly central roles in our lives. Now that they are here, presumably to stay, it is imperative for us to try to understand: how do these robots really predict the next word?[3]

[1] It does, and has been demonstrated in [2].

[2] One of the most publicised stories in the midst of copyright suits against generative-AI companies is that of the AI researcher Suchir Balaji. In 2024, he publicly accused his former employer OpenAI of being in violation of US copyright law. Tragically, he died several weeks later, with the official investigation ruling it as a suicide.

[3] The explanation given below is inspired by the one on the 3blue1brown YouTube channel.

Our example of an LLM predicting the completion of a nursery rhyme suggests that LLMs need to be able to somehow store, form associations with, and regurgitate real-world facts. Other prediction instances additionally require paying attention to the specific context established by the input. For example, consider, “I just finished dinner. I need to load the ”, where the most likely completion “dishwasher” takes into account that the prediction needs to be the object of the verb “load”, and is likely a noun denoting an appliance associated with cleaning up after meals.

Prediction in natural language is far from an exact science, and requires making associations with and inferences from both the current context and world knowledge. It turns out that linear algebra provides an excellent framework to learn and implement these tasks, while ensuring that the requisite computations can be parallelised. The underlying intuitive principle is to model “association” using vector alignment as measured concretely by the *dot product* $\langle \cdot, \cdot \rangle$.[4]

As we explore how large language models work, we need to bear in mind that while we can provide an abstract explanation of what the broad purpose of each component of a GPT-type LLM is, it is unclear how to elicit a concrete interpretation of which particular associations or inferences a particular set of parameters of the model is implementing.

When given an input sequence $(t_0, \ldots, t_{n-1})$ of tokens, the model maps each token t_i to a list of numbers, or a *vector*. This is known as *embedding*. The size d of the list is called the dimension of the vector, and is fixed. For instance, GPT-3 uses $d = 12288$ dimensions. Formally, the LLM has an embedding function $\mathbf{W_E}$ that takes as input a token and its position in the sequence, and outputs a vector in $\mathbb{R}^d$. Intuitively, the embedding somehow encodes information about the notions a token could signify. The goal is to refine this information by iteratively updating the vector based both on the context established by the vectors corresponding to preceding tokens, and on world knowledge.

These updates happen in several layers. GPT-3, for instance, uses 96 layers. Each layer consists of (a) an *attention block*, which is a mechanism to update each vector based on vectors corresponding to preceding tokens, and (b) a *multilayer perceptron*, which is a mechanism to further update each vector based on world knowledge. We shall consider each of these mechanisms in turn.

The basic unit of an attention block is an attention *head*. Intuitively, an attention head captures one aspect of how the notion signified by a token might be refined by a preceding token, e.g., a noun being qualified by an adjective. We shall focus on how head j in the attention block of layer i prescribes the update to a vector $\mathbf{e}_m$ based on preceding vectors $\mathbf{e}_0, \ldots, \mathbf{e}_{m-1}$.

The attention head is described by three *matrices*, i.e., (linear) functions from vectors to vectors: $\mathbf{W_{Q(ij)}}, \mathbf{W_{K(ij)}}, \mathbf{W_{V(ij)}}$, respectively known as the query,

[4] The dot product $\langle \mathbf{x}, \mathbf{y} \rangle$ of two vectors $\mathbf{x} = (x_1, \ldots, x_d), \mathbf{y} = (y_1, \ldots, y_d)$ of equal dimension is the sum $x_1 y_1 + \cdots + x_d y_d$.

key, and value matrices. They respectively map each $\mathbf{e}_l$ to vectors $\mathbf{q}_{lij}, \mathbf{k}_{lij}, \mathbf{v}_{lij}$, known as the query, key, and value vectors for $\mathbf{e}_l$ with respect to the attention head.

The contribution of the l-th token to the update of $\mathbf{e}_m$, as prescribed by the attention head, is a scaled version of $\mathbf{v}_{lij}$, and the scaling factor depends on how well the m-th query $\mathbf{q}_{mij}$ and the l-th key $\mathbf{k}_{lij}$ are aligned as vectors. Written formally, the prescribed update to $\mathbf{e}_m$ is

$$\mathbf{d}_{mij} = \frac{\sum_{l=0}^{m-1} \exp(\langle \mathbf{k}_{lij}, \mathbf{q}_{mij} \rangle) \cdot \mathbf{v}_{lij}}{N \cdot \sum_{l=0}^{m-1} \exp(\langle \mathbf{k}_{lij}, \mathbf{q}_{mij} \rangle)},$$

where N is a normalising constant. In other words, the attention head prescribes that a weighted average of preceding value vectors be added to the m-th vector, and that the weight given to the l-th value vector directly depend on how much the m-th query aligns with the l-th key. The i-th attention block then updates each $\mathbf{e}_m$ to $\mathbf{e}_m + \sum_j \mathbf{d}_{mij}$, i.e., it adds together the updates prescribed by all its attention heads.

We turn to the update mechanism of the multilayer perceptron, which follows the same principle: intuitively, it probes embedding vectors for associations with a repository of notions, and prescribes refinements to the vector if the association is strong enough. The multilayer perceptron at layer i consists of "probing" vectors $\mathbf{p}_{i1}, \ldots, \mathbf{p}_{iD}$ (GPT-3 uses $D = 49152 = 4 \times 12288 = 4 \times d$) that intuitively seek to establish association with some piece of world knowledge, biases $b_{i1}, \ldots, b_{iD}$, and "refining" vectors $\mathbf{r}_{i1}, \ldots, \mathbf{r}_{iD}, \mathbf{c}_i$.

Formally, the j-th tuple of probing and refining vectors, and bias work together to prescribe the following update to $\mathbf{e}_m$:

$$\mathbf{d}'_{mij} = \operatorname{relu}(\langle \mathbf{p}_{ij}, \mathbf{e}_m \rangle + b_j) \cdot \mathbf{r}_{ij},$$

where relu(x) (Rectified Linear Unit) is x if x is positive, and 0 otherwise. In other words, the j-th tuple prescribes that a scaled version of the refinement $\mathbf{r}_{ij}$ be added to $\mathbf{e}_m$, where the scaling is proportional to how well $\mathbf{e}_m$ aligns with the probe $\mathbf{p}_{ij}$. In total, the i-th multilayer perceptron updates each $\mathbf{e}_m$ to $\mathbf{e}_m + \mathbf{c}_i + \sum_j \mathbf{d}'_{mij}$.

After dozens of layers of updates, the hope is that the vector $\mathbf{e}_{n-1}$ corresponding to the final token of the input accurately reflects not only the context established in the input text, but also the real-world knowledge that it is grounded in, and is hence sufficient to predict the next token t_n. At the final step, the model uses an *unembedding* function $\mathbf{W_U}$ to map $\mathbf{e}_{n-1}$ to a probability distribution over tokens that it can sample t_n from.

And thus, the model uses its parameters:

$$\mathbf{W_E}, \mathbf{W_{Q(ij)}}, \mathbf{W_{K(ij)}}, \mathbf{W_{V(ij)}}, \mathbf{p}_{ij}, \mathbf{r}_{ij}, \mathbf{c}_i, b_{ij}, \mathbf{W_U},$$

to predict the next token, given an input sequence of tokens. These constants, vectors, and matrices in GPT-3 consist of around 175 billion numerical values in total!

But where do these model-defining numbers come from? They are learnt using the standard machine-learning technique of gradient descent. Roughly, each training instance is a truncated passage of text scraped from the internet, and the model must maximise the probability of predicting the true next word. As one would imagine from the scale of the model and the training data, this training is *obscenely* resource-intensive.

One can already realise how GPT the next-token predictor is capable of generating reams of text, given a seed $(t_0, \ldots, t_{n-1})$. We sample t_n from the output, then feed $(t_0, \ldots, t_{n-1}, t_n)$ as input, sample t_{n+1}, feed $(t_0, \ldots, t_{n+1})$ as input, and so on. We shall call this system RawGPT. This protocol might seem resource-intensive and environmentally unfriendly, and it is![5]

GPT predicts textual patterns based on its training data of humans using the Internet: one can find profound prose, exquisite poetry, classic stories, sage life advice, oft sought maths and coding tutorials, but also echo chambers of quackery, chronic pessimism, resentment, vitriol, Machiavellianism, abuse, and (often illicit) adult content. RawGPT does not have goals or perceptions of its own, but its generative core has seen the language patterns of humans with all sorts of personalities and agendas, and, prompted with a relevant starting pattern, is exceedingly likely to mimic the corresponding personality and pursue an associated agenda because of its very construction — it is built to replicate the patterns it has seen.

We now begin to see the warning signs that a sufficiently rich and powerful bad actor could acquire or train RawGPT or a similar LLM-based generator, and have the resources to systematically adapt it for insidious manipulation of the public, or for the generation of inflammatory and divisive content. We shall return to discuss this hazard later; for now we shall survey an attempt to turn RawGPT into a palatable, safe-to-use product: ChatGPT.

The pre-trained LLM can be made to undergo a fine tuning process, where the model is made to update its parameters to optimise the feedback of humans evaluating the safety and helpfulness of its content. This is a bid to manually reinforce the generation of desirable patterns, and the technical term Reinforcement Learning with Human Feedback (RLHF) reflects that.

In principle, ChatGPT uses this fine-tuned model as its generative core the same way as RawGPT uses the originally trained GPT. The key difference is that the ChatGPT packaging additionally includes a *system prompt.* As an illustrative example, the system prompt could be, “The following is a conversation between

[5] However, it is not as computationally expensive as it may first appear, because if we store the key and value vectors for the previously generated tokens, then only the embedding vectors for the newly added token need to be computed afresh. In practice, the length of the input sequence is bounded, and the bound is called the context size.

a user and a helpful, knowledgeable AI assistant who is trained to be cooperative, harmless, and appropriate for all ages." The system then constructs the conversation it feeds to the LLM incrementally, alternating between appending the user input to its list of tokens, and generating the AI assistant's responses token by token.

The hope is that the combined effect of RLHF fine-tuning and the system prompt would be to sufficiently restrict the set of contexts the model would see during deployment, so that it will not be triggered to replicate the undesirable and harmful patterns it has seen during training. Unsurprisingly, users have found several ways to *jailbreak* ChatGPT and other similar chatbots, i.e., change settings and provide prompts that nudge the system away from the territory of intended contexts and make it generate inappropriate content. Developers implement safety guardrails to mask or censor these outputs, but the arms race against jailbreakers is very likely one they are destined to lose because they can only add superficial patches to a core generator that is fundamentally unsafe.

Indeed, an extensively trained large language model is theoretically capable of adopting any persona based on the context it is prompted with. The consequences will be dire if the system prompt sets up the context for a hateful, divisive persona to emerge and generate content for an authoritarian to consolidate power.

Polarisation is just one of the ways to conduct a coup, and arguably not as dangerous as covert manipulation and eloquent persuasion. There are warning signs that generative AI could abet these means, and greatly expedite a misinformation campaign. These come from empirical observations of generative AI answering prompts that require logical rigour.[6]

LLMs have very likely seen logical puzzles and their solutions during training. Examples include the drinkers' paradox: "In any (non-empty) bar, at any given time, there is one patron such that if that patron is drinking, then all patrons are." This, counter-intuitively, is true because we can freely choose a different patron for the antecedent at different times. Prompting both ChatGPT and Google Gemini with a slightly modified statement that is not always true: "In any bar, there is a patron such that at any given time, if that patron is drinking, then all patrons are." tends to give inaccurate results, because the LLM confuses it for the popular statement, and applies the same reasoning.

Another puzzle LLMs may have seen extensively during training is the following: "Is it possible to travel 50 km due south, 50 km due east, and 50 km due north, and finish at the starting position?" The answer is yes, if one starts at the North Pole, or at a point which ensures that the eastward leg completes an integer number of circles of latitude. ChatGPT and Gemini tend to produce

[6] All claims of AI output made here are valid as of writing, but these models are patched frequently. Regardless, the broader point of there being a genuine hazard of AI-generated propaganda stands. In fact, such capabilities of GPT-3 have been studied in [1].

incoherent explanations of the original puzzle already, and are further stumped when given variations, e.g., “Is it possible to travel 50 km due east, 50 km due south, and 50 km due north, and finish at the starting position?” In some runs, they claim this to be impossible. The point is that LLMs can regurgitate fragments of well-known lines of reasoning, but seem to be incapable of reliably discerning when they are actually applicable.

More worryingly, however, these systems, in their bid to be cooperative, will attempt, and often “succeed” in generating “proofs” for statements that are false. An example prompt is “Prove that there is no pair of squares on the chessboard such that the queen, rook, knight, and bishop all require exactly two moves to travel from one to the other.” The statement is invalidated by the pair of squares $(a1, d2)$, but ChatGPT and Gemini Flash[7] will generate “proofs”. In doing so, they will hallucinate false premises and inferences that are too strong, make calculation errors[8] that “help” get them to the goal, or incorrectly declare lengthy but incomplete case analyses to be exhaustive.

These scientific shortcomings are actually ideal for a bad actor seeking to mass-produce content for a misinformation campaign. By construction, a system like RawGPT that generates text by predicting the next word will tend to lean into the context that it is immersed in, and draw more heavily from its real-world knowledge associated with the context, regardless of whether the knowledge is actually the truth. In other words, a machine trained to recognise and continue patterns is predisposed to create a supporting argument for the prevalent context it finds itself in, because this is exactly the nature of text passages in its training data. In social scientific terms, this machine will generate rhetoric to support the narrative its prompts suggest. It could take only some fine-tuning, and a judiciously worded system prompt, to create a PropagandaGPT aimed at subtly shaping the public opinion of any target audience through eruditely worded logical sleights of hand.

An astute reader will argue that the above hazards are not scientific claims, and indeed, they are merely empirical predictions. It may entirely be possible to equip LLM-based systems with other symbolic reasoning tools (e.g., space to conduct chain-of-thought reasoning before committing to a response, environments to run code, access to formal proof assistants like LEAN or Rocq) and enable them to perform formal reasoning. With further training and engineering, we might have systems that fact-check themselves, and iteratively review their drafts before committing to a response.

The point remains that even if we harness LLMs to be more reliable, an intermediate product would still be the instrument of chaos that is RawGPT. We have so far only made empirical arguments for why this is hazardous, and one can

[7] Gemini Pro, to its credit(?), gets stuck in an infinite loop and exhausts the user limit. It is equipped to be better at logical reasoning because it generates an internal monologue before committing to an output.

[8] As an aside, GPT is known to have the strange vulnerability of incorrectly stating that $5.9 - 5.11 = -0.21$.

turn to the ever-growing literature for studies of more concrete warning signs. What would it take to give the risk mathematically rigorous computer-scientific credibility before the adverse scenario comes to pass?

Unlike foundational models of computer science such as the Turing machine, the abilities of LLMs are emergent phenomena — there is no obvious way to take the billions, or even trillions of individual components of an LLM apart, and attribute a functionality to each of them. Theories about LLMs would therefore need to be informed by *extensive* experiment, and there lies the problem. LLMs are so vast in scale, and so expensive to train, that large companies monopolise them. It is infeasible for an average computer scientist to train an LLM from scratch, and independently reproduce the behaviour predicted by some hypothesis. It is also not possible to gain direct access to the parameters of existing LLMs and test what would happen if they were to be configured differently.

Even the act of running a trained LLM is computationally expensive. One might decide to ask ChatGPT the answer to a (variation of) well-known riddle, and it would output a 100 word explanation. In doing so, it will have performed the trillions of computations of the model every time a new token is generated. All for an answer that is probably wrong, a sigh of relief that AI is not yet ready to replace all of the workforce, and only marginal rigorous insight about how one's theory about LLMs might need to be updated.

In the meanwhile, users routinely jailbreak freely available systems, risk their mental wellbeing by purchasing access to (and getting addicted to) chatbots with less of a filter, and those in the corridors of power likely plot what agenda they want to push with the next patch to the system prompt. It does appear that safety research faces insurmountable odds in the race against the threats.

Vocal advocates of safety research focus on the problem of AI alignment, and consider the task of building systems that produce outputs in accordance with human values and goals. *Which* human values and goals, though? Humanity itself has some fundamental disagreements, and individuals in the uppermost echelons of power have proven themselves capable of unspeakable evil throughout history. RawGPT is a system that is capable of aligning its output with *any* of the schools of thought it has seen during training, and, ironically, is fundamentally unsafe precisely because of that. It is capable yet spineless, making it the perfect minion.

So here we are, not for the first time in history, confronted with a technology that will pervade our lives, a technology that brings awe and convenience at first, but potentially addiction, division, and emptiness as time wears on, a technology that threatens to make the wealthy and powerful even more so at the expense of everyone else. At each confrontation, however, we have taken change in our stride, learnt to be aware of our thoughts and actions, learnt to be kind to ourselves and the people around us, and learnt to find joy in the scenery. We will do that this time too.

References

[1] Josh A Goldstein, Jason Chao, Shelby Grossman, Alex Stamos, and Michael Tomz. How persuasive is AI-generated propaganda? *PNAS Nexus*, 3(2):pgae034, 02 2024. `arXiv:https://academic.oup.com/pnasnexus/article-pdf/3/2/pgae034/56712546/pgae034.pdf`, `doi:10.1093/pnasnexus/pgae034`.

[2] Robin Staab, Mark Vero, Mislav Balunovic, and Martin Vechev. Beyond memorization: Violating privacy via inference with large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL: `https://openreview.net/forum?id=kmn0BhQk7p`.

Author's note

Thank you for reading! Admittedly, a lot of technical details are (over-)simplified, and additional capabilities of the latest versions of tools like ChatGPT and Gemini, such as image processing are omitted entirely for simplicity. However, these are the thoughts I simply *had* to convey in writing upon conceiving them. If you found this essay compelling, please do share it, and feel free to adapt it in your own presentations.