Everybody Calm Down

The Wonder. *Of course* I'm annoyed. Over and over again, I have been finding myself in the middle of an argument that, as it appears, became less academic and more personal. The never-ending altercation between generative linguists and computer scientists, both of which encompass some of my greatest role models as well as my dearest friends, about the reality of the wonder of modern science: machine-generated language.

The problem (over)simplifies as follows: on the one hand, generative linguists get triggered upon almost any mention of large language models (LLMs) for it always comes with this modest reminder that their work is obsolete, that there is no further need to understand the underlying structures of the human language as natural speech generation is better off without them. On the other hand, with each little improvement natural language processing (NLP) engineers bring to the table, they get reminded that what they produce is nothing but a fake, an illusion, a false promise they can never fulfil, that no machine can possibly compete with a human speaker because memorising data cannot compare to good old human thinking.

Stranded. I'm well aware that I have little audience left for I refuse to take sides: I'm too much of a traditional linguist to most of the NLP community and too much of an LLM optimist to most generative linguists. This essay is meant for the remaining handful who sense that this perceived need for self-defence in either of these two (complementary rather than contesting) fields might be superfluous, self-inflicted, and eventually counterproductive. It aims to appeal to those who are, instead, genuinely open to understanding the synergies between traditional linguistic theories and the advancements in machine-generated language.

Inside the Machine's Mind. Before entering such a debate, everyone needs to be on the same page about what machine-generated language actually is. Above all, it's a product of artificial intelligence (AI). It is maybe worth noting that any modern AI is not trained to merely 'be smart' but rather to solve very specific kinds of problems. Even amongst LLMs, the type of AI trained to communicate with humans, one could find a wide range of architectures. Hence, the following script is limited to what's necessary to understand where machine-generated language comes from rather than a profound explanation of the mechanics of modern language models.

The Three Phases. Any AI, including LLMs, acquires its intelligence through machine learning (ML) which usually spreads across three phases: (i) training, (ii) validation, and (iii) testing. In training, AI browses examples of problems paired with their respective solutions and adjusts the *parameters* of its prediction algorithm based on the relationships it observes. The most powerful LLMs typically refine an astounding number of parameters (OpenAI's GPT-4 has over 1.7 trillion weights, while its predecessor, GPT-3, had 175 billion), and every single one of them pays attention to every bit of data the model receives.

During validation, AI applies its pre-trained algorithm to a set of entirely new problems. Complex models occasionally ace the training yet show poor accuracy on validation data, which often implies *overfitting*: the phenomenon of AI learning to fit its training set (e.g., by overfocusing on its noise) but failing to generalise to unfamiliar examples. While sporadic errors during validation fine-tune the algorithm (much like in training), frequent errors that suggest overfitting might require interventions in the model's architecture.

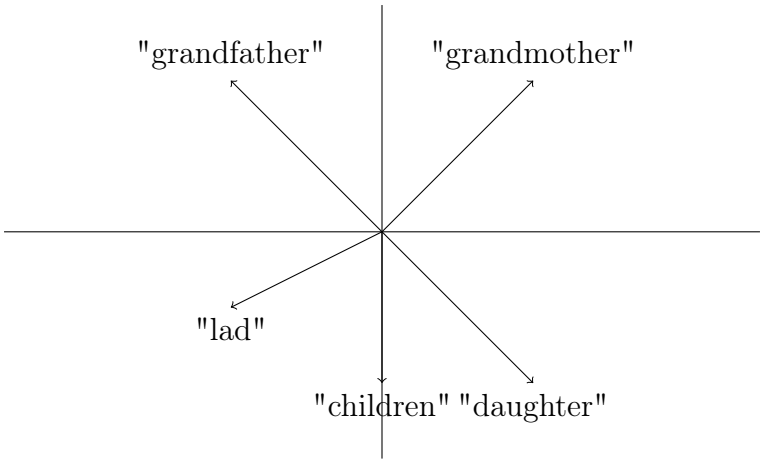
Training and validation repeat in epochs until the model attains sufficient accuracy to

undergo testing. Unlike validation, the testing phase is no longer used to adjust the prediction algorithm but rather to provide a final evaluation of the model's overall expected functionality 'out in the wild'.[1] The whole ML process—from training to testing—requires a lot of computational capacity: training a model of GPT-3's degree on a regular home computer would take ∼355 years.

Figure 1: This flowchart shows the process of machine learning.

Language to Numbers. To tackle NLP tasks, LLMs get introduced to examples of natural language (written or transcribed)—which, to a machine, is initially nothing but a meaningless stream of characters. Instead of viewing human language as a sequence of sounds or words, LLMs split it into a sequence of *tokens*. A token is a labelled (usually indexed) part of the text[2] which enables LLMs to spot its every instance in the training data, keep track of its frequency, and store its unique additional information.

Most popular LLMs are trained on billions of tokens. They don't need all this data to memorise the patterns. They need it to be able to evaluate each (unique) token's *embedding*. Token embedding is an n-dimensional vector representing the token's semantics in an n-dimensional space of parameters (where n is a very big number). This vector probably contains some basic information about the token such as its usual position in a sentence and word category but also more complex details such as animacy and argument structure. I'm saying probably because modern computer science has so far succeeded in interpreting AI models with a single hidden layer. GPT-4 has 120.



[1] A curious reader can access more detailed insights regarding training, validation, and testing data, accompanied by tangible examples, on MLU-ExplAIned's interactive website.

[2] One token may but does not necessarily need to correspond to one complete word—it may as well correspond to a smaller unit such as a morpheme. An interested reader may try to break their own example text into tokens using OpenAI's online Tokenizer.

Figure 2: This chart shows five vocabulary words represented in a 2-dimensional space (with the parameter of *age* on the vertical and the parameter of *gender* on the horizontal axis). LLMs such as OpenAI's GPT-3 have a unique representation for each token in a vector space with thousands of dimensions. All of the vectors' parameters are learnt in training rather than prescribed.

Find the Next Token. As mentioned earlier, AI first needs a problem to be able to search for a suitable solution. LLMs are trained on billions of tokens, but how do these tokens help them solve any actual problems? Before answering that question, it is important to recognise that the primary purpose of generative LLMs is to talk to us, to communicate in a language humans understand. Somewhat unsurprisingly, the biggest problem an LLM encounters then turns out to be neatly human: 'What do I say next?'

When LLMs browse their training data, they look at every individual token in the example sequence as a form of input and adjust their prediction algorithm to try to generate the next best token as an output. During the performance, they randomly pick a first token and proceed to generate the rest of the response. (Note that the chances of picking any given initial token are not uniform across all possible tokens but rather biased based on the prompt the model receives or previous discourse.)

Creativity. A careful reader might ask: if AI learns to find the next best token, how come the responses generated based on the same data are not always identical? LLMs do not merely remember that the most common token to follow token x is token y. They also know that token y follows token x $p\%$ of the time and that token z follows token x $q\%$ of the time. Subsequently, an LLM will proceed to generate token y after token x $p\%$ of the time and token z $q\%$ of the time, preventing thus monotonous output.[3]

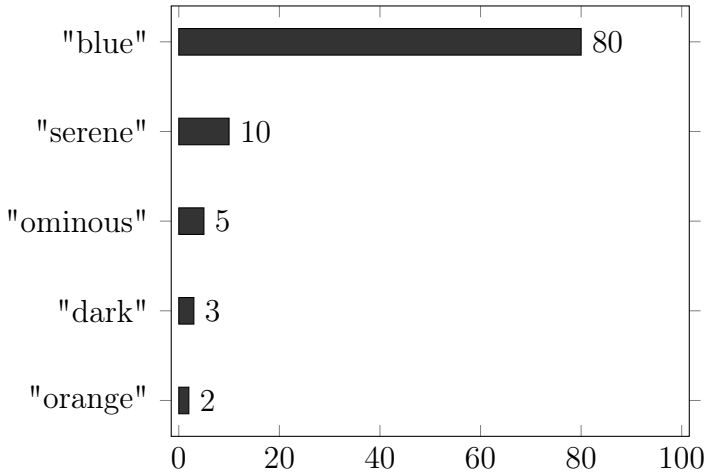


Figure 3: Visual representation of the next best token to continue the sequence: "The sky is...".

However, LLMs do not merely remember which token is to come next—they, crucially, remember what *kind* of a token is likely to follow. (Recall that LLMs have access to a lot more

[3]This probabilistic parameter is called *temperature* and it is used to control the randomness and diversity of generated output. LLMs with high temperature will be more prone to choosing the less common next tokens than LLMs with low temperature.

information embedded in each token such as its usual position and semantic representation.) This enables them to produce grammatically correct sentences, describe logical relations, maintain the sentiment and tone, and much more—even when asked to give an inventive response to a prompt they have never faced before.

One More Cave Away. After all these lines that explained how LLMs generate language, it is finally time someone clarified what it is like, to look at the world through the eyes of a machine. With a little caution and a little more enthusiasm, I subscribe to the idea of comparing human and AI realities to learning about the world from a cave. Millennia before the idea of AI even crossed anyone's mind, Plato in Ancient Greece argued that rather than engaging with one singular reality, humans are like prisoners in a cave constructing their own realities based purely on any sensory input available to them.[4]

This privilege of immediate access to the world we live in is available to humans but not to machines. Machines can't see, hear, smell, taste, or touch anything we can, but they can still construct their reality based on what humans say about their experiences. Unlike little humans, machines do not acquire languages through a word-to-world mapping process but through building a network of expressions that relate to one another based on the conversations they overhear us having—from one more cave away. Do they need more data to learn language compared to humans? Sure, but I think that's a fair compensation.

The Real Deal. I've heard some say that we don't even have a vocabulary to describe the process by which machines acquire their problem-solving skills, but we do. Let's call machine learning for what it is: it's learning. Not memorisation and not pattern-matching. *Learning.* The massive, interconnected system brings machines as close to consciousness as it gets. It makes them aware of the fine line between the real and the unreal, the possible and the impossible. And I'd encourage anyone who's still not on board to ask one eye-opening question: 'Just because something thinks differently from you, does that mean it's not thinking?'

Why Bother. Now that we've established—thank you very much—that mankind has built intelligent machines generating real thoughts and real language, we have basically won over generative linguistics, right? I claim that only someone who has never touched any generative linguistics can say something along those lines. In fact, whoever says that formal linguists should just call it a day surely also thinks that interpretability[5] research is unnecessary. Right?

The times when generative linguistics was merely about composing a set of rules that would formulate any and all sentences in any language are long gone. Modern syntax is about considering the black box called the human mind and figuring out how come it generates speech one way but not another. The field shows, step by step, that grammar is not just an incidental surface-level morphological decoration but a [real thing] reflecting underlying cognitive processes.

The Puzzle. Is generative linguistics going to cure cancer? Probably not. Most sciences won't. There is little space in the field for folks like myself (who have internalised the

[4]An interested reader is encouraged to further explore the idea of reality and its perception through Plato's Allegory of the Cave.

[5]For those not familiar with the topic, AI interpretability is a key area of research in AI safety, helping humans understand the hidden part of machine decision-making and contributing thus to a more aligned technology.

potentially unhealthy idea that if they don't change this scary world for the better, nobody will), yet I can't seem to give it up because I've come across nothing else that would entertain my intellect as much as the puzzles of theoretical syntax.

(1) a. [$_{\text{subject}}$ žek-ā] is b-exu-r-s
[$_{\text{subject}}$ man-**ERG**] bull.ABS.III III-die-CAUS-PST.WIT
'The man killed the bull.'

b. [$_{\text{subject}}$ is] b-exu-s.
[$_{\text{subject}}$ bull.ABS.III] III-die-PST.WIT
'The bull died.'

Figure 4: This is ergative, my favourite case. It spreads unevenly across subjects in some languages. It marks agency, and so it will never appear on subjects that experience events but do not perform any action willfully or by choice.

Generative linguistics explains processes we all do subconsciously, effortlessly, and in real-time yet [overwhelmingly] correct. It works with data that is computationally difficult to crack. Only the smartest get it right. But when they do, the results are unrealistically satisfying. They offer a new perspective to our understanding of the world by explaining language, something so uniquely human. And I think that's beautiful.

Bridges. To all my nerdy friends: I know what it's like to treat machines with a great deal of respect—and a little bit of love. I know the very specific kind of excitement that comes when the machine does the expected—and the very specific kind of pride when it comes up with something novel all by itself. It doesn't feel fake. It's all very real. But I also think there's more to understand about language than merely how to generate it, and that's where I've seen syntacticians making almost unbelievably accurate predictions.

To all my fellow linguists: much like most of you, I share the belief that humans are inherently linguistic creatures, that language is so essential to our existence that there must be an inborn faculty such that it allows us to further acquire language regardless of where the lottery of birth places us. But I've also grown up in an alien world. I know what it's like to learn about concepts one doesn't have immediate access to through alternative means. I know that there is more than a singular way to do real intelligence. And I think machines are worth listening to, too.

I'm still not taking sides. This is not a competition. There *are* no sides to be taken. One can be a computational linguist *and* a generative syntactician. The hope is that the reader can see it's not an oxymoron now.