1 Fundamental Principles of NLP

Before explaining the basic mechanisms of LLMs, it is important to understand that there is (currently) no singular AI that gets trained to ‘be smart’. Instead, various kinds of AI get trained to solve various kinds of specific problems. Even amongst LLMs, one will encounter a rich diversity of architectures and training principles. Rather than a universal explanation of LLMs, this section thus aims to provide a simplified overview of various aspects of NLP relevant to this paper’s latter arguments.[1]

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 170 trillion weights (its predecessor, GPT-3, had 175 billion) (OpenAI, 2023).

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 memorising 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’.[2] The whole ML process—from training to testing—may take several months (e.g., GPT-3 took $\sim$34 days) and requires considerable computational power (Naranayan et al., 2021).

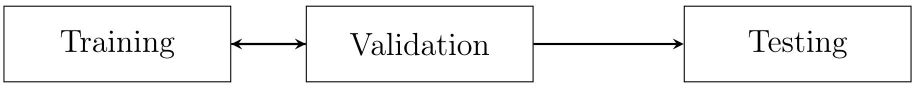

Figure 1: This flowchart shows the process of machine learning.

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[3] 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 (e.g., GPT-3 was trained on 300 billion (Li, 2020)). Thanks to all this data, LLMs can evaluate each (unique) token’s positional encoding (the token’s usual position in a sequence) as well as its token embedding (an n-dimensional vector representing the token’s semantics in an n-dimensional space of parameters).[4] Every token thus becomes associated with a substantial amount of learnt information

[1] Non-critical but potentially useful pieces of information such as this one will be posted in the footnotes.

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

[3] One token may but does not necessarily need to correspond to one complete word. An interested reader may try to break their own example text into tokens using OpenAI’s online Tokenizer.

[4] LLMs such as OpenAI’s GPT-3 have a unique representation for each token in a vector space with thousands of dimensions (Greene et al., 2022). All of the vectors’ parameters are learnt in training rather than pre-programmed.

ready for further analysis and future use in problem-solving.

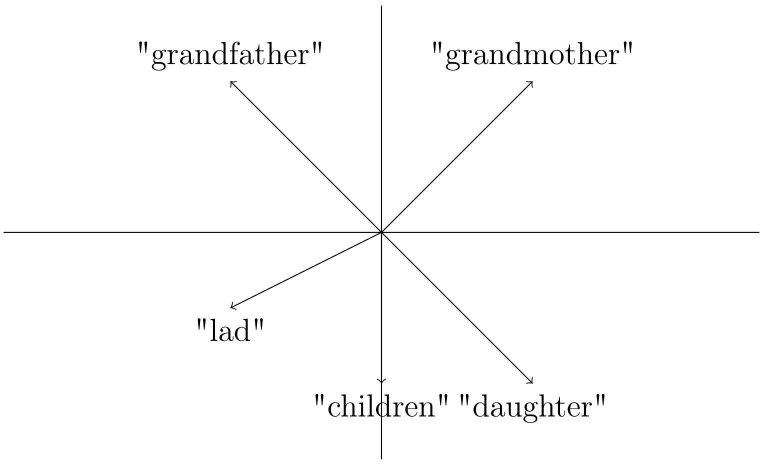

Figure 2: This chart shows five vocabulary words represented in a 2-dimensional space (with the parameter of *age* on the vertical and the parameter of *gender* on the horizontal axis). LLMs usually account for thousands of these parameters.

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.)

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.[5]

[5] 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.

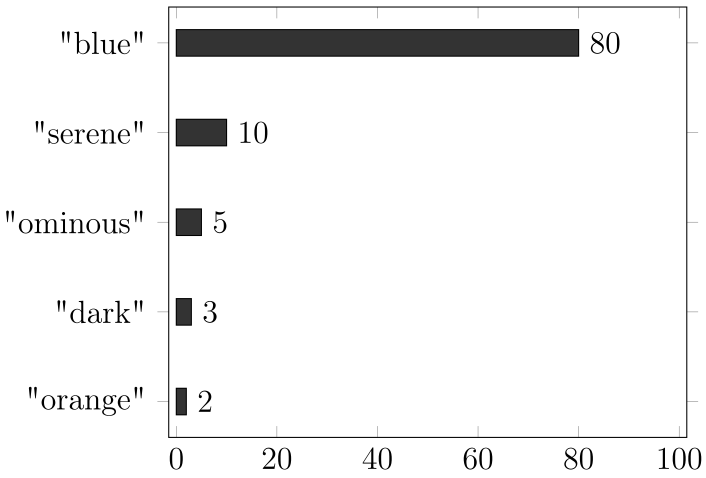

Figure 3: Visual representation of the next best token to continue the sequence: "The sky is...".

Moreover, LLMs do not merely remember which token is to come next—they also remember what *kind* of a token is likely to follow. (Recall that LLMs have access to a lot more information associated with each token such as its usual position and semantic representation.) This enables them to produce grammatically correct sentences, replicate 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. While the above paragraphs have explained how LLMs produce human-like language, none clarified whether AI understands what this world is actually about. 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 using the only available sensory input, the shadows from the outside.[6] AI has no access to any senses; does it have any reality at all?

An attentive reader may have noticed that LLMs do not merely acquire an immense bank of independent tokens. Instead, they have a knowledge of how many unique tokens relate to one another. This is how AI creates its own reality—not by directly interacting with the world but by overhearing the stories humans tell about their interactions and recognising the relationships between them (Kulveit, 2023). AI is a mighty learner—it just learns about this world from one more cave away. Does this necessarily imply it does not understand it?

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