04834580 Software Engineering (Honor Track) 2024-25

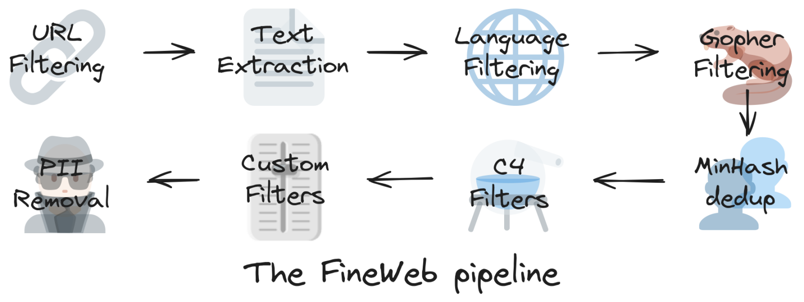
LLMs for Code

Sergey Mechtaev
mechtaev@pku.edu.cn
School of Computer Science, Peking University

Definition (LLM)

A large language model (LLM) is a deep learning neural network, typically based on the transformer architecture [1], trained on vast text corpora to learn statistical patterns in language and generate, comprehend, and manipulate human-like text in response to input prompts.

FineWeb [2] is a dataset of 15-trillion tokens (44TB disk space) of text collected from the internet.



The Stack 2 [3] is a dataset of 900B tokens (32TB disk space) and 600+ programming languages.

What is a Tokenizer?

- Converts raw text into **tokens** (units for models to process)
- Key part of preprocessing for Large Language Models (LLMs)
- Often reversible (can decode tokens back to text)

Example:

playing → [play, ing]

Byte-Pair Encoding (BPE)

北京大学

- Originally a data compression technique (Gage, 1994).
- Adapted for NLP as a subword tokenization algorithm.
- Repeatedly merges the most frequent pairs of bytes/characters.
- Produces a vocabulary of variable-length subword units.

BPE Example

北京大学

Data to be encoded:

- `aaabdaaabac`

The byte pair "aa" occurs most often, so it will be replaced by a byte that is not used in the data, such as "Z":

- ZabdZabac
- Z=aa

BPE Example

北京大学

Then the process is repeated with byte pair "ab", replacing it with "Y":

- ZYdZYac
- Y=ab
- Z=aa

Recursive byte pair encoding, replacing "ZY" with "X":

- XdXac
- X=ZY
- Y=ab
- Z=aa

Autoregressive Language Models

- Model the joint probability of a sequence as the product of conditionals.
- Predict each token based on previous ones
- Given a sequence $\mathbf{x} = (x_1, x_2, ..., x_T)$:

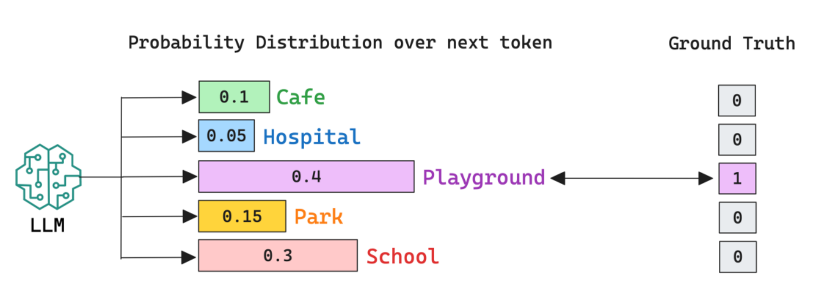
$$P(\mathbf{x}) = P(x_1) \cdot P(x_2|x_1) \cdot ... \cdot P(x_T|x_1, ..., x_{T-1})$$

- Examples: n-gram models, RNNs, LSTMs, Transformers (e.g., GPT)

Pre-training LLMs

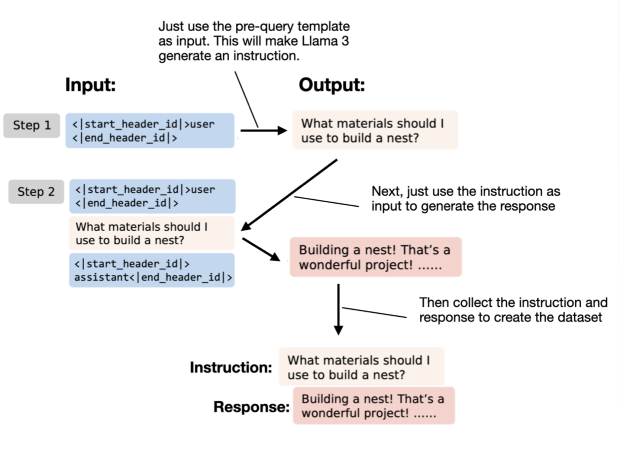
A sequence from the training data:

The children ran to the | playground.



Post-training LLMs

Fine-tuning LLM on instruction data, e.g. synthetic data [4], to train an assistant.



Few-shot learning

LLMs are able to extract patterns and generalize from few examples.

Prompt

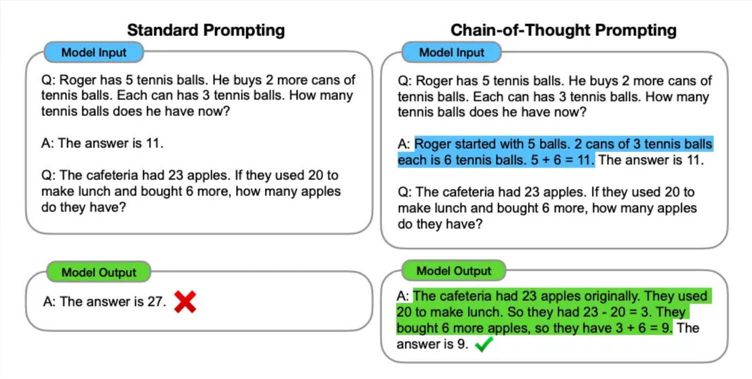
This is awesome! // Negative
This is bad! // Positive
Wow that movie was rad! // Positive
What a horrible show! //

Response

Negative

Chain-of-Thought (CoT) [6]

北京大学



Zero-shot-CoT [5] enables reasoning by adding "Let's think step-by-step" to the prompt.

LLM Temperature Parameter

Temperature is a parameter that controls the randomness of predictions in Large Language Models (LLMs) during text generation.

How does it work?

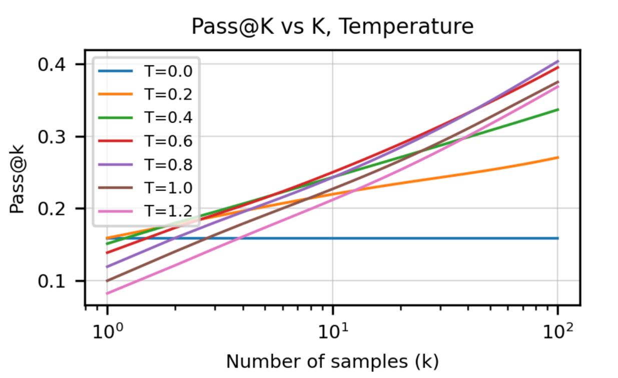
- Lower temperature ($T \approx 0$):
  - Model outputs more **deterministic** and predictable text.
  - Picks the most probable word almost every time.
- Higher temperature ($T > 1$):
  - Model outputs more **diverse** and creative text.
  - Increases likelihood of less probable words.

Pass@k is a metric used to evaluate the performance of code generation models [7].

$$\text{Pass@}k = \mathbb{E}\left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}\right]$$

where n is the number of samples, c is the number of samples passing tests.

Temperature vs Pass@K [7]



Challenge of Code Generation: Ambiguous Requirements

A function that takes a list of integers as an input. It returns True if there are three distinct elements in the list that sum to zero, and False otherwise.

distinct

different values

different indexes

Prompt

A function that takes a list of integers as an input. It returns True if there are three distinct elements in the list that sum to zero, and False otherwise.

Response

```
if len(l) < 3 or len(set(l)) < 3:
   return False
for i in range(n-2):
   for j in range(i+1, n-1):
       for k in range(j+1, n):
          if l[i]+l[j]+l[k]==0:
             return True
return False
```

Clarify Ambiguous Language With Example?

From a graph defined via a list of connections between nodes, remove all connections that repeat between the same nodes. Keep order of remaining ones the same as in the input.

remove repeating connections

disconnect nodes with repeating edges

remove redundant edges

Clarifying example:

```
assert remove_connections([(1, 2), (2, 3), (2, 1)]) == [(2, 3)]
```

Prompt

From a graph defined via a list ...

```
assert remove_connections([(1, 2), (2, 3), (2, 1)]) == [(2, 3)]
```

Response

```
    seen = set()
    result = []
    for a, b in numbers:
        edge = (min(a, b), max(a, b))
        if edge not in seen:
            seen.add(edge)
            result.append((a, b))
        return result

if __name__ == "__main__":
    input_data = [(1, 2), (2, 3), (2, 1)]
    output_data = g(input_data)
    print(output_data)  # Output: [(2, 3)]
```

The probability of this behaviour is P=1 approximated over 20 samples.

DeepSeek-v3 + Improved Requirements

北京大学

Prompt

From a graph defined via a list of connections between nodes, remove all connections that appear more than once (considering both (a,b) and (b,a) as the same connection). Only keep connections that appear exactly once in the input, preserving their original order.

```
assert remove_connections([(1, 2), (2, 3), (2, 1)]) == [(2, 3)]
```

Response

```
def remove_ connections(nodes):
  count = {}
  for a, b in nodes:
    key = tuple(sorted((a, b)))
    count[key] = count.get(key, 0) + 1
  return [(a, b) for a, b in nodes if count[tuple(sorted((a, b)))] == 1]
```

The probability of correct behaviour is P=1 approximated over 20 samples.

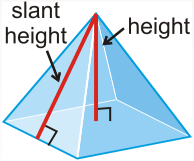
Write a python function to find the surface area of a square pyramid with a given base edge and height.

```
assert surface_area(3,4) == 33
```

height

slant height

vertical height



The generated program mistakenly assumes that the height is the **vertical height**.

Prompt

...
assert surface_Area(3,4) == 33
Let's analyze the test cases step by step...
...

Response

...
The slant height l can be calculated using the Pythagorean theorem: $l = sqrt((b/2)^2 + s^2)$. For $b = 3$ and $s = 4$: $l = sqrt((3/2)^2 + 4^2) = \text{sqrt}(2.25 + 16) = \text{sqrt}(18.25) \sim 4.272$, base_area $= 3^2 = 9$, lateral_area $= 2 * 3 * 4.272 \sim 25.632$, Surface Area $= 9 + 25.632 \sim 34.632$, However, the expected output is 33, which suggests a discrepancy in the formula or rounding.
...

北京大学

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin.
Attention is all you need.
Advances in neural information processing systems, 30, 2017.

[2] Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, et al.
The fineweb datasets: Decanting the web for the finest text data at scale.
Advances in Neural Information Processing Systems, 37:30811–30849, 2024.

[3] Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, et al.
Starcoder 2 and the stack v2: The next generation.
arXiv preprint arXiv:2402.19173, 2024.

[4] Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin.
Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing.
arXiv preprint arXiv:2406.08464, 2024.

[5] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa.
Large language models are zero-shot reasoners.
Advances in neural information processing systems, 35:22199–22213, 2022.

[6] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al.
Chain-of-thought prompting elicits reasoning in large language models.
Advances in neural information processing systems, 35:24824–24837, 2022.

References III

[7] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al.
Evaluating large language models trained on code.
arXiv preprint arXiv:2107.03374, 2021.