INTERACTION WITH LARGE SCALE MODELS

LIACD/1

University of Beira Interior,
Department of Informatics

Hugo Pedro Proença,
hugomcp@di.ubi.pt, 2024/2025

INTERACTION WITH LARGE SCALE MODELS

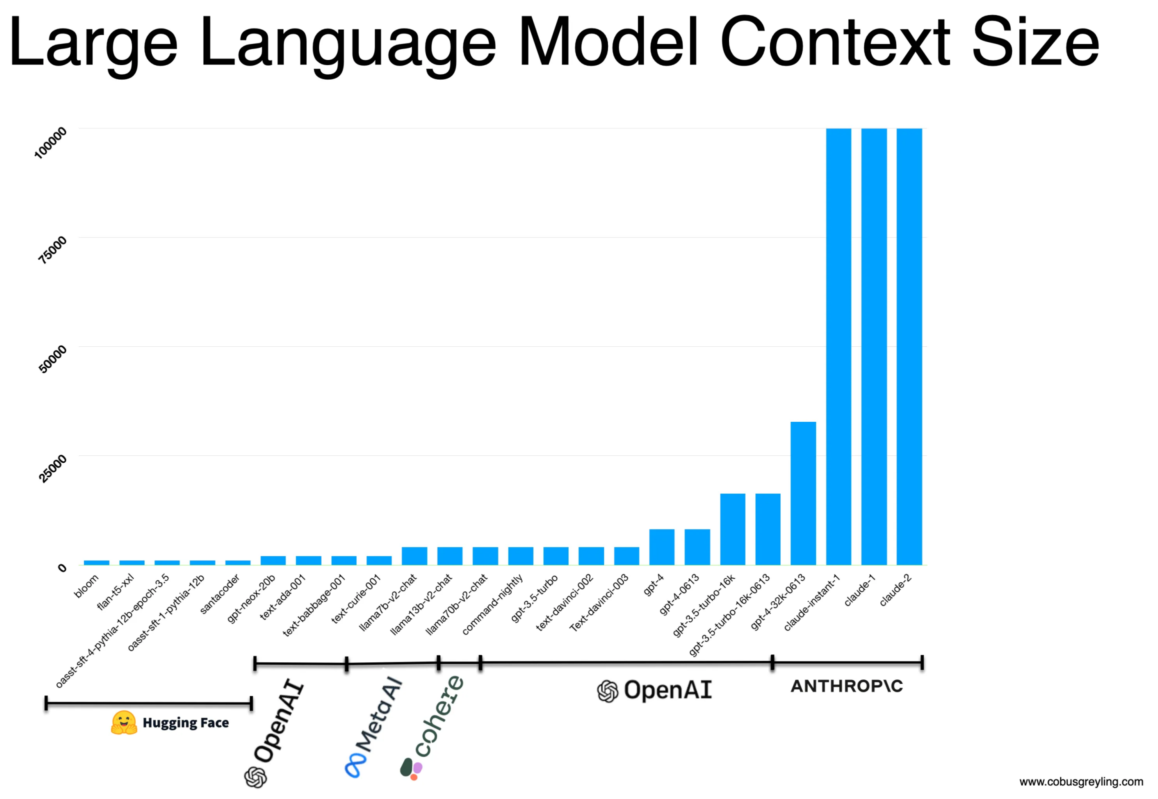
[03]

Syllabus

- Context Length: How much information a model can remember
- Hyperparameters: How different settings affect model responses
- Importance of understanding model internals for better prompt crafting

Context Length

- **Definition:** The maximum number of tokens a language model can proces in a single prompt.
- **Intuition:** Think of it as a working memory"—it can only recall information within a certain limit.
- **Example:** A model with a 4,096-token context length can remember roughly 3,000 words (since tokens include words, punctuation, and subwords).



Source: https://cobusgreyling.medium.com/rag-llm-context-size-6728a2f44beb

Context Length

- **Too Short Contexts (e.g., 512 tokens)**
  - Can handle small prompts but forgets earlier details in long conversations.
  - Example: If summarizing a book chapter, it may lose track of the main theme if the input exceeds the limit.
- **Too Long Contexts (e.g., 32,000 tokens)**
  - Can process entire documents, enabling deep reasoning and better memory retention.
  - Example: A legal AI model can read an entire contract and answer specific questions without missing details.
- **Trade-offs**
  - More memory means slower processing and higher computational cost.
  - Long contexts don't always mean better results—models still struggle with retrieving relevant parts.

Positional Encoding

- State-of-the-art models process all input tokens at once. How does it know the order of the words? The answer is positional encoding, the model’s way of telling the model which word goes first.
- Positional encoding helps the model **differentiate positions** in the input sequence.
- There are two widely used positional encoding methods:
  - **Sinusoidal encoding** (used in GPT): A wave-like function based on the token’s position is added to the input token embedding.
  - **Rotary encoding** (used in Llama): It similarly uses a wave-like function. But instead of adding to the input, it is multiplied with the keys and queries in every layer.
- A transformer layer doesn’t care about the length of the input sequence (context length). All input lengths produce the same number of output vectors.

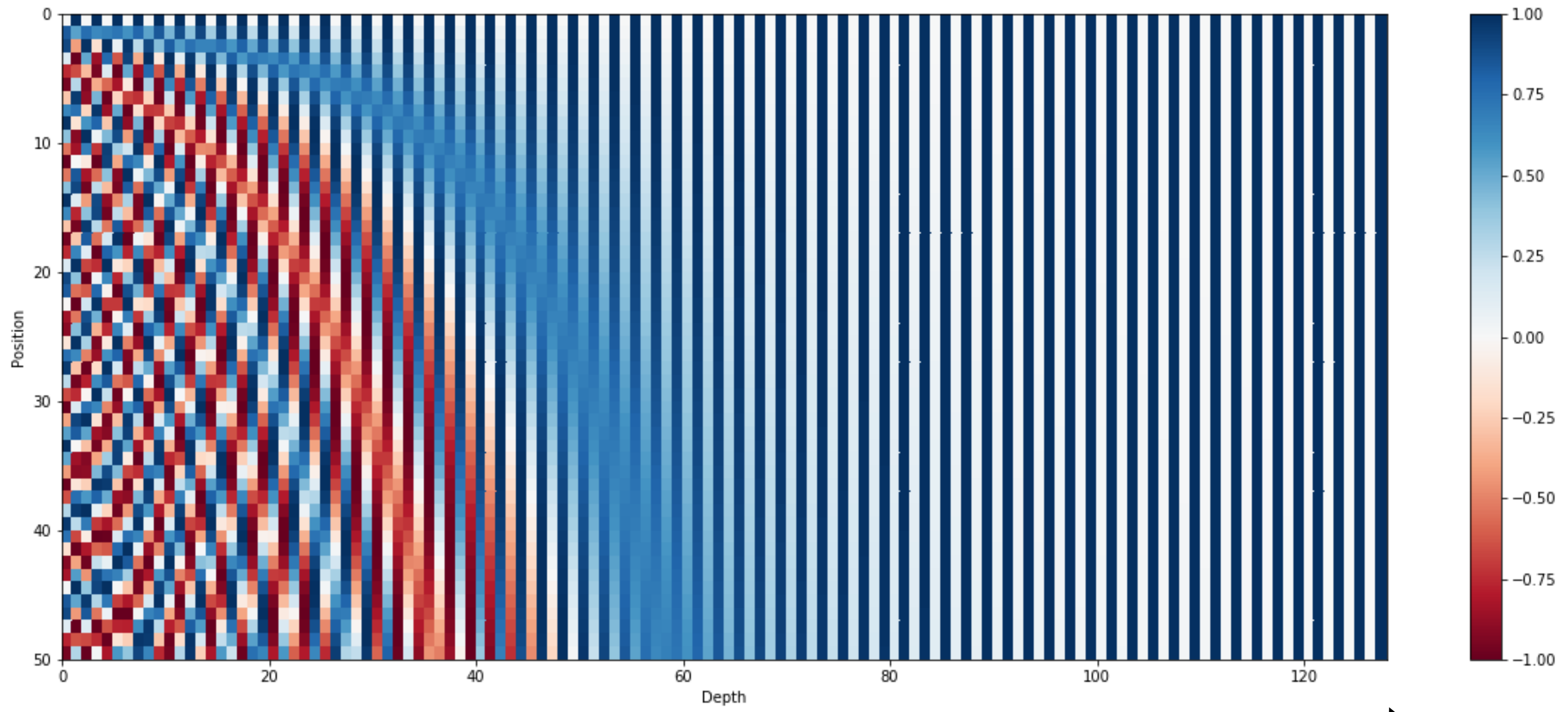
Positional Encoding

- Instead of assigning fixed positions (like simple numbers 1, 2, 3...), Transformers use sinusoidal functions (sine and cosine waves of different frequencies) to encode position.
  - Each position "*t*" in the sequence is mapped to a unique pattern of sinusoidal values.
  - These values are added to the token embeddings, allowing the model to recognize both content and position of words
- Why sinusoids?
  - Different frequencies allow the model to generalize to longer sequences than seen in training.
  - The wave pattern ensures smooth, continuous variation across positions, making it easy for the model to compute relative distances.

$$\vec{p_t} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1}$$

The positional vector contains pairs of sines and cosines of different frequencies. It is added to the embeddings. ("t" is the position of the token in the context)

Positional Encoding

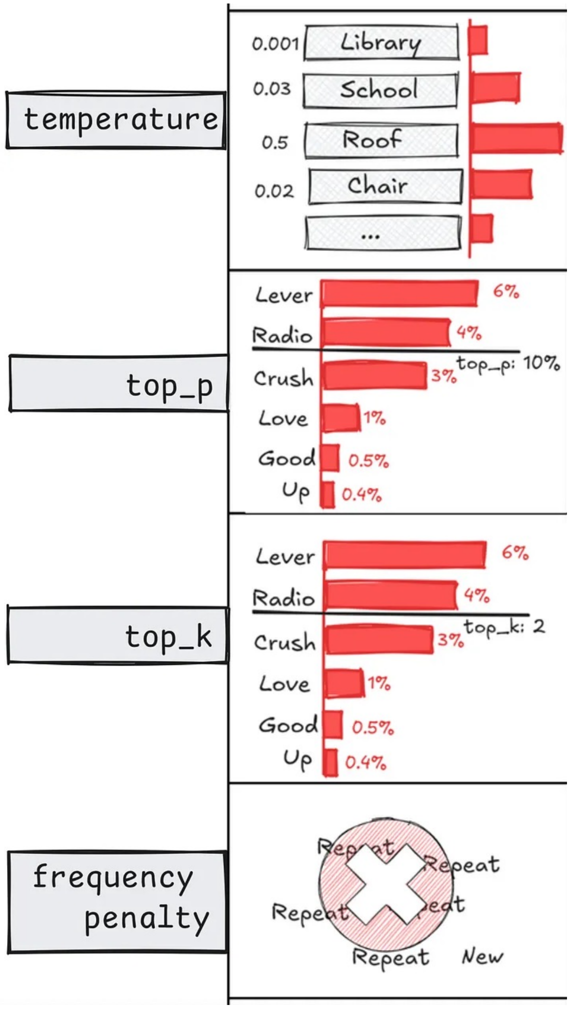


Source: https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Example of the positional encoding vectors produced for a token size = 128 and a sequence length = 50.

Model Hyperparameters

- There is a set of hyperparameters that define how models generate responses
- The most popular parameters are:
  - **Temperature**
    - Controls randomness in output. Higher values imply more creative and diverse responses (yet, less rigorous).
  - **Top-p (nucleus) sampling**
    - Limits the number of top probable tokens to sample from
  - **Top-k sampling**
    - Controls the probability distribution considering when sampling tokens
  - **Repetition penalty (Frequency/Presence)**
    - Penalizes repeated tokens, to avoid loops



Source: https://www.nb-data.com/p/7-llm-generation-parameters-to-know

Temperature

- Controls randomness in model responses
- High temperature (e.g., 0.8–1.2): More creative but inconsistent
- Low temperature (e.g., 0.1–0.3): More predictable but less diverse
- Examples:
  - "*Tell me a story about a dragon.”*
  - Temp 1.0 → Random, unpredictable story
  - Temp 0.2 → More structured, formulaic story
  - *"Summarize this legal document."*
    - Temp 0.2 → Accurate, focused summary
    - Temp 1.0 → Unreliable summary with possible hallucinations

High Temperature
(0.8–1.2)
Storytelling, poetry, brainstorming

Medium Temperature
(0.4–0.7)
Chatbots, balanced creative writing

Low Temperature
(0.1–0.3)
Coding, math, factual responses

Top-K and Top-P Sampling

- Top-K. Fixed number of choices
    - Selects from the top k most probable next words
        - k=50: Picks from 50 likely words
        - k=1: Always picks the single most likely word (greedy decoding)
    - Example: *"The cat sat on the..."*
        - k=50 → "mat, roof, table, floor, grass, book"
        - k=1 → "mat" (most common completion)
- Selects from a dynamic range of words based on cumulative probability
- Example:
    - p=0.9: Words until total probability is 90%
    - p=0.3: More constrained choices
- Top-p: Flexible, probability-based cutoff
    - Balanced configurations help achieve optimal results:
    - High temp + high k/p → Maximum creativity
    - Low temp + low k/p → Precise, deterministic responses

Top-p Sampling (Nucleus Sampling)

- Selects from a dynamic range of words based on cumulative probability
- Example:
  - p=0.9: Words until total probability is 90%
  - p=0.3: More constrained choices
- Top-k: Fixed number of choices
- Top-p: Flexible, probability-based cutoff
- Balanced configurations help achieve optimal results:
- High temp + high k/p → Maximum creativity
- Low temp + low k/p → Precise, deterministic responses