[CHAT-2.0] LARGE LANGUAGE MODELS

What is ChatGPT?

- It is an assistant or **AI chatbot** that has conversations (chats).
- It is not a **large language model** (LLM), but a user interface built around one.
- Chatbot arena: ChatGPT (OpenAI), Claude (Anthropic), Gemini (Google), Grok (X AI), Copilot (Microsoft), Coral (Cohere), Le Chat (Mistral AI), DeepSeek (High-Flyer), Qwen (Alibaba).

Early ChatGPT critique

- Not factual.
- Limited to training data timeframe.
- Limited to public training data, no access to proprietary information.
- No reasoning.
- Not good at math.
- Hallucination.

State of the art

- Advanced usage:
  - Instruction following.
  - In-context learning.
  - Chain-of-Thought (CoT) reasoning.
- Augmentation:
  - Retrieval-augmented generation (RAG).
  - Tool usage: calculator, web search, Python interpreter.

LLM taxonomy

- **Text generation** models: generate new text given another text, called the **prompt**. Other names: causal, autoregressive. Example: Google’s Gemini 2.5 Flash, OpenAi’s GPT-4o mini.
  - Code models.
  - General-purpose models.
  - **Reasoning** models.
- **Embedding models**: generate vector representations. Examples: Google’s text-embedding-004, OpenAI’s text-embedding-3-small.

Tokens

- The **tokens** are the “atoms” in which text is split by the LLM.
- They are typically words, subwords or punctuation. There are also tokens for the beginning and the end of a text.
- One component of a language model is **tokenizer**. The tokenizer splits the prompt based on a **vocabulary** of tokens. For every token, there is an **embedding vector**.

What is text generation?

- The model produces a reasonable continuation of the input text, based on what people have written on billions of webpages (the training data).
- It performs mathematical calculations with the input vectors, using a **neural network** architecture whose parameter values have been determined during the training of the model. The network architecture used by language models is called the **transformer**.
- The resulting vector is a set of **probabilities** for the output token, which is chosen according to these probabilities.

Continuation

- The output token is added to the input tokens. Then, a new output token is generated, and so on, until the end token is generated. The set of tokens generated is the “answer” of the model.
- **Context window**: the maximum number of tokens that the model can manage to respond a single prompt. For reasoning models, this is a relevant parameter.

How do we interact with a language model?

- Chat app: ChatGPT, Gemini, DeepSeek, Perplexity, Groq, LM Studio.
- Programmatic way:
  - Client-server (API): OpenAI, Gemini, DeepSeek, Groq, Hyperbolic.
  - Local model: Ollama, Hugging Face.

Is text generation free?

- Proprietary models: GPT (OpenAI), Claude (Anthropic), Gemini (Google), Grok (X AI).
- Open source: Llama (Meta), Mistral (Mistral AI), DeepSeek (High-Flyer), Qwen (Alibaba), Gemma (Google), Granite (IBM), Command (Cohere).