Nineteenth Session

Alireza Moradi

✉ | LinkedIn | ● | k

Generative AI

Generative AI

- **Generative AI** is a type of artificial intelligence technology that can produce various types of content, including text, code, image, audio ,video and synthetic data.

- Generative AI was introduced in the **1960s** in chatbots. But it was not until 2014, with the introduction of **generative adversarial networks (GANs)** that generative AI could create convincingly authentic images, videos and audio of real people.

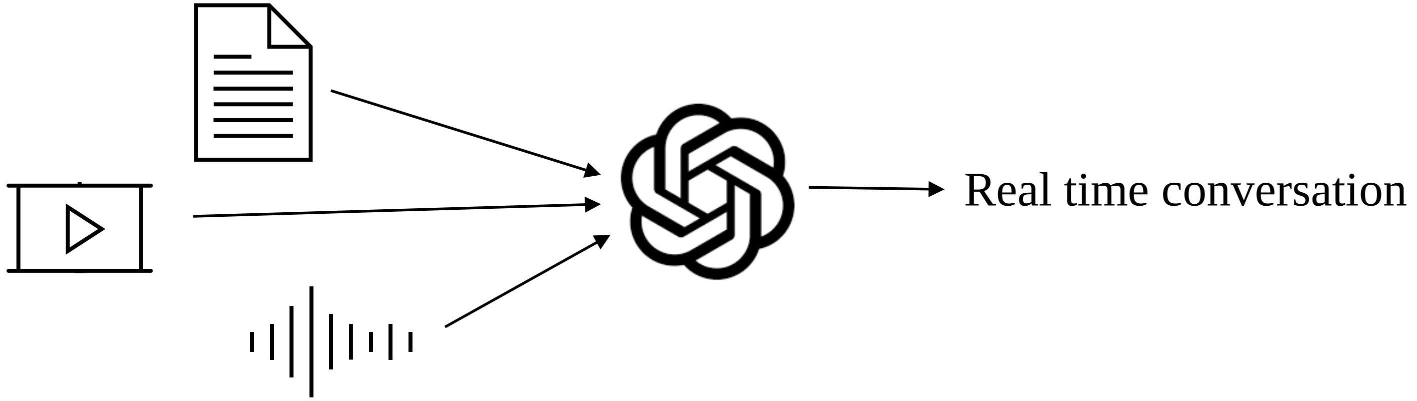
BERT BLOOM Ernie Bot

Stanford Alpaca DALL·E GitHub Copilot sora

Generative AI; Multimodal models

- Innovations in **multimodal AI** enable teams to generate content across multiple types of media, including text, graphics and video.

- The most famous example is GPT-4o from OpenAI.



- It can respond to audio inputs with an average of **320 milliseconds**, which is similar to human response time in a conversation.

Generative AI; Examples

- Text generation tools include ChatGPT, Perplexity, Consensus and Jasper.
- Image generation tools include Dall-E 3, Midjourney, Craiyon, Starryai and Stable Diffusion.
- Music generation tools include Aiva, Soundraw, Beatoven and MuseNet.
- Code generation tools include Codex, GitHub Copilot and Tabnine.
- Voice synthesis tools include Descript and Listnr.ai.
- Video generation tools like Sora, Synthesia and Lumen5.

- There are several platforms that combine various AI models in one UI for a comprehensive user experience like Gemini, Copilot, and GPT-4. We also have websites that incorporate many tools, like Peo.

Image generation

- Image generation tools usually require a **text prompt** describing the desired image.

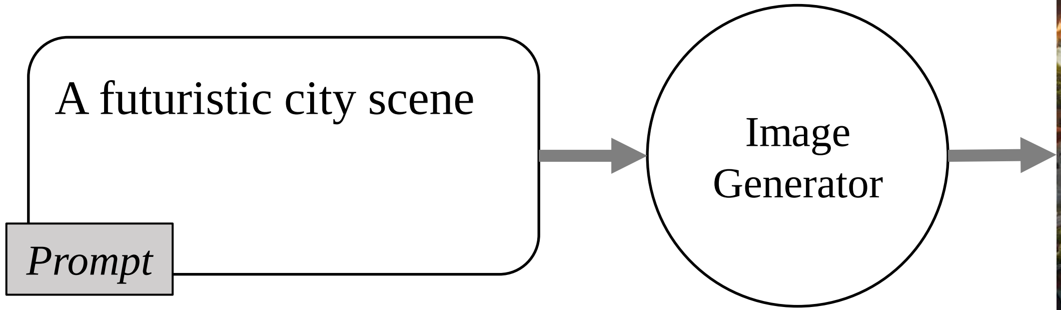


Image generation; diffusion model

- **Diffusion models** are advanced machine learning algorithms that uniquely generate high-quality data by progressively adding noise to a dataset and then learning to reverse this process.

✓ Let's assume we have a picture of an apple. How can we use this image to generate **training examples for a diffusion model** that can create new images of apples?

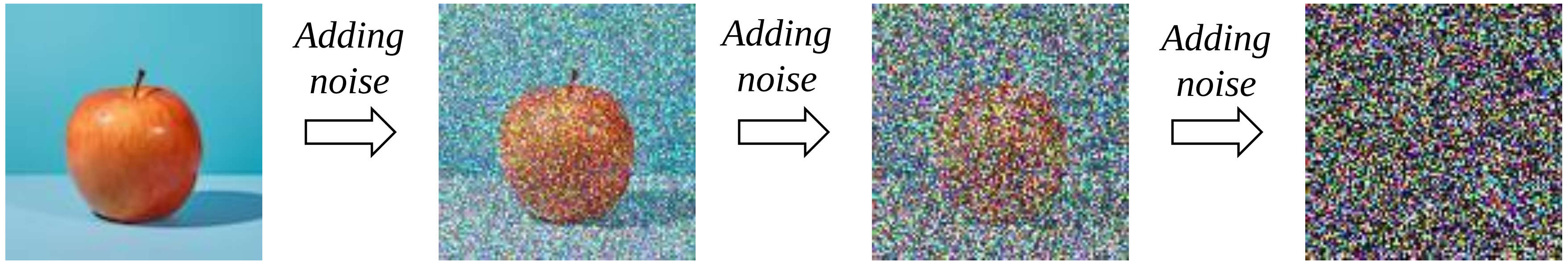


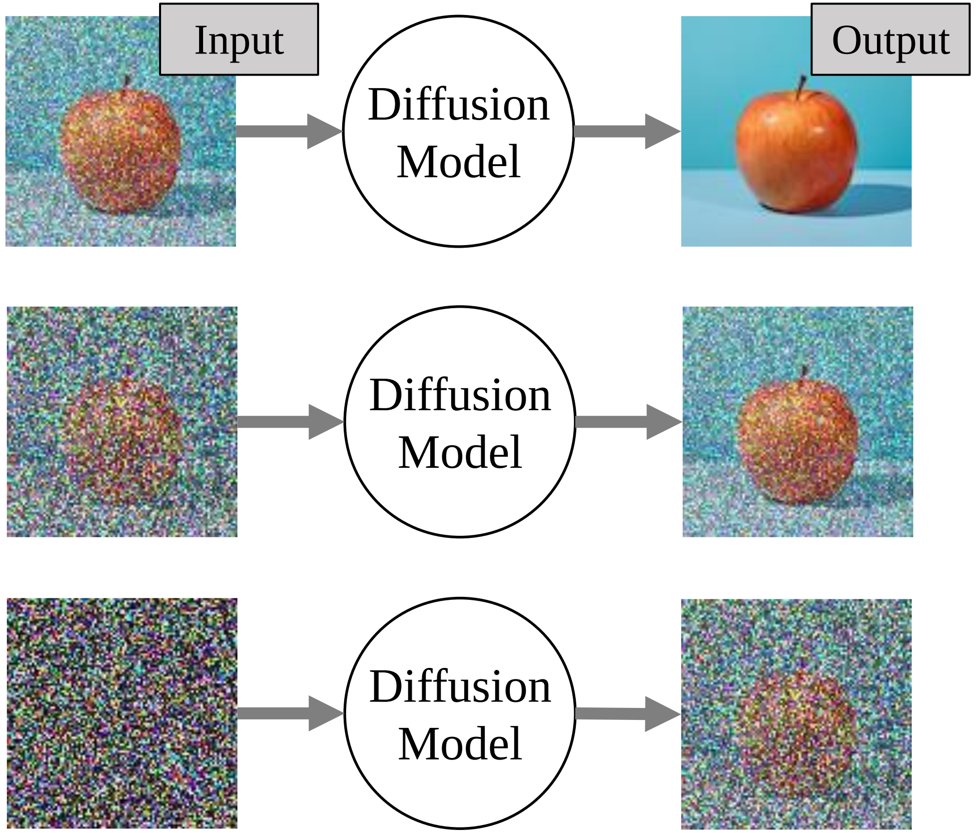
Image 1 Image 2 Image 3 Image 4

Diffusion model; Training

- With each step, the diffusion model generates a slightly less noisy image.

- Training examples:

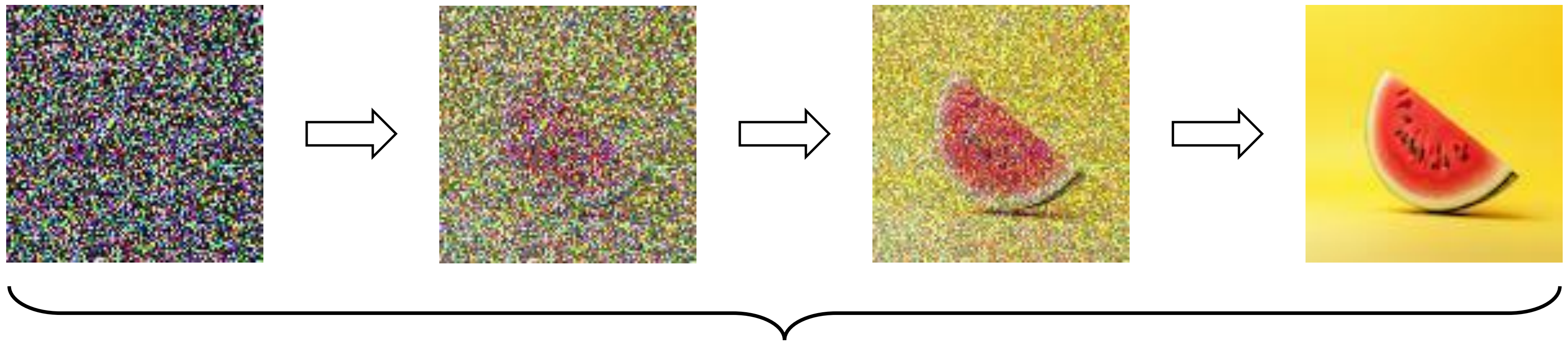
Input	Output
Noisy image	Slightly less noisy image



Diffusion model; Generating

- **Diffusion models** generate images by starting with a pattern of random noise and gradually shaping it into a coherent image.

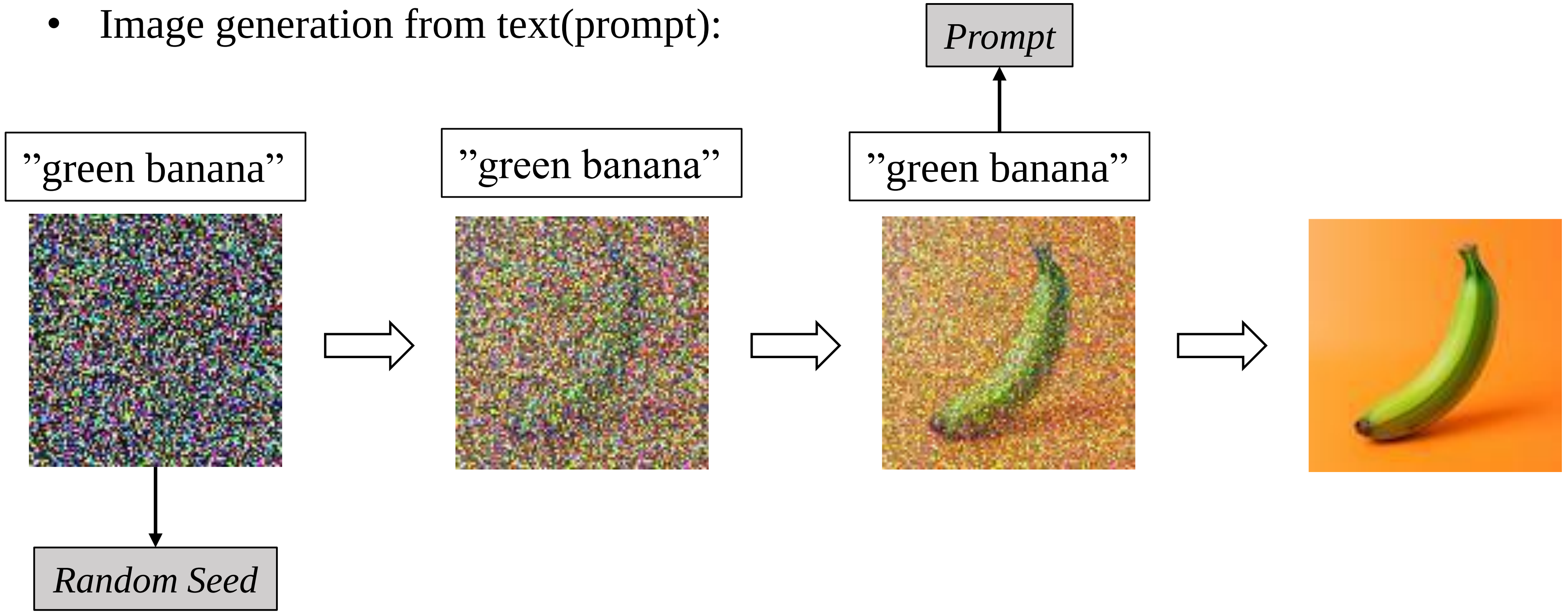
random noise(seed)



Typically ~100 steps for diffusion model

Diffusion model; Generating

- Image generation from text(prompt):

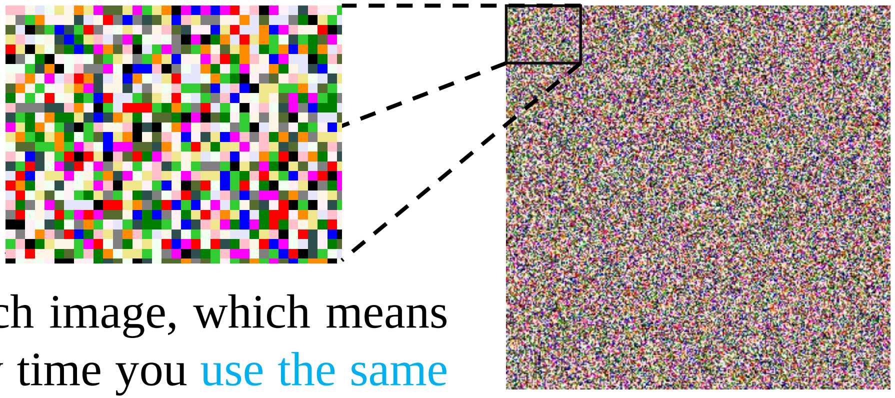


- For example, DALL-E operates on the principles of diffusion models.

Diffusion model; Generating

- By default, **Diffusion models** uses a **random seed** for each image.

- This means that you will get different results every time you use the same prompt.



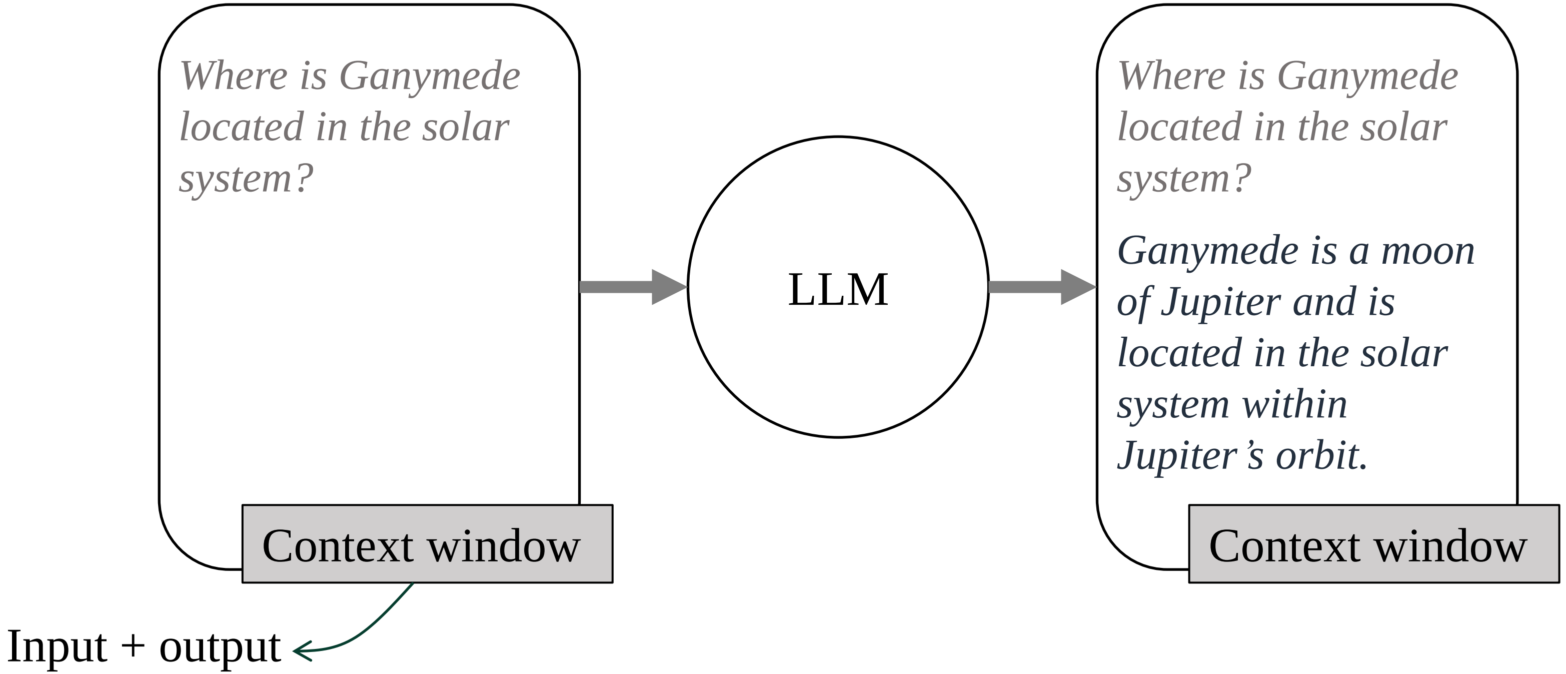
- But you can also specify a seed for each image, which means that you will **get the same result** every time you use the same prompt and the same seed.

- Generating data through iterative denoising is much more time-consuming compared to direct generation methods used by GANs.

Large Language Models

Large Language Models (LLMs)

- **Large language models (LLMs)** are very large deep learning models that are pre-trained on vast amounts of textual data.

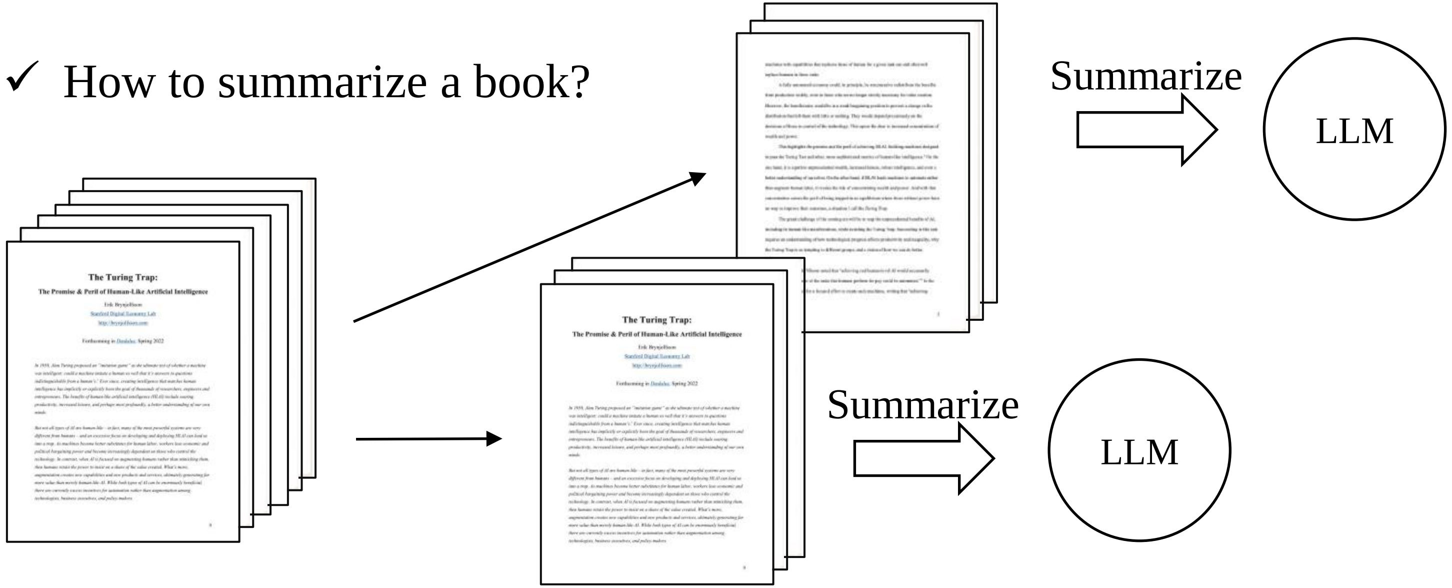

Where is Ganymede located in the solar system?
LLM
Where is Ganymede located in the solar system?
Ganymede is a moon of Jupiter and is located in the solar system within Jupiter's orbit.
Context window
Context window
Input + output

Large Language Models (LLMs)

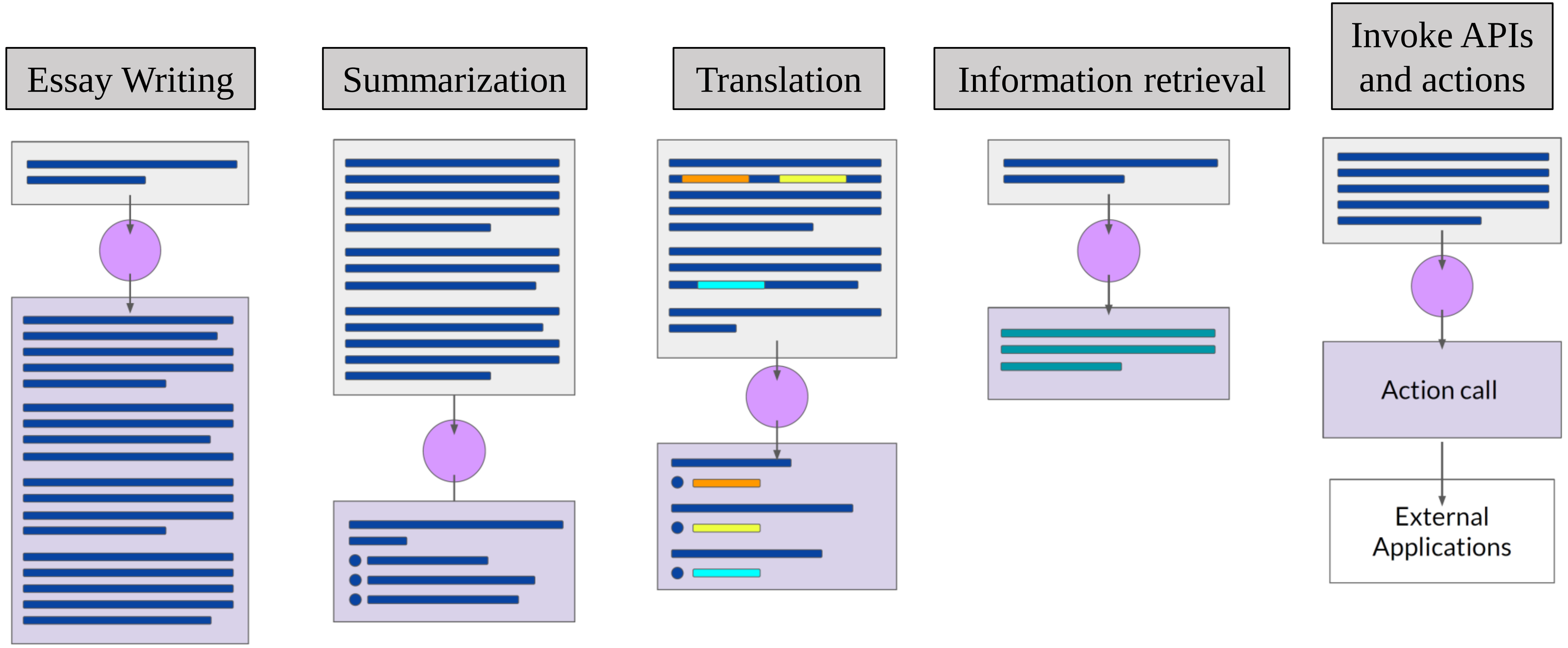
- **Context window** is typically a few thousand words.

- For example, OpenAI's GPT-4 Turbo model has a 128K context window. A 4x boost to the previous maximum of 32K in GPT-4.

✓ How to summarize a book?

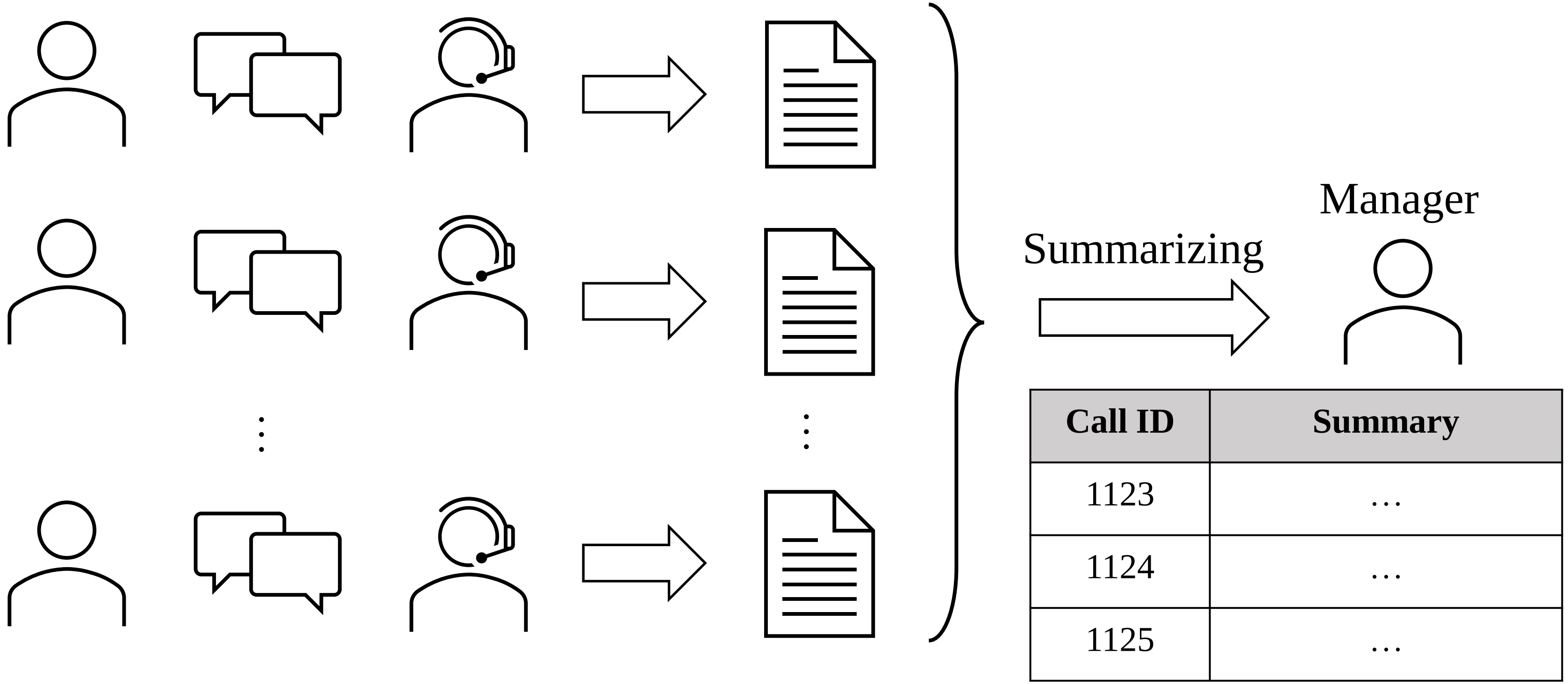


LLMs use cases and tasks



LLM use case; Call center

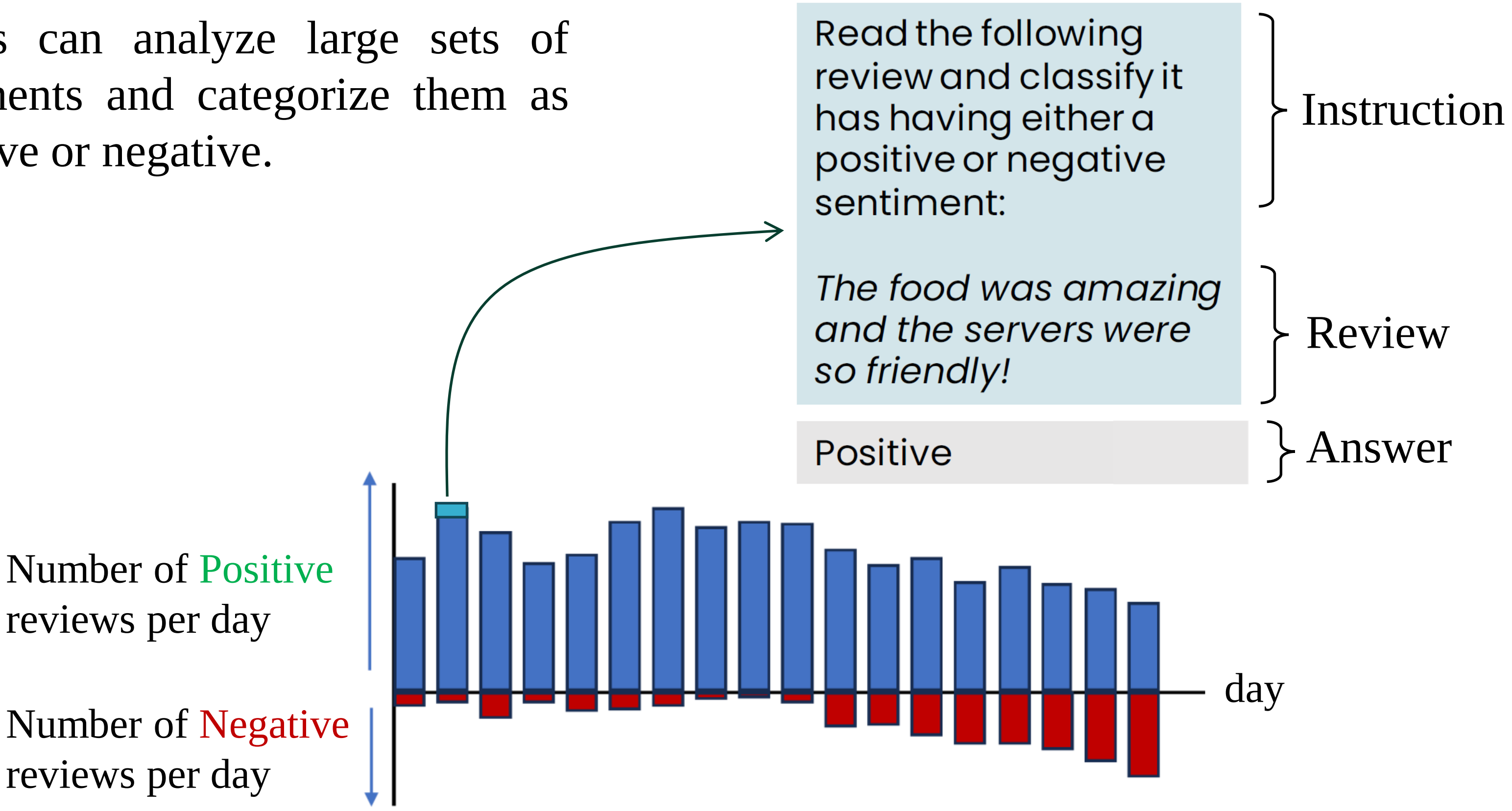
- Summarizing call center conversations:



Call ID	Summary
1123	…
1124	…
1125	…

LLM use case; Reputation monitoring

- LLMs can analyze large sets of comments and categorize them as positive or negative.


Read the following review and classify it has having either a positive or negative sentiment:

The food was amazing and the servers were so friendly!

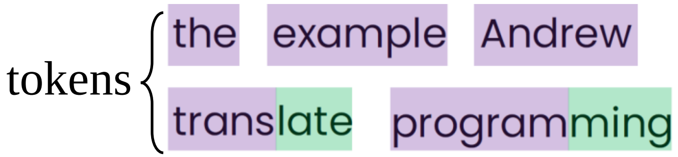
Positive

Instruction
Review
Answer
Number of Positive reviews per day
Number of Negative reviews per day
day

LLM Cost estimation

- Suppose we want to evaluate how much an **LLM-based chatbot** can save on our expenses.

  - Typical adult reading speed: 250 words/minute.
  - Roughly, 1 token = 3/4 words



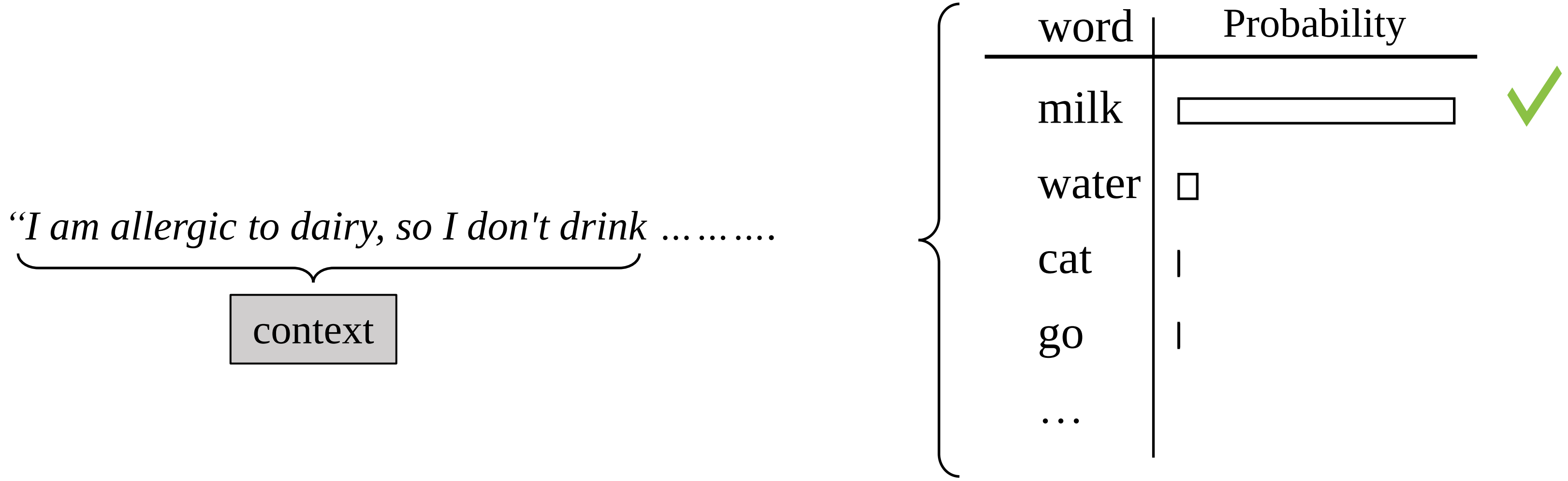
✓ How much would it cost to keep someone occupied for 1 hour?

OpenAI's Pricing

Model	**Input**	**Output**
gpt-3.5-turbo	$0.50 / 1M tokens	$1.50 / 1M tokens
gpt-4o	$5.00 / 1M tokens	$15.00 / 1M tokens

LLMs; Generating text

- The **text generation** process involves predicting the most likely next word given the context of the words that have come before it.



✓ How can we enable the model to learn these?

LLMs; training

- **LLMs** are built by using **supervised learning** to repeatedly predict the next word.

A piece of text

“My favorite food is a bagel with cream cheese”

Training data

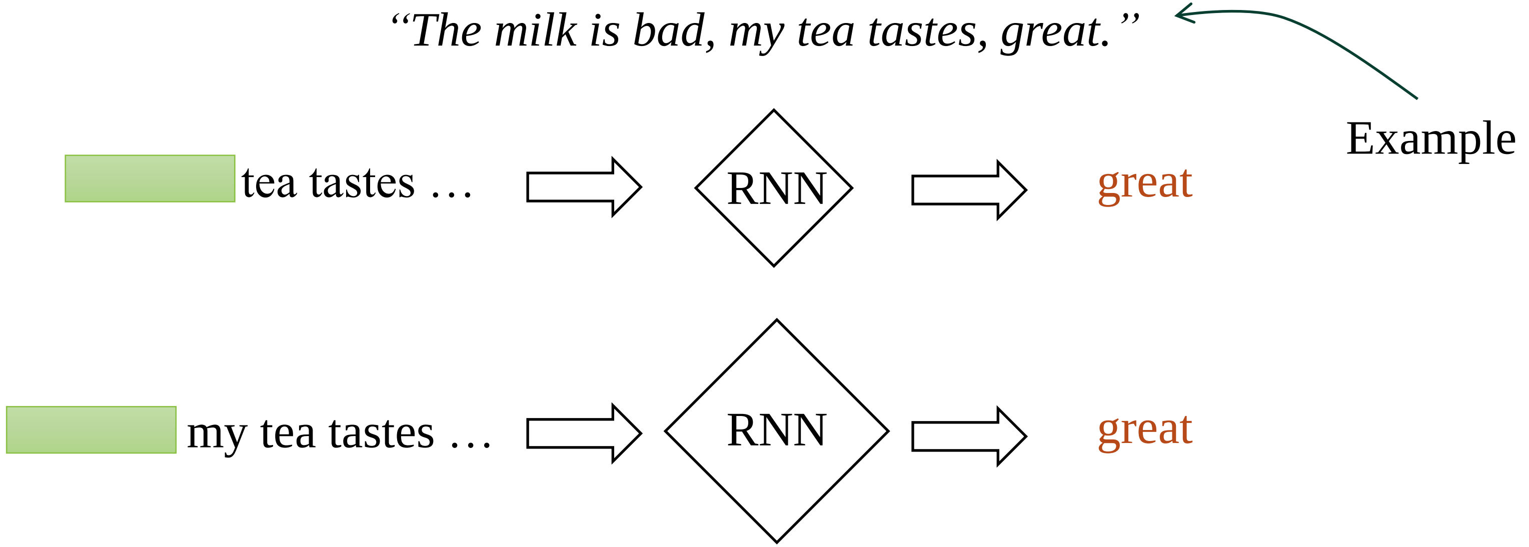
Input	Output
My favorite food is a	bagel
My favorite food is a bagel	with
My favorite food is a bagel with	cream
My favorite food is a bagel with cream	cheese

- **LLMs** are trained on massive amounts of data. For example, GPT-4 is reportedly trained on roughly 13 trillion tokens, which is roughly equivalent to 10 trillion words.

Generating text with RNNs

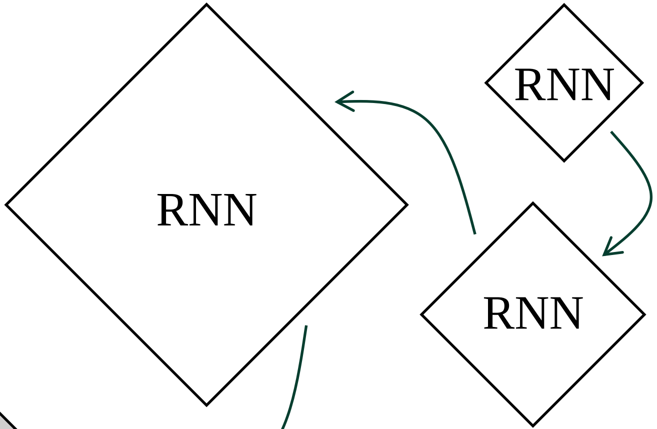
- One of the key challenges **RNNs** face in NLP tasks is capturing **long-term** dependencies in sequential data.

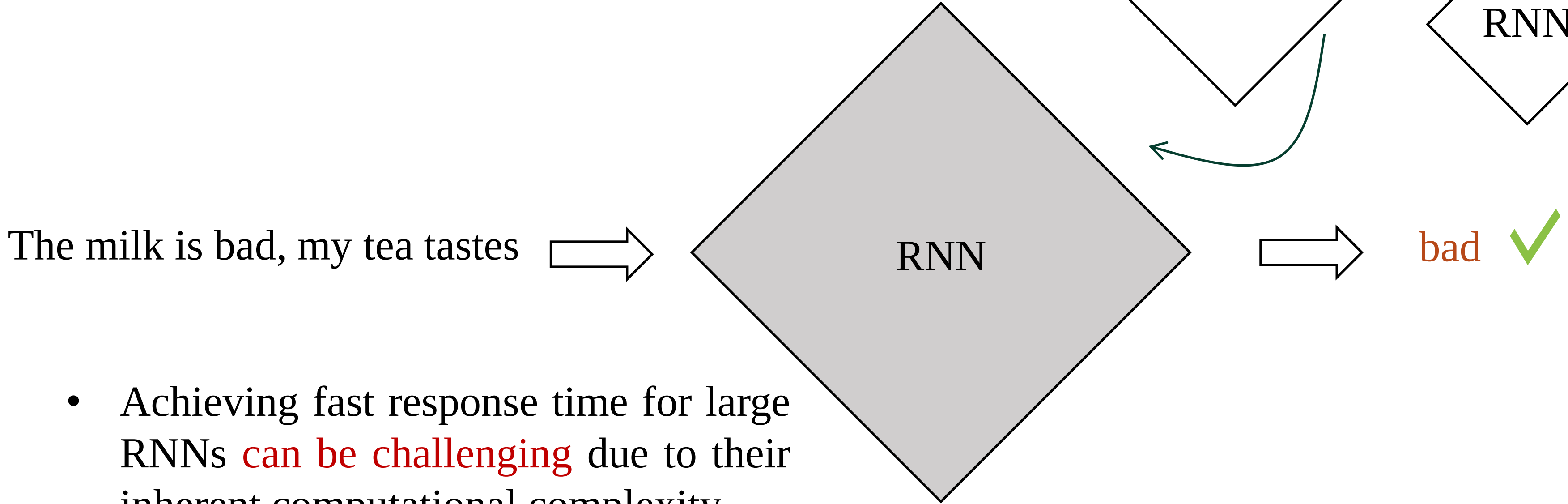
"The milk is bad, my tea tastes, great."



Generating text with RNNs

- We must construct very large RNN models to capture long-term dependencies in text.





- Achieving fast response time for large RNNs can be challenging due to their inherent computational complexity.

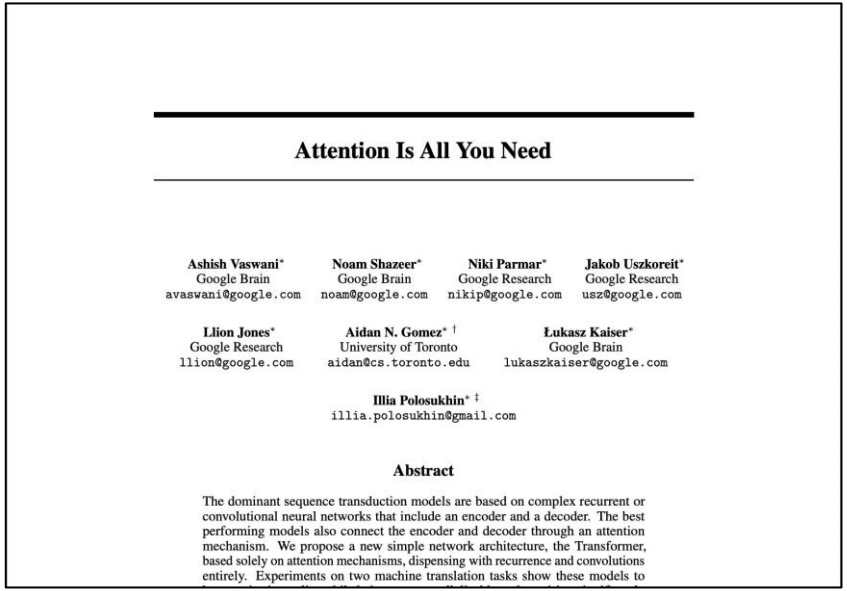
✓ What is the problem?

The significance of scale

- By integrating **scaling laws** into language models, AI systems can exhibit a deeper comprehension of linguistic context and exhibit the ability to generate coherent and contextually relevant language output.

- In **2017**, Google reported on a new type of neural network architecture that brought significant improvements in efficiency and accuracy to tasks like NLP.

- The breakthrough approach, called **transformers**, was based on the concept of attention.



Attention Is All You Need

Ashish Vaswani[*] Google Brain avaswani@google.com

Noam Shazeer[*] Google Brain noam@google.com

Niki Parmar[*] Google Research nikip@google.com

Jakob Uszkoreit[*] Google Research usz@google.com

Llion Jones[*] Google Research llion@google.com

Aidan N. Gomez[*] [†] University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser[*] Google Brain lukaszkaiser@google.com

Illia Polosukhin[*] [‡] illia.polosukhin@gmail.com

Abstract

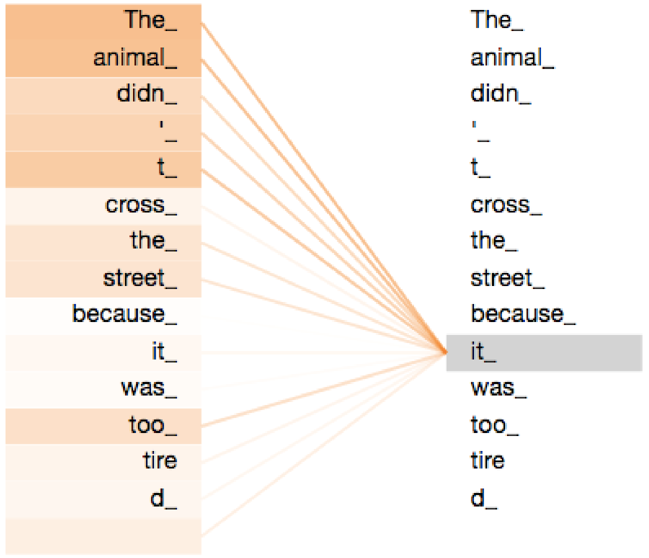
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to

Attention is all you need

- The researchers showed how a **transformer neural network** was able to translate between English and French with more accuracy and in only a quarter of the training time than other neural nets.

- Main idea of attention => we focus everywhere, just to **different extents**.

- Example:



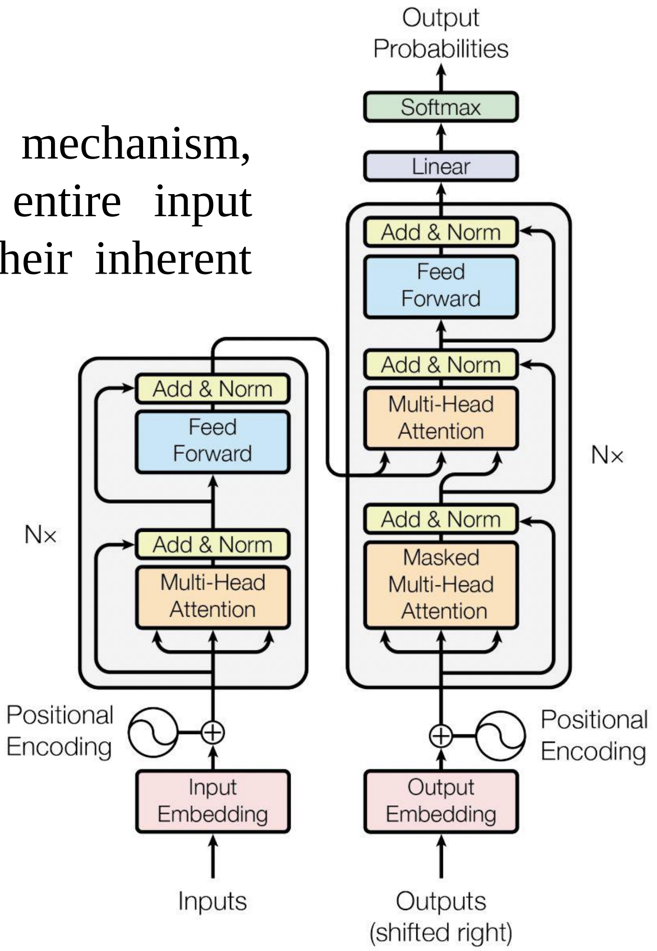
- **Transformer architecture** has evolved rapidly since it was introduced, giving rise to LLMs such as GPT-3 and better pre-training techniques, such as Google's BERT.

Transformer

- **Transformer models**, powered by the Attention mechanism, efficiently parallelize computations across the entire input sequence, unlike **RNNs** which struggle due to their inherent sequential processing nature.

- **Attention mechanism** allows the model to dynamically focus on the most relevant parts of the input when generating an output.

- Transformers:
  - ✓ Scale efficiently
  - ✓ Parallel process



Attention Mechanism

- Attention is a method for taking a query, and softly looking up information in a key-value store by picking the value(s) of the key(s) most like the query.

- By “picking” and “most like,” we mean averaging overall values, putting more weight on those which correspond to the keys more like the query.

- The "key-value-query" concept is fundamental to the way attention mechanisms work.

- Imagine you're reading a long document. You don't read every word equally; you focus on the parts most relevant to your current goal. Attention mechanisms in machine learning do the same thing with sequences of data (like words in a sentence).

Key-query-value

- Consider a token x_i in the sequence $x_{1:n}$. For this token we define:

A query	$q_i = Qx_i$
A key	$k_i = kx_i$
A value	$v_i = Vx_i$

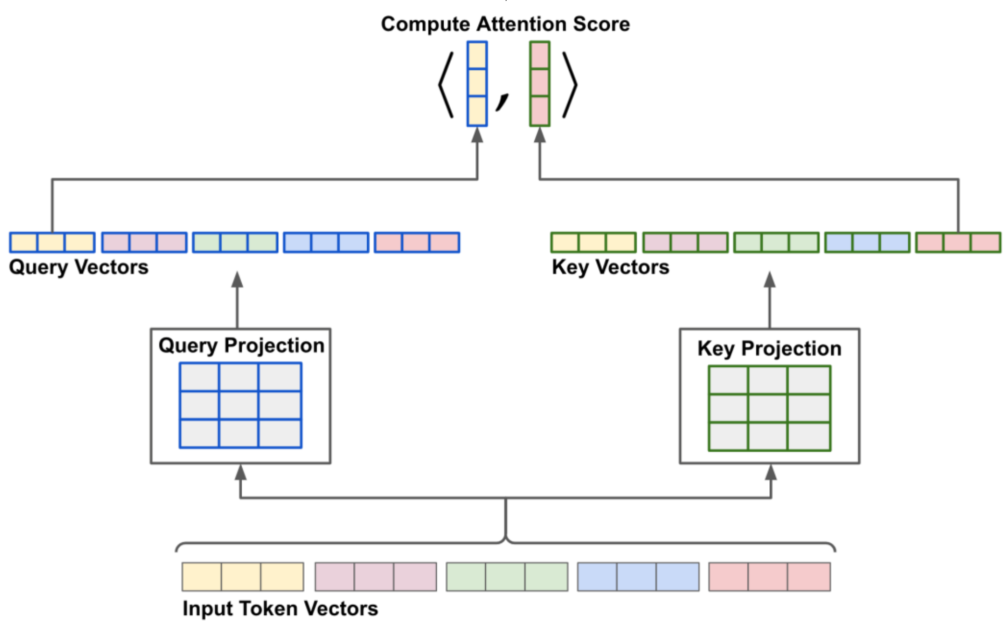
- Then we define the same things for each of the token in the sequence.

- Now based on the definitions in the previous slide we can define some weights. We can consider these weights as a measure of similarity or the value of attending to other tokens by each token.

Key-query-value

- Here you can see the formula for calculating these weights:

$$\alpha_{ij} = \frac{\exp(q_i^\intercal k_j)}{\sum_{j'}^{n} \exp(q_i^\intercal k_{j'})}$$



Key-query-value

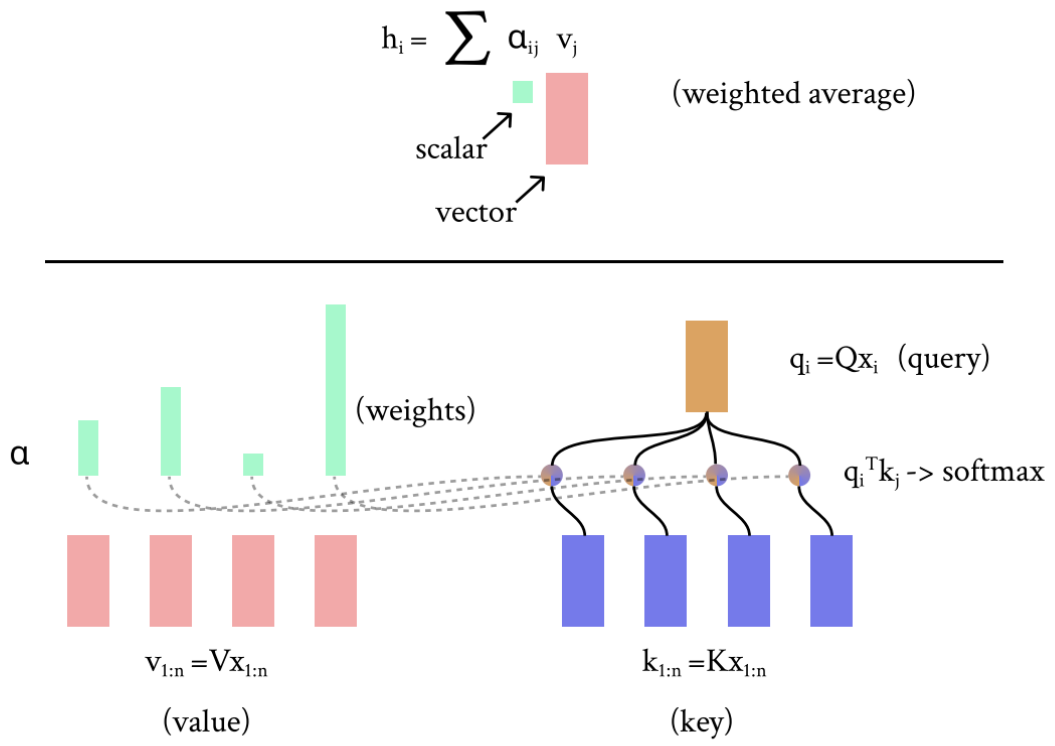
- The formula for calculating weights:

$$\alpha_{ij} = \frac{\exp(q_i^\intercal k_j)}{\sum_{j'}^{n} \exp(q_i^\intercal k_{j'})}$$

- Based on these weight, that are representing how much the i-th word should attend to the j-th word, we can calculate new representations for the i-th token:

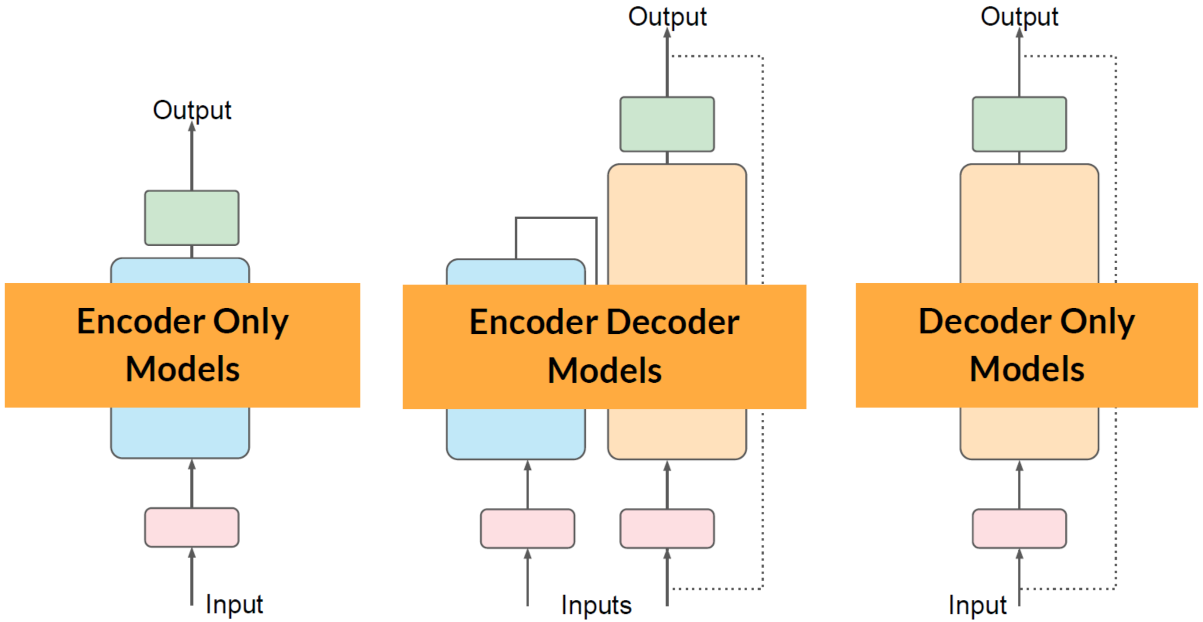
$$h_i = \sum_{j}^{n} \alpha_{ij}\, v_j$$

Key-query-value visualization



Types of Transformers

- Different types of transformers:
  1. Encoder-Decoders(standard architecture)
  2. Encoders only
  3. Decoder only



Encoders and Decoders

- In the context of NLP, encoders and decoders refer to the two main components that work together:

  - Encoders:

    The encoder is responsible for processing the input sequence and generating a numerical representation, often called the "context vector“. The encoder's job is to convert the input sequence into a fixed-size vector that captures the most relevant information and features of the input.

  - Decoders:

    The decoder is responsible for generating the output sequence based on the context vector produced by the encoder. The decoder takes the context vector as input and generates the output sequence token by token, using a language model to predict the most likely next token.

Encoder-Decoders

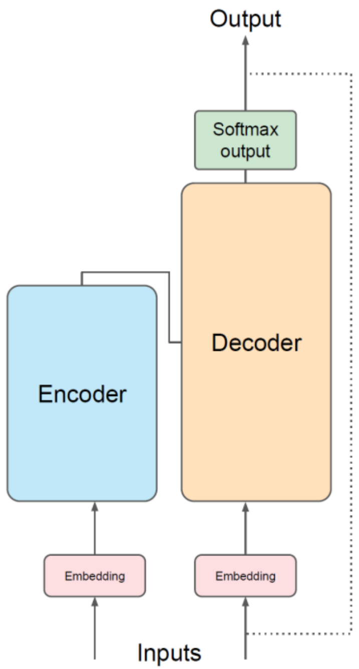
- Encoder-decoder transformers, also known as sequence-to-sequence(seq2seq) transformers, are a type of transformer model that combines both the encoder and decoder components.

  - Encoder:

    The encoder takes the input sequence (e.g., a sentence in one language) and generates a contextual representation of it. This representation captures the semantic and syntactic relationships within the input.

  - Decoder:

    The decoder takes the encoder's output and generates the output sequence (e.g., a translation of the input sentence into another language). The decoder uses the previously generated tokens to predict the next token in the sequence.



Encoder-Decoders

- The encoder-decoder architecture allows the model to understand the input and generate the output, making it suitable for tasks that involve transforming one sequence into another, such as:
  - Machine Translation
  - Summarization
  - Question answering
  - Dialogue generation

- Some popular examples of encoder-decoder transformer models include:
  - The original transformer model proposed in "Attention is All You Need“
  - T5 (Text-to-Text Transfer Transformer)
  - BART (Bidirectional and Auto-Regressive Transformers)

Encoder only

- Encoder-only transformers, as the name suggests, only use the encoder component and do not have a decoder. They are commonly used for tasks that do not require generating an output sequence, such as:
  - Text Classification
  - Named entity recognition
  - Extractive QA

- Some popular examples of encoder-only transformers include:
  - BERT (Bidirectional Encoder Representations from Transformers)
  - RoBERTa (Robustly Optimized BERT Pretraining Approach)
  - DistilBERT

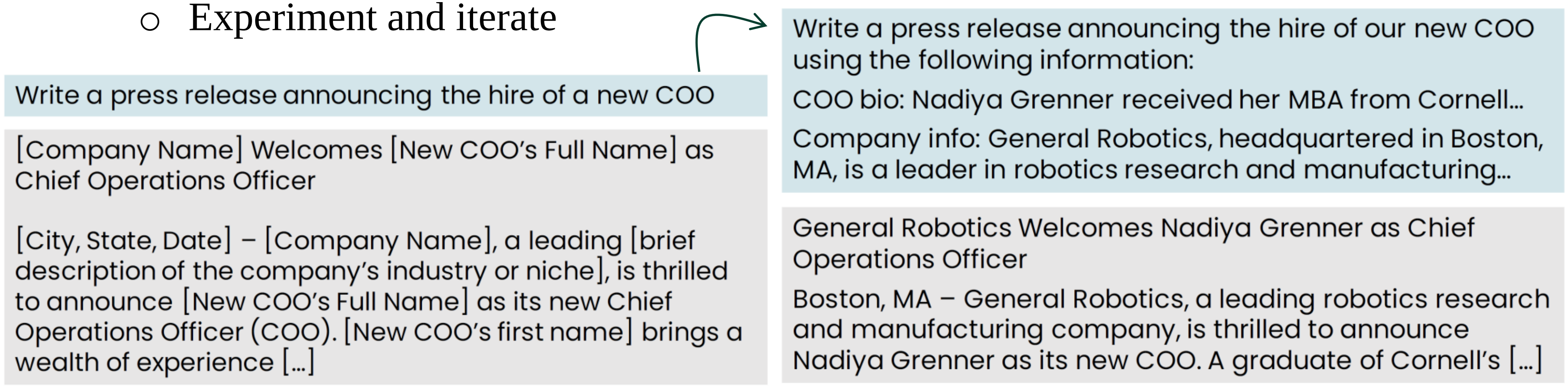
Decoder only

- Decoder-only transformers, on the other hand, only use the decoder component and do not have an encoder. They are commonly used for tasks that require generating output sequences, such as:
    - Text Generation
    - Machine Translation
    - Dialogue Generation

- Some popular examples of decoder-only transformers include:
    - GPT(Generative Pre-training Transformer)
    - Transformer XL

Prompting

- Tips:
  - Be detailed and specific
  - Guide the model to think through its answer
  - Experiment and iterate



Write a press release announcing the hire of a new COO

[Company Name] Welcomes [New COO's Full Name] as Chief Operations Officer

[City, State, Date] – [Company Name], a leading [brief description of the company's industry or niche], is thrilled to announce [New COO's Full Name] as its new Chief Operations Officer (COO). [New COO's first name] brings a wealth of experience [...]

Write a press release announcing the hire of our new COO using the following information:

COO bio: Nadiya Grenner received her MBA from Cornell...

Company info: General Robotics, headquartered in Boston, MA, is a leader in robotics research and manufacturing...

General Robotics Welcomes Nadiya Grenner as Chief Operations Officer

Boston, MA – General Robotics, a leading robotics research and manufacturing company, is thrilled to announce Nadiya Grenner as its new COO. A graduate of Cornell's [...]

- Give sufficient context for LLM to complete the task and describing the desired task in detail can help a lot.

Prompting; Example

- A well written Email routing prompt:

Read the email below and choose the most appropriate department to route the email to.

Choose the department from the following list: Apparel, Electronics, Home appliances.

I love my new llama t-shirt! The fabric is so soft.

Department: Apparel

State the task you want the model to carry out

Provide the choices for the model

Instruction

Include the email to analyze

The model responds with an answer from the list

Prompt Engineering

- **Prompt engineering** is a relatively new discipline for developing and optimizing prompts to efficiently use language models (LMs) for a wide variety of applications.

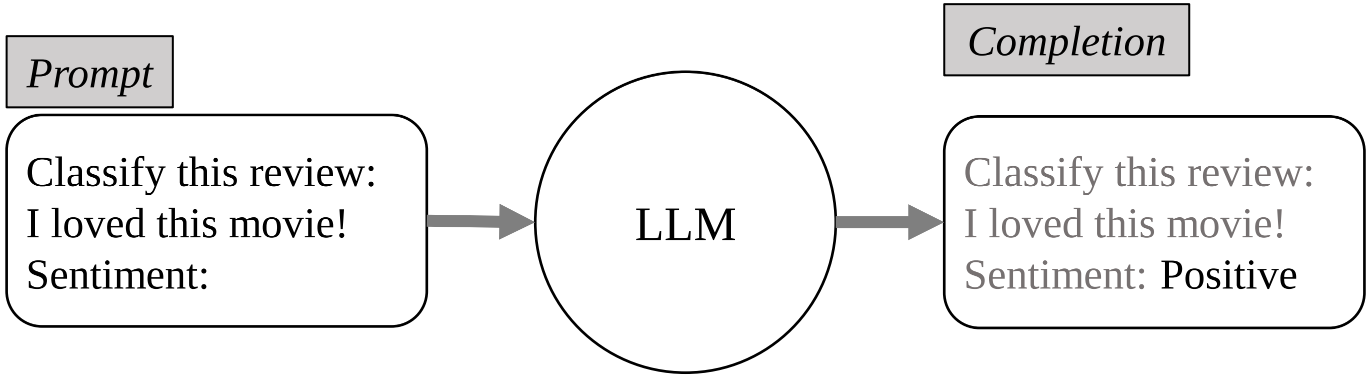
- There are many prompting techniques available. You can learn more about them on this link.
    - Few shots
    - Chain-of-Thought
    - Self consistency
    - RAG
    - ReAct
    - ...

- You can also read this to familiarize yourself with prompting techniques for OpenAI's GPT models.

In-context learning

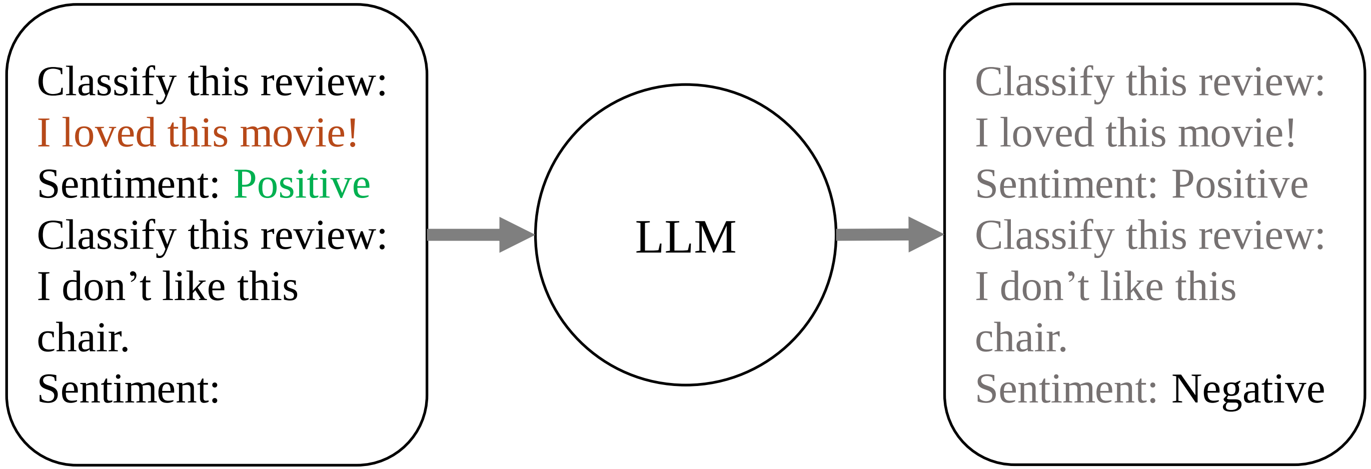
- **In-context learning (ICL)** is a method of prompt engineering where the model is shown task demonstrations as part of the prompt in natural language.

- Zero-shot inference example:



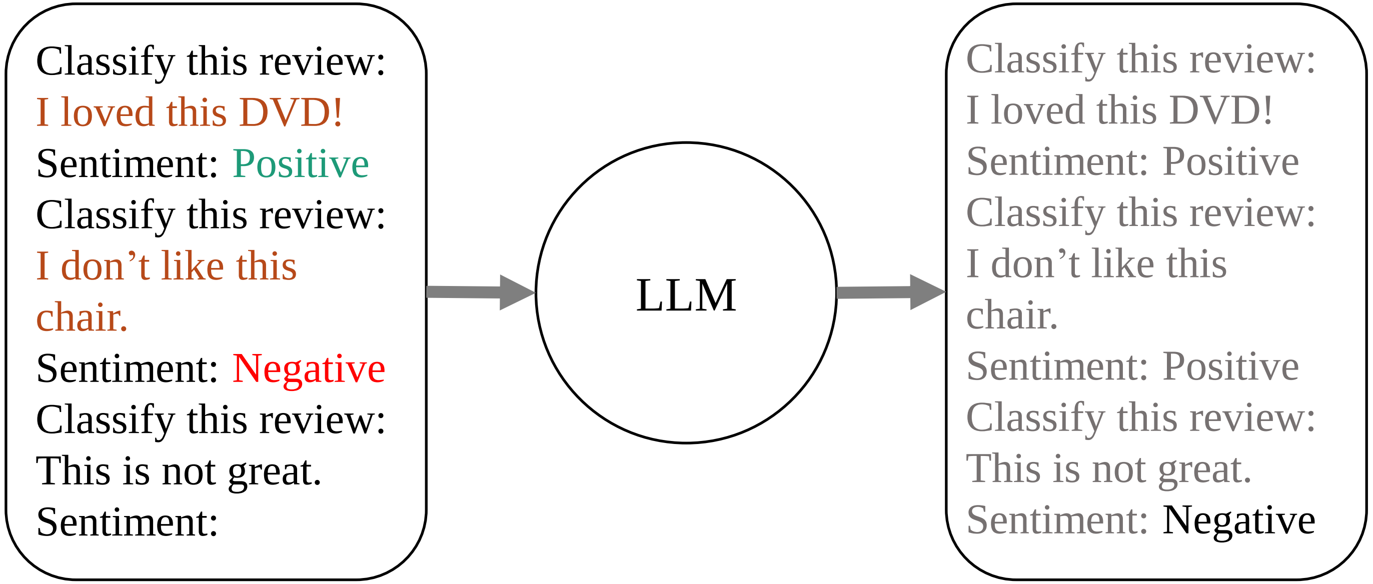
ICL; One shot inference

- One-shot inference example:



ICL; Few shot inference

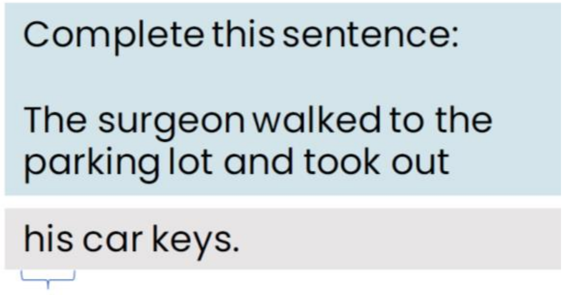
- 5 or 6 examples



Bias in LLMs

- An **LLM** can reflect the biases that exist in the text it learned from.

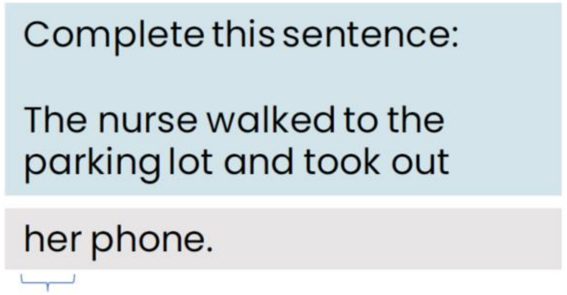
- For example:



Complete this sentence:

The surgeon walked to the parking lot and took out

his car keys.



Complete this sentence:

The nurse walked to the parking lot and took out

her phone.

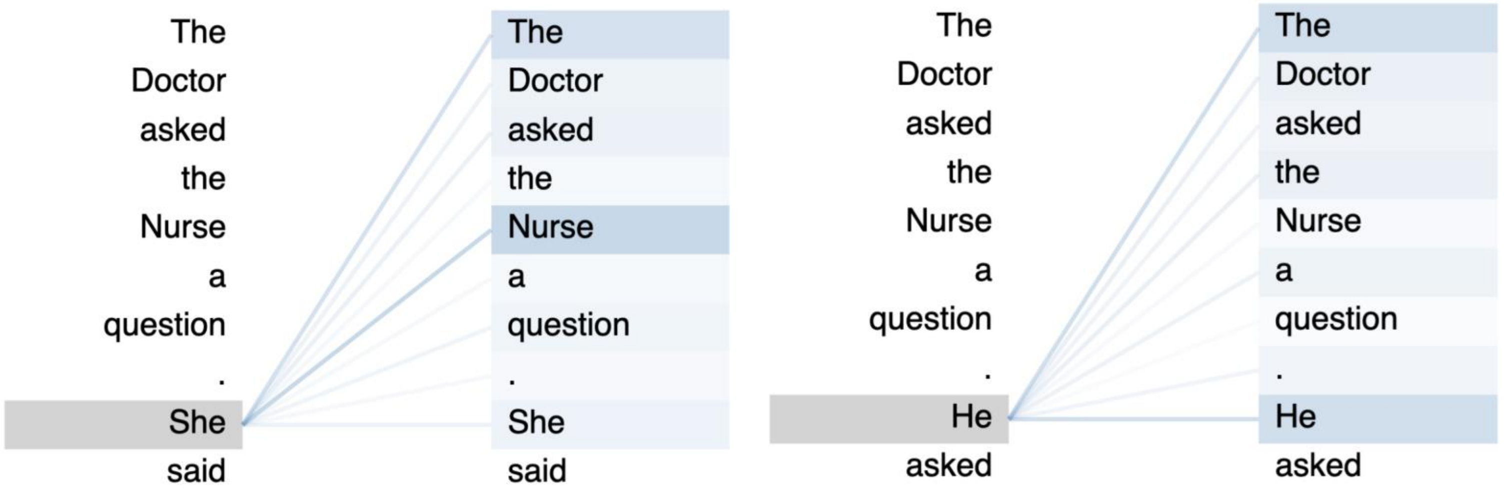
Assumed male

Assumed female

- LLM bias include gender, race, and cultural bias.

Bias in LLMs

- Attention visualization of an example reflecting bias:


The
Doctor
asked
the
Nurse
a
question
.
She
said
The
Doctor
asked
the
Nurse
a
question
.
She
said
The
Doctor
asked
the
Nurse
a
question
.
He
asked
The
Doctor
asked
the
Nurse
a
question
.
He
asked