COMPSC 762
Advanced Neural Networks

Neural Networks IV

Instructor : Thomas Lacombe

Week 11

Outline

Introduction

Artificial Neural Networks (ANN)

- Single Unit: Architecture of Perceptron (NN1)
- Connection to Shallow Machine Learning (NN1)
- Multi-Layer Feed-Forward Neural Network (NN2)

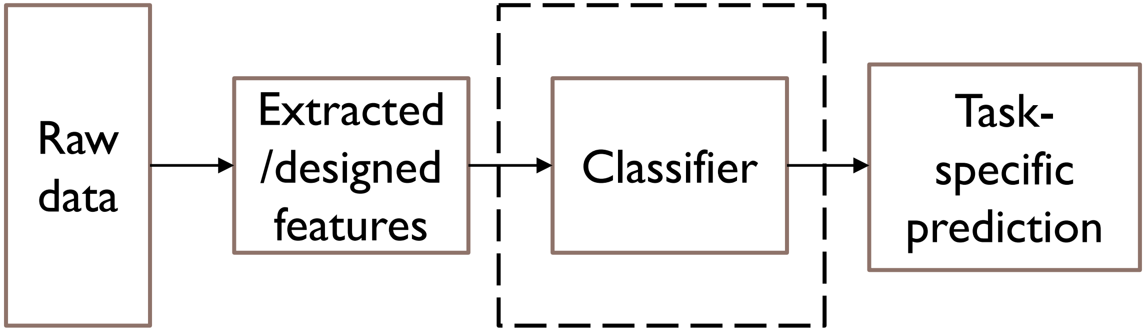
Design Issues (NN3)

Deep Learning / Large Language Models (NN4)

Evolution of Machine Learning

- Evolution of ML paradigms:

1. Feature design/extraction (expert knowledge) + task specific



Decision tree,
Probabilistic classifier,
kNN,
…

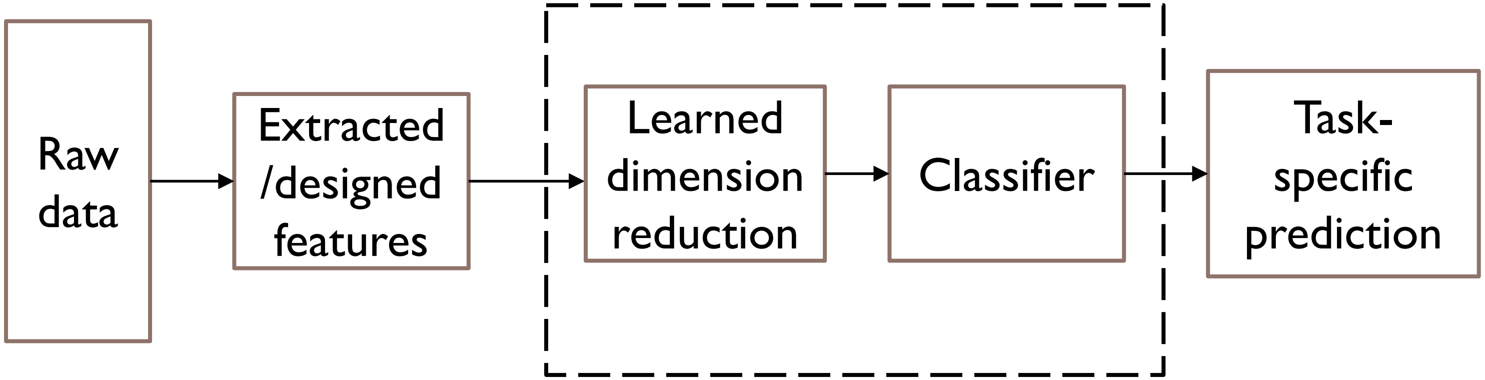
E.g.,

- predicting house price from house features,
- classifying images from extracted features from image processing techniques, …

Evolution of Machine Learning

- Evolution of ML paradigms:

1. Feature design/extraction (expert knowledge) + task specific
2. Learned dimensionality reduction + task specific

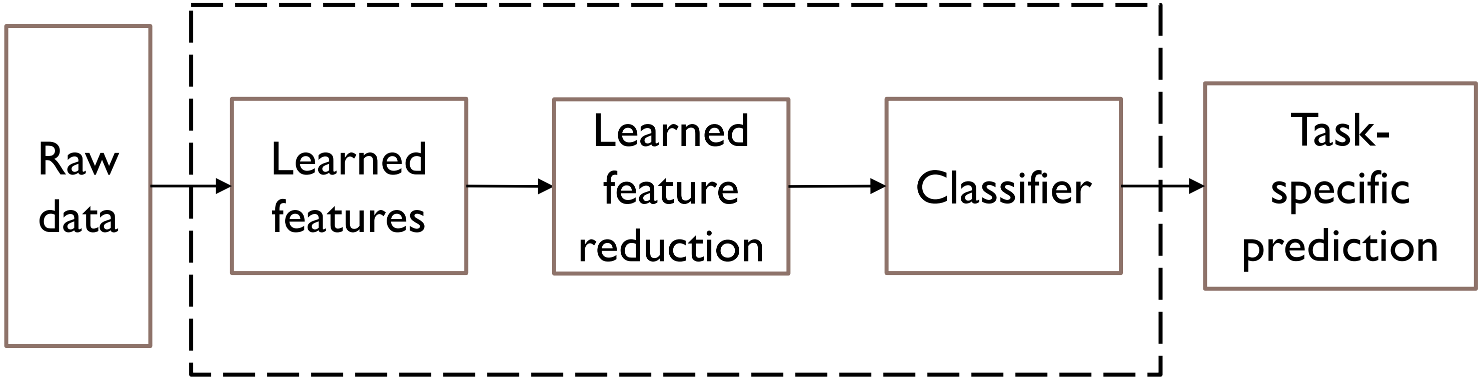


- Perceptron
- SVM
- Random forest
- Small MLP

Evolution of Machine Learning

- Evolution of ML paradigms:

1. Feature design/extraction (expert knowledge) + task specific
2. Learned feature reduction + task specific
3. Learned features + task specific

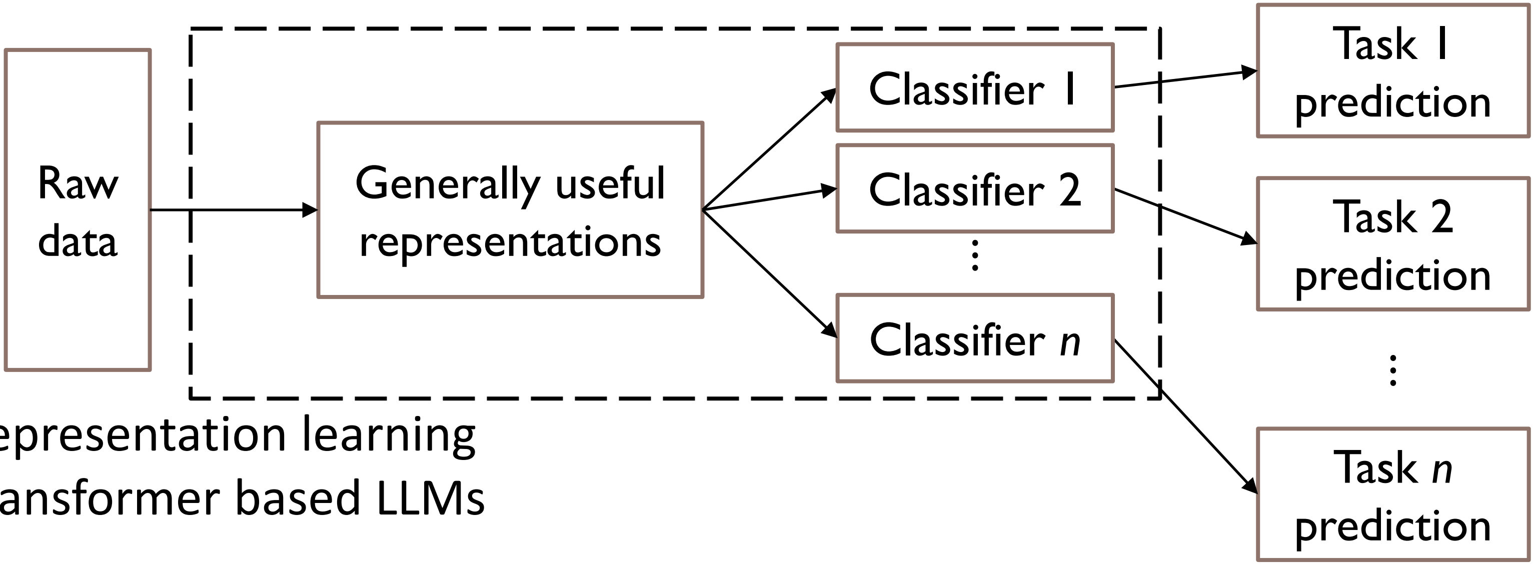


- ➢ Deep neural networks
- ➢ Auto-encoders
- ➢ Attention mechanism
- ➢ Transformers

Evolution of Machine Learning

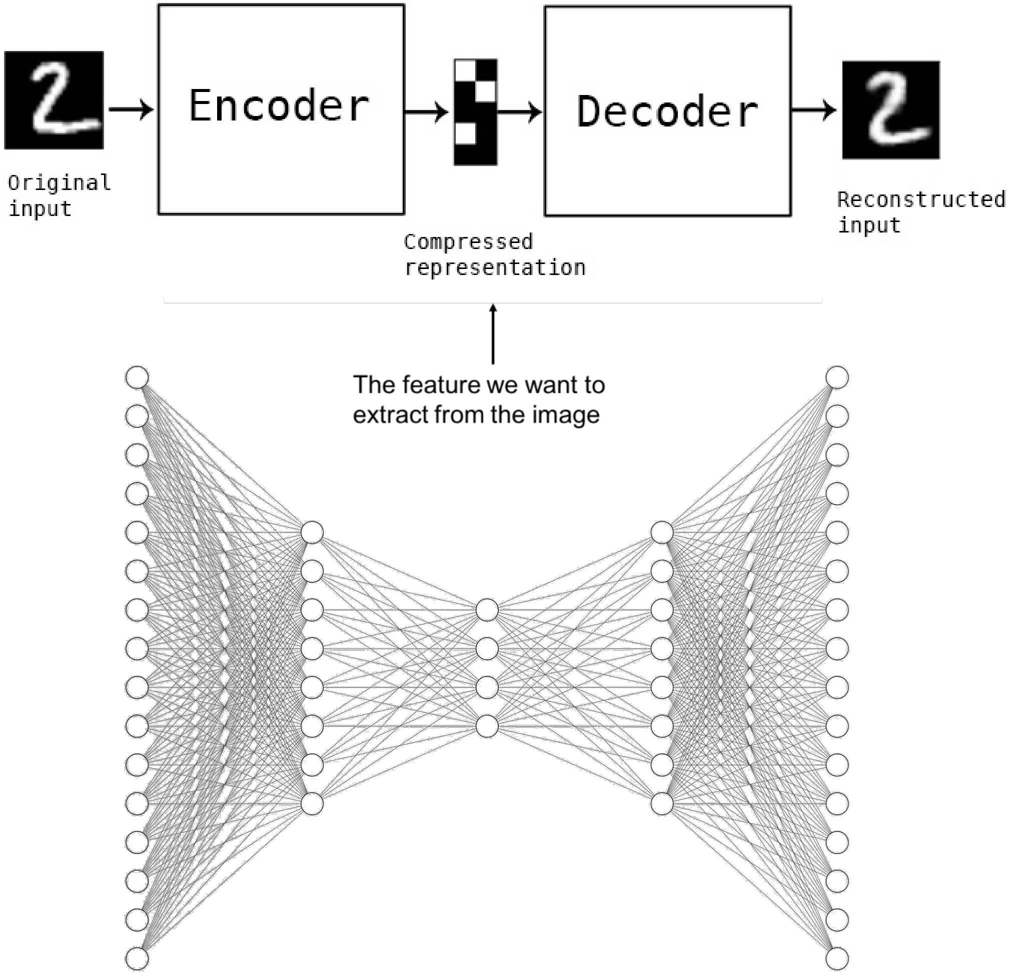
- Evolution of ML paradigms:

1. Feature design/extraction (expert knowledge) + task specific
2. Learned feature reduction + task specific
3. Learned features + task specific
4. Generally useful learned representations + multi tasks



- Representation learning
- Transformer based LLMs
- …

Auto-encoders



The feature we want to extract from the image

Auto-encoder

An auto-encoder is a neural network composed of two "sub-networks": the encoder and the decoder. The encoder learns to compress the input data into a low-dimensional representation (latent representation). The decoder learns to decompress the latent representation back to the original data.

The encoder learns relevant dimentionality reduction and produces efficient representations of the data.

The decoder learn to retrieve the orginal meaning of the data from the compressed representation.

Attention mechanism

Attention allows the model to weigh the importance of different parts of the input when making predictions. Instead of processing input features independently, it at includes correlations between the inputs in its learning.

E.g., in natural languages, the semantic/meaning of a word is closely related to its context. I.e., words that appear frequently together are likely to be semantically related.

Attention allows the model to “focus” differently on inputs in a sequence depending on their relevance to the task at hand.

Word embeddings

Individual words are represented as vector of numerical values in a lower dimension space. Such representations are called word embeddings.

Word embeddings aim at capturing the meaning of words and their relationship to other words (semantic and syntax).

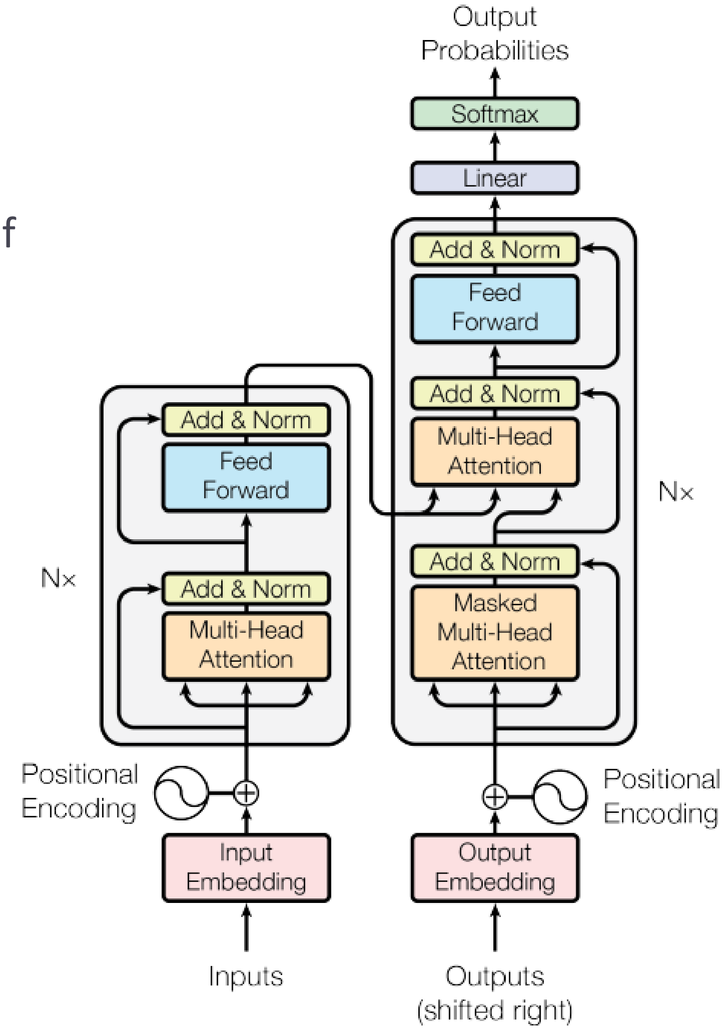
- Bag of words (BOW) are one of the simplest word embedding, but they can be intensive to compute and they fail to capture the relationship between words (i.e., do not consider the order).
- Modern word embeddings are learned through ML and take in considering the local or global context of the words. A few popular embeddings: Word2Vec, GloVe, FastText and ElMo.

Transformer – Full architecture

- Seq2Seq architecture (encoder/decoder)
  - Encoder: takes an input sequence and produces a set of hidden representations, also known as context vector.
  - Decoder: takes the context vector and generates the output sequence.

- Leverages the **attention mechanism** to retain information about which parts of the sequence are important to make the prediction.

- Autoregressive inferences/predictions (decoder): predict the next element/token in a sequence.



[Attention Is All You Need, Vaswani 2017]

Transformer – Full architecture

Several encoder and decoder blocks are stacked.

- In original Transformer paper: $N_x = 6$

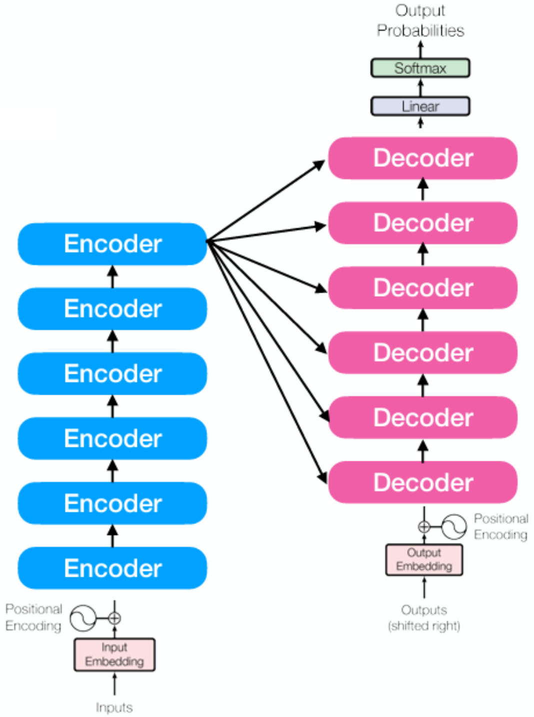
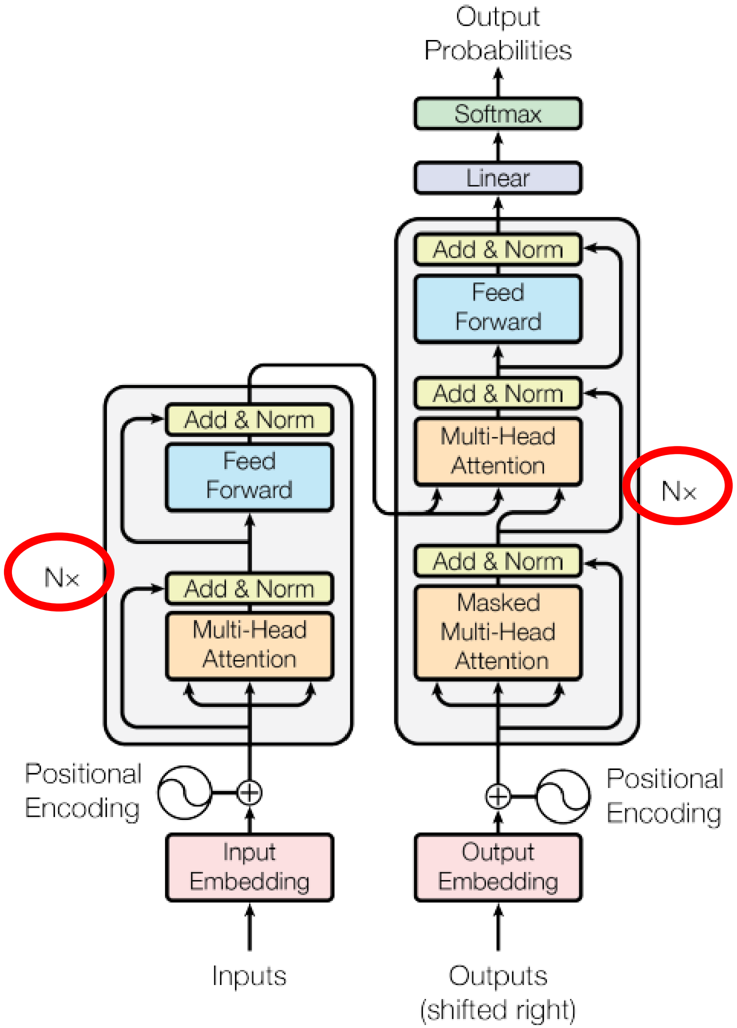


Image source: Sebastian Raschka, STAT 453: Intro to Deep Learning



[Attention Is All You Need, Vaswani 2017]

Transformer – Full architecture

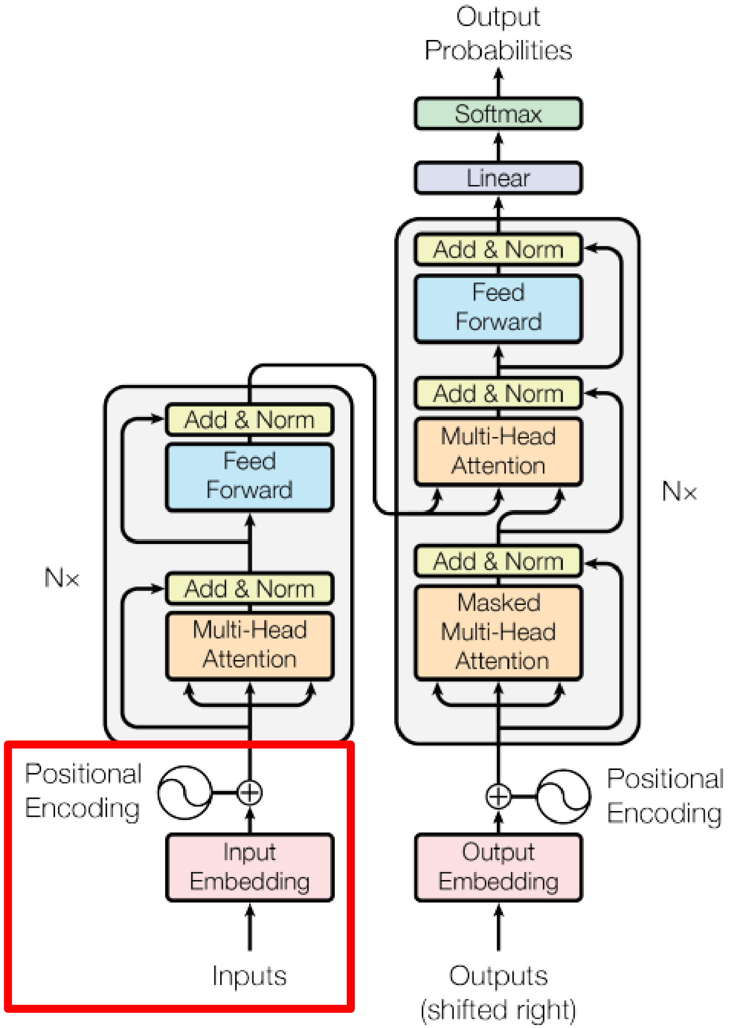
Input/output embedding:

- Original embeddings size: $d_{model} = 512$
- Embeddings are learned during the training process.

Positional encoding:

- Transformers do not process the data sequentially.
- By design, they do not have access to the position of the tokens in a sequence.
- Positional encoding is used to retrieve the order information.

Full explanation with example for positional embedding (Hedu AI - Youtube)

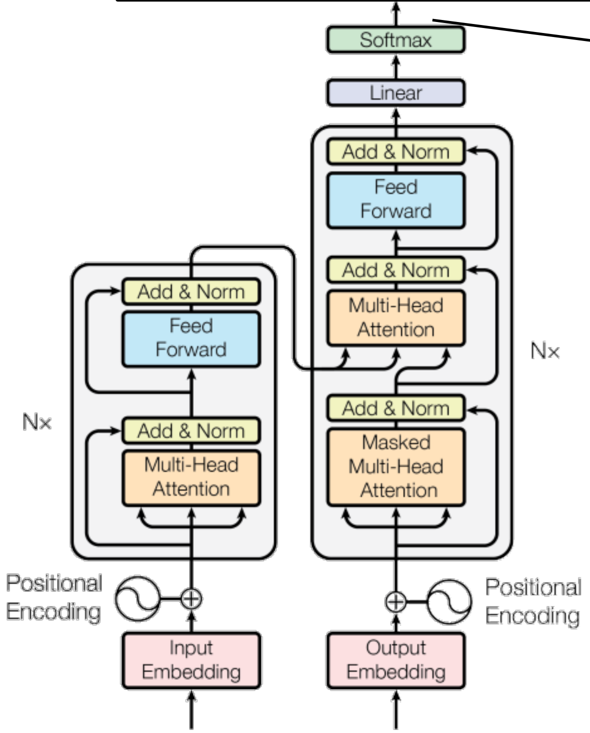


[Attention Is All You Need, Vaswani 2017]

Transformer inference – Translation

Predicted word = highest softmax probability.

Trees



Output = probabilities of each word of the vocabulary to be predicted.

Pr(Trees) = 0.8

Pr(The) = 0.2

Les arbres sont grands

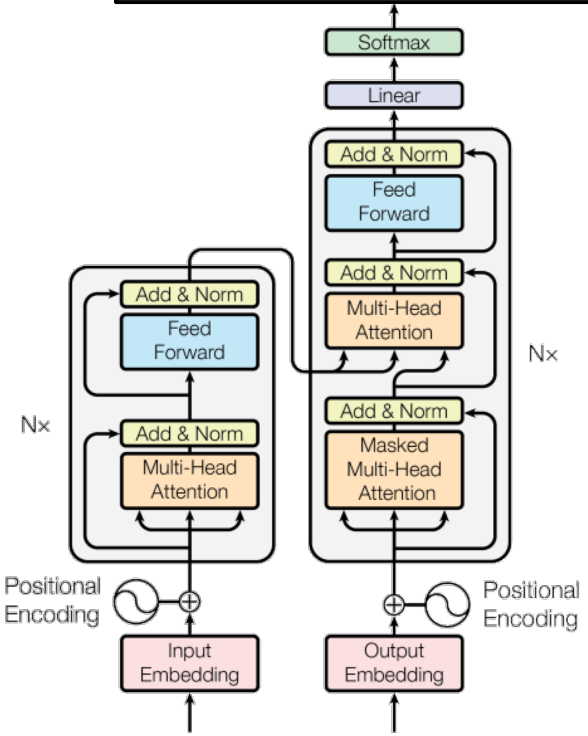
input

SOS

Transformer inference – Translation

Trees **are**

Pr(are) = 0.95
Pr(is) = 0.05



Les arbres sont grands

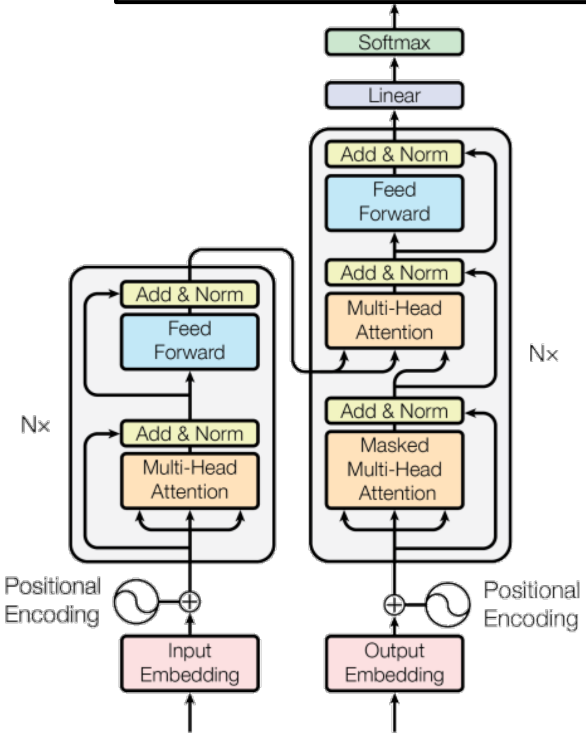
SOS Trees

input

Transformer inference – Translation

Trees are **tall**

Pr(tall) = 0.65
Pr(big) = 0.25
Pr(huge) = 0.1



Les arbres sont grands

input

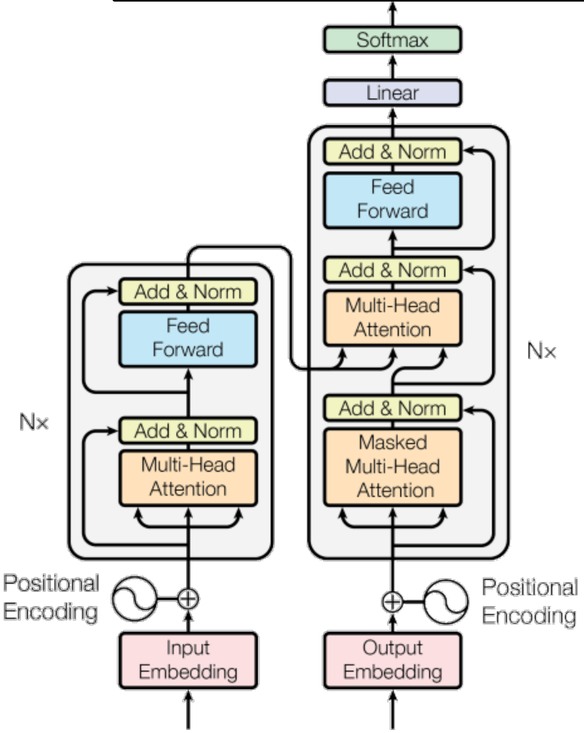
SOS Trees are

Transformer inference– Translation

Trees are tall **ESO**

Pr(EOS) = 0.99

…



Les arbres sont grands

input

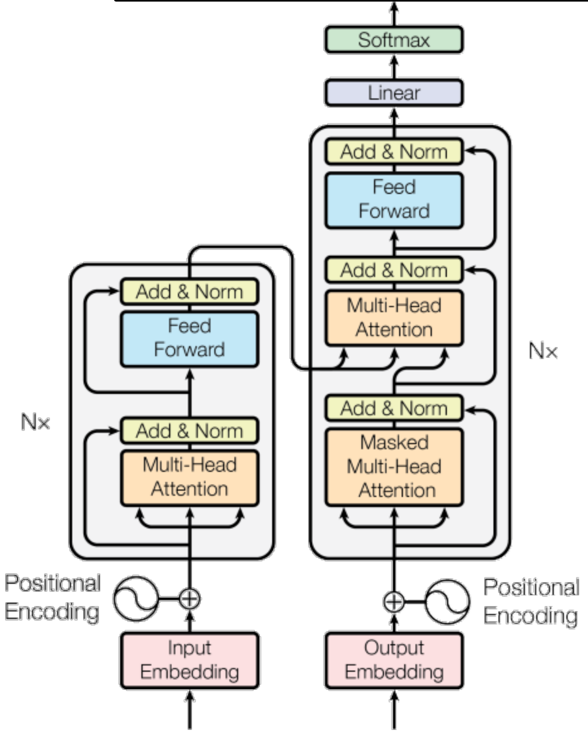
SOS Trees are tall

Transformer inference– Translation

Trees are tall **ESO**

Pr(EOS) = 0.99

…

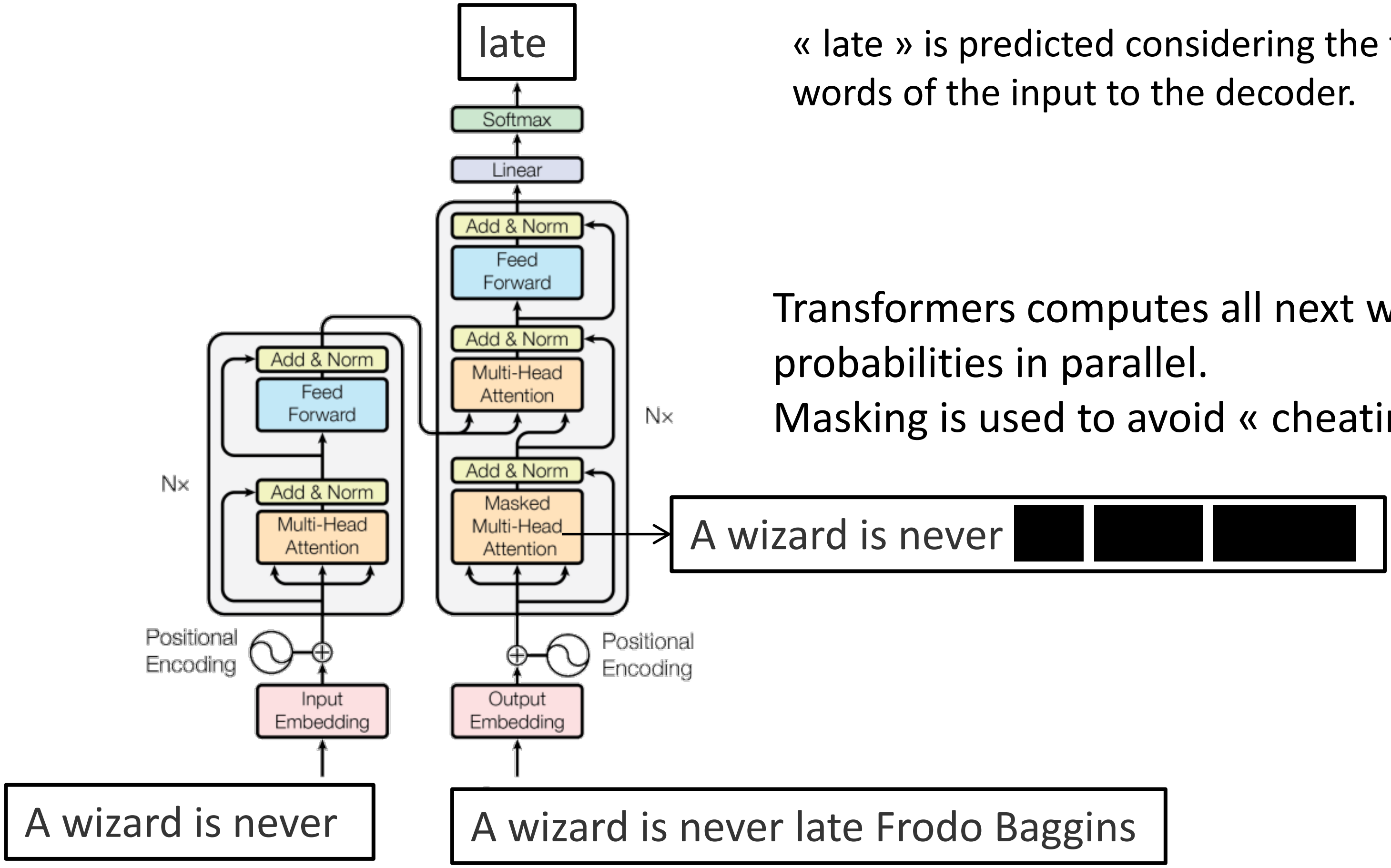


Les arbres sont grands

input

SOS Trees are tall EOS

Transformer training – Next word prediction



A wizard is never [redacted]

A wizard is never

input

A wizard is never late Frodo Baggins

whole output (**only when training**)

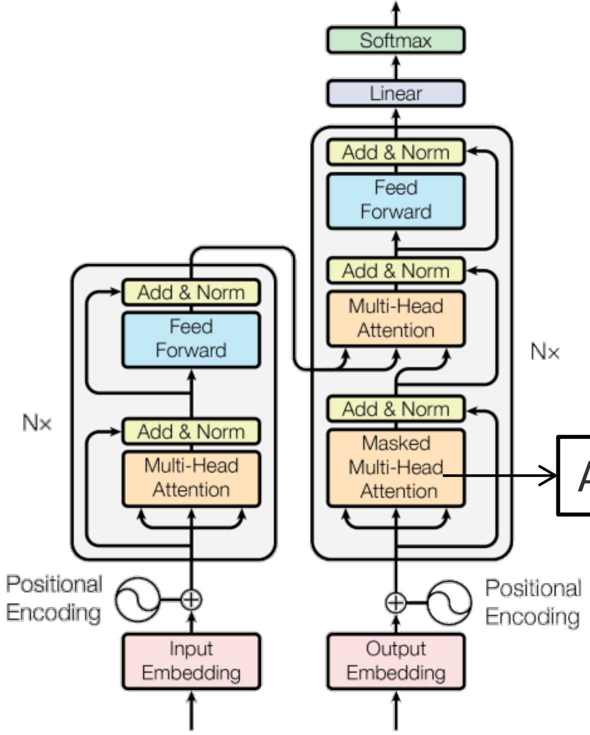
« late » is predicted considering the first 4 words of the input to the decoder.

Transformers computes all next words probabilities in parallel.
Masking is used to avoid « cheating ».

Transformer – Next word prediction

Frodo

« Frodo » is this time predicted considering the first 5 words of the input to the decoder.



A wizard is never late [redacted] [redacted]

Transformers computes all next words probabilities in parallel.
Masking is used to avoid « cheating ».

A wizard is never

input

A wizard is never late Frodo Baggins

whole output (**only when training**)

Large language models

A **large language model** (LLM) is a general purpose language model consisting of a neural network with many parameters (typically billions of weights or more). LLMs trained on large quantities of unlabelled text perform well at a wide variety of tasks, a development which, since their emergence around 2018, has shifted the focus of natural language processing research away from the previous paradigm of training specialized supervised models for specific tasks.

Wikipedia

A **large language model** is an artificial neural network designed to analyze and generate natural language data. It is trained on vast amounts of text data and can perform various language tasks such as translation, summarization, and sentiment analysis. Large language models have revolutionized natural language processing, allowing machines to understand and generate human-like language with high accuracy.

ChatGPT

LLMs - A parameter story

GPT-4
170 trillions

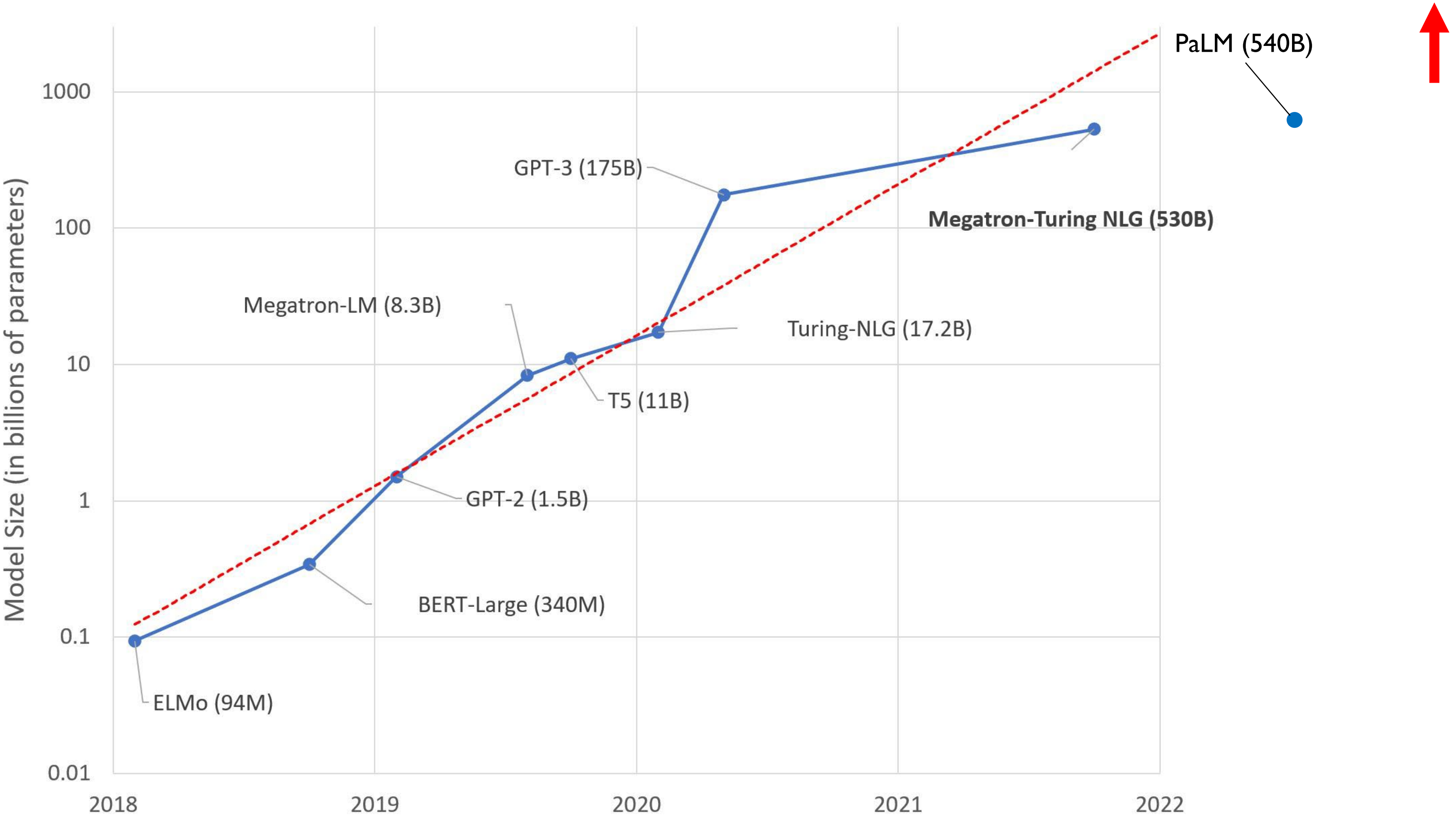


Image source: huggingface.co/blog/large-language-models

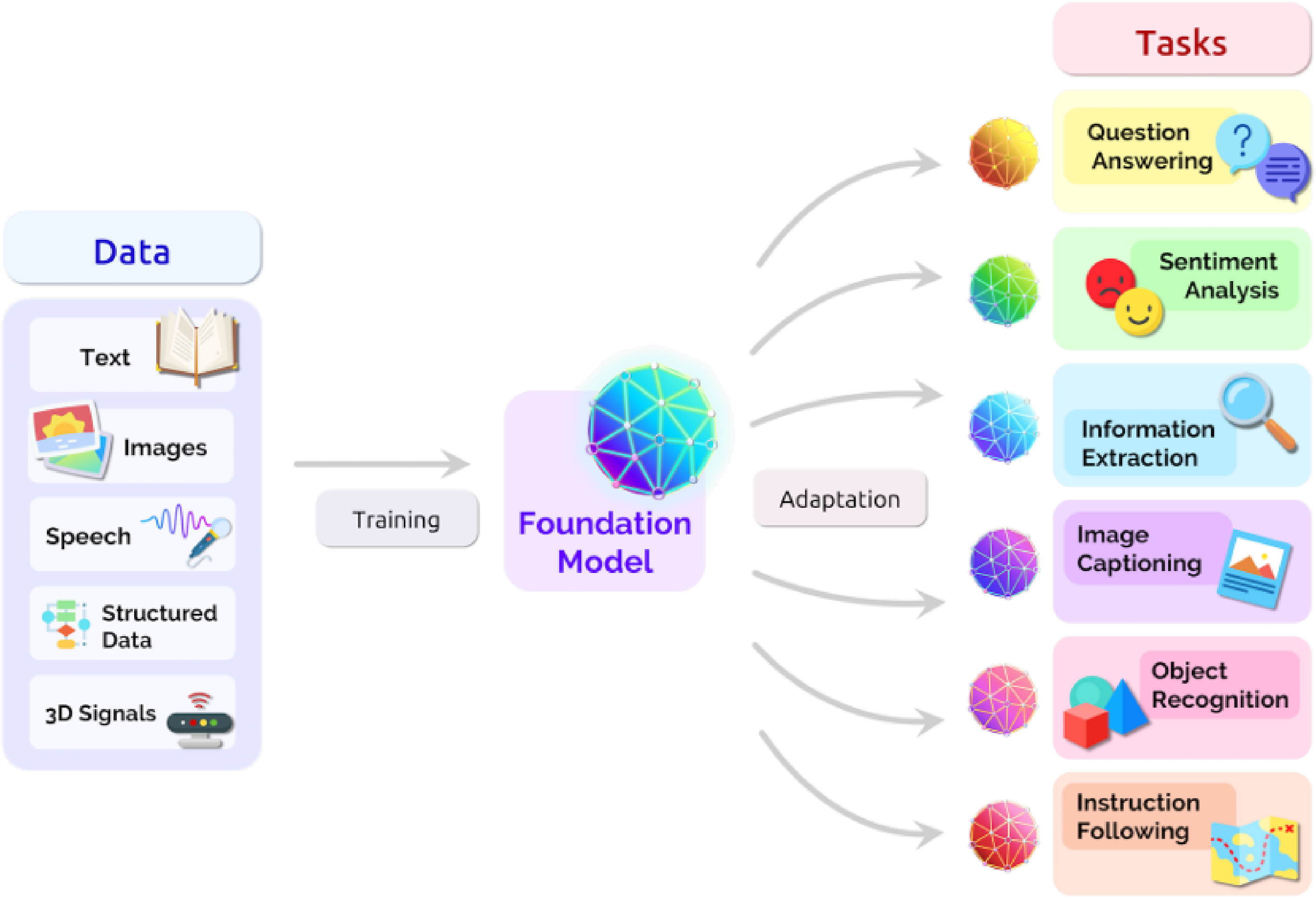
LLMs - Training

LLMs are trained following 2 phases:

1. Pre-training
   - Large amount of unlabelled data
   - Self-supervised learning (learn one part of the input from another part of the input)
   - General training (not task specific)
   - Computationally expensive

2. Fine-tuning
   - Labelled data
   - Specific to down-stream task (e.g., translation, summarisation, Q&A, …)
   - Computationally cheaper

LLMs – Foundation models

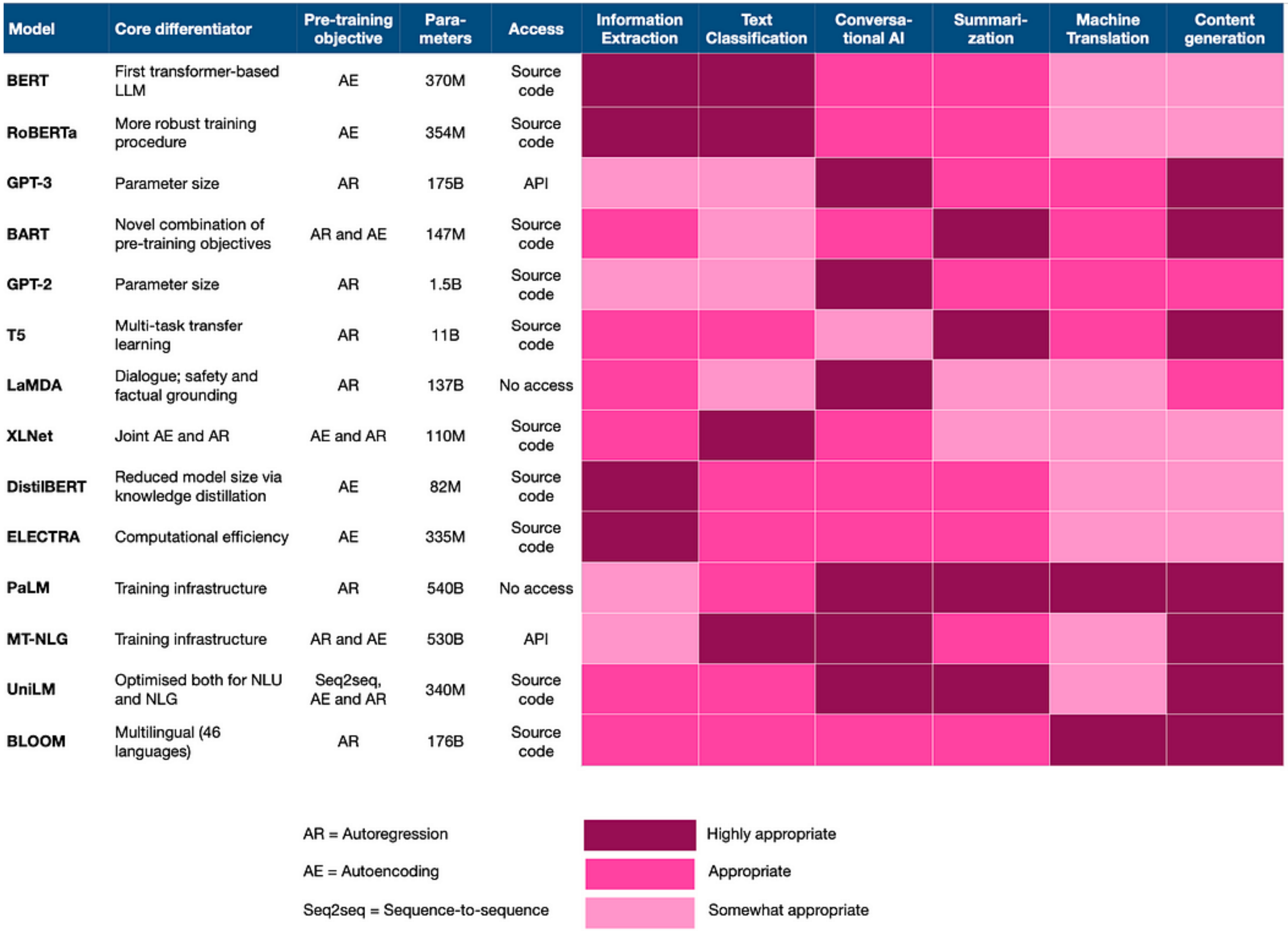


[On the Opportunities and Risks of Foundation Models, Bommasani et al., 2022]

LLMs – Popular models

- BERT (Google AI) and derivatives ALBERT, RoBERTa, CamemBERT, …
- GPT-2, GPT-3 (OpenAI)
- BART (Facebook AI)
- T5 (Google AI)
- Turing-NLG (Microsoft)
- Megatron-LM (NVIDIA) and
- Megatron Turin NLG (NVIDIA and Microsoft)
- PaLM (Google AI)

LLMs – Features and tasks



Model	Core differentiator	Pre-training objective	Para-meters	Access	Information Extraction	Text Classification	Conversa-tional AI	Summari-zation	Machine Translation	Content generation
BERT	First transformer-based LLM	AE	370M	Source code	■	■	▣	▣	□	□
RoBERTa	More robust training procedure	AE	354M	Source code	■	■	▣	▣	□	□
GPT-3	Parameter size	AR	175B	API	□	□	■	▣	▣	■
BART	Novel combination of pre-training objectives	AR and AE	147M	Source code	▣	□	▣	■	▣	■
GPT-2	Parameter size	AR	1.5B	Source code	□	□	■	▣	▣	▣
T5	Multi-task transfer learning	AR	11B	Source code	▣	▣	□	■	▣	■
LaMDA	Dialogue; safety and factual grounding	AR	137B	No access	▣	□	■	□	□	▣
XLNet	Joint AE and AR	AE and AR	110M	Source code	▣	■	▣	□	□	□
DistilBERT	Reduced model size via knowledge distillation	AE	82M	Source code	■	▣	▣	▣	□	□
ELECTRA	Computational efficiency	AE	335M	Source code	■	▣	▣	▣	□	□
PaLM	Training infrastructure	AR	540B	No access	□	▣	■	■	■	■
MT-NLG	Training infrastructure	AR and AE	530B	API	□	■	■	▣	□	■
UniLM	Optimised both for NLU and NLG	Seq2seq, AE and AR	340M	Source code	▣	▣	■	■	□	■
BLOOM	Multilingual (46 languages)	AR	176B	Source code	▣	▣	▣	▣	■	■

AR = Autoregression

AE = Autoencoding

Seq2seq = Sequence-to-sequence

■ Highly appropriate

▣ Appropriate

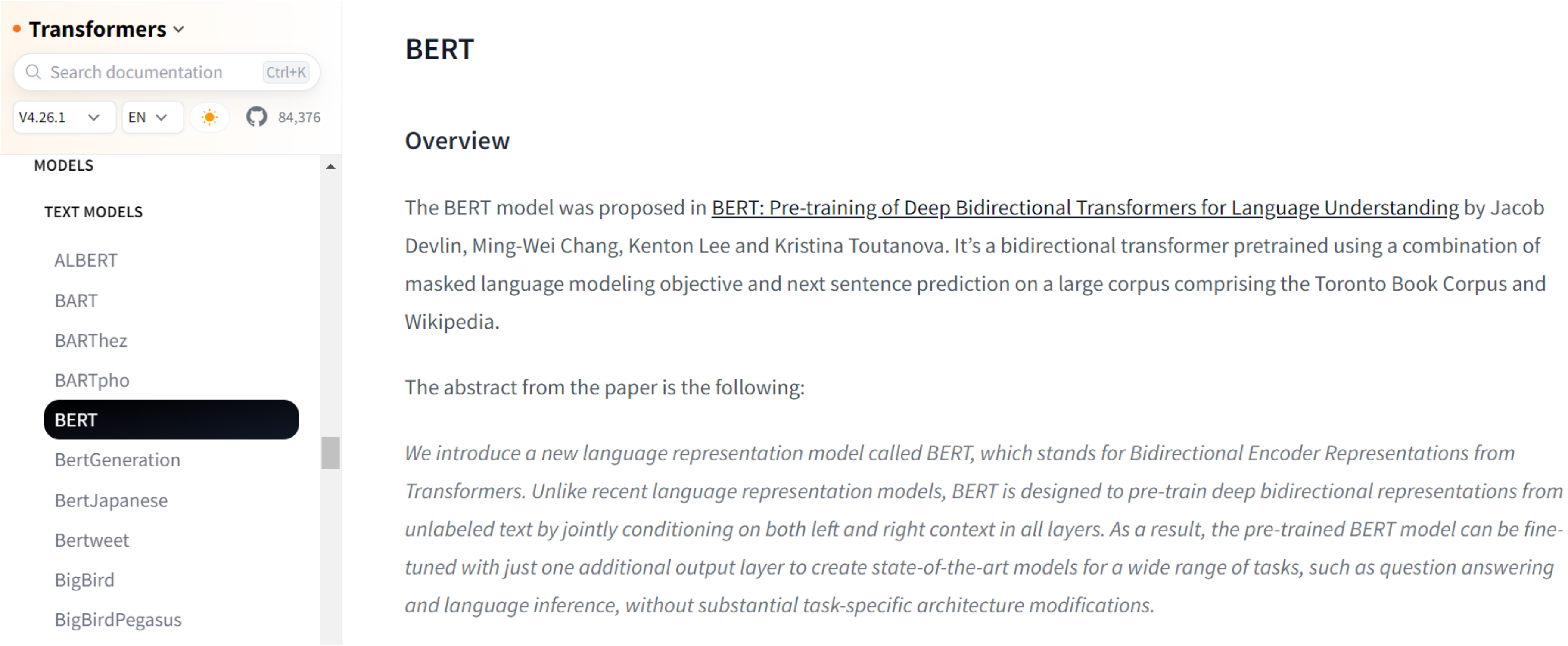
□ Somewhat appropriate

Image source: Janna Lipenkova - Choosing the right language model for your NLP use case

LLMs – Documentation and ressources

HuggingFace Transformers library

- Large collection of documentations and ressources about models and datasets.



BERT

Overview

The BERT model was proposed in BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding by Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. It's a bidirectional transformer pretrained using a combination of masked language modeling objective and next sentence prediction on a large corpus comprising the Toronto Book Corpus and Wikipedia.

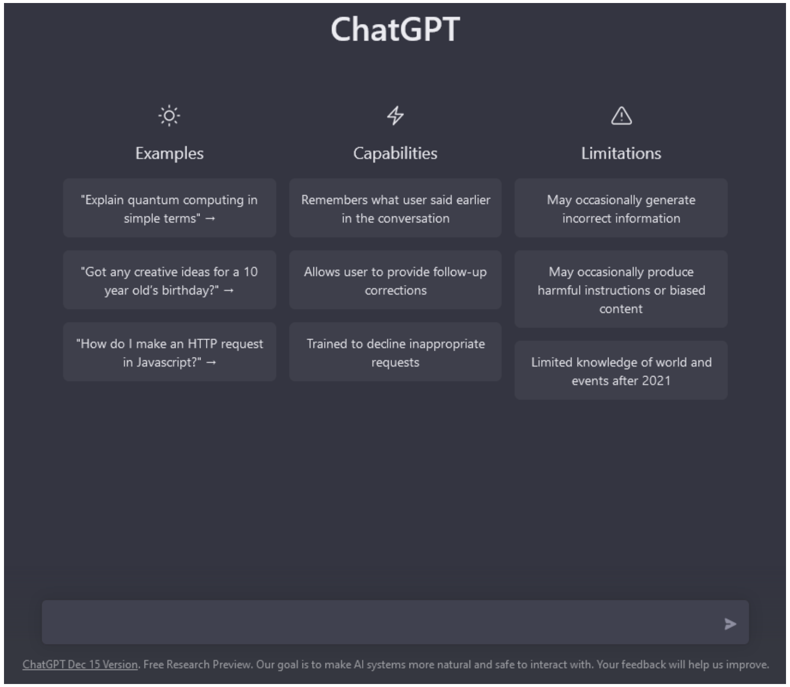
The abstract from the paper is the following:

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

https://huggingface.co/docs/transformers/v4.26.1/en/model_doc/index

Guiding the model with prompts

- A prompt is a short piece of text that is used to guide the output of a large language model (LLM).



ChatGPT

Examples

"Explain quantum computing in simple terms" →

"Got any creative ideas for a 10 year old's birthday?" →

"How do I make an HTTP request in Javascript?" →

Capabilities

Remembers what user said earlier in the conversation

Allows user to provide follow-up corrections

Trained to decline inappropriate requests

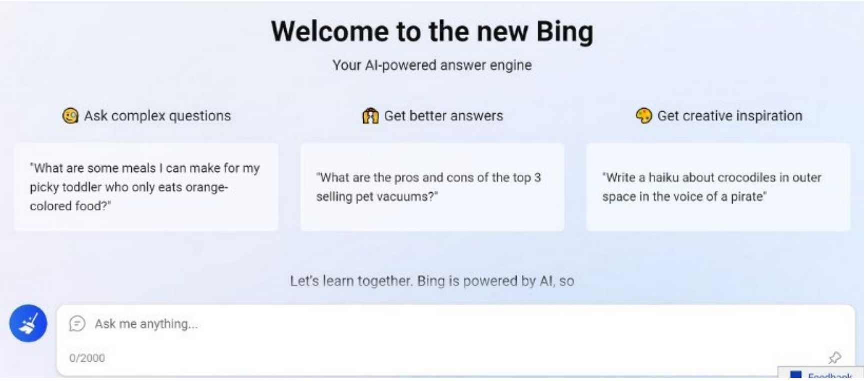
Limitations

May occasionally generate incorrect information

May occasionally produce harmful instructions or biased content

Limited knowledge of world and events after 2021

ChatGPT Dec 15 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.



Welcome to the new Bing

Your AI-powered answer engine

Ask complex questions

"What are some meals I can make for my picky toddler who only eats orange-colored food?"

Get better answers

"What are the pros and cons of the top 3 selling pet vacuums?"

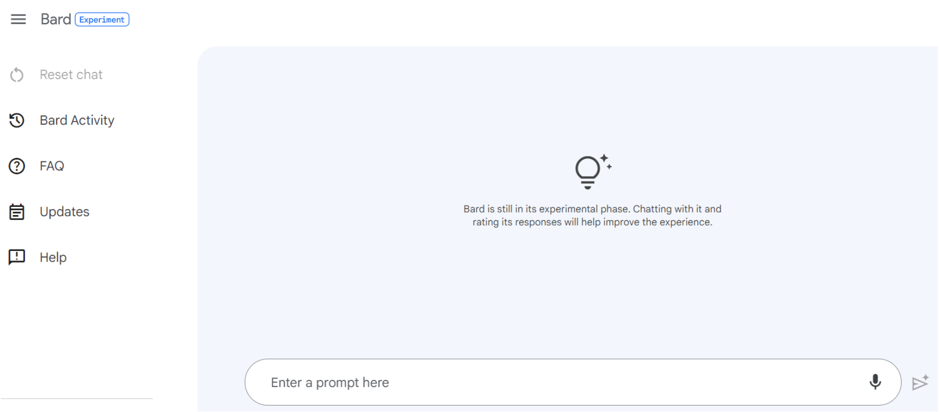
Get creative inspiration

"Write a haiku about crocodiles in outer space in the voice of a pirate"

Let's learn together. Bing is powered by AI, so

Ask me anything...

0/2000



Bard Experiment

Reset chat

Bard Activity

FAQ

Updates

Help

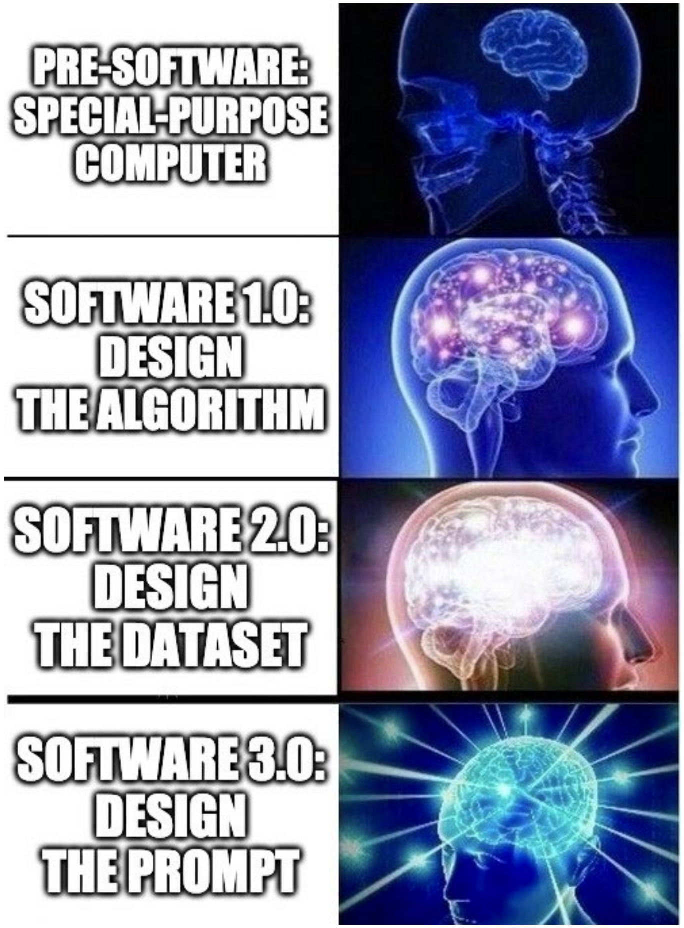
Bard is still in its experimental phase. Chatting with it and rating its responses will help improve the experience.

Enter a prompt here

Guiding the model with prompts

- A prompt provides *context* for the model to generate a *completion*.
- The amount of tokens (e.g., words) in a single conversation (set of prompts + completions) is limited by memory constrains (e.g., ~ 4000 tokens for original version of ChatGPT, up to 32000 tokens for most recent versions).
- This limits the context the LLM has access to when completing future prompts.

Designing prompts


PRE-SOFTWARE: SPECIAL-PURPOSE COMPUTER
SOFTWARE 1.0: DESIGN THE ALGORITHM
SOFTWARE 2.0: DESIGN THE DATASET
SOFTWARE 3.0: DESIGN THE PROMPT

LLMs can « hallucinate »

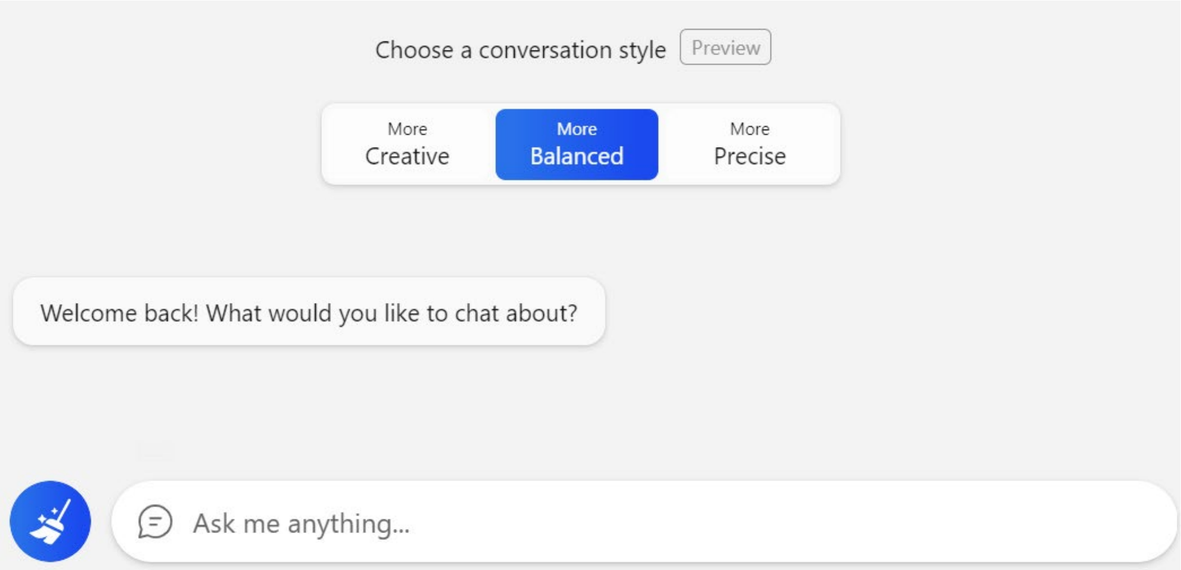
- Hallucination happens when the model generates text that is semantically or syntactically plausible but is in fact incorrect or nonsensical.

- LLMs are not doing any explicit fact checking when answering questions. If it gives a true information, it can be seen as a “secondary effect” of its original training task.

- Hallucination is a complex and not well understood yet phenomenon, but it can be linked to:
  - Incorrect information in the training data.
  - Model not being trained on enough data (“improvising”).
  - Model trying to generate completion which is too creative or original.

Example: Bing Chat (GPT-4)

- User can choose between 3 conversation styles:
  - Creative: answers are more original and creative (can generate images through Dall-E model) → more hallucinations?
  - Precise: answers are short and to the point.
  - Balanced: answers are balanced between the 2 styles.



Choose a conversation style Preview

Welcome back! What would you like to chat about?

ChatGPT

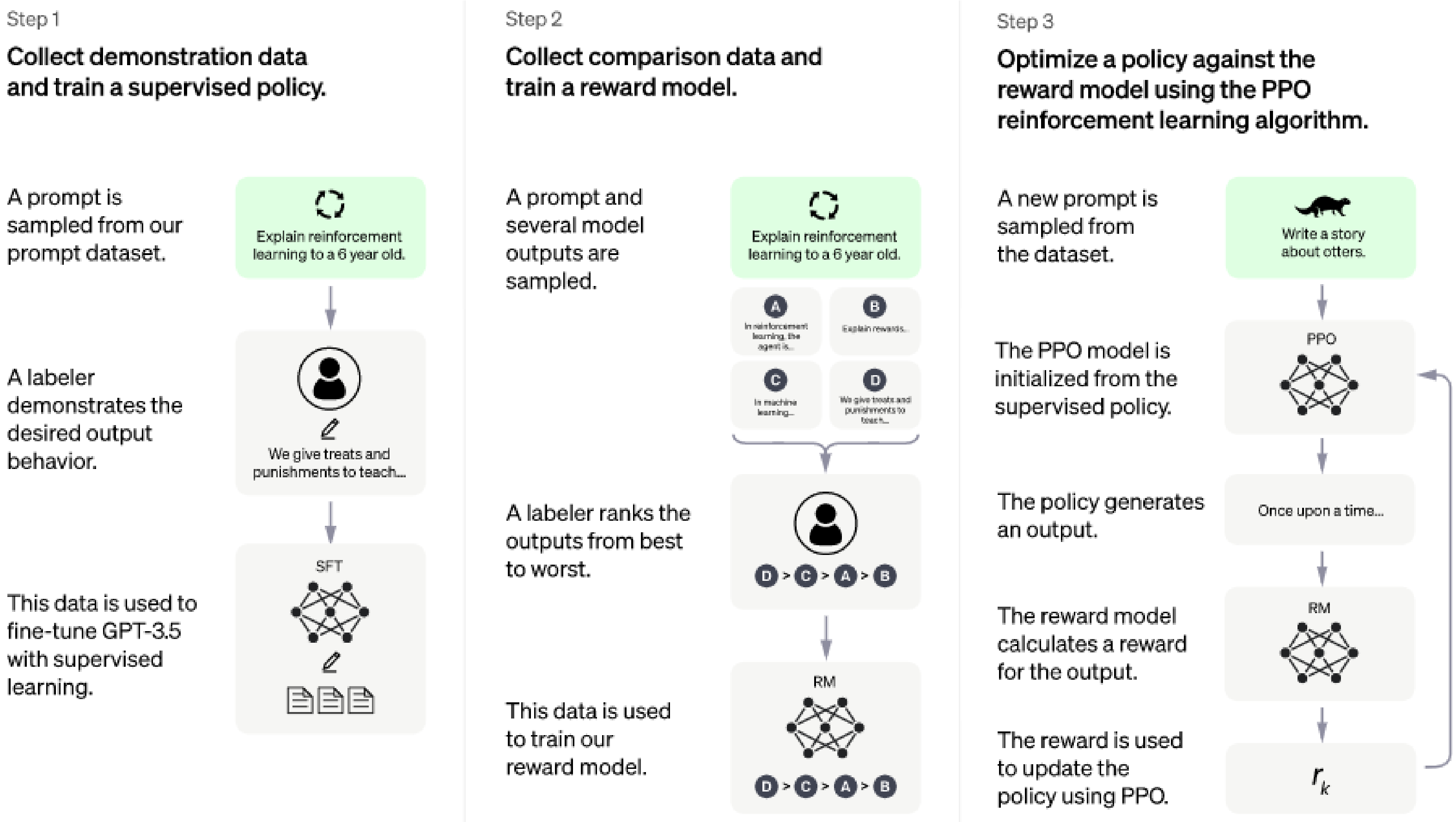
- Chat-GPT is a chatbot based on the GPT-3.5 LLMs series.

- Fine-tuned on a variety of NLP tasks including translation, summarisation, Q&A and dialogue generation.

The model used for Chat-GPT is specially fine-tuned for chatbot applications, where the goal is to generate human-like responses to user inputs in a conversational manner.

ChatGPT

- Also fine-tuned using reinforcement learning.



Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.

Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

Explain reinforcement learning to a 6 year old.
We give treats and punishments to teach...
SFT
Explain reinforcement learning to a 6 year old.
A
In reinforcement learning, the agent is...
B
Explain rewards...
C
In machine learning...
D
We give treats and punishments to teach...
D > C > A > B
RM
D > C > A > B
Write a story about otters.
PPO
Once upon a time...
RM
r_k

Image source: https://openai.com/blog/chatgpt/

ChatGPT

- Good at:
  - Changing style, summarising, picking up errors in code, …
  - Sounding like a human!

- But it can also:
  - Make up stuff (« hallucinate »)
  - And sound confident about it! (also true for some other tools using LLMs)

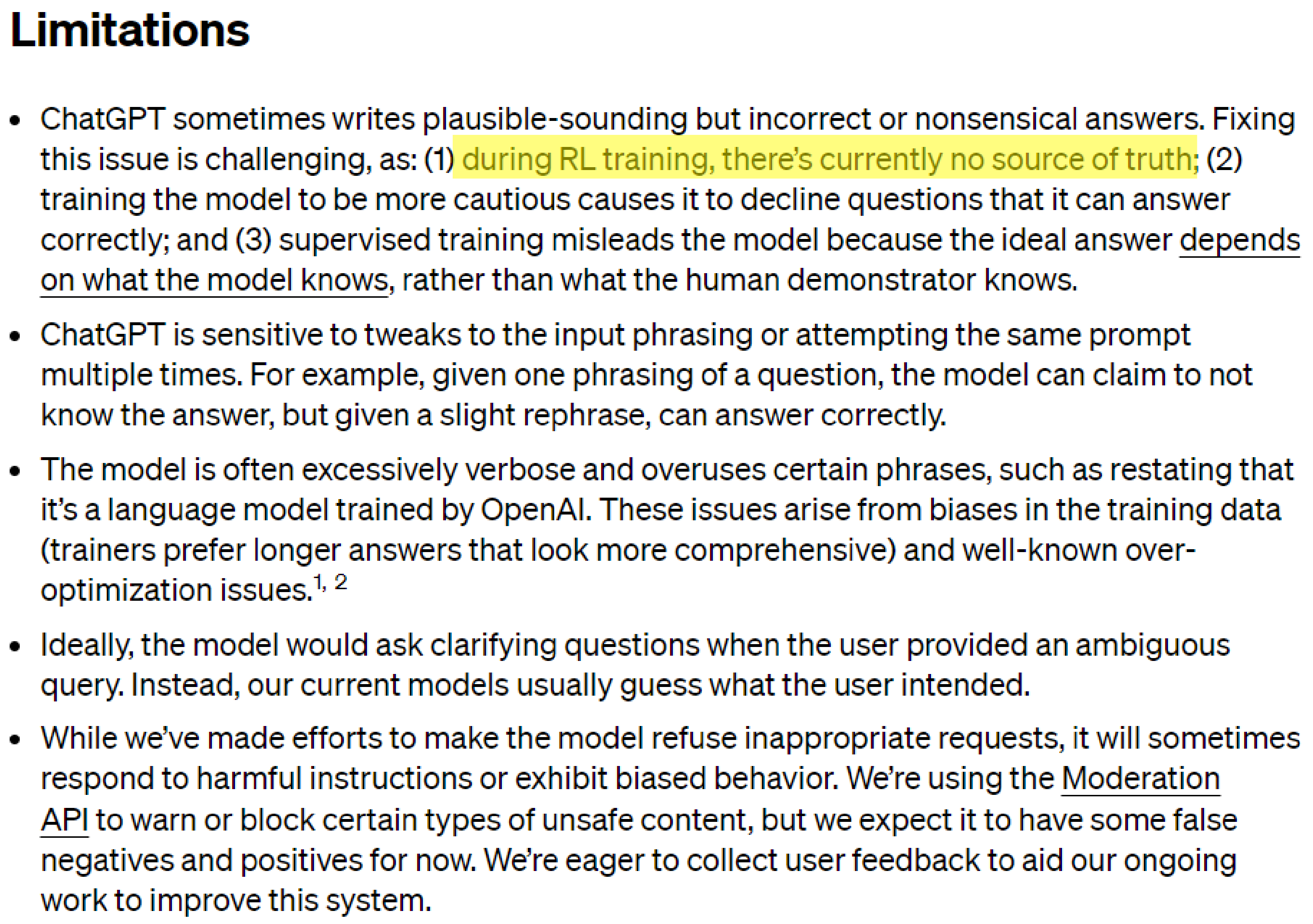
- Be careful with:
  - Factual information (does not do explicite fact checking),
  - Complexe tasks,
  - Logical reasonning.

ChatGPT – A few things to keep in mind

ChatGPT is not doing any explicit fact checking when “answering” questions. It can “hallucinate”.

ChatGPT is fine-tuning based on RL using ranking from human labellers, with their own subjectivity.

ChatGPT - Limitations



Limitations

- ChatGPT sometimes writes plausible-sounding but incorrect or nonsensical answers. Fixing this issue is challenging, as: (1) during RL training, there's currently no source of truth; (2) training the model to be more cautious causes it to decline questions that it can answer correctly; and (3) supervised training misleads the model because the ideal answer depends on what the model knows, rather than what the human demonstrator knows.
- ChatGPT is sensitive to tweaks to the input phrasing or attempting the same prompt multiple times. For example, given one phrasing of a question, the model can claim to not know the answer, but given a slight rephrase, can answer correctly.
- The model is often excessively verbose and overuses certain phrases, such as restating that it's a language model trained by OpenAI. These issues arise from biases in the training data (trainers prefer longer answers that look more comprehensive) and well-known over-optimization issues.[1, 2]
- Ideally, the model would ask clarifying questions when the user provided an ambiguous query. Instead, our current models usually guess what the user intended.
- While we've made efforts to make the model refuse inappropriate requests, it will sometimes respond to harmful instructions or exhibit biased behavior. We're using the Moderation API to warn or block certain types of unsafe content, but we expect it to have some false negatives and positives for now. We're eager to collect user feedback to aid our ongoing work to improve this system.

Image source: https://openai.com/blog/chatgpt/

Storing knowledge vs retrieving it

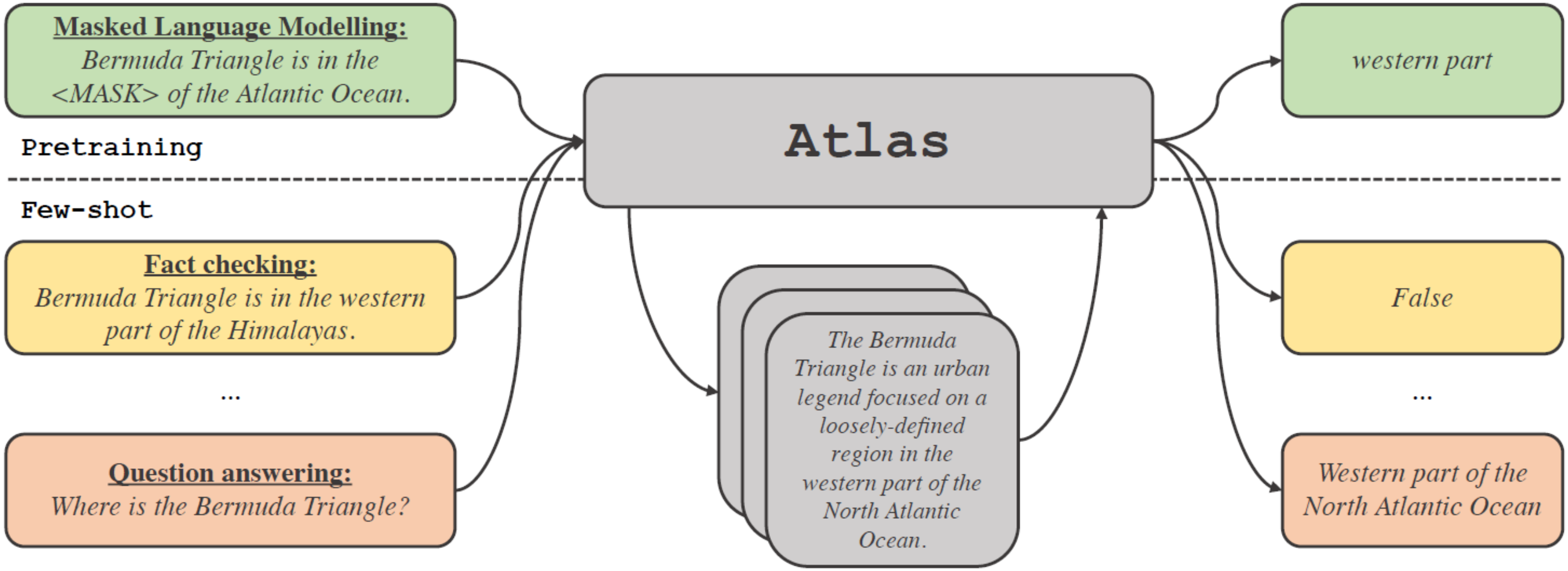
- LLMs encode knowledge in their parameters. It is hard to say if they « know » and what they « know ».

- In task where factual knowledge is important (e.g., Question Answering, Fact Checking), large parameter counts usually leads to better results, but it still leads to hallucinations.

- Some works focus on training smaller networks and augmenting the query with relevant documents retrieved from a database.

- Retrieval augmented networks show good performances in knowledge intensive tasks.

Example: ATLAS


Masked Language Modelling:
Bermuda Triangle is in the <MASK> of the Atlantic Ocean.
Pretraining
Few-shot
Fact checking:
Bermuda Triangle is in the western part of the Himalayas.
...
Question answering:
Where is the Bermuda Triangle?
Atlas
The Bermuda Triangle is an urban legend focused on a loosely-defined region in the western part of the North Atlantic Ocean.
western part
False
...
Western part of the North Atlantic Ocean

Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., ... & Grave, E. (2022). Few-shot learning with retrieval augmented language models. arXiv preprint arXiv:2208.03299.

Cost of training large laguage models

- Cost of training vs model size:
  - \$2.5k- \$50k (110 million parameter model)
  - \$10k- \$200k (340 million parameter model)
  - \$80k- \$1.6m (1.5 billion parameter model)

[The cost of training nlp models: A concise overview, Sharir et al., 2020]

ChatGPT:

- Newest version (gpt-3.5-turbo): \$0.002 per 1000 tokens (10x less than a few months ago)

- Training GPT-3 consumed an estimated 1,287 MWh (~65k inhabitant city consumption per day in NZ) and produced 552 CO2e (~80 Auckland-London return flights in economy class).

[The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink, Patterson et al., 2022]

Litterature

- Transformers/Attention: [Attention Is All You Need, Vaswani et al., 2017]
  https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

- Popular models based on Transformers:
  - Google AI BERT [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", Devlin et al., 2018]

    https://arxiv.org/pdf/1810.04805.pdf
  - OpenAI GPT-2 [Language Models are Unsupervised Multitask Learners, Radford et al., 2018]

    https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
  - OpenAI GPT-3 [Language Models are Few-Shot Learners, Brown et al., 2020]

    https://arxiv.org/pdf/2005.14165.pdf
  - Facebook AI BART [BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, Lewis et al., 2019]

    https://arxiv.org/pdf/1910.13461
  - Google AI T5 [Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, Raffel et al., 2020]

    https://arxiv.org/pdf/1910.10683.pdf

Litterature

- NVIDIA Megatron-LM [Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism, Shoeybi et al., 2020]

  https://arxiv.org/abs/1909.08053

- Microsoft Turing & NVIDIA [Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model, Smith et al., 2022]

  https://arxiv.org/pdf/2201.11990

- [Training language models to follow instructions with human feedback., Ouyang, Long, et al., 2022]

  https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf

- [Few-shot learning with retrieval augmented language models, Izacard, Gautier, et al., 2022]

  https://arxiv.org/pdf/2208.03299