Transformer Case Study: LLMs

CSE 849 Deep Learning
Spring 2025

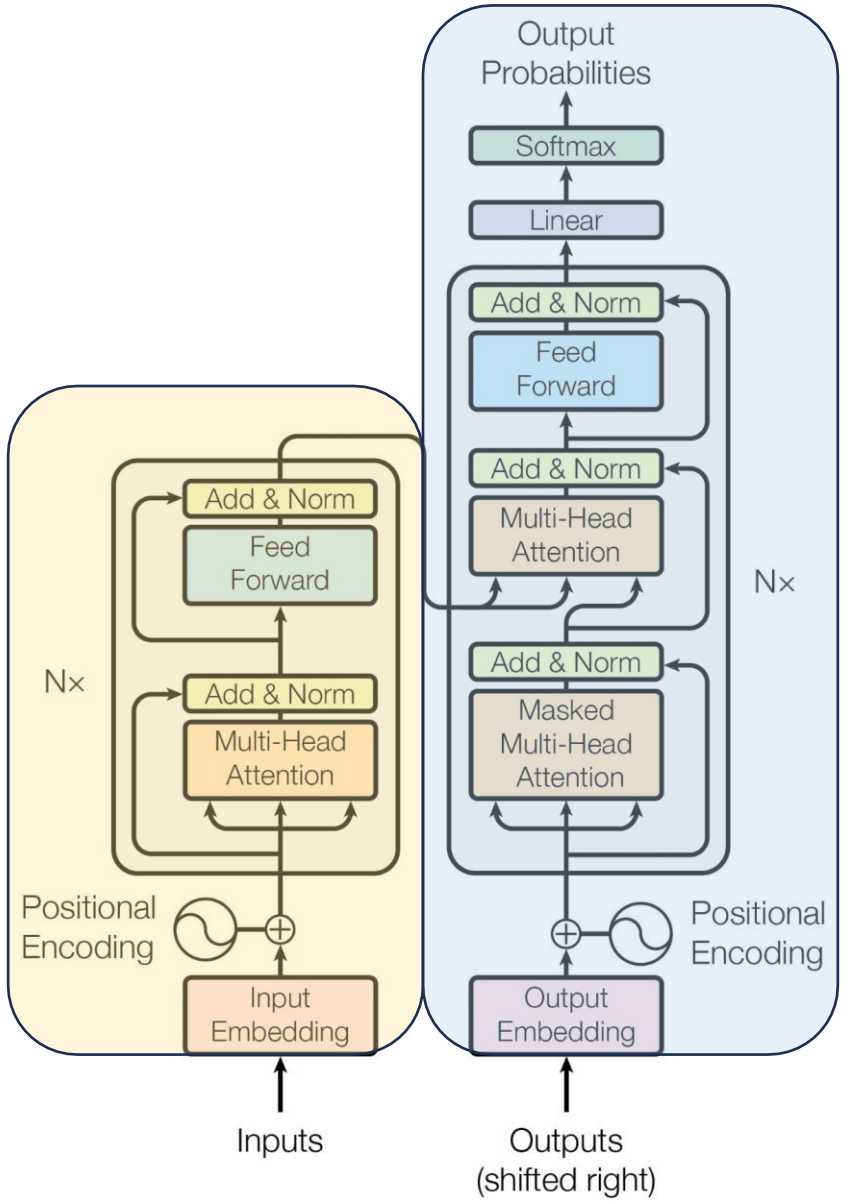
Zijun Cui

Some slides are adapted from DanqiChen, ChenyanXiong and Graham Neubig

Table of contents

Transformers, mid-2017

Representation

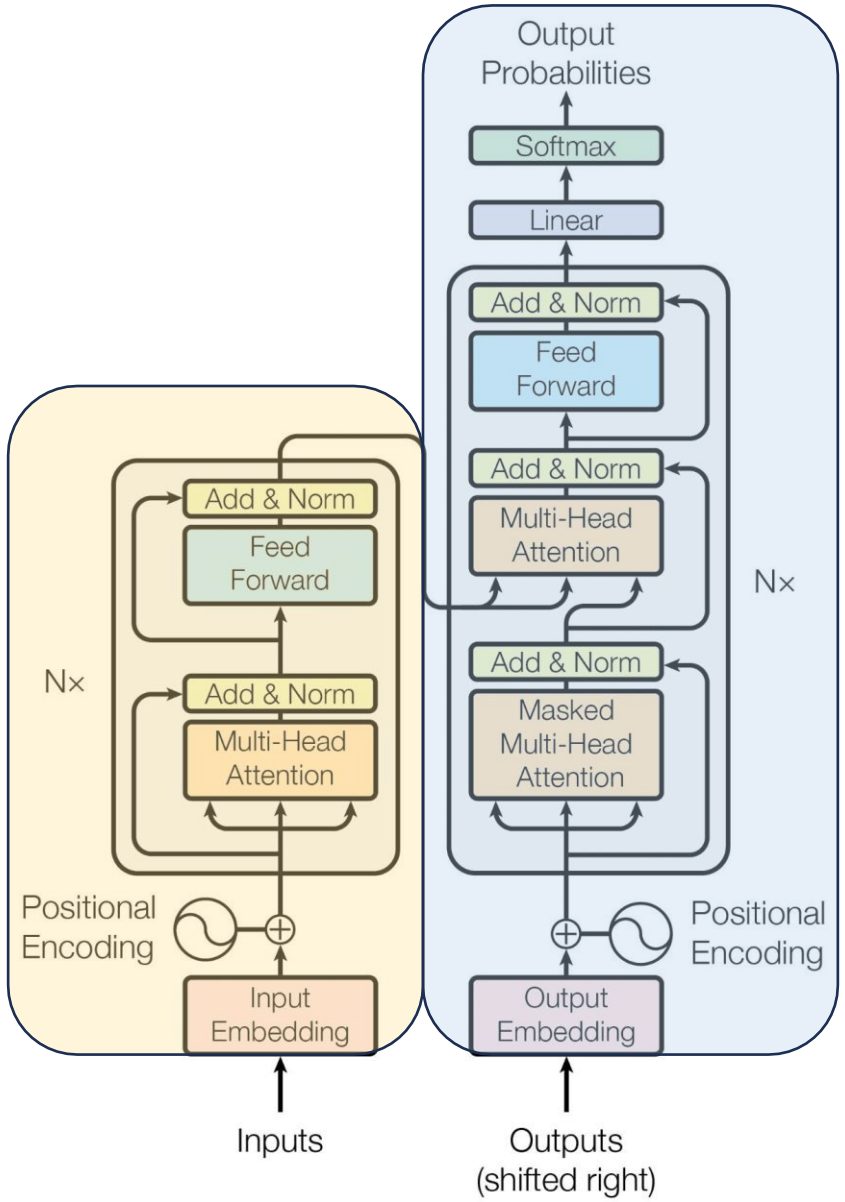


Generation

2018 – Inception of the LLM Era

BERT
Oct 2018

Representation

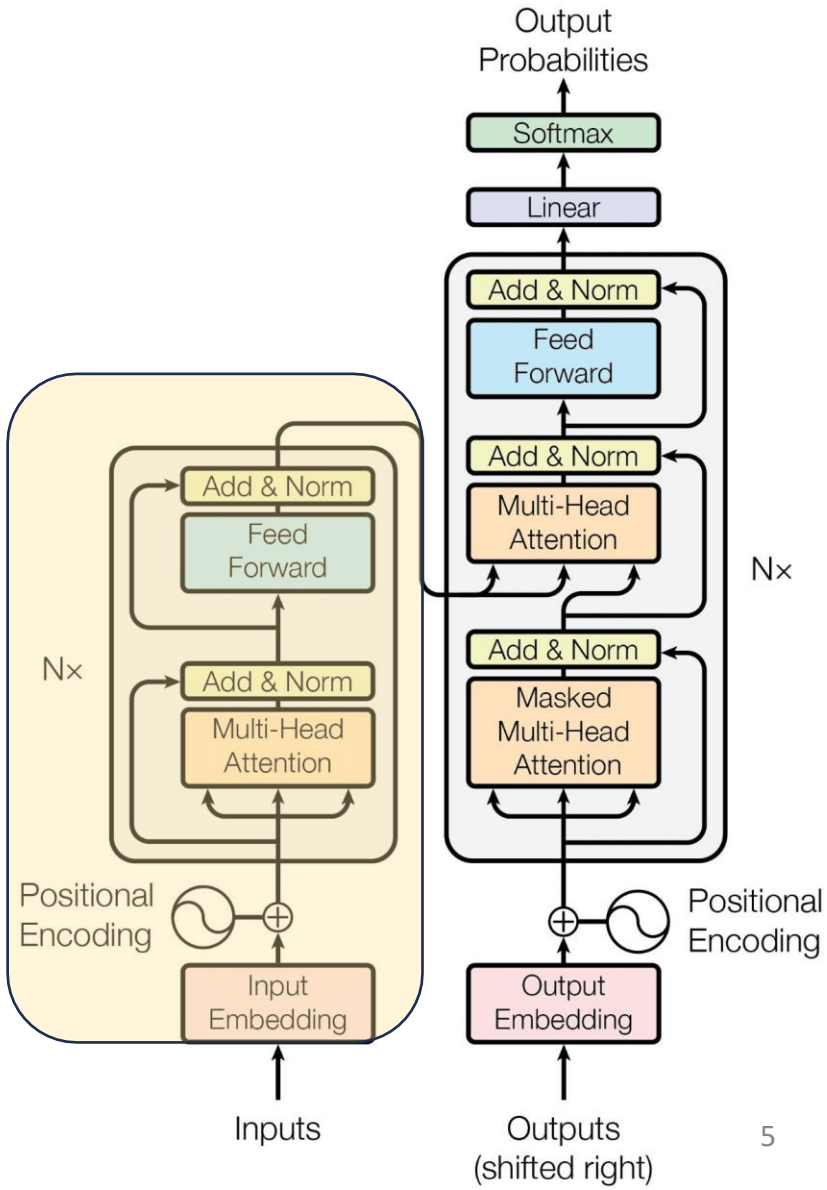


GPT
June 2018

Generation

BERT - Bidirectional Encoder Representations

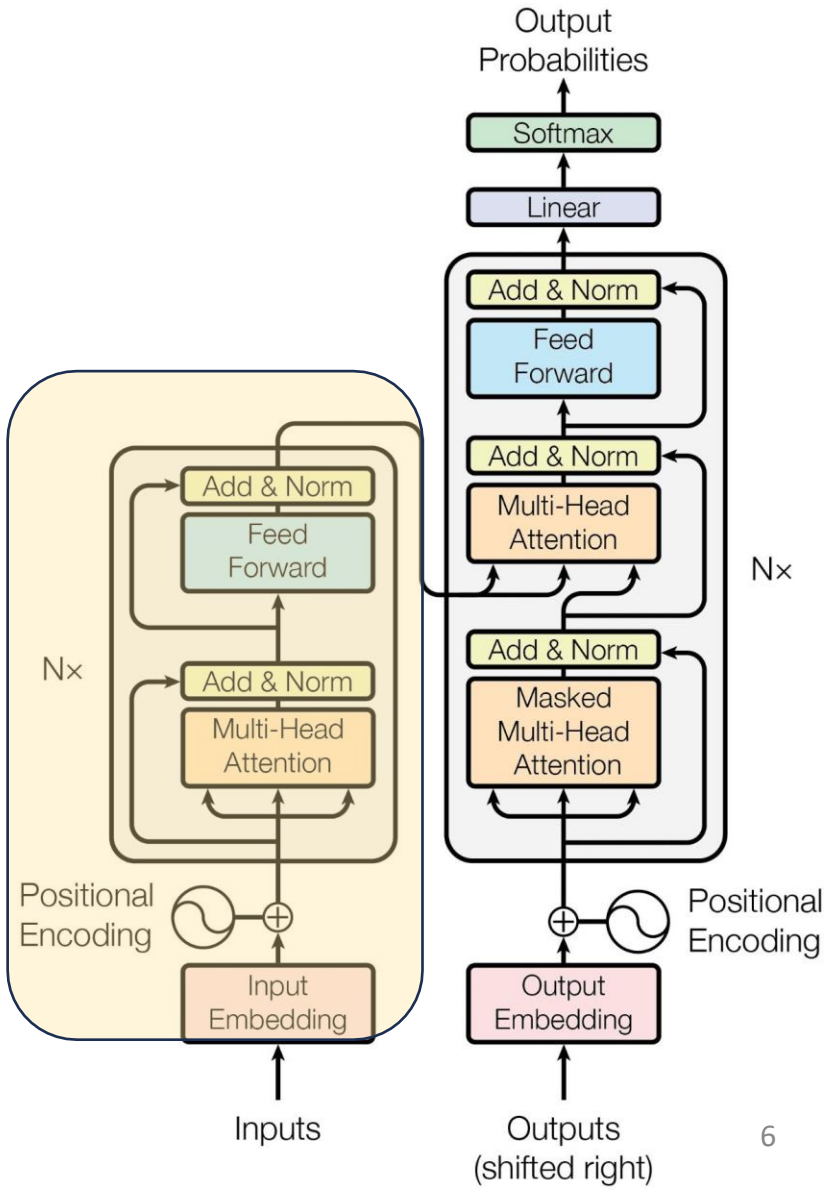
- One of the biggest challenges in LM-building used to be the lack of task-specific training data.
- What if we learn an effective representation that can be applied to a variety of downstream tasks?
  - Word2vec (2013)
  - GloVe (2014)



BERT - Bidirectional Encoder Representations

BERT Pre-Training Corpus:

- English Wikipedia - 2,500 million words
- Book Corpus - 800 million words



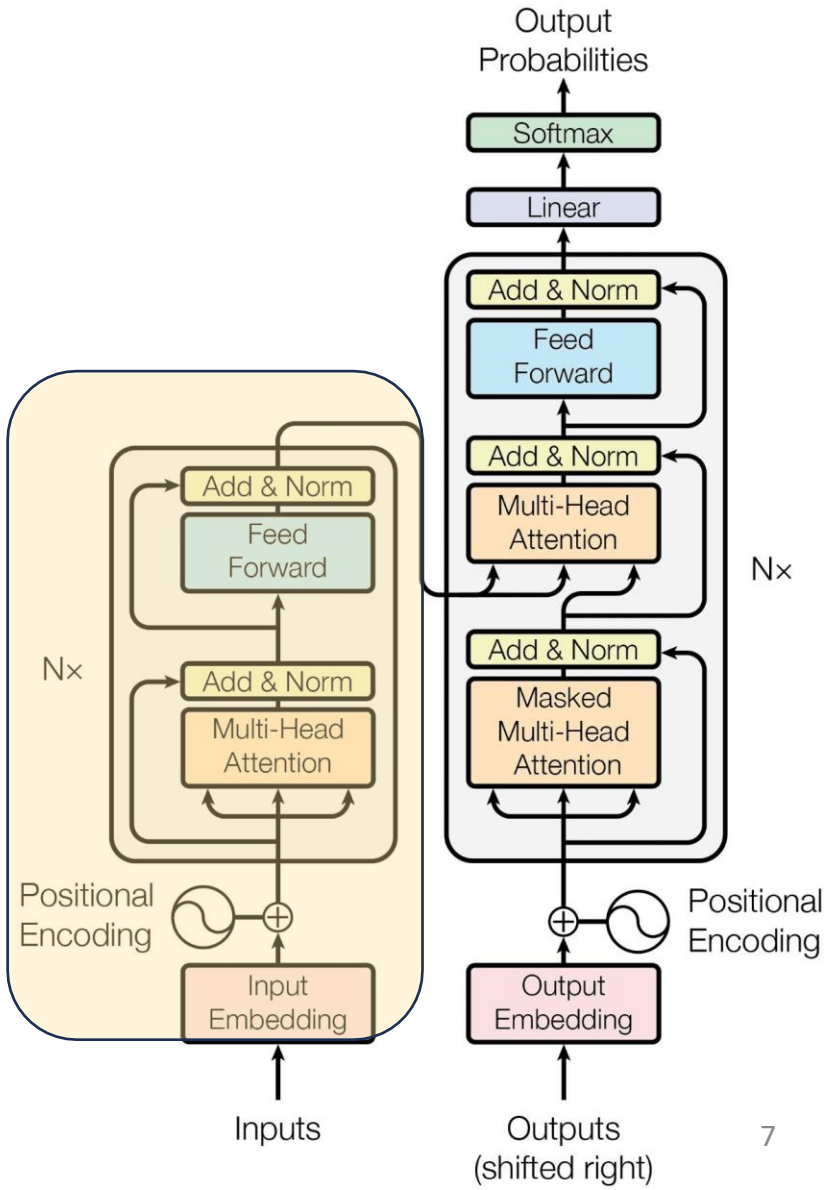
BERT - Bidirectional Encoder Representations

BERT Pre-Training Corpus:

- English Wikipedia - 2,500 million words
- Book Corpus - 800 million words

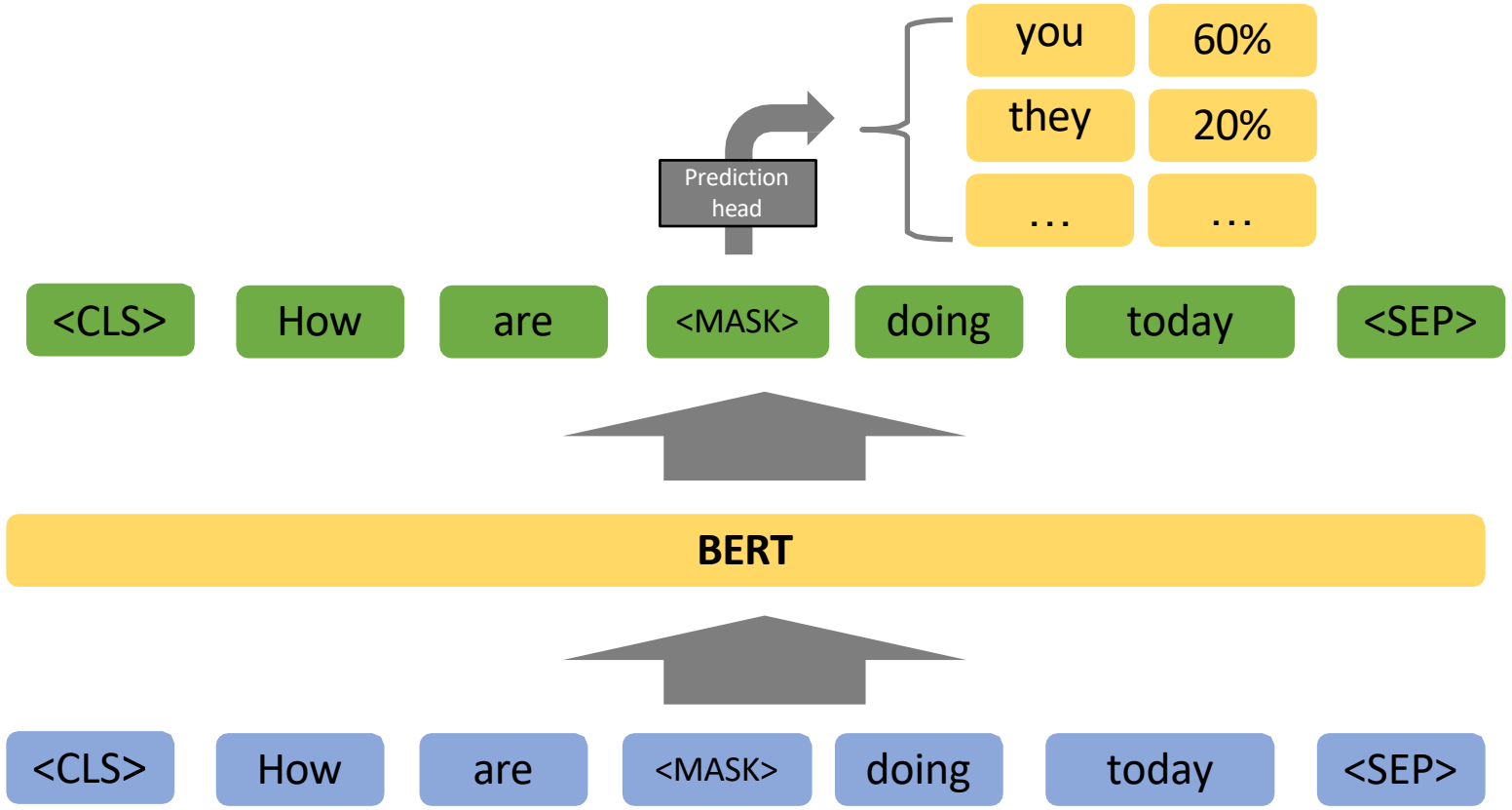
BERT Pre-Training Tasks:

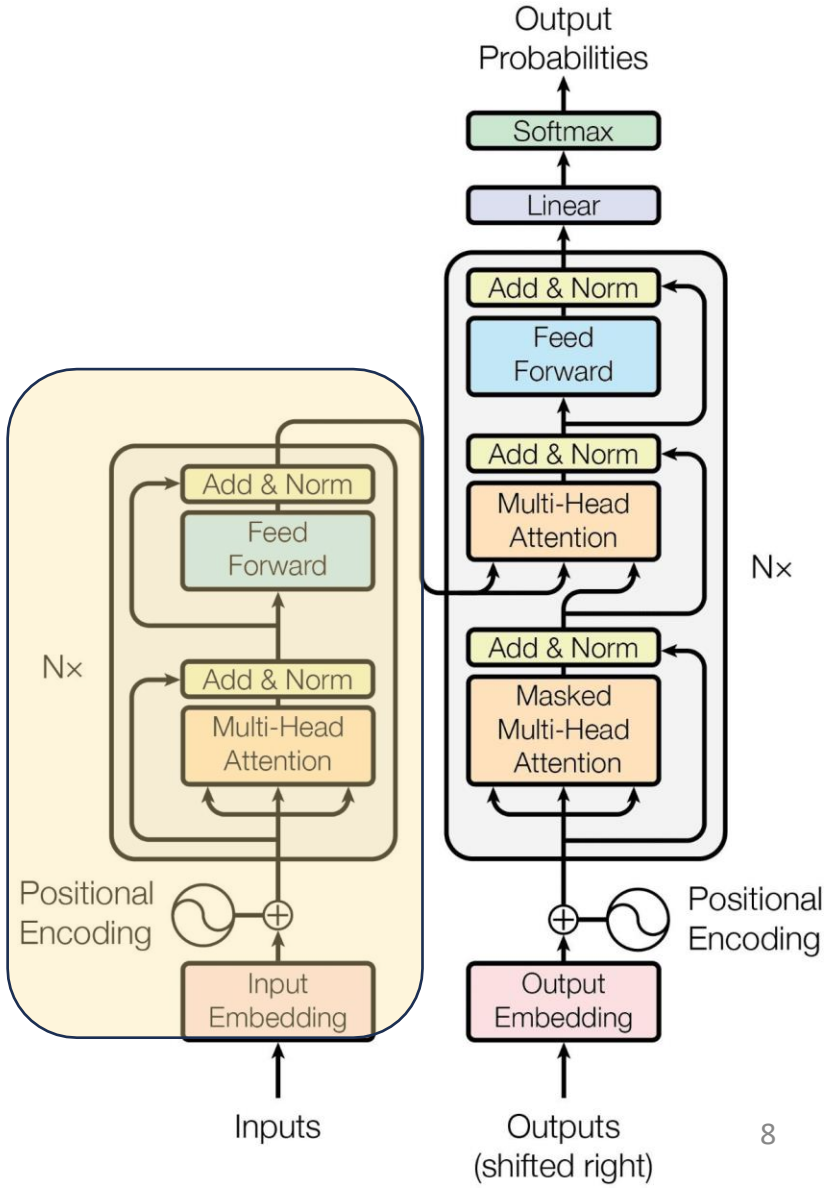
- MLM (Masked Language Modeling)
- NSP (Next Sentence Prediction)



BERT - Bidirectional Encoder Representations

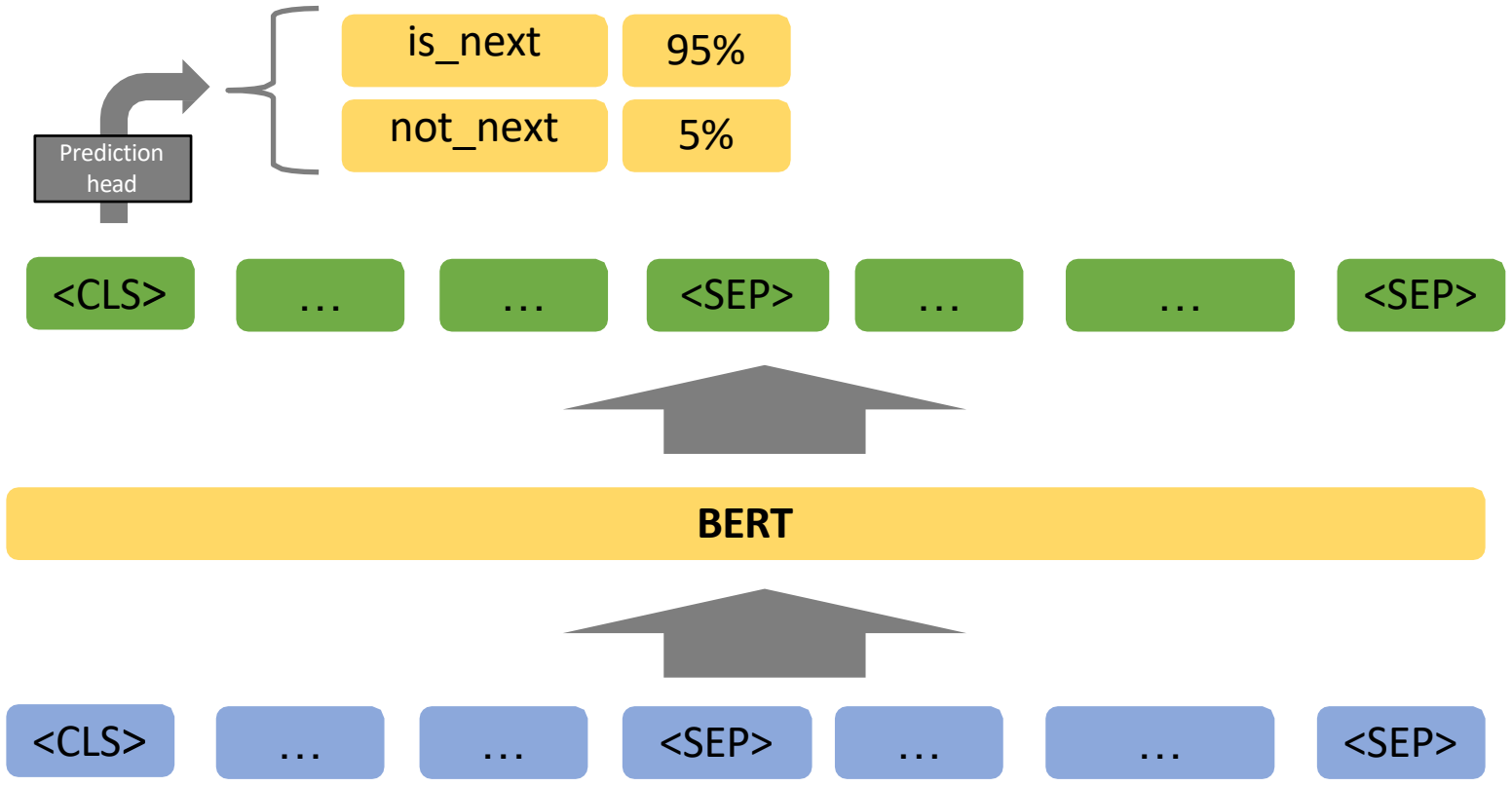
MLM (Masked Language Modeling)

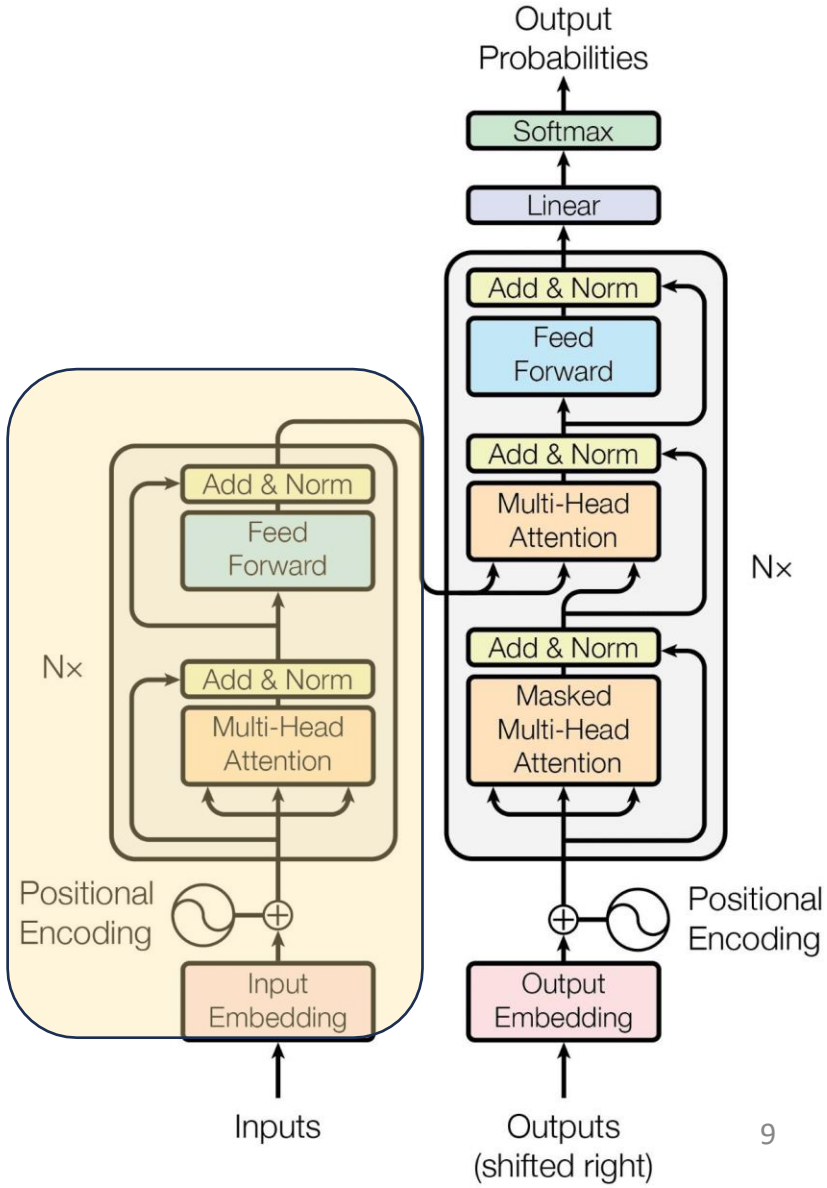




BERT - Bidirectional Encoder Representations

NSP (Next Sentence Prediction)





BERT - Bidirectional Encoder Representations

BERT Pre-Training Corpus:

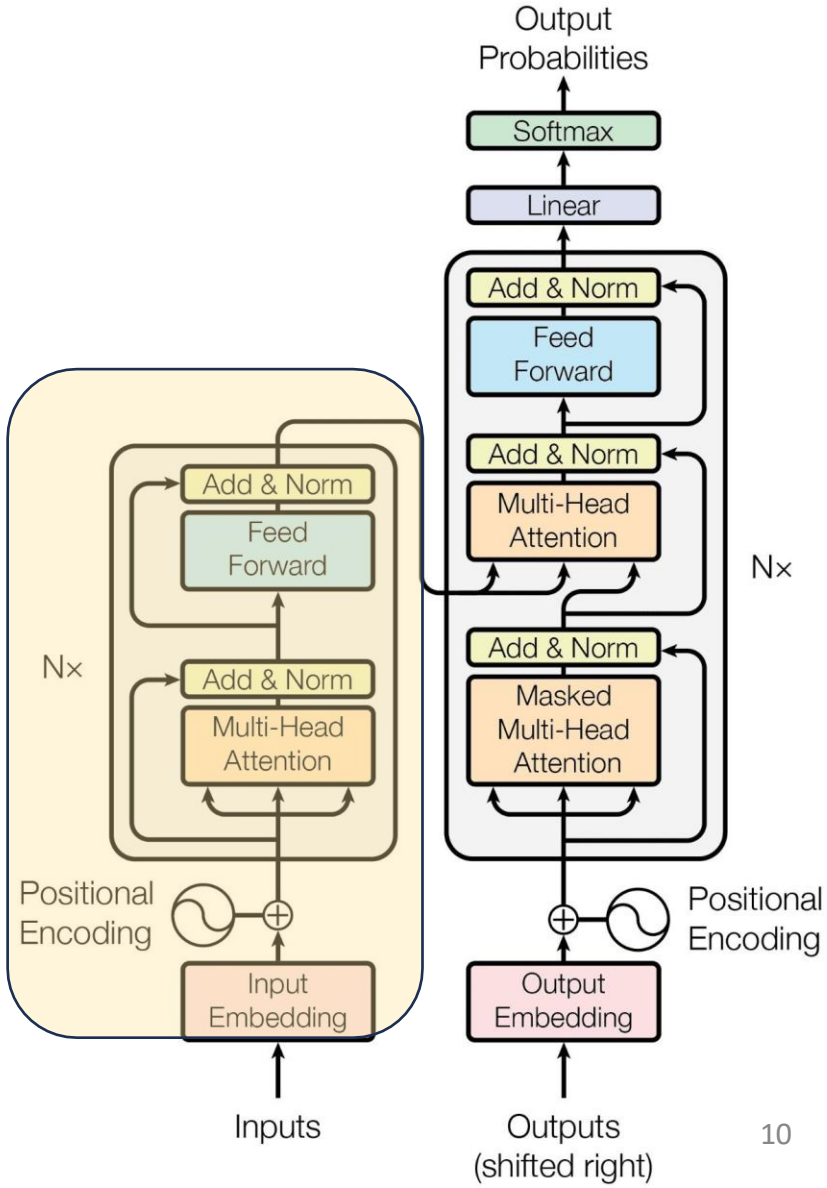
- English Wikipedia - 2,500 million words
- Book Corpus - 800 million words

BERT Pre-Training Tasks:

- MLM (Masked Language Modeling)
- NSP (Next Sentence Prediction)

BERT Pre-Training Results:

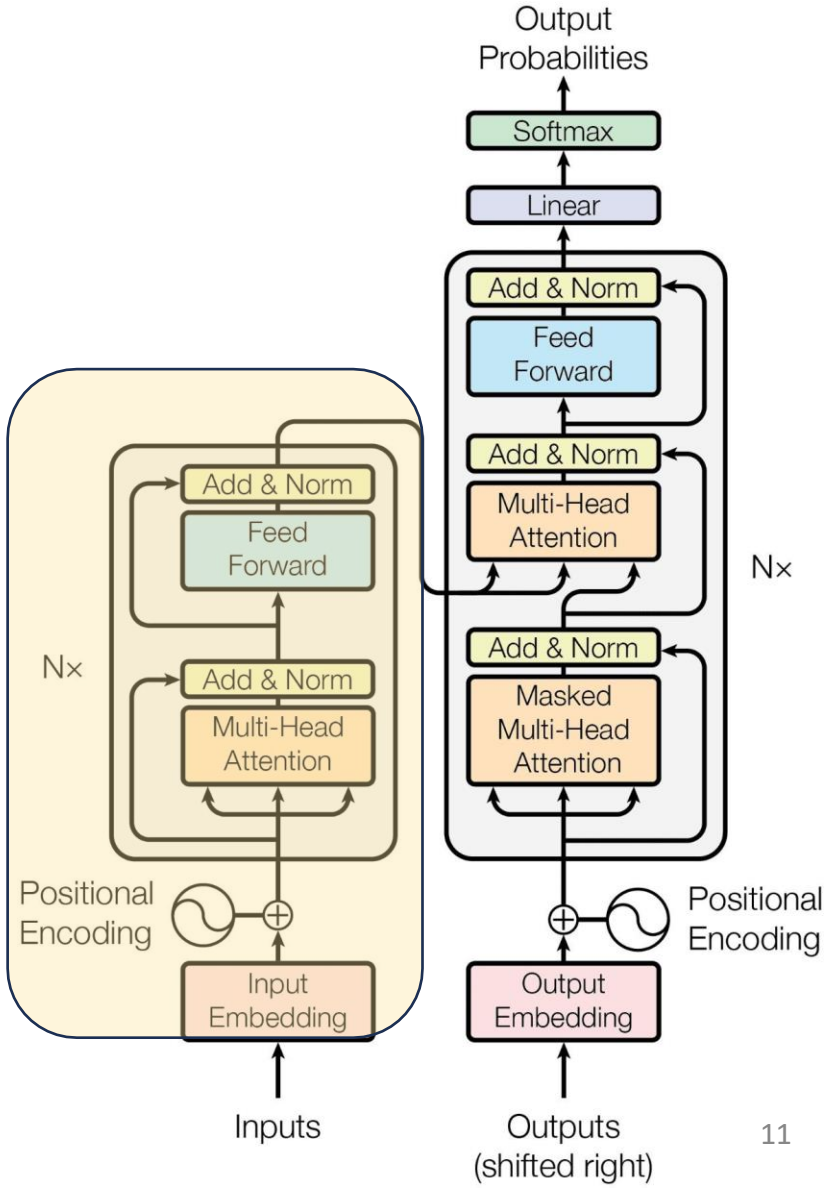
- BERT-Base – 110M Params
- BERT-Large – 340M Params



BERT - Bidirectional Encoder Representations

BERT Fine-Tuning:

- Simply add a task-specific module after the last encoder layer to map it to the desired dimension.
  - *Classification Tasks:*
    - *Add a feed-forward layer on top of the encoder output for the [CLS] token*
  - *Question Answering Tasks:*
    - *Train two extra vectors to mark the beginning and end of answer from paragraph*
  - ...

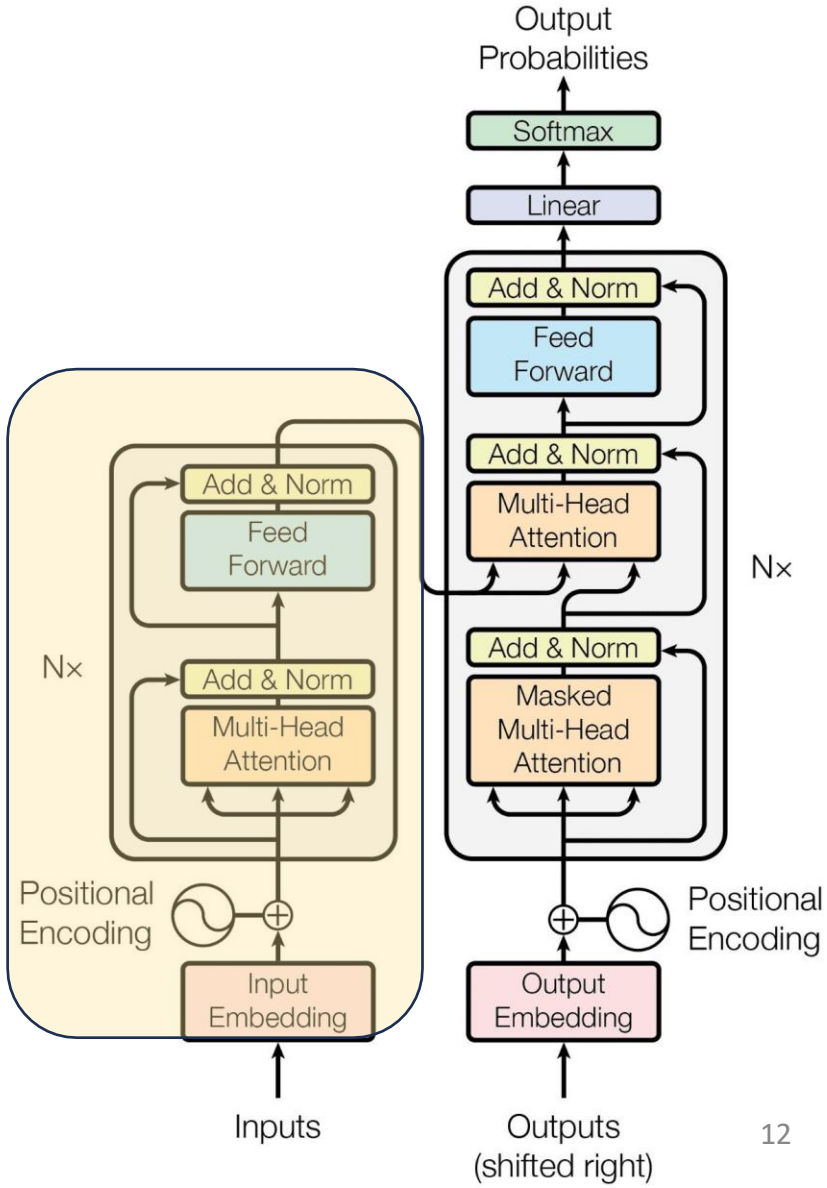


BERT - Bidirectional Encoder Representations

BERT Evaluation:

- General Language Understanding Evaluation (GLUE)
  - Sentence pair tasks
  - Single sentence classification
  - Several datasets, e.g., MNLI, QQP, CoLA, etc

- Stanford Question Answering Dataset (SQuAD)



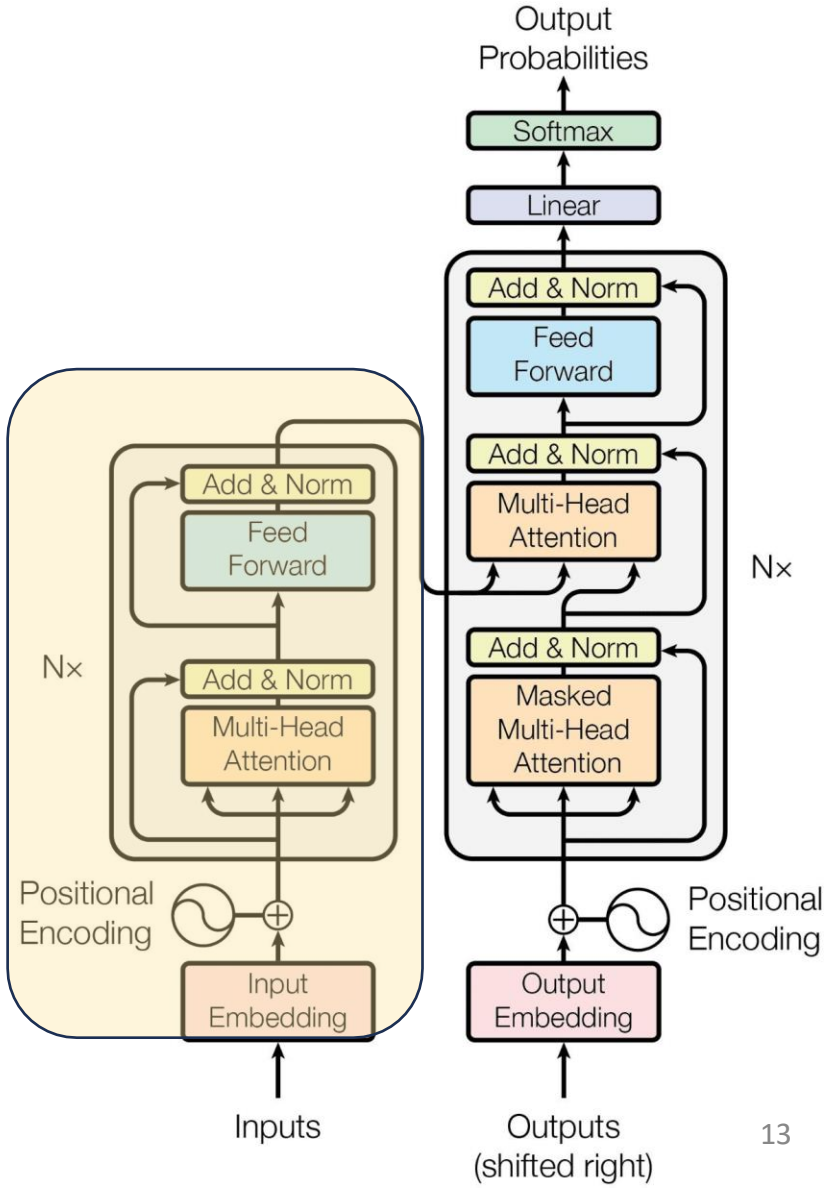
BERT - Bidirectional Encoder Representations

BERT Evaluation:

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

System	Dev EM	Dev F1	Test EM	Test F1
Leaderboard (Oct 8th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
#1 Single - nlnet	-	-	83.5	90.1
#2 Single - QANet	-	-	82.5	89.3
Published				
BiDAF+ELMo (Single)	-	85.8	-	-
R.M. Reader (Single)	78.9	86.3	79.5	86.6
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

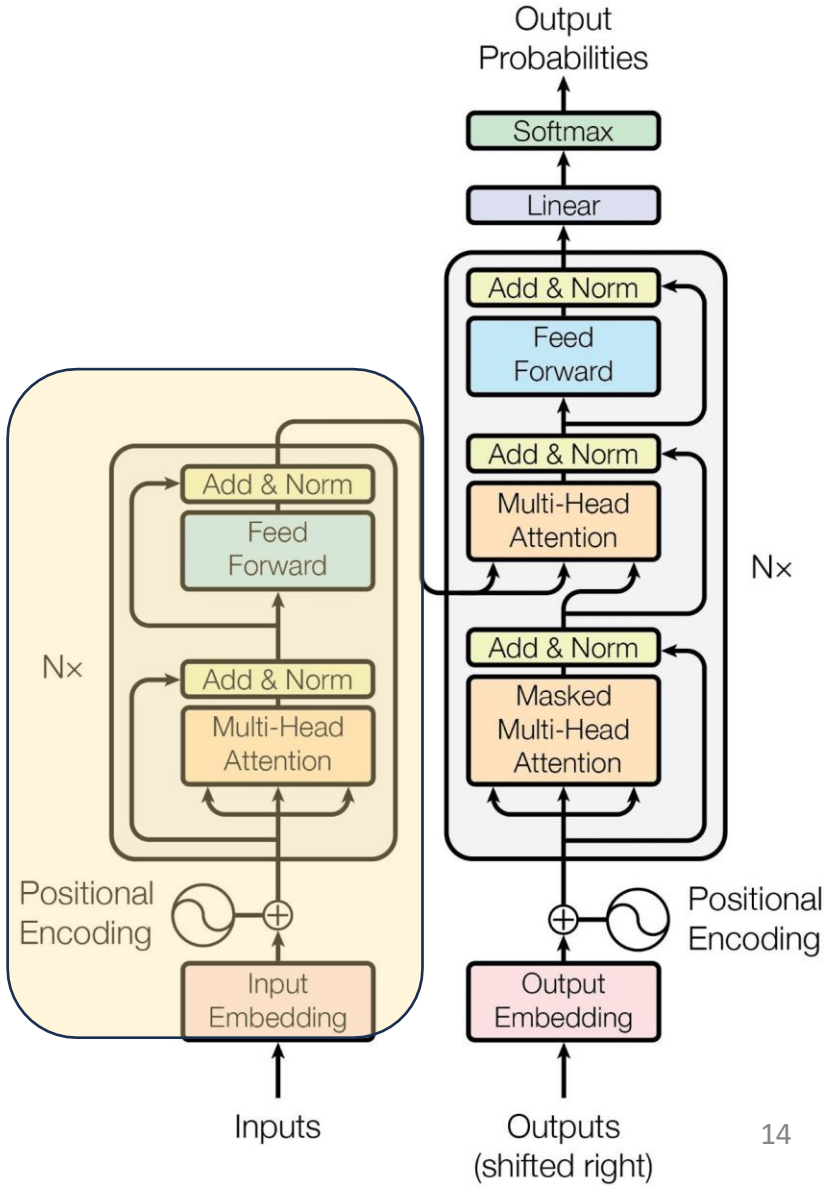
Table 2: SQuAD results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.



BERT - Bidirectional Encoder Representations

What is our takeaway from BERT?

- **Pre-training tasks can be invented flexibly…**
  - Effective representations can be derived from a flexible regime of pre-training tasks.

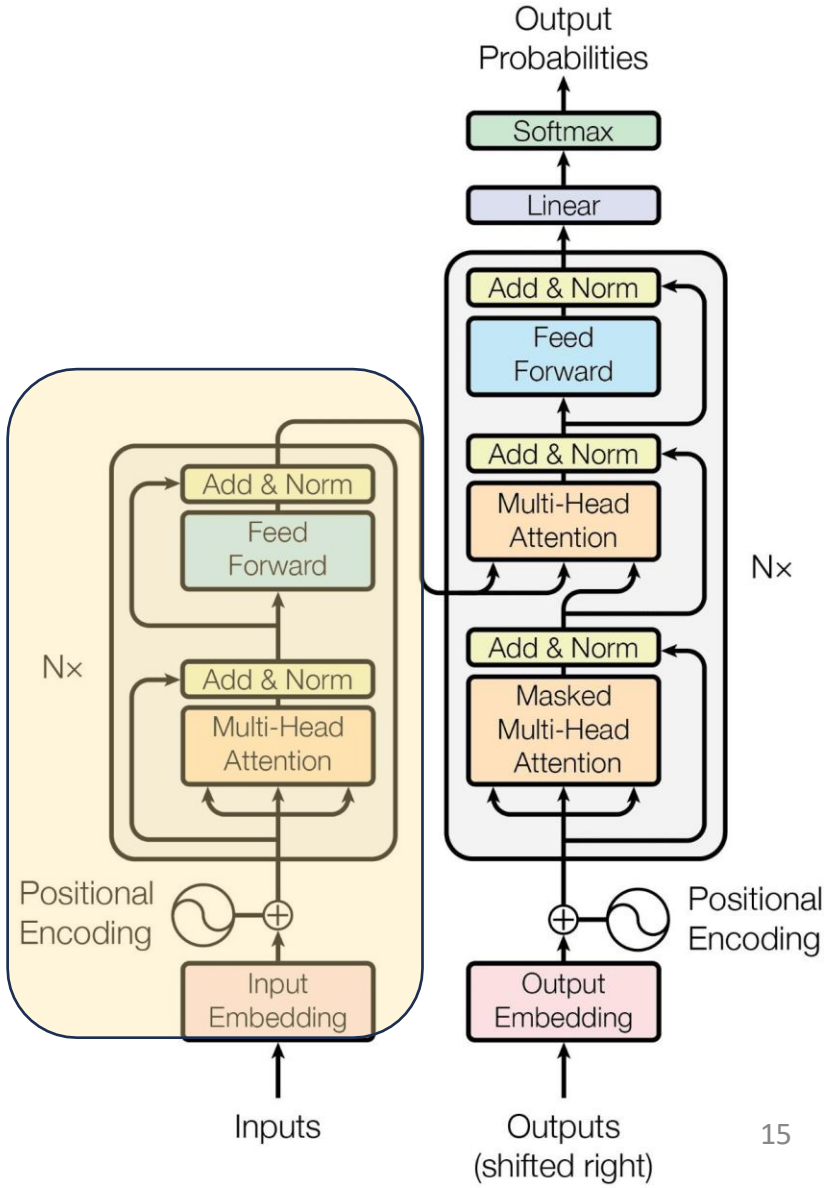


BERT - Bidirectional Encoder Representations

What is our takeaway from BERT?

- **Pre-training tasks can be invented flexibly…**
  - Effective representations can be derived from a flexible regime of pre-training tasks.

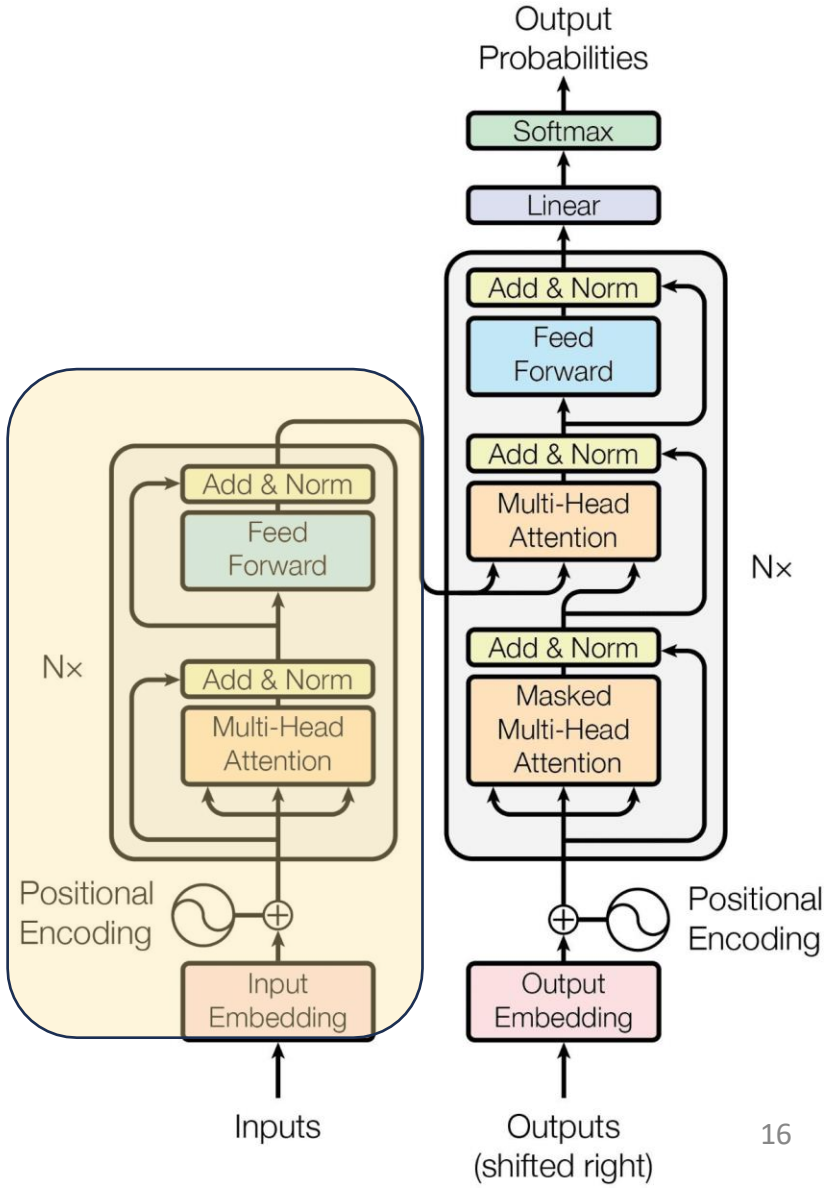
- **Different NLP tasks seem to be highly transferable with each other...**
  - As long as we have effective representations, that seems to form a general model which can serve as the backbone for many specialized models.



BERT - Bidirectional Encoder Representations

What is our takeaway from BERT?

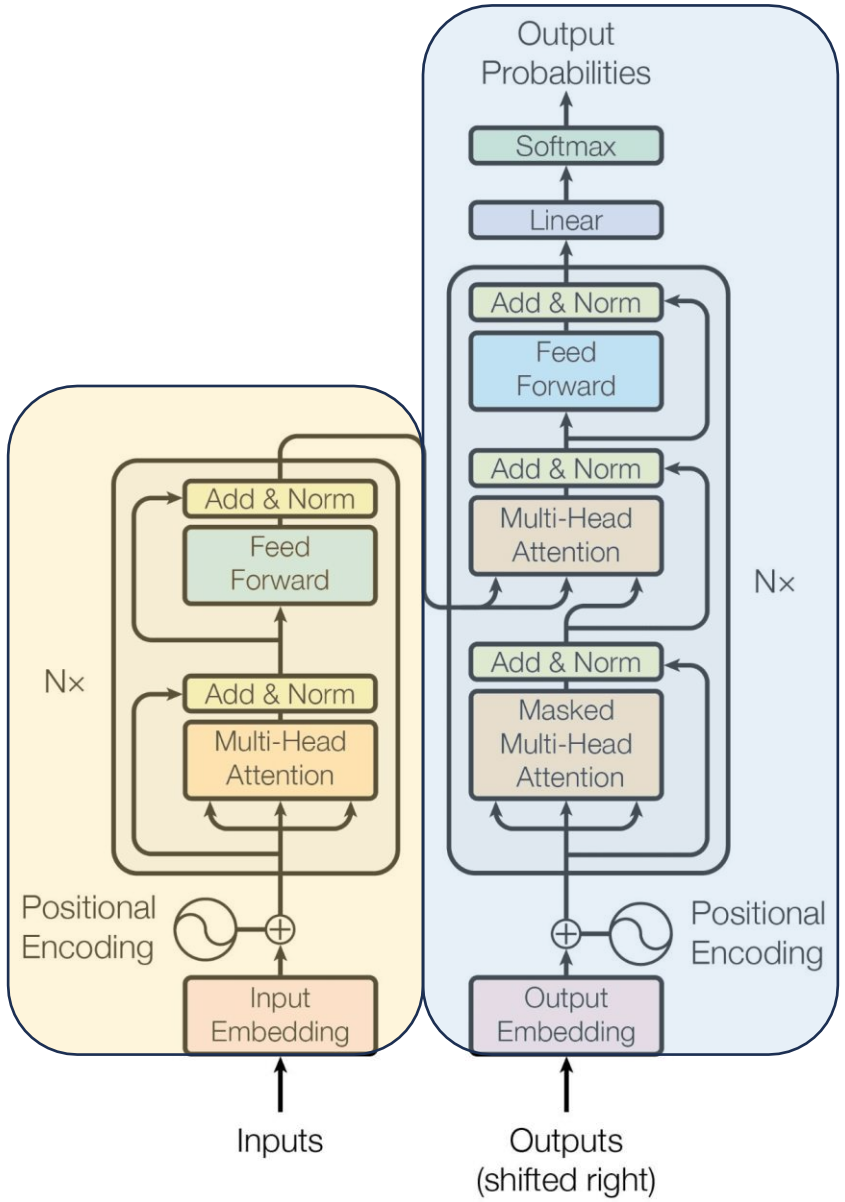
- **Pre-training tasks can be invented flexibly…**
  - Effective representations can be derived from a flexible regime of pre-training tasks.
- **Different NLP tasks seem to be highly transferable with each other...**
  - As long as we have effective representations, that seems to form a general model which can serve as the backbone for many specialized models.
- **And scaling works!!!**
  - 340M was considered large in 2018



2018 – Inception of the LLM Era

BERT
Oct 2018

Representation

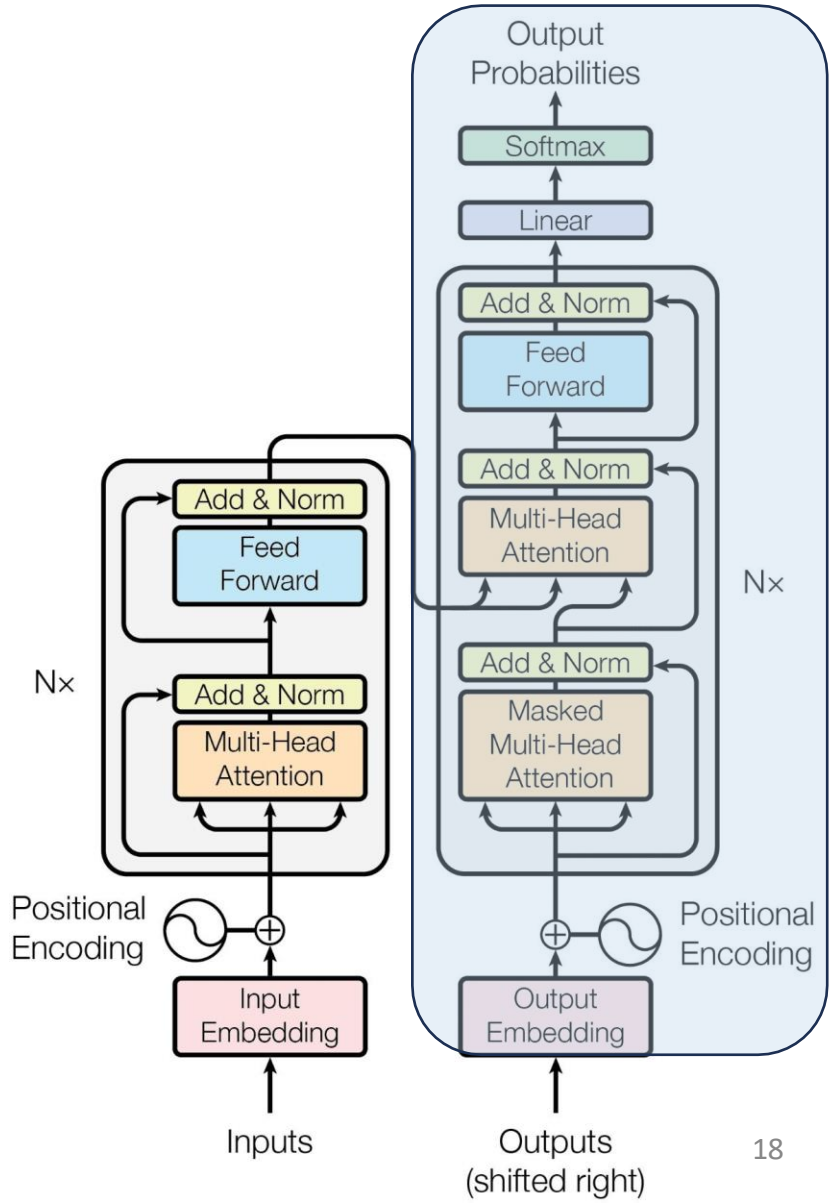


GPT
June 2018

Generation

GPT – Generative Pretrained Transformer

- Similarly motivated as BERT, though differently designed
  - Can we leverage large amounts of unlabeled data to pretrain an LM that understands general patterns?

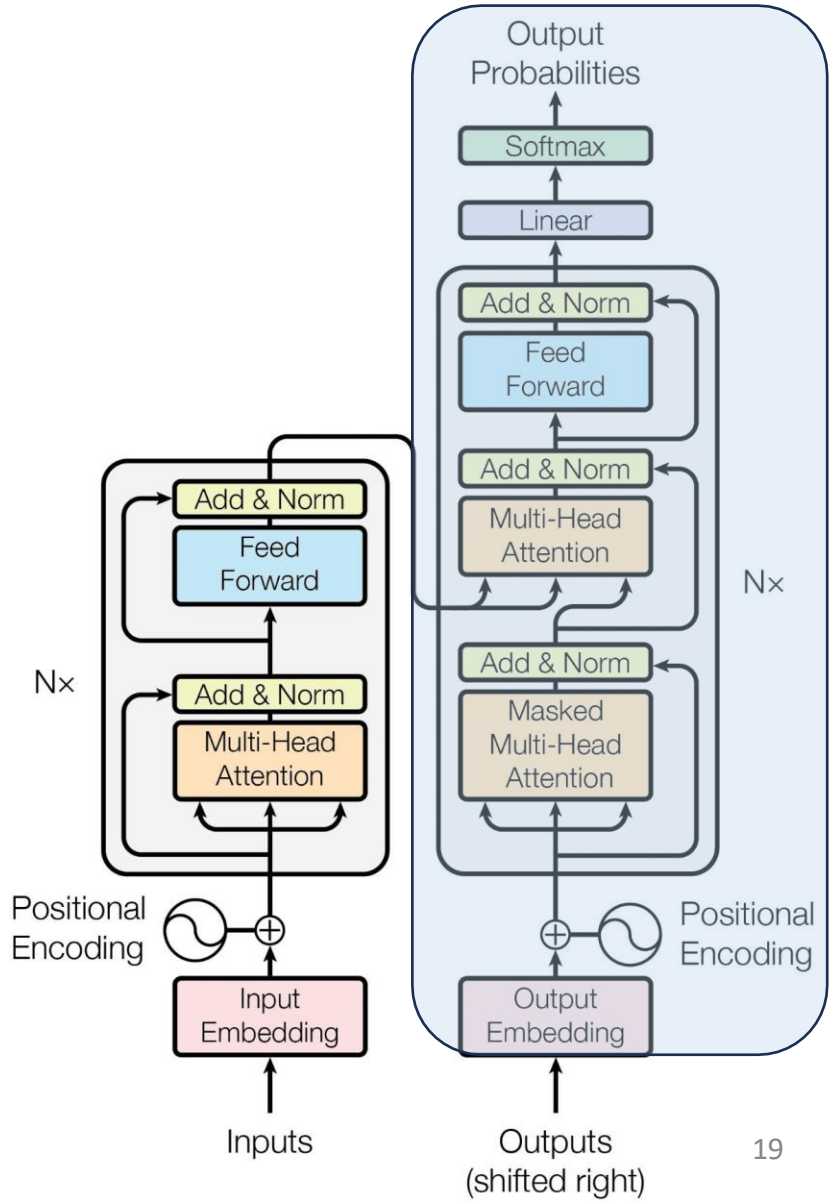


GPT – Generative Pretrained Transformer

GPT Pre-Training Corpus:
- Similarly, BooksCorpus and English Wikipedia

GPT Pre-Training Tasks:
- Predict the next token, given the previous tokens
  - More learning signals than MLM

GPT Pre-Training Results:
- GPT – 117M Params
  - Similarly competitive on GLUE and SQuAD



GPT – Generative Pretrained Transformer

GPT Fine-Tuning:

- Prompt-format task-specific text as a continuous stream for the model to fit

Summarization

Summarize this article:

XXXXXXXXXXXXXX

The summary is:

[Output]

QA

Answer the question based on the context.

Context:

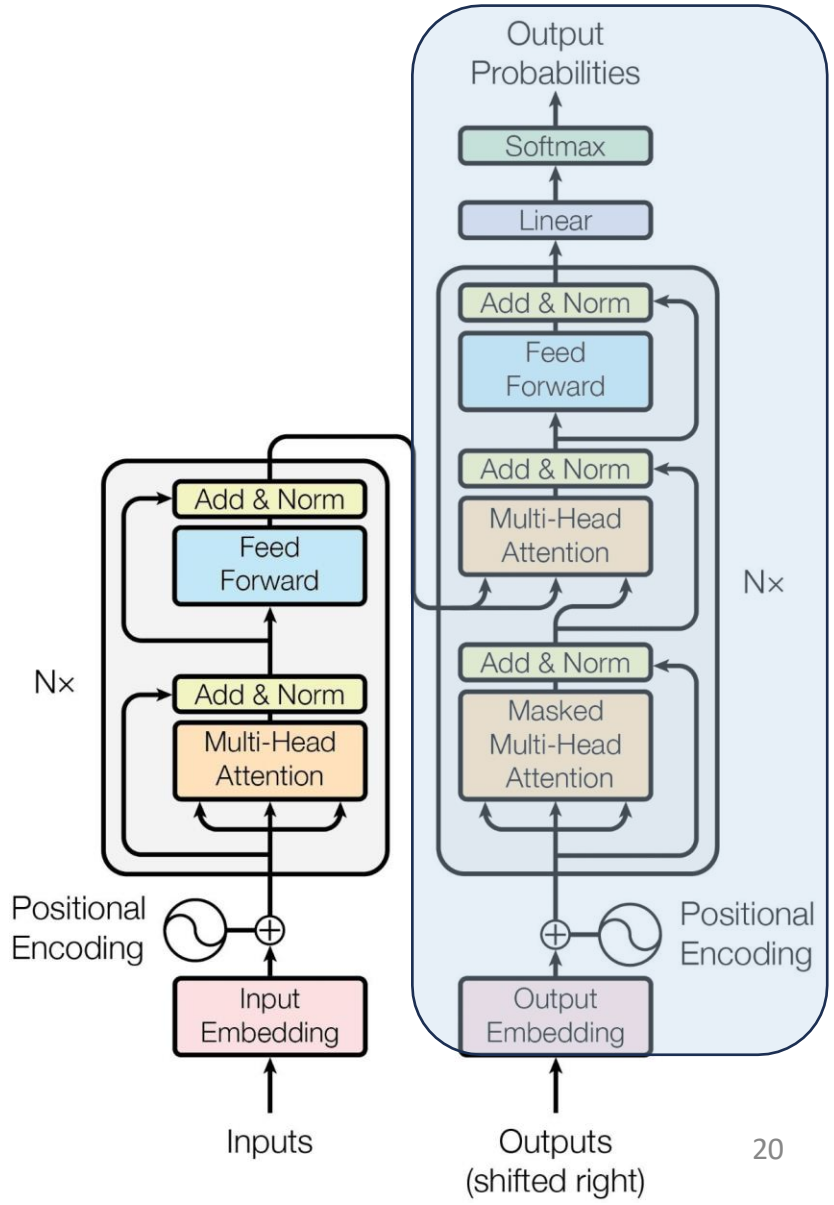
XXXXXXXXXXXXXXXXXXXX

Question:

XXXXXXXXXXX

Answer:

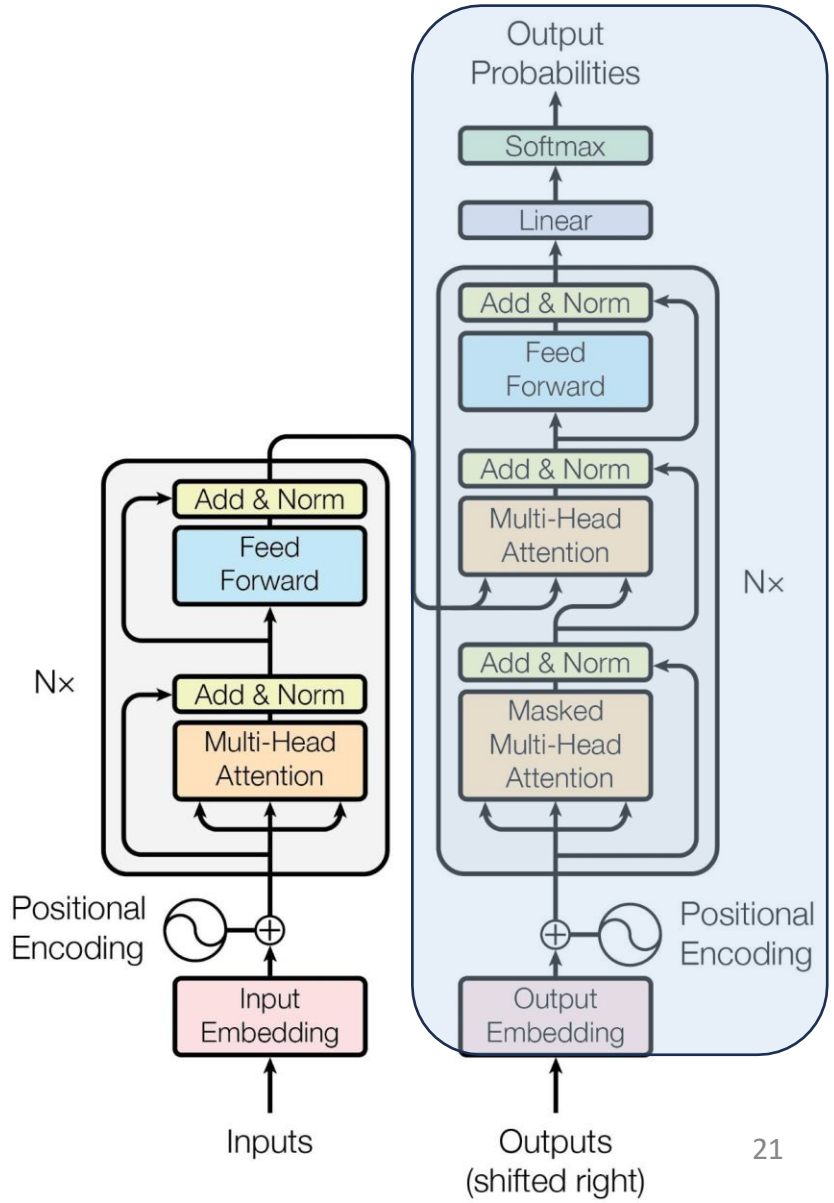
[Output]



GPT – Generative Pretrained Transformer

What is our takeaway from GPT?

- **The Effectiveness of Self-Supervised Learning**
  - Specifically, the model seems to be able to learn from generating the language *itself*, rather than from any specific task we might cook up.

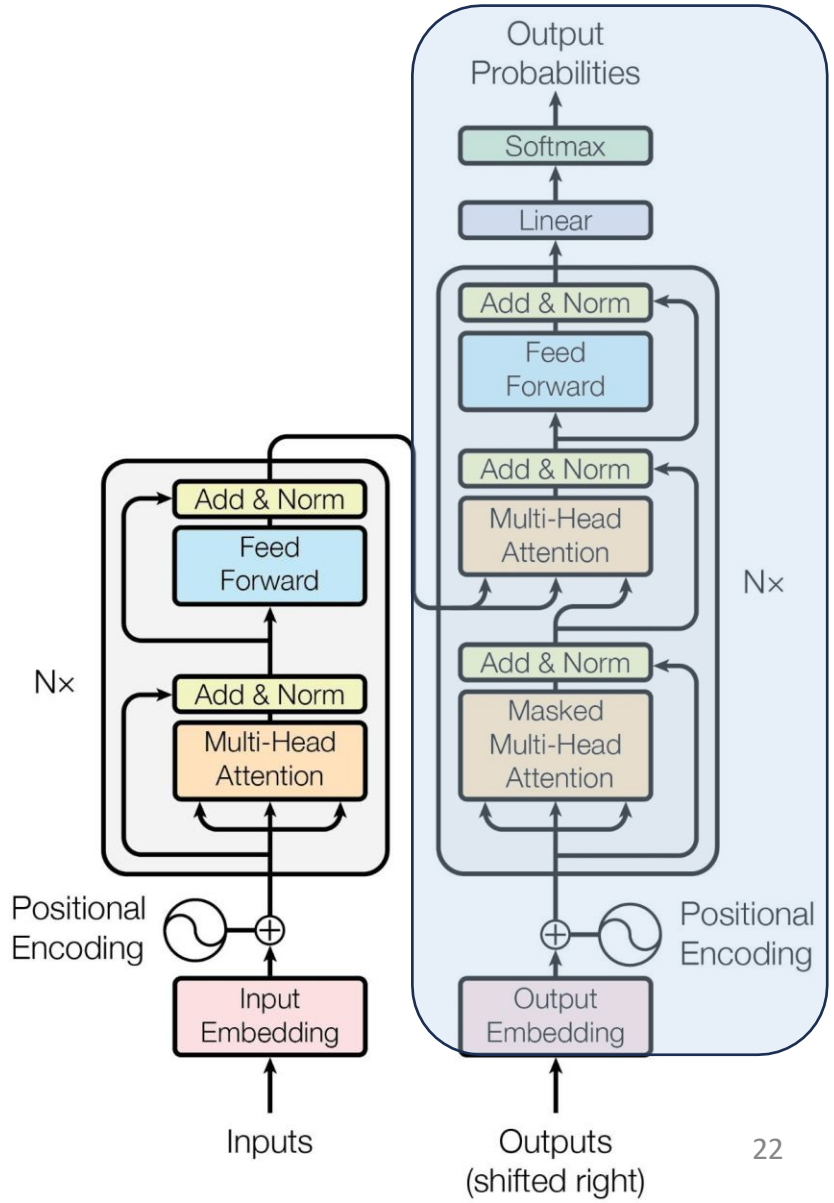


GPT – Generative Pretrained Transformer

What is our takeaway from GPT?

- **The Effectiveness of Self-Supervised Learning**
  - Specifically, the model seems to be able to learn from generating the language *itself*, rather than from any specific task we might cook up.

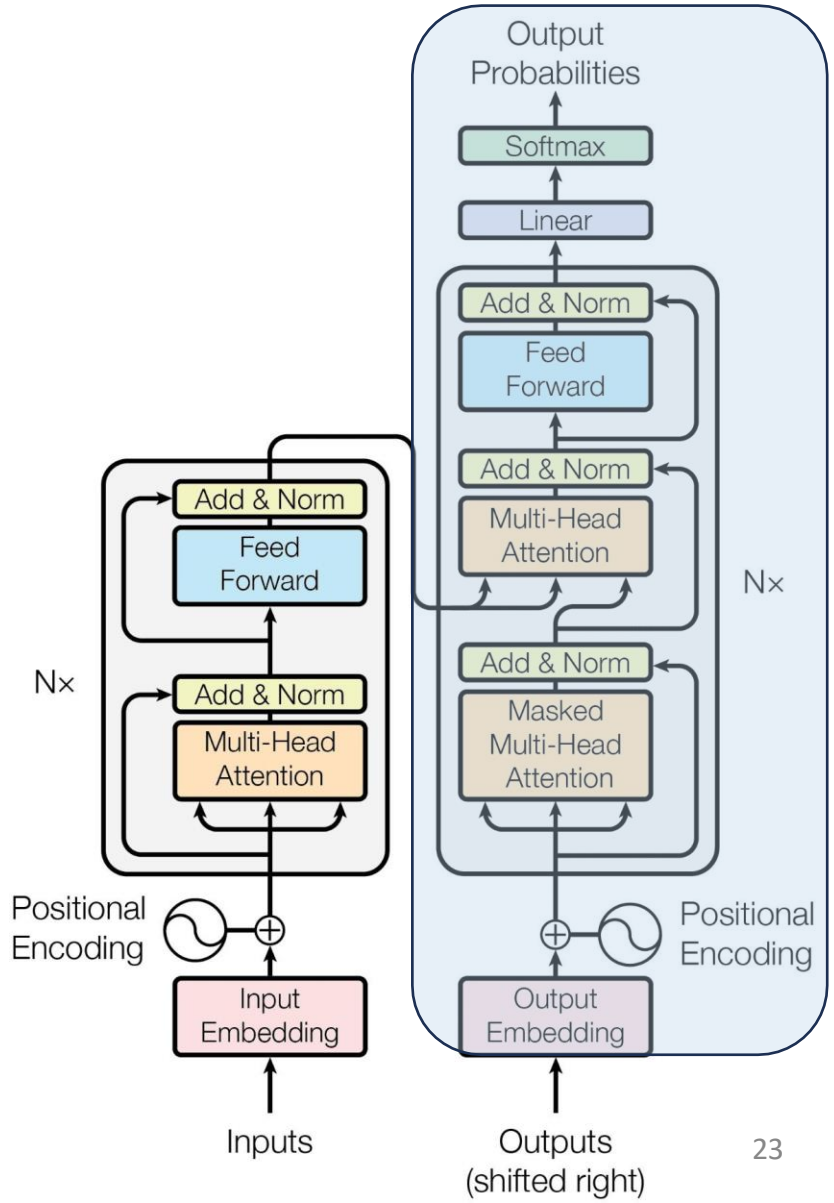
- **Language Model as a Knowledge Base**
  - Specifically, a generatively pretrained model seems to have a decent zero-shot performance on a range of NLP tasks.



GPT – Generative Pretrained Transformer

What is our takeaway from GPT?

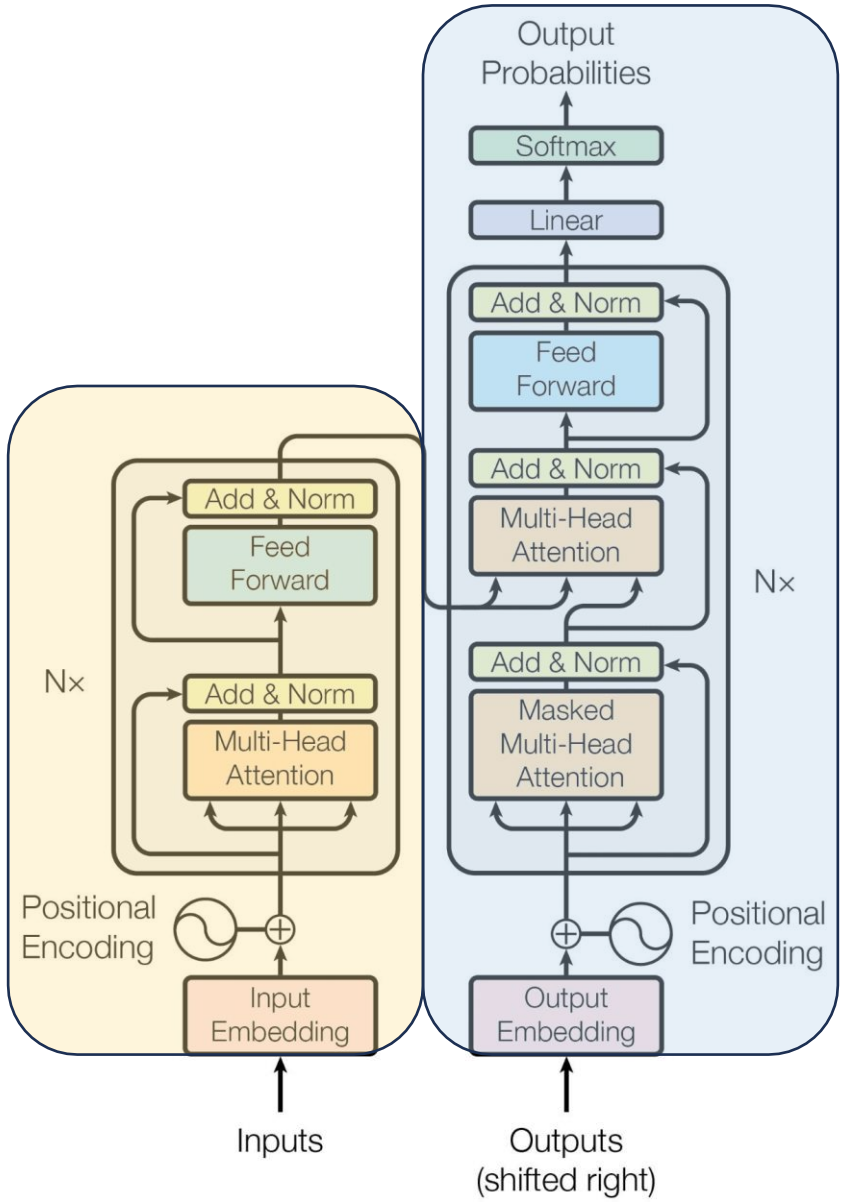
- **The Effectiveness of Self-Supervised Learning**
  - Specifically, the model seems to be able to learn from generating the language *itself*, rather than from any specific task we might cook up.
- **Language Model as a Knowledge Base**
  - Specifically, a generatively pretrained model seems to have a decent zero-shot performance on a range of NLP tasks.
- **And scaling works!!!**



The LLM Era – Paradigm Shift in Machine Learning

BERT
Oct 2018

Representation



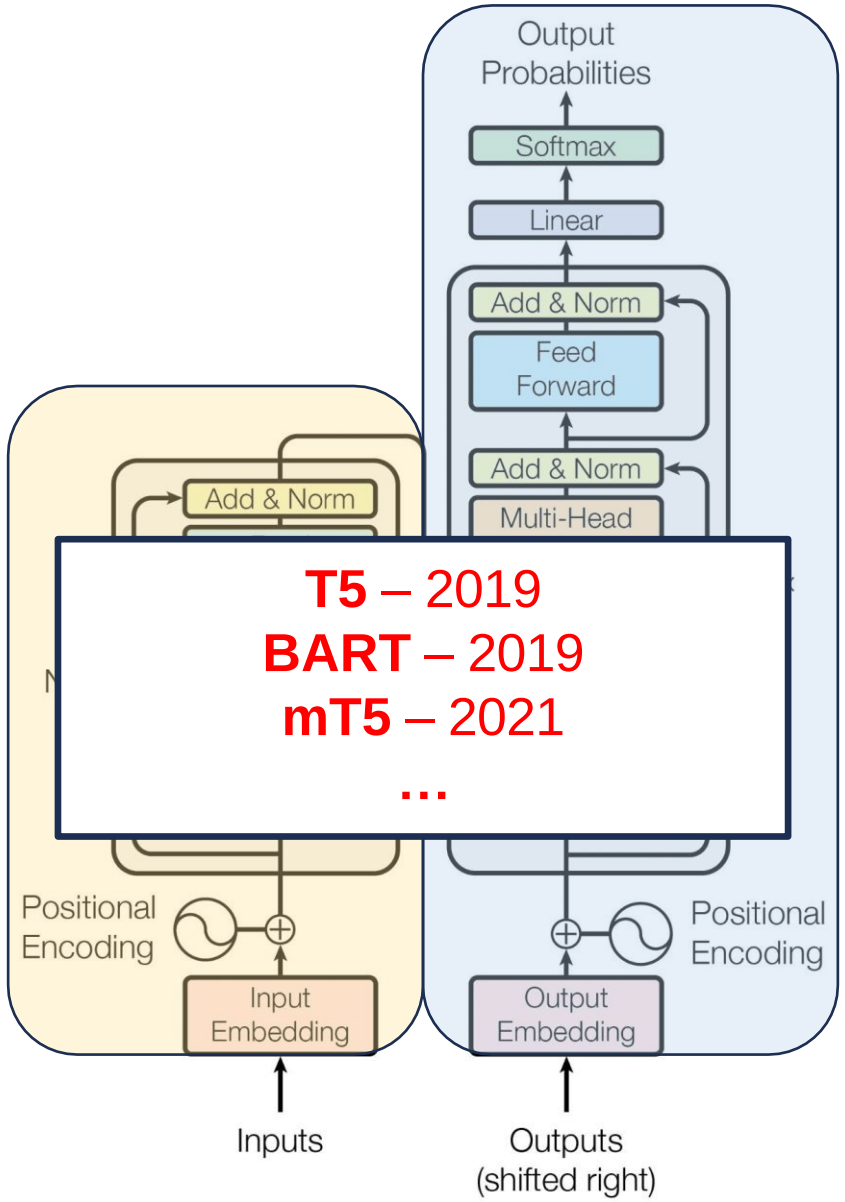
GPT
June 2018

Generation

The LLM Era – Paradigm Shift in Machine Learning

BERT – 2018
DistilBERT – 2019
RoBERTa – 2019
ALBERT – 2019
ELECTRA – 2020
DeBERTa – 2020
…

Representation



T5 – 2019
BART – 2019
mT5 – 2021
…

GPT – 2018
GPT-2 – 2019
GPT-3 – 2020
GPT-Neo – 2021
GPT-3.5 (ChatGPT) – 2022
LLaMA – 2023
GPT-4 – 2023
…

Generation

The LLM Era – Paradigm Shift in Machine Learning

From both BERT and GPT, we learn that…

- Transformers seem to provide a new class of generalist models that are capable of capturing knowledge which is more fundamental than task-specific abilities.

Before LLMs	Since LLMs
• **Feature Engineering** • How do we design or select the best features for a task?	

The LLM Era – Paradigm Shift in Machine Learning

From both BERT and GPT, we learn that…

- Transformers seem to provide a new class of generalist models that are capable of capturing knowledge which is more fundamental than task-specific abilities.

Before LLMs

- **Feature Engineering**
  - How do we design or select the best features for a task?
- **Model Selection**
  - Which model is best for which type of task?

Since LLMs

The LLM Era – Paradigm Shift in Machine Learning

From both BERT and GPT, we learn that…

- Transformers seem to provide a new class of generalist models that are capable of capturing knowledge which is more fundamental than task-specific abilities.

Before LLMs

- **Feature Engineering**
  - How do we design or select the best features for a task?
- **Model Selection**
  - Which model is best for which type of task?
- **Transfer Learning**
  - Given scarce labeled data, how do we transfer knowledge from other domains?

Since LLMs

The LLM Era – Paradigm Shift in Machine Learning

From both BERT and GPT, we learn that…

- Transformers seem to provide a new class of generalist models that are capable of capturing knowledge which is more fundamental than task-specific abilities.

Before LLMs

- **Feature Engineering**
  - How do we design or select the best features for a task?
- **Model Selection**
  - Which model is best for which type of task?
- **Transfer Learning**
  - Given scarce labeled data, how do we transfer knowledge from other domains?
- **Overfitting vs Generalization**
  - How do we balance complexity and capacity to prevent overfitting while maintaining good performance?

Since LLMs

The LLM Era – Paradigm Shift in Machine Learning

From both BERT and GPT, we learn that…

- Transformers seem to provide a new class of generalist models that are capable of capturing knowledge which is more fundamental than task-specific abilities.

Before LLMs

- **Feature Engineering**
  - How do we design or select the best features for a task?
- **Model Selection**
  - Which model is best for which type of task?
- **Transfer Learning**
  - Given scarce labeled data, how do we transfer knowledge from other domains?
- **Overfitting vs Generalization**
  - How do we balance complexity and capacity to prevent overfitting while maintaining good performance?

Since LLMs

- **Pre-training and Fine-tuning**
  - How do we leverage large scales of unlabeled data out there previously under-leveraged?

The LLM Era – Paradigm Shift in Machine Learning

From both BERT and GPT, we learn that…

- Transformers seem to provide a new class of generalist models that are capable of capturing knowledge which is more fundamental than task-specific abilities.

Before LLMs

- **Feature Engineering**
  - How do we design or select the best features for a task?
- **Model Selection**
  - Which model is best for which type of task?
- **Transfer Learning**
  - Given scarce labeled data, how do we transfer knowledge from other domains?
- **Overfitting vs Generalization**
  - How do we balance complexity and capacity to prevent overfitting while maintaining good performance?

Since LLMs

- **Pre-training and Fine-tuning**
  - How do we leverage large scales of unlabeled data out there previously under-leveraged?
- **Zero-shot and Few-shot learning**
  - How can we make models perform on tasks they are *not* trained on?

The LLM Era – Paradigm Shift in Machine Learning

From both BERT and GPT, we learn that…

- Transformers seem to provide a new class of generalist models that are capable of capturing knowledge which is more fundamental than task-specific abilities.

Before LLMs

- **Feature Engineering**
  - How do we design or select the best features for a task?
- **Model Selection**
  - Which model is best for which type of task?
- **Transfer Learning**
  - Given scarce labeled data, how do we transfer knowledge from other domains?
- **Overfitting vs Generalization**
  - How do we balance complexity and capacity to prevent overfitting while maintaining good performance?

Since LLMs

- **Pre-training and Fine-tuning**
  - How do we leverage large scales of unlabeled data out there previously under-leveraged?
- **Zero-shot and Few-shot learning**
  - How can we make models perform on tasks they are *not* trained on?
- **Prompting**
  - How do we make models understand their task simply by describing it in natural language?

The LLM Era – Paradigm Shift in Machine Learning

From both BERT and GPT, we learn that…

- Transformers seem to provide a new class of generalist models that are capable of capturing knowledge which is more fundamental than task-specific abilities.

Before LLMs

- **Feature Engineering**
  - How do we design or select the best features for a task?
- **Model Selection**
  - Which model is best for which type of task?
- **Transfer Learning**
  - Given scarce labeled data, how do we transfer knowledge from other domains?
- **Overfitting vs Generalization**
  - How do we balance complexity and capacity to prevent overfitting while maintaining good performance?

Since LLMs

- **Pre-training and Fine-tuning**
  - How do we leverage large scales of unlabeled data out there previously under-leveraged?
- **Zero-shot and Few-shot learning**
  - How can we make models perform on tasks they are *not* trained on?
- **Prompting**
  - How do we make models understand their task simply by describing it in natural language?
- **Interpretability and Explainability**
  - How can we *understand* the inner workings of our own models?

The LLM Era

- What are LLMs?
- Modern LLM Architecture
- LLM Training Procedure
- LLM Inference – Prompting, In-Context Learning and Chain of Thought
- Evaluating LLMs
- Multimodal LLMs

GPT 2 – Generalizing to Unseen Tasks

- LMs can be used for different tasks by pre-training a “base” model and then fine-tuning for the task(s) of interest

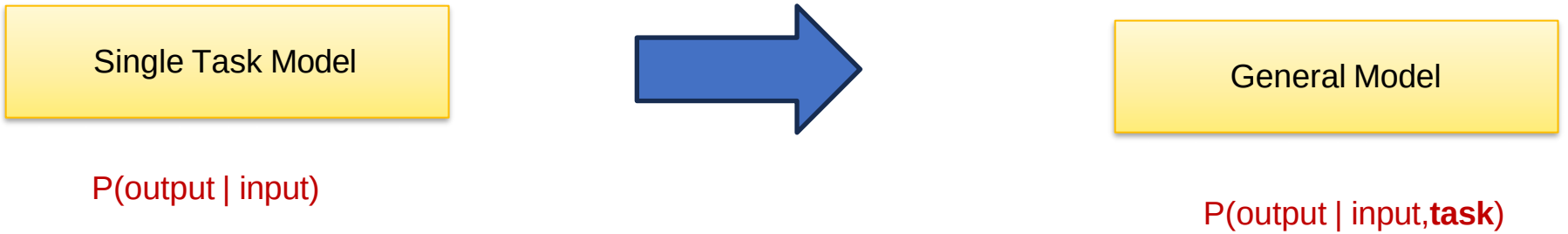
- Practical Issues:
  - Too many copies of the model
  - Need for large-scale labeled data for fine-tuning

- Multi-task Training?
  - Data remains a challenge
  - Humans don’t need such large volumes of data to learn – can we do better?

- Train a model that can perform NLP tasks in a zero-shot manner

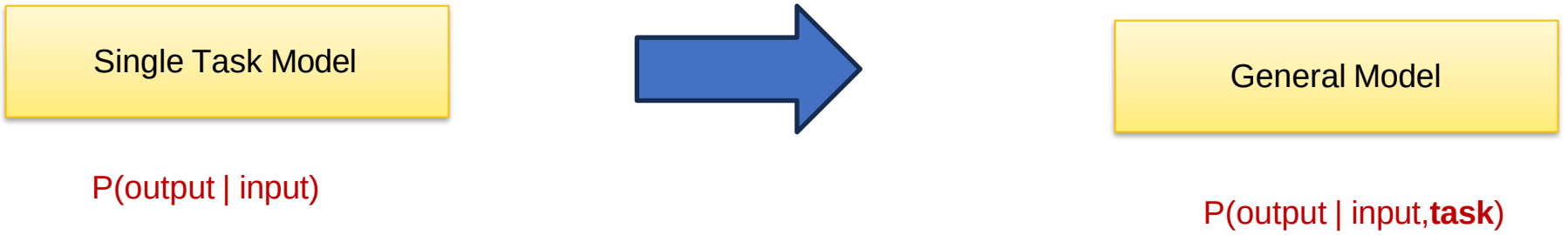
GPT 2 – Task Specifications

- Primary shift comes from modeling assumptions from single-task to general model



GPT 2 – Task Specifications

- Primary shift comes from modeling assumptions from single-task to general model



- Task descriptions may be provided as text – for example, translate this French text to English

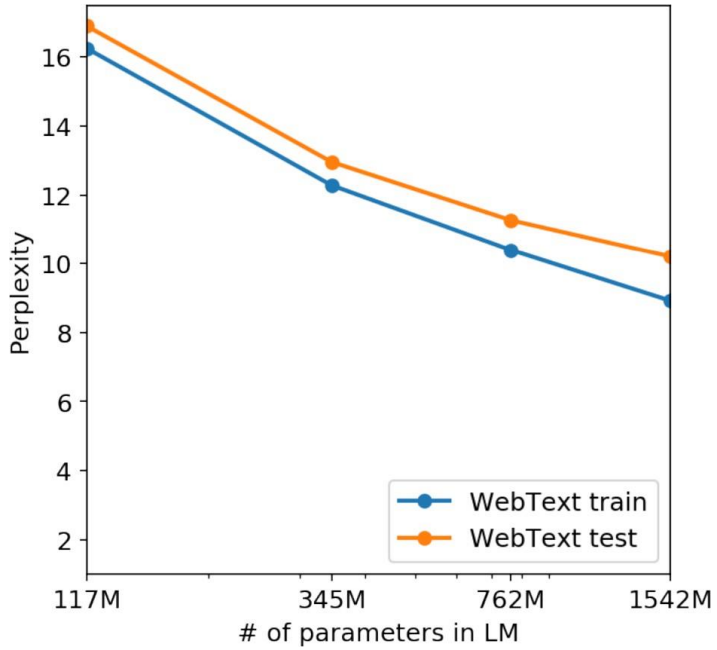
GPT 2 – what makes such an LM work ?

- Diverse training data
  - Model can do many disparate tasks with no training at all!
- Scaling model capacity and data

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

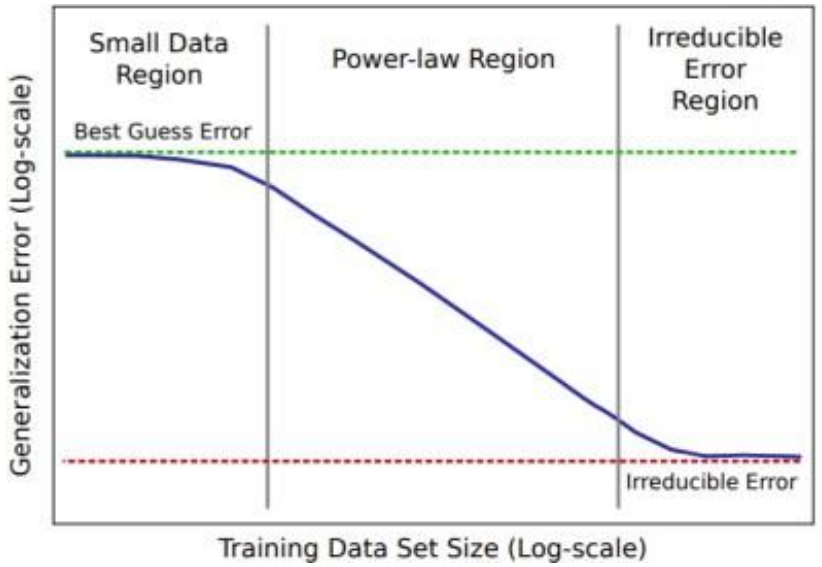
Scaling in GPT-2

- Scaling improves the perplexity of the LM and improves performance



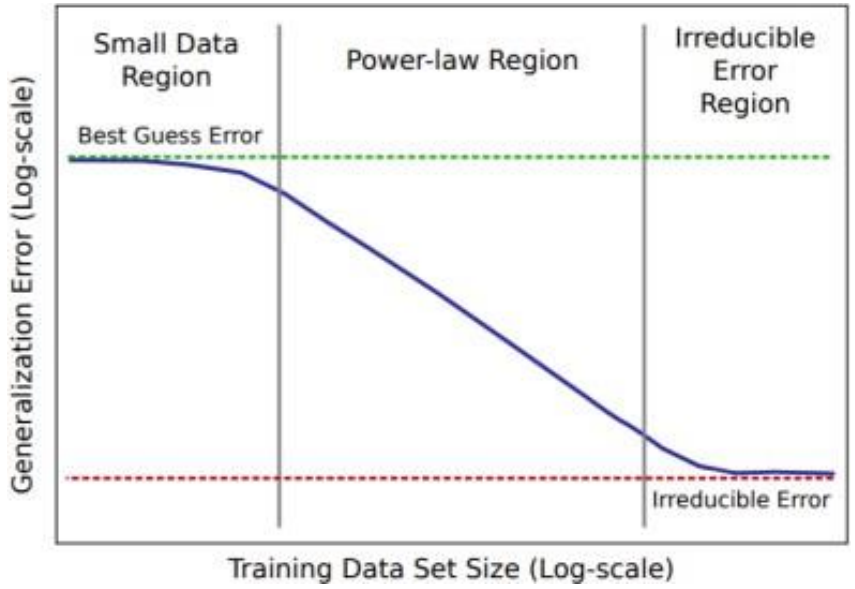
Why is this interesting? Look at data scaling

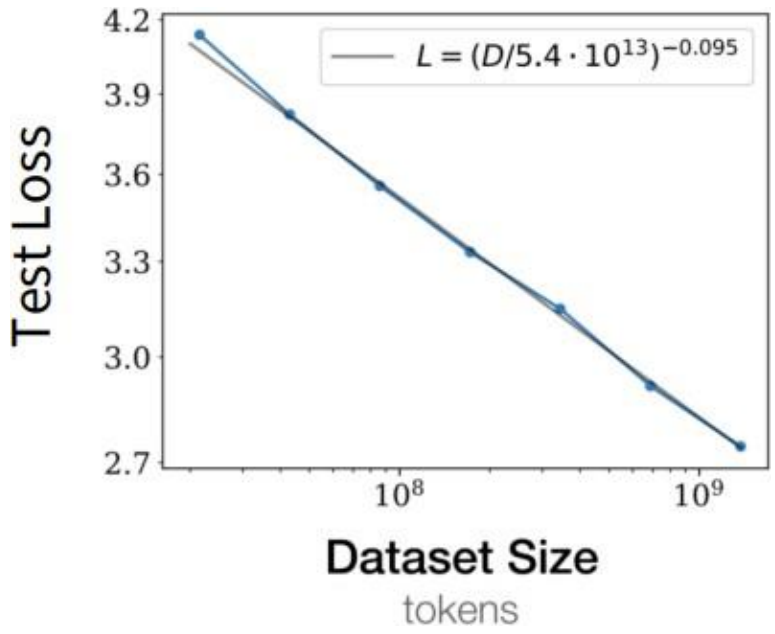
- We know that typical scaling effects look like this when we increase the amount of training data



Why is this interesting? Look at data scaling

- Loss and dataset size is linear on a log-log plot
- This is “power-law scaling”





Scaling - (Kaplan,2020)

- Can we understand scaling by positing scaling laws ?
- With scaling laws, we can make decisions on architecture, data, hyperparameters by training smaller models
- Open AI Study : **Scaling Laws for Neural Language Models** (Kaplan et al. 2020)

Scaling - (Kaplan,2020)

- Open AI Study : **Scaling Laws for Neural Language Models** (Kaplan et al. 2020)

- Key Findings:
  - Performance depends strongly on scale, and weakly on the model shape
    - Larger models are more sample-efficient
    - Smooth power laws

$$y = ax^k$$

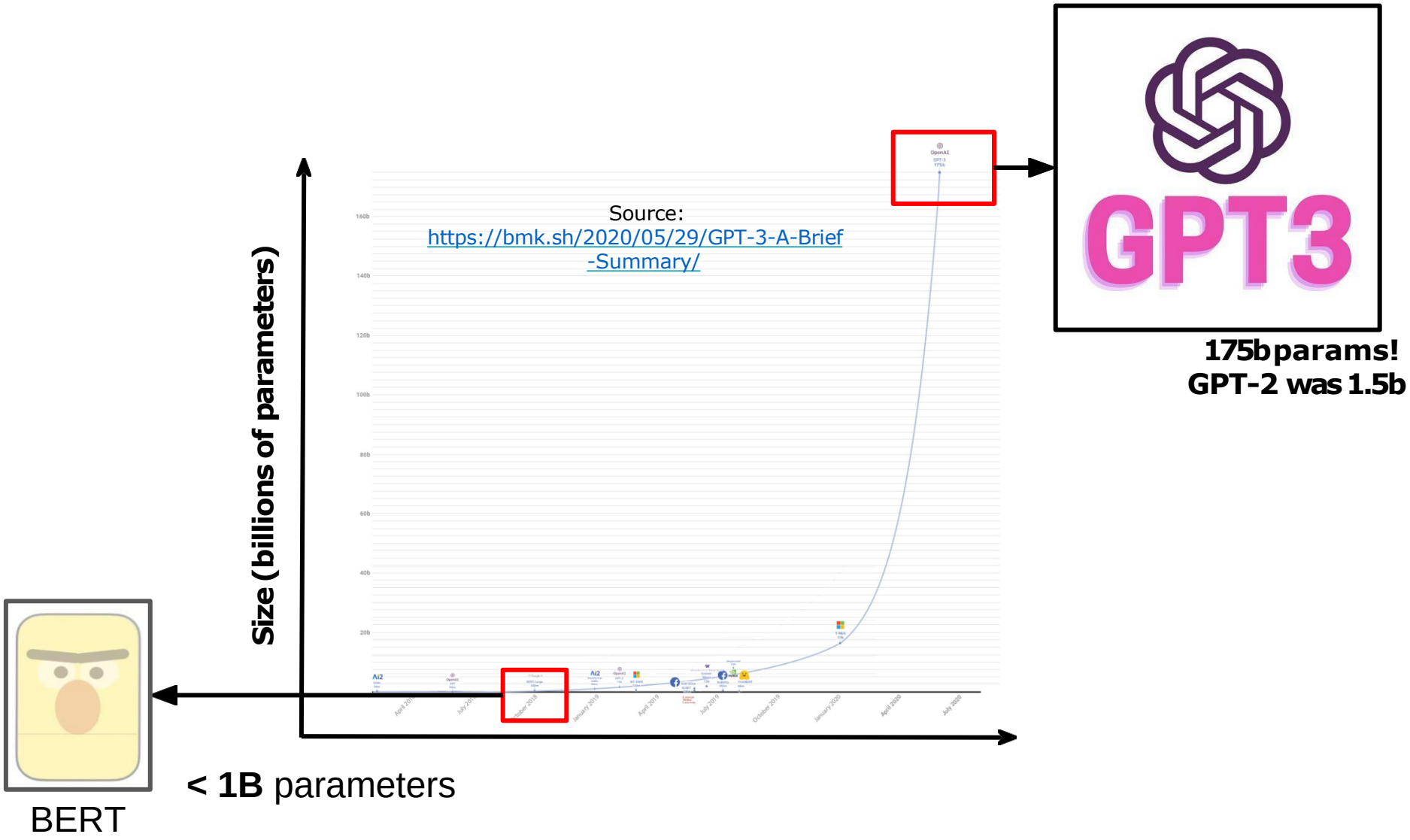
relationship between empirical performance y and x (one of N, D or C)
N - parameters, D - dataset size, C - compute

Scaling Effects

- The effect of some hyperparameters on big LMs can be predicted before training – optimizer (Adam v/s SGD), model depth, LSTM v/s Transformer

- Idea:
  - Train a few smaller models
  - Establish a scaling law (e.g. ADAM vs SGD scaling law)
  - Select optimal hyper param based on the scaling law prediction

Model Scaling: GPT-3



Emergent Abilities with GPT-3 – Wei et. al 2022

- Emergent abilities:
  - not present in smaller models but is present in larger models
  - Do LLMs like GPT3 have these ?

- Findings:
  - GPT-3 trained on text can do arithmetic problems like addition and subtraction
  - Different abilities “emerge” at different scales
  - Model scale is not the only contributor to emergence – for 14 BIG-Bench tasks, LaMDA 137B and GPT-3 175B models perform at near-random, but PaLM 62B achieves above-random performance
  - Problems LLMs can’t solve today may be emergent for future LLMs

Large Language Models

- Language models that have many parameters (over 1B) and can perform multiple tasks through prompting

- Eg. GPT, Llama2, Gemini, PaLM, Mistral, Mixtral etc.

LLM Realization - Architecture

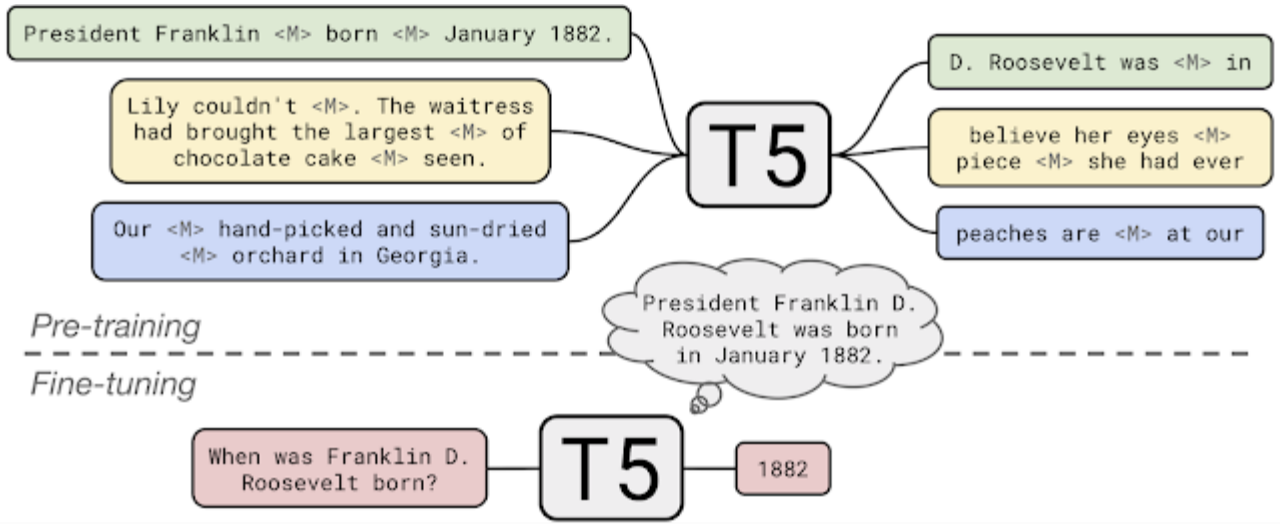
- ✓ Encoder-only (BERT)
  - Pre-training : Masked Language Modeling (MLM)
  - Great for classification tasks, but hard to do generation

- ✓ Decoder-only (GPT)
  - Pre-training: Auto-regressive Language Modeling
  - Stable training, faster convergence
  - Better generalization after pre-training

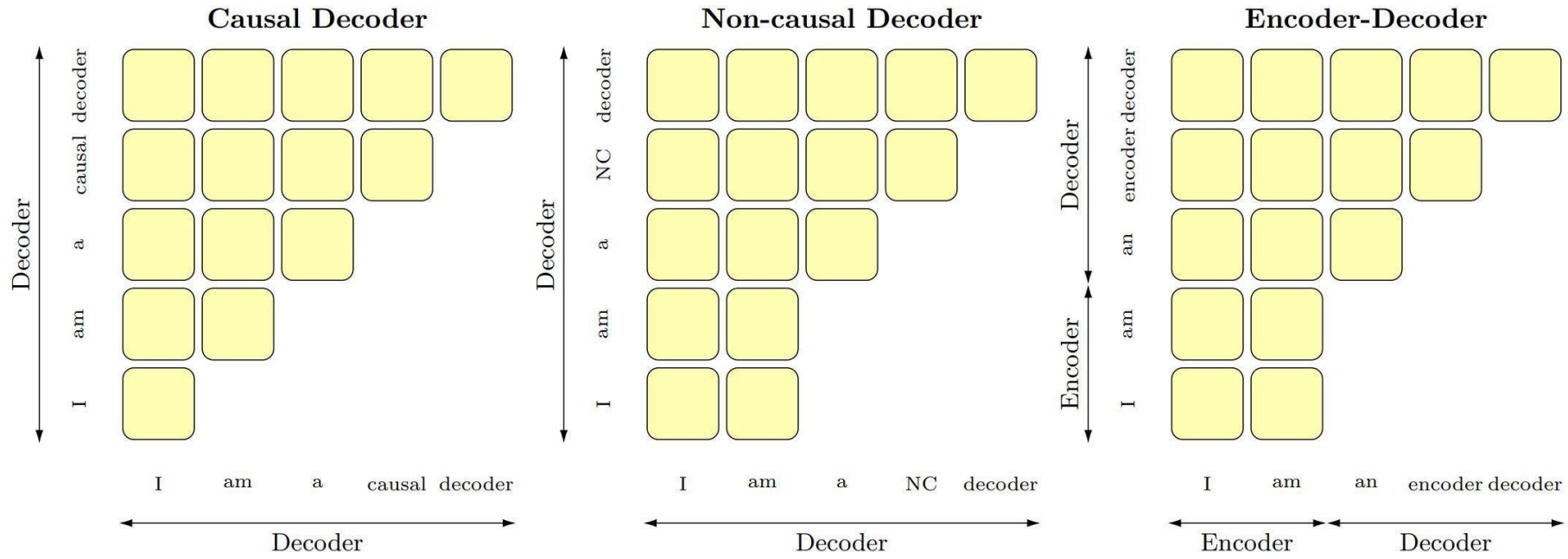
- Encoder-decoder (T0/T5)
  - Pre-training : Masked Span Prediction
  - Good for tasks like machine translation, summarization

T5/ T0 : Masked Span Prediction

- Masked span prediction involves:
  - Mask continuous set of tokens (span) in input
  - Predict this masked span from the decoder



Attention patterns (Wang et. al)



- Causal decoder -- each token attends to the previous tokens only.
- In both non-causal decoder and encoder-decoder, attention is allowed to be bidirectional on any conditioning information.
- For the encoder-decoder, that conditioning is fed into the encoder part of the model.

Empirical Observations (Wang et.al 2022)

- Decoder-only models outperform encoder-decoder models using similar configuration

	EAI-EVAL	T0-EVAL
Causal decoder	**44.2**	**42.4**
Non-causal decoder	43.5	41.8
Encoder-decoder	39.9	41.7
Random baseline	32.9	41.7

Llama 2 Architecture (Ouyang et. al. 2023)

- Decoder-only model
- Changes in transformer module:
  - Norm after sublayer -> Norm before sublayer
  - LayerNorm -> RMSNorm for stability
  - Activation: ReLU -> SwiGLU(x)
  - Position Embedding: Absolute/Relative -> RoPE (Rotary PE)
  - Long contexts : Multi-head attention -> Grouped-query attention

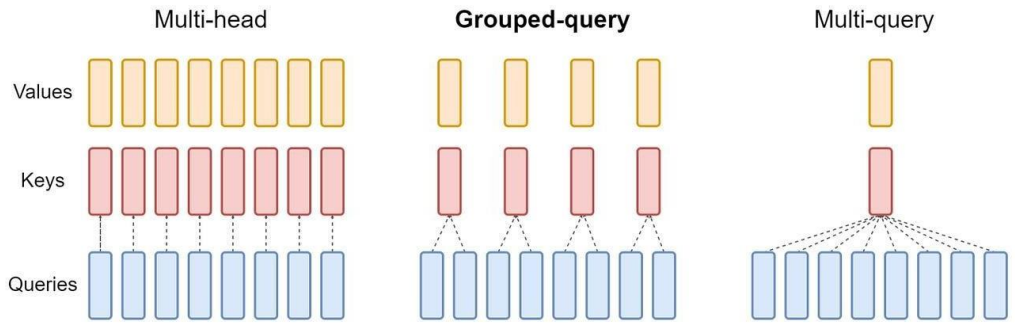


Figure 2: Overview of grouped-query method. Multi-head attention has H query, key, and value heads. Multi-query attention shares single key and value heads across all query heads. Grouped-query attention instead shares single key and value heads for each *group* of query heads, interpolating between multi-head and multi-query attention.

Training of Decoder-only LLMs – Llama 2

1. Auto-regressive Pre-training - Train to predict the next token on very large scale corpora (~3 trillion tokens)
2. Instruction Fine-tuning/ Supervised Fine-tuning (SFT) - Fine-tune the pre-trained model with pairs of (instruction+input,output) with large dataset and then with small high-quality dataset

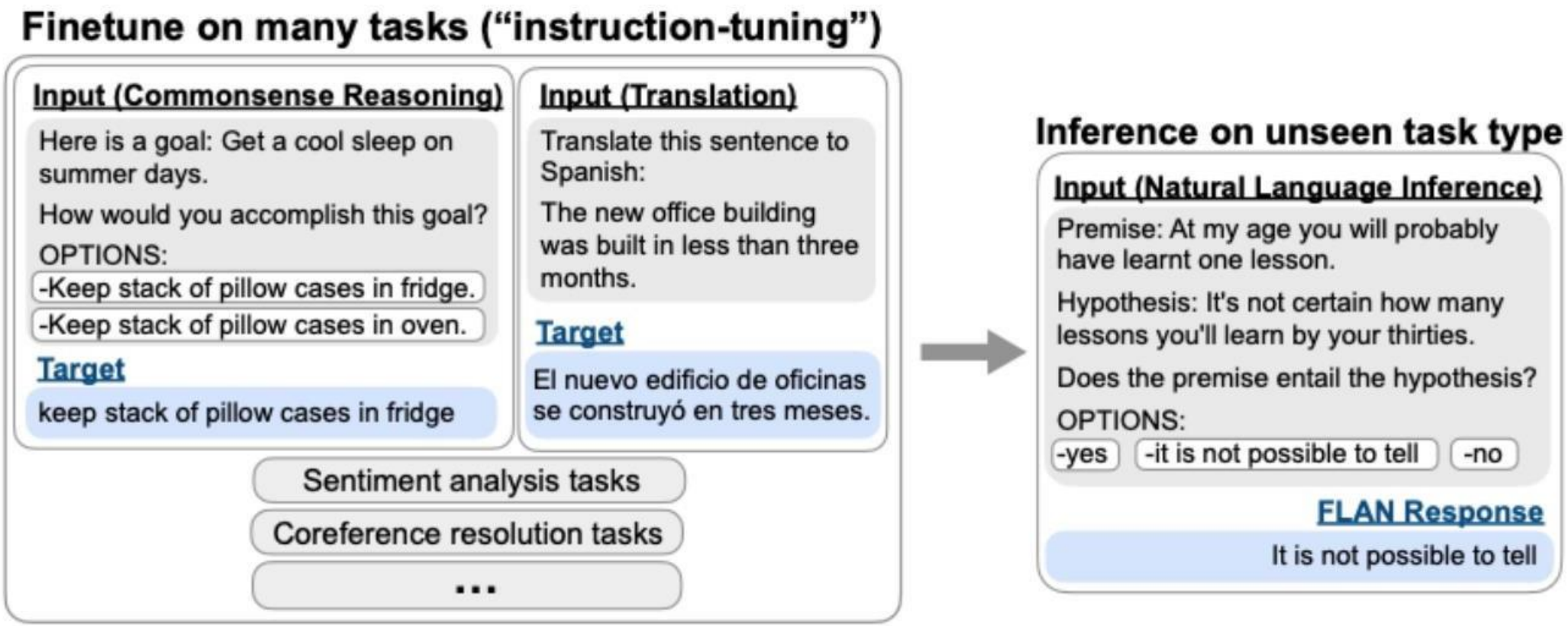
Instruction fine-tuning provides as a prefix a natural language description of the task along with the input.

- E.g. Translate into French this sentence: my name is -> je m’appelle

Supervised Fine-tuning versus Pre-training

- Objective function
  - Loss computed only for target tokens in SFT, all tokens are targets in pre-training
- Input and Target
  - Instruction + input as input with the target as output in SFT
  - only input as input with shifted input as target in pre-training
- Purpose
  - Pre-training makes good generalist auto-completes but good SFT builds models that can do many unseen tasks
  - SFT can also guide nature of outputs in terms of safety and helpfulness

Instruction Tuning (Wei et. al. 2021)



Unsafe Outputs – Alignment Problem

- LLMs may produce
  - Harmful text – unparliamentary language, bias and discrimination
  - Text that can cause direct harm – allowing easy access to dangerous information
- Therefore, LLMs should be trained to produce outputs that align with human preferences and values
- Modern LLMs do so by using SFT and by using human preference directly in model training

Training of Decoder-only LLMs – Llama 2

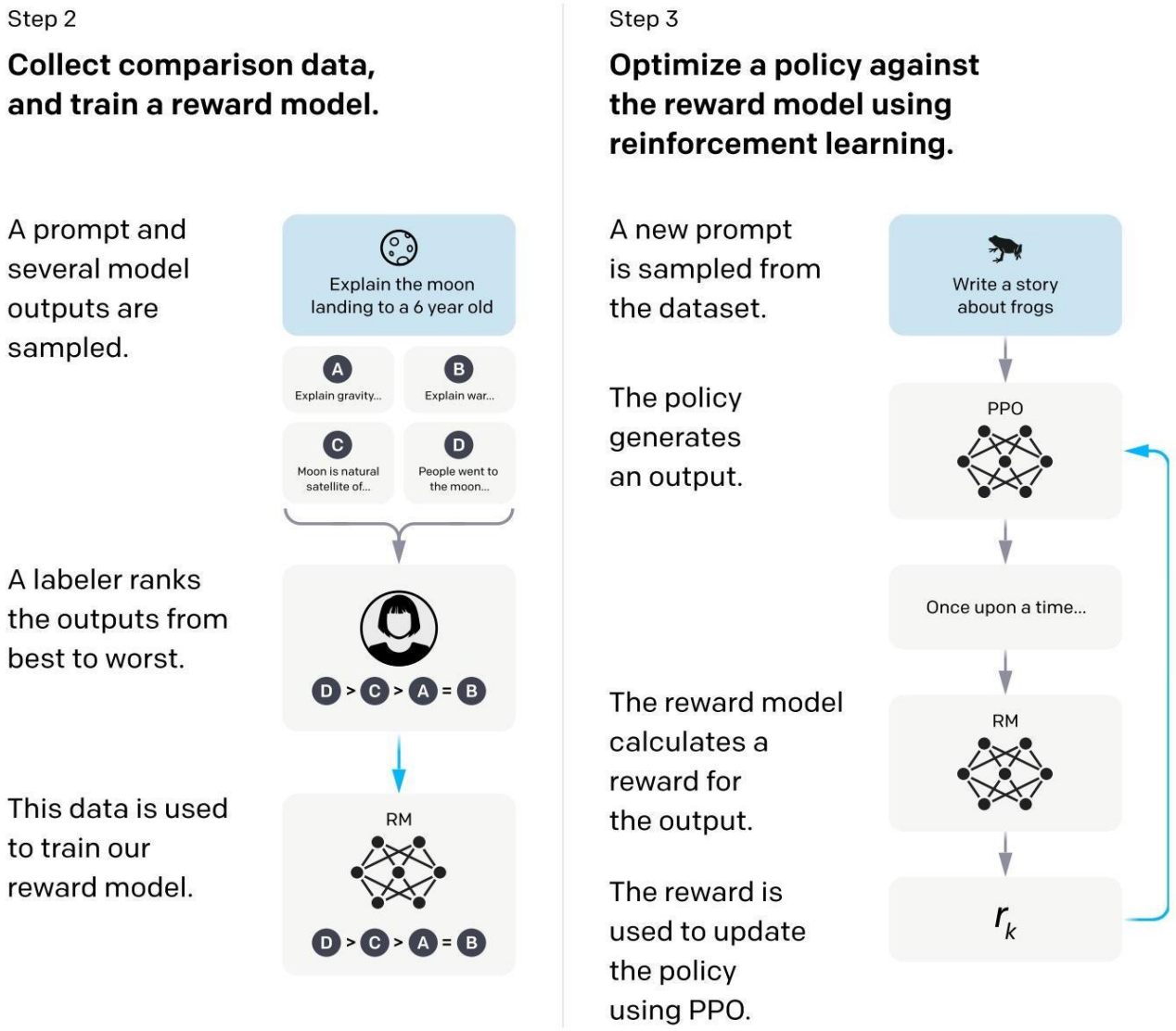
1. Auto-regressive Pre-training - Train to predict the next token on very large scale corpora (~3 trillion tokens)

2. Instruction Fine-tuning/ Supervised Fine-tuning (SFT) - Fine-tune the pre-trained model with pairs of (instruction+input,output) with large dataset and then with small high-quality dataset

3. Safety - Design a reward model based on human feedback and use policy gradient methods with the trained reward model to update LLM parameters so that outputs align with human values

   -- RLHF (Reinforcement-learning with Human Feedback)

RLHF



- However, these techniques, like RLHF, can’t guarantee LLMs will never produce harmful outputs

LLM Inference: Prompting

- Prompts
  - Tell the model what to do in natural language
  - For example, generate a textual summary of this paragraph:
  - Can be as short or long as required

- Prompt Engineering
  - The task of identifying the correct prompt needed to perform a task
  - General rule of thumb be as specific and descriptive as possible
  - Can be manual or automatic (prefix-tuning, paraphrasing etc.)

ChatGPT Prompt example

```
messages=[
  {
    "role": "system",
    "content": "You are an assistant that translates corporate jargon into plain English."
  },
  {
    "role": "system",
    "name":"example_user",
    "content": "New synergies will help drive top-line growth."
  },
  {
    "role": "system",
    "name": "example_assistant",
    "content": "Things working well together will increase revenue."
  },
  ...,
  {
    "role": "user",
    "content": "This late pivot means we don't have time to boil the ocean for the client deliverable."
  },
]
```

In-context learning/ Few-shot prompting (Brown,21)

- Provide a few examples along with the instruction

Instruction | Please classify movie reviews as 'positive' or 'negative'.

Examples |
```
Input: I really don't like this movie.
Output: negative

Input: This movie is great!
Output: positive
```

Chain of thought prompting (Wei, 2021)

- Get the model to work through the steps of the problem

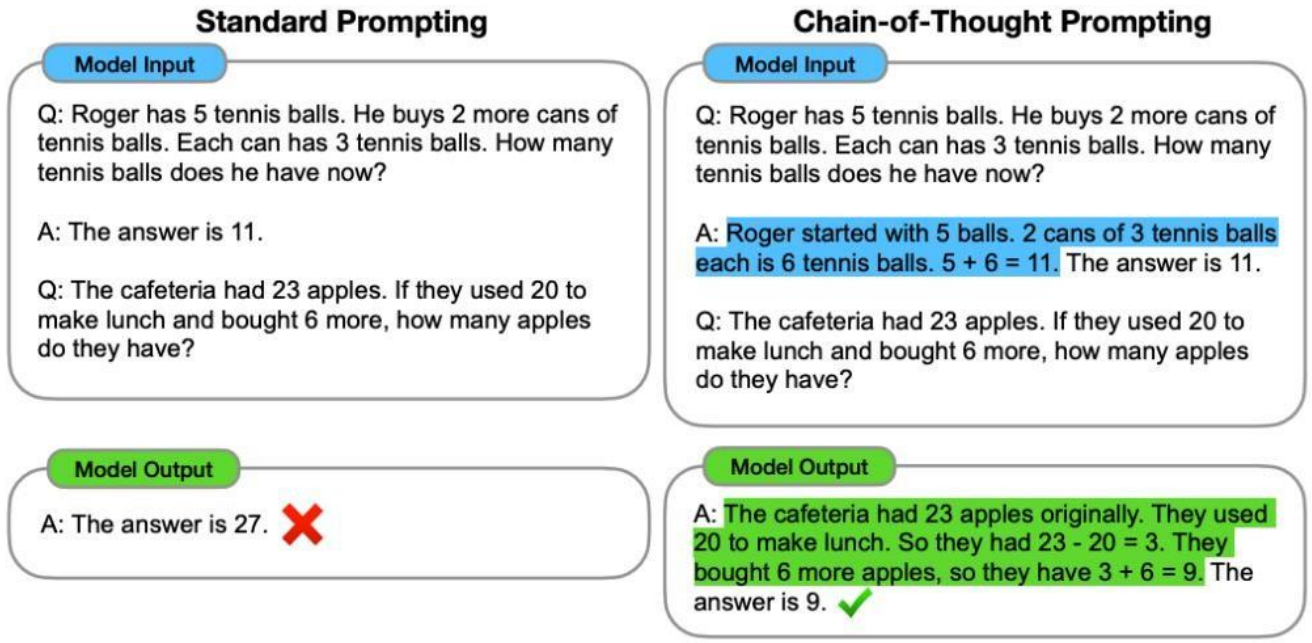
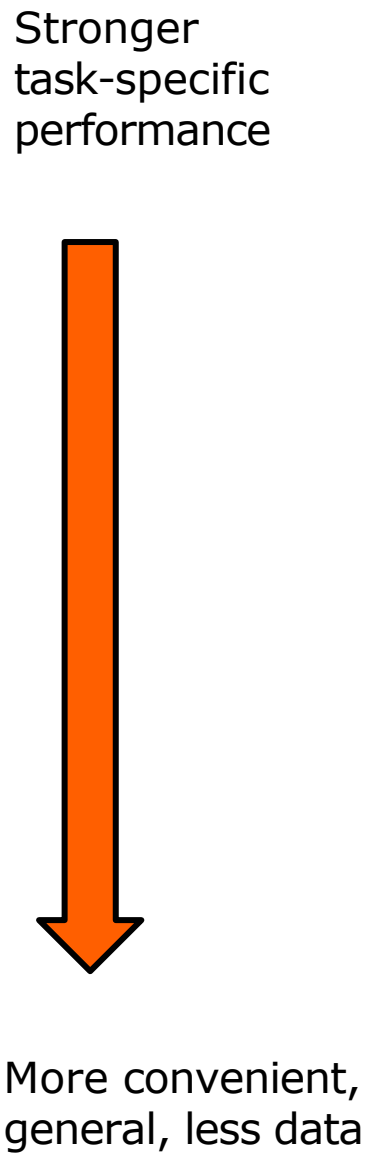


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

What to Pick?

1. Full Fine-tuning (FT)
   a. +Strongest performance
   b. - Need curated and labeled dataset for each new task (typically 1k-100k+ ex.)
   c. - Poor generalization, spurious feature exploitation
2. Few-shot (FS)
   a. +Much less task-specific data needed
   b. +No spurious feature exploitation
   c. - Challenging
3. One-shot (1S)
   a. +"Most natural," e.g. giving humans instructions
   b. - Challenging
4. Zero-shot (0S)
   a. +Most convenient
   b. - Challenging, can be ambiguous



Note on Parameter Efficient Fine-tuning

- When we don’t have large enough data for SFT
  - Freeze the LM and keep some parameters trainable
  - Add an external adapter module to adapt model parameters to the task
  - Perform Low-rank Adaptation (LoRA)

Evaluating LLMs

- Evaluation is challenging
  - Evaluate on as many datasets and tasks as possible

Benchmark (shots)	GPT-3.5	GPT-4	PaLM	PaLM-2-L	LLAMA 2
MMLU (5-shot)	70.0	**86.4**	69.3	78.3	68.9
TriviaQA (1-shot)	–	–	81.4	**86.1**	85.0
Natural Questions (1-shot)	–	–	29.3	**37.5**	33.0
GSM8K (8-shot)	57.1	**92.0**	56.5	80.7	56.8
HumanEval (0-shot)	48.1	**67.0**	26.2	–	29.9
BIG-Bench Hard (3-shot)	–	–	52.3	**65.7**	51.2

Table 4: Comparison to closed-source models on academic benchmarks. Results for GPT-3.5 and GPT-4 are from OpenAI (2023). Results for the PaLM model are from Chowdhery et al. (2022). Results for the PaLM-2-L are from Anil et al. (2023).

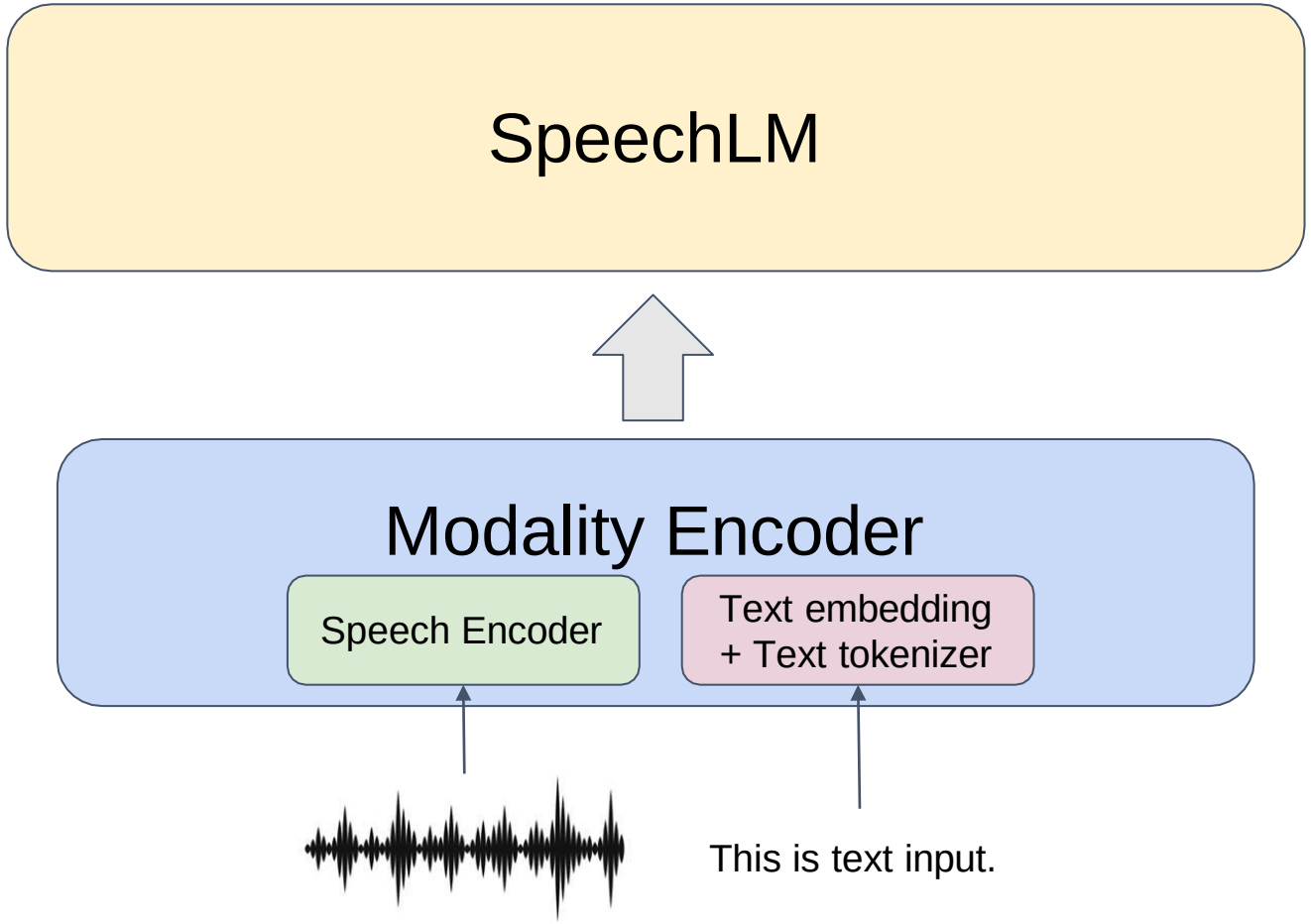
Multimodal LLMs

- Text is only part of the picture
  - We want LLMs that can understand the world by seeing and listening as well
  - Models should be able to do cross-modal reasoning and learning

- Multimodality can be introduced
  - From pre-training: Gemini
  - From instruction-tuning: AudioGPT, Flamingo

Modelling data using continuous representations

- Using continuous speech representations
  - Pros
    - Rich information
    - Good performance
  - Cons
    - Computationally heavy
    - Storage heavy

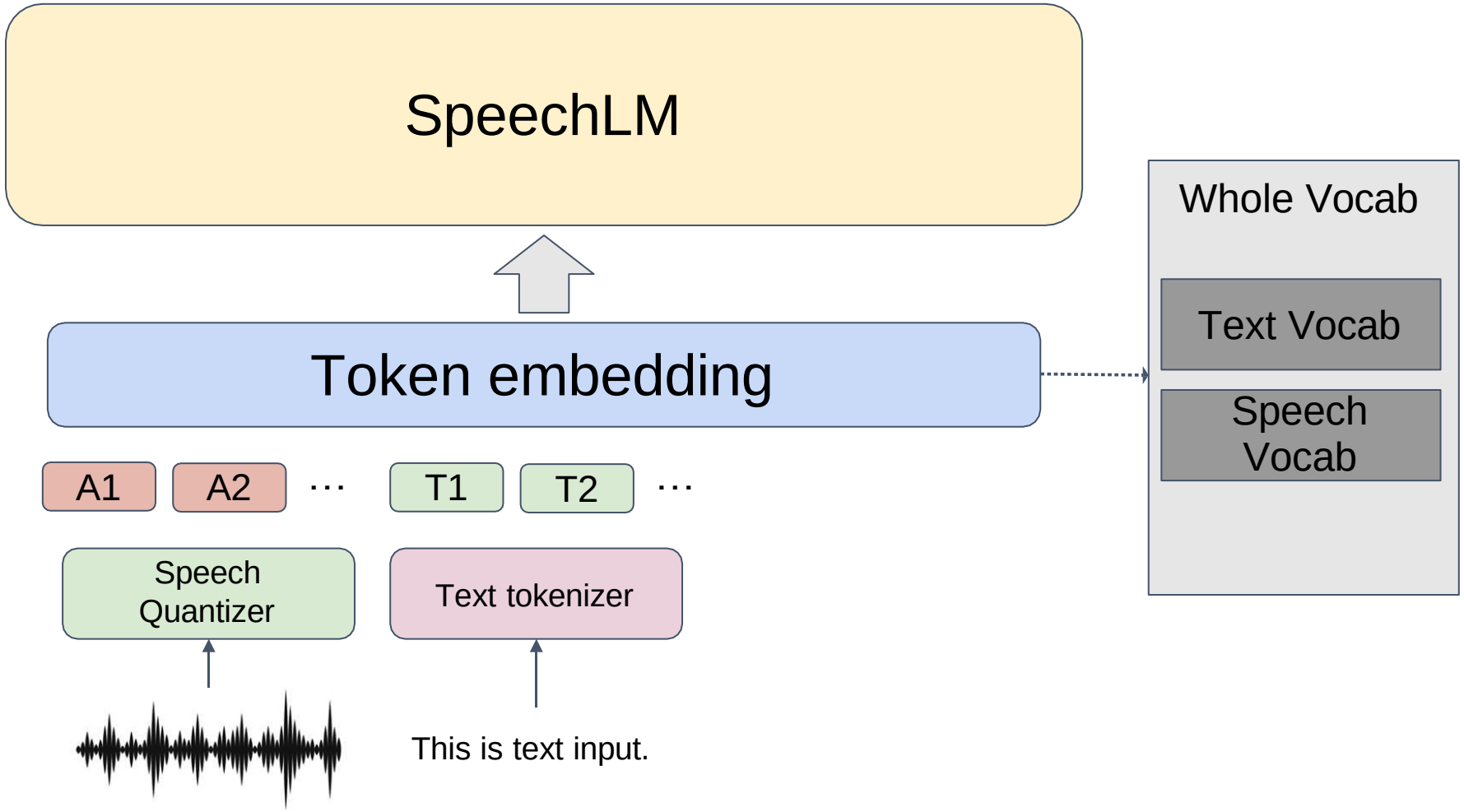


Modeling data using Discrete Unit

- Recently discrete units shows promising performance and benefit

Chang, Xuankai, et al. "Exploring Speech Recognition, Translation, and Understanding with Discrete Speech Units: A Comparative Study."
arXiv preprint arXiv:2309.15800 (2023).

- Storage decreases
- Sequence length (> 50% reduction)
- Performance is okay



Multimodal LLMs – Representing Images

- Continuous embeddings
  - concatenated with the embeddings of text inputs to LLMs
  - Pre-trained independently
  - Ex: CLIP

- Discrete representations
  - Extracted from self-supervised audio models like VQ-VAEs

Open Challenges - LLMs

- New Capabilities
  - Multimodal
  - Multi-lingual
  - More Complex Tasks
- Performance
  - Reduce Hallucinations
  - Improve Alignment with Human Preference
  - Increase Context Length Efficiently
  - Improve Data, Training Strategy, and Model Architecture
- Efficiency
  - Computational cost, time, and money
  - Compute architecture – GPU/ TPU/ HPU
- Safety
  - Reduce Harm
  - Improve Adversarial Robustness
  - Privacy Concerns
- Interpretability
  - Why do LLMs do what they do?