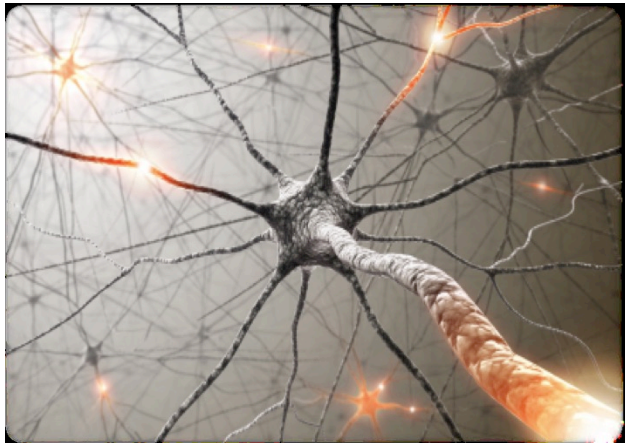
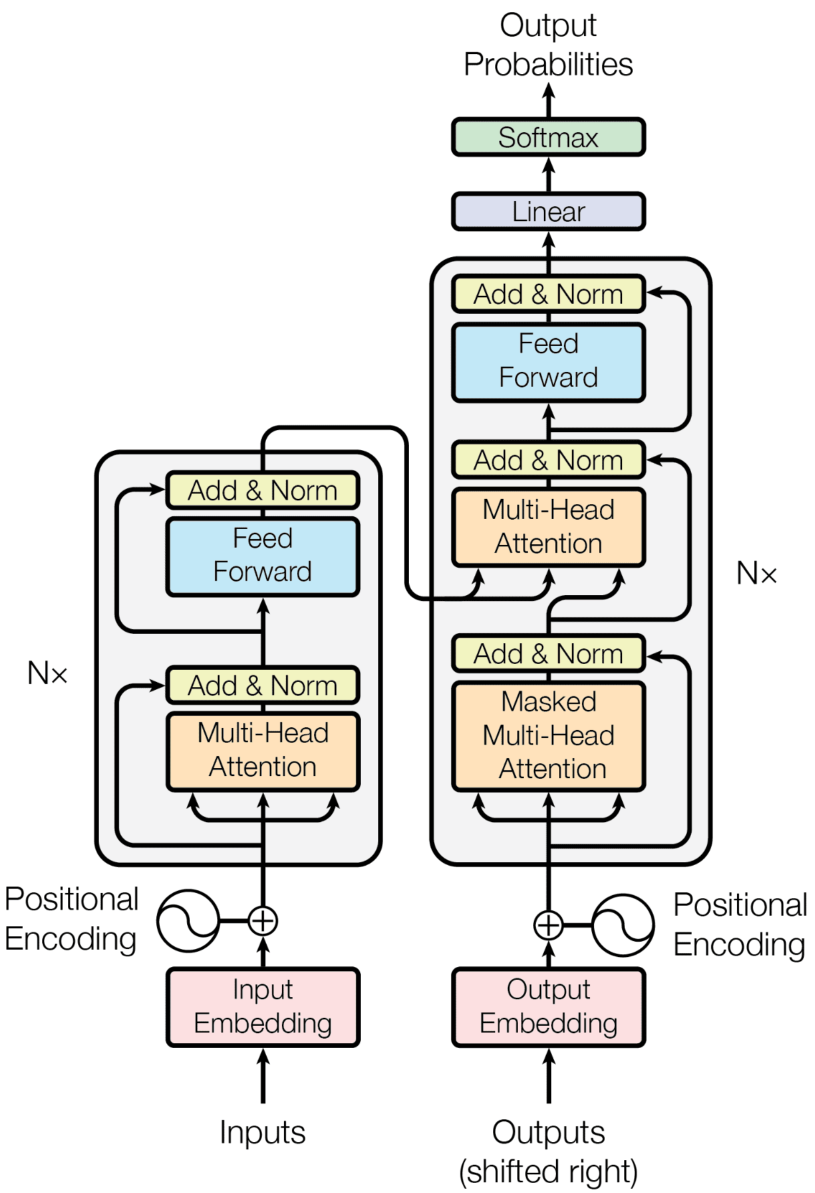
06 LLMs (Large Language Models)

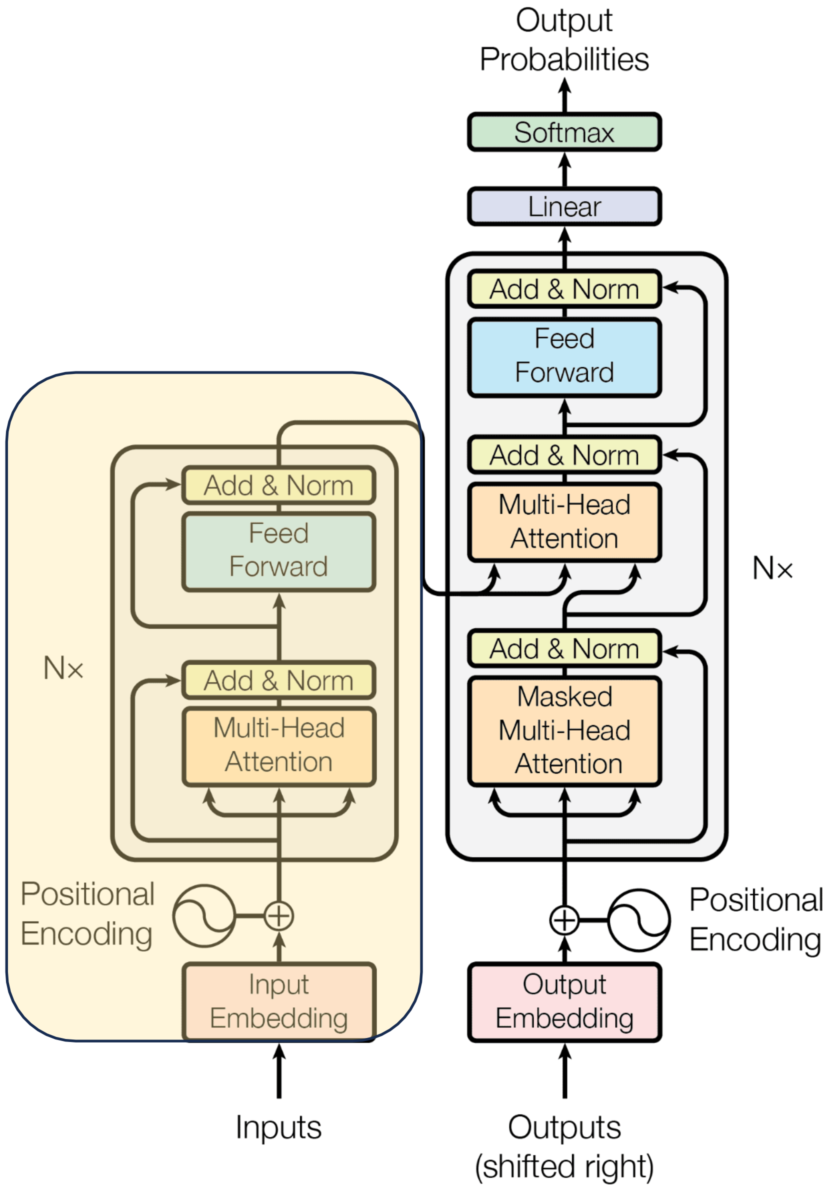


Transformers, mid-2017



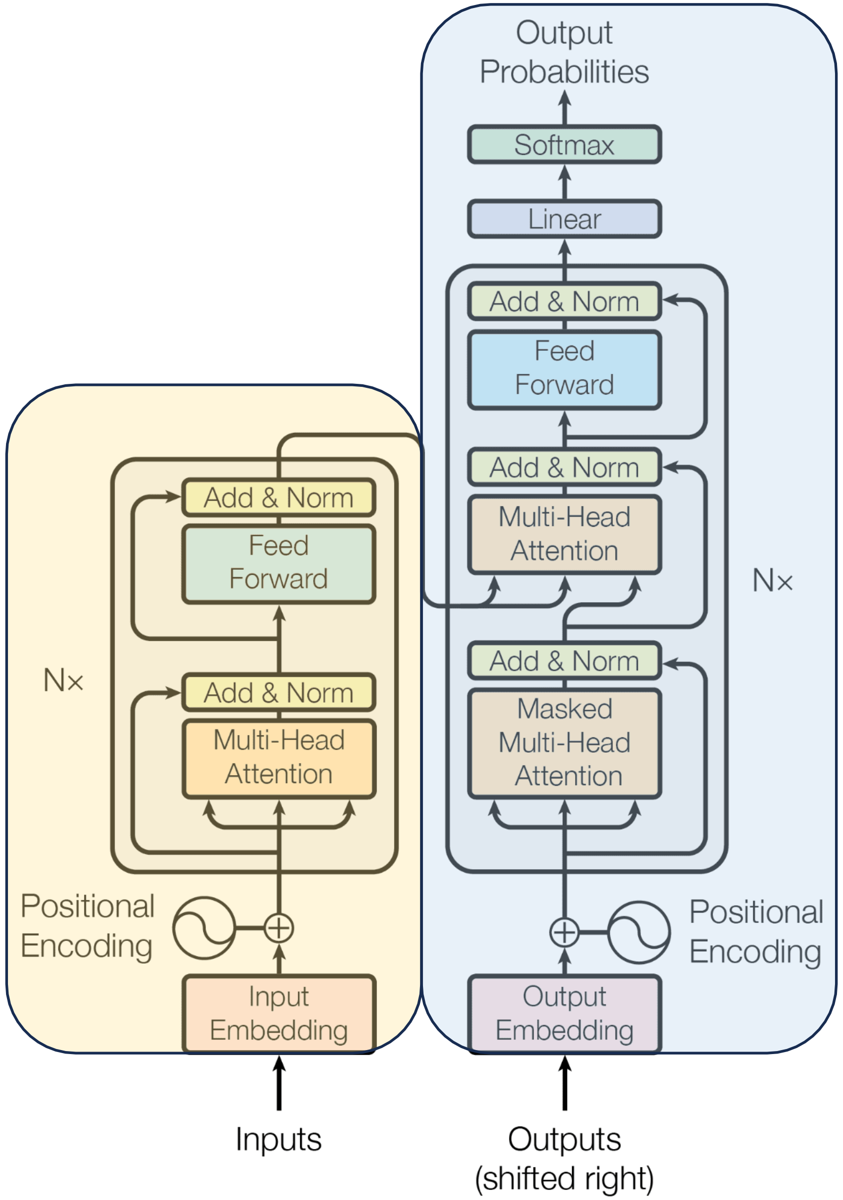
Transformers, mid-2017

Representation



Transformers, mid-2017

Representation

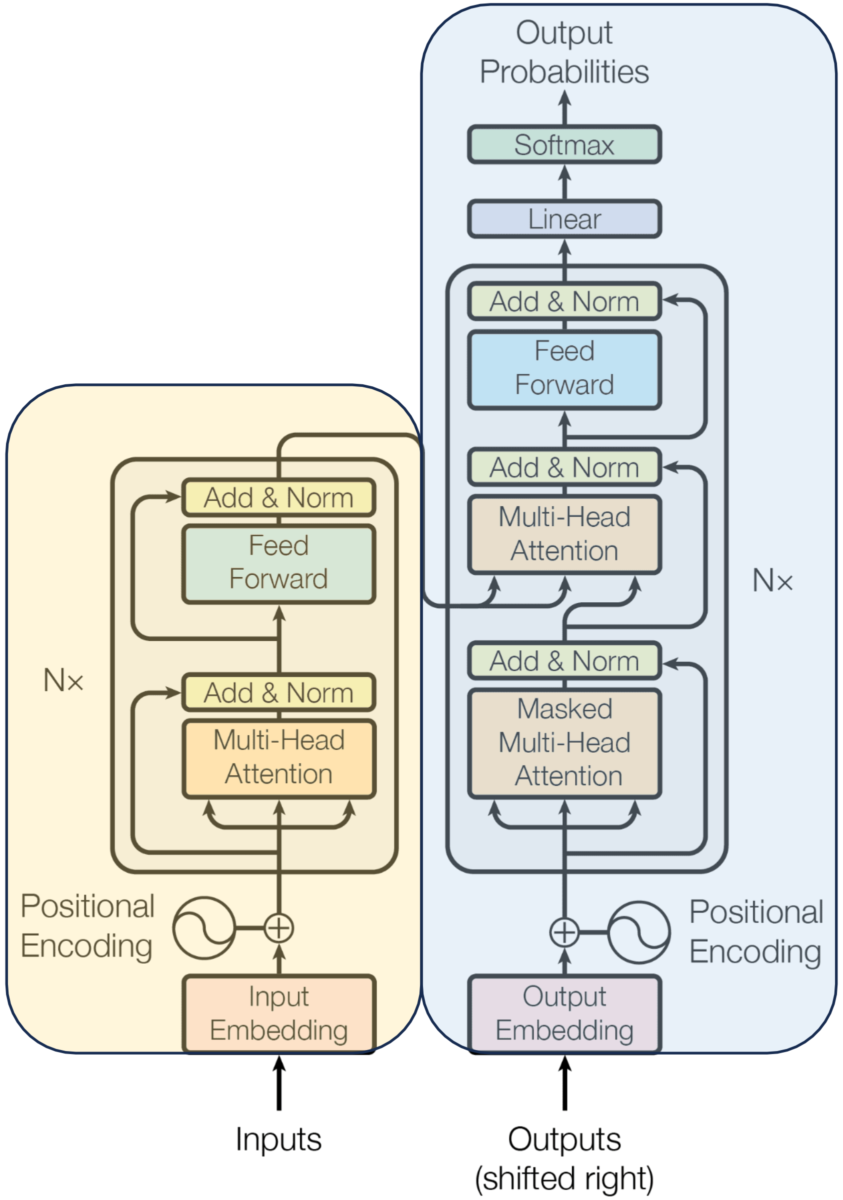


Generation

Transformers, mid-2017

Input – input tokens
Output – hidden states

Representation



Input – output tokens and hidden states*
Output – output tokens

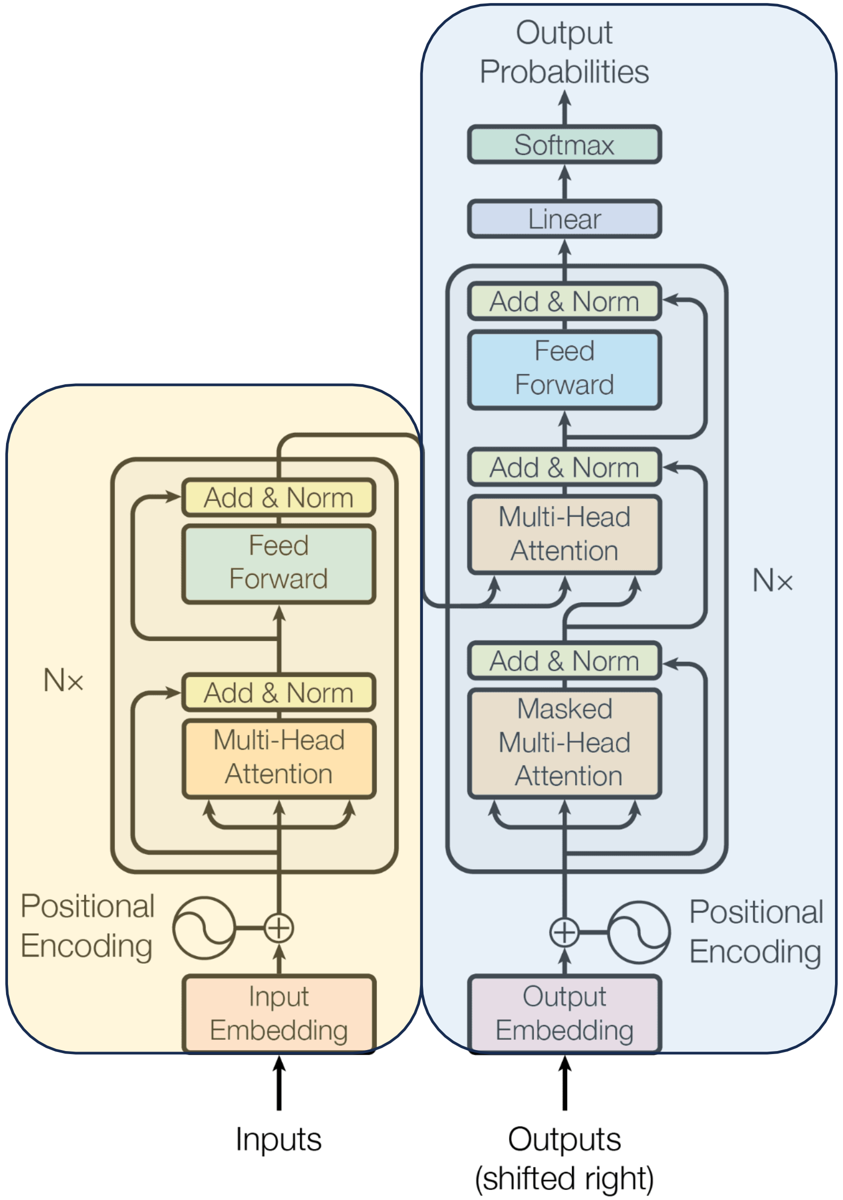
Generation

Transformers, mid-2017

Input – input tokens
Output – hidden states

Model can see all timesteps

Representation



Input – output tokens and hidden states*
Output – output tokens

Model can only see previous timesteps

Generation

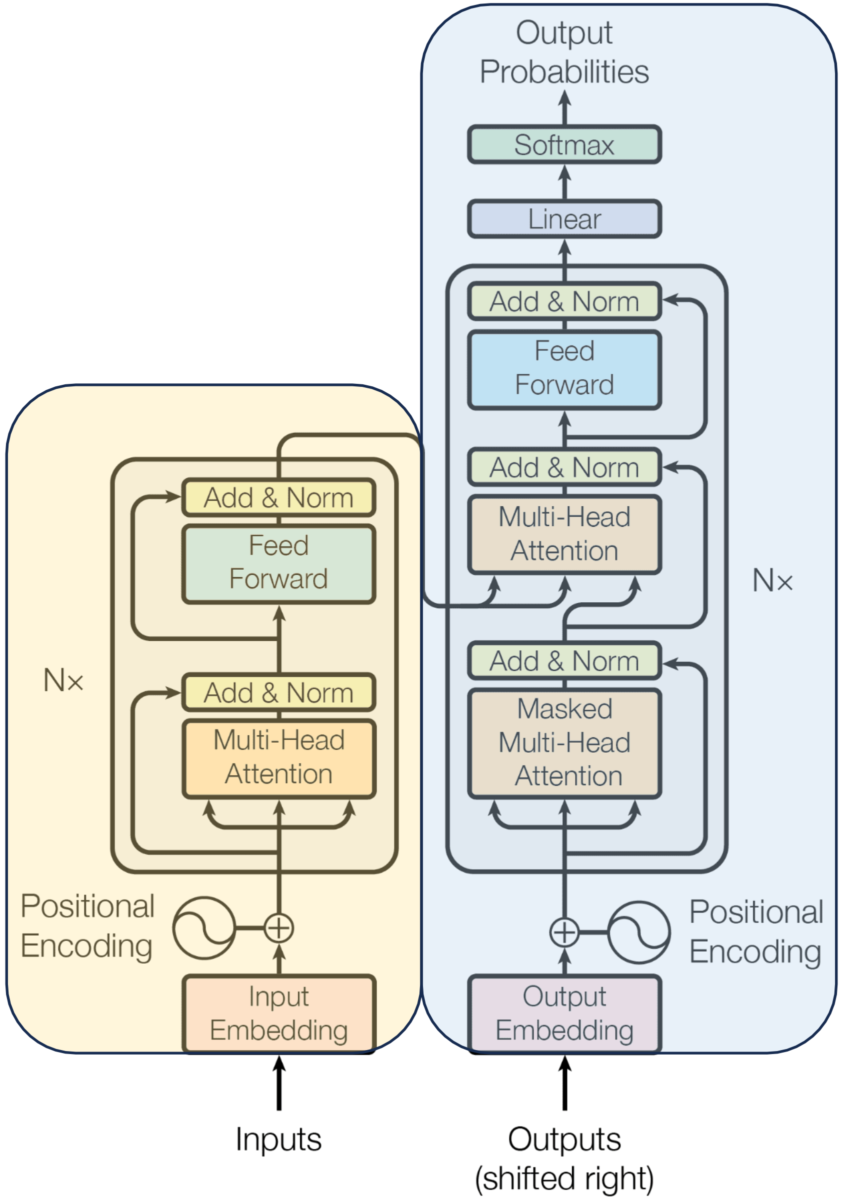
Transformers, mid-2017

Input – input tokens
Output – hidden states

Model can see all timesteps

Does not usually output tokens, so no inherent auto-regressivity

Representation



Input – output tokens and hidden states*
Output – output tokens

Model can only see previous timesteps

Model is auto-regressive with previous timesteps' outputs

Generation

Transformers, mid-2017

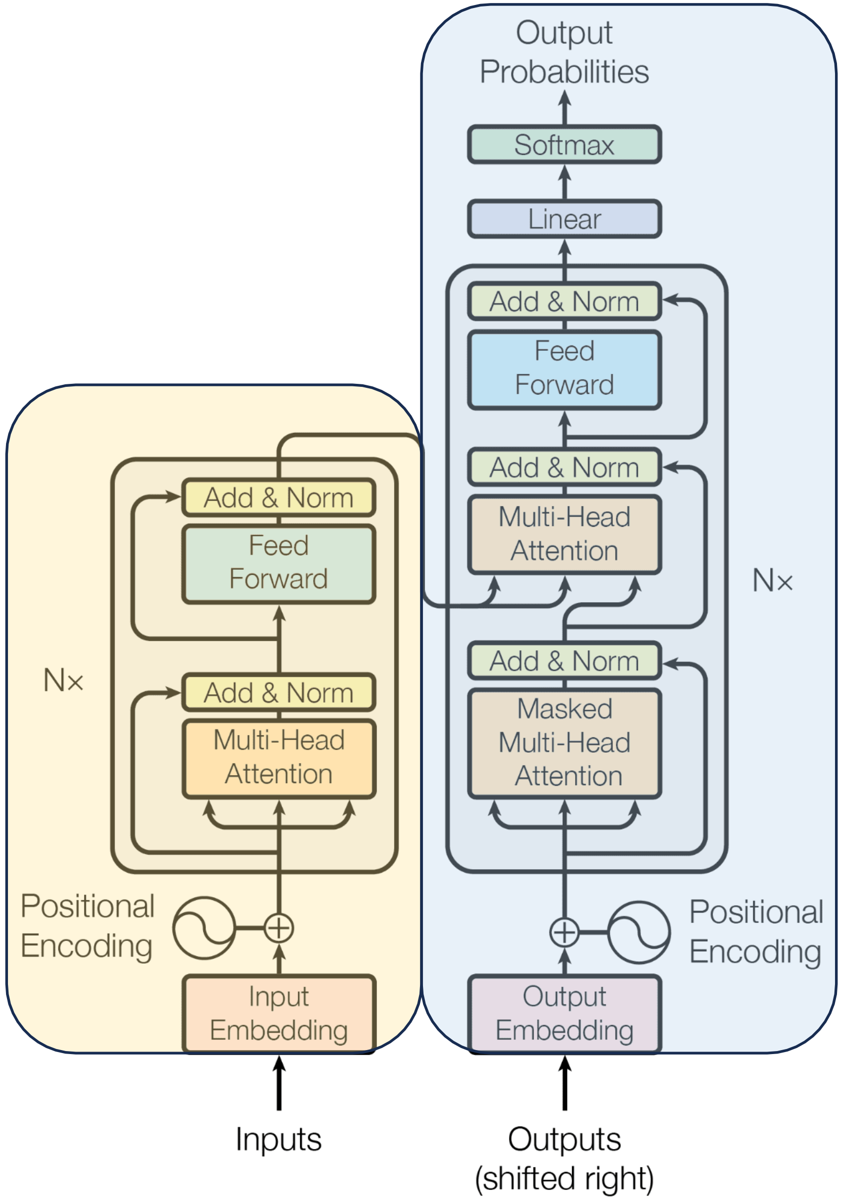
Input – input tokens
Output – hidden states

Model can see all timesteps

Does not usually output tokens, so no inherent auto-regressivity

Can also be adapted to generate tokens *by appending a module that maps hidden state dimensionality to vocab size*

Representation



Input – output tokens and hidden states*
Output – output tokens

Model can only see previous timesteps

Model is auto-regressive with previous timesteps' outputs

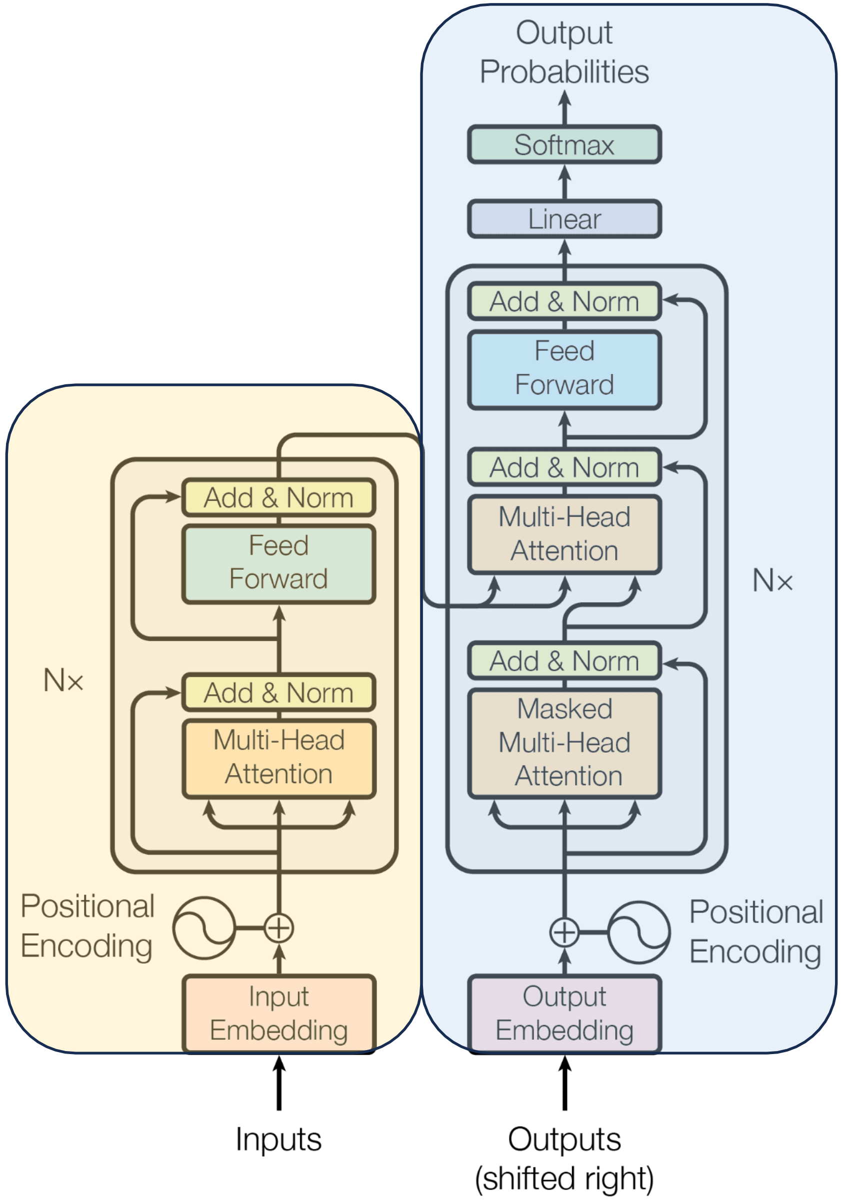
Can also be adapted to generate hidden states *by looking before token outputs*

Generation

2018 – The Inception of the LLM Era

BERT
Oct 2018

Representation



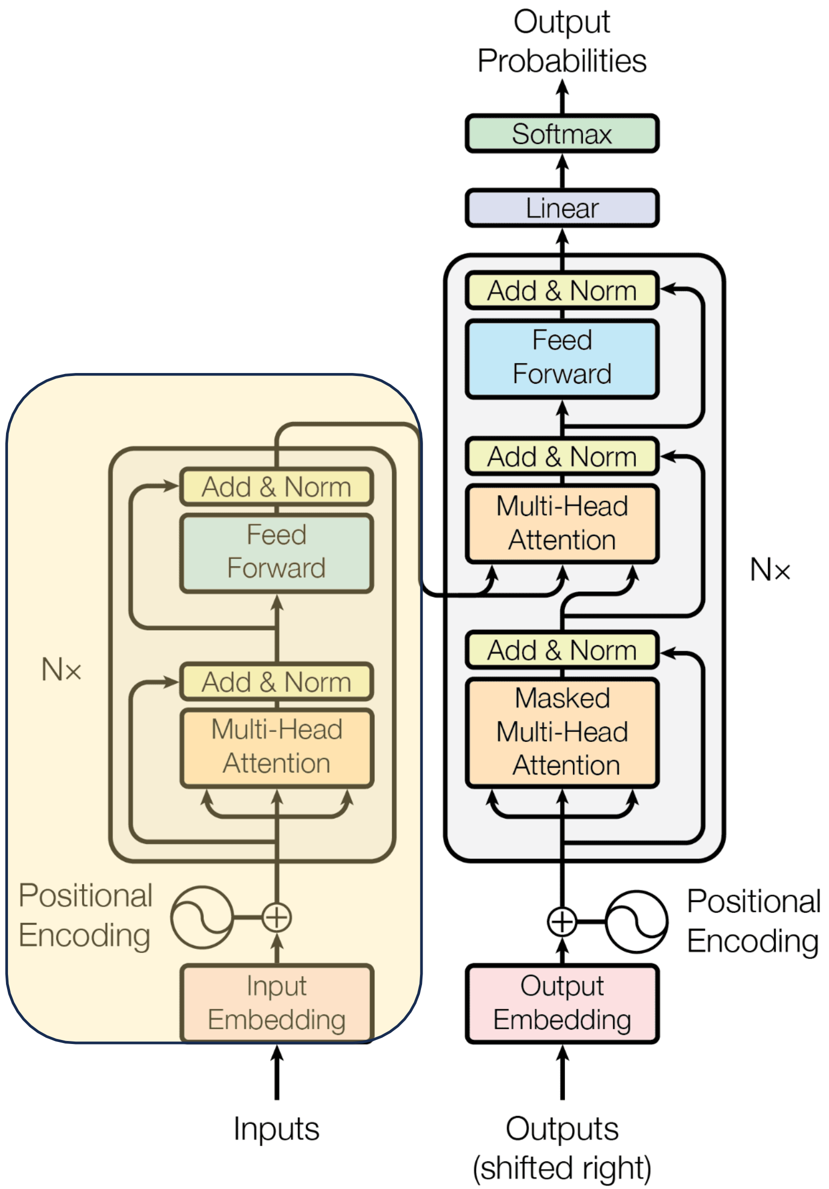
GPT
Jun 2018

Generation

BERT - Bidirectional Encoder Representations

- One of the biggest challenges in LM-building used to be the lack of task-specific training data.

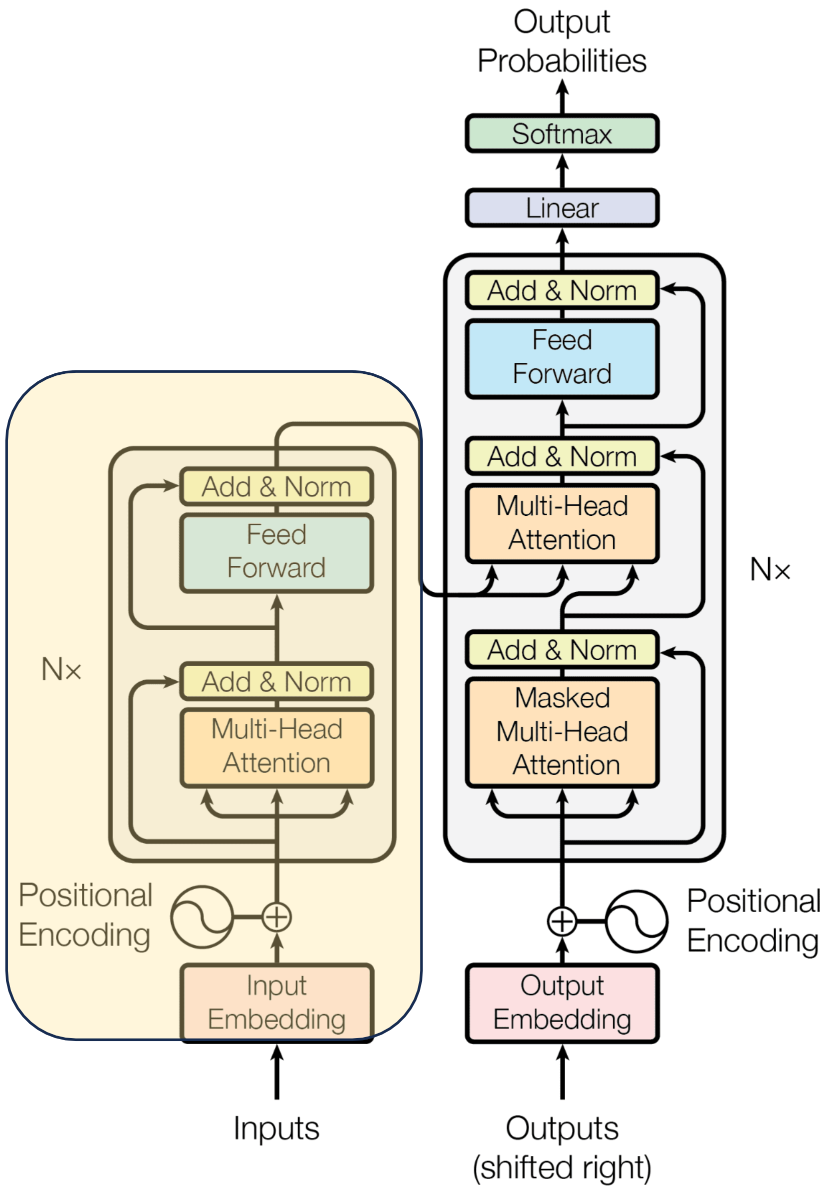
- What if we learn an effective representation that can be applied to a variety of downstream tasks?
  - Word2vec (2013)
  - GloVe (2014)



BERT - Bidirectional Encoder Representations

BERT Pre-Training Corpus:

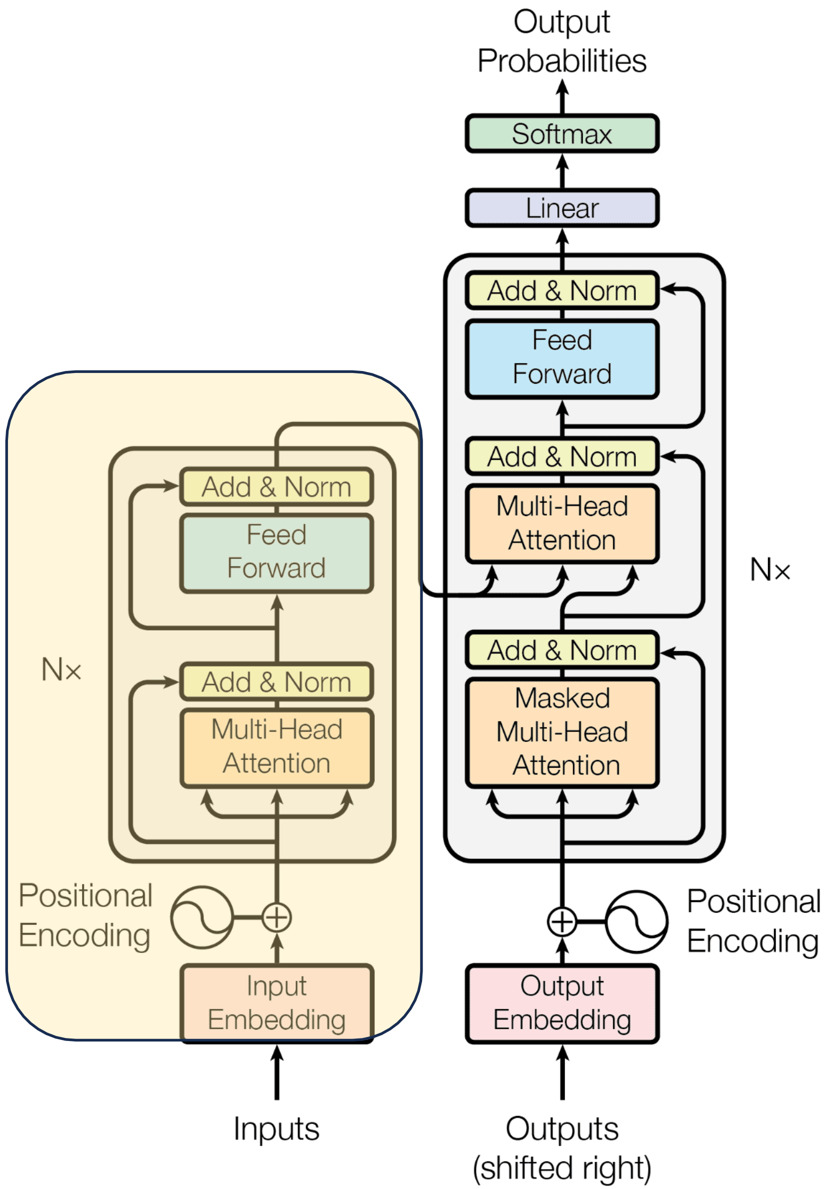
- English Wikipedia - 2,500 million words
- Book Corpus - 800 million words



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)

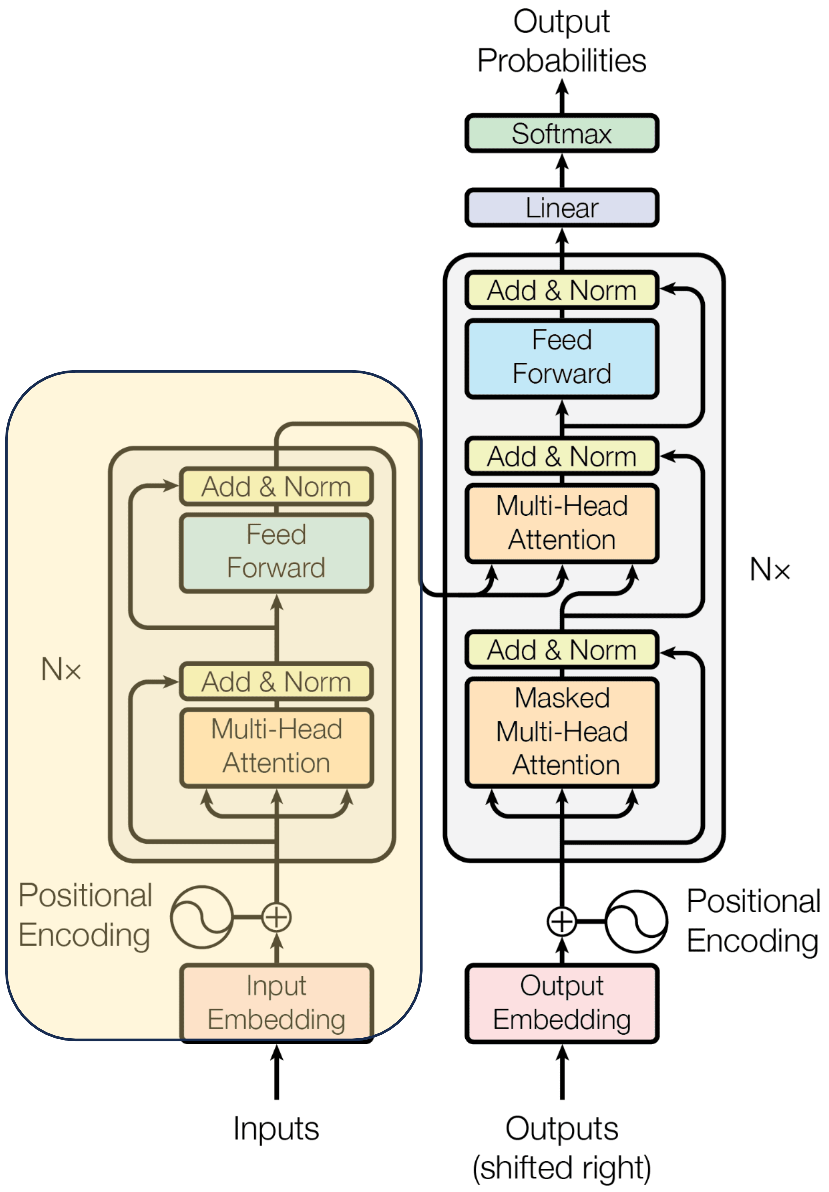


BERT - Bidirectional Encoder Representations

BERT Pre-Training Corpus:
- English Wikipedia - 2,500 million words
- Book Corpus - 800 million words

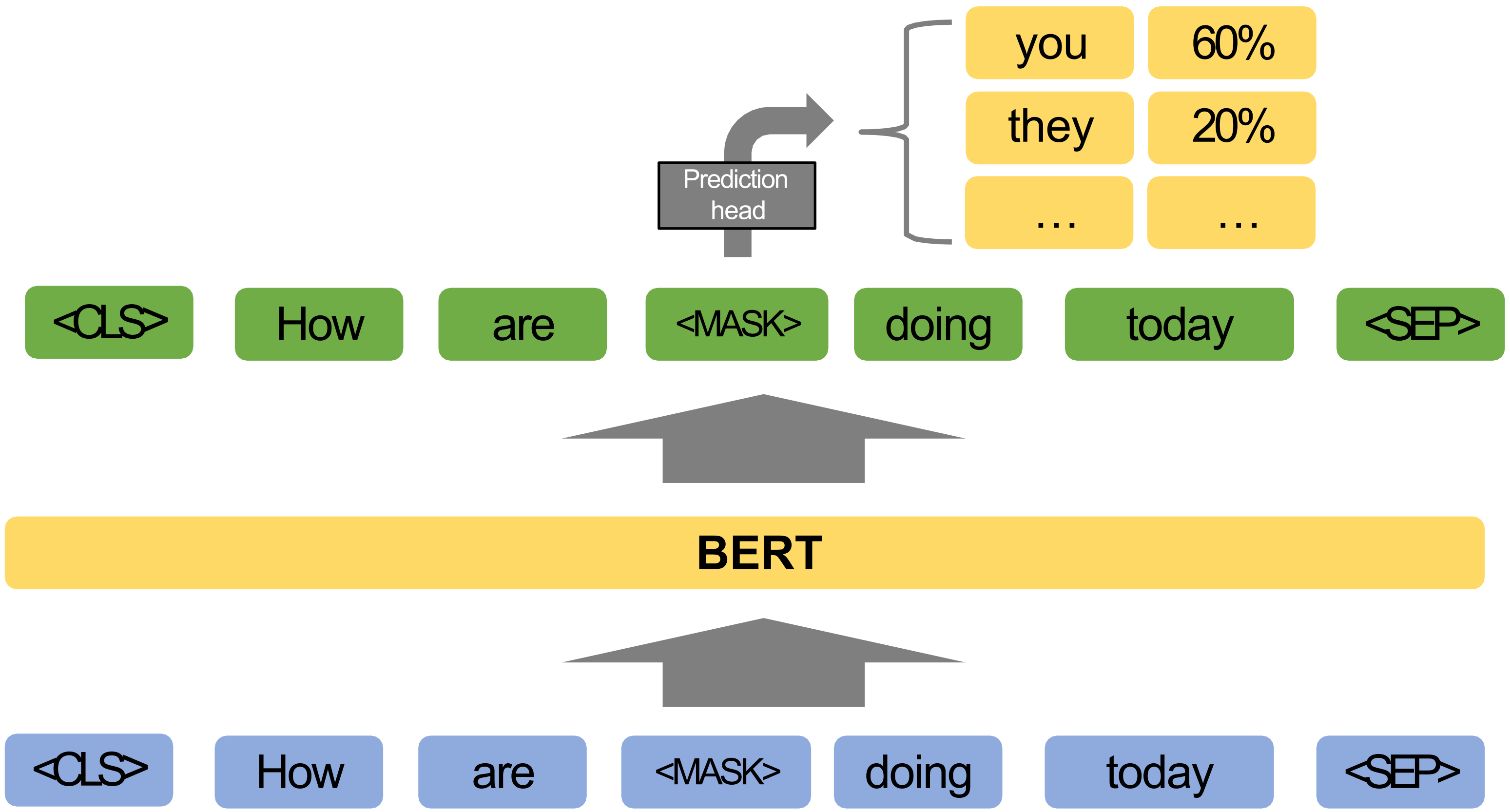
BERT Pre-Training Tasks:
- MLM (Masked Language Modeling)
- NSP (Next Sentence Prediction)

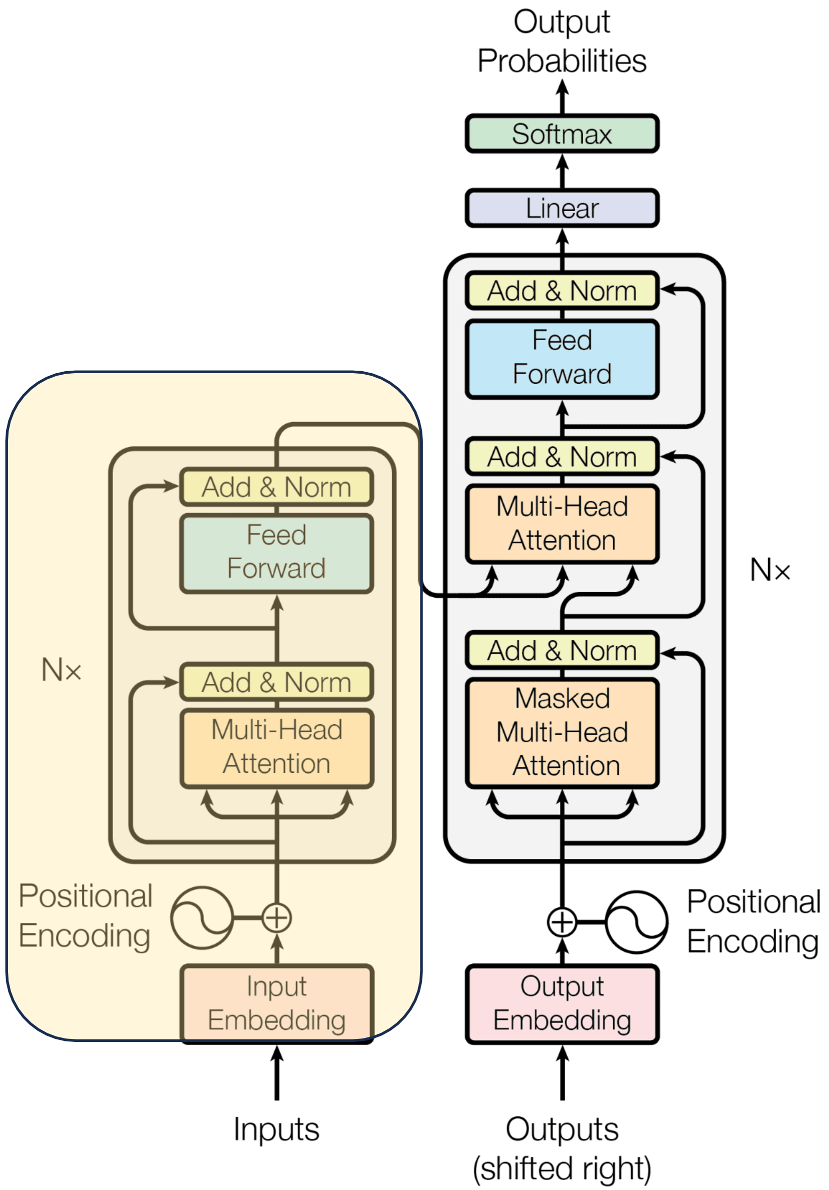
BERT Pre-Training Results:
- BERT-Base – 110M Params
- BERT-Large – 340M Params



BERT - Bidirectional Encoder Representations

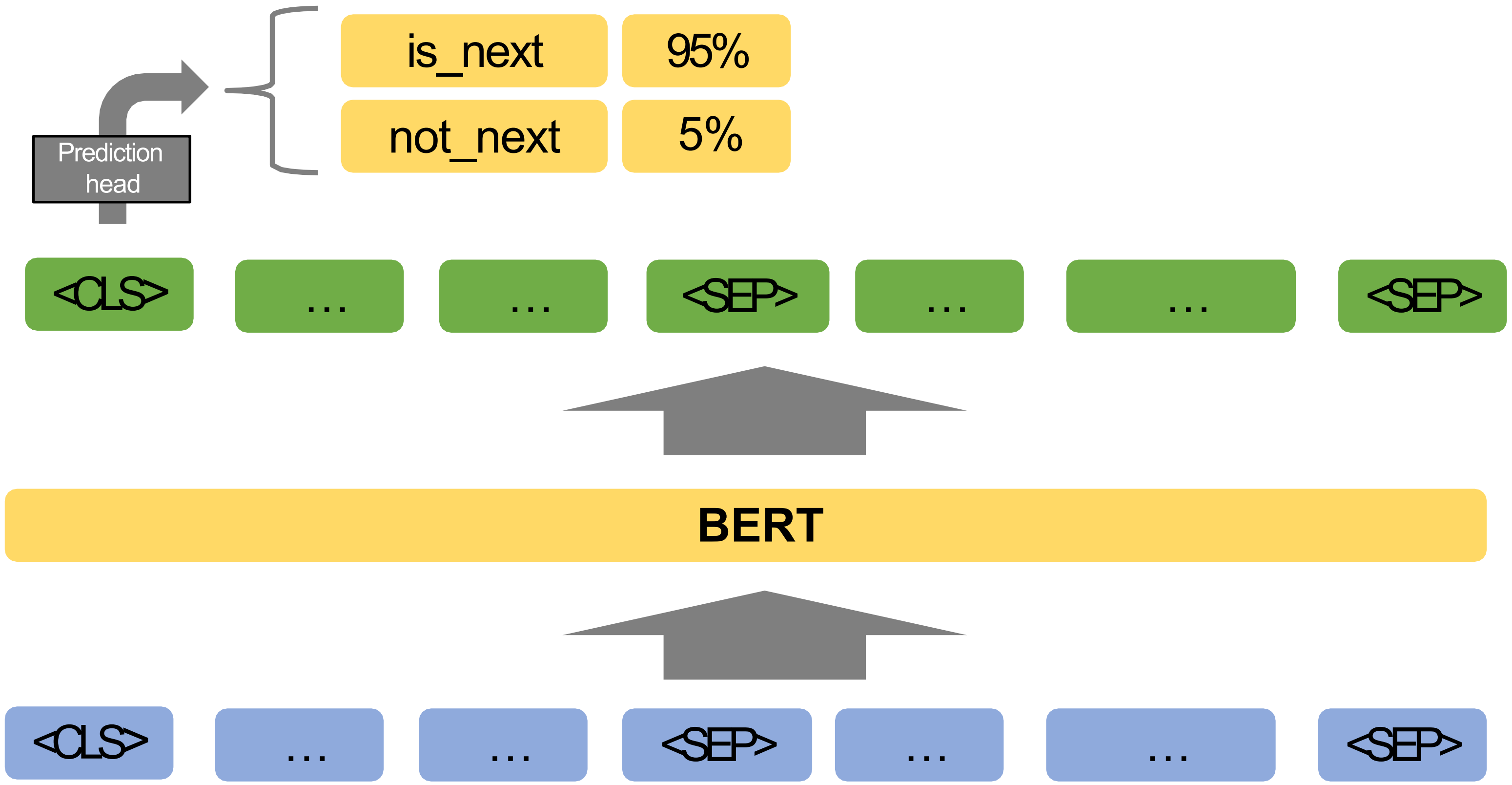
MLM (Masked Language Modeling)

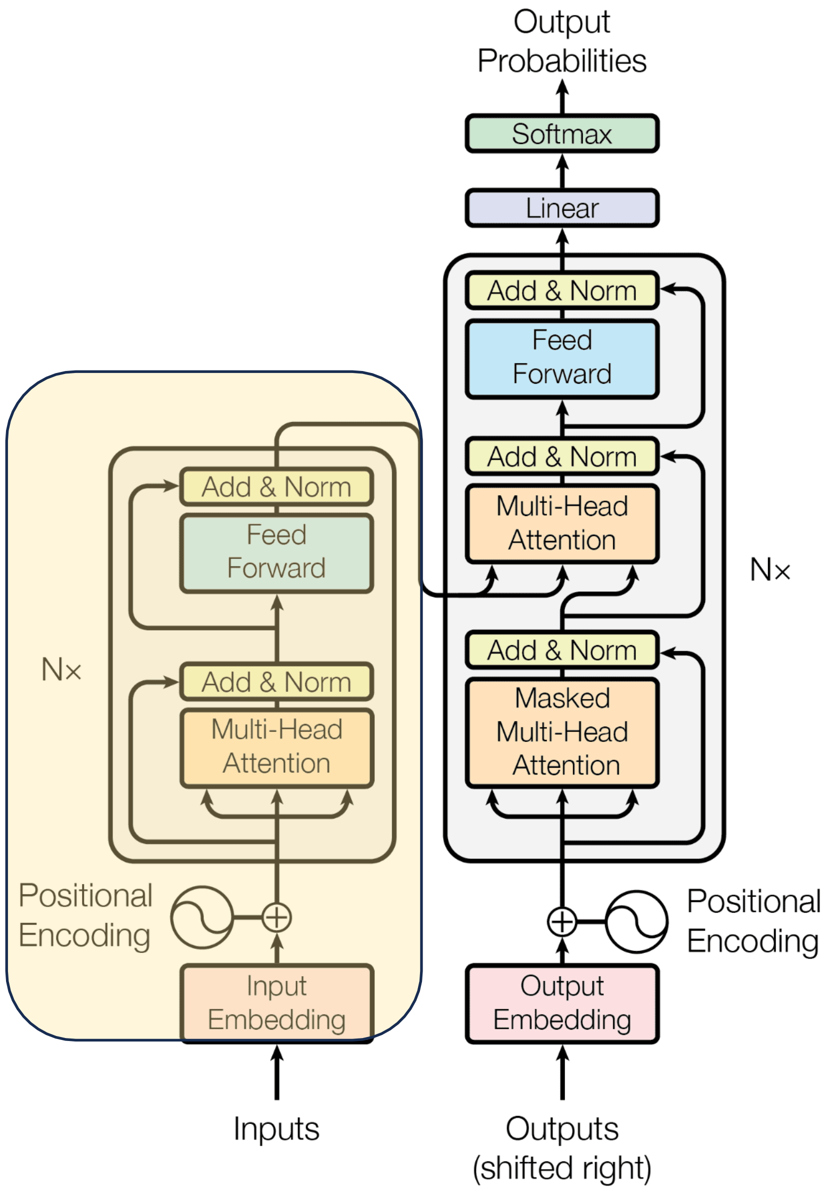




BERT - Bidirectional Encoder Representations

NSP (Next Sentence Prediction)



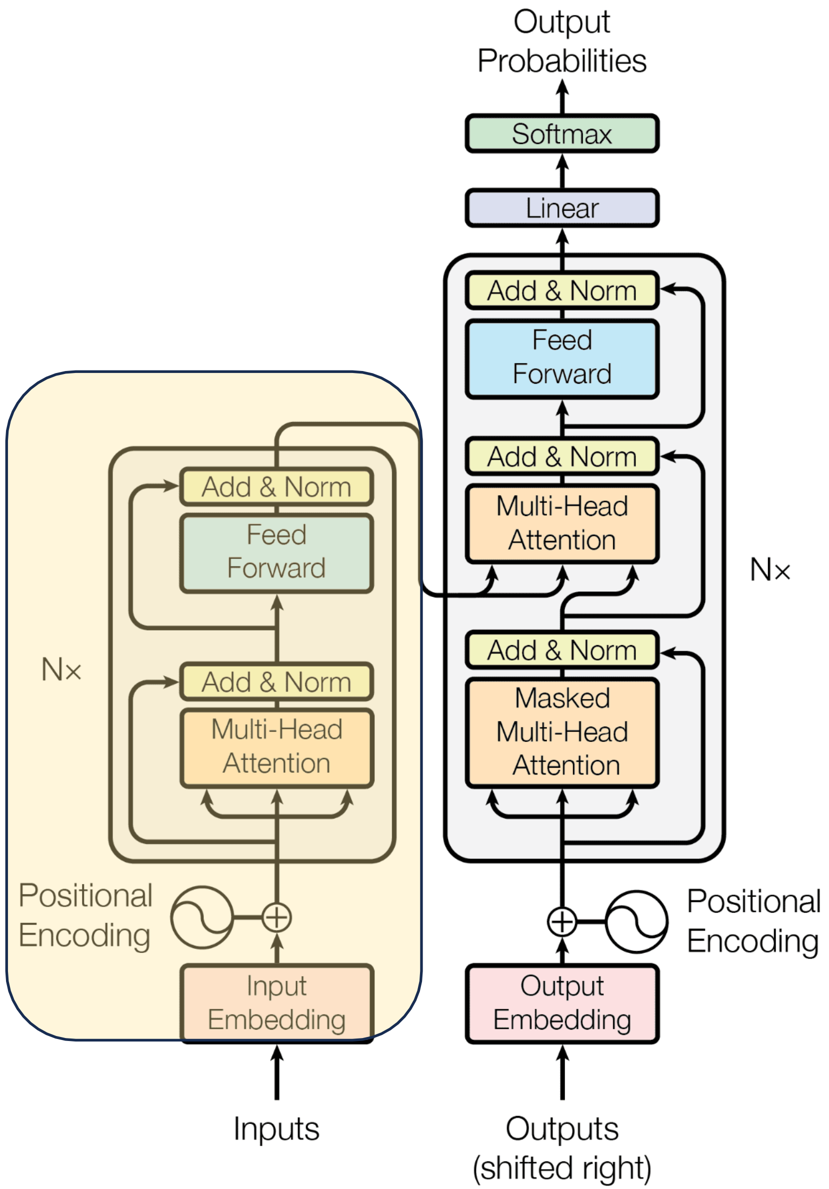


BERT - Bidirectional Encoder Representations

BERT Fine-Tuning:

- Simply add a task-specific module after the last encoder layer to map it to the desired dimension.

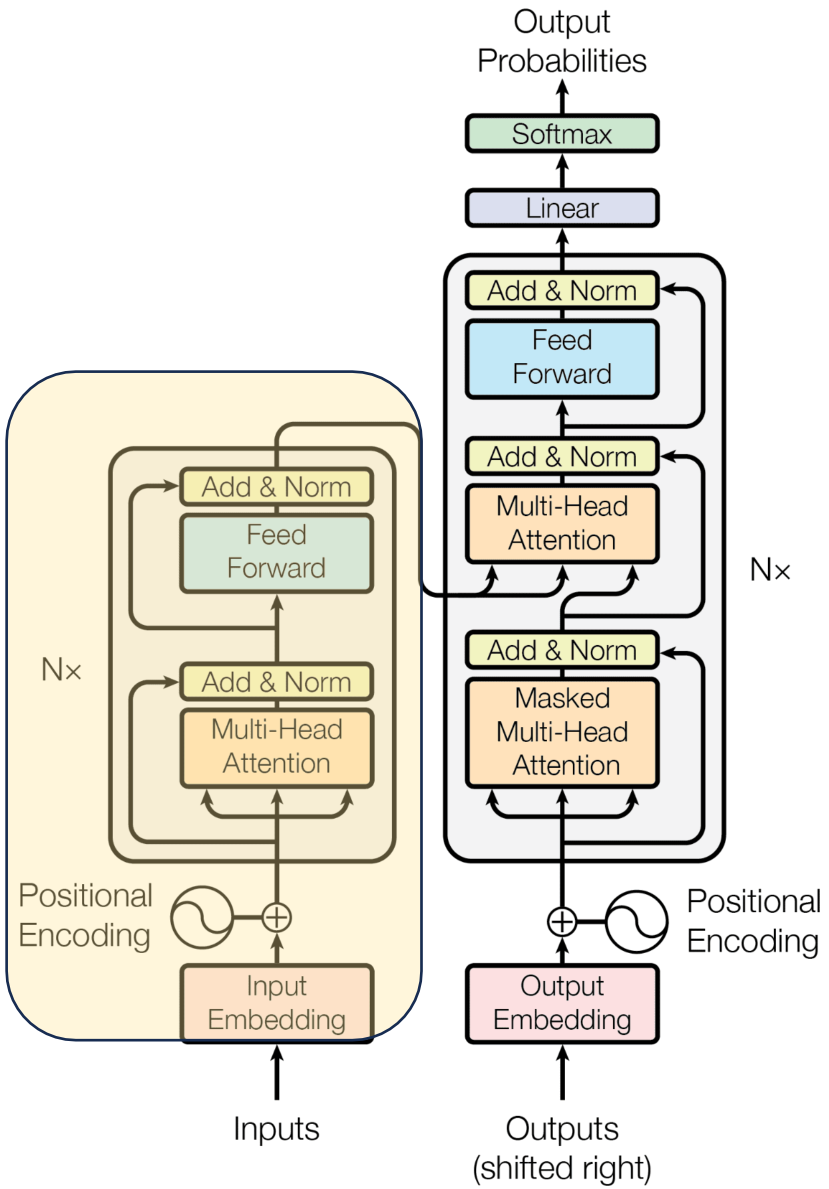
  - *Classification Tasks:*
    - *Add a feed-forward layer on top of the encoder output for the [CLS] token*
  - *Question Answering Tasks:*
    - *Train two extra vectors to mark the beginning and end of answer from paragraph*
  - *...*



BERT - Bidirectional Encoder Representations

BERT Evaluation:

- General Language Understanding Evaluation (GLUE)
  - Sentence pair tasks
  - Single sentence classification
- Standford Question Answering Dataset (SQuAD)



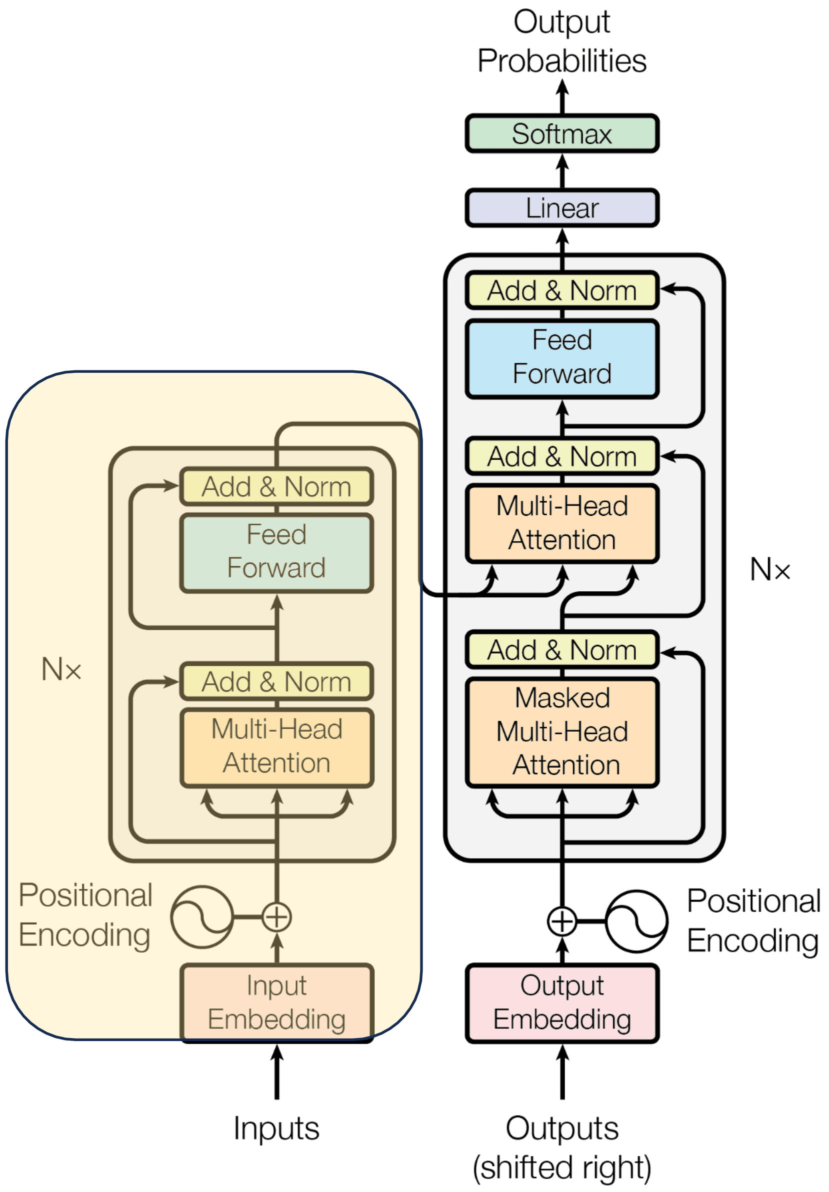
BERT - Bidirectional Encoder Representations

BERT Evaluation:

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
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**

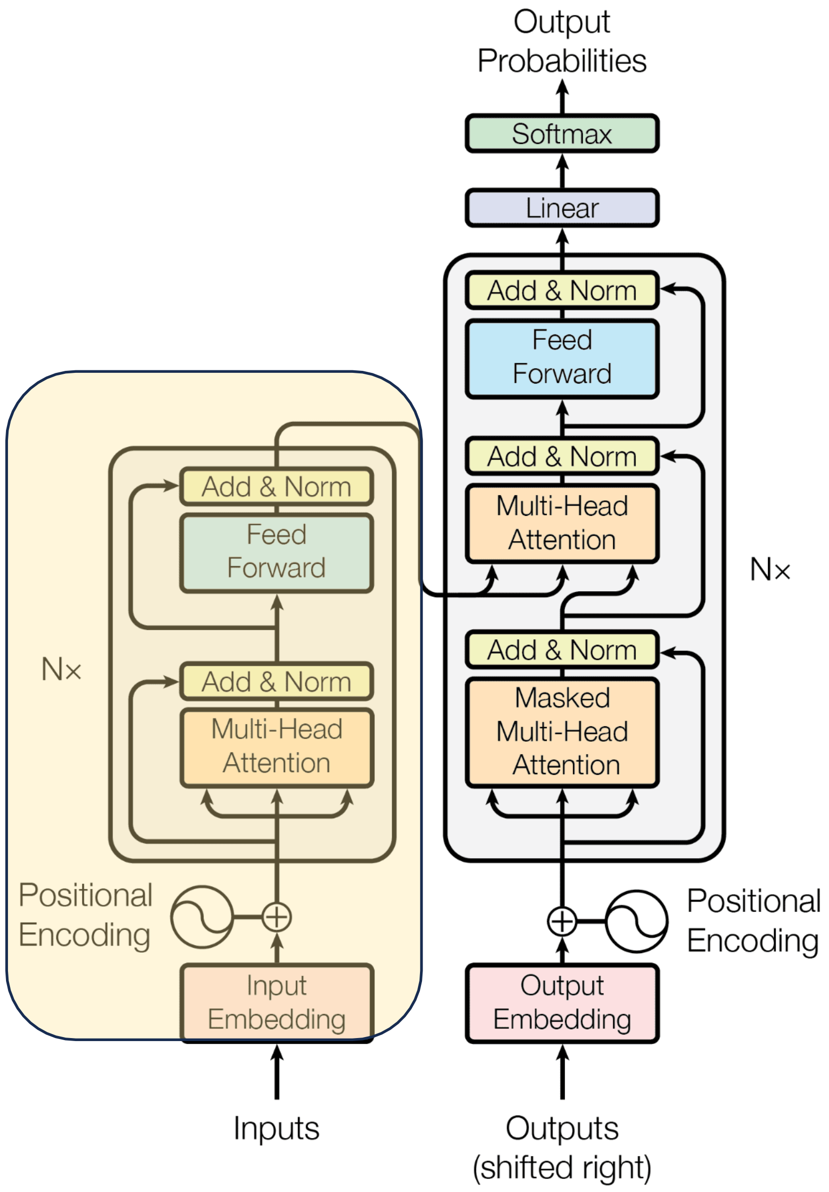
Table 2: SQuAD results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.



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.

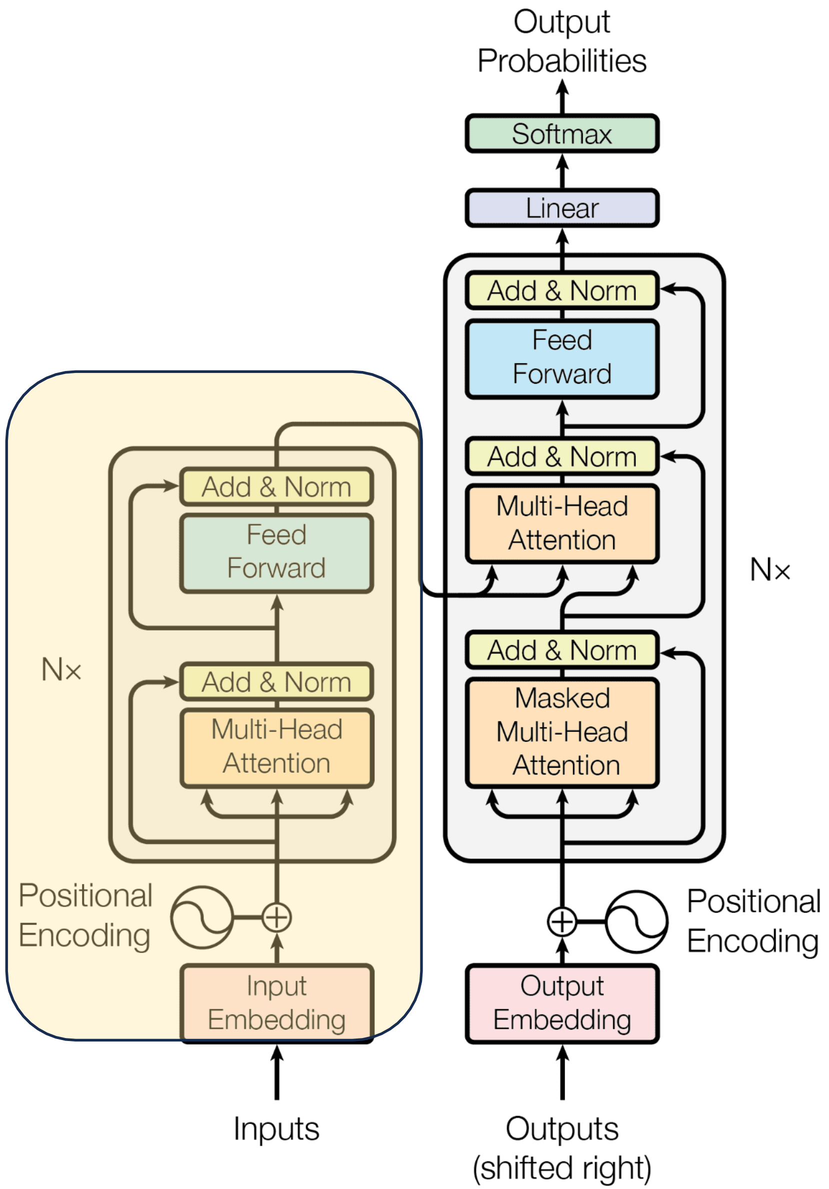


BERT - Bidirectional Encoder Representations

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.

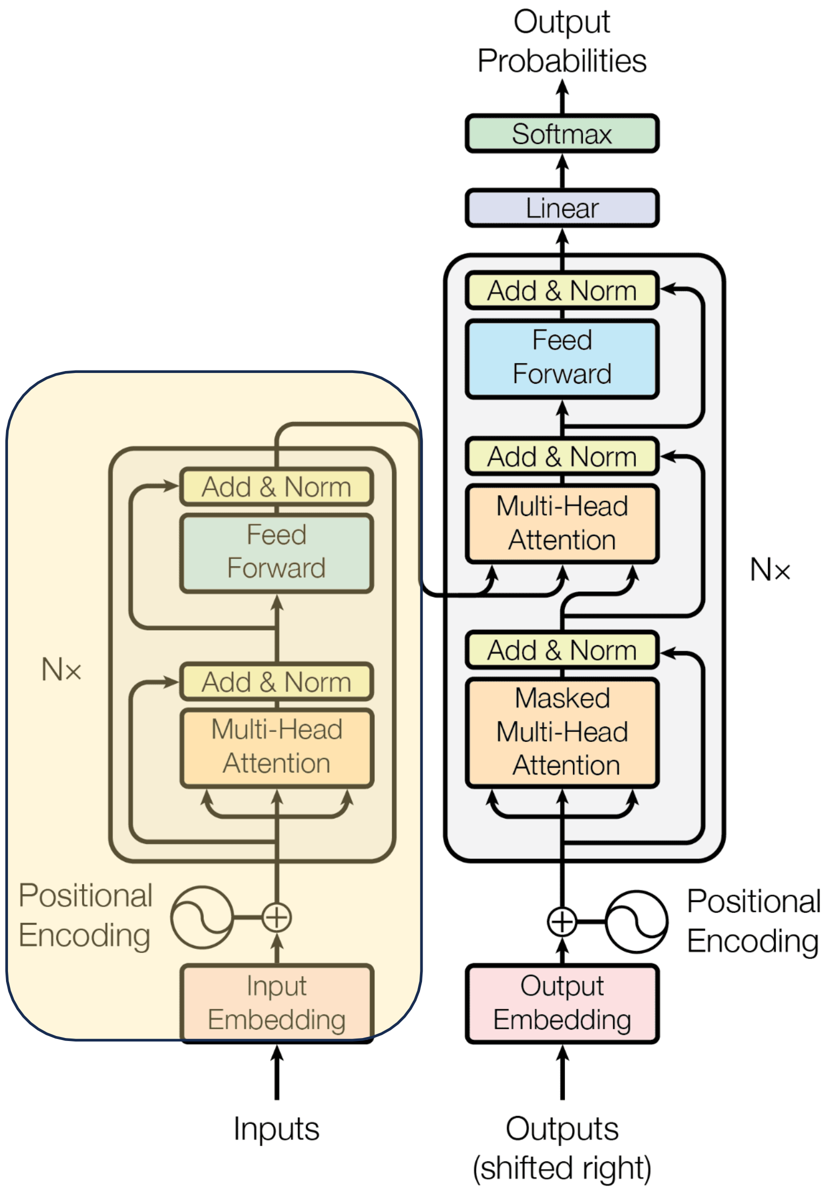
- **Different NLP tasks seem to be highly transferable with each other...**
  - As long as we have effective representations, that seems to form a general model which can serve as the backbone for many specialized models.



BERT - Bidirectional Encoder Representations

What is our takeaway from BERT?

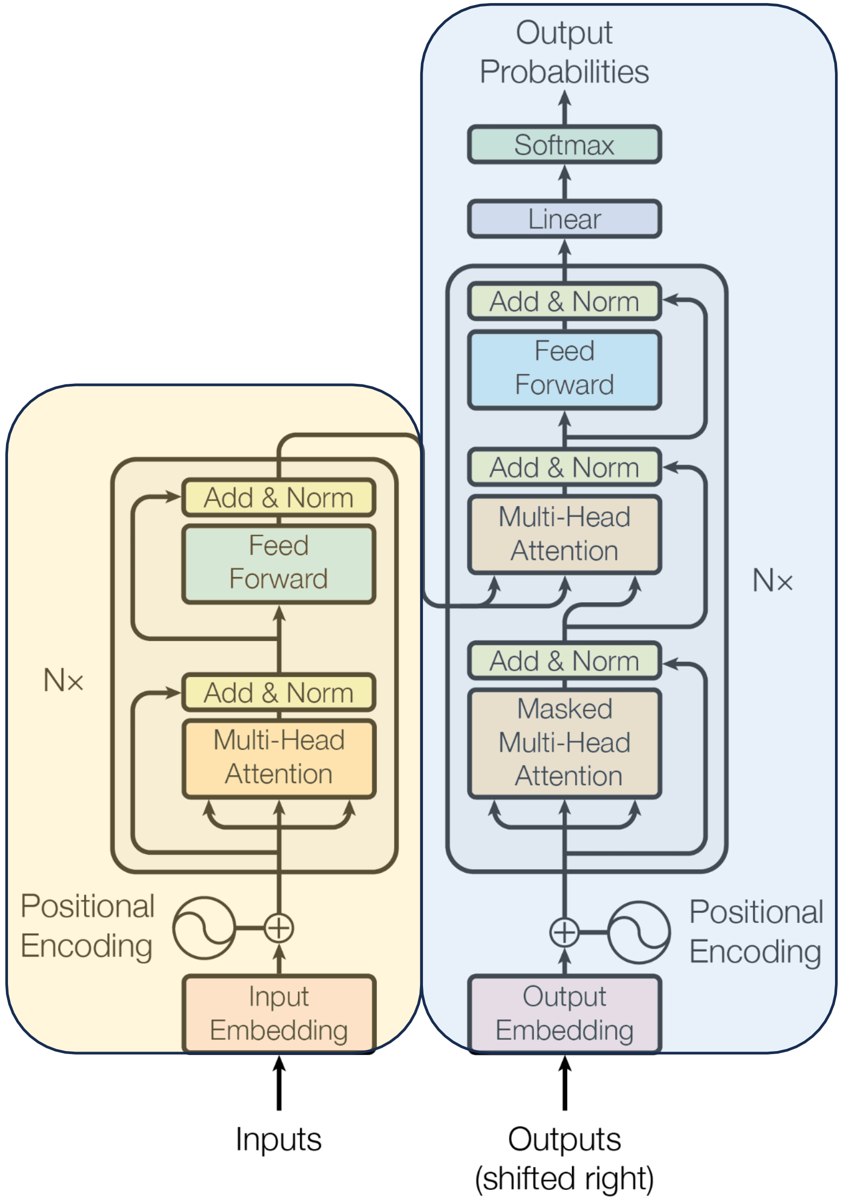
- **Pre-training tasks can be invented flexibly…**
  - Effective representations can be derived from a flexible regime of pre-training tasks.
- **Different NLP tasks seem to be highly transferable with each other...**
  - As long as we have effective representations, that seems to form a general model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



2018 – The Inception of the LLM Era

BERT
Oct 2018

Representation

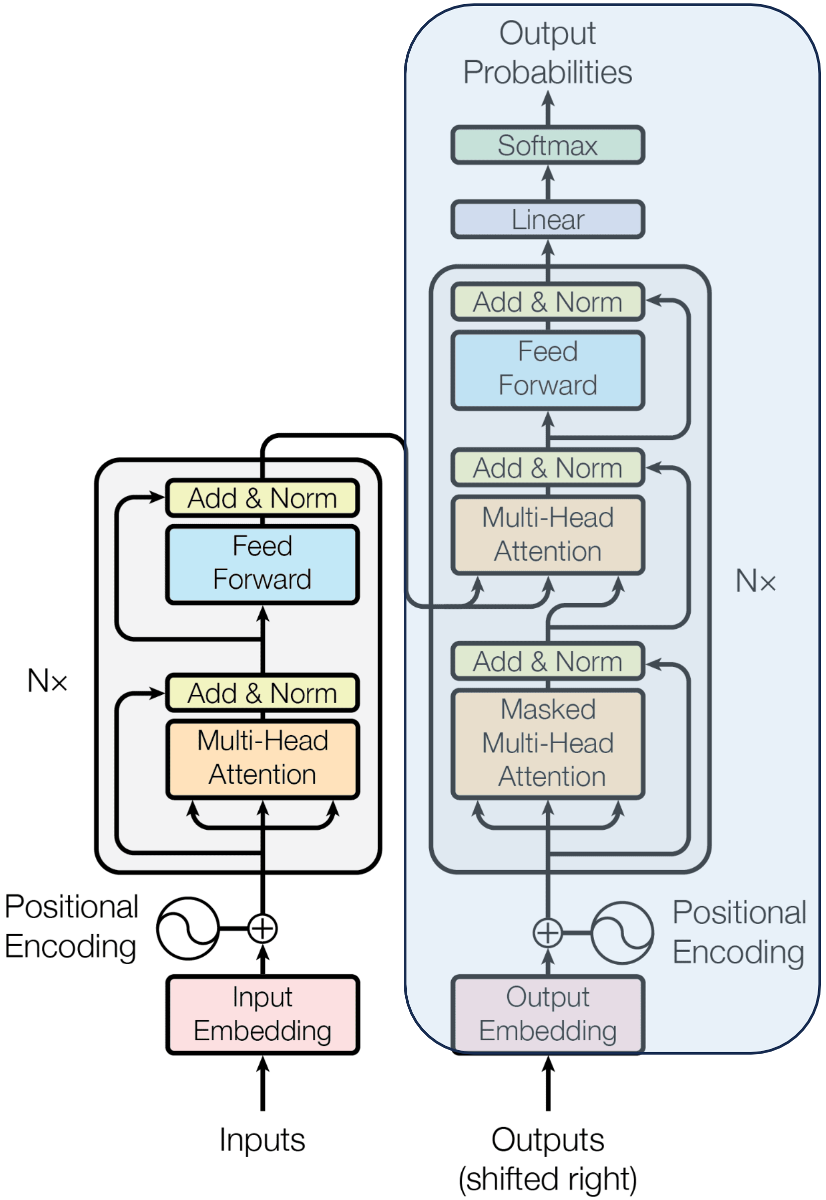


GPT
Jun 2018

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?

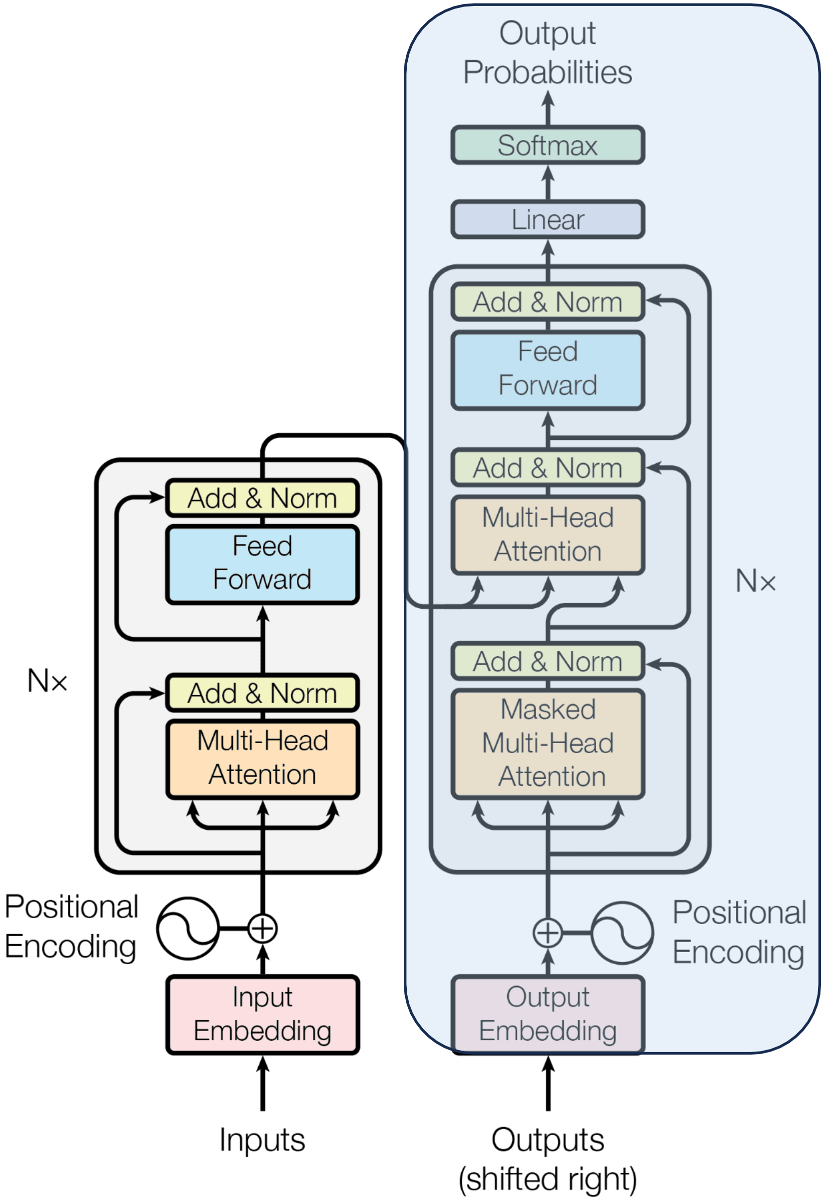


GPT – Generative Pretrained Transformer

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:

The summary is:

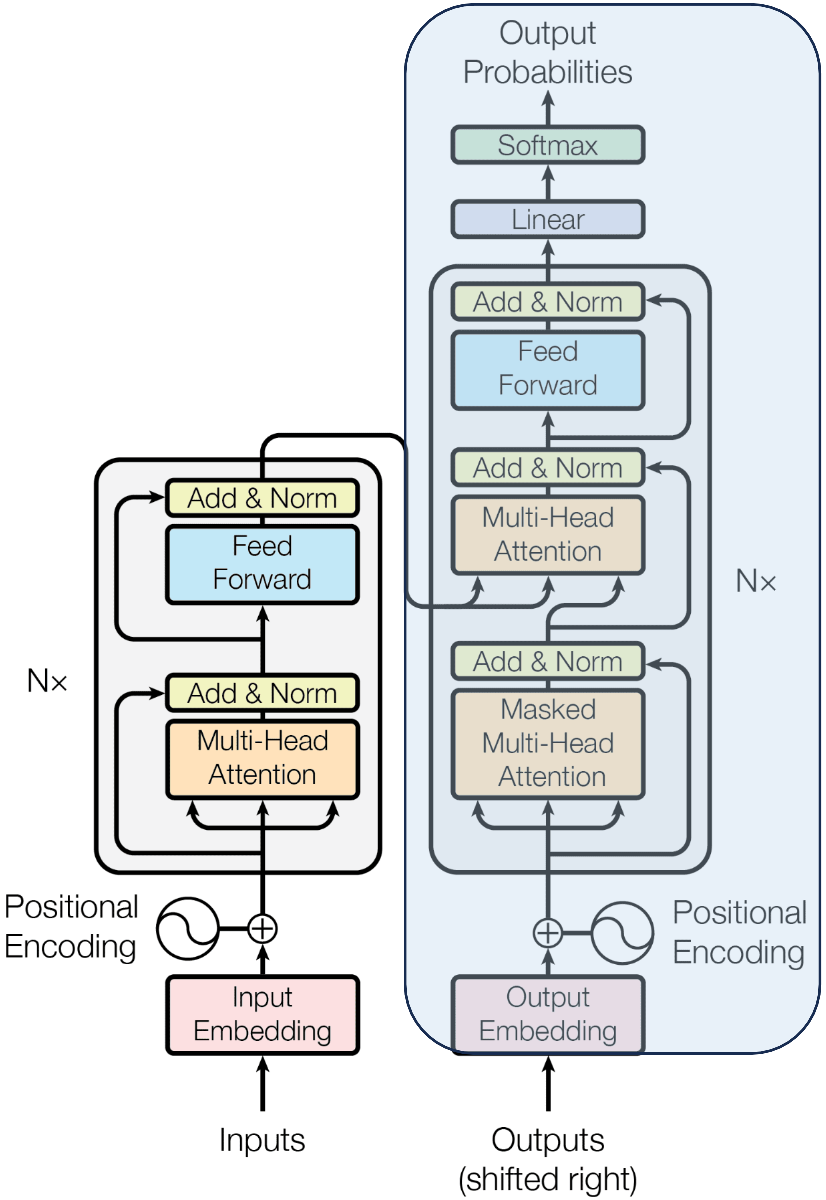
QA

Answer the question based on the context.

Context:

Question:

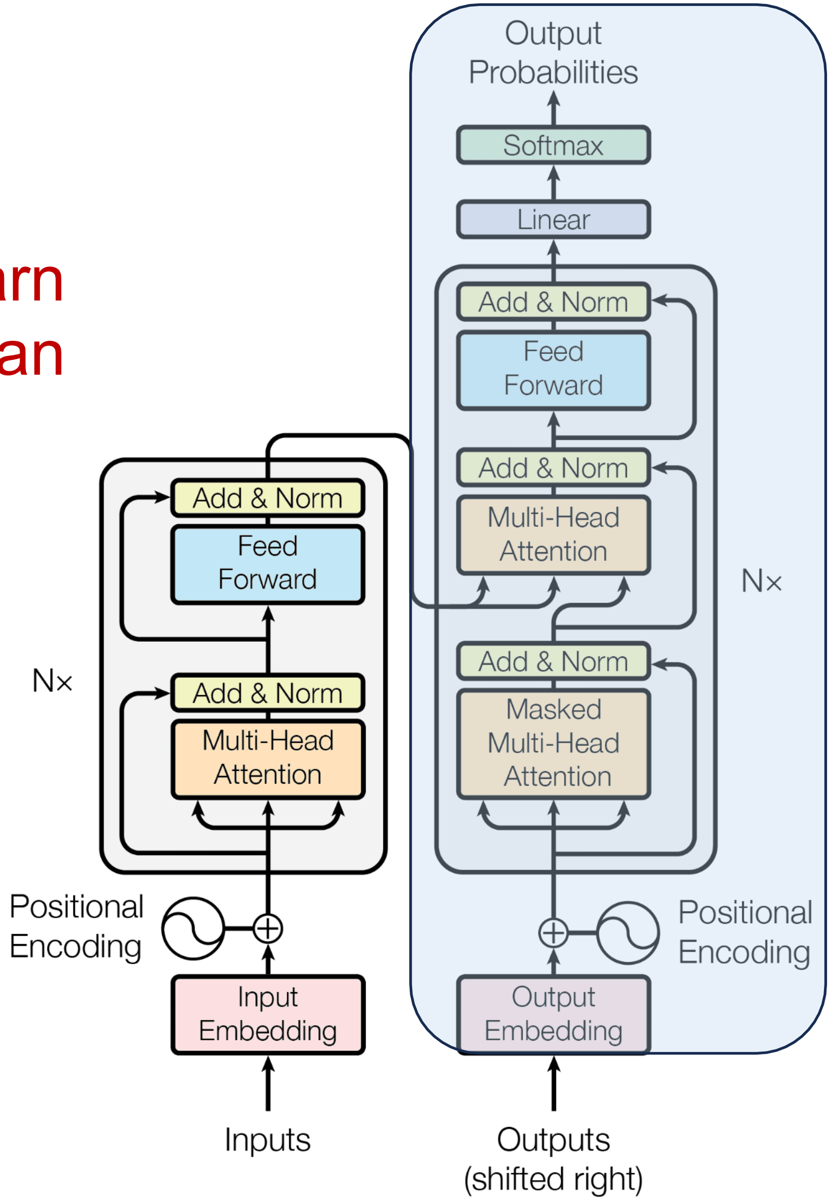
Answer:



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.

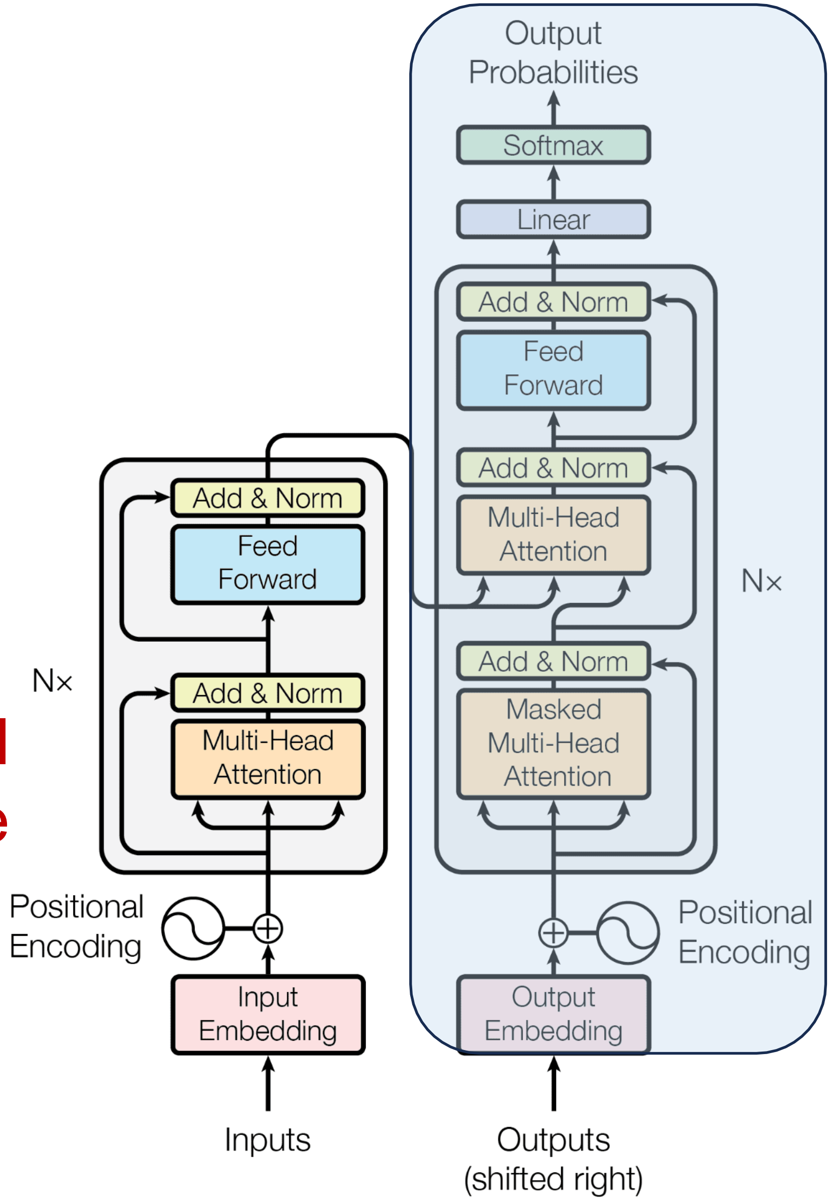


GPT – Generative Pretrained Transformer

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.



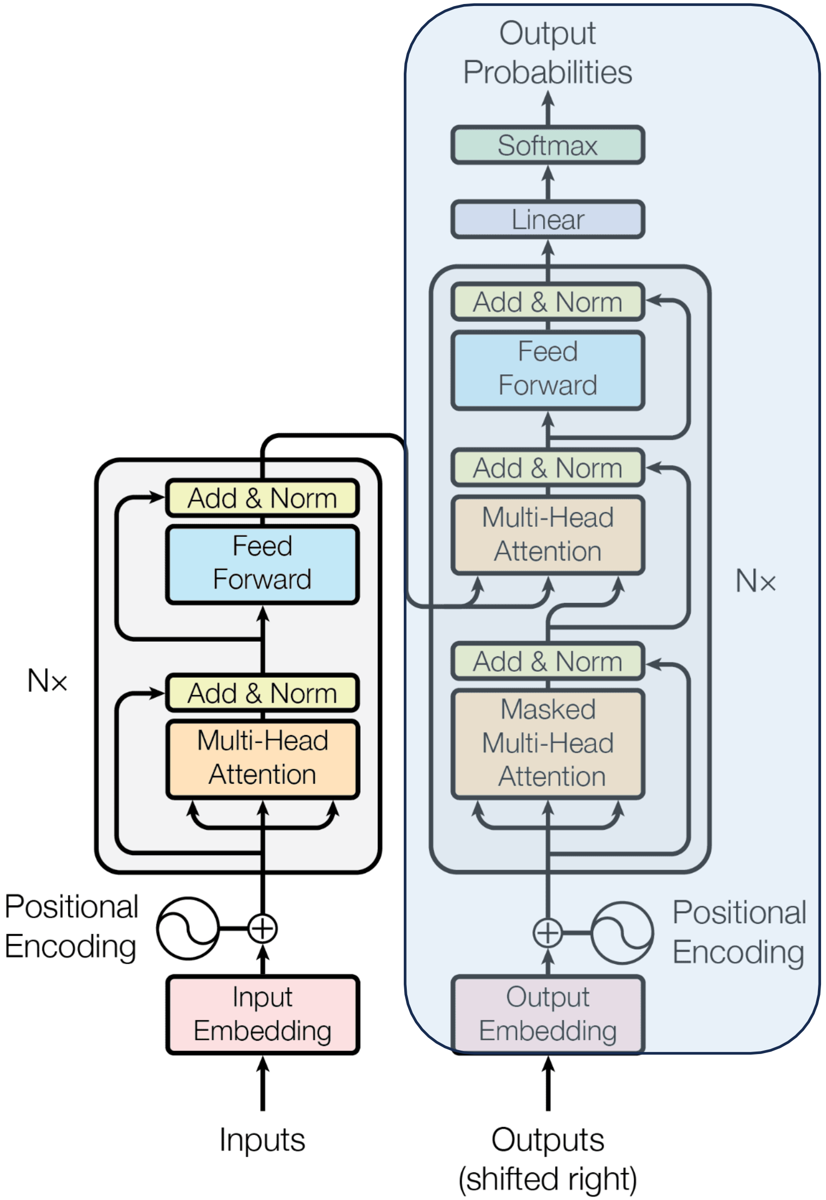
GPT – Generative Pretrained Transformer

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.

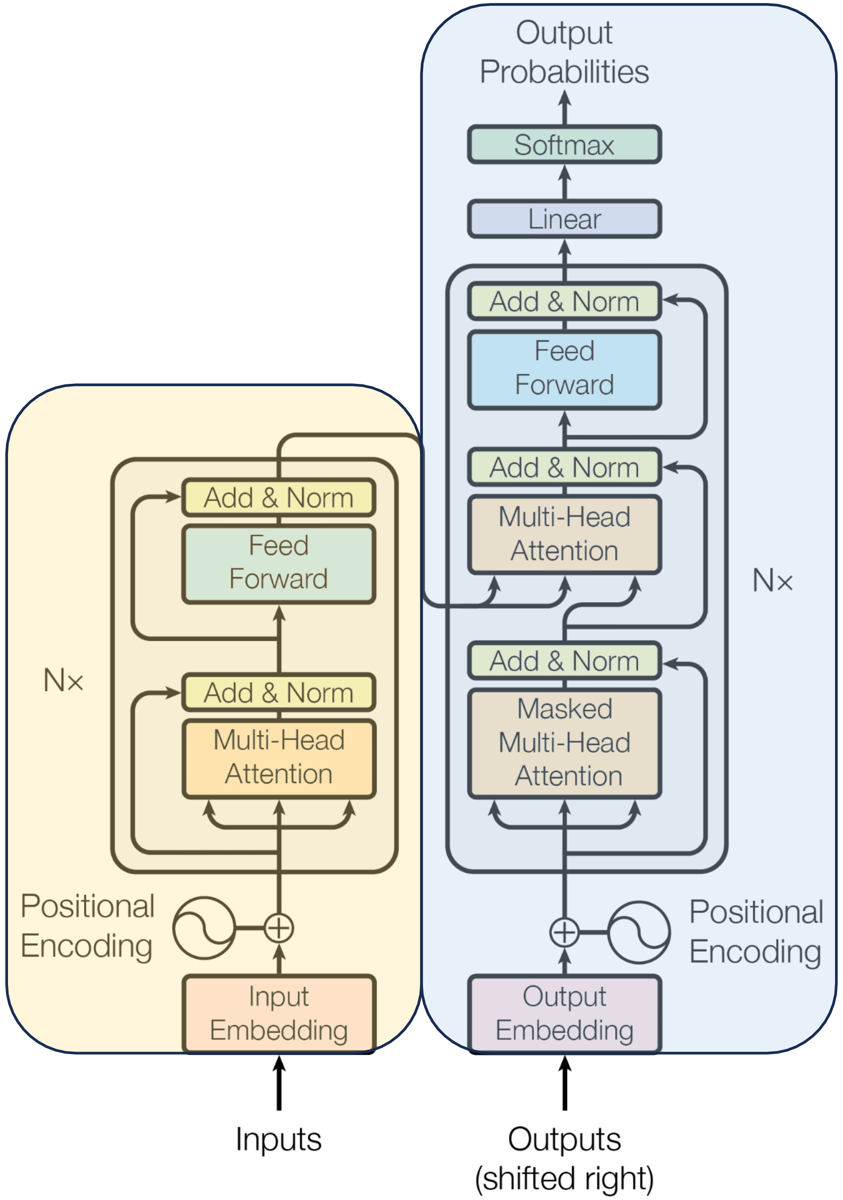
- **And scaling works!!!**



The LLM Era – Paradigm Shift in Machine Learning

BERT
Oct 2018

Representation



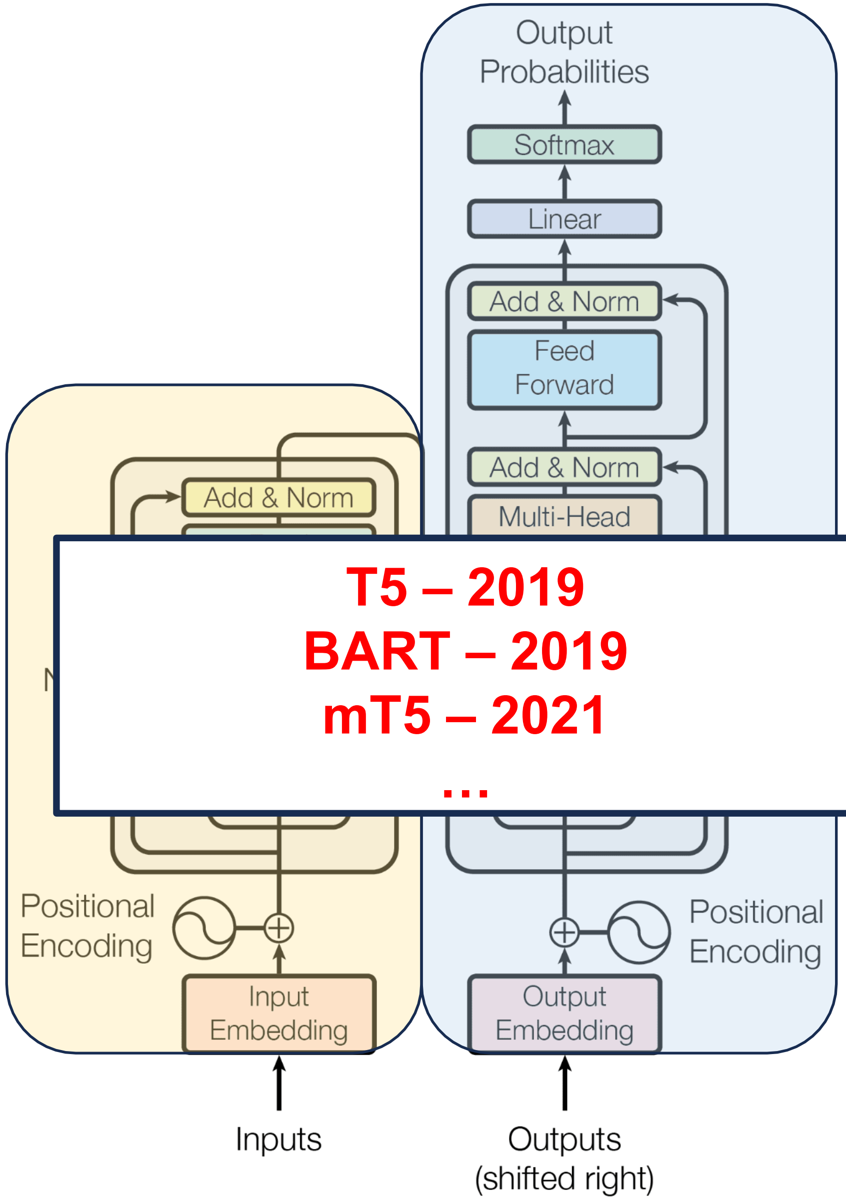
GPT
Jun 2018

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.

<u>Before LLMs</u>

- **Feature Engineering**
  - How do we design or select the best features for a task?

<u>Since LLMs</u>

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.

<u>**Before LLMs**</u>

<u>**Since LLMs**</u>

- **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?

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.

<u>**Before LLMs**</u>

<u>**Since LLMs**</u>

- **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?

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.

<u>Before LLMs</u>

- **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?

<u>Since LLMs</u>

- **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 *<u>not</u>* 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 – Paradigm Shift in Machine Learning

- **What has caused this paradigm shift?**

The LLM Era – Paradigm Shift in Machine Learning

- **What has caused this paradigm shift?**

  - **Problem in recurrent networks**
    - Information is effectively lost during encoding of long sequences
    - Sequential nature disables parallel training and favors late timestep inputs

The LLM Era – Paradigm Shift in Machine Learning

- **What has caused this paradigm shift?**

  - **Problem in recurrent networks**
    - Information is effectively lost during encoding of long sequences
    - Sequential nature disables parallel training and favors late timestep inputs

  - **Solution: Attention mechanism**
    - Handling long-range dependencies
    - Parallel training
    - Dynamic attention weights based on inputs

The LLM Era – Paradigm Shift in Machine Learning

- **Attention and Transformer – is this the end?**

The LLM Era – Paradigm Shift in Machine Learning

- **Attention and Transformer – is this the end?**
    - **Problem in current Transformer-based LLMs??**

The LLM Era – Paradigm Shift in Machine Learning

- **Attention and Transformer – is this the end?**

  - **Problem in current Transformer-based LLMs??**
    - True understanding the material vs. memorization and pattern-matching
    - Cannot reliably follow rules – factual hallucination e.g. inability in arithmetic

The LLM Era – Paradigm Shift in Machine Learning

- **Attention and Transformer – is this the end?**

  - **Problem in current Transformer-based LLMs??**
    - True understanding the material vs. memorization and pattern-matching
    - Cannot reliably follow rules – factual hallucination e.g. inability in arithmetic

  - **Solution: ???**

Looking Back

It is true that language models are just programmed to predict the next token. But that is not as simple as you might think.

In fact, all animals, including us, are just programmed to survive and reproduce, and yet amazingly complex and beautiful stuff comes from it.

- Sam Altman*

***Paraphrased by IDL TAs**

Acknowledgement

Reference and thanks to:

- **CMU 11-785 Course:**

Introduction to Deep Learning
https://deeplearning.cs.cmu.edu/F23/