Large Language Models (LLMs)

Naeemullah Khan
naeemullah.khan@kaust.edu.sa

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

LMH
Lady Margaret Hall

July 3, 2025

Large Language Models

BCV

Application Layer

Copywriting
Jasper
copy.ai
Headlime
co:here
HyperWrite
Writesonic
Contenda
unbounce
copysmith

Coding
tabnine
MUTABLEAI
Codiga
co:here
GitHub Copilot
CODEGEN

Dev Tools
algolia
warp
Mintlify
cogram
Debuild
repl.it

Chat / Comms
MessageBird
Replier.ai
Sapling
FABLE

BizOps
viable
Enterpret
tabulate
Anecdote
OTHERSIDE AI
casetext
Dover

Infrastructure Layer

Model Creation
AI21labs
OpenAI
NVIDIA
Adept
EleutherAI
ANTHROPIC
Google AI

Hardware
habana
NVIDIA
SambaNova SYSTEMS
cerebras
GRAPHCORE

Fine Tuning
Google AI
aws
OpenAI
Hugging Face

Inference
OpenAI
Hugging Face

Table of Contents

1. Motivation
2. Learning Outcomes
3. BERT and GPT Concepts
4. Scaling Laws for LLMs
5. Pre-training Overview
6. Tokenization: Why It Matters?
7. Level of Tokenization
8. Byte-Pair Encoding (BPE)
9. Token-Free LLMs
10. Context Window Problems
11. Moving Towards Larger Context Windows
12. Limitations of LLMs
13. Future Directions

Motivation

- Why LLMs?
  - Transforming NLP: ChatGPT, Claude, Gemini, etc.
  - Achieving human-like generation and comprehension.
  - Pivotal for tasks like summarization, translation, reasoning.
- Need for deeper understanding:
  - LLMs are expensive to train and operate.
  - Design decisions impact performance significantly (e.g., tokenization, scaling laws).

Learning Outcomes

By the end of this session, you should be able to:

- Explain core architectures like BERT and GPT.
- Understand scaling laws for LLM development.
- Describe tokenization strategies and their impact.
- Discuss limitations of current models (e.g., context window).
- Explore emerging directions like token-free LLMs.

BERT and GPT Concepts

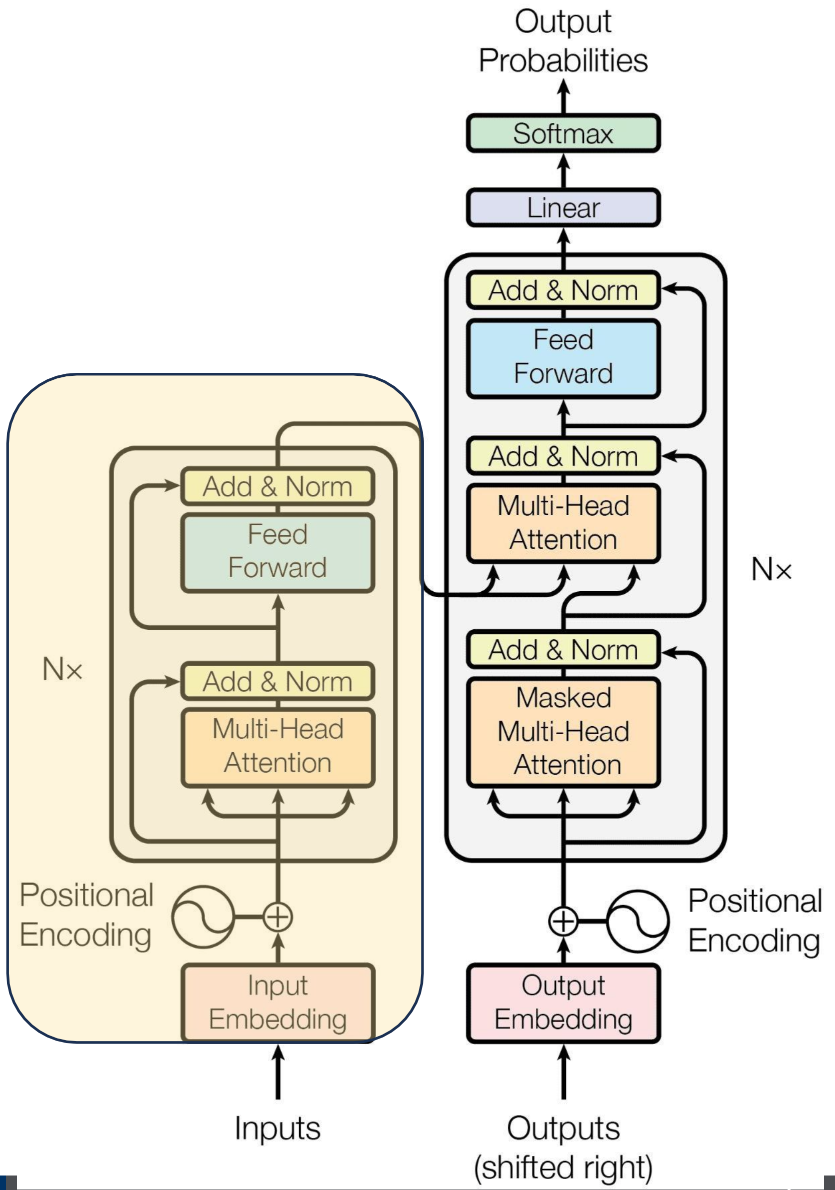
أكاديمية كاوست
KAUST ACADEMY

LMH
Lady Margaret Hall

- **BERT (Bidirectional Encoder Representations from Transformers)**
  - Uses transformer encoder only.
  - Trained with Masked Language Modeling (MLM).
  - **Bidirectional:** Considers both left and right context.
  - Fine-tuned for tasks like QA, classification.
  - **Architecture:**
    - Layers of encoder blocks.
    - Positional encodings added to embeddings.
    - Self-attention heads capture dependencies.

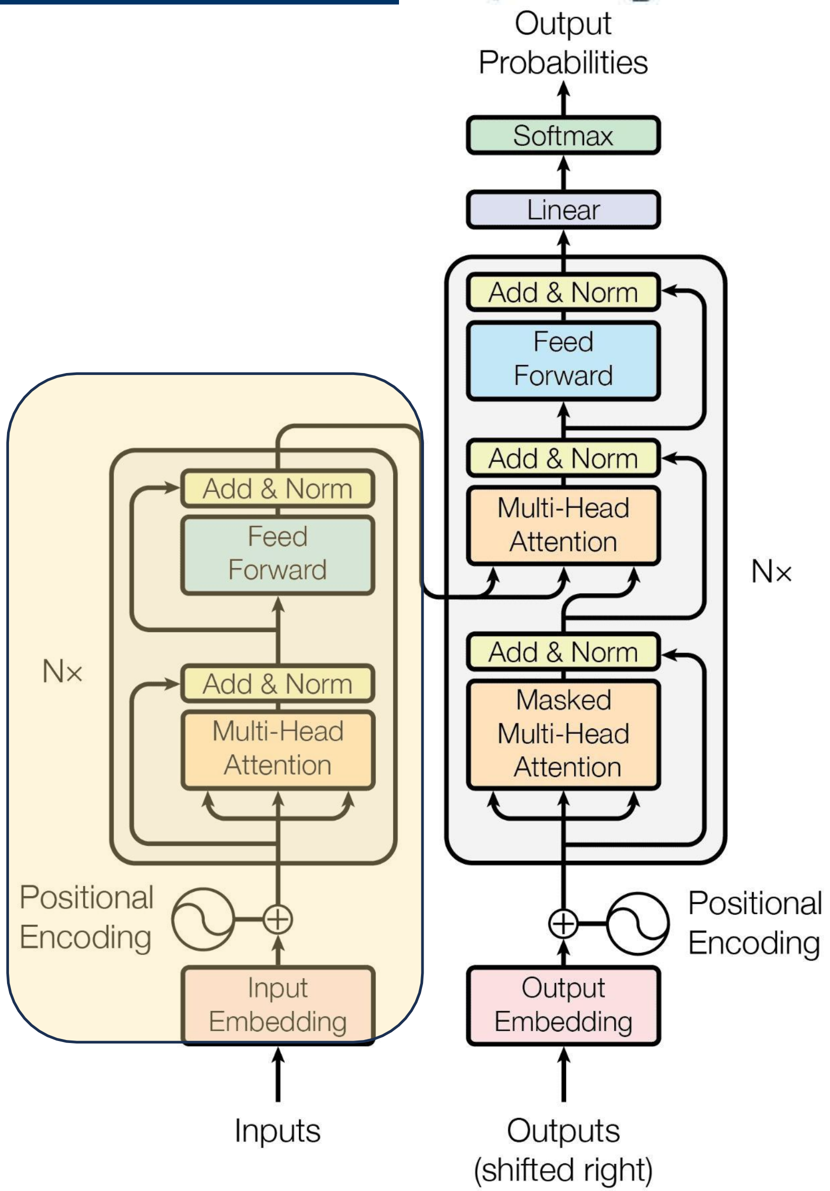
BERT - Bidirectional Encoder **Representations**

- 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 Pre-Training Corpus:

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



KAUST ACADEMY أكاديمية كاوست

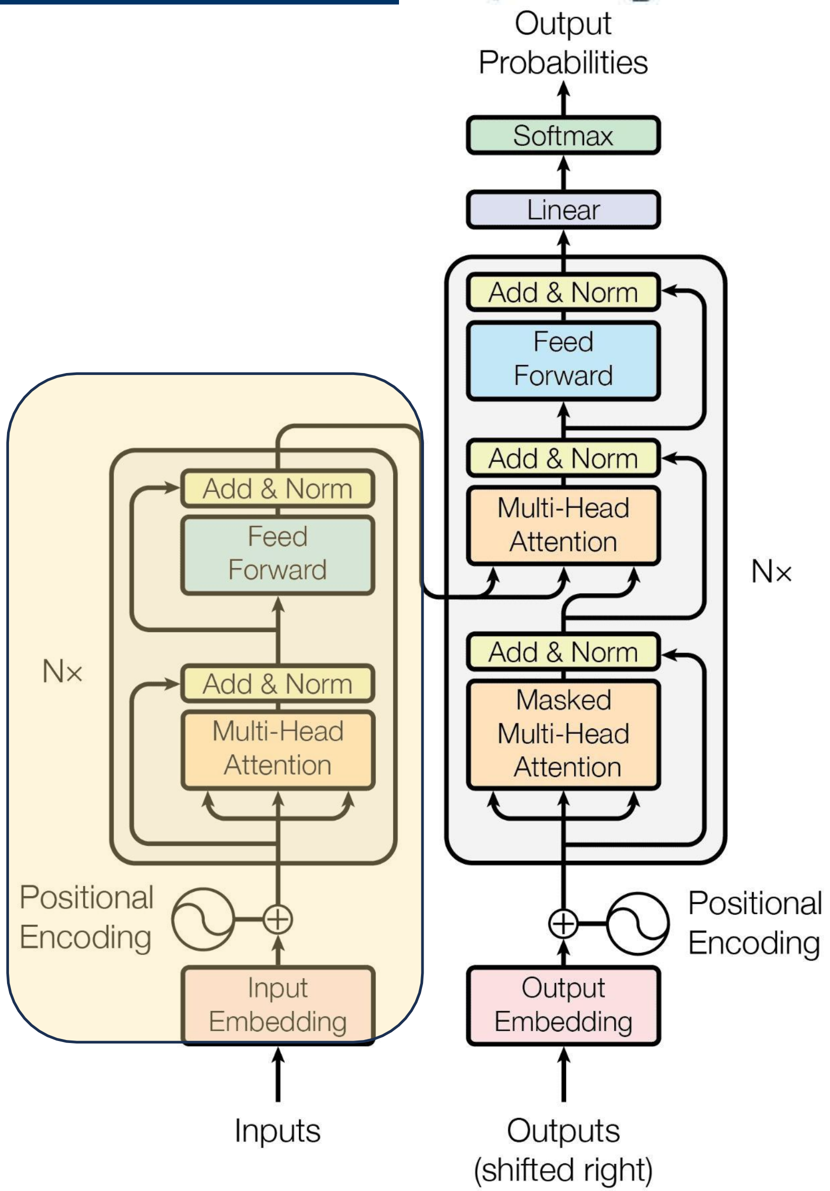
Lady Margaret Hall LMH

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

أكاديمية كاوست
KAUST ACADEMY

Lady Margaret Hall

BERT Pre-Training Corpus:

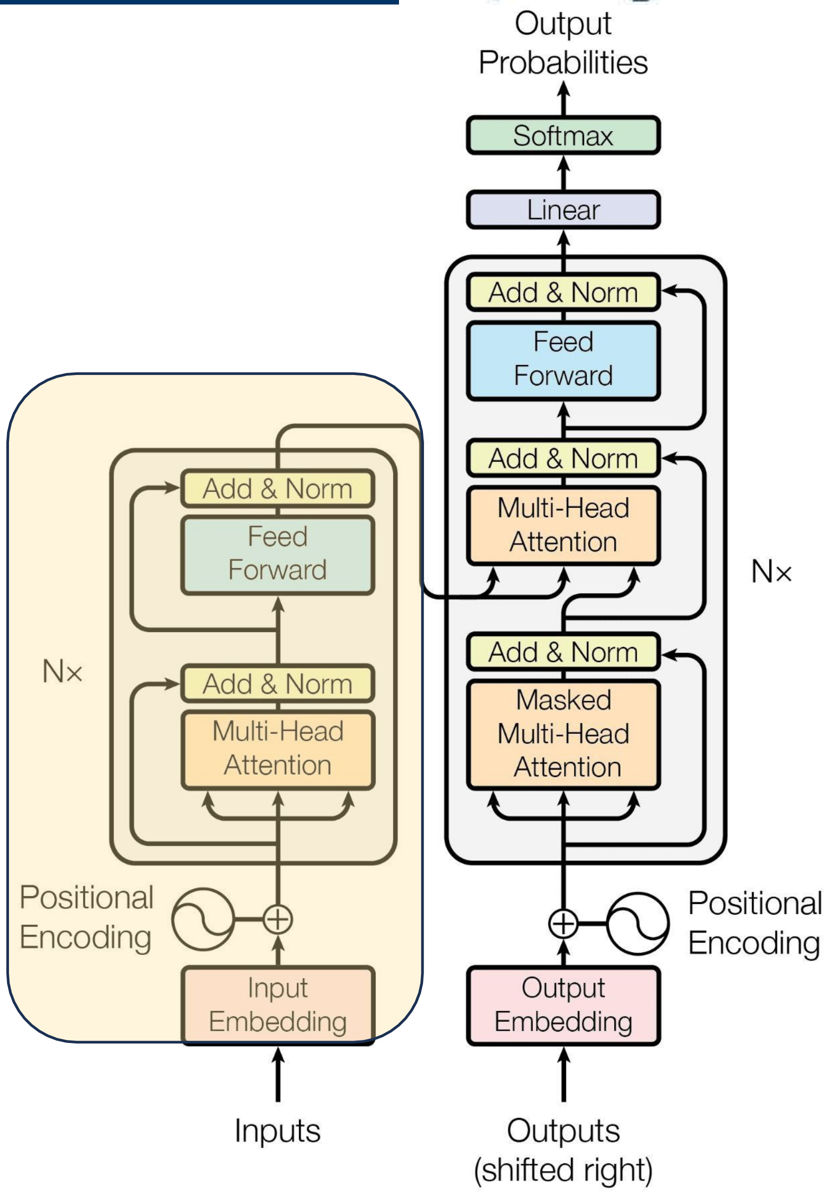
- English Wikipedia - 2,500 million words
- Book Corpus - 800 million words

BERT Pre-Training Tasks:

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

BERT Pre-Training Results:

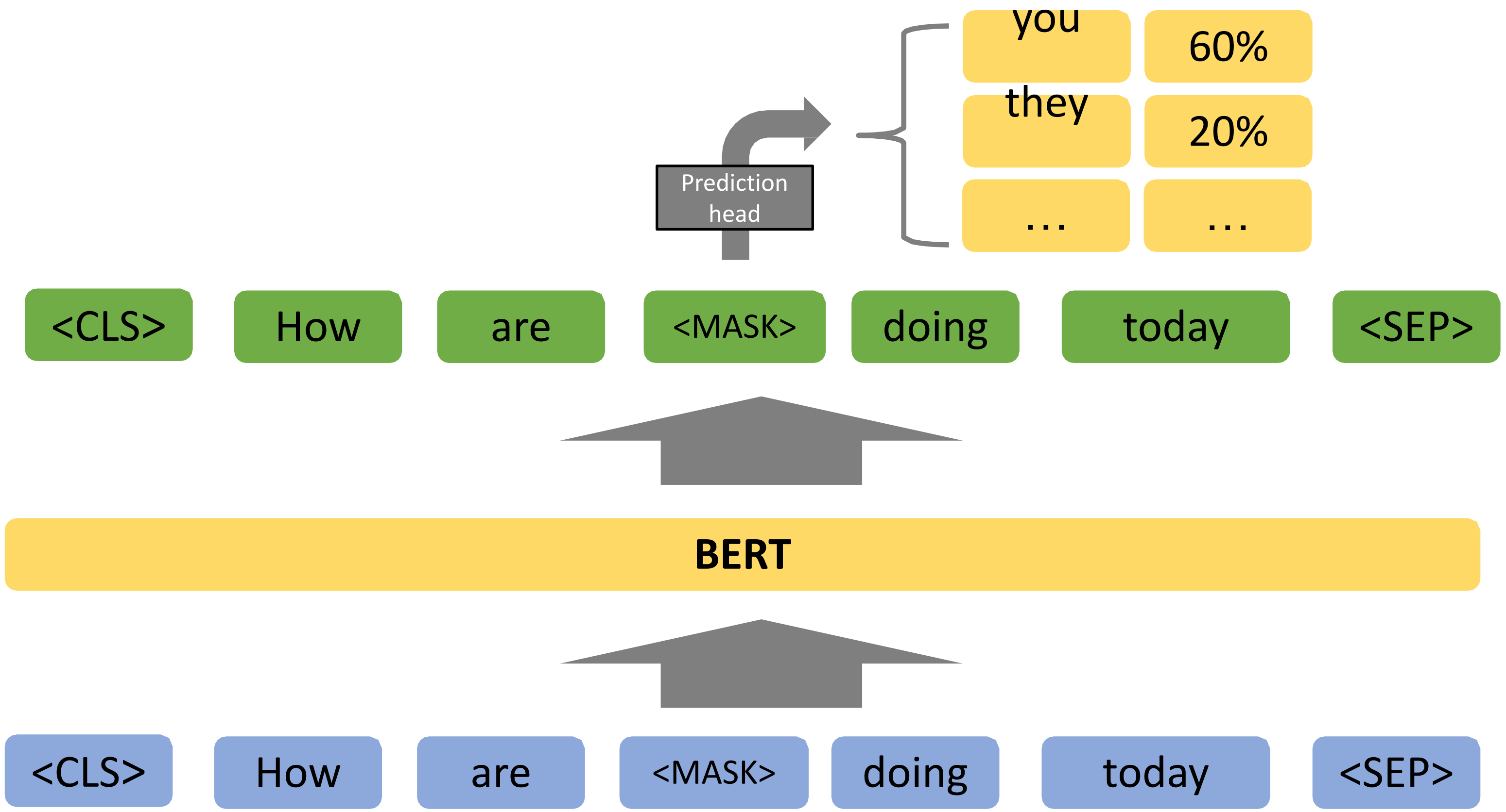
- BERT-Base – 110M Params
- BERT-Large – 340M Params

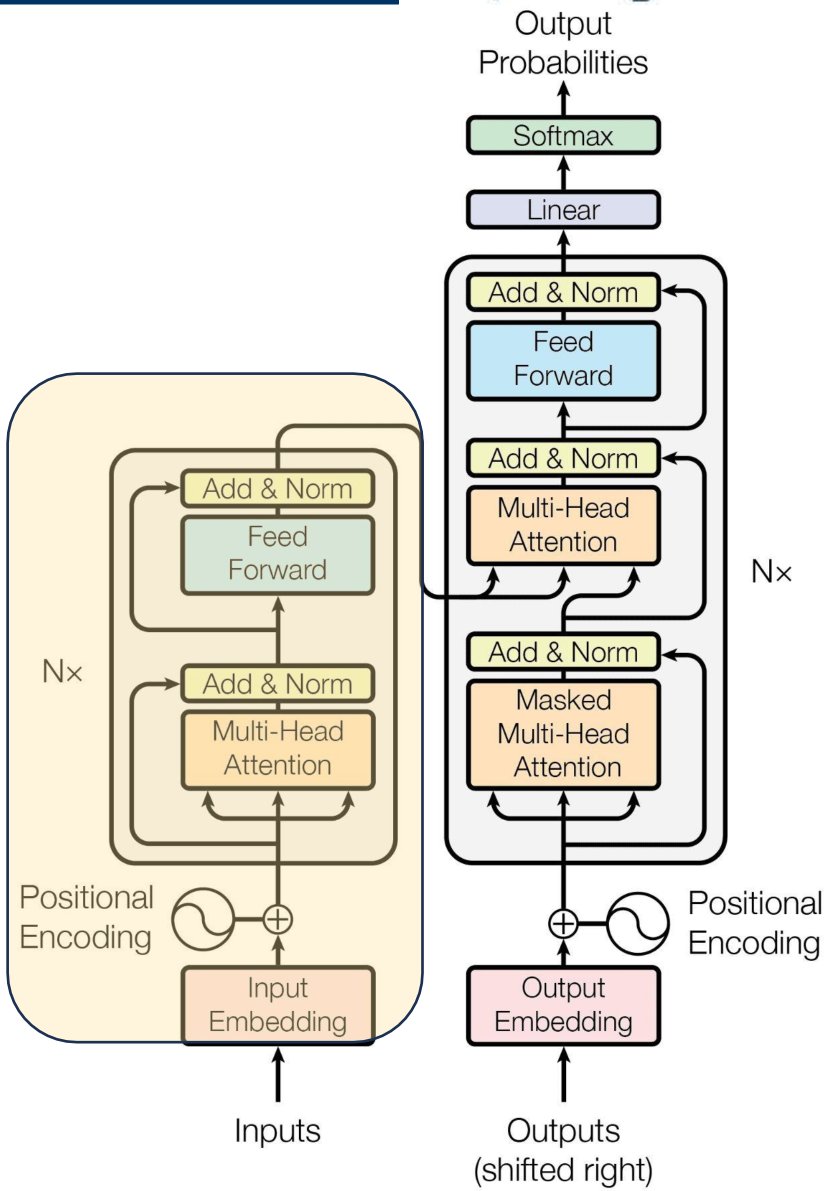


أكاديمية كاوست
KAUST ACADEMY

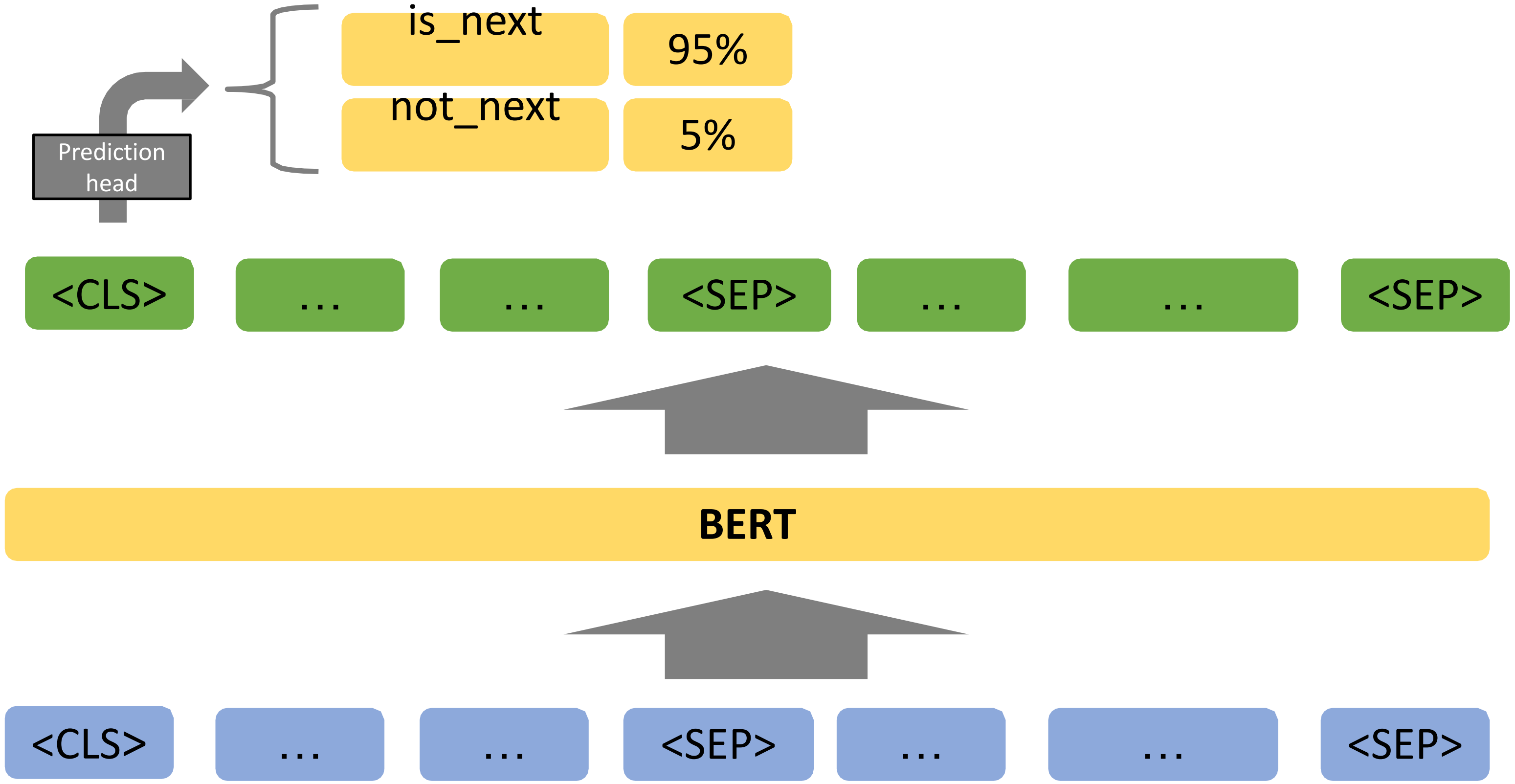
LMH
Lady Margaret Hall

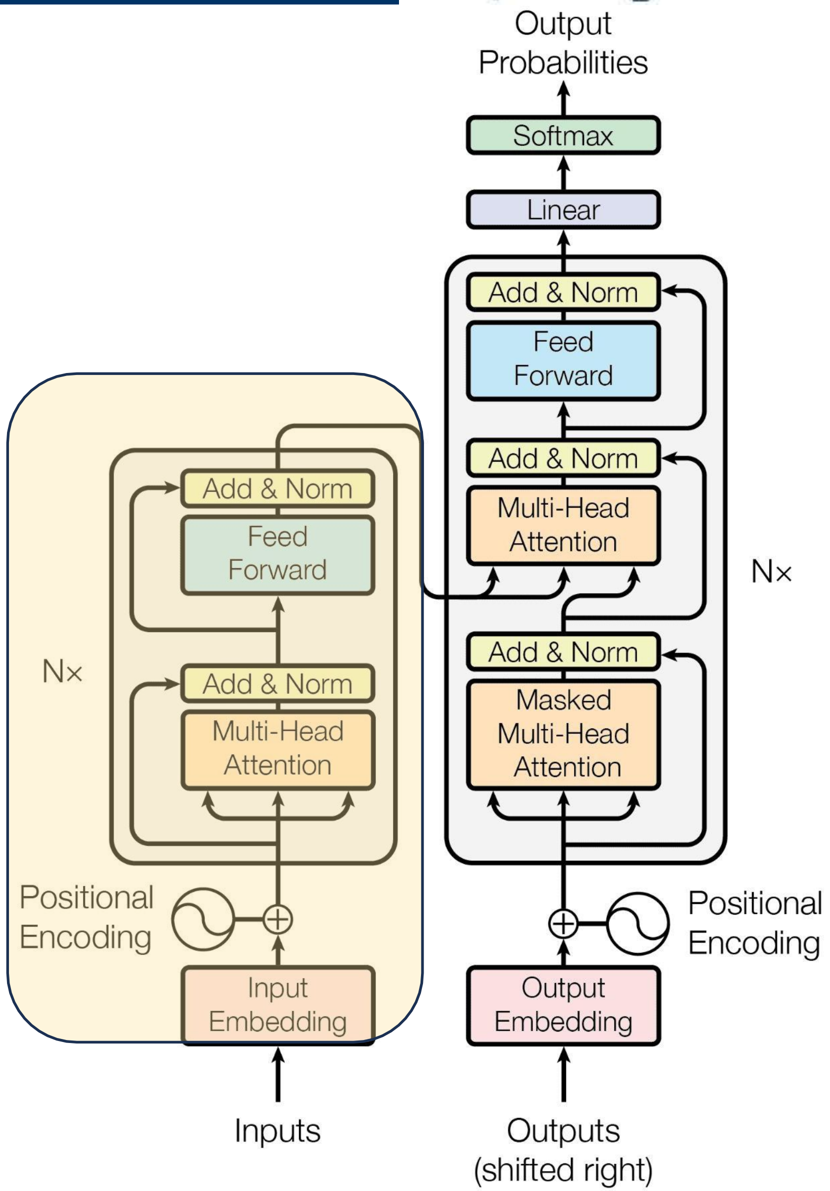
MLM (Masked Language Modeling)





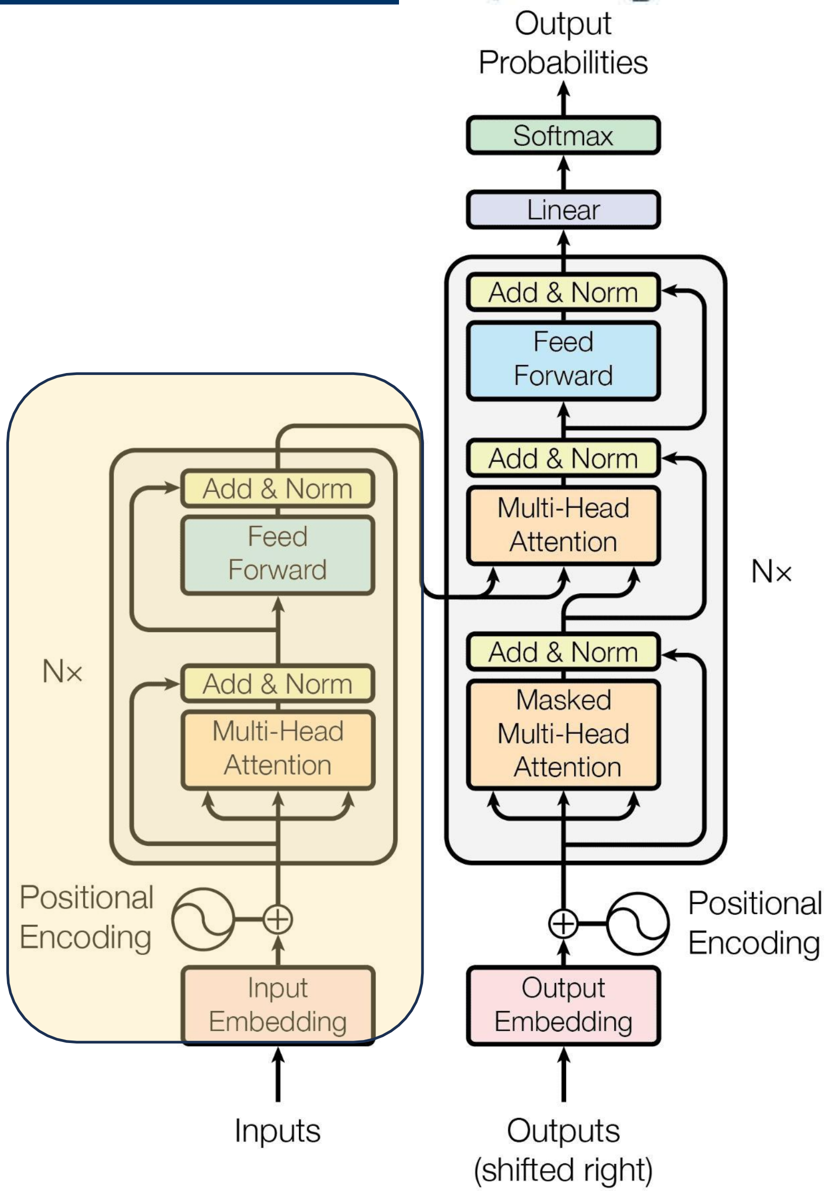
NSP (Next Sentence Prediction)





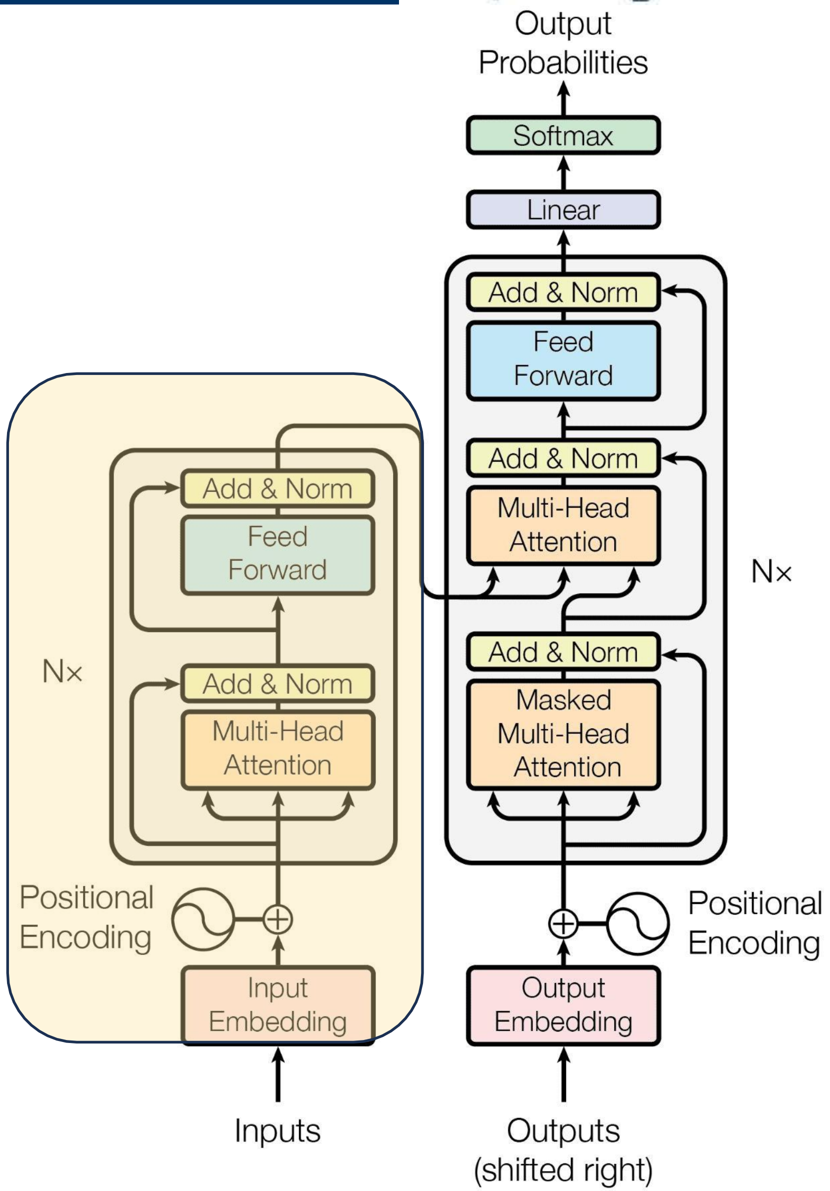
BERT Fine-Tuning:

- 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 Evaluation:

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



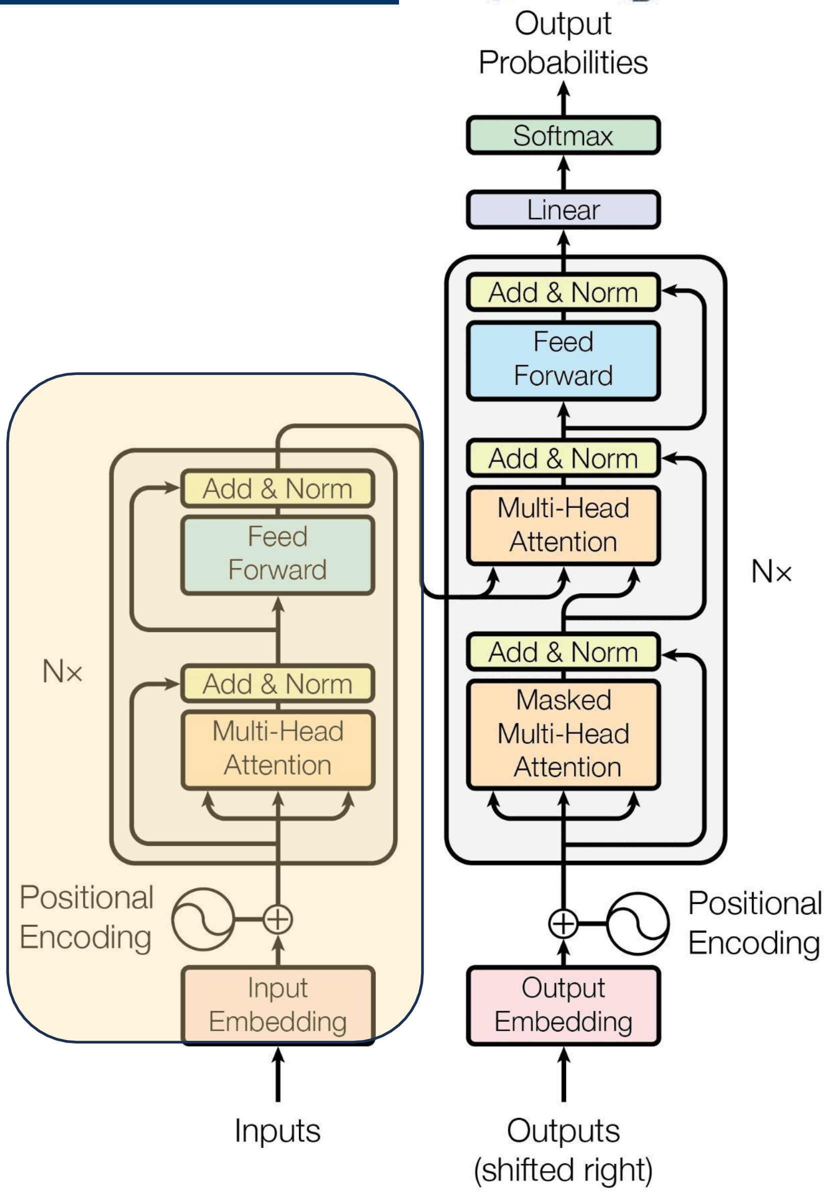
BERT - Bidirectional Encoder **Representations**

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**

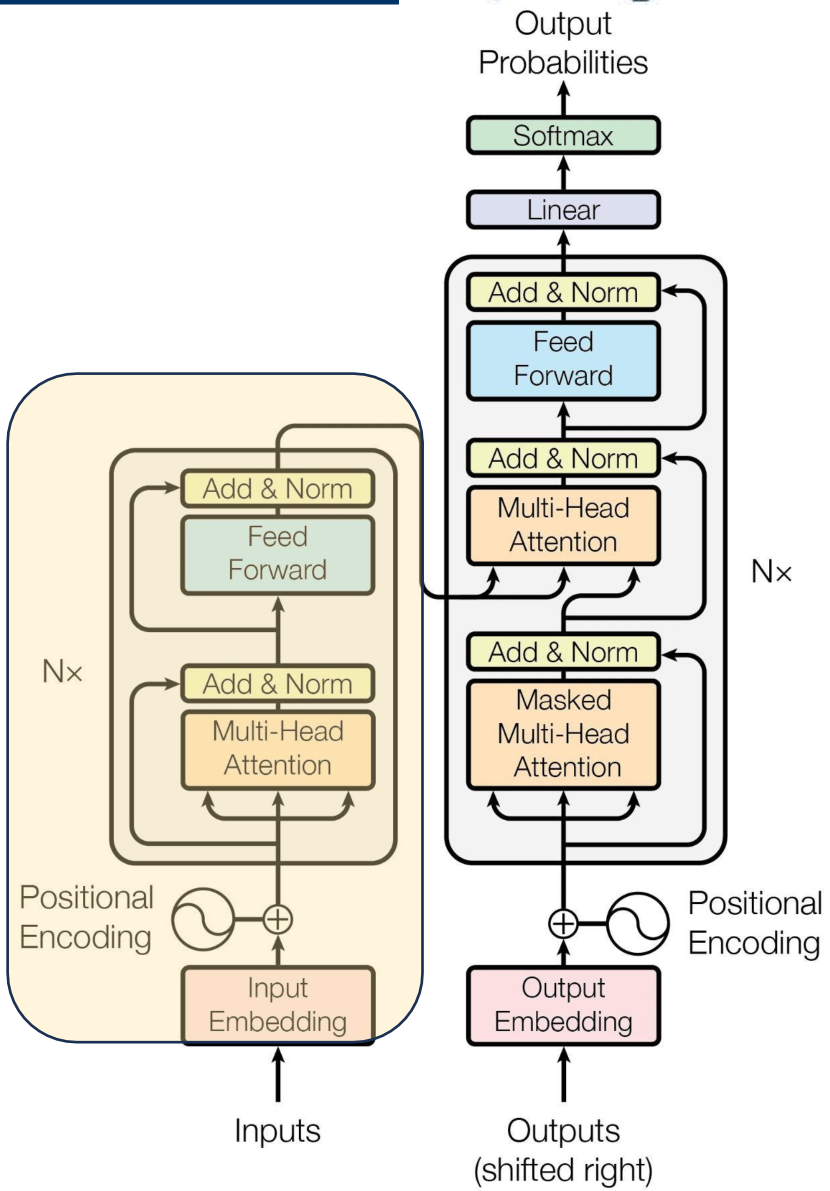
System	Dev EM	Dev F1	Test EM	Test F1
Leaderboard (Oct 8th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
#1 Single - nlnet	-	-	83.5	90.1
#2 Single - QANet	-	-	82.5	89.3
Published				
BiDAF+ELMo (Single)	-	85.8	-	-
R.M. Reader (Single)	78.9	86.3	79.5	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.



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.



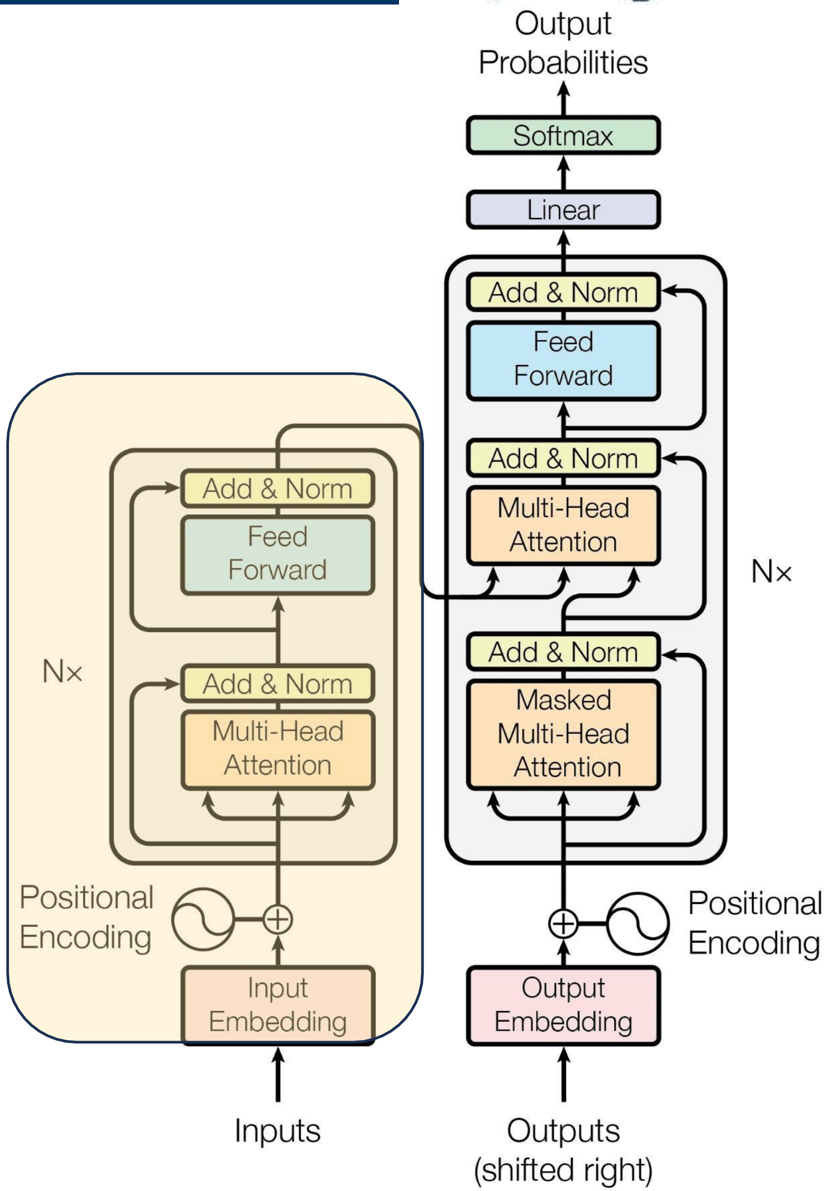
أكاديمية كاوست
KAUST ACADEMY

LMH
Lady Margaret Hall

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.

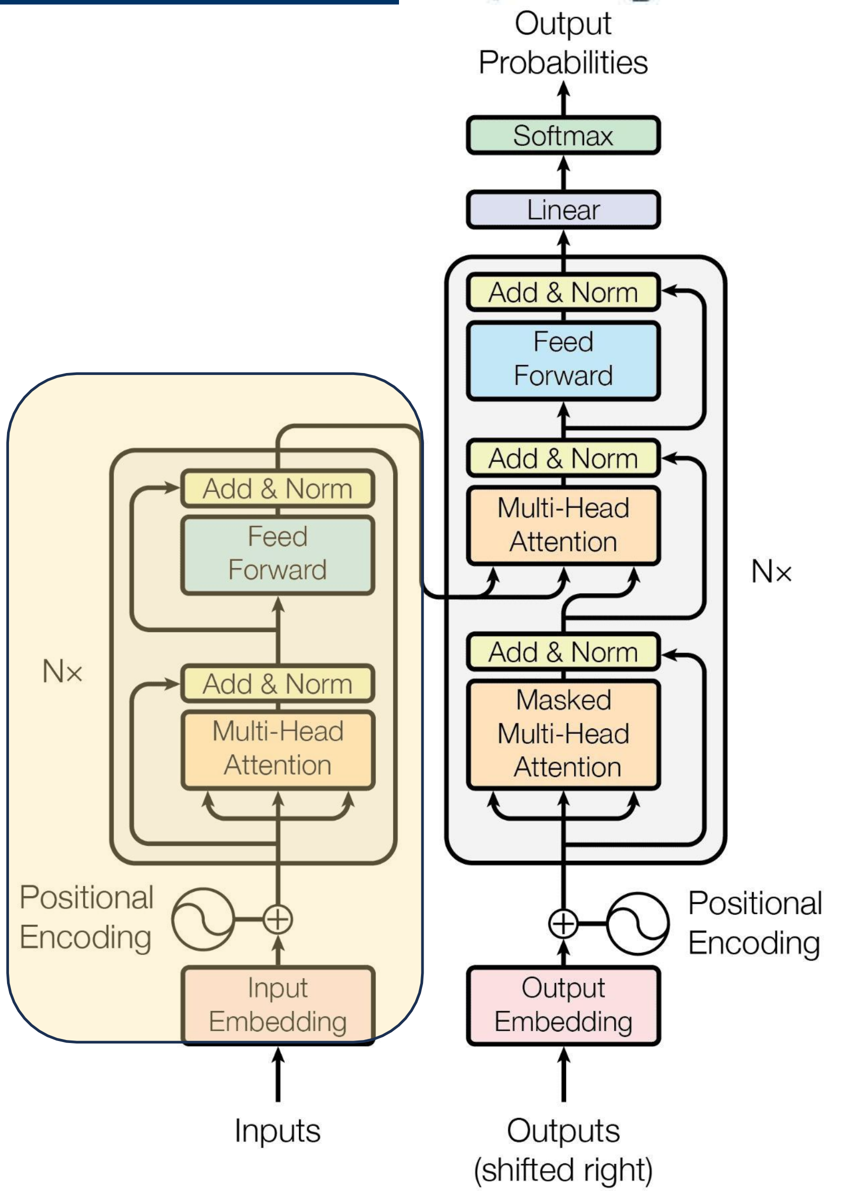
- **Different NLP tasks seem to be highly transferable with each other...**
  - As long as we have effective representations, that seems to form a general model which can serve as the backbone for many specialized models.



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



BERT and GPT Concepts

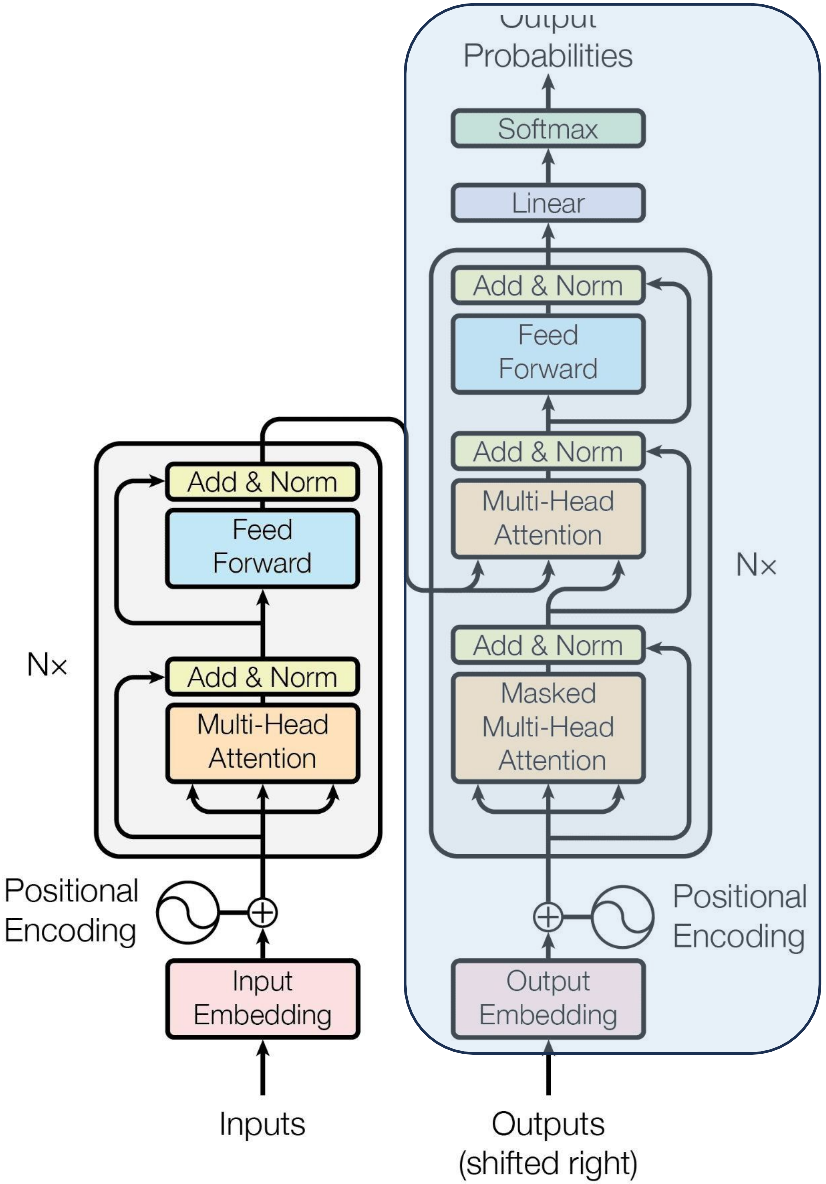
أكاديمية كاوست
KAUST ACADEMY

LMH
Lady Margaret Hall

- **GPT (Generative Pre-trained Transformer)**
  - Uses transformer decoder only.
  - Trained with next-token prediction.
  - **Unidirectional:** Considers left context only.
  - Fine-tuned for text generation, dialogue systems.
  - **Architecture:**
    - Layers of decoder blocks.
    - Causal masking to prevent future token access.
    - Self-attention heads capture sequential dependencies.

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?

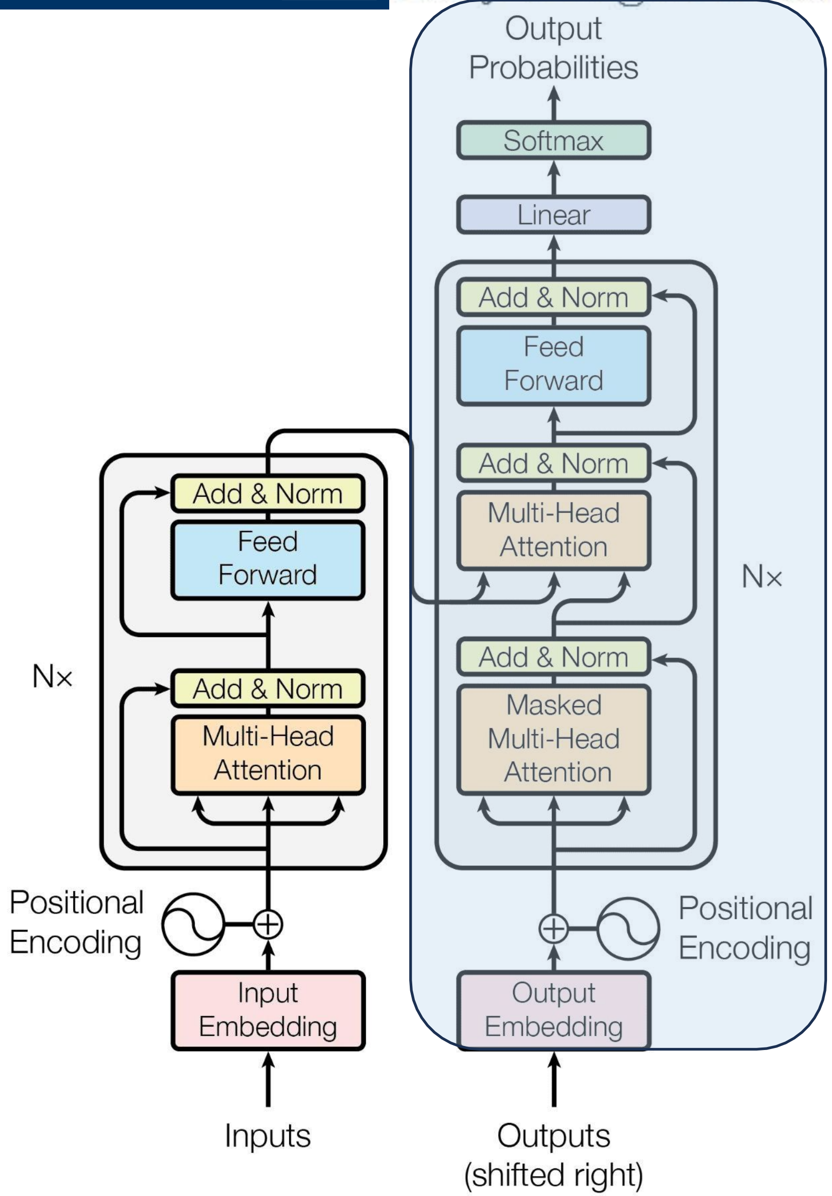


GPT – Generative Pretrained Transformer

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 – Generativ e 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:

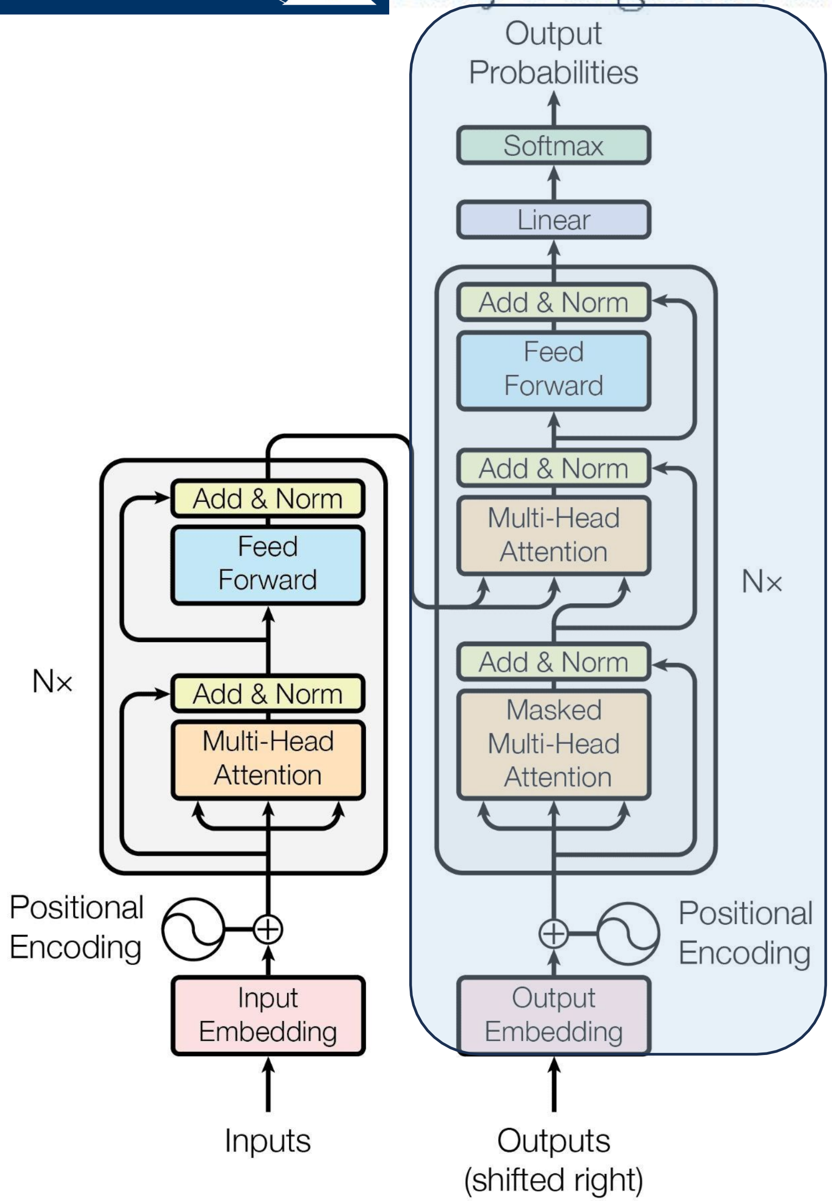
QA

Answer the question based on the context.

Context:

Question:

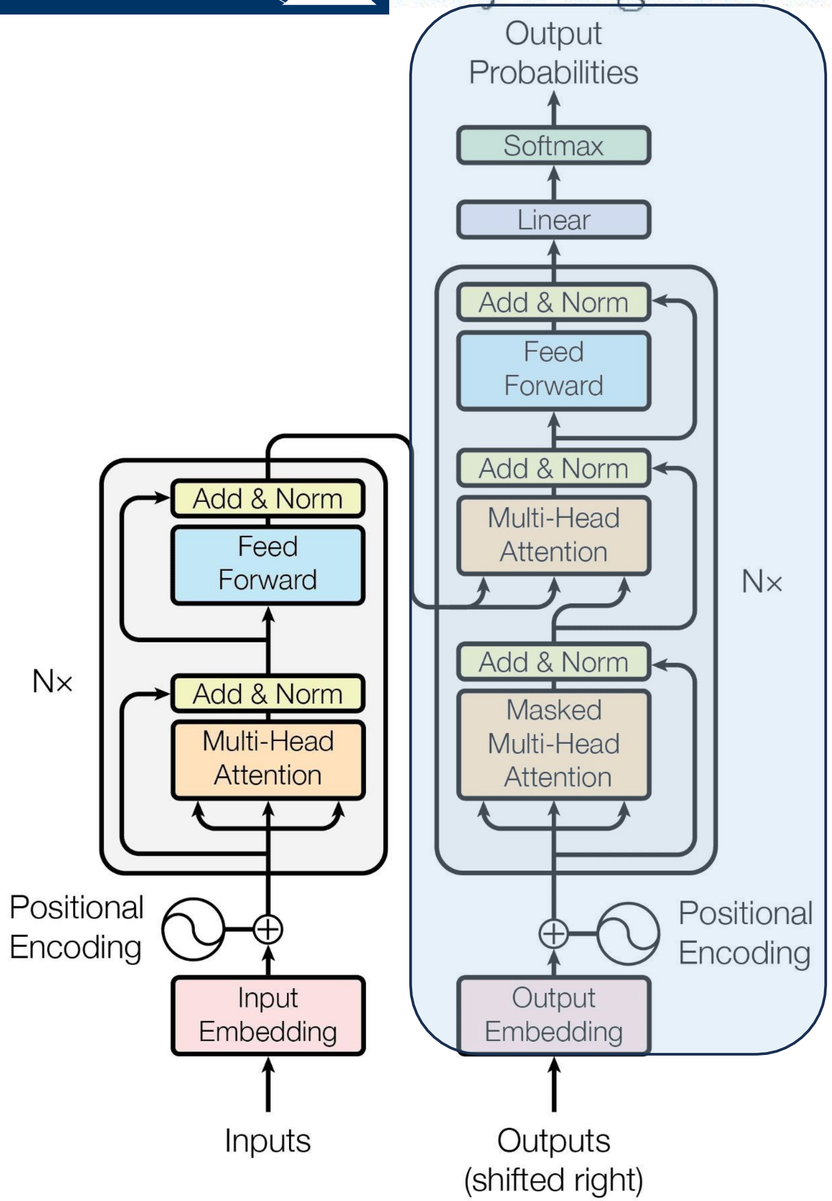
Answer:



GPT – Generative Pretrained Transformer

What is our takeaway from GPT?

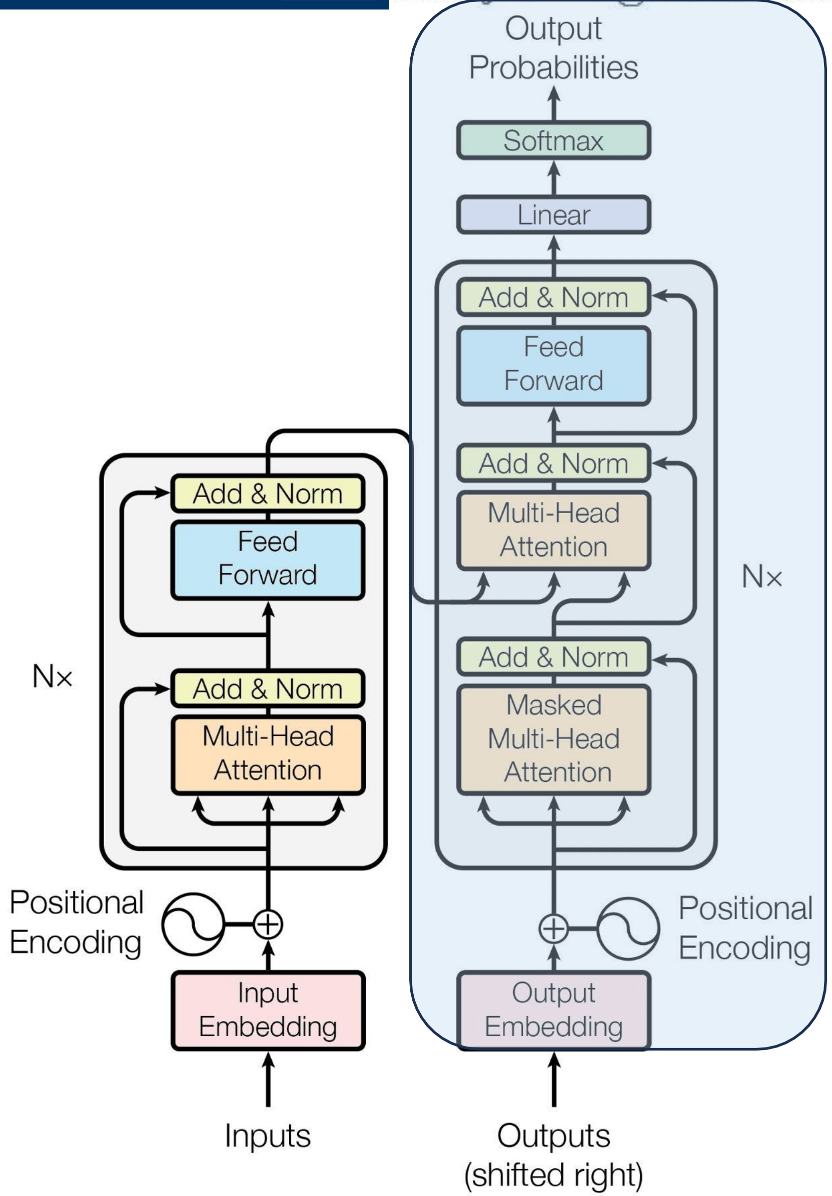
- **The Effectiveness of Self-Supervised Learning**
  - Specifically, the model seems to be able to learn from generating the language *itself*, rather than from any specific task we might cook up.



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.



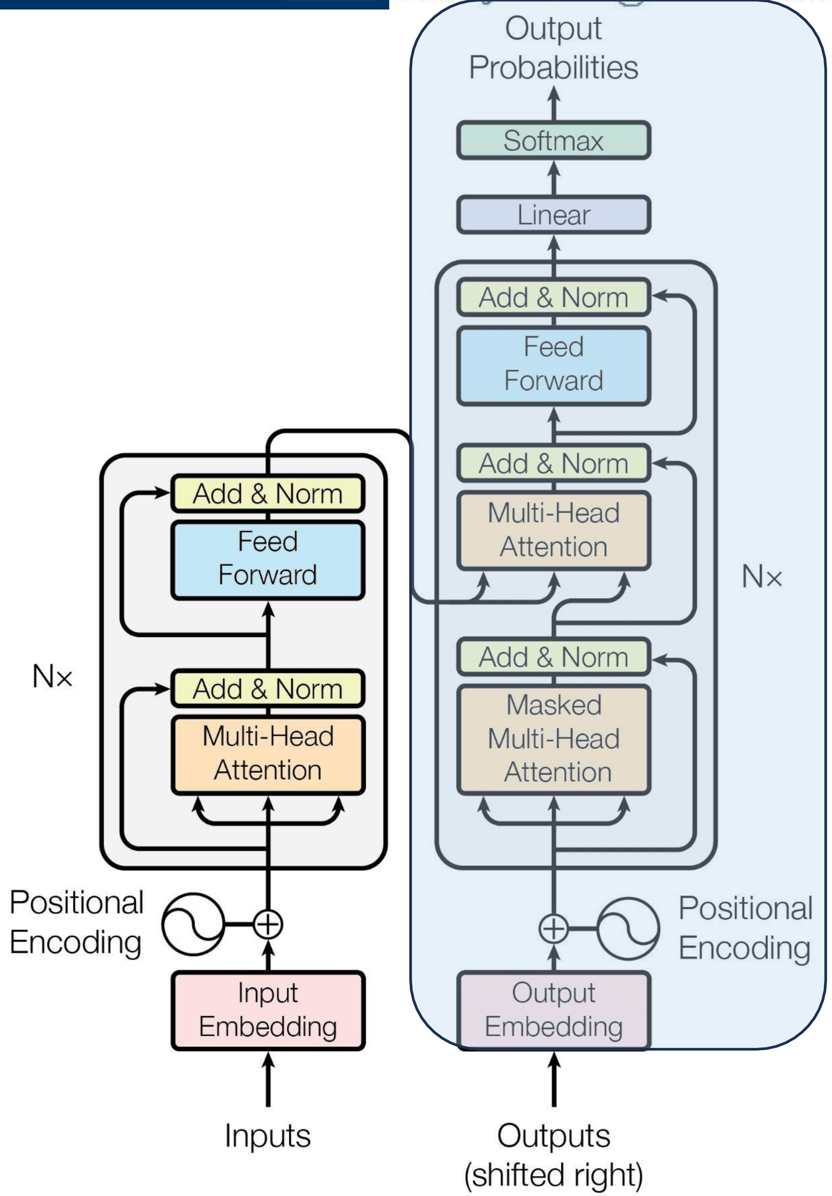
GPT – Generative Pretrained Transformer

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!!!**



BERT and GPT Concepts

- **Key Differences:**

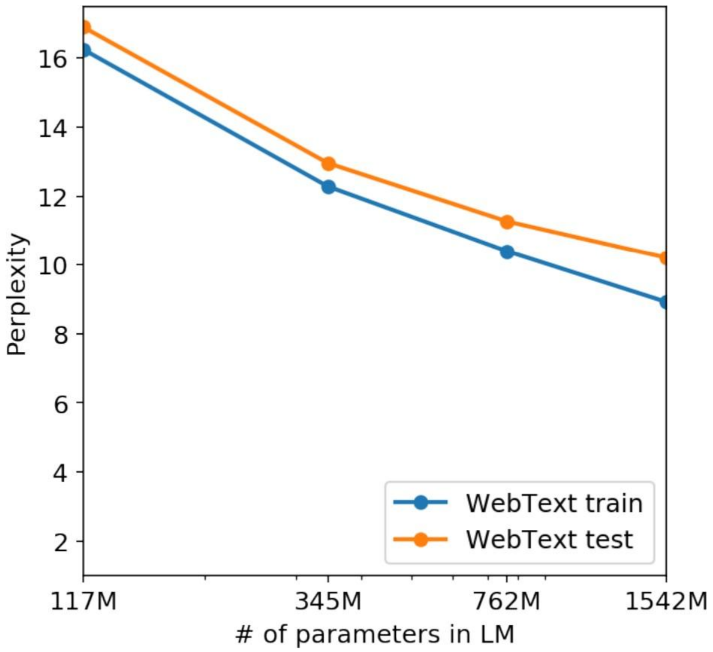
Feature	**BERT**	**GPT**
Directionality	Bidirectional	Unidirectional
Objective	Masked Language Modeling (MLM)	Causal Language Modeling (CLM)
Output	Contextual embeddings	Text generation
Usage	Downstream tasks (e.g., classification, QA)	Generation, few-shot learning

Scaling Laws for LLMs

- **Kaplan et al. (2020):** "Scaling Laws for Neural Language Models"
- Performance improves predictably with:
  - More parameters
  - More compute
  - Larger datasets
- **Optimal allocation of compute:** Train bigger models with less data, rather than small models with lots of data.
- **Implication:** LLMs like GPT-3 (175B), GPT-4 (est. >500B) are products of scaling laws.

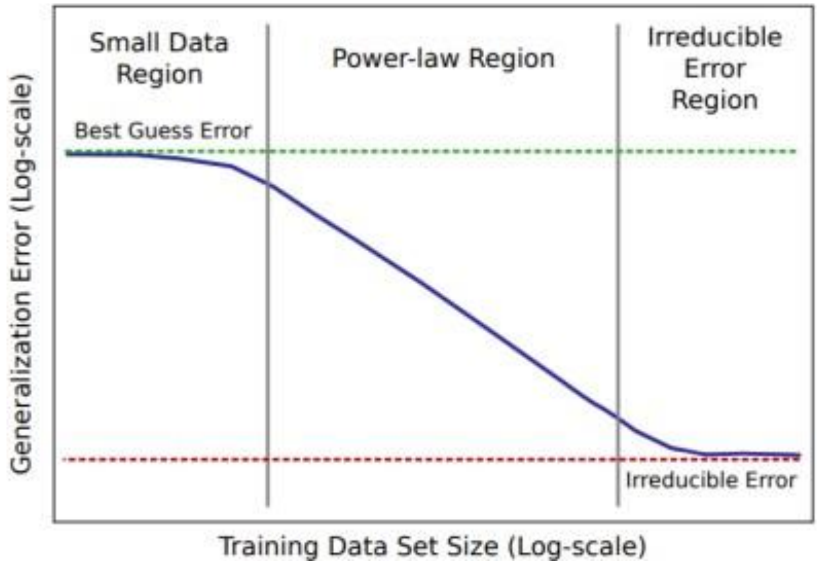
Scaling in GPT-2

- Scaling improves the perplexity of the LM and improves performance



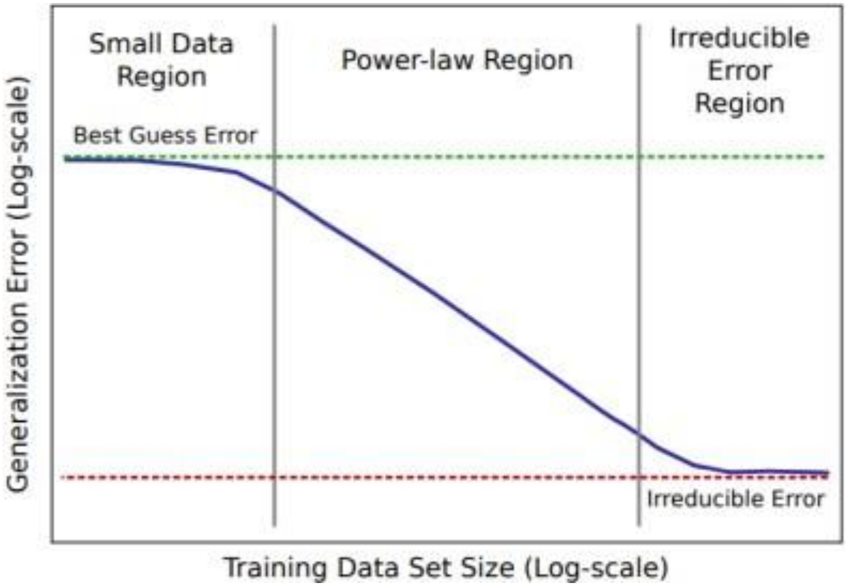
Why is this interesting? Look at data scaling

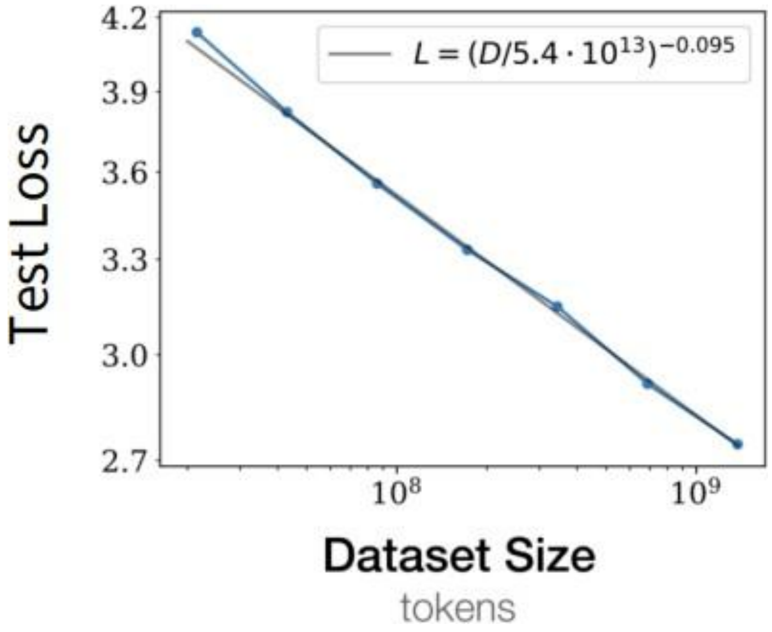
- We know that typical scaling effects look like this when we increase the amount of training data



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)

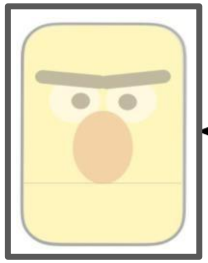
- Key Findings:
  - Performance depends strongly on scale, and weakly on the model shape
  - Larger models are more sample-efficient
  - Smooth power laws ($y = ax^k$) b/w empirical performance & N - parameters, D - dataset size, C - compute

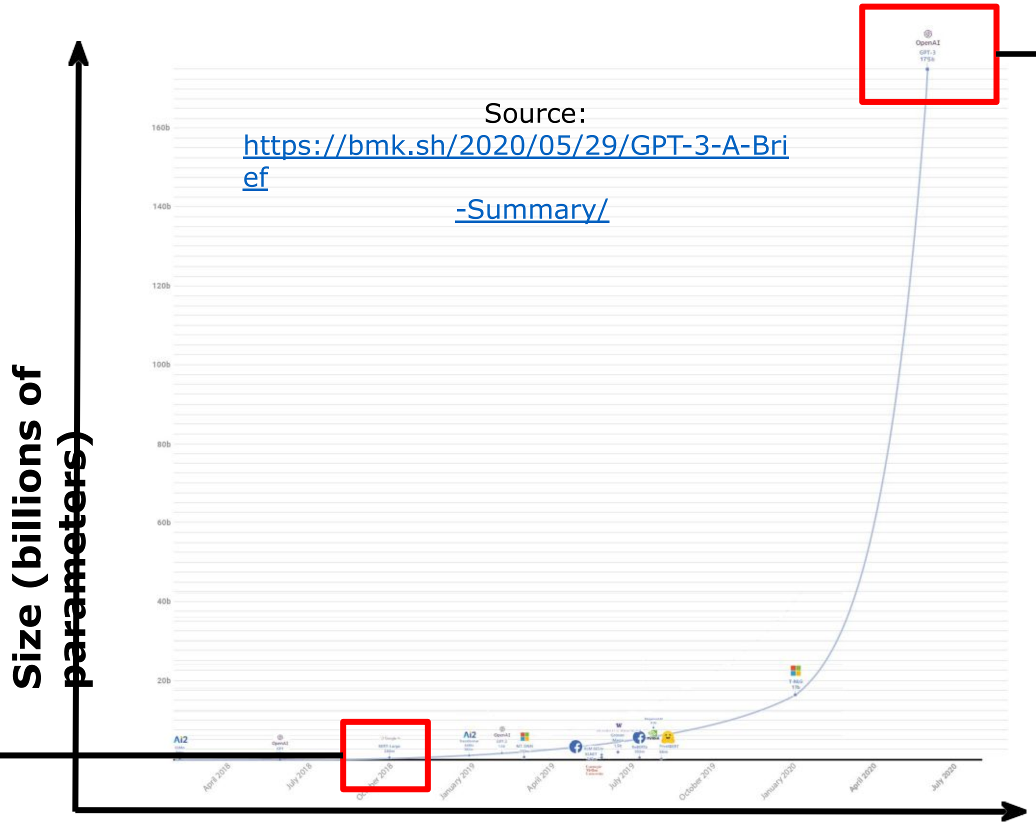
Scaling Effects

- 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





Source: https://bmk.sh/2020/05/29/GPT-3-A-Brief-Summary/

175b params! GPT-2 was 1.5b

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

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

أكاديمية كاوست
KAUST ACADEMY

LMH
Lady Margaret Hall

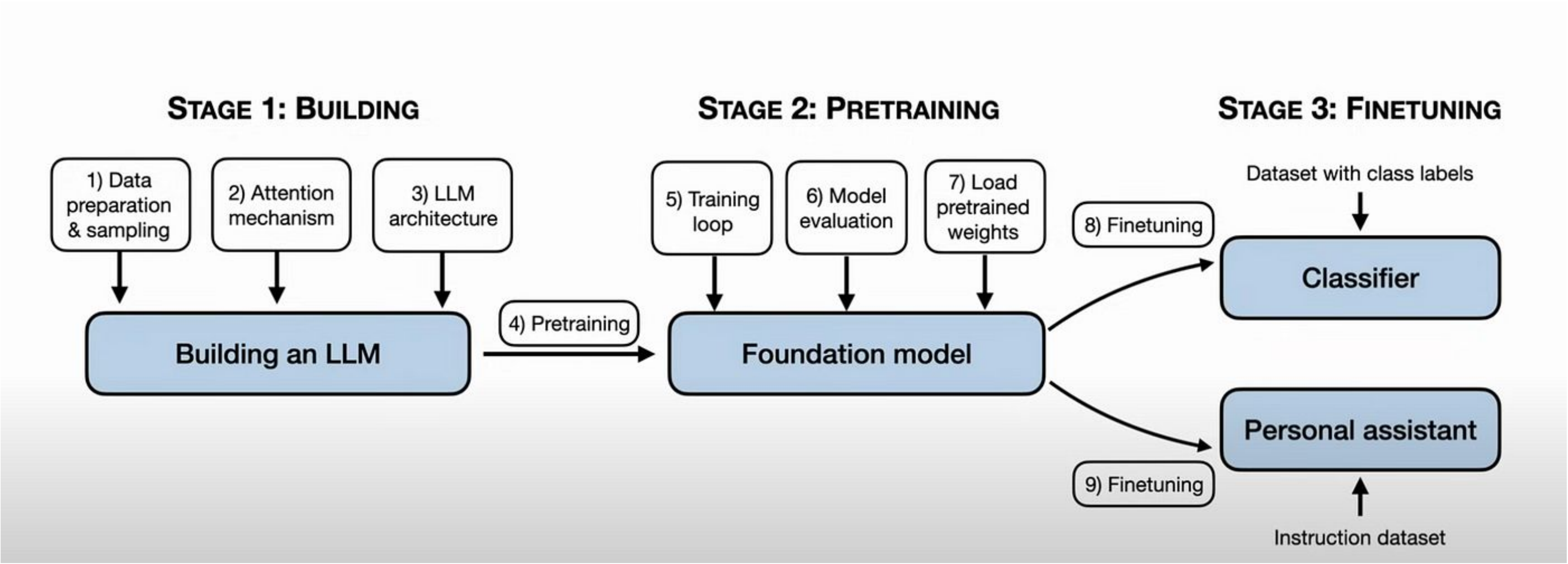
- Emergent abilities:
  - not present in smaller models but is present in larger models
  - Do LLMs like GPT3 have these ?

- 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

Pre-training Overview

- **Pre-training Phase:**
  - Large-scale unsupervised training on corpus (e.g., Common Crawl, Books)
  - Objective: learn general-purpose language representations
- **Why Pre-train?**
  - Data-efficient fine-tuning
  - Enables zero-shot and few-shot capabilities
  - Foundation for instruction tuning, alignment
- **Challenges:**
  - Massive compute costs
  - Environmental concerns (carbon footprint)

Pre-training Overview



Training of Decoder-only LLMs – Llama 2

أكاديمية كاوست
KAUST ACADEMY

LMH
Lady Margaret Hall

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

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

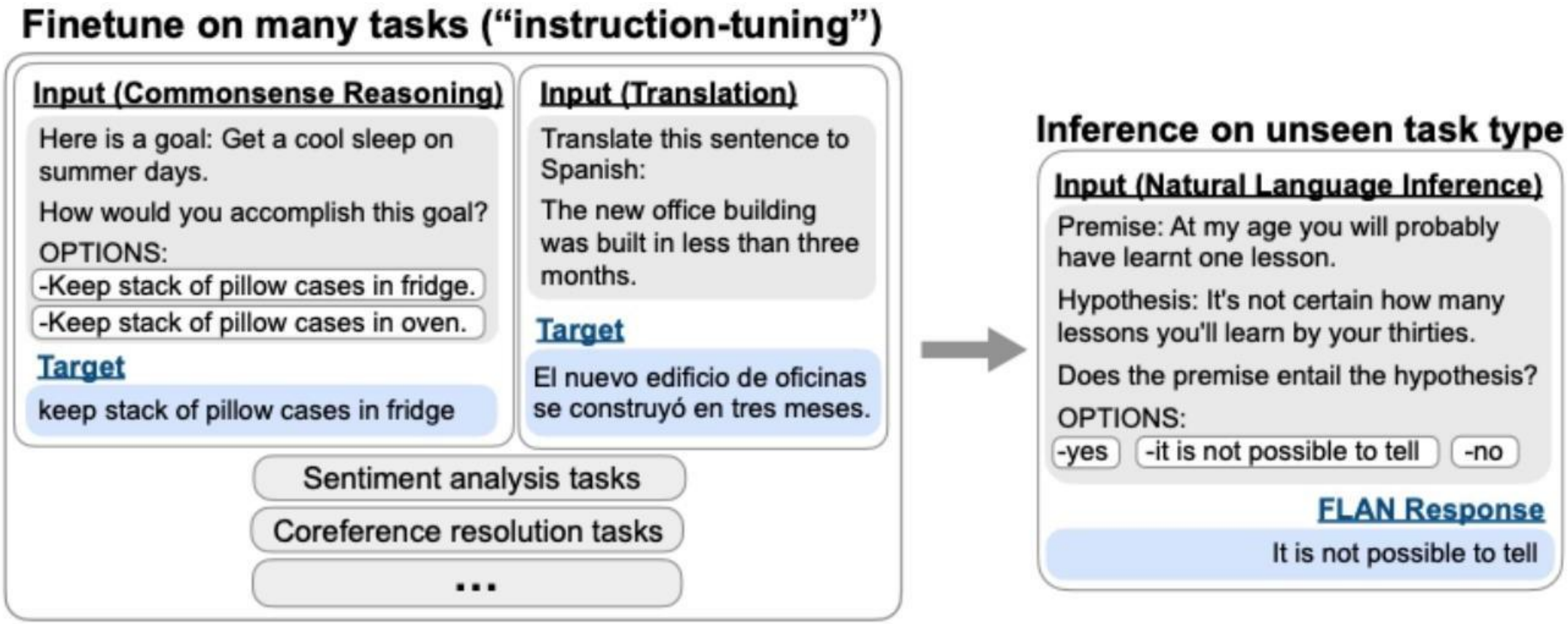
Instruction fine-tuning provides as a prefix a natural language description of the task along with the input.

- 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 in SFT and only input as input with shifted input as target
- 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

Tokenization: Why It Matters?

- **Language $\neq$ Input**
  - Text must be converted to numbers $\rightarrow$ tokens.
- **Tokenization:**
  - Splits text into manageable units.
  - Balances granularity and vocabulary size.
- **Good tokenizer =**
  - Efficient sequence length
  - High vocabulary coverage
  - Robust across domains (code, multilingual, etc.)

Machine Translation

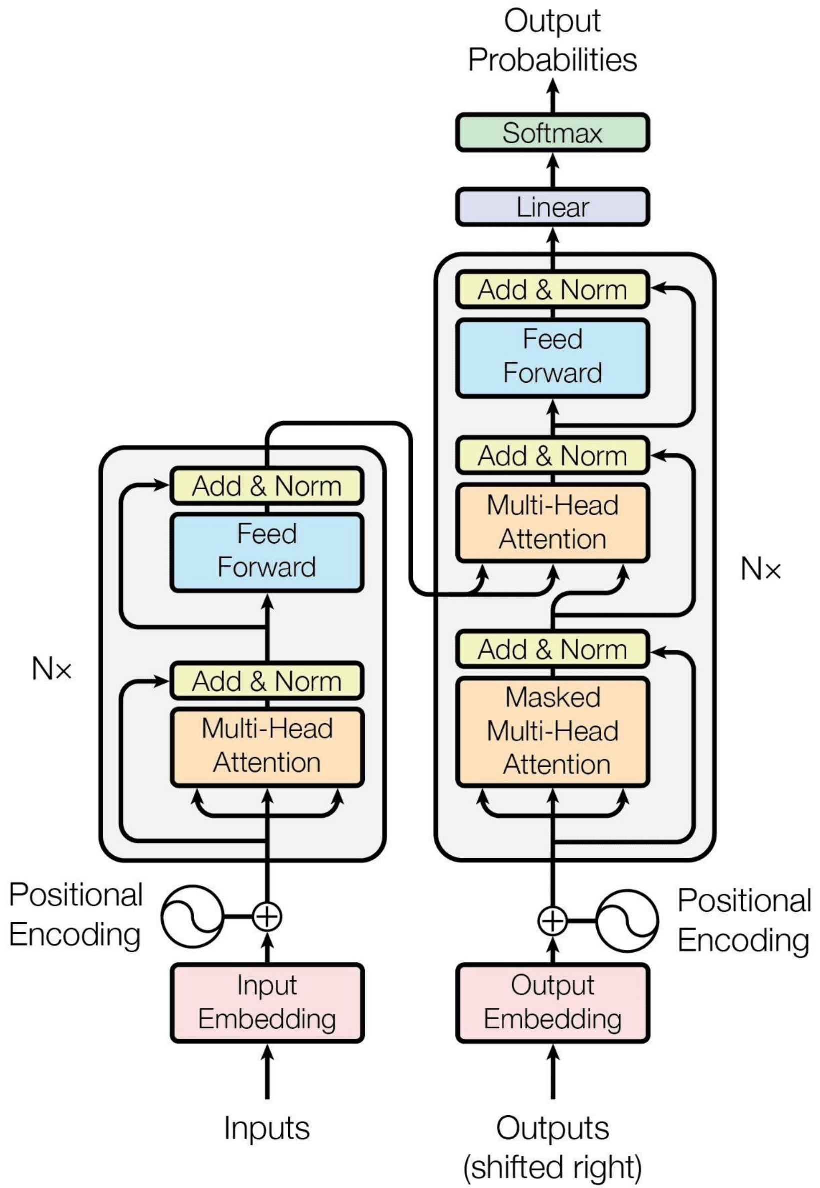
Targets

Ich habe einen
Apfel gegessen

⬆

Inputs

I ate an apple



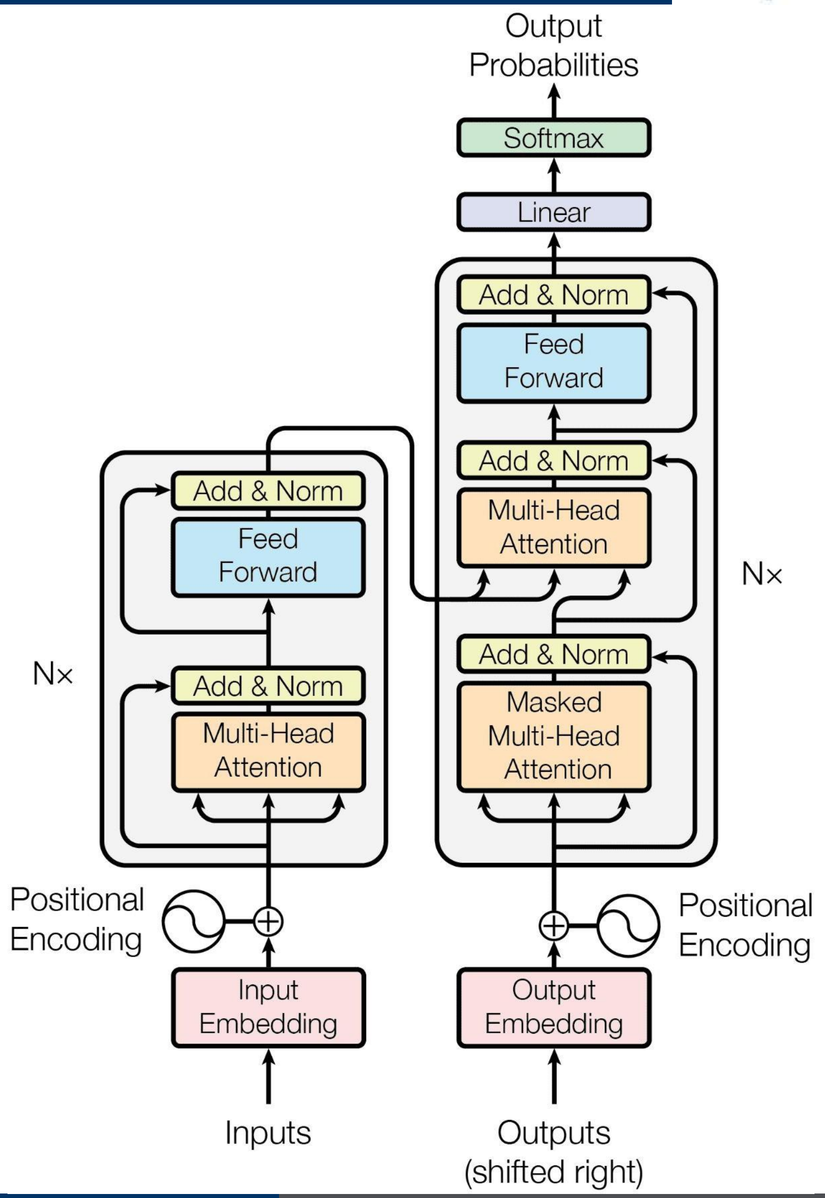
أكاديمية كاوست
KAUST ACADEMY

LMH
Lady Margaret Hall

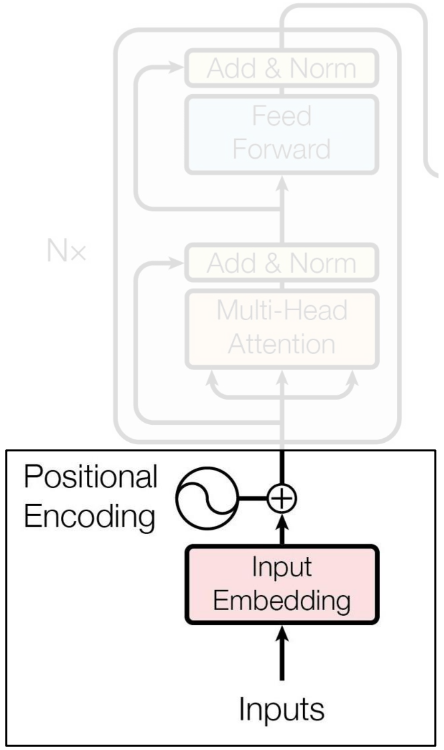
Processing Inputs

Inputs

I ate an apple



Tokenization



Tokenizer (split into individual words)

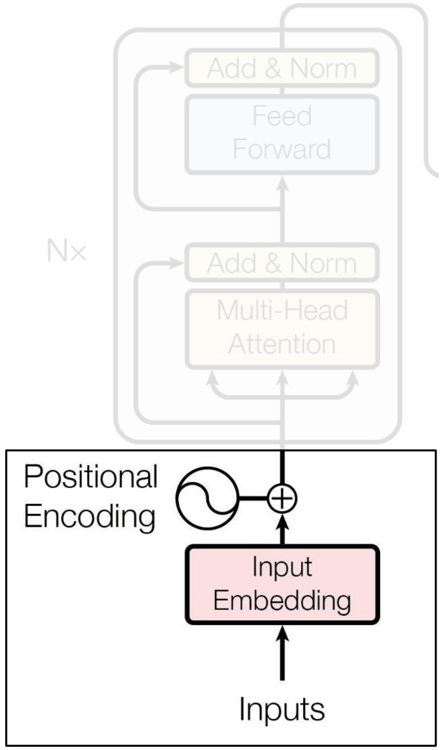
I ate an apple

Tokenization

I | ate | an | apple | <eos>

Tokenizer (split into individual words)

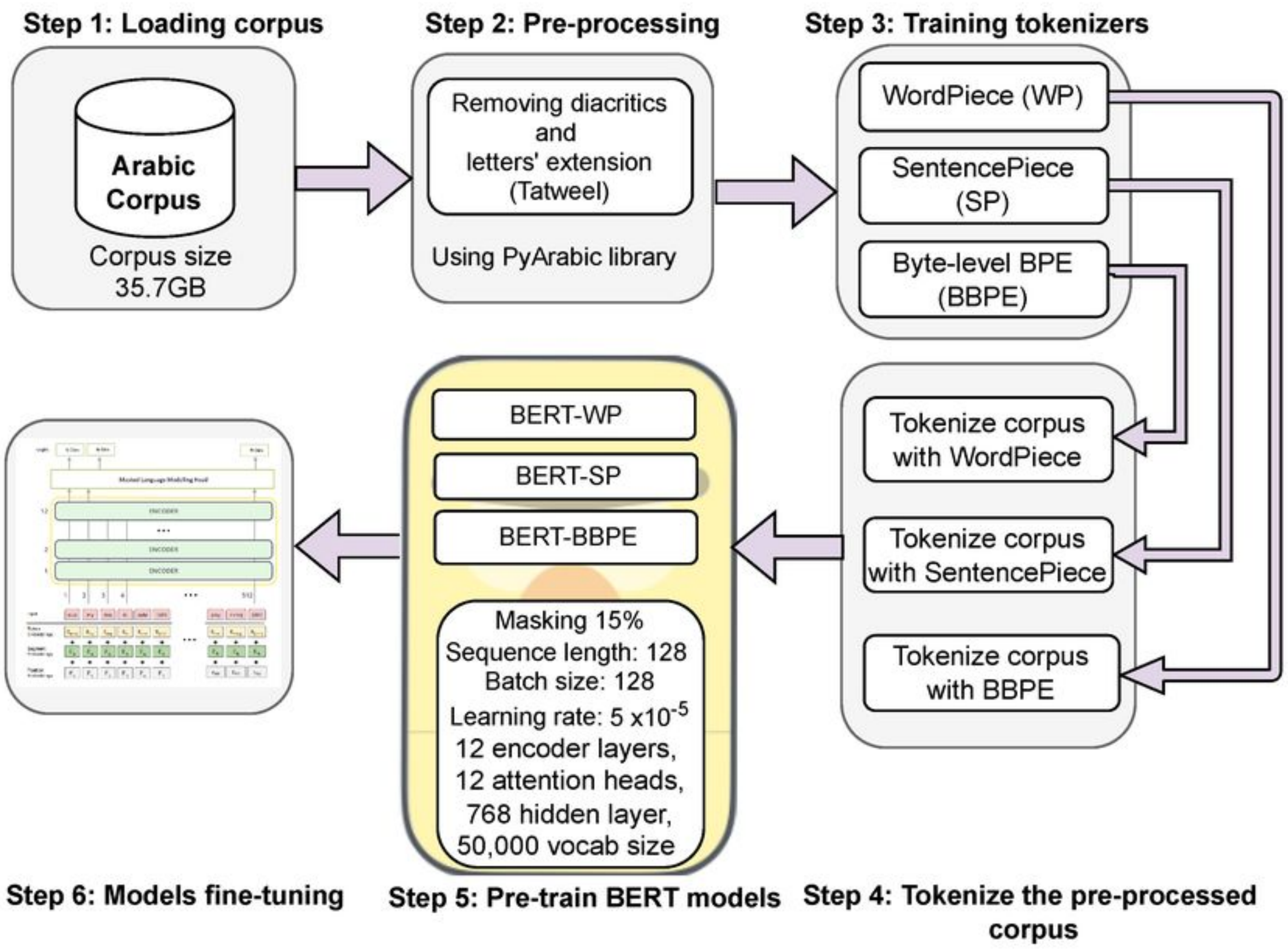
I ate an apple



Different Levels of Tokenization

- **Character-level**
  - Fine-grained, handles unknowns
  - Long sequences, slow training
- **Word-level**
  - Intuitive, natural boundaries
  - Large vocab, OOV (Out-of-Vocab) issues
- **Subword-level**
  - Balance of generalization & compactness
  - Requires segmentation algorithm
- **SentencePiece/Unigram LM**
  - Learned from data, language-agnostic

Different Levels of Tokenization

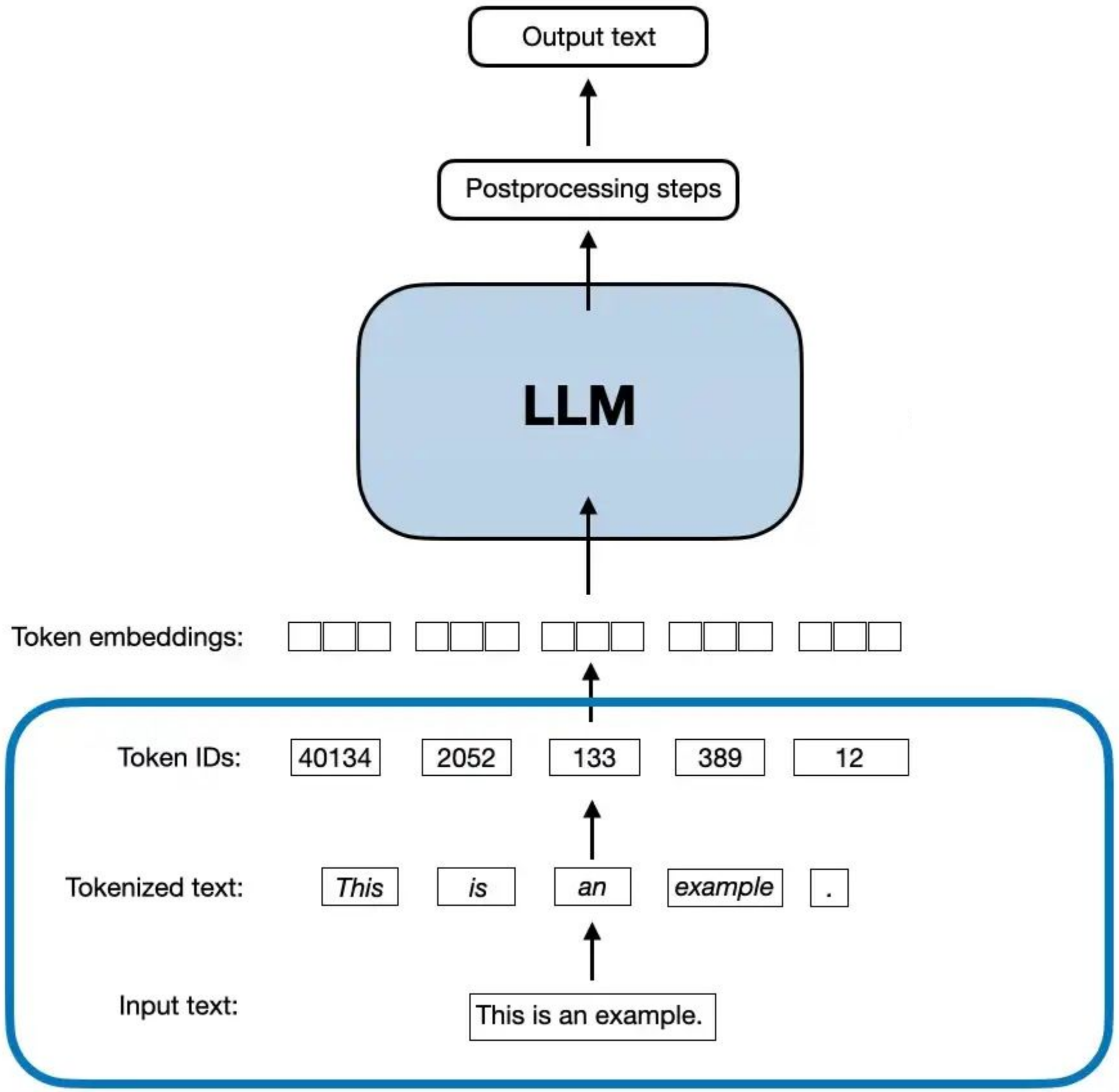


Different Levels of Tokenization

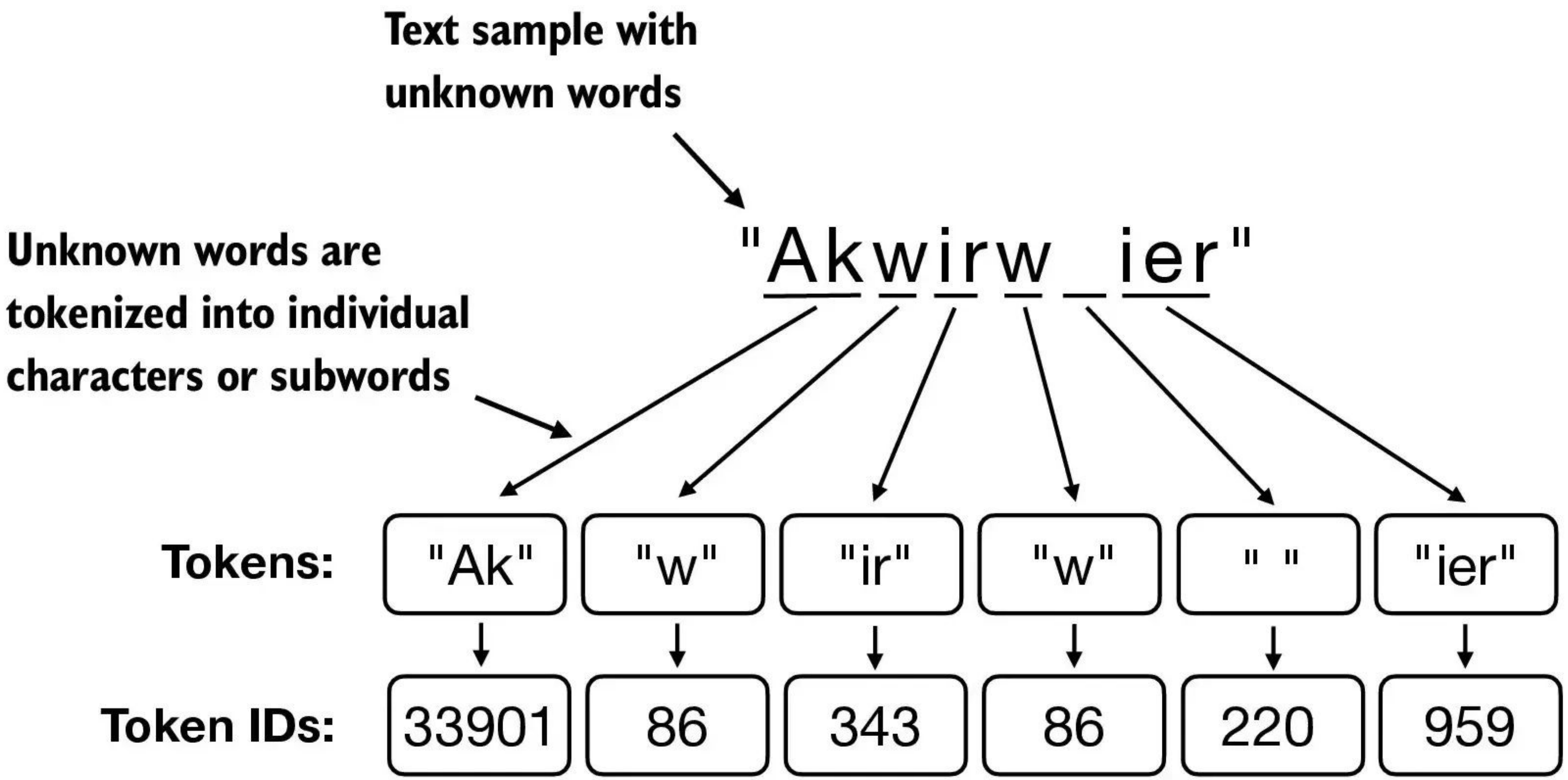
Byte-Pair Encoding (BPE)

- **How BPE works:**
  - Start with characters.
  - Merge most frequent pair (e.g., "t", "h" $\rightarrow$ "th").
  - Repeat until vocab size reached.
- **Advantages:**
  - Handles unknown words well (e.g., `unbelievably` $\rightarrow$ `un` + `believ` + `ably`)
  - Reduces sequence length
- **Limitations:**
  - Not optimal for all scripts (e.g., Chinese)
  - Word boundaries may be unclear

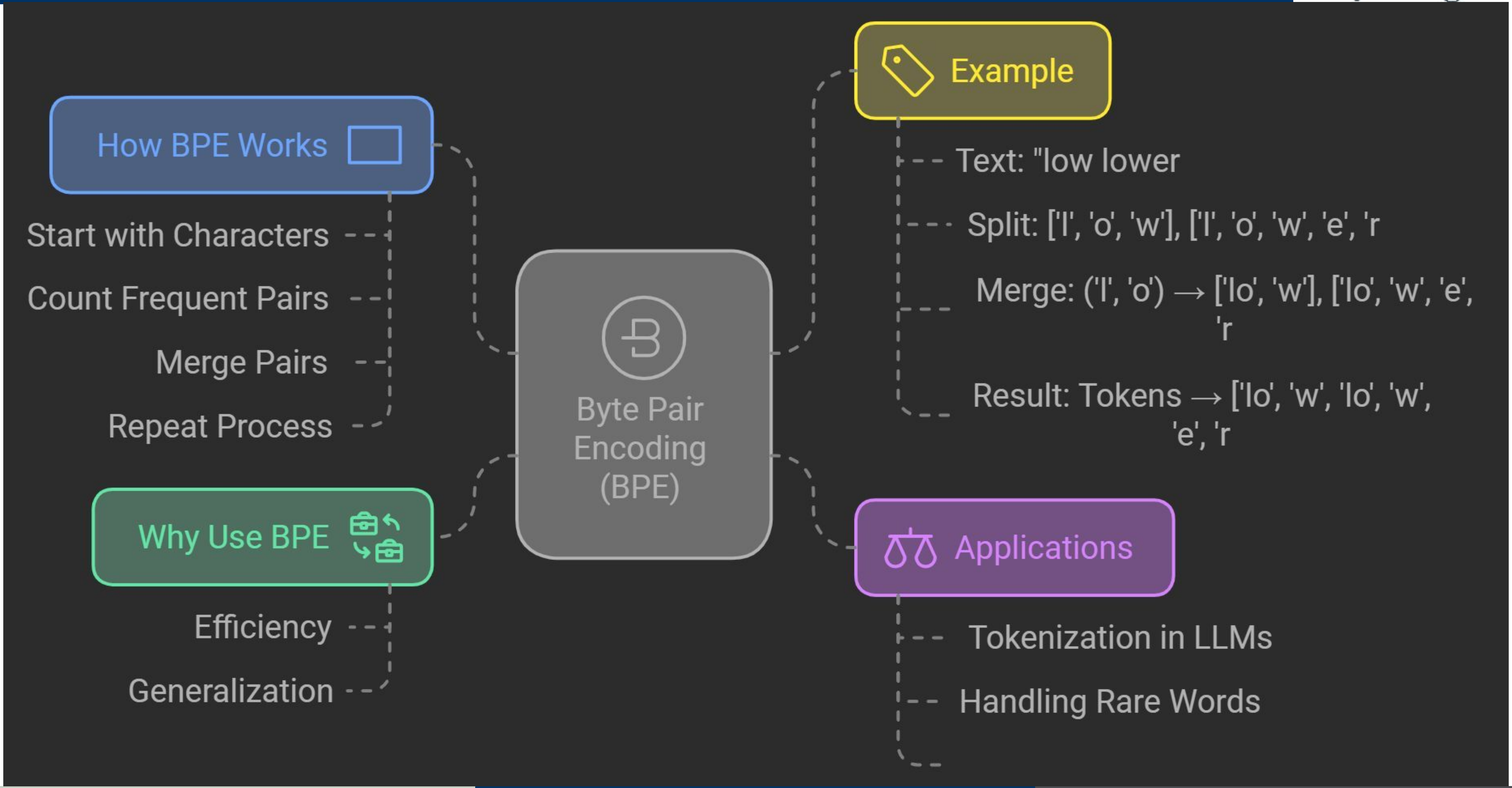
Byte-Pair Encoding (BPE)



Byte-Pair Encoding (BPE)



Byte-Pair Encoding (BPE)



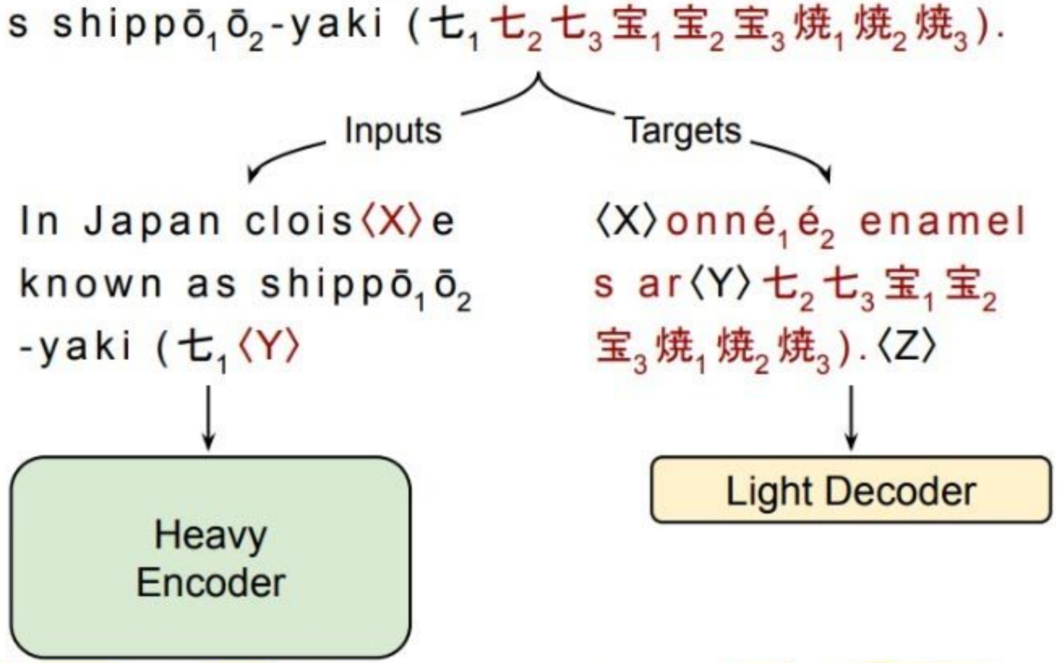
Token-Free LLMs

- **What are Token-Free LLMs?**
  - Models that operate directly on raw text without tokenization.
  - Aim to eliminate the need for pre-defined tokens.
- **Advantages:**
  - Avoids tokenization errors and biases.
  - Can handle arbitrary text lengths and formats.
  - Potentially more efficient for certain tasks.
- **Challenges:**
  - Requires novel architectures to process raw text effectively.
  - May struggle with long-range dependencies without tokenization.

Token-Free LLMs

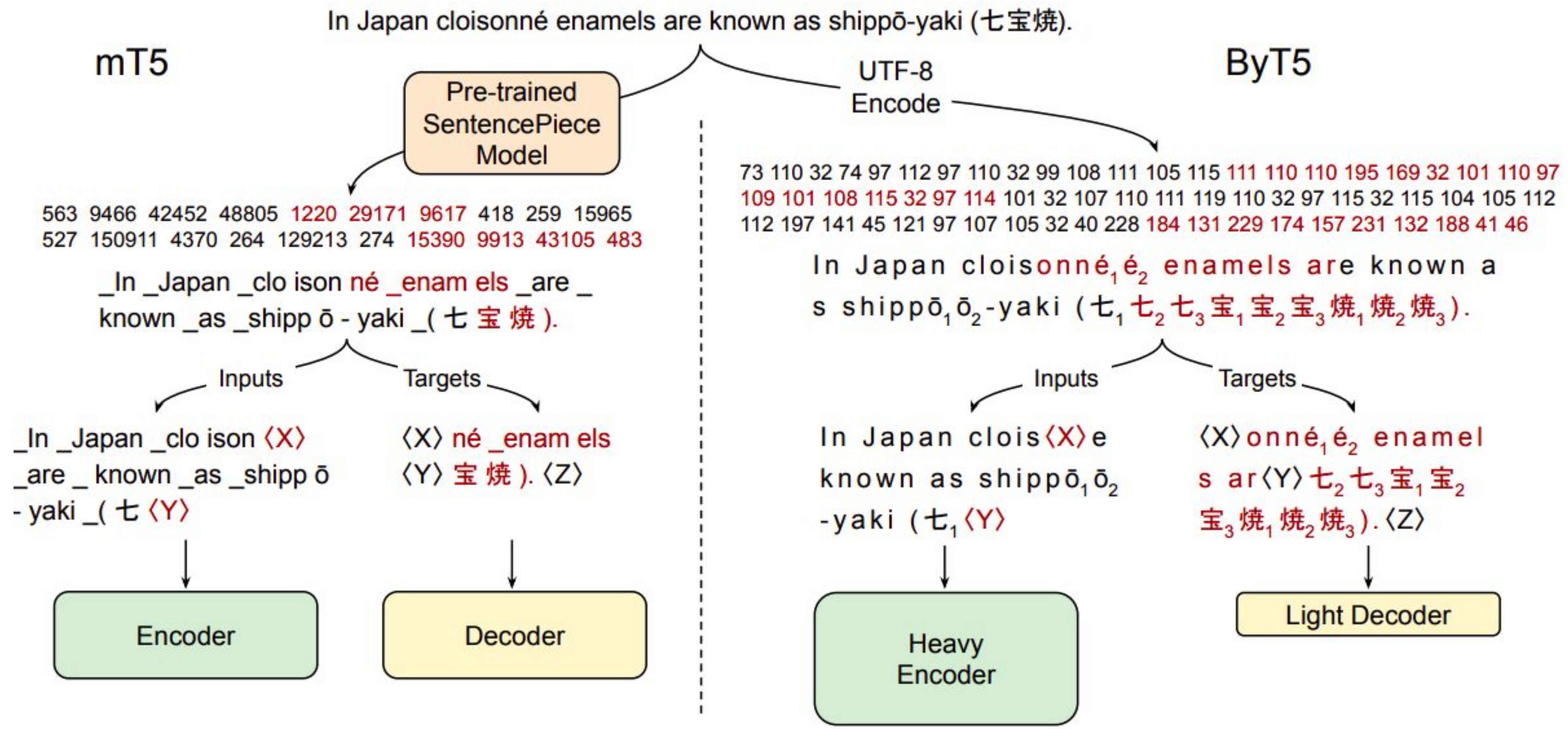
- **Example: ByT5 (Xue et al., 2022)**
  - Character-level variant of T5 that processes raw bytes instead of tokens.
  - Eliminates the need for a tokenizer, enabling better multilingual and low-resource language support.
- **Pros:**
  - No preprocessing or tokenization pipeline required.
  - Handles any language or script without modification.
  - Performs better on low-resource and unseen languages.
- **Cons:**
  - Training is slower due to longer input sequences.
  - Increased computational requirements.
  - May require more data to achieve comparable performance.

ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models



NLP Journal Club

Token-Free LLMs



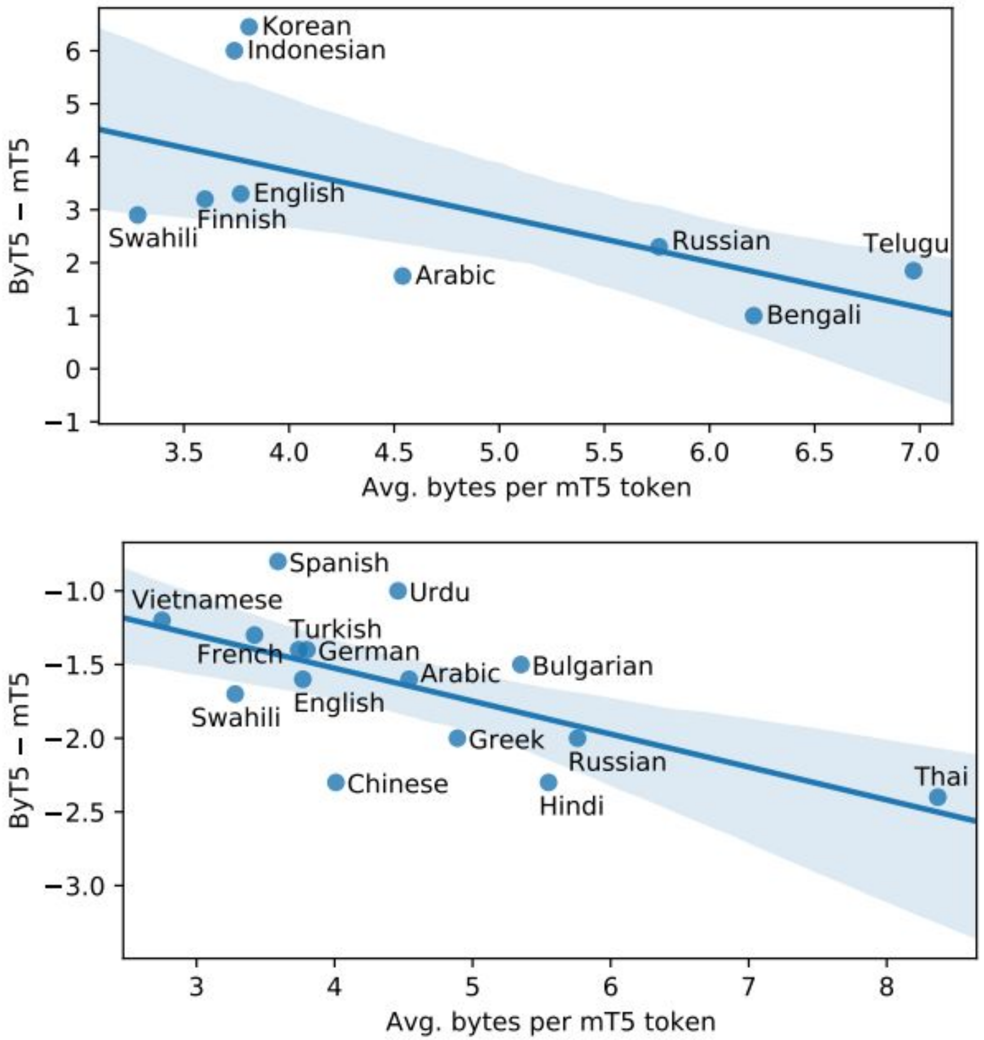


Figure 3: Per-language performance gaps between ByT5-Large and mT5-Large, as a function of each language's "compression rate". **Top**: TyDiQA-GoldP gap. **Bottom**: XNLI zero-shot gap.

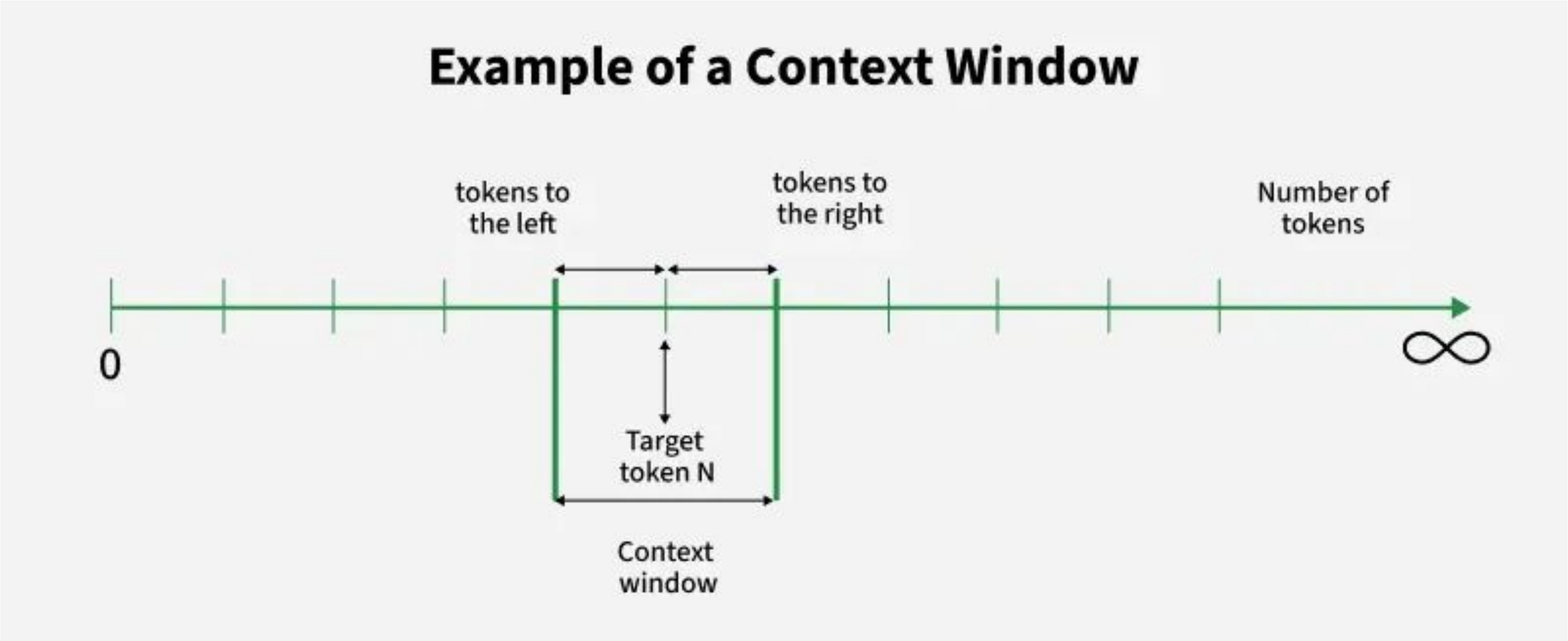
Token-Free LLMs

- **Current Research:**
  - Exploring architectures like *Raw Transformer* and *Text2Vec*.
  - Investigating how to maintain performance without tokenization.
- **Future Directions:**
  - Integrating token-free approaches with existing LLMs.
  - Enhancing efficiency and scalability of token-free models.

Context Window Problems

- Current LLMs have a finite "context window":
  - GPT-3: 2K tokens
  - GPT-4: Up to 128K tokens
  - Claude 3.5: Up to 200K tokens
- **Problems:**
  - Long documents get truncated
  - Token limit affects reasoning
  - Difficult to do document-level QA or summarization

Context Window Problems



Context Window Problems

أكاديمية كاوست
KAUST ACADEMY

LMH
Lady Margaret Hall

Why Bigger Context Windows Aren't Always Better?

1 Information Overload

- Excess data can overwhelm the LLM, reducing focus and slowing performance.

2 Getting Lost in Data

- Large windows prioritize edges, missing key middle info.

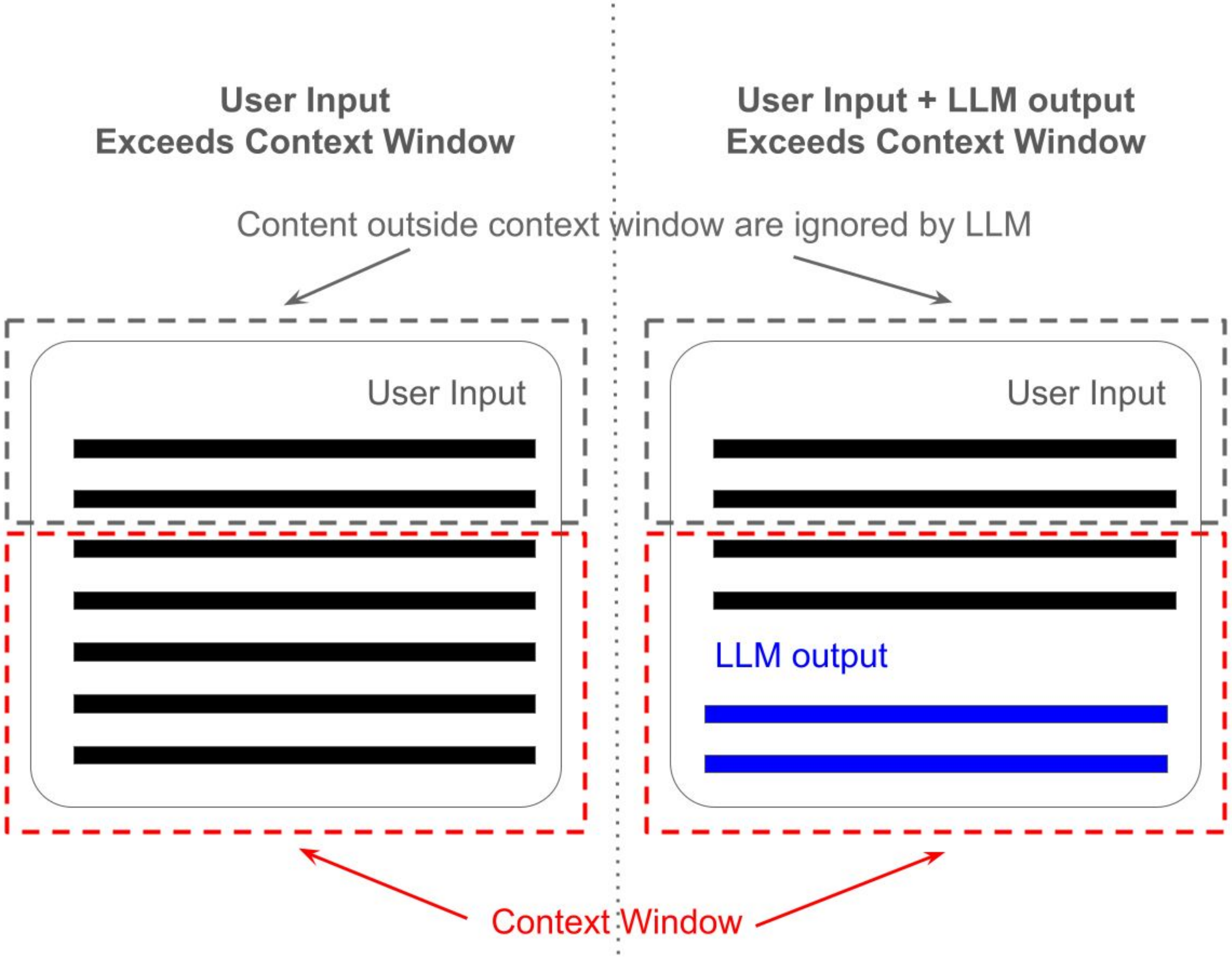
3 Poor Management

- More data leads to noise, redundancy, and potential bias.

4 Long-Range Issues

- Understanding relationships becomes harder with large context windows.

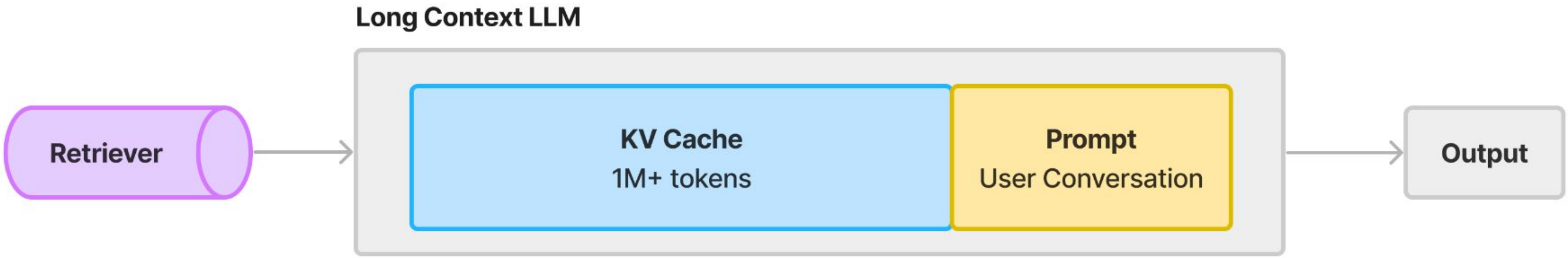
Context Window Problems



Moving Towards Larger Context Windows

- **Efficient Attention Variants:**
  - Longformer, BigBird, FlashAttention-2
- **Memory-Augmented Models:**
  - Retrieval-augmented generation (RAG)
  - Memory layers (e.g., RetNet)
- **Chunking + Re-ranking:**
  - Process in parts, summarize/join results
- **Recurrence or State Passing:**
  - Transformer-XL, RWKV (RNN-inspired)

Moving Towards Larger Context Windows



Limitations of LLMs

أكاديمية كاوست
KAUST ACADEMY

LMH
Lady Margaret Hall

- High inference costs & latency
- Bias and toxic outputs
- Hallucination and factual inconsistency
- Limited interpretability
- Require vast pre-training data
- Poor domain generalization (medical, legal, etc.)

Future Directions for LLMs

- **Long Context Handling:** Efficient token reuse, sparse attention, recurrence.
- **Token-Free Models:** Improved byte/character models.
- **Multimodal LLMs:** Integrating vision, speech, and more.
- **Smaller, Efficient LLMs:** Distillation, quantization, sparse models.
- **Open-Weight & Ethical LLMs:** Responsible, accessible models.

Summary

- BERT and GPT are foundational LLM architectures.
- Scaling laws dictate optimal growth paths.
- Tokenization is central to model performance.
- Context window size remains a bottleneck.
- Innovations in token-free modeling and memory-efficient transformers are shaping the future.

References

These slides have been adapted from

- Bhiksha Raj & Rita Singh, 11-785 Introduction to Deep Learning, CMU

References

[1] Kaplan et al., "Scaling Laws for Neural Language Models", 2020.

[2] Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers", 2018.

[3] Brown et al., "Language Models are Few-Shot Learners (GPT-3)", 2020.

[4] Xue et al., "ByT5: Towards a token-free future with pre-trained byte-to-byte models", 2022.

[5] Beltagy et al., "Longformer: The Long-Document Transformer", 2020.

[6] Tay et al., "Efficient Transformers: A Survey", 2020.

[7] OpenAI, "Technical Report on GPT-4", 2023.

References

[8] Google Research Blog: Scaling Transformer Models.

[9] Anthropic, "Claude 3.5 Release Notes", 2024.

[10] FlashAttention: https://arxiv.org/abs/2205.14135

Credits

Dr. Prashant Aparajeya

Computer Vision Scientist — Director(AISimply Ltd)

p.aparajeya@aisimply.uk

This project benefited from external collaboration, and we acknowledge their contribution with gratitude.