Washington University in St. Louis
James McKelvey School of Engineering

CSE 561A: Large Language Models

Spring 2024

Lecture 2: Language Model Architectures

Course Announcements

- The course capacity has been expanded.
- The course is published on canvas: https://wustl.instructure.com/courses/129974
- The sign-up sheet is out: https://docs.google.com/spreadsheets/d/1xSCaIOjiri17V7IjP2di-kFwbOgInPb_azBfZKeTgmc/edit
- The dates of the presentation are tentative, might be one week or two floating around the original date on the syllabus.

Content

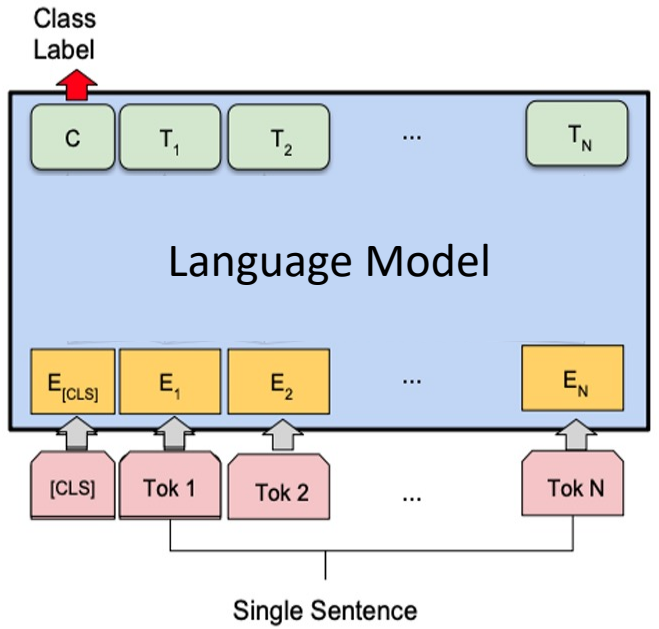
- **Transformers**
- Different Architectures of Pre-trained Language Models
  - Decoder-Only Models (GPT-2)
  - Encoder-Only Models (BERT, RoBERTa, ELECTRA)
  - Encoder-Decoder Models (T5, BART)

Common NLP Tasks

- Sentence-Level Tasks:
  - Single-sentence classification tasks: text classification, sentiment analysis, …
  - Sentence-pair classification tasks: sentence entailment, …
  - Sentence generation tasks: machine translation, question answering, …
- Token-Level Tasks:
  - part-of-speech tagging, named entity recognition, …

Common NLP Tasks

- Single-sentence classification tasks
  - Text Classification Tasks
    - Input: The bike is too small and I want to return it.
    - Output: <refund, return, check_status>
  - Sentiment Analysis
    - Input: The restaurant is crowded and I waited my food for 30 minutes!
    - Output: <positive, negative>



Common NLP Tasks

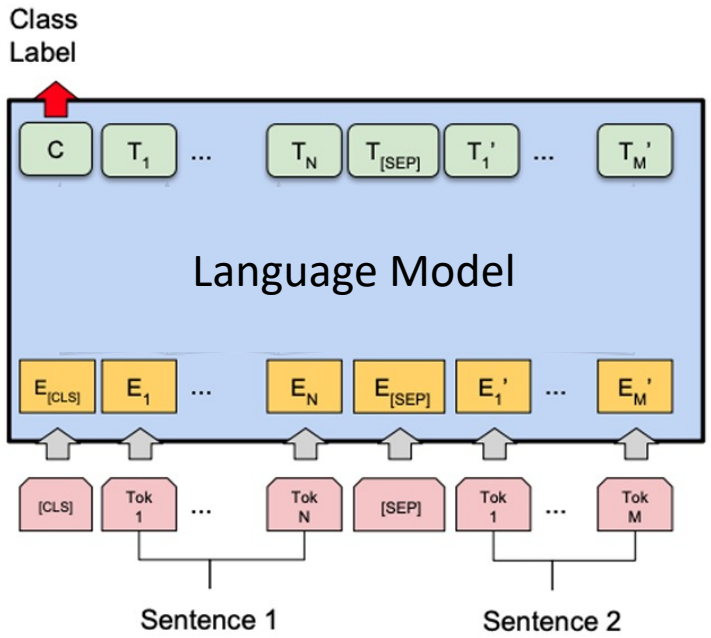
- Sentence-pair classification tasks
  - Sentence entailment
    - Input:

      Sentence 1: Our Large Language Model Course meets on Tuesdays and Thursdays in WashU.

      Sentence 2: There is a large language model course in WashU.

    - Output:

      <entailment, contradiction, neutral>

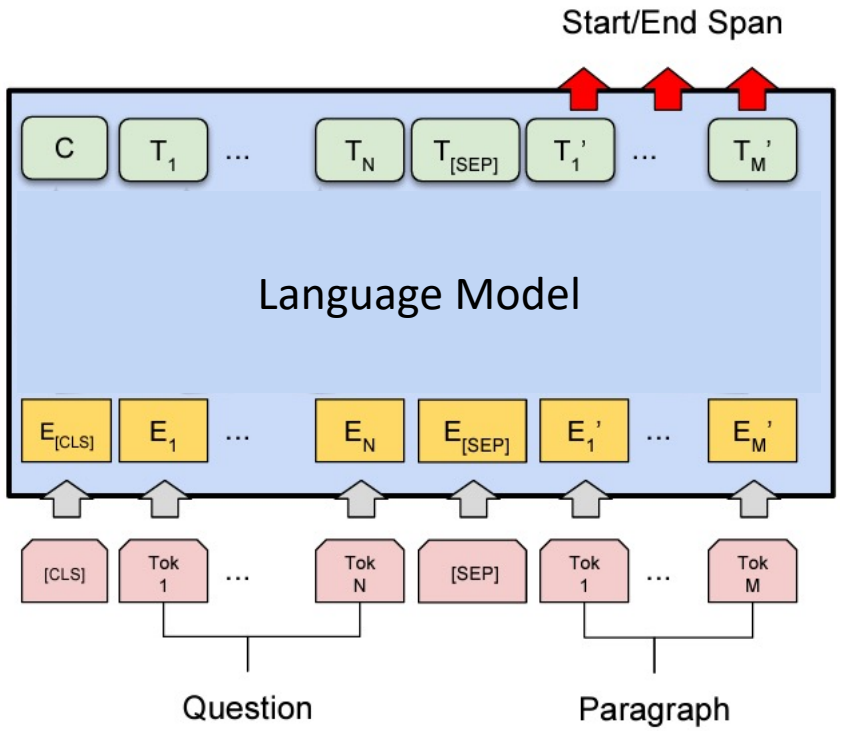


Common NLP Tasks

- Sentence generation tasks
  - Machine Translation
    - Input:

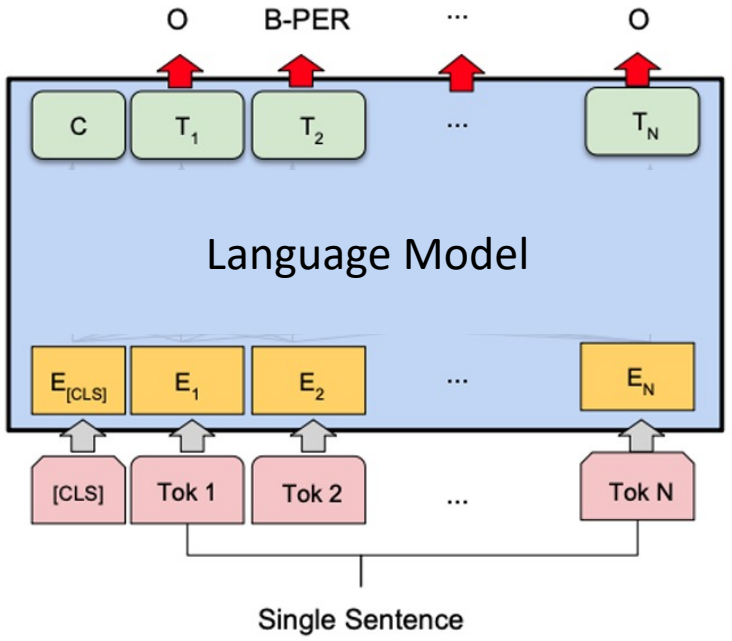
    English: This is good. Germany:
    - Output:

    Das ist gut.



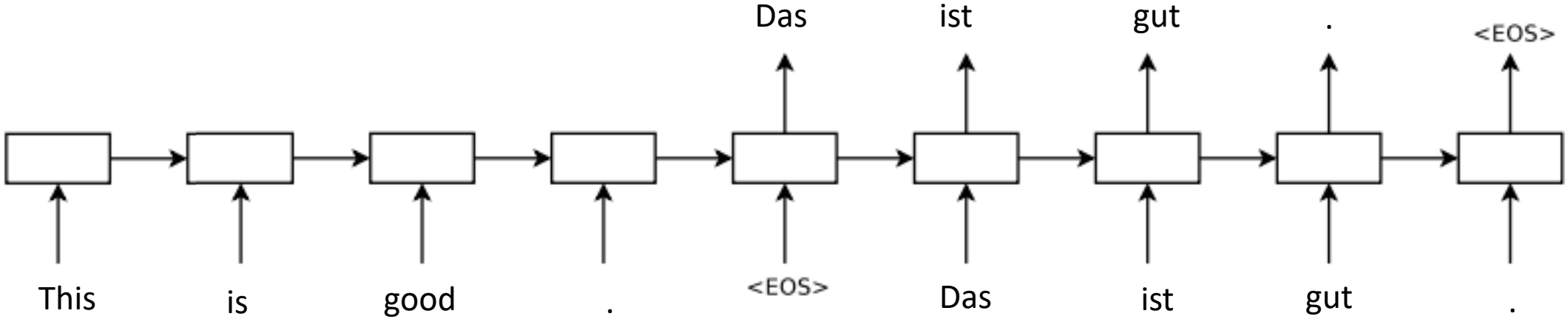
Common NLP Tasks

- Token-level tasks
  - Named Entity Recognition
  - Input: St. Louis is located in the state of Missouri .
  - Output: <Begin-Location> <Inside-location> O O O O O O <Begin-Location> O



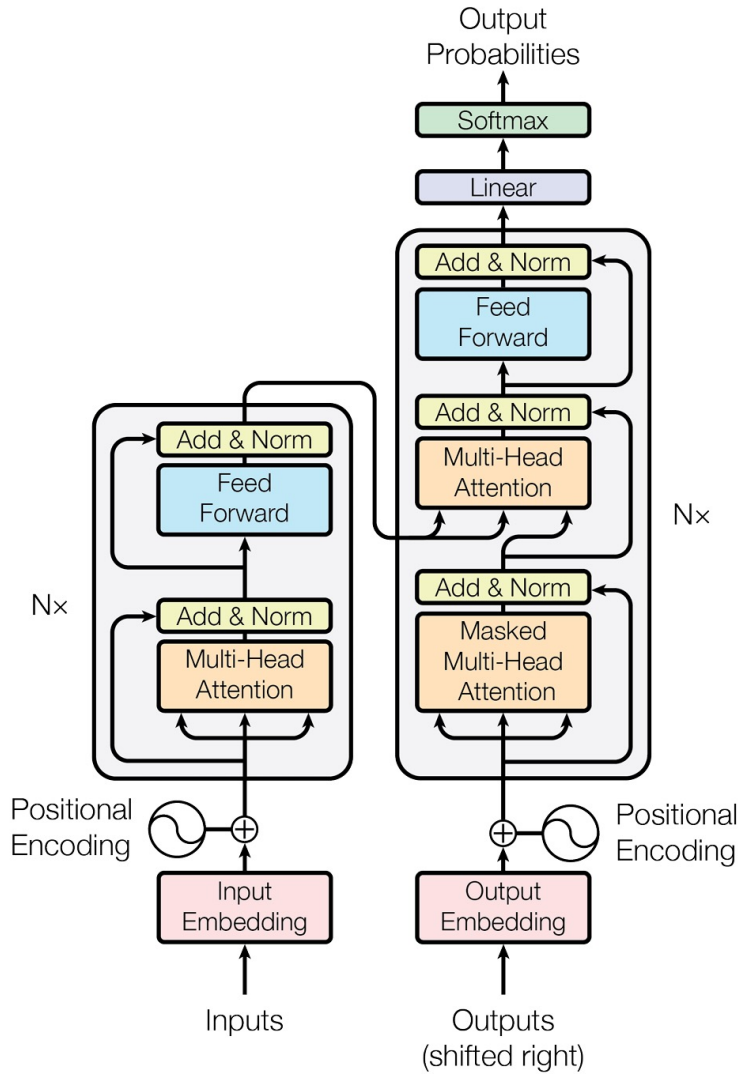
Encoder and Decoder

- NLP tasks can be generally decomposed into language understanding and language generation.
- Encoder models are generally used to understand input sentences, and decoder models are generally used to generate sentences.



Transformer Model Architecture

- Input Embedding
- Positional Encoding
- 12 Transformer layers
  - 6 encoder layers
  - 6 decoder layers
- Linear + Softmax layer for next word prediction



Encoder Model

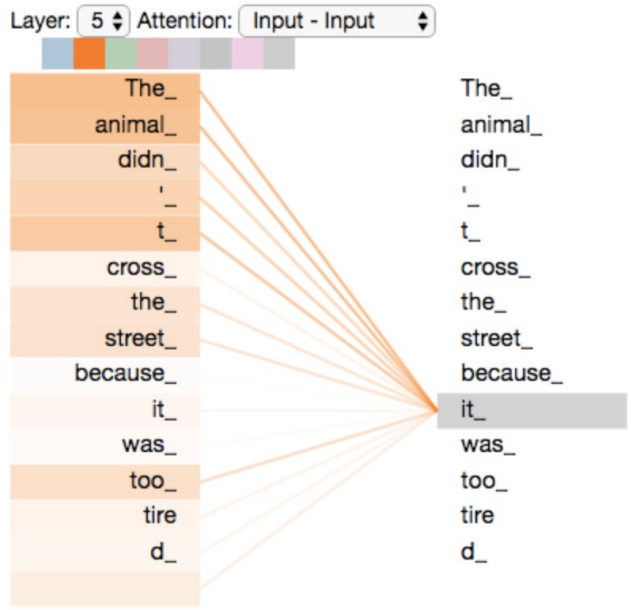
- Multi-head attention layer captures information from different subspaces at different positions

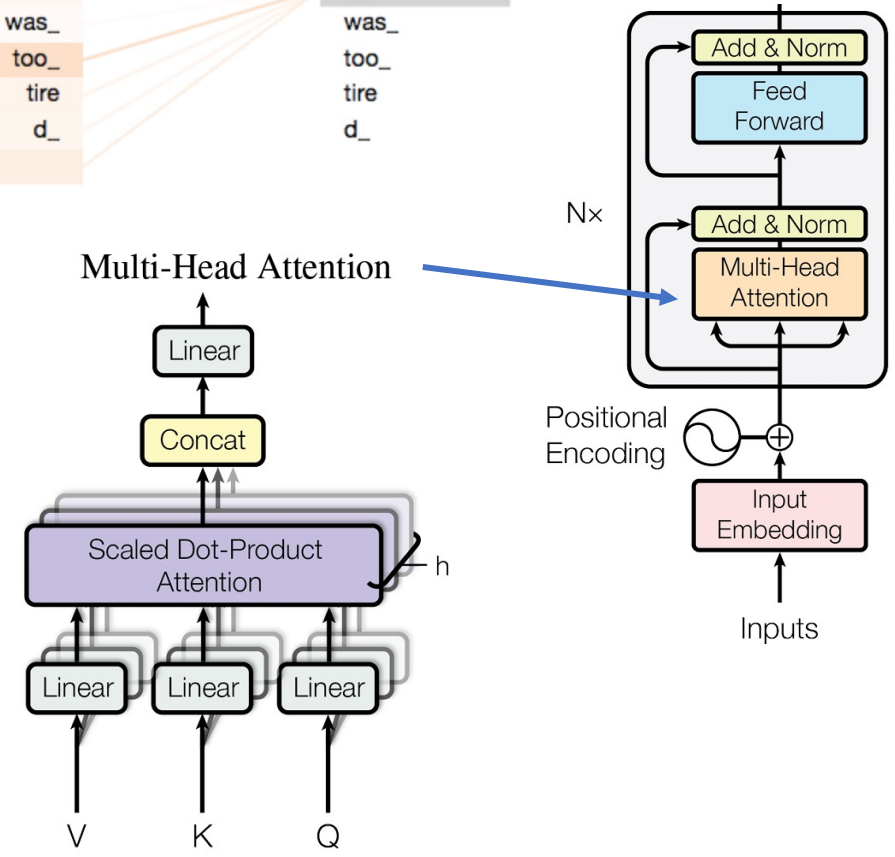
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

- Feed-forward layer is applied to each token position without interaction with other positions

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

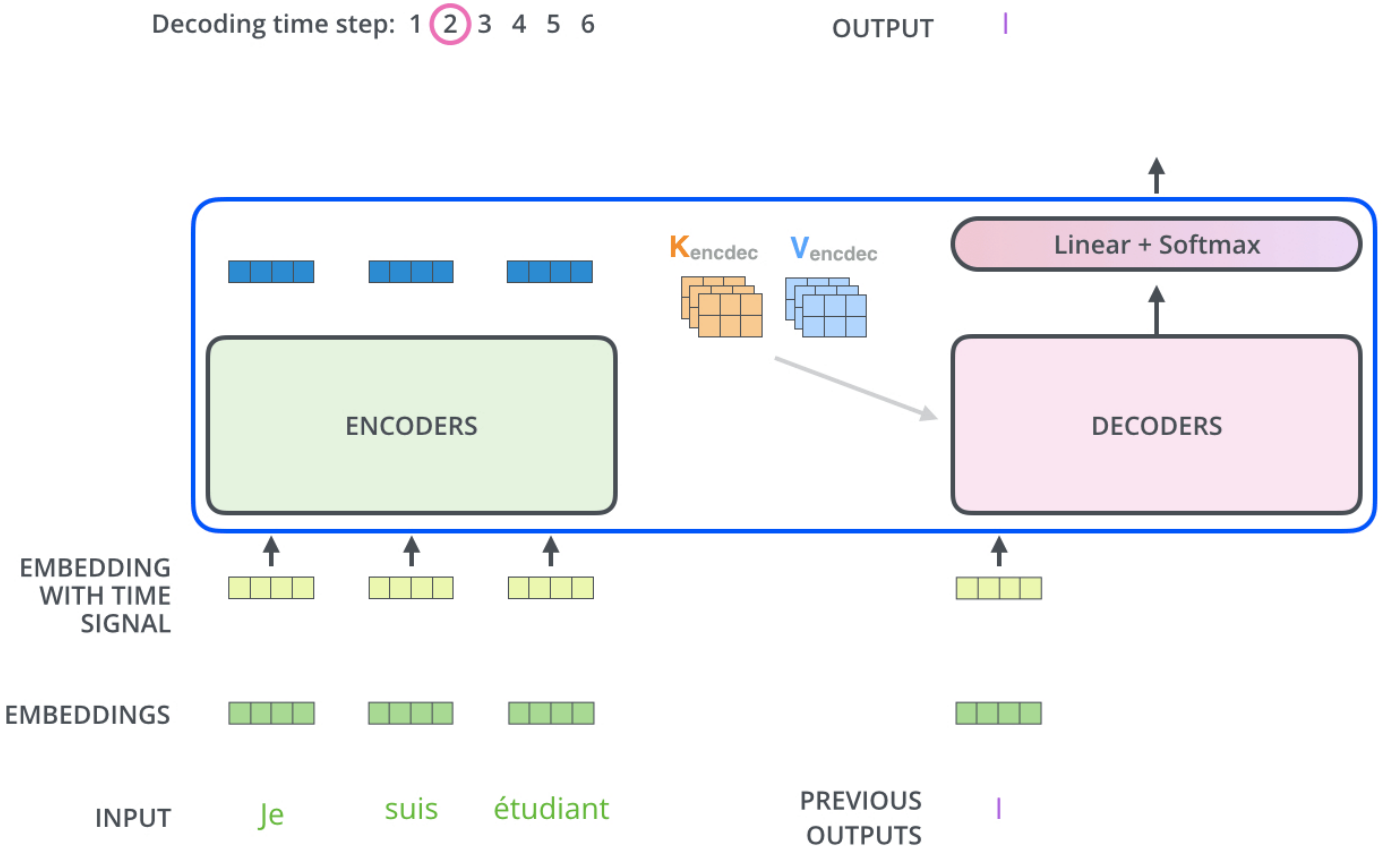
- Residual connection and layer normalization





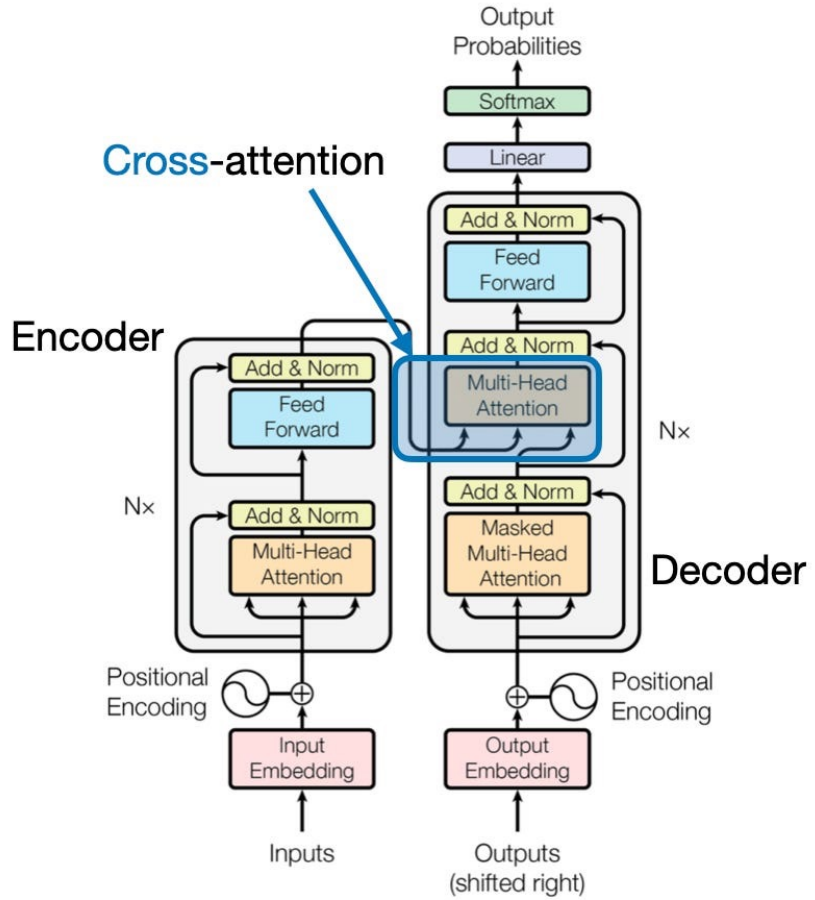
Decoder Model

- Demo from https://jalammar.github.io/illustrated-transformer/



Decoder Model

- Multi-head self-attention: only allowed to attend to earlier positions (left side).
  - Q, K, V matrices are both from the previously generated tokens
- Multi-head cross-attention: attend to the input sequence.
  - Q is from the generated tokens
  - K, V matrices are from the input tokens



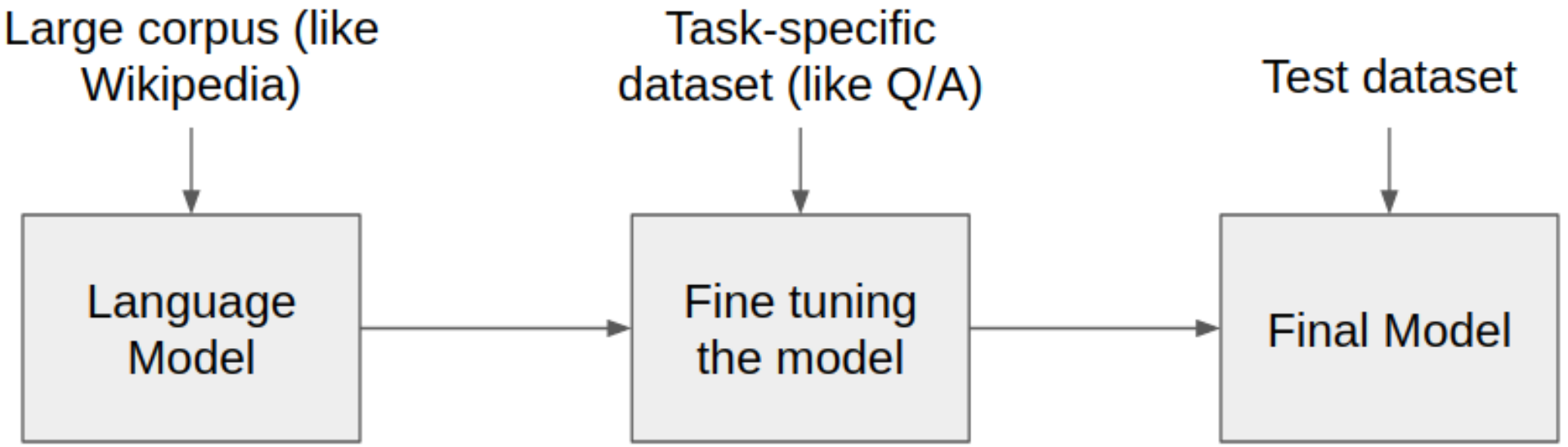
Source: “Attention Is All You Need” (https://arxiv.org/abs/1706.03762)

Content

- Transformers
- **Different Architectures of Pre-trained Language Models**
  - Decoder-Only Models (GPT-2)
  - Encoder-Only Models (BERT, RoBERTa, ELECTRA)
  - Encoder-Decoder Models (T5, BART)

Pre-trained Language Models

- “Pretraining”: Train deep language models (usually Transformer models) via **self-supervised** objectives on **large-scale general-domain corpora**
- “Fine-tuning”: Adapt the pretrained language models (PLMs) to downstream tasks using task-specific data
- The power of PLMs: Encode generic linguistic features and knowledge learned through large-scale pretraining, which can be effectively transferred to the target applications



Different Architectures for PLMs

- **Decoder-Only (Unidirectional) PLM** (e.g., GPT): Predict the next token based on previous tokens, usually used for **language generation tasks**
- **Encoder-Only (Bidirectional) PLM** (e.g., BERT, XLNet, ELECTRA): Predict masked/corrupted tokens based on all other (uncorrupted) tokens, usually used for **language understanding/classification tasks**
- **Encoder-Decoder (Sequence-to-Sequence) PLM** (e.g., T5, BART): Generate output sequences given masked/corrupted input sequences, can be used for both **language understanding and generation tasks**

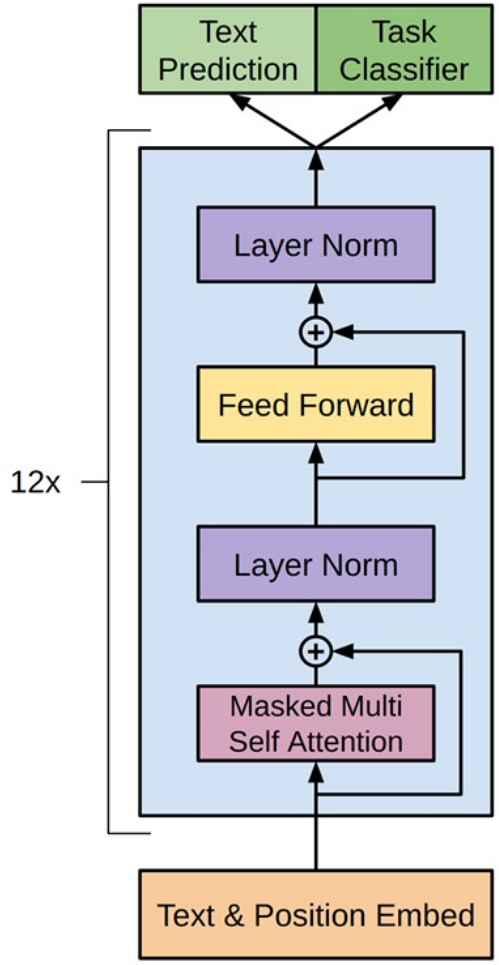
Generative Pretraining (GPT)

- Model Architecture: A multi-layer transformer decoder
- Leverage unidirectional context (usually left-to-right) for next token prediction (i.e., language modeling)

k previous tokens as context

$$\mathcal{L}_{\text{LM}} = -\sum_i \log p(x_i \mid x_{i-k}, \ldots, x_{i-1})$$

- The Transformer uses **unidirectional** attention masks (i.e., every token can only attend to previous tokens)



[1] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI blog.
[2] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.
[3] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. NeurIPS.

Content

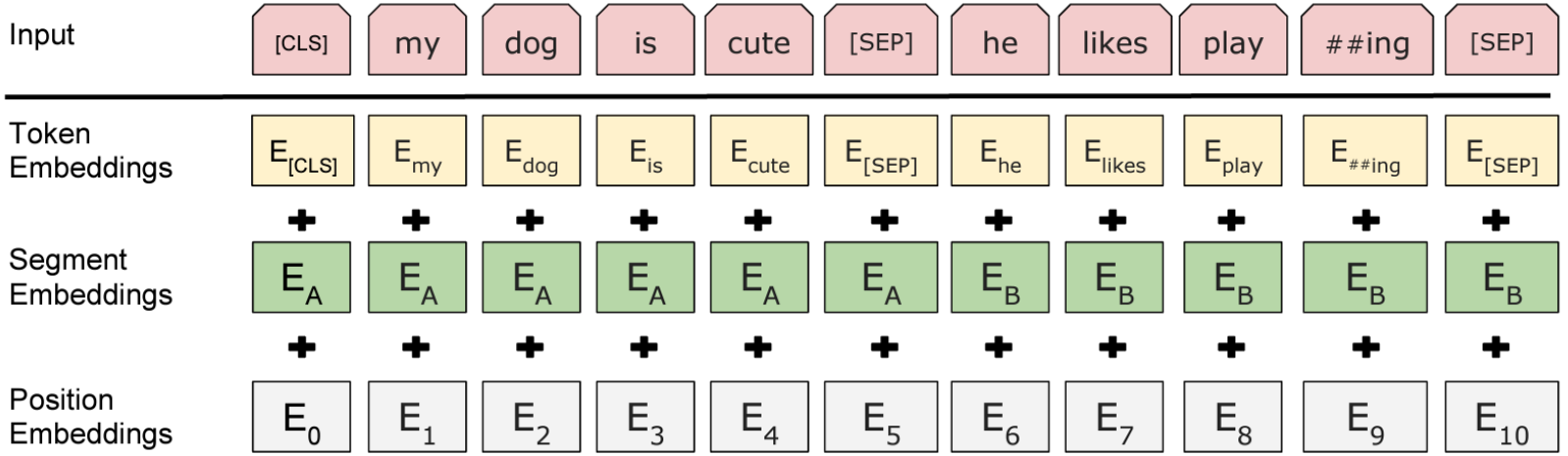
- Transformers (continued)
- Different Architectures of Pre-trained Language Models
  - Decoder-Only Models (GPT-2)
  - **Encoder-Only Models (BERT, RoBERTa, ELECTRA)**
  - Encoder-Decoder Models (BART, T5)

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [NAACL'19 best paper]

- Main Contributions:
- Pre-training corpus: Wikipedia (2.5B) + BooksCorpus (0.8B)
- Pre-training objectives
  - Masked language modeling + Next Sentence Prediction
- Groundbreaking performance on a wide range of token-level and sentence-level tasks

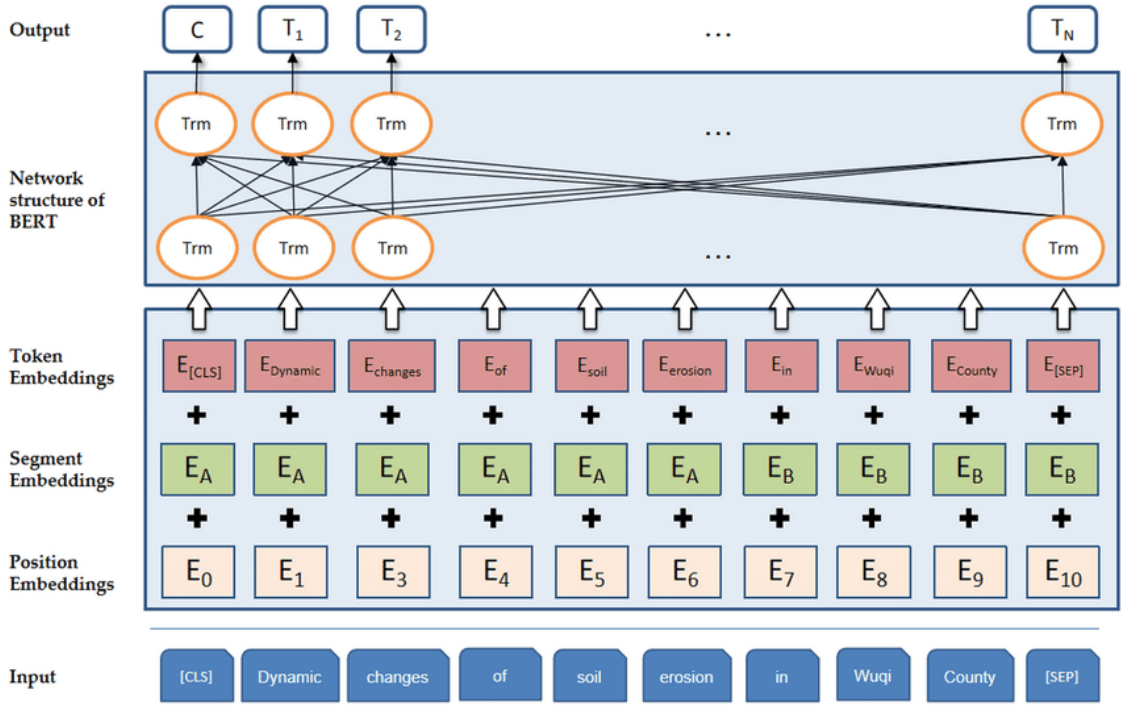
BERT Model Architecture

- Input: Sentence Pairs with special tokens [CLS] and [SEP].
  - Pair-wise tasks: question answering, translation, sentence entailment
  - [CLS]: beginning of a sentence
  - [SEP]: separation of two sentences
- WordPiece embedding



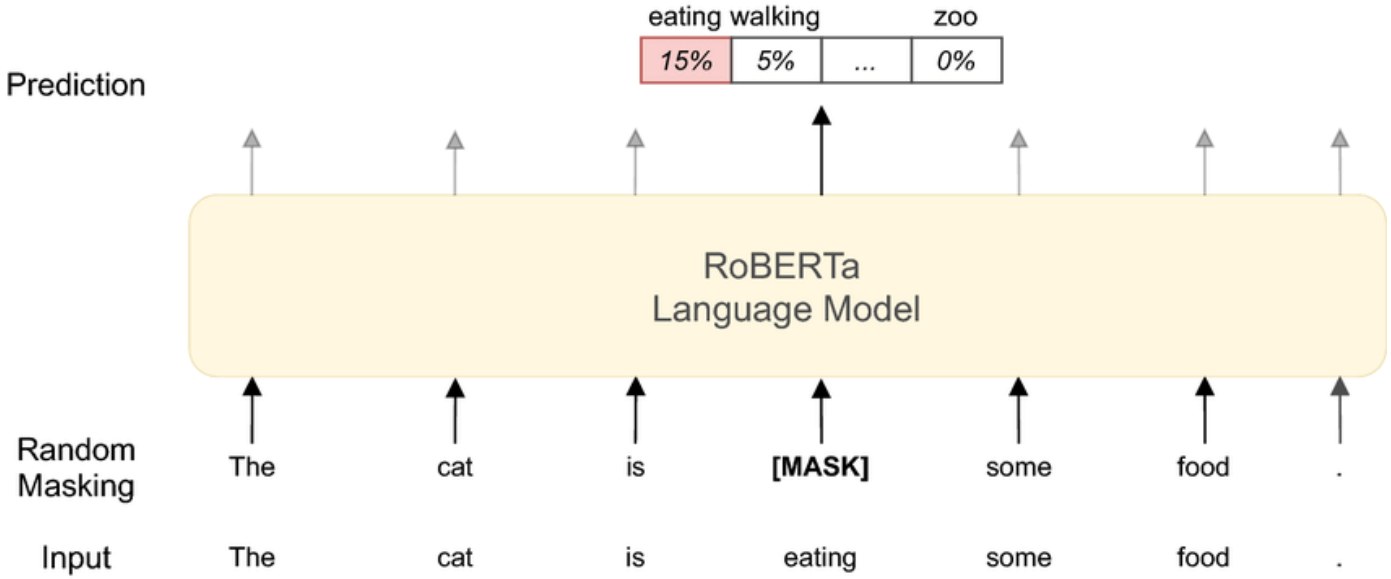
BERT Model Architecture

- Multi-layer Transformer encoder model
  - Bert-base: 12 layers, 768 hidden size, 12 attention heads, 110M parameters
  - Bert-large: 24 layers, 1024 hidden size, 16 attention heads, 340M parameters
- Bidirectional: each token can attend to its left and right context for self-attention



BERT Training Objective (I): Masked Language Modeling

- Masked Language Modeling (MLM)
  - Randomly mask a few words in the original sentences.
  - Predict the masked words with its left and right context.
  - Mask ratio: 15%
  - Demo: https://huggingface.co/bert-base-cased

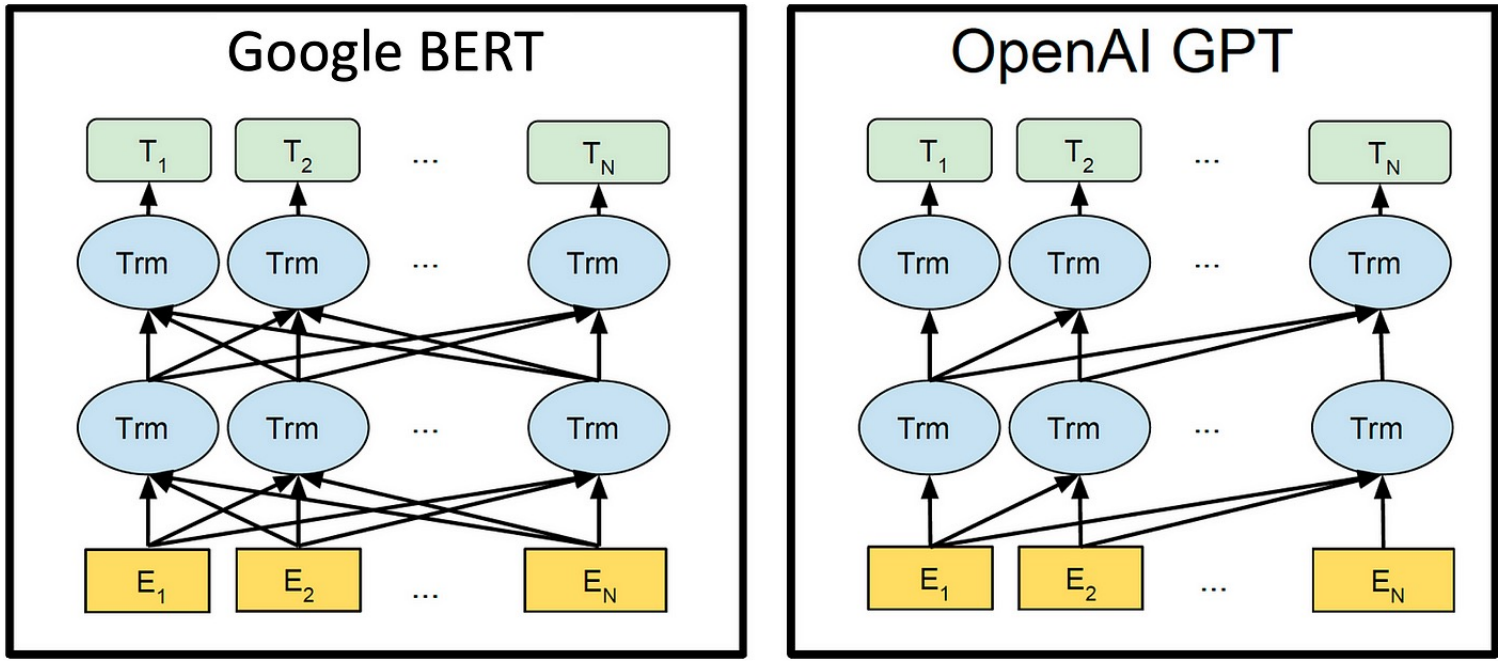


BERT Training Objective (I): Masked Language Modeling

- Discrepancy between pre-training and fine-tuning: There is no [MASK] word in a test set!
- 80-10-10 replacing rule for the sampled 15% tokens.
  - 80%: Replace the original word with [MASK] token
    - cat eating food -> cat [MASK] food
  - 10%: Replace the original word with a random token
    - cat eating food -> cat paper food
  - 10%: Do not replace the original word
    - cat eating food -> cat eating food

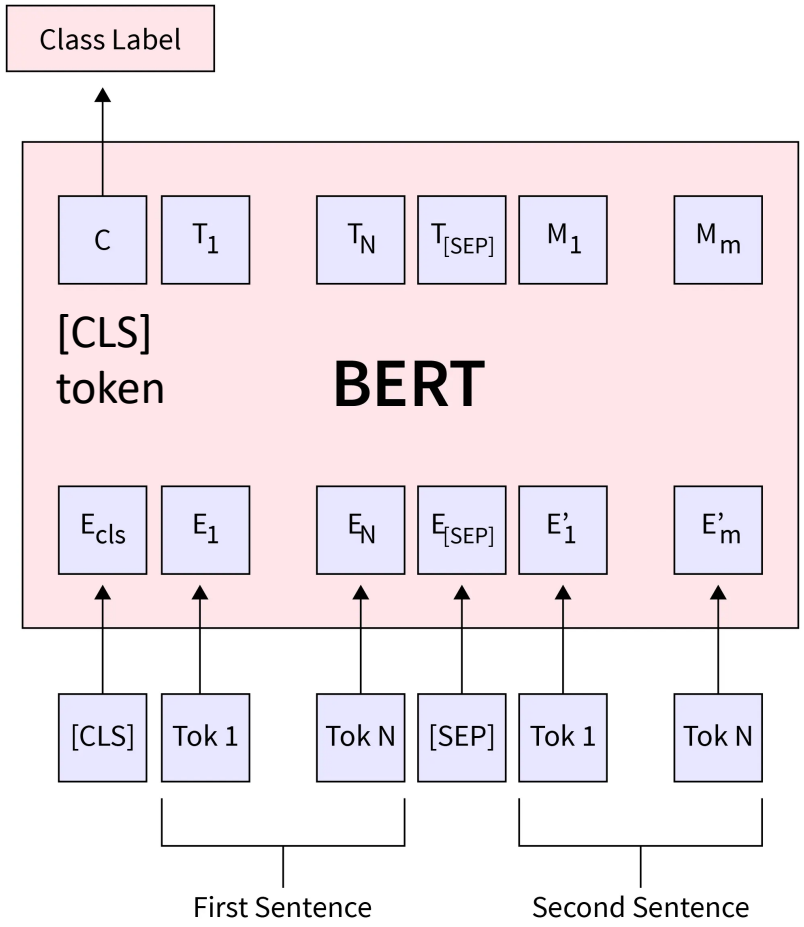
Comparison with GPT-2 Model

- Training objective: MLM prediction vs. left-to-right token prediction



BERT Training Objective (II): Next Sentence Prediction

- Next Sentence Prediction (NSP)
  - Predict whether Sentence B is the next sentence of Sentence A.
  - Positive samples: two contiguous sentences in the corpus.
  - Negative samples: sample another sentence for sentence A.
  - Class Labels: <is_next, not_next>



BERT Experiment Results

- GLUE Benchmark for natural language understanding

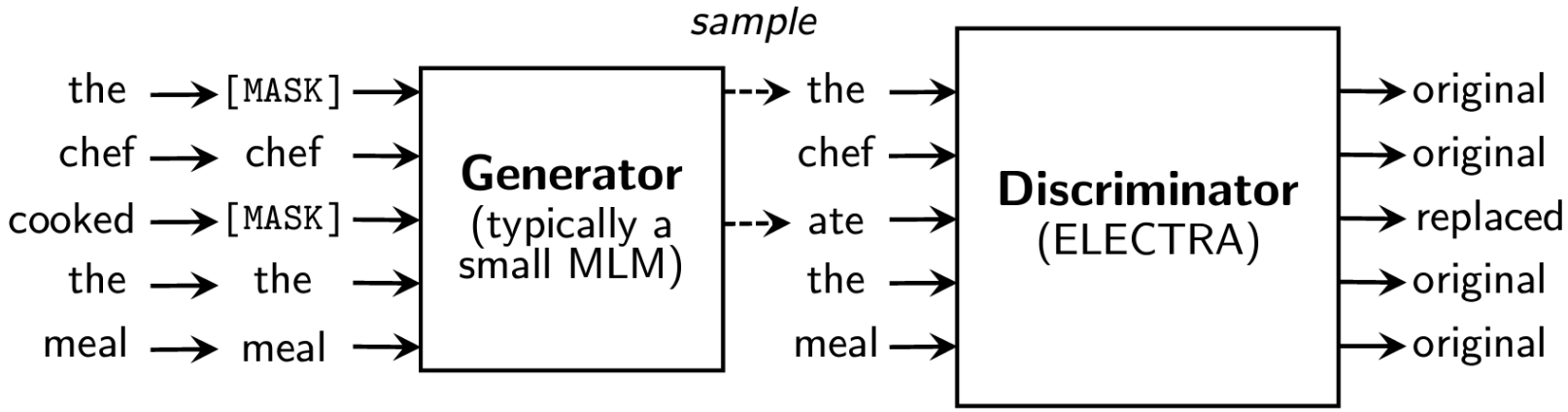
System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	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**

Variants of BERT Model

- RoBERTa (RoBERTa: A Robustly Optimized BERT Pretraining Approach. Liu et al. 2019)
  - Training the model longer on more data with bigger batches
  - Remove the next sentence prediction objective
  - Dynamically change the [MASK] patterns in each epoch

Variants of BERT Model

- ELECTRA (ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. Clark et al. 2020)
  - Replaced token detection by corrupting text sequences with an auxiliary MLM
  - Works better than BERT because the input text for ELECTRA does not contain [MASK] tokens (no discrepancy between training and test data)

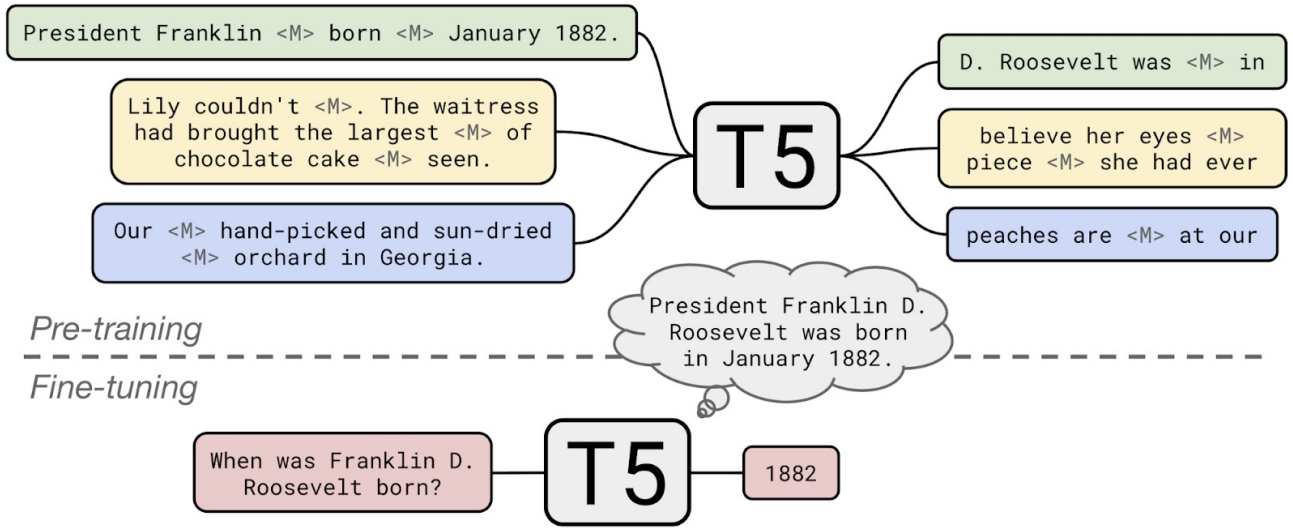


Content

- Transformers
- Different Architectures of Pre-trained Language Models
  - Decoder-Only Models (GPT-2)
  - Encoder-Only Models (BERT, RoBERTa, ELECTRA)
  - **Encoder-Decoder Models (T5, BART)**

T5 Model

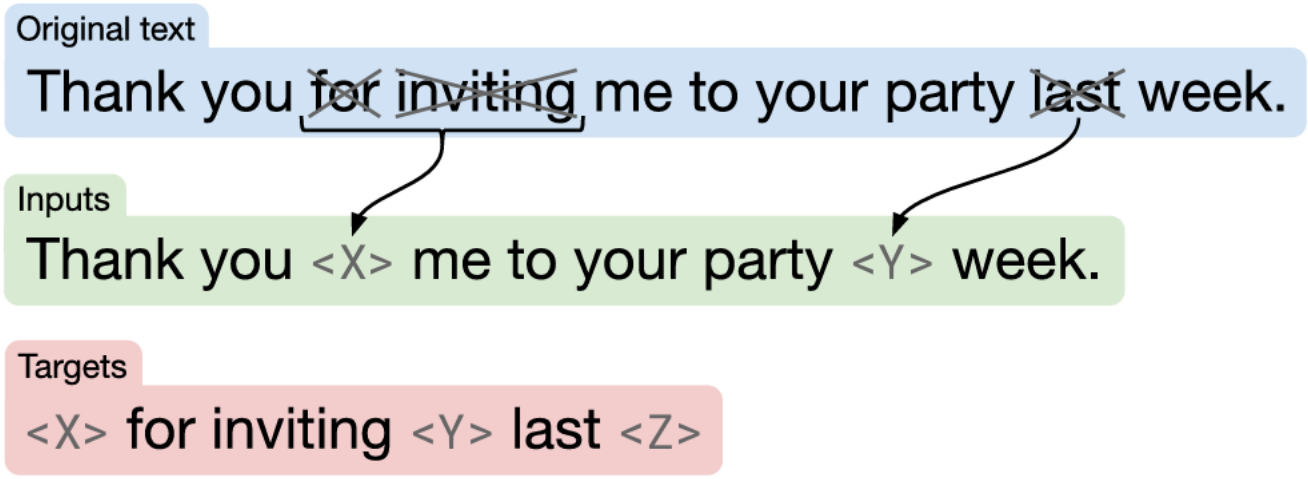
- How to predict a span of masked tokens within a sentence?
- BERT model requires the number of [MASK] token to be given in prior, while GPT models are causal left-to-right models
- T5: **T**ext-**t**o-**T**ext **T**ransfer **T**ransformer (parameters: 60M~11B)



Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR.

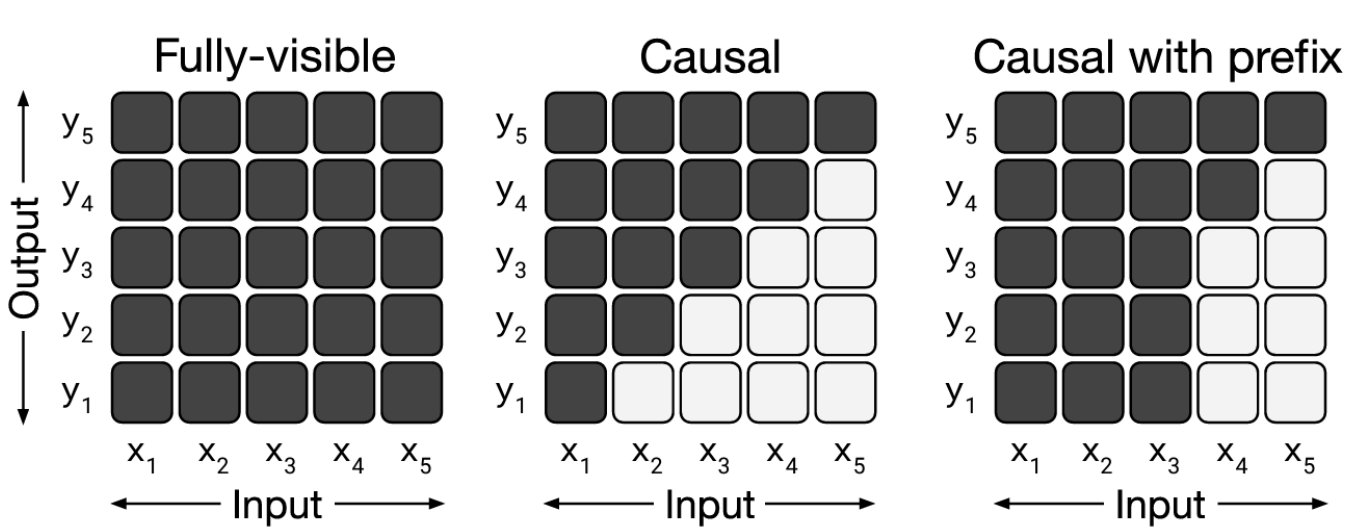
Training of T5 Model

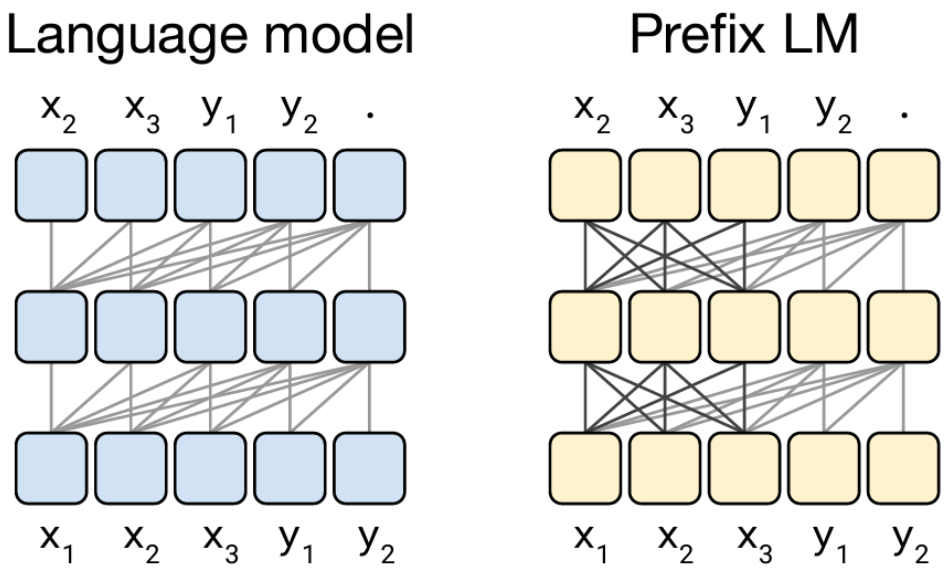
- Pretraining: Mask out spans of texts; generate the original spans
- Fine-Tuning: Convert every task into a sequence-to-sequence generation problem



T5 Attention

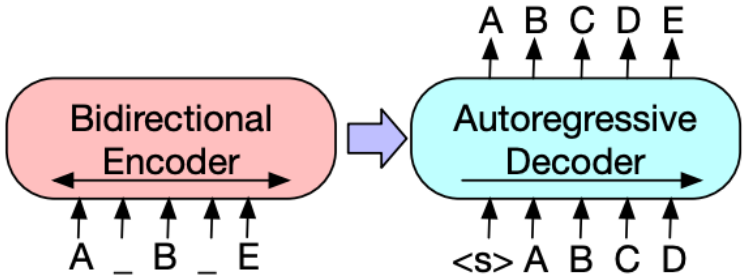
- A “fully-visible” attention mechanism is placed at the input sequence.
  - Input Sequence:
    - translate English to German : That is good . target :
  - Target Output:
    - Das ist gut .



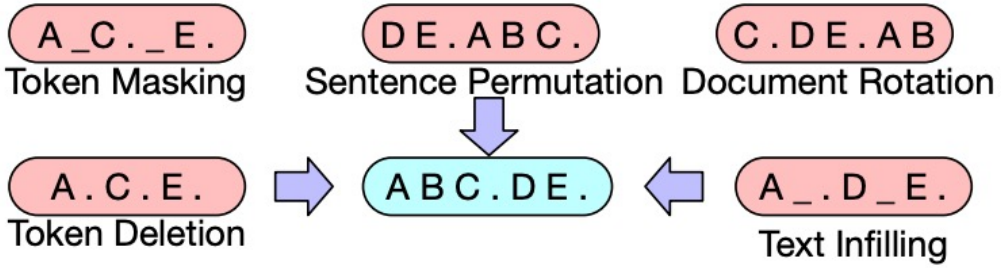


BART Model

- BART: Denoising autoencoder for pretraining sequence-to-sequence models
- Pretraining: Apply a series of noising schemes (e.g., masks, deletions, permutations…) to input sequences and train the model to recover the original sequences



BART architecture



BART pretraining objectives

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. ACL.

Scaling up Language Models

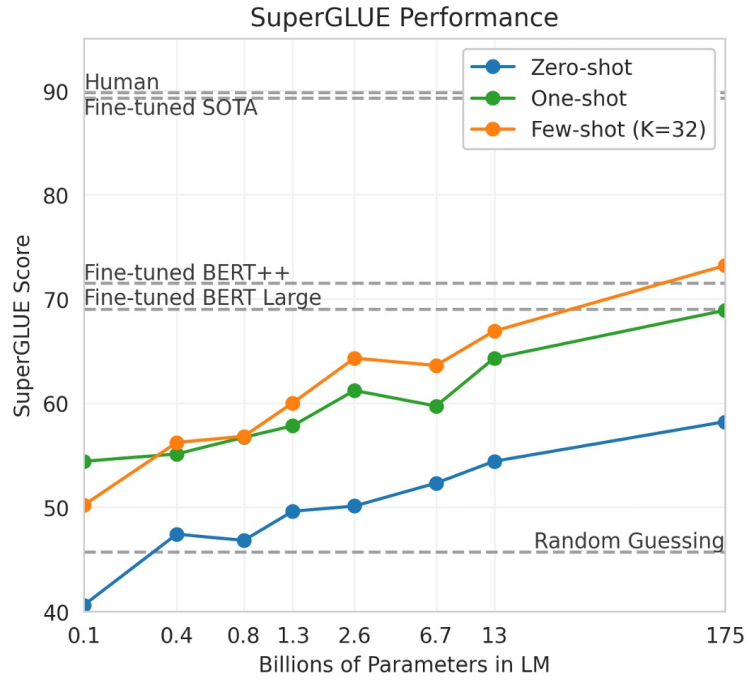
- GPT-2 model size: 1.5 billion parameters
- Pre-trained model can be very, very large (GPT-3 has 175 billion parameters!) and have very strong text generation abilities.

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Performance of Zero-Shot/Few-Shot GPT-3

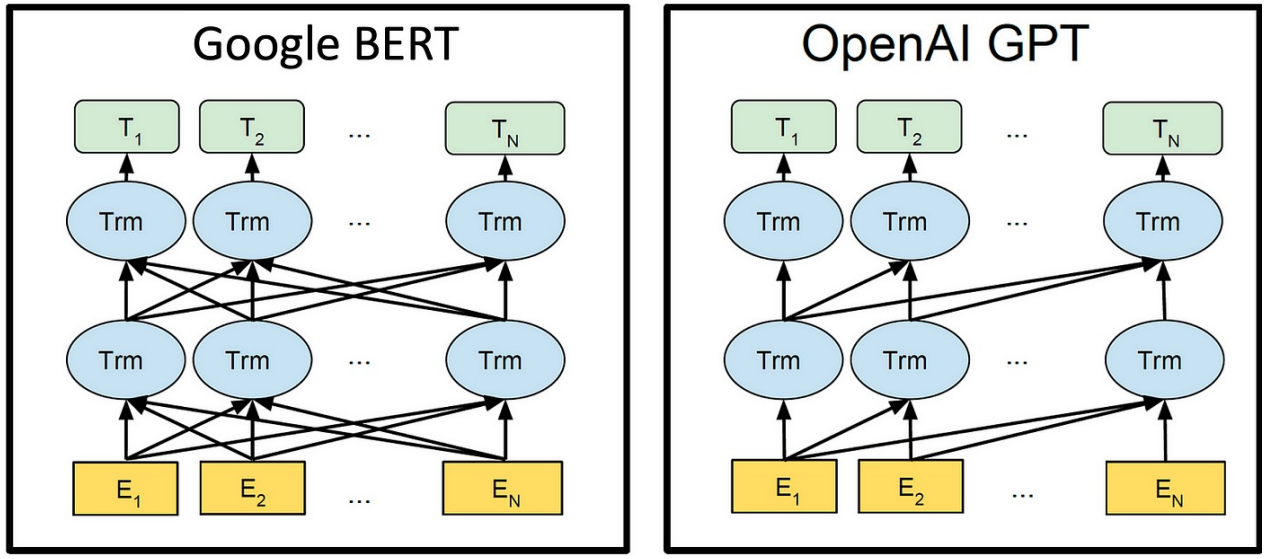
	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

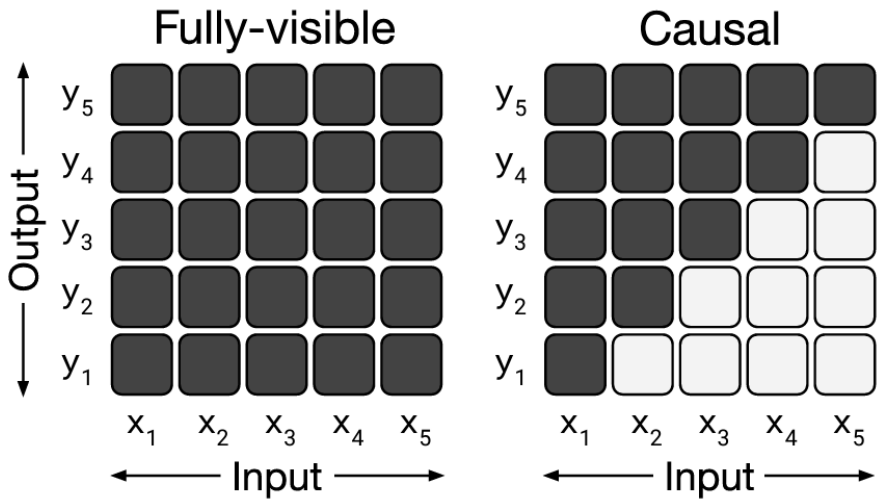
	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1



Discussion Question

- There has been fewer successful attempts to scaling up BERT-based bidirectional models (e.g., 100x) than unidirectional models. Why does unidirectional model has better scaling performance than bidirectional models?





Next Class: Instruction Tuning