Transformer Family

Sukanya Nath
Data Science Lab (DSL)
University of Bern

Transformer Applications

Encoder

- Sentence classification
- Named Entity Recognition
- Extractive Question Answering

Decoder

- Text Generation

Encoder Decoder

- Summarization
- Translation
- Generative question answering

Transformer Family

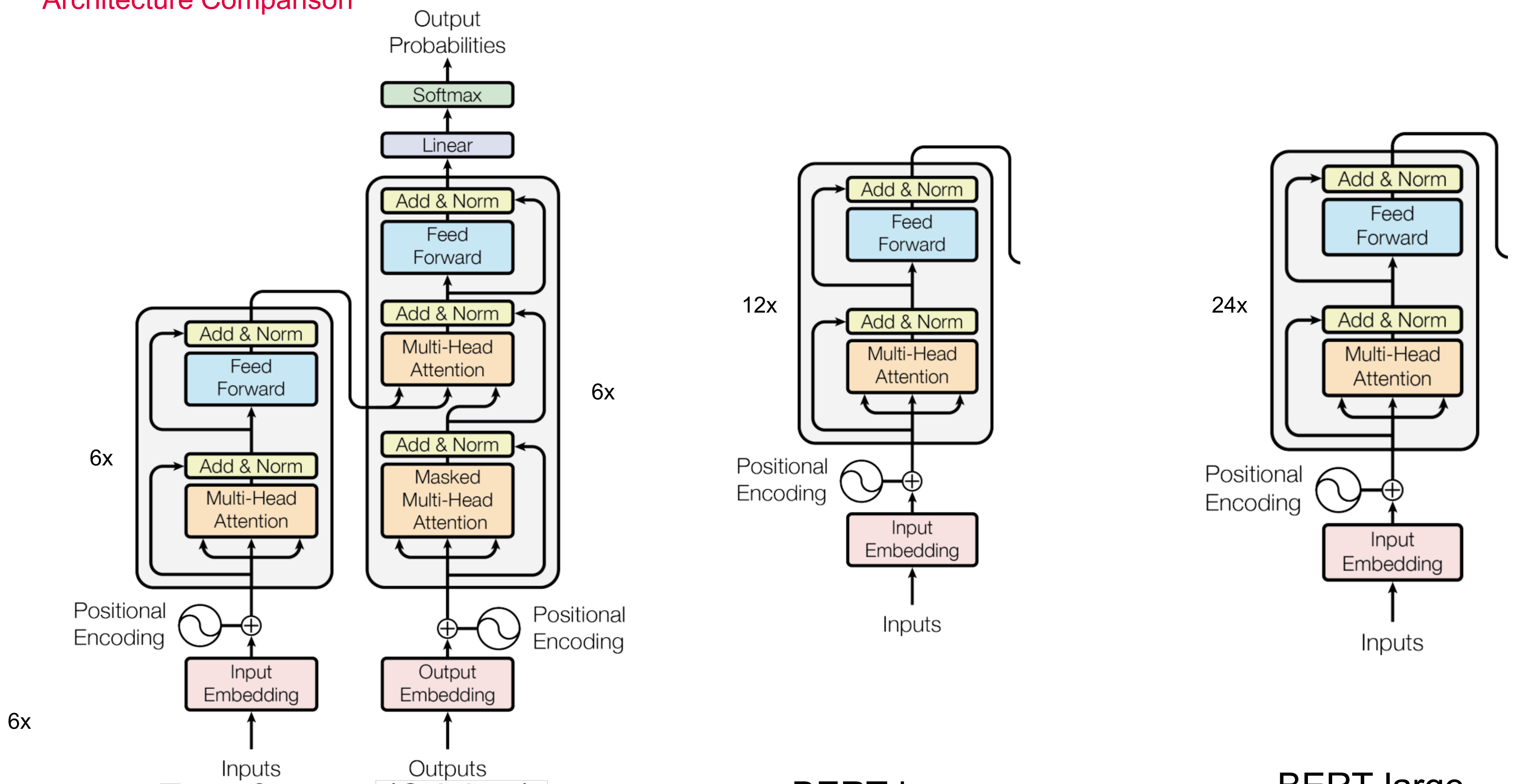
Encoder	Decoder	Encoder Decoder
BERT	GPT	BART
DistilBERT	GPT2	mBART
RoBERTa	Transformer XL	T5
ELECTRA	CTRL	
ALBERT		

Transformer Family

Encoder	Decoder	Encoder Decoder
BERT	GPT	BART
DistilBERT	GPT2	mBART
RoBERTa	Transformer XL	T5
ELECTRA	CTRL	
ALBERT		

Bidirectional Encoder Representations from Transformers

Architecture Comparison



Transformers (Original)

BERT-base

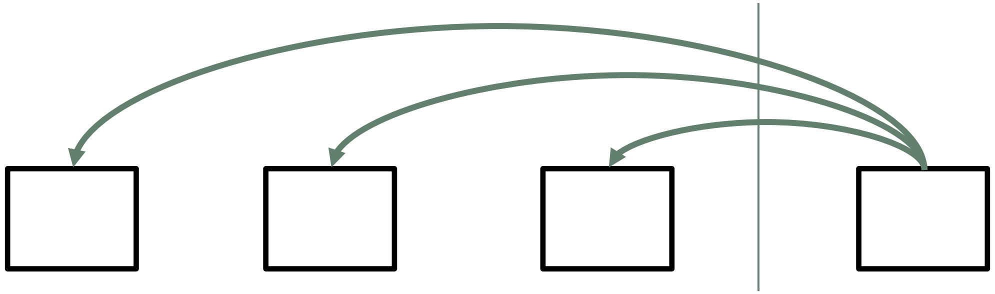
BERT-large

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

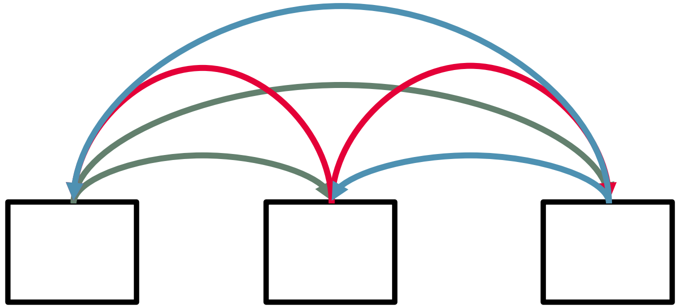
Dimension Comparison

	Transformer (Vaswani et al 2017)	BERT- base	BERT - large
N (Number of encoder layers)	6	12	24
d_{model} (Model Dimensions)	512	768	1024
A (Number of attention heads)	8	12	16
z_a (Dimension of each head)	512/8 = 64	768/12 = 64	1024/12 = 64

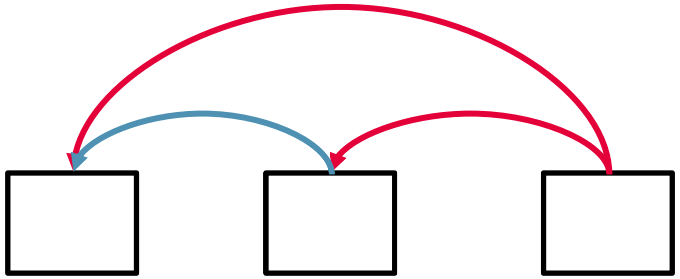
Three ways of attention



Encoder Decoder attention



Encoder Self-Attention



Masked Decoder Self-Attention

BERT Pre-Training Objective 1

Masked Language Modelling

- MLM randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary id of the masked word based only on its context.
- The MLM objective helps in training in a deeply bidirectional manner by fusing the left and the right context.
- 15% of the tokens are masked in each sequence at random

I liked it because it was a Bluebird

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

BERT Pre-Training Objective 1

1. Masking whole sequence by a decoder
   – *I liked <Masked Sequence>*

2. Masking by MLM
   – *I liked <Mask> because it was a Bluebird*

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

BERT Pre-Training Objective 2

Next Sentence Prediction

- Binary classification if Sentence B follows Sentence A.
- 50% of the time B is the actual next sentence that follows A
- Special tokens added
  - [CLS] for binary classification
  - [SEP] denoting end of a sequence.

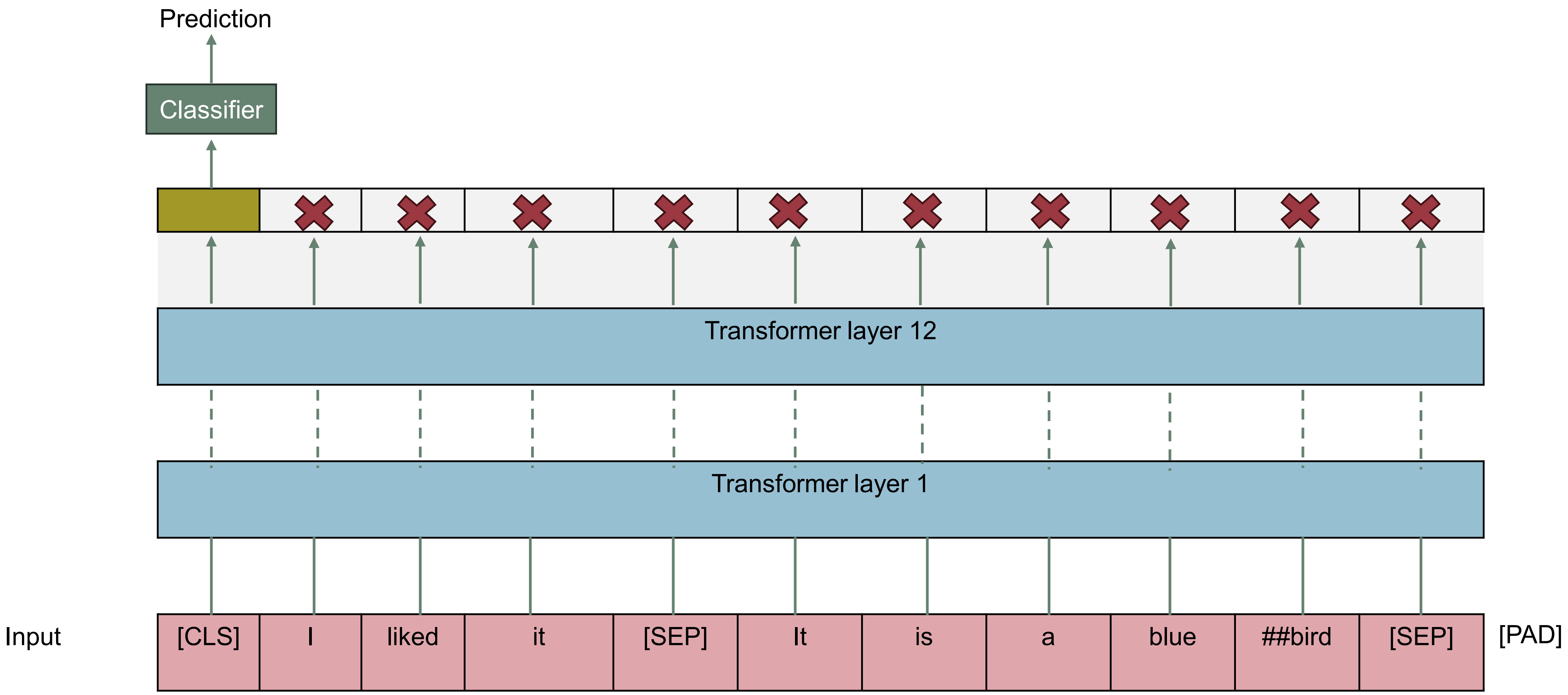
I liked it. It was a Bluebird → *[CLS] i liked it [SEP] it was a Bluebird [SEP]*

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

BERT Input Representation

Input	[CLS]	I	liked	it	[SEP]	It	is	a	blue	##bird	[SEP]
Token Embeddings	$E_{[CLS]}$	E_i	E_{liked}	E_{it}	$E_{[SEP]}$	E_{it}	E_{is}	E_a	E_{blue}	$E_{\#\#bird}$	$E_{[SEP]}$
	+	+	+	+	+	+	+	+		+	+
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B	E_B
	+	+	+	+	+	+	+	+	+	+	+
Positional Encoding	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}

BERT Architecture



Pre-Training BERT Steps

1. Prepare dataset in the desired language.
   – Original BERT used the BookCorpus dataset.
2. Train a Tokenizer on the training dataset.
3. Preprocess the dataset.
4. Pre-train BERT using MLM and NSP objectives.

Fine Tuning BERT Steps

1. Initialise a pre-trained BERT model with the respective configurations.
2. Prepare labelled training data for the downstream task (e.g. Text Classification, Question Answering).
3. Tokenize the input text using the BERT tokenizer.
4. Fine tune Model by training.

Limitations of BERT

1. Masking is performed in a static manner only once during the pretraining
   – [MASK] token is never seen during finetuning.
   – Only 15% tokens are masked and predicted which could mean data is underutilized.
2. Computational Complexity and large size of the BERT model can lead to higher latency.

Limitations of BERT

1. Masking is performed in a static manner only once during the pretraining
   - [MASK] token is never seen during finetuning.
   - Only 15% tokens are masked and predicted which could mean data is underutilized.
2. Computational Complexity and large size of the BERT model can lead to higher latency.

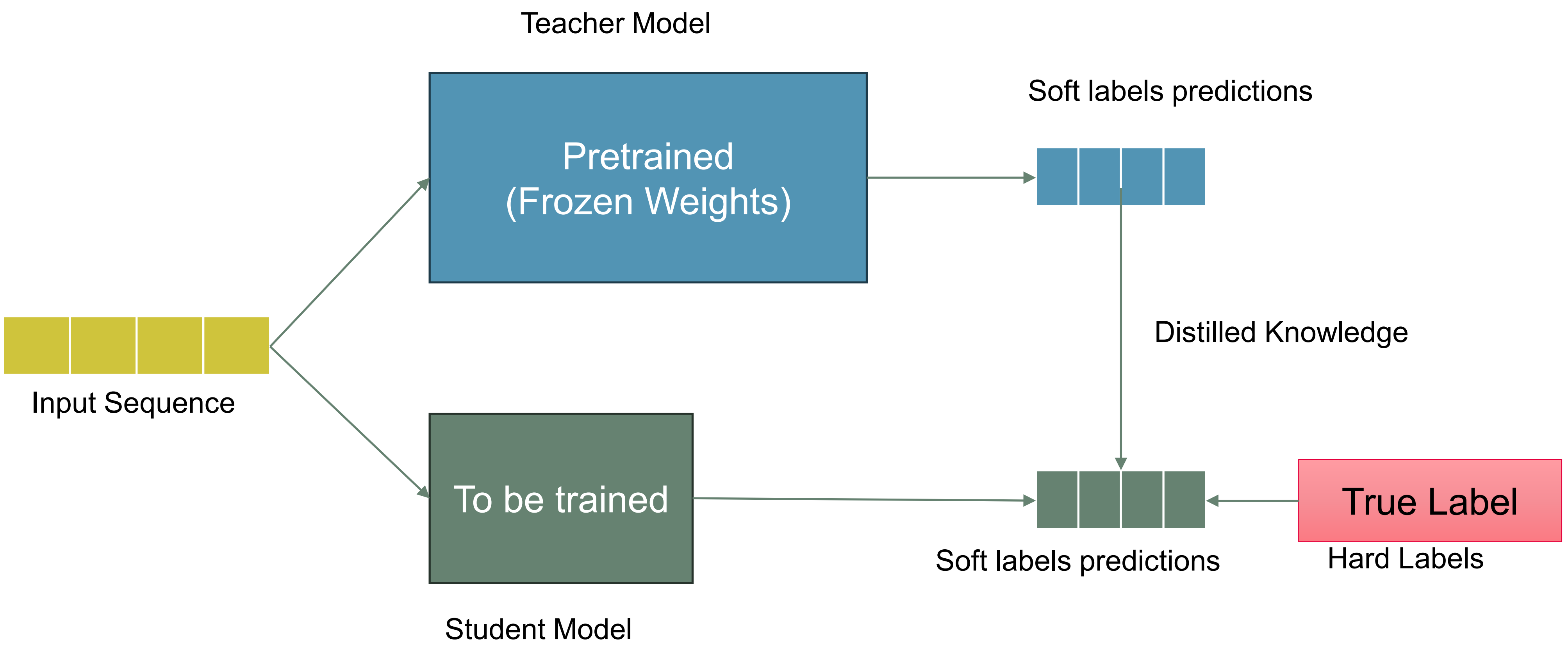
What are our alternatives?

DistilBERT

1. Uses Distillation technique which is a compression technique in which a compact model - the student - is trained to reproduce the behaviour of a larger but better performing model - the teacher - or an ensemble of models
2. DistilBERTas shown to achieve 97% of BERT’s results with 40% less memory and with 60% higher speed.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Knowledge Distillation



RoBERTa

A Robustly Optimized BERT Pretraining Approach

1. Modified the pre-training approach
   - Dynamic Masking in MLM that is randomly generated every time a sample is fed into the model.
   - Removed NSP task.
2. Trained on a much larger dataset with longer sequences and bigger batch sizes as compared to BERT
   - Datsets used: BookCorpus, CC-News, OpenWebText, STORIES
3. Removing NSP improved performance.
4. RoBERTa shown to outperform BERT by a large margin for NLU and QA tasks

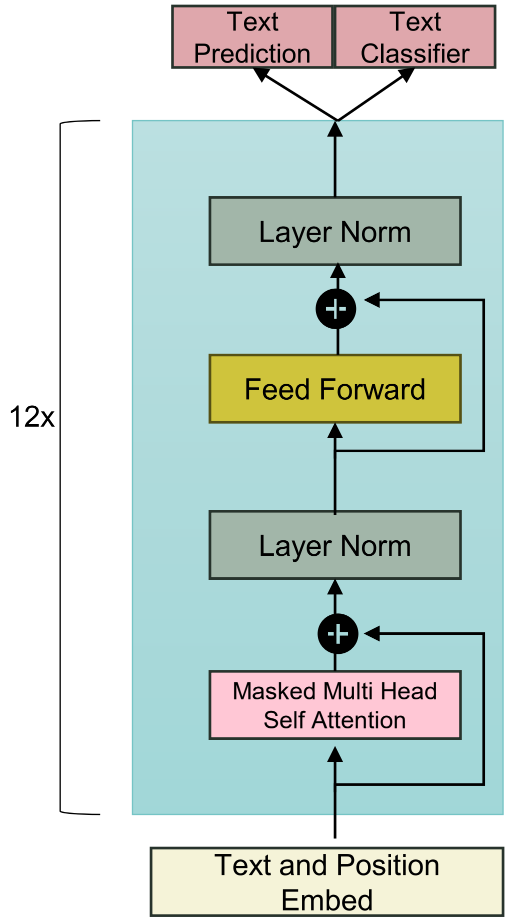
Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Transformer Family

Encoder	Decoder	Encoder Decoder
BERT	GPT	BART
DistilBERT	GPT2	mBART
RoBERTa	Transformer XL	T5
ELECTRA	CTRL	
ALBERT		

Generative Pre-trained Transformer (GPT)

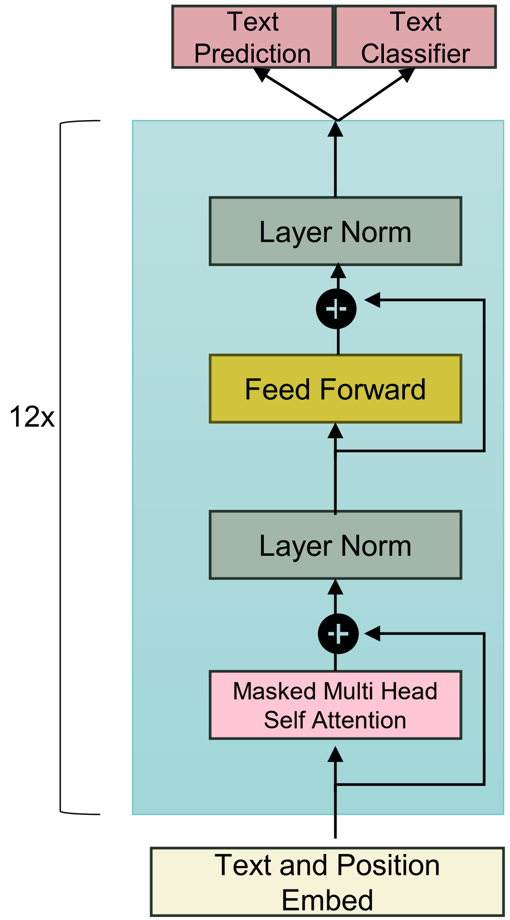
1. It is an autoregressive model that uses attention unidirectionally i.e to predict the next token in a sequence based on the previous tokens
2. Architecture: 12-layer decoder-only transformer with masked self-attention heads ($d_{model} = 768$).



Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

Generative Pre-trained Transformer (GPT)

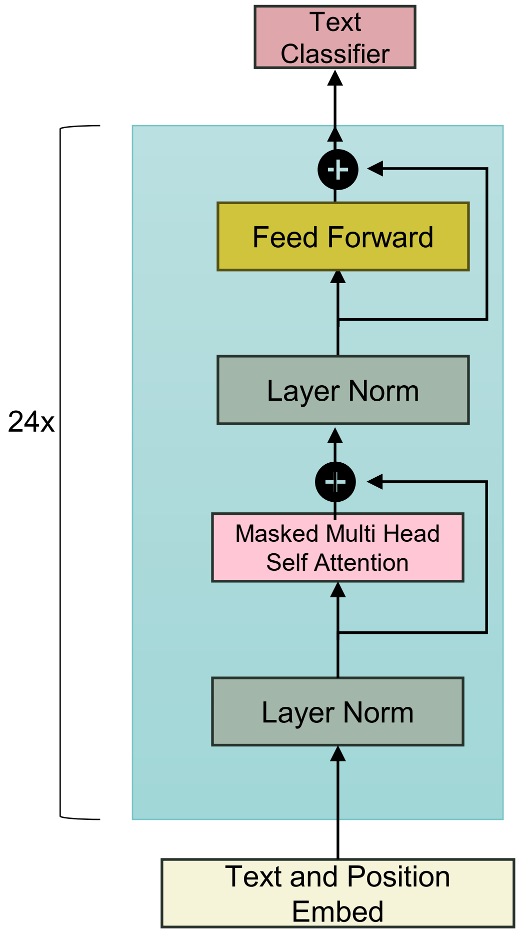
4. Proposes a two step Training:
   - Generative pre-training (large unlabeled data)
   - Discriminative fine-tuning (small labeled data)
5. Dataset: Book Corpus
6. Number of parameters: 100M
7. Showed to outperform existing models (original transformer, LSTM) on reasoning, question answering, textual entailment.



Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

GPT-2

1. Similarities with GPT
   - Unidirectional language modelling
2. Improvements compared to GPT
   - Larger Dataset (WebText 8M Documents)
   - Larger Model (1.5B Parameters)
   - No Fine tuning (Zero Shot learning)
   - Architecture changes



Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

GPT-2

Architecture:

– 24 - 48 layer decoder-only transformer with masked self-attention heads (d_{model} ranges from 1024 -1600)

– Rearranged the layer norm and residual layers

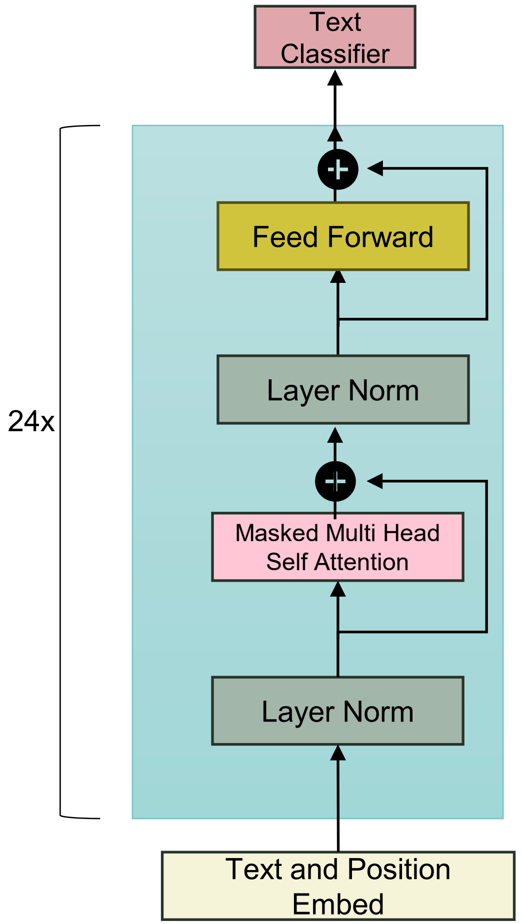
– Vocabulary size increased (30k -> 50k)

– Context Size increased (512 -> 1024 tokens)

Performance

– Increasing model size increased performance

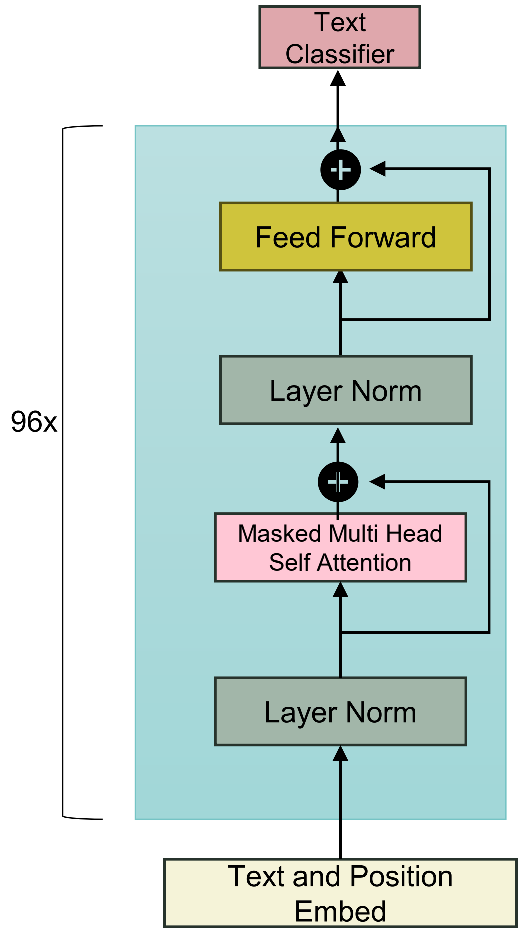
– Beat the SOTA models on Zero shot learning tasks such as Common Sense Reasoning, Question Answering, Summarization etc.



Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

GPT-3

1. Similarities with GPT-2
   - Unidirectional language modelling
   - Architecture
2. Improvements compared to GPT-2
   - Larger Dataset (300B tokens from Common Crawl, WebText2, Books1&2, Wikipedia)
   - Larger Model (175B Parameters)
   - Zero Shot, One Shot and Few Shot Task Learning
3. Implicit Task Learning via in-context learning



Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877-1901.

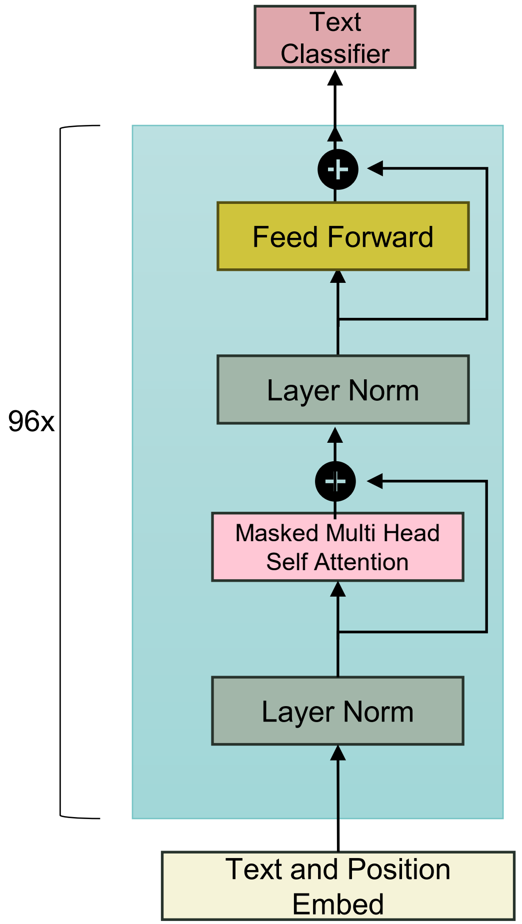
GPT-3

Architecture:

- 96-layer decoder-only transformer with masked self-attention heads (d_{model}= 12288)
- Context Size increased (1024 -> 2048 tokens)
- Alternating dense and locally banded sparse attention patterns in the layers of the transformer.

Performance:

- Beat the SOTA models on Question Answering, Summarization etc.



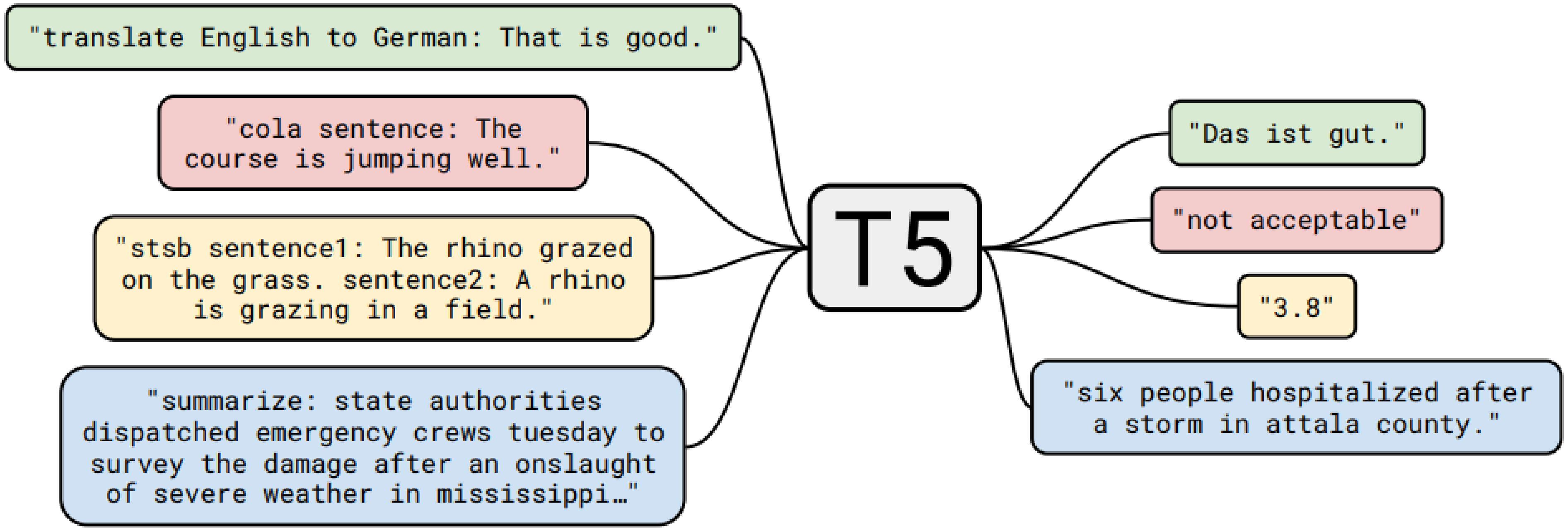
Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Transformer Family

Encoder	Decoder	Encoder Decoder
BERT	GPT	BART
DistilBERT	GPT2	mBART
RoBERTa	Transformer XL	T5
ELECTRA	CTRL	
ALBERT		

Text-to-Text Transfer Transformer (T5)

Unifies NLU and NLG tasks by converting them to text-to-text generation.



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. *The Journal of Machine Learning Research*, *21*(1), 5485-5551.

Text-to-Text Transfer Transformer (T5)

1. Architecture
   - Roughly equivalent to the original Transformer
   - 12 layers of encoder and decoder
   - d_{model}= 768
2. Number of parameters = 220M
3. During Pretraining, 15% of the tokens are dropped randomly, masking consecutively dropped tokens with a single sentinel token

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. *The Journal of Machine Learning Research*, *21*(1), 5485-5551.

Text-to-Text Transfer Transformer (T5)

1. Dataset: Colossal Clean Crawled Corpus (C4).
2. Results
   - Performance was greatly improved by pretraining.
   - The Encoder Decoder architecture with denoising performed the best.
   - Sharing parameters can reduce the number of parameters to half with minimal loss of performance.