Lecture 8-4: GPT

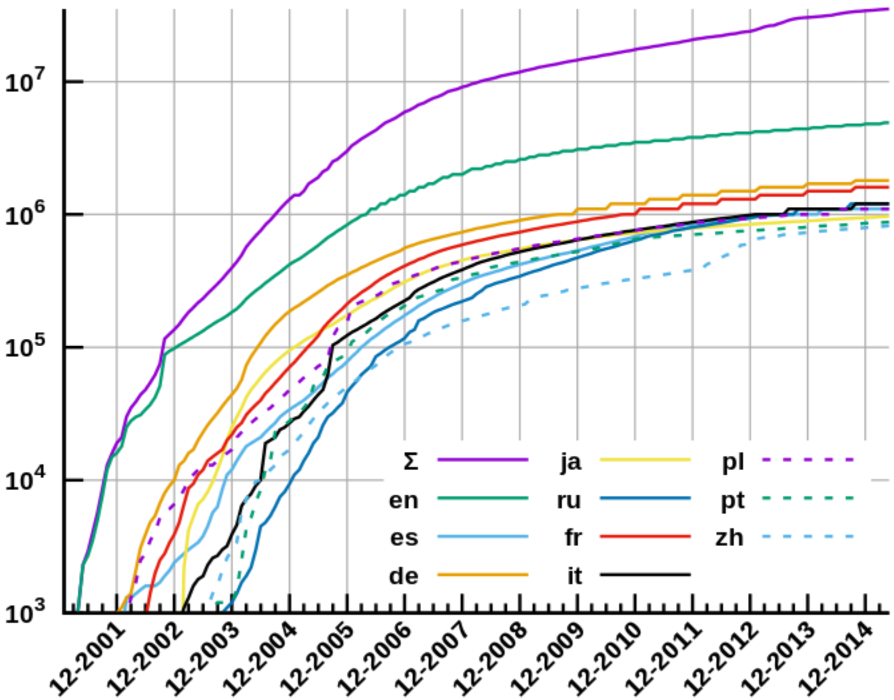
Pilsung Kang
School of Industrial Management Engineering
Korea University

GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

- Backgrounds

Unlabeled dataset



As of 24 February 2020, there are 6,020,081 articles in the English Wikipedia containing over 3.5 billion words.

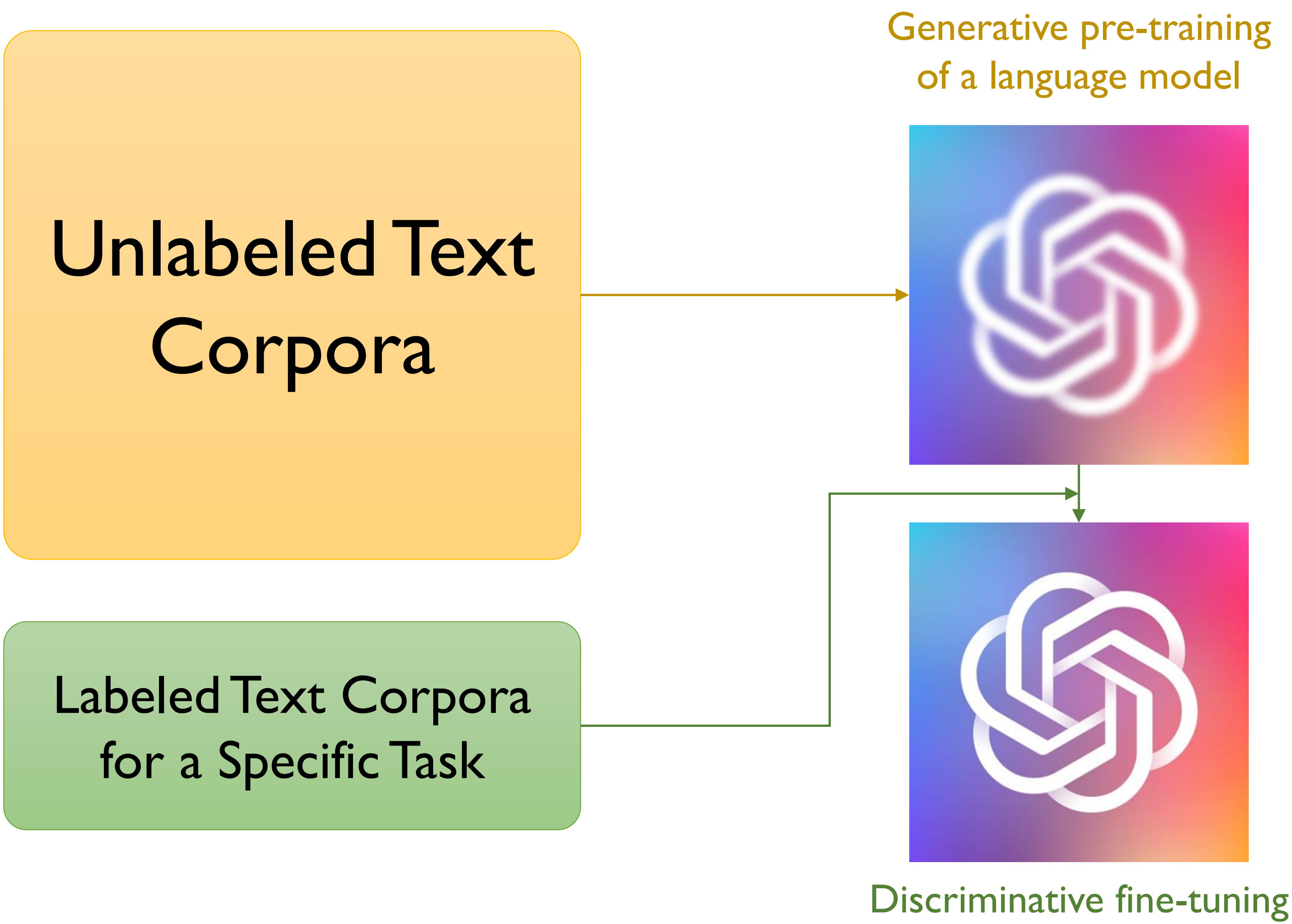
Labeled dataset

- STS Benchmark for sentence similarity: 8,628 sentences
- Quora question pairs: 404,290 question pairs
- CoLA dataset: 10,657 sentences

GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

- Motivation



GPT: Generative Pre-Training of a Language Model

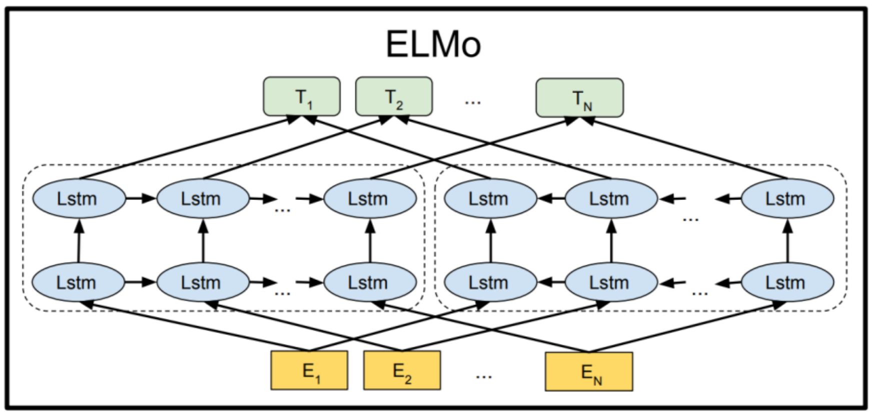
Radford et. al (2018)

- Leveraging more than word-level information from unlabeled text is challenging
  - ✓ It is unclear what type of optimization objectives are most effective at learning text representations that are useful for transfer
    - Language modeling (Peters et al., 2018), machine translation (McCann et al., 2017), discourse coherence (Jernite et al., 2017), etc.
  - ✓ There is no consensus on the most effective way to transfer these learned representations to the target task
    - Making task-specific changes to model architecture, using intricate learning scheme, adding auxiliary learning objective

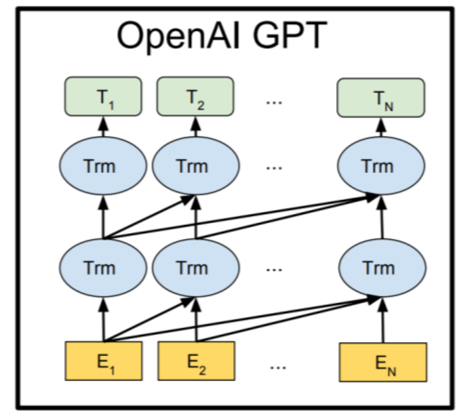
GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

- GPT: Unsupervised pre-training
  - ✓ Illustrative difference between EMLo and GPT



VS



  - ✓ Given an unsupervised corpus of tokens, $\mathcal{U} = (u_1, u_2, ..., u_n)$, a standard language modeling objective to maximize the following likelihood is used:

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, ..., u_{i-1}; \Theta)$$

    - k is the size of context window
    - P is the conditional probability modeled using a neural network with parameter Θ

GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

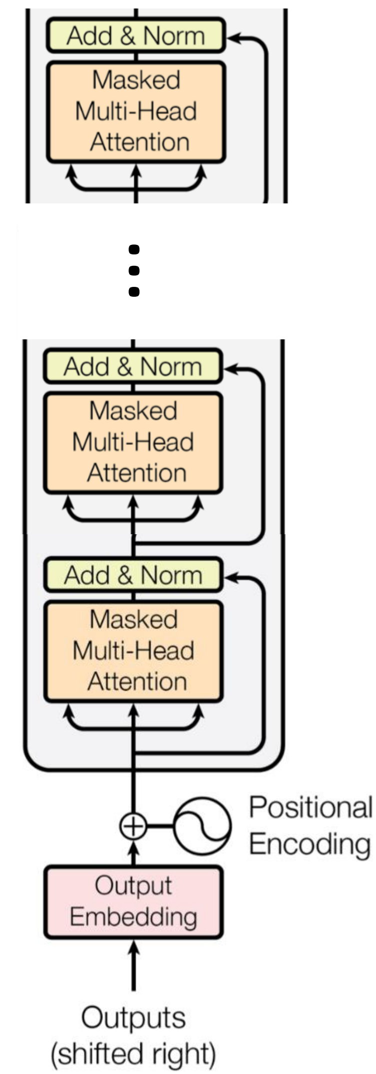
- GPT: Unsupervised pre-training
  - ✓ A multi-layer Transformer decoder is used for language model

$$h_0 = UW_e + W_p$$

$$h_l = transformer\_block(h_l - 1), \quad \forall i \in [1, n]$$

$$P(u) = softmax(h_n W_e^T)$$

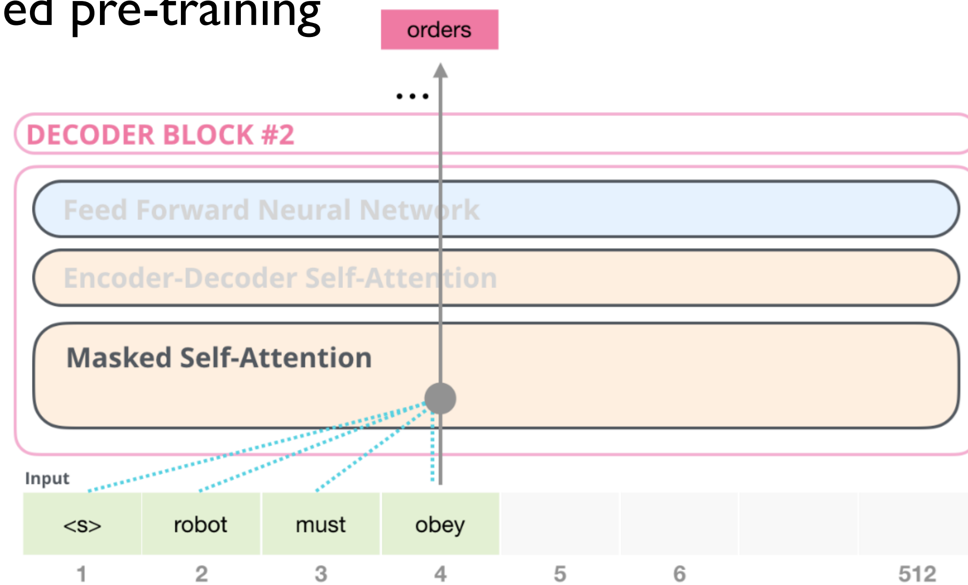
- $U = (u_{-k}, ..., u_{-1})$: the context vector of tokens
- n: the number of layers
- W_e: token embedding matrix
- W_p: position embedding matrix

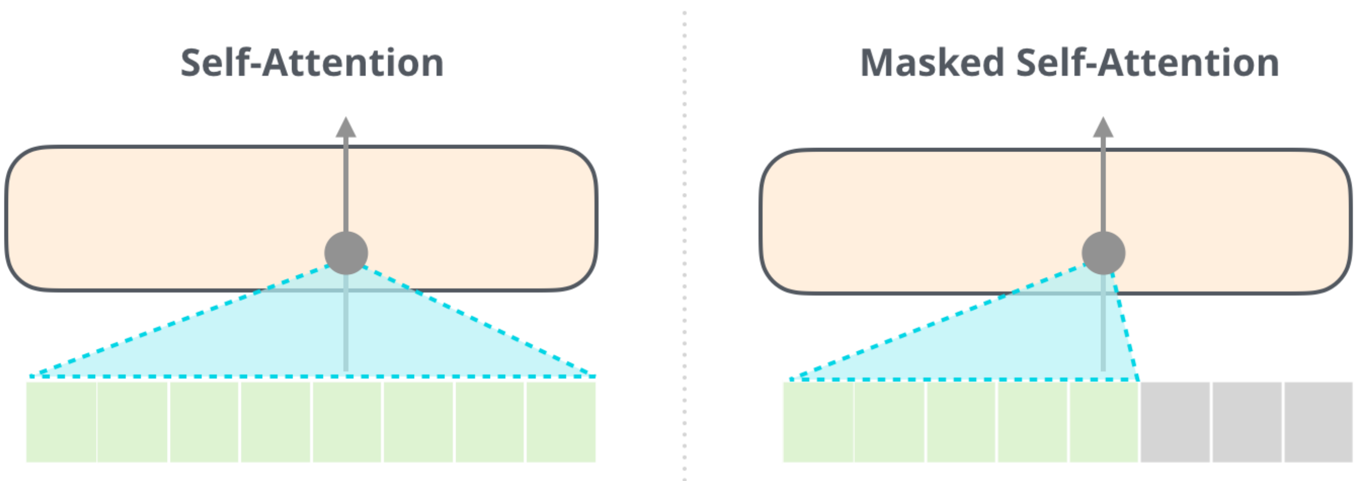


GPT: Generative Pre-Training of a Language Model

Alammar (GPR-2)

- GPT: Unsupervised pre-training

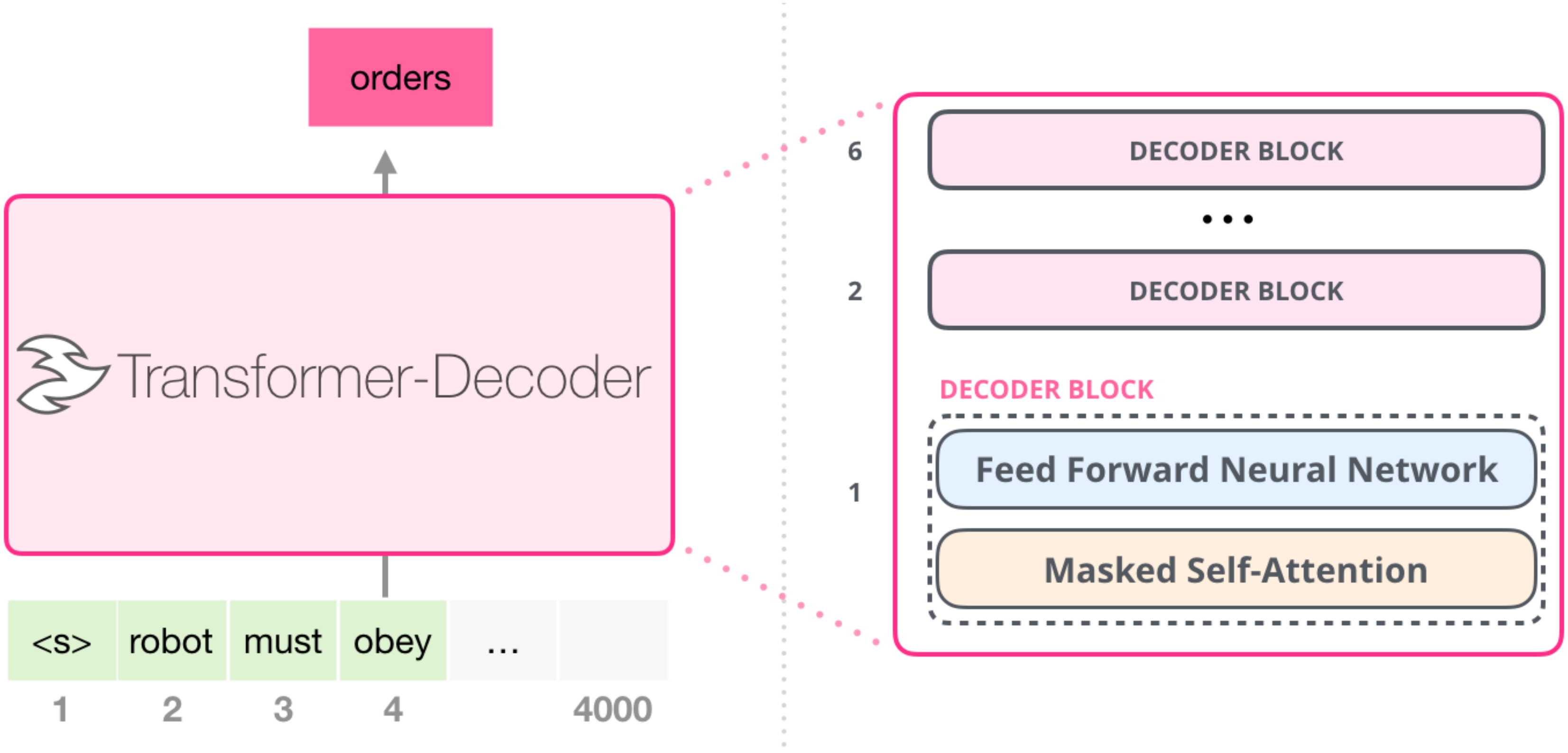




GPT: Generative Pre-Training of a Language Model

Alammar (GPR-2)

- GPT: Unsupervised pre-training
  - ✓ Decoder-only block



GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

- GPT: Supervised fine-tuning
  - ✓ A labeled dataset $\mathcal{C}$ with each instance consisting of a sequence of input tokens, x^1, …, x^m, along with a label y
  - ✓ The inputs are passed through the pre-trained model to obtain the final transformer block's activation h_l^m, which is then fed into an added linear output layer with parameter W_y to predict y:

$$P(y|x^1, ..., x^m) = softmax(h_l^m W_y)$$

  - ✓ This gives us the following objective to maximize:

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, ..., x^m)$$

GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

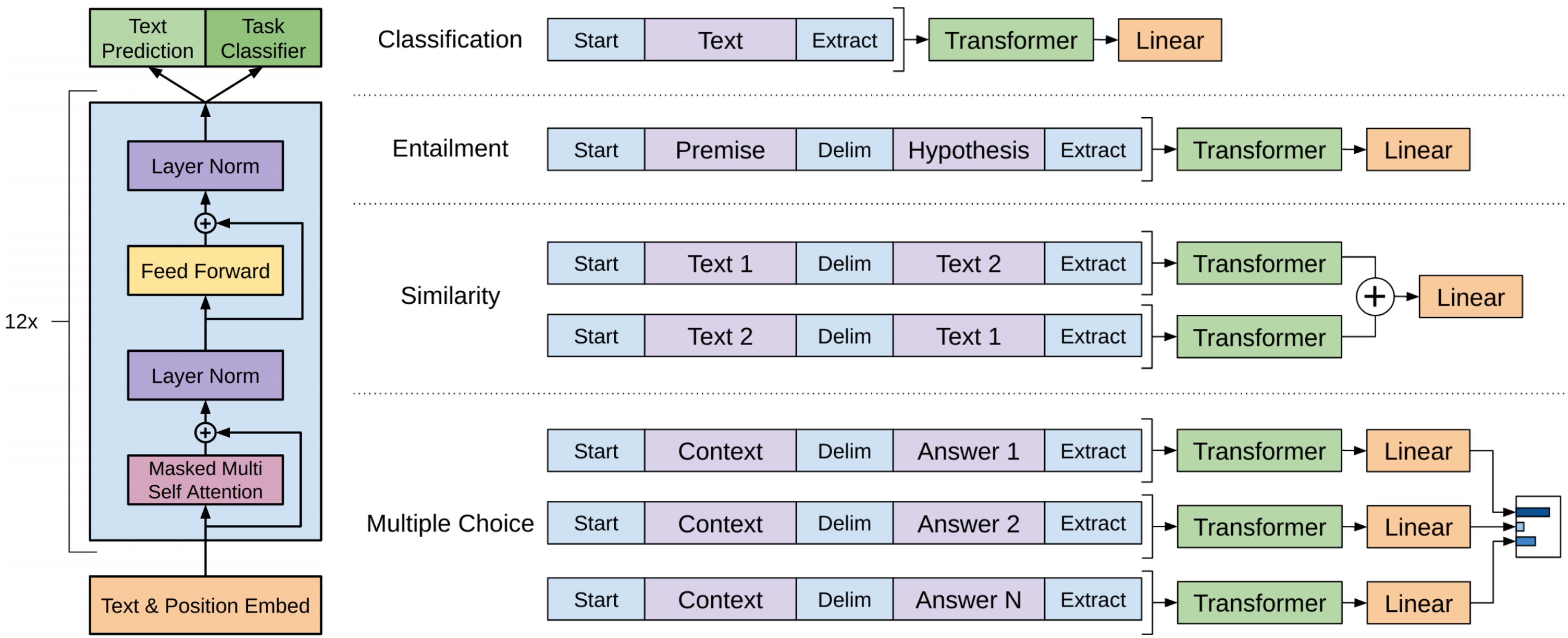
- GPT: Supervised fine-tuning
  - ✓ The authors additionally found that including language modeling as an auxiliary objective to the fine-tuning helped learning by
    - Improving generalization of the supervised model
    - Accelerating convergence

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda \times L_1(\mathcal{C})$$

GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

- GPT: Task-specific input transformations



GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

- Experiments
  - ✓ Pre-training
    - ▪ BookCorpus

# of books	# of sentences	# of words	# of unique words	mean # of words per sentence	median # of words per sentence
11,038	74,004,228	984,846,357	1,316,420	13	11

- ▪ 1 Billion Word Language Model Benchmark (used by ELMo)
  - https://www.statmt.org/lm-benchmark/

1 Billion Word Language Model Benchmark

paper | code | data | output probabilities

The purpose of the project is to make available a standard training and test setup for language modeling experiments.

The training/held-out data was produced from the WMT 2011 News Crawl data using a combination of Bash shell and Perl scripts distributed here.

This also means that your results on this data set are reproducible by the research community at large.

Besides the scripts needed to rebuild the training/held-out data, it also makes available log-probability values for each word in each of ten feld-out data sets, for each of the following baseline models:

- unpruned Katz (1.1B n-grams),
- pruned Katz (~15M n-grams),
- unpruned Interpolated Kneser-Ney (1.1B n-grams),
- pruned Interpolated Kneser-Ney (~15M n-grams)

Happy benchmarking!

Service Unavailable

The server is temporarily unable to service your request due to maintenance downtime or capacity problems. Please try again later.

GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

- Experiments
  - ✓ Tasks & Datasets

Task	Datasets
Natural language inference	SNLI [5], MultiNLI [66], Question NLI [64], RTE [4], SciTail [25]
Question Answering	RACE [30], Story Cloze [40]
Sentence similarity	MSR Paraphrase Corpus [14], Quora Question Pairs [9], STS Benchmark [6]
Classification	Stanford Sentiment Treebank-2 [54], CoLA [65]

GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

- Experiments
  - ✓ Natural Language Inference

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	<u>89.3</u>	-	-	-
CAFE [58] (5x)	80.2	79.0	<u>89.3</u>	-	-	-
Stochastic Answer Network [35] (3x)	<u>80.6</u>	<u>80.1</u>	-	-	-	-
CAFE [58]	78.7	77.9	88.5	<u>83.3</u>		
GenSen [64]	71.4	71.3	-	-	<u>82.3</u>	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

  - ✓ Question & Answering

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	<u>77.6</u>	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	<u>60.2</u>	<u>50.3</u>	<u>53.3</u>
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

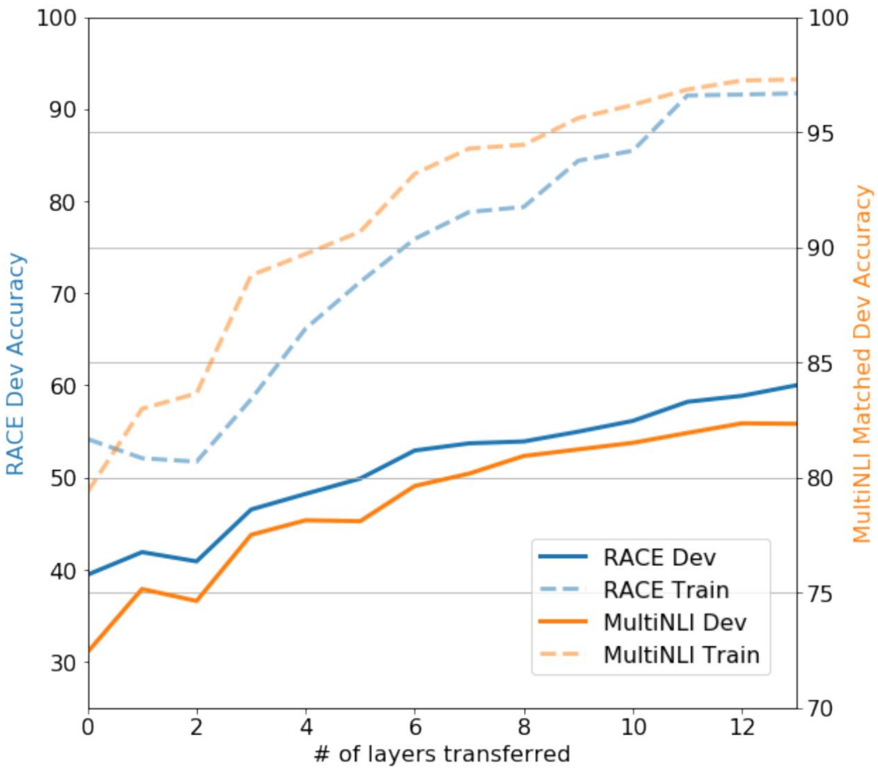
- Experiments
  - ✓ Semantic Similarity & Classification

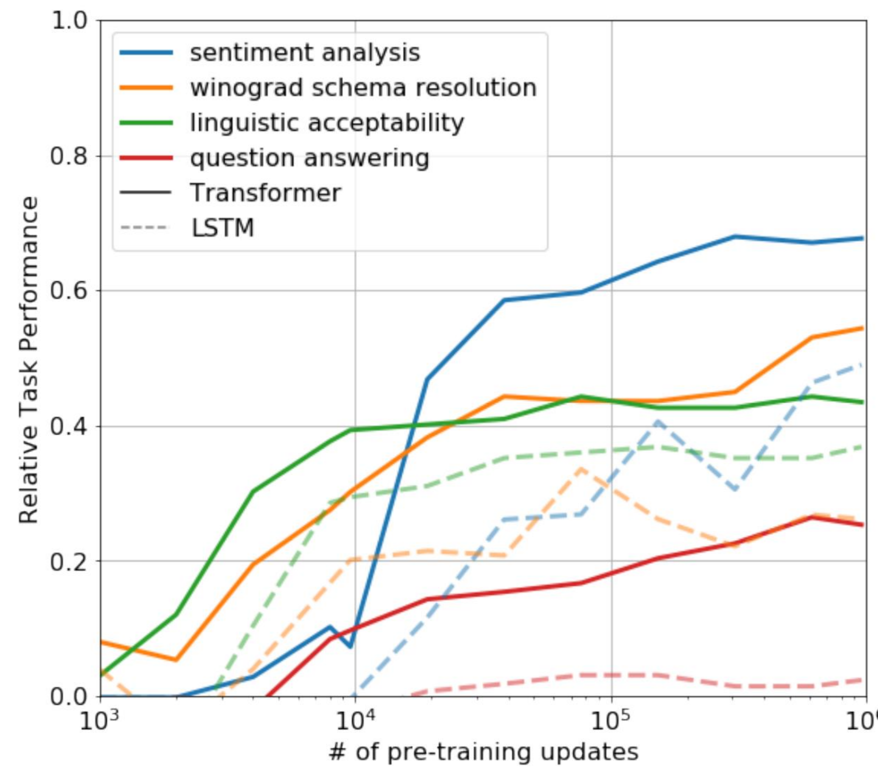
Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

- Experiments
  - ✓ Impact of number of layered transferred and Zero-shot behaviors





GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

- Experiments
  - ✓ Ablation studies

Method	Avg. Score	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	MNLI (acc)	QNLI (acc)	RTE (acc)
Transformer w/ aux LM (full)	74.7	45.4	91.3	82.3	82.0	**70.3**	**81.8**	**88.1**	**56.0**
Transformer w/o pre-training	59.9	18.9	84.0	79.4	30.9	65.5	75.7	71.2	53.8
Transformer w/o aux LM	**75.0**	**47.9**	**92.0**	**84.9**	**83.2**	69.8	81.1	86.9	54.4
LSTM w/ aux LM	69.1	30.3	90.5	83.2	71.8	68.1	73.7	81.1	54.6

  - Larger datasets benefit from the auxiliary objective but smaller datasets do not
  - LSTM only outperforms the Transformer on one dataset


ANY
questions?