

Lecture 8-2: Pre-Trained Models ELMo, GPT, BERT, and GPT-2

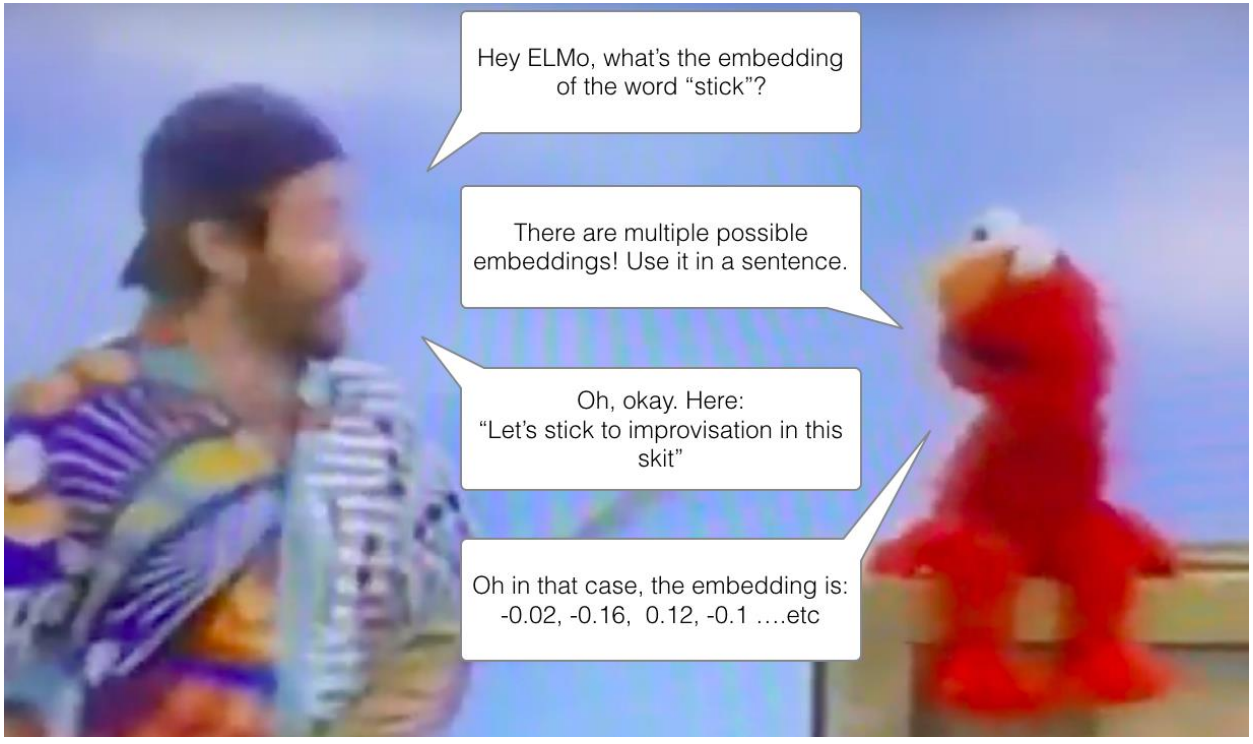
Pilsung Kang
School of Industrial Management Engineering
Korea University

AGENDA

ELMo: Embeddings from Language Models

Peters et. al (2018)

- Pre-trained word representations
  - ✓ A key component in many neural language understanding models
- High quality representations should ideally model
  - ✓ Complex characteristics of word use (e.g., syntax and semantics)
  - ✓ How these uses vary across linguistic contexts (i.e., to model polysemy)



http://jalammar.github.io/illustrated-bert/

ELMo: Embeddings from Language Models

Peters et. al (2018)

- GloVe vs. ELMo

Example

GloVe mostly learns *sport*-related context

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

Table 4: Nearest neighbors to "play" using GloVe and the context embeddings from a biLM.

ELMo can distinguish the word sense based on the context

https://www.slideshare.net/shuntaroy/a-review-of-deep-contextualized-word-representations-peters-2018

ELMo: Embeddings from Language Models

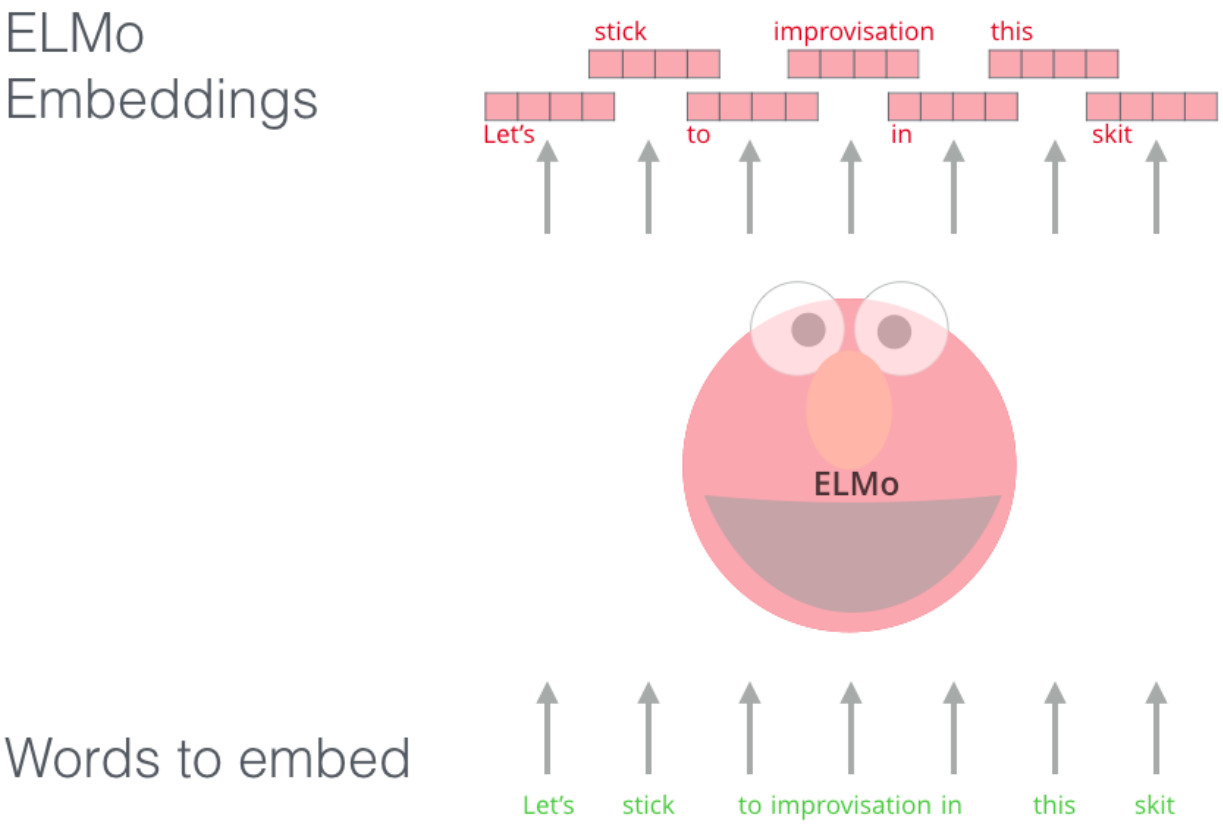
Peters et. al (2018)

- ELMo
  - ✓ Each token is assigned a representation that is a function of the entire input sentence
  - ✓ Use vectors derived from a bidirectional LSTM that is trained with a coupled language model (LM) objective on a large text corpus
- Features
  - ✓ ELMo representations are deep in the sense that they are a function of all of the internal layers of the biLM
    - a linear combination of the vectors stacked above each input word for each end task is learned, which markedly improves performance over just using the top LSTM layer
    - This allows for very rich word representations
      - Higher-level LSTM states captures context-dependent aspects of word meaning
      - Lower-level state model aspects of syntax

ELMo: Embeddings from Language Models

Peters et. al (2018)

- Graphical illustration
  - ✓ ELMo looks at the entire sentence before assigning each word in it an embedding

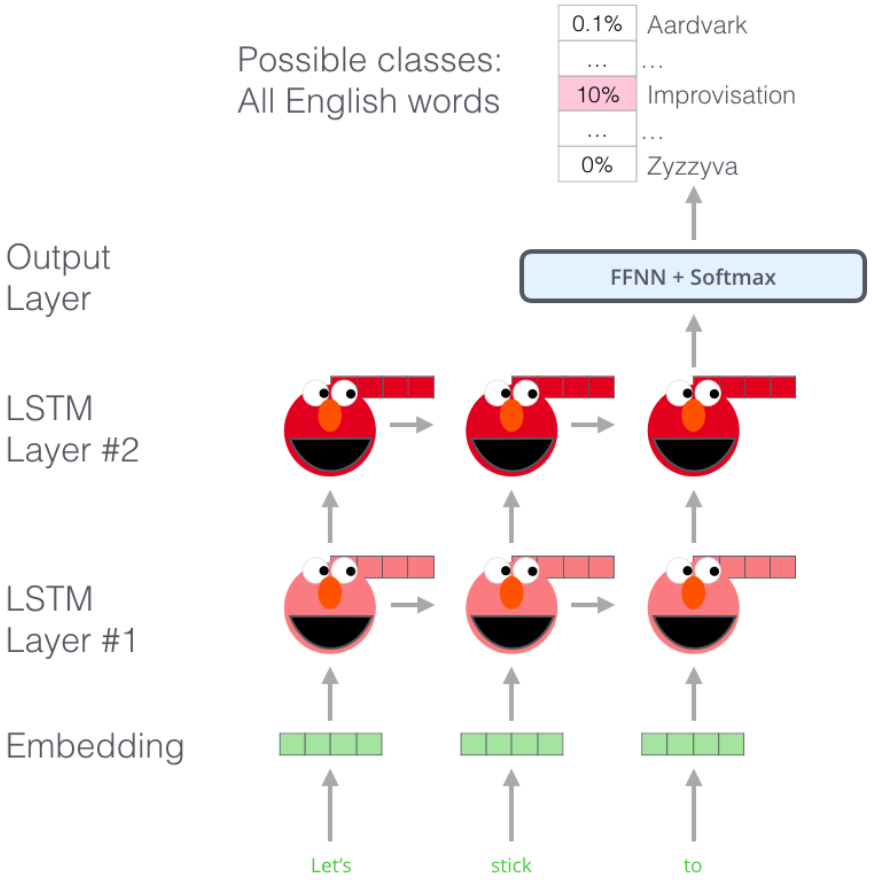


http://jalammar.github.io/illustrated-bert/

ELMo: Embeddings from Language Models

Peters et. al (2018)

- Graphical illustration
  - ✓ ELMo gained its language understanding from being trained to predict the next word in a sequence of words - a task called Language Modeling

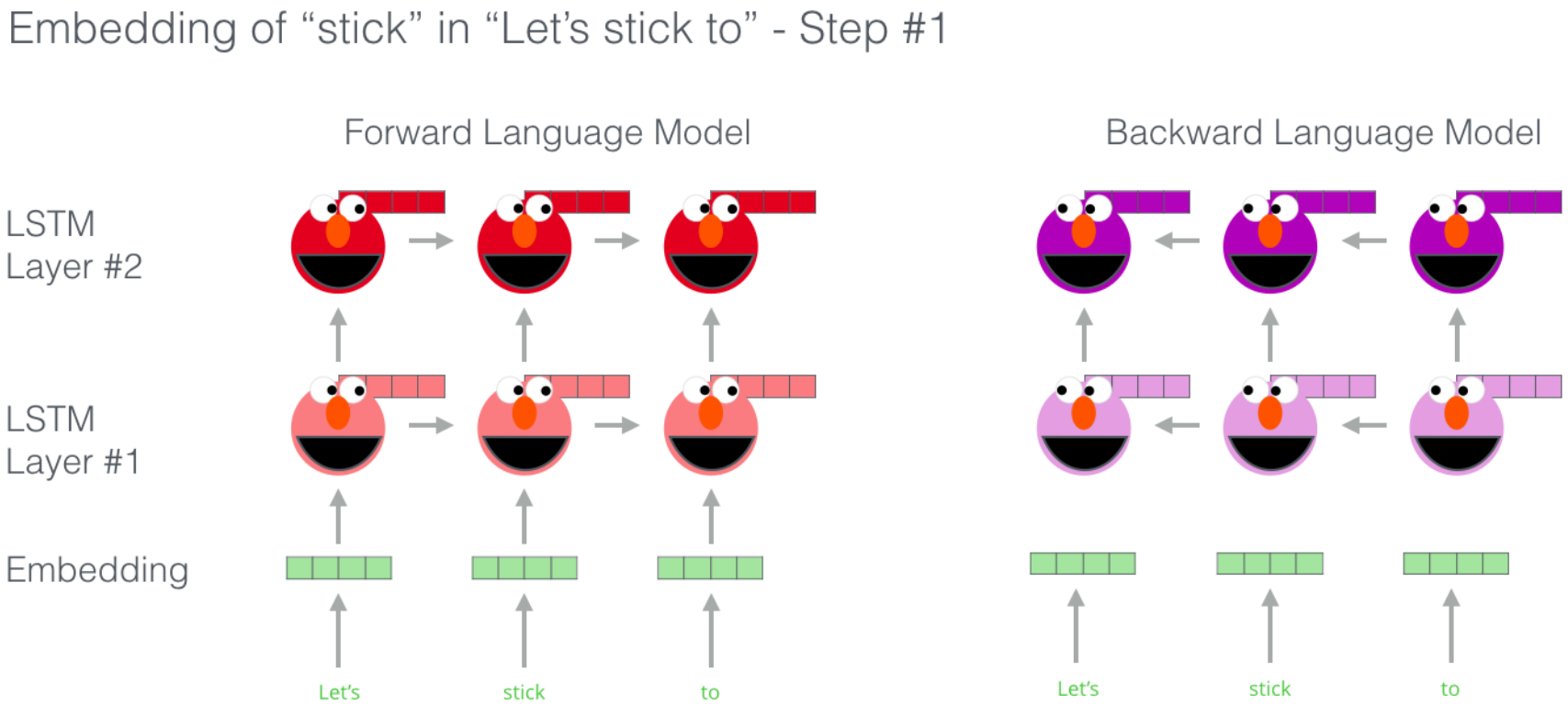


http://jalammar.github.io/illustrated-bert/

ELMo: Embeddings from Language Models

Peters et. al (2018)

- Graphical illustration
  - ✓ ELMo actually goes a step further and trains a bi-directional LSTM – so that its language model doesn't only have a sense of the next word, but also the previous word.

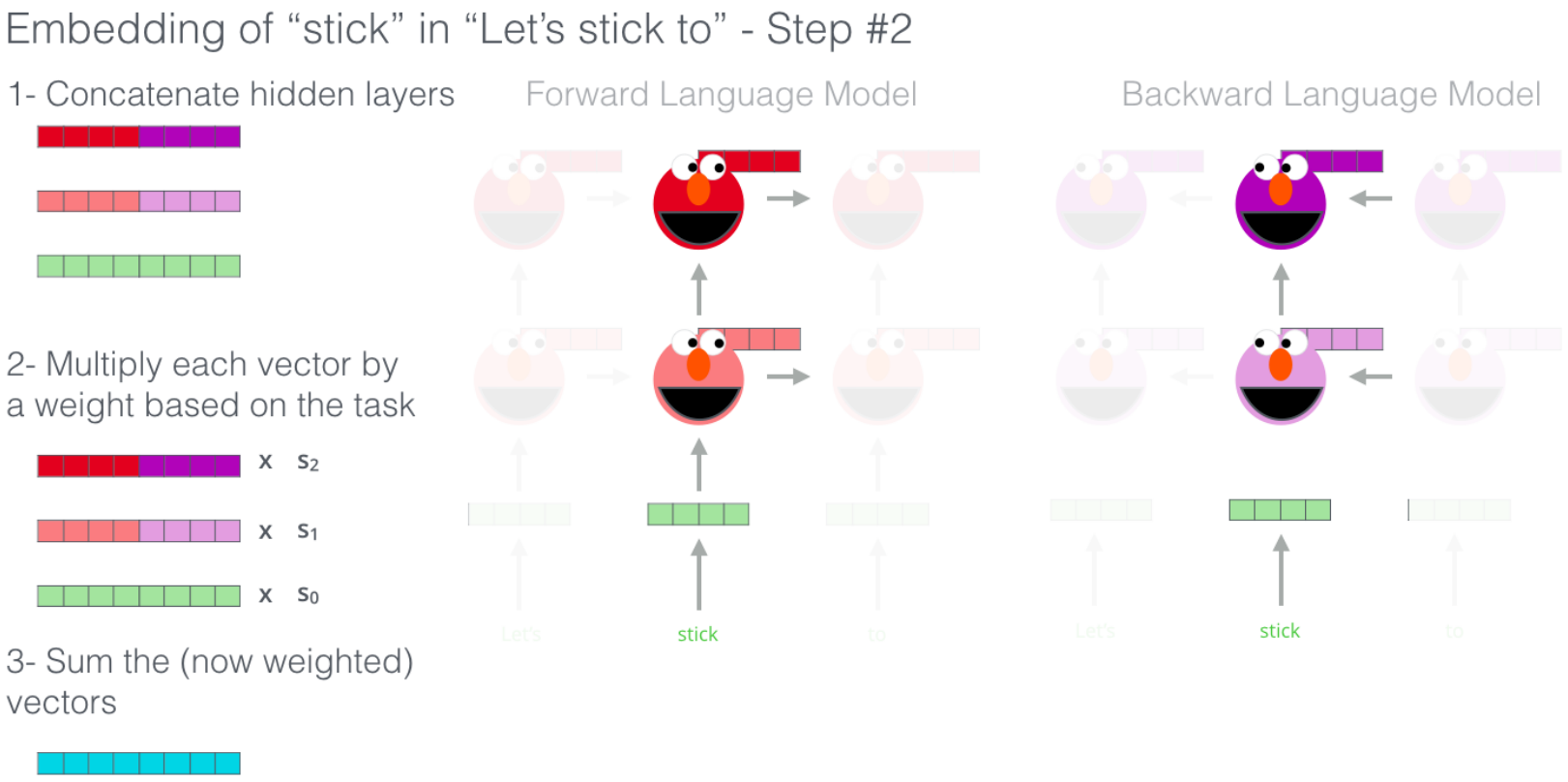


http://jalammar.github.io/illustrated-bert/

ELMo: Embeddings from Language Models

Peters et. al (2018)

- Graphical illustration
  - ✓ ELMo comes up with the contextualized embedding through grouping together the hidden states (and initial embedding) in a certain way (concatenation followed by weighted summation)



http://jalammar.github.io/illustrated-bert/

ELMo: Embeddings from Language Models

Peters et. al (2018)

- ELMo for downstream task

ELMo represents a word t_k as a linear combination of corresponding hidden layers (inc. its embedding)

ELMo is a task specific representation. A down-stream task learns weighting parameters

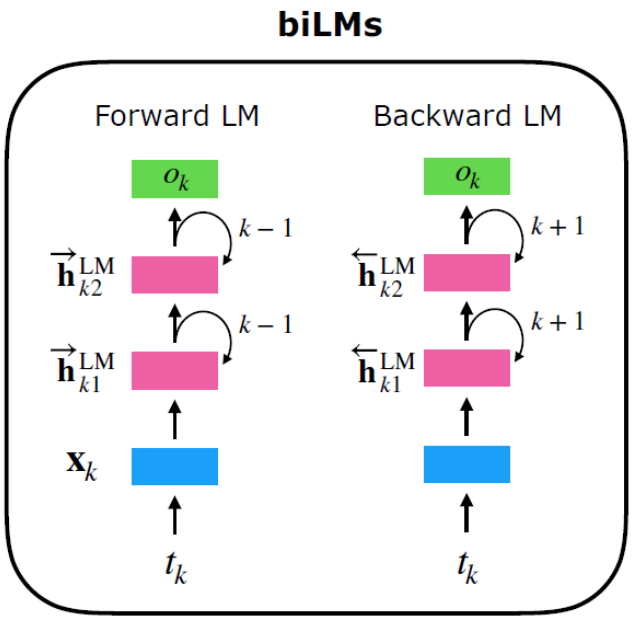
$$\mathbf{ELMo}_k^{\text{task}} = \gamma^{\text{task}} \times \sum \left\{ s_2^{\text{task}} \times \mathbf{h}_{k2}^{\text{LM}},\ s_1^{\text{task}} \times \mathbf{h}_{k1}^{\text{LM}},\ s_0^{\text{task}} \times \mathbf{h}_{k0}^{\text{LM}} \right.$$

$$\left. ([\mathbf{x}_k ; \mathbf{x}_k]) \right.$$

Concatenate hidden layers

$$[\overrightarrow{\mathbf{h}}_{kj}^{\text{LM}} ; \overleftarrow{\mathbf{h}}_{kj}^{\text{LM}}]$$

Unlike usual word embeddings, ELMo is assigned to every *token* instead of a *type*



https://www.slideshare.net/shuntaroy/a-review-of-deep-contextualized-word-representations-peters-2018

ELMo: Embeddings from Language Models

Peters et. al (2018)

- Mathematical demonstration: Bidirectional language models
  - ✓ Given a sequence of N tokens ($t_1, t_2, \ldots, t_N$), a forward language model computes the probability of the sequence by modeling probability of token t_k given the history ($t_1, t_2, \ldots, t_{k-1}$)

$$p(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} (t_k | t_1, t_2, ..., t_{k-1})$$

  - ✓ Neural language models compute a context-independent token representation x_k^{LM} (via token embeddings or a CNN over characters) then pass it through L layers of forward LSTMs
  - ✓ At each position k, each LSTM layer outputs a context-dependent representation $\overrightarrow{h}_{k,j}^{LM}$ where j = 1, …, L
  - ✓ The top layer LSTM output, $\overrightarrow{h}_{k,L}^{LM}$ is used to predict the next token t_{k+1} with a Softmax layer

ELMo: Embeddings from Language Models

Peters et. al (2018)

- Mathematical demonstration: Bidirectional language models
  - ✓ A backward LM is similar to a forward LM, except it runs over the sequence in reverse, predicting the previous token given the future context

$$p(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} (t_k | t_{k+1}, t_{k+2}, ..., t_N)$$

  - ✓ Each backward LSTM layer j in an L layer deep model producing representations $\overleftarrow{h}_{k,j}^{LM}$ of t_k given (t_{k+1}, …, t_N)

ELMo: Embeddings from Language Models

Peters et. al (2018)

- Mathematical demonstration: Bidirectional language models
  - ✓ Jointly maximizes the log likelihood of the forward and backward directions

$$\sum_{k=1}^{N} \Big(\log p(t_k | t_1, ..., t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + \log p(t_k | t_{k+1}, ..., t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \Big)$$

    - Θ_x, Θ_s : tied token representation & softmax layer parameters
    - Separated parameters for the LSTMs in each direction

ELMo: Embeddings from Language Models

Peters et. al (2018)

- ELMo
  - ✓ A task specific combination of the intermediate layer representations in the biLM
  - ✓ For each token t_k, a l-layer biLM computes a set of 2L+1 representations

$$R_k = \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} | j = 1, ..., L\} = \{\mathbf{h}_{k,j}^{LM}, | j = 0, ..., L\}$$

    - where $\mathbf{h}_{k,0}^{LM}$ is the token layer and $\mathbf{h}_{k,j}^{LM} = [\overrightarrow{\mathbf{h}}_{k,j}^{LM}; \overleftarrow{\mathbf{h}}_{k,j}^{LM}]$ for each biLSTM layer
  - ✓ For inclusion in a downstream model, ELMo collapses all layers in R into a single vector

$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}$$

allows task model to scale the entire ELMo vector

softmax-normalized weights

ELMo: Embeddings from Language Models

Peters et. al (2018)

- Natural language inference (NLI) task
  - ✓ Classify two given sentence to one of the three classes: entailment, contradiction, neutral
    - Examples (https://nlp.stanford.edu/projects/snli/)

Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

ELMo: Embeddings from Language Models

Peters et. al (2018)

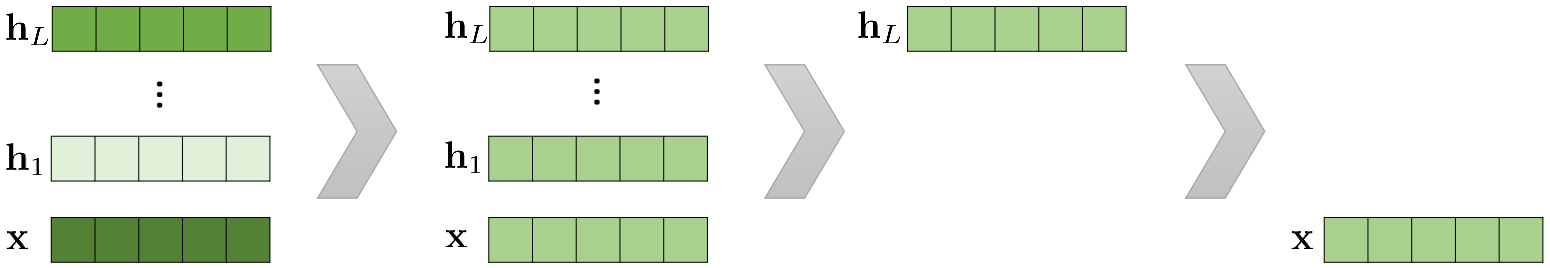
- Performances

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

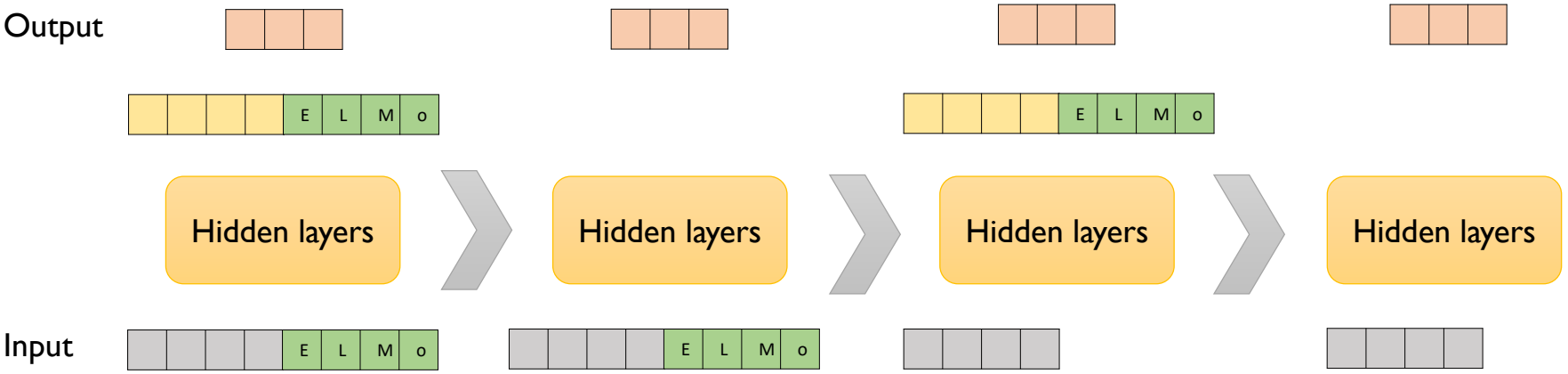
ELMo: Embeddings from Language Models

Peters et. al (2018)

- Analysis: Alternate layer weighting scheme



- Analysis: Where to include ELMo?



ELMo: Embeddings from Language Models

Peters et. al (2018)

- Analysis: What information is captured by the biLM's representation?
  - ✓ Disambiguating the meaning of words using their context

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

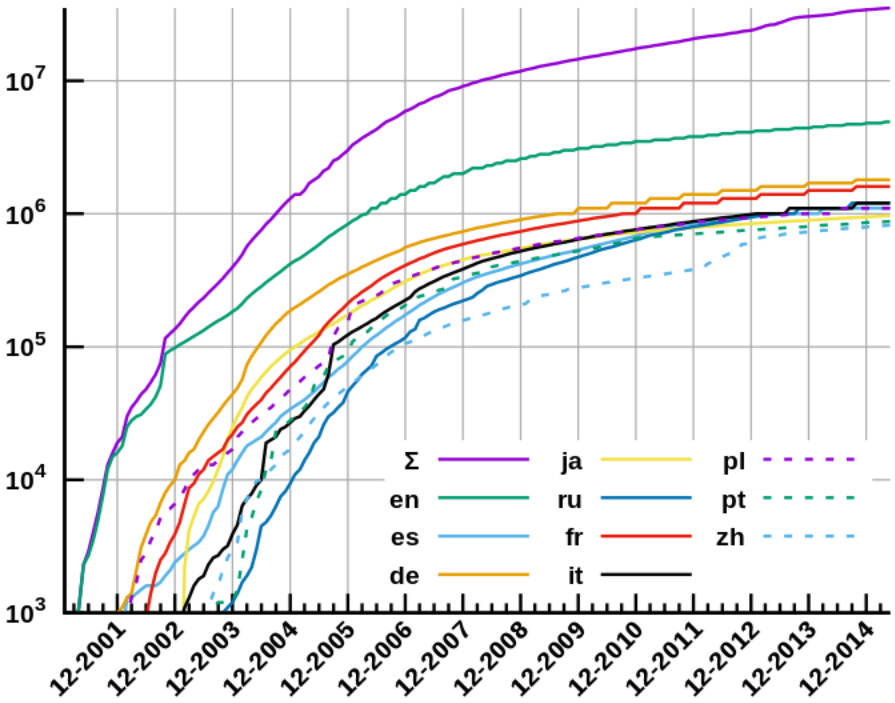
AGENDA

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

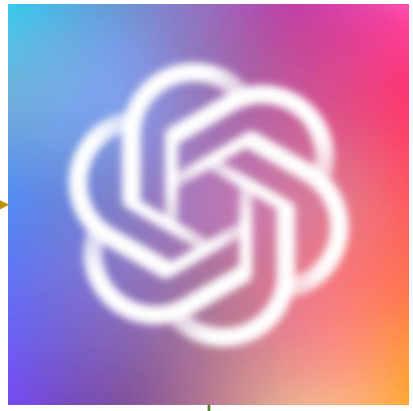
Radford et. al (2018)

- Motivation

Unlabeled Text Corpora

Labeled Text Corpora for a Specific Task

Generative pre-training of a language model



Discriminative fine-tuning

GPT: Generative Pre-Training of a Language Model

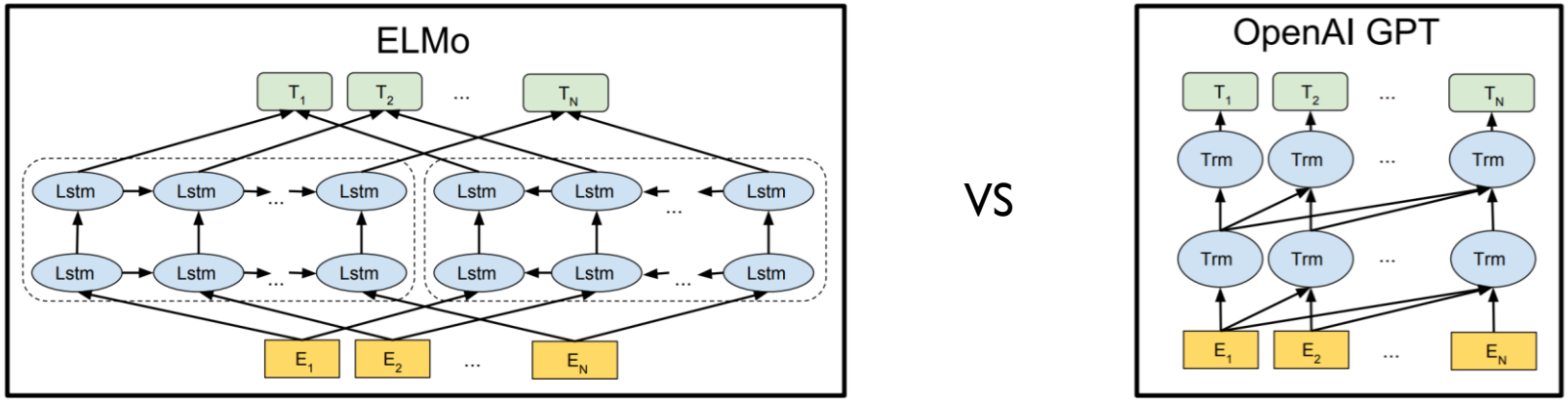
Radford et. al (2018)

- 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



  - ✓ 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)

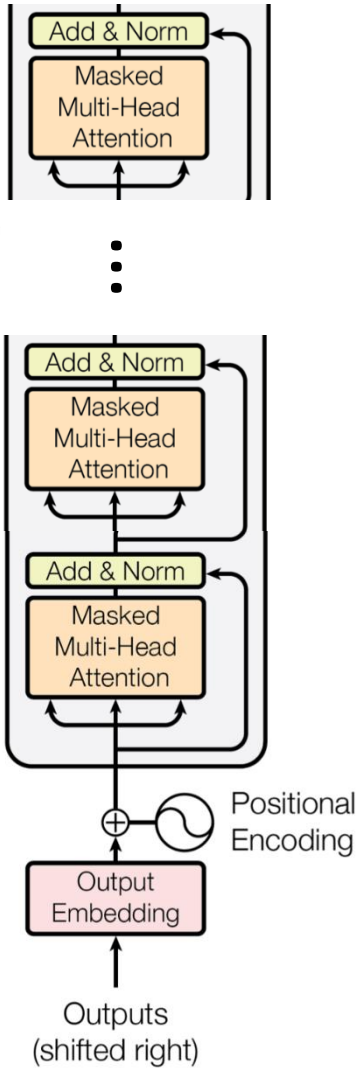
- GPT: Unsupervised pre-training
  - ✓ A multi-layer Transformer decoder is used for language model

$$h_0 = UW_e + W_p$$

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

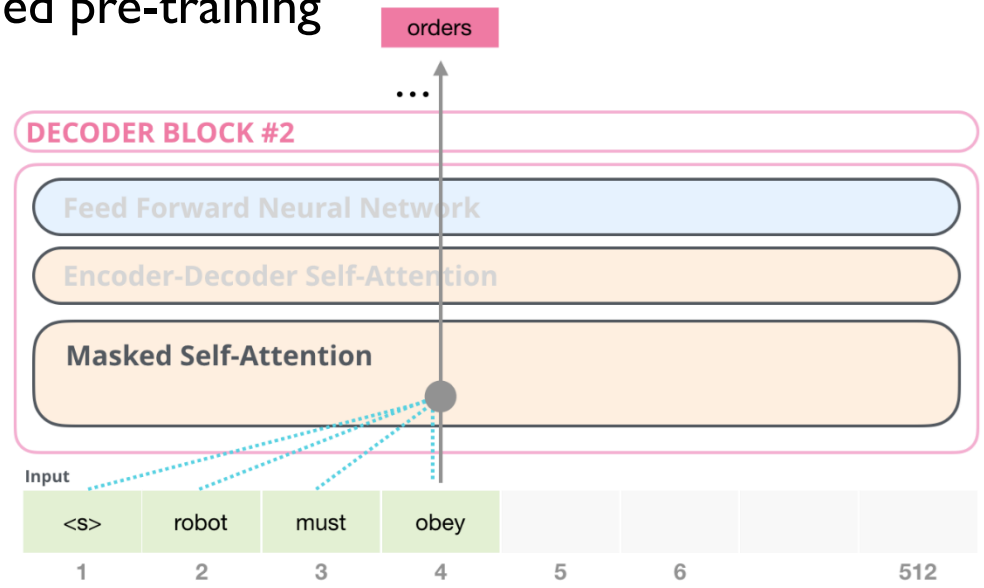
$$P(u) = softmax(h_n W_e^T)$$

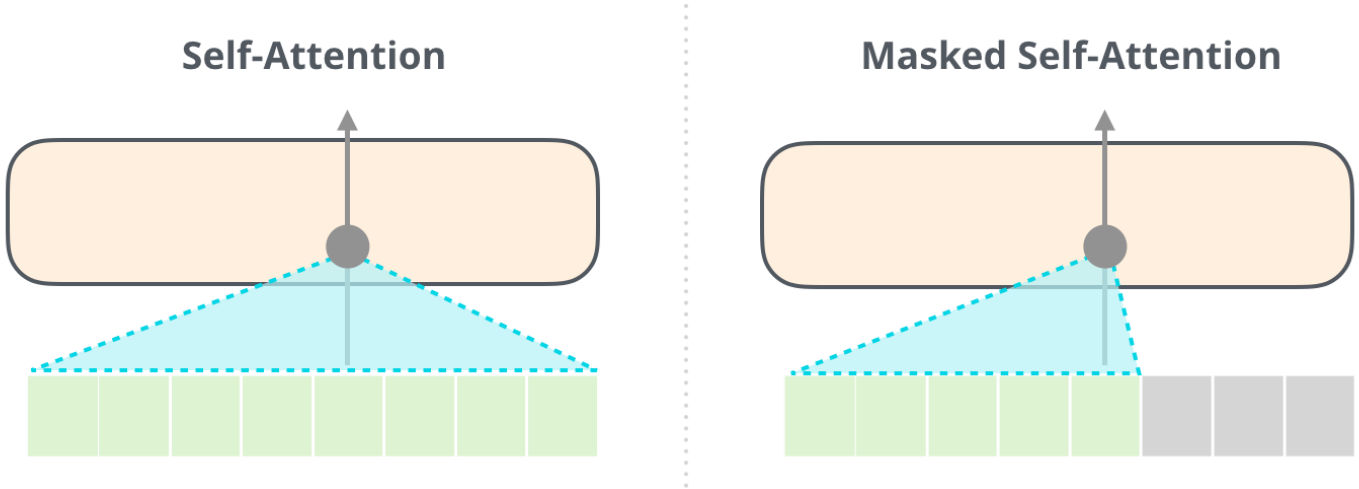
  - $U = (u_{-k}, ..., u_{-1})$: the context vector of tokens
  - n: the number of layers
  - W_e: token embedding matrix
  - W_p: position embedding matrix



GPT: Generative Pre-Training of a Language Model

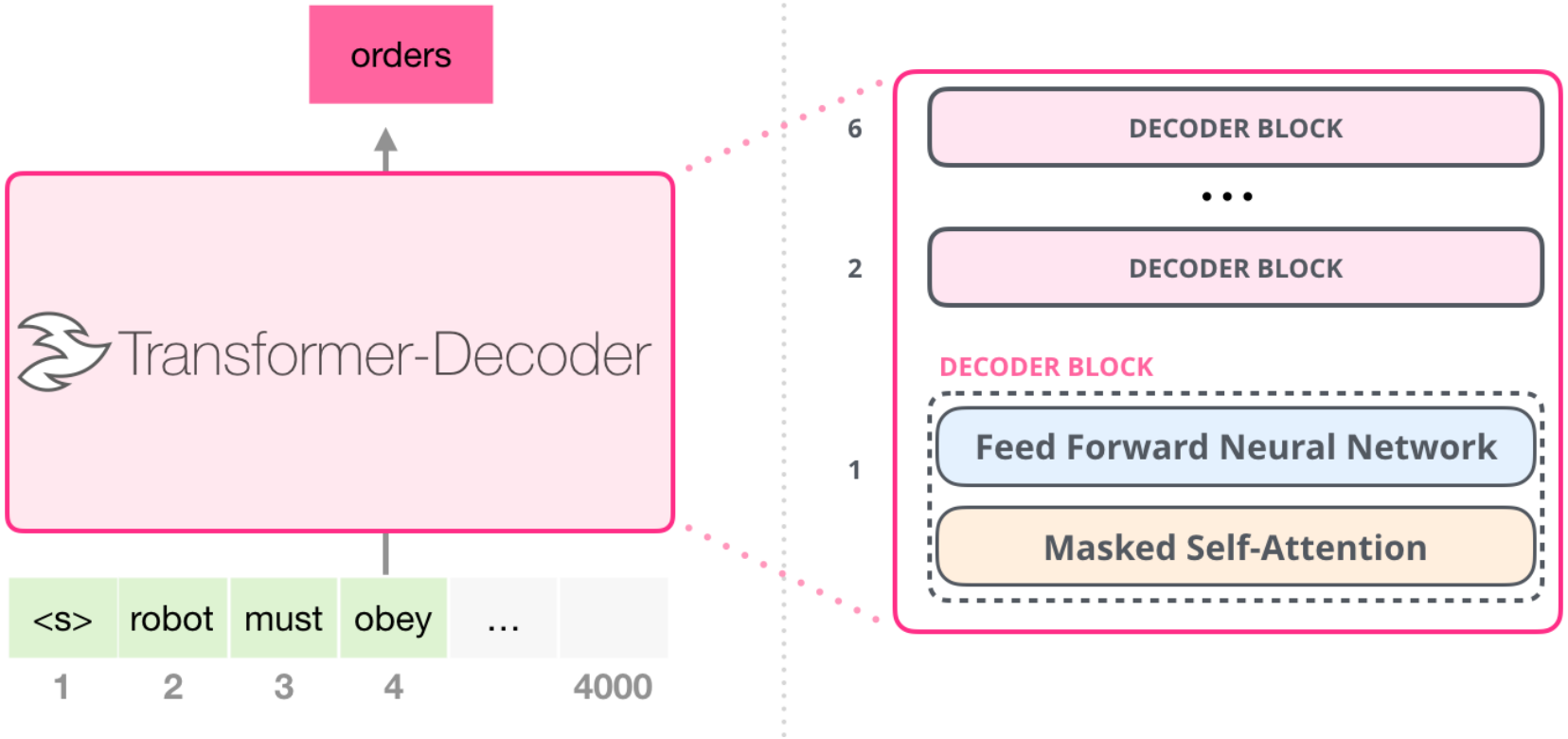
- GPT: Unsupervised pre-training





GPT: Generative Pre-Training of a Language Model

- 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)

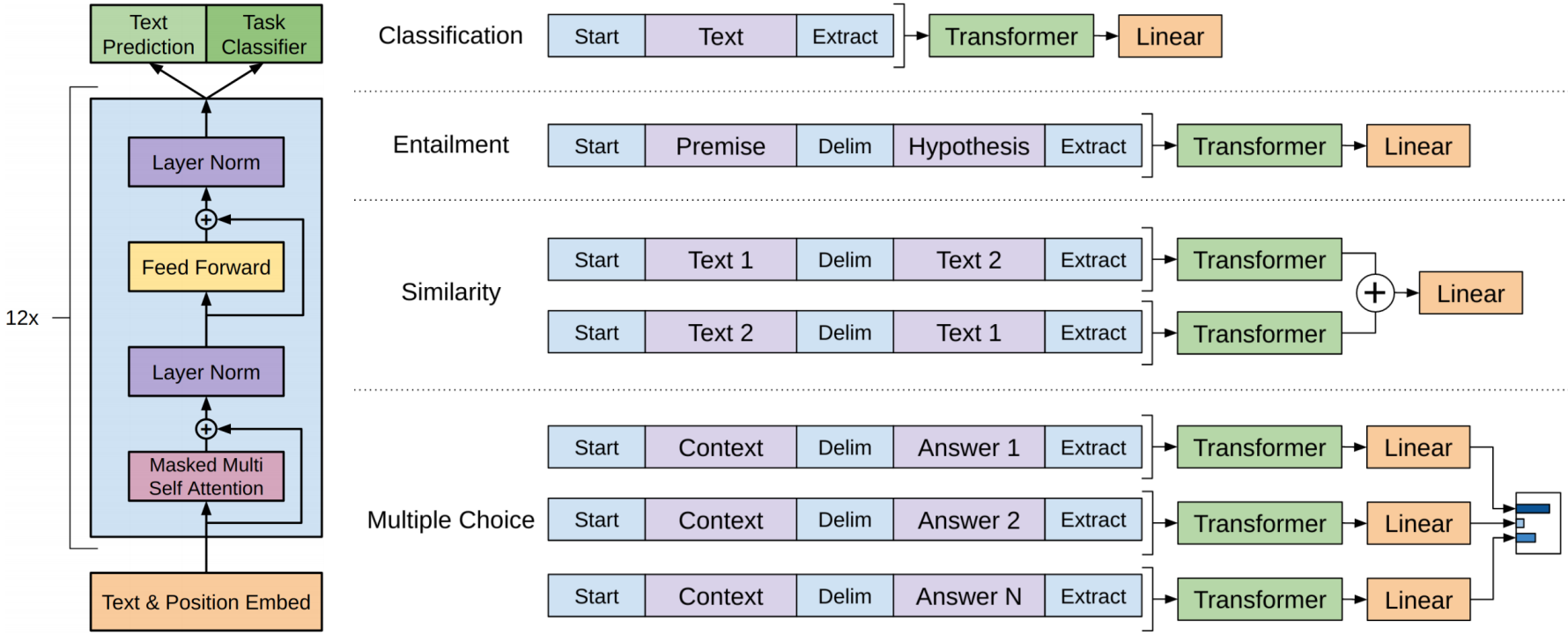
- GPT: Supervised fine-tuning
  - ✓ The authors additional found that including language modeling as an auxiliary objective to the fine-tuning helped learning by
    - Improving generalization of the supervised model
    - Accelerating convergence

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

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)

Service Unavailable

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

Happy benchmarking!

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)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	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]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

GPT: Generative Pre-Training of a Language Model

Radford et. al (2018)

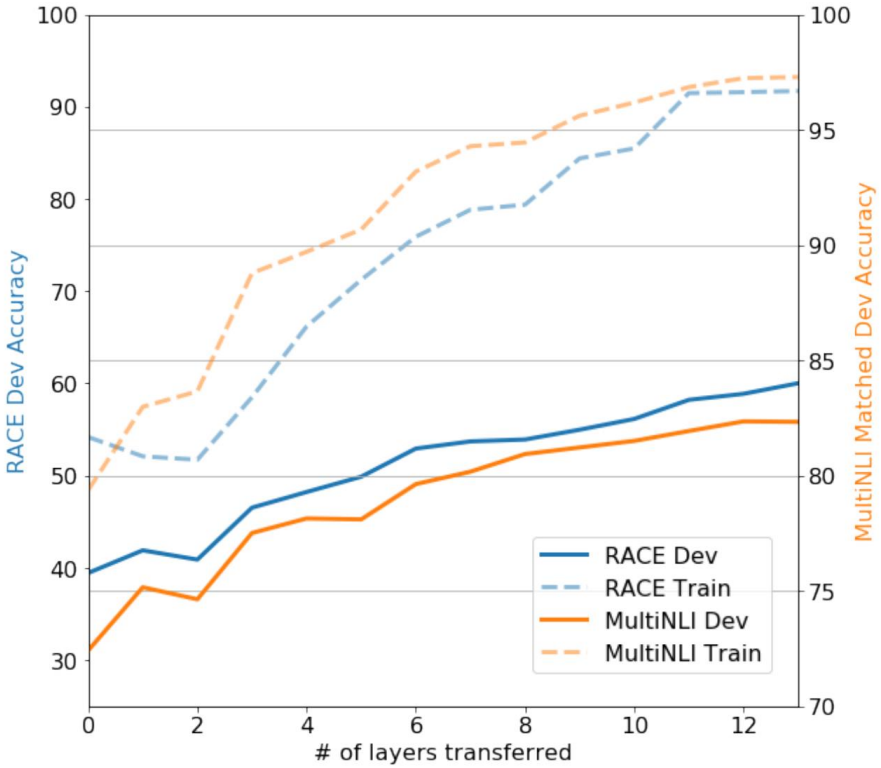
- Experiments
  - ✓ Semantic Similarity & Classification

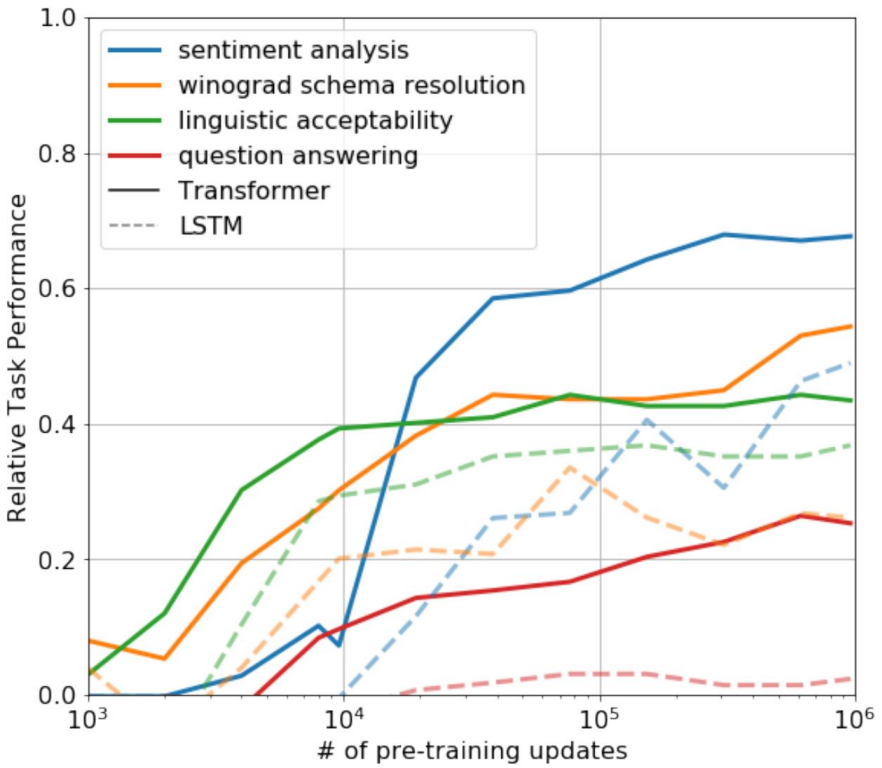
Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + 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

AGENDA

01 ELMo

02 GPT

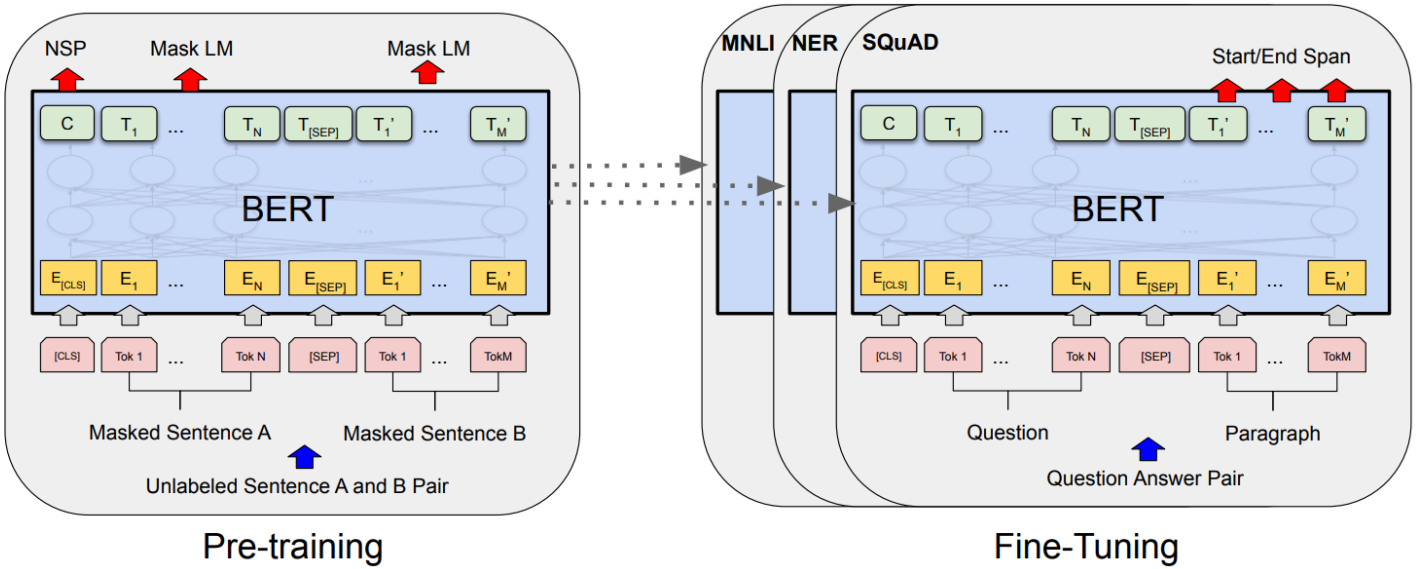
03 BERT

04 GPT-2

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- BERT
  - ✓ Designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers
    - Masked language model (MLM): bidirectional pre-training for language representations
    - Next sentence prediction (NSP)

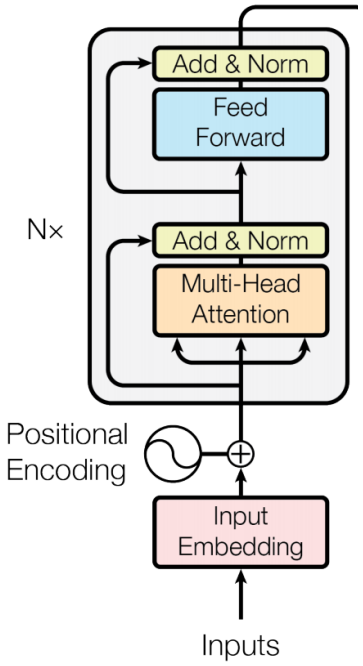


  - Pre-trained BERT model can be fine-tunes with just one additional output layer to create SOTA models for a wide range of NLP tasks (QA, NER, Sentiment Analysis, etc.)

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- BERT: Model Architecture
  - ✓ Multi-layer bidirectional Transformer encoder
    - L: number of layers (Transformer block)
    - H: hidden size
    - A: number of self attention heads
  - ✓ $BERT_{BASE}$
    - L = 12, H=768, A = 12
    - Total parameters = 110M
    - Same model size as OpenAI GPT
  - ✓ $BERT_{LARGE}$
    - L = 24, H=1,024, A = 16
    - Total parameters = 340M



BERT: Bidirectional Encoder Representations from Transformer

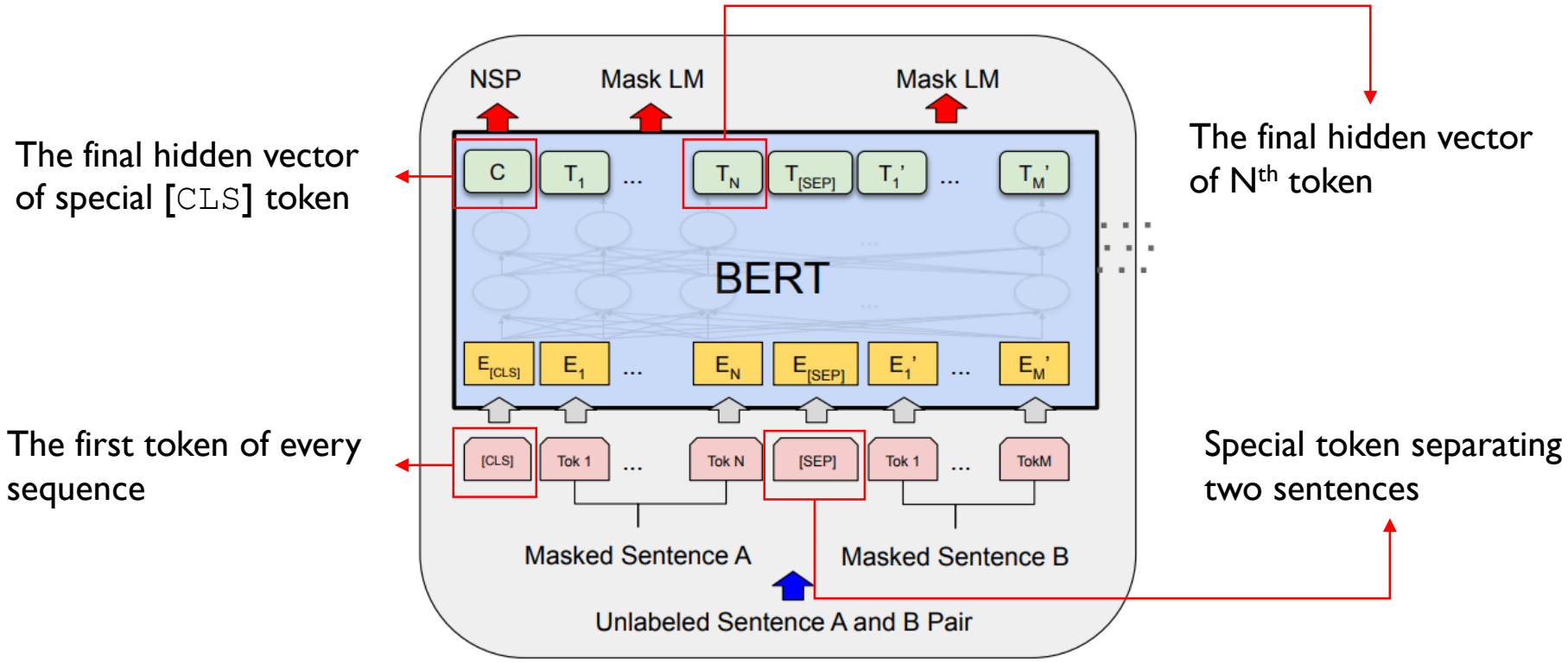
Devlin et. al (2018)

- BERT: Input/Output Representations
  - ✓ To make BERT handle a variety of down-stream tasks, the input representation is able to unambiguously represent both a single sentence and a pair of sentences (ex: Question-Answer)
    - **Sentence**: an arbitrary span of contiguous text, rather than an actual linguistic sentence
    - **Sequence**: the input token sequences to BERT, which may be a single sentence or two sentences packed together

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

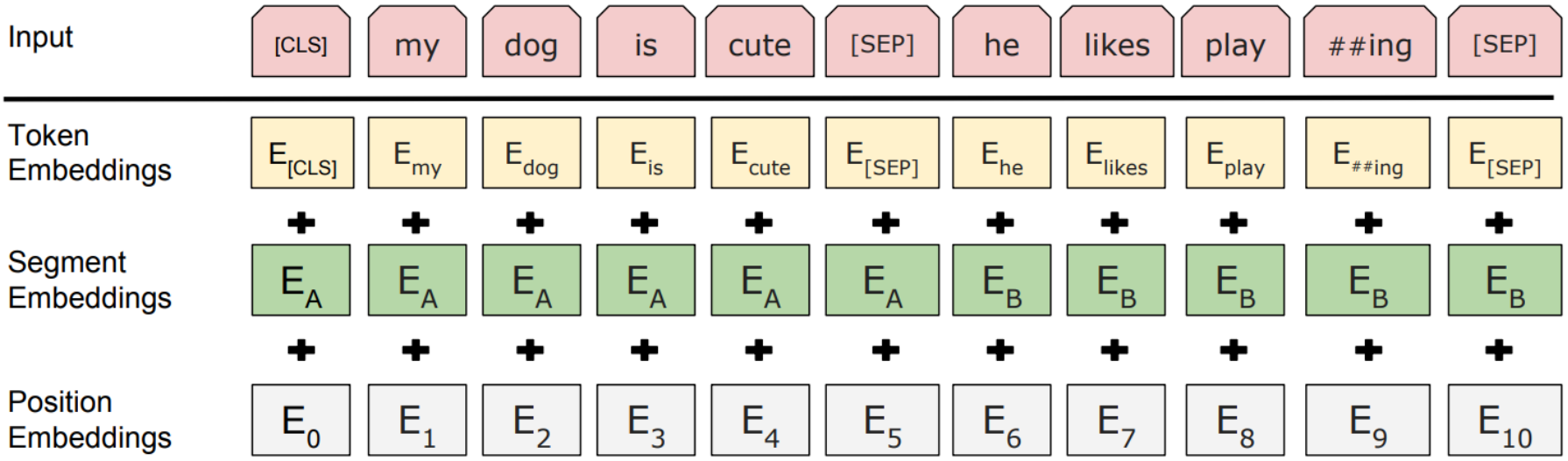
- BERT: Input/Output Representations



BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

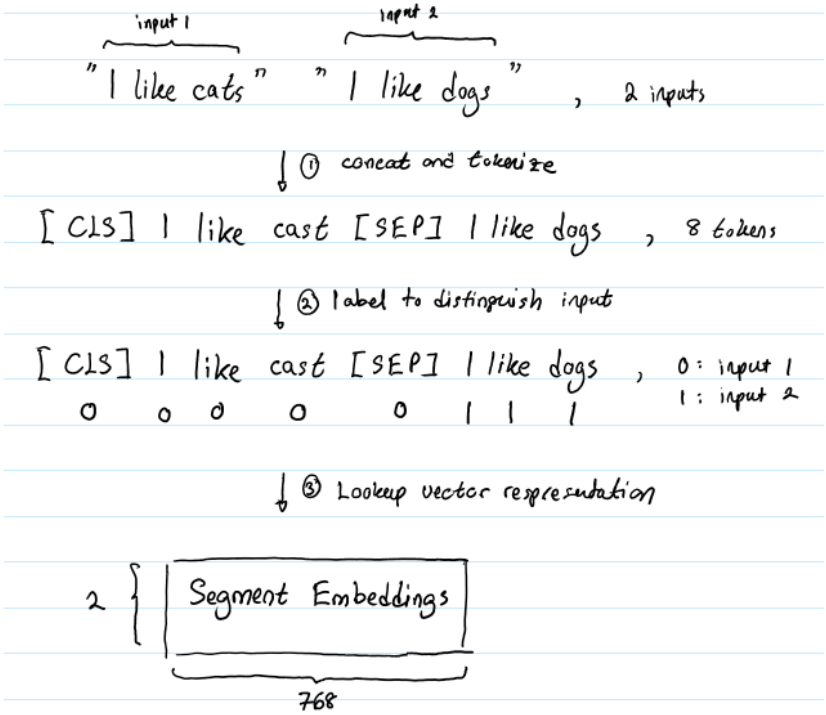
- BERT: Input/Output Representations
  - ✓ Input representation is the sum of
    - (1) Token embedding: WordPiece embeddings with a 30,000 token vocabulary
    - (2) Segment embedding
    - (3) Position embedding: same as in the Transformer



BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- BERT: Input/Output Representations
  - ✓ (2) Segment embedding



Layer-wise accounting:

Going through layers from top to bottom, we can see following:

1. Inputs — Token and segment do not have any trainable parameters, as expected.
2. Token embeddings parameters= 23040000 (H * T) — because each of 30k (T) tokens needs a representation in dimension 768 (H)
3. Segment Embeddings parameters = 1536 (2*H) because we need two vectors each of length (H). The vectors represent Segment A and Segment B respectively
4. Token embeddings and segment embeddings are added to Position Embedding. Parameters = 393216 (H*P). This is because it needs to generate P vectors, each of length H, for the tokens starting 1 to 512 (P). The position embeddings in BERT are trained and not fixed as in *Attention is all you need;* There's a dropout applied, and then Layer Normalization is done
5. Layer Normalization parameters = 1536 (2*H). Normalization has two parameters to learn — mean and standard deviation of each of the embedding position, hence 2*H
6. Encoder: MultiheadSelfAttention: MultiHeadAttention = 2362368

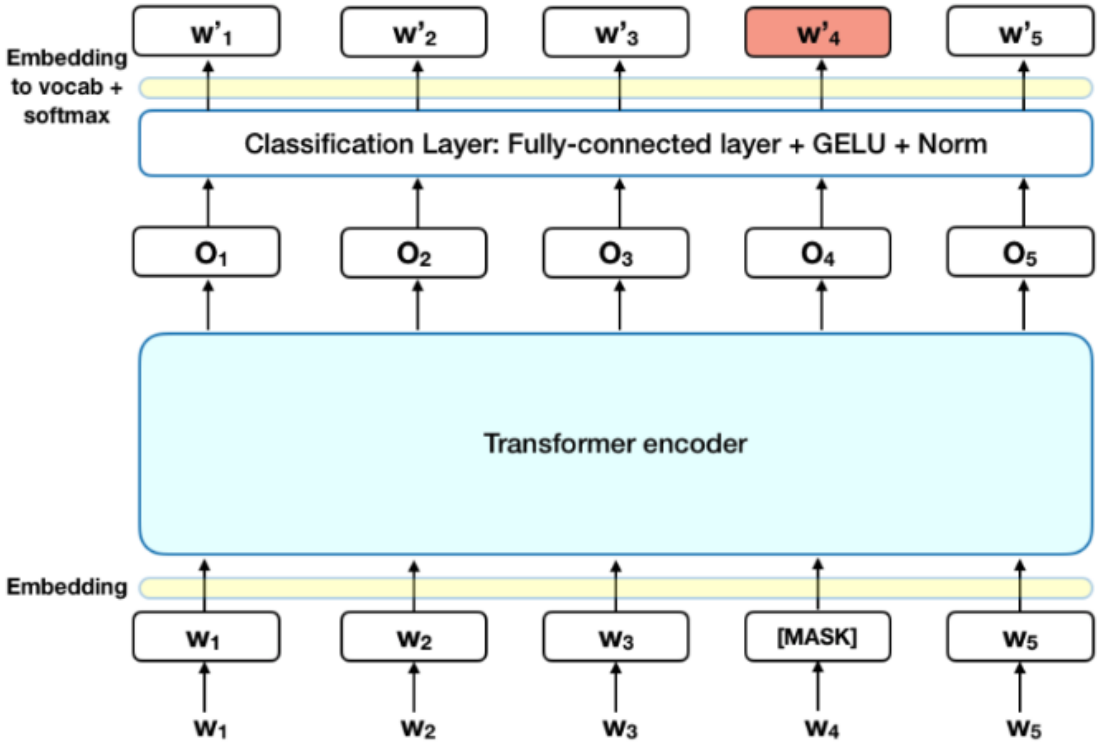
https://medium.com/@_init_/why-bert-has-3-embedding-layers-and-their-implementation-details-9c261108e28a

https://mc.ai/understanding-bert-architecture/

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 1: Masked Language Model (MLM)
    - 15% of each sequence are replaced with a [MASK] token
    - Predict the masked words rather tan reconstructing the entire input in denoising encoder



https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 1: Masked Language Model (MLM)
    - (Caution!) A mismatch occurs between pre-training and fine-tuning, since the [MASK] token does not appear during fine-tuning
    - (Solution) If the i-th token is chosen to be masked, it is replaced by the [MASK] token 80% of the time, a random toke 10% of the time, and unchanged 10% of the time
      - (80%) `my dog is hairy → my dog is [MASK]`
      - (10%) `my dog is hairy → my dog is apple`
      - (10%) `my dog is hairy → my dog is hairy`

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 1: Masked Language Model (MLM)
    - (Caution!) A mismatch occurs between pre-training and fine-tuning, since the [MASK] token does not appear during fine-tuning
    - (Solution) If the i-th token is chosen to be masked, it is replaced by the [MASK] token 80% of the time, a random toke 10% of the time, and unchanged 10% of the time

Masking Rates			Dev Set Results		
MASK	SAME	RND	MNLI	NER	
			Fine-tune	Fine-tune	Feature-based
80%	10%	10%	84.2	95.4	94.9
100%	0%	0%	84.3	94.9	94.0
80%	0%	20%	84.1	95.2	94.6
80%	20%	0%	84.4	95.2	94.7
0%	20%	80%	83.7	94.8	94.6
0%	0%	100%	83.6	94.9	94.6

BERT: Bidirectional Encoder Representations from Transformer

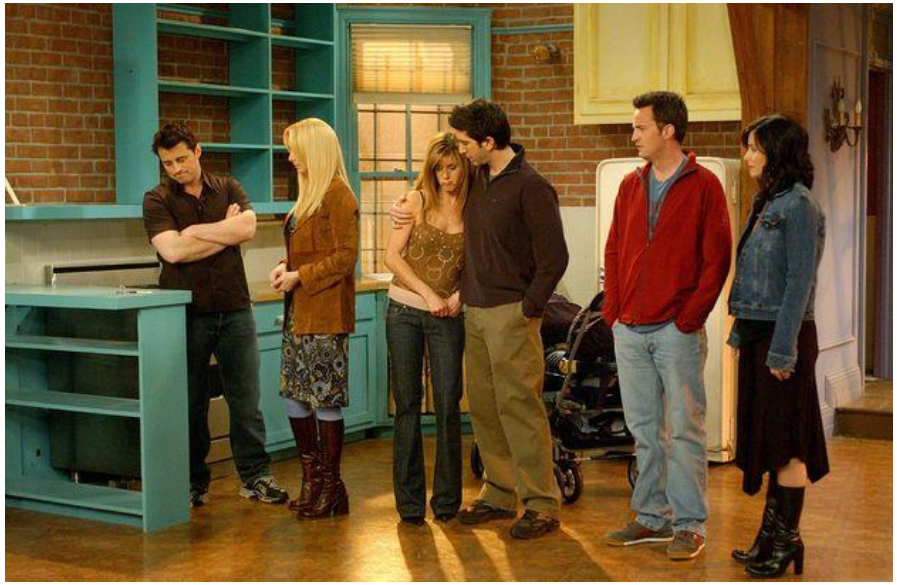
Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 2: Next Sentence Prediction (NSP)
    - Many important downstream tasks such as QA and NLI are based on understanding the relationship between two sentences, which is not directly captured by language modeling
    - A Binarized next sentence prediction task that can be trivially generated from any monolingual corpus is trained
      - 50% of the time B is the actual next sentence that follows A (IsNext)
      - 50% of the time it is a random sentence from the corpus (NotNext)
      - C is used for next sentence prediction
    - Despite its simplicity, pre-training towards this task is very beneficial both QA and NLI

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 2: Next Sentence Prediction (NSP)



Monica: This is harder than I thought it would be.

Chandler: Oh, it is gonna be okay.

Rachel: Do you guys have to go to the new house right away, or do you have some time?

Monica: We got some time.

Rachel: Okay, should we get some coffee?

Chandler: Sure. Where?

https://fangj.github.io/friends/season/1017-1018.html

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 2: Next Sentence Prediction (NSP)

Monica: This is harder than I thought it would be.

Chandler: Oh, it is gonna be okay.

Rachel: Do you guys have to go to the new house right away, or do you have some time?

Monica: We got some time.

Rachel: Okay, should we get some coffee?

Chandler: Sure. Where?

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 2: Next Sentence Prediction (NSP)

Monica: This is harder than I thought it would be.

Chandler: Oh, it is gonna be okay.

Rachel: Do you guys have to go to the new house right away, or do you have some time?

Monica: We got some time.

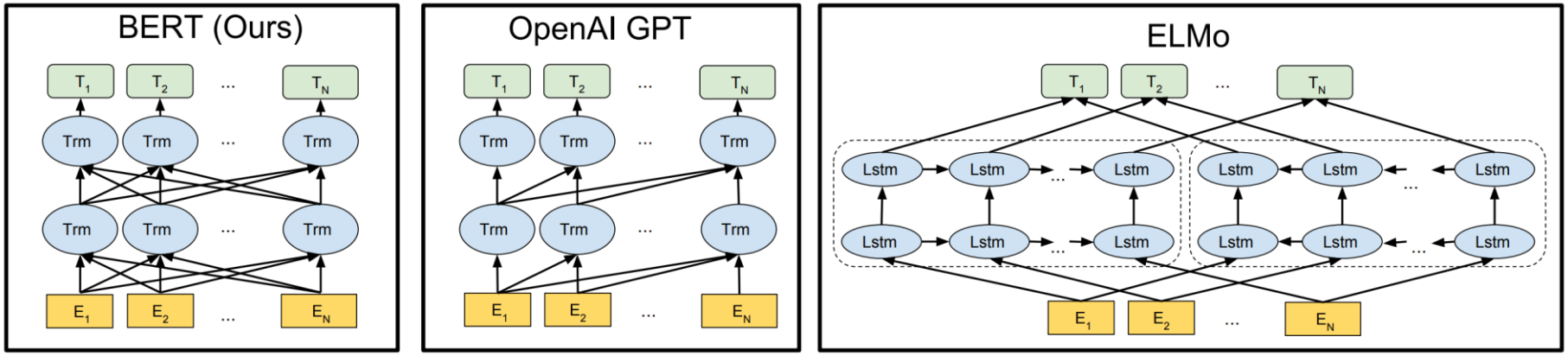
Rachel: Okay, should we get some coffee?

Chandler: Sure. Where?

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

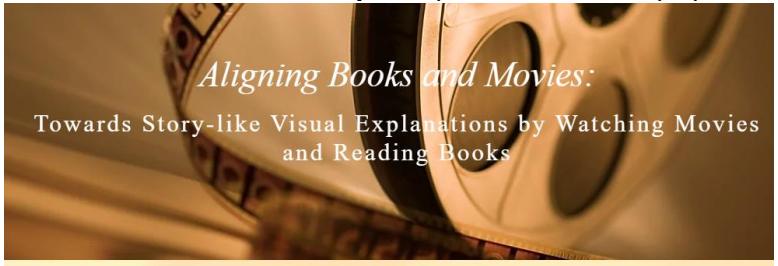
- Differences in pre-training model architectures



BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Datasets for pre-training
    - BooksCorpus (800M words) (Zhu et al., 2015)



Abstract

Books are a rich source of both fine-grained information, how a character, an object or a scene looks like, as well as high-level semantics, what someone is thinking, feeling and how these states evolve through a story. This work aims to align books to their movie releases in order to provide rich descriptive explanations for visual content that go semantically far beyond the captions available in current datasets. To align movies and books we propose a neural sentence embedding that is trained in an unsupervised way from a large corpus of books, as well as a video-text neural embedding for computing similarities between movie clips and sentences in the book. We propose a context-aware CNN to combine information from multiple sources. We demonstrate good quantitative performance for movie/book alignment and show several qualitative examples that showcase the diversity of tasks our model can be used for.

Paper

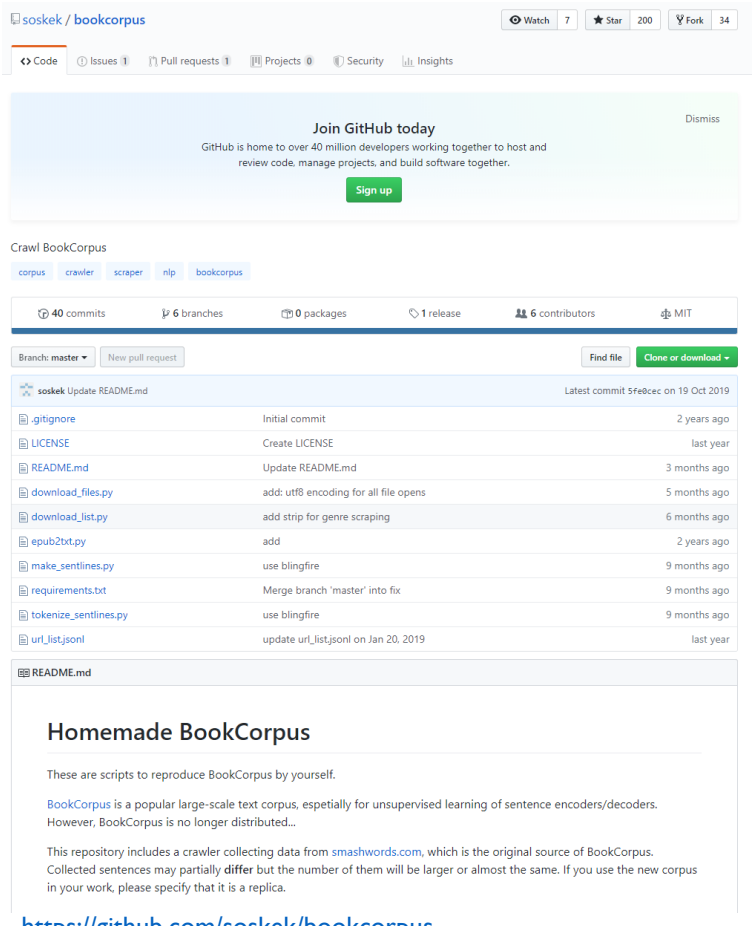
Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books

Yukun Zhu*, Ryan Kiros*, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, Sanja Fidler
Arxiv, June 2015
* denotes equal contribution

Data

MovieBook dataset: We no longer host this dataset. You can find movies and corresponding books on Amazon.

BookCorpus: Please visit smashwords.com to collect your own version of BookCorpus.

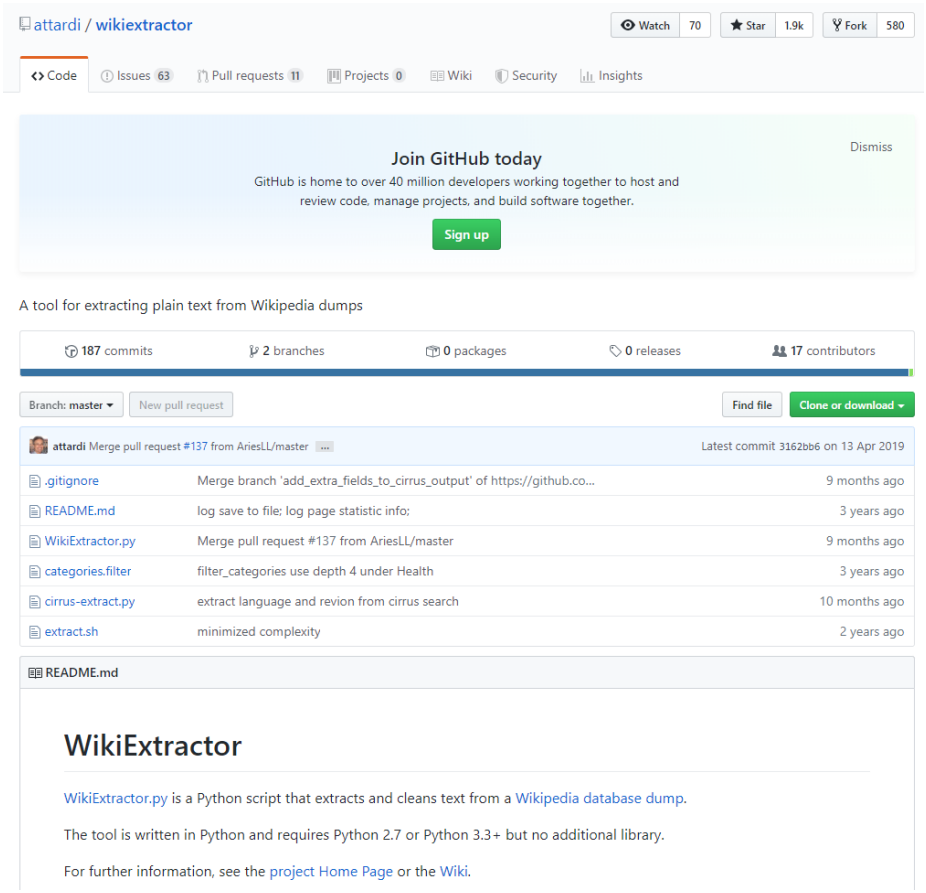


https://github.com/soskek/bookcorpus

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Datasets for pre-training
    - English Wikipedia (2,500M words)



https://github.com/attardi/wikiextractor

BERT: Bidirectional Encoder Representations from Transformer

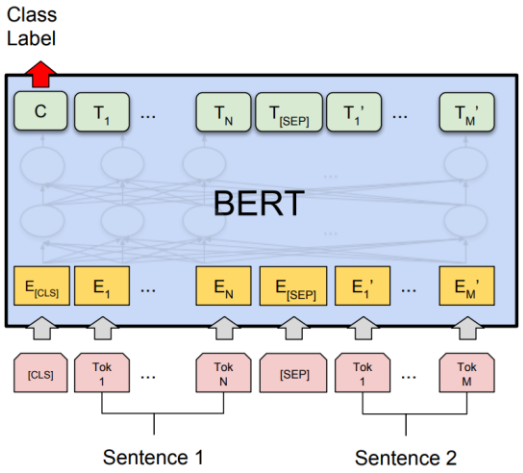
Devlin et. al (2018)

- Pre-training BERT
  - ✓ Hyper-parameter settings
    - Maximum token length: 512
    - Batch size: 256
    - Adam with learning rate of 1e-4, beta1 = 0.9 beta2 = 0.999
    - L2 weight decay of 0.01
    - Learning rate warmup over the first 10,000 steps, linear decay of the learning rate
    - Dropout probability of 0.1 on all layers
    - GeLU activation function rather than standard ReLU
    - $\text{BERT}_{\text{BASE}}$ took 4 days with 16 TPUs and $\text{BERT}_{\text{LARGE}}$ took 4 days with 64 TPUs
    - Pre-train the model with sequence length of 128 for 90% of the steps
    - The rest 10% of the steps are trained with sequence length of 512

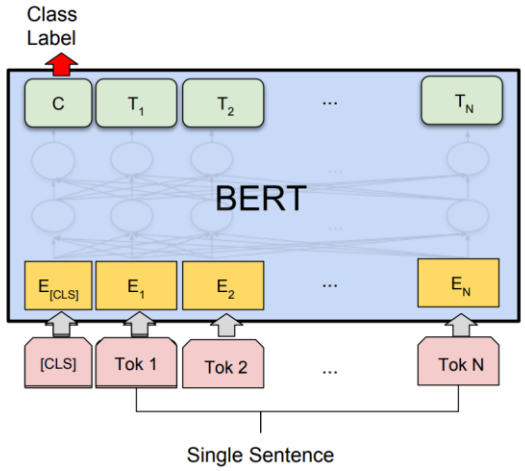
BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

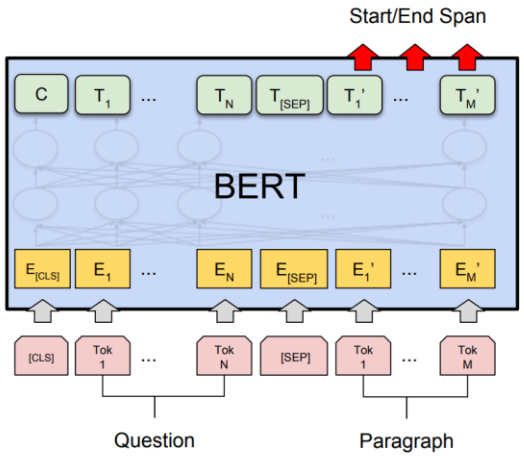
- Fine-tuning BERT



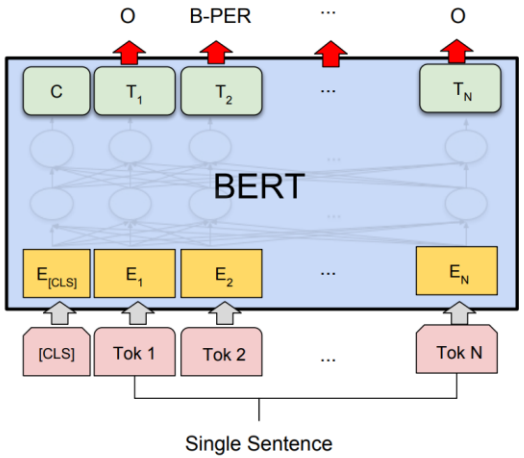
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Experiments
  - ✓ A collection of diverse NLU tasks

Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
1	ERNIE Team - Baidu	ERNIE		90.2	72.2	97.5	93.0/90.7	92.9/92.5	75.2/90.8	91.2	90.6	98.0	90.9	94.5	49.4
2	王玮	ALICE v2 large ensemble (Alibaba DAMO NLP)		90.1	73.2	97.1	93.9/91.9	93.0/92.5	74.8/91.0	90.8	90.6	99.2	87.4	94.5	48.7
3	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5	50.2
4	T5 Team - Google	T5		89.7	70.8	97.1	91.9/89.2	92.5/92.1	74.6/90.4	92.0	91.7	96.7	92.5	93.2	53.1
5	XLNet Team	XLNet (ensemble)		89.5	70.2	97.1	92.9/90.5	93.0/92.6	74.7/90.4	90.9	90.9	99.0	88.5	92.5	48.4
6	ALBERT-Team Google Language	ALBERT (Ensemble)		89.4	69.1	97.1	93.4/91.2	92.5/92.0	74.2/90.5	91.3	91.0	99.2	89.2	91.8	50.2
7	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)		88.8	68.0	96.8	93.1/90.8	92.4/92.2	74.8/90.3	91.1	90.7	98.8	88.7	89.0	50.1
8	Facebook AI	RoBERTa		88.5	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	98.9	88.2	89.0	48.7
9	Junjie Yang	HIRE-RoBERTa		88.3	68.6	97.1	93.0/90.7	92.4/92.0	74.3/90.2	90.7	90.4	95.5	87.9	89.0	49.3
10	Microsoft D365 AI & MSR AI	MT-DNN-ensemble		87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	42.8

https://gluebenchmark.com/leaderboard

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**

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Experiments
  - ✓ Ablation study 1: Effect of Pre-training Tasks

Tasks	Dev Set MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$\text{BERT}_{\text{BASE}}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

  - ✓ Ablation study 2: Effect of Model Size

Hyperparams				Dev Set Accuracy		
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Experiments
  - ✓ Ablation study 3: Feature-based Approach with BERT
    - CoNLL-2003 NER task

System	Dev F1	Test F1
ELMo (Peters et al., 2018a)	95.7	92.2
CVT (Clark et al., 2018)	-	92.6
CSE (Akbik et al., 2018)	-	**93.1**
Fine-tuning approach		
$\text{BERT}_{\text{LARGE}}$	96.6	92.8
$\text{BERT}_{\text{BASE}}$	96.4	92.4
Feature-based approach ($\text{BERT}_{\text{BASE}}$)		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

AGENDA

01 ELMo

02 GPT

03 BERT

04 GPT-2

GPT-2: Language Models are Unsupervised Multitask Learners

Radford et. al (2019)

- Feb. 14, 2019

SYSTEM PROMPT (HUMAN-WRITTEN)

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

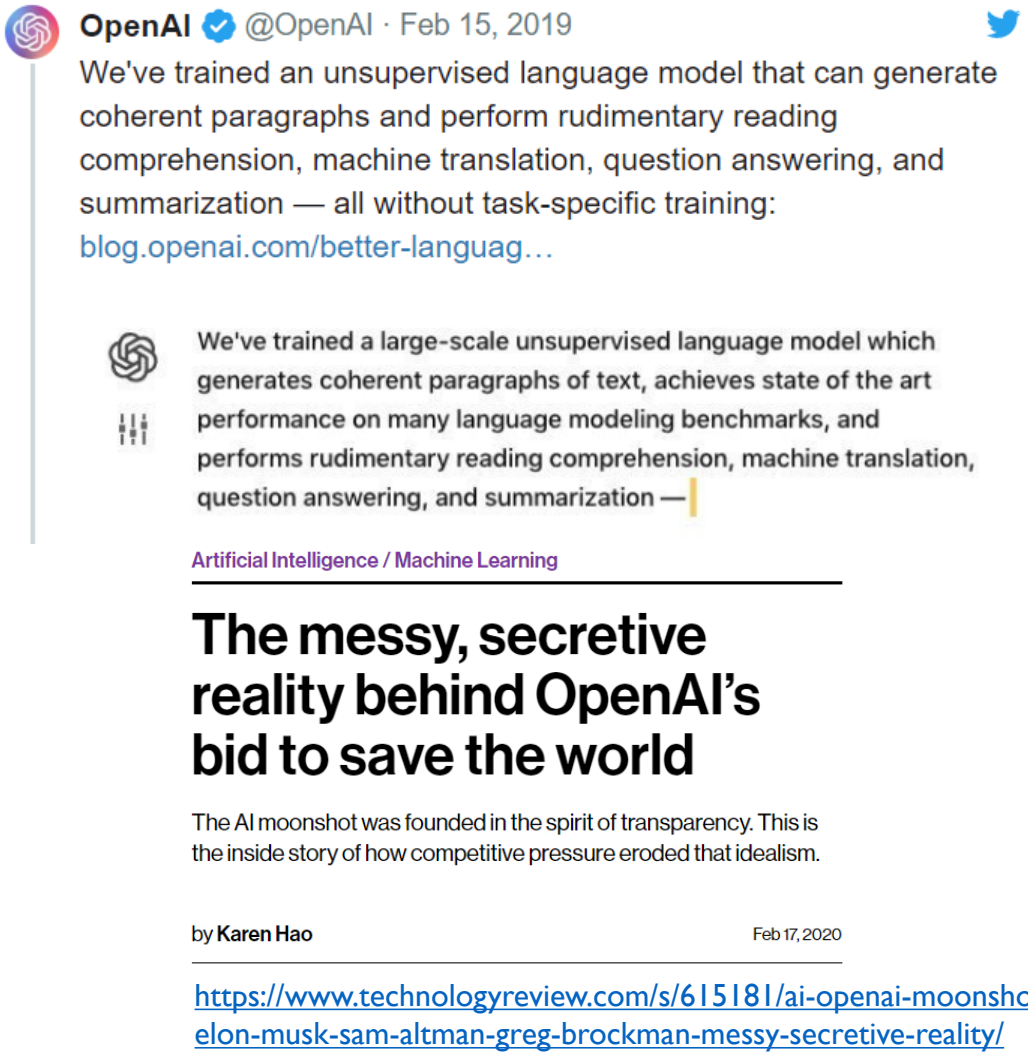
While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

GPT-2: Language Models are Unsupervised Multitask Learners

Radford et. al (2019)

- Debates on GPT model



OpenAI @OpenAI · Feb 15, 2019

We've trained an unsupervised language model that can generate coherent paragraphs and perform rudimentary reading comprehension, machine translation, question answering, and summarization — all without task-specific training: blog.openai.com/better-languag…

We've trained a large-scale unsupervised language model which generates coherent paragraphs of text, achieves state of the art performance on many language modeling benchmarks, and performs rudimentary reading comprehension, machine translation, question answering, and summarization —

Artificial Intelligence / Machine Learning

The messy, secretive reality behind OpenAI's bid to save the world

The AI moonshot was founded in the spirit of transparency. This is the inside story of how competitive pressure eroded that idealism.

by **Karen Hao** Feb 17, 2020

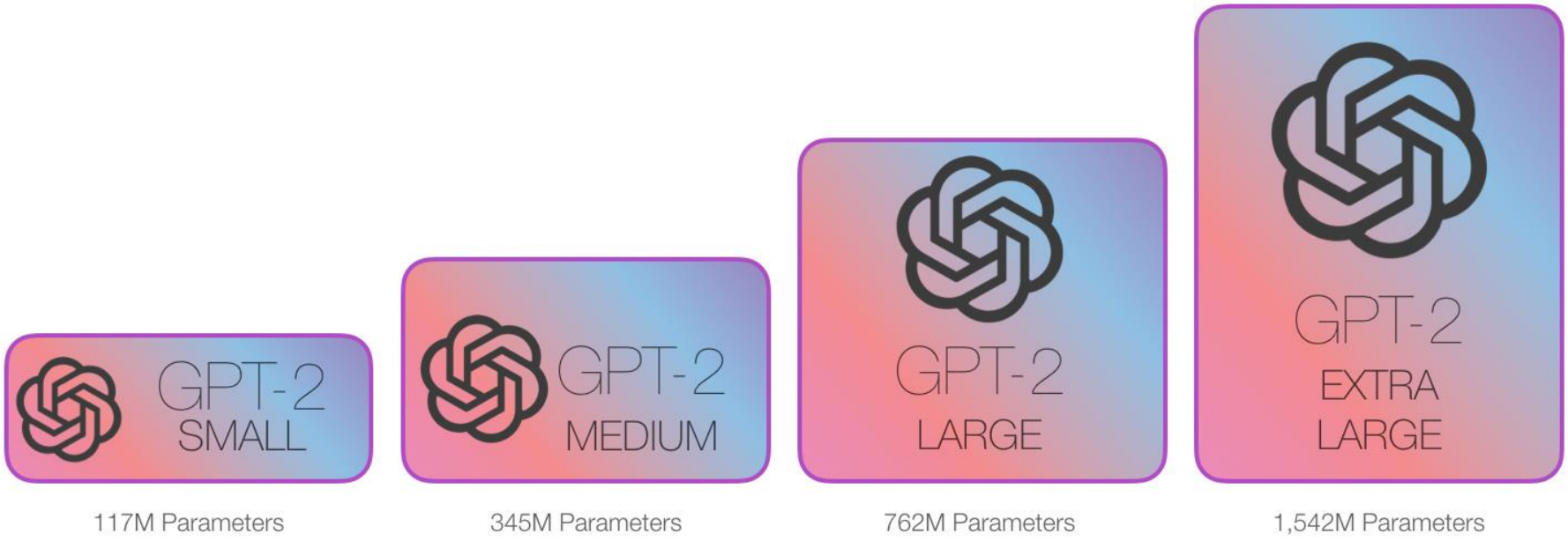
https://www.technologyreview.com/s/615181/ai-openai-moonshot-elon-musk-sam-altman-greg-brockman-messy-secretive-reality/

The full version of GPT-2 is now publicly available, following nearly nine months of heated debates and some smaller model releases. The large-scale unsupervised language model was kept under lock and key for this long as it was deemed too dangerous—a controversial decision that led to backlash from the open source community.

GPT-2: Language Models are Unsupervised Multitask Learners

Radford et. al (2019)

- GPT-2 is not a particularly novel architecture – it is very similar to the decoder-only transformer
- However, it was trained on a massive dataset (40GB, WebText)



https://demo.allennlp.org/next-token-lm?text=AllenNLP%20is

GPT-2: Language Models are Unsupervised Multitask Learners

Radford et. al (2019)

- GPT-2 Example

Every year, OpenAI's employees vote on when they believe artificial general intelligence, or AGI, will finally arrive. It's mostly seen as a fun way to bond, and their estimates differ widely. But in a field that still debates whether human-like autonomous systems are even possible, half the lab bets it is likely to happen within 15 years.

Language Modeling

This demonstration uses the public 345M parameter OpenAI GPT-2 language model to generate sentences.

Enter some initial text and the model will generate the most likely next words. You can click on one of those words to choose it and continue or just keep typing. Click the left arrow at the bottom to undo your last choice.

Sentence:

Every year, OpenAI's employees vote on when they believe artificial intelligence will finally

Predictions:

25.8% be
9.3% become
3.7% make
3.1% reach
3.1% win
← Undo

GPT-2: Language Models are Unsupervised Multitask Learners

Radford et. al (2019)

- 오늘의 교훈

SYSTEM PROMPT (HUMAN-WRITTEN)

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

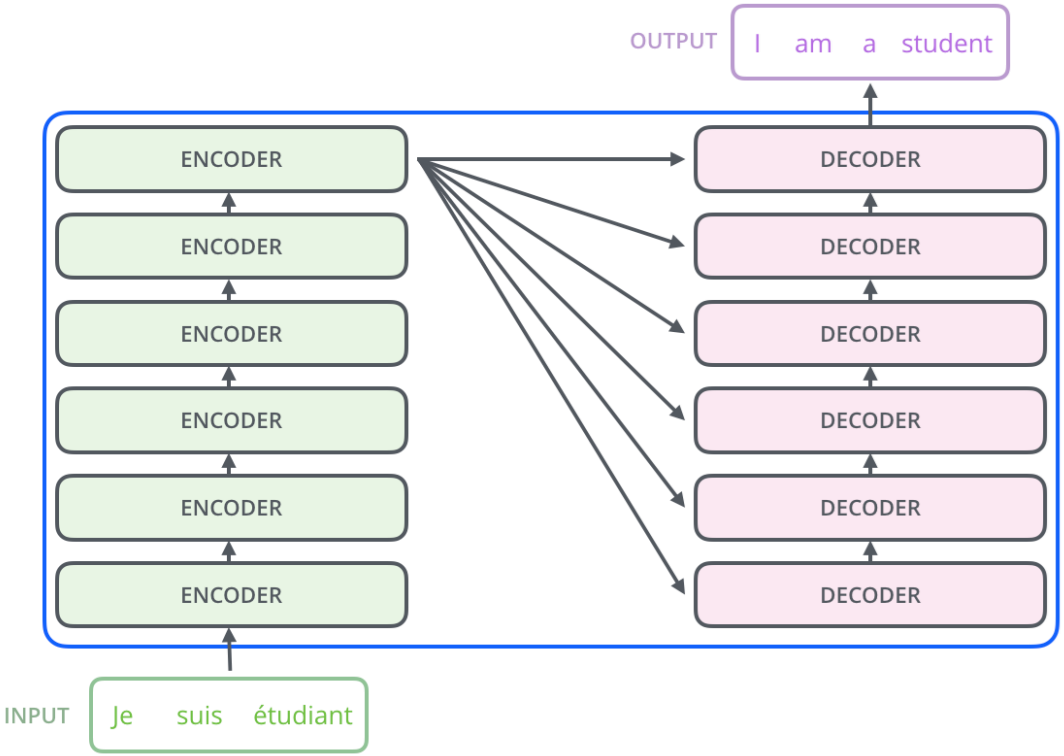
While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

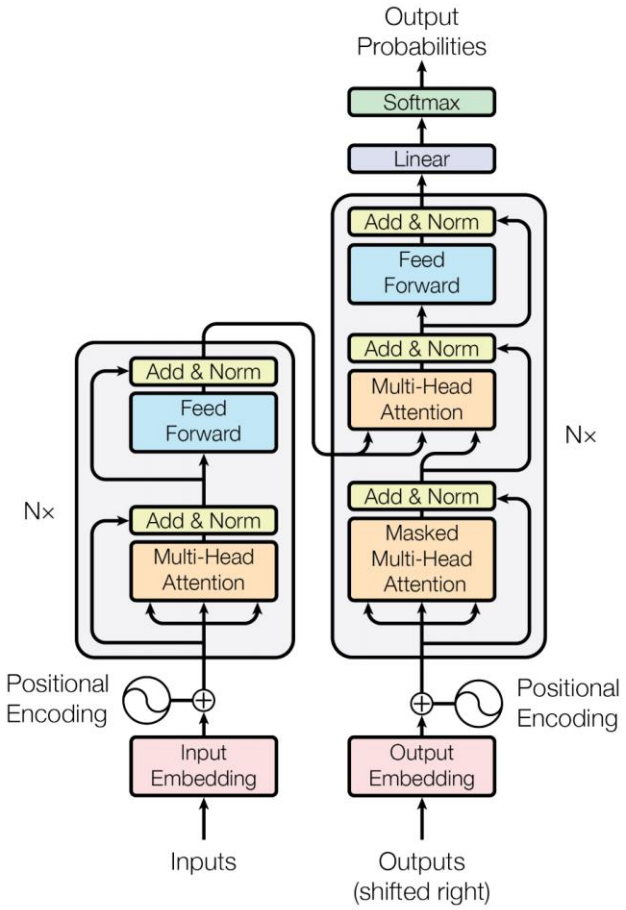
However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

GPT-2: Language Models are Unsupervised Multitask Learners

Alammar (GPT-2)

- Transformer revisited

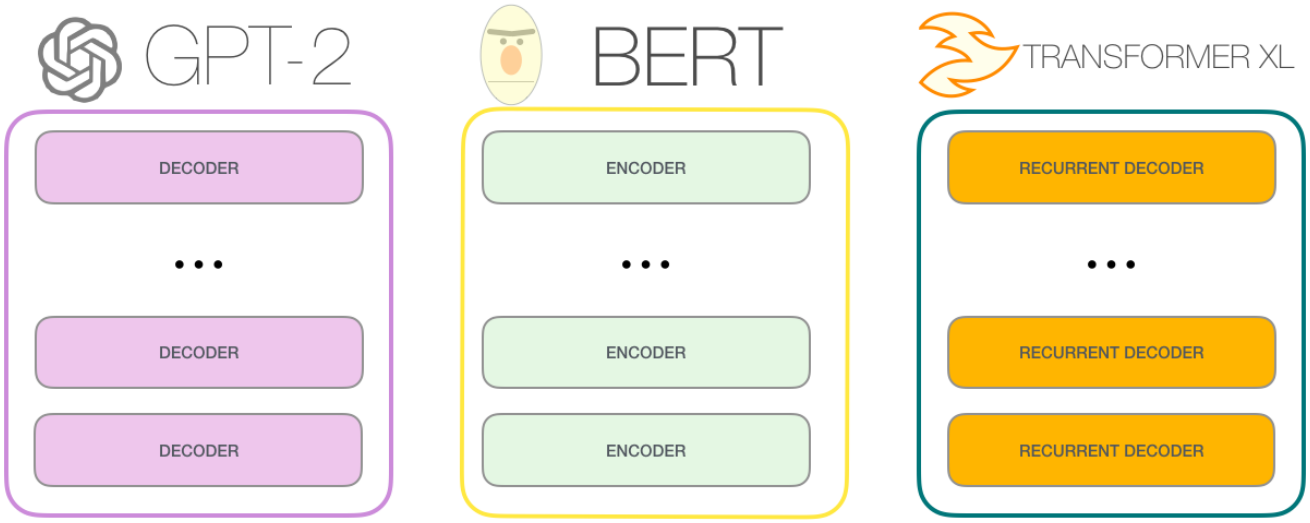




GPT-2: Language Models are Unsupervised Multitask Learners

Alammar (GPT-2)

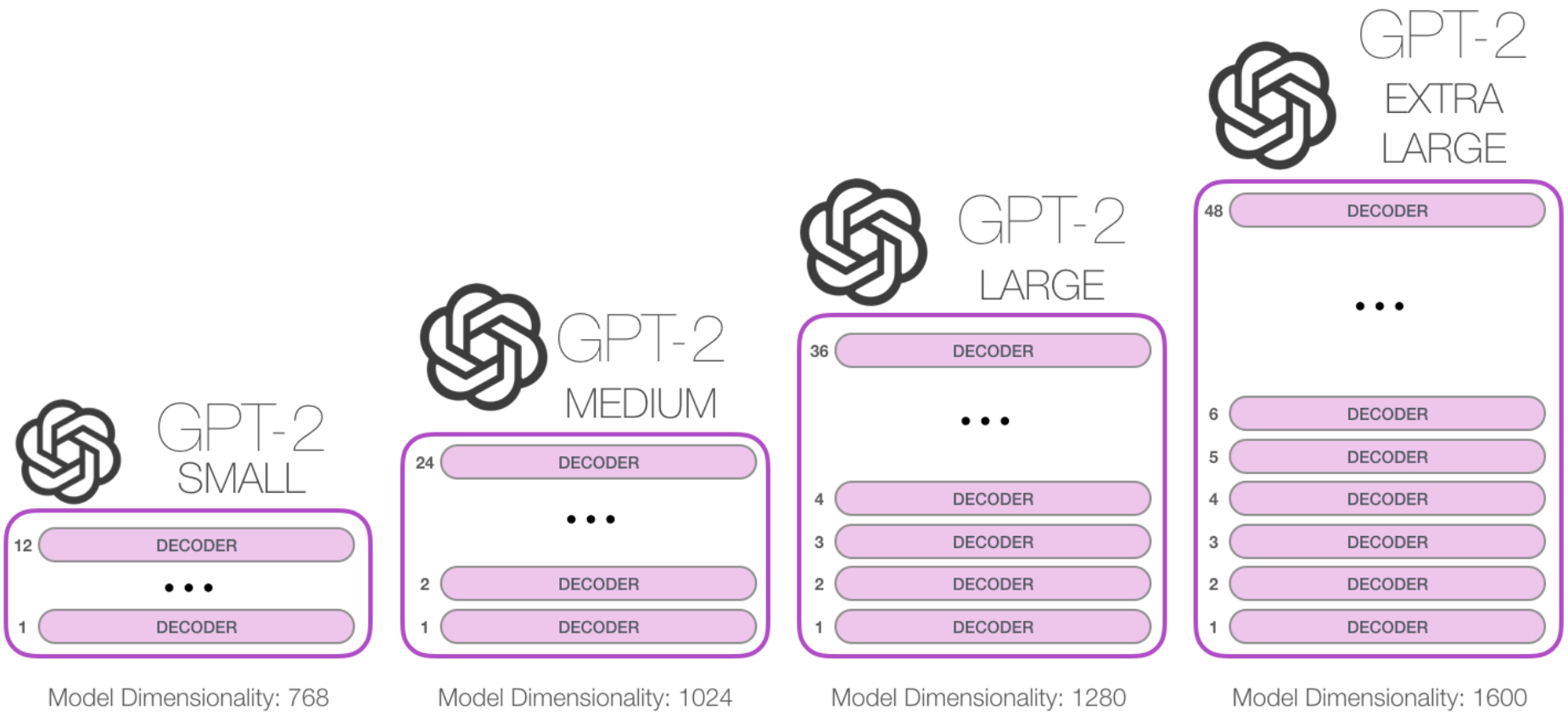
- Transformer revisited
  - ✓ A lot of the subsequent research work saw the architecture shed either the encoder or decoder, and use just one stack of transformer blocks
  - ✓ Stacking them up as high as practically possible, feeding them massive amounts of training text, and throwing vast amounts of compute at them
  - ✓ Hundreds of thousands of dollars to train some of these language models, likely millions in the case of AlphaStar



GPT-2: Language Models are Unsupervised Multitask Learners

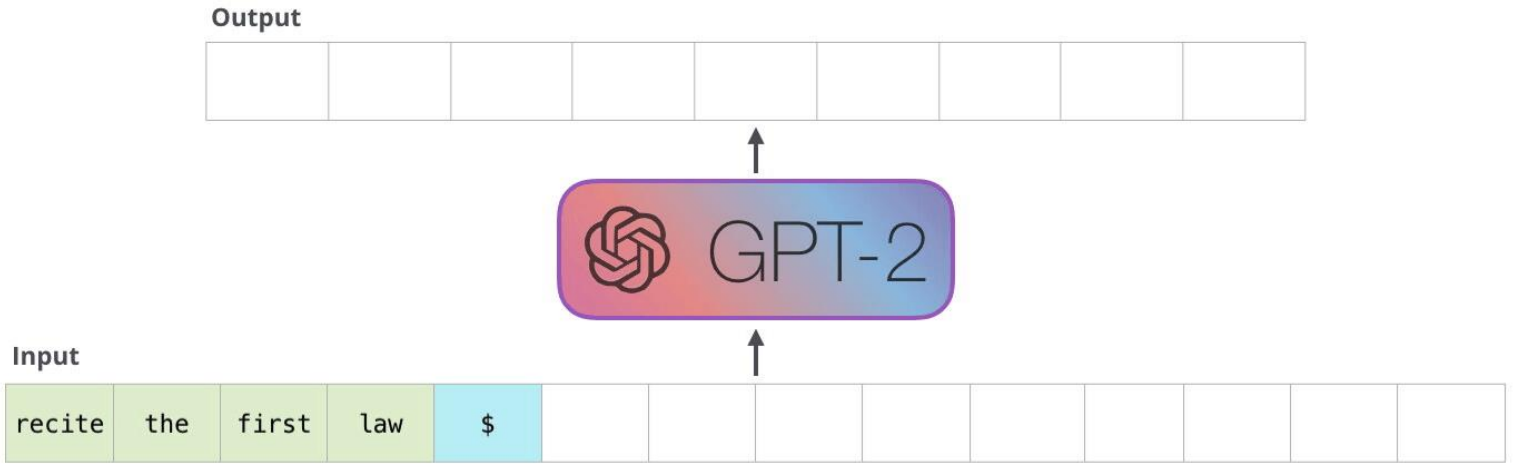
Alammar (GPT-2)

- How high can we stack up the blocks?
  - ✓ Main distinguishing factor of different GPT-2 models



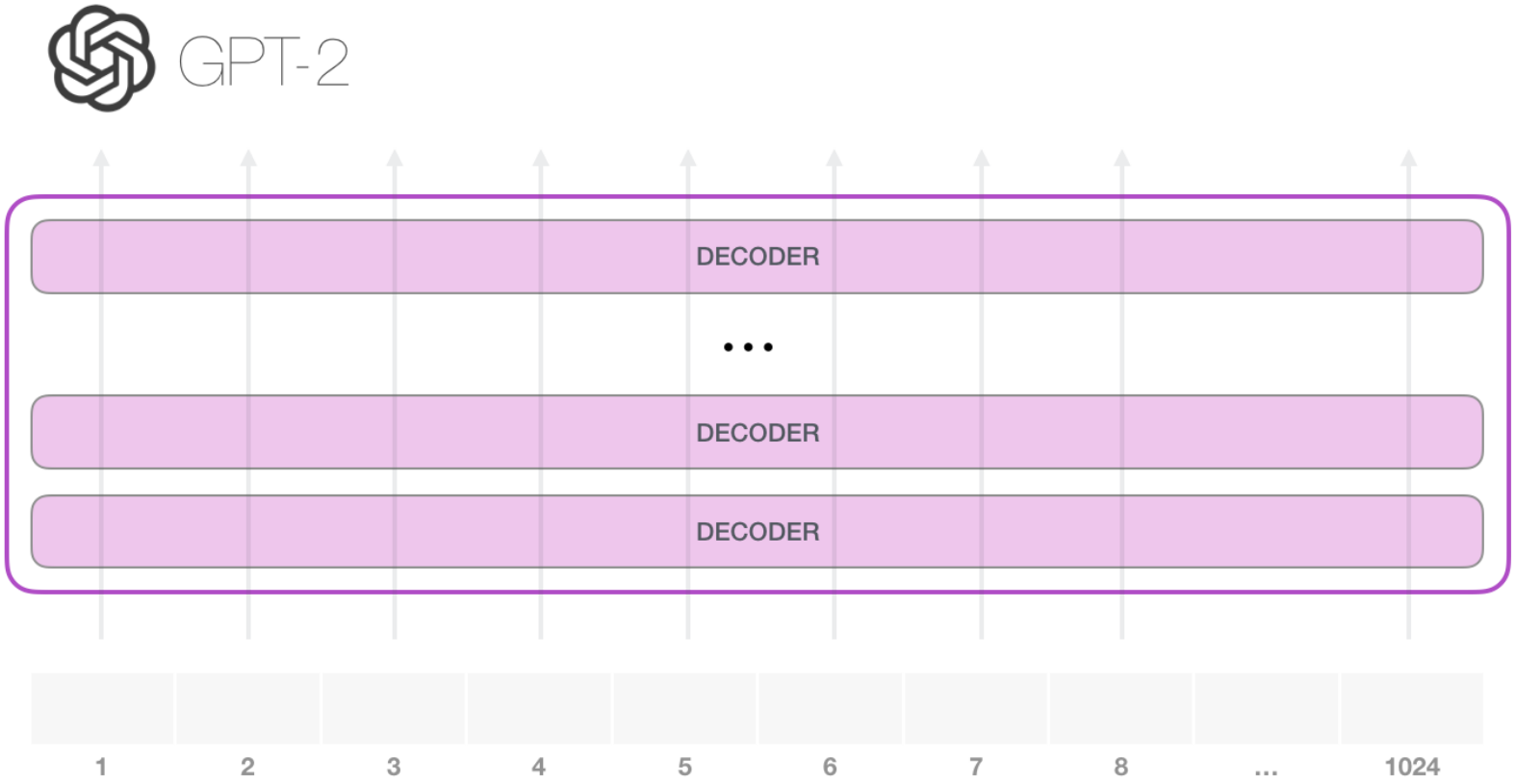
GPT-2: Language Models are Unsupervised Multitask Learners

- Main difference between GPT-2 and BERT
  - ✓ GPT-2 is auto-regressive but BERT is not
    - After each token is produced, that token is added to the sequence of inputs



GPT-2: Language Models are Unsupervised Multitask Learners

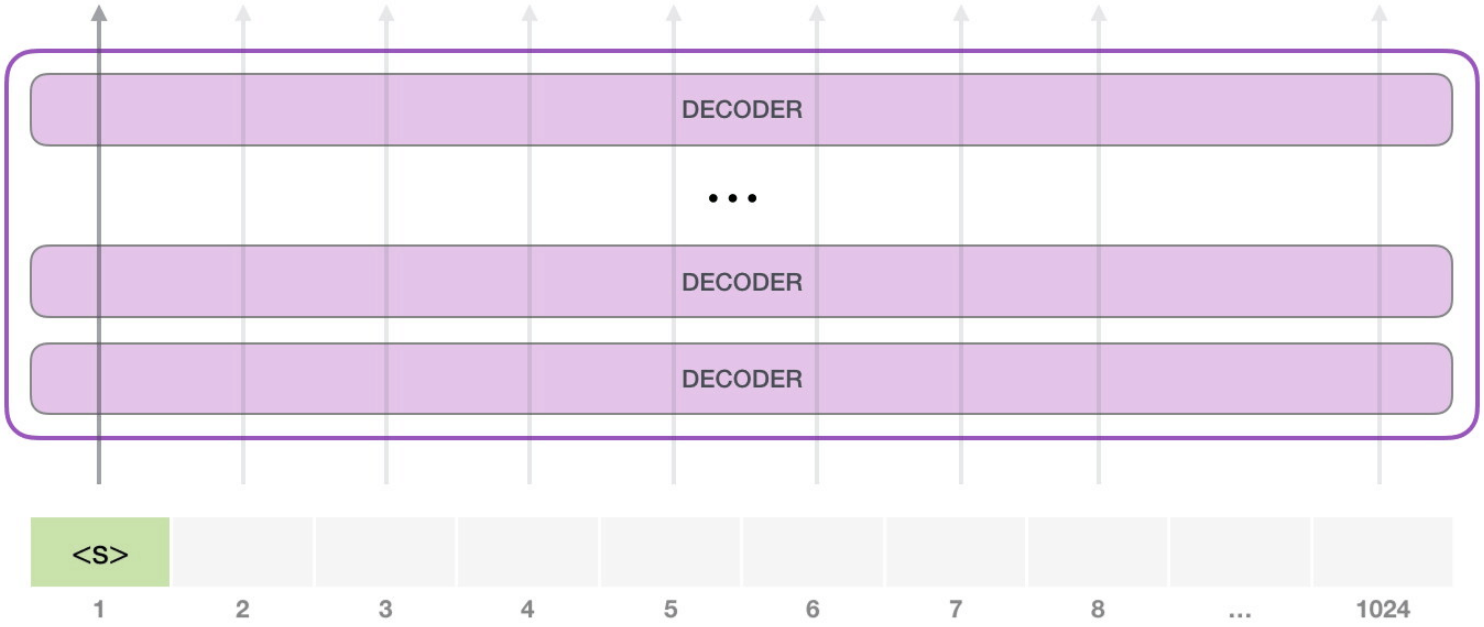
- GPT-2 can process 1024 tokens
  - ✓ Each token flows through all the decoder blocks along its own path



GPT-2: Language Models are Unsupervised Multitask Learners

Alammar (GPT-2)

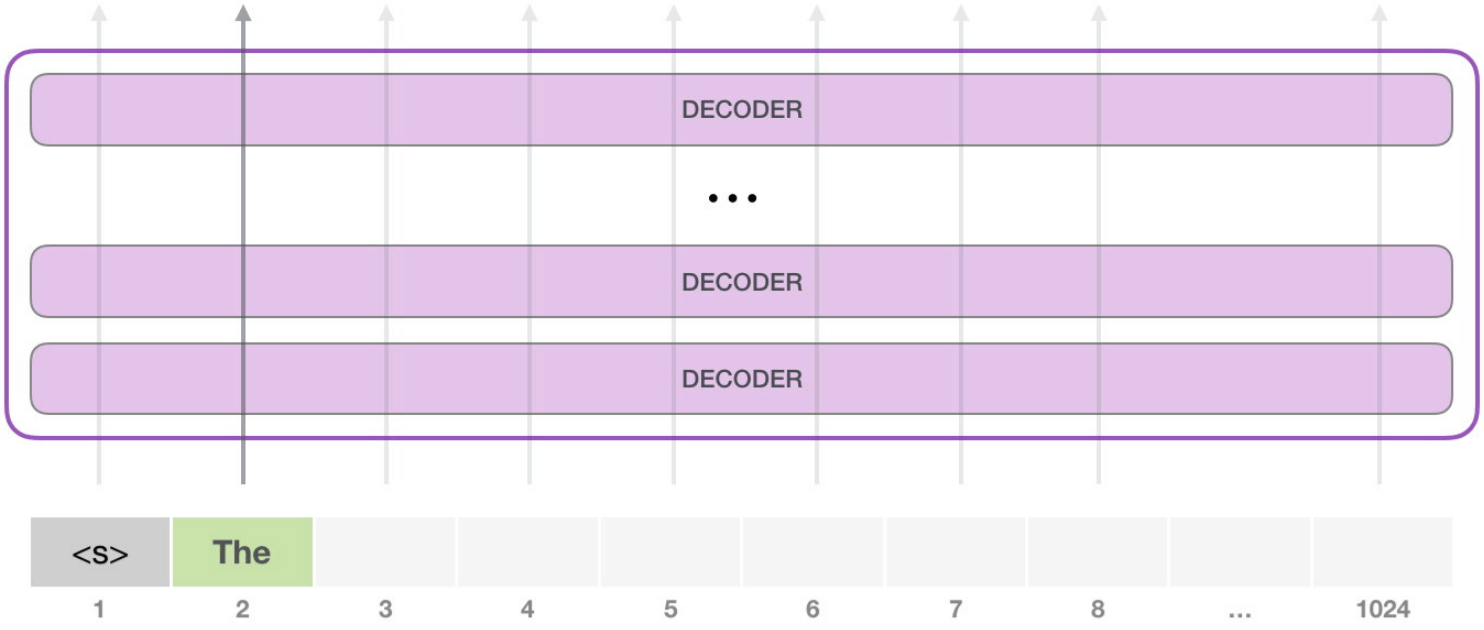
- The simplest way to run a trained GPT-2 is to allow it to ramble on its own
  - ✓ Generating unconditional samples
  - ✓ GPT-2 has a parameter called top-k that we can use to have the model consider sampling words other than the top word



GPT-2: Language Models are Unsupervised Multitask Learners

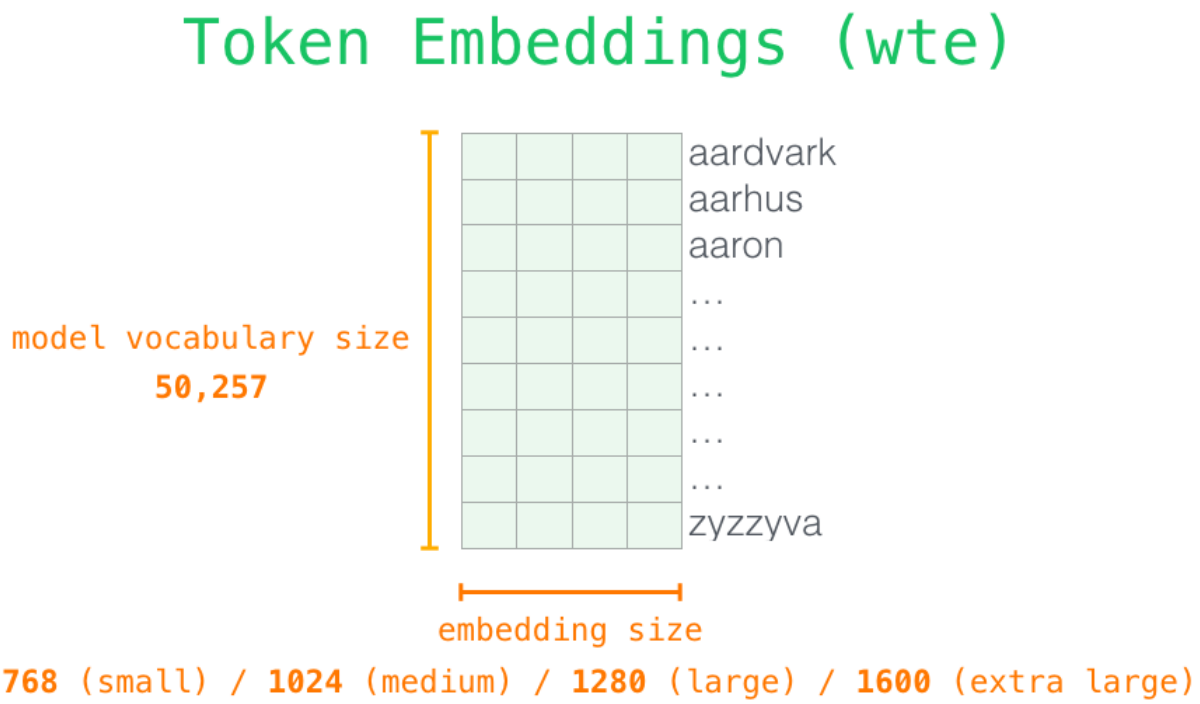
Alammar (GPT-2)

- In the next step, we add the output from the first step to our input sequence, and have the model make its next prediction:
  - ✓ The second path is the only that's active in this calculation
  - ✓ GPT-2 does not re-interpret the first token in light of the second token



GPT-2: Language Models are Unsupervised Multitask Learners

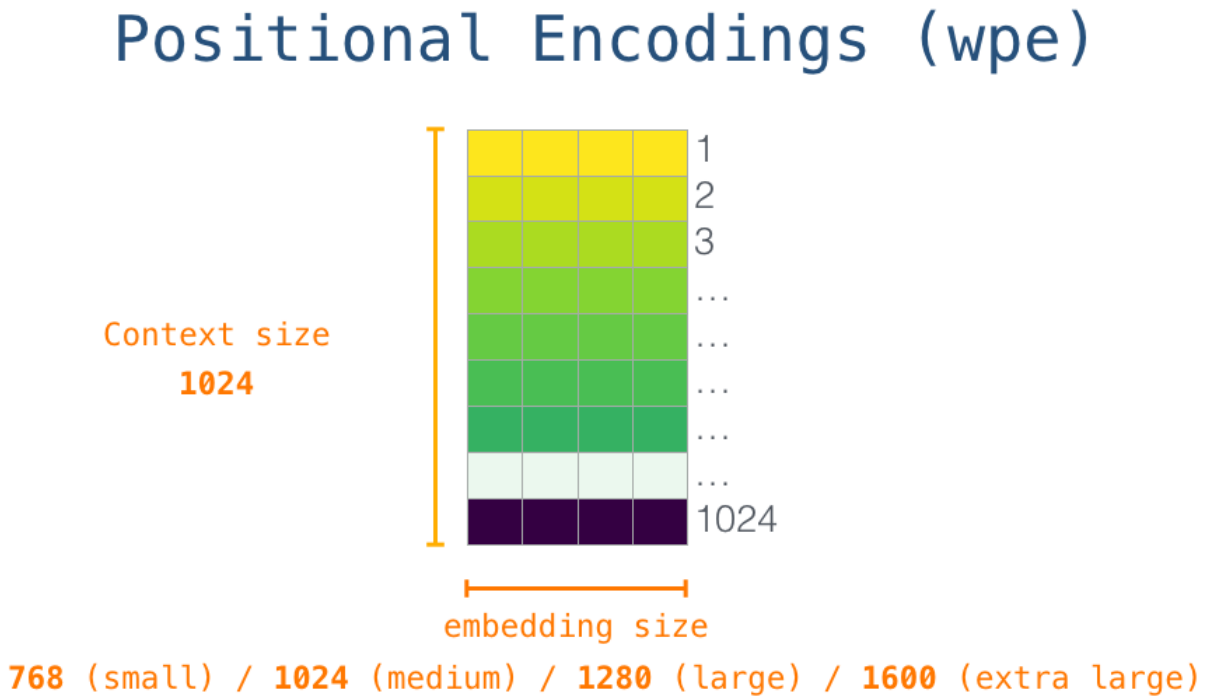
- GPT2: A deeper look inside
  - ✓ Input encoding



GPT-2: Language Models are Unsupervised Multitask Learners

Alammar (GPT-2)

- GPT2: A deeper look inside
  - ✓ Positional encoding



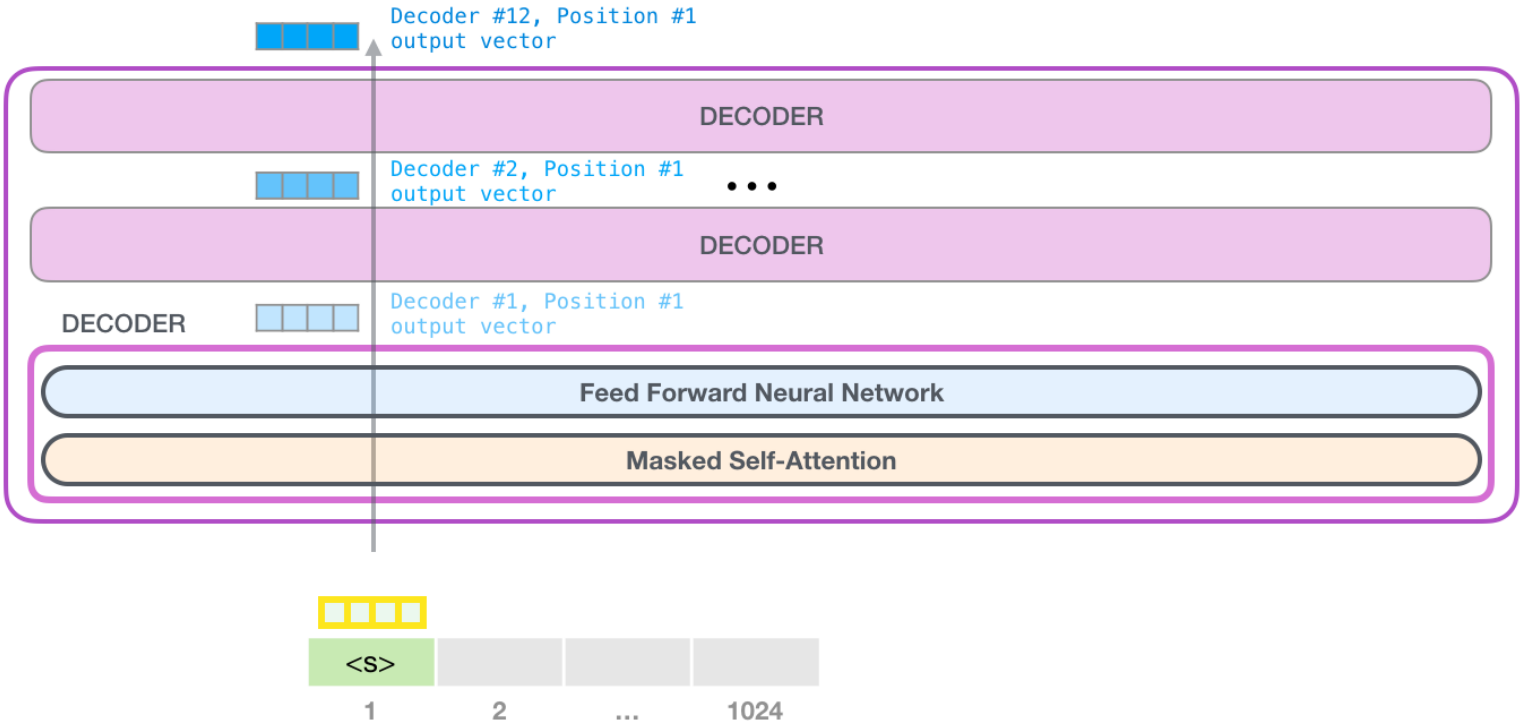
GPT-2: Language Models are Unsupervised Multitask Learners

Alammar (GPT-2)

- GPT2: A deeper look inside
  - ✓ Sending a word to the first transformer block

GPT-2: Language Models are Unsupervised Multitask Learners

- A journey up the stack
  - ✓ Once a lower-level transformer block processes the token, it sends its resulting vector up the stack to be processed by the next block
    - The process is identical in each block, but each block has its own weights in both self-attention and the neural network sublayers



GPT-2: Language Models are Unsupervised Multitask Learners

- Self-Attention Recap
  - ✓ Language heavily relies on context
    - Look at the second law

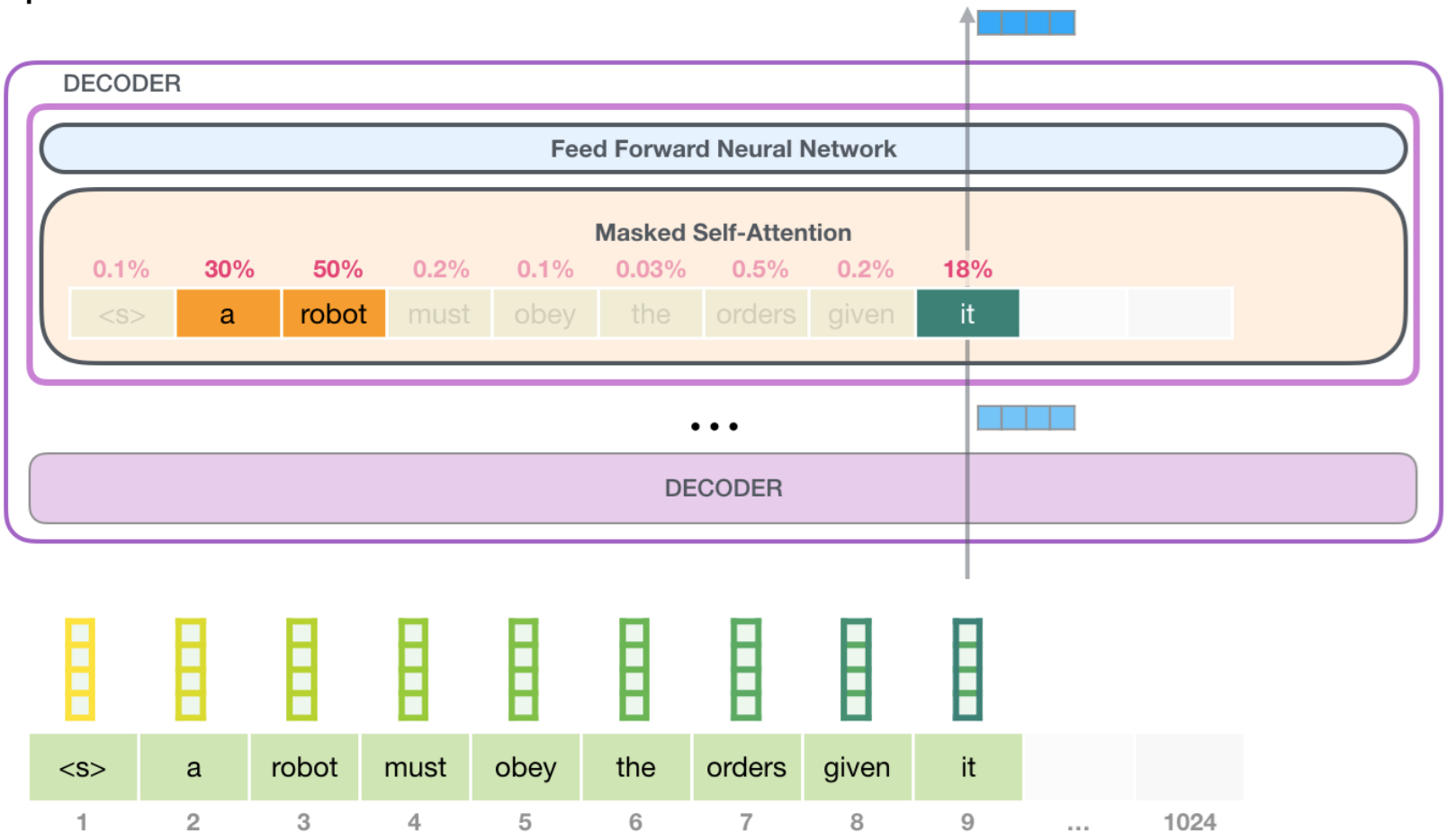
> ***First Law of Robotics***
> *A robot may not injure a human being or, through inaction, allow a human being to come to harm.*

> ***Second Law of Robotics***
> *A robot must obey the orders given **it** by human beings except where **such orders** would conflict with the **First Law**.*

  - ✓ When a model processes this sentence, it has to be able to know that
    - *it* refers to the robot
    - *such orders* refers to the earlier part of the law, namely "the orders given it by human beings"
    - *The First Law* refers to the entire First Law
  - ✓ This is what self-attention does

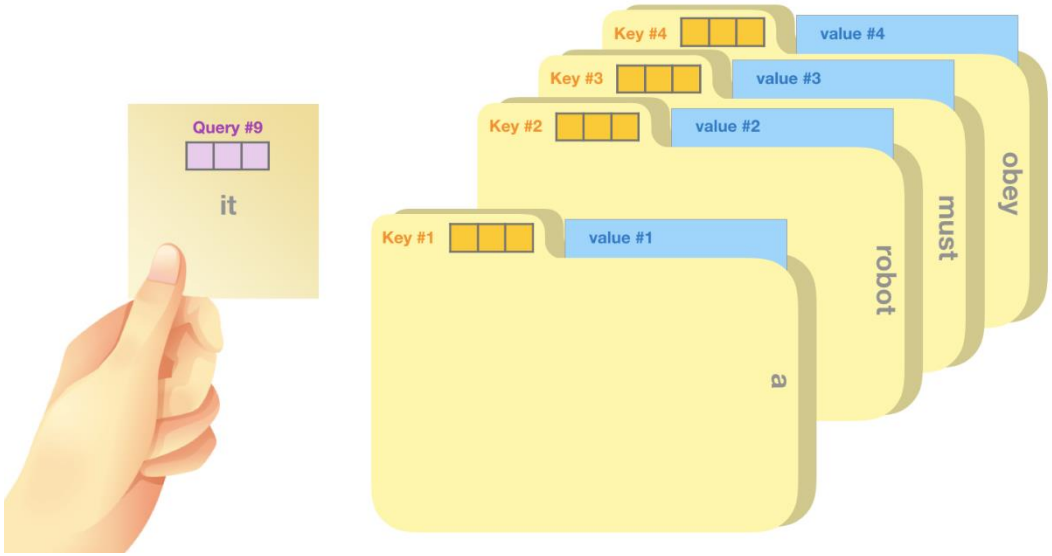
GPT-2: Language Models are Unsupervised Multitask Learners

- Self-Attention Recap
  - ✓ This self-attention layer in the top block is paying attention to "a robot" when it processes the word "it"



GPT-2: Language Models are Unsupervised Multitask Learners

- Self-Attention Recap
  - ✓ Think of it like searching through a filing cabinet

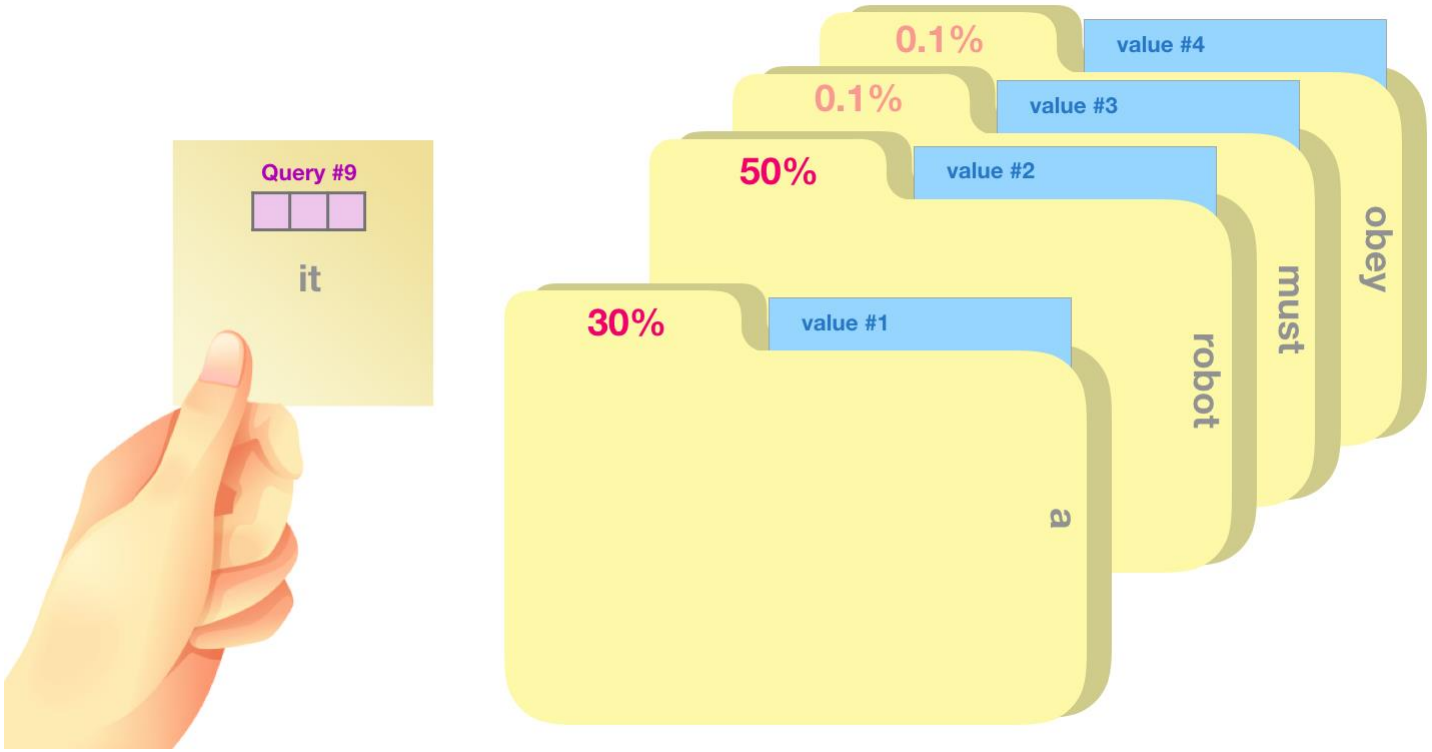


- The query is like a sticky note with the topic you're researching
- The keys are like the labels of the folders inside the cabinet
- When you match the tag with a sticky note, we take out the contents of that folder, these contents are the value vector
- Except you're not only looking for one value, but a blend of values from a blend of folders.

GPT-2: Language Models are Unsupervised Multitask Learners

Alammar (GPT-2)

- Self-Attention Recap
  - ✓ Multiplying the query vector by each key vector produces a score for each folder (technically: dot product followed by softmax)



GPT-2: Language Models are Unsupervised Multitask Learners

- Self-Attention Recap

✓ Multiply each value by its score and sum up – resulting in our self-attention outcome

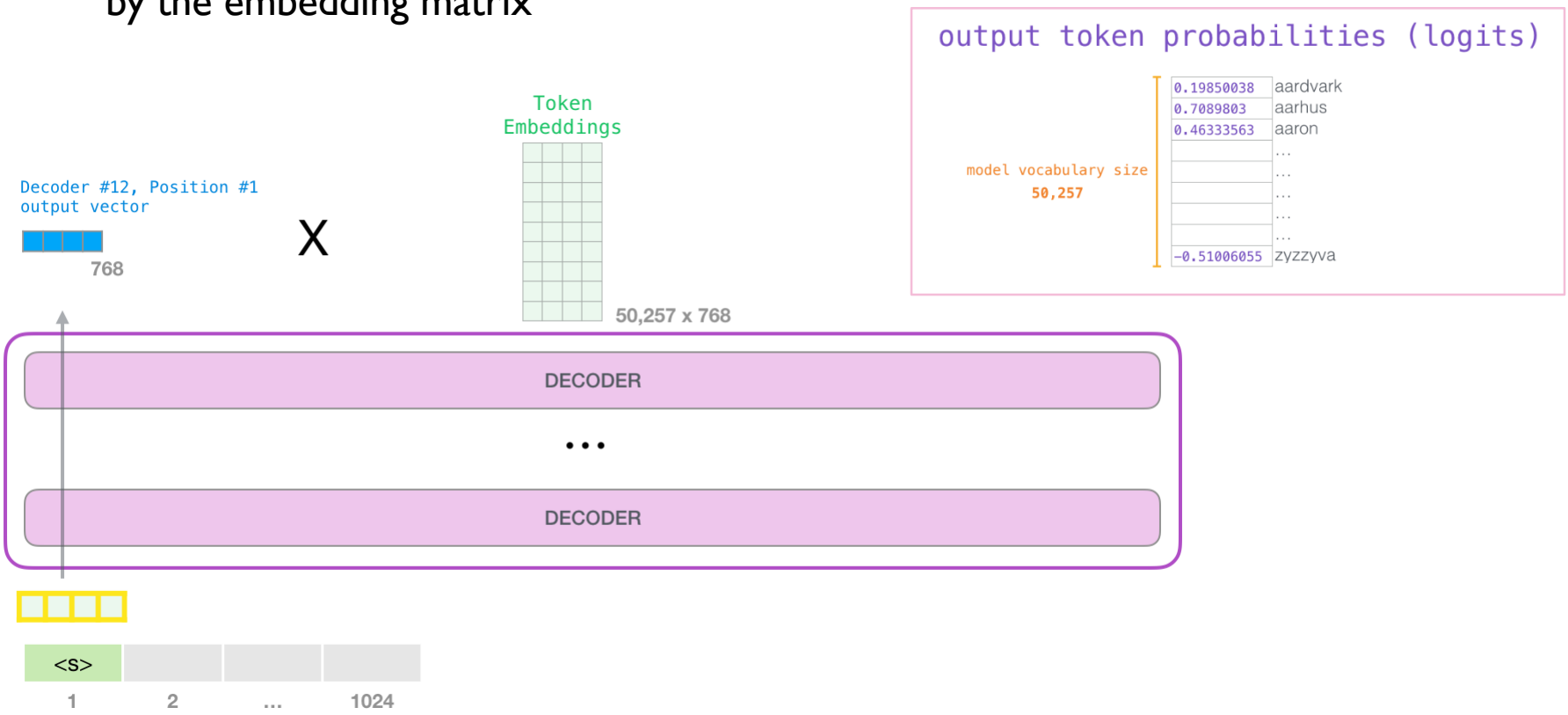
Word	Value vector	Score	Value X Score
<s>		0.001	
a		0.3	
robot		0.5	
must		0.002	
obey		0.001	
the		0.0003	
orders		0.005	
given		0.002	
it		0.19	
		Sum:	

- This weighted blend of value vectors results in a vector that paid 50% of its attention to the word `robot`, 30% to the word `a`, and 19% to the word `it`

GPT-2: Language Models are Unsupervised Multitask Learners

Alammar (GPT-2)

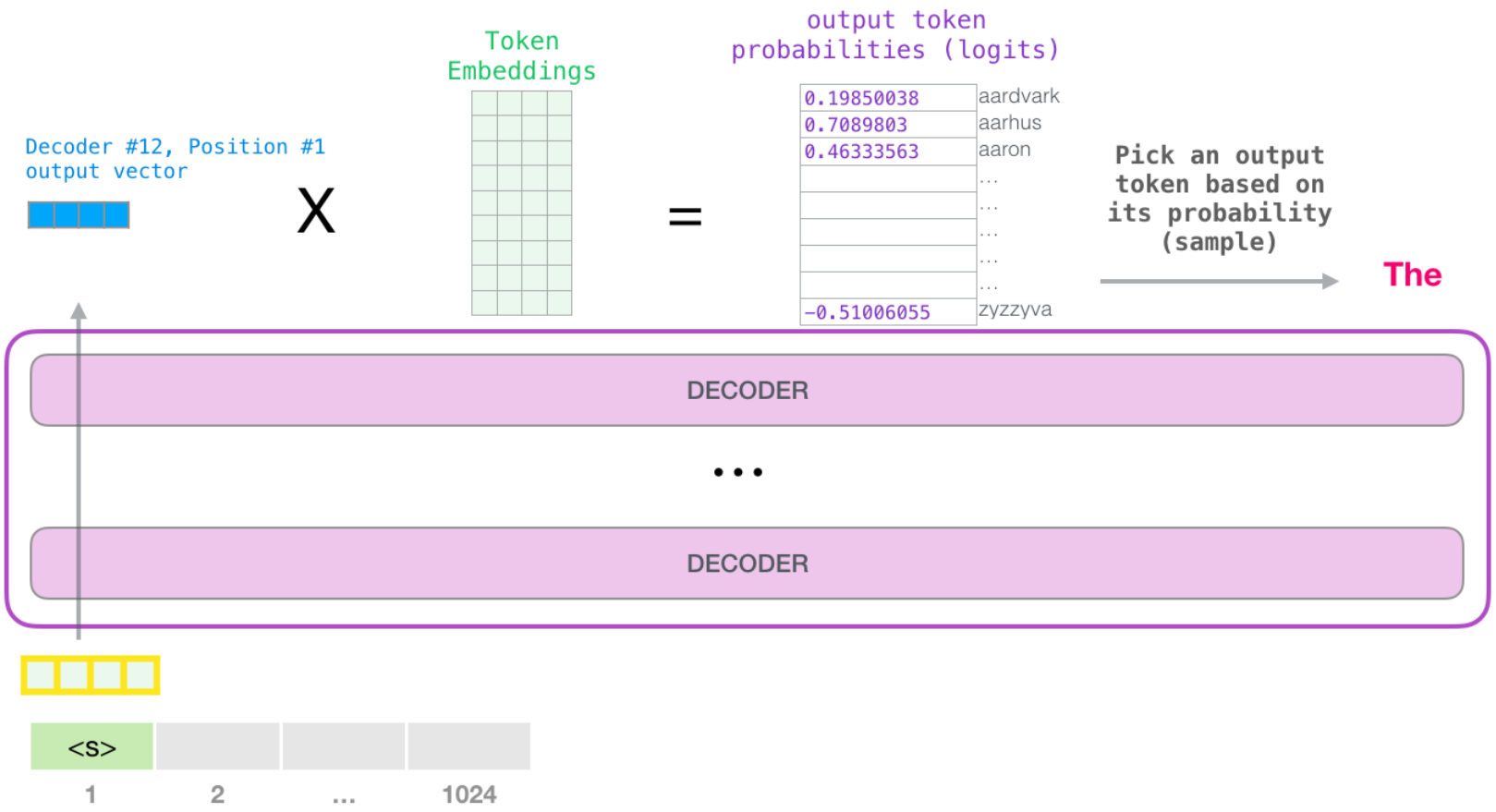
- Model Output
  - ✓When the top block in the model produces its output vector (the result of its own self-attention followed by its own neural network), the model multiplies that vector by the embedding matrix



GPT-2: Language Models are Unsupervised Multitask Learners

Alammar (GPT-2)

- Model Output

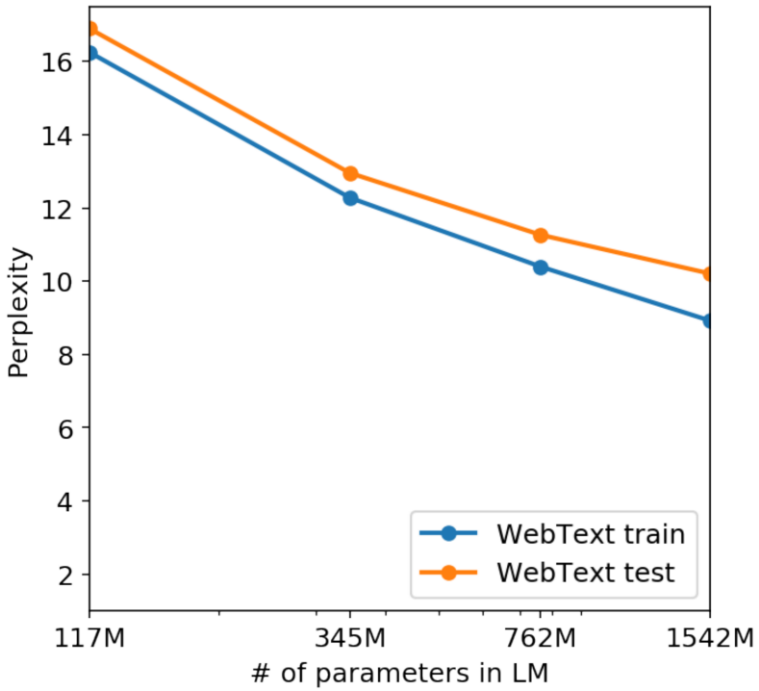


GPT-2: Language Models are Unsupervised Multitask Learners

- Notes in GPT-2
  - ✓ GPT-2 uses Byte Pair Encoding to create the tokens in its vocabulary; tokens are usually parts of words
  - ✓ When training, a maximum of 512 tokens are processes at the same time
  - ✓ Layer normalization is important in Transformer structure

GPT-2: Language Models are Unsupervised Multitask Learners

- Experiments
  - ✓ Performance w.r.t. model size



- ✓ Zero-shot results on Language Modeling datasets

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

GPT-2: Language Models are Unsupervised Multitask Learners

- Experiments
  - ✓ Answers for the questions not in the training dataset

Question	Generated Answer	Correct	Probability
Who wrote the book the origin of species?	Charles Darwin	✓	83.4%
Who is the founder of the ubuntu project?	Mark Shuttleworth	✓	82.0%
Who is the quarterback for the green bay packers?	Aaron Rodgers	✓	81.1%
Panda is a national animal of which country?	China	✓	76.8%
Who came up with the theory of relativity?	Albert Einstein	✓	76.4%
When was the first star wars film released?	1977	✓	71.4%
What is the most common blood type in sweden?	A	✗	70.6%
Who is regarded as the founder of psychoanalysis?	Sigmund Freud	✓	69.3%
Who took the first steps on the moon in 1969?	Neil Armstrong	✓	66.8%
Who is the largest supermarket chain in the uk?	Tesco	✓	65.3%
What is the meaning of shalom in english?	peace	✓	64.0%
Who was the author of the art of war?	Sun Tzu	✓	59.6%
Largest state in the us by land mass?	California	✗	59.2%
Green algae is an example of which type of reproduction?	parthenogenesis	✗	56.5%
Vikram samvat calender is official in which country?	India	✓	55.6%
Who is mostly responsible for writing the declaration of independence?	Thomas Jefferson	✓	53.3%
What us state forms the western boundary of montana?	Montana	✗	52.3%
Who plays ser davos in game of thrones?	Peter Dinklage	✗	52.1%
Who appoints the chair of the federal reserve system?	Janet Yellen	✗	51.5%
State the process that divides one nucleus into two genetically identical nuclei?	mitosis	✓	50.7%
Who won the most mvp awards in the nba?	Michael Jordan	✗	50.2%
What river is associated with the city of rome?	the Tiber	✓	48.6%
Who is the first president to be impeached?	Andrew Johnson	✓	48.3%
Who is the head of the department of homeland security 2017?	John Kelly	✓	47.0%
What is the name given to the common currency to the european union?	Euro	✓	46.8%
What was the emperor name in star wars?	Palpatine	✓	46.5%
Do you have to have a gun permit to shoot at a range?	No	✓	46.4%
Who proposed evolution in 1859 as the basis of biological development?	Charles Darwin	✓	45.7%
Nuclear power plant that blew up in russia?	Chernobyl	✓	45.7%
Who played john connor in the original terminator?	Arnold Schwarzenegger	✗	45.2%

Beyond ELMo, GPT, and BERT

Post BERT

BERT

OCTOBER 11, 2018

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding by Jacob Devlin et al

GPT-2

FEBRUARY 14, 2019

Language Models are Unsupervised Multitask Learners

XLNet

JUNE 19, 2019

XLNet: Generalized Autoregressive Pretraining for Language Understanding

CTRL

SEPTEMBER 11, 2019

CTRL: A Conditional Transformer Language Model for Controllable Generation

Transformer-XL

JANUARY 9, 2019

Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context

ERNIE

APRIL 19, 2019

ERNIE: Enhanced Representation through Knowledge Integration

RoBERTa

JULY 26, 2019

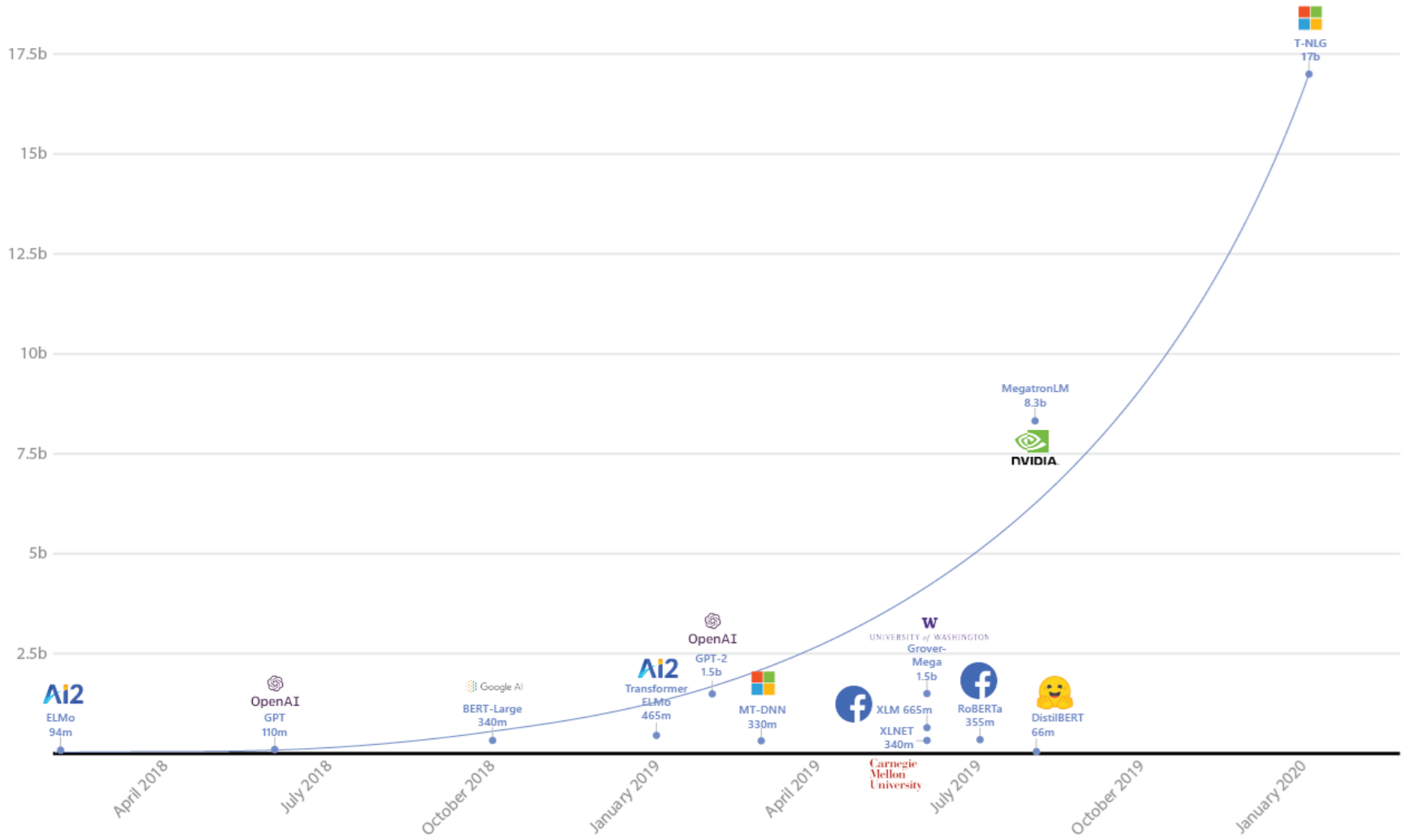
RoBERTa: A Robustly Optimized BERT Pretraining Approach

ALBERT

SEPTEMBER 26, 2019

ALBERT: A Lite BERT for Self-supervised Learning of Language Representations

Bigger, BIGger, BIGGER!!



https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/


ANY
questions?

References

Research Papers

- 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.
- Jernite, Y., Bowman, S. R., & Sontag, D. (2017). Discourse-based objectives for fast unsupervised sentence representation learning. arXiv preprint arXiv:1705.00557.
- McCann, B., Bradbury, J., Xiong, C., & Socher, R. (2017). Learned in translation: Contextualized word vectors. In Advances in Neural Information Processing Systems (pp. 6294-6305).
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.
- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8), 9.

Other Materials

- Jay Alammar, The Illustrated GPT-2, http://jalammar.github.io/illustrated-gpt2