Improving Language Understanding by Generative Pre-Training

Jishnu Jaykumar Padalunkal

CS 6301.004 - Deep Learning For NLP

Spring 2023

Authors

OpenAI

Alec Radford

Karthik Narasimhan

Tim Salimans

Ilya Sutskever

Coming back to the **Title**

Improving **Language Understanding** by **Generative Pre-Training**

Language Understanding

- Natural language understanding comprises a wide range of diverse tasks such as
  - **Textual entailment** - involves determining the **directional relationship** between two pieces of text . The goal is to determine if the hypothesis (Text2) is entailed (true), contradicted (false), or neutral with respect to the premise (Text1).
  - Y = TE(Text1, Text2)
    - Y ∈ {Entail, Neutral, Contradiction}

```
Premise: "The dog is running in the park."
Hypothesis: "The animal is exercising."
Entailment: The premise entails the hypothesis because the dog is running, which is a form of exercising.

Premise: "The dog is running in the park."
Hypothesis: "The animal is sleeping."
Contradiction: The premise contradicts the hypothesis because the dog is running and not sleeping

Premise: "The dog is running in the park."
Hypothesis: "The dog is in the park."
Neutral: The premise and the hypothesis convey the same information
```

Language Understanding

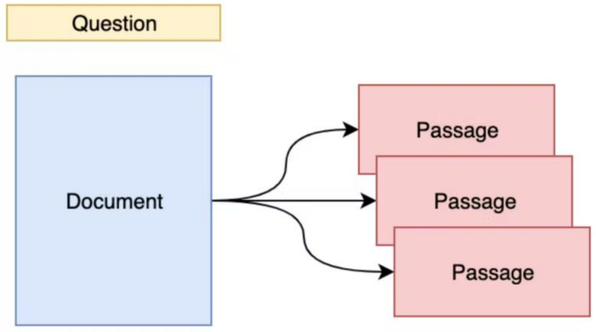
- Natural language understanding comprises a wide range of diverse tasks such as
  - **Textual entailment** - involves determining the **directional relationship** between two pieces of text . The goal is to determine if the hypothesis (Text2) is entailed (true), contradicted (false), or neutral with respect to the premise (Text1).
  - Y = TE(Text1, Text2)
    - $Y \in$ {Entail, Neutral, Contradiction}

Positive	Text **(Sentence 1)** implies hypothesis **(Sentence 2)**
Negative	Text **(Sentence 1)** contradicts hypothesis **(Sentence 2)**
Neutral	Text **(Sentence 1)** cannot prove or disprove hypothesis **(Sentence 2)**

Image: https://www.oreilly.com/content/textual-entailment-with-tensorflow

Language Understanding

- Natural language understanding comprises a wide range of diverse tasks such as
  - **Question answering -** is a task where a system is given a **question** in natural language and a **set of documents** or **text** as <u>context</u>, and it is expected to return the correct answer to the question.



The goal of QA is to understand the question and find the **answer** within the context.

Image: https://www.deepset.ai/blog/modern-question-answering-systems-explained

Language Understanding

- Natural language understanding comprises a wide range of diverse tasks such as
  - **Semantic similarity assessment** - involves determining the similarity between two pieces of text. The goal is to measure how closely related the meaning of two texts are.
  - **Document classification** - involves assigning predefined categories or labels to a given document. The goal is to automatically classify documents into one or more predefined categories based on their content.

Semantic similarity
Sentence 1: "The cat sat on the mat"
Sentence 2: "A feline was resting on a rug"

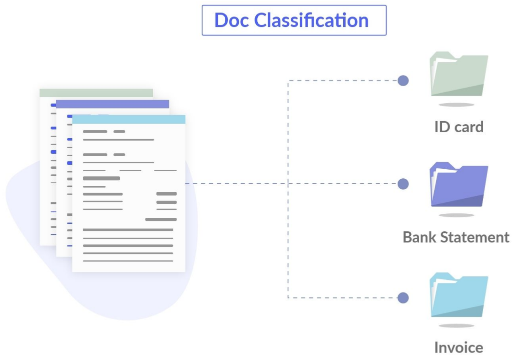


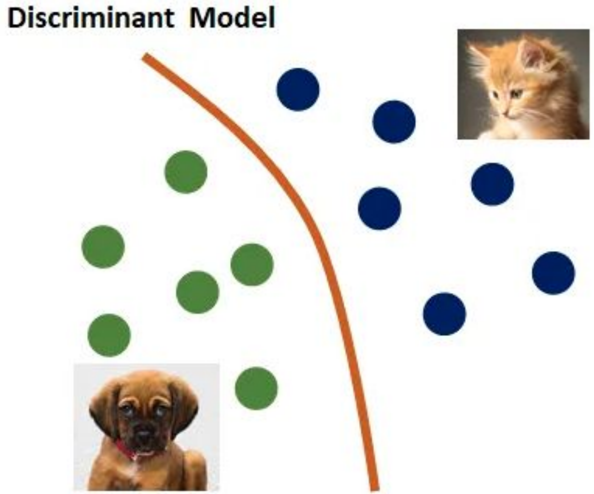
Image: https://www.docsumo.com/blog/auto-document-classification

Labeled Vs Unlabeled Text Data

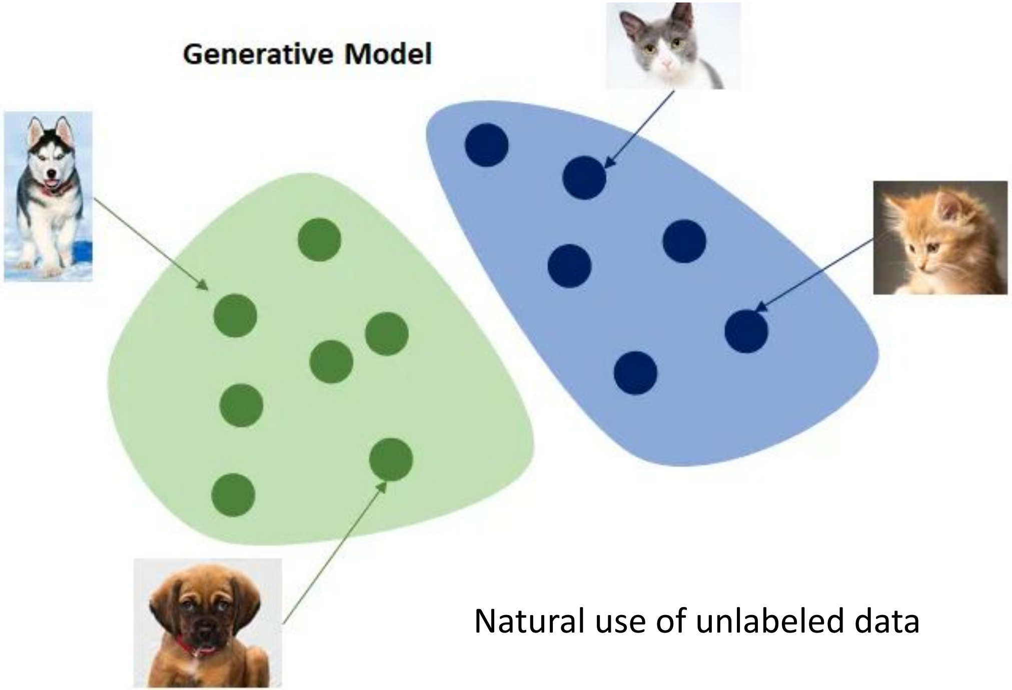
Can we leverage the vastly present **unlabeled data** to build a robust language model for **language understanding**?

Image: https://music-classification.github.io/tutorial/part4_beyond/semi-supervised-learning.html

Discriminative Vs Generative Models



Supervised, not designed for unlabeled data



Natural use of unlabeled data

Image: https://medium.com/@jordi299/about-generative-and-discriminative-models-d8958b67ad32

Related Work Vs GPT(1)

Semi-supervised learning for NLP
Word/Phrase/Sentence Level

High Level semantics

Unsupervised pre-training
Find a good initialization point instead of modifying the supervised learning objective.

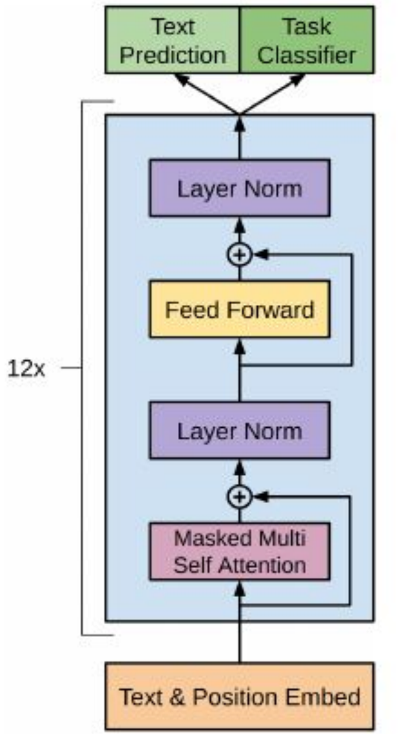
GPT

Performs unsupervised pretraining

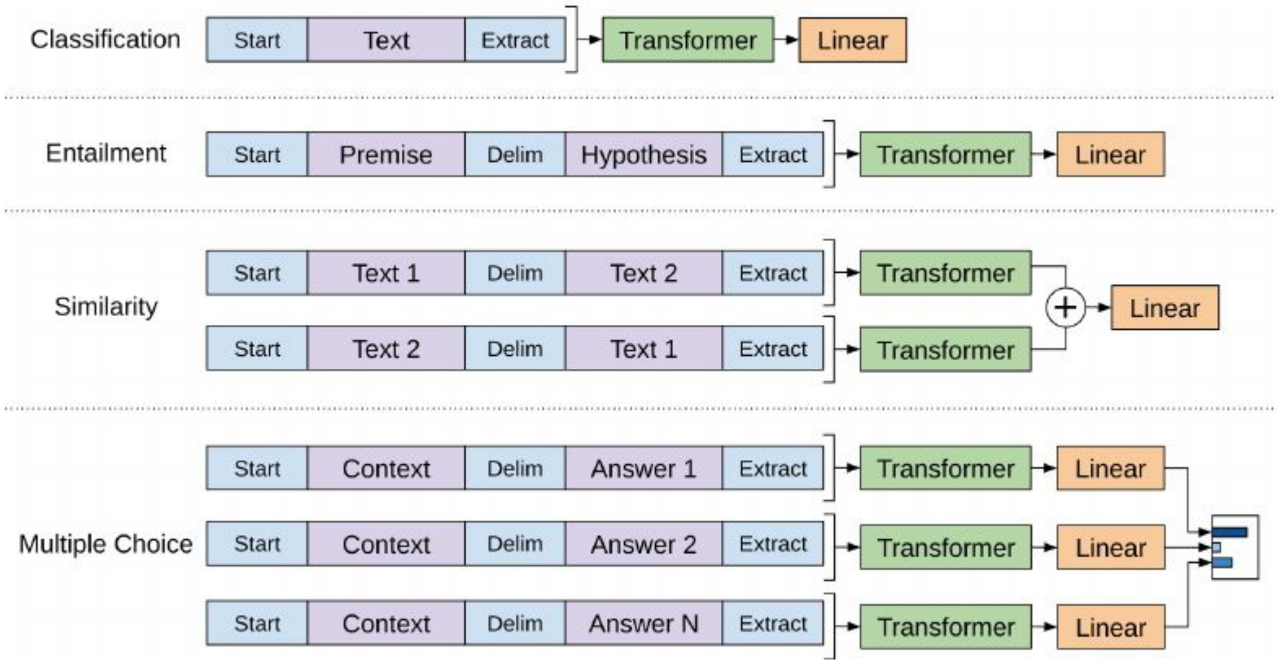
Auxiliary training objectives (ATO)
Task oriented

ATO is used but unsupervised pre-training already learns several linguistic aspects relevant to target tasks

GPT Framework

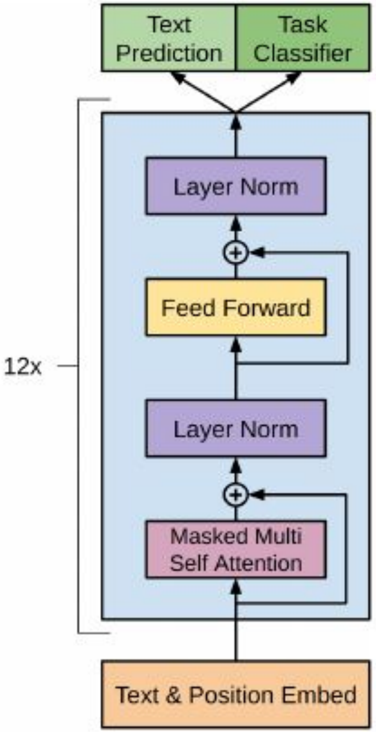


Stage-1: Unsupervised pre-training



Stage-2: Supervised fine-tuning

Stage-1: Unsupervised pre-training



Transformer decoder

https://arxiv.org/pdf/1801.10198.pdf

Given an unsupervised corpus of tokens $\mathcal{U} = \{u_1, \ldots, u_n\}$, we use a standard language modeling objective to maximize the following likelihood:

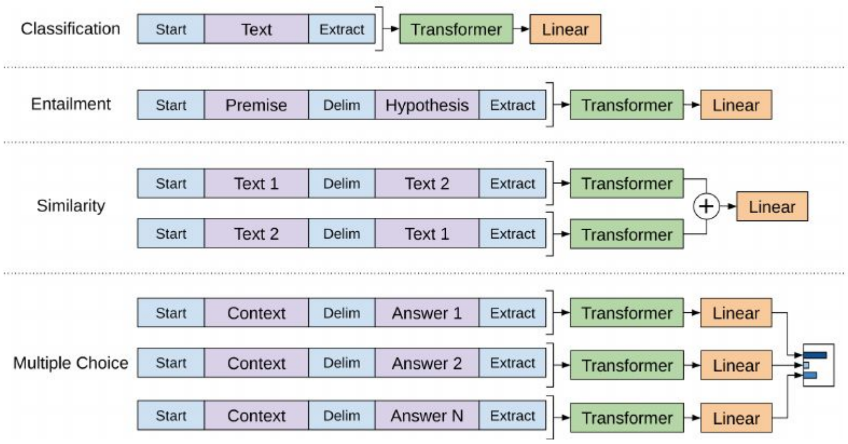
$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta) \tag{1}$$

where k is the size of the context window, and the conditional probability P is modeled using a neural network with parameters Θ. These parameters are trained using stochastic gradient descent [51].

$$\begin{aligned} h_0 &= UW_e + W_p \\ h_l &= \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n] \\ P(u) &= \texttt{softmax}(h_n W_e^T) \end{aligned}$$

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

Stage-2: Supervised fine-tuning



Assumption: A labeled dataset C, where each instance consists of a sequence of input tokens, x1 , . . . , xm, along with a label y

$$P(y|x^1,\ldots,x^m) = \texttt{softmax}(h_l^m W_y)$$

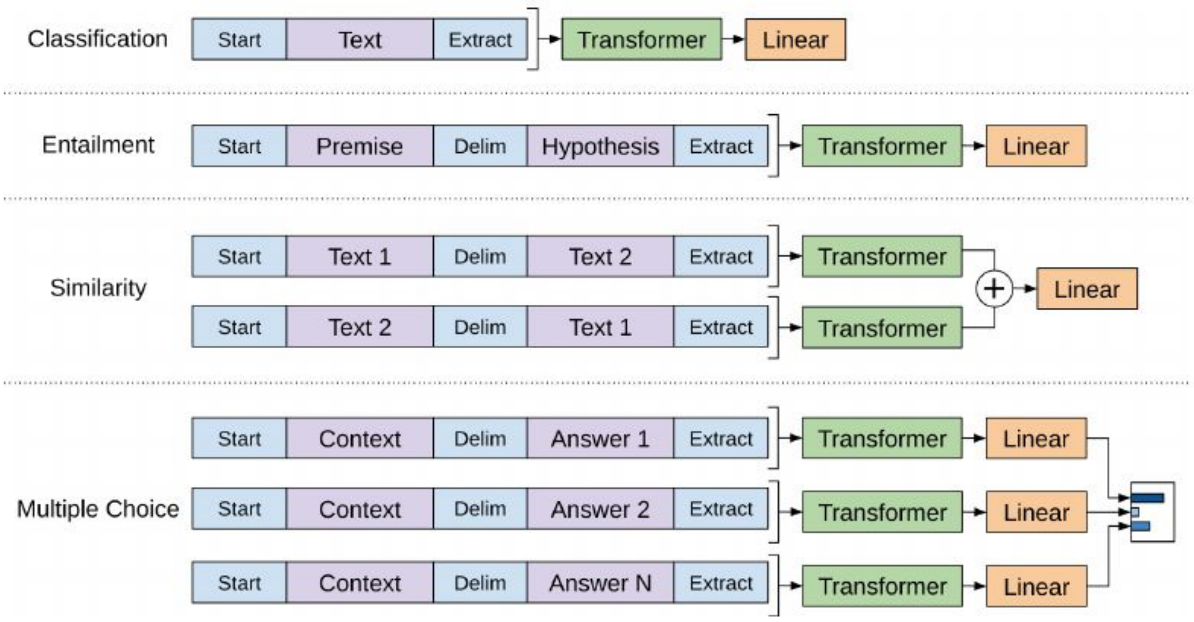
maximize:

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

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

Including language modeling as an auxiliary objective to the fine-tuning **helped learning** by (a) improving generalization of the supervised model, and (b) accelerating convergence.

Task-specific input transformations



All transformations include adding randomly initialized start and end tokens (<s>, <e>)

delimiter token ($)

document z, a question q, and a set of possible answers {a_k}.
[z; q; $; a_k]

These input transformations allow to avoid making extensive changes to the architecture across tasks

Experiments

- Unsupervised pre-training
  - Dataset: **BooksCorpus** dataset
  - **> 7,000** unique unpublished books from a variety of genres including Adventure, Fantasy, and Romance
  - Crucially, it contains **long stretches of contiguous text**, which **allows** the generative model to **learn** to **condition on long-range information**.
  - 1B Word Benchmark - approximately the same size - shuffled at a sentence level - achieved low token level **perplexity** of **18.4**

$$PP(W) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}}$$

https://towardsdatascience.com/perplexity-intuition-and-derivation-105dd481c8f3

Model specifications

Model specifications Our model largely follows the original transformer work [62]. We trained a 12-layer decoder-only transformer with masked self-attention heads (768 dimensional states and 12 attention heads). For the position-wise feed-forward networks, we used 3072 dimensional inner states. We used the Adam optimization scheme [27] with a max learning rate of 2.5e-4. The learning rate was increased linearly from zero over the first 2000 updates and annealed to 0 using a cosine schedule. We train for 100 epochs on minibatches of 64 randomly sampled, contiguous sequences of 512 tokens. Since layernorm [2] is used extensively throughout the model, a simple weight initialization of $N(0, 0.02)$ was sufficient. We used a bytepair encoding (BPE) vocabulary with 40,000 merges [53] and residual, embedding, and attention dropouts with a rate of 0.1 for regularization. We also employed a modified version of L2 regularization proposed in [37], with $w = 0.01$ on all non bias or gain weights. For the activation function, we used the Gaussian Error Linear Unit (GELU) [18]. We used learned position embeddings instead of the sinusoidal version proposed in the original work. We use the *ftfy* library[2] to clean the raw text in BooksCorpus, standardize some punctuation and whitespace, and use the *spaCy* tokenizer.[3]

Fine-tuning details

Fine-tuning details Unless specified, we reuse the hyperparameter settings from unsupervised pre-training. We add dropout to the classifier with a rate of 0.1. For most tasks, we use a learning rate of 6.25e-5 and a batchsize of 32. Our model finetunes quickly and 3 epochs of training was sufficient for most cases. We use a linear learning rate decay schedule with warmup over 0.2% of training. λ was set to 0.5.

Tasks and 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]

Results on NLI tasks (Metric: Accuracy)

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

5x indicates an ensemble of 5 models.

Results on question answering and commonsense reasoning

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**

9x means an ensemble of 9 models.

Results

Table 4: Semantic similarity and classification results, comparing our model with current state-of-the-art methods. All task evaluations in this table were done using the GLUE benchmark. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

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**

Analysis

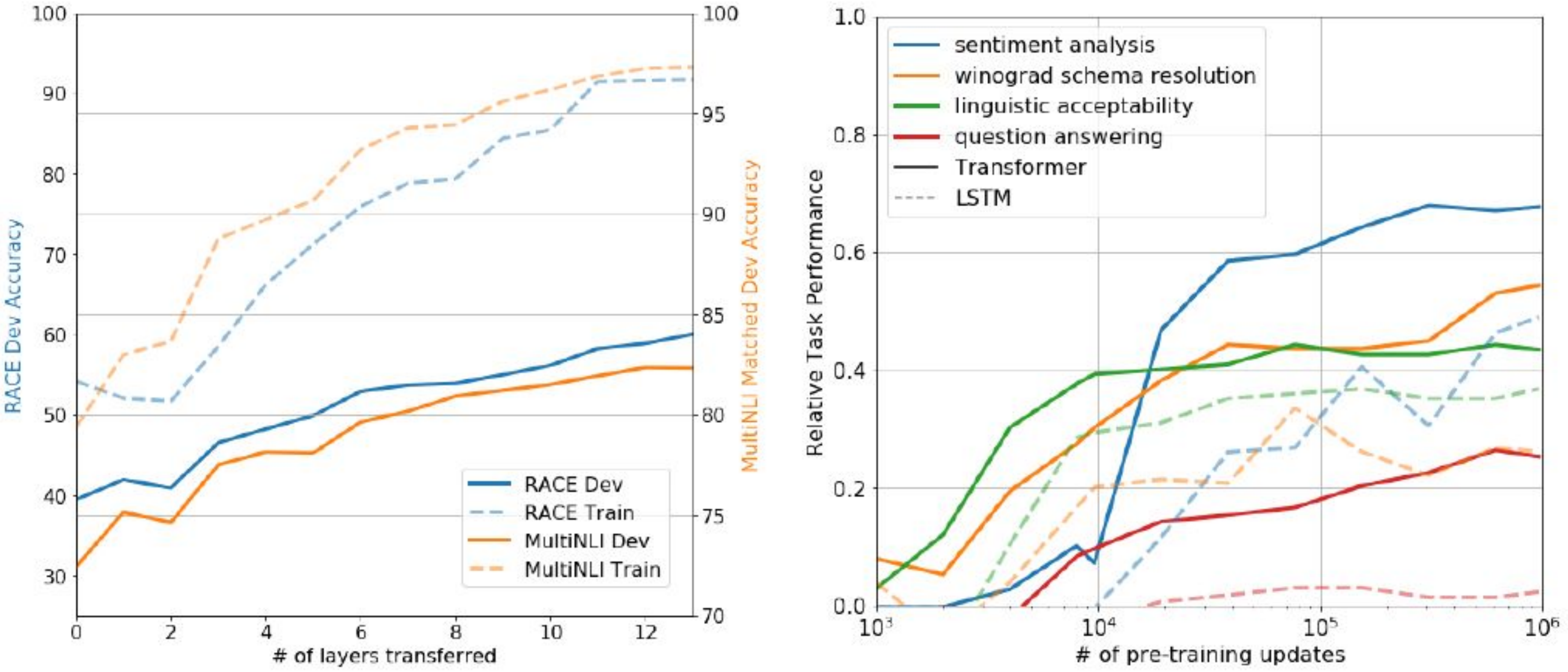


Figure 2: (**left**) Effect of transferring increasing number of layers from the pre-trained language model on RACE and MultiNLI. (**right**) Plot showing the evolution of zero-shot performance on different tasks as a function of LM pre-training updates. Performance per task is normalized between a random guess baseline and the current state-of-the-art with a single model.

Ablation Study

Table 5: Analysis of various model ablations on different tasks. Avg. score is a unweighted average of all the results. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

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

Conclusion

- **GPT(1)** - a **framework** for achieving strong natural language understanding with a **single task-agnostic model** through **generative pre-training** and **discriminative fine-tuning**.
- State of the art on 9 of the 12 datasets mentioned.
- Using **unsupervised (pre-)training** to **boost performance** on **discriminative tasks** has long been an important goal of Machine Learning research.
- This paper suggests that achieving significant performance gains is indeed possible, and offers hints as to what models (**Transformers**) and data sets (**text with long range dependencies**) work best with this approach.

Questions?