(GPT-2)
Language Models are Unsupervised Multitask Learners

Alec Radford, Jeffrey Wu, Rewon Child, David Luan,
Dario Amodei, Ilya Sutskever

2021-02-20

1.Introduction

narrow experts vs. competent generalist

- Machine learning systems now excel (in expectation) at tasks they are trained for by using a combination of large datasets, high-capacity models, and supervised learning . Yet these systems are brittle and sensitive to slight changes in the data distribution and task specification.

- We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one
- We demonstrate language models can perform down-stream tasks in a zero-shot setting – without any parameter or architecture modification

2.Approach

Language model & general system

$$p(x) = \prod_{i=1}^{n} p(s_n|s_1, ..., s_{n-1})$$

- Language modeling is usually framed as unsupervised distribution estimation from a set of examples (x1, x2, ..., xn) each composed of variable length sequences of symbols (s1, s2, ..., sn).
- Learning to perform a single task can be expressed in a probabilistic framework as estimating a conditional distribution **p(output|input)**.
- Since a general system should be able to perform many different tasks, even for the same input, it should condition not only on the input but also on the task to be performed. That is, it should model **p(output|input, task)**.

  ex) (translate to french, english text, french text), (answer the question, document, question, answer)

2.Approach

supervised objectivity vs. unsupervised objectivity

- Since the supervised objective is the the same as the unsupervised objective but only evaluated on a subset of the sequence, the global minimum of the unsupervised objective is also the global minimum of the supervised objective.

- Preliminary experiments confirmed that sufficiently large language models are able to perform multitask learning in this toy-ish setup but learning is much slower than in explicitly supervised approaches.

2.Approach

dataset: WebText

- We created a new web scrape which emphasizes document quality. To do this we only scraped web pages which have been curated/filtered by humans.

- We removed all Wikipedia documents from WebText since it is a common data source for other datasets and could complicate analysis due to overlapping training data with test evaluation tasks.

2.Approach

Input Representation: BPE(Byte Pair Encoding)

- aaabdaaabac

  -> ZabdZabac (aa->Z)

  -> ZYdZYac (ab->Y)

- size of vocabulary : 4(a,b,c,d) -> 5(a,c,d,Y,Z) / length of data : 11 -> 7
- **방법 정리:**

  1. 어휘 집합 구축 : 자주 등장하는 문자열(위의 예시에서 `aa`, `ab`들)를 병합하고 이를 어휘 집합(사전)에 추가한다. 이를 원하는 어휘 집합 크기가 될 때까지 반복한다.

  2.**토큰화** : 토큰화 대상 문장 내 각 어절(띄어쓰기로 문장을 나눈 것)에서 어휘 집합에 있는 서브워드가 포함되어 있을 경우 해당 서브워드를 어절에서 분리한다.

출처: https://ratsgo.github.io/nlpbook/docs/tokenization/bpe/

2.Approach

Model

[Unsupervised pre-training]

- **likelihood**

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

k: size of context window, Θ: parameter

- **multi-layer Transformer decoder**

$$h_0 = UW_e + W_p$$

$$h_l = \text{transformer\_block}(h_{l-1}) \quad \forall l \in [1, n]$$

$$P(u) = \text{softmax}(h_n W_e^T)$$

U: context vector of token
n: # of layers
We: token embedding matrix
Wp: positional embedding matrix

2.Approach

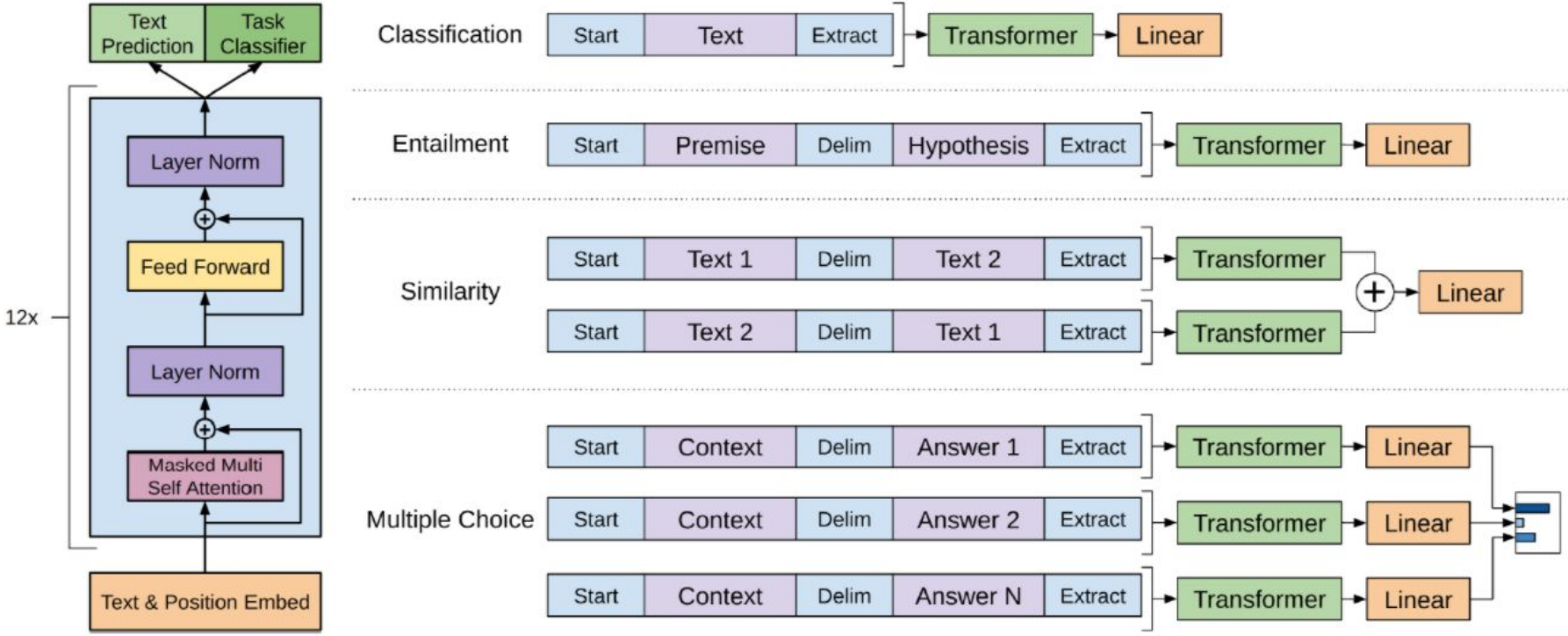
Model

[Supervised fine-tuning]

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

*__language model as an auxiliary objective__: (a) improving generalization of the supervised model, (b)accelerating convergence

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


Text Prediction
Task Classifier
12x
Layer Norm
Feed Forward
Layer Norm
Masked Multi Self Attention
Text & Position Embed
Classification
Start
Text
Extract
Transformer
Linear
Entailment
Start
Premise
Delim
Hypothesis
Extract
Transformer
Linear
Similarity
Start
Text 1
Delim
Text 2
Extract
Start
Text 2
Delim
Text 1
Extract
Transformer
Transformer
Linear
Multiple Choice
Start
Context
Delim
Answer 1
Extract
Start
Context
Delim
Answer 2
Extract
Start
Context
Delim
Answer N
Extract
Transformer
Linear

2.Approach

Model

1. Layer normalization was moved to the input of each sub-block
2. an additional layer normalization was added after the final self-attention block

3. We scale the weights of residual layers at initialization by a factor of $1/\sqrt{N}$ where N is the number of residual layers.

4. The vocabulary is expanded to 50,257
5. We also increase the context size from 512 to 1024 tokens
6. a larger batch size of 512 is used.

3.Experiments

Language Models are Unsupervised Multitask Learners

	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	56.25	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

- PPL(perplexity)

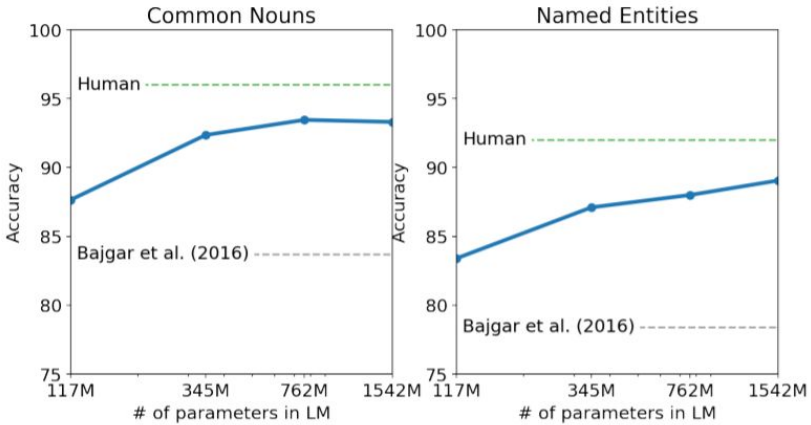
$$PPL(W) = \sqrt[N]{\frac{1}{P(w_1, w_2, w_3, \ldots, w_N)}} = \sqrt[N]{\frac{1}{\prod_{i=1}^{N} P(w_i | w_1, w_2, \ldots, w_{i-1})}}$$

3.Experiments

Children's Book Test

- CBT reports accuracy on an automatically constructed cloze test where the task is to predict which of 10 possible choices for an omitted word is correct.

- GPT-2 achieves new state of the art results of 93.3% on common nouns and 89.1% on named entities.



3.Experiments

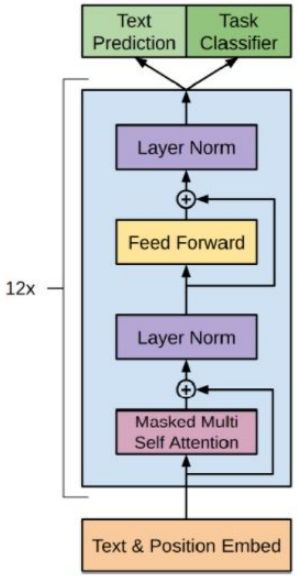
LAMBADA

- The task is to predict the final word of sentences which require at least 50 tokens of context for a human to successfully predict

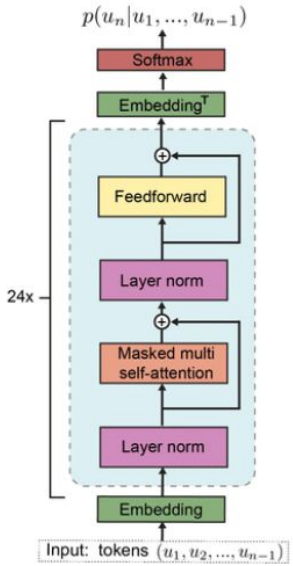
- GPT-2 improves the state of the art from 99.8 to 8.6 perplexity and increases the accuracy of LMs on this test from 19% to 52.66%.
- Investigating GPT-2's errors showed most predictions are valid continuations of the sentence, but are not valid final words. This suggests that the LM is not using the additional useful constraint that the word must be the final of the sentence. Adding a stop-word filter as an approximation to this further increases accuracy to 63.24%,

[Q&A]

Q. GPT-1과 GPT-2에서 Layer normalization의 위치가 어떻게 달라졌는지?



GPT-1



GPT-2

- GPT-1 모델 두번째 layer norm이 GPT-2 모델에서는 attention 전 단계인, embedded input을 받는 첫 단계로 위치가 달라진 것을 확인할 수 있었습니다
- 즉, GPT-1에서는 Attention->layer norm-> ff -> layer norm 순서로, GPT-2에서는 layer norm->attention -> layer norm -> ff 순서로 변경되었습니다.
- 구체적으로 변경된 이유는 언급되지 않았고, 변경된 GPT-2 구조는 pre-activation residual network와 유사하다고 합니다!

[Q&A]

Q. fine-tuning 과정에서 auxiliary objectivity가 성능을 높이는 데 도움이 되는지?

- Unsupervised pre-training objectivity $L_1(\mathcal{U}) = \sum_i \log P(u_i|u_{i-k}, \ldots, u_{i-1}; \Theta)$
- Supervised fine-tuning objectivity (without auxiliary objectivity) $L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m)$
- Supervised fine-tuning objectivity (with auxiliary objectivity) $L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$

발표 때 불명확하게 말씀드렸던 것 같아서 다시 한번 정리해보았습니다!

다음 장에서 성능에 관해 얘기하면,

[Q&A]

Q. fine-tuning 과정에서 auxiliary objectivity가 성능을 높이는 데 도움이 되는지?

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

GPT-1에서의 실험 결과에 따르면, task에 따라서 w/ w/o aux LMD은 성능을 높이는 데 도움이 되기도, 되지 않기도 하다는 것을 확인할 수 있습니다!