GPT

Generative Pre-trained Transformer

17기 신예진, 18기 홍진우

목차

GPT 1

GPT-1의 모델
구조 및 특징
모델 학습과정
BPE

GPT 2

모델의 특징
학습과정

GPT 3

GPT3 개요
및 모델 구조
학습과정
결과 및 한계
실습

GPT 개요

OpenAI

Bert와 함께 최신 NLP 연구의 양대산맥을 이루는 모델
Generative Pre-trained Transformer

Improving Language Understanding by Generative Pre-Training

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever
OpenAI
alec@openai.com karthikn@openai.com tim@openai.com ilyasu@openai.com

Abstract

Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by *generative pre-training* of a language model on a diverse corpus of unlabeled text, followed by *discriminative fine-tuning* on each specific task. In contrast to previous approaches, we make use of task-aware input transformations during fine-tuning to achieve effective transfer while requiring minimal changes to the model architecture. We demonstrate the effectiveness of our approach on a wide range of benchmarks for natural language understanding. Our general task-agnostic model outperforms discriminatively trained models that use architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied. For instance, we achieve absolute improvements of 8.9% on commonsense reasoning (Stories Cloze Test), 5.7% on question answering (RACE), and 1.5% on textual entailment (MultiNLI).

1 Introduction

The ability to learn effectively from raw text is crucial to alleviating the dependence on supervised learning in natural language processing (NLP). Most deep learning methods require substantial amounts of manually labeled data, which restricts their applicability in many domains that suffer from a dearth of annotated resources [61]. In these situations, models that can leverage linguistic information from unlabeled data provide a valuable alternative to gathering more annotation, which can be time-consuming and expensive. Further, even in cases where considerable supervision is available, learning good representations in an unsupervised fashion can provide a significant performance boost. The most compelling evidence for this so far has been the extensive use of pre-trained word embeddings [10, 39, 42] to improve performance on a range of NLP tasks [8, 11, 26, 45].

Leveraging more than word-level information from unlabeled text, however, is challenging for two main reasons. First, it is unclear what type of optimization objectives are most effective at learning text representations that are useful for transfer. Recent research has looked at various objectives such as language modeling [44], machine translation [38], and discourse coherence [22], with each method outperforming the others on different tasks.[1] Second, there is no consensus on the most effective way to transfer these learned representations to the target task. Existing techniques involve a combination of making task-specific changes to the model architecture [43, 44], using intricate learning schemes [21] and adding auxiliary learning objectives [50]. These uncertainties have made it difficult to develop effective semi-supervised learning approaches for language processing.

[1] https://gluebenchmark.com/leaderboard

GPT 1 - Improving Language Understanding by Generative Pre-Training

2018.06

Language Models are Unsupervised Multitask Learners

Alec Radford * [1] Jeffrey Wu * [1] Rewon Child [1] David Luan [1] Dario Amodei ** [1] Ilya Sutskever ** [1]

Abstract

Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential to the success of zero-shot task transfer and increasing it improves performance in a log-linear fashion across tasks. Our largest model, GPT-2, is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting but still underfits WebText. Samples from the model reflect these improvements and contain coherent paragraphs of text. These findings suggest a promising path towards building language processing systems which learn to perform tasks from their naturally occurring demonstrations.

1. Introduction

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 (Krizhevsky et al., 2012) (Sutskever et al., 2014) (Amodei et al., 2016). Yet these systems are brittle and sensitive to slight changes in the data distribution (Recht et al., 2018) and task specification (Kirkpatrick et al., 2017). Current systems are better characterized as narrow experts rather than competent generalists. 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.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent and identically distributed (IID) held-out examples. This has served well to make progress on narrow experts. But the often erratic behavior of captioning models (Lake et al., 2017), reading comprehension systems (Jia & Liang, 2017), and image classifiers (Alcorn et al., 2018) on the diversity and variety of possible inputs highlights some of the shortcomings of this approach.

Our suspicion is that the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. Progress towards robust systems with current architectures is likely to require training and measuring performance on a wide range of domains and tasks. Recently, several benchmarks have been proposed such as GLUE (Wang et al., 2018) and decaNLP (McCann et al., 2018) to begin studying this.

Multitask learning (Caruana, 1997) is a promising framework for improving general performance. However, multitask training in NLP is still nascent. Recent work reports modest performance improvements (Yogatama et al., 2019) and the two most ambitious efforts to date have trained on a total of 10 and 17 (dataset, objective) pairs respectively (McCann et al., 2018) (Bowman et al., 2018). From a meta-learning perspective, each (dataset, objective) pair is a single training example sampled from the distribution of datasets and objectives. Current ML systems need hundreds to thousands of examples to induce functions which generalize well. This suggests that multitask training many need just as many effective training pairs to realize its promise with current approaches. It will be very difficult to continue to scale the creation of datasets and the design of objectives to the degree that may be required to brute force our way there with current techniques. This motivates exploring additional setups for performing multitask learning.

The current best performing systems on language tasks

* **Equal contribution [1] OpenAI, San Francisco, California, United States. Correspondence to: Alec Radford <alec@openai.com>.

GPT 2 – Language Models are Unsupervised Multitask Learners

2019.02

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*
Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry
Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan
Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter
Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray
Benjamin Chess Jack Clark Christopher Berner
Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

Abstract

Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples or from simple instructions – something which current NLP systems still largely struggle to do. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic. At the same time, we also identify some datasets where GPT-3's few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora. Finally, we find that GPT-3 can generate samples of news articles which human evaluators have difficulty distinguishing from articles written by humans. We discuss broader societal impacts of this finding and of GPT-3 in general.

*Equal contribution
†Johns Hopkins University, OpenAI

Author contributions listed at end of paper.

GPT 3 – Language Models are Few-shot Learners

2020.07

GPT?

1) Generative Language model (언어모델)

이전 단어들이 주어졌을 때, 현재 알고 있는 단어들을 기반으로 다음 단어를 예측하는 모델

Ex) n-gram, RNN 계열의 Language model

Language model 의 특징

Language Model training doesn't need human labeled data!

Generative model

youtube deep learning tutorial

Train Data	Label
youtube	deep
youtube deep	learning
youtube deep learning	tutorial

Discriminative model

Titanic survivor data

sex	age	fare	Label
Man	10	100	survived
Woman	20	50	survived
woman	60	100	Died

학습 데이터 구축에 많은 시간과 비용이 들어가는 labeling이 필요한 Discriminative model이 아닌, Generative model을 사용

labeling을 하는 과정에서 우리가 모르는, 라벨링 과정에서 놓치게 되는 자연어의 내재적인 특징까지 학습 가능

GPT?

2) Pre-trained model (사전 학습 모델)

위키피디아, 책 등의 대용량 코퍼스로 미리 학습 시킨 Pretrained Model

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

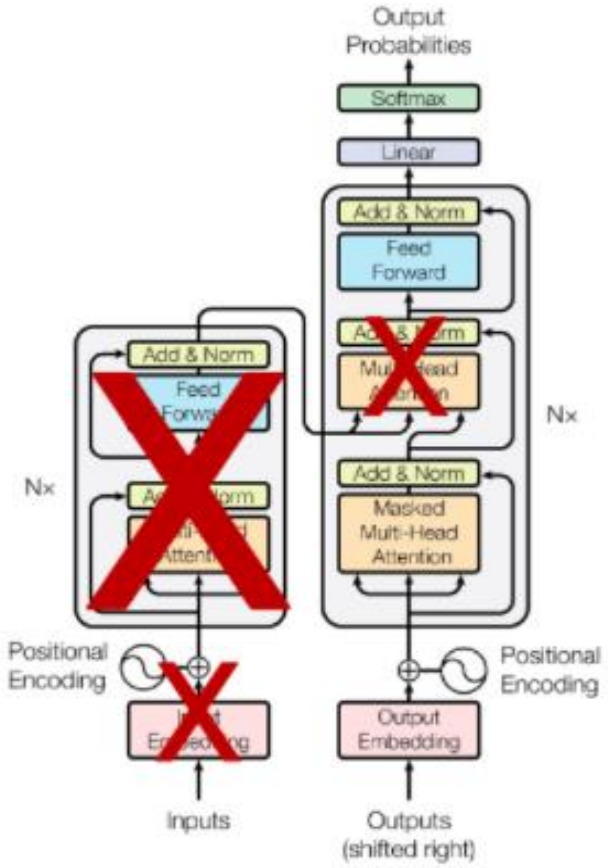
Table 2.2: Datasets used to train GPT-3. "Weight in training mix" refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets are seen less than once.

GPT 3의 경우 약 3억 개의 단어 token들을 학습

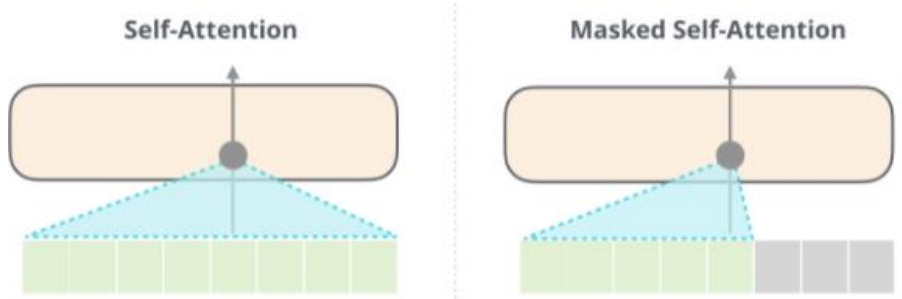
대용량 코퍼스에 대해 사전학습을 진행한 Language Model은 넓은 범위의 언어적 정보를 갖게 되며, 이러한 정보가 자연어 처리의 다양한 task를 해결할 때, 전이되는 효과

GPT?

3) Transformer (트랜스포머의 decoder 기반 모델)



BERT - Transformer의 인코더 아키텍처 활용 / GPT - Transformer의 디코더 아키텍처 활용



GPT 는 다음 단어를 예측하는 언어모델이기 때문에 Masked self-attention을 적용한 Transfomer Decoder를 사용

GPT?

1 Introduction

The ability to learn effectively from raw text is crucial to alleviating the dependence on supervised learning in natural language processing (NLP). Most deep learning methods require substantial amounts of manually labeled data, which restricts their applicability in many domains that suffer from a dearth of annotated resources [61]. In these situations, models that can leverage linguistic information from unlabeled data provide a valuable alternative to gathering more annotation, which can be time-consuming and expensive. Further, even in cases where considerable supervision is available, learning good representations in an unsupervised fashion can provide a significant performance boost. The most compelling evidence for this so far has been the extensive use of pre-trained word embeddings [10, 39, 42] to improve performance on a range of NLP tasks [8, 11, 26, 45].

Leveraging more than word-level information from unlabeled text, however, is challenging for two main reasons. First, it is unclear what type of optimization objectives are most effective at learning text representations that are useful for transfer. Recent research has looked at various objectives such as language modeling [44], machine translation [38], and discourse coherence [22], with each method outperforming the others on different tasks.[1] Second, there is no consensus on the most effective way to transfer these learned representations to the target task. Existing techniques involve a combination of making task-specific changes to the model architecture [43, 44], using intricate learning schemes [21] and adding auxiliary learning objectives [50]. These uncertainties have made it difficult to develop effective semi-supervised learning approaches for language processing.

[1] https://gluebenchmark.com/leaderboard

특정한 task에만 한정적으로 쓰이는 **supervised learning을 위해 labeling된 데이터셋**은 한정적

Labeled dataset만으로 학습한 모델보다는, **수많은 Unlabeled Text Corpora로부터 Generative Pre-training of a language model**을 구현 후, **Specific Task를 위한 Labeled Text Copora로 fine-tuning**을 통해 자연어 처리 성능 향상

GPT-1의 학습 과정

1) Unsupervised Pre-training with LM objective function

label이 없는 대용량 코퍼스를 사용해서 language model로의 비지도 학습을 진행함

이때, token의 비지도 말뭉치 $\mathcal{U} = u_1, \ldots, u_n$ 에 대해,

다음의 우도(likely hood)를 최대화하도록 표준언어모델링 목적함수 사용

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

k = context window (문맥 고려범위)

조건부확률 P : parameter가 Θ인 layer에서 u_i의 확률

문맥 고려범위 i-k ~ i-1까지의 token들이 이후에 i번째 token의 likelihood를 최대화하는 목적함수를 사용하여 학습

파라미터 Θ는 SGD(Stochastic Gradient Descent)로 학습

GPT-1의 학습 과정

1) Unsupervised Pre-training with LM objective function

GPT는 Transformer의 변형인 multi-layer Transformer decoder를 사용

입력 문맥 token에 multi-headed self-attention을 적용한 후, position-wise feedforward layer를 거쳐

target token에 대한 분포를 얻고, 다음 단어를 예측

$$h_0 = UW_e + W_p$$

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

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

$U = (u_{-k}, \ldots, u_{-1})$: token의 문맥벡터

n : layer의 수

W_e : token embedding 행렬

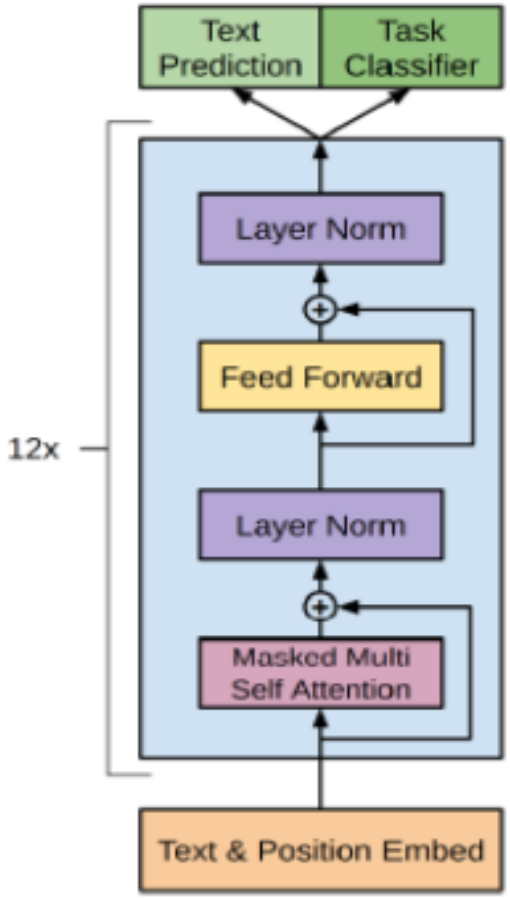
W_p : 위치 embedding 행렬

처음 h0에는 해당하는 token을 position embedding으로 순서 정보 값을 더해주고 -> transformer _block에 넣어서 학습 진행

-> 결과 P(u)는 학습된 마지막 값을 행렬 곱하여 text dictionary만큼 softmax로 다음 단어를 예측함

GPT-1의 학습 과정

1) Unsupervised Pre-training with LM objective function



언어모델은 이전 단어들이 주어졌을 때, 현재 알고 있는 단어들을 기반으로 다음단어를 예측해야 하기 때문에 masked Multi-head self Attention을 사용하는 transformer의 decoder 아키텍처 활용

$$h_0 = UW_e + W_p$$

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

입력 문맥 token에 multi-headed self-attention을 적용한 후

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

position-wise feedforward layer를 거쳐 목표 token을 출력하기 위한 P(u) 값 도출

GPT-1의 학습 과정

2) Supervised Fine Tuning

Pre-training을 거친 후, target task에 맞게 fine-tuning 수행

즉, 1단계에서 학습된 파라미터 Θ를 target task에 맞게 미세조정(fine-tuning)

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

Fine-tuning에 사용되는 labeled 데이터셋

: 입력 token $x^1,\ldots,x^m$ 들과 정답(label) y로 구성

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

입력은 사전 학습된 transformer decoder의 layer들을 거쳐서 결과를 전달하고, 이 결과는 다시 y값을 예측하기 위해 파라미터값들을 수정하며 지도학습 진행

최종 목적함수 $L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$ 를 최적화

GPT-1의 4가지 task

GPT 1 논문에서 제시한 4가지 task

- Natural Language Inference

The Stanford Natural Language Inference (SNLI) Corpus

Sentence 1	Sentence 2	Label
A man inspects the uniform of a figure in some East Asian country	The man is sleeping	contradiction
A soccer game with multiple males playing	Some men are playing a sport.	entailment

- Sematic Similarity

Sentence 1	Sentence 2	label
Deep learning is resolving many NLP problems recently.	Many NLP problems has been resolved by deep learning now a days.	similar

- Question Answering

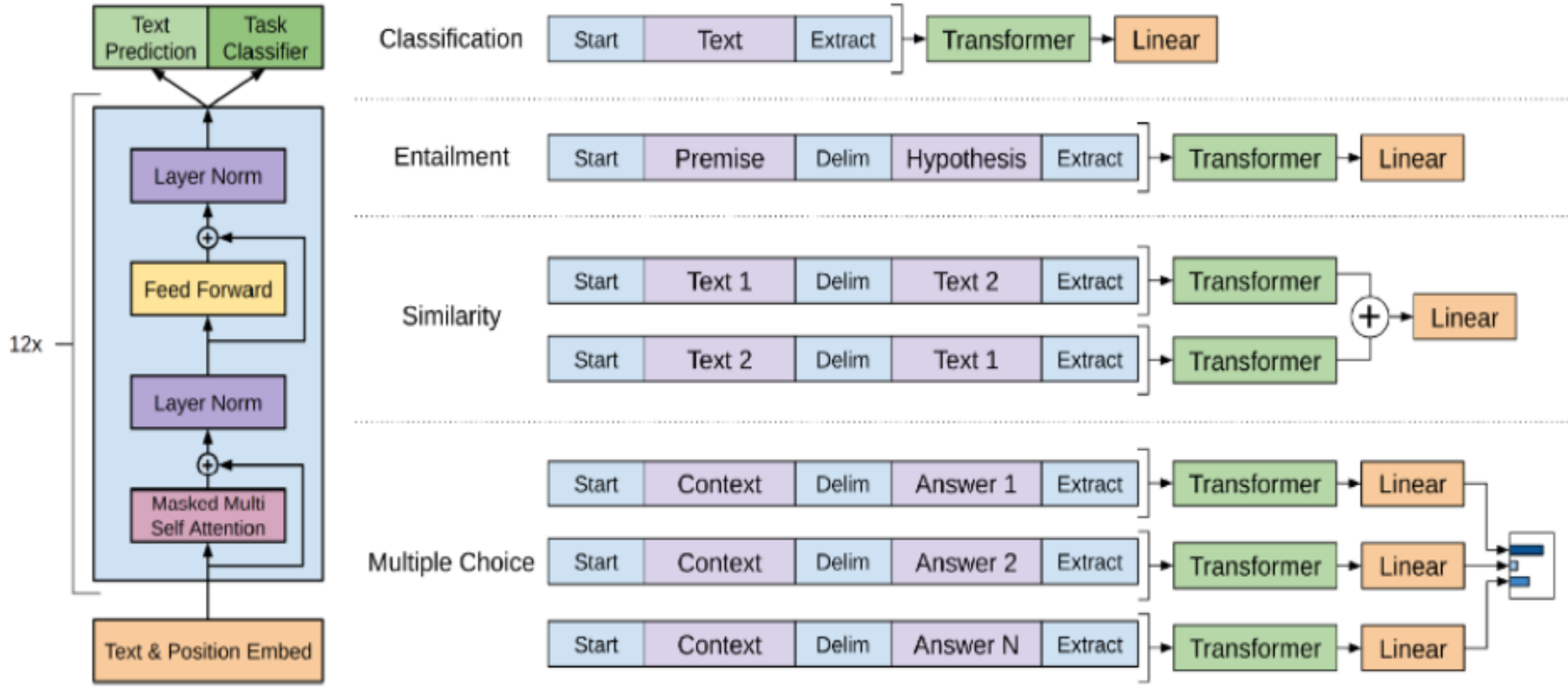
Stanford Question Answering Dataset (SQuAD)

Context	Question	Answer
The Panthers beat the Seattle Seahawks in the divisional round, running up a 31–0 halftime lead and then holding off a furious second half comeback attempt to win 31–24, avenging their elimination from a year earlier.	Who lost to the Panthers in the divisional round of the playoffs?	Seattle Seahawks

- Classification

Sentence 1	label
I am happy	Positive
I am sad	negative

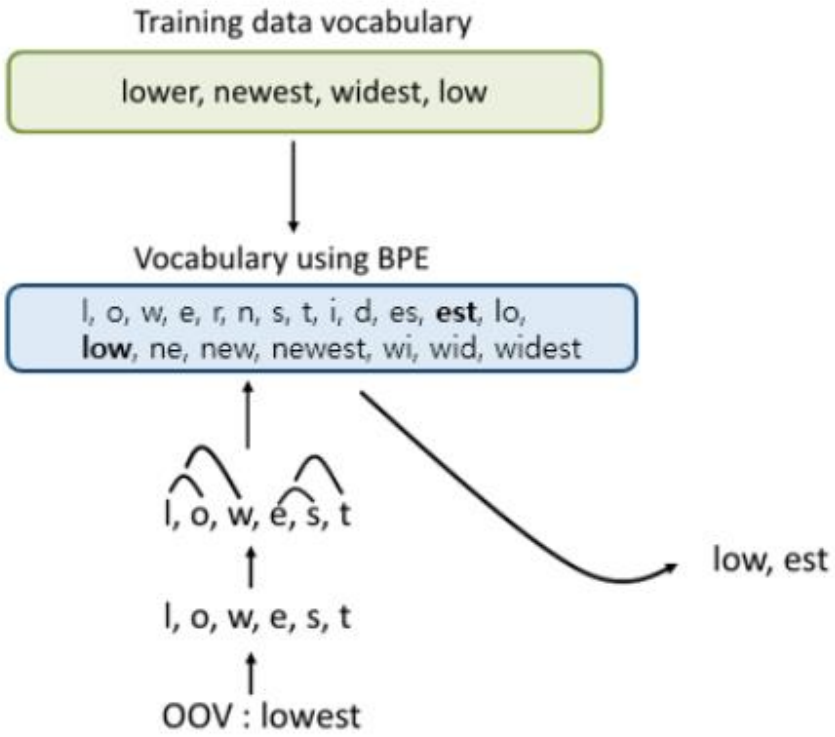
GPT-1의 4가지 task



Entailment, Similarity, Multiple Choice의 task의 경우 2개의 문장 사이에 delimeter(구분자)를 포함하여 하나의 input으로 concat

Fine-tuning과정에서 layer 추가 X, 단순히 모델 그대로 유지한 채로 레이블링된 데이터를 입력하고 최적화

Byte Pair Encoding



GPT는 기존의 Word2vec이 아닌 input token을 인코딩하기 위해

BPE(Byte pair encoding)을 사용

Word2vec

Low, lower, newest, widest라는 데이터셋이 존재한다고 가정하면, test 과정에서 lowest가 등장한다면, 해당 단어는 학습하지 않았기 때문에, 제대로 대응하지 못하는 OOV(Out-of-Vocabulary) 문제 발생

BPE는 기존에 있던 단어를 분리한다는 의미인 subword segmentation 알고리즘

Low, lower, newest, widest를 각각 바이트 즉, 글자 하나 단위로 분할 후 빈도수에 기반하여 단어들을 집합으로 묶는 메커니즘

Lowest가 들어와도, low 와 est를 보고 OOV문제 없이 인코딩이 가능함!

목차

GPT 1

GPT-1의 모델
구조 및 특징
모델 학습과정
BPE

GPT 2

모델의 특징
학습과정

GPT 3

GPT3 개요
및 모델 구조
학습과정
결과 및 한계
실습

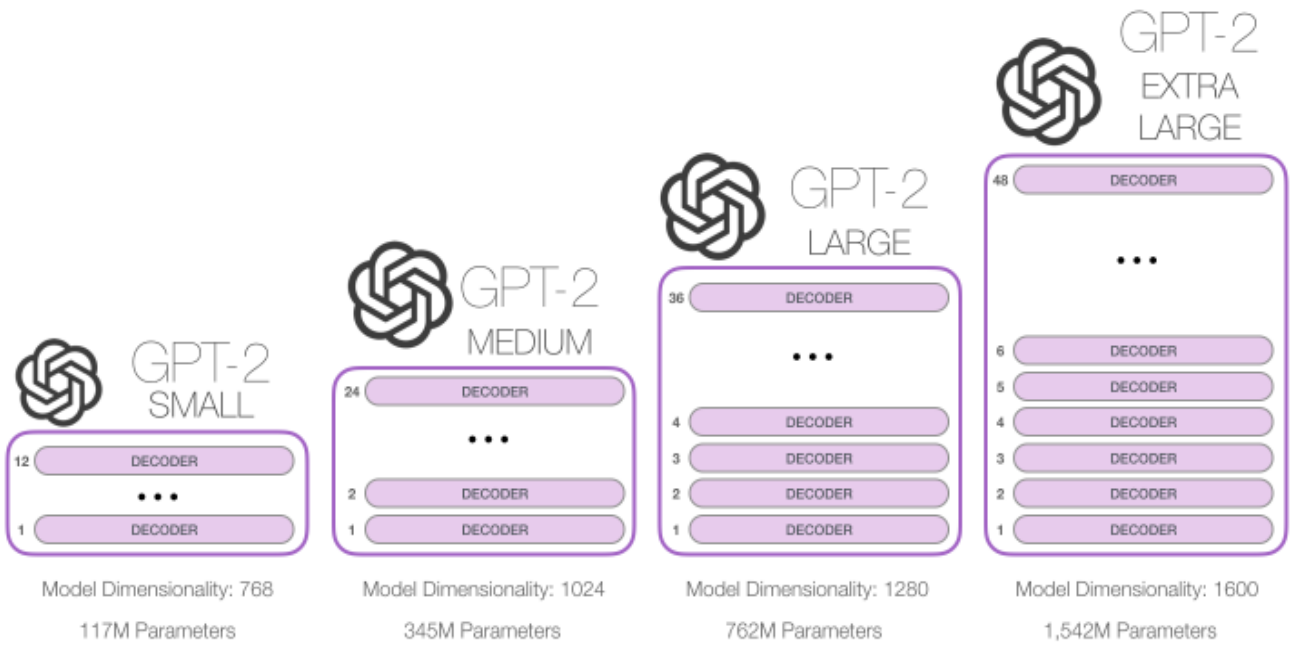
GPT-2

구조상으로 GPT1과 같은 아키텍처

차이점
1) 대용량 코퍼스 사용
2) fine-tuning 과정 제거

GPT-2

차이점 - 1) 대용량 코퍼스 사용



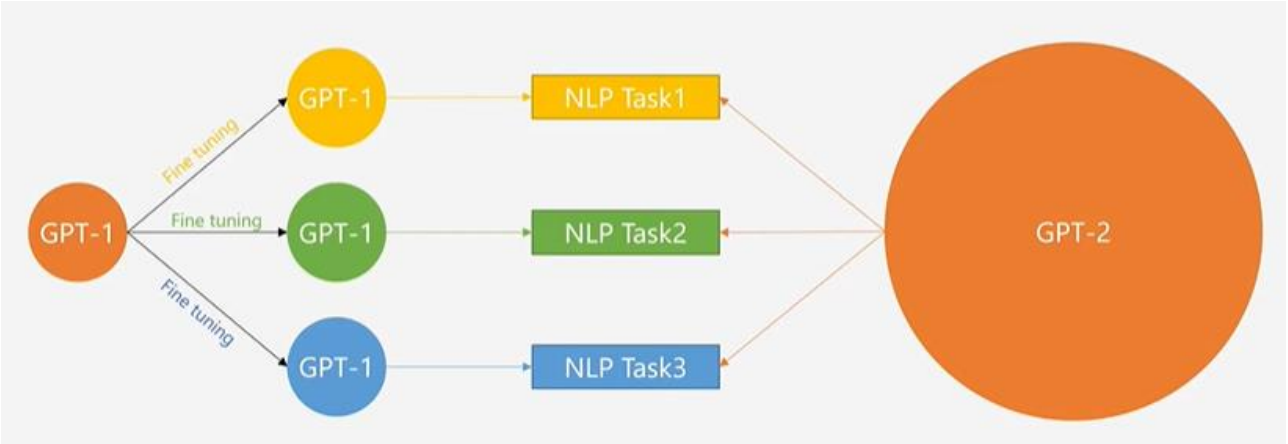
	GPT-1	GPT-2
parameters	117 millions	1.5 billions
layers	12	48
States dimension	768	1600
Context token size	512	1024
Batch size	64	512
etc		Layer normalization moved to the input of each sub block Layer normalization added after the final self attention block

GPT-2의 Pre-training에는 40GB정도의 코퍼스를 사용

GPT2 Extra Large의 경우,
모델의 **parameter 수가 15억 4,200만개**, 48개의 디코더 layer를 두고 **1600차원의 model dimensionality**로 임베딩

GPT-2

차이점 - 2) fine-tuning 과정 제거



기존의 GPT1은 pre-training 후, task에 맞는 fine-tuning을 거쳐야 했지만,

GPT2의 경우, 이러한 fine-tuning의 과정 없이 모델 그 자체를 다양한 task에 활용 가능

GPT-1	GPT-2
P(output | input)	P(output | input , task)

GPT1과 GPT2의 P값을 도출해내는 목적함수의 차이

GPT-2

language model의 구조만으로 labeling 데이터 없이 Generative model을 학습하여 다양한 task를 수행 가능

1) 언어모델

2) 기계번역

3) 질의응답

목차

GPT 1

GPT-1의 모델
구조 및 특징
모델 학습과정
BPE

GPT 2

모델의 특징
학습과정

GPT 3

GPT3 개요
및 모델 구조
학습과정
결과 및 한계
실습

연구 동기 및 목적

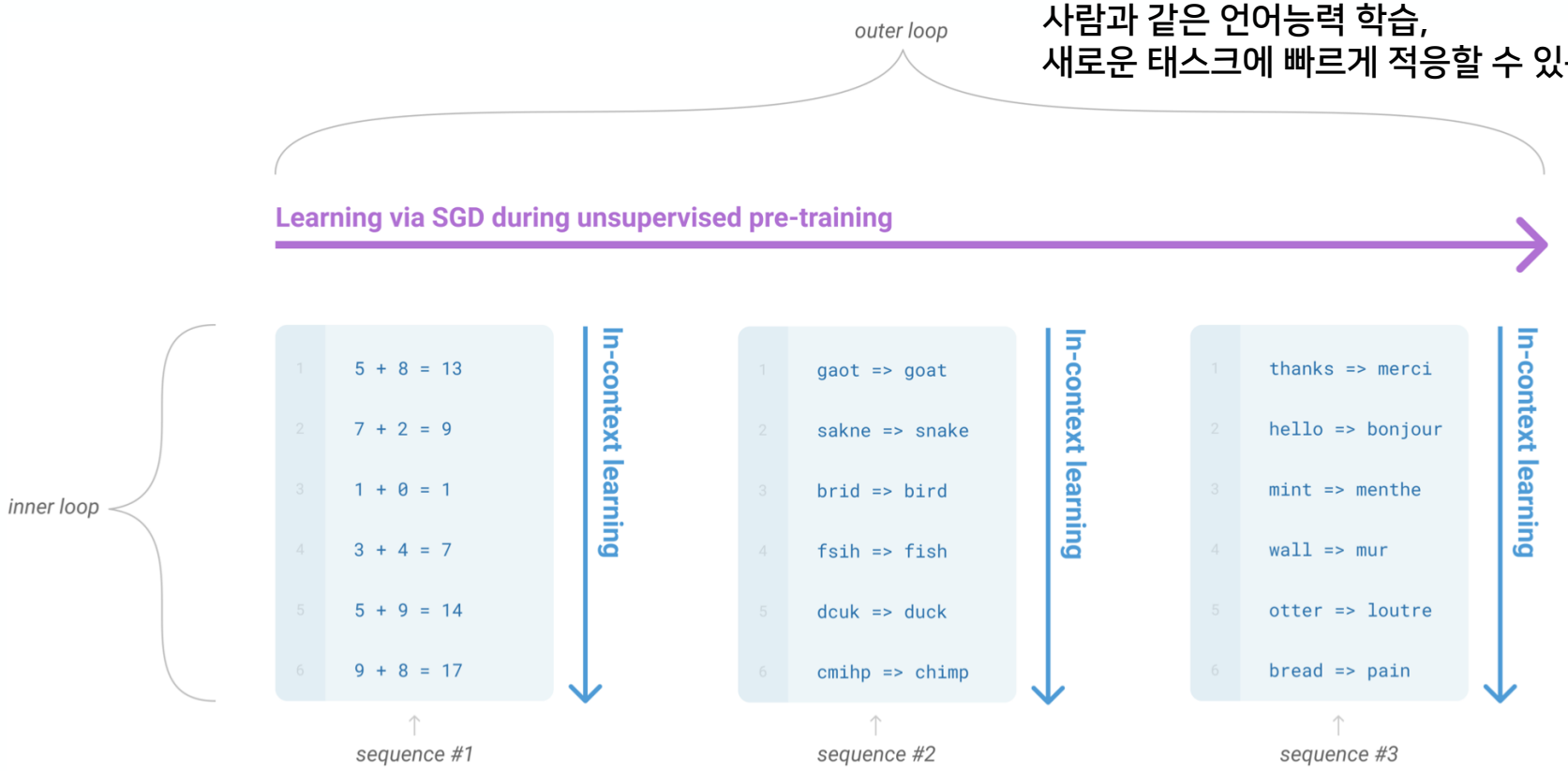
Fine Tuning을 하지 않고도 Task 를 수행할 수 있는 모델을 만들자 !

1. 새로운 task를 위해 매번 supervised training dataset을 구축하는 것이 어려우며 이는 언어모델의 적용 가능성을 제한한다.
2. 모델의 표현력과 훈련 분포의 협소함에 따라 훈련 데이터의 거짓 상관관계를 이용할 가능성이 근본적으로 증가하며, 훈련 데이터의 분포에 대해 한정 지어진 모델이 그 외의 영역은 잘 일반화하지 못한다.
3. 인간은 대규모 데이터셋을 요구하지 않는다 .

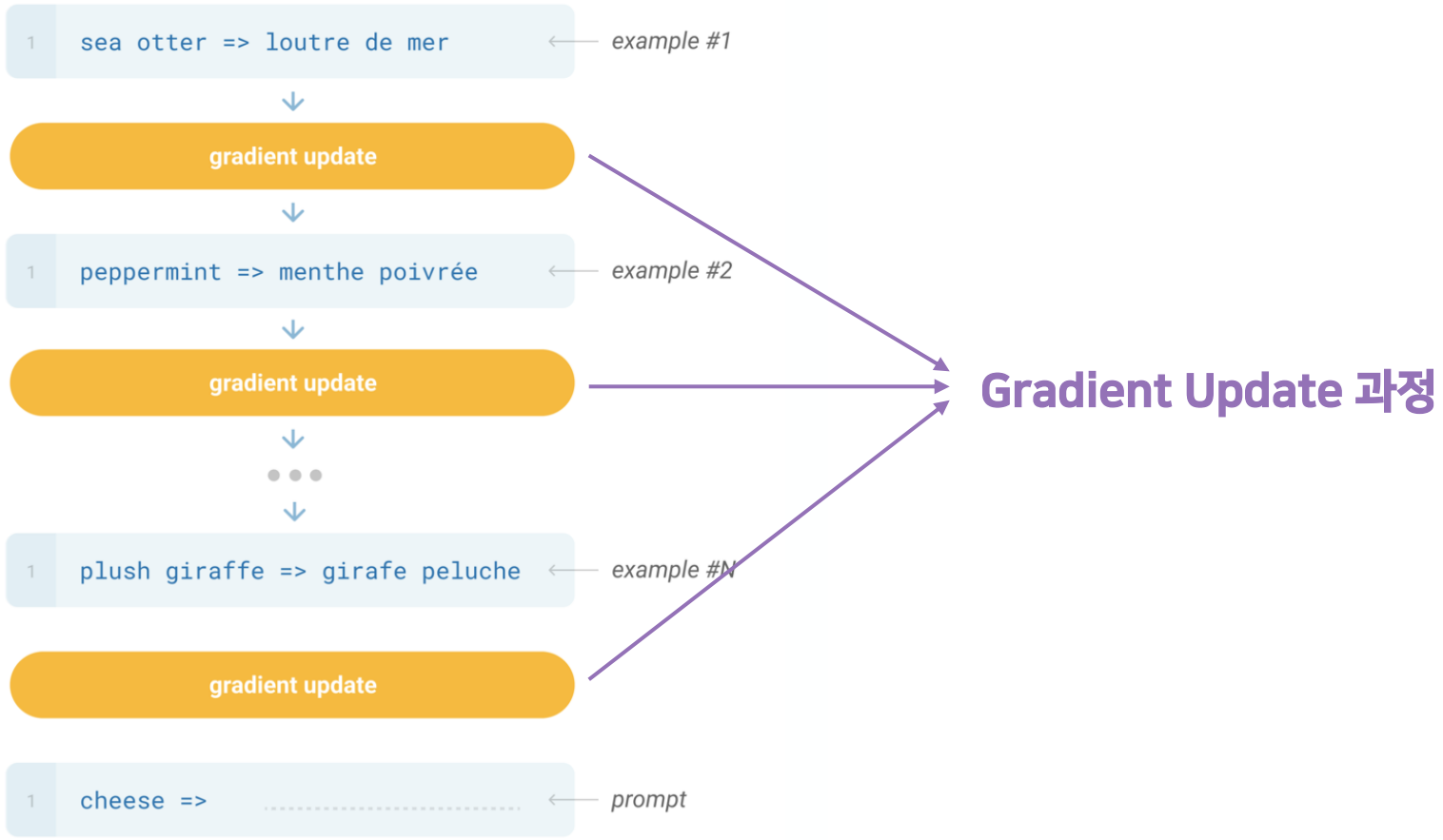
GPT-3: https://arxiv.org/pdf/2005.14165.pdf

Meta-Learning

사람과 같은 언어능력 학습,
새로운 태스크에 빠르게 적응할 수 있는 능력을 기른다.



Fine-Tuning


sea otter => loutre de mer
example #1
gradient update
peppermint => menthe poivrée
example #2
gradient update
plush giraffe => girafe peluche
example #N
gradient update
cheese =>
prompt
Gradient Update 과정

Zero, One, Few-shot

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

example 0개

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

example 1개

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée        ←
4  plush girafe => girafe peluche      ←
5  cheese =>                           ← prompt
```

example 10~100개

예시를 통한 Gradient Update 과정 없음 !

Fine-tuning이 아닌 in-context learning

Example

```
// Here are the 2 description:code pairs used to give GPT-3
some context for how to provide a response

// sample 1
description: a red button that says stop
code: <button style={{color: 'white', backgroundColor:
'red'}}>Stop</button>

//sample 2
description: a blue box that contains 3 yellow circles with
red borders
code: <div style={{backgroundColor: 'blue', padding: 20}}><div
style={{backgroundColor: 'yellow', border: '5px solid red',
borderRadius: '50%', padding: 20, width: 100, height: 100}}>
</div><div style={{backgroundColor: 'yellow', borderWidth: 1,
border: '5px solid red', borderRadius: '50%', padding: 20,
width: 100, height: 100}}></div><div style={{backgroundColor:
'yellow', border: '5px solid red', borderRadius: '50%',
padding: 20, width: 100, height: 100}}></div></div>
```

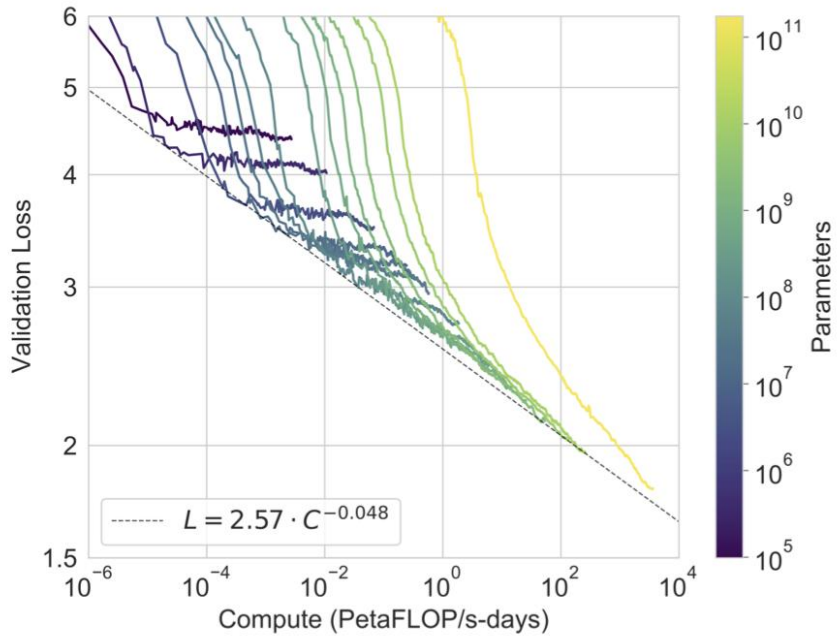

다른 예시가 궁금하다면? https://littlefoxdiary.tistory.com/45

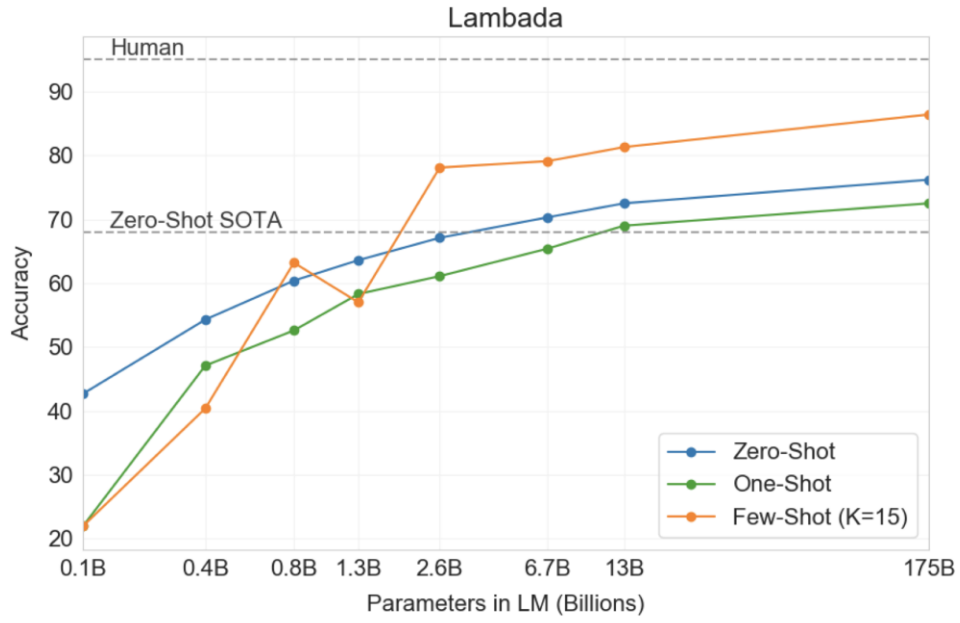
video: https://twitter.com/sharifshameem/status/1282676454690451457

가 설

언어모델의 크기와 학습 loss는 power-law 관계를 가진다.

언어모델의 크기가 커지면 In-context Learning Ability가 좋아질 것이다.





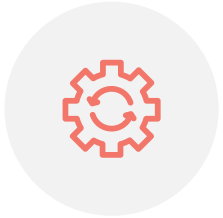
모델, 데이터셋, 학습 과정

Model

- GPT-2와 동일한 모델 사용 but 모델 크기가 125M에서 175B로 커짐
- 수정된 initialization, pre-normalization, reversible tokenization
- Transformer layer에서 alternating dense와 locally banded sparse attention을 사용

Training Dataset

- Common Crawl, WebText2, Books1, Books2, Wikipedia (450TB -> 570GB)
- 모든 데이터를 사용하지 않음.



Train Process

- 입력 크기는 2048 token
- 만약 짧은 문서라면 바로 다음 문서를 이어붙이고 중간에 delimiter를 사용
- 큰 모델 → 큰 배치사이즈, 작은 learning rate

실험 결과. 뉴스 기사 생성

	Mean accuracy	95% Confidence Interval (low, hi)	t compared to control (p-value)	"I don't know" assignments
Control	88%	84%–91%	-	2.7%
GPT-3 175B	52%	48%–57%	12.7 (3.2e-23)	10.6%

Table 3.12: People's ability to identify whether $\sim$ 500 word articles are model generated (as measured by the ratio of correct assignments to non-neutral assignments) was 88% on the control model and 52% on GPT-3 175B. This table shows the results of a two-sample T-Test for the difference in mean accuracy between GPT-3 175B and the control model (an unconditional GPT-3 Small model with increased output randomness).

실험 결과. 새로운 단어로 문장 생성

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:
We were traveling in Africa and we saw these very cute whatpus.

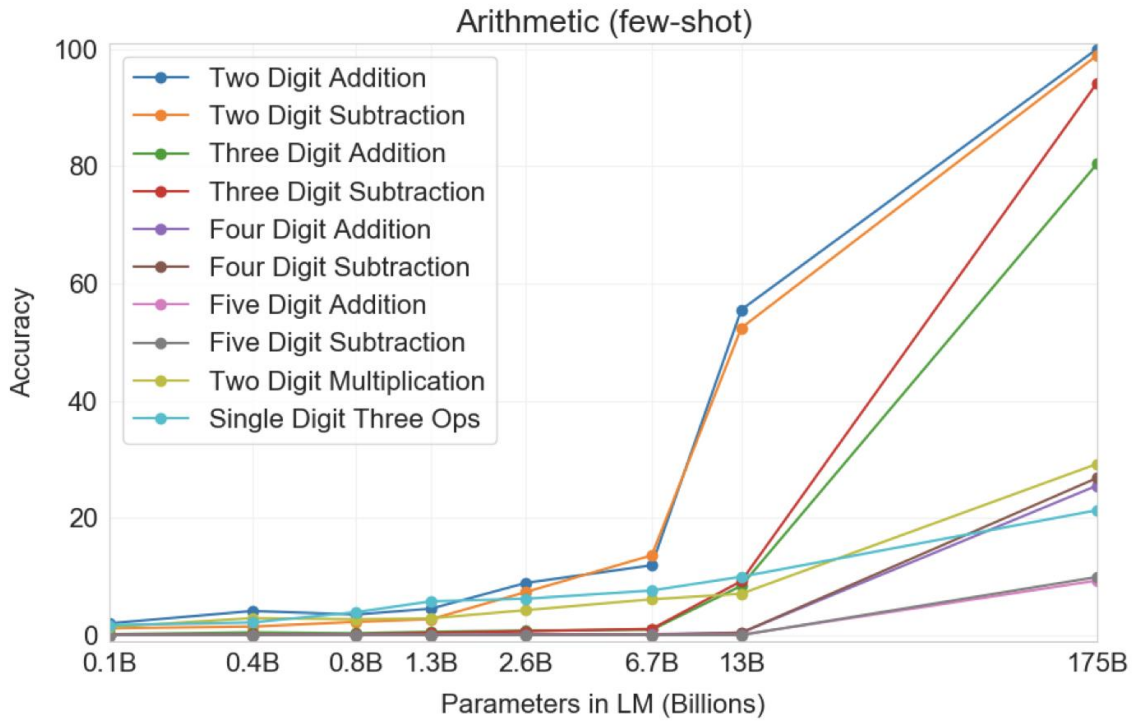
To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:
One day when I was playing tag with my little sister, she got really excited and she started doing these crazy farduddles.
A "yalubalu" is a type of vegetable that looks like a big pumpkin. An example of a sentence that uses the word yalubalu is:
I was on a trip to Africa and I tried this yalubalu vegetable that was grown in a garden there. It was delicious.

A "Burringo" is a car with very fast acceleration. An example of a sentence that uses the word Burringo is:
In our garage we have a Burringo that my father drives to work every day.

A "Gigamuru" is a type of Japanese musical instrument. An example of a sentence that uses the word Gigamuru is:
I have a Gigamuru that my uncle gave me as a gift. I love to play it at home.

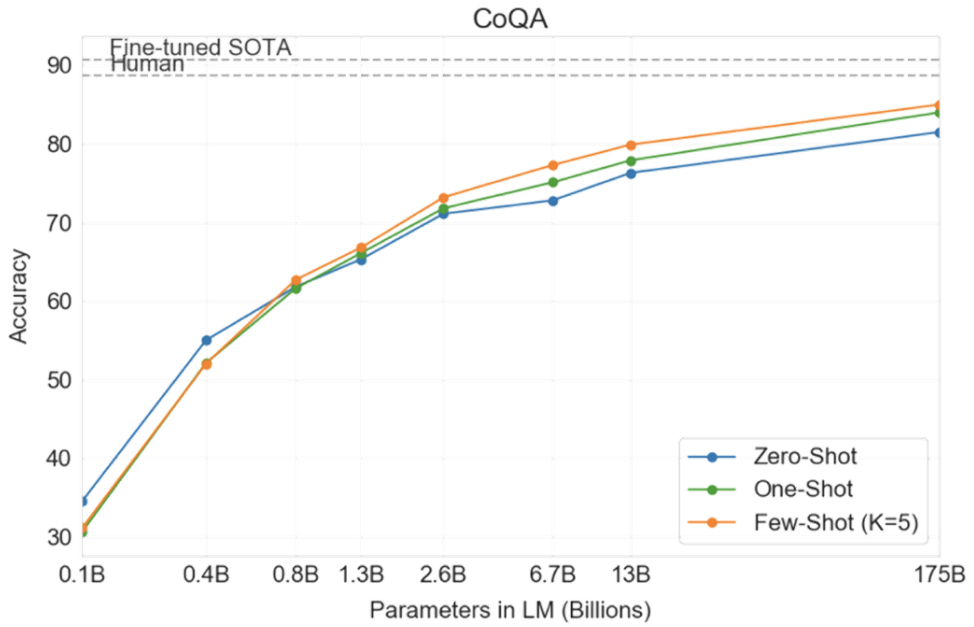
To "screeg" something is to swing a sword at it. An example of a sentence that uses the word screeg is:
We screeghed at each other for several minutes and then we went outside and ate ice cream.

실험 결과. 연산


Arithmetic (few-shot)
Two Digit Addition
Two Digit Subtraction
Three Digit Addition
Three Digit Subtraction
Four Digit Addition
Four Digit Subtraction
Five Digit Addition
Five Digit Subtraction
Two Digit Multiplication
Single Digit Three Ops
Accuracy
100
80
60
40
20
0
0.1B
0.4B
0.8B
1.3B
2.6B
6.7B
13B
175B
Parameters in LM (Billions)

실험 결과. 기계 독해

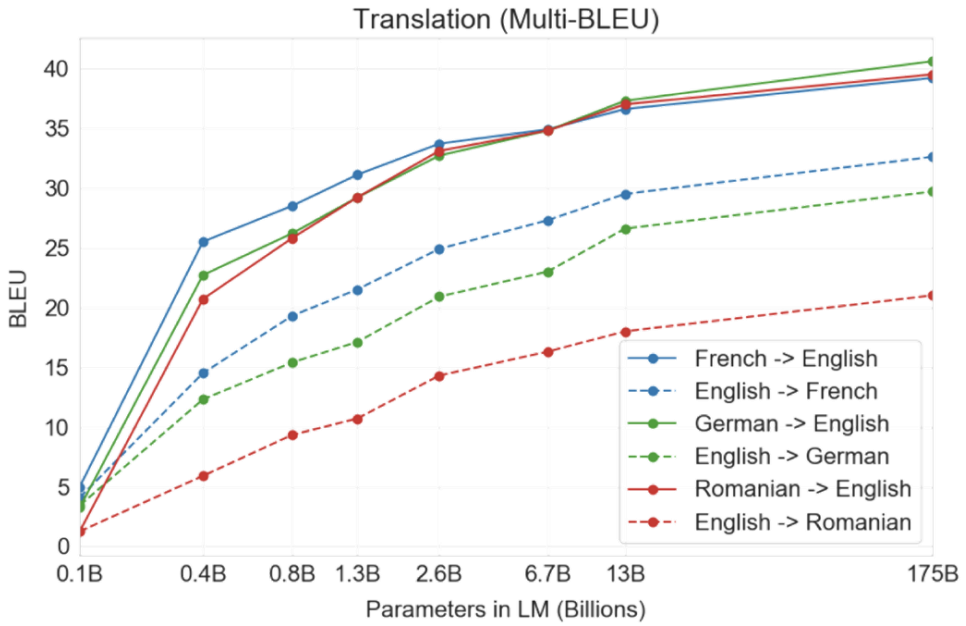
질문과 단락이 주어졌을 때, 정답을 인식하는 문제



Setting	CoQA	DROP	QuAC	SQuADv2	RACE-h	RACE-m
Fine-tuned SOTA	**90.7**[a]	**89.1**[b]	**74.4**[c]	**93.0**[d]	**90.0**[e]	**93.1**[e]
GPT-3 Zero-Shot	81.5	23.6	41.5	59.5	45.5	58.4
GPT-3 One-Shot	84.0	34.3	43.3	65.4	45.9	57.4
GPT-3 Few-Shot	85.0	36.5	44.3	69.8	46.8	58.1

실험 결과. 번역

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	**45.6**[a]	35.0 [b]	**41.2**[c]	40.2[d]	**38.5**[e]	**39.9**[e]
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ+19]	37.5	34.9	28.3	35.2	35.2	33.1
mBART [LGG+20]	-	-	29.8	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	39.2	29.7	40.6	21.0	39.5



한 계

1 성능적인 한계

- 문서 단위로 의미론적인 반복이 있었고, 충분히 긴 문장에서는 일관성을 잃거나, 모순이 발생했다.
- 상식 수준의 물리학 (common sense physics) 도메인을 잘 하지 못했다. 가령, 치즈를 냉장고에 넣으면 녹을까요?와 같은 질문에 잘 대답하지 못했다.
- In-context learning의 경우에도, WIC(두 단어가 한 문장에서 동일한 방식으로 사용되는지), ANLI(한 문장이 다른 문장을 의미하는지)와 같은 task에서는 큰 성능 향상이 없었다.

2 알고리즘적 한계

- 양방향성 X (auto-regressive에만 집중했기 때문)
- 모델의 규모를 키우는 것은 사전학습의 손실함수의 한계에 도달할 수 있다.
- 사전학습된 대규모 언어 모델은 실제 세상과의 상호작용이 없기 때문에 세상에 대한 context가 부족하다.

3 학습 데이터의 효율성

- GPT-3는 사전 학습동안 사람이 평생 보는 것보다도 더 많은 텍스트를 보지만, 사람만큼의 성능을 내지 못한다.

4 Few-shot learning의 해석 불가능함

- inference 과정에서 scatch로부터 패턴을 학습해내는건지, 아니면 트레이닝하는 과정에서 들어간 패턴들이 memorize 되었다가 나오는건지 모른다.
- 어떻게 working 하는지 알 수 없다.

[실습]

REFERENCES

논문

- https://arxiv.org/pdf/2005.14165.pdf

블로그

- https://littlefoxdiary.tistory.com/44
- https://ai-information.blogspot.com/2020/06/nl-069-language-models-are-few-shot.html
- https://jiho-ml.com/weekly-nlp-29/
- https://hyyoka-ling-nlp.tistory.com/8

유튜브

- https://www.youtube.com/watch?v=MV1l2JPAxVY
- https://www.youtube.com/watch?v=FeEmmylAF0o

실습 코드 관련

- https://github.com/NLP-kr/tensorflow-ml-nlp-tf2-colab
- <텐서플로2와 머신러닝으로 시작하는 자연어처리>

감사합니다