**Language Models are Unsupervised Multitask Learners (2018, Radford)**

[**https://hipgyung.tistory.com/25**](https://hipgyung.tistory.com/25)

[**https://greeksharifa.github.io/nlp(natural%20language%20processing)%20/%20rnns/2019/08/28/OpenAI-GPT-2-Language-Models-are-Unsupervised-Multitask-Learners/**](https://greeksharifa.github.io/nlp(natural%20language%20processing)%20/%20rnns/2019/08/28/OpenAI-GPT-2-Language-Models-are-Unsupervised-Multitask-Learners/)

[**https://ai-information.blogspot.com/2019/02/language-models-are-unsupervised\_21.html**](https://ai-information.blogspot.com/2019/02/language-models-are-unsupervised_21.html)

**Abstract**

WebText라는 수백만 데이터셋으로 LM 학습

LM의 capacity는 zero-shot task에 필수적: 좋은 LM은 pre-train 시에 보지 못했던 task에 대해 적용, test가 가능해야함.

GPT2는 1.5B Params, 8개 LM Dataset 중 7개에서 SOTA

제로샷..?

**Introduction**

기계학습은 큰 dataset과 고용량 모델 & 지도학습으로 발전 -> But 불안정이 남아 있고, 매우 좁은 범위의 문제에서만 능력 발휘 -> 데이터셋 수동 분류 없이 범용 모델이 필요

최근에 들어서야 GLUE benchmark 등 넓은 범위에 해당하는 dataset이 있다. 이전까지는 single task dataset -> 좁은 범위 과제에서만 효과적

다중 학습에서는  [Learning and Evaluating General Linguistic Intelligence](https://arxiv.org/abs/1901.11373) 와 같은 연구가 성능 향상을 이뤘지만 역시 수십만 개 \* each task 의 데이터(labeled?)가 필요하기 때문에 아직도 고민이 많다.

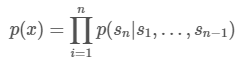
따라서 GPT2는 GPT1처럼 pre-training + supervised fine-tune을 이용하되 추가로 zero-shot setting을 이용하였다.

Supervised learning은 특정 task에서 성능 높이는데 필요하다.

**Approach**

핵심은 언어 모델링!

일련의 symbol(s1...sn)으로 구성된 비지도 분포 추정?!



트랜스포머는 이런 조건부 확률 잘 계산?

멀티 task에 대해서는 각 task에 대한 표시가 필요? 

**Training Dataset**

많은 선행 연구에서는 뉴스와 같은 한 영역의 데이터로 구성됨. -> 가능한 다양한 출처로부터 가져오는 것이 더 좋을 것이라 판단.

따라서 Common Crawl 같은 크롤링(web scraping 자료)가 좋을 것이라 판단 -> 하지만 unintelligible, 품질 떨어지는 데이터 많다.

-> 따라서 크롤링을 조금 변형시킴

1. 사람에 의해 필터링된 글만 사용. (Reddit에서 3 karma 이상의 글만 가져옴., 결과적으로 45M개의 링크)

2. 이 데이터셋 이름은 WebText

3. 링크에서 글 추출 위해서 Dragnet & newspaper 내용추출기 사용

4. 2017/12 이후의 글 & 위키피디아 글 제거, 중복제거 -> 8M개의 문서, 40GB 텍스트 확보 (위키피디아는 다른 dataset에서 흔하며, 학습과 평가 & 측정 단계에서 데이터가 겹치는 문제로 인해 분석이 복잡해질 수 있다.)

**Input Representation**

유니코드란 아스키 코드의 한계를 보완, 아스키 코드는 알파벳에만 한정됨 -> 모든 언어를 통합해 인코딩한 코드, 2바이트(2^16의 경우의 수 커버)

UTF8은 유니코드를 다시 한번 가변 길이로 인코딩 하는 것 같다. (전세계 공통)

BPE를 이용한다. Character와 word level의 중간 지점.

워드 레벨 LM의 경험적 편익 + 바이트(char) 레벨 접근법의 일반성을 결합 가능

어떤 유니코드 문자열에도 확률 할당 가능 -> pre-processing, tokenization, vocab size 관계 없이 모든 데이터셋에서 GPT2 LM을 평가 가능

현재 다른 대규모 언어모델은 소문자화, 토큰화, 모델링 가능한 문자열이 차지하는 공간을 제한하기 위한 사전 외 token과 같은 전처리 과정 거침.

Unicode 문자열을 UTF-8 형식으로 처리하는 것은 이를 만족?

Byte 수준 언어모델은 대규모 dataset에서 단어수준 언어모델에 뒤떨어짐..?

WebText에서도 그러한 성능차이 확인..?

BPE 알고리즘의 기본 원리는 가장 많이 등장한 문자열에 대하여 병합하는 작업을 반복 -> 원하는 단어 집합의 크기(단어의 개수)가 될 때까지 반복

BPE는 dog의 다양한 형태 dog., dog!, dog? 등을 가짐.

이 부분 다시, 유니코드?

**Model**

기존의 GPT 모델과 비슷하다. 차이점은 아래.

마지막 셀프 어텐션 이후의 normalization 레이어 추가.

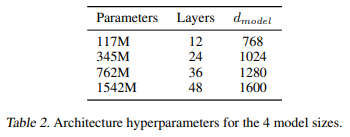
Normalization 레이어를 각 하위 블록의 입력으로 이동.

Context size를 512 -> 1024 tokens로 batchsize도 512로 좀 더 크게 사용

사전은 50257개로 확장

모델 깊이에 따른 residual path의 누적에 대해 N이 residual layer 수라 할 때, residual 가중치에  곱해서 사용.

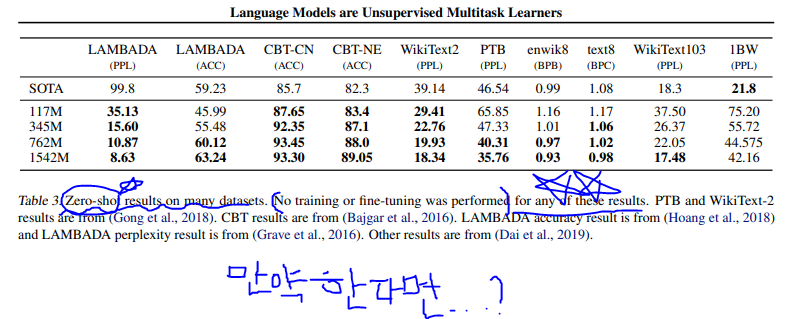
다른 시나리오에 맞추기 위해 다른 매개 변수가 있는 4개의 모델 학습



WebText의 5%를 떼서 만든 held-out sample을 사용해 수동 조정. (learning rate?)

모든 모델은 WebText에 대해 underfit -> 더 오래 학습시키면 더 좋은 성능?!

**Experiments**



가장 작은 -> GPT1, 2번째 -> BERT 크기와 동일, 가장 큰 -> 15억 개 파라미터

8개중 7개 SOTA, 특히 소규모 데이터셋 Task에서 더 큰 개선

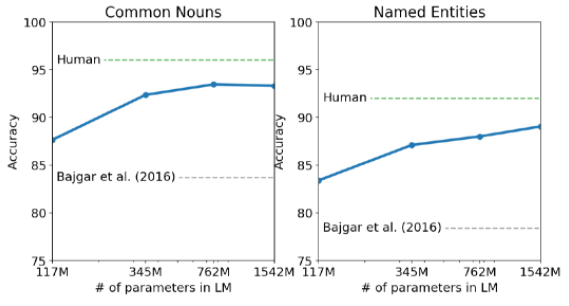
GPT2는 byte level -> 전처리, 토큰화 등 손실이 필요하지 않다. -> 어떤 언어 모델 Benchmark도 사용 가능.

평가는 로그 확률을 계산?? 일반 분포를 심하게 벗어난 규격화 된 텍스트, 분리된 구두점, 축약형 문장 등에 대해 평가 받음, <UNK>는 WebText에 400억 byte중 26번 밖에 등장하지 않음.

데이터셋 종류

Children’s Boot Test

품사 태깅하는 데이터셋?

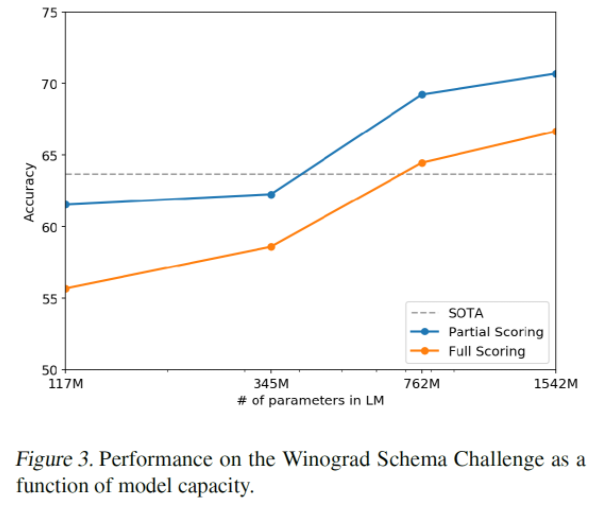


LAMBADA

Long-range dependencies 평가 데이터셋

Winograd Schema Challenge

텍스트에 중의성을 해석하는 일반 상식 추론 능력 평가 데이터셋



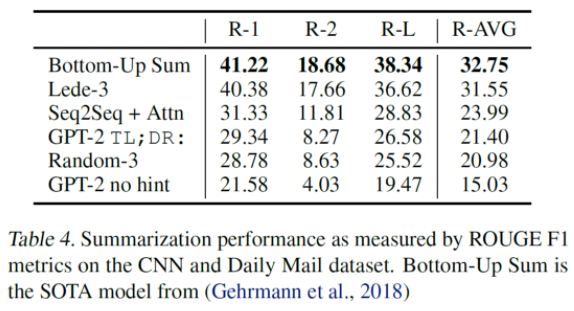
Reading Comprehension

CoQA을 7개 분야에서 가져온 질문자-답변자 대화 데이터셋, 지도 학습 없이 55 F1 Score, 고무적인 일, BERT는 89 F1 Score

독해 능력과 대화 기반 평가 데이터셋

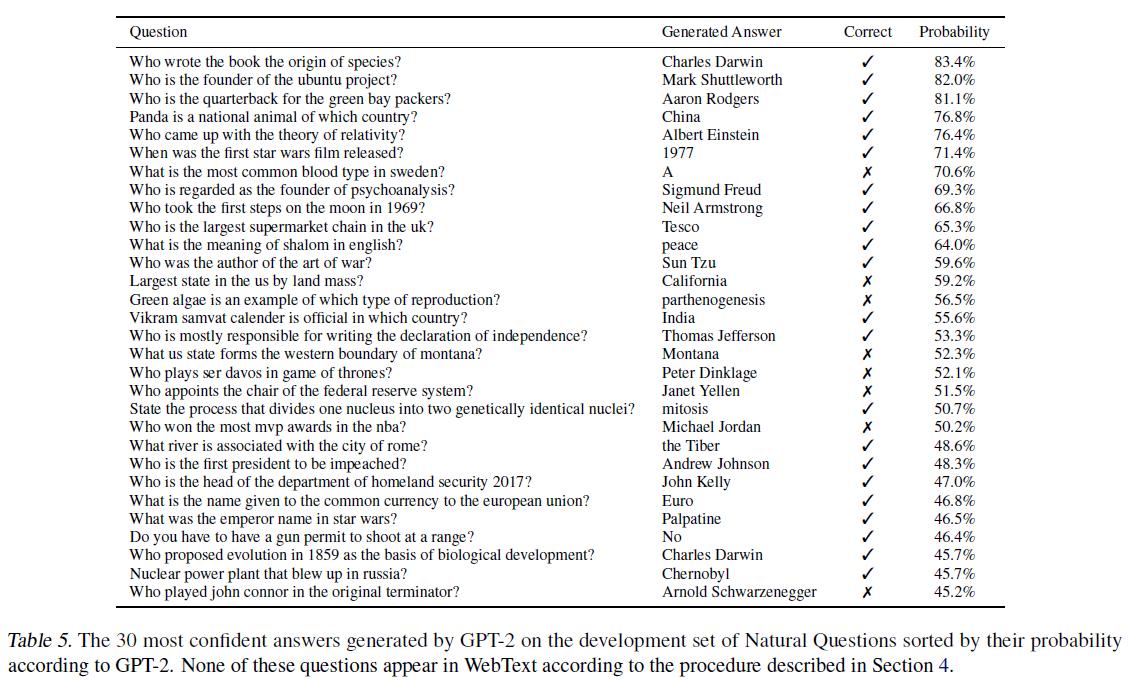
Summarization

Daily Mail Dataset



Translation은 좋지 못한 결과

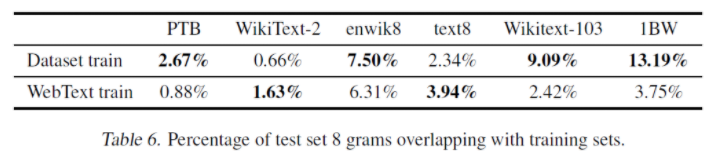
Question Answering



하지만 상용화 하기에는 역부족?(열린 분야 질답 시스템에 비하면 성능 떨어짐.)

테스트의 일반화를 위하여 트레인 셋 vs 테스트 셋 의 데이터 중복은 없을수록 일반화된 모델이 된다.

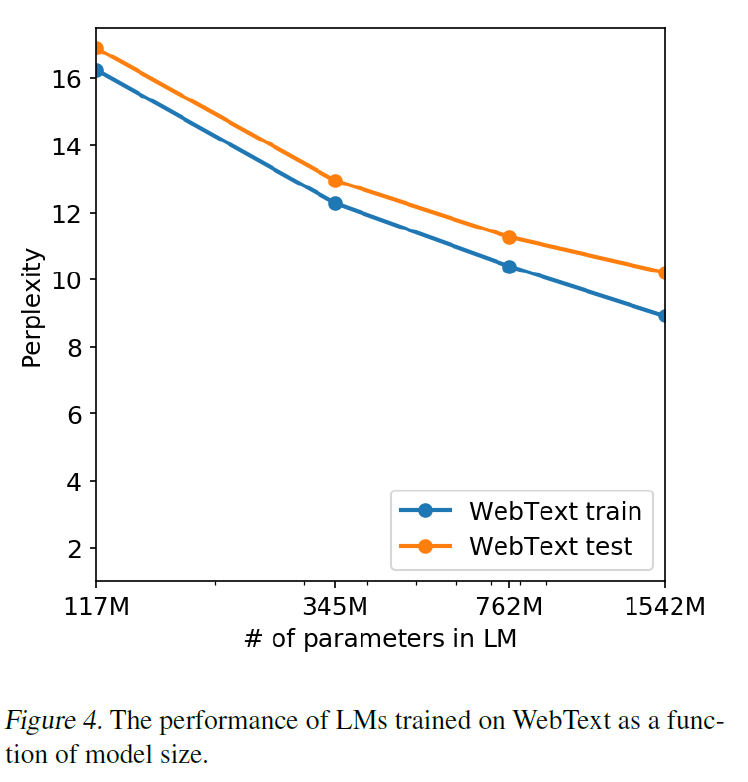
그런 overlap 평가를 위한 Bloom 필터 사용



WebText는 좋은 데이터라고 판단. 다른 데이터셋의 경우 겹침이 높음

-> 즉 n-gram overlap을 사용하는 것은 좋은 방법

또한 모델이 overfit 됐는지 평가하기 위해서 WebText에서 이미 학습 데이터로 썼던 것(train)과 그렇지 않은 것(test)로 평가했을 때 오히려 모두 underfit인 것을 볼 수 있고 train, test에 대한 PPL이 차이가 크지 않다는 것을 보아 overfit은 아님을 확인할 수 있다.



**Discussion**

연구 거리 아직 많다.

Zeroshot이 요약 등에서는 아직 기본적인 성능(독해에선 좋더라도), 여전히 실 사용은 무리.

미세조정을 하는 것에 대한 가능성도 있다.

BERT에서 언급한 단방향 표현의 비효율성을 극복할 만한 데이터셋 크기가 아니다.