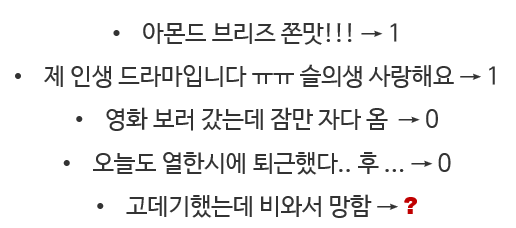
<https://littlefoxdiary.tistory.com/44>

<https://mangastorytelling.tistory.com/entry/Minsuk-Heo-GPT3-%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0>

**Language Models are Few-Shot Learners**

이전에 BERT, GPT1, GPT2(zero-shot 이지만 성능 더 높이려면 fine-tune 필요), ULMfit 등 task에 따라 매번 fine-tuning이 필요했다. => 높은 정확도의 장점이자 한계점



반면에 우리 인간은 몇 가지 예시, 위의 4개 문장만 보고 학습해도 답을 맞출 수 있다. = few-shot learning(메타 러닝 중 한 개)

제로 샷은 해당 task에 대해 배우지 않아도 그 task에 대해 적응하는 것이고.(단 사전훈련 데이터에 그 task에 관한 데이터가 있을 수는 있다.)  
퓨 샷은 해당 task에 대해 아주 조금만 배워 그 task에 적응하는 것이다.

하지만 BERT, GPT 모두 대량의 예제의 fine-tune이 필요하다.

GPT3는 이런 점을 파고든다.

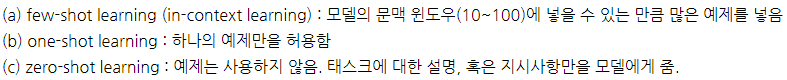
왜 그렇다면 이런 한계를 넘어서는 것이 중요한가?

1. 새 task 풀 때마다 현재는 많은 labeled 데이터가 필요한데 몇 가지 예제만으로 적응할 수 있다면, 문법 교정/ 생성 요약/ 비평문 쓰기등 레이블링 데이터 만들기 어려운 흥미로운 영역까지(즉, 까다로운 task)에 확장 가능.
2. Pretrain -> finetune 방법은 대량의 지식을 흡수한 뒤 아주 작은 task의 분포를 배우는 것이다. 따라서 finetune된 모델이 out of distribution 문제(해당 문제에만 너무 잘 맞춤(overfit for a task))를 발생하기도 한다. 따라서 다른 영역에서는 일반화 안된다는 소리, = 그 벤치마크에서는 성적이 좋은 것처럼 보여도 사람이 볼 때는 과장된 것처럼 느낌.(이런 단점 다루는 논문 있다. 블로그 참조)
3. 사람은 대부분 새로운 언어 task 처리 시 많은 예제 데이터를 필요로 하지 않음. 위에 처럼, 따라서 이런 유연성, 일반성이 NLP 모델에서도 있어야 한다고 생각.

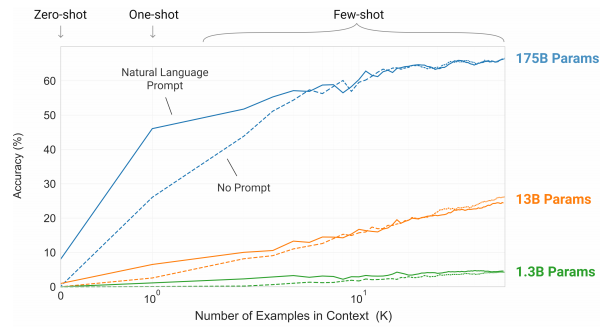
따라서 이를 극복하기 위해 meta-learning 분야가 활발히 연구. = 사전 훈련 때 몰빵하여 다양한 스킬이나 패턴 인식 => 다운스트림 task에 대해서는 빠르게 적응

GPT-2 에서는 in-context learning으로 진행(webtext에 다양한 task에 대한 데이터가 포함된 건가?) -> 사전 학습 모델에 풀고자 하는 task를 텍스트 인풋으로 넣음. -> 하지만 몇몇은 fine-tune 방식에 비해 아쉬운 성능

또한 또다른 이슈는 모델의 크기이다. 트랜스포머 -> 모델 사이즈 쉽게 늘릴 수 있다. 파라미터가 늘어날수록 downstream task에 대해 성능이 좋아졌다. -> GPT-3는 1750억 파라미터의 in-context learning 방식, 최대한 다양한 스킬 & task를 모델 파라미터에 담아야 한다.

24개의 NLP 데이터셋에 대해 3경우의 모델 성능 측정한다.  


잠깐, autoregressive language model: 알고 있는 개념임, 현재 출력 값을 다음 시점에서 사용하게 된다. (트랜스포머 디코더 구조)



단 성능 측정 시 그래디언트 업데이트 & fine-tuning은 일어나지 않음. 그냥 조금 맛보기 보여주는 느낌?

Prompt가 있을 때 성능이 높고 / few가 높을수록 성능이 높으며 / 모델이 클수록 in-context 정보(K개의 데이터) 활용 잘한다.(in-context learning: 사전 학습에 다양한, 풀고자하는 task 전부 넣어 몰빵, in-context 정보: few-shot에 쓰이는 예제)

대개 few>one>zero 의 성능이 이상적이며 사실 3개의 세팅에서 전부 좋은 성능!  
TriviaQA에서는 각각 64.3%, 68%, 71.2% 의 성능 -> 심지어 fine-tune보다 SOTA

근데 궁금한 것이 task 마다 모델의 끝부분(ex, classification은 2개, 빈칸 맞추기는 vocab 개수 일텐데 이 부분은 어떻게 조절하는 거지??)

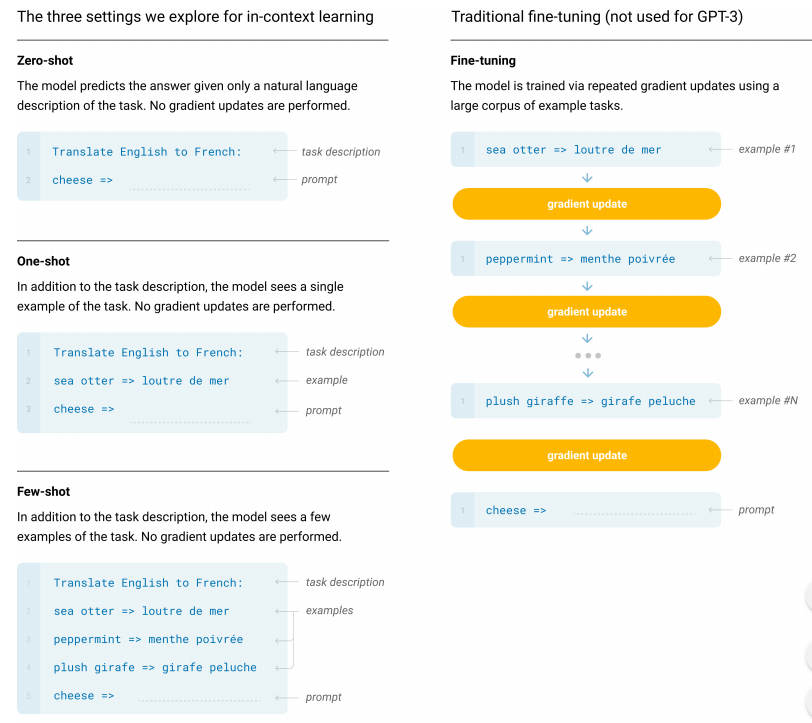
단어 순서 맞추기, 새로운 단어 사용, 3자리 수리 연산, 뉴스 생성 등 퓨샷으로 잘 수행.

하지만 ANLI, RACE, QuAC의 QA 셋에서는 감당이 안된다.

**Approach**

일단 모델, 데이터, 사전 훈련 기법 모두 GPT-2와 비슷, 하지만 평가 방법을 4가지로 세분화

1. Fine-tuning
2. Few-shot: 대부분 fine-tuning보단 성능이 떨어짐, 이런 데이터를 모델에게 예시로 보여주면된다.  
   “한국어를 영어로 번역하라: 집에 가고 싶어 -> I want go home. 배고파 -> I am hungry 치킨 사줘 -> \_\_\_”
3. One-shot
4. Zero-shot: task를 설명하는 자연어 문구만 준다. 엄청 어려운 과제, 사람도 카테고리만 주면 어려울 수 있다. 하지만 매우 편리

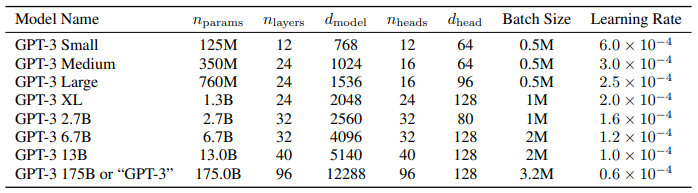


**Model & Architecture**

GPT2와 같다.

Modified initialization, pre-normalization, reversable tokenization 적용

어텐션 패턴에서 dense와 locally banded sparse attention을 번갈아 사용



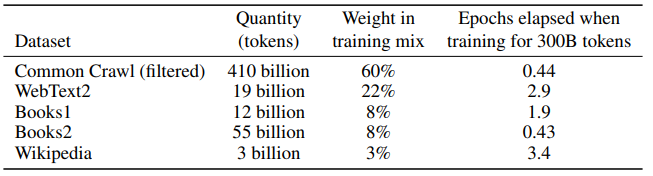
모든 모델은 BPE에 따른 3000억 토큰..?

모델 큼 -> 배치 큼 -> but learning rate는 작게

그래디언트의 noise scale을 측정해 배치 사이즈 정하는데 활용(따로 연구가 있음, 블로그 참조)

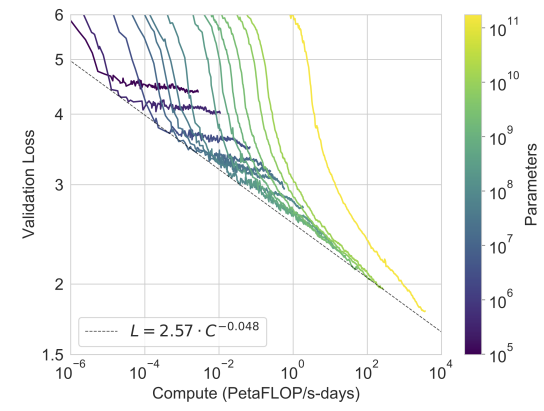
큰 모델 -> 메모리 부족 -> 행렬 곱에 있어 모델 병렬화, 레이어 사이의 모델 병렬화 섞어서 사용

**훈련 데이터셋**



아래 그림은 스케일에 따른 validation loss(사전 훈련 데이터셋도 일단 valid set은 존재하나 보다.)

모델이 클수록 컴퓨팅이 많고, Loss가 줄어든다.

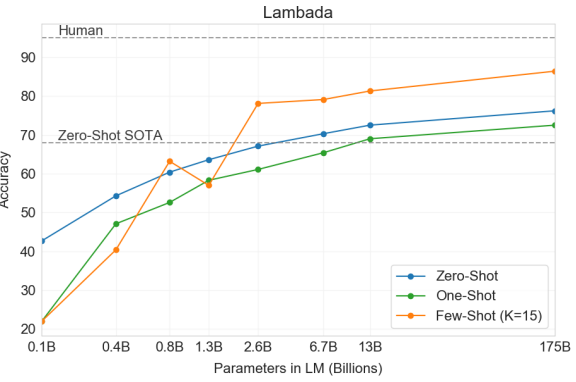


**Penn Tree Bank:** 기존 zero-shot SOTA 보다 15point 앞선 20.5 ppl

**LAMBADA:** 문장 완성/ 언어의 장기 의존성을 모델링하는 task



이전에 논점이 단순 데이터를 늘리는 것이 방도가 아니라 했지만 GPT3는 그런 방법을 포함하여 높은 결과



**HellaSwag:** 짧은 글이나 지시사항 끝맺기에 가장 알맞은 문장 고르기 task, 사람에게는 쉽지만 모델에게는 어려운 task

현 SOTA인 multi-task 학습 후 fine-tuning은 못이김(ALUM은 못이김)

**StoryCloze:** 다섯 문장의 긴 글을 끝맺기에 적절한 문장 고르는 task

퓨샷(K=70)으로 87.7% -> Bert보단 4%정도 낮지만 zero-shot 보다는 10% 높다.

**Closed book QA**

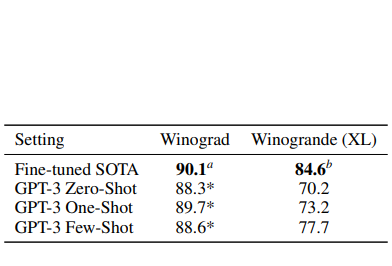
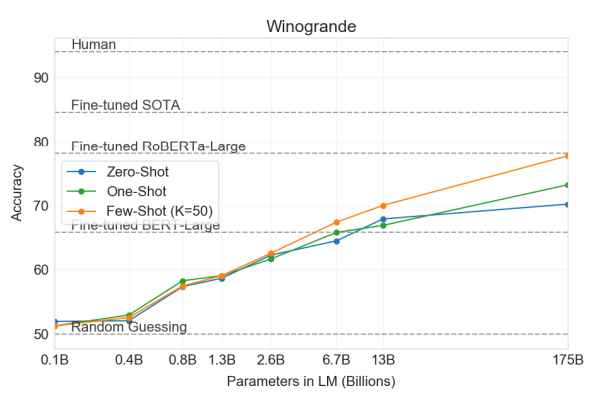
=> 폭 넓은 사실 기반의 지식에 대한 질문에(질문이 정말 포괄적) 답변을 잘 할까?  
=> open-domain QA 형태의 task에 대해 closed-book test

1. TriviaQA: T5-11B의 fine-tune 넘음  
2. WebQuestions  
3. Natural Questions

**번역은?**

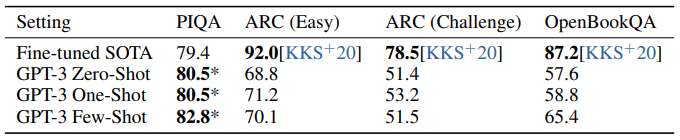
Common Crawl 데이터셋 -> ‘텍스트 질’ 필터만 적용 -> 7%의 기타언어(영어 제외) 학습 가능  
-> fine-tune SOTA보다 (불어->영어, 독어->영어)에서 좋은 성능..

**Winograd-syle task: 대명사 지칭**



Winograd는 in-context learning 많이 못함, but 데이터가 커진 winogrande에서 하니까 잘됨. 여기서는 in-context learning 잘했나..?

**Common sense reasoning**

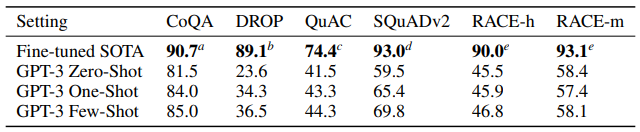


PIQA에서는 이김(하지만 data contamination issue 논쟁)

4지 선다형 문제인 ARC에서는 모두 SOTA 아래

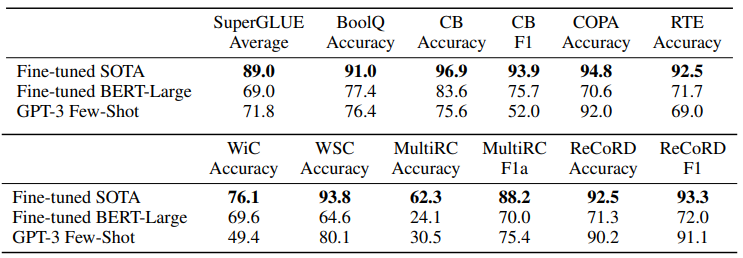
OpenBookQA는 zeroshot보다 fewshot이 높음.(SOTA 아래)

**기계 독해**

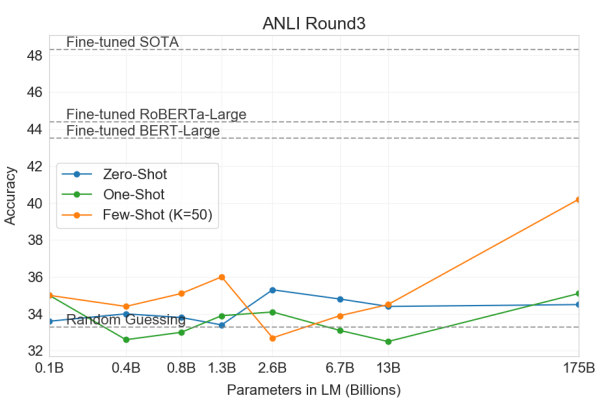


SOTA 아래

**SuperGLUE**



**NLI**



**Synthetic & Qualitive Task**

간단한 계산 추론 & 특이한 task에도 적용 가능 (자세한건 블로그 참조)

**모델의 오버핏 가능성(GPT-3가 사전학습 때 벤치마크 task 외웠다면..?)**

사전학습 데이터는 인터넷에서 구함 -> 벤치마크 테스트셋과 중복될 가능성이 있다. = 테스트 셋 오염(data contamination), 중요한 연구 분야

하지만 오염률이 그렇게 크지 않다면 성능에 크게 영향이 없다.

따라서 GPT3가 너무 커서 오버핏은 절대 없으며 오염 현상이 조금 있더라도 영향이 없을 것이라 예상

또한 13-gram overlap 되는 데이터를 삭제하는 ‘클린’ 테스트셋 만들려 했다.

조사한 결과 삭제된 데이터 때문에 모델의 성능이 높아진 근거도 없었고,

오염된 데이터가 많은 task에 대해서도 모델의 성능에는 큰 영향이 없었다.

=> 테스트셋을 외운 것이 아니다.

**모델의 한계**

1. 성능적 한계: 여전히 잘 못푸는 task 존재, ex) 물리학 일반상식을 잘 못함, “치즈를 냉장고에 넣어 놓으면 녹나?” -> 답 잘 못함.  
   생성에 있어 문단 레벨에서 동어 반복 현상 -> 긴 글 생성 시 가독성 떨어짐 & 모순 현상 & 관련 없는 문장  
   in-context learning이 WIC(두 단어가 문장에 같은 방식으로 사용되었는지 판별) & ANLI(한 문장은 다른 문장을 암시하는가?) -> zero & few shot이 효과적이지 않다.
2. 모델의 구조 & 알고리즘적 한계: 양방향적인 구조 & denoising 훈련 목적함수 고려 안함.  
   -> 빈칸 채우기/ 두 문단 비교 후 답하기/ 긴 문단 읽고 짧은 답변 생성 task에서 낮은 성능  
   GPT-3에서 양방향 모델 + zero/few 모델이 향후 연구 과제
3. 본질적인 한계: 사전학습 할 때 현재의 목적 함수는 모든 토큰에 대해 동일한 가중치를 적용/ 어떤 토큰을 예측하는 것이 더 중요한지를 반영 안함.  
   다른 연구는 관심 있는 토큰(개체)에 더 잘 예측하도록 커스터마이징..?  
     
   self-supervised 목적함수 사용하는 것은 토큰을 잘 예측할 것을 강요, 하지만 언어 시스템은 목적 지향적? -> 단순 예측을 넘은 논리적인 근거 필요??  
     
   실제 세계의 상호작용이 부족? 세상의 내용을 담은 사전 context가 부족? 따라서 단순히 규모만 키우기에는 한계가 있다.(세상을 다 담기에는 너무 힘듬.) -> 다른 접근 방법 필요?  
   -> 사람과 비슷한 목적함수 or 강화학습을 통해 finetune or 이미지등 다른 분야와 접목해서 문제 해결
4. 훈련 과정의 효율성: 사전학습 시 하나의 샘플이 모델에게 주는 정보에 대한 효율성을 높여야 한다. (사전 학습 시 방대한 데이터를 모델이 더 효율적으로 흡수해야한다.)
5. Few-shot 세팅 시 불확실성: 정말 새로운 task에서 배우는 건지 아니면 사전 학습에 포함된 task에 치중하여 답을 도출하는지 모호함. = 이 부분은 사람도 모호
6. 비용: 학습하기에 스케일이 너무 커서 비쌈, 추론 조차도 수행하기 쉽지 않음 -> 실용성이 매우 떨어짐 -> distillation이 좋은 방법 (현재 리소스도 빡셈)
7. Explanable이 떨어짐, 왜 이런 결과를 냈니?의 부족, calibration이 떨어지며, 예측할 때 답의 분산이 매우 크다. 또한 훈련 데이터에 대한 편향이 존재할 수 있다. -> 데이터의 편향, 예측의 편향은 위험한 결과 초래 가능

**GPT-3, 사회에 미치는 영향(블로그 참조)**