**Abstract**

문서와 질문을 조건으로 하는경우 언어 모델에서 생성된 답변은 CoQA 데이터 세트에서 55 F1 score에 도달한다?

Zero shot : 일반적으로 딥러닝은 training에 사용된 class만을 예측할 수 있다. 따라서 unseen data가 입력되면 seen class로 예측하는 바보가 되버리는데, Zero shot은 train set에 포함되지 않은 unseen class를 예측하는 분야이다. 즉, unseen data를 입력 받아도, seen data로 학습된 지식을 전이하여 unseen data를 unseen class로 예측할 수 있다.’

**Introduction**

대규모 데이터 셋, 고용량 모델 및 지도학습은 시스템 데이터 분포, 작업 사양의 약간의 변경에 취약하고 민감함

그래서 우리는 많은 작업을 수행할 수 있는 일반적인 시스템으로 이동하고 싶음

기존 ML은 train -> test(독립) 잘 하도록 모방하게 하는 거임

이러한 방식은 협소한 분야의 전문가들에게 도움됨(전체적으로 도움을 못준다)

우리는 단일 도메인 데이터 세트에 대한 단일 작업 train의 보급이 현재 시스템에서 관찰 되는 일반화 부족의 주요 원인이라 생각

현재 아키텍쳐 로 강력한 시스템으로 발전하려면, 광범위한 도메인 및 작업에 대한 train 및 성능 측정이 필요함 -> 최근 GLUE, decaNLP 에서 다룸

멀티테스크 nlp 는 아직 초기 단계다.

최근 근황 -> 약간의 성능 향상(Yogatama et al. 2019), 10개, 17개(데이터셋, 목표) 쌍에서 훈련시킨 Bowman et al(2018)

멀티태스킹 접근 방식이 잘 되기 위해서는 현재 아키텍쳐로는 효과적인 훈련쌍이 필요함

하지만 현재 기술로 무차별 대입에 필요한 정도로 데이터 세트 생성 및 목표 설계를 계속 확장하는 것은 매우 어려운 일이다.

현재 언어 과제 최고 성능 가지는건 사전 훈련 + 지도 fine tuning 조합 사용함

기존꺼 1. 셀프 어텐션 사용

2. 지도학습이 최소이거나, 비지도 학습일 경우 상식 추론, 감정분석과 같은 특정 작업을 수행하는 언어모델의 가능성 보여줌

이 논문에서 우리는 이 두가지를 연결할꺼고, 보다 일반적인 이전 방법의 추세를 계속 하거임

언어 모델은 매개변수나 아키텍처 수정 없이 제로샷 설정에서 다운스트림 작업을 수행 할 수 있음

우리는 제로샷 설정에서 광범위한 작업을 수행하는 언어모델의 능력을 강조함으로써 이 접근방식의 가능성을 보일거임

**Approach**

우리의 접근 방식의 핵심은 언어 모델링이다.

언어에는 자연스러운 순서가 있기 때문에 조건부 확률의 곱으로 기호 초과 결합 확률을 인수분해 하는게 일반적임 (Jelinek & Mercer 1980)(Bengio et al. 2003)

텍스트이(가) 표시된 사진

자동 생성된 설명

이 방식은 모든 조건에서 다루기 쉬운 샘플링 및 추정을 허용

일반적인 시스템은 동일한 입력이여도 다양한 작업을 수행할 수 있어야 하므로, 입력뿐만 아니라 수행할 작업에 대해서도 조건을 지정해야함

이는 멀티태스킹 및 메타러닝 환경에서 다양하게 공식화 되어있음

작업 조건화는 작업별 인코더, 디코더 같이 아키텍쳐 수준 또는 MAML의 내부 및 외부 루프 최적화 프레임(Finn et al. 2017)와 같은 알고리즘 수준에서 구현되는 경우가 많음

하지만 McCann (2018) 언어는 작업 입력 및 출력을 모두 기호의 순서로 지정하는 유연한 방법을 제공함

Ex) 독해 문제는 (answer, document, question, answer)로 작성 가능

이러한 유형의 형식을 사용하여 다양한 작업을 추론하고 수행함

언어 모델링 또한 원치적으로 McCann 의 작업을 학습할 수 있음

어떤 기호가 예측될 출력인지에 대한 명시적인 감독 필요없음

예비 실험해서 비지도 학습이 목표를 수렴으로 최적화 할 수 있는지 보니, 토이 모델에서 멀티태스킹 학습을 수행할 수 있지만, 지도학습보다 훨씬 느린걸 확인함

대화는 매력적인 접근방식이지만 지나치게 제한적이어서 걱정됨

인터넷에는 양방향 통신없이 수동적으로 사용할 수 있는 방대한 양의 정보가 포함되어 있음

우리의 추측은 충반한 능력을 가진 언어모델이 조달 방법에 관계 없이 비자연어 시퀀스를 더 잘 예측하기 위해 시연된 작업을 추론하고 수행하는 방법을 배우기 시작할 것 이라는 것임

언어모델이 이를 수행할 수 있다면 사실상 비지도 멀티태스킹을 수행하는 것임

우리는 다양한 작업에 대한 zero-shot 설정에서 언어 모델의 성능을 분석하여 이것이 사실인지 테스트 함

2.1 Training Dataset

대부분 이전 작업들은 뉴스기사, 위키피디아, 소설책과 같은 단일 텍스트 영역에서 언어 모델을 훈련함.

우리의 접근 방식은 가능한 한 다양한 도메인 및 컨텍스트에서 작업의 자연스러운 언어 데모를 수집하기 위해 가능한 한 크고 다양한 데이터 세트를 구축하도록 하려고함

웹 크롤러 가 좋은데 데이터 품질에대한 문제가 많이 제기됨 (Trinh & LE(2018))

텍스트이(가) 표시된 사진

자동 생성된 설명

자연발생한 WebText 트레이닝 셋

Eng to French

Frech to eng example

Trinh & Le의 가장 최고 결과의 대상 데이터 세트는 Winograd Schema challenge와 가장 유사한 문서만 포함하는 common crawl의 작은 하위 샘플을 사용함

특정 작업의 성능을 향상시키기 위한 실용적인 접근 방식이지만 하지만 우리는 미리 수행할 작업에 대해 가정하는걸 피하고 싶었음

그래서 우리는 문서 품질을 강조하는 새로운 웹 스크래핑을 만듬

이를 위해 인간이 선별/필터링한 웹페이지만 스크랩함

전체 웹스크래핑을 이렇게 하는건 매우 비싸므로, 우리는 소셜미디어 플랫폼인 reddit에서 모든 아웃바운드 링크를 스크랩했고, 최소 3 카르마(페이스북 좋아요 같은거) 이상인것만 가져옴

이것은 다른 사용자가 링크를 흥미롭거나, 교육적이거나, 재미있다고 보았는지에 대한 경험적 지표로 생각할수있음

결과 데이터셋에는 4,500백만 링크의 텍스트 하위집합이 포함됨

HTMLresponses에서 텍스트 추출하기 위해 Dragnet과 Newspaper content 추출기의 조합을 사용함

중복 제거 + 휴리스틱 기반 정리 후 40GB의 텍스트, 8백만이 약간 넘는 문서를 포함하는 예비버전의 webtext 사용

WebText는 다른 데이터 세트의 공통 데이터 소스이고 테스트 평가 작업과 훈련 데이터가 겹치기 때문에 분석을 복잡하게 할 수 있기 때문에 우리는 WebText에서 모든 Wikipedia 문서를 제거했습니다.

**Input Representation**

webtext에서 표준 바이트 수준 LM과 유사한 성능 격차 관찰

BPE는 byte이름에도 불구하고 유니코드 에서 작동하기 때문에 유니코드 기호의 전체공간을 포함해야함

가중 기호 토큰이 추가되기전 130,000개 이상의 기본 어휘 생성

이건 BPE가 자주 사용하는 32,000~64,000개의 토큰 어휘와 비교할 때 엄청 큼.

대조적으로 BPE의 바이트 수준 버전은 256개의 어휘만 필요

그러나 BPE를 바이트 시퀀스에 직접 적용하면 토큰 어휘를 구축하기 위해 그리드한 빈도 기반 휴리스티을 사용하는 BPE로 인해 차선의 병합이 발생

우리는 dog 에서도 dog. Dog! Dog? 같이 많은 버전 발생하는걸 관찰해서

제한된 어휘슬롯과 모델 용량이 차선으로 할당됨.

이를 피하기 위해 BPE가 모든 byte 시퀀스에 대한 문자 범주를 병합하는 것을 방지함.

압축 효율성을 크게 향상 시키면서 여러 어휘 토큰에 걸쳐 단어의 단편화를 최소화하는 공백에 대한 예외 추가

이 입력 표현을 통해 단어 수준 LM의 경험적 이점과 바이트 수준 접근의 일반성을 결합할 수 있습니다. 우리의 접근 방식은 모든 유니코드 문자열에 확률을 할당할 수 있기 때문에 사전 처리, 토큰화, 단어 크기에 관계없이 모든 데이터 세트에서 LM을 평가할 수 있습니다.

우리는 transformer 사용함 우리의 LMs 모델에 기반된

모델은 OpenAi GPT모델의 세부사항을 크게 따름 약간 우리가 수정한,

계층 정규화(Ba et al., 2016)는 활성화 전 잔차 네트워크(He et al., 2016)와 유사하게 각 하위 블록의 입력으로 이동되었으며 추가 계층 정규화가 최종 자체 주의 블록 뒤에 추가되었습니다. 모델 깊이가 있는 잔차 경로의 누적을 설명하는 수정된 초기화가 사용됩니다. 초기화 시 잔여 층의 가중치를 1=pN(여기서 Ni는 잔여 층 수)만큼 조정합니다. 어휘는 50,257로 확장됩니다. 또한 컨텍스트 크기를 512에서 1024 토큰으로 늘리고 512의 더 큰 배치 크기를 사용합니다.

테이블이(가) 표시된 사진

자동 생성된 설명

**3. Experiments**

우리는 대략 로그 간격의 크기로 4개의 LM을 훈련하고 벤치마킹했습니다. 아키텍처는 표 2에 요약되어 있습니다. 가장 작은 모델은 원래 GPT와 동일하고 두 번째로 작은 모델은 BERT의 가장 큰 모델과 동일합니다(Devlin et al., 2018). 우리가 GPT-2라고 부르는 가장 큰 모델은 GPT보다 훨씬 더 많은 매개변수를 가지고 있습니다. 각 모델의 학습률은 WebText의 5% 보류 샘플에서 최상의 난도를 위해 수동으로 조정되었습니다. 모든 모델은 여전히 Web-Text에 적합하지 않으며, 더 많은 훈련 시간을 감안할 때 지연된 당혹도가 아직 개선되었습니다.

**3.1 Language Modeling**

제로샷 작업 전송을 향한 초기 단계로 WebText LM이 교육을 받은 기본 작업인 언어 모델링에서 제로샷 도메인 전송을 수행하는 방법을 이해하는 데 관심이 있습니다. 우리 모델은 바이트 수준에서 작동하고 손실 전처리 또는 토큰화가 필요하지 않기 때문에 모든 언어 모델 벤치마크에서 평가할 수 있습니다. 언어 모델링 데이터 세트에 대한 결과는 일반적으로 표준 예측 단위당 평균 음수 로그 확률의 크기 조정 또는 지수 버전인 수량(일반적으로 문자, 바이트 또는 단어)으로 보고됩니다.

WebText LM에 따라 데이터 세트의 로그 확률을 계산하고 표준 단위 수로 나누어 동일한 양을 평가한다. 이러한 많은 데이터 세트의 경우 WebText LM은 400억 바이트에서 26번밖에 발생하지 않는 매우 드문 문자열 <UNK>와 같은 표준화된 텍스트, 끊어진 구두점 및 수축과 같은 토큰화 아티팩트를 적극적으로 예측해야 하므로 배포되지 않은 상태로 테스트됩니다.

우리는 가능한 많은 토큰화/사전 처리 아티팩트를 제거하는 역 토큰화 해제기를 사용하여 표 3에 주요 결과를 보고합니다. 이러한 de-tokenizer는 뒤집을 수 있기 때문에 데이터 세트의 로그 확률을 여전히 계산할 수 있으며 도메인 적응의 간단한 형태로 생각할 수 있습니다. 우리는 이러한 de-tokenizer를 사용하여 GPT-2에 대해 2.5에서 5의 당혹도 이득을 관찰했습니다.

WebText LM은 도메인 및 데이터 세트 간에 잘 전송되어 제로샷 설정에서 8개 데이터 세트 중 7개에서 최신 기술을 개선합니다. Penn Treebank 및 WikiText-2와 같이 훈련 토큰이 100만~200만 개에 불과한 소규모 데이터 세트에서 큰 개선 사항이 발견되었습니다. LAMBADA(Paperno et al., 2016) 및 Children's Book Test(Hill et al., 2015)와 같은 장기 종속성을 측정하기 위해 생성된 데이터세트에서도 큰 개선이 이루어졌습니다. 우리 모델은 여전히 One BillionWord Benchmark에 대한 이전 작업보다 훨씬 나쁩니다(Chelba et al., 2013). 이것은 가장 큰 데이터 세트이고 가장 파괴적인 전처리 기능이 있기 때문일 수 있습니다. 1BW의 문장 수준 셔플링은 모든 장거리 구조를 제거합니다.

**3.2 Childeren’s Book Test**

CBT(Children's Book Test)(Hill et al., 2015)는 명명된 개체, 명사, 동사, 전치사 등 다양한 단어 범주에 대한 LM의 성능을 조사하기 위해 만들어졌습니다. CBT는 평가 메트릭으로 난해함을 보고하는 대신 자동으로 구성된 클로즈 테스트에서 정확성을 보고합니다. 여기서 작업은 생략된 단어에 대한 10가지 가능한 선택 중 어느 것이 올바른지 예측하는 것입니다. 원본 논문에서 소개된 LMapproach에 따라 LM에 따라 각 선택과 이 선택을 조건으로 하는 나머지 문장의 확률을 계산하고 가장 높은 확률을 예측합니다. 그림 2에서 볼 수 있듯이 모델 크기가 증가함에 따라 성능이 꾸준히 향상되고 이 테스트에서 인간의 성능과의 차이의 대부분이 줄어듭니다. 데이터 중복 분석은 CBTtest 세트북 중 하나인 Rudyard Kipling의 The Jungle Book이 웹텍스트에 있음을 보여주므로 유의미한 중복이 없는 검증 세트에 대한 결과를 보고합니다. GPT-2는 일반 명사에서 93.3%, 이름에서 89.1%라는 새로운 최첨단 결과를 달성했습니다. CBT에서 PTB styletokenization 아티팩트를 제거하기 위해 de-tokenizer가 적용되었습니다.

**3.3 LAMBADA**

LAMBADA 데이터 세트(Paperno et al., 2016)는 장거리 종속성 텍스트를 모델링하는 시스템의 능력을 테스트합니다. 과제는 인간이 성공적으로 예측하기 위해 적어도 50개의 컨텍스트 토큰이 필요한 문장의 마지막 단어를 예측하는 것입니다. **GPT-2는 최신 기술을 99.8(Grave et al., 2016)에서 8.6 perplexity로 개선하고 이 테스트에서 LM의 정확도를 19%(Dehghani et al., 2018)에서 52.66%로 높입니다**. GPT-2의 오류를 조사한 결과 대부분의 예측은 문장의 유효한 연속이지만 유효한 마지막 단어는 아님을 보여주었습니다. 이것은 LM이 단어가 문장의 마지막이어야 한다는 추가적인 유용한 제약을 사용하지 않는다는 것을 암시합니다. 이에 대한 근사값으로 불용어 필터를 추가하면 정확도가 63.24%로 추가로 증가하여 이 작업에 대한 전체 기술 상태가 4% 향상됩니다. 이전의 기술 상태(Hoang et al., 2018)는 모델의 출력이 컨텍스트에 나타난 단어로만 제한되는 다른 제한된 예측 설정을 사용했습니다. GPT-2의 경우 이 제한은 19%의 답변은 컨텍스트에 있지 않습니다. 사전 처리 없이 데이터 세트 버전을 사용합니다.

**3.4 Winograd Schema Challenge**

Winograd Schema 챌린지(Levesque et al., 2012)는 텍스트의 모호성을 해결하는 능력을 측정함으로써 상식적인 추론을 수행하는 시스템의 능력을 측정하기 위해 구성되었습니다. 최근 Trinh & Le(2018)는 LM을 사용하여 더 높은 확률로 모호성의 해결을 예측함으로써 이 문제에 대한 상당한 진전을 보여주었습니다. 우리는 문제 공식을 따르고 그림 3의 전체 및 부분 채점 기술을 사용하여 모델의 성능을 시각화합니다. **GPT-2는 최첨단 정확도를 7% 향상시켜 70.70%를 달성합니다**. 데이터 세트는 273개의 예제로 매우 작기 때문에 Trichelair et al.을 읽는 것이 좋습니다. (2018) 이 결과를 맥락화하는 데 도움이 됩니다.

**3.5 Reading Comprehension**

대화 질문 응답 데이터 세트(CoQA)Reddy et al. (2018) 문서에 대한 질문자와 질문자 사이의 자연어 대화와 쌍을 이루는 7개의 다른 도메인 문서로 구성됩니다. CoQA는 독해 능력과 대화 이력(예: "왜?")에 의존하는 질문에 대답하는 모델의 능력을 테스트합니다. .문서, 관련 대화 기록 및 최종 토큰 A를 조건으로 하는 경우 **GPT-2에서 디그리 디코딩: 개발 세트에서 55 F1 달성. 이는 해당 기준선이 훈련된 127,000개 이상의 수동으로 수집된 질문 답변 쌍을 사용하지 않고 기준선 시스템 4개 중 3개의 성능과 일치하거나 초과합니다.** BERT 기반 시스템인 감독된 SOTA는 인간의 89 F1 성능에 근접하고 있습니다. GPT-2의 성능은 감독된 교육이 없는 시스템에서 흥미진진한 반면, 답변 및 오류의 일부 검사는 GPT-2가 종종 누가 질문에 대한 응답으로 문서의 이름으로 답변하는 것과 같은 간단한 검색 기반 휴리스틱을 사용한다고 제안합니다.

3.6 Summarization

CNN 및 Daily Mail 데이터 세트에 대한 요약을 수행하는 GPT-2의 기능을 테스트합니다(Nallapati et al., 2016). 요약 동작을 유도하기 위해 기사 뒤에 textTL;DR:을 추가하고 k=2를 사용하여 Top-krandom 샘플링(Fan et al., 2018)을 사용하여 100개의 토큰을 생성합니다. 이는 반복을 줄이고 욕심 많은 디코딩보다 추상적인 요약을 권장합니다**. 이 100개 토큰에서 처음 3개의 생성된 문장을 요약으로 사용합니다**. 세대는 질적으로 요약과 유사하지만 표 14에서 볼 수 있듯이 **기사의 최근 내용에 초점을 맞추거나 충돌 사고에 연루된 자동차 수 또는 모자나 셔츠에 로고가 있는지 여부와 같은 특정 세부 사항을 혼동하는 경우가 많습니다**. 일반적으로 보고되는 ROUGE 1,2,L 메트릭에서 생성된 요약은 고전적인 신경 기준선의 성능에 접근하기 시작하고 **기사에서 3개의 임의 문장을 선택하는 것보다 성능이 거의 우수합니다**. GPT-2의 성능은 작업 힌트가 제거될 때 집계 메트릭에서 6.4포인트 떨어집니다. 이는 자연 언어로 언어 모델에서 작업 특정 동작을 호출하는 기능을 보여줍니다.

* 기사에서 3개 임의 문장 보다, GPT-2 이용해서 3문장 요약한게 성능 더 좋다

**3.7 Translation**

우리는 GPT-2가 한 언어에서 다른 언어로 번역하는 방법을 배우기 시작했는지 테스트합니다. 이것이 원하는 작업임을 추론하는 것을 돕기 위해 우리는 형식의 예문 쌍의 컨텍스트에 따라 언어 모델을 조건화합니다. 번역으로. **WMT-14 English-Frenchtest 세트에서 GPT-2는 5 BLEU를 얻습니다. 이는 감독되지 않은 단어 번역에 대한 이전 작업에서 추론된 이중 언어 사전으로 단어별 대체보다 약간 나쁩니다**(Conneau et al., 2017b). . **WMT-14 프랑스어-영어 테스트 세트에서 GPT-2는 매우 강력한 영어 모델을 활용하여 훨씬 더 나은 성능을 발휘하여 11.5 BLEU를 달성할 수 있습니다. 이것은 (Artetxe et al., 2017) 및 (Lampleet al., 2017)의 몇 가지 감독되지 않은 기계 번역 기준을 능가하지만 현재 최고의 감독되지 않는 기계 번역 접근 방식의 33.5 BLEU보다 훨씬 나쁩니다**(Artetxe et al., 2019). **필터링 단계로 WebText에서 영어가 아닌 웹페이지를 의도적으로 제거했기 때문에 이 작업의 성능은 우리에게 놀라운 것이었습니다. 이를 확인하기 위해 우리는 WebText에서 바이트 수준 언어 탐지기를 실행했는데, 이 탐지기는 이전의 감독되지 않은 기계 번역 연구에서 흔히 볼 수 있는 단일 언어 프랑스어 코퍼스보다 약 500배 작은 프랑스어로 된 10MB의 데이터만 탐지했습니다.**

* **프랑스어가 거의 제거 됐을텐데, 이렇게 성능이 나왔다 -> 탐지해보니 기존 비지도 번역 연구에서 띄는 코퍼스 보다 약 500배나 작았다.**

**3.8 Question Answering**

언어 모델에 어떤 정보가 포함되어 있는지 테스트하는 잠재적인 방법은 팩토이드 형식의 질문에 대한 정답을 생성하는 빈도를 평가하는 것입니다. 모든 정보가 A Neural ConversationalModel(Vinyals & Le, 2015)과 같은 매개변수에 저장되는 신경 시스템에서 이러한 동작을 보여주는 이전에는 고품질 평가 데이터 세트의 부족으로 인해 정성적 결과가 보고되었습니다. 최근에 도입된 Natural Questions datase는 이를 보다 정량적으로 테스트할 수 있는 유망한 리소스입니다. 번역과 유사하게 언어 모델의 컨텍스트는 모델이 데이터 세트의 단답형 스타일을 추론하는 데 도움이 되는 예제 질문 답변 쌍으로 시드됩니다. **GPT-2는 읽기 이해에 일반적으로 사용되는 정확히 일치 메트릭으로 평가할 때 질문의 4.1%에 올바르게 답변합니다**. SQUAD.3과 같은 데이터 세트는 비교점으로, 가장 작은 모델은 각 질문 유형(누가, 무엇을, 어디서, 등...)에 대해 가장 일반적인 답변을 반환하는 믿을 수 없을 정도로 단순한 **기준선의 1.0% 정확도를 초과하지 않습니다. GPT-2는 5.3배 더 많은 질문에 정확하게 답하는데, 이는 모델 용량이 아직까지 이런 종류의 작업에서 신경 시스템의 성능 저하의 주요 요인이었음을 시사합니다**. GPT-2가 생성된 답변에 할당할 확률은 잘 보정되었으며 **GPT-2는 가장 자신 있는 질문의 1%에 대해 63.1%의 정확도를 갖습니다.** 개발 세트 질문에 대해 GPT-2가 생성한 가장 자신 있는 답변 30개 GPT-2의 성능은 정보 검색과 추출 문서 질의 응답을 혼합하는 30~50% 범위의 개방형 도메인 질의 응답 시스템보다 여전히 훨씬 나쁩니다(Alberti et al., 2019).

텍스트이(가) 표시된 사진

자동 생성된 설명

**4. Generalizaiton vs Memorization**

최근 컴퓨터 비전 연구에 따르면 일반적인 이미지 데이터 세트에는 거의 중복되지 않는 양의 이미지가 포함되어 있습니다. 예를 들어 **CIFAR-10은 훈련 이미지와 테스트 이미지 사이에 3.3%의 중첩이 있습니다**(Barz & Denzler, 2019). **그 결과 기계 학습 시스템의 일반화 성능이 과도하게 보고됩니다. 데이터 세트의 크기가 증가함에 따라 이 문제가 점점 더 가능성이 높아져 WebText에서 유사한 현상이 발생할 수 있음을 시사합니다.** 따라서 훈련 데이터에도 테스트 데이터가 얼마나 많이 나타나는지 분석하는 것이 중요합니다.

이를 연구하기 위해 8그램의 WebText 훈련 세트 토큰을 포함하는 Bloom 필터를 만들었습니다. 기억력을 향상시키기 위해 문자열은 구분 기호로 단일 공백이 있는 소문자 영숫자 단어만 포함하도록 정규화되었습니다. 블룸 필터는 위양성 비율이 1108로 상한이 되도록 구성되었습니다. 1M 문자열을 생성하여 낮은 거짓 긍정률을 추가로 확인했으며 그 중 필터에서 0개를 찾았습니다.

이러한 블룸 필터를 사용하면 데이터 세트가 주어지면 WebText 교육 세트에서도 발견되는 해당 데이터 세트의 8그램 비율을 계산할 수 있습니다. 표 6은 공통 LM 벤치마크의 테스트 세트에 대한 이러한 중첩 분석을 보여줍니다. CommonLM 데이터 세트의 테스트 세트는 Web-Text 트레인과 1-6% 겹칩니다. 평균 3.2% 겹칩니다**. 다소 놀랍게도 많은 데이터 세트가 평균 5.9%의 중복으로 자체 교육 분할과 더 큰 중복을 가지고 있습니다**. **우리의 접근 방식은 회상을 최적화하며 중복을 수동으로 검사하면 많은 공통 문구가 표시되지만 중복 데이터로 인해 더 긴 일치 항목이 많이 있습니다**. . **이것은 WebText에만 있는 것이 아닙니다. 예를 들어, 우리는 WikiText-103의 테스트 세트에 훈련 데이터 세트에도 있는 기사가 있다는 것을 발견했습니다. 테스트 세트에 60개의 기사만 있기 때문에 최소한 1.6%의 중복이 있습니다**.4 **잠재적으로 더 걱정스럽게도 1BW는 우리 절차에 따라 자체 훈련 세트와 거의 13.2%의 중복이 있습니다.**

Winograd Schema Challenge의 경우 WebTexttraining 세트와 8그램이 겹치는 스키마는 10개만 찾았습니다. 이 중 2개는 가짜 일치였습니다. 나머지 8개 중 1개의 스키마만 모든 컨텍스트에서 답변을 제공한 것으로 나타났습니다. **CoQA의 경우 뉴스 도메인의 문서 중 약 15%가 이미 WebText에 있으며 모델은 이러한 문서에서 약 3F1 더 나은 성능을 보입니다**. CoQA의 개발 세트 메트릭은 5개의 다른 도메인에 대한 평균 성능을 보고하며 다양한 도메인에 걸쳐 중첩으로 인해 약 0.5-1.0 F1의 이득을 측정합니다. 그러나 CoQA가 WebText의 링크에 대한 차단 날짜 이후에 릴리스되었기 때문에 WebText에는 실제 교육 질문이나 답변이 없습니다.

LAMBADA에서 평균 중첩은 1.2%입니다. GPT-2는 15% 이상의 중복이 있는 예에서 약 2개의 혼란을 더 잘 수행합니다. 중첩 이동이 있는 모든 예를 제외할 때 메트릭을 다시 계산하면 복잡도가 8.6에서 8.7로 감소하고 정확도가 63.2%에서 62.9%로 감소합니다. **전체 결과의 이 아주 작은 변화는 200개 중 1개의 예제에서만 상당한 중복이 있기 때문일 수 있습니다.**

전반적으로, 우리의 분석은 WebText 훈련 데이터와 특정 평가 데이터 세트 간의 데이터 중첩이 보고된 결과에 작지만 일관된 이점을 제공한다고 제안합니다. 그러나 대부분의 데이터 세트의 경우 표 6이 강조하는 것처럼 표준 학습 세트와 테스트 세트 사이에 이미 존재하는 것보다 훨씬 더 큰 중복을 발견하지 못했습니다.

매우 유사한 텍스트가 성능에 미치는 영향을 이해하고 수량화하는 것은 중요한 연구 질문입니다. 확장 가능한 퍼지 일치와 같은 더 나은 중복 제거 기술도 이러한 질문에 더 잘 답하는 데 도움이 될 수 있습니다. **현재로서는 새로운 NLP 데이터 세트에 대한 훈련 및 테스트 분할을 생성하는 동안 중요한 검증 단계 및 온전성 검사로 n-gram 중첩 기반 중복 제거를 사용하는 것이 좋습니다.** **WebText LM의 성능이 다음과 같은지 여부를 결정하는 또 다른 잠재적인 방법 암기는 자신의 고정 세트에서 자신의 성능을 검사하는 것입니다. 그림 4에서 볼 수 있듯이 WebText의 훈련 세트와 테스트 세트 모두에서 성능은 유사하며 모델 크기가 증가함에 따라 함께 향상됩니다**. 이것은 GPT-2조차도 여러 면에서 WebText에서 여전히 과소 적합하다는 것을 의미합니다. **GPT-2는 말하는 유니콘의 발견에 대한 뉴스 기사도 작성할 수 있습니다.** 예는 표 13에 나와 있습니다.

**5. Related work**

이 작업의 상당 부분은 더 큰 데이터 세트에서 훈련된 더 큰 언어 모델의 성능을 측정했습니다. 이것은 Jozefowicz 등의 작업과 유사합니다. (2016) 10억 WordBenchmark에서 RNN 기반 언어 모델을 확장했습니다. Bajgaret al. (2016) 또한 이전에 표준 교육 데이터 세트를 보완하기 위해 Project Gutenberg에서 훨씬 더 큰 훈련 데이터 세트를 생성하여 Children's Book Test의 결과를 개선했습니다. Hestness et al. (2017)은 다양한 딥러닝 모델의 성능이 모델 용량과 데이터 세트 크기의 함수로 어떻게 변하는지에 대한 철저한 분석을 수행했습니다. 우리의 실험은 작업 전반에 걸쳐 훨씬 더 시끄럽지만 목표의 하위 작업에 대해 유사한 경향이 유지되고 1B+ 매개변수 영역으로 계속된다는 것을 제안합니다.

생성 모델에서 흥미로운 학습 기능은 행 너비 추적 및 인용/주석 감지를 수행하는 RNN 언어 모델의 셀과 같이 이전에 문서화되었습니다. Karpathy et al. (2015). 우리 작업에 더 영감을 주는 것은 Wikipedia 기사를 생성하도록 훈련된 모델이 언어 간에 이름을 번역하는 방법도 배운 Liu et al.(2018)의 관찰이었습니다. 이전 작업은 iWeb Corpus와 같은 웹 페이지(Davies, 2018). 언어 작업을 위한 사전 훈련 방법에 대한 광범위한 작업이 있었습니다. 소개에서 언급한 것 외에도 GloVe(Pennington et al., 2014)는 모든 Common Crawl에 대해 단어 벡터 표현 학습을 확장했습니다. 심층 표현 학습 텍스트에 대한 초기 작업은 Skip-thought 벡터(Kiros et al., 2015)였습니다. McCannet al. (2017)은 기계 번역 모델에서 파생된 표현의 사용을 탐구했고 Howard & Ruder(2018)는 (Dai& Le, 2015)의 RNN 기반 미세 조정 접근 방식을 개선했습니다. (Conneau et al., 2017a) 자연어 추론 모델에 의해 학습된 표현의 전달 성능을 연구했고 (Subramanian et al., 2018) 대규모 멀티태스킹 훈련을 탐구했습니다.(Ramachandran et al., 2016) seq2seq 모델이 인코더 및 디코더로 사전 훈련된 언어 모델로 초기화됩니다. 보다 최근의 연구에 따르면 LM 사전 훈련은 잡담 대화 및 대화 기반 질문 답변 시스템과 같은 어려운 생성 작업을 미세 조정하는 경우에도 도움이 됩니다(Wolf et al., 2019)(Dinan et al., 2018).

**6. Discussion**

지도 및 비지도 사전 훈련 방법의 표현 학습(Hill et al., 2016), 이해(Levy & Goldberg, 2014) 및 비판적 평가(Wieting & Kiela, 2019)에 많은 연구가 진행되었습니다. 우리 결과 감독되지 않은 작업 학습이 탐구해야 할 추가 유망한 연구 영역이라고 제안합니다. 이러한 결과는 한계 내에서 이러한 사전 훈련 기술 중 하나가 감독된 적응 또는 수정 없이 작업을 직접 수행하는 방법을 배우기 시작한다는 것을 보여주기 때문에 다운스트림 NLP 작업에 대한 사전 훈련 기술의 광범위한 성공을 설명하는 데 잠재적으로 도움이 됩니다. 독해에 대해 GPT-2의 성능은 제로샷 설정에서 감독된 기준선과 경쟁적입니다. 그러나 요약과 같은 다른 작업에서는 작업을 정성적으로 수행하는 동안 그 성능은 여전히 ​​양적 측정 기준에 따라 초보적입니다. 연구 결과로 시사하는 바가 있지만, 실제 적용 측면에서 GPT-2의 제로샷 성능은 아직 사용 가능하지 않습니다. 우리는 많은 표준 NLP 작업에서 WebTextLM의 제로샷 성능을 연구했지만 많은 평가할 수 있는 추가 작업. 의심할 여지 없이 GPT-2의 성능이 여전히 무작위보다 좋지 않은 실제 작업이 많이 있습니다. 질문 답변 및 번역과 같이 우리가 평가한 일반적인 작업에서도 언어 모델은 충분한 용량이 있을 때만 사소한 기준선을 능가하기 시작합니다. 제로샷 성능이 많은 작업에서 GPT-2의 잠재적 성능 기준선을 설정하지만 명확하지 않습니다. 천장이 미세 조정되는 곳. 일부 작업에서 GPT-2의 완전히 추상적인 출력은 현재 많은 질문 응답 및 독해 데이터 세트에 대한 최첨단 기술인 추출 포인터 네트워크(Vinyals et al., 2015) 기반 출력에서 ​​크게 벗어났습니다. GPT를 튜닝하기 위해 decaNLP 및 GLUE와 같은 벤치마크에 대한 미세 조정을 조사할 계획입니다. 특히 GPT-2의 추가 교육 데이터와 용량이 BERT에서 입증된 단방향 표현의 비효율성을 극복하기에 충분한지 여부가 불분명하기 때문입니다.

**7. Conclusion**

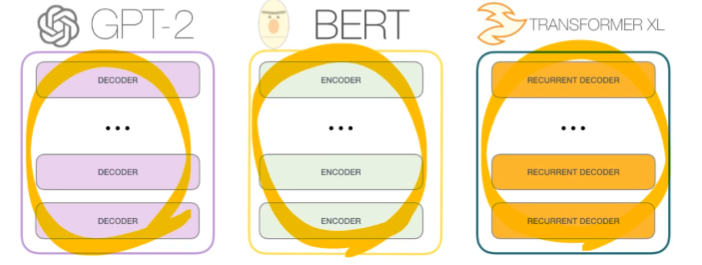
대규모 언어 모델이 충분히 크고 다양한 데이터 세트에서 훈련되면 많은 도메인과 데이터 세트에서 잘 수행할 수 있습니다. GPT-2는 테스트를 거친 언어 모델링 데이터 세트 8개 중 7개에서 최첨단 성능을 구현합니다. 모델이 제로샷 설정에서 수행할 수 있는 작업의 다양성은 충분히 다양한 텍스트 말뭉치의 가능성을 최대화하도록 훈련된 고용량 모델이 명시적 감독 없이도 놀라운 양의 작업을 수행하는 방법을 배우기 시작함을 시사합니다.

정리

GPT-2

GPT 보다 훨씬 큰 용량 사용

GTP-2 vs BERT vs TRANSFORMER XL



Auto-regressive

1024개 토큰 처리 가능

Top-k 개 뽑아서 예측

테이블이(가) 표시된 사진

자동 생성된 설명

모델별 레이어 차이

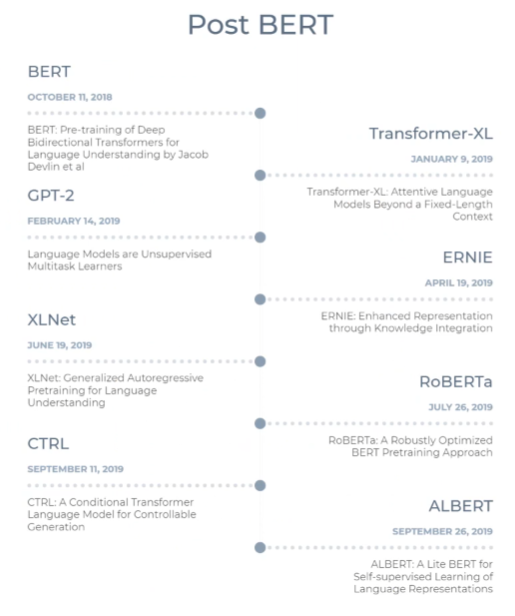
동시 최대 512 토큰 까지 처리 가능

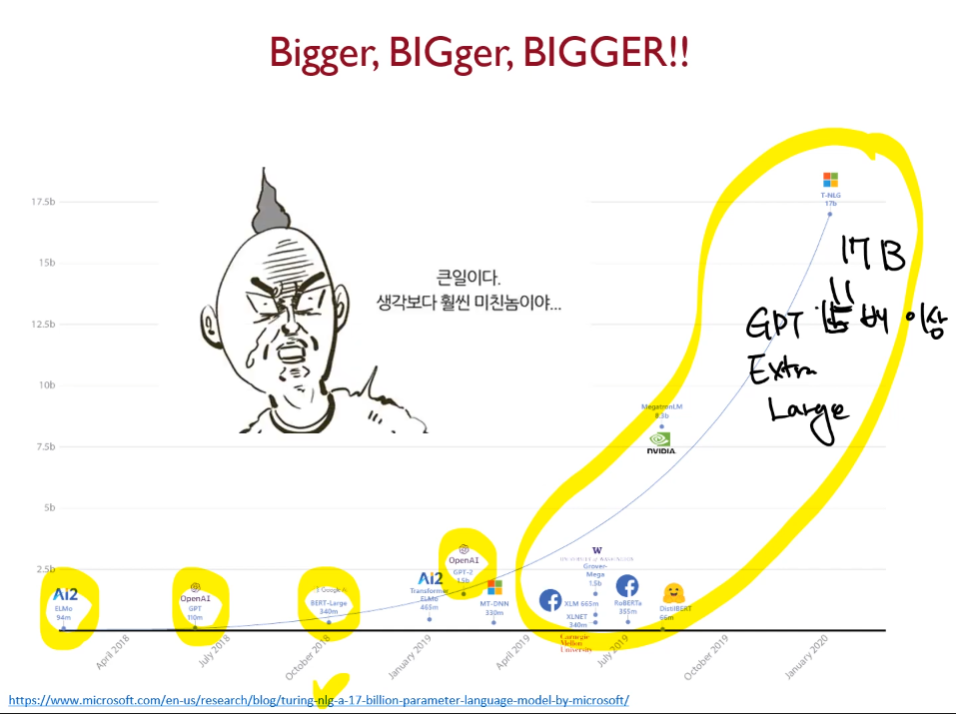
테이블이(가) 표시된 사진

자동 생성된 설명

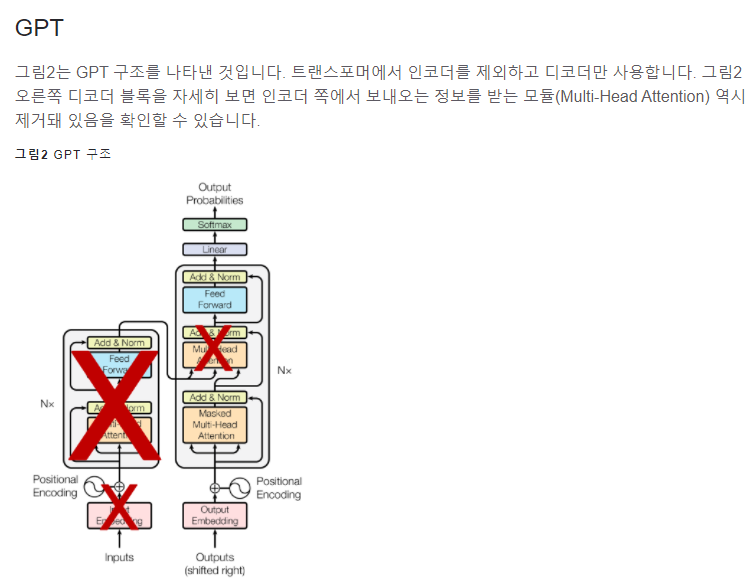
테이블이(가) 표시된 사진

자동 생성된 설명





복습) GPT 구조



Bert 구조

