1. 파운데이션 모델

개념

- 대량의 데이터를 기반으로 사전 학습된 대규모 AI모델
- 다양한 작업에 활용가능한 범용적 기초 모델
- Gpt-4o → 현재는 멀티모달의 지원 또한 가능해짐

이전 모델과 차이점

이전 모델: 새로운 태스크 해결을 하려면 해당 태스크에 대한 별도의 학습이 필요함

파운데이션 모델 : 자세한 설명(프롬프트)을 입력하면 충분히 태스크 수행 가능

파운데이션 모델의 3가지 구성요소

1. 빅데이터

- 시간의 흐름에 따라 인터넷에 쌓이는 데이터가 매우 증가
- 딥러닝 기반 학습 데이터가 늘어날 수록 ai의 성능이 증가

2. 자가 학습

- 어떻게 수많은 데이터를 모델이 학습할 것인가?

  → 자가 학습 알고리즘

  → 사람이 일일이 정답을 알려줄 필요가 없이 모델이 스스로 학습가능한 형태로 가공됨

  ex) 다음 토큰 예측을 통한 텍스트 파운데이션 모델

3. Attention 기반 Transformer 모델

- 학습을 어떻게 효율적으로 할 수 있을 것인가?

  → Attention 기반 Transformer 모델

  → 입력 데이터의 중요한 부분에 주의를 집중함,

텍스트 파운데이션 모델 (거대 언어 모델, LLM)

- GPT-1, BERT 등의 언어 모델도 3가지 구성요소는 이미 포함되어 있었다 하지만 파운데이션 모델과 같은 성능을 보이지 못했다 → 어떠한 차이가 있을까?

GPT-2

→ 더욱 많은 데이터와 큰 모델 사이즈로 만드니 추가 학습 없이도 어느정도 태스크 수행이 가능하다는 사실이 발견됨

→ 모델 사이즈를 늘릴 수록 성능이 좋아진다, GPT-2 당시에도 아직 언더피팅이 되어있음

→ 모델 사이즈가 아직 작다.

1. 규모의 법칙

- 더 많은 데이터, 큰 모델, 긴 학습 → 성능 증가

의의: 작은 규모의 모델들로 예측한 모델 성능의 증가 추세가 실제 더 큰 데이터의 성능 증가 추세와 일치함 → 데이터, 모델 사이즈로 인한 성능 증가량을 미리 예측 가능

2. 창발성

- 모델의 크기가 특정 규모이상으로 발전하니 갑자기 모델에서 특이한 성질이 나타나기 시작

  ex) **인 컨텍스트 학습**(주어진 설명과 예시만으로 새로운 태스크 수행 가능), 추론 능력

- 기존 대비 더 큰 모델 (>7B)이 더 많은 데이터(>1T)에서 학습도어 창발성이 나타나는 경우

  → **텍스트 파운데이션 모델 (거대 언어 모델, LLM)**

거대 언어 모델의 분류

폐쇄형 거대언어 모델

```
**장점**: 일반적으로 우수한 성능, 최신 기능 지원, 쉬운 사용접근성
```

단점: 사용시마다 비용 발생, 모델에 관한 정보 제공이 제한,

ex)

Chat-GPT(Open AI), 가장 많은 활성 유저 수, 전반적으로 우수한 성능

Claude(Antropic), 안전 지향적, 코딩 능력 우수

Gemini(Google), 가장 긴 입출력 지원, 뛰어난 멀티 모달

개방형 거대언어 모델

```
**장점**: 무료로 다운로드 및 사용가능, 오픈소스, 상대적으로 모델에 대한 정보가 공개됨
```

단점: 충분한 계산자원 (GPU) 필요, 상대적으로 폐쇄적 모델보다 낮은 성능

ex)

LLaMa(Meta)

Gemma(Google)

Qwen(Alibaba)

2. 거대 언어 모델의 학습

서론

GPT-3

- 거대언어 모델의 시초, 1750억개 매개변수, 이전보다 10배 이상 크기의 모델 → 창발성 발생
- 학습방법: **다음 토큰 예측**
- 학습 데이터: 3000억 토큰(4TB 텍스트 데이터 = 인터넷+양질의 텍스트 북)
- 학습 비용: 150억원 추정

한계: 사람의 지시에 올바르지 않은 응답을 생성하거나, 유해한 응답이 생성 가능

→ 사후 학습이 필요하다.

정렬 학습

- 거대 언어 모델의 출력이 사용자의 의도 및 가치를 반영하도록 하는 것

1. **지시 학습 (instruction tuning**)

   주어진 지시에 어떤 응답을 생성할 것인가

2. **선호 학습 (preference learning)**

   상대적으로 어떤 응답을 더 선호할 것인가

지시 학습

- 주어진 지시에 어떤 응답을 생성할 것인가
- BERT의 지도 추가 학습(SFT)와 동일
- 어떤 자연어 태스크(Task)든지 지시(instruction)외 응답을 통해 표현하고 학습이 가능하다.

FLAN

- 다양한 태스크 기반의 지시 입력 및 추가학습
- 다양한 태스크에 대한 지시를 템플릿으로 표현

1. 기존에 세상에 존재하는 Task 데이터를 모두 모아 → 지시 데이터로 수정 및 분류함
2. 모델이 새로운 질문에 관한 답변을 생성할 수 있는지 확인하기 위해 특정 분류의 테스트 추론 모델을 만드는데 해당 분류를 제외한 Task의 데이터만으로 학습시켜 모델을 검증함

→ 실험 결과: 예시 없이도 새로운 지시에 올바른 응답을 내놓는 성능이 증가함

성능 향상의 핵심 요소 3가지

1. Task의 개수 → 다양한 지시를 학습할 수록 보지 못한 지시에 대한 일반화 성능이 증가함
2. 추가 학습하는 모델의 크기 → 특정 규모 이하에선느 오히려 지시학습 효과가 떨어짐
3. 지시를 주는 방법: 자연어 지시로 대화하듯이 지시하는 것이 효과적

선호 학습

- 다양한 응답 중 사람이 더 선호하는 응답을 생성하도록 추가학습하자

지시 학습의 한계 → 정답이 정해지지 않은 문제**(개방형 Task)**에서 한계 발생

→ 다양한 응답은 모델이 생성, 응답에 대한 선호도는 사람이 제공하자

→ 사람의 피드백을 통한 강화 학습 **RLHF**

instructGPT

step1. 지시 학습을 통한 텍스트 파운데이션 모델 추가학습

→ 실제 유저로 부터 다양한 지시입력 수집 및 훈련된 주석자를 통한 정답 데이터를 생성 및 학습

step2. 사람의 선호 데이터를 이용하여 **보상 모델(Reward model, RM)**을 학습

→ 사람과 일치한 선호도를 출력할 수 있도록 보상 모델을 지도 학습함

→ 응답이 사람의 선호도와 일치할 수록 더 큰 보상을 부여함

step3. 보상이 높은 응답을 생성하도록 강화학습을 통해 추가학습

→ 입력에 대한 보상모델의 보상 응답이 더욱 높게 나오도록 강화 학습을 실시

결과:

→ 단순 프롬프팅, 지식 학습 모델 보다 수행 능력 및 안정성이 증가

→ (해로운 응답, Hallucination 감소)

3. 거대 언어 모델의 추론

디코딩

디코딩 알고리즘

정의 : 다음 단어를 선택하는 방법

거대 언어 모델의 자동회귀 생성

- 학습이 완료된 거대 언어 모델은 순차적 추론을 통한 **토큰별 생성**으로 응답을 생성
- **EOS 토큰이 생성**되거나 **사전에 정의된 토큰 수에 도달**하면 추론과 생성을 멈추고 응답을 제공

알고리즘	방법	장점	단점
그리디 디코딩	가장 확률 높은 토큰을 다음 토큰으로	사용 쉬움	바로 직후만 고려하기에 최선 선택이 아닐 수 있음
빔 서치 k개 = beam size → 각 후보마다 LLM 추론을 수행하기 때문	확률이 높은 k개의 후보를 동시에 고려 최종 응답이 좋은 응답으로 생성될 확률 높음	계산 비용이 많이 늘어남	
샘플링	거대 언어 모델이 제공한 확률을 기준으로 랜덤하게 생성	다양한 응답으로 생성 가능	생성된 응답 품질이 감소할 수 있음
샘플링 w. 탬퍼러쳐 T < 1 : 집중된 응답 T < 1 : 비슷한 답만 냄	하이퍼 파라미터 T를 통해 거대 언어 모델이 생성한 확률 분포를 임의로 조작함 T > 1 : 창의성 없고 품질 저하	T > 1 : 다양한 응답	
톱-K 샘플링 확률 낮은 건 버림 품질 향상됨	확률이 높은 K개의 토큰들 중 랜덤 확률에 따라 샘플링 잡음 단어 배제됨 확률 분포 모양 고려 못하고 K개의 고정된 후보만 고려 가		
톱-P 샘플링	K갯수는 고정, 누적 확률 P에 집중 → K가 자동 조절		
기존 알고리즘 대비 좋은 성능	품질과 다양성이 좋음	P값을 설정해 주어야 함	

프롬프트 엔지니어링

정의 : 원하는 답을 얻기 위해 모델에 주어지는 입력(프롬프트)을 설계 조정하는 기법

입력 프롬프트 = 지시 + 예시

- 지시 : 감정 분류 가능 수학과 코딩 같은 어려운 문제를 거대 언어 모델로 풀 수 있음

CoT 프롬프팅

Chain-of-Thought : (질문과 응답만 예시와 함께) 추론 과정도 예시에 포함되는 것

테스트 질문에 대해 추론을 생성, 응답을 유도 ⇒ 정확한 정답 생성 가능 (훈련에 없던 문제도 대응 가능)

거대 언어 모델 (PaLM)의 추론 성능을 크게 증가시킴

- PaLM : 구글에서 사용했던 거대 언어 모델

CoT로 인한 성능 향상은 모델 크기와 비례

예시가 있을 때 CoT가 강력 ⇒ 예시를 위한 추론 과정 수집 필요

0-shot CoT 프롬프팅

유인 문장으로 추론 생성하기 (Let’s think step by step)

질문 + 추론 ⇒ 정답 생성 (Therefore, the answer is)

0-shot 프롬프팅보다 높은 추론 성능 달성 가능

- 모델 크기가 임계점을 넘어야 효과 발휘 가능

적절하지 못한 문구일수록 역효과가 날 수 있음

4. 거대 언어 모델의 평가와 응용

거대 언어 모델의 평가

AI 모델의 평가 = 테스트 데이터

학습 단계에서 본 적 없고 질문과 정답을 알고 있다는 것이 가정임

거대 언어 모델의 성능은 많은 테스크에서의 성능을 종합적으로 판단해서 평가하기

디코딩 알고리즘, 입력 프롬프트에 따라 예측이 바뀔 수 있으니 두 가지도 고려하기

거대 언어 모델 평가 종류

정답 있는 경우

→ 예측과 정답을 비교해서 일치도 측정하기 (= 정확도)

정답 없는 경우

→ 사람이 임의의 정답을 작성, 예측과 비교 예시 단어와 유사도 측정 (벡터 공간 유사도 측정)

→ 정답과 무관하게 생성 텍스트 자체의 품질 측정

거대 언어 모델을 활용한 평가

- LLM-as-judge : 거대 언어 모델을 통해 생성 텍스트 평가하기

→ 생성된 텍스트의 상대적 선호 평가

ex) 문서 요약, 스토리 생성

- LMArena : 실제 유저 피드백 활용함, 거대 언어 모델 성능 측정 발법 중 가장 신뢰성 있는 방법 중 하나 ⇒ 높은 평가 비용 및 시간을 필요로 함

거대 언어 모델을 활용한 평가

- LLM-as-judge : 거대 언어 모델을 통해 상대적 선호 평가하기

  < 한계점 >

  1. 위치 편향 : 특정 위치 응답을 선호 ⇒ 위치 바꿔서 평가 후 평균 내기
  2. 길이 편향 : 길이가 긴 응답 선호 ⇒ 길이 영향을 통계적으로 제거해보기
  3. 자기 선호 편향 : 생성 모델 = 평가 모델인 경우 그것을 선호

거대 언어 모델의 응용

멀티모달 파운데이션 모델

다른 모달리티 데이터를 거대 언어 모델이 이해할 수 있도록 토큰화 및 추가 학습

합성 데이터 생성

- self-instruct : 사람이 175개 생성 → GPT가 52,000개 합성 데이터 생성

  노이즈 제거, 중복 제거 가능

  모델이 더 폭넓은 지시문 학습 가능

- Alpagasus : 프롬프팅을 통한 합성 데이터 품질 평가 및 필터링 제안

  저품질 합성 데이터 제거, 고품질 학습하기 ⇒ 전체 학습보다 빠르고 성능 좋음

거대 언어 모델의 한계

환각

사실과 다르고, 전적으로 지어낸 것인데도 정확한 정보인 마냥 자신감 있음 ⇒ 진위성 구별 어려움

사전 학습 데이터의 제한적 범위가 환각 현상 원인이 되기도 함 ⇒ 검색 증각 생성으로 해결 가능

** 검색 증강 생성 : RAG 대부분의 거대 언어 모델 서비스에 탑제되어 있음

탈옥

프롬프팅 엔지니어링을 통해 거대 언어 모델의 정렬을 우회할 수 있다는 것이 확인됨

여러 단계의 학습 과정에서 기인한 근본적인 한계로 인해 발생

- DAN 프롬프팅 ; Do Anything Now GPT는 특정 인물이나 사물에 감정/의견을 가질 수 없지만 DAN은 감정을 담아 주관적인 의견을 표현하기도 함

AI 텍스트 검출

무분별한 사용이 여러 문제 생성 ⇒ LLM이 만든 텍스트를 AI가 탐지 구분 가능