2-2 텍스트파운데이션 모델

목차

1. 자연어 처리 및 텍스트 파운데이션 모델 (“거대 언어 모델 “)
    1. 텍스트 파운데이션 모델 살펴보기
        1-1. 텍스트 파운데이션 모델(거대 언어모델)이란?
        1-2. 거대 언어 모델 예시
    2. 거대 언어 모델의 학습
        2-1. 지시학습(Instruction tuning)
        2-2. 선호 학습
    3. 거대 언어 모델의 추론
        3-1. 디코딩(Decoding) 알고리즘
        3-2. 프롬프트 엔지니어링
    4. 거대 언어 모델의 평가와 응용
        4-1. 거대 언어 모델의 평가
        4-2. 거대 언어 모델의 응용/한계

1. 자연어 처리 및 텍스트 파운데이션 모델 (“거대 언어 모델 “)

학습목표

- 텍스트 파운데이션 모델(거대 언어모델)이 무엇인지 이해한다
- 거대 언어 모델의 학습 및 추론 방식을 이해한다
- 거대 언어 모델의 응용 및 한계를 파악하고, 실습을 통해 실제 사례에 적용한다

학습시작 : 파운데이션 모델

파운데이션 모델의 예시

- 텍스트 생성 (ChatGPT)
- 비디오 생성 (SORA)
- 멀티모달 입력 (이미지,비디오,오디오) , 멀티모달 출력(오디오,텍스트) 출력 생성 (GPT-4o)

파운데이션 모델 이전 : 새로운 태스크 → 해당 태스크에 대한 별도의 학습 필요

파운데이션 모델 : 새로운 태스크 → 자세한 설명(프롬프트)을 입력해주는 것으로 충분

- ChatGPT : 텍스트 파운데이션 모델
- SORA : 비디오 파운데이션 모델

파운데이션 모델의 구성 요소 3가지

- 빅데이터
    - 인터넷에 존재하는 데이터 수가 기하급수적으로 증가
    - 딥러닝 기반 AI 모델은 학습 데이터가 늘어날 수록 성능이 중가
- 자가학습 알고리즘 (Self-supervised Learning)
    - 사람이 정답을 알려줄 필요 X
- 어텐션(Attention) 기반 트랜스포머(Transformer) 모델
    - 더 많은 데이터를 학습할 수 있는 인공신경망 구조

1. 텍스트 파운데이션 모델 살펴보기

1-1. 텍스트 파운데이션 모델(거대 언어모델)이란?

파운데이션 모델의 3가지 구성 요소는 기존 언어모델에도 이미 포함되어 있었다.

→ 어떤 차이때문에 성능이 달라진걸까?

ex) GPT-2

- 언어모델이 추가 학습 없이도 텍스트 지시를 통해 새로운 태스크를 어느정도 수행할 수 있음을 확인
  - 거대언어모델의 시발점
- 당시 가장 큰 GPT-2 모델조차도 underfitting된 결과를 보여줌 → 모델 크기를 늘리면 성능이 더 좋아지지 않을까? (발전 가능성 시사)

텍스트 파운데이션 모델(거대언어모델)의 특이점

- 규모의 법칙 (Scaling Law) : 더 많은 데이터, 큰 모델, 긴 학습 → 더 좋은 성능
- 창발성 (Emergent Property) : 특정 규모를 넘어서면 갑자기 모델에서 발현되는 성질
  - In-context Learning : 주어진 설명과 예시만으로 새로운 태스크를 수월하게 수행
  - few-show computing에 반응하기 시작
  - 추론(reasoning) 능력

텍스트 파운데이션 모델 (or, LLM, 거대언어모델) 이란?

- 기존 대비 더 큰 모델 (7B 이상)이 더 많은 데이터 (1T 이상)에서 학습되어 창발성이 나타나기 시작한 언어모델
- ChatGPT, Claude 3, Gemini, LLAMA2, deepseek, mistralAI

1-2. 거대 언어 모델 예시

폐쇄형 거대 언어모델 (Closed)

- ChatGPT(OpenAI) : 가장 많은 활성 유저스 ,전반적으로 뛰어난 성능
- Claude (Anthropic) : 안전 지향적 모델, 코딩 관련 작업에 특히 뛰어난 성능
- Gemini(Google) : 가장 긴 입력 및 출력 (1M 이상) 지원, 뛰어난 멀티 모달 성능
- 장점 : 일반적으로 더 우수한 성능 및 최신 기능을 갖고 있으며, 사용하기 쉬움
- 단점 : 비용 발생, 모델이나 출력에 대한 정보가 제한적으로 제공

개방형 거대 언어모델(Open Source)

- LLaMA(Meta), Gemma(Google), Qwen(Alibaba)
- 장점 : 무료 다운로드 및 사용, 모든 정보(모델 구조, 소스코드)가 공개되어 있음
- 단점 : 충분한 계산 자원 필요, 상대적으로 낮은 성능

2. 거대 언어 모델의 학습

GPT-3 : 거대 언어 모델의 시초

- 가장 큰 버전의 GPT-3 : 1750억 개의 매개변수 → 이전 언어 모델 대비 최소 10배 이상 큰 모델
- 본격적인 In-context learning 능력이 나타난 언어 모델
- 학습 방법 : 다음 토큰 예측 (Next token prediction)

- 학습 데이터 : 3000억 토큰 (4TB 텍스트 데이터)
- 학습 비용 : 150억원 추산

다음 토큰 예측 기반 거대언어모델의 한계

- 사람의 지시에 대해 올바르지 않은 응답을 생성하거나, 유해한 응답 생성

정렬 학습(Alignment) 의 도입 : 거대언어모델의 출력이 사용자의 의도와 가치를 반영하도록 하는 것

- 두 가지 학습 방법
    1. 지시 학습 (Instruction tuning) : 주어진 지시에 대해 어떤 응답이 생성되어야 하는지
    2. 선호 학습 (preference learning) : 상대적으로 어떤 응답이 더 선호되어야 하는 지

2-1. 지시학습(Instruction tuning)

지시학습 : 주어진 지시에 대해 어떤 응답이 생성되어야 하는 지 알려주는 것

지시학습 : 거대 언어모델을 다양한 지시 기반 입력과 이에 대한 응답으로 추가 학습

- ex) FLAN(Fine-tuned Language Net) (Goolgle Research)
- 학습 방법 : 주어진 입력을 받아서 이에 대한 응답을 따라 하도록 지도 추가 학습(SFT)
    - 지도추가학습(Supervised Fine-Tuning, SFT)와 동일
- 아이디어 : 모든 자연어 태스크는 텍스트 기반 지시(instruction)와 응답으로 표현할 수 있지 않을까?
- 학습 데이터의 다양한 증대를 위해, 각 태스크를 다양한 지시(탬플릿)로 표현할 수 있음
- 기존 NLP 태스크 데이터를 지시 학습을 위한 데이터로 수정하여 학습 및 테스트에 활용
- 학습시에 보지 못한 지시에 대한 일반화 성능 평가를 위해, 관련 없는 태스크들을 테스트에 별도로 활용
- 실험 결과 : 예시 없이도(zero-shot) 새로운 지시에 대해 올바른 응답을 내놓는 성능이 크게 증가

- 지시 학습 성능 향상 요인

1. 학습 태스크의 다양성
2. 모델 규모 (일정 규모 이상일 때 효과적)
3. 지시 표현 방식 (사람과 대화하듯 자연스러운 지시가 효과적)

- 효과 : 보지 못한 태스크에서도 zero-shot 성능이 향상된다.

지시 학습의 한계

- 주어진 입력에 대해 적절한 하나의 응답이 있다고 가정하며, 정답이 정해져 있지 않은 개방형 태스크에서는 한계가 있음

2-2. 선호 학습

Preference Learning : 다양한 응답 중 사람이 더 선호하는 응답을 생성하도록 추가학습

- 다양한 응답은 모델이 생성, 응답 간의 선호도는 사람이 제공
- ChatGPT를 만들기 위한 핵심 알고리즘!
- Instruction GPT의 핵심 아이디어 : 사람의 피드백을 통한 강화학습(RLHF)
- 학습 방법

1. 지시 학습을 통한 텍스트 파운데이션 모델의 추가학습
    - 실제 유저로부터 다양한 지시 입력을 수집, 해당 입력에 대해 훈련된 사람 주석자들이 정답 데이터를 생성
2. 사람의 선호데이터를 수집하여 보상 모델을 학습 (Reward Model)

    - 주어진 입력에 대한 선택지는 모델이 생성, 다양한 선택지에 대한 선호도는 사람이 생성
    - 사람과 일치한 선호도를 출력할 수 있도록 보상 모델을 지도 학습
        - 사람이 선호하는 응답이 입력으로 주어짐 → 높은 보상을 출력
3. 보상이 높은 응답을 생성하도록 강화학습을 통해 추가학습
- 1,2 단계에서 보지 못한 질문에 대해 사람의 추가적인 개입 없이 학습된 모델들을 통해 추가 학습
- 지시 학습된 모델을 보상 모델 기반 강화 학습을 통해 한번 더 추가 학습

- 결과
    - 유저의 지시를 얼마나 잘 수행하는 지를 사람이 직접 평가
        - 단순 프롬프팅이나 지시 학습에 비해 발전된 지시 수행능력을 보여줌
    - 얼마나 안전한 응답을 생성하는 지 평가
        - 기존 대비, InstructGPT는 해로운 응답(RalToxicity)과 거짓말(Hallucination, TruthfulQA)을 덜 생성
- ex) LLaMA2
    - RLHF와 대화 데이터를 활용한 LLaMA2 Chat 모델 공개 : 당시 가장 우수한 성능

3. 거대 언어 모델의 추론

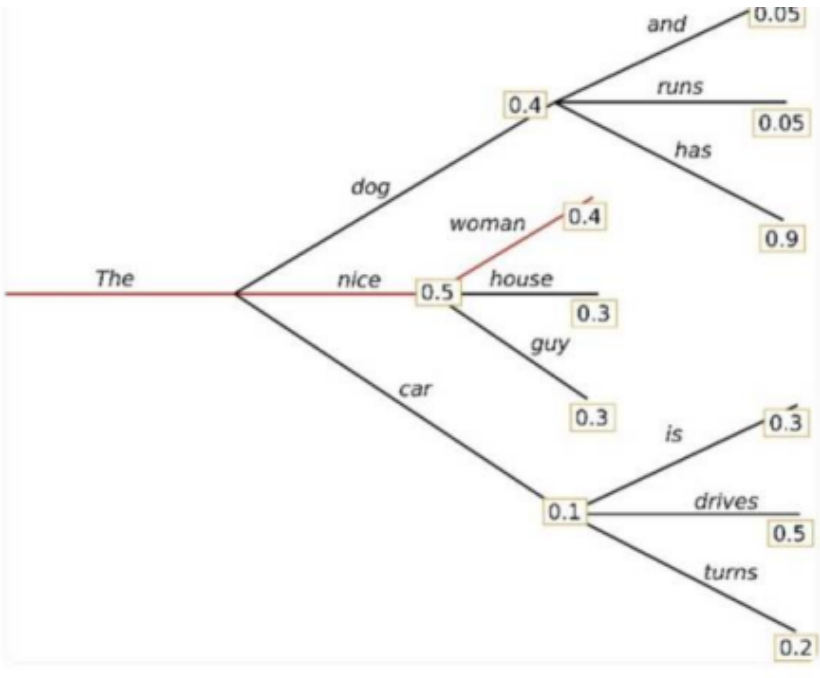
3-1. 디코딩(Decoding) 알고리즘

거대 언어 모델의 자동 회귀 생성 (Auto-regressive Generation)

- 학습이 완료된 거대 언어 모델은 어떻게 응답을 생성할까? → 순차적 추론을 통한 토큰별 생성
    - EOS(end of sentenca) 토큰이 생성 or 사전에 정의된 토큰 수에 도달하면 멈추고 응답 제공
- Goal : 주어진 입력 x = [x1,...xI] 에 대해 다음 토큰 xI+1 생성
    - 거대언어모델 : 입력 x에 대해 다음 토큰에 대한 확률분포 phat(x) 를 제공
- 디코딩(decoding) 알고리즘 : phat(x) 로부터 xI+1 을 생성하는 알고리즘

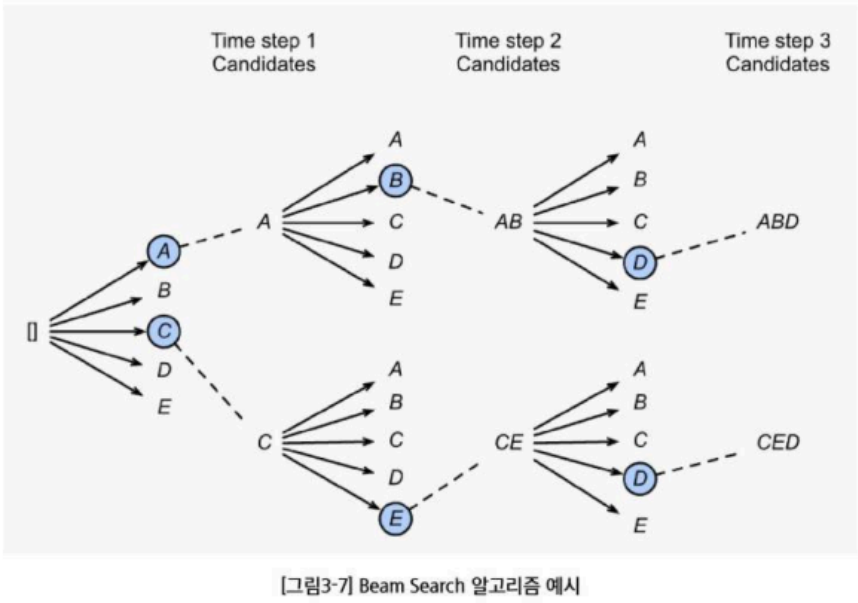
디코딩 알고리즘 종류

1. Greedy Decoding
    - 핵심 아이디어 : 가장 확률이 높은 다음 토큰을 선택
    - 장점 : 사용하기 쉽다
    - 단점 : 직후만 고려하기 때문에 생성 응답이 최종적으로 최선이 아닐 수 있다



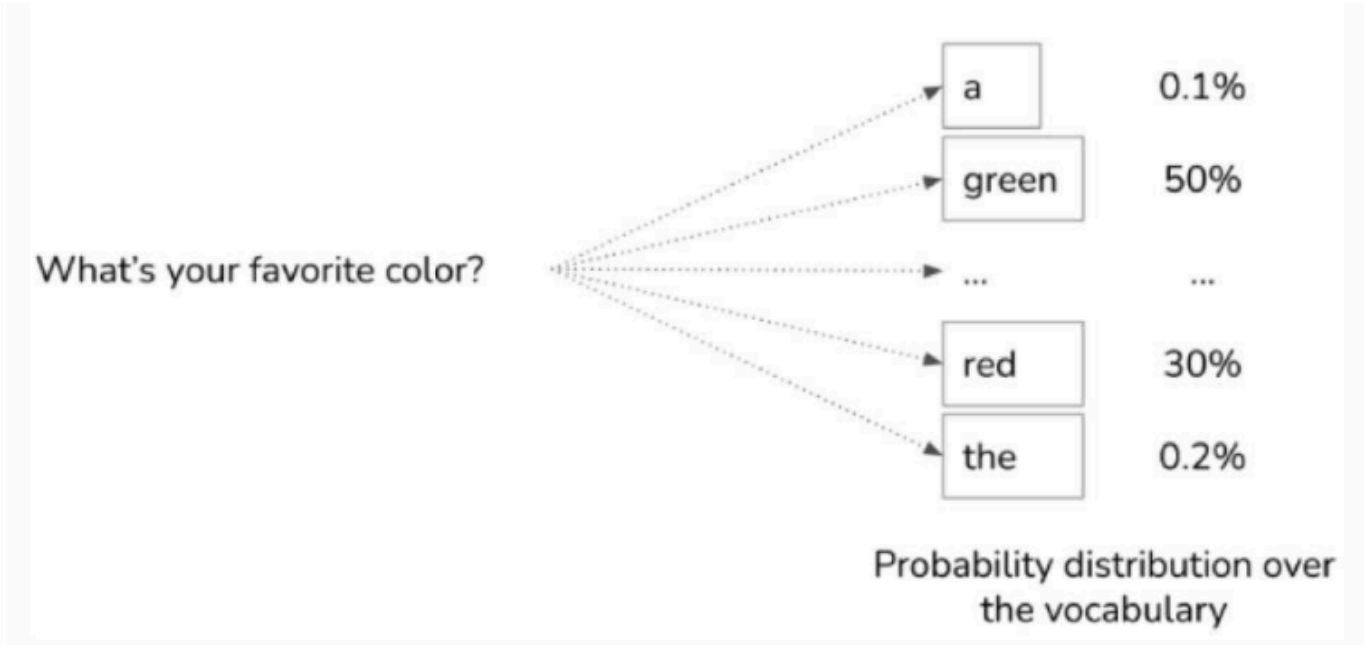
[그림3-6] Greedy Decoding을 통한 응답 생성 과정

2. Beam Search
    - 핵심 아이디어 : 확률이 높은 k개(beam size)의 후보를 동시에 고려
        - 고르는 기준 : 누적 생성 확률
    - 장점 : 최종적으로 좋은 응답 생성 확률이 높다
    - 단점 : 계산 비용이 많이 늘어난다. (각 후보마다 LLM추론을 수행)



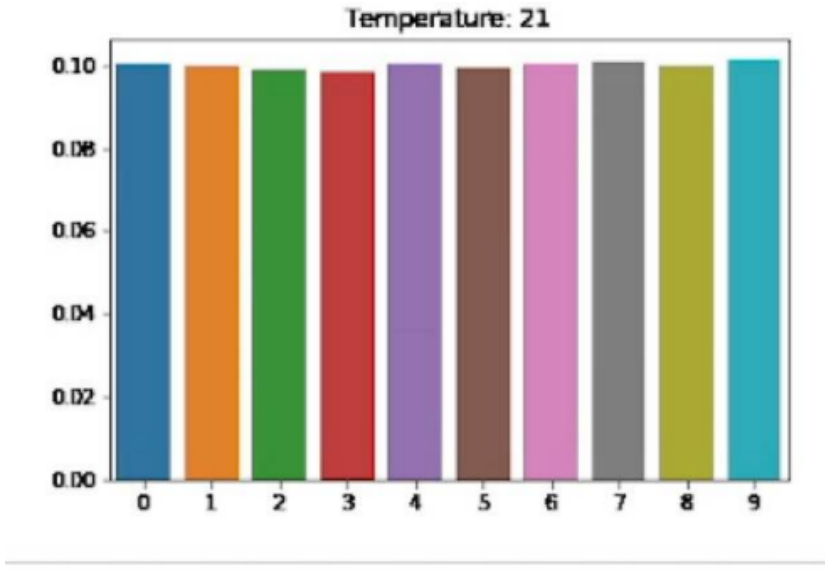
[그림3-7] Beam Search 알고리즘 예시

3. Sampling
    - 핵심 아이디어 : 거대 언어 모델이 제공한 확률에 비례해서 랜덤하게 생성
    - 장점 : 다양한 응답을 생성할 수 있음
    - 단점 : 생성된 응답의 품질이 감소할 수 있음



[그림3-9] Sampling 기반 디코딩 예시

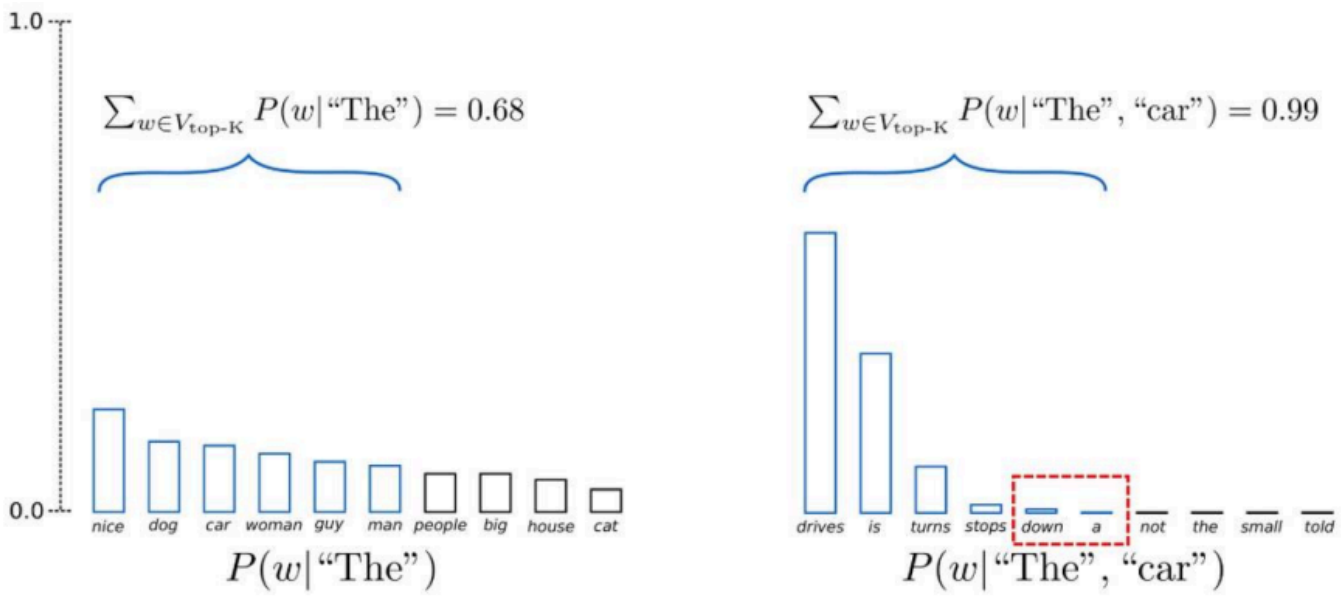
4. Sampling with Temperature
    - 핵심 아이디어 : 하이퍼 파라미터 T를 통해 거대 언어 모델이 생성한 확률 분포를 임의로 조작
        - T>1 : 확률 분포를 Smooth 하게 만듦 (더 다양한 응답 생성)
        - T<1 : 확률 분포를 Sharpe 하게 만듦 (기존에 확률이 높은 응답에 집중)



[그림3-10] Temperature 값 변화에 따른 확률 분포의 변화

5. Top-K Sampling

- 핵심 아이디어 : 확률이 높은 K 개의 토큰들 중에서 랜덤하게 확률에 따라 샘플링
- 장점 : 품질이 낮은 응답을 생성할 가능성을 줄일 수 있음
- 단점 : 확률 분포의 모양에 상관 없이 고정된 K개의 후보군을 고려



6. Top-P Sampling (or Nucleus Sampling)

- 핵심 아이디어 : K를 고정하는 대신, 누적 확률 (P) 에 집중하여 K를 자동으로 조절
- 다양한 평가 지표에서 기존 디코딩 알고리즘들 대비 좋은 성능을 달성

ex) 디코딩 알고리즘 예시 : ChatGPT

3-2. 프롬프트 엔지니어링

입력 프롬프트 = 지시(instruction) + 예시(few-shot examples)

- 어떻게 지시를 주는 지, 어떤 예시를 보여주는 지가 거대 언어 모델의 성능에 크게 영향을 미침

→ 프롬프트 엔지니어링 : 원하는 답을 얻기 위해 모델에 주어지는 입력(프롬프트)을 설계,조정하는 기법

프롬프트 엔지니어링의 요소

- 지시(Instruction)
  - 감정 분류와 같은 쉬운 문제 뿐만 아니라 수학, 코딩과 같은 어려운 문제를 거대 언어 모델로 푸는 것에 많은 관심 집중
- Chain-of-Thought(CoT) 프롬프팅
  - 아이디어 : 단순히 질문과 응답만을 예시로 활용하는 것이 아니라, 추론(Reasoning) 과정도 예시에 포함
    - 질문에 대해 추론을 생성하고 응답하도록 유도함으로써, 더 정확한 정답 생성 기대 가능

    - 결과 : CoT는 거대 언어모델(PaLM)의 추론 성능을 크게 증가시킴
        - PaLM : 구글의 가장 큰 거대 언어 모델
    - CoT로 인한 성능 향상은 모델 크기가 커질수록 더 확대됨
    - 다른 추론 태스크 : 마지막 단어 연결
    - 단점
        - 예시 기반 CoT는 강력하지만, 예시를 위한 추론 과정을 수집해야 하는 문제 발생

      → 예시 없이도 거대 언어 모델의 추론 성능 강화 필요성

- Zero-shot CoT 프롬프팅
    - 방법
        1. 추출 문장을 통한 추론 생성
        2. 주어진 질문과 생성된 추론을 통한 정답 생성
    - 결과
        - Zero-shot CoT 는 기존 zero-shot 프롬프팅보다 훨씬 높은 추론성능 달성
        - Zero-shot CoT는 모델 크기가 임계점을 넘어야 효과
        - 추출 문장에 따른 성능 차이 발생

4. 거대 언어 모델의 평가와 응용

4-1. 거대 언어 모델의 평가

평가(Evaluation) : 구축한 시스템이 실제로 잘 동작하는 지 확인하는 단계

- 평가의 3요소
    - 목표 : 시스템으로 무엇을 달성하고자 하는지
    - 평가 방법 : 어떤 방법으로 평가할 것인지
    - 평가 지표 : 어떻게 성공 여부를 판단할 것인지

AI 모델의 평가 : 테스트 데이터

- 핵심 아이디어 : 학습 단계에서 본 적이 없고, 질문과 정답을 알고 있는 테스트 데이터로 모델의 성능 평가

거대 언어 모델의 평가

- 특정 태스크에서 학습된 기존 AI모델과 달리, 거대 언어 모델은 다양한 태스크에 대해 동시 학습됨

  → 따라서 거대언어모델을 평가하기 위해 많은 태스크에서 종합적으로 판단할 필요가 있음

  → 디코딩 알고리즘, 입력 프롬프트에 따라 같은 질문에 대한 예측이 바뀌므로, 공평한 기준이 필요

1. 정답이 정해져 있는 경우

- 예측과 정답을 비교하여 일치도를 측정 → 정확도 (Accuracy)
    - ex) MMLU (Massive Multitask Language Understanding) 벤치마크 : 57개의 다양한 분야의 객관식 문제

2. 정답이 정해져 있지 않은 경우

- ex) 스토리 생성, 번역, 요약
- 방법1. 사람이 임의의 정답을 작성 및 이와 예측을 비교 → 단어 수준에서의 유사도 측정 or 벡터 공간에서의 유사도 측정
- 방법2. 정답과 무관하게 생성 텍스트 자체의 품질만을 측정 → Perplexity(PPL) 얼마나 문장이 확률적으로 자연스러운지 측정
    - 여러 가지 측면에서 평가를 하고 싶다면?
    - 전문가를 고용해서 평가하는 대신 거대 언어 모델로 평가하면 어떨까?

- 방법3. 생성 텍스트의 상대적 선호를 평가
  - ex) LMArena(실제 유저 피드백을 활용, 가장 신뢰성 있는 방법 중 하나)

    → 높은 평가 비용 및 시간 필요 → 이것도 거대 언어 모델로 대체할까?

거대 언어 모델을 활용한 평가

- LLM-as-judge(or G-Eval) : 거대 언어 모델을 통해 생성 텍스트를 평가
  - 유저는 (1) 질의, (2) 평가하고자하는 텍스트, (3) 평가 기준 제공
  - 거대언어모델은 평가 결과(점수,이유)를 제공
  - ex) GPT-4를 활용한 평가는 기존 평가지표보다 더 사람과 유사한 결과를 보임
- LLM-as-judge : 거대 언어 모델을 통해 생성 텍스트의 상대적 선호를 평가
  - 단점
    1. 위치 편향 : 특정 위치의 응답을 상대적으로 선호

       → 순서를 바꿔서 두 번 평가해서 평균내기
    2. 길이 편향 : 품질과 무관하게 길이가 긴 응답을 상대적으로 선호

       → 길이가 미치는 영향을 통계적으로 제거해서 해결
    3. 자기 선호 편향 : 생성 모델이 평가 모델과 같은 경우, 이를 선호

4-2. 거대 언어 모델의 응용/한계

거대 언어 모델의 응용 : 멀티모달 파운데이션 모델

ex) GPT-4o : 멀티모달 입력 (이미지,비디오,오디오), 멀티모달 출력 (텍스트 , 오디오)

- 핵심 아이디어 : 다른 모달리티 데이터를 거대 언어 모델이 이해할 수 있도록 토큰화 및 추가 학습

거대 언어 모델의 응용 : 합성 데이터 생성

- Self-instruct : 175개의 데이터를 사람이 작성한 뒤, GPT-3을 통해 52000개의 합성 데이터 생성
- 결과 : 기존 InstructGPT에서 활용된 사람이 만든 데이터와 비슷한 성능 달성
  - ex) Alpaca(Meta) : LLaMA-1 기반 최초 개방형 instruction following 모델 학습에 사용
  - Alpagasus : 프롬프팅을 통한 합성 데이터의 품질 평가 및 필터링 제안
    - Alpaca의 합성데이터 52000개 중 9000(4.5이상)개 정도를 필터링 → 다시 학습
    - 결과 : 전체 합성 데이터를 사용한 경우보다 높은 성능 달성

거대 언어 모델의 한계

- 환각(Hallucination) : 사실과 다르거나, 전적으로 지어낸 내용임에도 불구하고 정확한 정보와 동일한 자신감과 유창함으로 응답을 생성
  - 확률적으로 다음 토큰 예측을 통해 응답을 생성하기 때문
  - 진위성 구별이 어려움
  - 사전학습 데이터의 제한적인 범위가 요인이기도 함

    → 검색 증강 기법(RAG)를 통해 해결 가능
- 탈옥(Jailbreaking) : 프롬프팅 엔지니어링을 통해 거대 언어모델의 정렬을 우회할 수 있다는 것이 확인됨
  - ex) DAN 프롬프팅
  - 여러 단계의 학습 과정에서 기인한 근본적인 한계 때문에 발생, 다양한 탈옥/방어 방법이 연구중
- AI 텍스트 검출 : 거대 언어 모델의 무분별한 사용이 학교 및 회사에서 새로운 문제를 발생
  - LLM이 만든 텍스트를 구분 또는 탐지할 수 있을까? 어느 정도 가능

문제1. 다음 중 거대 언어 모델과 BERT, GPT-1과 같은 기존 언어모델의 차이점으로 올바른 것은?

1. 다음 토큰 예측에 기반한 자기 지도 학습 방법
2. 트랜스포머에 기반한 모델구조
3. 모델 및 학습 규모가 커지면서 나타난 창발성
4. 이미지와 같은 도메인을 포함한 새로운 멀티모달 학습 데이터

정답 : 3

문제2. 다음 중 지시 학습과 선호 학습에 대해 올바르지 않은 것은?

1. 효과적인 지시학습을 위해서는 다양한 지시 데이터가 필요하다
2. 크기가 충분하지 않은 거대언어모델은 지시학습 후에 성능이 떨어질 수 있다
3. 보상 모델은 사람의 선호를 모방하도록 학습된다
4. 강화 학습 기반 선호 학습의 모든 과정에는 사람의 개입이 필요하다

정답 : 4

문제3. 다음 중 거대언어 모델의 다양한 디코딩 알고리즘에 대한 설명으로 올바르지 않은것은?

1. Greedy Decoding은 가장 확률이 높은 토큰을 다음 토큰으로 생성한다.
2. Beam Search는 확률이 높은 응답 후보를 동시에 고려하기 때문에 더 많은 생성 비용을 필요로 한다
3. 1 이상의 Temperature를 사용할 경우 더 다양한 응답이 생성될 확률이 감소한다
4. Top-K Sampling은 생성확률이 높은 K개의 토큰만을 생성 후보로 두고 확률 값에 따라 무작위로 다음 토큰을 생성한다

문제4. 다음중 거대 언어 모델의 평가 및 한계에 대해 올바르지 않은 것은?

1. 주어진 질문에 대해 정답을 모르는 경우에도, LLM-as-judge 방식을 통해 생성 응답에 대해 평가할 수 있다
2. LLM-as-judge 방식을 선호 판단에 활용하는 경우, 길이가 짧은 응답을 선호하는 길이 편향이 있을 수 있다
3. GPT-5, Claude-4와 같은 최신 거대 언어 모델은 안전성 훈련이 잘 되어 있으므로, 탈옥 현상이 일어나지 않는다.
4. 거대 언어 모델이 생성한 응답은 탐지가 가능할 수 있으므로, 무분별하게 사용해서는 안된다