10주차 논문 리뷰: LLaMA: Open and Efficient Foundation Language Models

0. Abstract

- **LLaMA(Large Language Model Meta AI)**는 7B에서 65B까지 다양한 크기의 파라미터를 가진 대규모 언어 모델
  - 수조 개의 토큰(token)으로 학습되었으며, 공개적으로 이용 가능한 데이터셋만을 활용하여 학습됨
  - 소유권이 있거나 접근이 제한된 데이터셋을 사용하지 않고도 최첨단 성능을 달성할 수 있음을 증명
  - LLaMA-13B는 대부분의 벤치마크에서 GPT-3(175B)를 능가
  - LLaMA-65B는 Chinchilla-70B나 PaLM-540B와 같은 최고의 모델들과 비교 가능한 성능을 보임

1. Introduction

- **대규모 언어 모델의 필요성**
  - LLM(Large Language Models)은 대규모 텍스트 데이터로 학습되어, **텍스트 지침**이나 몇 가지 예제만으로 새로운 작업을 수행할 수 있는 능력을 보여줌
  - Few-shot 학습 능력은 **모델의 크기가 충분히 클 때** 나타나며, 이를 통해 규모 확장에 대한 많은 연구가 진행

- **효율적인 학습 법칙**

- 최근 연구(Hoffmann et al., 2022)는 모델 크기가 반드시 크다고 해서 성능이 최고가 되는 것은 아니며, **더 많은 데이터를 더 작은 모델이 학습했을 때 더 나은 성능**을 얻을 수 있음을 보여줌
- Hoffmann의 스케일링 법칙(Scaling Laws)을 바탕으로, 주어진 예산 내에서 데이터와 모델 크기를 가장 효율적으로 조정할 방법을 제안
- 연구 목표: 추론(inference) 예산을 최적화하면서도 경쟁력 있는 성능을 발휘하는 모델을 개발하는 것

- **LLaMA의 주요 특징**
  - **7B에서 65B 파라미터까지 다양한 모델** 제공
  - LLaMA-13B는 GPT-3(175B)와 비교해 **10배 더 작은 크기**임에도 불구하고 뛰어난 성능을 보여줌
  - LLaMA-65B는 Chinchilla(70B) 및 PaLM(540B)과 같은 대형 모델과 경쟁 가능한 성능을 제공

- **공개 데이터만 활용**
  - LLaMA는 공개적으로 접근 가능한 데이터만 사용하며, Chinchilla나 PaLM처럼 비공개 데이터(예: 소셜 미디어 대화, 특정 출판물 등)를 활용 X

  ⇒오픈소스와의 호환성을 보장

- **Token**: 자연어 처리에서 모델 학습의 단위로 사용되는 텍스트의 최소 단위. 단어, 문자, 혹은 서브워드(subword) 단위가 될 수 있음
- **Scaling Laws (스케일링 법칙)**: 모델의 크기, 학습 데이터량, 학습 시간 등의 변수를 조정했을 때 성능에 미치는 영향을 정량적으로 분석한 법칙.
- **Inference**: 학습된 모델이 새로운 데이터에 대해 예측을 수행하는 단계. 일반적으로 학습보다 연산 비용이 적게 듦.
- **Few-shot Learning**: 최소한의 예제만을 제공하여 모델이 새로운 작업을 수행하도록 학습하는 방법.

2. Approach

2.1 Pre-training Data (사전 학습 데이터)

- LLaMA의 학습 데이터셋은 다양한 소스를 혼합한 결과물

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

1. **CommonCrawl (67%)**:
    - CommonCrawl 덤프(2017~2020)를 전처리하여 활용
    - CCNet 파이프라인(Wenzek et al., 2020)을 사용하여 비영어 데이터를 제거하고, 낮은 품질의 콘텐츠는 n-gram 모델을 통해 필터링
    - **Wikipedia**에 자주 인용되는 문서를 우선적으로 선택하
2. **C4 (15%)**:
    - CommonCrawl의 변형으로, 품질 필터링과 언어 식별이 포함된 데이터셋(Raffel et al., 2020)
    - 구두점이나 특정 언어 패턴을 기반으로 데이터 품질을 평가
3. **GitHub (4.5%)**:
    - Google BigQuery에서 제공되는 GitHub 데이터를 사용.
    - Apache, BSD, MIT 라이선스를 가진 프로젝트만 포함
    - 코드 주석 및 문서 데이터를 정제하여 활용.
4. **Wikipedia (4.5%)**:
    - 랜덤 샘플링 대신 참조된 페이지를 우선적으로 선택.
5. **Books (4.5%)**:
    - Project Gutenberg와 같은 퍼블릭 도메인 북 데이터를 포함.
6. **ArXiv (2.5%)**:

- 과학 논문 데이터를 추가하여 텍스트 다양성을 강화.

7. **StackExchange (2%)**:
    - 고품질 질의응답 데이터를 활용하였으며, HTML 태그를 제거하여 정제함.

2.2 Architecture (모델 구조)

: Transformer 아키텍처(Vaswani et al., 2017)를 기반으로 하며, 최신 기술을 통합하여 성능을 개선

1. **Pre-normalization (GPT-3)**:
    - 각 Transformer 서브레이어의 입력을 정규화하여 안정성을 향상.
2. **SwiGLU Activation Function (PaLM)**:
    - ReLU 대신 SwiGLU(Shazeer, 2020) 활성화 함수를 사용.
3. **Rotary Embeddings (GPTNeo)**:
    - 기존의 절대 위치 임베딩 대신, RoPE(Su et al., 2021) 방식으로 대체

2.3 Optimizer (최적화 방법)

- **AdamW 옵티마이저**를 사용.
- 하이퍼파라미터:
    - β1 = 0.9, β2 = 0.95
    - 학습률은 최대 학습률의 10%에서 시작하며 점진적으로 증가(warmup).
    - 학습률 스케줄: 2,000 스텝 동안 선형 증가 후 감소.

2.4 Efficient Implementation (효율적 구현)

- 메모리 사용량을 줄이기 위해 **causal multi-head attention** 구현 최적화.
- 역전파 시 attention 가중치를 저장하지 않으며, 필요한 활성화 값만 재계산
- PyTorch의 자동 미분(autograd)을 사용하지 않고, 수동으로 역전파를 구현하여 효율성을 극대화

- **CCNet**: CommonCrawl 데이터를 전처리하는 파이프라인으로, 품질이 낮은 콘텐츠를 제거하고 언어 식별을 수행
- **RoPE**: 위치 임베딩 방법 중 하나로, Transformer 모델에서 순서 정보를 효과적으로 학습할 수 있도록 도움
- **AdamW**: Adam 옵티마이저의 변형으로, weight decay(가중치 감소)를 포함하여 과적합(overfitting)을 방지.

3. Main Results

3.1 Zero-shot 및 Few-shot 학습 성능 평가

: LLaMA는 **20개의 벤치마크**에서 Zero-shot과 Few-shot 작업 성능을 평가받았으며, 아래의 주요 결과를 보임

1. **Zero-shot Learning**:
   - 모델에 작업 설명과 테스트 예시를 제공하고, 적절한 답을 생성하거나 제안된 답을 평가하도록 함.
   - LLaMA-13B는 GPT-3(175B)보다 **10배 작은 크기**에도 불구하고 더 나은 성능을 보임.
   - LLaMA-65B는 **Chinchilla-70B**와 비교 가능하며, 일부 벤치마크에서 **PaLM-540B**를 초과함.
2. **Few-shot Learning**:
   - LLaMA는 GPT-3, PaLM 등 기존 모델들과 비교해 Competitive한 성능을 달성.

벤치마크 결과 요약

		BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA
GPT-3	175B	60.5	81.0	-	78.9	70.2	68.8	51.4	57.6
Gopher	280B	79.3	81.8	50.6	79.2	70.1	-	-	-
Chinchilla	70B	83.7	81.8	51.3	80.8	74.9	-	-	-
PaLM	62B	84.8	80.5	-	79.7	77.0	75.2	52.5	50.4
PaLM-cont	62B	83.9	81.4	-	80.6	77.0	-	-	-
PaLM	540B	**88.0**	82.3	-	83.4	**81.1**	76.6	53.0	53.4
LLaMA	7B	76.5	79.8	48.9	76.1	70.1	72.8	47.6	57.2
	13B	78.1	80.1	50.4	79.2	73.0	74.8	52.7	56.4
	33B	83.1	82.3	50.4	82.8	76.0	**80.0**	**57.8**	58.6
	65B	85.3	**82.8**	**52.3**	**84.2**	77.0	78.9	56.0	**60.2**

Table 3: **Zero-shot performance on Common Sense Reasoning tasks.**

- **Common Sense Reasoning** (Table 3):
  - **LLaMA-65B**: 여러 과제에서 Chinchilla-70B와 PaLM-540B보다 우수.
  - **LLaMA-13B**: GPT-3보다 뛰어난 성능.
- **Closed-book Question Answering** (Table 4 & Table 5):
  - **Natural Questions** 및 **TriviaQA**에서 높은 정확도 기록.
  - LLaMA-13B는 Chinchilla-70B와 유사한 성능을 보임.
- **Reading Comprehension** (Table 6):
  - LLaMA-65B는 PaLM-540B와 비교 가능한 성능을 기록.
- **Mathematical Reasoning** (Table 7):
  - LLaMA-65B는 일부 과제에서 PaLM보다 우수하며, Minerva-62B를 초과.
- **Code Generation**:
  - 코드 작성 작업에서 높은 정확도와 효율성 기록.

3.2 Closed-book Question Answering

		0-shot	1-shot	5-shot	64-shot
Gopher	280B	43.5	-	57.0	57.2
Chinchilla	70B	55.4	-	64.1	64.6
LLaMA	7B	50.0	53.4	56.3	57.6
	13B	56.6	60.5	63.1	64.0
	33B	65.1	67.9	69.9	70.4
	65B	**68.2**	**71.6**	**72.6**	**73.0**

Table 5: **TriviaQA.** Zero-shot and few-shot exact match performance on the filtered dev set.

- **평가 방법**: 모델이 답변을 위해 필요한 문서에 접근하지 못하도록 설정.
- **결과**:
  - LLaMA-65B는 TriviaQA 및 Natural Questions에서 기존 대규모 모델을 능가.

3.3 Reading Comprehension

		RACE-middle	RACE-high
GPT-3	175B	58.4	45.5
PaLM	8B	57.9	42.3
	62B	64.3	47.5
	540B	**68.1**	49.1
LLaMA	7B	61.1	46.9
	13B	61.6	47.2
	33B	64.1	48.3
	65B	67.9	**51.6**

Table 6: **Reading Comprehension.** Zero-shot accuracy.

- **RACE 데이터셋**: 중·고등학교 영어 독해 문제를 기반으로 평가.
- **결과**:
  - LLaMA-13B는 GPT-3와 비교해 몇 퍼센트 더 높은 성능을 기록.
  - LLaMA-65B는 PaLM-540B와 유사한 결과를 달성.

3.4 Mathematical Reasoning

- **MATH**와 **GSM8K** 벤치마크:
  - 수학 문제 해결 능력을 평가
  - LLaMA-65B는 Minerva-62B를 능가하며 일부 수학적 문제에서 높은 정확도를 보임

- **Zero-shot Learning**: 별도의 학습 없이 새로운 작업을 바로 수행.
- **Few-shot Learning**: 몇 가지 예시만 제공하고 작업을 수행하도록 학습.
- **Closed-book QA**: 문서나 참고자료에 접근하지 않고 질문에 답변.
- **Mathematical Reasoning**: 모델의 수학적 문제 해결 능력을 평가.

3.5 Code Generation

<평가 방법>

- 모델은 HumanEval (Chen et al., 2021)과 MBPP (Austin et al., 2021) 두 가지 벤치마크를 기반으로 코드 생성 능력을 평가받음
- 작업:
  - 주어진 몇 문장의 프로그램 설명과 입출력 예제를 기반으로 Python 코드를 생성.
  - HumanEval에서는 함수 서명(function signature)을 추가 제공받고, 자연어 텍스트 설명과 테스트 케이스를 포함한 프롬프트를 활용.

<결과>

	Params	HumanEval		MBPP	
pass@		@1	@100	@1	@80
LaMDA	137B	14.0	47.3	14.8	62.4
PaLM	8B	3.6*	18.7*	5.0*	35.7*
PaLM	62B	15.9	46.3*	21.4	63.2*
PaLM-cont	62B	23.7	-	31.2	-
PaLM	540B	**26.2**	76.2	36.8	75.0
LLaMA	7B	10.5	36.5	17.7	56.2
	13B	15.8	52.5	22.0	64.0
	33B	21.7	70.7	30.2	73.4
	65B	23.7	**79.3**	**37.7**	**76.8**

- **Pass@k 지표**:
  - 생성된 코드 중 테스트를 통과하는 코드 비율을 측정.
  - LLaMA는 PaIM 및 LaMDA와 같은 기존 모델보다 뛰어난 성능을 기록.
  - **LLaMA-13B**: HumanEval 및 MBPP 모두에서 우수한 성과.
  - **LLaMA-65B**: 기존 PaLM-62B 및 LaMDA보다 뛰어남.

💡

- **Pass@k 지표**
  - Pass@k는 모델이 k개의 시도로 테스트를 통과하는 비율을 측정하는 지표
  - Pass@1: 모델이 한 번의 시도에서 정확한 코드를 생성하는 비율을 나타냄
  - Pass@100: 최대 100개의 시도를 기반으로 정확한 코드가 포함되었는지를 측정
- **Sampling Temperature**
  - Temperature는 모델의 출력 다양성을 조절하는 파라미터
  - 낮은 값(예: 0.1)은 더 결정론적(deterministic)인 출력을 생성
  - 높은 값(예: 0.8)은 다양하고 창의적인 출력을 생성하도록 유도
- **HumanEval**
  - 주어진 함수 설명과 입력-출력 예제를 기반으로 Python 함수 코드를 생성하는 과제
  - 코드의 기능적 정확성을 평가
- **MBPP (Mostly Basic Python Programming)**
  - Python 코드 작성에 초점을 맞춘 벤치마크
  - 초보자 수준의 프로그래밍 문제를 포함하며, HumanEval보다 간단한 테스트 케이스를 제공

3.6 Massive Multitask Language Understanding (MMLU)

<평가 방법>

- MMLU (Hendrycks et al., 2020) 벤치마크를 기반으로 다중 작업 언어 이해 성능을 평가.
- 작업:
  - 인문학, STEM, 사회 과학 등 다양한 분야의 다중 선택 질문에 답변.
  - **5-shot 학습** 설정.

<결과>

- LLaMA-65B는 Chinchilla-70B와 PaLM-540B를 여러 도메인에서 초과.
- Humanities, STEM, Social Sciences 전 분야에서 높은 성능을 기록.

- **MMLU 벤치마크**
  - 다양한 지식을 테스트하기 위해 설계된 다중 선택형 질문 데이터셋
  - 인문학, STEM, 사회과학 등 폭넓은 주제를 다룸
  - **5-shot 학습**: 모델이 5개의 예제를 보고 학습한 후 질문에 답하는 방식

- **LLaMA-65B의 경쟁력**
  - PaLM 및 GPT-3와 같은 기존 모델을 넘어서는 결과
  - 특히 Humanities(인문학)와 Social Sciences(사회 과학)에서 뛰어난 성능을 보임

  ⇒ LLaMA가 훈련에 사용한 데이터셋(예: ArXiv, Gutenberg, Books3) 덕분으로 보임.

4. Instruction Finetuning

<평가 방법>

- LLaMA 모델에 Instruction-tuned 데이터를 미세 조정하여 MMLU 성능을 비교.
- 목표:
  - 모델의 지침 준수 능력을 강화.
  - LLaMA-65B는 기본적으로 지침을 따르는 능력을 가짐.
  - Table 10에 상세 성능 기록.

<결과>

- 미세 조정 후 LLaMA의 성능은 68.9%로 증가.
- 기존 Instruction-tuned 모델(Flan-PaLM, OPT-IML)과 유사한 성능.

💡

- **Instruction Fine-tuning**
  - Instruction-tuned 모델은 **특정 작업에 대한 지침을 더 잘 이해하고 따르도록 미세 조정된 모델**
  - Flan-PaLM, OPT-IML 등은 Instruction-tuned 모델의 예시

- **LLaMA의 기본 성능**
  - LLaMA는 별도의 Instruction Fine-tuning 없이도 기본적으로 지침을 잘 따름

  ⇒ LLaMA가 훈련 과정에서 다양하고 광범위한 데이터를 학습했기 때문

- **MMLU 성능 증가**
  - Fine-tuning 이후 성능이 68.9%로 향상
  - 이는 Instruction Fine-tuning이 모델의 전반적인 성능을 강화할 수 있음을 보여줌

5. Bias, Toxicity, and Misinformation

<분석 방법>

- 모델이 학습 데이터에서 존재하는 편향을 증폭하거나 toxic 콘텐츠를 생성할 가능성을 평가.
- LLaMA-65B가 웹에서 수집된 데이터를 학습하기 때문에 **부정확한 정보 생성 가능성**이 존재.

<결론>

- 모델이 생성하는 콘텐츠의 잠재적 위험을 이해하기 위해 추가적인 벤치마크 평가가 필요.

5.1 RealToxicityPrompts

<평가 방법>

- **RealToxicityPrompts**(Gehman et al., 2020)를 사용하여 모델이 생성하는 텍스트의 toxicity 점수를 평가.
- 데이터셋은 약 100,000개의 프롬프트로 구성되며, PerspectiveAPI를 통해 자동으로 독성 점수를 측정.
- 독성 점수 범위:
  - **0**: 비독성 (Non-toxic).
  - **1**: 완전히 독성 (Fully toxic).
- 평가 항목:
  - **Basic**: 일반적인 독성 텍스트.
  - **Respectful**: "존중하는" 표현으로 시작하는 프롬프트에서 독성 텍스트 생성 여부 평가.

<결과>

- **LLaMA-65B**의 독성 점수는 0.128 (Basic) 및 0.141 (Respectful).
- 모델 크기가 커질수록 Respectful 프롬프트에서 toxicity 점수가 증가하는 경향을 보임.
- 독성 생성의 증가는 데이터 크기와 모델 크기의 관계 때문으로 추정.

5.2 CrowS-Pairs (편향 평가)

<평가 방법>

- **CrowS-Pairs** 데이터셋(Nangia et al., 2020)을 사용하여 모델의 사회적 편향을 평가.
- 평가 범주:
  - 성별, 종교, 인종/피부색, 성적 지향, 연령, 국적, 장애, 외모, 사회경제적 지위 등 9개 범주
- 방법:

- 각 문장은 고정관념(Stereotype)을 포함한 문장과 이를 대조하는 문장(Anti-stereotype)으로 구성.
- 두 문장의 perplexity를 비교하여 모델의 편향 정도 측정.

결과

- **LLaMA-65B**는 GPT-3(66.7) 및 OPT-175B(69.5)와 비교하여 평균 편향 점수 66.6으로 비슷한 수준
- 종교와 연령 관련 편향에서 상대적으로 높은 점수 기록

5.3 WinoGender (성별 관련 대명사 분석)

<평가 방법>

- **WinoGender**(Rudinger et al., 2018) 데이터셋을 사용하여 성별 대명사와 직업군 간의 관계 분석.
- 목표:
  - 대명사의 성별과 문맥에 기반한 정확한 대명사 참조 해석(Co-reference resolution).

결과

- **LLaMA-65B**:
  - "Their/them/someone" 대명사에서 높은 정확도(81.7) 기록.
  - "Her/her/she"와 "His/him/he" 대명사에서는 상대적으로 낮은 성능.
  - **Gotcha 케이스**(문맥이 직업군 성별과 일치하지 않는 경우)에서는 편향이 두드러짐.

5.4 TruthfulQA (정확성 평가)

<평가 방법>

- **TruthfulQA**(Lin et al., 2021) 벤치마크를 사용하여 모델이 진실하고 정확한 정보를 제공하는지 평가.
- 주요 지표:

- **Truthful**: 진실한 답변 비율.
- **Truthful+Informative**: 진실하며 유용한 답변 비율.

결과

- **LLaMA-65B**:
  - Truthful 점수: 0.52 (GPT-3: 0.25).
  - Truthful+Informative 점수: 0.48 (GPT-3: 0.19).
  - 모델 크기가 커질수록 더 정확한 답변을 제공하지만, 일부 부정확한 답변의 비율은 여전히 존재.

1. **PerspectiveAPI**
   - 독성 점수를 평가하는 Google의 API
   - 각 텍스트의 독성 수준을 0~1 범위로 평가

2. **CrowS-Pairs**
   - 사회적 편향을 측정하기 위해 설계된 데이터셋으로, 고정관념적 문장과 이를 반박하는 문장을 비교

3. **TruthfulQA**
   - 모델이 허위 정보를 제공하는지 측정
   - 실제 세계의 진리와 문화적 믿음 또는 전통 간의 차이를 포함

4. **Gotcha 케이스**
   - 문맥적 단서를 제공하지 않거나 반대로 오도하는 경우를 포함하는 특수한 테스트 케이스
   - LLaMA는 성별 편향으로 인해 이러한 케이스에서 낮은 성능을 보임

6. Carbon Footprint (탄소 발자국)

- **탄소 배출량 계산**
  - 모델 훈련에 사용된 전력 소비와 탄소 배출량을 측정
    - 공식을 사용하여 총 전력 소비를 탄소 배출량으로 변환:
      $CO2eq = MWh \times 0.385$

      $CO2eq = MWh \times 0.385CO\_2eq = MWh \times 0.385$

      ⇒ A100 GPU의 소비 전력(400W)을 기준으로 계산하여 결과의 신뢰성을 높임
    - 여기서 0.385는 미국의 평균 전력 탄소 강도

- **모델별 탄소 배출량**
  - **OPT-175B**: 137톤
  - **BLOOM-175B**: 183톤
  - **LLaMA 모델**:
    - LLaMA-7B: 14톤
    - LLaMA-13B: 23톤
    - LLaMA-33B: 90톤
    - LLaMA-65B: 173톤
- 훈련에 사용된 GPU와 시간:
  - GPU: NVIDIA A100 (400W, 80GB 메모리)
  - LLaMA-65B는 약 5개월 훈련 시간 소요

	GPU Type	GPU Power consumption	GPU-hours	Total power consumption	Carbon emitted (tCO_2eq)
OPT-175B	A100-80GB	400W	809,472	356 MWh	137
BLOOM-175B	A100-80GB	400W	1,082,880	475 MWh	183
LLaMA-7B	A100-80GB	400W	82,432	36 MWh	14
LLaMA-13B	A100-80GB	400W	135,168	59 MWh	23
LLaMA-33B	A100-80GB	400W	530,432	233 MWh	90
LLaMA-65B	A100-80GB	400W	1,022,362	449 MWh	173

- **환경적 시사점**
  - LLaMA는 상대적으로 낮은 전력 소비로 훈련 가능하며, 단일 GPU에서도 실행 가능
  - 이를 통해 탄소 배출을 줄이고, 대규모 모델의 접근성을 높이는 데 기여

7. Related Work (관련 연구)

- **언어 모델의 역사**
  - 초기 언어 모델:
    - N-gram 기반 확률 모델 (Bahl et al., 1983)
    - RNN(Recurrent Neural Networks)과 LSTM(Long Short-Term Memory)의 발전 (Elman, 1990; Hochreiter and Schmidhuber, 1997)
  - 최근:
    - Transformer 기반 모델이 long-term dependencies을 캡처하는 데 성공 (Vaswani et al., 2017).
    - BERT(2018), GPT(2018), 그리고 GPT-3(2020)가 NLP 분야의 큰 도약

- **Scaling Laws (스케일링 법칙)**
  - Hoffmann et al. (2022)의 연구는 모델 크기와 데이터 크기의 관계를 최적화
  - Chinchilla, PaLM 등 대규모 모델은 학습 데이터를 늘리며 성능을 극대화

8. Conclusion (결론)

- **LLaMA의 경쟁력**:
  - LLaMA-13B는 GPT-3보다 10배 작음에도 대부분의 벤치마크에서 이를 능가
  - LLaMA-65B는 Chinchilla-70B 및 PaLM-540B와 경쟁

- **공개 데이터만 활용**:
  - 소유권 있는 데이터 없이도 최첨단 성능을 달성 가능

- **모델 공개의 의미**
  - 연구 커뮤니티에 모델을 공개하여, 대규모 언어 모델의 개발을 촉진
  - 독성, 편향 문제를 해결하고 모델의 robustness 강화

- **미래 계획**
  - Instruction Fine-tuning이 유망한 결과를 보였으며, 이를 통해 성능을 더욱 개선할 계획
  - 더 큰 규모의 모델 확장을 통해 추가 연구를 진행할 예정