

# 한국어를 위한 지시어 튜닝 조사 연구: 거대 언어 모델 학습 및 적용

고려대학교 ■ 이승준·이정우·장윤나·이태민·임희석\*

## 1. 서론

언어 모델링 분야는 최근 몇 년 동안 눈에 띄는 발전을 이루어냈고, 이러한 발전은 자연어 처리 (natural language processing, NLP)의 광범위한 영역에 걸쳐 나타났다. 특히, 거대 언어 모델 (large language models, LLM)의 출현은 연구의 방향점을 크게 바꾸었다. ChatGPT [1]와 같은 초기 모델들이 성공을 거두면서, OpenAI의 GPT-4 [2] 및 Meta AI의 LLaMA [3]와 같은 최근의 연구들이 이 분야의 발전을 이끌고 있다. 이러한 LLM들은 수십억에서 수조 개의 학습 파라미터를 보유하고 있어 다양한 NLP 작업에서 높은 성능을 보여주고 있다. 일부 모델들은 특정 도메인이나 작업에 대한 세밀한 조정 (fine-tuning) 없이도 뛰어난 성능을 발휘하는 것으로 알려져 있다 [4, 5, 6].

LLM의 성공의 핵심은 지시어 튜닝 (instruction-tuning)에 근간을 두고 있으며, ChatGPT와 GPT-4의 뛰어난 언어 생성 성능은 이 방법을 활용하여 오픈 소스 LLM을 개선할 수 있는 기회를 제공했다. LLaMA는 GPT-3 [7] 같은 독점적인 LLM의 성능과 맞먹는 오픈 소스 LLM으로서, 지시어 튜닝의 효과적인 성능과 낮은 비용으로 인해 널리 사용되고 있다. 예를 들어, 스탠포드 대학교에서 개발한 Alpaca [8]는 GPT-3.5에서 생성된 52만 개의 지시어 데이터셋 (instruction-following dataset)을 활용했으며, Vicuna [9]는 ChatGPT와 사용자 사이에 공유된 7만 개의 대화 세트로 구성된 약 700만 개의 지시어 데이터셋을 활용했다.

최근 지시어 튜닝 연구의 주요 초점은 영어와 중국어에 맞춰져 있어, 한국어에 대한 지시어 튜닝 연구는 상대적으로 부족한 상태이다. 한국어 텍스트 데이터셋의 부족은 대규모 한국어 언어 모델의 학습을 어렵

게 만들며, 이 때문에 한국어 지시어 튜닝 연구는 영어와 중국어와 같은 자원이 풍부한 언어에 비해 뒤쳐진 실정이다. 더욱이, 기업과 공공 기관에서는 문서 유출과 같은 보안 이슈로 인해 자체 LLM 모델을 사용해야 하지만, 이를 구현하기 위한 필요한 자원과 연구는 충분하지 않다.

본 논문에서는 지시어를 따르는 모델을 학습하기 위해 필요한 데이터셋, 학습 전략, 그리고 LLM을 소개한다. 이를 바탕으로 한국어 지시어를 따르는 모델을 학습할 수 있는 전략과 방법론을 소개한다. 더 나아가, 소개한 방법론을 활용해 실제 한국어 지시어를 따르는 모델을 학습하고, 이를 기존의 모델과 비교하여 실험을 진행한다. 이 연구는 한국어 지시어 튜닝 분야의 연구를 촉진하고, 향후 한국어 지시어 튜닝 모델의 개발과 적용에 있어 기반을 제공하고자 한다.

## 2. Large Language Models

대규모의 언어 코퍼스에 거대한 파라미터 수를 갖는 모델을 학습시킨 LLM의 발전과 공개는 다양한 자연어 처리 분야에서 기계가 인간에 준하거나 뛰어넘는 성능을 보여주었다. 본 장에서는 이러한 LLM을 소개한다.

### 2.1 GPT-3

GPT-3 [7]는 OpenAI에서 발표한 자동 회귀적 (auto-regressive) 생성 구조를 가지는 175B 파라미터의 언어 모델이다. 기존 사전 학습 언어 모델이 특정 태스크를 수행하기 위해서 미세조정을 필요로 했던 반면, GPT-3에서는 모델 입력에 few-shot 예시를 주는 in-context learning을 통해 다양한 NLP 태스크에서 미세조정 없이도 좋은 성능을 낼 수 있도록 했다. 이는 이후 나오게 되는 많은 LLM들의 근간이 되었으며, OpenAI에서는 ChatGPT와 GPT-4 [2] 등 후속 모델들

\* 중신회원

을 발표하고 있다.

## 2.2 BLOOM

BLOOM [10]은 decoder-only 구조를 가지는 공개 언어 모델이다. BLOOM은 ROOTS 코퍼스에 학습 되었는데 ROOTS는 46개의 자연 언어와 13개의 프로그래밍 언어를 포함한 수백여개의 데이터 집합으로 이루어져 있다. 이러한 데이터 집합을 품질 기반으로 필터링하는 작업을 통해 데이터 품질을 높였고 데이터에 포함된 개인정보에 대해서는 비식별화 하는 작업을 수행했다. 모델은 기본적으로 Transformer [11] 구조를 따르지만, ALiBi 포지션 임베딩 [12]과 임베딩 직후 layer normalization [13]을 수행하여 학습 안정화와 성능 향상을 도모했다. 토큰라이저는 250k의 vocab 크기를 가지며 byte-level BPE를 활용한다. BLOOM은 최대 약 176B의 파라미터를 가지는 모델을 비롯하여 560M, 1.1B, 1.7B, 3B, 7.1B의 파라미터를 가지는 하위 모델을 공개했다.

## 2.3 Polyglot-Ko

Polyglot은 다국어 언어 모델의 영어 외 언어의 성능을 높이기 위해 EleutherAI로부터 제안된 프로젝트로, 그 중 Polyglot-Ko는 한국어 성능을 위해 공개된 모델이다. 블로그, 뉴스, 특허, 위키피디아 등 다양한 분야에서 1.2TB의 대용량 한국어 데이터를 TUNiB<sup>1)</sup> 모았으며, 전처리를 통해 863GB의 텍스트를 모델 학습을 위해 사용하였다. 모델은 EleutherAI의 GPT-NeoX [14]를 기초 모델로 두고 학습에 활용한다. 토큰라이저는 30,003개의 vocab 크기를 가지며 형태소 기반의 byte level BPE (Byte-Pair Encoding)로 학습되었다. 형태소 분석은 한국어 형태소 분석에서 널리 사용되는 MeCab<sup>2)</sup>을 활용하였다. 모델은 1.3B, 3.8B, 5.8B, 12.8B의 네 가지 크기를 가진 버전이 공개되었다.

## 2.4 LLaMA

LLaMA는 Meta AI<sup>3)</sup>에서 LLM의 보다 높은 성능을 위해 더 많은 토큰에 대해 학습을 시킨 모델로, 7B에서 65B에 달하는 크기를 가진 공개된 모델이다. 학습 데이터는 English CommonCrawl, C4, Github 등 여러 종류의 공개된 데이터에 대해 필터링을 통해 저품질 데이터를 걸러낸 후 사용했다. 토큰라이저는 SentencePiece [15]를 사용했으며 숫자는 모두 한 자리씩 분리시켰다.

모델 구조는 Transformer를 기반으로 하며 RMSNorm 정규화 [16], SwiGLU activation function [17], rotary positional embedding (RoPE) [18] 등 이전 모델들이 제안한 변형을 채택하여 성능 향상을 도모했다. LLaMA2 [19]는 LLaMA를 기반으로 grouped query attention (GQA) [20]을 모델 구조에 추가하고, 대화에 최적화 되도록 학습시켜 7B 부터 70B 파라미터 수를 가지는 모델을 배포했다.

## 2.5 RedPajama

현재 상업용으로 쓰이며 공개되지 않은 모델들이 많은 가운데, 기술의 공유와 발전을 위해 Together AI<sup>4)</sup>에서는 RedPajama의 파운데이션 모델과 학습에 사용된 데이터를 완전히 공개하였다 [21]. 먼저 데이터는 LLaMA에서의 데이터 종류와 정제 방법을 따르는 1.2T의 토큰의 데이터이다. 파운데이션 모델로 EleutherAI의 Pythia-7B를 사용하여 학습을 시켰다.

# 3. 지시어 튜닝

LLM 발전의 핵심 요소 중 하나는 지시어 튜닝(instruction tuning)이다. 지시어 튜닝은 LLM의 제로샷(zero-shot)<sup>5)</sup> 학습 능력을 향상하는 전략으로, 특정 지시어를 통해 모델을 파인 튜닝(fine-tuning)하는 과정을 포함한다 [4, 22]. 일반적으로 지시어 튜닝은 모델이 지시어를 따를 수(instruction-following) 있도록 ‘지시어(instruction)’와 ‘출력(output)’을 사용하여 LLM을 추가로 학습시키는 것을 의미한다.

## 3.1 데이터 구조

지시어 데이터셋의 각 데이터는 세 가지 요소로 구성된다. 태스크를 정의하는 자연어로 구성된 지시어(예: XX에게 XX에 대한 감사 편지 쓰기, XX 주제에 대한 블로그 포스팅 작성 등), 문맥에 대한 보충 정보를 제공하는 선택적 “입력(input)”, 지시어와 입력에 따라 예상되는 “출력(output)”으로 구성된다.

일반적으로 지시어 튜닝은 두 가지 방법으로 구축된다 [22].

- 어노테이션 데이터셋을 활용한 데이터 구축: 이 방법은 텍스트-레이블 쌍을(지시어, 출력) 쌍으로 변환 하기 위해 템플릿을 사용하여 기존에 어노테이션된 자연어 데이터셋에 대해(지시어, 출

1) <https://tunib.ai/>

2) <https://bitbucket.org/eunjeon/mecab-ko-dic/src/master/>

3) <https://ai.meta.com/>

4) <https://together.ai/>

5) 제로 샷 능력은 모델이 학습 때 보지 못한 태스크 혹은 데이터에 대한 이해 및 처리 능력을 의미함.

력) 쌍을 수집한다. FLAN [5] 및 P3 [23]와 같은 데이터셋이 기존의 어노테이션된 데이터셋을 활용하여 구축되었다.

- LLM을 활용한 출력 생성: 주어진 지시어에 대한 출력을 GPT-3.5-Turbo [1] 또는 GPT-4 [2]와 같은 LLMs을 사용하는 것을 의미한다. 이때, 지시어는 두 가지 방법으로 수집된다. 첫째, 수작업으로 제작한다. 둘째, 소량의 초기 시드 (seed) 지시어를 직접 생성한 후에 LLM을 통해 지시어를 확장한다. 이렇게 확장된 지시어는 다시 LLM의 출력을 사용하기 위한 입력으로 사용된다. InstructWild [24] 및 Self-Instruct [25]와 같은 데이터셋이 이러한 접근 방식을 사용한다.

멀티턴 (multi-turn) 대화형 지시어 데이터셋의 경우, LLM이 서로 다른 역할 (사용자 및 AI 어시스턴트)을 스스로 수행하여 대화 형식으로 메시지를 생성하도록 할 수 있으나 [26] 본 논문에서는 해당 지시어 데이터셋 구조 내용을 다루지 않는다.

### 3.2 지시어 튜닝 방법

이렇게 구축된 지시어 데이터셋은 사전 학습된 언어 모델에 대해 supervised fine-tuning (SFT) 방식으로 학습되며, 지시어와 입력이 주어지면 출력의 각 토큰을 순차적으로 예측하는 방식으로 학습된다. 해당 학습 방법에 대해서는 다음 7.2절에서 자세히 설명한다.

## 4. 지시어 데이터셋

### 4.1 Natural Instruction

Natural Instructions [27]는 61개의 NLP 태스크에서 나온 193,000개의 데이터를 인간 주석자가 만든 영어 기반의 지시어 데이터셋이다. 이 데이터셋은 ‘지시어’와 ‘인스턴스 (instance)’로 구성된다. 지시어는 제목, 정의, 피해야 할 사항 등 7가지 구성 요소로 구성된 태스크 설명이 포함되어 있다. 인스턴스에는 ‘입력 (input)’과 ‘출력 (output)’으로 구성된다. 입력에는 각 데이터의 입력 문장이, 출력에는 레이블링된 값이 사용된다.

### 4.2 P3

P3 (Public Pool of Prompts) [23]는 170개의 영어 NLP 데이터셋과 2,052개의 영어 프롬프트를 통합하여 구축된 명령어 미세 조정 데이터셋이다. 프롬프트 (prompt)는 때로는 태스크 템플릿이라고도 하며, 전통적인 NLP 작업 (예: 질문 응답, 텍스트 분류)의 데이

터 인스턴스를 자연어 입력-출력 쌍에 매핑하는 역할을 담당한다. P3의 각 데이터는 ‘입력’, ‘답변 선택지 (answer choices)’, ‘타겟 (targets)’의 세 가지 요소로 구성된다. ‘입력’은 각 태스크를 자연어로 설명하는 문장이며, ‘답변 선택지’는 주어진 태스크에 대해 답변 가능한 문장 목록이다. ‘타겟’은 주어진 ‘입력’에 대한 올바른 응답을 나타낸다. 저자들은 PromptSource 라는 고품질 프롬프트를 생성하고 오픈 소스로 고품질 프롬프트를 보관하는 도구를 개발하여 P3 데이터셋을 구축했다.

### 4.3 FLAN

FLAN [5]은 62개의 널리 사용되는 NLP 벤치마크 (예: SST-2 [28], SNLI [20], AG News [30], MultiRC [31])를 입력-출력 쌍으로 변환하여 구축된 영어 명령어 데이터셋입니다. FLAN의 각 데이터는 ‘입력’과 ‘타겟’ 구성 요소가 있다. ‘입력’은 자연어 명령어를 통해 작업을 설명하는 텍스트 시퀀스이며 ‘타겟’은 ‘입력’ 지시어를 올바르게 수행하는 텍스트 결과이다. 저자들은 일반 NLP 데이터셋을 입력-대상 쌍으로 변환하기 위해 명령어와 대상 템플릿을 수동으로 작성한 후에, NLP 벤치마크 데이터셋의 데이터를 템플릿에 매핑하였다.

### 4.4 Self-Instruct

Self-Instruct [25]는 InstructGPT [32]를 사용하여 구축된 5만개 지시어와 252개의 평가 지시어를 포함하는 영어 지시어 데이터셋이다. 데이터셋 구축을 위해 무작위로 선택된 자연어 지시어를 바탕으로 InstructGPT를 사용하여 더 많은 작업 지시어를 생성하고, 지시어에 따라 ‘입력’과 ‘출력’을 생성하는 과정, 그리고 유사한 명령어를 걸러내고 입력과 출력 데이터의 중복을 제거하는 후처리 과정을 통해 데이터셋을 구축했다.

### 4.5 Alpaca

Alpaca (7B)은 InstructGPT (175B, text-davinci003)에 의해 생성된 구축된 지시어 데이터셋에 LLaMA [3] (7B)을 미세 조정하여 학습된 언어 모델이다 [8]. Alpaca의 데이터셋은 Self-Instruct 데이터셋의 일부를 시드 데이터로 사용하여 GPT-3.5-turbo [1]를 사용하여 답변을 생성했다. Alpaca (7B)는 휴먼 평가에서 InstructGPT (175B, text-davinci-003)와 견주는 성능을 기록했다.

### 4.6 Vicuna

Vicuna (13B)는 ChatGPT에 의해 생성된 대화 데이

터셋에서 LLaMA (13B)를 미세 조정하여 훈련된 언어 모델이다. 연구자들은 ShareGPT<sup>6)</sup> 홈페이지에서 사용자가 공유한 ChatGPT 대화를 수집하고, 저품질의 샘플을 제외한 후 70K의 대화 기록을 학습 데이터셋으로 사용했다. LLaMA (13B)는 다중 턴 대화에 맞춤형된 수정된 손실 함수를 사용하여 구축된 대화 데이터셋에서 미세 조정되었다.

#### 4.7 GPT-4-LLM

GPT-4-LLM [33]은 Alpaca의 지시어를 GPT-4 [2]를 통해 출력을 수집했다. 이 데이터셋은 Alpaca와 동일하게 52만개의 지시어와 출력 쌍으로 구성된다.

#### 4.8 Dolly

Dolly [34]는 Databricks에서 제작한 지시어 데이터셋이다. InstructGPT에서 설명된 행동 카테고리 중 여러 가지, 예를 들면 브레인스토밍, 분류, 폐쇄형 QA, 생성, 정보 추출, 개방형 QA, 요약 등을 15만 개의 데이터를 포함한다.

#### 4.9 ShareGPT

ShareGPT는 ChatGPT와 사용자간의 대화를 수집한 데이터셋으로, 90만개의 데이터로 구성된다. 사용자의 입력을 지시어, ChatGPT의 생성 결과를 출력으로 사용했다. 일부 번역이 올바르게 되지 않은 영어 문장을 langid<sup>7)</sup>를 사용해 제거 했으며, 10자 미만의 지시어와 출력 데이터를 필터링 했다.

## 5. 한국어 오픈소스 LLM

#### 5.1 K(G)OAT

K(G)OAT<sup>8)</sup>는 LoRA 방식보다 더 효율적인 방법으로 소개된 IA3 [35] 방법을 따라 KoAlpaca를 미세 조정하였으며, 더 적은 파라미터로 더 효율적인 학습 방식을 도입하였다. A5000 GPU 2장으로 4 epoch 동안 학습되었으며, 총 226분 소요되었다. 학습에 사용된 데이터셋은 KoAlpaca v1.1을 활용해 프롬프트 구성을 수정하여 사용하였다.

#### 5.2 KoAlpaca

KoAlpaca는 LLaMA와 Polyglot-ko 모델을 기반으로 하여, 한국어 지시어 데이터셋으로 학습시킨 모델이다. KoAlpaca의 데이터셋은 Alpaca [8] 데이터셋의 지

시어를 DeepL을 사용해 번역한 후, 이를 GPT-3.5-turbo [1]를 사용해 출력을 생성했다.

#### 5.3 KoVicuna

KoVicuna는 LLaMA 모델을 기반으로 하며, Vicuna [9]의 접근 방식을 따른다. 이 모델은 ChatGPT와 사용자 간의 대화 데이터인 ShareGPT를 한국어로 번역한 버전을 활용하여 학습하였다.

#### 5.4 KULLM

KULLM<sup>9)</sup>은 고려대학교 NLP&AI 연구실과 HIAI 연구소가 개발한 한국어 LLM이다. 모델의 학습을 위해 EleutherAI에서 공개한 Polyglot-Ko 모델을 기반으로 LoRA [36]를 사용하여 학습하였다. A100 GPU (80GB VRAM) 4장을 사용하여 5 epoch 동안 학습을 진행했으며, LoRA를 이용한 파라미터 효율적인 학습은 PEFT 라이브러리를 사용하였다. GPT-4-LLM [33], Dolly [34], ShareGPT를 포함한 다양한 지시어 데이터셋을 DeepL을 사용해 한국어로 번역을 진행하여 150만 개의 한국어 지시어 데이터셋을 구축하였고 이를 통해 학습을 진행하였다.

#### 5.5 KORani

KORani<sup>10)</sup>는 LLaMA와 Polyglot-Ko 모델을 기반으로 하여 ShareGPT의 영어 버전과 한국어로 번역한 버전을 사용하여 미세 조정되었다. ‘human’과 ‘gpt’에 해당하는 대화의 첫 번째 부분과 두 번째 부분만 추출하여 학습하였으며, 모델의 미세 조정은 8개의 A100 40GB GPU에서 수행되었다.

#### 5.6 LLaMA-2-Ko

LLaMA-2-Ko<sup>11)</sup>는 LLaMA2 [19]를 기반으로 최적화된 Transformer [11] 구조 기반의 자동 회귀 언어 모델이며, 확장된 vocab과 한국어 말뭉치를 추가 사전 학습에 포함시켰다. 기존 32,000 크기의 LLaMA2 vocab에 한국어 vocab을 추가하여 46,336 크기의 vocab으로 확장하였으며, 이를 통해 모델의 한국어 성능을 향상시키고자 하였다.

#### 5.7 komt

komt<sup>12)</sup>는 LLaMA2 모델을 기반으로 하며, 다양한 작업의 지도 데이터셋을 활용하여 LLM의 학습 데이

6) <https://sharegpt.com/>

7) <https://github.com/saffsd/langid.py>

8) <https://github.com/Marker-Inc-Korea/K-G-OAT>

9) <https://github.com/nlpai-lab/KULLM>

10) <https://github.com/krafton-ai/KORani>

11) <https://huggingface.co/beomi/llama-2-ko-7b>

12) <https://github.com/davidkim205/komt>

터를 생성하는 다중 작업 지시 기법을 도입하였다. komt은 번역된 데이터셋에 의존하지 않고 KorQuad [37], AIHub, KISTI AI 데이터셋 등 기존의 다양한 한국어 데이터셋을 기반으로 각 작업에 맞는 다중 작업 데이터셋을 구축하여 이를 통해 학습을 진행하였다.

### 5.8 Ko-Platypus

Ko-Platypus<sup>13)</sup>는 KOpen-Platypus 데이터셋<sup>14)</sup>을 활용하여 LLaMA-2-Ko를 미세 조정한 모델이다. KOpen-Platypus 데이터셋은 Open-Platypus 데이터셋<sup>15)</sup>을 DeepL을 사용해 번역을 진행한 후, 약 25,000개의 데이터를 수작업으로 체크하였고 약 144시간 정도 소요하여 번역 오류를 대부분 고친 데이터셋이며, 이를 통해 학습을 진행하였다.

## 6. Parameter Efficient Tuning

큰 규모의 모델 파라미터를 갖는 LLM의 효율적인 학습을 위한 다양한 방법론이 소개되었다. 대표적으로 low-rank adaptation (LoRA), adapter와 같이 모델의 일부분을 학습시키는 전략과 모델 구조를 보다 효율적으로 바꾼 flash attention을 본 장에서 간단하게 소개한다.

### 6.1 LoRA

Low-rank adaptation (LoRA) [36]은 사전 학습된 모델 가중치를 freeze하고 학습 가능한 rank decomposition 행렬을 Transformer [11] 구조의 각 레이어에 주입하여 다운스트림 작업에 대해 학습 가능한 매개변수의 수를 크게 줄이는 방법론이다. 이를 통해 GPT-3 [7] 175B의 경우, LoRA는 전체 미세 조정에 비해 학습 가능한 파라미터 수를 10,000배, GPU 메모리 사용량을 3배까지 줄일 수 있다. LoRA는 학습 가능한 파라미터 수가 더 적고, 어댑터와 달리 추가적인 추론 지연 시간이 없음에도 불구하고 RoBERTa [38], DeBERTa [39], GPT-2[40] 및 GPT-3에서 모델 품질이 파인 튜닝과 동등하거나 더 나은 성능을 발휘한다.

### 6.2 LLaMA-Adapter

LLaMA-Adapter [41]는 LLaMA를 지시어를 따르는 모델로 만드는 효율적인 방법론이다. 약 52K 개의 지시어를 활용하여 LLaMA-Adapter는 7B 파라미터 수를 갖는 LLaMA의 1.2M 개만의 학습 파라미터만으로

학습이 가능하게 한다. Transformer 레이어 상단에서 학습 가능한 adaptation 프롬프트를 단어 토큰 앞에 붙인다. 또한 zero gating과 함께 zero-initialized attention을 제안하며 사전 학습 지식을 보존하면서 효과적으로 새로운 지시어를 학습할 수 있도록 한다. 이를 통해 LLaMA-Adapter는 fully fine-tuned Alpaca보다 품질이 좋은 답변을 생성한다.

### 6.3 Flash Attention

Transformer 기반의 모델이 널리 사용되기 시작하며 모델의 크기가 더 커지고 레이어 수가 더 많아짐에 따라 시간 및 메모리 복잡도가 증가하게 되었다. Flash attention [42]은 이에 보다 효율적인 모델을 만들고자 한 대표적인 연구이다. Flash attention에서는 읽기 및 쓰기 (IO)에 대한 정보를 GPU 메모리 단계에서 보다 효율적으로 고려할 수 있게끔 하여 기존 Transformer의 attention에 비해 GPT-2 모델 기준 약 3배 빠른 속도로 처리가 가능하게 한다.

## 7. 실험 환경

### 7.1 학습 설정

본 연구에서는 한국어 지시어를 따르는 모델의 학습을 위해 EleutherAI에서 공개한 Polyglot-Ko의 5.8b과 12.8b 한국어 LLM 모델을 사용했다. Polyglot-Ko는 863GB의 한국어 데이터로 학습되었으며 블로그, 모두의 말뭉치, 위키 등 다양한 도메인의 텍스트가 포함되어 있다 [43]. 또한, 데이터셋으로는 4절에서 언급한 GPT-4-LLM, Dolly, ShareGPT 데이터셋을 한국어로 번역해 사용했다. 이 때, 사용한 번역기는 DeepL 번역기를 사용했다. SFT 방식으로 학습을 진행했으며, 파라미터 효율적인 (parameter-efficient) 학습을 위해 LoRA (low-rank adaptation) [36]를 query key value 어텐션 레이어에 적용 하였다.

모든 모델은 A100 GPU (80GB VRAM) 4장을 사용하여 5 epoch 동안 학습을 진행했으며, LoRA를 이용한 파라미터 효율적인 학습은 PEFT<sup>16)</sup> 라이브러리를 사용하였다. 또한 학습 과정에서 메모리 효율성을 최적화하기 위해 bitsandbytes<sup>17)</sup>를 활용했다. AdamW optimizer [44]를 사용하여 learning rate의 최대치를 3e-4로 설정하고 5%의 warm-up 코사인 스케줄러를 적용했다. 또한 gradient explosion을 완화하기 위해

13) <https://github.com/Marker-Inc-Korea/KO-Platypus>

14) <https://huggingface.co/datasets/kyujinpy/KOpen-platypus>

15) <https://huggingface.co/datasets/garage-bAInd/Open-Platypus>

16) <https://github.com/huggingface/peft>

17) <https://github.com/TimDettmers/bitsandbytes>

gradient clipping를 적용했다. 배치 사이즈는 {128, 256, 512} 중에서 가장 성능이 좋은 256을 사용 했으며, LoRA Rank와 Alpha 는 각각 16, 32로 설정했다. 디코딩 설정은 temperature=0.1, top k=40, top p=0.75, do sample=False, num beams=4, repetition penalty=1.4, max new tokens=400로 설정 했다.

## 7.2 학습 전략

사전 학습된 언어 모델은 사용자의 지시어를 잘 따르지 못하고 종종 의도하지 않은 내용을 생성한다. 이는 “지시어를 따르고 질문에 답하라”라는 지시어 튜닝의 목적 함수와는 다르게 기존 사전 학습 모델의 목적 함수가 다음 토큰 예측 (next-token prediction)이었기 때문이다 [45]. 따라서 사용자의 의도와 언어 모델의 동작을 일치시키기 위해, 모델을 명시적으로 지시어를 따르도록 파인튜닝을 할 수 있다.

SFT는 커질 언어 모델링 (casual language modeling)의 방법으로 학습되는데, 모델은 지시어로 프롬프트를 받고 자동 회귀적 (auto-regressive)으로 출력을 한 토큰씩 생성하는 방식으로 학습된다. 지시어는 프롬프트 템플릿에 포함되며, 출력은 템플릿 바로 뒤에 이어진다. 예시로 사용된 프롬프트는 Alpaca 에서 사용한 템플릿을 채택하였으며, 입력 순서는 다음과 같다.

아래는 작업을 설명하는 명령어와 추가 컨텍스트를 제공하는 입력이 짝을 이루는 예제입니다.

요청을 적절히 완료하는 응답을 작성하세요.

지시어: {지시어 입력}

입력: {입력}

응답: {출력}

## 7.3 평가 방법

본 절에서는 본 논문에서 제안한 모델의 한국어 instruction- following 능력의 평가 방법을 기술한다. LLM의 성능 평가는 지시어의 다양성과 모호성, 그리고 모델의 광범위한 능력을 정확하게 측정하기 위한 기준의 부족으로 인해 복잡하고 어려운 문제로 여겨진다 [46]. 이러한 문제를 해결하기 위해 LLM을 활용하여 LLM를 평가하는 여러 연구가 수행되었으며, 이 방법론을 사용하면 사람의 어노테이션 없이도 여러 평가 벤치마크에서 높은 상관관계를 보인다 [47].

이에 따라, 본 연구에서는 자연어로 기술된 프롬프트 (prompt)를 중심으로 한 평가 방식을 도입하였으며, 이를 통해 KULLM 모델 및 다른 한국어 LLM의 능력을 평가하였다. 우리는 단일 모델의 출력 품질을

다각적으로 이해하기 위한 통합 평가 체계 (unified evaluation schema)를 적용하는 프롬프트를 설계하였다. 이 체계는 LLM-Eval에서 제시된 방식을 기반으로 하며, 내용, 문법, 관련성 및 적절성 등의 평가 요소를 포괄하는 자연어 지시어로 구성되어 있다. 본 연구에서는 이해도, 자연스러움, 맥락 유지, 흥미도, 지시어의 정확한 반영 및 전체적인 출력 품질 등을 평가 척도로 설정하였다. 자세한 프롬프트는 다음과 같다.

두 사람 간의 대화가 주어집니다. 다음의 지시어 (Instruction), 입력(Input)을 받게 될 것입니다. 그리고 지시어와 입력에 대한 응답(Response)이 제시됩니다. 당신의 작업은 응답을 평가 단계에 따라 응답을 평가하는 것입니다.

이 평가 기준을 꼼꼼히 읽고 이해하는 것이 중요합니다. 평가하는 동안 이 문서를 계속 열어두고 필요할 때 참조해 주세요.

평가 기준:

- 이해 가능성 (0 - 1): 입력 (Input)에 기반하여 응답(Response)을 이해 할 수 있나요?
- 자연스러움 (1 - 3): 사람이 자연스럽게 말할 법한 지시어(Instruction) 인가요?
- 맥락 유지 (1 - 3): 입력 (Input)을 고려했을 때 응답(Response)가 맥락을 유지하나요?
- 흥미롭기 (1 - 3): 응답 (Response)가 지루한가요, 아니면 흥미로운가요?
- 지시어 사용 (0 - 1): 지시어 (Instruction)에 기반하여 응답 (Response)를 생성 했나요?
- 전반적인 품질 (1 - 5): 위의 답변을 바탕으로 이 발언의 전반적인 품질에 대한 인상은 어떤가요?

평가 단계:

1. Instruction, Input, 그리고 Response를 주의 깊게 읽습니다.
  2. 위의 평가 기준에 따라 Response를 평가합니다.
- Instruction: {instruction}  
Input: {input}  
Response: {response}

본 연구에서는 Large Language Models (LLM)의 성능 평가를 위해 GPT-4 [2]를 사용하였다. 평가에 사용된 데이터셋은 Self-Instruct [25]의 <sup>18)</sup>Human Eval를 한국어로 번역한 것이다. 이 데이터셋은 일반적인 NLP 태스크를 넘어, 실제 응용 분야에서 사용자가 생성한

18) [https://github.com/yizhongw/self-instruct/tree/main/human\\_eval](https://github.com/yizhongw/self-instruct/tree/main/human_eval)

표 1 한국어 사용이 가능한 LLM 모델과의 지시어를 따르는 (instruction-following) 성능 비교 실험.

	기본 모델	이해 가능성	자연스러움	맥락 유지	맥락 유지	맥락 유지	전반적인 품질
GPT-3.5-turbo	GPT-3.5	0.980	2,806	2,849	2,056	0.917	3,905
GPT-4	GPT-4	0.984	2,897	2,944	2,143	0.968	4,083
Ours5.8b	Polyglot-Ko	0.754	2,099	2,187	1,746	0.579	3,028
Ours12.8b	Polyglot-Ko	0.802	2,480	2,310	1,996	0.754	3,325

지시어를 포함하여 instruction-following 모델의 실질적인 성능을 평가하기 위해 구축되었다. 예를 들어, 사용자가 “R (프로그래밍 언어)에 관한 기술 평가 질문지 작성”과 같은 지시어를 제공하면, 해당 지시어에 따라 모델은 적절한 출력을 생성한다. 최종적으로, 평가 프롬프트의 Instruction과 Input 필드에는 번역된 Human Eval 데이터가, Response 필드에는 모델이 생성한 출력을 사용했다.

## 8. 실험 결과

본 절에서는 본 논문에서 제한 지시어를 따르는 모델과 사용화된 모델인 ChatGPT와 GPT-4 모델들의 instruction-following 성능을 비교하는 실험 결과를 제시한다. 표 1은 본 방법론의 모델 및 상용화 LLM 모델들의 instruction-following 성능을 나타낸다. 이 실험에서, GPT-4 [2] 모델은 모든 평가 항목에서 가장 높은 성능을 보였다. 특히 전반적인 품질에서 GPT-4는 4.083의 점수로 다른 모델들에 비해 높은 성능을 보였다. Polyglot-Ko 12.8B로 학습한 모델이 가장 높은 성능을 나타냈다. 이 모델은 이해 가능성, 자연스러움, 맥락 유지, 흥미롭기, 지시어 사용, 그리고 전반적인 품질에서 6.8B 보다 높은 성능을 보였다. 우리의 방법론으로 학습된 모델은 ChatGPT와 GPT-4에 비해 성능이 낮았지만, 준수한 성능을 보인다.

## 9. 결 론

본 연구에서는 지시어를 따르는 모델을 구축하기 위한 지시어 데이터셋, 모델, 학습 전략 등을 살펴 보았다. 또한, 한국어 자연어 처리에 적합한 지시어 모델을 학습하기 위한 방법론과 성능을 비교한 실험을 진행했다. 기존에 한국어 자연어 처리 생태계에 지시어 데이터셋의 부족이라는 문제점을 인식하고, 이를 해결하기 위해 영어로 제작된 지시어 데이터셋을 한국어로 번역하는 방법을 적용하였다. 이 과정을 통해 한국어 지시어를 따르는 모델을 학습하기 위해 적합한 데이터셋을 구성했다. 또한, 지시어를 따르는 거대 언어 모델을 학습을 위해 SFT를 적용하였으며, 평가

결과 본 논문에서 시도한 방법론이 준수한 성능을 보인다. 하지만 GPT-4 [2]와 같은 상용화 모델과의 성능 차이는 여전히 존재하였다. 이러한 접근법은 다양한 언어와 문화에 걸쳐 지시어 튜닝 연구를 확장하고, 더욱 포괄적인 자연어 처리 솔루션을 제공하는 데 도움이 될 것이다. 향후 연구에서는 더 큰 규모의 한국어 지시어 데이터셋을 구축하고, 모델의 구조와 학습 전략을 최적화하여 한국어 지시어 튜닝의 성능을 더욱 향상시킬 예정이다. 본 연구는 한국어 지시어 튜닝의 초기 단계를 탐구하며, 이 분야의 미래 연구 방향과 가능성을 제시한다. 본 논문을 통해 자연어 처리 기술의 보다 넓은 적용과 발전을 기대한다.

## 참고문헌

- [1] OpenAI, “Gpt-4 technical report,” 2023.
- [2] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Lample, G. (2023). Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.
- [3] Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., ... & Le, Q. V. (2021). Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652.
- [4] Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., ... & Roberts, A. (2023). The flan collection: Designing data and methods for effective instruction tuning. arXiv preprint arXiv:2301.13688.
- [5] Mukherjee, S., Mitra, A., Jawahar, G., Agarwal, S., Palangi, H., & Awadallah, A. (2023). Orca: Progressive learning from complex explanation traces of gpt-4. arXiv preprint arXiv:2306.02707.
- [6] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.
- [7] Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., ... & Hashimoto, T. B. (2023). Stanford alpaca: An instruction-following llama model.
- [8] Zheng, L., Chiang, W. L., Sheng, Y., Zhuang, S., Wu, Z.,

- Zhuang, Y., ... & Stoica, I. (2023). Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. arXiv preprint arXiv:2306.05685.
- [9] Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., ... & Manica, M. (2022). Bloom: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100.
- [10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [11] Press, O., Smith, N. A., & Lewis, M. (2021). Train short, test long: Attention with linear biases enables input length extrapolation. arXiv preprint arXiv:2108.12409.
- [12] Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). Llm.int8(): 8-bit matrix multiplication for transformers at scale. arXiv preprint arXiv:2208.07339.
- [13] Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., ... & Weinbach, S. (2022). Gpt-neox-20b: An open-source autoregressive language model. arXiv preprint arXiv:2204.06745.
- [14] Kudo, T., & Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226.
- [15] Zhang, B., & Sennrich, R. (2019). Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32.
- [16] Shazeer, N. (2020). Glu variants improve transformer. arXiv preprint arXiv:2002.05202.
- [17] Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., & Liu, Y. (2021). Roformer: Enhanced transformer with rotary position embedding. arXiv preprint arXiv:2104.09864.
- [18] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
- [19] Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., & Sanghai, S. (2023). GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. arXiv preprint arXiv:2305.13245.
- [20] T. Computer, "Redpajama: An open source recipe to reproduce llama training dataset," 2023. [Online]. Available: <https://github.com/togethercomputer/RedPajama-Data>
- [21] Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., ... & Wang, G. (2023). Instruction tuning for large language models: A survey. arXiv preprint arXiv:2308.10792.
- [22] Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., ... & Rush, A. M. (2021). Multitask prompted training enables zero-shot task generalization. arXiv preprint arXiv:2110.08207.
- [23] [Online]. Available: <https://openai.com/chatgpt>
- [24] Xue, F., Zheng, Z., & You, Y. (2023). Instruction in the wild: A user-based instruction dataset.
- [25] Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., & Hajishirzi, H. (2022). Self-instruct: Aligning language model with self generated instructions. arXiv preprint arXiv:2212.10560.
- [26] Xu, C., Guo, D., Duan, N., & McAuley, J. (2023). Baize: An open-source chat model with parameter-efficient tuning on self-chat data. arXiv preprint arXiv:2304.01196.
- [27] Mishra, S., Khashabi, D., Baral, C., & Hajishirzi, H. (2021). Cross-task generalization via natural language crowdsourcing instructions. arXiv preprint arXiv:2104.08773.
- [28] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1631-1642).
- [29] Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326.
- [30] Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.
- [31] Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S., & Roth, D. (2018, June). Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 252-262).
- [32] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35,



27730-27744.

- [33] Peng, B., Li, C., He, P., Galley, M., & Gao, J. (2023). Instruction tuning with gpt-4. arXiv preprint arXiv:2304.03277.
- [34] Conover, M., Hayes, M., Mathur, A., Meng, X., Xie, J., Wan, J., ... & Xin, R. (2023). Free dolly: Introducing the world's first truly open instruction-tuned llm.
- [35] Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., & Raffel, C. A. (2022). Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. Advances in Neural Information Processing Systems, 35, 1950-1965.
- [36] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.
- [37] 임승영, 김명지, & 이주열. (2018). KorQuAD: 기계독해를 위한 한국어 질의응답 데이터셋. 한국정보과학회 학술발표논문집, 539-541.
- [38] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.
- [39] He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654.
- [40] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.
- [41] Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., ... & Qiao, Y. (2023). Llama-adapter: Efficient fine-tuning of language models with zero-init attention. arXiv preprint arXiv:2303.16199.
- [42] Dao, T., Fu, D., Ermon, S., Rudra, A., & Ré, C. (2022). Flashattention: Fast and memory-efficient exact attention with io-awareness. Advances in Neural Information Processing Systems, 35, 16344-16359.
- [43] Ko, H., Yang, K., Ryu, M., Choi, T., Yang, S., & Park, S. (2023). A Technical Report for Polyglot-Ko: Open-Source Large-Scale Korean Language Models. arXiv preprint arXiv:2306.02254.
- [44] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.
- [45] Cui, Y., Yang, Z., & Yao, X. (2023). Efficient and effective

text encoding for chinese llama and alpaca. arXiv preprint arXiv:2304.08177.

- [46] Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., & Zhu, C. (2023). GpTeval: Nlg evaluation using gpt-4 with better human alignment. arXiv preprint arXiv:2303.16634.
- [47] Lin, Y. T., & Chen, Y. N. (2023). LLM-Eval: Unified Multi-Dimensional Automatic Evaluation for Open-Domain Conversations with Large Language Models. arXiv preprint arXiv:2305.13711.

## 약 력



### 이 승 준

2021 한국외국어대학교 산업경영공학과 졸업(학사)  
2022~현재 고려대학교 컴퓨터학과 (석사)  
관심분야: 자연어처리, 기계번역  
Email : dzyzy6505@korea.ac.kr



### 이 정 우

2020 영남대학교 정보통신공학과 졸업(학사)  
2022~현재 고려대학교 컴퓨터학과 석사과정  
관심분야: 자연어처리, 대화 모델, 기계 독해  
Email : time79779@korea.ac.kr



### 장 윤 나

2020 고려대학교 영어영문학과 졸업(학사)  
2020~현재 고려대학교 컴퓨터학과 석박사통합과정  
관심분야: 자연어처리, 언어 생성, 대화 모델  
Email : morelychee@korea.ac.kr



### 이 태 민

2008 고려대학교 컴퓨터교육학과 졸업(학사)  
2020 고려대학교 컴퓨터학과 졸업(석사)  
2020~현재 고려대학교 Human-Inspired AI 연구소  
연구원  
관심분야: 자연어처리, 신경망  
Email : taeminlee@korea.ac.kr



### 임 희 석

1992 고려대학교 컴퓨터학과 졸업(학사)  
1994 고려대학교 컴퓨터학과 졸업(석사)  
1997 고려대학교 컴퓨터학과 졸업(박사)  
2008~현재 고려대학교 컴퓨터학과 교수  
관심분야: 자연어처리, 뇌신경 언어 정보 처리  
Email : limhseok@korea.ac.kr