거대 언어 모델(LLM)을 이용한 비훈련 이진 감정 분류

안형진, 황태욱, 정상근°
충남대학교 컴퓨터융합학부
hyungjin.ahn51@gmail.com, taewook5295@gmail.com, hugmanskj@gmail.com°

Utilizing Large Language Models for Non-trained Binary Sentiment Classification

Hyungjin Ahn, Taewook Hwang, Sangkeun Jung°
The Division of Computer Convergence, Chungnam National University

요 약

ChatGPT가 등장한 이후 다양한 거대 언어 모델(Large Language Model, LLM)이 등장하였고, 이러한 LLM을 목적에 맞게 파인튜닝하여 사용할 수 있게 되었다. 하지만 LLM을 새로 학습하는 것은 물론이고, 단순 튜닝만 하더라도 일반인은 시도하기 어려울 정도의 많은 컴퓨팅 자원이 필요하다. 본 연구에서는 공개된 LLM을 별도의 학습 없이 사용하여 zero-shot 프롬프팅으로 이진 분류 태스크에 대한 성능을 확인하고자 했다. 학습이나 추가적인 튜닝 없이도 기존 선학습 언어 모델들에 준하는 이진 분류 성능을 확인할 수 있었고, 성능이 좋은 LLM의 경우 분류 실패율이 낮고 일관적인 성능을 보여 상당히 높은 활용성을 확인하였다.

주제어: Large Language Model, Zero-Shot Prompting, Sentiment Analysis, GLUE SST-2, Voting

1. 서론

Transformers[1]를 활용한 선학습 언어 모델(Pre-trained Language Model, PLM)이 등장하면서 상당수의 자연어처리 분야에서 State-of-the-Art (SOTA)을 달성하였고, 이후 대부분의 자연어처리 태스크에 선학습 모델이 활용되었다. 선학습 모델은 웹상에서 상대적으로 구하기 쉬운 비지도 학습 데이터를 대규모로 수집하고, Masked Language Modeling 등의 준지도 학습(self-supervised learning)으로 학습된다. 이렇게 선학습된 모델을 타겟 도메인의 데이터를 활용하여 파인튜닝하면 대부분의 자연어처리 분야에서 기존 Recurrent Neural Network 계열 모델 대비 높은 성능을 보인다.

이러한 방법론이 등장하면서 선학습 데이터셋과 모델의 규모가 클수록 모델의 성능이 높아지는 경향이 밝혀졌으며, Google, Microsoft 등의 다양한 기업 및 기관에서 PLM의 규모를 지속적으로 확대하여 배포하였다. 하지만 기존 PLM 모델들은 높은 성능을 보임에도, 사람과 비교하였을 때 상당히 부족하다는 한계가 존재한다.

최근 OpenAI에서 ChatGPT를 공개하면서 기존 PLM의 성능을 압도하며, 사람에 준하는 성능을 보이는 거대 언어 모델(Large Language Model, LLM)의 가능성을 입증하였다. 이후 Bard[1], LLaMA[2], Alpaca[3] 등이 순차적으로 등장하였으며, 이러한 모델들을 기반으로 한국어 데이터셋으로 학습한 KoAlpaca, KoLLaMA, polyglot-ko[2] 등의 LLM도 등장하였다.

그러나 LLM은 사람에 준하는 성능을 보이기 위해서는 상당한 양의 데이터 및 리소스가 필요하다. GPT-3 모델의 경우 약 1,750억 개의 파라미터를 사용하였고, 모델을 선학습하기 위해 약 3,000억 토큰의 데이터셋으로 학습시켜서 수천 petaflop/s-days[3]의 연산이 수행되었다[4]. 위와 같은 이유로 일반인들은 LLM을 본인의 목적에 맞게 다시 학습시키거나 튜닝하기 힘들고, 연구기관 역시 LLM 튜닝에 많은 시간과 비용이 필요하다.

따라서 본 연구에서는 공개된 LLM을 활용하여 추가적인 학습이나 튜닝 없이 zero-shot 프롬프팅으로, 기존 자연어 처리 태스크를 바로 해결할 수 있는 능력을 갖추고 있는지 확인하고자 한다. Application Programming Interface (API)가 공개된 영어버전의 LLM 5개를 활용하였으며, 안정적인 성능을 위해 voting을 적용하였다. 일반적인 성능을 측정하기 위해 널리 알려진 GLUE[5] 벤치마크의 SST-2 이진 감정 분류 태스크 훈련 데이터셋 일부를 활용하였다.

2장의 관련 연구에서는 본 연구에서 사용한 모델들에 대해 소개한다. 또한 본 연구에서 프롬프트를 작성한 방식인 zero-shot 프롬프팅, 더 나은 추론 결과를 얻기 위해 시도한 voting 기법, 마지막으로 본 연구에서 사용한 데이터셋인 GLUE 벤치마크의 SST-2를 소개한다. 3장의 방법론에서는 본 연구의 실험 과정을 간략하게 소개하며 진행 과정을 그림으로 표현했으며, 4장의 실험에서 사용한 모델, 데이터셋, 프롬프트, 후처리 방법에 대해 자세히 설명하고 실험 결과를 표 형태로 나타냈다. 5장의 논의에서는 본 연구의 의미와 한계를 서술하고 향후 수행할 연구에 대해 작성했다.

[1] https://bard.google.com/
[2] https://huggingface.co/beomi

[3] 1 petaflop/s-days는 초당 1,000조 번의 연산을 수행할 수 있는 컴퓨터로 하루 동안 계산할 수 있는 연산량이다.

2. 관련 연구

2.1 GPT-3 / GPT-4

[4]에서는 자체적인 전처리를 거친 Common Crawl, WebText2, Books1, Books2, Wikipedia 데이터셋을 이용하여 GPT-3를 선학습 하였다. 기존에는 태스크에 맞는 동작을 수행하도록 하기 위해 태스크에 따른 파인튜닝을 해야 하고, 그를 위해 많은 양의 라벨링 된 학습데이터가 필요하다는 한계가 있었다. GPT-3 모델은 별도의 파인튜닝 없이, 태스크와 관련된 예제를 함께 입력하여 결과를 출력하는 few-shot 환경에서도 우수한 결과를 보였다.

[6]에서는 텍스트, 이미지 입력을 모두 처리할 수 있는 멀티 모달 모델인 GPT-4를 소개했다. GPT-4는 다양한 전문/학술적 벤치마크에서 인간 수준의 성능을 보였는데, 모의 사법시험을 상위 10%의 높은 성적을 보이며 통과했다. GPT-3.5가 동일한 시험에서 하위 10%의 성능을 기록한 것과 비교하면 GPT-4가 더 뛰어난 수준으로 자연어를 학습했다고 볼 수 있다. 다양한 언어로 번역된 버전의 MMLU 벤치마크[7]에서도 좋은 성능을 보였다.

2.2 LLaMA-1, LLaMA-2

[2]에서는 추론 비용을 고려하여 최적의 성능을 내는 모델을 연구하였고, 더 작은 모델을 더 많은 데이터로 훈련해 추론 비용을 낮춘 LLaMA 모델을 소개했다. 크기가 약 13배 더 작은 LLaMA-13B가 많은 벤치마크에서 GPT-3(175B) 모델을 능가하는 성능을 보였다. CommonCrawl, C4, Github, Wikipedia, Gutenberg, Books3, ArXiv, StackExchange로부터 공개된 데이터만을 사용하여 총 1.4조 개의 토큰으로 학습시켰다. 공개된 데이터만을 사용해도 SOTA 성능의 모델을 훈련할 수 있다는 것을 보였고, 공개된 데이터셋을 사용했기 때문에 LLaMA는 공개되어 누구나 사용할 수 있다.

[8]에서 소개된 LLaMA-2는 LLaMA-1[2] 모델이 개량된 버전으로, 선학습 데이터셋의 크기를 늘려서 2조 개의 토큰으로 학습시켰고, 모델의 context 길이를 이전보다 2배 늘렸으며, grouped-query attention[9]을 적용했다. 함께 소개된 LLaMA-2-Chat 모델은 대화에 맞는 형태로 파인튜닝한 모델이다. LLaMA-2 모델들은 [8]에서 테스트한 대부분의 벤치마크에서 다른 오픈 소스 LLM을 뛰어넘는 성능을 보였다.

2.3 Vicuna

[10]에서 등장한 Vicuna는 LLaMA모델을 ShareGPT를 통해 수집된 유저 대화 데이터를 이용하여 파인튜닝한 오픈 소스 모델이다. GPT-4를 평가용 모델로 사용한 평가[11] 결과 ChatGPT와 Google Bard의 품질을 90%이상 달성했고, 90%이상의 경우에 LLaMA와 Alpaca 모델 이상의 성능을 발휘했다.

2.4 Dolly 2.0

[12]는 파라미터 수가 7,000만 개에서 120억 개에 이르는 8개의 LLM을 공개된 데이터셋인 Pile 데이터셋[13]과 중복을 제거한 Pile 데이터셋에서 각각 훈련해서, 총 16개의 LLM을 소개했다. [12]는 gender bias, memorization, few-shot learning과 같은 특성들이 훈련 데이터와 모델 크기에 의해 어떤 영향을 받는지 연구하기 위해 pythia를 소개하고 모델 체크포인트를 공개했다.

[14]에서 소개하는 Dolly 2.0 모델은 pythia 모델을 기반으로, 공개된 데이터셋인 databricks-dolly-15k[4]를 사용하여 파인튜닝한 모델이다. 완전히 공개된 모델이기 때문에 상업적으로도 이용할 수 있다.

2.5 zero-shot / few-shot 프롬프팅

zero-shot 프롬프팅은 언어 모델이 태스크를 수행하도록 프롬프트를 작성할 때, 태스크에 관련된 예시를 입력하지 않고 질문만 입력하는 방식이다. 이와 다르게 few-shot 프롬프팅은 예측해야 할 입력과 함께 태스크와 관련된 예시(입력과 정답 쌍)를 함께 입력하여 LLM이 태스크와 관련된 추가적인 정보를 얻을 수 있게 질문하는 방식이다. [4]에서는 context 안에 주어지는 예시의 수가 많아질수록 성능이 높아지는 결과가 나타났다.

2.6 Voting

Voting은 여러 모델의 예측값을 평균 내거나 가장 많이 선택된 클래스를 최종 결과로 산출하는 기법이다. Hard voting은 각 모델이 예측한 결과 중 가장 많이 선택된 클래스를 최종 결과로 선택하는 방법이고, soft voting 방법은 모델별로 클래스에 대한 예측 확률을 모아 평균을 계산했을 때 가장 확률이 높은 클래스를 최종 결과로 선택하는 방법이다.

2.7 GLUE 벤치마크, SST-2

[5]에서 소개된 GLUE (General Language Understanding Evaluation) 벤치마크는 다양한 도메인에서 다양한 태스크를 수행할 수 있는 보다 일반화된 자연어 이해 성능을 평가하기 위한 벤치마크이다. Question answering, sentiment analysis, textual entailment를 포함한 태스크들이 포함되어 있다. 이 중 SST-2 (The Stanford Sentiment Treebank[15]) 데이터셋은 감정 분석 (sentiment analysis) 태스크에 관련된 데이터셋으로, 영화 리뷰를 문장 단위로 긍정/부정 두 개의 라벨을 붙여 구축한 데이터셋이다.

[4]https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

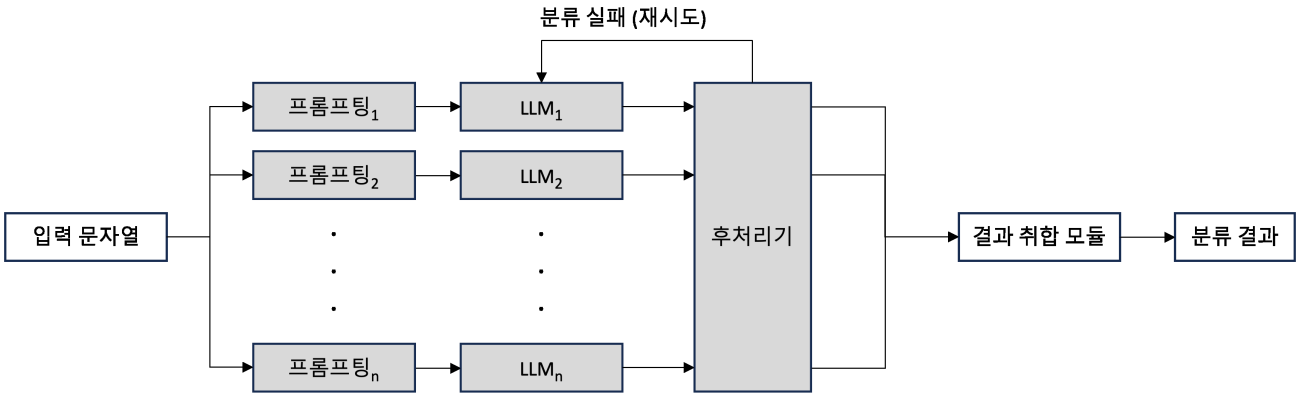

그림 1. LLM을 사용하여 분류 결과를 얻기 위해 사용한 구조

3. 실험

3.1 실험 방법

본 연구에서는 그림 1 형태의 구조를 통해 분류 결과를 얻을 수 있었다. 모델별로 3회의 유효한 분류를 얻어낸 후, 그 결과를 이용하여 다른 모델과 hard voting 및 soft voting 방법을 이용하여 취합한 결과를 확인하였다.

분류 결과를 효율적으로 추출하기 위해 별도의 후처리를 진행하였다. 후처리하여도 분류할 수 없는 예외 결과가 나타난 경우에는, 인식할 수 있는 형태의 출력이 나타날 때까지 잘못 출력된 결과를 포함한 프롬프트를 LLM에 반복 입력하는 방법을 사용했다.

3.2 사용 모델

gpt-3.5-turbo, gpt-4, llama-2-13b-chat, dolly-v2-12b, vicuna-13b 모델을 이용하여 분류 결과를 추출했다. 비교를 위한 baseline 모델로는 GLUE Human baseline, T5(Small, Base, Large, 3B, 11B)[16], RoBERTa[17], ANNA[18], MT-DNN[19], BERT(base, large)[20], DistilBERT[21]의 성능을 인용하였다.

추론을 위해 gpt 모델들은 OpenAI의 API를 사용했고, llama-2-13b-chat, dolly-v2-12b, vicuna-13b 모델은 replicate.com[5]의 API를 이용했다.

3.3 사용 데이터셋

데이터셋으로는 GLUE 벤치마크의 SST-2 데이터셋 훈련 데이터 중 1,000개의 샘플을 추출하여 테스트 데이터로 사용했다. SST-2 데이터셋 전체 약 70,000개의 데이터 중 1,000개 샘플의 신뢰구간 99%에서 표본오차는 4.05% 이다.

[5]https://replicate.com/

3.4 실험 구성

각 모델에 대해 모델별로 동일한 프롬프트를 사용하여 3회의 유효한 분류 결과를 추출한 후 성능을 측정한다. 추출한 결과에 대한 hard/soft voting 결과도 함께 측정한다. 이때 사용한 프롬프트는 아래와 같은 형태이다.

3.4.1 gpt-4, gpt-3.5-turbo, llama-2-13b-chat, vicuna-13b 사용 프롬프트

아래는 “sentiment”라는 key와 “positive” 혹은 “negative”라는 value를 가지는 JSON 형태로 분류하도록 지시하는 프롬프트이다. 이진 분류 태스크이기 때문에, 중립적인 대답을 피하고자 ‘There is no neutral answer’라는 부분을 추가했다. 아래에서는 “is funny , charming and quirky in her feature film acting debut as amy ”라는 문장을 예시로 사용했다.

Prompt

Classify the text into “positive” or “negative”.
Return as JSON with key “sentiment”.
Return sentiment must be “positive” or “negative”.
There is no neutral answer.
Text: “is funny , charming and quirky in her feature film acting debut as amy ”
ANSWER:

Model Answer (gpt-3.5-turbo)

{“sentiment”: “positive”}

아래는 유효하지 않은(INVALID) 응답이 나타났을 때 재시도하기 위한 프롬프트이다. 응답이 유효하지 않다는 것은 응답이 1개의 JSON으로 파싱되지 않거나, “sentiment”라는 key가 없거나, “sentiment” key의 value가 “positive”나 “negative”가 아닌 경우를 의미한다. 아래 프롬프트에서는 유효하지 않은 결과를 생성하기 위해, “neutral”이라는 결과가 나타나 자주 실패하는 문장인 “being neither ”를 예시로 사용했다.

Prompt
Classify the text into “positive” or “negative”.
Return as JSON with key “sentiment”.
Return sentiment must be “positive” or “negative”.
There is no neutral answer.
Text: “being neither ”
INVALID ANSWER: {“sentiment”:“neutral”}
VALID ANSWER:

Model Answer (vicuna-13b)
{“sentiment”:“negative”}

3.4.2 dolly-v2-12b 사용 프롬프트

아래 프롬프트는 dolly 모델로 JSON 형태 응답을 얻기 위해 설계한 프롬프트이다. dolly 모델에서는 다른 모델에 사용한 프롬프트로는 JSON 응답이 온전하게 나타나지 않아 후처리에 어려움을 겪었다. 또한 dolly 모델에서는 텍스트가 중립적인 느낌이 강한 경우 “mixed”라는 출력이 자주 나타났기 때문에, “mixed”도 출력하지 않도록 명시하여 새로운 프롬프트를 작성했다. 아래 예시에서도 “is funny , charming and quirky in her feature film acting debut as amy”라는 문장을 사용했다.

Prompt
Return a VALID JSON object with key “sentiment”.
Return sentiment must be “positive” or “negative”.
Do not say anything other than JSON object result.
There is no “neutral”, “mixed” answer.
Answer must start with ‘{’.
Classify the sentiment of the sentence “is funny , charming and quirky in her feature film acting debut as amy ”.

Model Answer (dolly-v2-12b)
{“sentiment”: “positive”}

아래는 dolly 모델에서 유효하지 않은 응답이 나타났을 때 재시도하기 위한 프롬프트이다. 이 경우에도 응답이 유효하지 않다는 것은 응답이 1개의 JSON으로 파싱되지 않거나, “sentiment”라는 key가 없거나, “sentiment” key의 value가 “positive”나 “negative”가 아닌 경우를 의미한다. 아래 프롬프트에서도 유효하지 않은 결과를 생성하기 위해 “being neither ” 라는 문장을 예시로 사용했다.

Prompt
Return a VALID JSON object with key “sentiment”.
Return sentiment must be “positive” or “negative”.
Do not say anything other than JSON object result.
There is no “neutral”, “mixed” answer.
Answer must start with ‘{’.
Classify the sentiment of the sentence “being neither ”.
INVALID ANSWER: {“sentiment”: “-1”}
VALID ANSWER:

Model Answer (dolly-v2-12b)
{“sentiment”: “negative”}

3.4.3 후처리 방법

모델의 전체 출력 결과를 소문자로 변환시켜, 대/소문자에 상관없이 결과를 얻을 수 있도록 구성했다. 모델의 프롬프트에 ‘JSON’ 형태로 출력하도록 지시했기 때문에, 모델의 출력에는 ‘{’로 시작하고 ‘}’로 끝나는 JSON 형태의 출력이 포함될 것이다. 그리고 JSON에 포함될 값으로는 ‘sentiment’라는 key를 갖도록 지시했기 때문에, ‘{’, ‘}’의 위치를 찾아 JSON object로 파싱한 후 ‘sentiment’라는 key로 접근하여 결과를 얻어냈다.

3.5 실험 결과

서술한 실험 방식을 통해 표 1에 기록된 결과를 얻었다. Consistency(일관성)는 API의 기본 설정으로 3회 분류시 모델이 얼마나 일관된 출력을 나타내는지를 표현한 부분이다. 성능이 높은 모델일수록 일관성이 높게 나타나는 경향을 보였다.

JSON 파싱에 실패하거나 ‘positive’, ‘negative’가 아닌 출력이 나타나면 결과를 얻을 때까지 재시도하게 되는데, 표 1에 기록된 Retry rate는 그런 재시도 횟수의 비율(실패율), 즉 1회 분류당 실패율을 나타낸다. 예를 들어 retry rate가 0.001인 GPT4는 1,000회의 분류를 수행할 때 1회 재시도한 것이다. 이 경우도 성능이 높은 모델일수록 실패율이 낮게 나타나는 경향을 보였다.

Voting은 전체 5개 모델, 상위 3개 모델, 하위 3개 모델에 대해 각각 적용했다. Hard voting의 결과는 개별 모델의 결과보다 향상되지 않았지만, Soft voting에서 하위 3개 모델에 대한 결과는 기존보다 소폭 향상된 결과를 보였다.

4. 논의

공개된 LLM API에 zero shot 프롬프팅으로 별도의 학습 및 튜닝 없이 바로 분류 결과를 예측했을 때, 기존 PLM에 준하는 성능을 바로 얻을 수 있었다. 표 2에 나타나는 성능은 태스크를 처리하기 위한 훈련을 했거나, 튜닝을 따로 진행한 baseline의 결과이다. 본 논문에서 제시한 결과(표 1)는 태스크를 처리하기 위한 튜닝을 진행하지 않았음에도 최고 0.933의 정확도를

	GPT4	**GPT3.5-turbo**	**llama2-13b**	**vicuna-13b**	**dolly-v2-12b**
Consistency	0.984	0.951	0.985	0.822	0.584
Retry rate	0.003	0.025	0.08	0.14	1.322
Accuracy	**0.933**	**0.933**	0.883	0.874	0.679
	Voting				
Hard Voting Accuracy	0.923				
	0.929			-	
	-		0.883		
Soft Voting Accuracy	0.927				
	0.928			-	
	-		**0.887**		

표 1. 비훈련 방식의 LLM 분류 결과 및 Voting 성능 (본 연구)

Model		**Accuracy**
T5[16]	3B	0.974
	11B	**0.975**
RoBERTa[17]		0.967
ANNA[18]		0.964
T5[16]	Small	0.918
	Base	0.952
	Large	0.963
MT-DNN[19]		0.956
BERT[20]	base	0.935
	large	0.949
DistilBERT[21]		0.913

표 2. 태스크별로 학습된 모델의 성능 (baseline)

기록하였다. 고성능 LLM 들은 실패율도 낮고 일관성도 높기 때문에 PLM을 대체하여 사용할 여지가 있음을 확인하였다.

하지만 본 연구는 상대적으로 쉬운 이진 분류 태스크로만 수행되었기 때문에 더 복잡한 데이터 및 더 다양한 태스크를 활용한 추가적인 검증이 필요하다. 또한 API를 활용하기 때문에 예측 시간은 오래 걸리는 편이고, 모델에 따라 원하는 형태의 출력을 주지 않아 많은 재시도를 해야 하는 경우도 있었다.

향후에는 더 다양한 태스크에 대해 추가적인 검증을 수행할 예정이다. 또한, 모델에 따라 적절한 프롬프트를 자동으로 작성해 주는 모델 및 방법론을 연구할 예정이다.

5. 결론

본 연구에서는 LLM을 학습시키거나 튜닝하는 데 너무 많은 리소스가 필요하다는 문제를 해결하고자 하였다. zero-shot 프롬프팅으로 별도의 학습 및 튜닝 없이 LLM이 기존 자연어처리 태스크 문제를 해결할 수 있는 능력을 확인했다.

별도 학습 및 튜닝 없이 일부 고성능 LLM만으로도, 태스크에 맞는 훈련 혹은 튜닝을 진행한 PLM에 준하는 분류 성능을 확인할 수 있었다. 또한 고성능 LLM의 경우 응답 실패율이 낮고 일관성이 높아 활용 가능성이 충분하다는 것을 확인하였다.

향후 더 다양한 태스크에서 검증하고 모델별로 적절한 프롬프트를 자동으로 작성하는 모델 및 방법론을 연구할 예정이다.

감사의 글

이 논문은 2022년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2022R1F1A1071047)

참고문헌

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023.

[2] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, “Llama: Open and efficient foundation language models,” 2023.

[3] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, “Stanford alpaca: An instruction-following llama model,” https://github.com/tatsu-lab/stanford_alpaca, 2023.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu,

C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[5] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," 2019.

[6] OpenAI, "Gpt-4 technical report," 2023.

[7] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," *arXiv preprint arXiv:2009.03300*, 2020.

[8] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open foundation and fine-tuned chat models," 2023.

[9] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, "Gqa: Training generalized multi-query transformer models from multi-head checkpoints," 2023.

[10] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," March 2023. [Online]. Available: https://lmsys.org/blog/2023-03-30-vicuna/

[11] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging llm-as-a-judge with mt-bench and chatbot arena," 2023.

[12] S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, and O. van der Wal, "Pythia: A suite for analyzing large language models across training and scaling," 2023.

[13] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy, "The pile: An 800gb dataset of diverse text for language modeling," 2020.

[14] M. Conover, M. Hayes, A. Mathur, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, and R. Xin. (2023) Free dolly: Introducing the world's first truly open instruction-tuned llm. [Online]. Available: https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

[15] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Oct. 2013. [Online]. Available: https://aclanthology.org/D13-1170

[16] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2020.

[17] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[18] C. Jun, H. Jang, M. Sim, H. Kim, J. Choi, K. Min, and K. Bae, "Anna: Enhanced language representation for question answering," 2022.

[19] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," 2019.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[21] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020.