
Amazon KDD Cup 2024: Multi-Task Online Shopping Challenge for LLMs

Team Ensemble 심규재, 하지원, Matti Zinke

Index

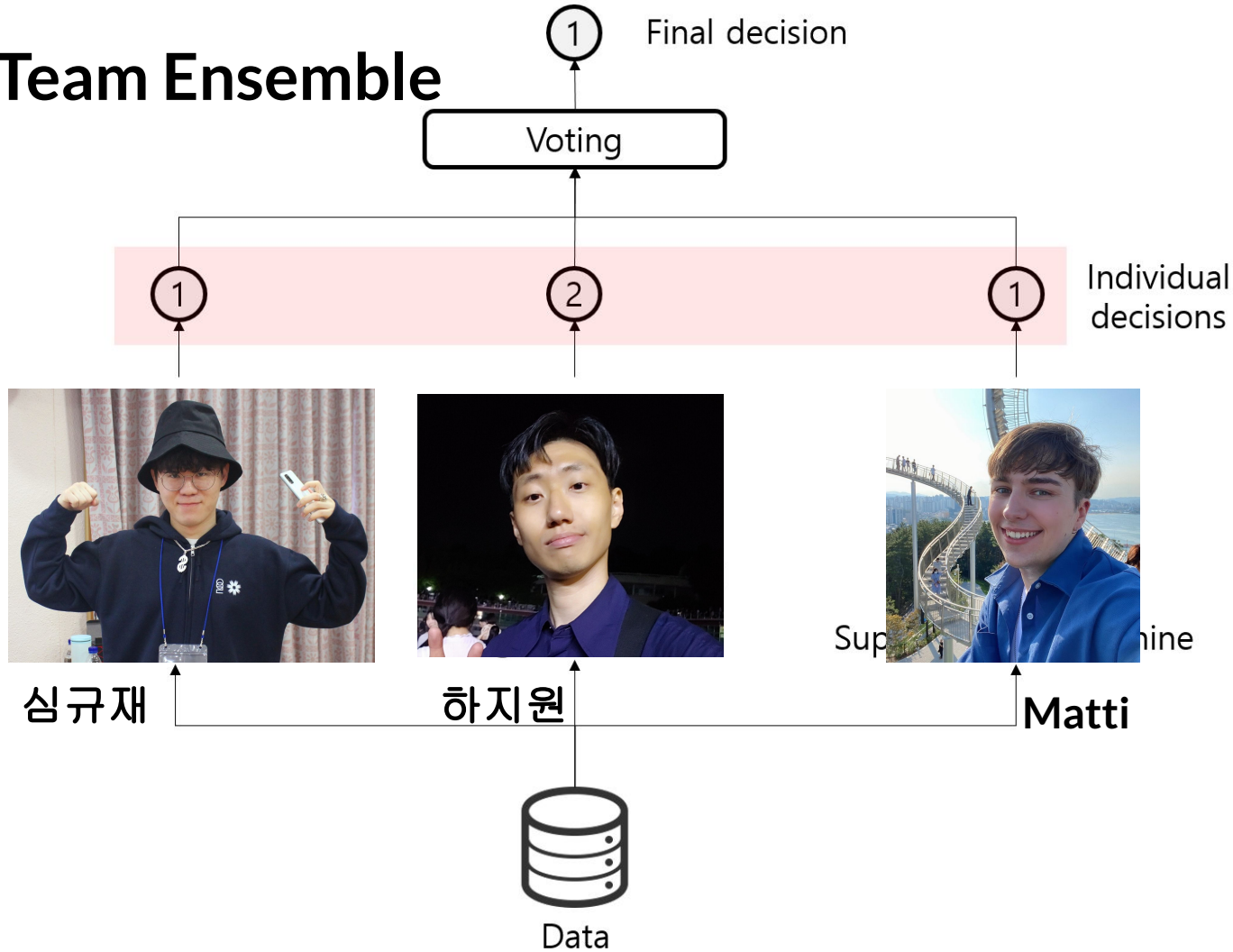
1. Team members
2. What is KDD CUP?
3. Code interpretation
4. Model Performance
5. DPO & DARE TIES
6. Future research & thoughts



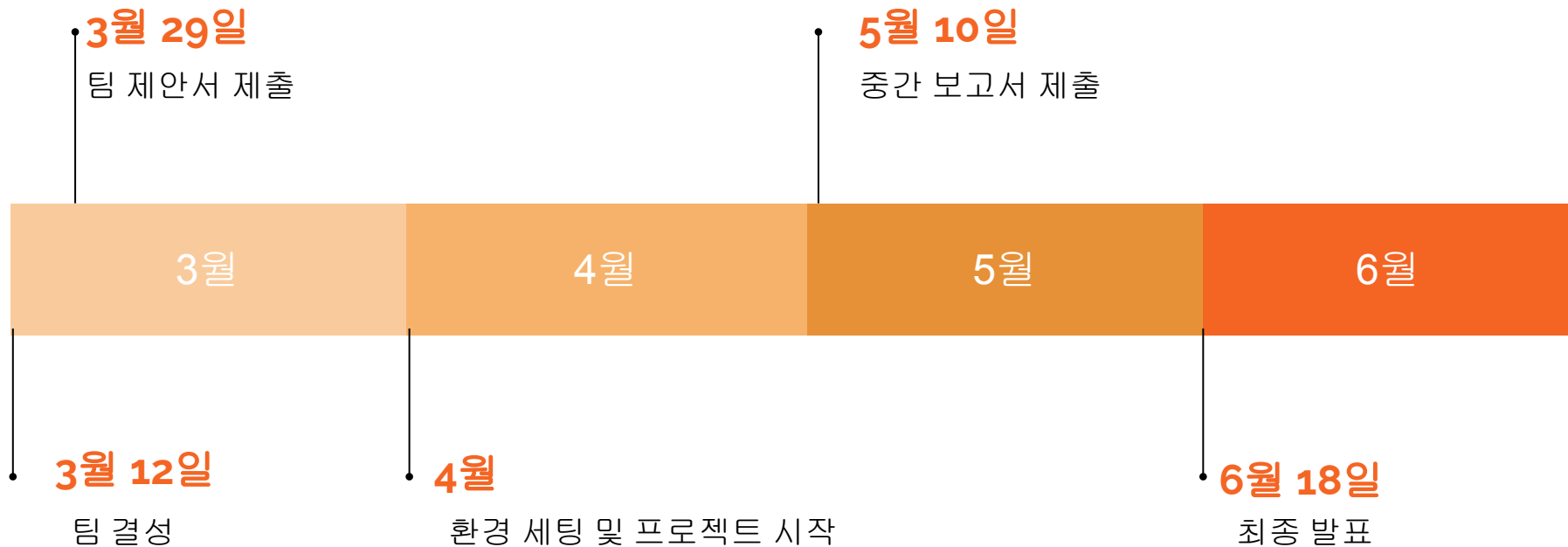
1

Team members

Team Ensemble



Timeline





2. What is KDD CUP?



**Knowledge
Discovery
Data Mining**



Track & Goal

Track1	shopping concept understanding
Track2	shopping knowledge reasoning
Track3	user behavior alignment
Track4	multi-lingual abilities
Track5	All-around

Shopbench Dataset

input_field:	This field contains the instructions and the question that should be answered by the model.
output_field:	This field contains the ground truth answer to the question.
task_type:	This field contains the type of the task (Details in the next Section, "Tasks")
task_name:	This field contains the name of the task. However, the exact task names are redacted, and we only provide participants with hashed task names (e.g. task1, task2).
metric:	This field contains the metric used to evaluate the question (Details in Section "Evaluation Metrics").
track:	This field specifies the track the question comes from.

from Shopbench to development.json

# Tasks	# Questions	# Products	# Product Category	# Attributes	# Reviews	# Queries
57	20598	~13300	400	1032	~11200	~4500




#ShopBench is split into a few-shot development set and a test set

Tasks: 18

Questions: 96

Example in development.json

```
{  
  "input_field": "Instructions: Tell me what this product category is about\nInput: Toggle  
Switch\nOutput:",  
  "output_field": "A toggle switch is an electric switch operated by means of a projecting  
lever that is moved up and down.",  
  "task_name": "task1",  
  "task_type": "generation",  
  "metric": "sent-transformer",  
  "is_multiple_choice": False,  
  "track": "amazon-kdd-cup-24-understanding-shopping-concepts"  
}
```



3 Code Implementation

DummyModel

```
11 class DummyModel(ShopBenchBaseModel):  
12     """  
13     A dummy model implementation for ShopBench, illustrating how to handle both  
14     multiple choice and other types of tasks like Ranking, Retrieval, and Named Entity Recognition.  
15     This model uses a consistent random seed for reproducible results.  
16     """
```

- 쇼핑 관련 문제를 풀기 위한 기본적인 모델

```

22 def predict(self, prompt: str, is_multiple_choice: bool) -> str:
23     """
24     Generates a prediction based on the input prompt and task type.
25
26     For multiple choice tasks, it randomly selects a choice.
27     For other tasks, it returns a list of integers as a string,
28     representing the model's prediction in a format compatible with task-specific parsers.
29
30     Args:
31         prompt (str): The input prompt for the model.
32         is_multiple_choice (bool): Indicates whether the task is a multiple choice question.
33
34     Returns:
35         str: The prediction as a string representing a single integer[0, 3] for multiple choice tasks,
36             or a string representing a comma separated list of integers for Ranking, Retrieval tasks,
37             or a string representing a comma separated list of named entities for Named Entity Recognition tasks.
38             or a string representing the (unconstrained) generated response for the generation tasks
39             Please refer to parsers.py for more details on how these responses will be parsed by the evaluator.
40     """
41     possible_responses = [1, 2, 3, 4]
42
43     if is_multiple_choice:
44         # Randomly select one of the possible responses for multiple choice tasks
45         return str(random.choice(possible_responses))
46     else:
47         # For other tasks, shuffle the possible responses and return as a string
48         random.shuffle(possible_responses)
49         return str(possible_responses)
50     # Note: As this is dummy model, we are returning random responses for non-multiple choice tasks.
51     # For generation tasks, this should ideally return an unconstrained string.
52

```

기본적으로 문제가 단답형인지 아닌지 구분

단답형일 경우, 1, 2, 3, 4 중에서 무작위로 숫자 하나를 골라 문자열로 반환

단답형이 아닐 경우, [1, 2, 3, 4] 리스트를 무작위로 나열한 결과를 문자열로 반환

Baseline Model-Vicuna



LLaMA와 Alpaca에 영감을 받아 개발되었고, **ShardGPT***에서 수집된 사용자들의 대화로 이뤄진 데이터로 LLaMA를 파인튜닝하여 구축된 챗봇

*: 사용자 프롬프트와 **ChatGPT**의 해당 답변 결과를 서로 공유할 수 있는 웹사이트

```

1  from typing import List, Union
2  import random
3  import os
4
5  from transformers import AutoTokenizer, AutoModelForCausalLM
6
7  from .base_model import ShopBenchBaseModel
8
9  # Set a consistent seed for reproducibility
10 AICROWD_RUN_SEED = int(os.getenv("AICROWD_RUN_SEED", 3142))
11
12
13 class Vicuna2ZeroShot(ShopBenchBaseModel):
14     def __init__(self):
15         random.seed(AICROWD_RUN_SEED)
16
17         model_path = 'lmsys/vicuna-13b-v1.3'
18
19         self.tokenizer = AutoTokenizer.from_pretrained('./models/vicuna-13b-v1.3/', trust_remote_code=True)
20         self.model = AutoModelForCausalLM.from_pretrained('./models/vicuna-13b-v1.3/', device_map='auto', torch_dtype='auto', trust_remote_code=True, do_sample=True)
21         self.system_prompt = "You are a helpful online shopping assistant. Please answer the following question about online shopping and follow the given instructions.\n\n"

```

- LLM을 사용하고, 토큰화 방식을 정의하기 위해 **transformers** 모듈에서 **AutoTokenizer**, **AutoModelForCasualLLM** 메소드 불러오기
- **Vicuna** 모델을 불러온 뒤, 온라인 쇼핑 도우미로서 지시 사항을 따라서 질문에 답할 것을 시스템 프롬프트로 명령
- 방금 입력한 시스템 프롬프트에 추가 명령을 내리기 위한 프롬프트를 정의하여 이를 최종 프롬프트로 결정
- LLM이 이 최종 프롬프트를 이해할 수 있도록 토큰화하여 모델에 입력
- 단답형 문제의 경우 출력할 최대 토큰 개수를 1로 설정하고, 아닐 경우 100으로 설정

```

24  def predict(self, prompt: str, is_multiple_choice: bool) -> str:
25      prompt = self.system_prompt + prompt
26      inputs = self.tokenizer(prompt, return_tensors='pt')
27      inputs.input_ids = inputs.input_ids.cuda()
28
29      if is_multiple_choice:
30          generate_ids = self.model.generate(inputs.input_ids, max_new_tokens=1, temperature=1)
31      else:
32          generate_ids = self.model.generate(inputs.input_ids, max_new_tokens=100, temperature=1)
33
34      result = self.tokenizer.batch_decode(generate_ids, skip_special_tokens=True, clean_up_tokenization_spaces=False)[0]
35      generation = result[len(prompt):]
36
37      return generation
38

```




4. Model Performance

-Top 2 Model Performance

→ **1st : DARE_TIES_13B**

0.650502408251804

→ **2nd : 13B_MATH_DPO**

0.6148088357017373

Local Evaluation Scores on 20 LLM

LLM 모델명	local eval score	LLM 모델명	local eval score
Vicuna-7B-v1.5	0.30195231568858446	stablelm-2-12b-chat	0.46852982312194846
Meta-Llama-3-8B-Instruct	0.5223284217275423	yam-peleg/Experiment26-7B	0.5280649422643182
Qwen1.5-7B-sft-0502	0.5610776246185725	MaziyarPanahi/Calme-7B-Instruct-v0.2	0.5153258876929895
Qwen-14B-Llamafied	0.41060220095147437	chihoonlee10/T3Q-EN-DPO-Mistral-7B	0.5261771492540045
Qwen1.5-MoE-A2.7B-Chat	0.45975688578281115	zhengr/MixTAO-7Bx2-MoE-v8.1	0.5896902518997831
Hermes-2-Theta-Llama-3-8B	0.5125085712022551	01-ai/Yi-9B	0.4544301238436799
DARE_TIES_13B	0.5956676126566732 → 0.6213757661369984	01-ai/Yi-6B	0.30425252628858446
yunconglong/13B_MATH_DPO	0.6148088357017373	yunconglong/Truthful_DPO_ToanGrc_FusionNet_7Bx2_MoE_13B	0.6140449173494128
yunconglong/MoE_13B_DPO	0.5788059831556638	vicgalle/CarbonBeagle-11B-truthy	0.5593031144819792
BarraHome/Mistroll-7B-v2.2	0.5682951110016166	TwT-6/cr-model-v1	0.577217490607002

Analysis on Performance Indicator

Model Name	average	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K	localscore
DARE_TIES_13B	77.1	74.32	89.5	64.47	78.66	88.08	67.55	0.621375766
yunconglong/13B_MATH_DPO	77.08	74.66	89.51	64.53	78.63	88.08	67.1	0.614808836
yunconglong/Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B	77.44	74.91	89.3	64.67	78.02	88.24	69.52	0.614044917
zhengr/MixTAO-7Bx2-MoE-v8.1	77.5	73.81	89.22	64.92	78.57	87.37	71.11	0.589690252
yunconglong/MoE_13B_DPO	77.05	74.32	89.39	64.48	78.47	88	67.63	0.578805983
TwT-6/cr-model-v1	77.32	70.65	87.85	74.73	80.47	83.66	66.57	0.577217491
BarraHome/Mistroll-7B-v2.2	76.76	72.78	89.16	64.35	78.1	85	71.19	0.568295111
Qwen1.5-7B-sft-0502	61.99	55.12	77.18	61.68	50.72	71.67	55.57	0.561077625
vicgalle/CarbonBeagle-11B-truthy	76.1	82.27	89.31	66.55	78.55	83.82	66.11	0.559303114
yam-peleg/Experiment26-7B	76.64	73.88	89.15	64.32	78.24	84.93	70.43	0.528064942
chihoonlee10/T3Q-EN-DPO-Mistral-7B	76.43	73.04	89.3	64.13	78.71	85.32	68.08	0.526177149
Meta-Llama-3-8B-Instruct	66.87	60.75	78.55	67.07	51.65	74.51	68.89	0.522328422
MaziyarPanahi/Calme-7B-Instruct-v0.2	76.61	73.12	89.19	64.36	78	84.93	70.05	0.515325888
Hermes-2-Theta-Llama-3-8B	68.1	66.04	84.95	63.36	55.75	78.06	60.42	0.512508571
Qwen/Qwen1.5-14B	66.7	56.57	81.08	69.36	52.06	73.48	67.63	0.499776516
stablelm-2-12b-chat	68.38	64.85	85.96	61.06	62.01	78.53	57.85	0.468529823
Qwen1.5-MoE-A2.7B-Chat	57.22	53.67	80.54	60.97	50.56	69.38	28.2	0.459756886
01-ai/Yi-9B	63.17	61.18	78.82	70.06	42.45	77.51	48.98	0.454430124
Qwen-14B-Llamafied	63.09	55.2	82.31	66.11	45.6	76.56	52.77	0.410602201
01-ai/Yi-6B	54.08	55.55	76.57	64.11	41.96	74.19	12.13	0.304252526



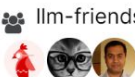


Correlation Analysis

	average	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K	localscore
average	1							
ARC	0.922854	1						
HellaSwag	0.930937	0.914986	1					
MMLU	0.177081	0.071087	-0.01644	1				
TruthfulQA	0.950588	0.924376	0.947515	0.039973	1			
Winogrande	0.920184	0.923232	0.91484	0.090632	0.901755	1		
GSM8K	0.862531	0.67405	0.687896	0.223539	0.701635	0.652652	1	
localscore	0.816958	0.699199	0.680244	0.077637	0.769455	0.675213	0.807009	1

Regression Analysis

회귀분석 통계량								
다중 상관계수	0.883908							
결정계수	0.781293							
조정된 결정계수	0.653715							
표준 오차	0.045697							
관측수	20							
분산 분석								
	자유도	제곱합	제곱 평균	F 비	유의한 F			
회귀	7	0.08952	0.012789	6.124007	0.003254			
잔차	12	0.025059	0.002088					
계	19	0.114579						
	계수	표준 오차	t 통계량	P-값	하위 95%	상위 95%	하위 95.0%	상위 95.0%
Y 절편	1.094865	0.551996	1.983467	0.070673	-0.10783	2.297561	-0.10783	2.297561
average	0.001916	0.056407	0.033963	0.973465	-0.12098	0.124816	-0.12098	0.124816
ARC	-0.00049	0.005062	-0.09649	0.924722	-0.01152	0.010542	-0.01152	0.010542
HellaSwag	-0.01206	0.011697	-1.03116	0.322802	-0.03755	0.013424	-0.03755	0.013424
MMLU	-0.00311	0.009207	-0.33756	0.741527	-0.02317	0.016952	-0.02317	0.016952
TruthfulQA	0.004677	0.010528	0.444257	0.664763	-0.01826	0.027615	-0.01826	0.027615
Winogrande	0.001078	0.015964	0.067554	0.947253	-0.0337	0.035861	-0.0337	0.035861
GSM8K	0.002707	0.009745	0.277812	0.785885	-0.01852	0.023939	-0.01852	0.023939

Team's Final Ranking

▼ 23		0.689	0.764	0.168	0.639	0.631	3	Wed, 12 Jun 2024 15:34
▲ 23		0.689	0.745	0.253	0.725	0.610	3	Sun, 16 Jun 2024 12:47
▼ 25		0.673	0.753	0.298	0.712	0.533	7	Fri, 7 Jun 2024 07:57
▼ 25		0.673	0.768	0.218	0.656	0.545	4	Tue, 11 Jun 2024 17:58
▼ 27		0.669	0.783	0.365	0.409	0.615	4	Mon, 10 Jun 2024 00:04



5. DPO & DARE TIES

-가장 높은 성능을 낸 모델은 어떤
기술들을 기반으로 하는가?

→ What is DPO?

→ What is DARE_TIES?

DPO

Direct Preference Optimization: Your Language Model is Secretly a Reward Model

대규모 비지도(unsupervised) 언어 모델은 학습 방식의 비지도성 때문에 행동을 정밀하게 제어하기 쉽지 않음.

인간에 의한 레이블을 수집하여 비지도 언어 모델을 RLHF(reinforcement learning from human feedback) 기법으로 인간의 선호도에 맞추어 fine-tuning하는 것이 기존의 방식.

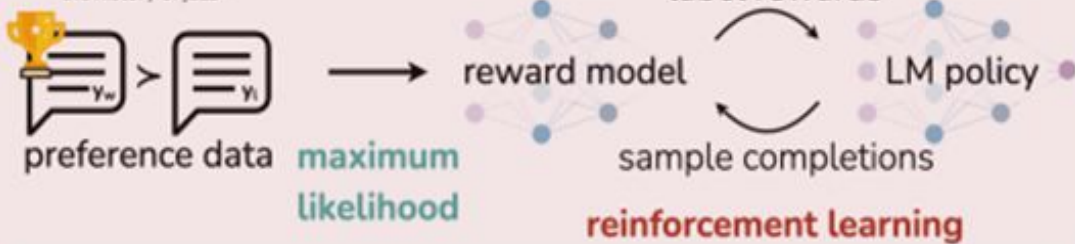
하지만 RLHF는 보상 모델을 별도로 학습해야 하기 때문에 복잡하고 불안정한 절차이며, 계산 비용이 높은 기법임.

반면 DPO는 RLHF의 보상모델에 새로운 매개변수를 도입해서 일반적인 classification loss만으로 RLHF 문제를 해결하게 해줌. 즉, 보상모델을 따로 학습할 필요를 없애줌.

DPO

Reinforcement Learning from Human Feedback (RLHF)

x: "write me a poem about
the history of jazz"



Direct Preference Optimization (DPO)

x: "write me a poem about
the history of jazz"



DARE

Drop And REscale

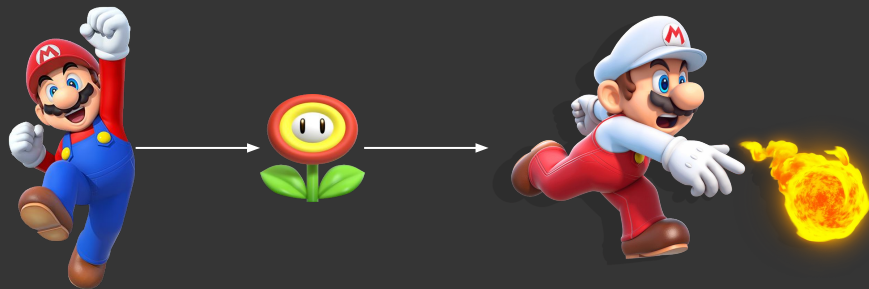
Language Models are Super Mario: Absorbing Abilities from Homologous Models as a Free Lunch

언어모델이 재훈련하거나 GPU를 사용하지 않고 상응하는 모델의 파라미터를 흡수함으로써 새 기능을 얻게 해주는 기술

Drop: 델타 파라미터를 (파인튜닝된 모델과 초기학습된 모델 간 차이) 무작위로 p비율만큼 골라서 제거(값을 0으로 감소).

Rescale: 나머지 파라미터의 값은 $1/(1-p)$ 비율로 재조정하여 해당 영역이 담당하는 작업에 대한 성능을 상승.

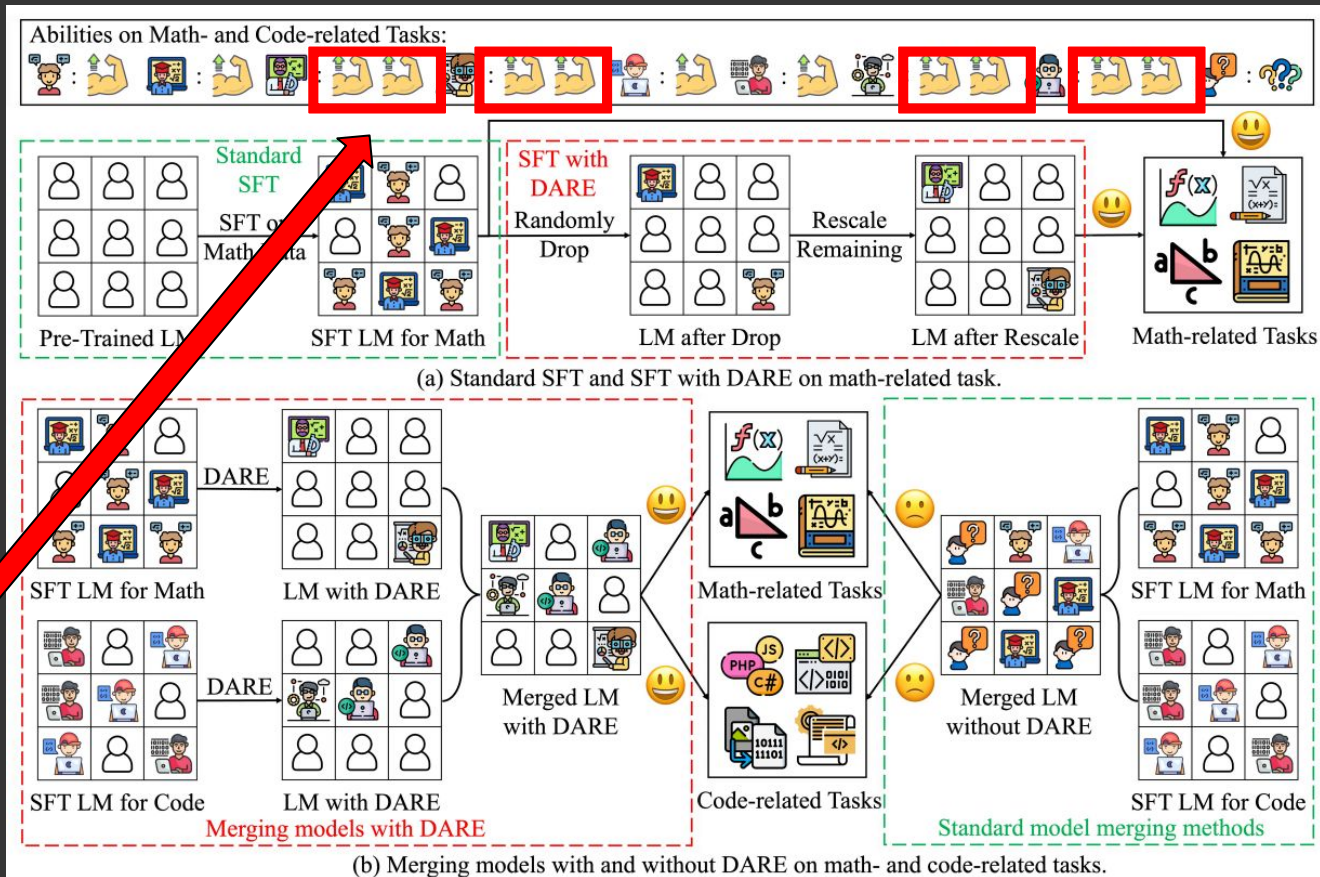
출처: <https://arxiv.org/abs/2311.03099>



팁

슈퍼 마리오 게임 시리즈의 주인공 마리오가 파이어 플라워를 획득해서 파이어볼이라는 새 능력을 얻는 것을 생각하면 쉽다!

DARE



근육 아이콘이 2개일수록 성능이 뛰어나다는 뜻!

TIES-Merging:

Resolving Interference When Merging Models

Prateek Yadav et al.(2023)

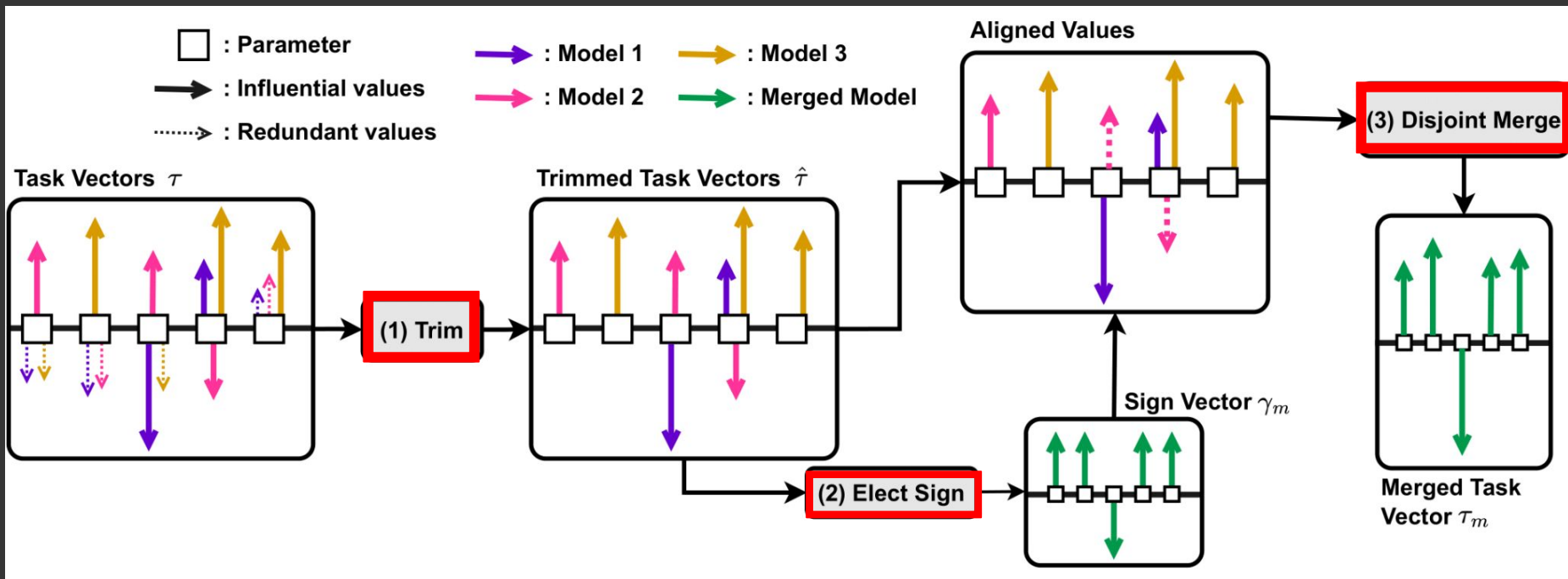
하나의 Task에 fine-tuning된 모델을 병합하여 하나의 모델로 복수의 Task를 다루려는 시도.


병합과정에 두 가지 주요 간섭 원인이 있는데 (a) 중복 매개변수 값으로 인한 간섭과 (b) 모델 전반에 걸쳐 주어진 매개변수 값의 부호에 대한 불일치임.

이를 해결하는 것이 TRIM(1), ELECT SIGN(2) & MERGE(TIES-Merging)(3) 방법임.

(1) 미세 조정 중에 약간만 변경된 매개변수를 재설정하고, (2) 부호 충돌 문제를 해결, (3) 최종 합의된 부호와 일치하는 매개변수만 병합하여 간섭을 완화한다.

TIES-Merging:





6 Future research & Thoughts

Future research

Shopbench의
모든 데이터를 사용
가능하다면?

LLM에 API를 통해
아마존 사의 쇼핑
사이트를 연결
해본다면?

복수의 문제 해결
LLM을 연결 할 수
있다면?

Thoughts

Task별 runtime이
한정되어 있어 더 많은
모델을 시도하지 못한점이
아쉬웠다.

-규재

GPU등의 설비가 없는 참가
팀들에게 시작 환경 세팅
등의 부분부터 제약이 있어
쉽지 않은 프로젝트 였으나,
잘 마무리했고 유의미한
성능을 얻어냈다.

-지원

프롬프트의 변경 만으로도
성능개선이 이루어졌다.
충분한 시간이 주어졌다면
다양한 프롬프트
엔지니어링 기법을
적용해보고 싶다.

-마티

—

Thank you for listening our presentation! :D