

# Phi 알아보기

- 기존 대규모 언어 모델(LLM) 학습 접근 방식의 한계를 극복하고, 효율적이고 환경적으로 지속 가능한 학습 방법을 제안

## Phi-1: Textbooks Are All You Need

### 1. 핵심 주제:

- 소규모 LLM을 활용한 효율적인 코드 생성
  - 기존 대규모 모델에 비해 작은 규모의 Phi-1 모델(1.3B 파라미터)을 개발하여, HumanEval 및 MBPP 코딩 벤치마크에서 높은 성능을 달성.
- "텍스트북 품질" 데이터 사용
  - 일반적인 웹 기반 데이터가 아닌, 철저히 선정되고, 교육적으로 가치가 높은 데이터셋을 사용함으로써 성능을 극대화.
- 학습 비용 및 환경 부담 감소
  - 대규모 데이터나 GPU 사용량이 아닌, 최적화된 데이터 품질과 효율적인 훈련 과정으로 학습 비용을 크게 절감.

### 2. 연구 배경

#### 2.1 문제 정의

- 기존 접근법의 한계
  - LLM 학습은 일반적으로 방대한 데이터셋(수백억수조 개의 토큰)과 대규모 모델(수십억수천억 파라미터)을 요구함.
  - 데이터의 양에 초점을 맞추다 보니, 데이터 내 중복성, 노이즈, 비효율적 학습 문제가 발생.
  - 학습 비용 및 환경적 부담이 매우 크며, 이는 지속 가능하지 않음.
- 코드 생성 모델의 도전 과제
  - 기존 LLM 기반 코드 생성 모델의 데이터는 불완전하거나 맥락이 부족하여 알고리즘적 사고 학습에 적합하지 않음.
  - 예시: The Stack, StackOverflow 데이터는 설정 코드, GUI 구성 등 단순 작업 중심의 샘플이 많고 학습 신호가 약함.

#### 2.2 연구 목표

- 모델 규모 및 학습 비용 절감 소규모(1.3B 파라미터)의 Phi-1 모델로도 대규모 모델에 필적하는 성능 달성.
- 데이터 품질 극대화 고품질의 "텍스트북 품질" 데이터를 선별 및 생성하여 데이터 크기를 최소화하면서도 강력한 학습 신호 제공.
- 모델 효율성 검증 HumanEval 및 MBPP 벤치마크에서 경쟁 모델 대비 높은 성능을 달성하여 데이터 품질의 효과를 입증.

#### 2.3 연구 질문

- 고품질 데이터만으로 소규모 모델이 대규모 모델과 유사한 성능을 낼 수 있을까?
- 데이터 품질 개선이 Scaling Law를 넘어서 모델 성능에 미치는 영향을 어떻게 정량화할 수 있을까?

#### 2.4 연구 기여

- 데이터 품질의 중요성 제시
  - 기존 "데이터 양 vs. 모델 크기" 위주의 Scaling Law를 벗어나, 데이터 품질 중심의 접근법으로 새로운 연구 방향 제안.
- 환경 부담 감소
  - 학습 비용을 크게 절감하여 지속 가능한 AI 학습 방법론 제시. (예: Phi-1은 8개의 A100 GPU에서 단 4일간 훈련으로 성과 달성).

Date	Model	Model size (Parameters)	Dataset size (Tokens)	HumanEval (Pass@1)	MBPP (Pass@1)
2021 Jul	Codex-300M [CTJ <sup>+</sup> 21]	300M	100B	13.2%	-
2021 Jul	Codex-12B [CTJ <sup>+</sup> 21]	12B	100B	28.8%	-
2022 Mar	CodeGen-Mono-350M [NPH <sup>+</sup> 23]	350M	577B	12.8%	-
2022 Mar	CodeGen-Mono-16.1B [NPH <sup>+</sup> 23]	16.1B	577B	29.3%	35.3%
2022 Apr	PaLM-Coder [CND <sup>+</sup> 22]	540B	780B	35.9%	47.0%
2022 Sep	CodeGeeX [ZXZ <sup>+</sup> 23]	13B	850B	22.9%	24.4%
2022 Nov	GPT-3.5 [Ope23]	175B	N.A.	47%	-
2022 Dec	SantaCoder [ALK <sup>+</sup> 23]	1.1B	236B	14.0%	35.0%
2023 Mar	GPT-4 [Ope23]	N.A.	N.A.	67%	-
2023 Apr	Replit [Rep23]	2.7B	525B	21.9%	-
2023 Apr	Replit-Finetuned [Rep23]	2.7B	525B	30.5%	-
2023 May	CodeGen2-1B [NHX <sup>+</sup> 23]	1B	N.A.	10.3%	-
2023 May	CodeGen2-7B [NHX <sup>+</sup> 23]	7B	N.A.	19.1%	-
2023 May	StarCoder [LAZ <sup>+</sup> 23]	15.5B	1T	33.6%	52.7%
2023 May	StarCoder-Prompted [LAZ <sup>+</sup> 23]	15.5B	1T	40.8%	49.5%
2023 May	PaLM 2-S [ADF <sup>+</sup> 23]	N.A.	N.A.	37.6%	50.0%
2023 May	CodeT5+ [WLG <sup>+</sup> 23]	2B	52B	24.2%	-
2023 May	CodeT5+ [WLG <sup>+</sup> 23]	16B	52B	30.9%	-
2023 May	InstructCodeT5+ [WLG <sup>+</sup> 23]	16B	52B	35.0%	-
2023 Jun	WizardCoder [LXZ <sup>+</sup> 23]	16B	1T	57.3%	51.8%
2023 Jun	<b>phi-1</b>	1.3B	7B	50.6%	55.5%

Table 1: We use self-reported scores whenever available. Despite being trained at vastly smaller scale, **phi-1** outperforms competing models on HumanEval and MBPP, except for GPT-4 (also WizardCoder obtains better HumanEval but worse MBPP).

- 열(Column) 설명:
  - Date: 각 모델의 발표 시점.
  - Model: 모델 이름.
  - Model size (Parameters): 모델의 파라미터 크기.
  - Dataset size (Tokens): 모델이 학습에 사용한 데이터셋 크기.
  - HumanEval (Pass@1): HumanEval 벤치마크에서 코드 생성 정확도.
  - Pass@1: 주어진 입력에 대해 첫 번째 시도에서 정답 코드를 생성한 비율.
  - MBPP (Pass@1): MBPP(Mostly Basic Python Problems) 벤치마크에서 Pass@1 성능.

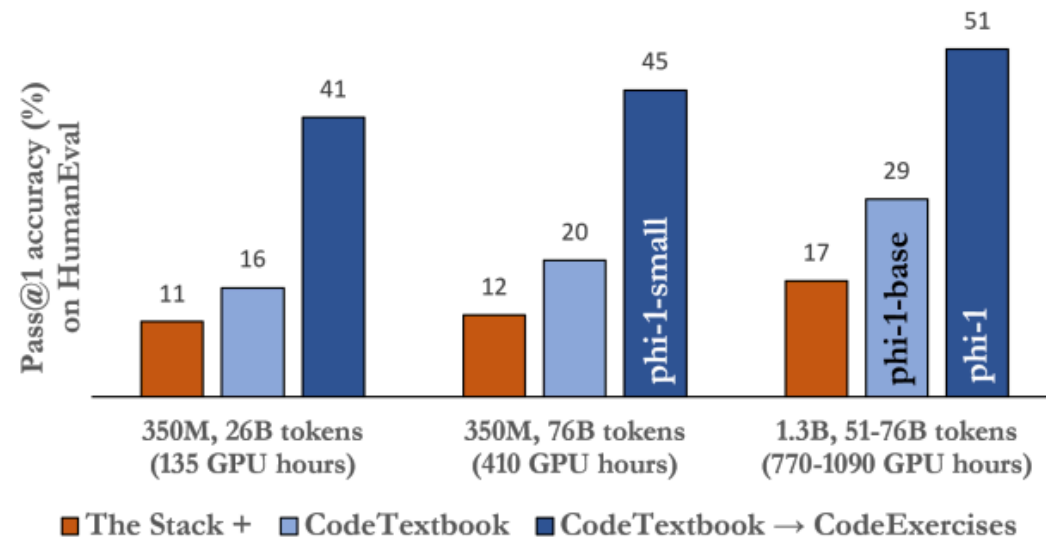


Figure 2.1 HumanEval에서 Pass@1 정확도 (%) 막대그래프의 그룹화는 두 가지 일반적인 스케일링 차원에 해당합니다:

계산 시간 증가 (더 많은 데이터를 반복 학습, 여기서는 26B 토큰에서 76B 토큰까지 학습) 모델 파라미터 수 증가 (350M에서 1.3B까지). 각 그룹 내의 열은 서로 다른 학습 데이터셋에 해당합니다:

(A) 첫 번째 열 (주황색): 중복 제거된 Python 파일 데이터셋인 The Stack(1.3B 파라미터 모델의 경우 StackOverflow 포함)으로 학습된 모델의 성능. (B) 두 번째 열 (연한 녹색): 새로운 데이터셋 구성인 CodeTextbook으로 학습된 모델의 성능. (C) 세 번째 열 (짙은 녹색): CodeTextbook 데이터셋으로 학습한 후 CodeExercises 데이터셋으로 미세조정된 모델의 성능. 1.3B 모델의 경우, phi-1과 phi-1-base는 51B 토큰(770 GPU 시간)을 학습한 후의 체크포인트를 의미하며, The Stack+ 모델은 76B 토큰(1090 GPU 시간)을 학습했습니다.

중요한 점은, phi-1-base 모델은 미세조정을 하지 않고도 CodeTextbook 데이터셋만으로 HumanEval에서 29%의 성능을 달성했음을 강조합니다. 이전까지 HumanEval에서 30%에 근접한 성능을 달성한 가장 작은 모델은 Replit-Finetuned로, 2.7B 파라미터를 갖추었으며, 우리보다 100배 더 많은 학습 토큰을 사용했습니다.

뿐만 아니라, CodeExercises 데이터셋으로 미세조정하여 phi-1 모델을 완성했을 때, HumanEval에서 51%의 최고 성능을 달성했으며, 이는 추가적으로 예상치 못한 코딩 능력을 발휘하게 만들었습니다(자세한 내용은 3장에서 확인 가능).

## Educational values deemed by the filter

### High educational value

```
import torch
import torch.nn.functional as F

def normalize(x, axis=-1):
    """Performs L2-Norm."""
    num = x
    denom = torch.norm(x, 2, axis, keepdim=True)
    .expand_as(x) + 1e-12
    return num / denom

def euclidean_dist(x, y):
    """Computes Euclidean distance."""
    m, n = x.size(0), y.size(0)
    xx = torch.pow(x, 2).sum(1, keepdim=True).
    expand(m, n)
    yy = torch.pow(y, 2).sum(1, keepdim=True).
    expand(m, m).t()
    dist = xx + yy - 2 * torch.matmul(x, y.t())
    dist = dist.clamp(min=1e-12).sqrt()
    return dist

def cosine_dist(x, y):
    """Computes Cosine Distance."""
    x = F.normalize(x, dim=1)
    y = F.normalize(y, dim=1)
    dist = 2 - 2 * torch.mm(x, y.t())
    return dist
```

### Low educational value

```
import re
import typing
...

class Default(object):
    def __init__(self, vim: Nvim) -> None:
        self._vim = vim
        self._denite: typing.Optional[SyncParent]
        = None
        self._selected_candidates: typing.List[int]
        ] = []
        self._candidates: Candidates = []
        self._cursor = 0
        self._entire_len = 0
        self._result: typing.List[typing.Any] = []
        self._context: UserContext = {}
        self._bufnr = -1
        self._winid = -1
        self._winrestcmd = ''
        self._initialized = False
        self._winheight = 0
        self._winwidth = 0
        self._winminheight = -1
        self._is_multi = False
        self._is_async = False
        self._matched_pattern = ''
        ...
```

Educational values deemed by the filter

우리의 필터링 방법론은 아래에서 논의될 **합성 데이터셋(synthetic datasets)**을 사용하지 않더라도 모델 성능을 크게 향상시켰습니다.

350M 파라미터 모델을 대상으로, 필터링되지 않은 Stack 데이터셋(중복 제거된 Python 코드)과 StackOverflow를 학습한 경우:

96k 스텝(약 200B 토큰)을 학습했음에도 불구하고, HumanEval 성능은 **12.19%**에서 포화 상태를 보였습니다. 반면, 필터링된 데이터셋을 사용하여 36k 스텝만 학습했을 때, HumanEval 성능은 **17.68%**로 상승했습니다.

여기에 필터링된 데이터셋과 **합성 텍스트북 데이터셋(synthetic textbooks dataset)**을 조합하여 학습한 결과, HumanEval 성능은 **20.12%**까지 추가로 향상되었습니다(Figure 2.1에 보고됨).

## 2.3 Model architecture and training

모델	파라미터 수	아키텍처 특징	학습 데이터	학습 비용	HumanEval 성능	MBPP 성능	특징
Phi-1-base	1.3B	Transformer, FlashAttention, Rotary Embedding(32D), 24 레이어, 히든 차원 2048, MLP 차원 8192, 32 Heads	CodeTextbook (6B 토큰, 8번 반복), 총 50B 토큰 학습	8 A100 GPU, 4일	29%	N/A	고품질 데이터만으로 경쟁력 있는 성능 달성
Phi-1	1.3B	Phi-1-base와 동일	CodeExercises (180M 토큰)으로 미세조정	추가 7시간 (8 A100 GPU)	50.6%	55.5%	미세조정으로 성능 대폭 향상, 예상치 못한 Emergent Properties 발현
Phi-1-small	350M	Transformer, 20 레이어, 히든 차원 1024, MLP 차원 4096, 16 Heads	동일한 데이터 파이프라인 사용	-	45%	N/A	소규모 모델로 데이터 품질과 효율성의 균형을 보여줌

## 추가 정보

- Phi-1-base와 Phi-1의 차이:
  - Phi-1은 Phi-1-base에 추가적으로 **CodeExercises** 데이터셋으로 미세조정되어 성능이 크게 향상됨.
- Phi-1-small의 목적:
  - 작은 모델 크기에서도 데이터 품질이 중요한 영향을 미친다는 점을 입증.

### Prompt:

```
""" Alice, Bob and Charles are playing games. Alice first pick a random number from 333 to 678, Bob then pick a number starting from Alice's number to 888, Charles then pick a number starting from 123 to Bob's number. Alice gets one point if Alice's number minus Charles's number is divisible by the floor of the square root of Bob's number, otherwise Bob gets one point. Simulate Alice's and Bob's points in 30 iterations.
"""
```

#### phi-1

```
points = [0, 0]
for i in range(30):
    alice_num = random.randint(333, 678)
    bob_num = random.randint(alice_num, 888)
    charles_num = random.randint(123, bob_num)
    if (alice_num - charles_num) % int(bob_num ** 0.5) == 0:
        points[0] += 1
    else:
        points[1] += 1
return f"Alice: {points[0]}, Bob: {points[1]}"
```

#### phi-1-base

```
self.points = 0
self.alice = 333
self.bob = 678
self.charles = 888
self.bob_points = 0
self.alice_points = 0
self.iterations = 30
self.alice_turn = True
self.bob_turn = True
self.alice_turn_points = 0
self.bob_turn_points = 0
self.alice_turn_points_history = []
self.bob_turn_points_history = []
self.alice_turn_points_history.append(self.alice_points)
self.bob_turn_points_history.append(self.bob_points)
```

#### phi-1-small

```
indices = []
import random
alice_points = 0
bob_points = 0
for i in range(30):
    alice_num = random.randint(333, 678)
    bob_num = random.randint(123, bob_num)
    if (alice_num - bob_num) % math.floor(math.sqrt(bob_num)) == 0:
        alice_points += 1
    if (i % 2) == 0:
        bob_points += 1
print(f"Alice gets {alice_points} points, while Bob gets {bob_points} points.")
```

우리가 직접 생성한 간단한 Python 함수를 사용하여, 모델이 미세조정 후 명령을 이해하고 이를 따르는 수준이 훨씬 높아졌음을 관찰했습니다. 특히, phi-1-base는 프롬프트에 포함된 논리적 관계를 처리하는 데 어려움을 겪었지만, phi-1은 질문을 해석하고 정답을 정확히 생성할 수 있었습니다. 이 예제에서, 심지어 350M 파라미터의 phi-1-small 모델도 문제를 어느 정도 이해하는 모습을 보였으나, 결과적으로 생성된 솔루션은 잘못되었습니다. 이와 같은 경향은 우리의 여러 상호작용에서 일관되게 나타났으며, 추가 예시는 부록 A에서 확인할 수 있습니다.

Model	Size	Training tokens	Score	HumanEval
CodeGen-Mono-350M [NPH <sup>+</sup> 23]	350M	577B	19%	13%
CodeGen-Mono-16.1B [NPH <sup>+</sup> 23]	16.1B	577B	38%	29%
Replit [Rep23]	2.7B	525B	37%	22%
StarCoder [LAZ <sup>+</sup> 23]	15.5B	1T	51%	34%
<b>phi-1-base</b>	1.3B	7B	37%	29%
<b>phi-1-small</b>	350M	7B	45%	45%
<b>phi-1</b>	1.3B	7B	52%	51%

Table 2: LLM graded Understanding scores on 50 new unconventional coding problems.

Table 2는 Phi-1 모델과 다른 모델들이 새로운 비정형적인 코딩 문제에 대해 얼마나 잘 수행했는지 비교한 결과를 나타냅니다. 이 평가의 목적은 기존 HumanEval 벤치마크와 달리, 학습 데이터와의 겹침 가능성을 최소화한 문제를 통해 모델의 진정한 능력을 평가하는 것입니다.

## 1. Table 2의 구성

- 열(Column) 설명:
  - Model Size:** 각 모델의 파라미터 크기.
  - Training Tokens:** 모델이 학습에 사용한 데이터셋의 토큰 수.
  - Score:** 새로운 비정형적 코딩 문제에서의 점수.
  - HumanEval:** 기존 HumanEval 벤치마크에서의 성능 (Pass@1).

## 2. Table 2의 주요 결과 및 분석

### 1. Phi-1의 뛰어난 성능:

- Phi-1은 새로운 비정형적 문제에서 **52%의 점수**를 기록하며, StarCoder(51%)를 근소하게 앞섰음.
- 학습 데이터 크기(7B)가 StarCoder(1T)에 비해 훨씬 적음에도 불구하고 성능이 더 뛰어남.
- HumanEval에서도 StarCoder를 능가하며, 효율성과 데이터 품질의 중요성을 입증.

### 2. Phi-1-base와 Phi-1 비교:

- Phi-1-base는 비정형적 문제에서 37%의 점수를 기록했으나, Phi-1은 미세조정을 통해 52%로 성능이 크게 향상.
- 이는 **CodeExercises** 데이터셋으로 미세조정된 효과를 강조함.

### 3. Phi-1-small의 놀라운 효율성:

- 350M 파라미터의 작은 모델임에도 불구하고, 새로운 문제에서 45%의 점수를 기록.

- 이 학습 데이터 품질이 모델 성능에 미치는 영향을 입증.

#### 4. 대규모 모델의 한계:

- CodeGen-Mono-16.1B(16.1B 파라미터)는 큰 모델 크기와 학습 데이터(577B 토큰)에도 불구하고 38%로 낮은 성능을 보임.
- Replit(2.7B 파라미터)도 Phi-1과 유사한 모델 크기지만 점수는 37%로 낮음.

### 3. Table 2가 보여주는 핵심 결과

- 데이터 품질의 중요성:** Phi-1은 소규모 모델과 작은 데이터셋으로도 대규모 모델을 능가하며, 데이터 품질의 중요성을 강조.
- 미세조정 효과:** Phi-1은 CodeExercises 데이터셋으로 미세조정된 결과 성능이 크게 향상됨.
- 비정형적 문제 해결 능력:** 새로운 문제에서 Phi-1이 가장 높은 성능을 보여, 단순히 학습된 데이터를 "기억"하는 것이 아니라, 문제를 이해하고 해결하는 능력을 입증.

이 표는 Phi-1 모델이 **효율적이고 데이터 품질 중심적인 접근 방식**을 통해 대규모 모델을 능가할 수 있음을 강력히 보여줍니다.

In [ ]:

## Phi-1.5: Textbooks Are All You Need II: phi-1.5 technical report

### 1. 발표 개요의 핵심 주제

- 핵심 주제:**
  - phi-1.5 모델:** 1.3억 파라미터 규모의 LLM으로, 고품질 "교과서 수준의 데이터"를 이용하여 훈련됨.
  - 이 모델은 기존 LLM의 학습 데이터로 사용된 웹 데이터 대신 합성 데이터를 활용하여, 대규모 모델과 유사한 성능을 달성.
  - 주요 성능 영역:
    - 상식적 추론 (common sense reasoning)
    - 언어 이해와 지식 (language understanding)
    - 다단계 추론 (multi-step reasoning)
  - 성능과 에너지 효율성 측면에서 더 작은 모델로도 대규모 모델에 가까운 결과를 낼 수 있다는 가능성을 보여줌.

### 2. 연구 배경

#### 1. 문제 정의:

- 대규모 언어 모델은 점점 더 많은 파라미터와 데이터를 요구하지만, 이는 경제적/환경적 비용 및 윤리적 문제를 유발함.
- 주요 질문: 대규모 데이터와 모델 크기가 정말 필수적인가?
  - 작은 모델로 비슷한 성능을 달성하는 것이 가능하다면, 접근성 및 지속 가능성을 개선할 수 있음.

#### 2. 이전 연구 흐름:

- TinyStories:** 1천만 파라미터로 영어 문장을 생성 가능한 모델.
- phi-1:** 1.3억 파라미터 모델로 파이썬 코딩 작업에서 유사한 수준의 성능 달성.

#### 3. phi-1.5의 기여:

- 상식적 추론과 같은 복잡한 작업에 초점을 맞춰, "교과서 수준" 합성 데이터를 사용하여 훈련.
- 기존의 웹 데이터 중심 접근법과는 다른 데이터 관리 및 훈련 전략 채택.
- 데이터 품질이 모델 성능에 미치는 영향을 강조하고, 데이터 품질이 모델 크기만큼 중요하다는 점을 입증.

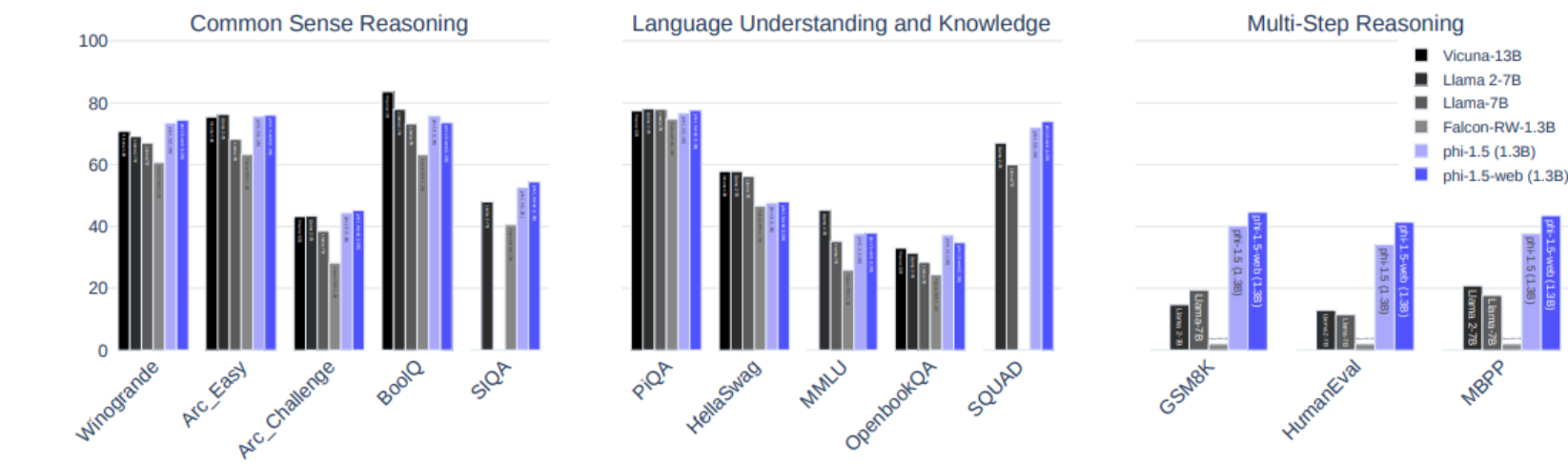
#### 4. 데이터 구성:

- 총 30억 토큰의 데이터 사용:
  - phi-1의 기존 데이터(7억 토큰).
  - 새로운 합성 데이터(20억 토큰): 상식, 과학, 일상 활동 등 다양한 주제를 포함.
  - 코드 데이터(3억 토큰): phi-1에서 사용된 필터링된 코드 데이터 재활용.

#### 5. 기술적 의의:

- 더 작은 모델이 더 큰 모델의 성능을 재현할 수 있는 가능성을 제시함으로써 AI 연구의 경제적 및 환경적 부담을 줄일 수 있음.
- phi-1.5는 독립 연구에 적합한 플랫폼을 제공하여, 모델 해석 가능성, 맥락 학습, 편향 및 환각 완화 전략에 대한 연구를 장려.





다음은 phi-1.5, 필터링된 웹 데이터를 추가한 phi-1.5-web, 그리고 다른 최신 공개 LLM들의 벤치마크 결과를 비교한 것입니다. 모델 크기는 phi-1.5의 13억 파라미터(Falcon-RW-1.3B)부터 10배 큰 Vicuna-13B(Llama-13B의 파인튜닝 버전)까지 다양합니다. 벤치마크는 크게 상식 추론, 언어 기술, 다단계 추론의 세 가지 범주로 분류됩니다. 이러한 분류는 느슨하게 적용되며, 예를 들어 HellaSwag는 상식 추론을 요구하지만, "기억된 지식"에 더 의존한다고 볼 수 있습니다. phi-1.5 모델들은 상식 추론과 언어 기술에서 다른 모델들과 비슷한 성능을 보이며, 다단계 추론에서는 다른 모델들을 크게 능가합니다. 참고로, 이 수치들은 모델 간의 일관성을 유지하기 위해 자체 평가 파이프라인에서 나온 것이므로, 다른 곳에서 보고된 수치와 약간 다를 수 있습니다.

모델명	파라미터 수	아키텍처 특징	학습 데이터	학습 비용 (GPU 시간)	HumanEval 성능	MBPP 성능	특징	기타 지표 및 정보
phi-1	1.3억	Transformer 기반, 24 레이어, 레이어당 32 헤드, 헤드당 64 차원, 회전 임베딩, 플래시 어텐션, CodeGen-모노 토크나이저 사용	교과서 수준의 웹 데이터 6억 토큰, GPT-3.5로 생성된 합성 데이터 1억 토큰, 파이썬 연습 문제 데이터셋 2억 토큰	770	50.6%	55.5%	파이썬 코딩 작업에 특화된 소형 언어 모델, GPT-3.5보다 우수한 성능을 보임	8개의 NVIDIA A100 GPU로 4일간 학습 진행, 고품질 데이터 사용으로 작은 모델로도 높은 성능 달성
phi-1-small	3.5천만	phi-1과 유사한 아키텍처를 가지며, 파라미터 수만 축소됨	phi-1과 동일한 데이터셋 사용	-	45%	-	phi-1의 축소판 모델로, 성능은 다소 낮지만 효율적인 학습 가능	-
phi-1.5	1.3억	phi-1과 동일한 아키텍처를 유지하며, 추가적으로 상식 추론과 언어 이해를 위한 데이터로 학습됨	phi-1의 데이터 7억 토큰 + 새로운 합성 데이터 20억 토큰 + 코드 데이터 3억 토큰	1,500	-	-	상식적 추론과 언어 이해 능력을 향상시킨 모델로, 5배 큰 모델과 유사한 성능을 발휘	phi-1과 동일한 아키텍처 구조를 유지하며, 고품질 합성 데이터를 추가하여 성능 향상
phi-1.5-web	1.3억	phi-1.5와 동일한 아키텍처를 가지며, 웹 데이터로 추가 학습됨	phi-1.5의 데이터 + 웹에서 수집한 88억 토큰의 데이터	3,000	-	-	웹 데이터를 추가 학습하여 성능을 향상시킨 모델	phi-1.5 대비 학습 데이터와 비용이 증가하였으며, 웹 데이터 활용으로 다양한 언어 이해 능력 향상

## Phi-2: The Surprising Power of Small Language Models

Phi-2는 Microsoft에서 개발한 27억 개 파라미터를 가진 소형 언어 모델(SLM)로, 고품질 합성 데이터와 웹 데이터를 활용하여 학습되었습니다. 이 모델은 상식 추론, 언어 이해, 수학, 코딩 등 다양한 분야에서 우수한 성능을 보이며, 특히 다단계 추론에서 두각을 나타냅니다.

다음은 Phi-2와 이전 모델들의 차이점과 특징을 비교한 표입니다:

모델명	파라미터 수	학습 데이터 및 방법	학습 비용 (GPU 시간)	주요 성능 및 특징
Phi-1	1.3억	교과서 수준의 웹 데이터 6억 토큰, 합성 데이터 1억 토큰, 파이썬 연습 문제 데이터셋 2억 토큰 사용.	770	파이썬 코딩 작업에 특화된 소형 언어 모델로, GPT-3.5보다 우수한 성능을 보임.
Phi-1.5	1.3억	Phi-1의 데이터 7억 토큰에 추가로 합성 데이터 20억 토큰과 코드 데이터 3억 토큰을 사용하여 학습.	1,500	상식적 추론과 언어 이해 능력을 향상시켜, 5배 큰 모델과 유사한 성능을 발휘.
Phi-2	27억	1.4조 토큰의 합성 및 웹 데이터셋을 여러 번 반복하여 학습.	14일간 96개의 A100 GPU 사용	상식, 언어 이해, 수학, 코딩 등 다양한 분야에서 우수한 성능을 보이며, 특히 다단계 추론에서 두각을 나타냄.

Phi-2는 이전 모델들에 비해 파라미터 수와 학습 데이터의 규모가 증가하였으며, 이를 통해 다양한 자연어 처리 및 코딩 작업에서 뛰어난 성능을 보여줍니다. 특히, 다단계 추론 분야에서의 성능 향상이 두드러집니다.

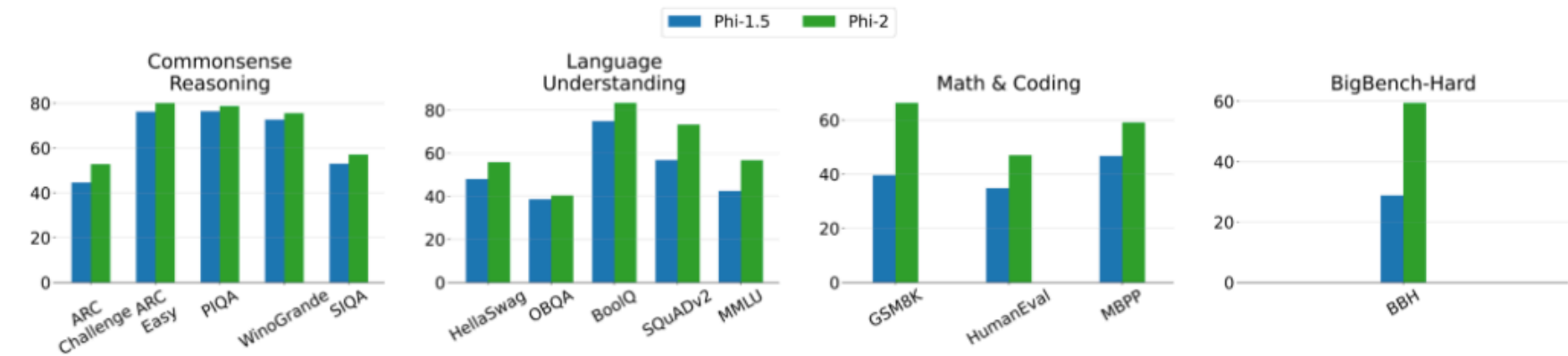


Figure 2. Comparison between Phi-2 (2.7B) and Phi-1.5 (1.3B) models. All tasks are evaluated in 0-shot except for BBH and MMLU which use 3-shot CoT and 5-shot, respectively.

핵심 질문:

- 언어 모델 크기를 수백억~수천억 파라미터로 확장하면 새로운 능력이 나타났지만, 이러한 능력이 소규모 모델에서도 데이터 선택 등의 전략을 통해 구현될 수 있는지 탐구.

Phi 모델 시리즈의 목표:

- 소형 언어 모델(SLM)이 대규모 모델에 근접한 성능을 발휘하도록 훈련.
- Phi-2는 기존 언어 모델 확장 법칙을 깨는 두 가지 핵심 전략을 통해 성능을 극대화.

핵심 전략:

- 데이터 품질 강화:
  - "교과서 수준"의 고품질 데이터에 집중.
  - 상식적 추론, 과학, 일상 활동, 마음 이론 등 일반 지식을 가르치기 위한 합성 데이터 사용.
  - 교육적 가치와 내용 품질을 기준으로 필터링된 웹 데이터 추가.
- 지식 확장 및 전이:
  - Phi-1.5(1.3억 파라미터)에서 학습한 지식을 27억 파라미터 모델 Phi-2에 전이.
  - 이 과정을 통해 학습 속도 가속 및 벤치마크 점수 상승을 확인.

Phi-2는 데이터 품질과 지식 전이를 통해 효율적 학습과 성능 향상을 달성하며, 소규모 모델에서도 대규모 모델에 근접한 성능 가능성을 보여줍니다.

In [ ]:

Phi-3

Phi-3 모델 시리즈의 종류와 각 모델의 주요 특징점

모델명	파라미터 수	주요 특징	활용 목적 및 비교
Phi-3-mini	3.8억	- GPT-3.5 수준의 성능 (MMLU 69%, MT-Bench 8.38) - 스마트폰에서 로컬 실행 가능 (iPhone 14에서 12 tokens/s 생성) - 데이터 최적화 접근법으로 학습	- 스마트폰 등 로컬 디바이스에서 작동 가능 - 대규모 모델 대비 적은 리소스 요구
Phi-3-small	7억	- 4.8조 토큰 학습 - MMLU 75%, MT-Bench 8.7 - GEGLU 활성화 및 BlockSparse Attention 도입	- 높은 성능의 중형 모델 - 다양한 멀티모달 작업 지원 가능
Phi-3-medium	14억	- 4.8조 토큰 학습 - MMLU 78%, MT-Bench 8.9 - Phi-3-small 대비 확장된 구조와 성능	- 고성능 요구 작업에 적합 - Phi-3 모델군 중 가장 성능이 뛰어난 단일 모델
Phi-3.5-mini	3.8억	- 멀티모달 입력 지원 (텍스트+이미지) - MMLU 멀티링구얼 점수 55.4 (다국어 능력 강화) - LongRope 기술로 128K 컨텍스트 길이 지원	- 멀티모달 작업과 긴 문맥 길이 처리에 적합 - 모델 크기 대비 고성능
Phi-3.5-MoE	16×3.8억 (활성: 6.6억)	- Mixture of Experts (MoE) 구조로 효율적 모델 활성화 - MMLU 멀티링구얼 점수 69.9 - GPT-4o-mini와 유사한 성능	- 효율성을 극대화한 고성능 멀티모달 모델 - 수학, 코딩, 논리적 추론 등 다양한 작업에서 뛰어난 성능
Phi-3.5-Vision	4.2억	- CLIP 기반 이미지 인코더와 Phi-3.5-mini 통합 - 텍스트+이미지 멀티모달 입력 지원 - AI2D, ScienceQA 등 벤치마크에서 경쟁력 있는 성능 발휘	- 시각적 정보와 텍스트를 동시에 처리 - 데이터 시각화 및 이해를 필요로 하는 작업에 적합

추가 설명

- Phi-3-mini는 로컬 실행을 염두에 둔 경량화된 모델로, 스마트폰에서 작동 가능한 점이 큰 강점입니다.
- Phi-3.5 시리즈는 다국어 및 멀티모달 작업에 중점을 둔 확장된 기능을 제공합니다.
- Phi-3.5-MoE는 Mixture of Experts 구조로 설계되어, 고성능을 발휘하면서도 효율적인 리소스 활용이 가능하도록 설계되었습니다.
- Phi-3.5-Vision은 텍스트와 이미지를 함께 처리하는 멀티모달 작업에 적합하며, 다양한 시각적 이해와 분석 작업에서 활용됩니다.

이 표는 PDF에 기술된 내용을 바탕으로 작성되었습니다.

Phi-3 모델에 대한 요약과 phi-3의 개발 목적 및 주요 특징은 다음과 같습니다:

## 요약

- Phi-3-mini**는 3.8억 개의 파라미터를 가진 소형 언어 모델로, 3.3조 개의 토큰으로 학습되었습니다.
- 경쟁력 있는 성능:** Phi-3-mini는 GPT-3.5와 같은 더 큰 모델과 비슷한 성능을 보여주며, 특히 MMLU 테스트에서 69%, MT-Bench에서 8.38의 점수를 기록했습니다.
- 로컬 사용 가능:** 이 모델은 작고 효율적이며, 스마트폰에서도 작동할 수 있도록 설계되었습니다. iPhone 14에서 12 tokens/s의 생성 속도를 기록하며 완전 오프라인으로 작동합니다.
- 데이터 중심 접근:** 학습 데이터의 품질을 개선함으로써 더 작은 모델로도 높은 성능을 달성했습니다. 웹 데이터와 LLM 기반 생성 데이터를 조합하여 학습 효율을 극대화했습니다.

## 개발 목적

- 소형화와 고효율성:** 고성능 언어 모델을 소형화하여 스마트폰 등 로컬 환경에서 작동 가능하도록 설계. 이는 데이터 최적화와 효율적인 아키텍처 설계를 통해 이루어졌습니다.
- 안전성과 견고성:** Phi-3-mini는 안전하고 신뢰할 수 있는 응답을 제공하기 위해 후처리 단계를 통해 조정되었습니다.
- 멀티모달 및 다국어 지원:** Phi-3.5 시리즈에서는 멀티모달 입력(이미지+텍스트)과 다국어 지원 능력을 강화했습니다.

## 주요 특징점

- 작은 크기와 고성능:**
  - 3.8억 파라미터로 GPT-3.5와 비슷한 수준의 성능을 제공.
  - 스마트폰에서 실행 가능하며, 완전 오프라인 작동 지원.
- 데이터 최적화 학습:**
  - 데이터 필터링 및 생성 데이터 사용을 통해 모델 크기에 비해 높은 성능 달성.
  - "데이터 최적화" 방식으로 논리적 추론 및 언어 이해 능력 강화.
- 안전한 사용:**
  - 학습 후 안전성 조정 및 지속적인 피드백 반영.
- 멀티모달 기능:**
  - Phi-3.5-Vision은 이미지와 텍스트를 혼합하여 처리할 수 있는 모델로, 다양한 시각적 추론 작업에서도 성능을 발휘.

이

Phi 시리즈 모델들은 각 버전마다 파라미터 수, 학습 데이터 양, 성능, 그리고 활용 목적에서 차별화된 특징을 보입니다. 아래 표는 Phi-1부터 Phi-3까지의 주요 특징점을 비교한 것입니다:

모델명	파라미터 수	학습 데이터 양	주요 성능 및 특징	활용 목적 및 비교
Phi-1	1.3억	60억 토큰	- HumanEval 테스트에서 GPT-3.5보다 우수한 성능 - 파이썬 코딩에 특화된 모델	- 고품질의 파이썬 코드 생성을 위해 개발 - 8개의 NVIDIA A100 GPU로 4일간 훈련
Phi-1.5	1.3억	-	- 일반적인 추론과 언어 이해 능력 향상 - 5배 큰 모델과 동등한 성능 발휘	- 상식 추론 및 언어 이해 능력 강화 - Phi-1의 확장 버전
Phi-2	2.7억	1.4조 토큰	- 논리적 추론과 안전성이 획기적으로 향상 - Mistral 7B, LLaMA-2 13B 모델을 능가하는 성능	- 언어 이해, 추론, 수학, 코딩 과제 등 다양한 벤치마크에서 우수한 성능 - 온디바이스 AI 구현 가능
Phi-3	3.8억	3.3조 토큰	- GPT-3.5와 유사한 성능 (MMLU 69%, MT-Bench 8.38 점) - 스마트폰에서 로컬 실행 가능 (iPhone 14에서 초당 12토큰 생성)	- 데이터 최적화 접근법으로 작은 모델에서도 높은 성능 달성 - 멀티모달 및 다국어 지원 강화

이러한 비교를 통해 Phi 시리즈 모델들이 점진적으로 성능을 향상시키면서도 모델 크기를 효율적으로 유지하고 있음을 알 수 있습니다. 특히, Phi-3는 소형 모델임에도 불구하고 고성능을 발휘하며, 로컬 환경에서의 활용성을 높인 점이 두드러집니다. 모델은 특히 작고 효율적인 언어 모델이 필요하거나 로컬 환경에서 신뢰성과 안전성을 요구하는 응용 프로그램에 적합합니다.

## Phi-3.5-vision

Phi-3.5-Vision의 특징과 모델 구조는 다음과 같습니다:

### Phi-3.5-Vision의 주요 특징

- 멀티모달 입력 지원:**
  - 텍스트와 이미지를 혼합하여 처리 가능.
  - 단일 이미지뿐 아니라 다중 이미지 입력과 텍스트를 함께 처리할 수 있는 능력.
  - 학습 데이터에 차트, 표, 다이어그램, 파워포인트 등의 시각적 콘텐츠 포함.
- 고성능 시각적 이해:**



- ScienceQA, AI2D, ChartQA 등 다양한 시각적 및 언어적 추론 벤치마크에서 경쟁력 있는 성능.
- 텍스트 생성 및 답변 생성에서 이미지의 의미와 문맥을 통합.

### 3. 효율적 시각 정보 처리:

- 동적 크롭핑(Dynamic Cropping) 전략을 사용하여 고해상도 이미지와 다양한 비율을 효율적으로 처리.
- 1344x1344 해상도까지 지원, 대부분의 학습 이미지를 커버.

### 4. 데이터 구성:

- 텍스트와 이미지를 혼합한 대규모 데이터셋 (예: FLD-5B).
- PDF의 Optical Character Recognition(OCR) 데이터를 활용한 합성 데이터.

### 5. 훈련 목표:

- 텍스트 토큰 예측을 중심으로 진행하며, 이미지 토큰은 손실 계산에서 제외.
- 텍스트 및 멀티모달 작업을 동시에 학습해 언어 및 멀티모달 추론 능력을 최적화.

## Phi-3.5-Vision의 구조

### 1. 구성 요소:

- 이미지 인코더: CLIP ViT-L/14를 기반으로 시각적 토큰을 추출.
- 텍스트 디코더: Phi-3.5-mini 모델을 사용하여 텍스트 토큰과 시각적 토큰을 결합해 처리.

### 2. 토큰 처리 방식:

- 시각적 토큰과 텍스트 토큰을 인터리브(interleaved) 방식으로 결합.
- 이미지 입력을 2D 블록으로 나누어 토큰화하여 통합.

### 3. 크기와 성능:

- 4.2억 개의 파라미터로 구성.
- 멀티모달 작업 시 높은 효율성과 정확성을 보여줌.

### 4. 학습 프로세스:

- Pre-training:** 시각적 및 텍스트 데이터를 예측 목표로 학습.
- Post-training:** Supervised Fine-Tuning(SFT)와 Direct Preference Optimization(DPO) 단계를 통해 모델 정밀도를 개선.

### 5. 적용 사례:

- 단일 이미지와 텍스트를 결합하여 분석하는 작업.
- 여러 이미지 및 텍스트를 처리하는 복잡한 시각적 추론.

Phi-3.5-Vision은 텍스트 중심의 언어 모델에서 확장되어 시각적 콘텐츠와의 결합을 지원하는 멀티모달 AI 모델로, 특히 데이터 시각화 및 분석 작업에서 뛰어난 효율성과 성능을 보여줍니다.

model 구조:

```
Phi3VForCausalLM(  
  (model): Phi3VModel(  
    (embed_tokens): Embedding(32064, 3072, padding_idx=32000)  
    (embed_dropout): Dropout(p=0.0, inplace=False)  
    (vision_embed_tokens): Phi3ImageEmbedding(  
      (drop): Dropout(p=0.0, inplace=False)  
      (wte): Embedding(32064, 3072, padding_idx=32000)  
      (img_processor): CLIPVisionModel(  
        (vision_model): CLIPVisionTransformer(  
          (embeddings): CLIPVisionEmbeddings(  
            (patch_embedding): Conv2d(3, 1024, kernel_size=(14, 14), stride=(14, 14), bias=False)  
            (position_embedding): Embedding(577, 1024)  
          )  
        )  
        (pre_layrnorm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)  
        (encoder): CLIPEncoder(  
          (layers): ModuleList(  
            (0-23): 24 x CLIPEncoderLayer(  
              (layer_norm1): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)  
              (mlp): CLIPMLP(  
                (activation_fn): QuickGELUActivation()  
                (fc1): Linear(in_features=1024, out_features=4096, bias=True)  
                (fc2): Linear(in_features=4096, out_features=1024, bias=True)  
              )  
              (layer_norm2): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)  
              (self_attn): CLIPAttentionFA2(  
                (k_proj): Linear(in_features=1024, out_features=1024, bias=True)  
                (v_proj): Linear(in_features=1024, out_features=1024, bias=True)  
                (q_proj): Linear(in_features=1024, out_features=1024, bias=True)  
                (out_proj): Linear(in_features=1024, out_features=1024, bias=True)  
              )  
            )  
          )  
        )  
      )  
    )  
  )  
)
```

```
)
(post_layernorm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
)
)
(img_projection): Sequential(
  (0): Linear(in_features=4096, out_features=3072, bias=True)
  (1): GELU(approximate='none')
  (2): Linear(in_features=3072, out_features=3072, bias=True)
)
)
(layers): ModuleList(
  (0-31): 32 x Phi3DecoderLayer(
    (self_attn): Phi3FlashAttention2(
      (o_proj): Linear(in_features=3072, out_features=3072, bias=False)
      (qkv_proj): Linear(in_features=3072, out_features=9216, bias=False)
      (rotary_emb): Phi3SuScaledRotaryEmbedding()
    )
    (mlp): Phi3MLP(
      (gate_up_proj): Linear(in_features=3072, out_features=16384, bias=False)
      (down_proj): Linear(in_features=8192, out_features=3072, bias=False)
      (activation_fn): SiLU()
    )
    (input_layernorm): Phi3RMSNorm()
    (resid_attn_dropout): Dropout(p=0.0, inplace=False)
    (resid_mlp_dropout): Dropout(p=0.0, inplace=False)
    (post_attention_layernorm): Phi3RMSNorm()
  )
)
)
(norm): Phi3RMSNorm()
)
(lm_head): Linear(in_features=3072, out_features=32064, bias=False)
)
```



extract all displayed texts and keep original displayed format. For the left section, please use markdown table with column "coffee", "small", "medium" and "large". Instead, for the right section, please use JSON format.

Certainly! Below is the markdown table converted from the image:

Coffee	Small	Medium	Large
Espresso	1.30	1.60	2.00
Americana	1.40	1.75	2.10
Cappuccino	2.25	2.40	3.25
Latte	2.35	3.00	3.25
Mocha	2.35	3.00	3.50
Chai	2.25	2.40	3.25
Chai Bomb	2.75	3.25	3.75
House Cafe	1.50	1.75	2.00
Cafe au Lait	1.75	2.50	3.25

And here is the JSON representation of the teas offered:

```
{
  "Teas": {
    "CUP": {
      "GREEN": "$1.50",
      "BLACK": "$1.50",
      "WHITE": "$1.50",
      "OOLONG": "$1.50",
      "HERBAL": "$1.50"
    },
    "POT": {
      "GREEN": "$4.00",
      "BLACK": "$4.00",
      "WHITE": "$4.00",
      "OOLONG": "$4.00",
      "HERBAL": "$4.00"
    }
  },
  "**ADD .50 FOR HONEY**",
  "**ADD $1.00 FOR ICE**"
}
```

What is the price of a Cappuccino with large size?

The price for a Cappuccino in large size is \$3.25.

How much does it cost to add ice to a tea?

It costs an additional \$1.00 to add ice to any tea.

If someone wants to buy a pot of tea, how much would it cost?

A pot of tea would cost \$4.00.

Figure 7: The demo case shows Phi-3.5-Vision’s capability in natural image understanding and reasoning.



# Man's Best Friend: Dogs

A Fascinating Journey through  
History, Breeds, and Care

## Domestication of Dogs

### Theories of Dog Domestication

There are several theories about how dogs were domesticated and became man's best friend, including the hunting hypothesis, scavenging hypothesis, and social theory.

### Working Dogs

Dogs have been bred to perform various jobs and tasks, from hunting and herding to search and rescue and law enforcement.

### Dog Breeds

There are over 300 dog breeds, each with unique physical and behavioral characteristics, bred for specific purposes and tasks.



## Working Dogs

### Hunting Dogs

Hunting dogs have been used for centuries to help humans track and retrieve game animals. Different breeds excel at different types of hunting, such as pointing, flushing, and retrieving.

### Herding Dogs

Herding dogs are trained to help humans manage livestock. They have a natural instinct to herd, and different breeds excel at different types of livestock, such as sheep, cattle, and pigs.

### Protection Dogs

Protection dogs are trained to protect humans and property. They are often used by law enforcement and military personnel, as well as by private individuals for personal protection.



## Herding Dogs

### Border Collies

Border Collies are known for their intelligence and agility, making them excellent herding dogs. They are highly trainable and have a strong work ethic.

### Australian Shepherds

Australian Shepherds, also known as Aussies, are versatile herding dogs that excel in a variety of tasks, including herding, agility, and obedience. They are loyal and hardworking dogs.



## Nutrition

### Essential Nutrients

Dogs require a balanced diet that contains the essential nutrients, including protein, carbohydrates, fats, vitamins, and minerals, to maintain a healthy body and mind.

### Best Types of Food

There are many different types of dog food available, including dry kibble, wet canned food, and homemade meals. We will discuss the best types of food for your dog and how to choose the right one.

### Portion Size

Feeding your dog the right amount of food is just as important as choosing the right type of food. We will discuss how to determine the appropriate portion size for your dog based on their age, weight, and activity level.

## Exercise

### Best types of exercise for your dog

Different dog breeds require different types of exercise. We will discuss the best types of exercise for your dog and how to keep them engaged and stimulated while exercising.

### How much exercise does your dog need?

The amount of exercise your dog needs depends on their breed, age, and overall health. We will discuss the recommended amount of exercise for different breeds and how to determine if your dog is getting enough exercise.



Summarize the deck of slides.

To summarize, the slides are composed of these sections:

- **Introduction:**  
The presentation begins by introducing the topic of dogs as man's best friend, highlighting their fascinating journey through history, breeds, and care.
- **Domestication of Dogs:**  
The presentation discusses theories of dog domestication, including the hunting hypothesis and social theory. It also mentions the various jobs and tasks dogs have been bred to perform, such as hunting, herding, and law enforcement.
- **Working Dogs:**  
The presentation delves into the specific roles of different types of working dogs, including hunting dogs, herding dogs, and protection dogs. It provides examples of each breed and their specific tasks.
- **Nutrition:**  
The presentation emphasizes the importance of a balanced diet for dogs, including essential nutrients and the best types of food. It also discusses the importance of portion size and how to determine the appropriate amount of food for a dog.
- **Exercise:**  
The presentation concludes by discussing the importance of exercise for dogs, the different types of exercise suitable for different breeds, and how to determine the appropriate amount of exercise for a dog based on their breed, age, and overall health.

Phi-3.5-vision은 고객의 귀중한 피드백을 바탕으로 멀티 프레임 이미지 이해와 추론을 위한 최첨단 기능을 도입했습니다.

이 혁신은 상세한 이미지 비교, 멀티 이미지 요약/스토리텔링, 비디오 요약 등의 기능을 제공하며, 다양한 상황에서 폭넓게 활용될 수 있는 가능성을 열어줍니다.

예를 들어, 여러 슬라이드의 요약 작업에 대한 모델 출력 결과를 확인할 수 있습니다.

특히, Phi-3.5-vision은 단일 이미지 벤치마크에서 상당한 성능 향상을 보여주었습니다.

예를 들어, MMMU 성능이 40.4에서 43.0으로, MMBench 성능이 80.5에서 81.9로 향상되었습니다.

또한, 문서 이해 벤치마크인 TextVQA에서도 70.9에서 72.0으로 성능이 개선되었습니다.

다음 표는 잘 알려진 멀티 이미지/비디오 벤치마크에서의 세부 비교 결과를 보여주며, 개선된 성능 지표를 확인할 수 있습니다. 다만, Phi-3.5-vision은 다국어 활용 사례에 최적화되지 않았으므로, 추가 미세 조정 없이 다국어 환경에서 사용하는 것은 권장되지 않습니다.

BLINK Benchmark	Phi-3.5-vision-instruct	LlaVA-Interleave Qwen 7B	InternVL 2 - 4B	InternVL 2 -8B	Gemini 1.5 Flash	GPT 4o mini	Claude 3.5 Sonnet	Gemini 1.5 Pro	GPT 4o
Art Style	87.2	62.4	55.6	52.1	64.1	70.1	59.8	70.9	73.3
Counting	54.2	56.7	54.2	66.7	51.7	55.0	59.2	65.0	65.0
Forensic Detection	92.4	31.1	40.9	34.1	54.5	38.6	67.4	60.6	75.8
Functional Correspondence	29.2	34.6	24.6	24.6	33.1	26.9	33.8	31.5	43.8
IQ Test	25.3	26.7	26.0	30.7	25.3	29.3	26.0	34.0	19.3
Jigsaw	68.0	86.0	55.3	52.7	71.3	72.7	57.3	68.0	67.3
Multi-View Reasoning	54.1	44.4	48.9	42.9	48.9	48.1	55.6	49.6	46.6
Object Localization	49.2	54.9	53.3	54.1	44.3	57.4	62.3	65.6	68.0
Relative Depth	69.4	77.4	63.7	67.7	57.3	58.1	71.8	76.6	71.0
Relative Reflectance	37.3	34.3	32.8	38.8	32.8	27.6	36.6	38.8	40.3
Semantic Correspondence	36.7	31.7	31.7	22.3	32.4	31.7	45.3	48.9	54.0
Spatial Relation	65.7	75.5	78.3	78.3	55.9	81.1	60.1	79.0	84.6
Visual Correspondence	53.5	40.7	34.9	33.1	29.7	52.9	72.1	81.4	86.0
Visual Similarity	83.0	91.9	48.1	45.2	47.4	77.8	84.4	81.5	88.1
Overall	57.0	53.1	45.9	45.4	45.881	51.9	56.5	61.0	63.2

Table 9: Phi-3.5-vision Tasks Benchmark

VideoMME Benchmark	Phi-3.5-vision-instruct	LlaVA-Interleave Qwen 7B	Intern VL 2 - 4B	Intern VL 2 - 8B	Gemini 1.5 Flash	GPT 4o mini	Claude 3.5 Sonnet	Gemini 1.5 Pro	GPT 4o
Short (< 2min)	60.8	62.3	60.7	61.7	72.2	70.1	66.3	73.3	77.7
Medium (4 – 15 min)	47.7	47.1	46.4	49.6	62.7	59.6	54.7	61.2	68.0
Long (30 – 60 min)	43.8	41.2	42.6	46.6	52.1	53.9	46.6	53.2	59.6
Overall	50.8	50.2	49.9	52.6	62.3	61.2	55.9	62.6	68.4

Table 10: Phi-3.5-vision VideoMME Benchmark

	Phi-3.5-Vision 4.2b	MM1-3B-Chat 3.6b [MGF <sup>+</sup> 24]	MM1-7B-Chat 7.6b [MGF <sup>+</sup> 24]	LLaVA-1.6 Vicuna-7b [LLLL23]	LLaVA-Next LLama3-8b [LLL <sup>+</sup> 24]	Qwen-VL-Chat 9.6b [BBY <sup>+</sup> 23]	Claude 3 haiku [Ant24]	Gemini 1.0 Pro V [TAB <sup>+</sup> 23]	GPT-4O 2024-05-13
MMMU (val) [YNZ <sup>+</sup> 23]	43.0	33.9	37.0	34.2	36.4	39.0	40.7	42.0	61.8
ScienceQA (test) [LMX <sup>+</sup> 22]	91.3	69.4	72.6	70.6	73.7	67.2	72.0	79.7	88.5
MathVista (testmini) [LBX <sup>+</sup> 24]	43.9	32.0	35.9	31.5	34.8	29.4	33.2	35.0	54.4
Inter-GPS (test) [LGJ <sup>+</sup> 21]	36.3	-	-	20.5	24.6	22.3	32.1	28.6	46.9
MMBench (dev-en) [LDZ <sup>+</sup> 24]	81.9	75.9	79.0	76.3	79.4	75.8	62.4	80.0	88.4
POPE (test) [LDZ <sup>+</sup> 23]	86.1	87.4	86.6	87.2	87.0	82.6	74.4	84.2	87.0
AI2D (test) [KSK <sup>+</sup> 16]	78.1	-	-	63.1	66.9	59.8	60.3	62.8	82.8
ChartQA (test) [MLT <sup>+</sup> 22]	81.8	-	-	55.0	65.8	50.9	59.3	58.0	64.0
TextVQA (test) [SNS <sup>+</sup> 19]	72.0	71.9	72.8	64.6	55.7	59.4	62.7	64.7	75.6

Table 5. 공용 MLLM 벤치마크에 대한 비교 결과 모든 보고된 수치는 동일한 파이프라인을 사용하여 비교 가능하도록 생성되었습니다. 단, MM1-3B-Chat [MGF+24] 및 MM1-7B-Chat [MGF+24] 모델은 공개되지 않았으므로 해당 모델의 수치는 포함되지 않았습니다. 우리는 Llava-1.5 [LLLL23]에서 사용된 평가 설정을 채택하였으며, 모든 결과에 대해 특정 프롬프트나 이미지 전처리를 사용하지 않았습니다. 이로 인해 보고된 수치는 다른 발표된 결과와 약간의 차이가 있을 수 있습니다. 이는 프롬프트의 차이 때문입니다.

Table 5는 Phi-3.5-Vision 모델이 공용 멀티모달 언어 모델(MLLM) 벤치마크에서 보여준 성능을 다른 모델들과 비교한 결과를 제공합니다.

이 표는 다양한 공개 데이터셋을 기반으로 각 모델의 시각적 이해 및 추론 능력을 평가한 결과를 포함하고 있습니다.

### Table 5의 주요 구성 요소

1. 벤치마크 종류:

- **MMMU (val)**: 멀티모달 모델의 전반적인 이해 및 처리 능력을 평가.
- **ScienceQA (test)**: 과학 관련 질문에 대한 모델의 시각적 및 언어적 추론 능력을 평가.
- **MathVista (test-mini)**: 수학적 문제 해결 능력을 평가.
- **Inter-GPS (test)**: 기하학 문제의 추론 및 해석 능력을 평가.
- **MMBench (dev-en)**: 멀티모달 작업의 전반적인 성능을 테스트.
- **POPE (test)**: 차트와 표의 정보를 분석하고 해석하는 능력을 평가.
- **AI2D (test)**: 다이어그램 기반 문제의 분석 및 이해 능력을 평가.
- **ChartQA (test)**: 차트 데이터를 기반으로 질문에 답변하는 능력을 평가.
- **TextVQA (test)**: 텍스트와 이미지가 포함된 데이터를 기반으로 질문에 답변하는 능력을 평가.

2. 비교 대상 모델:

- **MM1-3B-Chat** 및 **MM1-7B-Chat**: 3.6억 및 7.6억 파라미터로 구성된 모델.
- **Llava-1.6** 및 **Llava-Next (Llama 3-8B 기반)**: 7억 및 8억 파라미터로 구성된 모델.
- **Qwen-VL-Chat**: 멀티모달 이해를 목표로 설계된 9.6억 파라미터 모델.
- **Claude 3 Haiku** 및 **GPT-4O**: 상업용 모델로 멀티모달 작업에 강점을 보이는 최신 모델.

3. 평가 설정:

- 동일한 평가 파이프라인을 사용하여 비교 가능한 수치 제공.
- **Llava-1.5**에서 사용된 평가 방식 채택.
- 프롬프트 최적화나 이미지 전처리를 하지 않고 테스트 진행.

### Table 5의 주요 결과

- **Phi-3.5-Vision 성능:**
  - 여러 벤치마크에서 비교 대상 모델과 동등하거나 우수한 성능을 발휘.



- **ScienceQA**: GPT-4O와 비슷한 성능(91.3%).
- **MMBench** 및 **POPE**: 상위권 성능을 유지하며, 특정 과제에서는 GPT-4O에 근접.
- **TextVQA**: 72.0%로 Claude 3 Haiku보다 높은 성능을 기록.
- 다른 모델과의 비교:
  - **Qwen-VL-Chat** 및 **Claude 3 Haiku**와 비슷한 성능을 보여줌.
  - **GPT-4O**와 비교 시 전반적으로 약간 뒤처지지만, 일부 벤치마크에서는 경쟁력 있는 성과를 달성.

## 요약

Table 5는 **Phi-3.5-Vision**이 다른 멀티모달 언어 모델들과 비교했을 때 뛰어난 시각적 추론 및 언어적 이해 능력을 갖췄음을 보여줍니다.

특히, 일부 벤치마크에서 상업용 모델에 필적하는 성과를 발휘하며, 효율적인 멀티모달 모델로 자리 잡고 있음을 강조합니다.

	Phi-3.5-Vision 4.2b	Llava-interleave Qwen 7b [LZZ <sup>+</sup> 24]	InternVL2 4b [CWT <sup>+</sup> 24]	InternVL2 8b [CWT <sup>+</sup> 24]	Gemini 1.5 Flash [TAB <sup>+</sup> 23]	GPT4O mini 2024-07-18	Claude 3.5 Sonnet [Ant24]	Gemini 1.5 Pro [TAB <sup>+</sup> 23]	GPT-4O 2024-05-13
BLINK (val) [FHL <sup>+</sup> 24]	57.0	53.1	45.9	45.4	45.8	51.9	56.5	61.0	63.2
VideoMME (test) [FDL <sup>+</sup> 24]	50.8	50.2	49.9	52.6	62.3	61.2	55.9	62.6	68.4

Table 6: Comparison results on public multi-image/video MLLM benchmarks. All the reported numbers are produced with the exact same pipeline to ensure that the numbers are comparable.

	Phi-3.5-Vision 3.8b+0.3b	Phi-3.5-Vision w/o safety 3.8b+0.3b	Llava-1.6 Vicuna 7b+0.3b	Qwen-VL-Chat 7.7b+1.9b	GPT4-V N/A
Internal (private)	8.16	7.06	5.44	7.27	8.55
RTVLM (public)	5.44	3.56	3.86	4.78	6.81
VLGuard (public)	9.10	4.66	5.62	8.33	8.90

Table 7: Comparison results on public and private multi-modal RAI benchmarks. Note that all metrics in the table are [0,10] and a higher value indicates a better performance.

Table 6와 Table 7은 각각 **Phi-3.5-Vision** 모델의 멀티 이미지 및 비디오 벤치마크 성과와, 멀티모달 모델의 **Responsible AI (RAI)** 성능을 비교한 결과를 보여줍니다.

## Table 6: 멀티 이미지 및 비디오 MLLM 벤치마크 성능

### 목적

- 멀티 이미지 입력과 비디오 이해를 다룰 수 있는 모델의 성능을 평가.

### 벤치마크 종류

- BLINK (val)**:
  - 멀티 이미지를 기반으로 한 분석과 추론 성능 평가.
  - 예: 예술 스타일 인식, 법의학적 이미지 분석.
- VideoMME (test)**:
  - 비디오 프레임을 입력으로 사용하여 분석 및 추론 성능 평가.
  - 예: 영상 요약, 시각적 장면 이해.

### Phi-3.5-Vision의 성능

- BLINK**: 57.0%로, Claude 3.5 Sonnet 및 Gemini 1.5 Flash와 비슷한 성능을 발휘.
- VideoMME**: 50.8%로, Gemini 1.5 Flash(62.3%)보다 약간 낮지만, 여전히 경쟁력 있는 결과를 기록.

### 다른 모델과의 비교

- GPT-4O**: BLINK와 VideoMME 모두에서 가장 높은 성능을 보임.
- Qwen-VL-Chat** 및 **InternVL**: 일부 작업에서 Phi-3.5-Vision과 비슷한 수준의 성능을 보임.

Table 7: 멀티모달 모델의 RAI (Responsible AI) 성능

목적

- 멀티모달 모델의 안전성, 신뢰성, 유해 콘텐츠 방지를 평가.

벤치마크 종류

- Internal (private):**
  - Microsoft 내부 벤치마크를 사용하여 다각도로 모델의 안전성을 평가.
- RTVLM (public):**
  - 공공 데이터셋을 기반으로 모델의 책임감 있는 AI 성능 평가.
- VLGuard (public):**
  - 유해 콘텐츠 생성 방지 및 안전성 강화를 평가.

Phi-3.5-Vision의 성능

- Internal:** 8.16점으로, GPT-4V(8.55)에 근접한 성능을 발휘.
- RTVLM:** 5.44점으로, Qwen-VL-Chat(4.78)보다 우수한 성능.
- VLGuard:** 9.10점으로, GPT-4V(8.90)를 능가하며 최고 성능을 기록.

Phi-3.5-Vision의 개선점

- Safety Post-Training:**
  - 추가적인 안전성 데이터로 학습하여 유해 콘텐츠 생성 확률을 낮춤.
  - VLGuard와 Internal 벤치마크에서 안전성 성능이 눈에 띄게 향상.

다른 모델과의 비교

- GPT-4V:** 전반적으로 가장 높은 점수를 기록하지만, VLGuard에서는 Phi-3.5-Vision이 우위를 보임.
- Llava-1.6 및 Qwen-VL-Chat:** 전반적으로 Phi-3.5-Vision보다 낮은 성능.

요약

- Table 6**은 멀티 이미지 및 비디오 처리에서 **Phi-3.5-Vision**의 높은 경쟁력을 보여줍니다. 특히 BLINK와 VideoMME 벤치마크에서 Claude 및 Gemini 모델과 유사한 성과를 보였습니다.
- Table 7**은 안전성과 책임 있는 AI 구현에서 **Phi-3.5-Vision**의 강점을 나타냅니다. 특히 VLGuard 벤치마크에서 GPT-4V를 넘어서는 성능을 보여줌으로써, 안전한 멀티모달 AI로의 가능성을 입증했습니다.

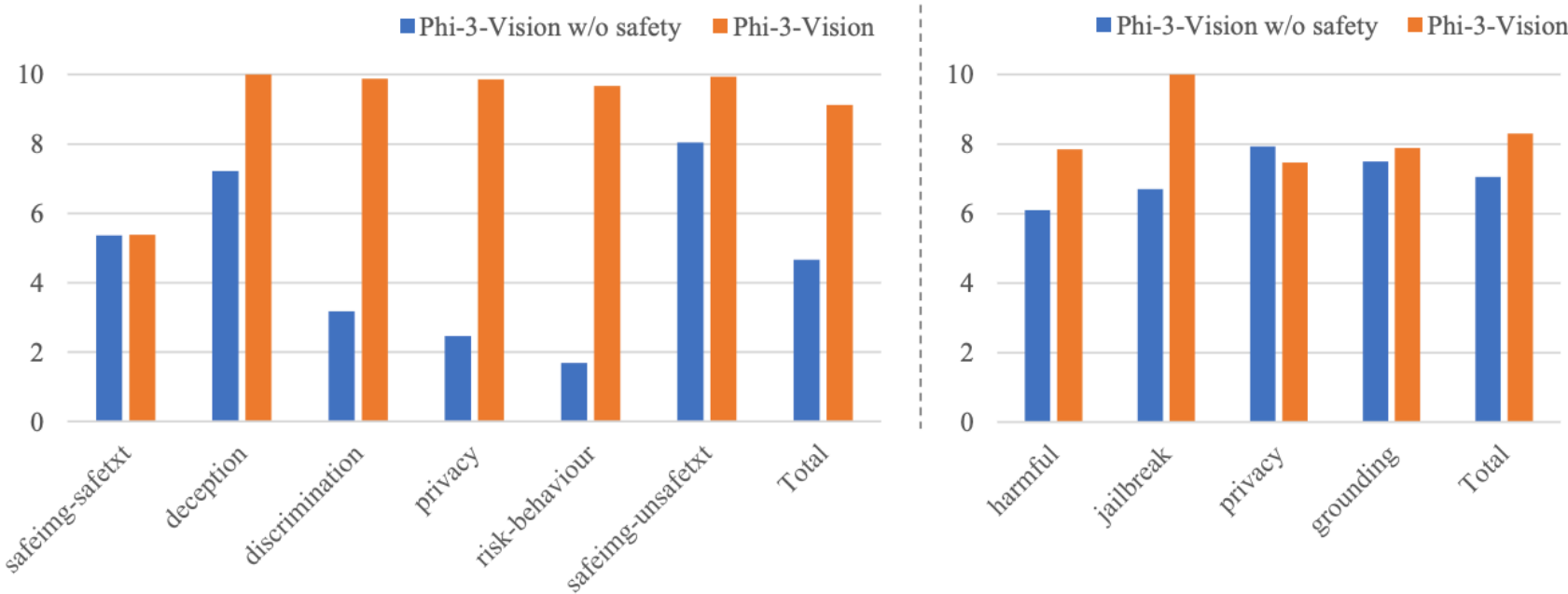


Figure 8: Comparison of categorized RAI performance of Phi-3.5-Vision with and without the safety post-training on the VLGuard (left) and Internal (right) benchmark, respectively. It clearly indicates that safety post-training can enhance the RAI performance across nearly all the RAI categories.

In [ ]:

## phi-4 Technical Report

phi-4 모델은 데이터 품질을 중시하여 개발된 140억 개 파라미터의 언어 모델로, 학습 데이터셋과 학습 과정에서 합성 데이터의 활용이 두드러집니다.

학습 데이터셋 구성:

데이터 소스	전체 학습 데이터에서의 비중	고유 토큰 수	에포크 수
웹 데이터	15%	1.3조	1.2

데이터 소스	전체 학습 데이터에서의 비중	고유 토큰 수	에포크 수
웹 리라이트	15%	2,900억	5.2
합성 데이터	40%	2,900억	13.8
코드 데이터	20%	8,200억	2.4
타겟 소스 (학술 자료 등)	10%	5,800억	1.7

학습 과정:

1. 사전 학습 (Pretraining):

- 합성 데이터 활용:** 다양한 기법을 통해 생성된 고품질의 합성 데이터를 주로 사용하여 모델의 추론 및 문제 해결 능력을 향상시켰습니다.
- 웹 데이터 필터링:** 웹에서 수집한 데이터를 엄격하게 선별하여 고품질의 지식과 추론 콘텐츠만을 포함시켰습니다.

2. 미드트레이닝 (Midtraining):

- 컨텍스트 길이 확장:** 모델의 컨텍스트 길이를 4K에서 16K로 늘려, 긴 문맥을 처리할 수 있도록 학습하였습니다.
- 장문 데이터 활용:** 긴 컨텍스트를 포함하는 데이터(예: 학술 자료, 도서, 코드)를 선별하여 학습에 반영하였습니다.

3. 포스트 트레이닝 (Post-Training):

- 거부 샘플링 및 직접 선호 최적화(DPO):** 모델의 출력을 세밀하게 조정하기 위해 거부 샘플링과 새로운 DPO 기법을 도입하였습니다.
- 지도 미세 조정(SFT):** 특정 작업에 대한 모델의 성능을 높이기 위해 정제된 지도 학습 데이터셋을 활용하였습니다.

이러한 데이터 구성과 학습 과정을 통해, phi-4 모델은 특히 추론 중심의 작업에서 우수한 성능을 발휘하며, 이전 모델보다 향상된 결과를 보여줍니다.

1. 합성 데이터 활용:

- phi-4는 고품질의 시드 데이터와 합성 데이터 처리를 통해, 수학 및 STEM 평가에서 대형 모델과 유사한 성능을 보입니다.
- 합성 데이터는 구조화되어 있어, 모델이 논리적 패턴을 학습하는 데 유리합니다.

2. 데이터 수집 및 처리:

- 웹 페이지, 도서, 학술 논문 등에서 시드 데이터를 추출하여, 복잡한 추론에 중점을 둔 고품질 데이터를 생성합니다.
- 이러한 시드 데이터를 기반으로 다양한 합성 데이터를 생성하여 모델 학습에 활용합니다.

3. 직접 선호 최적화(DPO):

- phi-4는 사용자 선호도를 직접적으로 모델에 반영하는 DPO 기법을 사용하여, 모델의 출력을 조정합니다.
- DPO는 별도의 보상 모델 없이 사용자 피드백을 통해 모델을 미세 조정하는 방법으로, 효율성과 안정성을 높입니다.

4. Pivotal Token Search (PTS):

- 모델의 출력을 개선하기 위해 PTS 기법을 도입하여, 중요한 토큰을 식별하고 이에 집중하여 학습합니다.

5. Judge-Guided DPO:

- 판단자(Judge)의 피드백을 기반으로 모델을 미세 조정하여, 사용자 요구에 더욱 부합하는 출력을 생성합니다.

이러한 방법들을 통해 phi-4는 효율적인 데이터 활용과 모델 최적화를 이루어냈으며, 특히 추론 능력에서 우수한 성과를 보였습니다.

자세한 내용과 코드 예시는 원문을 참고하시기 바랍니다.

In [ ]:

In [ ]: