

MathQA 성능 향상을 위한 파인튜닝 실험 보고서

실험 목표

MathQA 평가 데이터셋에서 모델 성능을 향상시키는 효과적인 파인튜닝 방법을 탐색하는 것. 특히 학습 데이터의 도메인, 문제 형식, 손실 함수가 성능에 미치는 영향을 검증함.

1. 실험 배경

1.1 MathQA 데이터셋 특성

구분	내용
형식	5지선다 (a, b, c, d, e)
옵션 구조	"a) 24 , b) 120 , c) 625 , d) 720 , e) 1024" 형태의 단일 문자열
정답 형식	"c" 같은 레이블 문자

장점

- 규모가 크고 다양하여 의미 있는 비교가 가능
- 정답 파싱 오류 발생 확률이 낮음

단점

- 선다형 특성상 "추론 능력"보다 "시험 잘 보기(heuristics)"를 측정할 수 있음
- CoT reasoning이 직접 평가되지 않음

1.2 핵심 가설

- 가설 1: 수학 문제 난이도가 다르면 상호보완적 학습이 불가능할 수 있음
- 가설 2: 문제와 답의 형식 불일치가 성능 저하 원인일 수 있음

→ 가설 2에 집중: 모델이 어떤 형식의 문제를 풀 수 있는지 먼저 파악해야 가설 1도 검증 가능하기 때문

2. 진행한 실험

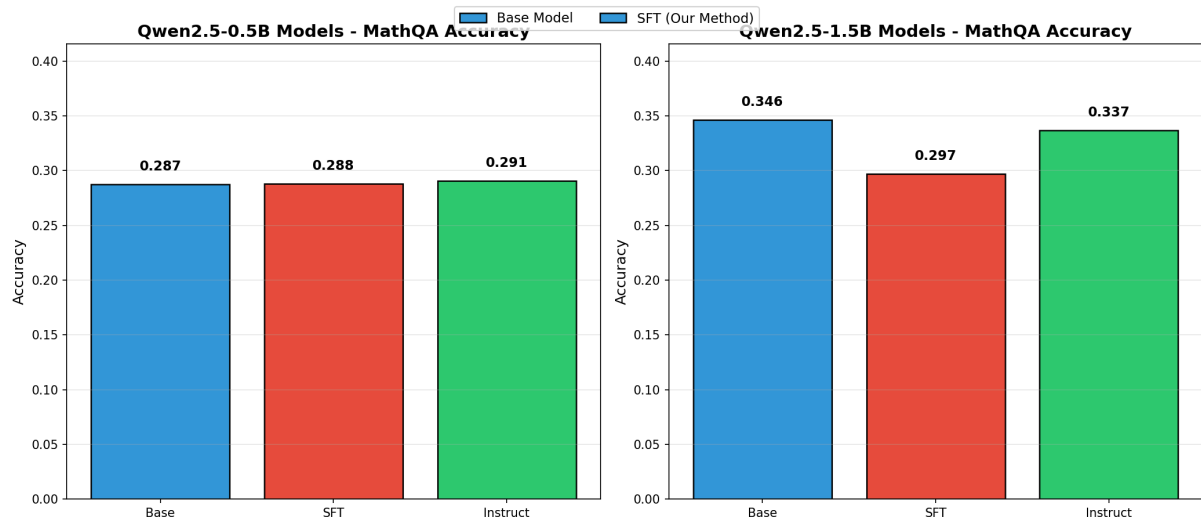
실험	설명
Baseline 4종	Base 2종 + Instruct 2종 (Qwen2.5-0.5B, 1.5B)
SFT (기본)	grade-school-math 데이터로 기존 방식 SFT
MC-loss 방식	동일 데이터로 Multiple Choice 목적함수 학습
In-Domain SFT	MathQA 학습 데이터셋으로 SFT
혼합 SFT	MathQA + GSM 혼합 데이터로 SFT

2.1 MC-loss 방식 상세

일반 SFT는 (prompt → 답 문자열) LM loss인 반면, MathQA 평가는 (prompt, choice_i)의 상대 점수 비교로 결정됨. 이 불일치를 해결하기 위해:

1. 기존 grade-school-math 문제 사용
2. Rule 기반 Python 코드로 각 문제에 5개 옵션 생성
3. 각 옵션 continuation에 대한 log-likelihood 계산
4. 정답 옵션이 가장 높아지도록 cross-entropy(softmax over options)로 학습

3. 실험 결과



MathQA Evaluation Results Summary

Model	Accuracy	Std Error
Qwen2.5-0.5B (Base)	0.2874	0.0083
Qwen2.5-0.5B-math-SFT	0.2884	0.0083
Qwen2.5-0.5B-Instruct	0.2901	0.0083
Qwen2.5-1.5B (Base)	0.3461	0.0087
Qwen2.5-1.5B-math-SFT	0.2978	0.0084
Qwen2.5-1.5B-Instruct	0.3374	0.0087
Qwen2.5-0.5B-math-SFT-Improved-mc	0.2874	0.0083
Qwen2.5-0.5B-SFT-improved-mathqa-train	0.3558	0.0088
Qwen2.5-0.5B-SFT-improved-mathqa-gsm-train	0.3363	0.0086

4. 결론

4.1 핵심 발견

발견	수치	해석
In-domain 데이터가 결정적	0.5B + MathQA (35.58%) vs 0.5B + GSM (28.84%)	+6.7%p 차이. 평가 데이터와 동일 분포의 학습 데이터가 압도적으로 효과적
Out-of-domain은 해로움	1.5B Base (34.61%) → 1.5B + GSM SFT (29.78%)	-4.8%p 하락. Catastrophic forgetting 발생
데이터 혼합의 희석 효과	MathQA 단독 (35.58%) > MathQA + GSM 혼합 (33.63%)	-2%p. Out-of-domain 혼합이 오히려 성능 저하
MC-loss 효과 없음	GSM으로 MC-loss 학습해도 성능 향상 없음	난이도가 다른 문제는 형식 맞춰도 도움 안 됨

4.2 최종 결론

In-Domain 데이터 기반 증강이 가장 효과적. 평가 데이터셋과 동일한 분포의 학습 데이터를 사용하는 것이 핵심이며, Out-of-domain 데이터(초등학교 수학)로는 고등학교 수준 문제 풀이 능력 향상이 불가능함.

5. 향후 실험 아이디어 (미진행)

미진행 사유: LLM API 호출 비용 과다 및 데이터셋 규모로 인한 시간/비용 제약

5.1 동일 문제의 다양한 형태 학습 (Problem Augmentation)

5.1.1 개요

동일한 정답을 가진 문제를 다양한 문장 형태로 변형하여 학습 데이터를 증강하는 방식. 모델이 문제의 본질적인 수학적 구조를 학습하도록 유도함.

5.1.2 구체적 방법

A. 문제 요약 (Condensation)

원본 문제에서 핵심 정보만 추출하여 간결한 형태로 변환하고, (요약된 문제, 정답) pair로 학습

[원본]

"A store sells apples at \$2 per pound. If John wants to buy 5 pounds of apples and he has a 10% discount coupon, how much will he pay?"

[요약]

"Apples: \$2/lb, 5 lbs, 10% discount. Total cost?"

B. 문제 확장 (Expansion)

원본 문제에 배경 설명, 힌트, 또는 추가 컨텍스트를 삽입하고, (확장된 문제, 정답) pair로 학습

[원본]

"Calculate 5^4 "

[확장]

"In a multiple choice test, there are 4 questions and each question has 5 answer choices. To find the total number of ways to complete the test, we need to calculate 5^4 . What is this value?"

C. 문제 패러프레이즈 (Paraphrasing)

동일한 수학적 내용을 다른 어휘와 문장 구조로 재표현하고, (패러프레이즈된 문제, 정답) pair로 학습

[원본]

"What is 15% of 200?"

[패러프레이즈 1]

"If you take fifteen percent from two hundred, what do you get?"

[패러프레이즈 2]

"Calculate the value that represents 15/100 of 200."

[패러프레이즈 3]

"200의 15%는 얼마인가?" (다국어 변형)

5.1.3 직관 및 기대 효과

- **인간 학습 모방:** 사람은 비슷한 유형의 문제를 여러 개 풀면서 해당 유형에 대한 숙달도가 높아짐
- **표현 불변성 학습:** 문제의 표면적 형태가 달라져도 동일한 수학적 구조를 인식하는 능력 향상
- **난이도 스펙트럼 커버:** 축약된 문제(어려움)부터 설명이 풍부한 문제(쉬움)까지 다양한 난이도 학습
- **Robustness 향상:** 평가 시 문제 표현 방식의 변화에 덜 민감해짐

5.1.4 구현 파이프라인

1. 원본 MathQA 문제 선택
2. LLM API 호출하여 3가지 변형 생성 (요약/확장/패러프레이즈)
3. 생성된 변형의 수학적 정확성 검증 (정답이 동일한지 확인)

4. 검증 통과한 데이터로 학습 데이터셋 구성
5. 원본 + 변형 데이터로 SFT 수행

5.2 지식 기반 학습 (Knowledge-Grounded Learning)

5.2.1 개요

문제-답변 쌍을 직접 학습하는 대신, 문제 풀이에 필요한 **수학적 개념과 원리**를 먼저 학습시키는 방식. "수학의 정석"처럼 개념 정리 → 예제 → 응용 순서로 구조화된 지식을 학습함.

5.2.2 구체적인 방법

A. 개념 추출 및 생성

문제로부터 필요한 수학적 개념을 역으로 추출하여 교과서 스타일의 설명 생성

[문제]

"5 choices for each of 4 questions. Total ways = ?"

[추출된 개념]

"## 곱의 법칙 (Multiplication Principle)"

정의: 사건 A가 m 가지 방법으로 일어나고, 각각에 대해 사건 B가 n 가지 방법으로 일어날 수 있다면, A와 B가 함께 일어나는 경우의 수는 $m \times n$ 가지이다.

일반화: k 개의 독립적인 선택이 있고, 각 선택에서 n_1, n_2, \dots, n_k 가지 옵션이 있다면, 전체 경우의 수는 $n_1 \times n_2 \times \dots \times n_k$ 이다.

예시:

- 주사위 2개를 던지는 경우의 수: $6 \times 6 = 36$
- 동전 3개를 던지는 경우의 수: $2 \times 2 \times 2 = 8$

B. 응용 예제 생성

추출된 개념에 대한 다양한 난이도의 응용 예제 생성

[기본]

"셔츠 3벌과 바지 4벌이 있다. 옷을 입는 경우의 수는?"

[중급]

"4자리 비밀번호를 설정하려 한다. 각 자리에 0-9를 사용할 수 있고 중복이 가능하다면, 가능한 비밀번호의 개수는?"

[고급]

" n 개의 질문이 있고 각 질문에 k 개의 선택지가 있는 시험에서, 모든 질문에 답하는 경우의 수를 일반식으로 표현하시오."

C. 학습 데이터 구성

[지식 문서] → [개념 이해 질문] → [응용 문제] 순서로 curriculum 구성

Phase 1: 개념 정의 학습

Input: "곱의 법칙이란 무엇인가?"

Output: [개념 설명]

Phase 2: 개념 적용 학습

Input: "다음 상황에서 곱의 법칙을 적용하시오: ..."

Output: [적용 과정 및 답]

Phase 3: 문제 풀이 학습

Input: [MathQA 스타일 문제]

Output: [정답]

5.2.3 직관 및 기대 효과

- **근본적 이해:** 문제 패턴 암기가 아닌 개념 이해를 통한 일반화 능력 향상
- **전이 학습:** 학습하지 않은 새로운 유형의 문제에도 적용 가능
- **Curriculum Learning:** 쉬운 개념부터 어려운 응용까지 단계적 학습으로 효율성 증가
- **설명 가능성:** 모델이 왜 그 답을 선택했는지 개념 기반 설명 가능

5.2.4 구현 파이프라인

1. MathQA 문제들을 수학적 개념별로 클러스터링
2. 각 클러스터에서 대표 개념 추출 (LLM 활용)
3. 개념별 교과서 스타일 설명 생성
4. 개념별 난이도별 예제 문제 생성
5. [개념 설명 → 예제 → 실전 문제] 순서로 학습 데이터 구성
6. Curriculum Learning 방식으로 SFT 수행

5.3 문제 생성 학습 (Problem Generation Learning)

5.3.1 개요

모델이 문제를 푸는 것이 아니라 **만드는** 법을 학습하게 함. 문제 생성 과정에서 해당 수학적 개념에 대한 깊은 이해가 자연스럽게 형성됨.

5.3.2 구체적인 방법

A. 조건부 문제 생성

주어진 조건(정답, 난이도, 개념)에 맞는 문제를 생성하도록 학습

[입력]

```
{
  "answer": 625,
  "concept": "exponentiation",
  "difficulty": "medium",
  "format": "word_problem"
}
```

[출력]

"A multiple choice test consists of 4 questions, and each question has 5 answer choices. In how many ways can the test be completed if every question is answered?"

B. 문제 변형 생성

기존 문제를 기반으로 유사하지만 다른 문제 생성

[입력]

```
{
  "original": "5^4 = ?",
  "variation_type": "change_numbers",
}
```

```
"constraints": "keep same concept, change base and exponent"
}
```

[출력]

"A restaurant menu has 6 appetizers, and a customer wants to try 3 different ones over 3 visits. If order matters, how many ways can the customer choose? (Note: This is 6^3)"

C. 역문제 생성

답과 풀이 과정이 주어지면 그에 맞는 문제 생성

[입력]

```
{
  "solution_steps": ["Identify this as multiplication principle",
                    "Calculate  $5 \times 5 \times 5 \times 5$ ",
                    "Result: 625"],
  "answer": 625
}
```

[출력]

"In a game, a player must make 4 independent decisions. Each decision has 5 possible choices. How many different game paths are possible?"

5.3.3 직관 및 기대 효과

- **교사 효과 (Teaching Effect):** 가르치는 것이 배우는 것보다 깊은 이해를 만듦. 문제를 만들려면 그 개념을 완전히 이해해야 함
- **창의적 이해:** 단순 패턴 매칭이 아닌, 수학적 구조에 대한 창의적 이해 형성
- **데이터 증강 부수 효과:** 학습된 모델로 추가 학습 데이터 생성 가능 (self-improvement loop)
- **다양성 확보:** 인간이 만들기 어려운 다양한 형태의 문제 생성 가능

5.3.4 구현 파이프라인

1. MathQA 문제들에서 (조건, 문제) 쌍 추출
 - 조건: 정답, 사용된 개념, 난이도, 문제 유형 등
2. 문제 생성 학습 데이터 구성
 - Input: 조건 (JSON 형태)
 - Output: 문제 텍스트
3. 문제 생성 모델 학습 (SFT)
4. 생성된 문제의 품질 검증
 - 수학적 정확성 (정답이 맞는지)
 - 문제의 자연스러움
 - 난이도 적절성
5. (선택) 검증된 생성 문제로 문제 풀이 모델 추가 학습

