

# 하나를 보면 열을 아는 뉴로 심볼릭 코드 학습

2026. 2. 3. 류연희, 허기홍

# 훌륭한 언어



- 스펙과 코드를 함께 작성하는 프로그래밍 언어
  - ▶ A verification-aware programming language
  - ▶ AI시대의 스펙 주도 개발 Spec-driven development with AI에 적합한 언어!

# 훌륭한 언어



- 스펙과 코드를 함께 작성하는 프로그래밍 언어
  - ▶ A verification-aware programming language
  - ▶ AI시대의 스펙 주도 개발 Spec-driven development with AI에 적합한 언어!

```
method smallestNum(list:array<int>) returns (res:int)
  requires list.Length > 0
  ensures forall i :: 0 <= i < list.Length ==> res <= list[i]
  ensures exists i :: 0 <= i < list.Length && res == list[i]
{
  var smallest := list[0];
  var i := 1;
  while i < list.Length
    invariant 1 <= i <= list.Length
    invariant smallest <= list[0]
  {
    if list[i] < smallest
      ...
  }
```

# 훌륭한 언어



함수의  
실행 전후 조건

- 스펙과 코드를 함께 작성하는 프로그래밍 언어
  - ▶ A verification-aware programming language
  - ▶ AI시대의 스펙 주도 개발 Spec-driven development with AI에 적합한 언어!

```
method smallestNum(list:array<int>) returns (res:int)
  requires list.Length > 0
  ensures forall i :: 0 <= i < list.Length ==> res <= list[i]
  ensures exists i :: 0 <= i < list.Length && res == list[i]
{
  var smallest := list[0];
  var i := 1;
  while i < list.Length
    invariant 1 <= i <= list.Length
    invariant smallest <= list[0]
  {
    if list[i] < smallest
      ...
  }
}
```

# 훌륭한 언어



함수의  
실행 전후 조건

반복문 불변식

- 스펙과 코드를 함께 작성하는 프로그래밍 언어
  - ▶ A verification-aware programming language
  - ▶ AI시대의 스펙 주도 개발 Spec-driven development with AI에 적합한 언어!

```
method smallestNum(list:array<int>) returns (res:int)
  requires list.Length > 0
  ensures forall i :: 0 <= i < list.Length ==> res <= list[i]
  ensures exists i :: 0 <= i < list.Length && res == list[i]
{
  var smallest := list[0];
  var i := 1;
  while i < list.Length
    invariant 1 <= i <= list.Length
    invariant smallest <= list[0]
  {
    if list[i] < smallest
      ...
  }
```

# 훌륭한 언어, 그렇지 못한 모델

GPT-4

# 훌륭한 언어, 그렇지 못한 모델

GPT-4

```
method smallestNum(list: array<int>) returns (res: int)
  requires list.Length > 0
  {
    res := list[0];
    for i in 1..list.Length
    {
      if list[i] < res
      {
        res := list[i];
      }
    }
  }
}
```

# 훌륭한 언어, 그렇지 못한 모델

GPT-4

```
method smallestNum(list: array<int>) returns (res: int)
  requires list.Length > 0
{
  res := list[0];
  for i in 1..list.Length
  {
    if list[i] < res
    {
      res := list[i];
    }
  }
}
```

파이썬처럼 작성



# 훌륭한 언어, 그렇지 못한 모델

QwenCoder-1.5B

# 훌륭한 언어, 그렇지 못한 모델

QwenCoder-1.5B

```
method smallestNum(list: array<int>) returns (res: int)
  requires list != null && list.Length > 0
  ensures res >= 0 && res == min(list)
{
  var minVal := list[0];
  for i in 1 ..< list.Length do
    if list[i] < minVal then
      minVal := list[i];
  return minVal;
}
```

# 훌륭한 언어, 그렇지 못한 모델

QwenCoder-1.5B

파이썬처럼 작성

```
method smallestNum(list: array<int>) returns (res: int)
  requires list != null && list.Length > 0
  ensures res >= 0 && res == min(list)
{
  var minVal := list[0];
  for i in 1 ..< list.Length do
    if list[i] < minVal then
      minVal := list[i];
  return minVal;
}
```

# 훌륭한 언어, 그렇지 못한 모델

QwenCoder-1.5B

```
method smallestNum(list: array<int>) returns (res: int)
  requires list != null && list.Length > 0
  ensures res >= 0 && res == min(list)
{
  var minVal := list[0];
  for i in 1 ..< list.Length do
    if list[i] < minVal then
      minVal := list[i];
  return minVal;
}
```

파이썬처럼 작성

문법 키워드 처럼 생겼지만  
틀린 문법

# 훌륭한 언어, 그렇지 못한 모델, 뒤늦은 규칙

QwenCoder-1.5B

+ 조건부 생성

Constrained Decoding

```
method smallestNum(list: array<int>) returns (res: int)
  requires list != null && list.Length > 0
  ensures res >= 0 && res == min(list)
{
  var minVal := list[0];
  for i := 0
```

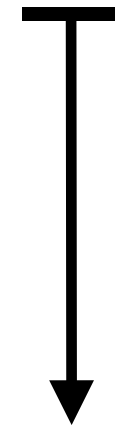
# 훌륭한 언어, 그렇지 못한 모델, 뒤늦은 규칙

QwenCoder-1.5B

+ 조건부 생성

Constrained Decoding

```
method smallestNum(list: array<int>) returns (res: int)
  requires list != null && list.Length > 0
  ensures res >= 0 && res == min(list)
{
  var minVal := list[0];
  for i := 0
```



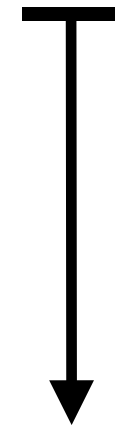
# 훌륭한 언어, 그렇지 못한 모델, 뒤늦은 규칙

QwenCoder-1.5B

+ 조건부 생성

Constrained Decoding

```
method smallestNum(list: array<int>) returns (res: int)
  requires list != null && list.Length > 0
  ensures res >= 0 && res == min(list)
{
  var minVal := list[0];
  for i := 0
```



다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

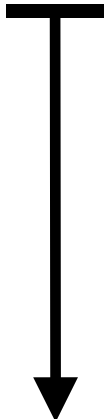
# 훌륭한 언어, 그렇지 못한 모델, 뒤늦은 규칙

QwenCoder-1.5B

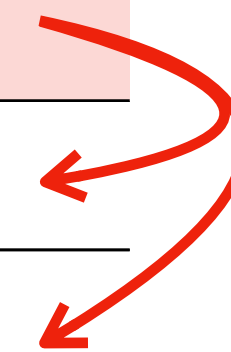
+ 조건부 생성

Constrained Decoding

```
method smallestNum(list: array<int>) returns (res: int)
  requires list != null && list.Length > 0
  ensures res >= 0 && res == min(list)
{
  var minVal := list[0];
  for i := 0
```



다음 토큰 후보	모델 예측 확률	조건부 생성 확률
in	50%	0%
do	30%	60%
to	10%	20%
...	...	...

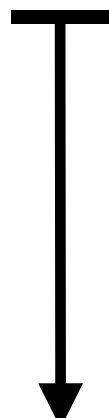




# 훌륭한 언어, 그렇지 못한 모델, 뒤늦은 규칙

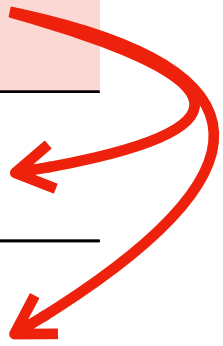
QwenCoder-1.5B  
+ 조건부 생성  
Constrained Decoding

```
method smallestNum(list: array<int>) returns (res: int)
  requires list != null && list.Length > 0
  ensures res >= 0 && res == min(list)
{
  var minVal := list[0];
  for i := 0
```



다음 토큰 후보	모델 예측 확률	조건부 생성 확률
in	50%	0%
do	30%	60%
to	10%	20%
...	...	...

"downto"의 일부  
로 인식하고 허용



# 훌륭한 언어, 그렇지 못한 모델, 뒤늦은 규칙

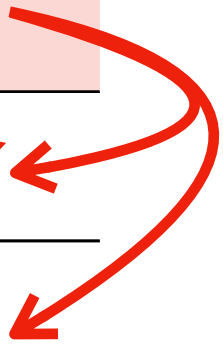
QwenCoder-1.5B  
+ 조건부 생성  
Constrained Decoding

```
method smallestNum(list: array<int>) returns (res: int)
  requires list != null && list.Length > 0
  ensures res >= 0 && res == min(list)
{
  var minVal := list[0];
  for i := 0 do
```



다음 토큰 후보	모델 예측 확률	조건부 생성 확률
in	50%	0%
do	30%	60% ★
to	10%	20%
...	...	...

"downto"의 일부  
로 인식하고 허용



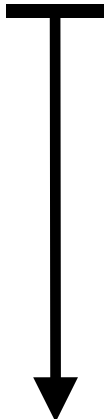
# 훌륭한 언어, 그렇지 못한 모델, 뒤늦은 규칙

QwenCoder-1.5B

+ 조건부 생성

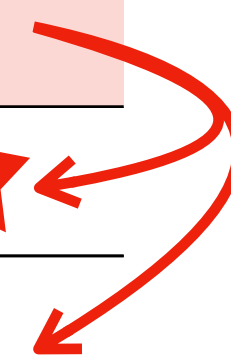
Constrained Decoding

```
method smallestNum(list: array<int>) returns (res: int)
  requires list != null && list.Length > 0
  ensures res >= 0 && res == min(list)
{
  var minVal := list[0];
  for i := 0 downto list.Length-1 { ... }
```



다음 토큰 후보	모델 예측 확률	조건부 생성 확률
in	50%	0%
do	30%	60% ★
to	10%	20%
...	...	...

"downto"의 일부  
로 인식하고 허용

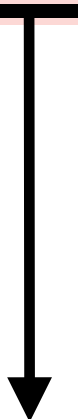


# 훌륭한 언어, 그렇지 못한 모델, 뒤늦은 규칙

QwenCoder-1.5B  
+ 조건부 생성  
Constrained Decoding

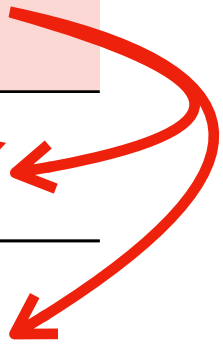
```
method smallestNum(list: array<int>) returns (res: int)
  requires list != null && list.Length > 0
  ensures res >= 0 && res == min(list)
{
  var minVal := list[0];
  for i := 0 downto list.Length-1 { ... }
```

Out of Bound!



"downto"의 일부  
로 인식하고 허용

다음 토큰 후보	모델 예측 확률	조건부 생성 확률
in	50%	0%
do	30%	60% ★
to	10%	20%
...	...	...

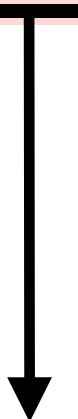


# 훌륭한 언어, 그렇지 못한 모델, 뒤늦은 규칙

QwenCoder-1.5B  
+ 조건부 생성  
Constrained Decoding

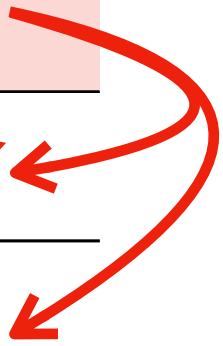
```
method smallestNum(list: array<int>) returns (res: int)
  requires list != null && list.Length > 0
  ensures res >= 0 && res == min(list)
{
  var minVal := list[0];
  for i := 0 downto list.Length-1 { ... }
```

Out of Bound!



다음 토큰 후보	모델 예측 확률	조건부 생성 확률
in	50%	0%
do	30%	60% ★
to	10%	20%
...	...	...

"downto"의 일부  
로 인식하고 허용



미리 "to"를  
선택했어야 했는데...

# 얼마나 못할까?

- MBPP-DFY [FSE'24]: Dafny 언어로 번역된 MBPP 228개 과제

# 얼마나 못할까?

- MBPP-DFY [FSE'24]: Dafny 언어로 번역된 MBPP 228개 과제

언어별 테스트 성공율

# 얼마나 못할까?

- MBPP-DFY [FSE'24]: Dafny 언어로 번역된 MBPP 228개 과제

## 언어별 테스트 성공율

모델
GPT-5-Nano
QwenCoder-7B
QwenCoder-3B
QwenCoder-1.5B



# 얼마나 못할까?

- MBPP-DFY [FSE'24]: Dafny 언어로 번역된 MBPP 228개 과제

## 언어별 테스트 성공율

모델	Python
GPT-5-Nano	94.7%
QwenCoder-7B	83.0%
QwenCoder-3B	77.7%
QwenCoder-1.5B	77.2%

# 얼마나 못할까?

- MBPP-DFY [FSE'24]: Dafny 언어로 번역된 MBPP 228개 과제

## 언어별 테스트 성공율

모델	Python	Dafny
GPT-5-Nano	94.7%	38.8%
QwenCoder-7B	83.0%	13.3%
QwenCoder-3B	77.7%	11.8%
QwenCoder-1.5B	77.2%	8.3%

# 얼마나 못할까?

- MBPP-DFY [FSE'24]: Dafny 언어로 번역된 MBPP 228개 과제
- Python 의 절반도 못한다!

## 언어별 테스트 성공율

모델	Python	Dafny
GPT-5-Nano	94.7%	38.8%
QwenCoder-7B	83.0%	13.3%
QwenCoder-3B	77.7%	11.8%
QwenCoder-1.5B	77.2%	8.3%

# 얼마나 못할까?

- MBPP-DFY [FSE'24]: Dafny 언어로 번역된 MBPP 228개 과제
- Python 의 절반도 못한다!

언어별 테스트 성공율

모델	Python	Dafny
GPT-5-Nano	94.7%	38.8%
QwenCoder-7B	83.0%	13.3%
QwenCoder-3B	77.7%	11.8%
QwenCoder-1.5B	77.2%	8.3%

Dafny 테스트 실패 이유

# 얼마나 못할까?

- MBPP-DFY [FSE'24]: Dafny 언어로 번역된 MBPP 228개 과제
- Python 의 절반도 못한다!

언어별 테스트 성공율

모델	Python	Dafny
GPT-5-Nano	94.7%	38.8%
QwenCoder-7B	83.0%	13.3%
QwenCoder-3B	77.7%	11.8%
QwenCoder-1.5B	77.2%	8.3%

Dafny 테스트 실패 이유

검증 실패
14.2%
10.0%
4.7%
2.2%

# 얼마나 못할까?

- MBPP-DFY [FSE'24]: Dafny 언어로 번역된 MBPP 228개 과제
- Python 의 절반도 못한다!

언어별 테스트 성공율

모델	Python	Dafny
GPT-5-Nano	94.7%	38.8%
QwenCoder-7B	83.0%	13.3%
QwenCoder-3B	77.7%	11.8%
QwenCoder-1.5B	77.2%	8.3%

Dafny 테스트 실패 이유

검증 실패	문법 오류
14.2%	36.6%
10.0%	76.1%
4.7%	82.5%
2.2%	88.7%

# 얼마나 못할까?

- MBPP-DFY [FSE'24]: Dafny 언어로 번역된 MBPP 228개 과제
- Python 의 절반도 못한다!

언어별 테스트 성공율

모델	Python	Dafny
GPT-5-Nano	94.7%	38.8%
QwenCoder-7B	83.0%	13.3%
QwenCoder-3B	77.7%	11.8%
QwenCoder-1.5B	77.2%	8.3%

Dafny 테스트 실패 이유

검증 실패	문법 오류	loop-do
14.2%	36.6%	0.3%
10.0%	76.1%	2%
4.7%	82.5%	8%
2.2%	88.7%	19%

조건부 생성으로  
해결하기에는  
이미 낮은 패턴

# 얼마나 못할까?

- MBPP-DFY [FSE'24]: Dafny 언어로 번역된 MBPP 228개 과제
- Python 의 절반도 못한다!

언어별 테스트 성공율

모델	Python	Dafny
GPT-5-Nano	94.7%	38.8%
QwenCoder-7B	83.0%	13.3%
QwenCoder-3B	77.7%	11.8%
QwenCoder-1.5B	77.2%	8.3%

Dafny 테스트 실패 이유

검증 실패	문법 오류	loop-do	for-in
14.2%	36.6%	0.3%	33%
10.0%	76.1%	2%	13%
4.7%	82.5%	8%	9%
2.2%	88.7%	19%	8%

조건부 생성으로  
해결하기에는  
이미 낮은 패턴

다른 언어를 잘 학습  
해서 틀리는 패턴



# 얼마나 못할까?

- MBPP-DFY [FSE'24]: Dafny 언어로 번역된 MBPP 228개 과제
- Python 의 절반도 못한다!

언어별 테스트 성공율

모델	Python	Dafny
GPT-5-Nano	94.7%	38.8%
QwenCoder-7B	83.0%	13.3%
QwenCoder-3B	77.7%	11.8%
QwenCoder-1.5B	77.2%	8.3%

Dafny 테스트 실패 이유

검증 실패	문법 오류	loop-do	for-in
14.2%	36.6%	0.3%	33%
10.0%	76.1%	2%	13%
4.7%	82.5%	8%	9%
2.2%	88.7%	19%	8%

조건부 생성으로  
해결하기에는  
이미 낮은 패턴

다른 언어를 잘 학습  
해서 틀리는 패턴

학습하자!

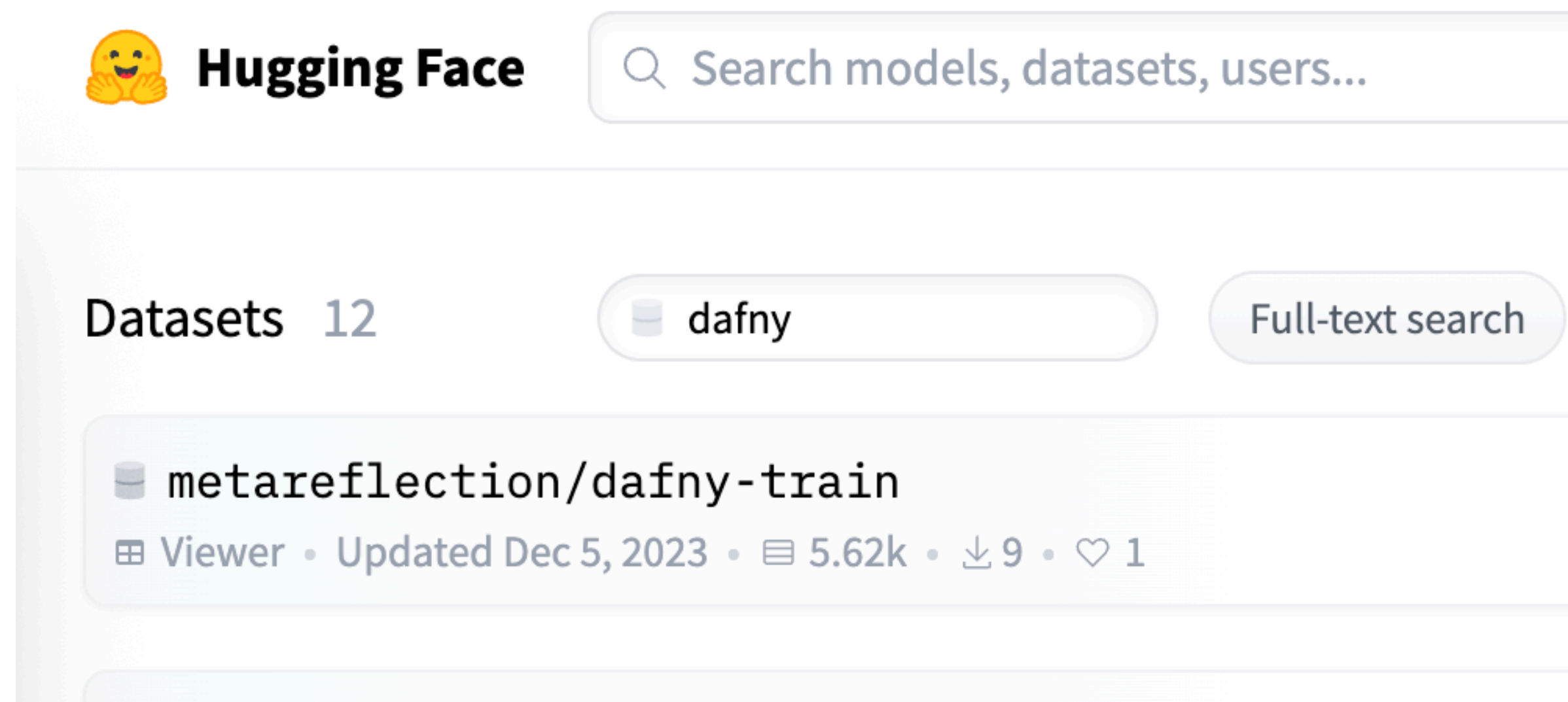
# 훌륭하지만 비주류

# 출렁하지만 비주류

- 데이터가 없다

# 훌륭하지만 비주류

- 데이터가 없다
  - ▶ HuggingFace 오픈 데이터: 최대 5.6K



# 훌륭하지만 비주류

- 데이터가 없다
  - ▶ HuggingFace 오픈 데이터: 최대 5.6K
  - ▶ GitHub 에서 Dafny 로 검색되는 파일 수: 54.3K

The image shows two overlapping screenshots. The background screenshot is from Hugging Face, displaying search results for 'dafny' under the 'Datasets' tab. It shows 12 datasets, with the top result being 'metareflection/dafny-train', which has 5.62k files and was updated on Dec 5, 2023. The foreground screenshot is from GitHub, showing search results for 'language:dafny'. It displays 585 repositories, with the top result being 'Consensys/evm-dafny', an EVM interpreter in Dafny, which has 131 stars and was updated on 2024년 10월 24일. The GitHub interface also shows a 'Filter by' sidebar with counts for Code (54.3k), Repositories (585), Issues (771), Pull requests (1k), and Discussions (0).

**Hugging Face** Search models, datasets, u

**Datasets 12** dafny

metareflection/dafny-train  
Viewer • Updated Dec 5, 2023 • 5.62k • 9 • 1

**GitHub** language:dafny

Filter by

- <> Code 54.3k
- Repositories 585
- Issues 771
- Pull requests 1k
- Discussions 0

585 results (2 s) Sort by: Most stars Save ...

**Consensys/evm-dafny** Star  
An EVM interpreter in Dafny  
Dafny • 131 • Updated on 2024년 10월 24일

**Consensys/eth2.0-dafny** Star

# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터    **for**  $i := 0$

# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터     **for** i := 0

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...



# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터     **for** i := 0 **to**

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터     **for** i := 0 **to**

학습 전 모델

다음 토큰 학습하기

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터    **for** i := 0 **to**

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

다음 토큰 학습하기

다음 토큰 후보
in
do
to
...

# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법


학습 데이터    **for** i := 0 **to**

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

다음 토큰 학습하기

다음 토큰 후보
in
do
to
...



# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법


학습 데이터    **for** i := 0 **to**

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

다음 토큰 학습하기

다음 토큰 후보	모델 예측 확률
in	40%
do	25%
to	25%
...	...



# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터    **for** i := 0 **to**

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

다음 토큰 학습하기

다음 토큰 후보	모델 예측 확률
in	40%
do	25%
to	25%
...	...

정답 "to" 를 예측할  
확률 일부 개선

# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터    **for**  $i := 0$  **to**    + 규칙    **for**  $i := 0$  **in**

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

다음 토큰 학습하기

다음 토큰 후보	모델 예측 확률
in	40%
do	25%
to	25%
...	...

정답 "to" 를 예측할  
확률 일부 개선

# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터    **for** i := 0 **to**    + 규칙    **for** i := 0 **in**

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

다음 토큰 학습하기

다음 토큰 후보	모델 예측 확률
in	40%
do	25%
to	25%
...	...

틀린 토큰도 학습하기

정답 "to" 를 예측할  
확률 일부 개선



# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터    **for** i := 0 **to**    + 규칙    **for** i := 0 **in**

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

다음 토큰 학습하기

다음 토큰 후보	모델 예측 확률
in	40%
do	25%
to	25%
...	...

틀린 토큰도 학습하기

다음 토큰 후보
in
do
to
...

정답 "to" 를 예측할  
확률 일부 개선

# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터    **for** i := 0 **to**    + 규칙    **for** i := 0 **in**

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

다음 토큰 학습하기

다음 토큰 후보	모델 예측 확률
in	40%
do	25%
to	25%
...	...

틀린 토큰도 학습하기

다음 토큰 후보
in
do
to
...

정답 "to" 를 예측할  
확률 일부 개선

# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터    **for** i := 0 **to**    + 규칙    **for** i := 0 **in**

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

다음 토큰 학습하기

다음 토큰 후보	모델 예측 확률
in	40%
do	25%
to	25%
...	...

틀린 토큰도 학습하기

다음 토큰 후보
in
do
to
...

정답 "to" 를 예측할  
확률 일부 개선

# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터    **for** i := 0 **to**    + 규칙    **for** i := 0 **in**

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

다음 토큰 학습하기

다음 토큰 후보	모델 예측 확률
in	40%
do	25%
to	25%
...	...

정답 "to" 를 예측할  
확률 일부 개선

틀린 토큰도 학습하기

다음 토큰 후보	모델 예측 확률
in	15%
do	25%
to	50%
...	...

# 뉴로 심볼릭 코드 학습: 먼저 온 규칙

- 뉴로 심볼릭 학습: 규칙을 이용하여 적은 데이터에서도 효율적으로 학습하는 방법

학습 데이터    **for** i := 0 **to**    + 규칙    **for** i := 0 **in**

학습 전 모델

다음 토큰 후보	모델 예측 확률
in	50%
do	30%
to	10%
...	...

다음 토큰 학습하기

다음 토큰 후보	모델 예측 확률
in	40%
do	25%
to	25%
...	...

정답 "to" 를 예측할  
확률 일부 개선

틀린 토큰도 학습하기

다음 토큰 후보	모델 예측 확률
in	15%
do	25%
to	50%
...	...

하나의 데이터에서도  
동시에 여러 정보 학습 가능!

# 문법 규칙을 활용하는 뉴로 심볼릭 코드 학습

# 문법 규칙을 활용하는 뉴로 심볼릭 코드 학습

- Dafny 데이터 18K + 문법 마스크 학습

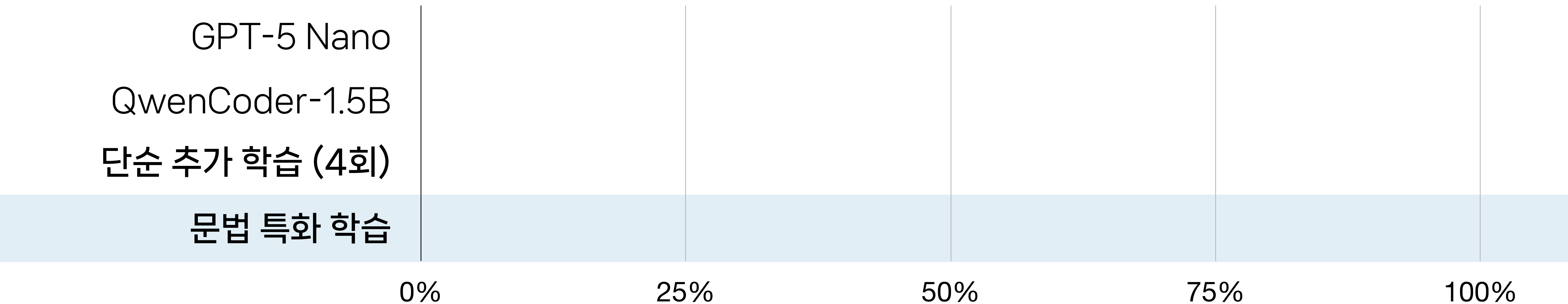
# 문법 규칙을 활용하는 뉴로 심볼릭 코드 학습

- Dafny 데이터 18K + 문법 마스크 학습
- QwenCoder 1.5B + LoRA



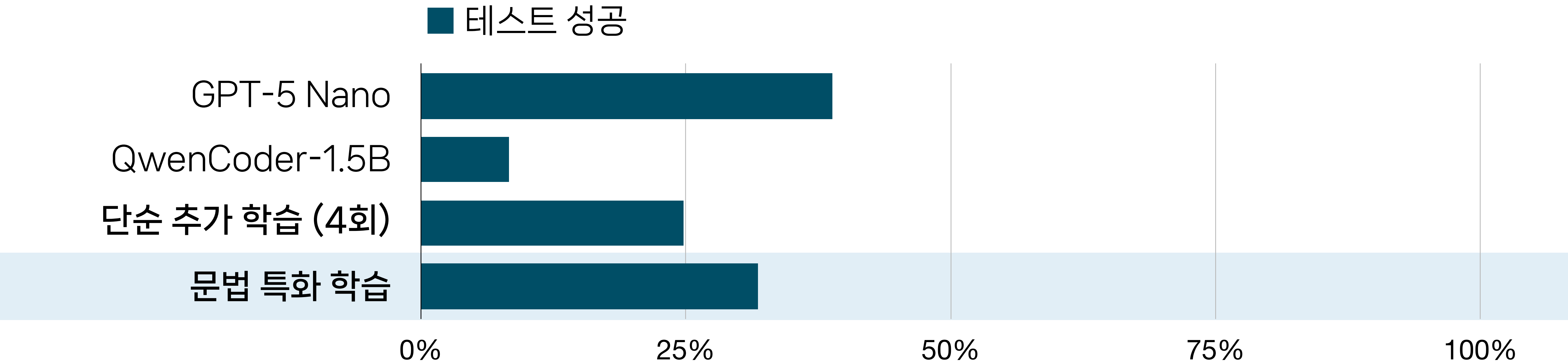
# 문법 규칙을 활용하는 뉴로 심볼릭 코드 학습

- Dafny 데이터 18K + 문법 마스크 학습
- QwenCoder 1.5B + LoRA



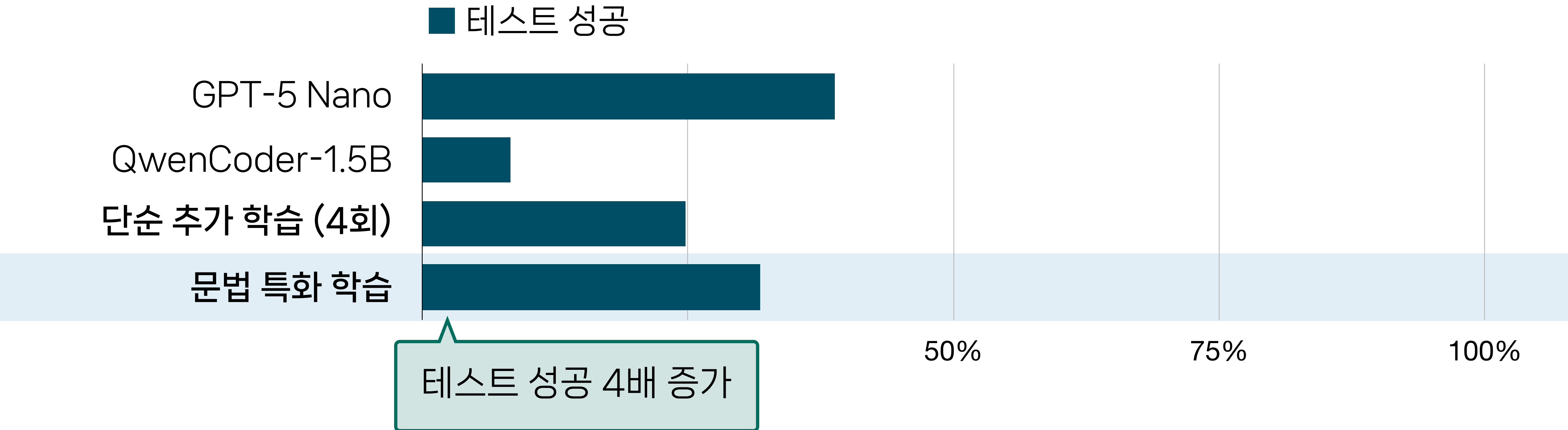
# 문법 규칙을 활용하는 뉴로 심볼릭 코드 학습

- Dafny 데이터 18K + 문법 마스크 학습
- QwenCoder 1.5B + LoRA



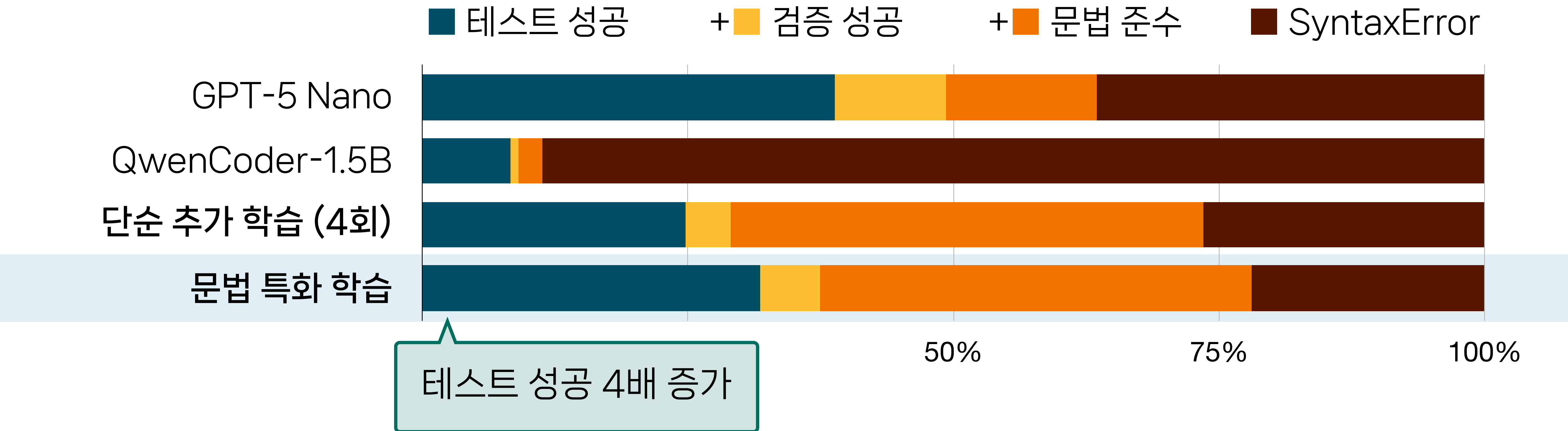
# 문법 규칙을 활용하는 뉴로 심볼릭 코드 학습

- Dafny 데이터 18K + 문법 마스크 학습
- QwenCoder 1.5B + LoRA



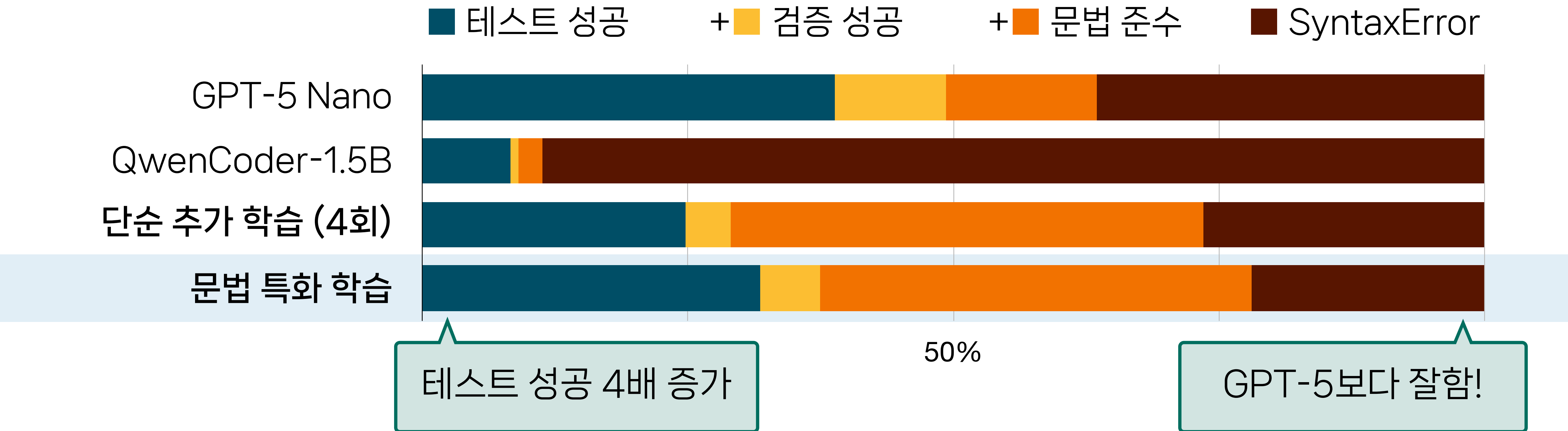
# 문법 규칙을 활용하는 뉴로 심볼릭 코드 학습

- Dafny 데이터 18K + 문법 마스크 학습
- QwenCoder 1.5B + LoRA



# 문법 규칙을 활용하는 뉴로 심볼릭 코드 학습

- Dafny 데이터 18K + 문법 마스크 학습
- QwenCoder 1.5B + LoRA



# 하나를 보면 열을 아는 뉴로 심볼릭 코드 학습

## 하나를 보면 열을 아는 뉴로 심볼릭 코드 학습

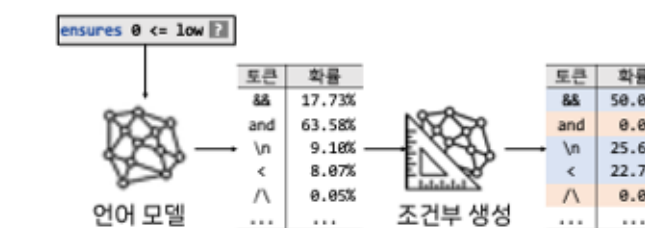
류연희, 허기홍

KAIST

Programming Systems Laboratory

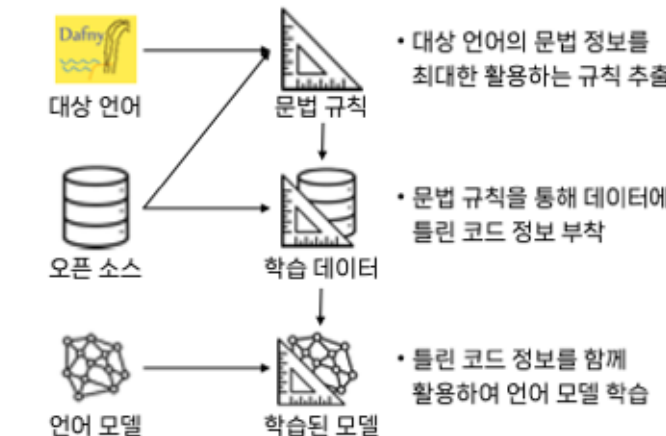
### 1 배경: 조건부 생성Constrained Decoding

- 언어 모델이 출력하는 확률을 규칙에 따라 조작
  - 문법에 맞는 코드만 출력 (Ugare et al., TML 2025, Park et al., ICML 2025)
  - 타입이 올바른 코드만 출력 (Mandler et al., PLDI 2025)



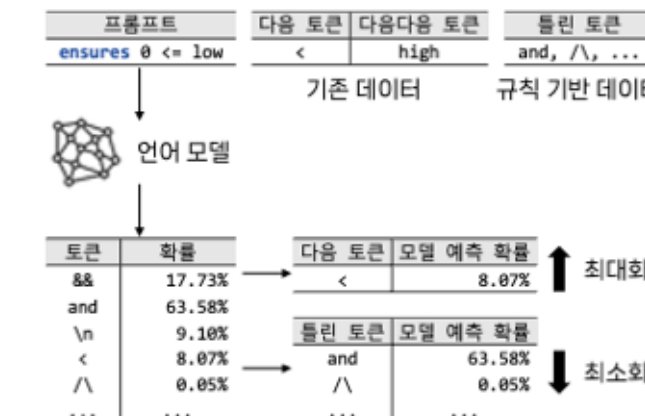
### 3 해결: 뉴로 심볼릭 학습

- 오픈 소스로부터 대상 언어 코드를 수집



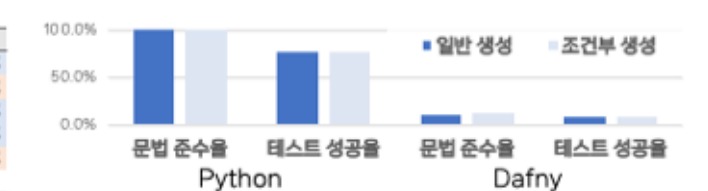
### 5 규칙 기반 언어 모델 학습

- 규칙 기반 데이터로부터 "다음 토큰 예측 확률" 과 "틀린 토큰 예측 확률"을 동시에 학습



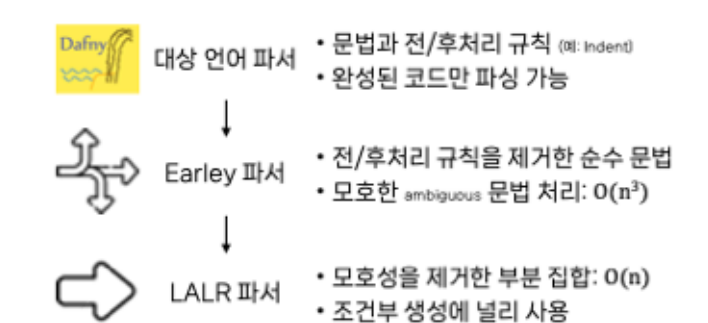
### 2 배경: 비주류 언어와 언어 모델

- 주류 언어에 비해 언어 모델의 성능 부족
- 학습할만한 데이터 부족
  - GitHub 저장소 검색량: Python 5.8M vs Dafny 728
  - HuggingFace 검색량: Python 1.5K vs Dafny 10



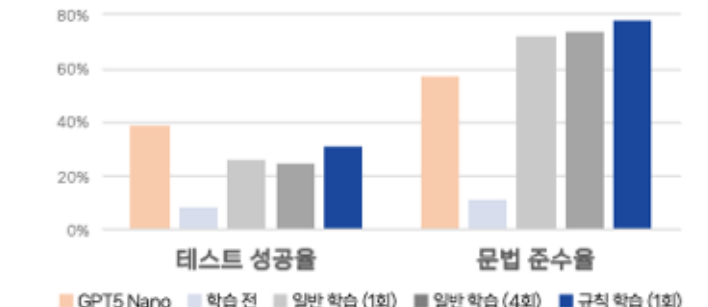
### 4 문법 규칙 최적화

- 대상 언어의 문법을 조건부 생성에 적합하게 변환
  - 선형 시간  $O(n)$  상황식 부분 파서 (Bottom up Incremental 필요)
- 학습 데이터에서 많은 정보를 수집하도록 최적화



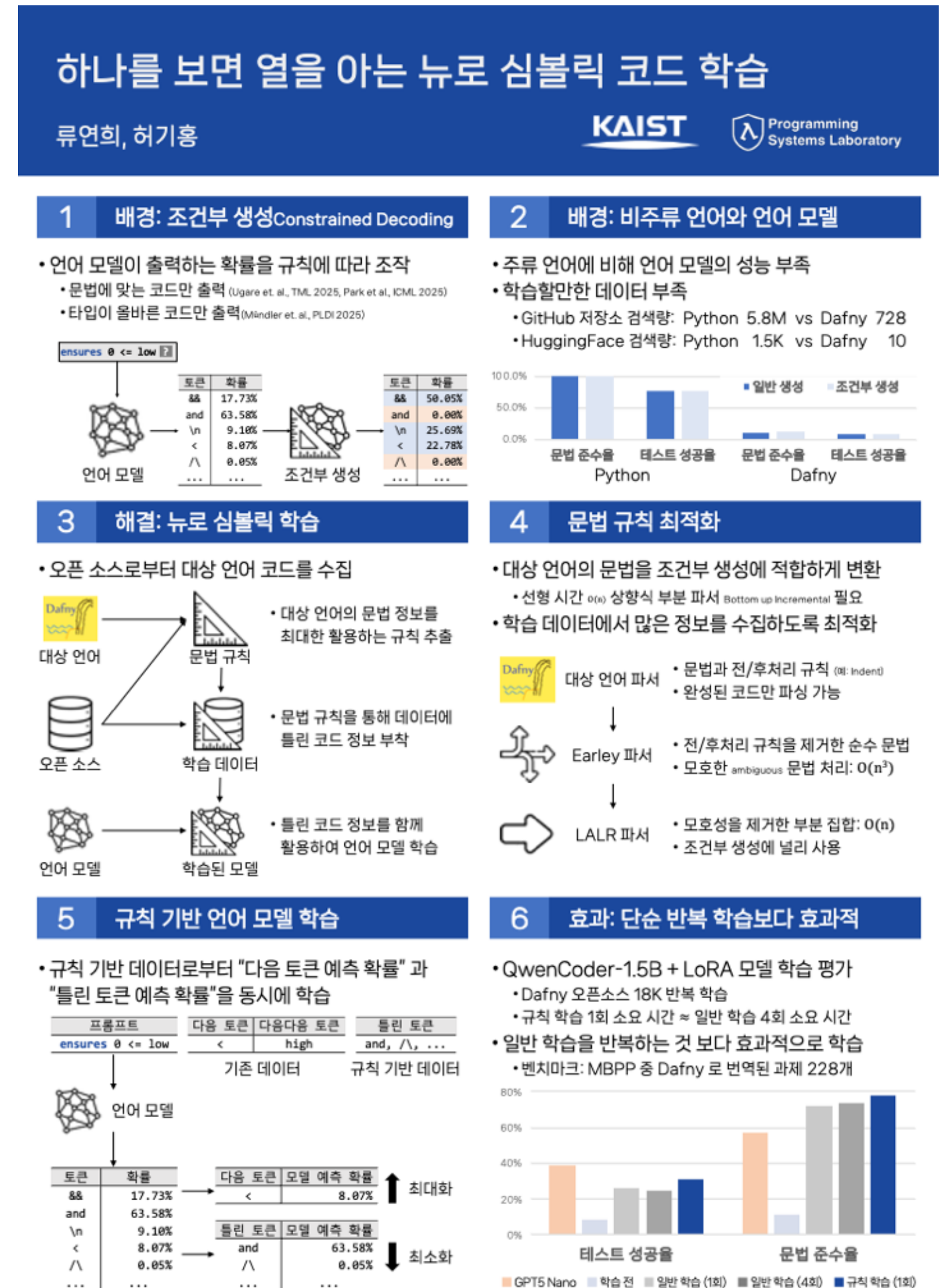
### 6 효과: 단순 반복 학습보다 효과적

- QwenCoder-1.5B + LoRA 모델 학습 평가
  - Dafny 오픈소스 18K 반복 학습
  - 규칙 학습 1회 소요 시간  $\approx$  일반 학습 4회 소요 시간
- 일반 학습을 반복하는 것 보다 효과적으로 학습
  - 벤치마크: MBPP 중 Dafny 로 번역된 과제 228개



# 하나를 보면 열을 아는 뉴로 심볼릭 코드 학습

- 언어 모델을 이용한 비주류 언어 생성 품질 문제
  - 조건부 생성 하기에는 기본 모델이 너무 못함
  - 학습하기에는 오픈 소스 데이터가 너무 없음





# 하나를 보면 열을 아는 뉴로 심볼릭 코드 학습

- 언어 모델을 이용한 비주류 언어 생성 품질 문제
  - 조건부 생성 하기에는 기본 모델이 너무 못함
  - 학습하기에는 오픈 소스 데이터가 너무 없음
- 뉴로-심볼릭 학습: 데이터 부족 문제를 규칙으로 해결!
  - 1.5B 모델로도 GPT-5 Nano 수준의 생성 품질
  - 단순 반복 학습에 비해 효과적

