

Enhancing LLM Code Generation with Ensembles: A Similarity-Based Selection Approach

Tarek Mahmud
Texas State University

Bin Duan
The University of Queensland

Corina Pasareanu
Carnegie Mellon University

Guowei Yang
The University of Queensland

The 48th IEEE/ACM International Conference on Software Engineering (ICSE 2026)

背景

LLMのコード生成性能

- 生成されたコードはまだ完璧じゃない

HumanEvalでの正答率

GPT-4
約 82%

CodeLlama
42~50%

十分な正解率ではない

Codex [14] の HumanEval での結果

pass@k : k 個生成し, どれか1つでも正解なら成功

pass@1
28.8%



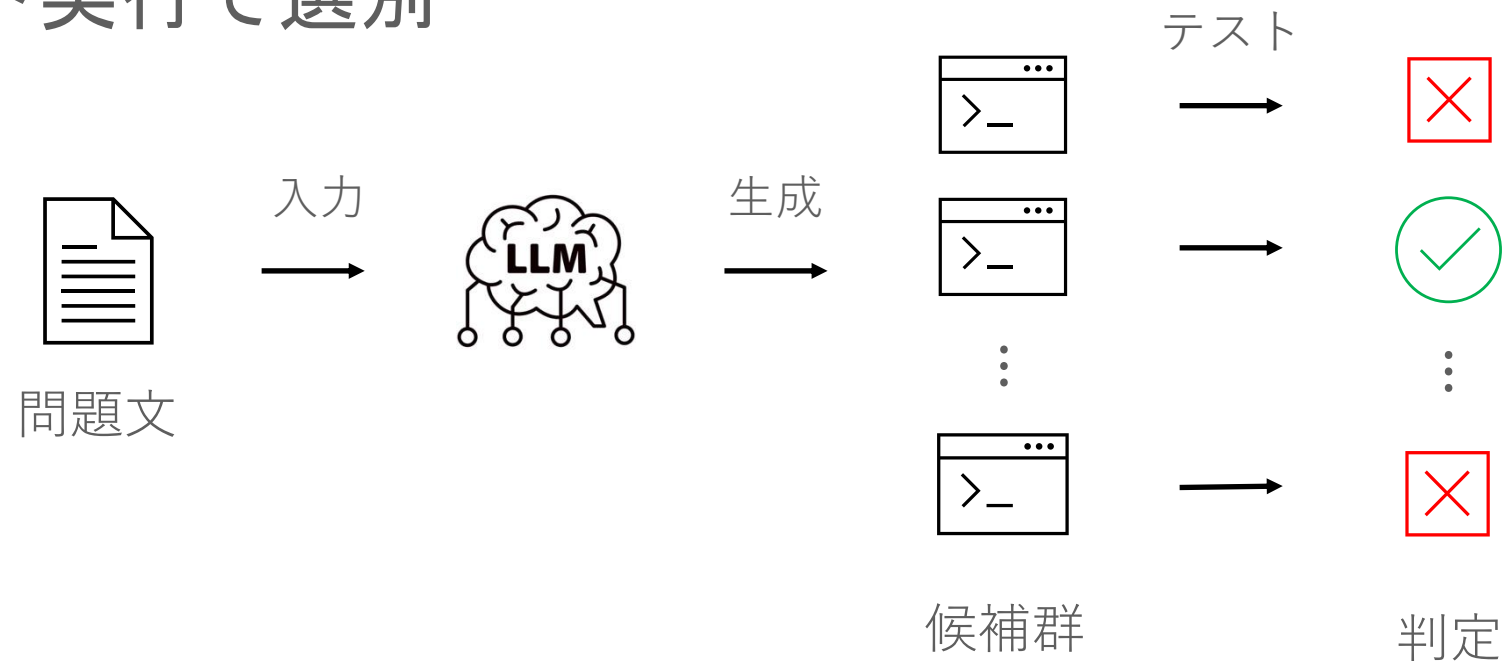
pass@100
70.2%

正解は候補の中にあることが多い

候補の選び方

先行研究^{[1][2]}

- テスト実行で選別



- 問題点

一つのモデルで生成させるため候補の**多様性が限られる**

同じような間違いをしやすい

アンサンブル学習的なアプローチが取れないか？

アンサンブル学習

アンサンブル学習：複数のモデルの予測を組み合わせる手法

分類タスクとコード生成の違い

分類問題

モデル1：猫

モデル2：猫

モデル3：犬

多数決で
「猫」

同じラベルを数えるだけ

コード生成

モデル 1

```
while left <= right:
    mid = (left+right)//2
    if arr[mid] == target:
        return mid
    ...
```

モデル 2

```
if left > right:
    return -1
mid = (left+right)//2
...
return bs(mid+1, right)
```

文字列で判断できない

機能的に同じコードでも文字列は同じではない可能性

→多数決が取れない

目的・アプローチ

目的

- アンサンブル学習をコード生成に適用し，コード生成の精度を高める

アプローチ

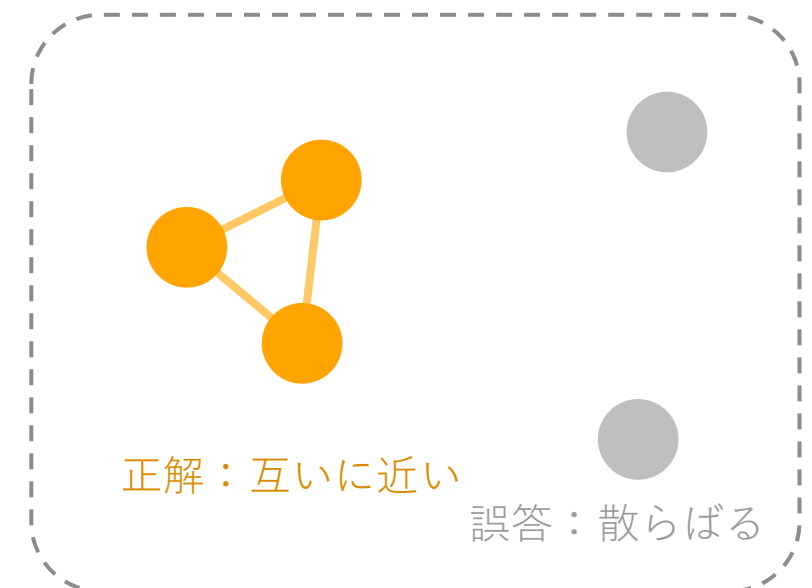
- 類似度ベースの投票

コード同士の類似度を測り，他の候補と最も似ているコードを選択

前提

異なる LLM は同じ間違いをしにくい

正解同士は互いに似て，誤答はばらばらになる



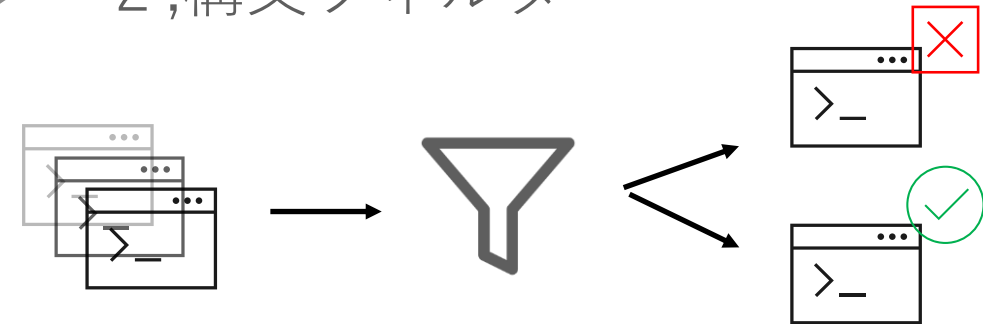
提案手法 | 概略

1, 候補の生成



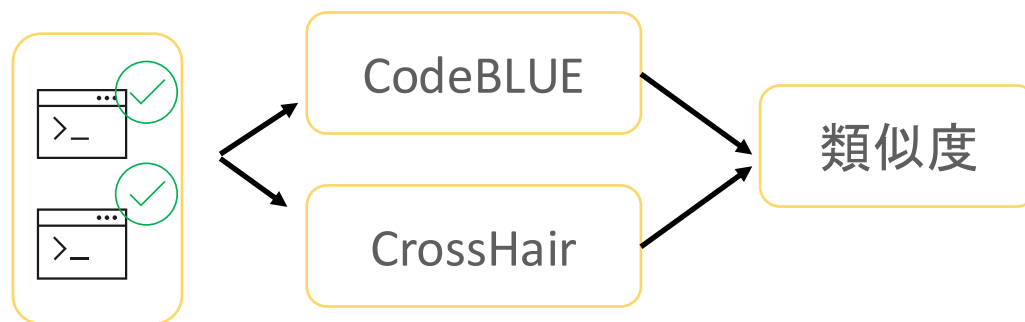
14モデルで生成

2, 構文フィルタ



Pylint(静的解析) で
構文エラーのある候補を除外

3, 類似度計算



候補の全ペアで類似度を計算

4, 選択

P1	P2 0.9	P3 0.625	P4 0.7	2.225
P2	P1 0.9	P3 0.625	P4 0.75	2.275 ✓
P3	P1 0.625	P2 0.625	P4 0.5	1.75
P4	P1 0.7	P2 0.75	P3 0.5	1.95

区切り = その候補との類似度 (票)

類似度の合計が最大の候補を出力

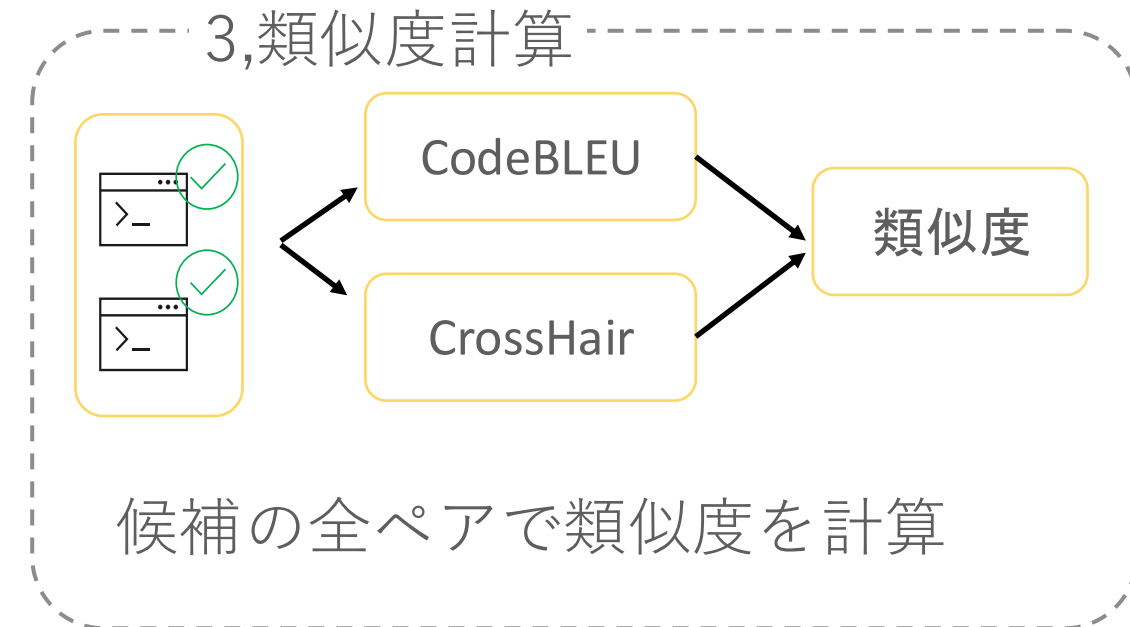
提案手法 | 類似度計算

CodeBLEU

- コード向けの類似度計算

字面, 構文, 意味を考慮

今回字面は使用しない



CrossHair

- Python関数の解析ツール

2つの関数に同じ入力を与え, 出力が食い違う入力を探す

出力が食い違うたびに減点

提案手法 | 選択

各候補ペアの類似度を合計

- 例

	P1	P2	P3	P4	P5	合計
P1	—	0.9	0.8	0.4	0.3	2.4
P2	0.9	—	0.9	0.4	0.3	2.5
P3	0.8	0.9	—	0.3	0.3	2.3
P4	0.4	0.4	0.3	—	0.2	1.3
P5	0.3	0.3	0.3	0.2	—	1.1

合計が高いP2を選択

評価実験 | RQs・ベンチマーク

RQ

- 単体のLLMと比べて精度は上がるか
- CodeBLEUとCrossHairはそれぞれどれだけ寄与するか
- オープンソースLLMだけでどこまで精度を出せるか

ベンチマーク

- HumanEval(164問)
- LiveCodeBench (511問)

評価実験 | モデル・評価指標

使用したLLM

商用 (5種)
API で利用

GPT-4o, GPT-4, GPT-3.5, CoPilot, Gemini

無料 (9種)
Ollama で実行

OpenChat, CodeBERT, Llama 3.2, Qwen2, Codestral,
Gemma 2, DeepSeekCoder, CodeLlama, DolphinCoder

評価指標

Pass@1:各 LLM は1回だけ生成. EnsLLM は選んだ1つで正誤を判定

- 達成可能精度 (提案手法にとっての上限)

$$\text{達成可能精度} = C / M \times 100$$

C : 少なくとも1つの LLM が正解した問題数 M : 全問題数

候補から選ぶだけなので、これを超えることはできない

結果 | RQ1（単体との比較）

LLM	HumanEval (%)	LiveCodeBench (%)
GPT-4o	83.5	43.4
GPT-4	82.3	42.2
GPT-3.5	76.8	39.1
CoPilot	74.4	41.6
OpenChat (F)	71.3	37.3
CodeBERT (F)	68.9	37.2
Llama 3.2 (F)	68.9	36.7
Gemini	66.5	36.1
Qwen2 (F)	62.1	36.4
Codestral (F)	59.2	37.2
Gemma 2 (F)	52.1	31.3
DeepSeekCoder (F)	50.6	25.6
CodeLlama (F)	42.7	23.4
DolphinCoder (F)	34.8	22.2
EnsLLM（全14）	90.2 (90.9)	50.2 (53.8)
EnsLLM（上位5）	87.2 (90.9)	48.3 (53.8)
EnsLLM（無料9）	80.5 (83.2)	41.6 (44.1)

(F)：無料のモデル 括弧内：達成可能精度（上限）

単体で最良：GPT-4o

83.5 / 43.4



EnsLLM（全14モデル）

90.2 / 50.2

+6.7pt / +6.8pt

(HumanEval / LiveCodeBench, %)

全ての単体 LLM を上回る

結果 | RQ1 (単体との比較)

LLM	HumanEval (%)	LiveCodeBench (%)
GPT-4o	83.5	43.4
GPT-4	82.3	42.2
GPT-3.5	76.8	39.1
CoPilot	74.4	41.6
OpenChat (F)	71.3	37.3
CodeBERT (F)	68.9	37.2
LLaMA-3.2 (F)	68.0	36.7
Gemini-1.5	67.8	36.5
Qwen2.5-72B	67.7	36.4
Claude-3.5	67.6	36.3
Gemini-1.5 Pro	67.5	36.2
DolphinCode (F)	54.0	22.2
EnsLLM (全14)	90.2 (90.9)	50.2 (53.8)
EnsLLM (上位5)	87.2 (90.9)	48.3 (53.8)
EnsLLM (無料9)	80.5 (83.2)	41.6 (44.1)

上限との差

HumanEval : 0.7pt(1 問)

LiveCodeBench : 3.6pt(18問)

正解が候補にあればほぼ正確に選んでいる

(F) : 無料のモデル 括弧内 : 達成可能精度 (上限)

結果 | RQ1（単体との比較）

LLM	HumanEval (%)	LiveCodeBench (%)
GPT-4o	83.5	43.4
GPT-4	82.3	42.2
GPT-3.5	76.8	39.1
CoPilot	74.4	41.6
OpenChat (F)	71.3	37.3
CodeBERT (F)	68.9	37.2
LLaMA 3.2 (F)	68.0	36.7
Gemini 1.5 Pro	67.8	36.5
Qwen2.5 72B	67.7	36.4
Claude 3.5 Sonnet	67.6	36.3
Gemini 1.5 Flash	67.5	36.2
DolphinCode (F)	54.0	22.2
EnsLLM（全14）	90.2 (90.9)	50.2 (53.8)
EnsLLM（上位5）	87.2 (90.9)	48.3 (53.8)
EnsLLM（無料9）	80.5 (83.2)	41.6 (44.1)

候補モデル数の差

上位5モデルより全14モデルの方が高い

性能が低いモデルも貢献している

(F)：無料のモデル 括弧内：達成可能精度（上限）

結果 | RQ2

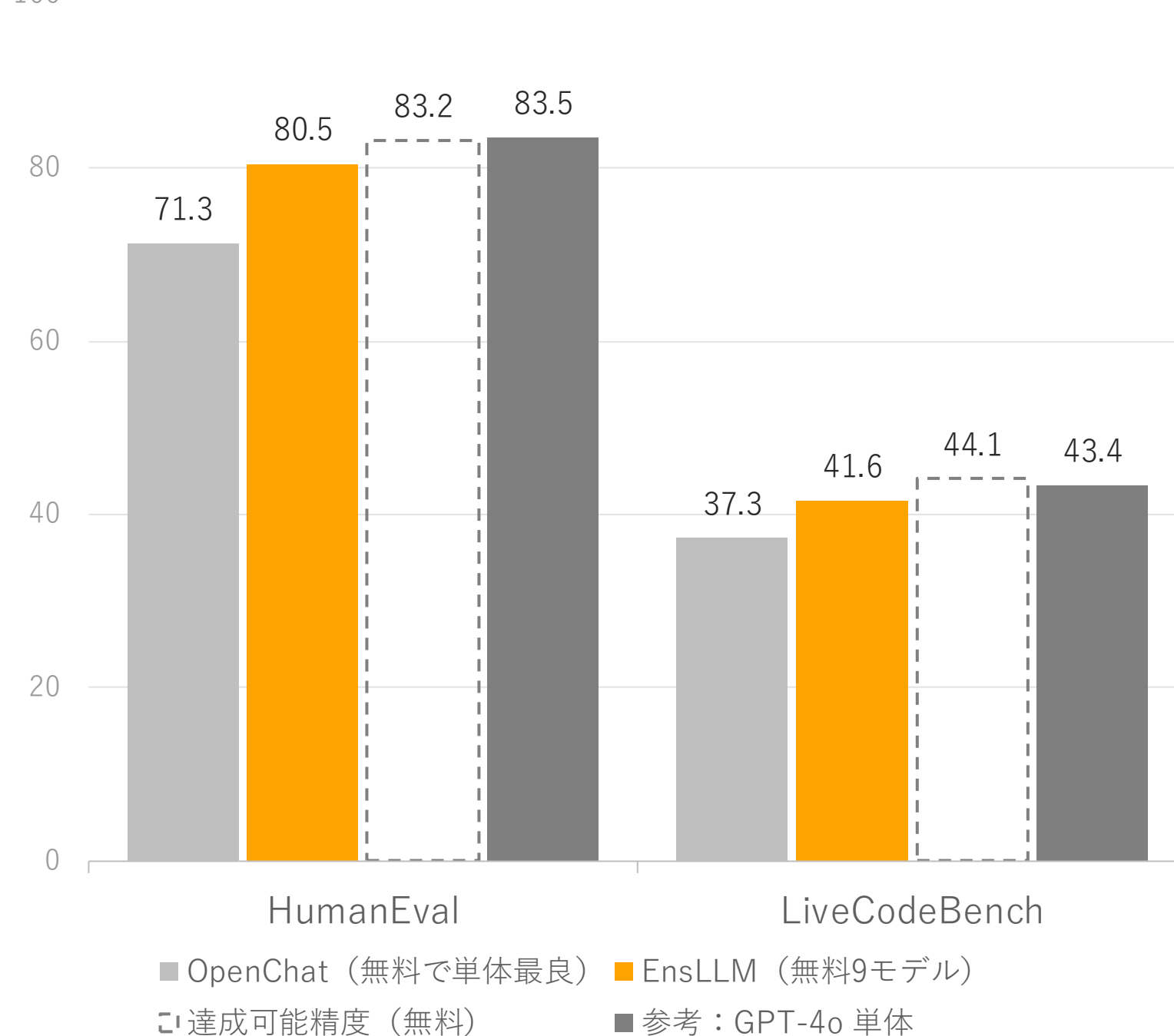
CodeBLEU と CrossHair の重みを変えて比較

λ	意味	HumanEval (%)	LiveCodeBench (%)
1	CodeBLEU のみ	85.9	49.5
0.75		87.2	49.5
0.5	半々 (採用)	90.2	50.2
0.25		89.6	48.4
0	CrossHair のみ	89.6	47.9

- 両方を半々で使うと、どちらのベンチマークでも最も高い
- CodeBLEU のみ：振る舞いの誤りを見逃す (HumanEval で **-4.3pt**)
- CrossHair のみ：LiveCodeBench で低下 (**-2.3pt**) → 構文・意味の情報も必要

結果 | RQ3 (無料 LLM のみ)

正解率 (%)



無料モデル

単体で最良の OpenChat から

+9.2pt / +4.3pt

(HumanEval / LiveCodeBench)

GPT-4o 単体 (83.5 / 43.4)
には届かない

→ 商用モデルを使えない
環境での選択肢

まとめ

複数のLLMの出力から一つを選ぶ手法を提案

CodeBLEU（構文・意味）とCrossHair（振る舞い）を組み合わせる

単体最良のモデルを上回った

性能の低いモデルも貢献していた

一問あたりの計算コストは高い（約6.5分）

