

Issue解決タスクにおけるIssue特徴を 考慮したルーティング手法の提案

オープンソースソフトウェア工学研究室
S2520048 陸井優希

背景

大規模言語モデルの利活用が一般化

- モデルの数も増加し続けており種類も豊富
オープンソースモデル共有プラットフォームで公開

Models 2,863,854

2026/05/10時点のモデル登録数

モデルの人気度などを参考に利用するモデルを選択

SE分野におけるLLM利用

コード生成



2つの整数を受け取り，合計を返す関数を書け



フィボナッチ数列のうち素数でもある
n番目の値を返す関数を書け

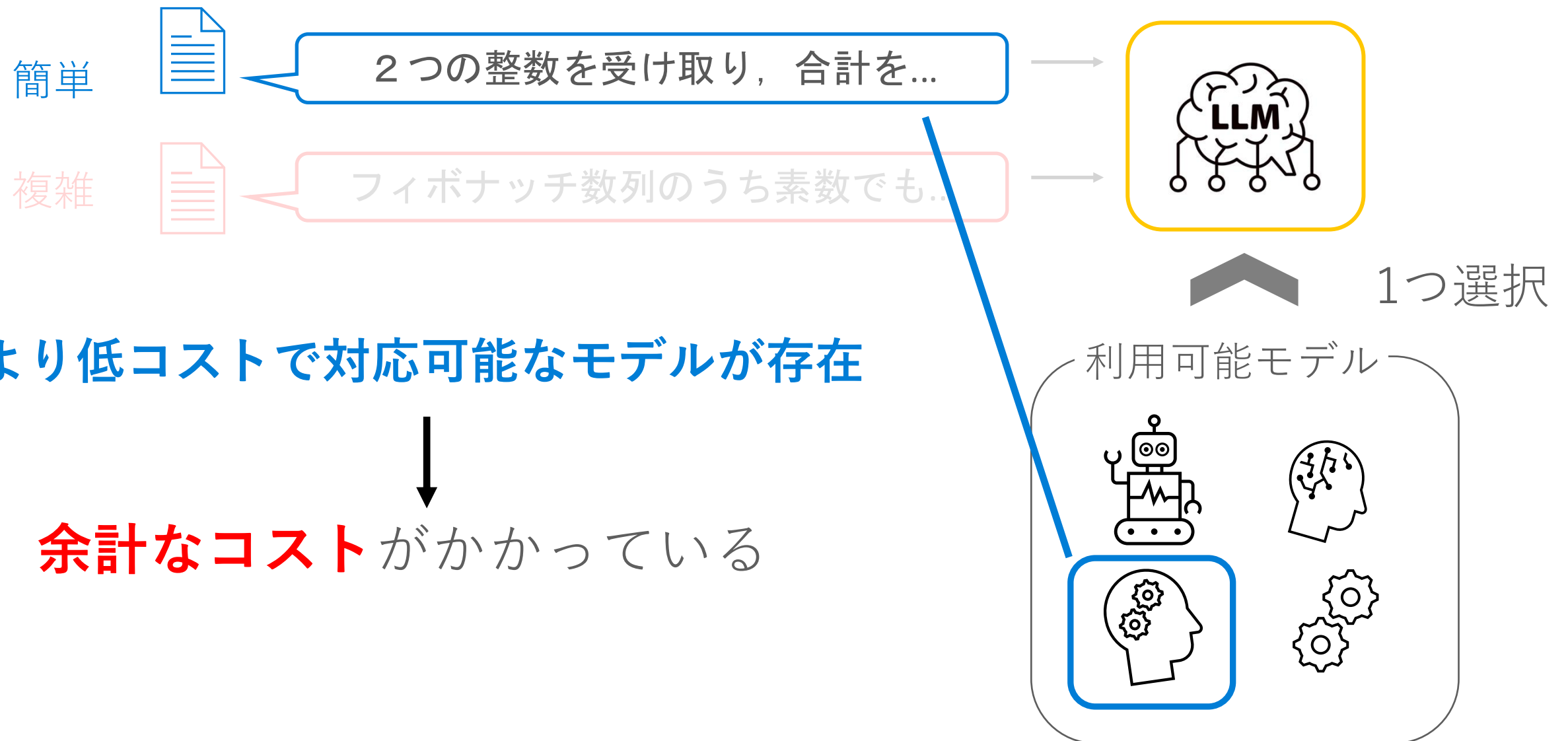
入力に関わらず一つのモデルで処理

SEタスク全般で同じ傾向^{[1][2][3][4]}

単一モデル利用の問題点

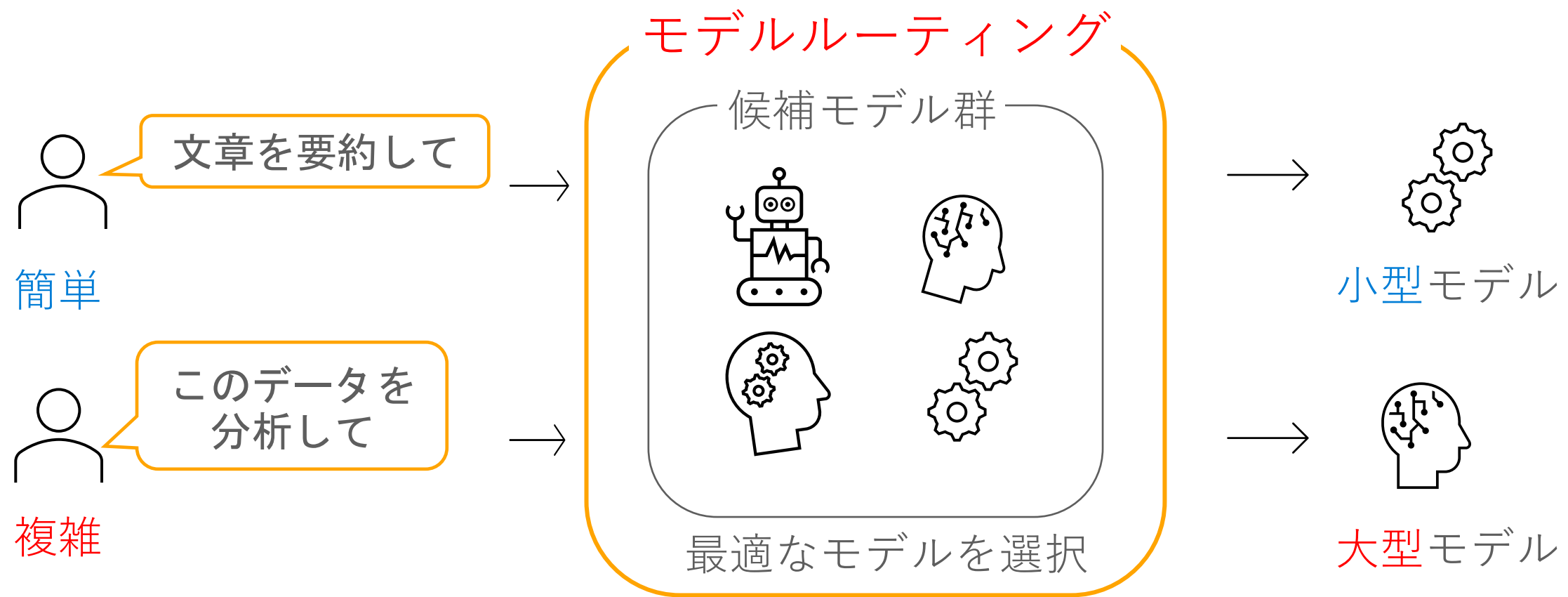
LLM利用にかかるコスト

- 高性能なモデルほど高価である傾向



NLP分野での解決策

クエリごとに最適なモデルを自動的に選択する技術



NLPタスクで複数の手法が提案・検証されている[5][6]

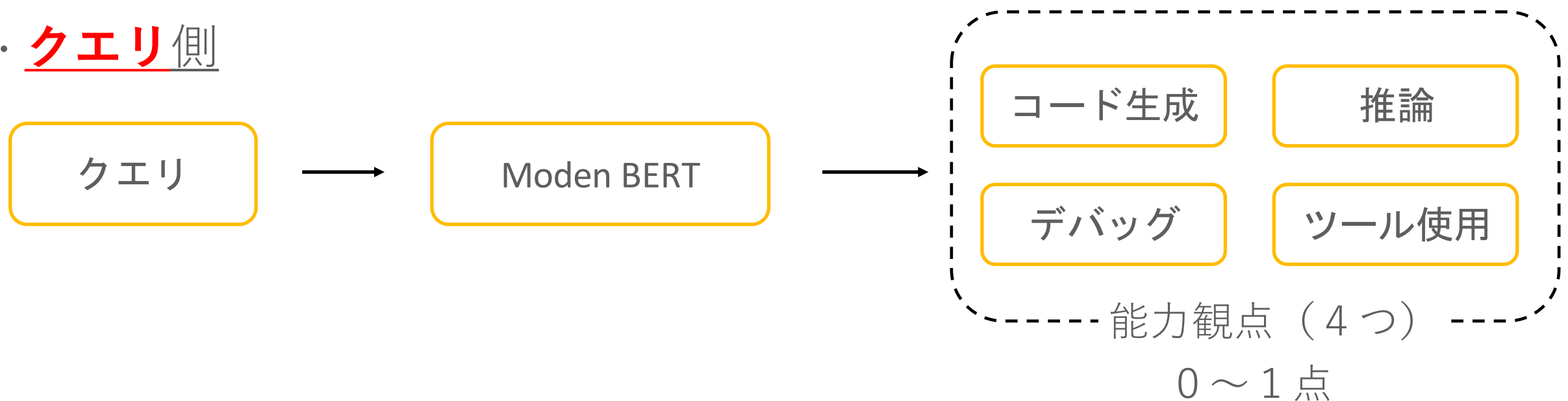
クエリごとに最適なモデルを選択することでコスト削減

SE分野でのモデルルーティング

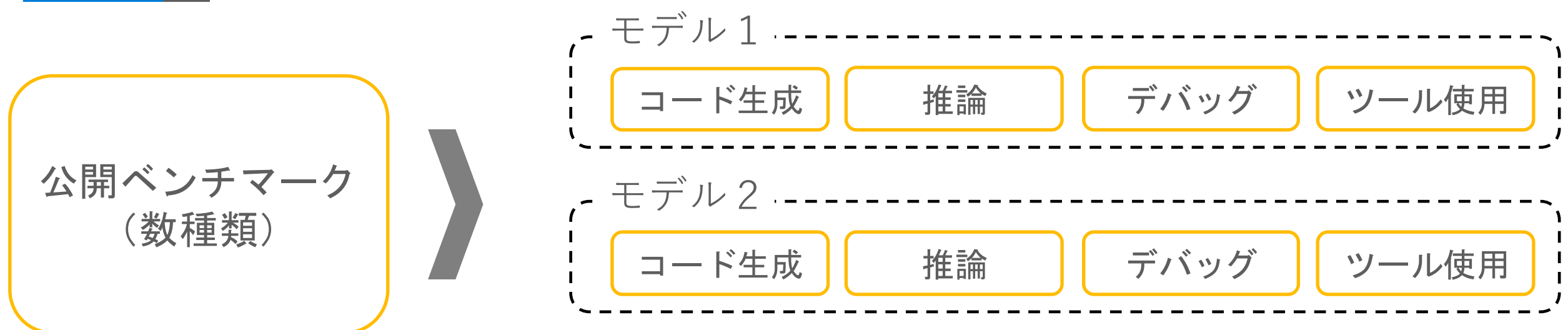
先行研究

- クエリが要求する能力の推定

・ クエリ側



・ モデル側



SE分野でのモデルルーティング

先行研究

- クエリが要求する能力の推定

クエリが要求する観点を満たす最も安いモデルを選択

コード生成

推論

デバッグ

ツール使用

能力観点（4つ）

0～1点

・ モデル側

公開ベンチマーク
(数種類)



モデル 1

コード生成

推論

デバッグ

ツール使用

モデル 2

コード生成

推論

デバッグ

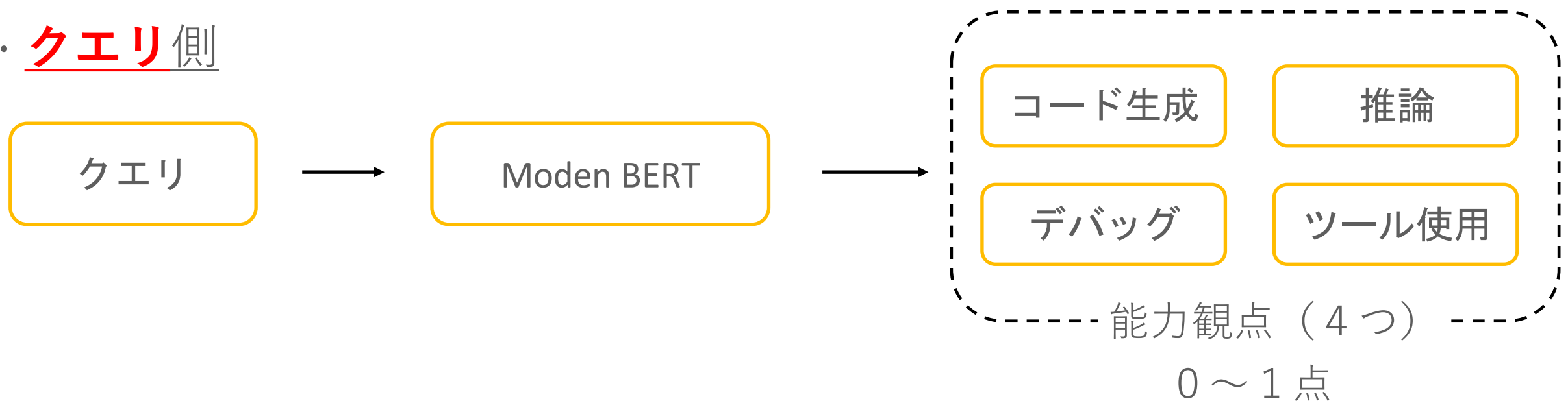
ツール使用

先行研究の問題点

先行研究

- クエリが要求する能力の推定

・ クエリ側



問題点

- クエリ（Issue）から4項目しか考慮していない

Issue記述の品質・完全性に関連する項目^{[8][9][10]}を含んでいない

└ 解決率に影響し得る

目的・アプローチ

目的

- Issue記述の品質・完全性を考慮したルーティングを提案する

アプローチ

- Issue特徴の調査

Issueを分析した既存研究を調査

- ルータを構築

調査で発見した特徴量を入力としたルータを構築

- 既存手法と比較

NLP分野の手法・SE分野の手法と精度・コストを比較

Issue特徴の調査

6本の論文から15の特徴を抽出

- バグ報告の品質・Issue記述の失敗分析に関する論文

特徴名	説明	出典
has_error_message	エラー語（Error/Exception/Warning）の有無	[8]
has_stack_trace	スタックトレースの有無	[9]
has_code_example	コード片（コードフェンス/REPL）の有無	[9]
has_test_case	テストコード（def test/assert/pytest 等）の有無	[10]
has_steps_to_reproduce	再現手順の有無	[10]/ [11]
has_expected_behavior	期待挙動の明示（"expected"）の有無	[10]
has_observed_behavior	観測挙動の明示（"actual"/"but got" 等）の有無	[10]
has_version_mention	バージョン情報の有無	[11]
n_info_items	8種の情報項目の充足数（0-8）	[12]

+ 6 特徴（詳細は参考資料）

ルータの構築

各候補モデルが正解する確率を予測



設定した閾値より高いモデルのうち安いモデルを選択（例 閾値 = 0.7, 価格順 : A > B > C）



評価実験

データセット

- SWE-bench Verified

評価指標

- 品質保持率 (QR)

$$QR = \frac{\text{ルータの解決数}}{\text{理想ルータの解決数}}$$

- コスト削減率 (CS)

$$CS = 1 - \frac{\text{ルータのコスト}}{\text{最良モデルのコスト}}$$

- 誤ルーティング率

より安いモデルで解けたのに高いモデルを選んだ割合

候補モデル

Together AI上のオープンソースモデル

- TogetherAI : オープンソースモデルをAPI経由で提供
ドルベースの料金比較が可能

モデル	正解率（500問中）	コスト
Qwen3-coder-480B	250 (50%)	\$1,394
DeepSeek-V3.1	241 (48.2%)	\$271
gpt-oss-120b	103 (20.6%)	\$ 28
gpt-oss-20b	13 (2.6%)	\$ 8
gemma-3n-E4B-it	4 (0.8%)	\$ 5

結果

手法	QR	CS	Mis
always-Qwen	78.86	0.00	0.48
理想ルータ	100.00	29.88	0.00
MetaLLM	75.71	14.87	0.44
RouterDC	76.66	60.73	0.27
IRT-Router	76.03	78.49	0.22
提案	81.70	30.66	0.35

always-Qwenより理想に近くコスト削減できた

まとめ・今後の予定

| Issueの特徴を文献から収集しルーティングに利用

| NLP分野で提案された手法と比較

単一モデル利用の精度を超えつつコスト削減を実現した

| 今後の予定

- SE分野のルーティングの再現および比較
- 候補モデルでは解けないと判断する機構の追加
さらにコスト削減を目指す

特徴名	説明	出典
has_error_message	エラー語（Error/Exception/Warning）の有無	[8]
has_stack_trace	スタックトレースの有無	[9]
has_code_example	コード片（コードフェンス/REPL）の有無	[9]
has_test_case	テストコード（def test/assert/pytest 等）の有無	[10]
has_steps_to_reproduce	再現手順の有無	[10]/ [11]
has_expected_behavior	期待挙動の明示（"expected"）の有無	[10]
has_observed_behavior	観測挙動の明示（"actual"/"but got" 等）の有無	[10]
has_version_mention	バージョン情報の有無	[11]
n_info_items	8種の情報項目の充足数（0-8）	[12]
itemization	箇条書き・番号付きリストの有無	[11]
patch_proposal	報告者の修正案・diff の有無	[9]
intermittency	非決定性を示す語（sometimes/flaky 等）の有無	[13]
stmt_len	問題文（problem_statement）の文字数	[14] / [9]
hints_text_len	PR 議論テキストの文字数	[15]
hints_text_present	PR 議論テキストの有無	[15]

参考文献

- [1] Dong et al., Self-collaboration code generation via ChatGPT, TOSEM 2024
- [2] Yuan et al., Evaluating and improving ChatGPT for unit test generation, FSE 2024
- [3] Xia et al., Automated program repair in the era of large pre-trained language models, ICSE 2023
- [4] Du et al., Vul-rag: Enhancing LLM-based vulnerability detection, 2024
- [5] MetaLLM: A High-performant and Cost-efficient Dynamic Framework for Wrapping LLMs Quang H. Nguyen et al., <https://arxiv.org/abs/2407.10834>
- [6] RouterDC: query-based router by dual contrastive learning for assembling large language models Chen, Shuhao et al., Proceedings of the 38th International Conference on Neural Information Processing Systems
- [7] IRT-Router: Effective and Interpretable Multi-LLM Routing via Item Response Theory Wei Song et al. ACL 2025 <https://aclanthology.org/2025.acl-long.761/>
- [8] BEST-Route: Adaptive LLM Routing with Test-Time Optimal Comput Dujian Ding et al. ICML 2025 <https://arxiv.org/abs/2506.22716>