

LLMを使ったソフトウェア開発において、トレードオフ における優先に応じて適したLLMを選択するための ベンチマークの分析

鳥飼優有

前回のあらすじ

LLMのコード生成能力と重要なプロジェクトの機能特性について関係を明らかにする方針

指摘:

- ・**LLM**のコード生成能力が既存の研究に基づいているのか
- ・重要なプロジェクトの機能特性をどのように判断するのか, 実際にあると言えるのか

背景(1/3):重要なプロジェクトの機能特性

現在ソフトウェアはあらゆる産業で活用されており、機能はユーザの利用目的によって様々

プロジェクトによって重要なソフトウェアの機能特性が異なる

速さと正確さはトレードオフの関係にあり、どちらがどれだけ優先されるかは重要なソフトウェアの機能特性によって異なる

極端な例

医療機器に使用されるソフトウェア → 正確な処理

ライブ中継に使用されるソフトウェア → 高速な処理

背景(2/3):LLMのコード生成能力

近年自然言語からコードを自動生成できる**LLM**の開発が盛ん
開発者は使用する**LLM**の選択を迫られるようになる

しかし

ハイパーパラメータや最適化手法などの実装細部が**LLM**の
性能に大きな影響を与える

トレーニングデータやアーキテクチャの概要といった表層の情報から**LLM**の性能を正確に予測することは困難

LLMのコード生成能力を評価することは重要な課題

背景(3/3):ベンチマーク

HumanEval

コード生成タスクにおける代表的なベンチマーク

164の評価タスクに対して正しい振る舞いのコードを生成できるかを評価

コード生成能力を包括的(※)に評価

※大学院の研究では包括的な評価ではないと主張する予定

重要なプロジェクトの機能特性に応じた**LLM**コード生成能力を測るベンチマークはない

LLMのコード生成能力と重要なプロジェクトの機能特性との関係の理解が重要

研究目的とアプローチ

研究目的: **HumanEval**で測定できる**LLM**のコード生成能力とトレードオフとの関係を明らかにする

アプローチ

① **HumanEval**の各評価タスクを評価する**LLM**のコード生成能力によって分類

② トレードオフにおける優先が極端なプロジェクトをそれぞれ選択しソースコードを抽出

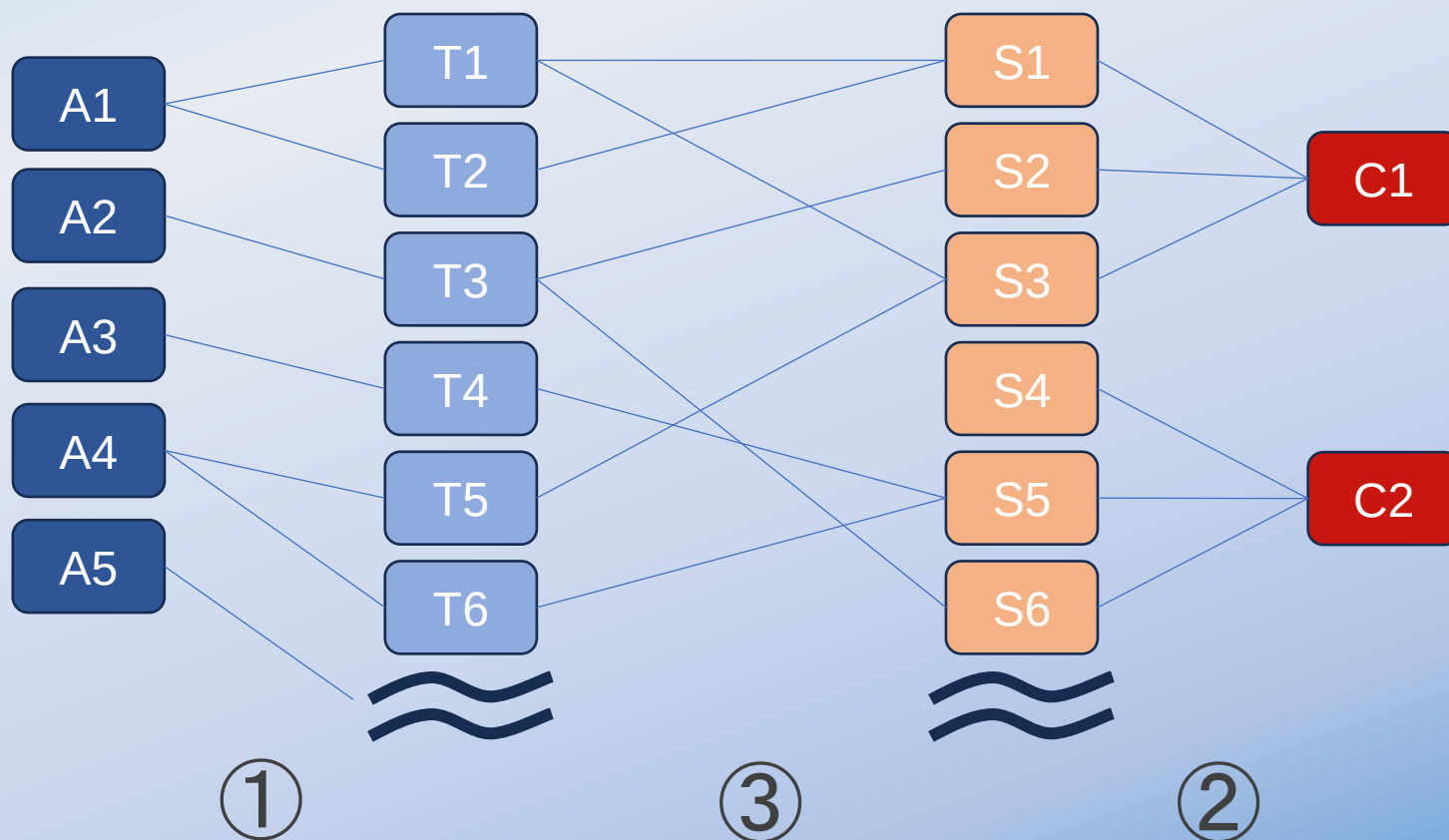
③ 評価タスクとソースコードを紐づけ

評価タスクに対して正しい振る舞いをするコードとソースコードの類似度が高いものなど

イメージ図

A:LLMのコード生成能力
T:ベンチマークの評価タスク

C:トレードオフにおける優先
S:ソフトウェアのソースコード



LLMのコード生成能力の分類

2パターンを検討

◆アルゴリズムやデータ構造による分類

どのようなアルゴリズムやデータ構造に適しているかといった生成能力に注目

“Classification and Recommendation Using CNNs”

畳み込みニューラルネットワークを使用し、プログラミングアプローチによる分類

◆コードのトピックモデリングによる分類

LLMの構造的に、トークンによって振る舞いが変わると考える

“Mining Source Code with Topic Models: Combining LDA with code analysis”

自然言語処理を使用し、コードのトピックによる分類

貢献と懸念点

貢献

HumanEvalの評価タスクにトレードオフにおける優先に応じた重みづけを行う事によって, トレードオフにおける優先に沿った**LLM**の選択が可能になる

懸念点