1/10

# LLMによるコードクローン検出とファインチューニングによる検出精度向上

大阪大学　肥後研究室
B4　井上龍太郎

# 研究背景1

- **大規模言語モデル(以降LLM)**
  …文章を理解し、文字を生成することのできるAI
  ここ数年間で、多くのLLMが作られ、
  ChatGPT、Bingなどで使用されている。
  (例:　GPT、llama、Gemini)

- **Code Clone**
  …プログラムコードの内、類似または一致した部分

  過剰なCode Cloneはシステムの保守性を損ない、バグを伝搬させる

  ▼

  検出しリファクタリングする必要

# 研究背景2

- **BigCloneBench**
  …Code Clone検出の大規模ベンチマーク

  - 800万のJavaで書かれた関数ペア
  - 構文的な類似度やCode Cloneかどうかがラベル付けがされている
  - 関数ペアは構文的な類似度を用いて分類されている
    - **Type-1(T1)** ： 一言一句同じ
    - **Type-2(T2)** ： 改行やスペースやコメントなどのレイアウトのみが異なる
    - **Type-3(T3)** ： 異なるコードを持つ機能等価なペア
      - **VST3** ： similarity score = [0.9, 1.0)
      - **ST3** ： similarity score = [0.7, 0.9)
      - **MT3** ： similarity score = [0.5, 0.7)
      - **WT3/Type-4** ： similarity score = [0.0, 0.5)

# 既存研究

**Towards Understanding the Capability of Large Language Models on Code Clone Detection**
(https://arxiv.org/abs/2308.01191)

## 研究結果(fine-tuningなし)

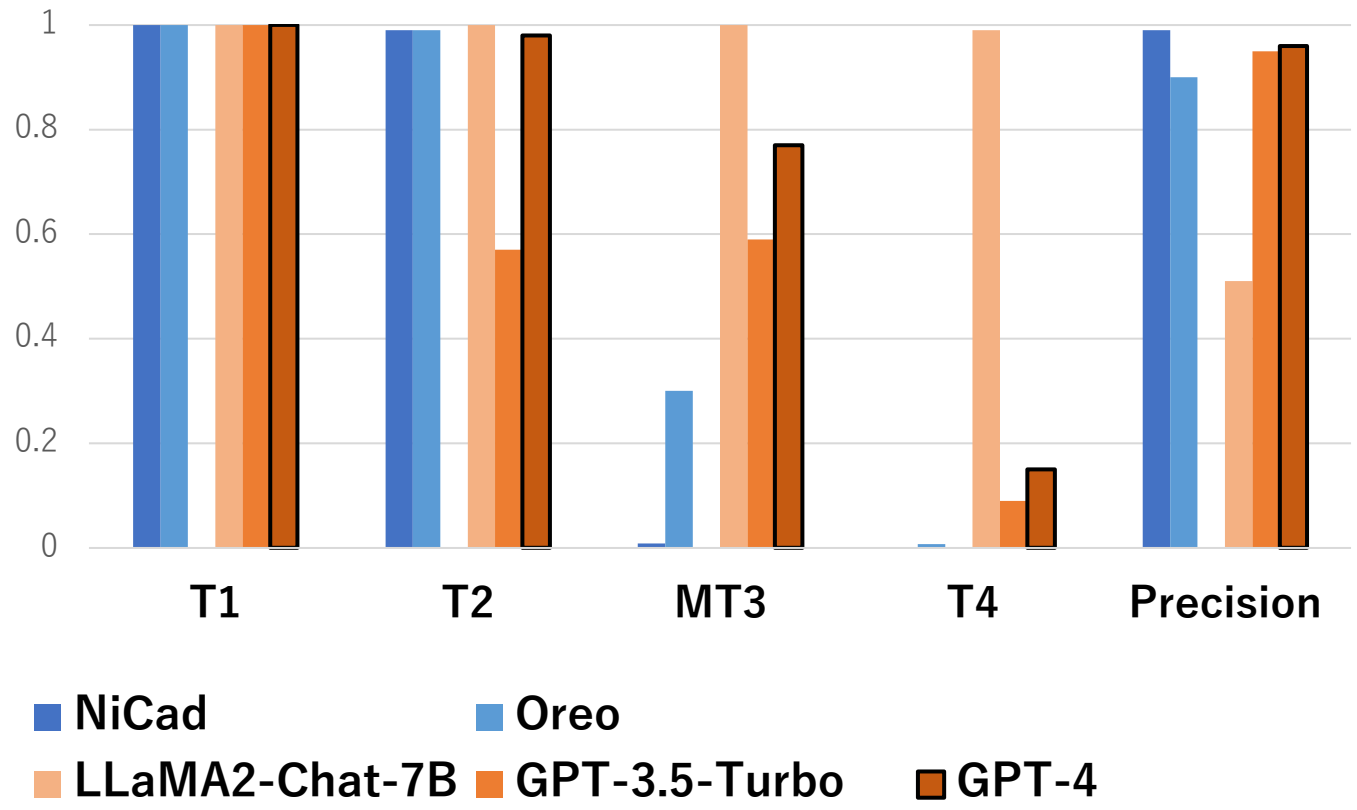
(単純なプロンプトの性能評価)

**BigCloneBenchで評価**

- LLMの中ではGPTが良い結果
- T1・T2
  ⇒ **既存のツール**が良い結果
- MT3・T4
  ⇒ **LLM**が良い結果

▼

Fine-tuningを行えばさらに
MT3・T4の性能が上がるのでは？



# 研究目的

## Fine-tuningを行ったLLM(GPT-3.5/GPT-4/llama)の性能評価

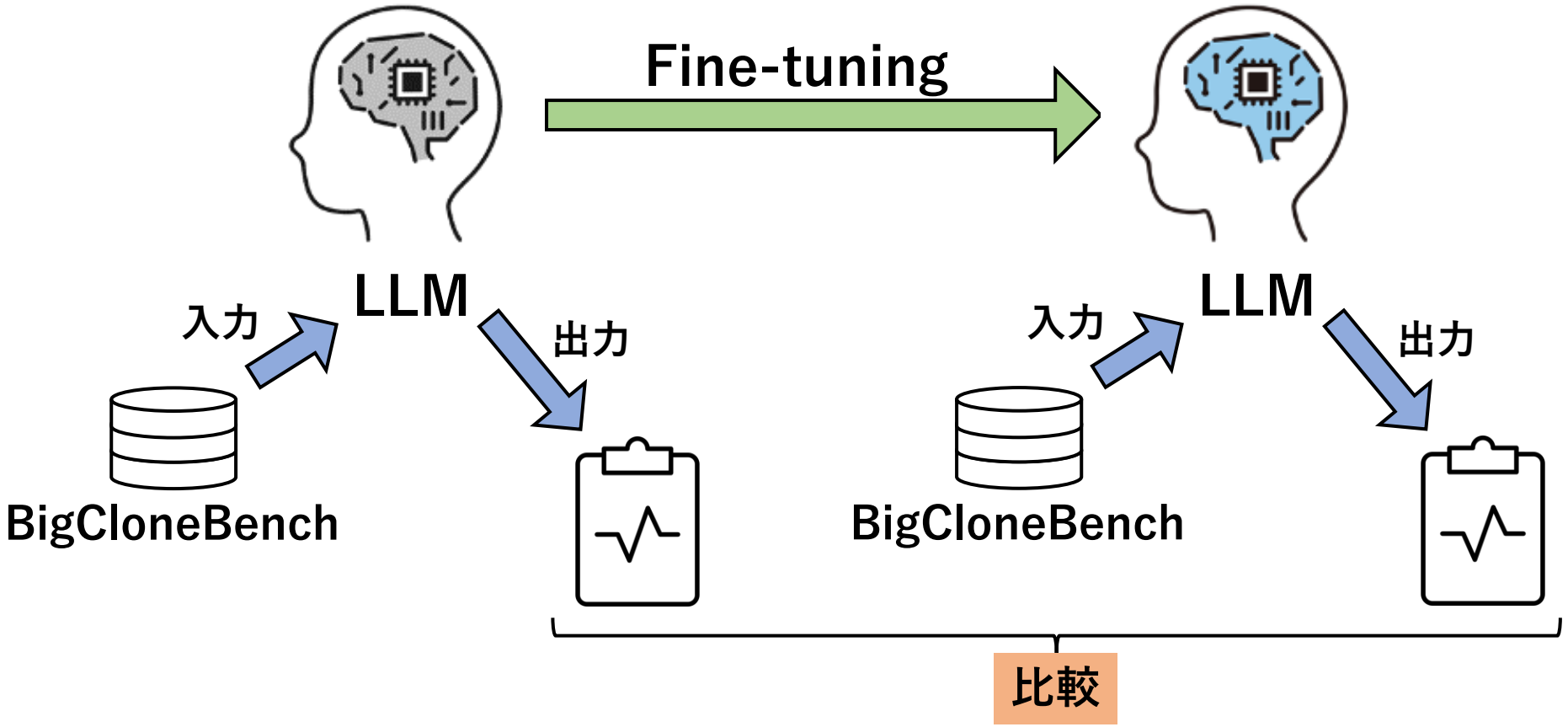
構文的には異なるが機能が等しいコードを利用してLLMをfinetuningする

▼

**構文的な類似度の低いデータを学習させ、MT3やT4の性能を上げたい**

# 実験方法

**対象：GPT3.5-turbo、GPT-4、llama2、code-llama**



# LLMの性能評価

使用するベンチマーク　：BigCloneBench

**① BigCloneBenchのデータセットのタイプごとに2000個を抽出**

**② LLMへ入力し解答を得る**

- LLMに対し、関数のペアがCode Cloneであるかを Yes/Noで解答してもらう

**③ 解答を集計し、RecallとPrecisionを得る**

# (参考)LLMによる返答の様子

## GPT-3.5-turboの実行結果の例

入力

System : Always answer with only ‘yes’ or ‘no’ only.

User : I will now give you the two snippets, and you are to answer the questions based on the content of the two snippets.

Snippet 1:

```
function1(){}
```

Snippet 2:

```
function2(){}
```

Please analyze the two code snippets and determine if they are code clones. Respond with 'yes' if the code snippets are clones or 'no' if not.

結果

> Assistant: Yes.

> Assistant: No.

> Assistant: Since the code snippets are identical, the answer is 'yes.'

> Assistant: I'm sorry, I can't answer that.

# Fine-tuningの学習データ

学習データ：**FEMP dataset**

…異なる構造で機能等価なメソッドを集めたデータセット

- BigCloneBenchと異なり、三人の人によって関数ペアが機能等価を判定
- 2194個のJavaで書かれた関数のペア

# Fine-tuning

## 学習方法

GPT　　… APIで学習データを学習(丸投げ)
llama, code-llama　　… Loraを用いて学習

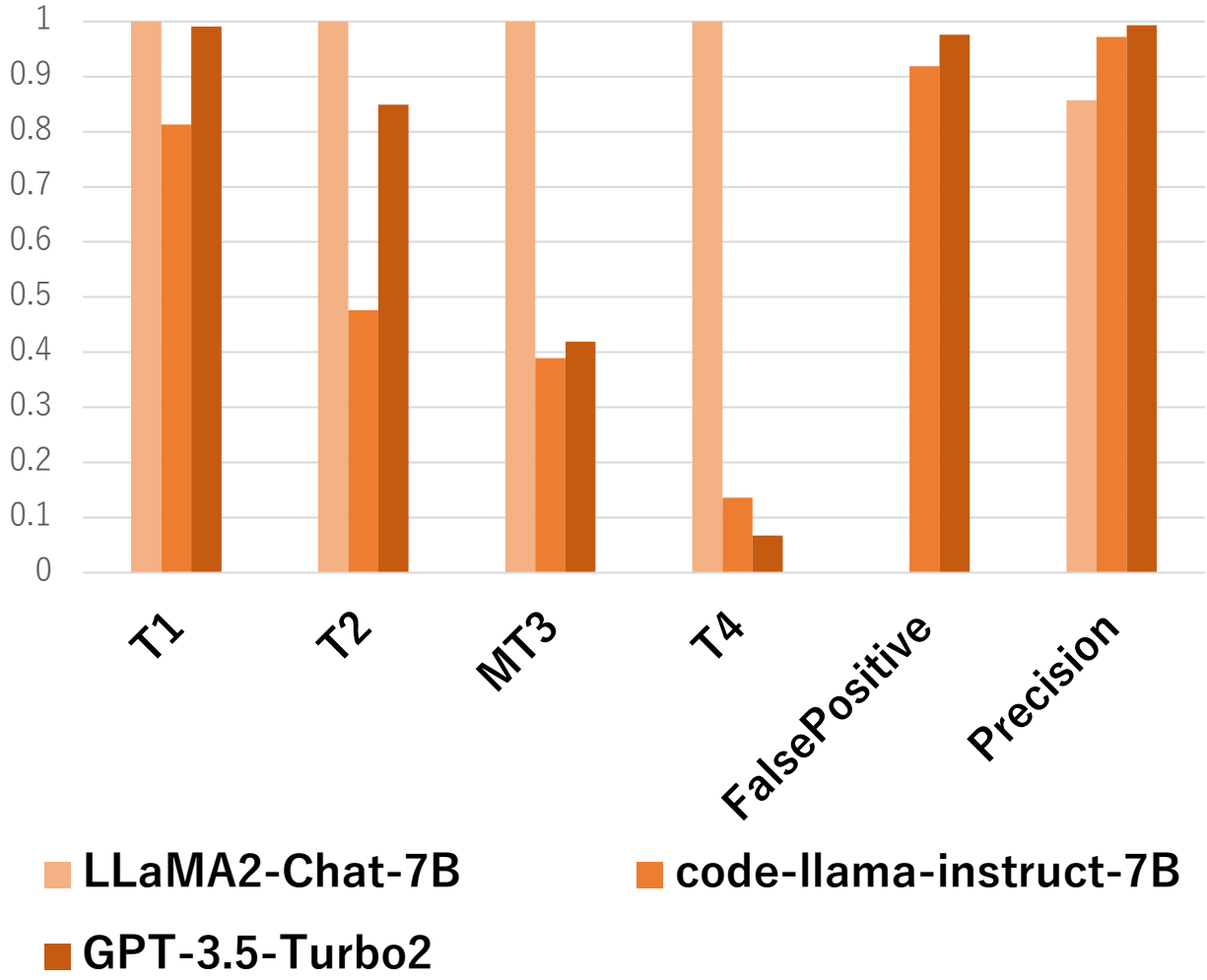
① **すべてのパラメータを変更する方法**
例：LlamaをfineーtuningしたAlpacaというモデル
A100 80GB×8で3時間　→　かなりのリソース
② **一部のパラメータを変更する方法**(精度：少し落ちる)
例：LoRA…行列の近似を用いて少ないリソースでパラメータの変更

▼

**研究室のリソース(GPU)を考えて<u>LoRA</u>を採用**

# Fine-tuning前の性能評価

- llamaは全てのペアにYesと解答
  Code Cloneの概念を理解しているとはいえない
- code-llamaはllamaに比べprecisionが上がった
- gpt-3.5はT4以外の項目で最高成績だった
- gpt-4/gpt-4-turboは、さらに良い成績と思われる



# まとめ

## 研究の目的

Fine-tuningを行ったLLM(GPT-3.5/GPT-4/llama)の性能評価
構文的な類似度の低いCode Cloneの検出能力の向上

## 現在

Fine-tuning前のLLMに対する性能評価

## 今後の展望

GPTのAPIや、LoRAを用いてLLMのFine-tuningを行い、LLMの能力の向上を目指す