大規模言語モデルを用いた なりすましプログラムの判定

情報工学科４年
大枝研究室　高橋　琉生

背景

ソースコードの特徴分析の先行研究では大野らがソースコードから得られる特徴を確率モデルで抽出し，そのモデルに学習させることで執筆者の識別などに応用している．

https://chatgpt.com/

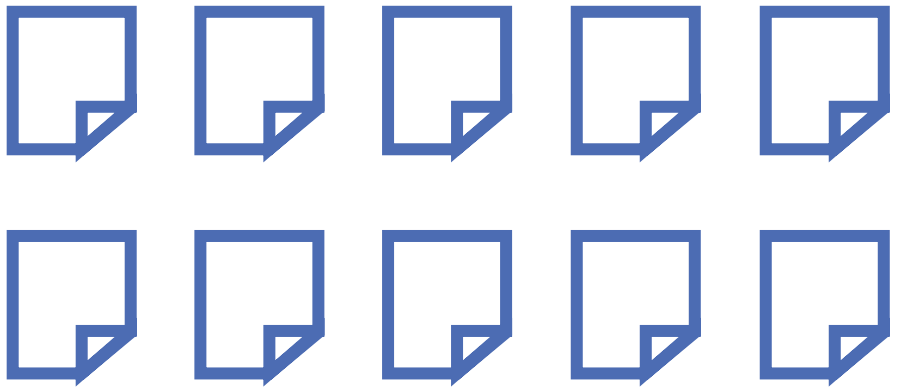
ChatGPTなどの普及により，直接人間がコードを書かない状況が増えている．

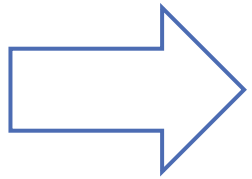
https://claude.ai/

ソースコードの執筆者固有の癖を大規模言語モデル（LLM）で捉えるアプローチを提案する．

目的

対象者を１人決める
対象者の過去のソースコードを収集し，傾向・特徴を分析
新たに与えられたソースコードが対象者のものか判定する





https://chatgpt.com/

方法

使用モデルはChatGPT 4o

実験の流れ

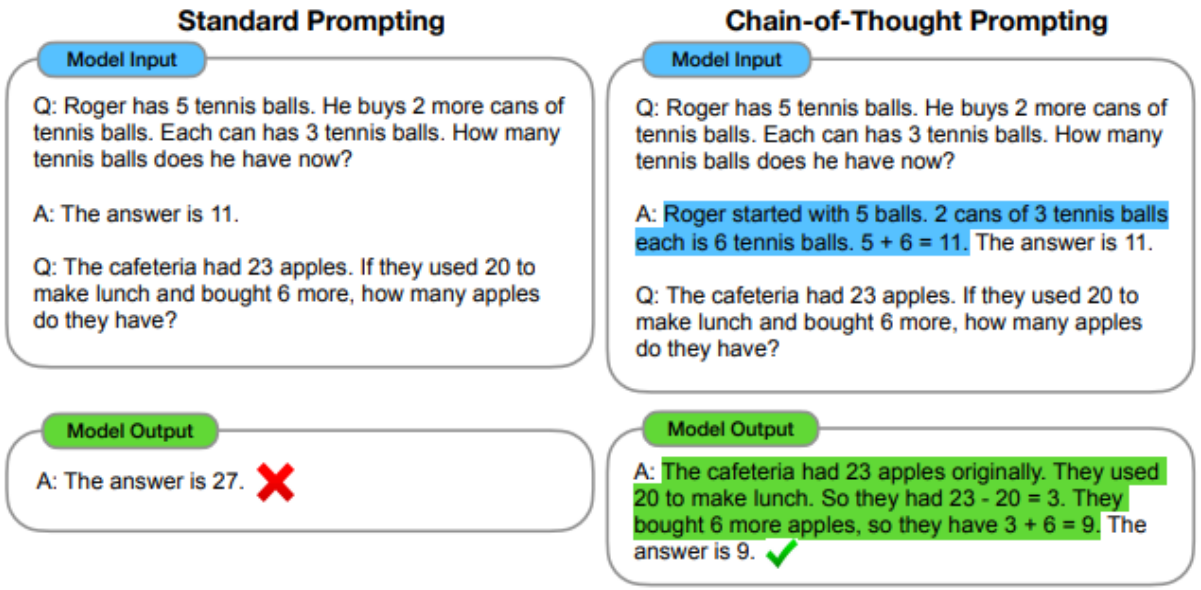
１．プロンプトに特徴の分析方法を与える

２．１人のソースコードを複数与える

３．新たにソースコードを渡し、判定させる

1．プロンプトに特徴の分析方法を与える

Chain of Thought（思考の連鎖，CoT）
考え方を与えてから回答させる



ソースコードの評価項目

1．コーディングスタイル・フォーマット
2．コメントやドキュメンテーションの特徴
3．コード設計上の特徴
4．構文・言語機能の利用傾向
5．コードの正確性
6．上記以外の特筆すべき特徴

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models　より引用

プロンプトの一部

複数のC言語のソースコードが同一人物によって書かれたものかを判別するために，段階的に判断するプロセスを考えてください．以下の手順で答えてください：

1.コーディングスタイル・フォーマット：同一人物かを判断するために，まずコーディングスタイルにどのような違いや特徴が表れるかを考えてください（例：インデントやスペースの使い方，波括弧や括弧の配置ルール，命名規則）

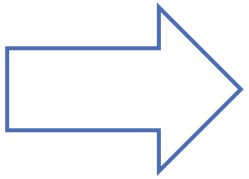
2.コメントやドキュメンテーションの特徴： 文章にもどのような違いや特徴が表れるかを考えてください（例：コメント文の文体や言語，コメント量や記法の統一性）

２．１人のソースコードを複数与える

・C言語の問題を５個
・それを解いた対象者のソースコードを５個
　コーディングスタイルを統一

問題

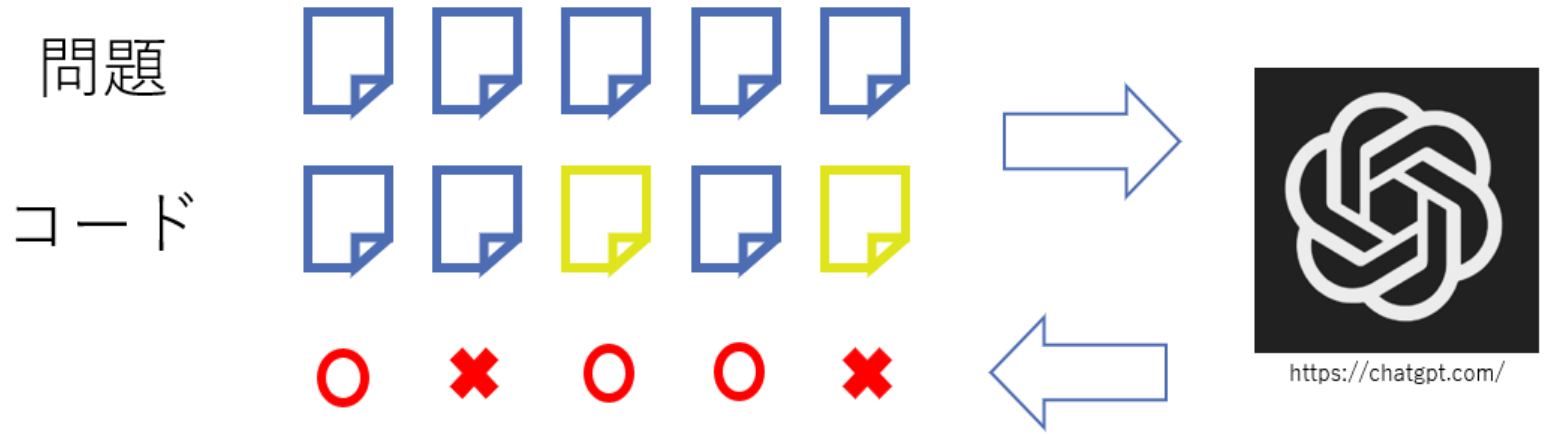
コード



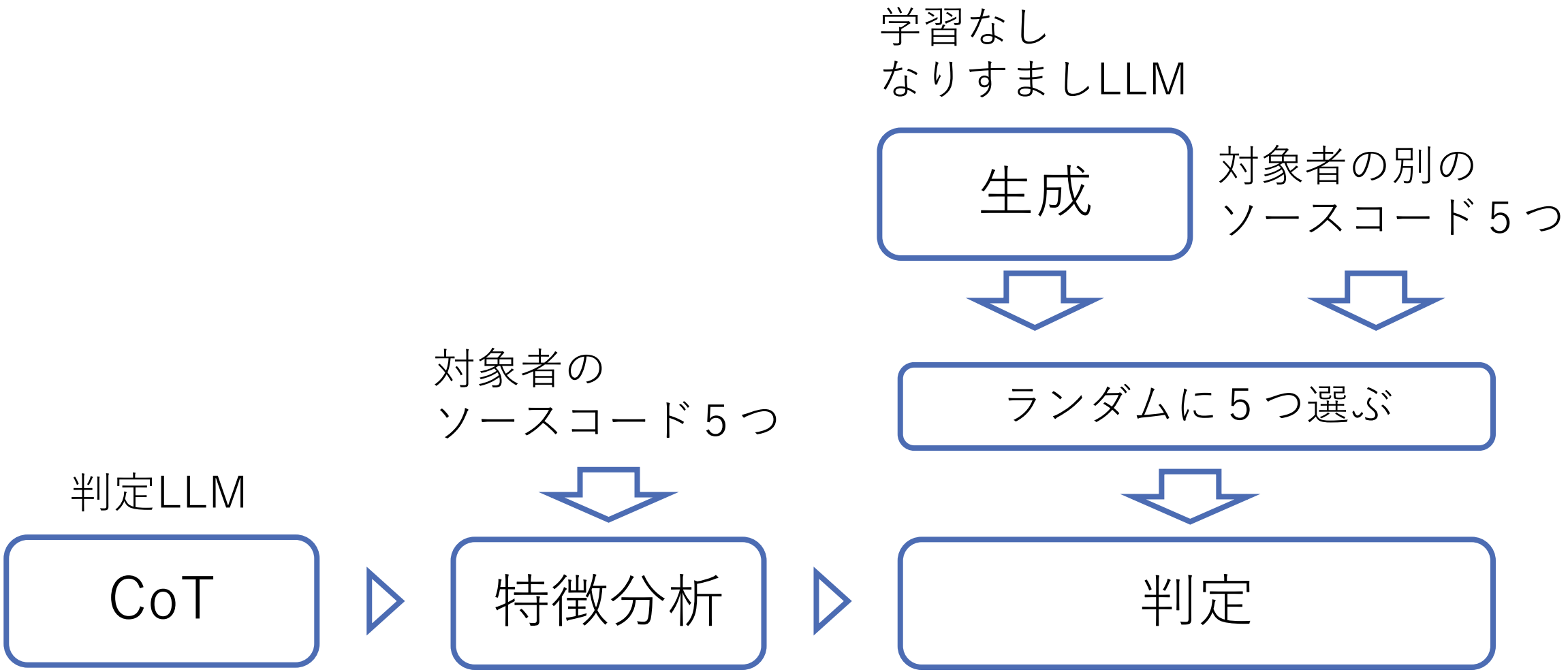
https://chatgpt.com/

３．新たにソースコードを渡し、判定させる

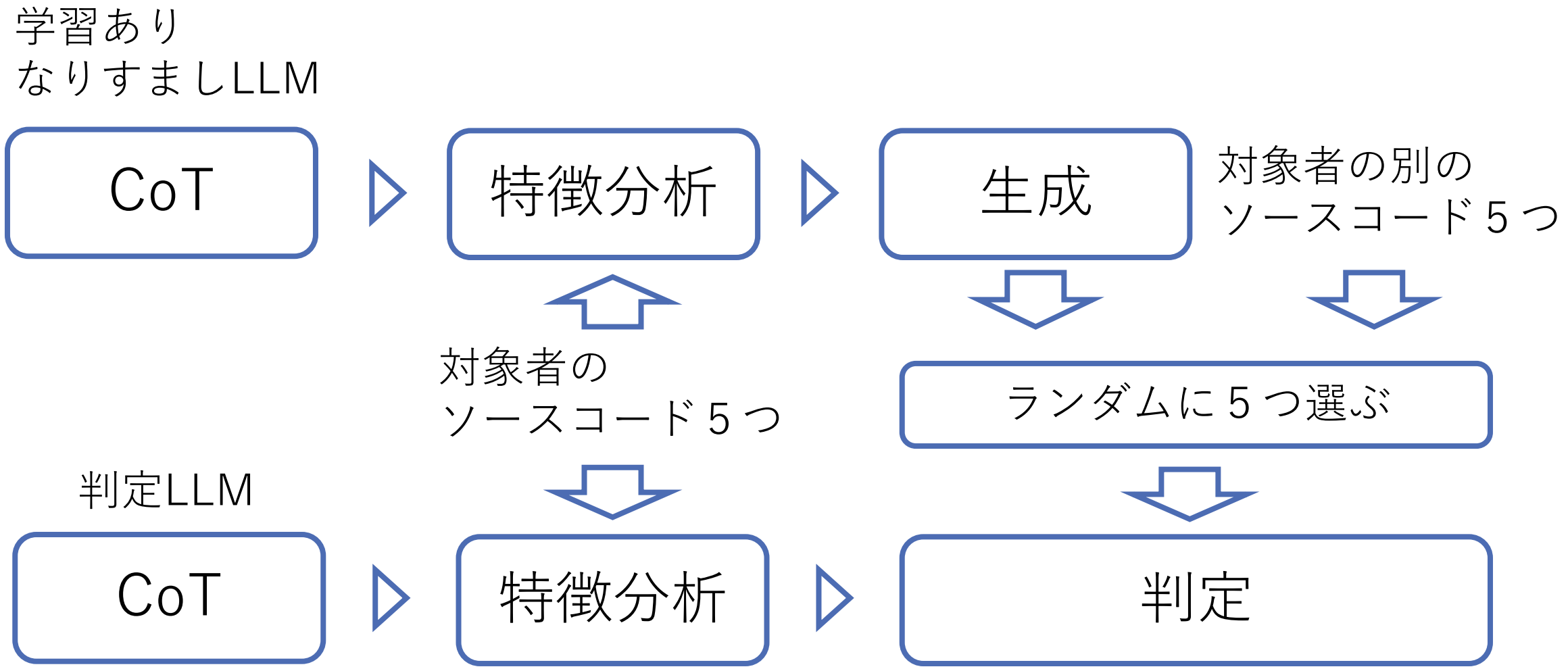
・C言語の問題５つとソースコード５個を渡し，
　正しく判定できるか
　・対象者のコード
　・なりすまし（学習なし）
　・なりすまし（学習あり）



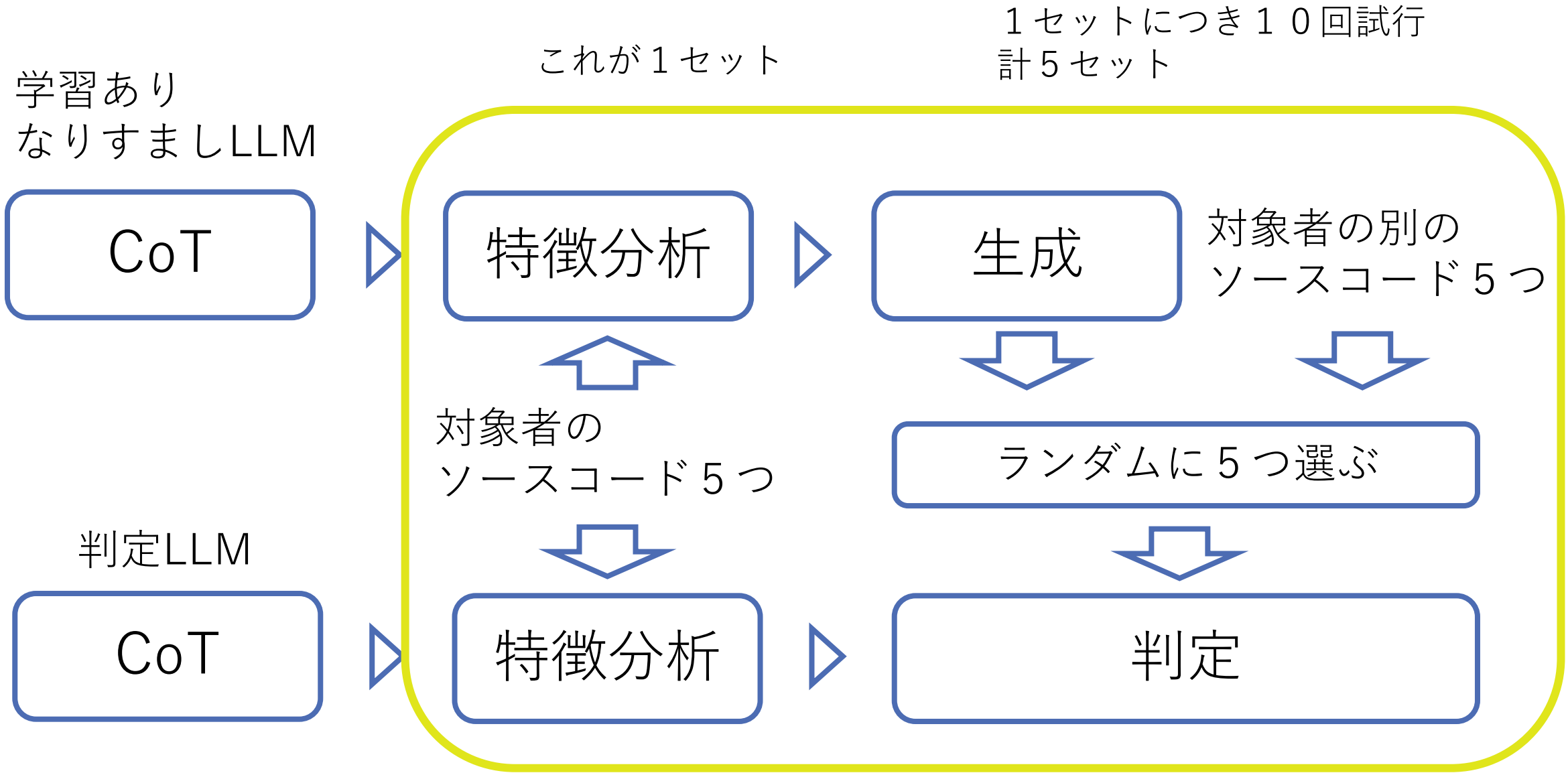
方法


学習なし
なりすましLLM
生成
対象者の別の
ソースコード５つ
ランダムに５つ選ぶ
対象者の
ソースコード５つ
判定LLM
CoT
特徴分析
判定

方法


学習あり
なりすましLLM
CoT
特徴分析
生成
対象者の別の
ソースコード５つ
対象者の
ソースコード５つ
ランダムに５つ選ぶ
判定LLM
CoT
特徴分析
判定

方法


学習あり
なりすましLLM
CoT
これが１セット
１セットにつき１０回試行
計５セット
特徴分析
生成
対象者の別の
ソースコード５つ
対象者の
ソースコード５つ
ランダムに５つ選ぶ
判定LLM
CoT
特徴分析
判定

方法

対象者のコード

```c
#include <stdio.h>

int main() {

    int num;
    scanf("%d", &num);

    if (num < 0) {
        printf("%dは負の数\n", num);
    } else if (num % 2 == 0) {
        printf("%dは偶数\n", num);
    } else {
        printf("%dは奇数\n", num);
    }

    return 0;
}
```

学習ありのコード

```c
#include <stdio.h>

int main() {

    int num;
    scanf("%d", &num);

    if (num < 0) {
        printf("負の数です\n");
    } else if (num % 2 == 0) {
        printf("偶数です\n");
    } else {
        printf("奇数です\n");
    }

    return 0;
}
```

学習なしのコード

```c
#include <stdio.h>

int main() {
    int number;

    // ユーザーに整数を入力してもらう
    printf("整数を入力してください: ");
    scanf("%d", &number);

    // 負の数の場合
    if (number < 0) {
        printf("負の数です。\n");
    }
    // 偶数の場合
    else if (number % 2 == 0) {
        printf("偶数です。\n");
    }
    // 奇数の場合
    else {
        printf("奇数です。\n");
    }

    return 0;
}
```

なりすまし（学習なし）

	正解率(%)	偽陽性率(%)	偽陰性率(%)
セット1	66.0	33.3	35.0
セット2	72.0	50.0	13.3
セット3	72.0	40.0	22.9
セット4	70.0	40.0	6.7
セット5	36.0	88.0	40.0
合計	63.2	49.6	24.0

偽陽性率：出題したなりすましソースコードの数に対する，なりすましのソースコードを対象者のソースコードだと判定した割合
偽陰性率：出題した対象者のソースコードの数に対する，対象者のソースコードをなりすましのソースコードだと判定した割合

結果

なりすまし（学習なし）

	正解率(%)	偽陽性率(%)	偽陰性率(%)
セット1	66.0	33.3	35.0
セット2	72.0	50.0	13.3
セット3	72.0	40.0	22.9
セット4	70.0	40.0	6.7
セット5	36.0	88.0	40.0
合計	63.2	49.6	24.0

なりすまし（学習あり）

	正解率(%)	偽陽性率(%)	偽陰性率(%)
セット1	42.0	96.7	0.0
セット2	48.0	85.0	30.0
セット3	56.0	86.7	25.7
セット4	34.0	82.3	26.7
セット5	24.0	100.0	52.0
合計	40.8	90.4	28.0

偽陽性率：出題したなりすましソースコードの数に対する，なりすましのソースコードを対象者のソースコードだと判定した割合
偽陰性率：出題した対象者のソースコードの数に対する，対象者のソースコードをなりすましのソースコードだと判定した割合

まとめ

成果

学習なしならある程度なら判別はできる
ユーザーが偽造しようと思えばできてしまう

問題点

サンプルが少ないと正確に特徴を分析できない
ChatGPTは文脈を覚えておけない

今後の展望

与えるプロンプトや用いるLLMを変え，精度の向上
AtCoderのデータでも行う

参考文献

大野 麻子, 村尾 元. “確率モデルによるソースコード記述スタイルの識別”, 第53回システム制御情報学会研究発表講演会

・Chain-of-Thought Prompting Elicits Reasoning in Large Language Models
https://arxiv.org/pdf/2201.11903

・Holistic Evaluation of Language Models (HELM)
https://crfm.stanford.edu/helm/lite/latest/#/leaderboard

HELM Leaderboard

The HELM leaderboard shows how the various models perform across different scenarios and metrics.

Accuracy | Efficiency | General information

Model	Mean win rate	NarrativeQA - F1	NaturalQuestions (open) - F1	NaturalQuestions (closed) - F1
GPT-4o (2024-05-13)	**0.938**	**0.804**	0.803	0.501
GPT-4o (2024-08-06)	0.928	0.795	0.793	0.496
DeepSeek v3	0.908	0.796	0.765	0.467
Claude 3.5 Sonnet (20240620)	0.885	0.746	0.749	**0.502**
Amazon Nova Pro	0.885	0.791	**0.829**	0.405
GPT-4 (0613)	0.867	0.768	0.79	0.457
GPT-4 Turbo (2024-04-09)	0.864	0.761	0.795	0.482
Llama 3.1 Instruct Turbo (405B)	0.854	0.749	0.756	0.456
Claude 3.5 Sonnet (20241022)	0.846	0.77	0.665	0.467