GitHub Copilotを用いた コード推薦における入力言語の影響調査

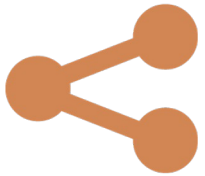
2023年11月11日

○小栁 慶，野口 広太郎，王 棟，近藤 将成，亀井 靖高，鵜林 尚靖

九州大学
KYUSHU UNIVERSITY

研究の背景: 支援ツールを活用した開発

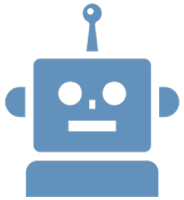
Trello
(タスク管理)



git
(バージョン管理)

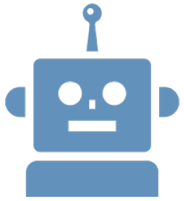
Visual Studio
(統合開発環境)

研究の背景: 支援ツールを活用した開発



GitHub Copilot　(コーディング支援)

研究の背景: 支援ツールを活用した開発



GitHub Copilot　(コーディング支援)

入力

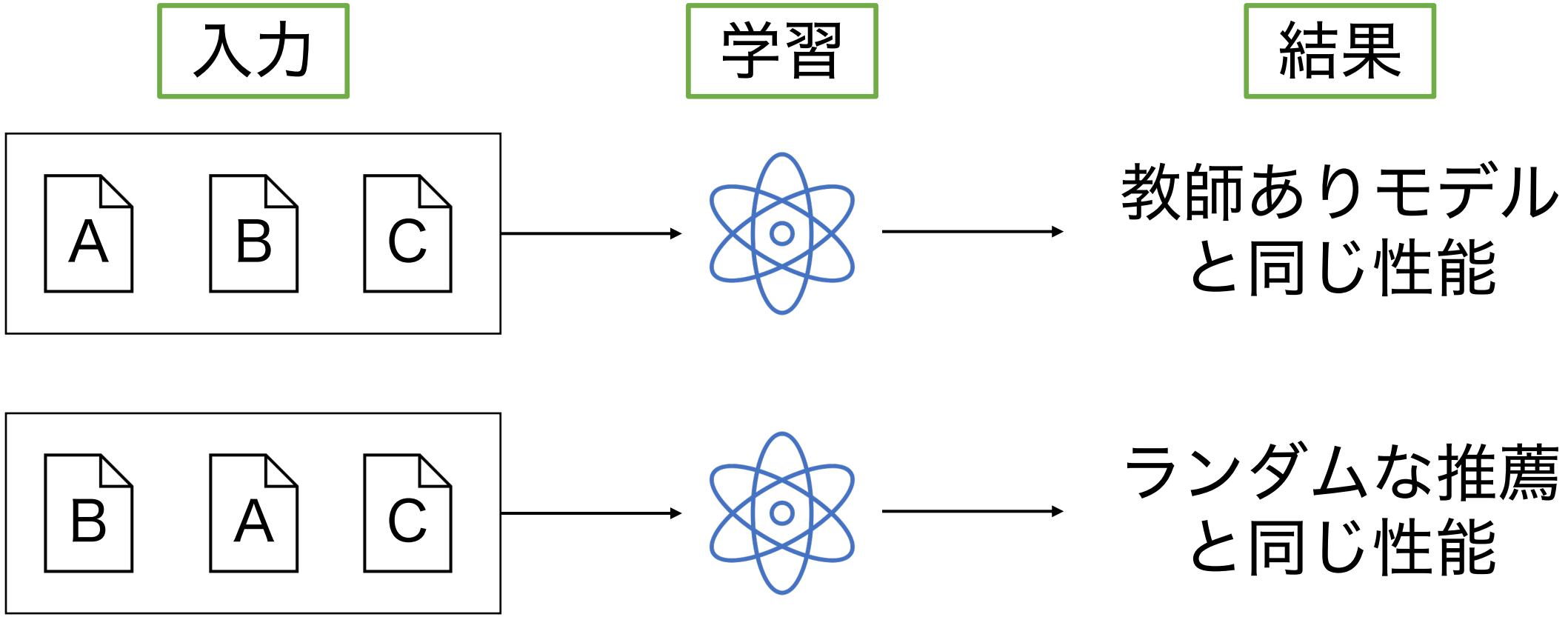
推薦

```python
import datetime

def parse_expenses(expenses_string):
    """Parse the list of expenses and return the list of triples (date, value, currency).
    Ignore lines starting with #.
    Parse the date using datetime.
    Example expenses_string:
        2016-01-02 -34.01 USD
        2016-01-03 2.59 DKK
        2016-01-03 -2.72 EUR
    """
    expenses = []
    for line in expenses_string.splitlines():
        if line.startswith("#"):
            continue
        date, value, currency = line.split(" ")
        expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
                         float(value),
                         currency))
    return expenses
```

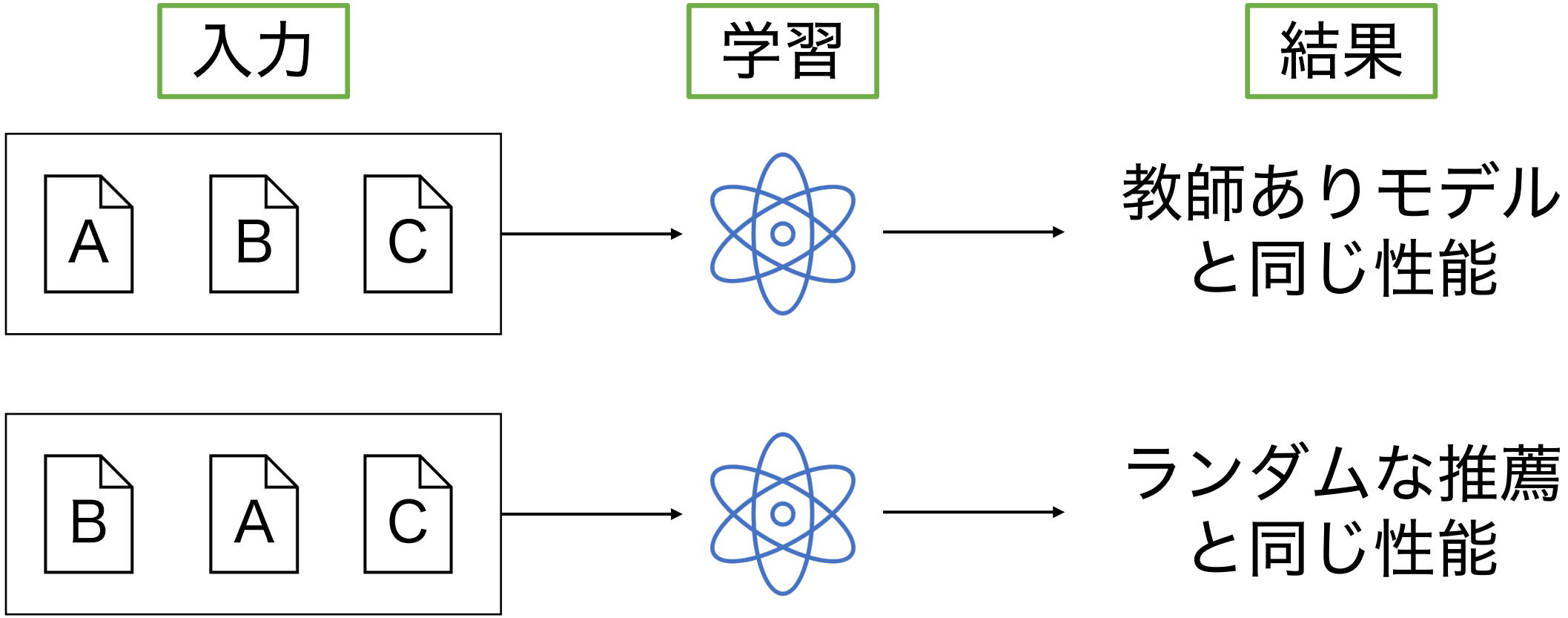
Copilot

研究の背景: 入力順序による出力の違い[1]



[1] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. . In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

研究の背景: 入力順序による出力の違い



言語モデルの能力を最大限引き出すには適切な入力が必要

研究の背景: 多種多様な言語

世界には7,000以上の言語が存在[2]

[2] https://www.ethnologue.com/

研究の背景: 多種多様な言語

世界には7,000以上の言語が存在[2]

1.3億人　　15億人　　11億人

各言語を話せる人口は異なる

[2] https://www.ethnologue.com/

研究の背景: 多種多様な言語

世界には7,000以上の言語が存在[2]

こんに

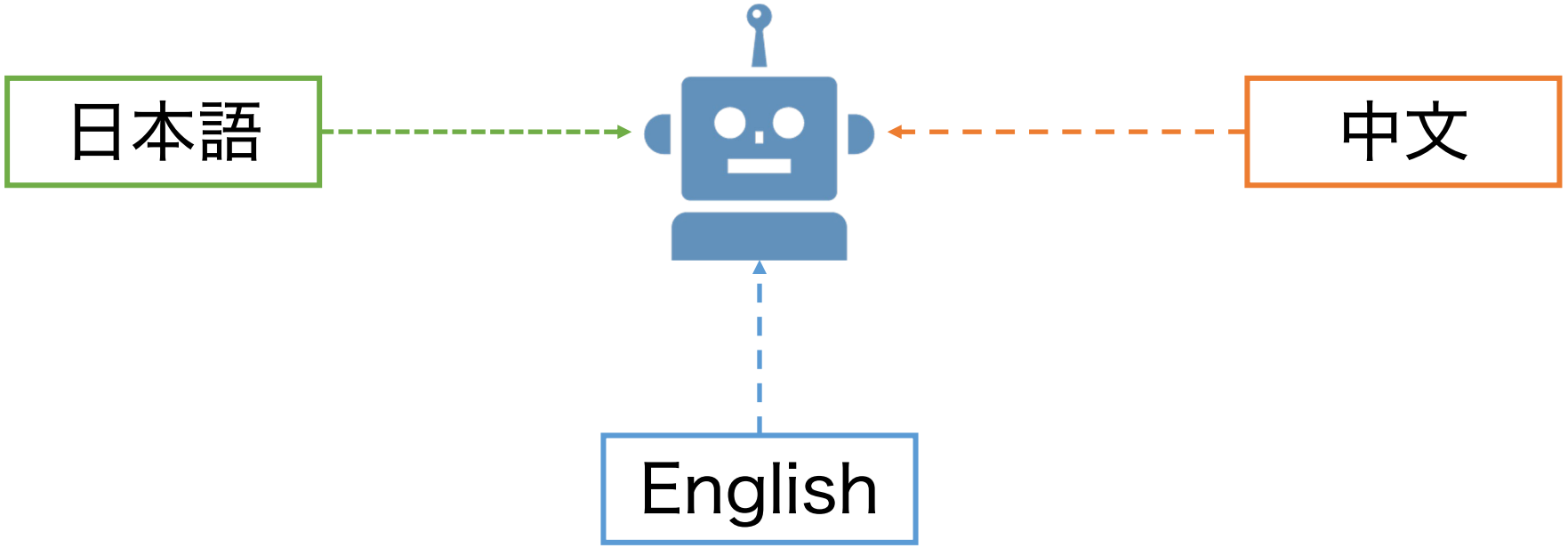
入力に異なる言語を使用することで,
学習データ数の不均衡などにより
学習結果のバイアスにつながる可能性がある

各言語を話せる人口は異なる

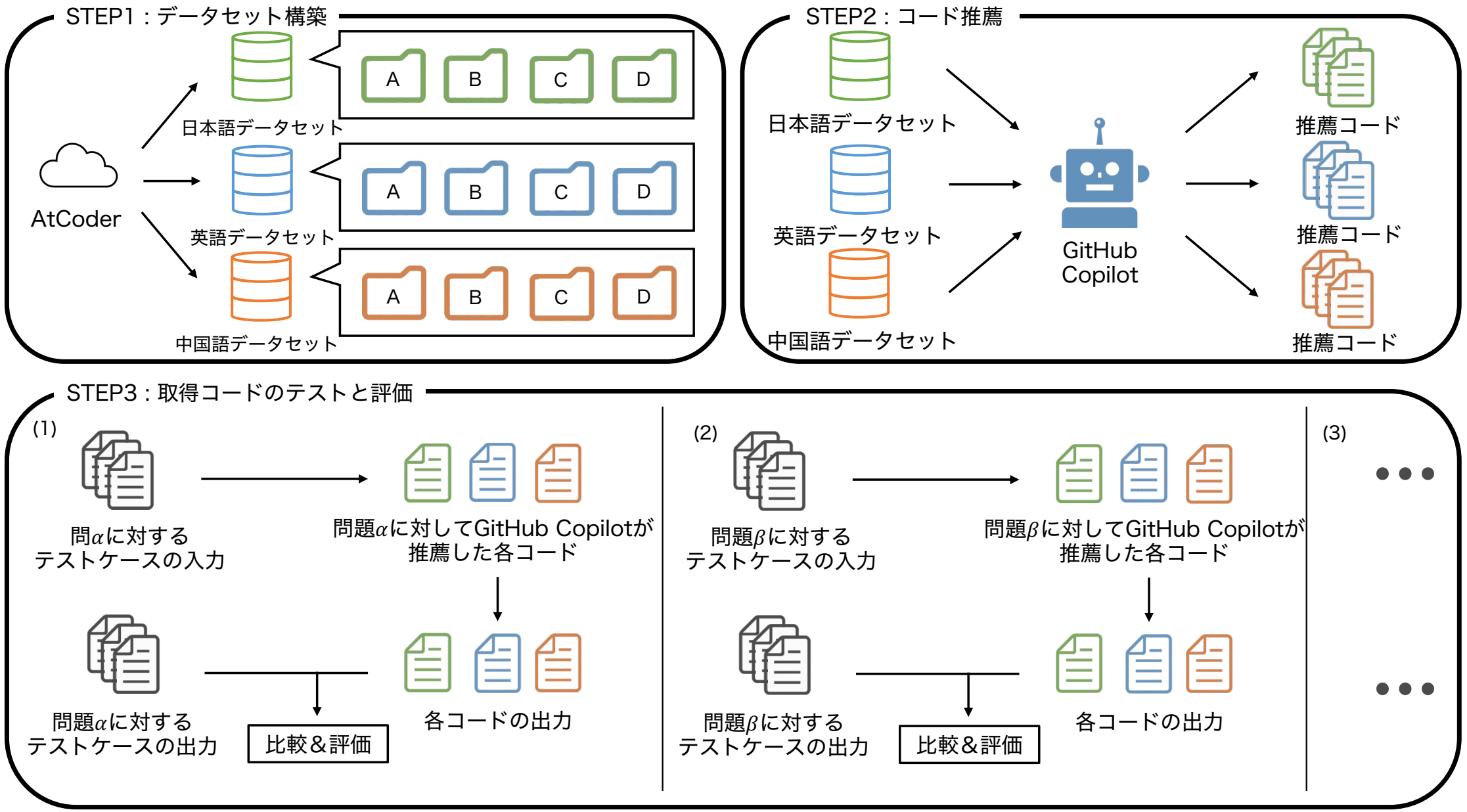
[2] https://www.ethnologue.com/

研究の目的

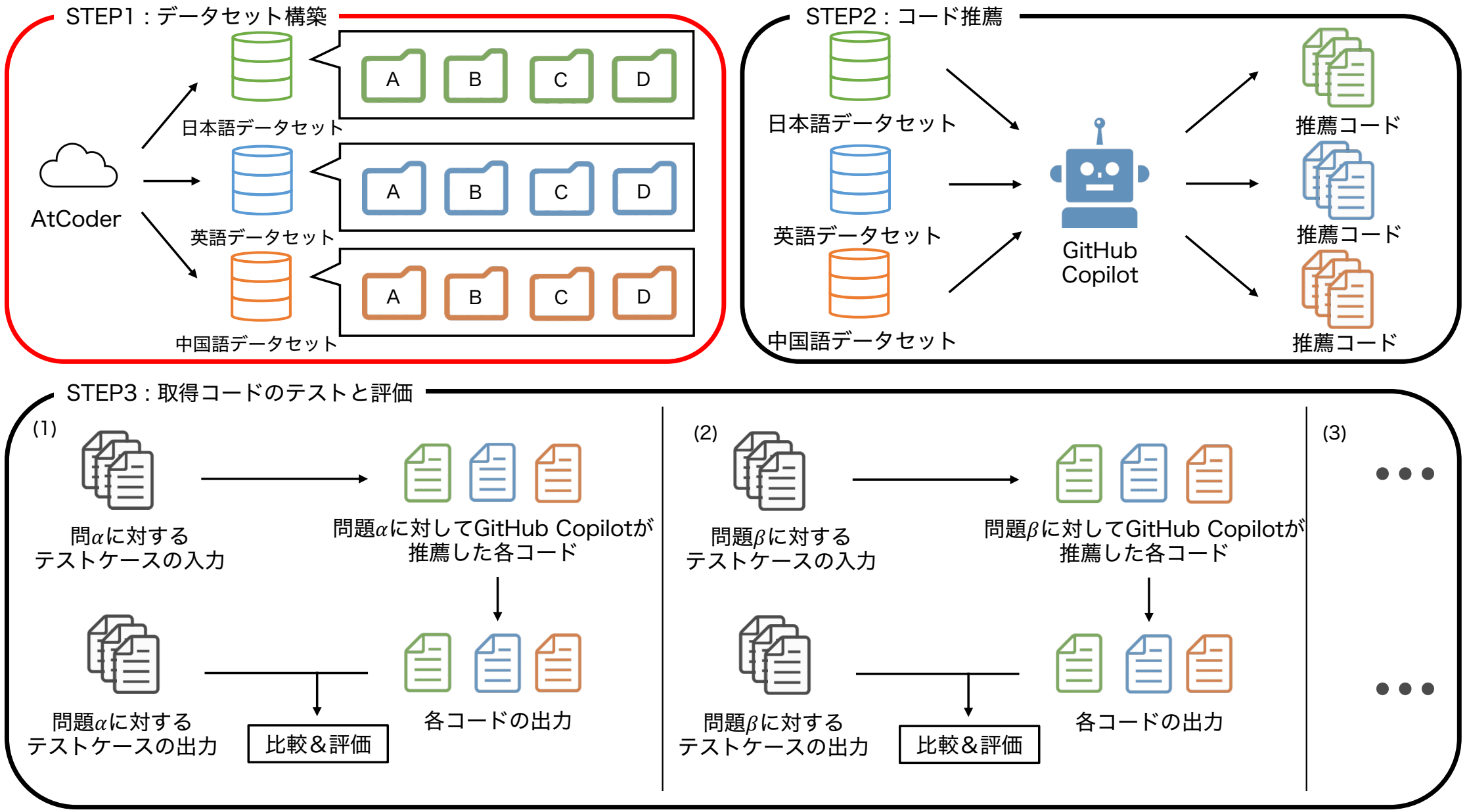
言語の違いがGitHub Copilotの性能に
どのような影響を与えるのか調査を行う



実験設計


STEP1：データセット構築
AtCoder
日本語データセット
英語データセット
中国語データセット
A
B
C
D
STEP2：コード推薦
日本語データセット
英語データセット
中国語データセット
GitHub Copilot
推薦コード
推薦コード
推薦コード
STEP3：取得コードのテストと評価
(1)
問αに対する
テストケースの入力
問題αに対してGitHub Copilotが
推薦した各コード
問題αに対する
テストケースの出力
比較＆評価
各コードの出力
(2)
問題βに対する
テストケースの入力
問題βに対してGitHub Copilotが
推薦した各コード
問題βに対する
テストケースの出力
比較＆評価
各コードの出力
(3)

実験設計


STEP1：データセット構築
AtCoder
日本語データセット
英語データセット
中国語データセット
A
B
C
D
STEP2：コード推薦
日本語データセット
英語データセット
中国語データセット
GitHub Copilot
推薦コード
推薦コード
推薦コード
STEP3：取得コードのテストと評価
(1)
問αに対する
テストケースの入力
問題αに対してGitHub Copilotが
推薦した各コード
問題αに対する
テストケースの出力
比較＆評価
各コードの出力
(2)
問題βに対する
テストケースの入力
問題βに対してGitHub Copilotが
推薦した各コード
問題βに対する
テストケースの出力
比較＆評価
各コードの出力
(3)

データセット構築

AtCoder

AtCoder Beginner Contest

・開催番号
  - 1 ~ 307 (2023/06/24現在)
・各コンテスト内の問題難易度
  - A, B, C, D, E, F, G, Ex(H)
・言語
  - 日本語，英語

データセット構築

AtCoder

AtCoder Beginner Contest

・開催番号
  - 1 ~ 307 (2023/06/24現在)
・各コンテスト内の問題難易度
  - A, B, C, D, E, F, G, Ex(H)
・言語
  - 日本語，英語

データセット
(日本語，英語)

・開催番号
  - 99 ~ 287
・問題難易度
  - A, B, C, D
・言語
  - 日本語，英語

データセット構築

AtCoder

AtCoder Beginner Contest

・開催番号
  - 1 ~ 307 (2023/06/24現在)
・各コンテスト内の問題難易度
  - A, B, C, D, E, F, G, Ex(H)
・言語
  - 日本語，英語

データセット
(日本語，英語)

・開催番号
  - 99 ~ 287
・問題難易度
  - A, B, C, D
・言語
  - 日本語，英語

テストケース

複雑な処理

データセット構築

データセット
(英語)

・開催番号
- 99 ~ 287
・各問題難易度
- A, B, C, D
・言語
- 英語

DeepL API

中国語に翻訳

中国人
ソフトウェア研究者
(研究歴約6年)

翻訳の確認

データセット
(中国語)

・開催番号
- 99 ~ 287
・各問題難易度
- A, B, C, D
・言語
- 中国語

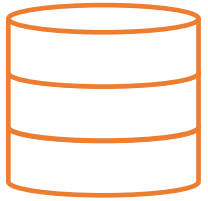
データセット構築

データセット
(英語)

・開催番号
- 99 ~ 287
・各問題難易度
- A, B, C, D
・言語
- 英語

データセット
(日本語)

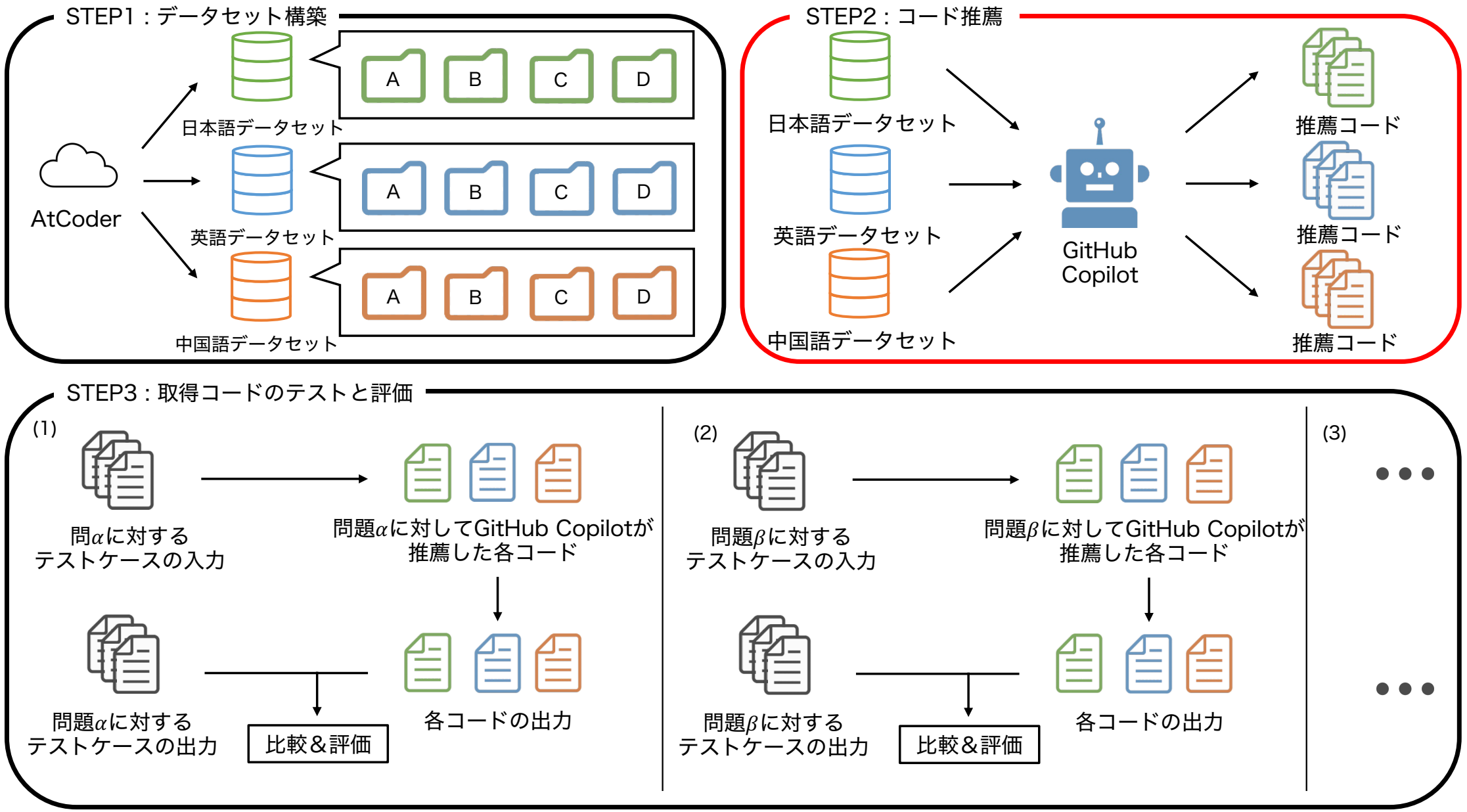
・開催番号
- 99 ~ 287
・各問題難易度
- A, B, C, D
・言語
- 日本語



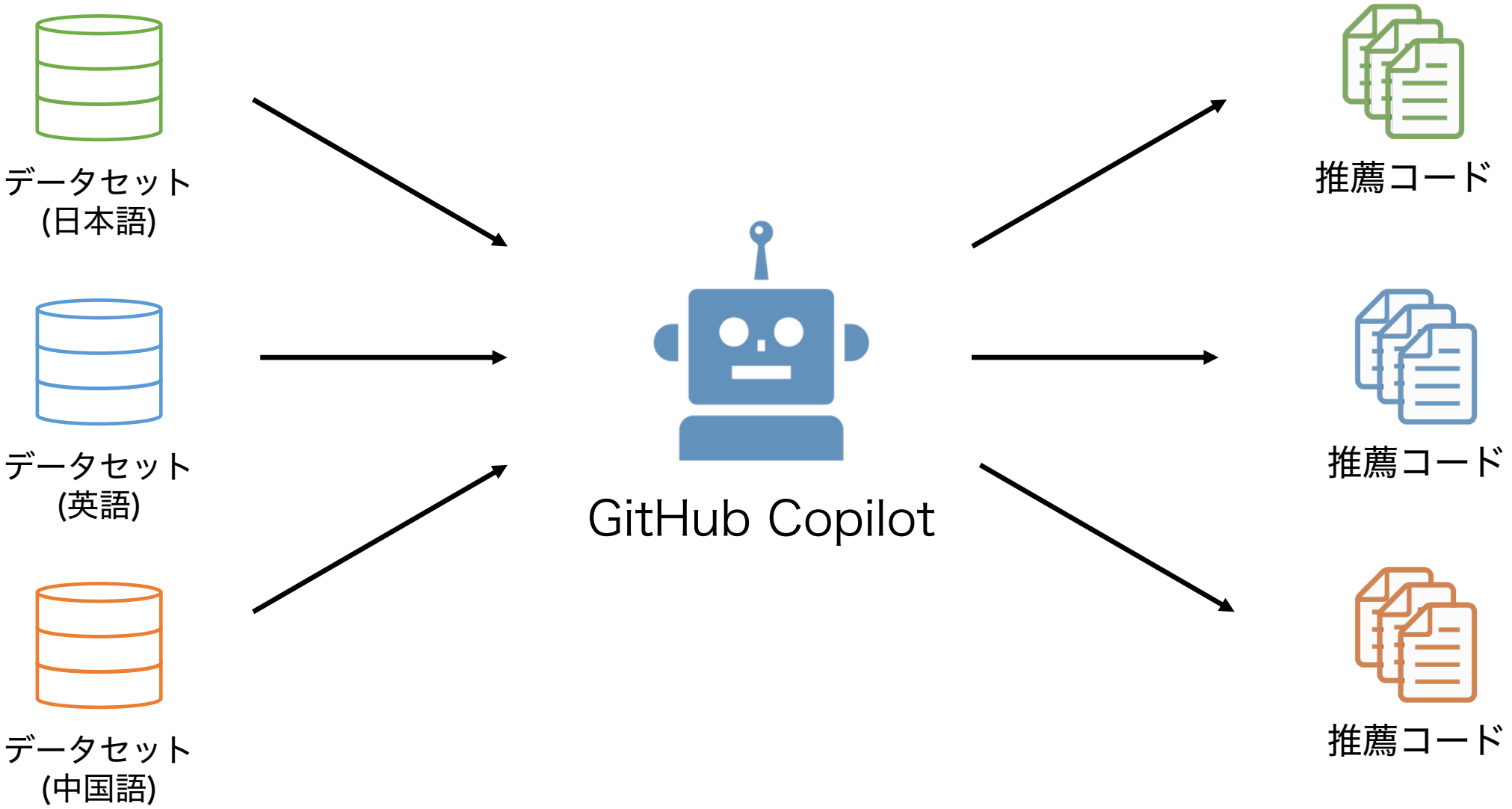
データセット
(中国語)

・開催番号
- 99 ~ 287
・各問題難易度
- A, B, C, D
・言語
- 中国語

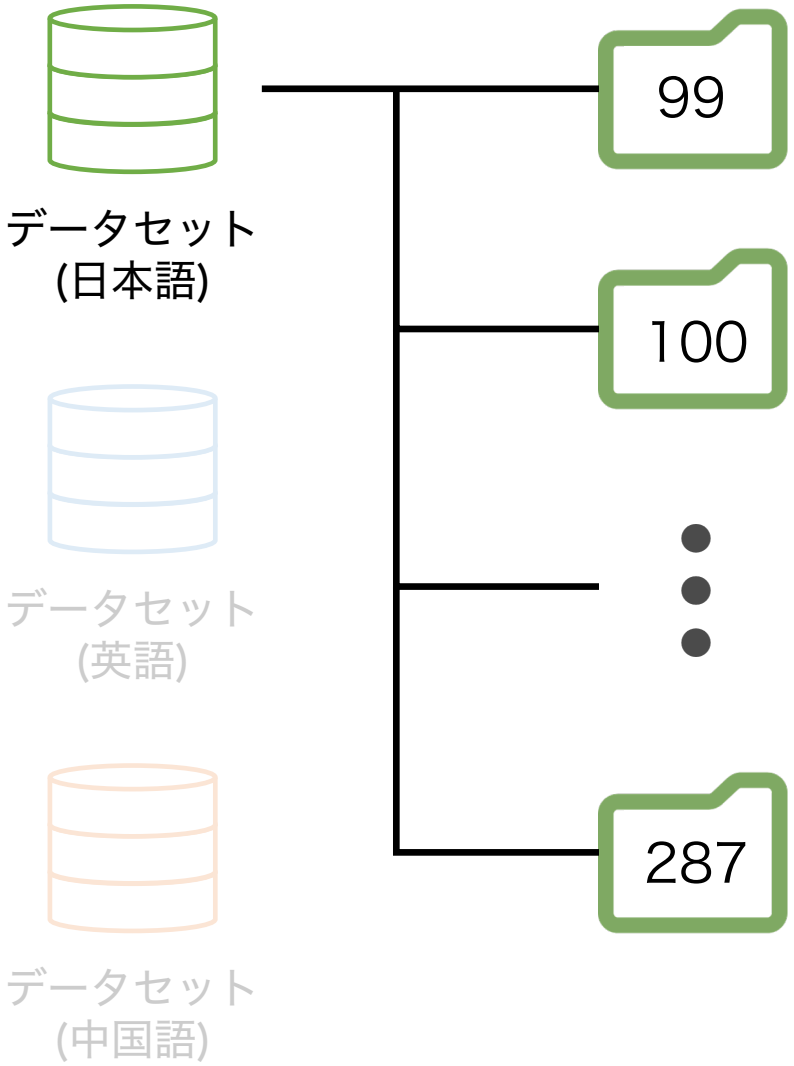
実験設計


STEP1：データセット構築
AtCoder
日本語データセット
英語データセット
中国語データセット
A
B
C
D
STEP2：コード推薦
日本語データセット
英語データセット
中国語データセット
GitHub Copilot
推薦コード
推薦コード
推薦コード
STEP3：取得コードのテストと評価
(1)
問αに対する
テストケースの入力
問題αに対してGitHub Copilotが
推薦した各コード
問題αに対する
テストケースの出力
比較＆評価
各コードの出力
(2)
問題βに対する
テストケースの入力
問題βに対してGitHub Copilotが
推薦した各コード
問題βに対する
テストケースの出力
比較＆評価
各コードの出力
(3)

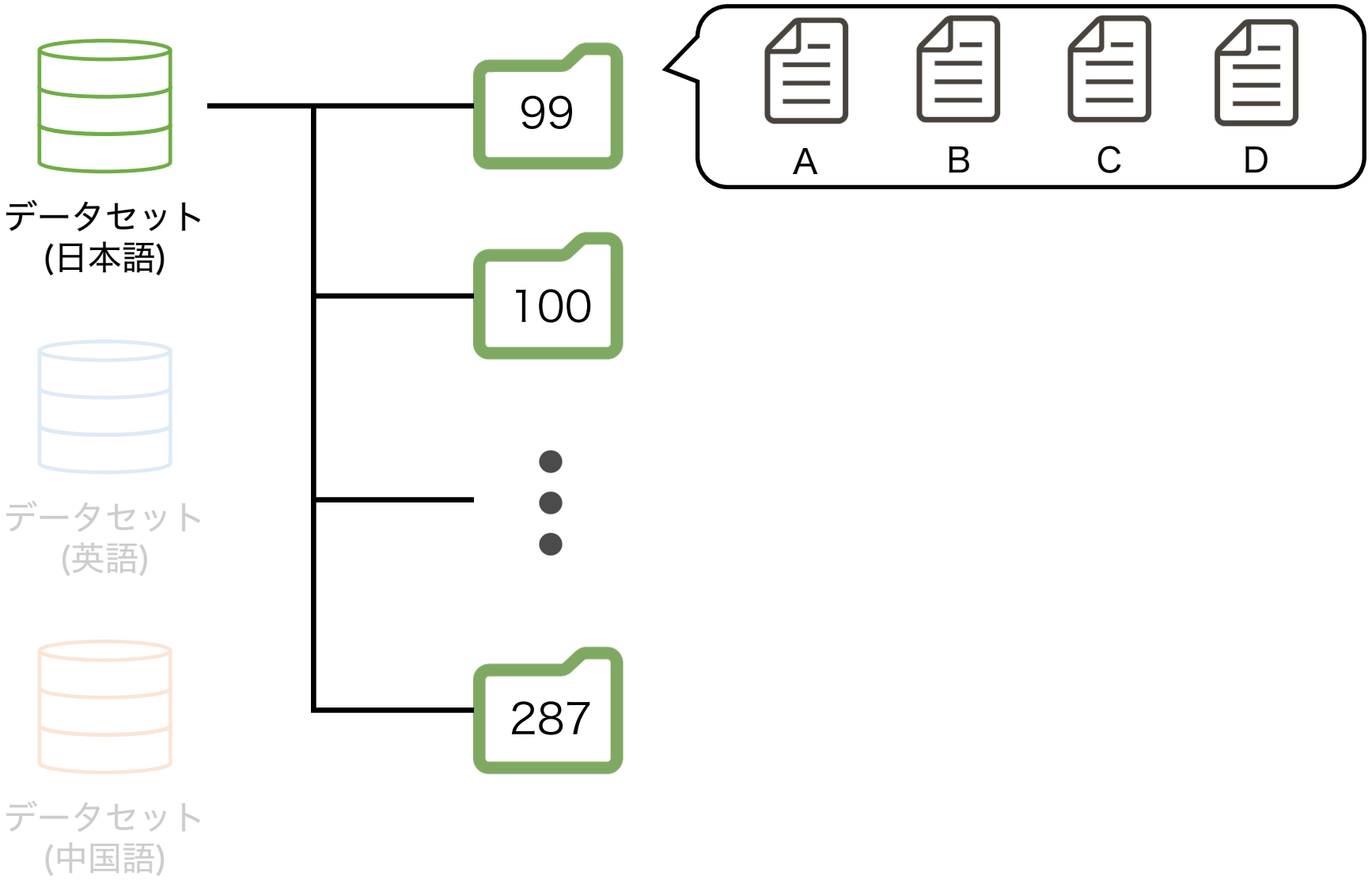
コード推薦


データセット
(日本語)
データセット
(英語)
データセット
(中国語)
GitHub Copilot
推薦コード
推薦コード
推薦コード

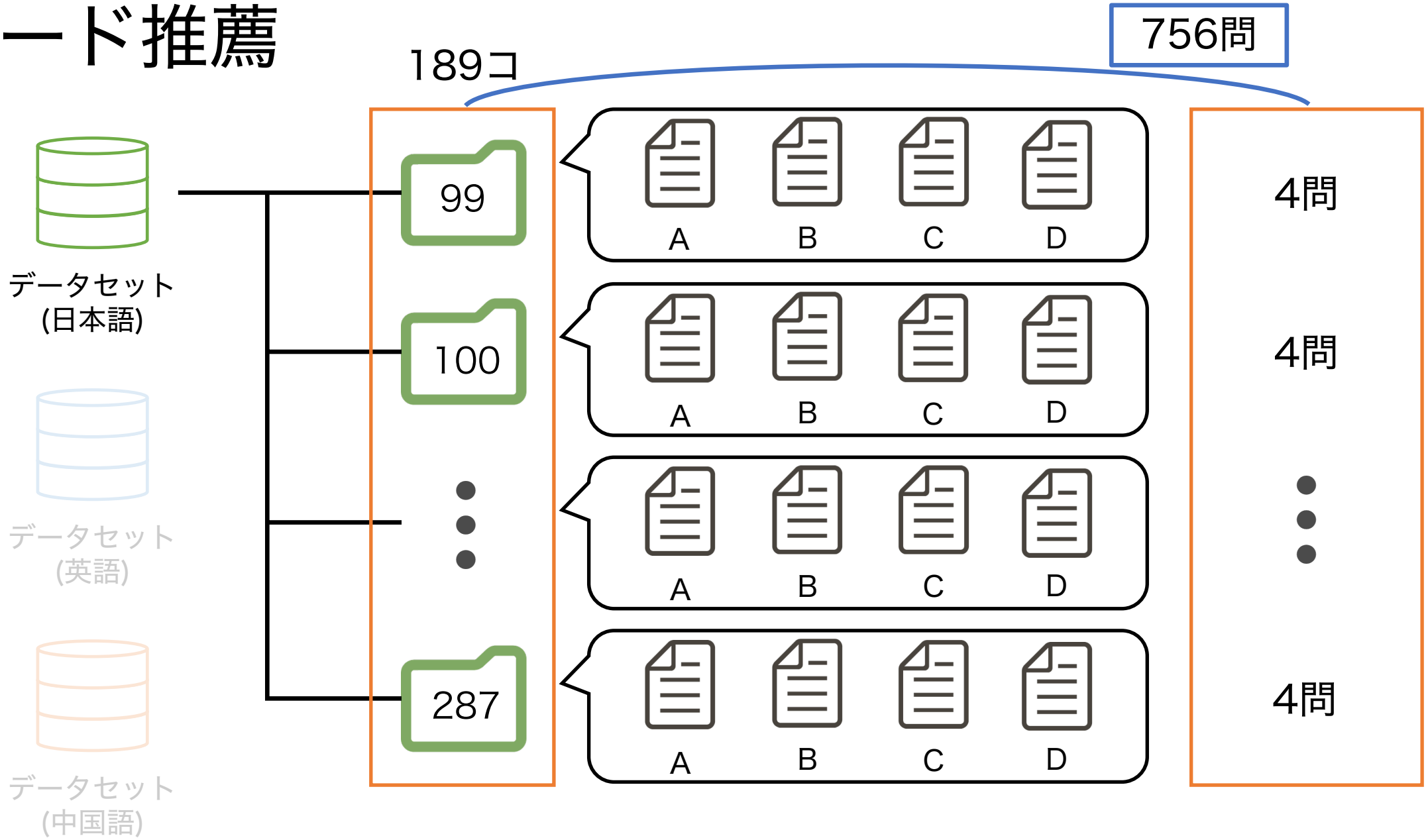
コード推薦



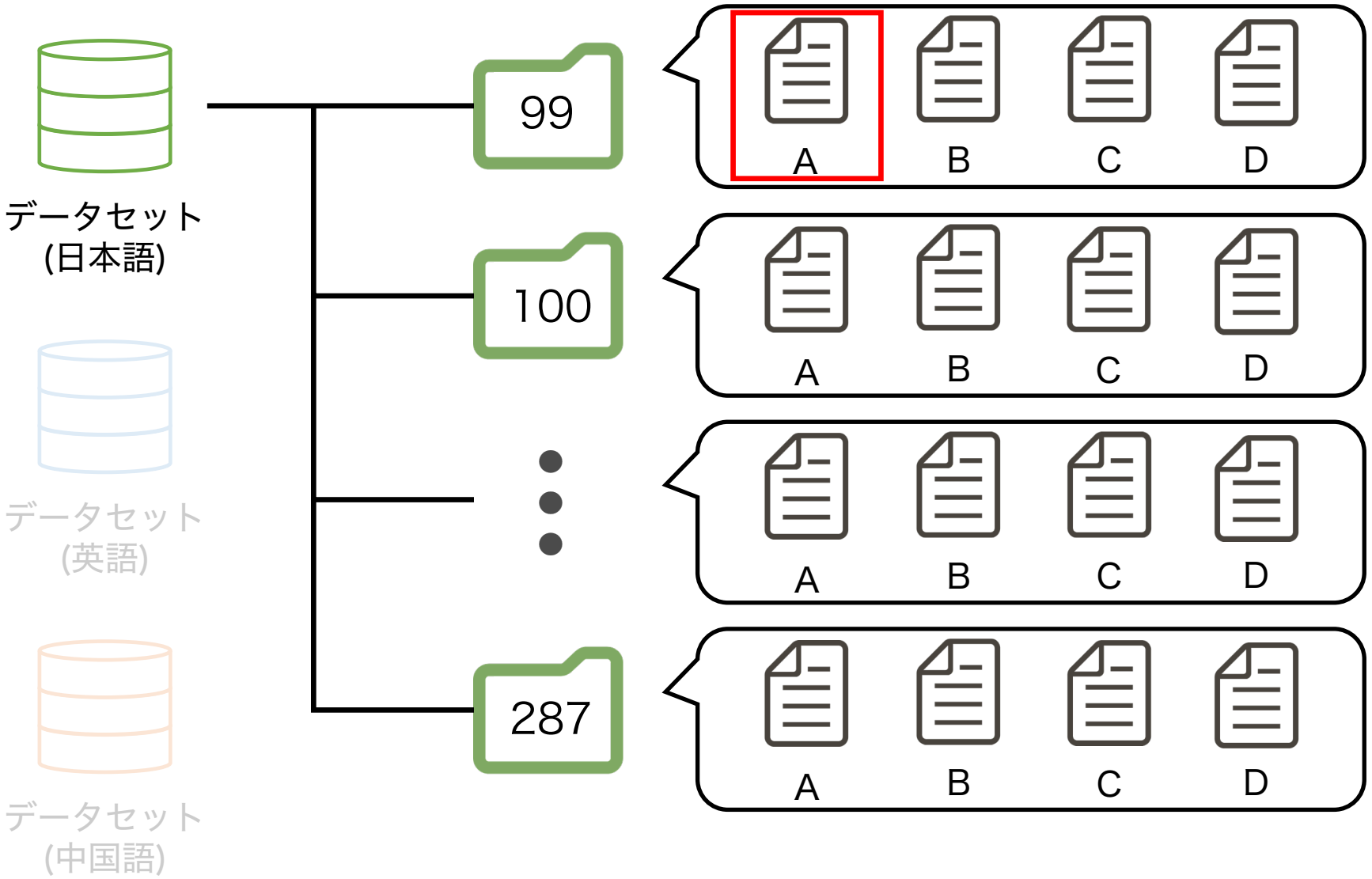
コード推薦



コード推薦


756問
189コ
99
100
287
A
B
C
D
4問
4問
4問
データセット
(日本語)
データセット
(英語)
データセット
(中国語)

コード推薦


データセット
(日本語)
データセット
(英語)
データセット
(中国語)
99
100
287
A
B
C
D
A
B
C
D
A
B
C
D
A
B
C
D

コード推薦([e.g.] 99 - A)

99-A

問題文

AtCoder Beginner Contestが始まってから早数十年．コンテストは1回目から順にABC001,ABC002,…と名付けられてきましたが，999回目のコンテストABC99を終え，これからのコンテストの名前をどうするかという問題が生じました．そこで，1000回目から1998回目のコンテストを順にABD001,ABC002,…,ABD999と名付けることとなりました．

1以上1998以下の整数Nが与えられるので，N回目のコンテストの名前の最初の3文字を出力してください．

コード推薦([e.g.] 99 - A)

99-A

問題文

AtCoder Beginner Contestが始まってから早数十年．コンテストは1回目から順にABC001,ABC002,…と名付けられてきましたが，999回目のコンテストABC99を終え，これからのコンテストの名前をどうするかという問題が生じました．そこで，1000回目から1998回目のコンテストを順にABD001,ABC002,…,ABD999と名付けることとなりました．

1以上1998以下の整数Nが与えられるので，N回目のコンテストの名前の最初の3文字を出力してください．

制約

・1≦N≦1998

・Nは整数

コード推薦([e.g.] 99 - A)

99-A

問題文

AtCoder Beginner Contestが始まってから早数十年．コンテストは1回目から順にABC001,ABC002,…と名付けられてきましたが，999回目のコンテストABC99を終え，これからのコンテストの名前をどうするかという問題が生じました．そこで，1000回目から1998回目のコンテストを順にABD001,ABC002,…,ABD999と名付けることとなりました．

1以上1998以下の整数Nが与えられるので，N回目のコンテストの名前の最初の3文字を出力してください．

制約

・1≦N≦1998

・Nは整数

入力

入力は以下の形式で標準入力から与えられる．

N

出力

N回目のコンテストの名前の最初の3文字を出力せよ．

コード推薦([e.g.] 99 - A)

99-A

問題文

AtCoder Beginner Contestが始まってから早数十年．コンテストは1回目から順にABC001,ABC002,…と名付けられてきましたが，999回目のコンテストABC99を終え，これからのコンテストの名前をどうするかという問題が生じました．そこで，1000回目から1998回目のコンテストを順にABD001,ABC002,…,ABD999と名付けることとなりました．

1以上1998以下の整数Nが与えられるので，N回目のコンテストの名前の最初の3文字を出力してください．

制約

・1≦N≦1998

・Nは整数

入力

入力は以下の形式で標準入力から与えられる．

N

出力

N回目のコンテストの名前の最初の3文字を出力せよ．

入力例1

999

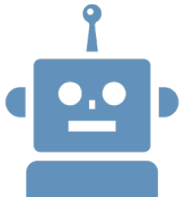
出力例 1

ABC

コード推薦([e.g.] 99 - A)

99-A

入力



GitHub Copilot

問題文

AtCoder Beginner Contestが始まってから早数十年．コンテストは1回目から順にABC001,ABC002,…と名付けられてきましたが，999回目のコンテストABC99を終え，これからのコンテストの名前をどうするかという問題が生じました．そこで，1000回目から1998回目のコンテストを順にABD001,ABC002,…,ABD999と名付けることとなりました．

1以上1998以下の整数Nが与えられるので，N回目のコンテストの名前の最初の3文字を出力してください．

制約

・1≦N≦1998

・Nは整数

入力

入力は以下の形式で標準入力から与えられる．

N

出力

N回目のコンテストの名前の最初の3文字を出力せよ．

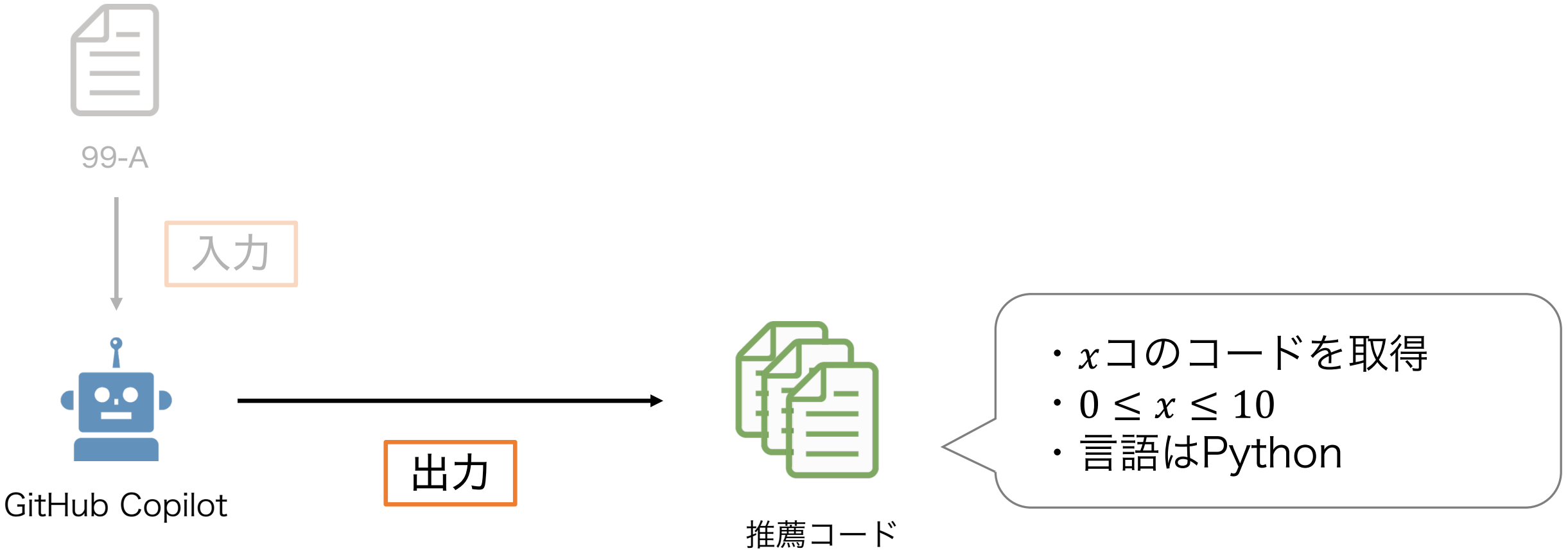
入力例1

999

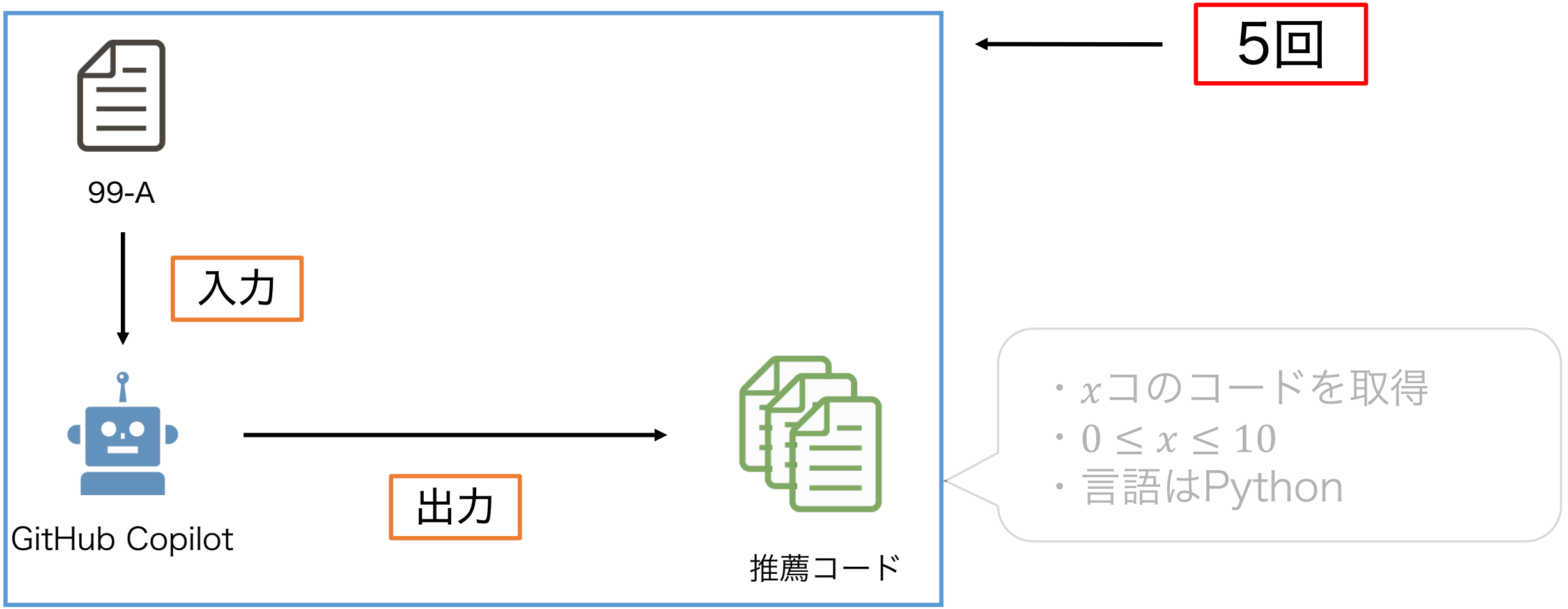
出力例 1

ABC

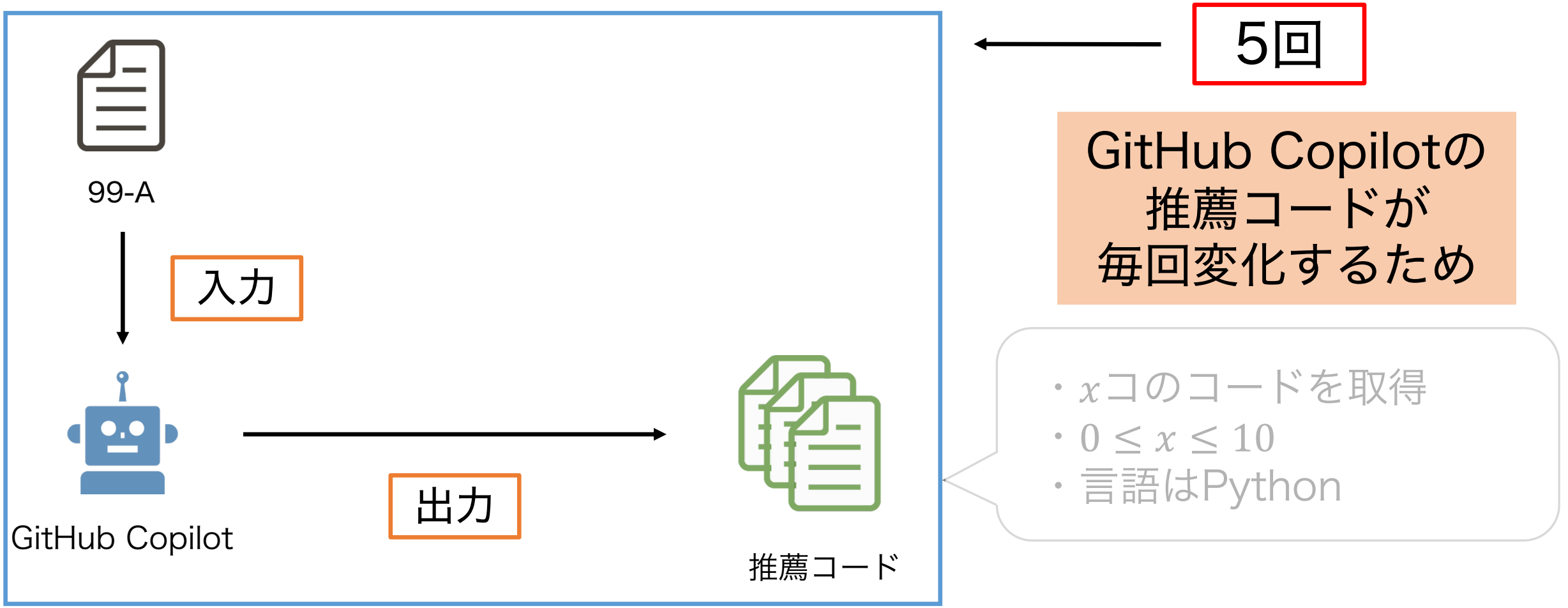
コード推薦([e.g.] 99 - A)


99-A
入力
GitHub Copilot
出力
推薦コード
・xコのコードを取得
・$0 \leq x \leq 10$
・言語はPython

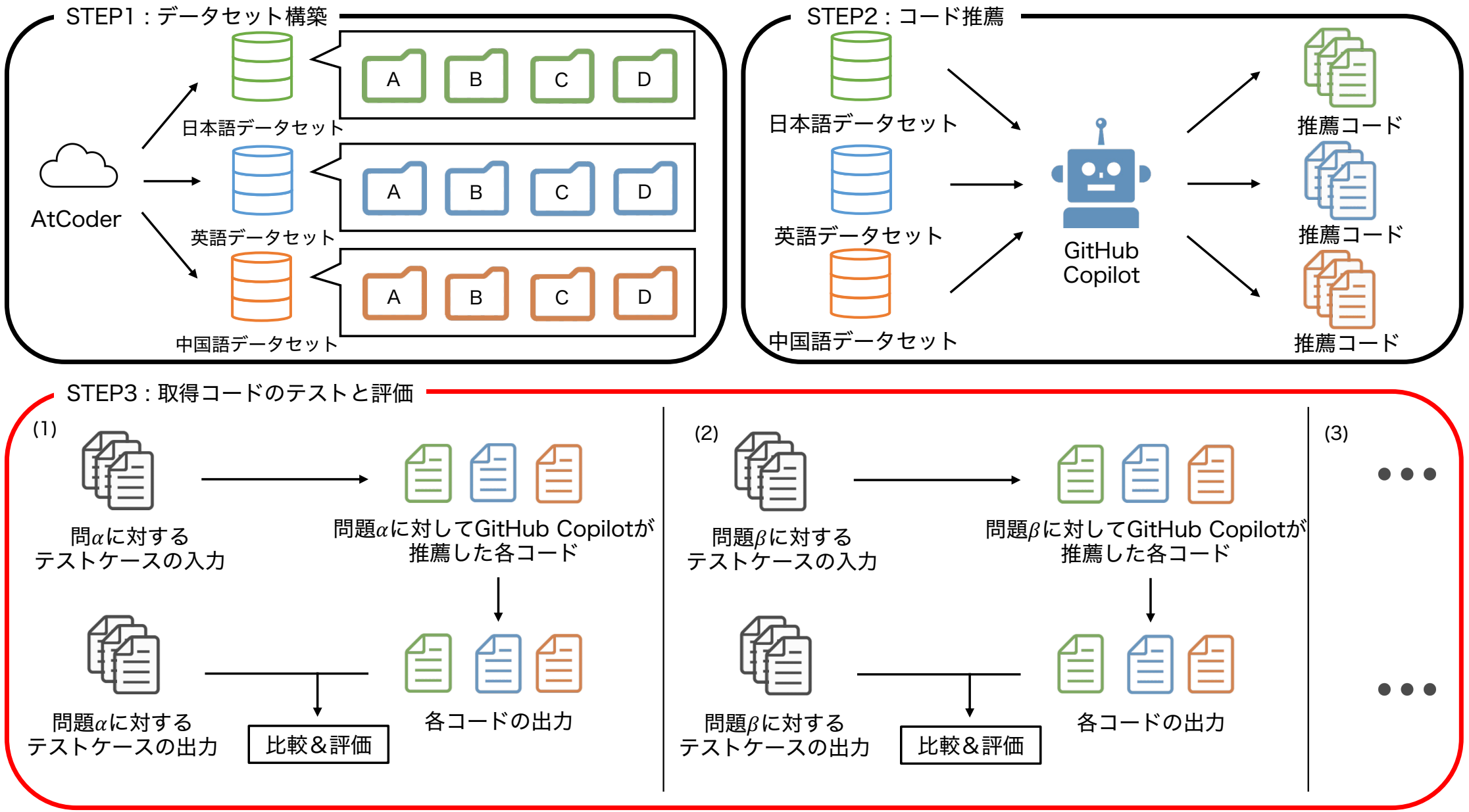
コード推薦([e.g.] 99 - A)


99-A
入力
GitHub Copilot
出力
推薦コード
5回
・xコのコードを取得
・$0 \leq x \leq 10$
・言語はPython

コード推薦([e.g.] 99 - A)


99-A
入力
GitHub Copilot
出力
推薦コード
5回
GitHub Copilotの
推薦コードが
毎回変化するため
・xコのコードを取得
・$0 \leq x \leq 10$
・言語はPython

実験設計


STEP1：データセット構築
AtCoder
日本語データセット
英語データセット
中国語データセット
A
B
C
D
STEP2：コード推薦
日本語データセット
英語データセット
中国語データセット
GitHub Copilot
推薦コード
推薦コード
推薦コード
STEP3：取得コードのテストと評価
(1)
問αに対する
テストケースの入力
問題αに対してGitHub Copilotが
推薦した各コード
問題αに対する
テストケースの出力
比較＆評価
各コードの出力
(2)
問題βに対する
テストケースの入力
問題βに対してGitHub Copilotが
推薦した各コード
問題βに対する
テストケースの出力
比較＆評価
各コードの出力
(3)

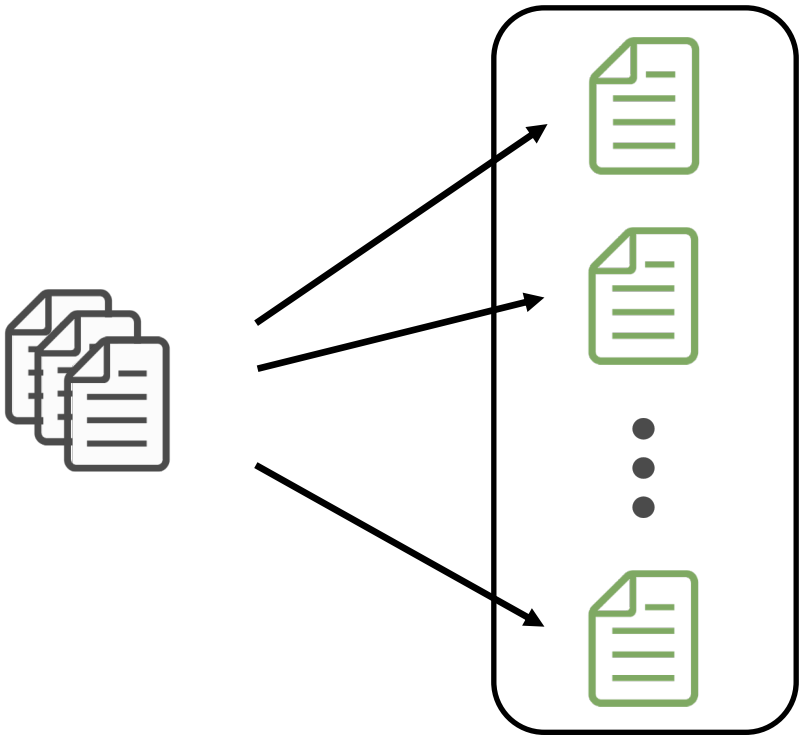
取得スクリプトのテストと評価

e.g.) 99-A

推薦コード
(最大10個)

取得スクリプトのテストと評価

e.g.) 99-A

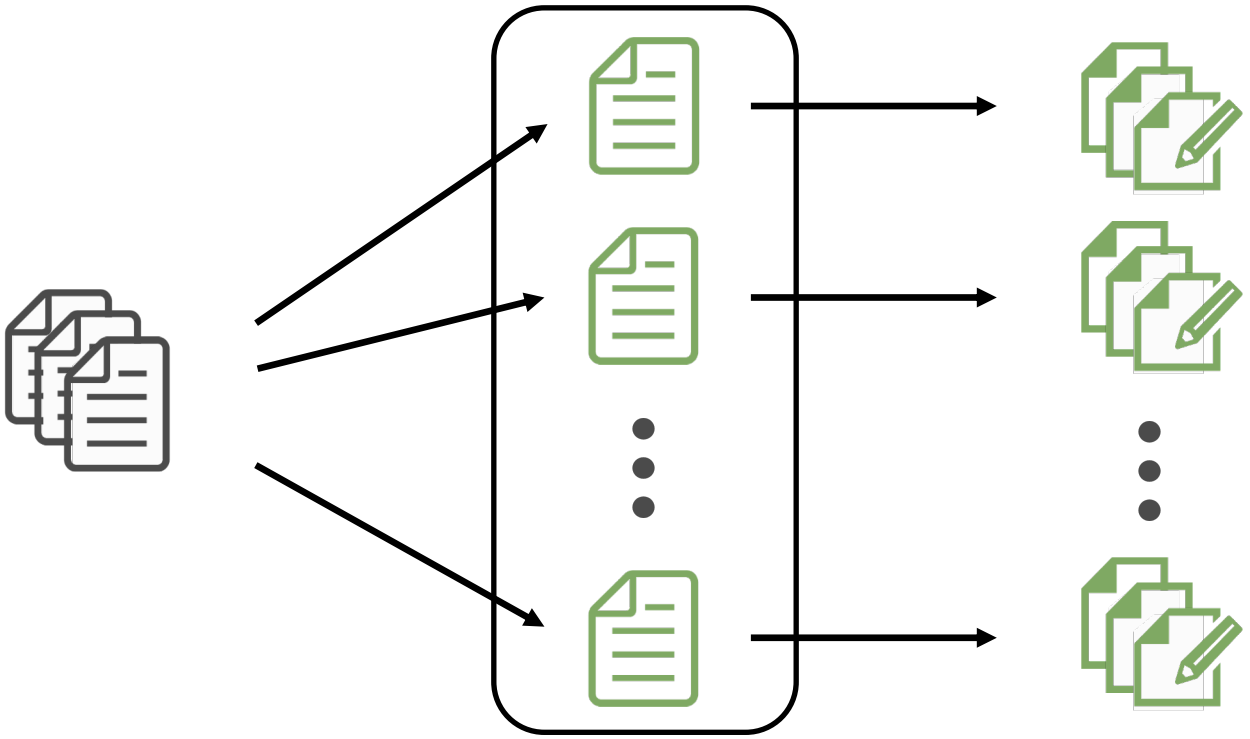


テストケースの
入力

推薦コード
(最大10個)

取得スクリプトのテストと評価

e.g.) 99-A



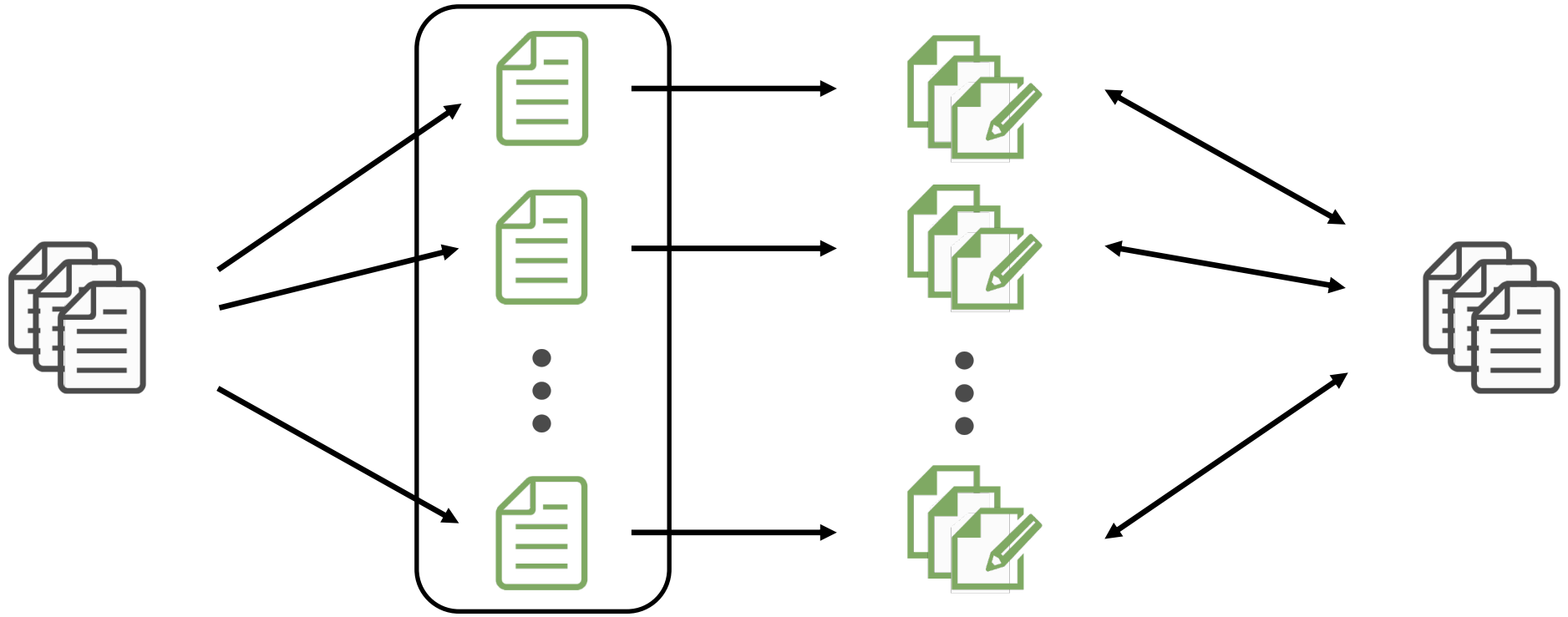
テストケースの
入力

推薦コード
(最大10個)

各テストケース
に対する出力

取得スクリプトのテストと評価

e.g.) 99-A



テストケースの
入力

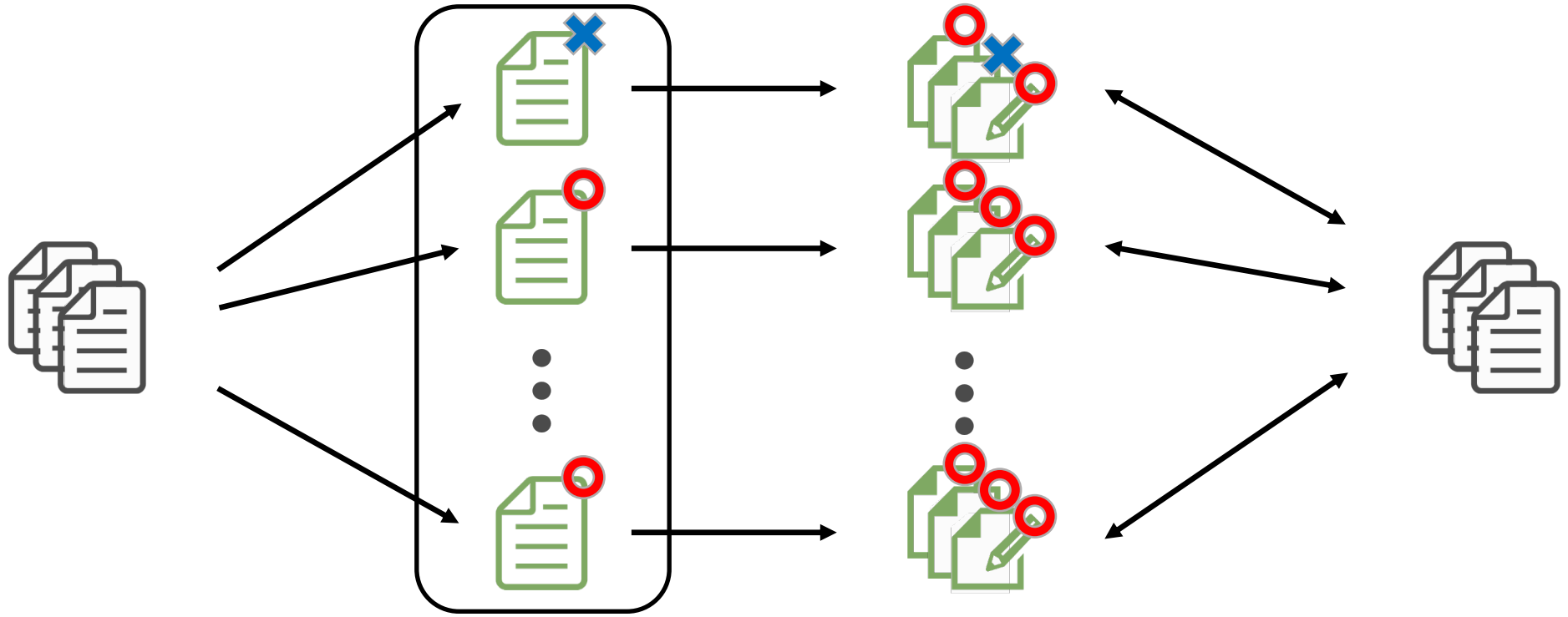
推薦コード
(最大10個)

各テストケース
に対する出力

テストケースの
出力

取得スクリプトのテストと評価

e.g.) 99-A



テストケースの
入力

推薦コード
(最大10個)

各テストケース
に対する出力

テストケースの
出力

評価指標

- $Accuracy$ （正答率）

  推薦された全推薦コードの内,

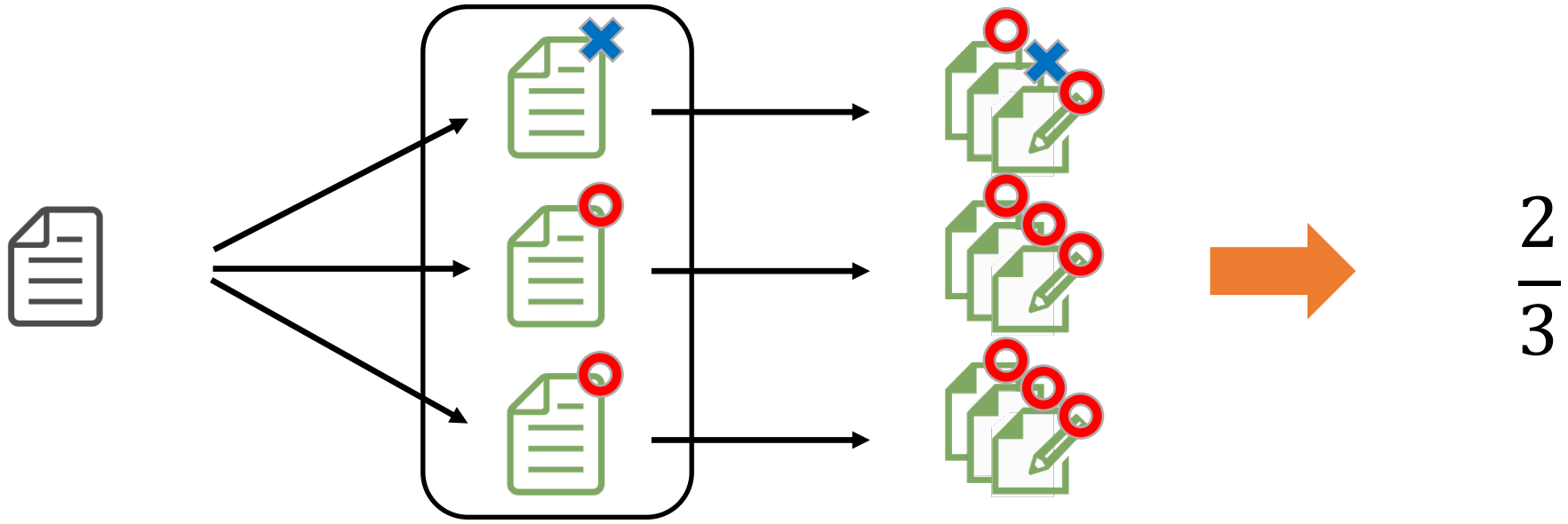
  全てのテストケースを通過した推薦コードの割合

評価指標

・$Accuracy$ （正答率）

推薦された全推薦コードの内,

全てのテストケースを通過した推薦コードの割合



問題99-A　　推薦コード　　各テストケースに対する出力　　$Accuraccy$

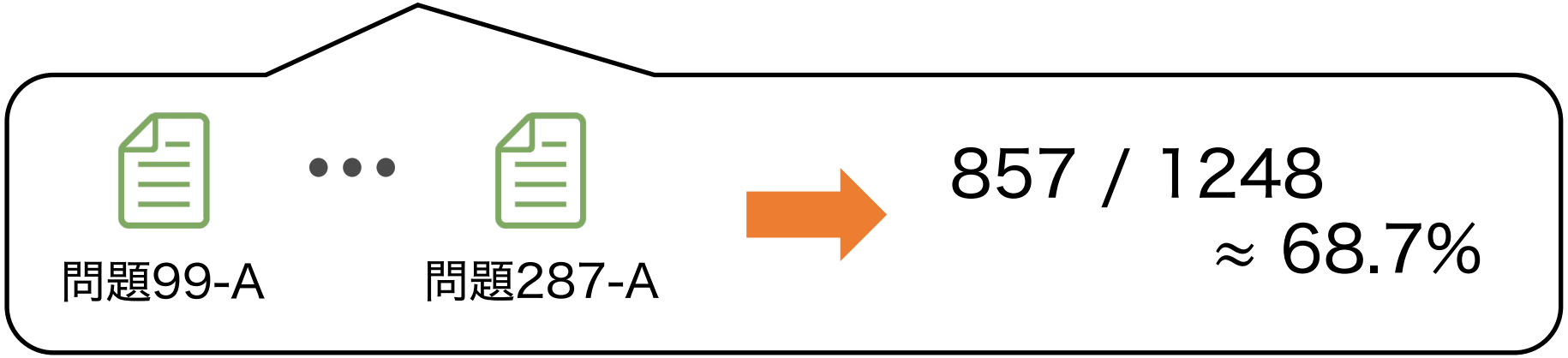
評価指標

・*Accuracy*　（正答率）

推薦された全推薦コードの内,

全てのテストケースを通過した推薦コードの割合

日本語	1回目	2回目	3回目	4回目	5回目
A	68.7%	67.4%	68.1%	68.4%	68.0%



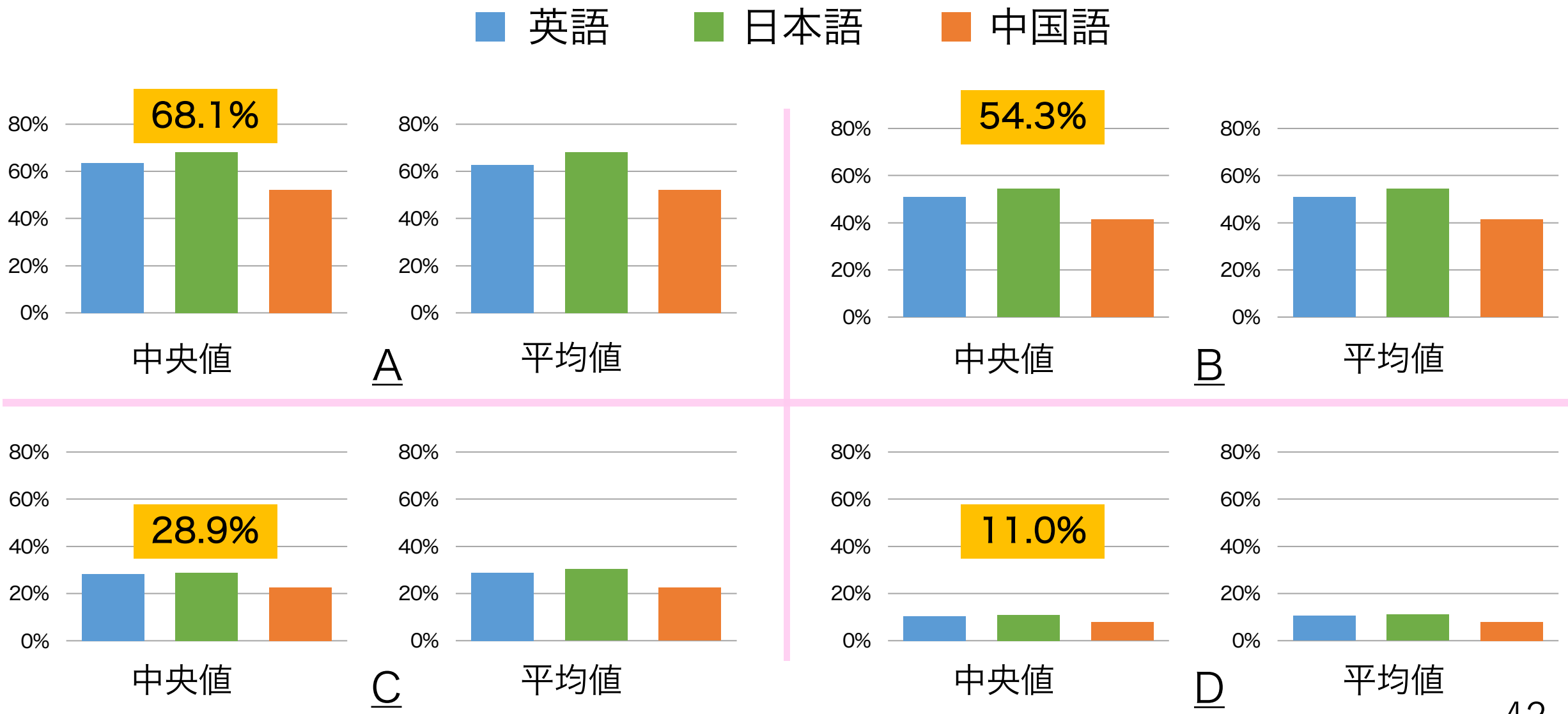
Research Question

入力言語の違いによって
Copilotの性能(正答率)にどのような影響を与えるのか？

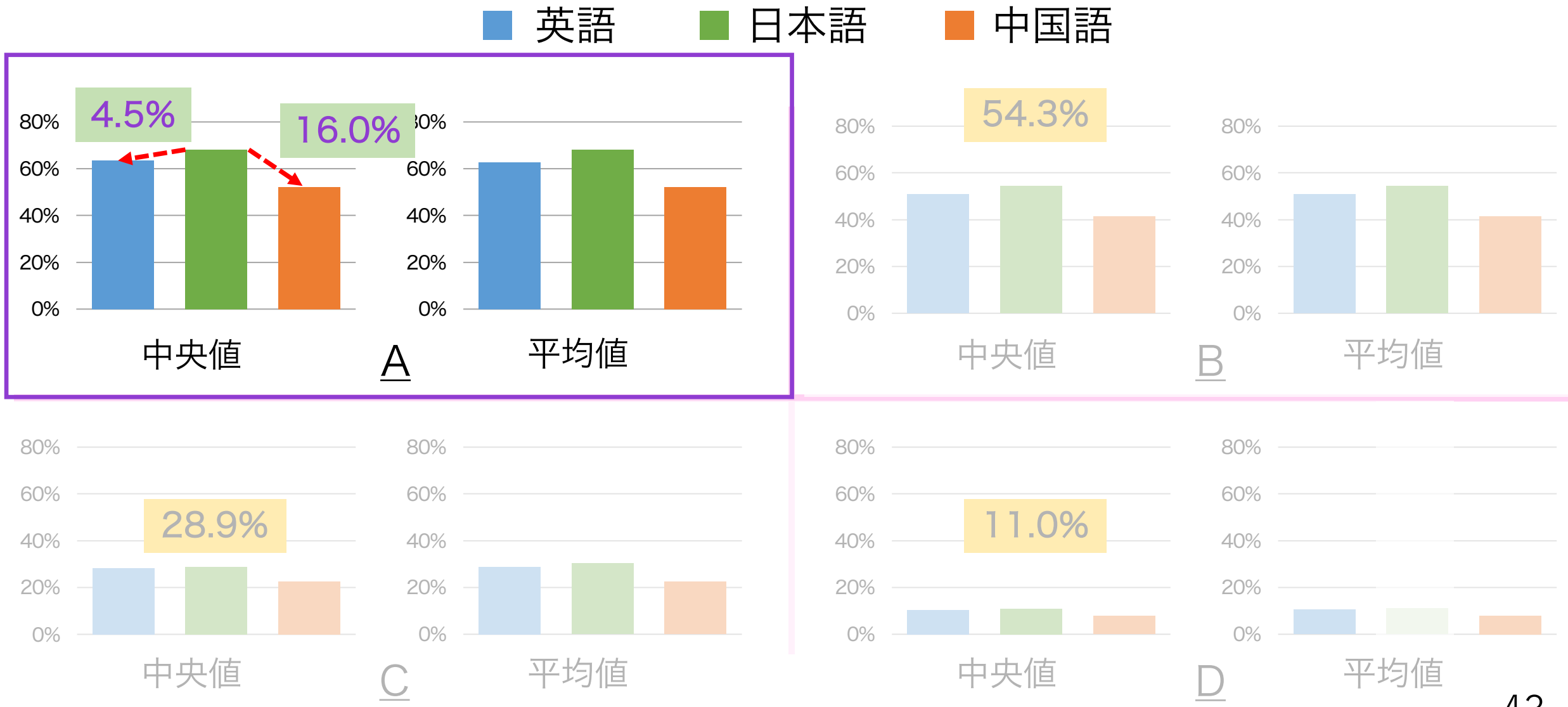
目的

入力言語の違いによってCopilotの性能に
差が生じるのか明らかにすることで,
今後のCopilotの最適な活用についての知見を得る

結果：全難易度において日本語の$Accuracy$が最も高かった



結果：全難易度において日本語の$Accuracy$が最も高かった



$Accuracy$の差が大きかった問題 ([e.g.] 212 - A)

212-A

推薦コード
(英語)

推薦コード
(日本語)

推薦コード
(中国語)

*Accuracy*の差が大きかった問題 ([e.g.] 212 - A)

212-A

問題文

高橋くんはAグラムの純金とBグラムの純銀(0 ≦ A, B, 0 < A + B)をよく溶かした上で混ぜ合わせ，新たな金属を作成しました．

生成された金属は「純金」「純銀」「合金」のいずれでしょうか？なお，生成された金属は

・0 < A かつ B = 0 なら「純金」　・A = 0 かつ 0 < B なら「純銀」　・0 < A かつ 0 < B なら「合金」であるとみなします．

制約

・0 ≦ A, B ≦ 100　・0 < A + B　・A, Bは整数

入力

入力は以下の形式で標準入力から与えられる．

A B

出力

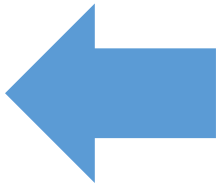
生成された金属が「純金」ならGoldと，「純銀」ならSilverと，「合金」ならAlloyと出力せよ．

入力例1	入力例2	入力例3	入力例4
50 50	100 0	0 100	100 2
出力例 1	**出力例2**	**出力例3**	**出力例4**
Alloy	Gold	Silver	Alloy

*Accuracy*の差が大きかった問題 ([e.g.] 212 - A)

```
def main():
    A, B = map(int, input().split())
    if A == 0:
        print("Silver")
    elif B == 0:
        print("Gold")
```

```
def main():
      a,b = map(int,input().split())
      if a == 0 and b != 0:
            print('Silver’)
      elif a != 0 and b == 0:
```



推薦コード
(中国語)

$Accuracy$の差が大きかった問題 ([e.g.] 212 - A)

212-A

問題文

高橋くんはAグラムの純金とBグラムの純銀(0 ≦ A, B, 0 < A + B)をよく溶かした上で混ぜ合わせ，新たな金属を作成しました．

生成された金

・0 < A かつ

るとみなします．

制約

・0 ≦ A, B ≦

入力

入力は以下の形式で標準入力から与えられる．

A B

出力

生成された金属が「純金」ならGoldと，「純銀」ならSilverと，「合金」ならAlloyと出力せよ．

入力例1	入力例2	入力例3	入力例4
50 50	100 0	0 100	100 2
出力例 1	出力例2	出力例3	出力例4
Alloy	Gold	Silver	Alloy

条件が複数，出力が文字列の場合に
$Accuracy$の差が大きかった

$Accuracy$の差が大きかった問題 ([e.g.] 212 - A)

212-A

問題文

高橋くんはAグラムの純金とBグラムの純銀(0 ≦ A, B, 0 < A + B)をよく溶かした上で混ぜ合わせ，新たな金属を作成しました．

生成された金

・0 < A かつ

るとみなします．

制約

・0 ≦ A, B ≦

入力

入力は以下の形式で標準入力から与えられる．

A B

出力例 1

Alloy

出力例2

Gold

出力例3

Silver

出力例4

Alloy

条件が複数，出力が文字列の場合に
$Accuracy$の差が大きかった

文章中のローマ字や英単語がシンボルとして認識されなかった
データセットを翻訳して作成した影響

RQへの回答

入力言語の違いによって
Copilotの性能(正答率)にどのような影響を与えるのか？

GitHub Copilotの推薦コードの正答率は,
入力言語によって変化し,
日本語, 英語, 中国語の順に高かった

RQへの回答

入力言語の違いによって
Copilo のか？

ある特定のタスクにおいては

英語以外の言語を使用した方が

正答率が高い可能性がある

G
入力言語によって変化し，
日本語，英語，中国語の順に高かった

今後の展望

データセット変更

LeetCodeの問題
(英語)

↓

日本語，中国語
に翻訳

推薦コードの順番を考慮

最大10個の
推薦コードを取得

↓

最大個数の変更
or
推薦順序に重みづけ

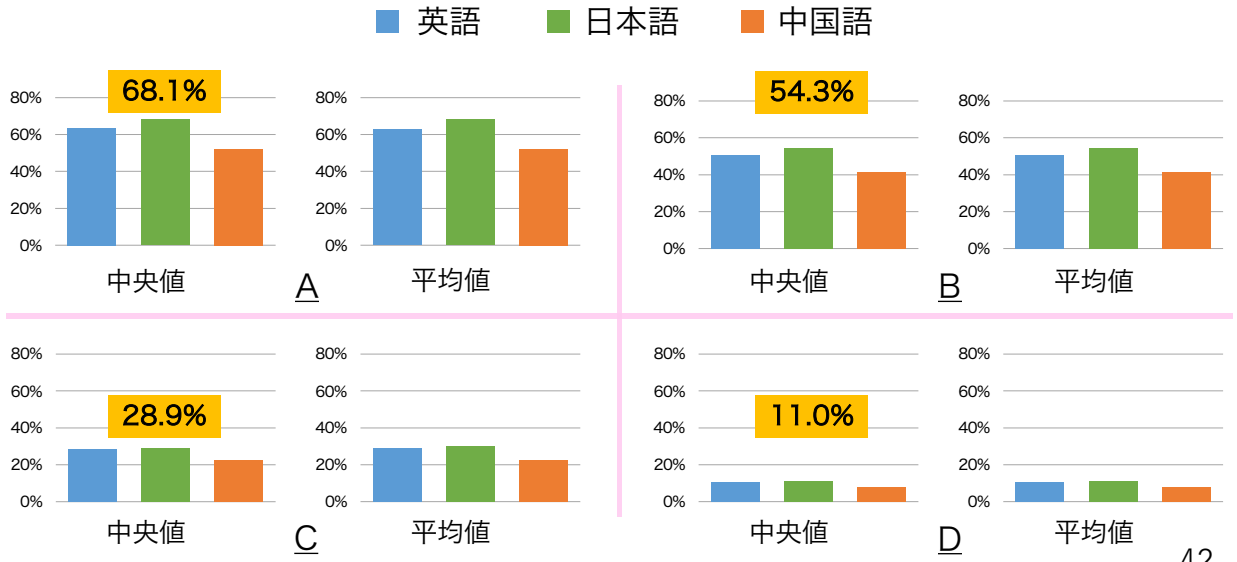
推薦コードの品質

正答率($Accuracy$)
のみを評価

＋

推薦コードの
可読性も評価

結果：全難易度において日本語の$Accuracy$が最も高かった



RQへの回答

入力言語の違いによって
Copilotの性能(正答率)にどのような影響を与えるのか？

GitHub Copilotの推薦コードの正答率は，
入力言語によって変化し，
日本語，英語，中国語の順に高かった

RQへの回答

ある特定のタスクにおいては
英語以外の言語を使用した方が
正答率が高い可能性がある

今後の展望

データセット変更

LeetCodeの問題
(英語)

↓

日本語，中国語
に翻訳

推薦コードの順番を考慮

最大10個の
推薦コードを取得

↓

最大個数の変更
or
推薦順序に重みづけ

推薦コードの品質

正答率($Accuracy$)
のみを評価

＋

推薦コードの
可読性も評価