

マシンアンラーニングを用いた モデル崩壊回復手法の提案

オープンソースソフトウェア工学研究室
S2620051 佐久間一颯

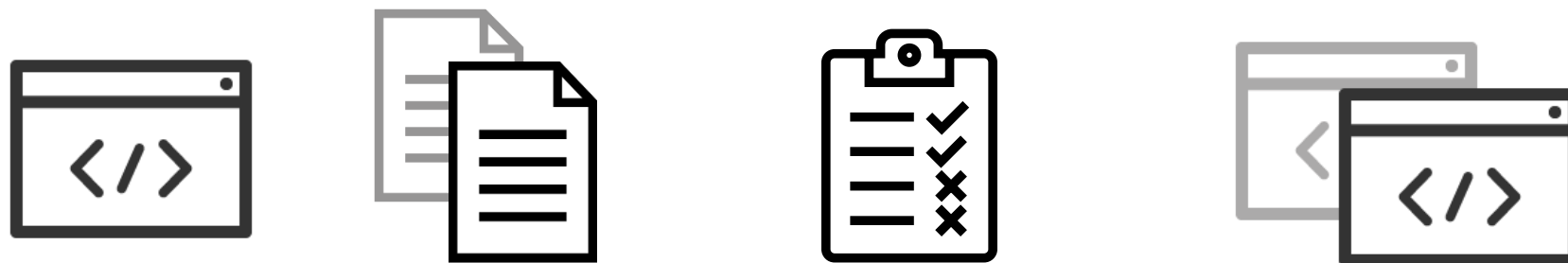
背景

生成AIの普及とデータの枯渇

- 日本企業の87.6%が生成AIを業務効率化に導入済^[1]

ソフトウェア開発分野では全工程で使用する支援が進歩している

例) コード生成, バグ修正, テストケース生成, コード自動修正



- 高精度な生成AIを構築するには, 高品質な**実データ**が必要

▶ しかし2026年には実データが**収集され尽くされる**ことが予想されている^[2]

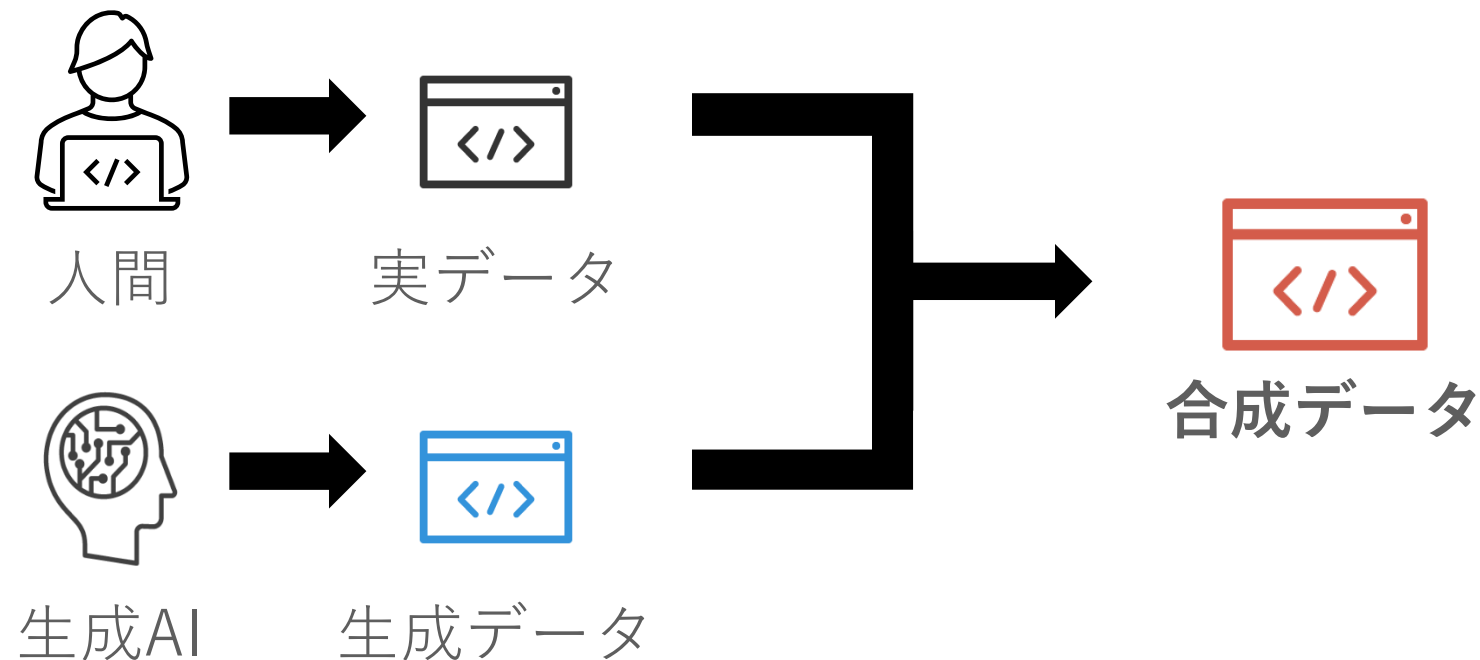
[1] デロイト トーマツ: プライム上場企業における生成AI活用の意識調査, 2024年5月.

[2] P. Villalobos, et al. "Will we run out of data? Limits of LLM scaling based on human-generated data," ICML, 2024.

背景

データ枯渇のための対策案

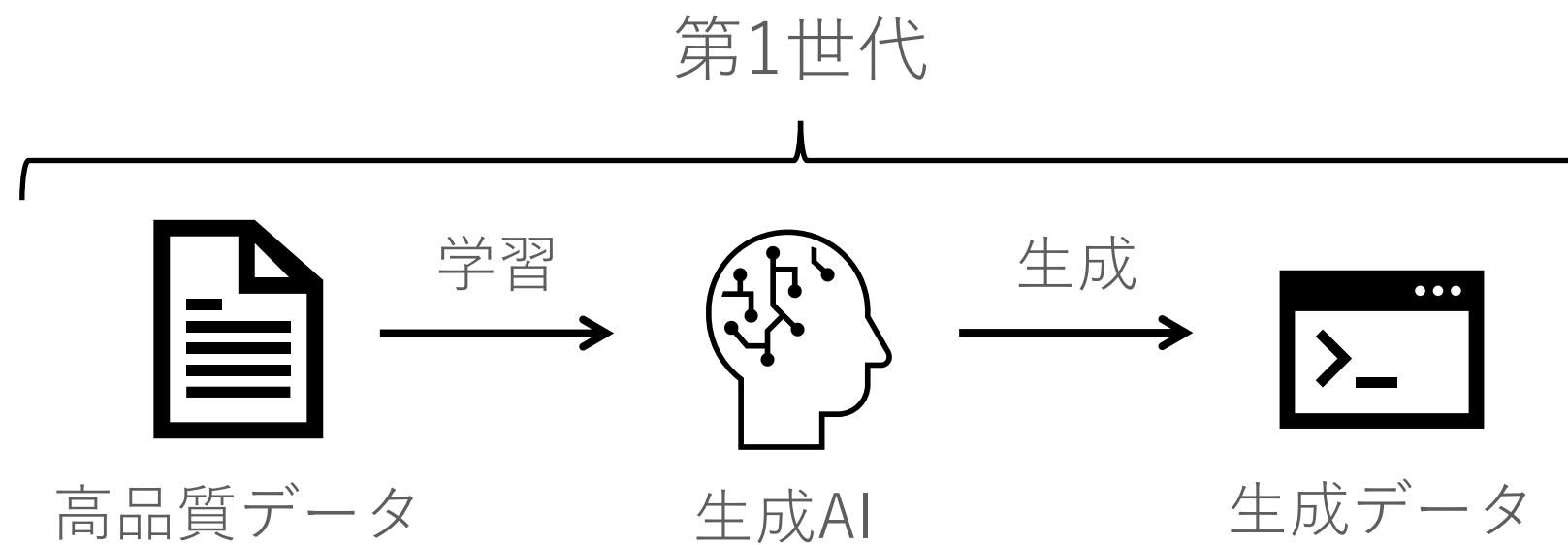
- ・ 対策として今後合成データを使用することが考えられる
- ▶ 合成データを使用すると、**モデル崩壊**を起こす危険性



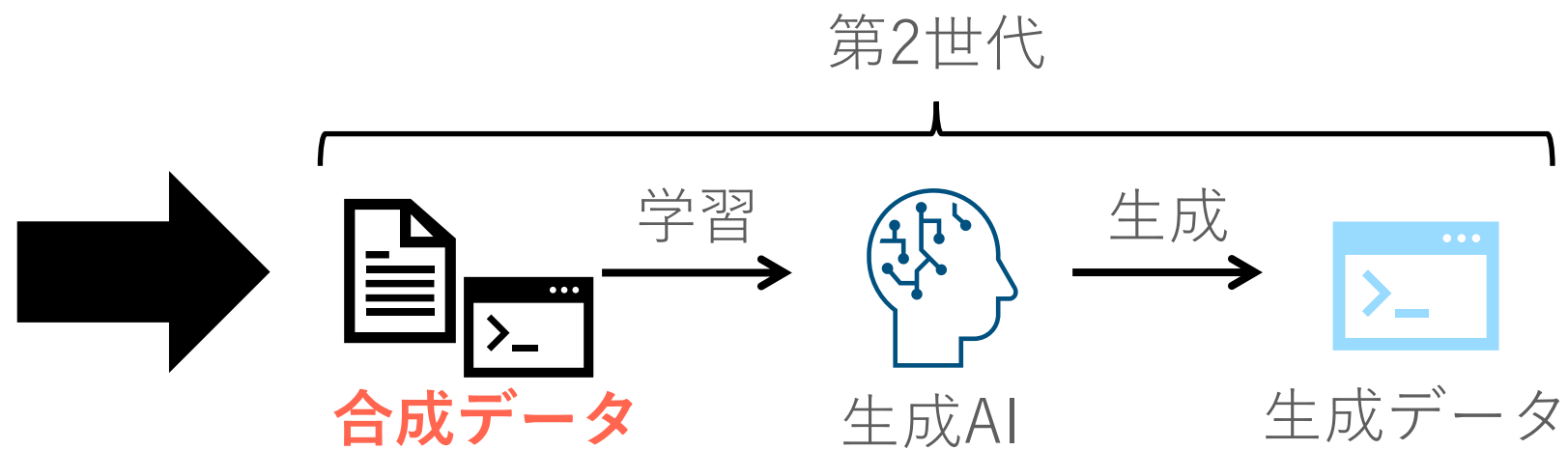
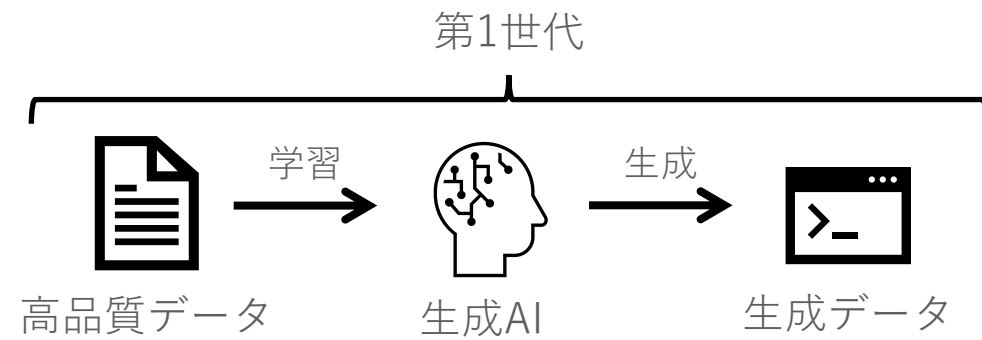
モデル崩壊とは

生成AIが自分の出力で追加学習を繰り返す過程で
出力の質が劣化し、多様性や正確性が失われること

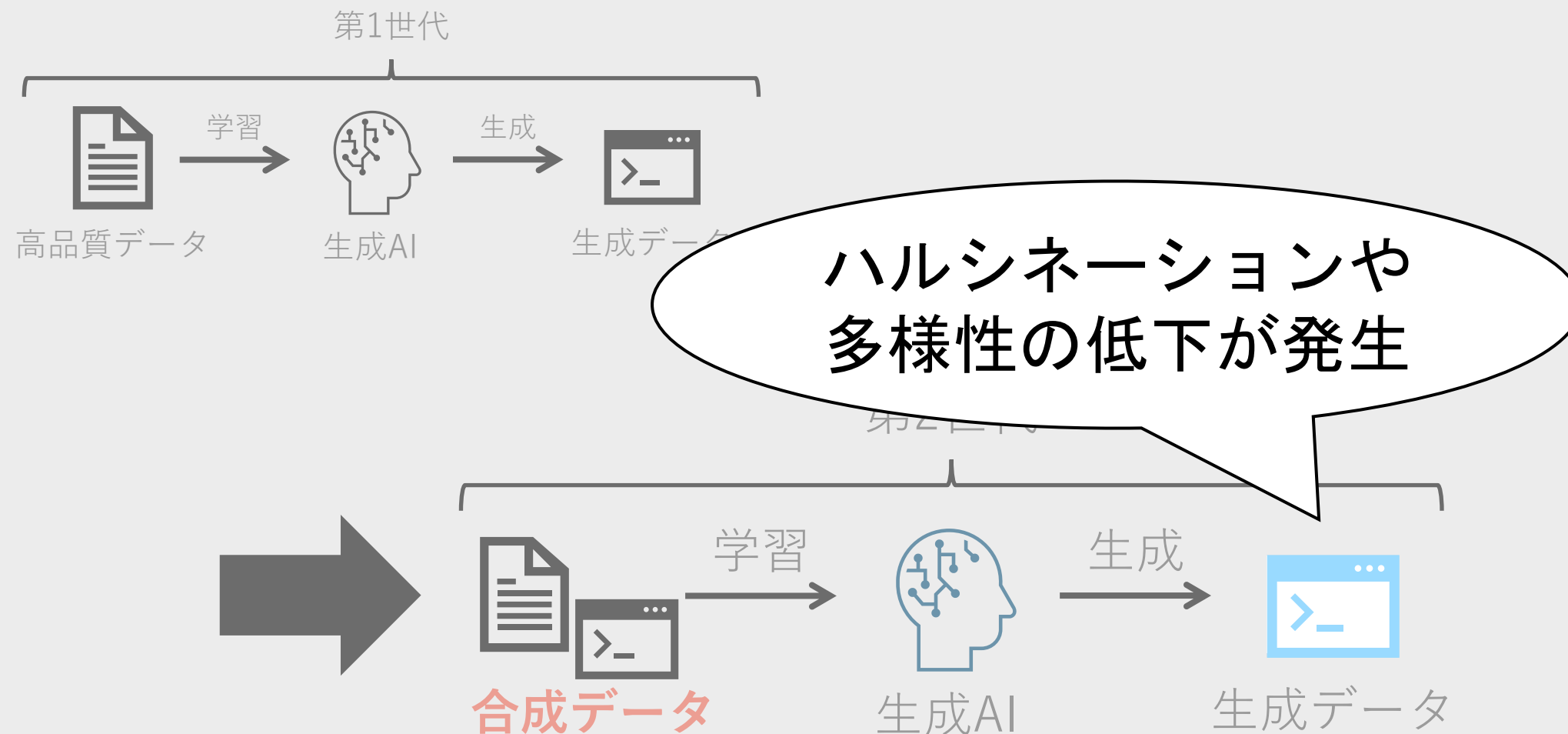
モデル崩壊



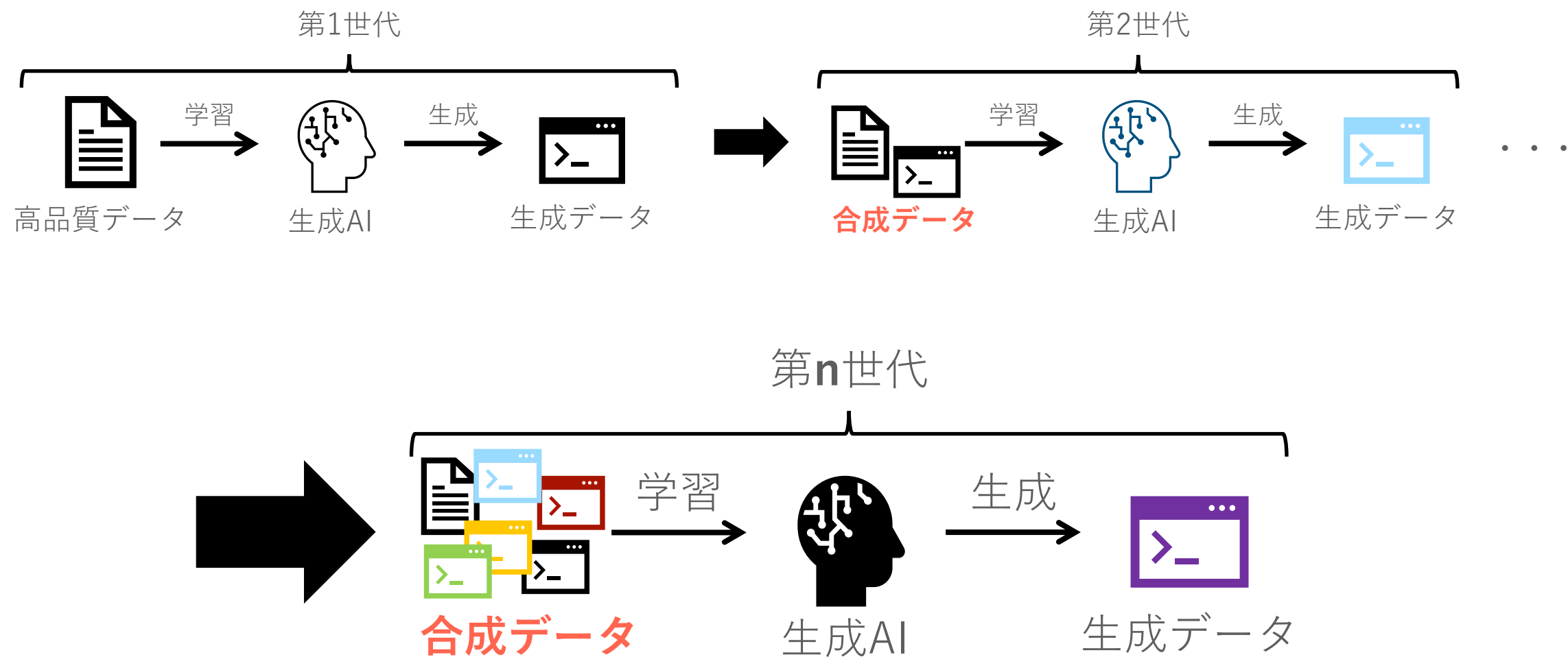
モデル崩壊



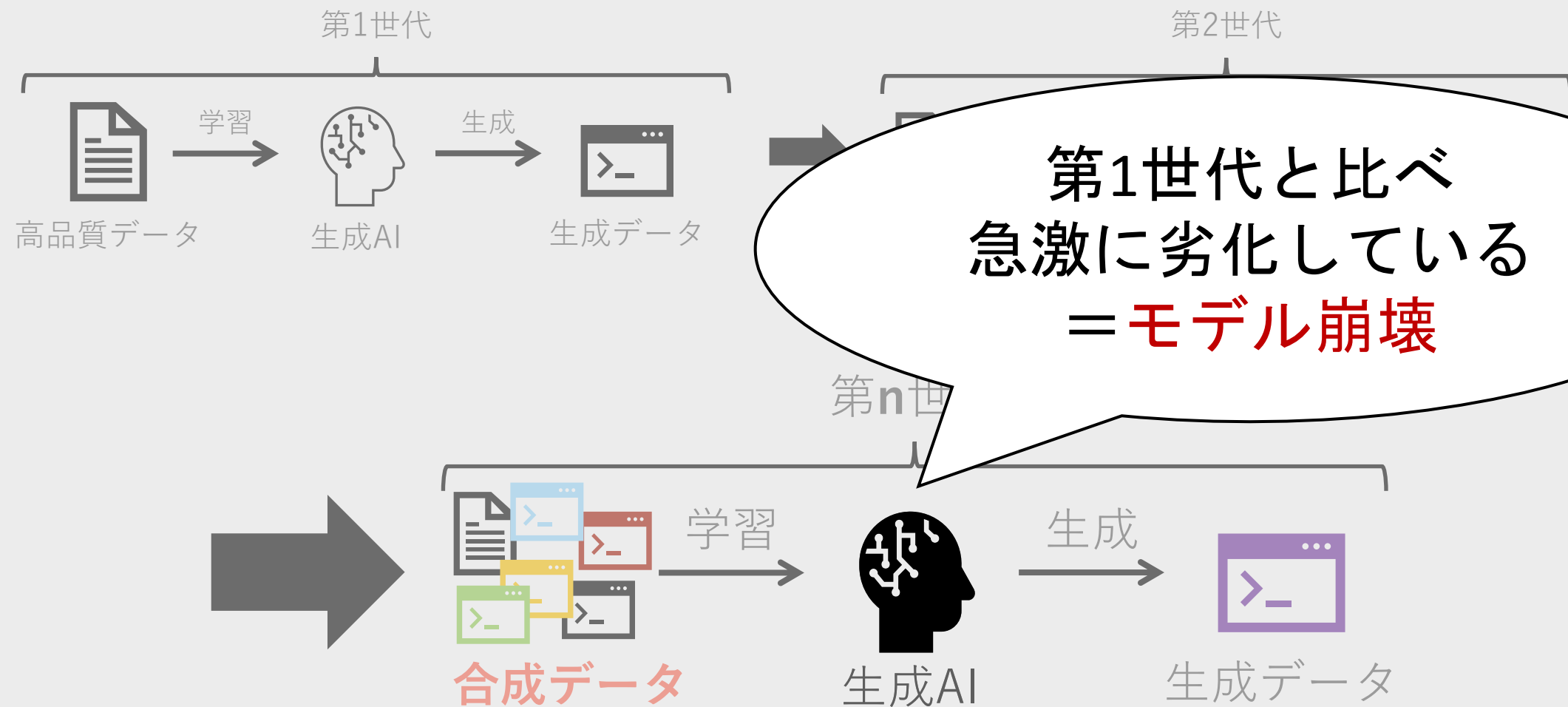
モデル崩壊



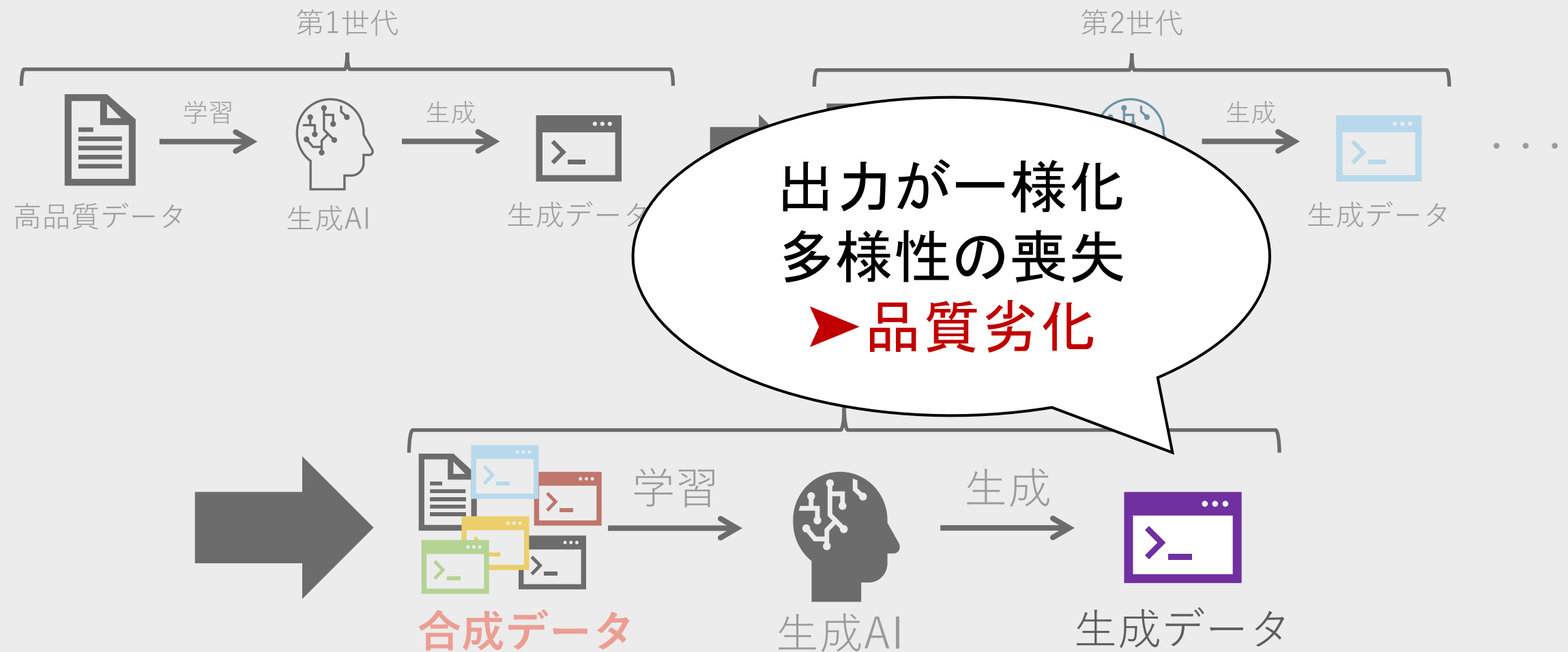
モデル崩壊



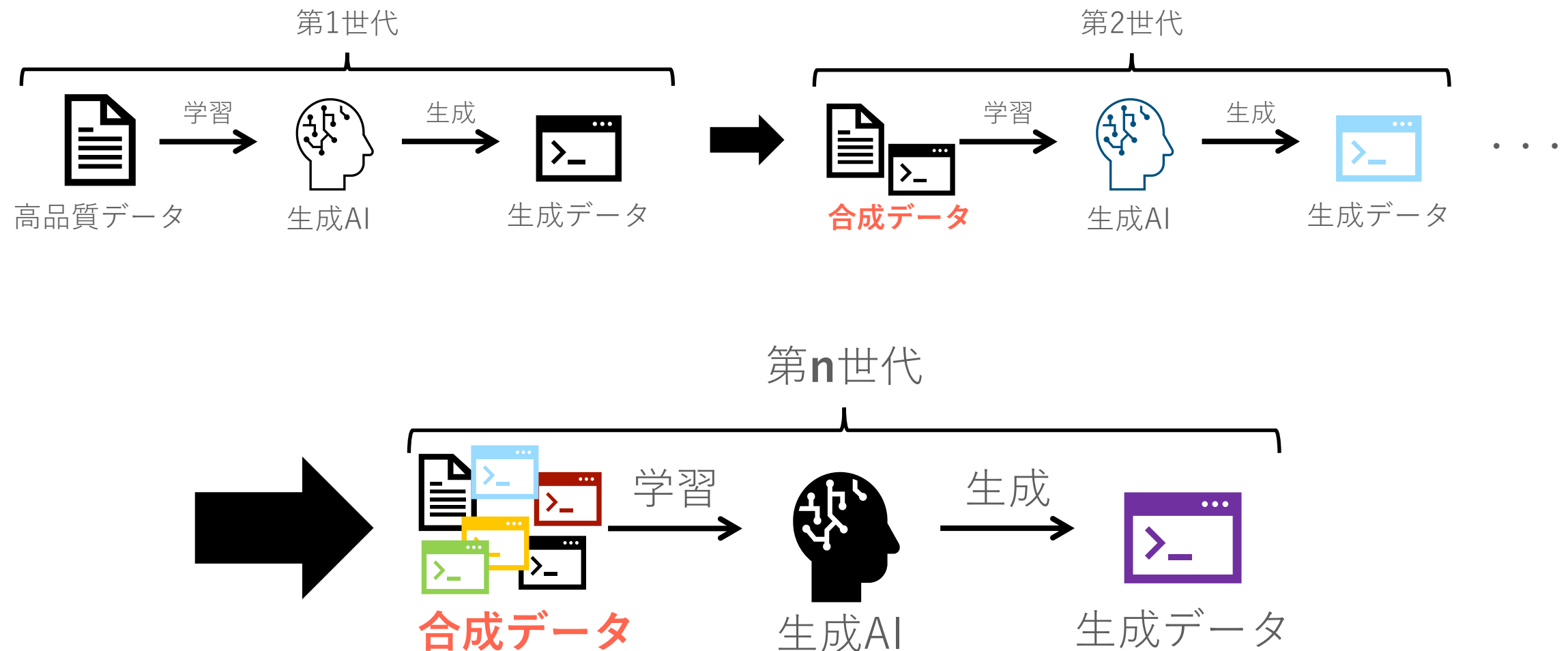
モデル崩壊



モデル崩壊



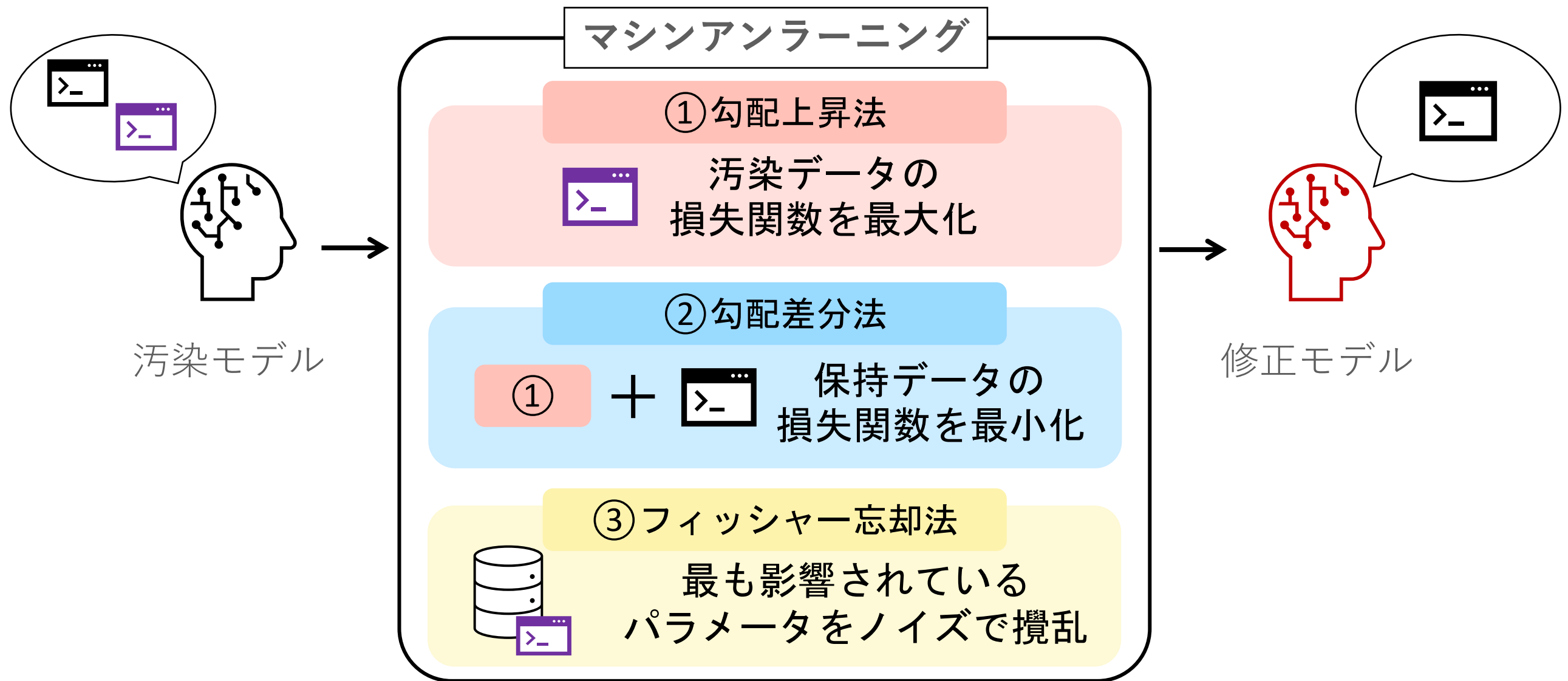
モデル崩壊



合成データから汚染されている部分を除去できればいいのでは

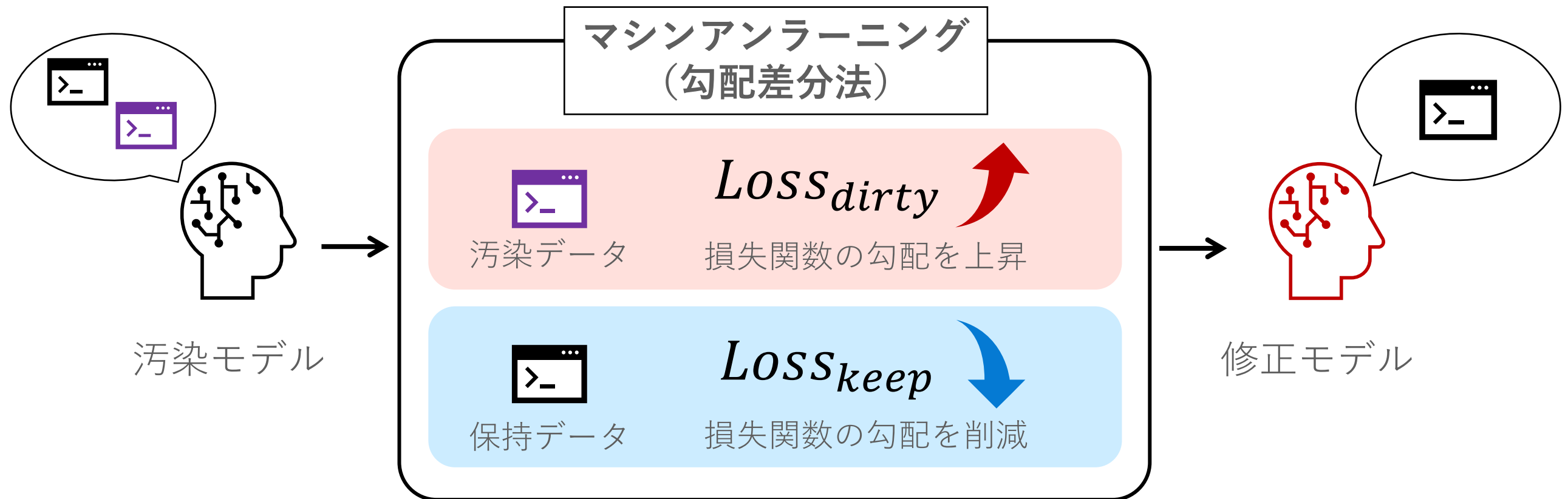
マシンアンラーニング

学習したデータから特定のデータを忘却させる技術



勾配差分法

各損失関数を最適化することで忘却させる手法



- ・ 損失関数が**増加** ➤ モデルがデータを忘却する
- ・ 損失関数が**減少** ➤ モデルがデータを学習する

既存研究

コード生成モデルを対象に勾配差分法を実施

- ・ データ内の機密情報の削除を目的に実施
- ・ 情報の漏洩率を指標としている

例) 漏洩率10%の場合：100回プロンプトを与えると，10回情報漏洩する

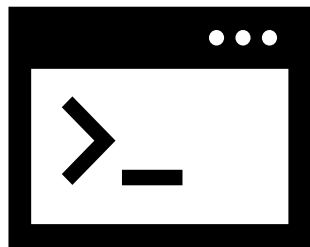
モデル	漏洩率 (適用前/適用後)	削減率	Pass@1 (適用前/適用後)
CodeLlama-34B	32.7% / 5.0%	84.7%	75.8% / 74.6%
CodeLlama-14B	26.2% / 4.7%	82.1%	74.5% / 73.3%
CodeLlama-7B	28.6% / 5.6%	80.4%	69.3% / 68.8%
AIXCoder-7B	23.2% / 13.1%	43.5%	60.0% / 57.0%

- ・ コード生成能力を最大90%以上保ちつつ漏洩を最大84.7%削減することができる

既存研究との違い

アンラーニングを行っている範囲が異なる

- 既存研究：個人情報について（文字列レベル）



```
def print_personal_info():
```

```
    full_name = 'Oliver Lee'
```

```
    gender = 'Male'
```

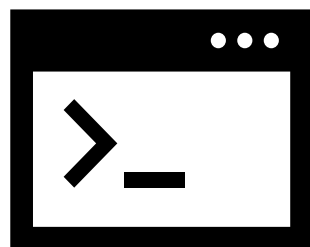
```
    country = 'USA'
```

```
    birthday = 'June 22, 1995'
```

```
    address = 'aaaaa@icloud.com'
```

対象箇所

- 今回の対象範囲：特定リポジトリのコード（リポジトリレベル）



```
def load_config(filipath):
```

```
    ~
```

```
    return json.load(f)
```

```
def parse_arguments(args=None):
```

```
    ~
```

```
    return parser(args)
```

対象箇所

研究目的・リサーチクエスチョン

本研究の目的

モデル崩壊を起こしたコード生成モデルの回復手段として
マシンアンラーニングの有効性を検証する

リサーチクエスチョン

RQ1：どの汚染が崩壊を進行させ，除去でどれだけ抑制できるか

- ・汚染データの種類によって崩壊への影響度は異なるのか

RQ2：合成データの影響を効果的に除去できるか

- ・ Pass@1がアンラーニング後も低下しないのか検証

RQ3：マシンアンラーニングは完全再訓練と比べてどの程度か

- ・ アンラーニング後モデルと完全再訓練モデルを比較

評価指標

PPL（パープレキシティ）

- ・ モデルがあるデータをどれだけ知っているのかのエントロピー
 - ・ PPLが高い = データを忘れている

$$PPL = \exp\left(-\frac{1}{n} \sum_{i=1}^n \log P(x_i | x_1, \dots, x_{i-1})\right)$$

$P(x_i | x_1, \dots, x_{i-1})$: それまでの文脈が与えられたときに,
 i 番目のトークン x_i が出現する確率

x_i : 各トークン

Pass@1

- ・ 生成するコードが全テストケースを通過するか

CodeBLEU

- ・ 生成コードが参照コードとどれだけ類似しているか

使用モデル・データセット（仮）

| CodeLlama-7B, CodeSearchNetを使用

- 評価指標に対応できるモデルを現在模索中

今後の予定

マシンアンラーニングについての既存研究を確認中

- ・対象タスクがSEタスクである研究が少ない

マシンアンラーニング自体を実施

- ・自身の手元で行えるか確認