テーマ発表

分散表現を用いたDockerfileからの記述パターンの抽出

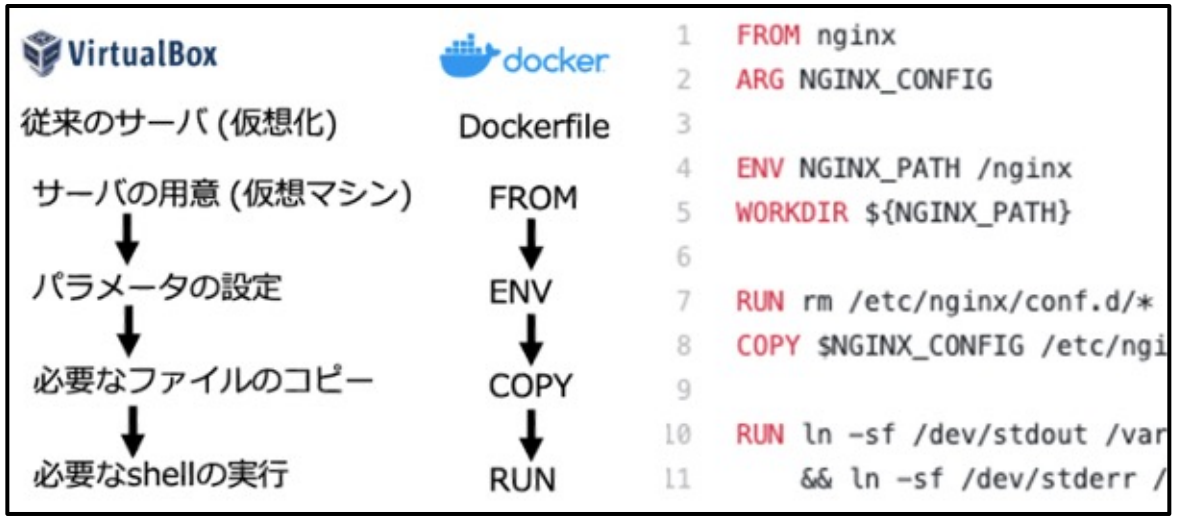
ソフトウェア基礎技術研究室

中村碧海

研究背景

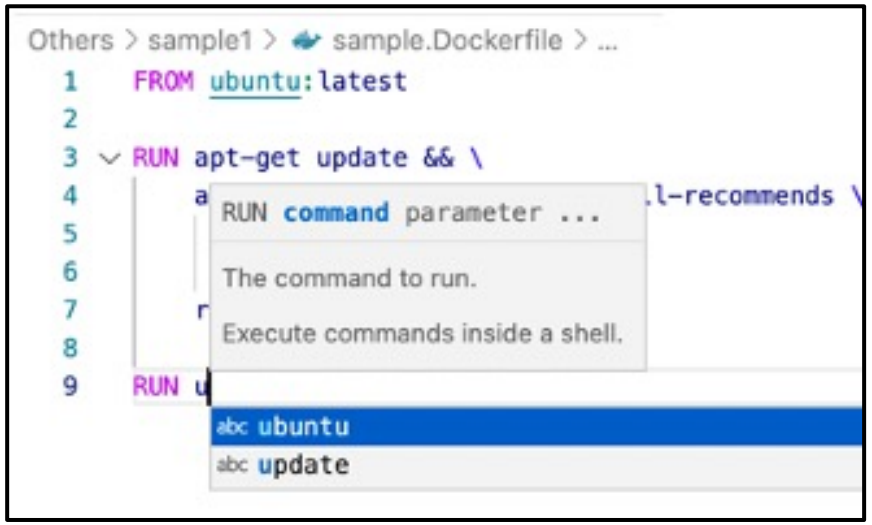
- コンテナ型の仮想化技術としてDockerが注目
  - パッケージングされたランタイムの実行が可能

- コンテナを構築する手法のひとつにDockerfileを用いた手法がある



  - コンテナの構築手順をコードで管理
  - コンテナのバージョン管理や共有, 配布が可能

- 問題点
  - IDEにおけるコード補完機能が極端に弱い

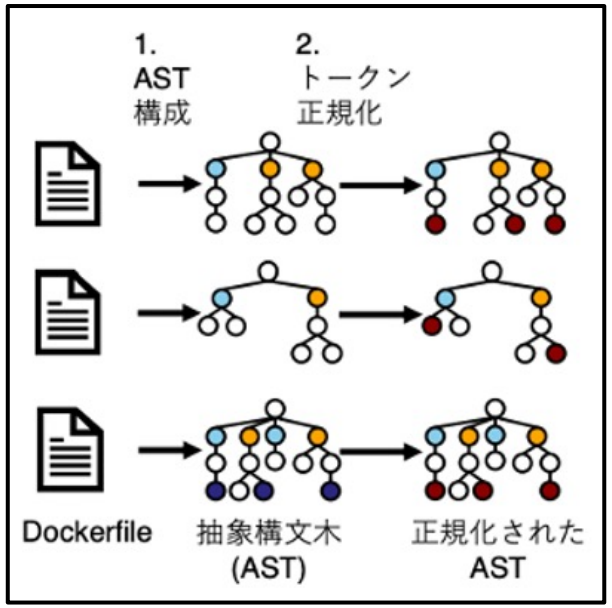


  - Dockerfileの記述の支援としては不十分

既存研究

◘Type-2のコードクローンを検出

- ■ 構文解析した後に, 正規化処理を施しクローンを検出



◘問題点

- ■ 構文解析を頻繁に利用される上位50個のBashに限定している
- ■ 揺らぎに対して, 人的処理(正規化)で対応

→ 汎用性に欠ける

◘構文解析の限界

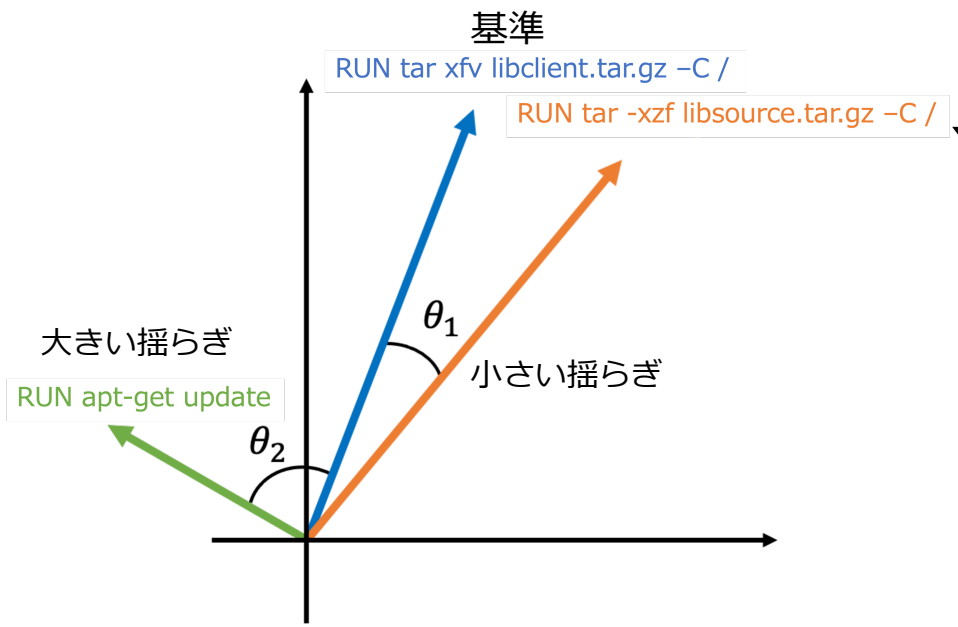
- ■ Dockerfile内に複数の言語を組み込むことができるため

```
Dockerfile > ...
1   FROM python:3.8.3
2
3   RUN apt-get update \
4       && apt-get install -y --no-install-recommends \
5       git \
6       locales \
7       locales-all \
8       pandoc \
9       && rm -rf /var/lib/apt/lists/*
10
11  RUN curl https://sh.rustup.rs -sSf | sh -s -- -y
12  RUN ~/.cargo/bin/cargo install fd-find sd && ~/.cargo/bin/cargo
13
14  RUN pip install --no-cache-dir --upgrade pip
15  RUN pip install --no-cache-dir torch==1.5.1+cpu torchvision==0.6
16
```

- ■ 言語別の構文解析器の用意や対応には無理がある

アプローチ

- 潜在表現(ベクトル)を利用
  - ある程度の記述の揺らぎを許容
  - 完全一致の必要もない



インデントの
親子関係の処理

```
----------変換前----------
COPY scripts/sccache.sh /scripts/
RUN set -ex; \
    \
親  find /usr/local -depth \
        \( \
      子  \( -type d -a \( -name test -o -name tests -o -name
            -o \
            \( -type f -a \( -name '*.pyc' -o -name '*.pyo' \)
        \) -exec rm -rf '{}' +; \
    rm -f get-pip.py

----------変換後----------
["COPY", "scripts/sccache.sh", "/scripts/"],
["RUN", "set", "-ex", "AND", "BACK"],
["RUN", "find", "/usr/local", "-depth"],
["RUN", "find", "/usr/local", "-depth", "-type", "d", "-a", "-
["RUN", "find", "/usr/local", "-depth", "-type", "d", "-a", "-
["RUN", "rm", "-rf", "get-pip.py"]
```

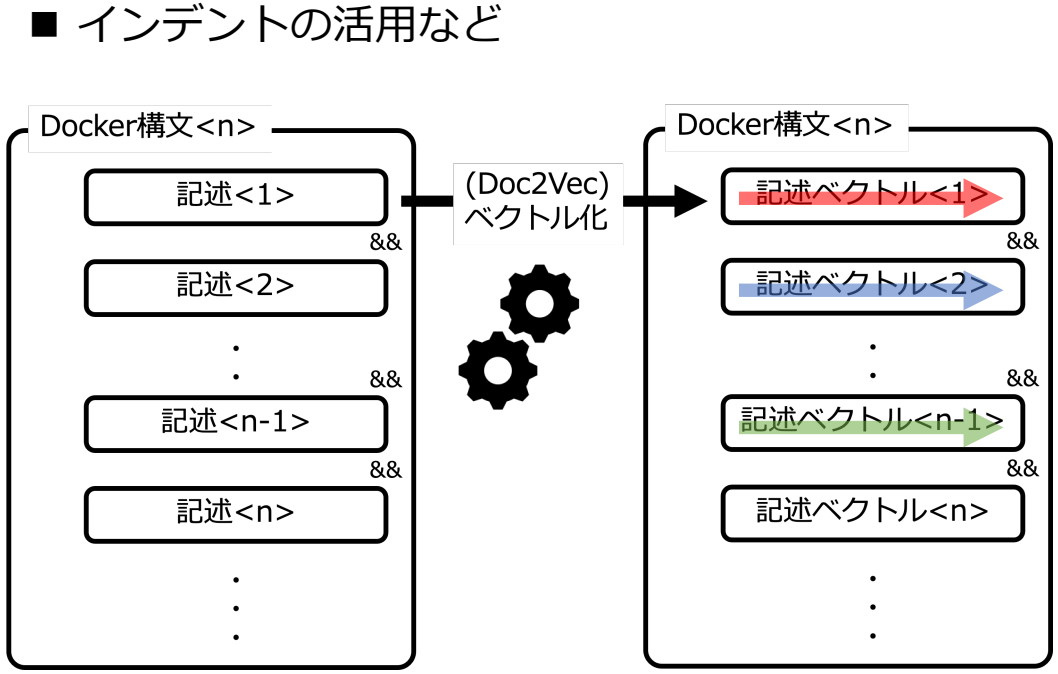
- 効果
  - 潜在的な構造や, 依存関係の検出を期待
  - オプションの順序などの振る舞いに与える影響の少ない記述も許容
  - パターンを限定する必要がないため, 汎用性も高い

手法の概要

1. 潜在表現(ベクトル)への変換

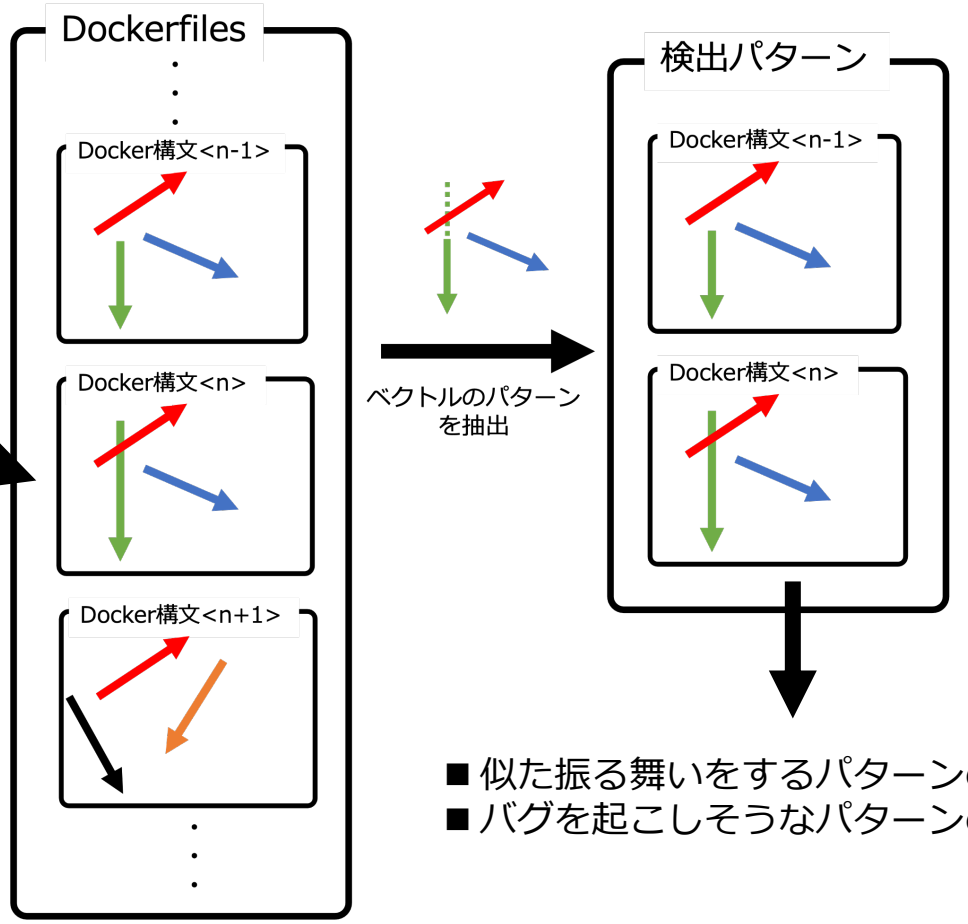
■ Doc2Vecを拡張したDock2Vecを提案

■ インデントの活用など



2. パターンの検出

■ ベクトルの組み合わせからパターンを検出



■ 似た振る舞いをするパターンの検出
■ バグを起こしそうなパターンの検出など

予備実験

◘ Dock2Vecを用いた検出例

◘準備

- ■ binnacle-icse2020 -> ”apt-get”
- ■ 約75000個のDockerfileを対象

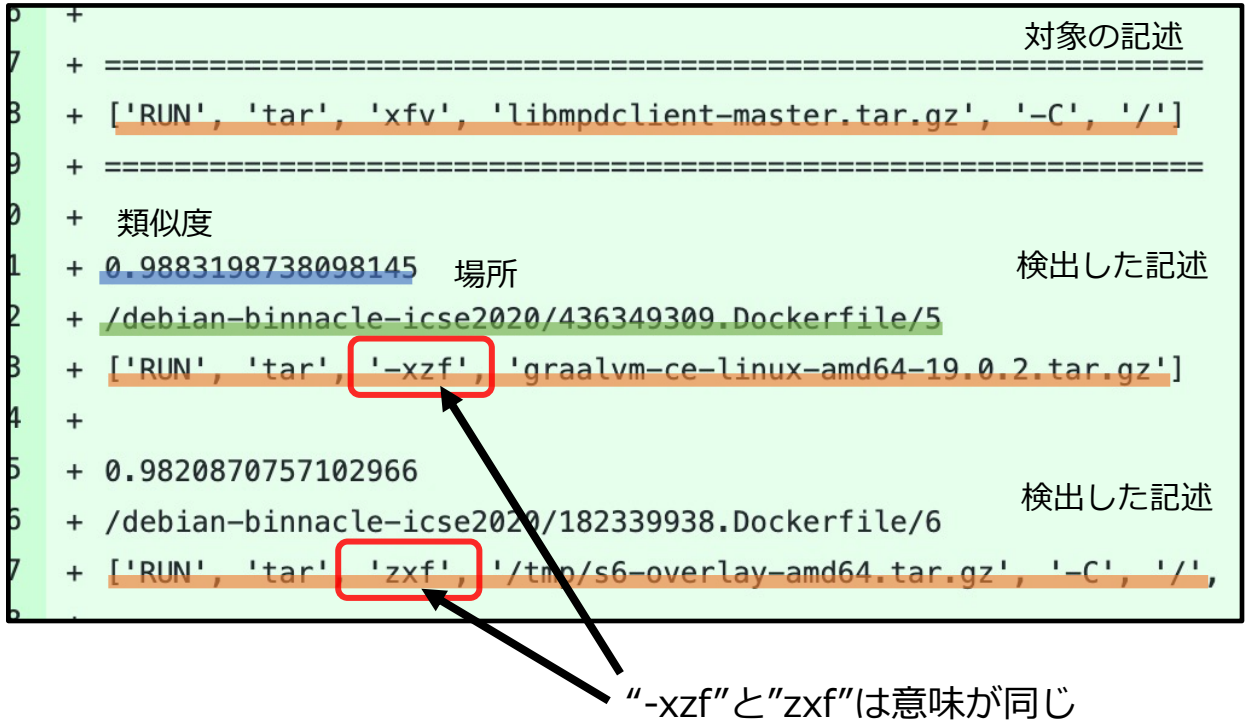
◘手順

1. Doc2Vecを利用して記述をベクトルに変換
2. コサイン類似度を利用して, 類似した記述を取得

◘結果

- ■ 揺らぎを許容していそうな記述を取得可能
- ■ 他の記述でも同様の結果

◘ 結果



今後の予定

- 調査
  - 適切な学習モデルを選択をするために文献を調査
- 検証
  - 分散表現での記述の揺らぎの許容
  - 類似したパターンの取得
  - 学習モデルの選定