

ソースコードと ドキュメントを紐づける 特徴の固有表現抽出

発表者 中谷祐太

背景

ドキュメント

ソフトウェアの機能,使用方法等についてまとめられたもの

```
./bin/pyspark
```

Or if PySpark is installed with pip in your current environment:

```
pyspark
```

Spark's primary abstraction is a distributed collection of items called a Dataset. Datasets can be created from Hadoop InputFormats (such as HDFS files) or by transforming other Datasets. Due to Python's dynamic nature, we don't need the Dataset to be strongly-typed in Python. As a result, all Datasets in Python are Dataset[Row], and we call it DataFrame to be consistent with the data frame concept in Pandas and R. Let's make a new DataFrame from the text of the README file in the Spark source directory:

ドキュメントの例

背景

ソースコードが更新されるとドキュメントも更新

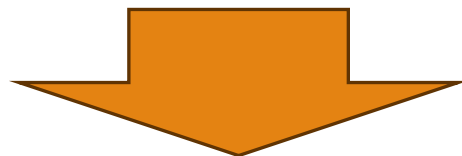
- しかし、ドキュメントの更新が遅れることがある

原因

開発者が更新すべきドキュメントを把握できていない

背景

ドキュメントとソースコードを明示的に紐づけ



紐づけるために必要な要素を抽出

- 正規表現による抽出(先行研究)
 - 実装が比較的簡単
 - 決まったパターンのみを抽出
- NER(固有表現抽出)
 - 文脈などで柔軟に抽出

研究目的と手法

研究目的

ドキュメントとソースコードを紐づけるための要素を抽出する

アプローチ

正規表現

NER(固有表現抽出)

データセット

先行研究のjira issueデータセットから以下の条件でフィルタ

1. issueのタグがdocのもの
2. 別のissueへの引用があるもの
3. 引用の数が1つだけのもの
4. 両方にプルリクエストがあるもの
5. docのプルリクにmdファイル,codeのプルリクにjava,R,scala,pyのいずれかがあるもの

フィルタ	Issue件数
1	2209
2	488
3	151
4	74
5	28

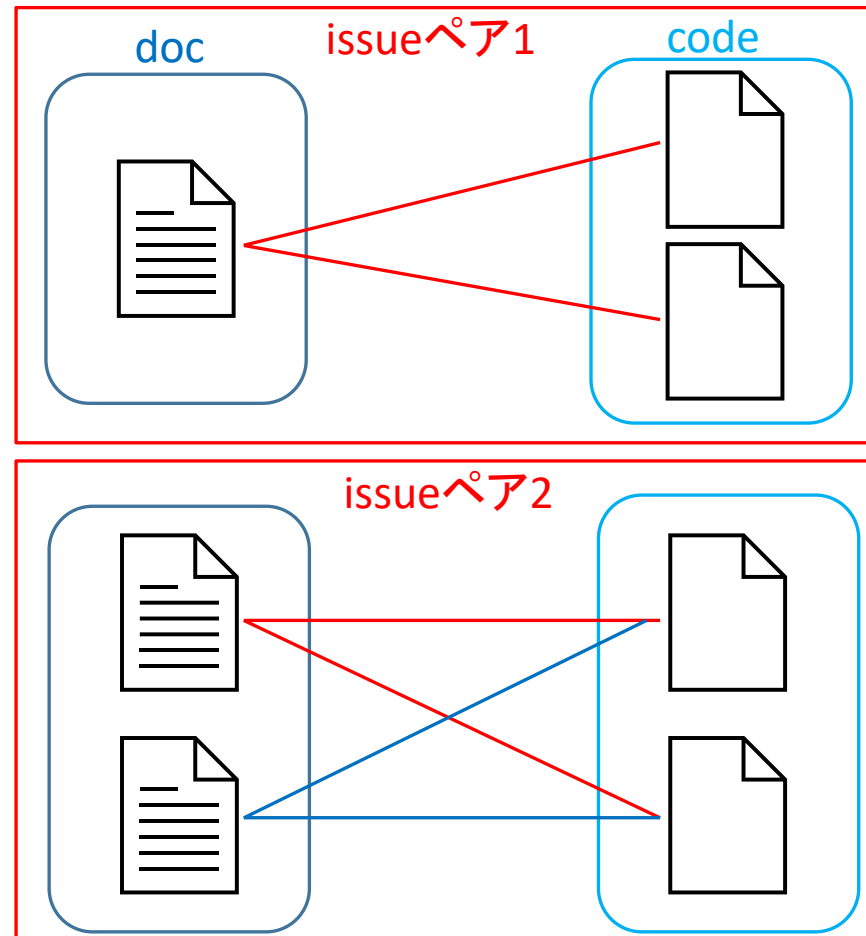
データセット

- issueペアに紐づいているプルリクからmd,java,R,scala,pyファイルのdiffを取り出す

	md	py	java	R	scala
ファイル数	34	13	0	18	145

- 同一issueペアに存在するmdファイルとcodeのファイルを一対一のdiffペアとして扱う

206件のdiffペアを収集



正解集合の収集

diffペアに以下の処理を行う

1. docのdiffをnltkでトークン化
2. 以下の条件に当てはまるトークンを除外
 - 1文字だけのもの
 - 品詞タグがCC,TO,INのもの
3. トークンがcodeのdiffに存在するか確認(大文字,小文字の区別なし)

全体で1112種類の単語を正解集合とした

各アプローチでの予測

mdファイル34件のdiffに対し各アプローチで単語を抽出

- 正規表現
- NER
 - bert-base-NER
 - codebert-base-finetuned-stackoverflow-ner

抽出した単語から以下の条件で除外

- 1文字だけのもの
- 正解集合を収集する際に除外した単語

	検出単語数
正規表現	622
bert-base	146
codebert	926

結果

	正解集合	検出単語数	TP	precision	recall	F1
正規表現	1112	622	137	0.220	0.123	0.158
bert-base	1112	146	24	0.164	0.022	0.039
codebert	1112	926	175	0.189	0.157	0.172
正規表現+ codebert	1112	1500	284	0.189	0.255	0.217

$$precision = \frac{TP}{\text{検出単語数}}$$
$$recall = \frac{TP}{\text{正解集合}}$$
$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

- 正規表現はprecisionが高く,codebertはrecallが高い

今後の予定

- 各手法で検出できた単語の傾向の調査
- 各手法で検出できなかった単語の傾向の調査