

ライセンスルール自動生成を目的とした ライセンス記述パターン抽出手法

2016 07/26 研究発表

東 裕之輔

OSSの再利用

- 企業では、開発コスト削減のため、オープンソースソフトウェア(OSS)を再利用した派生開発が行なわれている

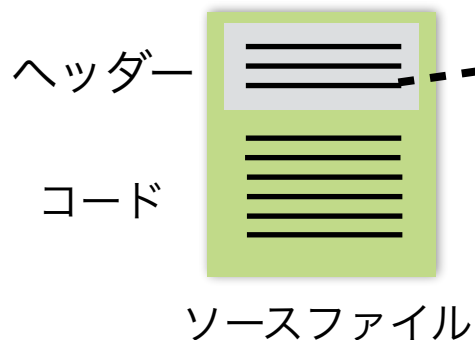


- OSSは、ソースファイルごとに決定されている
OSSライセンスを遵守することで再利用することができる

ソースファイルのライセンス

- OSSライセンスは、ソースファイルのヘッダー部分に記述されている

BSD2のライセンス記述



```
#####  
# Redistribution and use in source and binary forms, with or without  
# modification, are permitted provided that the following conditions are met:  
#  
# 1. Redistributions of source code must retain the above copyright notice,  
#   this list of conditions and the following disclaimer.  
# 2. Redistributions in binary form must reproduce the above copyright  
#   notice, this list of conditions and the following disclaimer  
#   in the documentation and/or other materials provided  
#   with the distribution.  
#  
#####
```

- OSSのライセンスを遵守するには、全てのソースファイルのライセンスを確認しなければならない

OSSに含まれるソースファイル数

- OSSには、多くのソースファイルが含まれている



Linux

ドメイン : OS

ソースファイル数 : 22,310



ドメイン : データベース

ソースファイル数 : 1,452



ドメイン : サーバー

ソースファイル数 : 291

- 再利用するOSSに含まれる、全てのソースファイルのライセンスを手作業で確認するのは、労力がかかる
 - Linux, MySQL, Apacheを再利用すると24,000ファイル以上のライセンス記述を確認しなければならない

ライセンス特定ツール[1][2]

- ライセンスルール（正規表現）を用いて，自動的にソースファイルのライセンスを特定する



[1] Ninka : Daniel M German, Yuki Manabe, and Katsuro Inoue, "A Sentence-Matching Method for Automatic License Identification of Source Code Files" In Proc. of ASE'10, pp. 437–446

[2] Fossology : Robert Gobeille, "The fossology project," In Proc. of MSR'08, 2008, pp. 47–50.

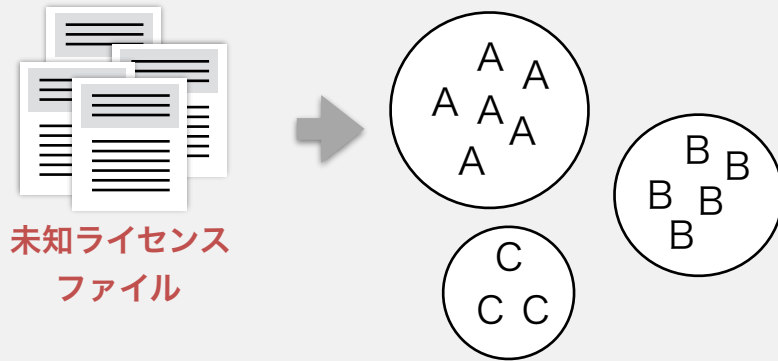
ライセンス特定ツールの課題

- 新しいライセンスや、未知の表記ゆれは、年々増加する
- ライセンス特定ツールは、**コングスタント**に新しいルールを追加し続けていくことが重要



手作業でのライセンスルール作成手順

STEP1. 未知ライセンスファイル集合を ライセンス毎に分類



STEP2. 各クラスターの頻出記述と 表記ゆれを調査する

頻出記述

This[...] is free software.
it under the term of the A license

[...]部の表記ゆれ

1	program
2	software
3	library

STEP3. ライセンスルール作成

1	This program is free ~
2	This software is free~
3	This library is free~

新ライセンスルール

This (program|software|library) is ~

問題点

大量の未知ライセンスファイルが検出
された場合、大変な労力がかかる

最終的なゴール

✧ ゴール：ライセンスルール自動生成手法の構築

✧ アプローチ

- ライセンスルール作成手順における各ステップを、機械的処理に置き換える

ライセンス記述クラスタリング手法[3]を提案

1.未知ライセンス記述の分類



2.頻出記述と表記ゆれの調査



3.正規表現の作成



階層クラスタリング

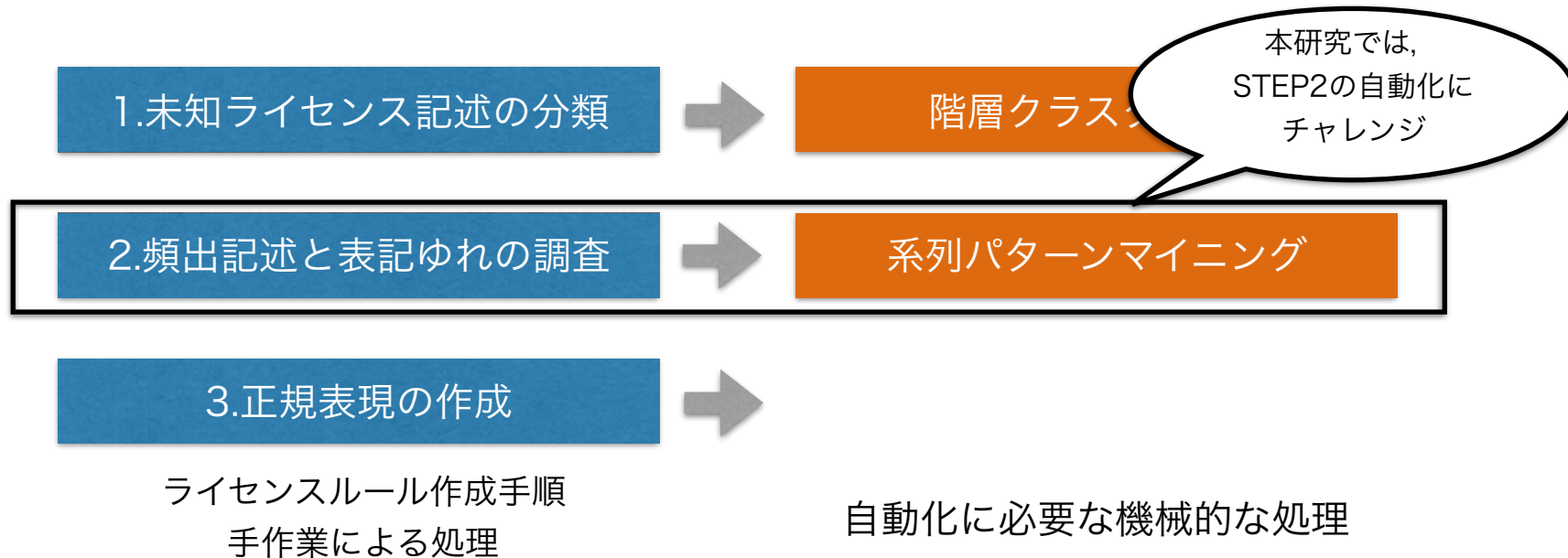
ライセンスルール作成手順
手作業による処理

自動化に必要な機械的な処理

[3] Yunosuke Higashi, Yuki Manabe, and Masao Ohira. Clustering OSS license statements toward automatic generation of license rules. In Proc. of IWESEP'16 pp. 30–35, 3 2016.

最終的なゴール

- ✧ ゴール：ライセンスルール自動生成手法の構築
- ✧ アプローチ
 - ライセンスルール作成手順における各ステップを、機械的処理に置き換える



本研究の目的

- 目的：既存手法[3]により作成された未知ライセンスファイルのクラスタから**ライセンス記述パターン**を抽出する
- ライセンス記述パターンとは
 - 各クラスタに頻出する語句の系列パターン

クラスタAのライセンス記述	
	This program is free software. it under the term of the A license
	This software is free software. it under the term of the A license
	This library is is free software. it under the term of the A license



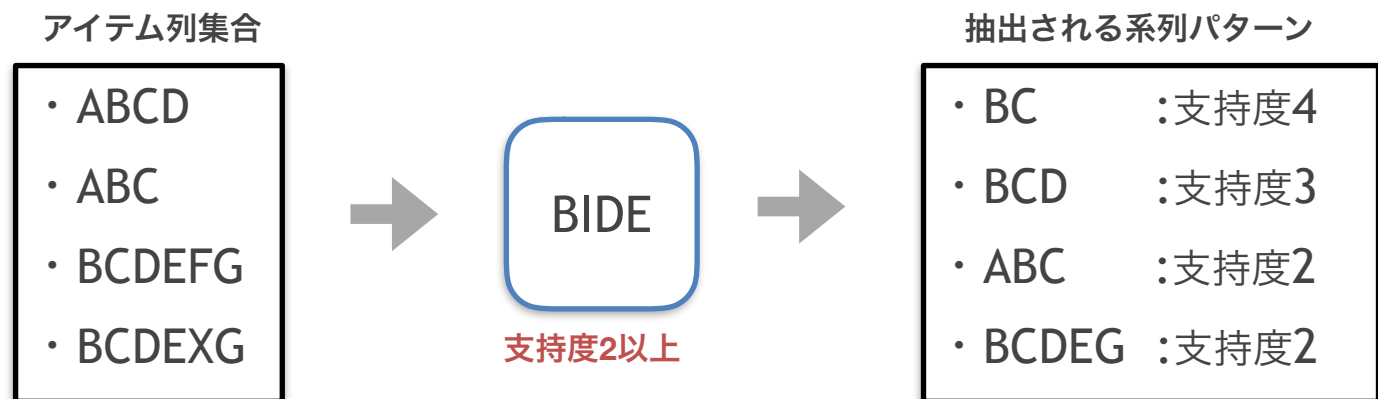
ライセンス記述パターン

This is free software.
it under the term of the A license

- アプローチとして**系列パターンマイニング**を用いる

系列パターンマイニング

- 大量のアイテム列に頻出する系列（時系列）パターンを抽出する
 - 系列パターンの特徴：出現順序が同じであれば、間に無関係な要素が追加されても抽出されるパターンは変わらない
- BIDEアルゴリズム
 - 発生回数が同じで、かつ他の抽出したパターンの部分集合となるパターンを抽出しない



Step2.自動化における技術的課題1

- 1.単一のライセンスからなるクラスタから記述パターンを抽出したい
 - 複数のライセンスから成り立つクラスタからは、複数のライセンスを特定してしまう不適切なライセンスルールを生成する



- 既存手法は、クラスタリングを用いているため、複数のライセンスから成るクラスタをどうしても作成してしまう

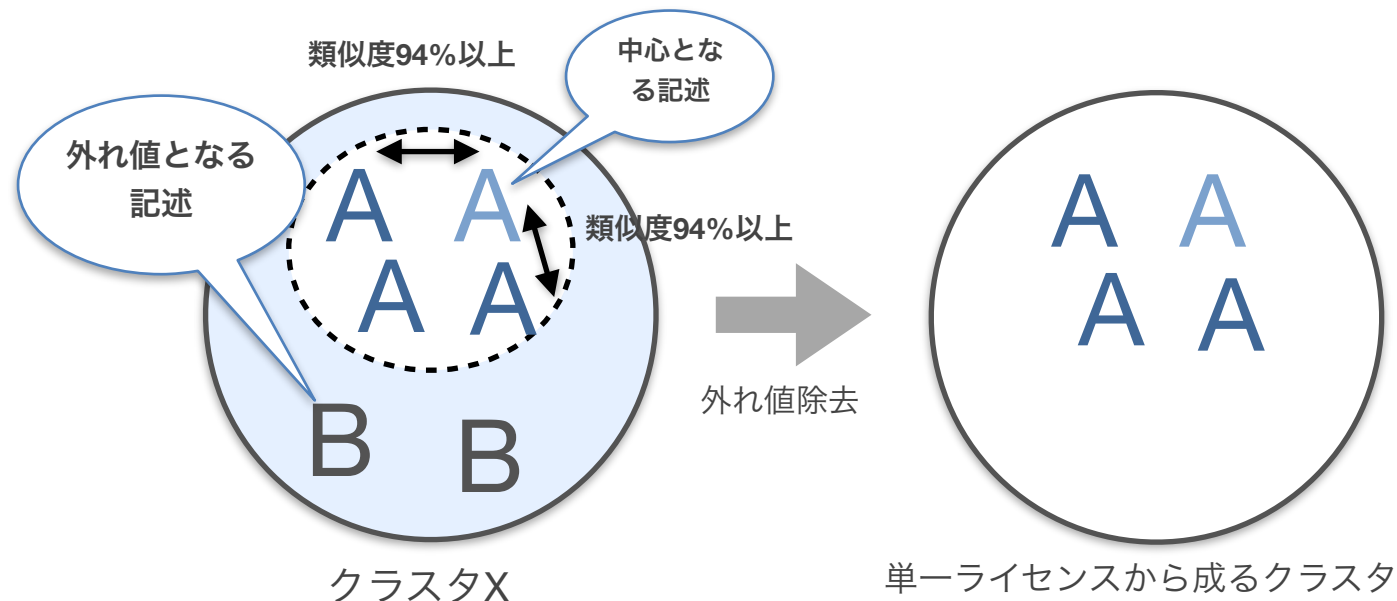
Step2.自動化における技術的課題2

- 2.ライセンスルール作成に必要なパターンを抽出したい
 - 系列パターンマイニングの欠点は、大量の系列パターンを出力してしまう
 - “ of the ”のようなパターンは、どのライセンス記述ではなくても成り立ってしまうため、**ルール作成には不必要である**
 - 1種類のライセンスでのみ成立する記述パターンを抽出できるようにしたい

BIDEで抽出されたパターン	有用/不要	支持度	ライセンス
of the	不要	8	MPLv2.0
This Source Code Form is subject to the terms of the Mozilla Public License, v<dot> 2<dot>0.	有用	4	MPLv2.0
2 - This Source Code Form is subject to the terms of the Mozilla Public License, v<dot> 2<dot>0.	有用	3	MPLv2.0
If a copy of the MPL was not distributed with this file, You can obtain one at http<colon>//mozilla.org/MPL/2.0/ .	有用	4	MPLv2.0

技術的課題1 解決のアプローチ

- クラスタ内で大きく異なる記述を外れ値としてフィルタリング
 - クラスタの中心となる記述グループを検出する
 - クラスタの中心となる記述：編集距離による類似度が**94%**以上ある記述を最も多く持つ記述（予備調査により決定）



編集距離

- 編集距離とは、ある文字列Aから、文字列Bにするまで、何回の操作(削除, 置換, 挿入)が必要となるか
 - 例 kittenとsitting 距離=4
 - 1 「kitten」
 - 2 「sitten」 (「k」を「s」に置換)
 - 3 「sittin」 (「e」を「i」に置換)
 - 4 「sitting」 (「g」を挿入して終了)
- 本手法では、距離を最大文字数で**正規化**した編集距離類似度を用いる
 - バージョンなどの細かい差異を区別することができる

$$NLD = 1 - \frac{LD(A, B)}{\text{maxlength}(A, B)}$$

NLD : 正規化された編集距離
LD : 編集距離
maxlength : 最大字数

技術的課題2解決のアプローチ

- 他の系列パターンの部分集合になるような系列パターンを削除する
 - BIDEは、異なる支持度の系列パターン間では、他の部分集合となる系列パターンを出力してしまう

有用/不要	グループAから生成されたパターン
不要	of the
有用	This Source Code Form is subject to the terms of the Mozilla Public License, v2.0.
有用	2 - This Source Code Form is subject to the terms of the Mozilla Public License, v2.0.
有用	If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/

部分集合
部分集合
部分集合となる
パターンを削除



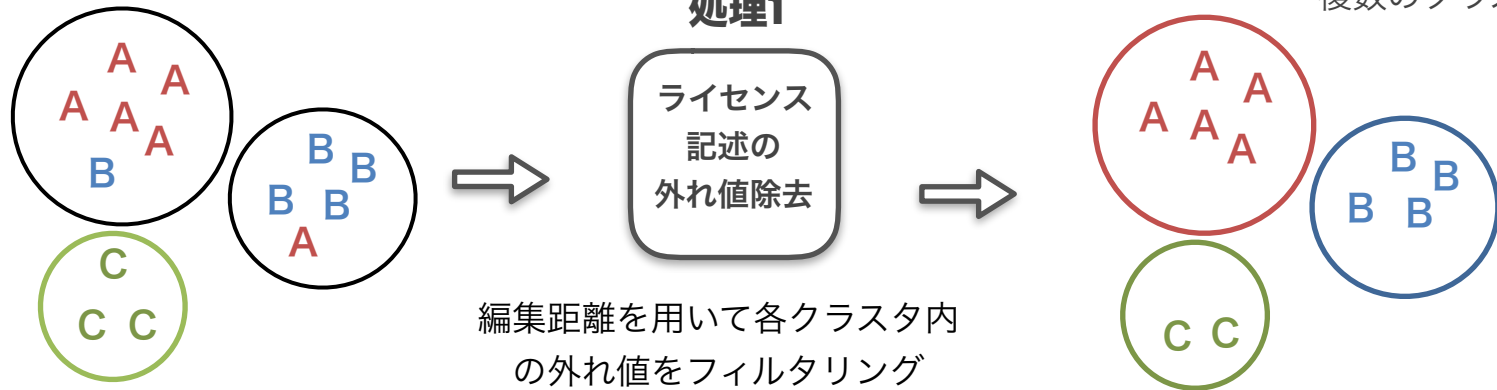
有用/不要	グループAから生成されたパターン
有用	2 - This Source Code Form is subject to the terms of the Mozilla Public License, v2.0.
有用	If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/

- 他の系列パターンの部分集合になるような系列パターンを削除することで、不要なパターンを減らす

提案手法の概要

既存手法のクラスタリングによって分類された未知ライセンスファイルのクラスタ

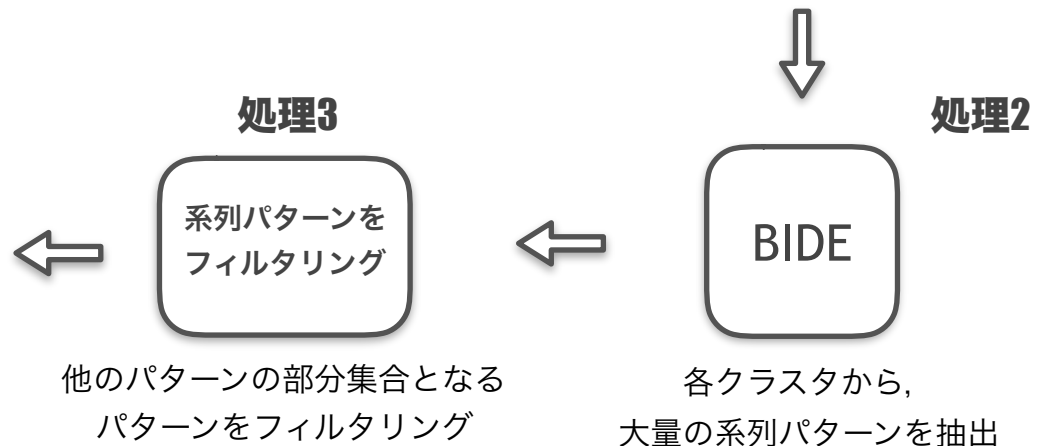
外れ値を除去した複数のクラスタ



グループAから生成されたパターン
2 - This Source Code Form is subject to the terms of the Mozilla Public License, v.2.0.
If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/ .

グループBから生成されたパターン

グループCから生成されたパターン



ケーススタディ

- 目的：提案手法は、2つの技術的課題をどの程度クリアできるかどうかを検証する
 - 課題1.単一のライセンスからなるクラスタからライセンス記述パターンを抽出したい
 - 課題2.ライセンスルール作成に必要なライセンス記述パターンを抽出したい
- データセット
 - Debian v7.8.0で検出された2,446の未知ライセンスファイルを既存手法によりクラスタリングした結果を用いる

Debian-7.8.0	クラスタ数	単一のライセンスから成るクラスタ	複数のライセンスから成るクラスタ
2,446ファイル	246	178(72%)	69(28%)

実験方法

- 課題1解決の評価実験方法

1. データセットにおける各クラスタについて大きく異なるライセンス記述をフィルタリングする
 2. 単一ライセンスから成るクラスタ, 複数のライセンスから成るクラスタの数とその割合を調べる
- フィルタリングされたファイルや, 類似度が94%以上であるライセンス記述が存在しなかったクラスタは, **全て単一のファイル** (記述パターンを作成できない) として扱う

- 課題2解決の評価実験方法

1. データセットにおける各クラスタについて大きく異なるライセンス記述をフィルタリングした後, 各クラスタにBIDEを適用
2. BIDEによって出力された記述パターンのうち, 1種類のライセンスのみに存在するものがどれくらいあるか, データセットを使って調査する
3. 記述パターンをフィルタリングした後, 同じ調査を行い, 1種類のライセンスのみに存在する記述パターンの割合を調査する

課題1 解決の実験結果

Debian-7.8.0	外れ値除去	クラスタ数	単一のライセンス から成るクラスタ	複数のライセンス から成るクラスタ	単一ファイル
2,446ファイル	データセット	246	178(72%)	69(28%)	48
	外れ値除去後	127	126(99%)	1(1%)	2008

- データセットの99%のクラスタをSLCにすることができた
- ただし、フィルタリングにより、大量の単一ファイルが出力されてしまっている
 - 記述パターンを抽出できるファイル数が減少する

課題2解決の実験結果

	生成されたパターン数	1種類のライセンスのみに存在したパターン数	割合
フィルタリング前	1498	375	25%
フィルタリング後	371	276	74%

提案手法が抽出したライセンス記述パターンの一部

#Licは、ライセンス記述パターンが成立するライセンスの種類数を示す。

記述パターン	ライセンス名	#Lic
See the License for the specific language governing rights and limitat[...]	AOLserver	6
If you do not delete the provisions above, a recipient may use your ve[...]	AOLserver	1
If you wish to allow use of your version of this file only under the t[...]	AOLserver	1
The Original Code is AOLserver Code and related documentation distribu[...]	AOLserver	1
The Initial Developer of the Original Code is America Online, Inc. Por[...]	AOLserver	1
The contents of this file are subject to the AOLserver Public License [...]	AOLserver	1
Alternatively, the contents of this file may be used under the terms o[...]	AOLserver	1
Software distributed under the License is distributed on an "AS IS" ba[...]	AOLserver	5
You may obtain a copy of the License at http://aolserver.com/ .	AOLserver	1
This software/database is a "United States Government Work" under the [...]	PublicDomain	1
PUBLIC DOMAIN NOTICE National Center for Biotechnology Information	PublicDomain	1
The National Library of Medicine and the U.S. Government have not plac[...]	PublicDomain	1
The NLM and the U.S. Government disclaim all warranties, express or im[...]	PublicDomain	1
2 - This Source Code Form is subject to the terms of the Mozilla Publi[...]	MPLv2.0	1
If a copy of the MPL was not distributed with this file, You can obtai[...]	MPLv2.0	1
This program is free software; you can redistribute it and/or modify i[...]	GPLv2+	1
This program is distributed in the hope that it will be useful, but WI[...]	GPLv2+	27

実験結果1の考察

- 大量のライセンス記述がフィルタリングされてしまったため、記述パターンを抽出できる記述数が減少している

外れ値除去	クラスタ数	単一のライセンスから成るクラスタ	複数のライセンスから成るクラスタ	単一ライセンスから成るクラスタのライセンス数
データセット	246	178(72%)	69(28%)	45
外れ値除去後	127	126(99%)	1(1%)	48

- しかし、記述パターンを抽出できるライセンス記述数は減少しているが、ライセンス数は減少していない
- 提案手法は、適切なライセンスルールをより多くの種類のライセンスから作成することができる

実験結果2の考察

- 記述パターンのフィルタリングを行っても、全ての記述パターンが1種類のライセンスで成立するものではなかった
 - そもそも複数のライセンスに共通するライセンス記述が存在する BSD 2 とBSD3, 免責条項など
 - 複数のライセンスで共通するライセンス記述を事前に検出し、一つにまとめておく必要がある
- 記述パターンを正規表現化することで、そのままライセンスルールとして用いるものも抽出できた (MPLv2.0など)
 - NinkaはどういうわけかMPLv2.0を特定することができない[4]
 - 調査の結果、正規表現作成における人為的ミスが原因
 - 正規表現作成における人為的ミスを防ぐことができる

[4] Yuhao Wu, Yuki Manabe, Tetsuya Kanda, Daniel M. German, and Katsuro Inoue. A method to detect license inconsistencies in large scale open source projects. In Proc. of MSR '15, pp. 324–333, May 2015.

まとめと今後の課題

- まとめ

- 未知ライセンスのクラスタから、記述パターンを抽出する手法を提案した
- Debian v7.8.0を対象としたケーススタディで分かったこと
 - 外れ値除去により、クラスタリング結果からほとんどのNon-SLCをなくすことができた
 - ライセンスルール作成に有用なライセンス記述パターンを抽出できた

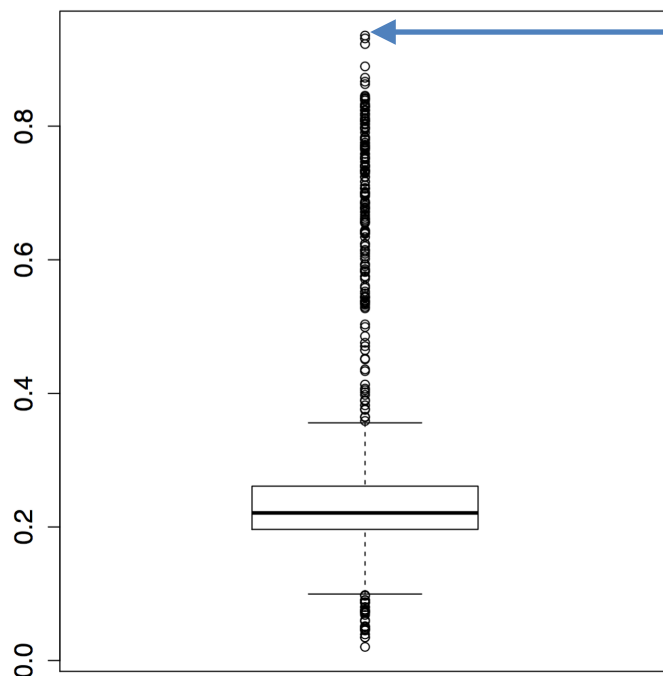
- 今後の課題

- 異なるライセンスに共通するライセンス記述の正規化したのちにBIDEを適用するように改善する
- ライセンスルール作成手順のSTEP3自動化 : ライセンス記述パターンを使って表記ゆれを列挙し、正規表現を作成する

編集距離の閾値決定

- 予備調査

- Debian v7.8.0から、ランダムに取り出した、異なるライセンス記述のペアにおける類似度を1000組調査



→ 最大値は0.9356

閾値は0.94を採用

異なるライセンスのペアでも、類似度が93%もある例が見つかった

閾値が高ければ高いほど、
処理後のファイル数が少ないが信頼性が高い