

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 10/26 研究発表

東 裕之輔

OSSの再利用

- 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

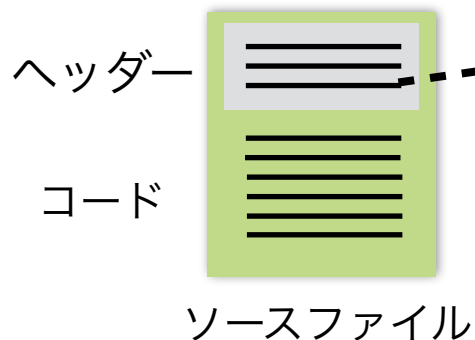


- OSSは、ソースファイルごとに決定されている
OSSライセンスを遵守することで再利用することができる

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・OSSライセンスが含まれている



Linux

ドメイン : OS

ソースファイル数 : 22,310

ライセンス種類数 : 53種類以上



ドメイン : DB

ソースファイル数 : 1,452

ライセンス種類数 : 18種類以上



ドメイン : Server

ソースファイル数 : 291

ライセンス種類数 : 4種類以上

- 多くのOSSは、他のOSSを再利用しているため、複数のOSSライセンスが含まれている

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

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.

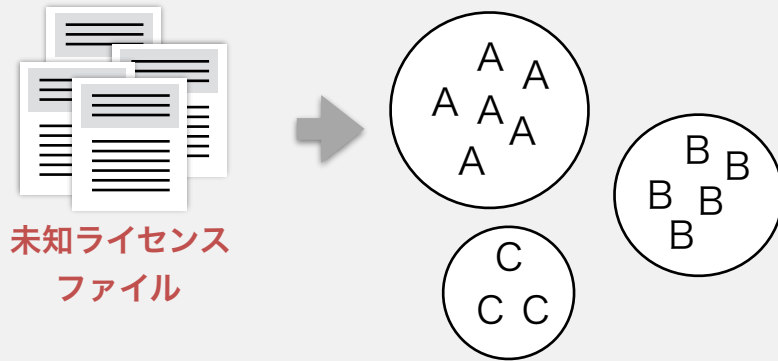
ライセンス特定ツールの課題

- 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

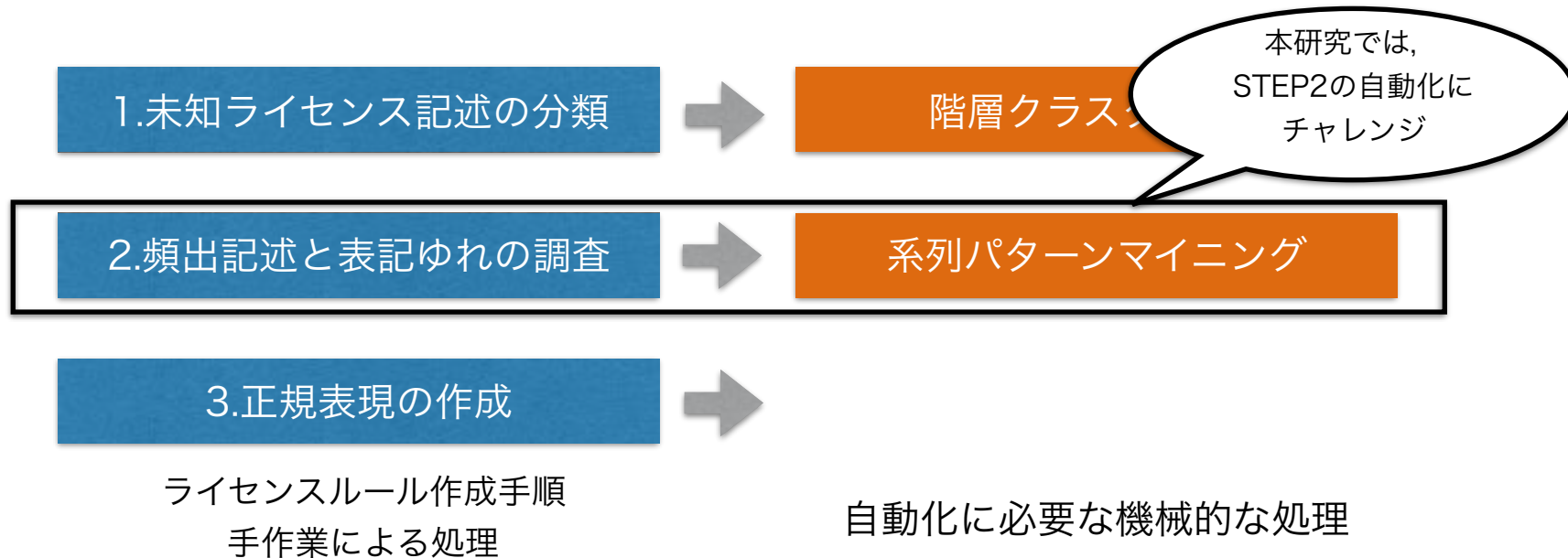
ライセンスルール作成手順
手作業による処理

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

- **本研究の目的**：既存手法によって生成されたクラスタから**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**：特定のライセンスにおけるライセンス記述に頻出する語句の出現順序（系列）

クラスタAの
未知ライセンス記述

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



本研究の目的
(Step2の自動化)

クラスタAから抽出された
ライセンス記述パターン

(This),(is),(free),(software.),(is),(free)
(It),(under),(the),(term),(of),(the),(A),(license)



Step.3の自動化

ライセンスルール生成

This (**program|software**) is free software.
It under the term of the A license

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、どうしても複数のライセンスから成るクラスタを作成してしまう
 - データセットには、ライセンス名は異なるが、ライセンス記述が酷似しているものが多数含まれている

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有用な系列パターンのみ抽出したい
 - 系列パターンマイニング：頻出する(出現記述数が2以上の)系列パターンを高速に抽出

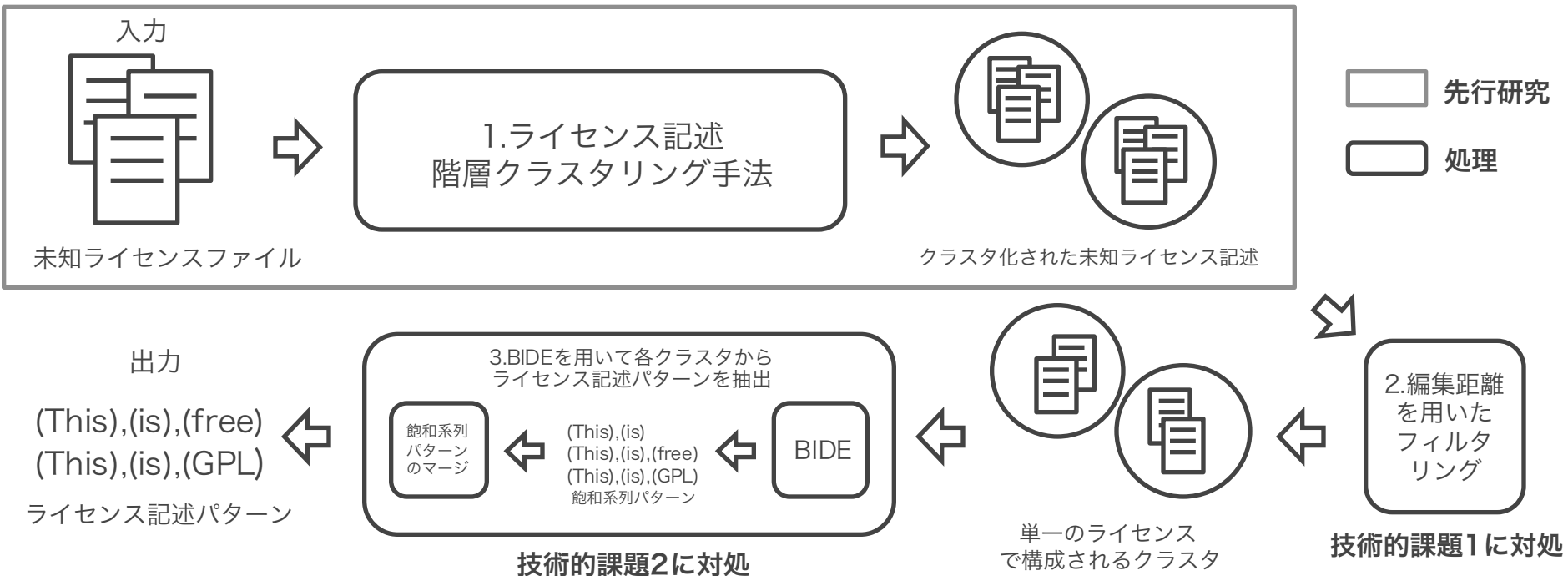
- 欠点：ルール作成に不要な大量の系列パターンを出力してしまう

例：LGPLのクラスタに頻出する系列パターン（一部）

```
-This program is distributed in the hope that it will be useful, but WITHOUT  
ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  
-This is or  
-This is the the  
-This is it the of  
-Free Software  
-See the GNU Lesser General Public License for more details.  
-by  
-of the GNU Lesser General Public  
-of  
-third-party contributions are distributed under license by Red Hat
```

- “This is the the”, “by”のような系列パターンは、どのライセンス記述でも成り立ってしまうため、ルール作成には**不適切**である

提案手法の概要



技術的課題1のアプローチ

- 編集距離を用いたフィルタリング
 - 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

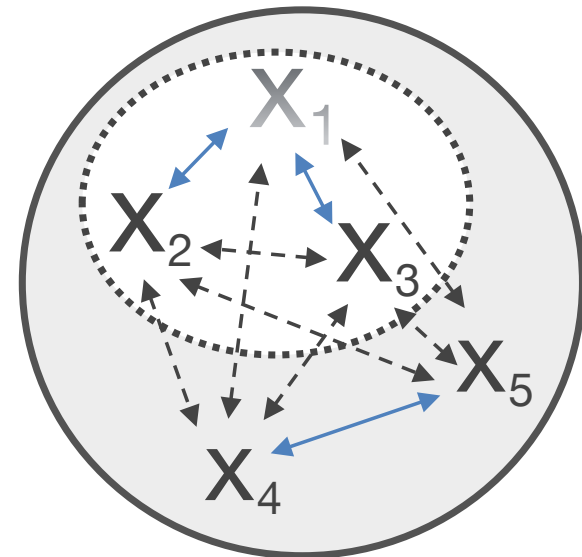
←-----→ 類似度が94%未満

←————→ 類似度が94%以上

X 未知ライセンス記述

X クラスタの中心となる記述
(←————→を最も多く持つ記述)

○ フィルタリングされる記述群



クラスタX

技術的課題2のアプローチ(1/2)

– 系列パターンマイニング

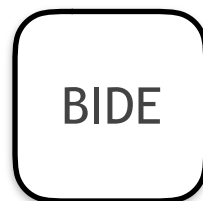
- アイテム列に頻出する系列パターンを抽出するマイニング手法

– BIDEアルゴリズム

- 系列パターンマイニングアルゴリズムの1つ
- 出現記述数が同じで、かつ、他の系列パターンの部分集合となる系列パターンを抽出しない

アイテム列集合

ABCD
ABC
BCDEFG
BCDEXG



出現記述数
2以上



BIDEによって抽出される系列パターン

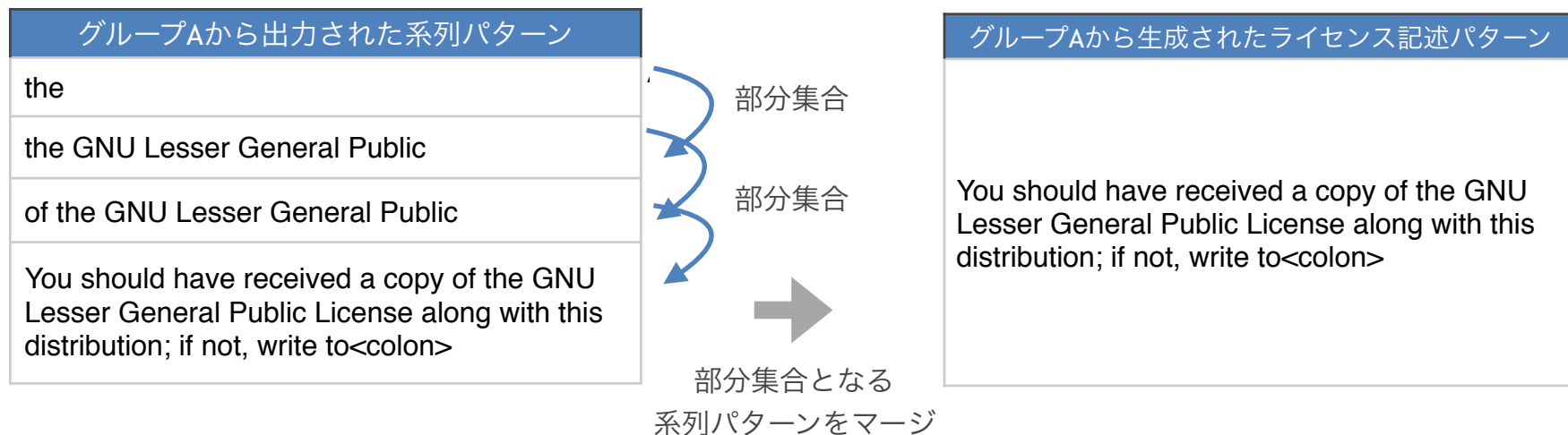
系列	出現記述数
BC	4
BCD	3
ABC	2
BCDEG	2

~~AB~~	~~2~~
~~BCDE~~	~~2~~

抽出しない

技術的課題2のアプローチ(2/2)

- 他の系列の部分集合になるような系列パターンをマージする
 - BIDEは、出現記述数が異なる系列間では、他の部分集合となる系列パターンを出力してしまう



- 不要なパターンをマージしたものをライセンス記述パターンとする

ケーススタディ

- 実験の目的：提案手法は、2つの技術的課題をどの程度解決できるか2種類の実験を行い検証する
 - 評価1. データセットとなるクラスタにおける単一のライセンスから成るクラスタの割合
 - 評価2. 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いくつ作成できるか
- 既存手法によるクラスタリング結果
 - 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	クラスタにならなかったファイル
2,446ファイル	246	170(69.1%)	76(30.9%)	48

実験結果1

- 実験：既存手法によるクラスタリング結果に対し、編集距離を用いたフィルタリングを行う
- 評価：単一のライセンスから成るクラスタの割合
- 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	クラスタにならなかったファイル
2,446ファイル	外れ値除去前	246	170(69.1%)	69(30.9%)	48
	外れ値除去後	126	120(95%)	6(5%)	2008

- 全体の95%のクラスタを単一のライセンスから成るクラスタにできた

実験結果2

- 実験：
 - 正規表現化したライセンス記述パターンを条件にgrepによるマッチングをデータセットに対し行う
 - 1種類のライセンスにマッチしたライセンス記述をライセンスルール作成に有用なライセンスルールとし、その数をカウントする
- 評価：ライセンスルール作成に有用なライセンスルールを何個作成することができたか
- 結果：ライセンスルール作成に有用なパターンを276個作成することができた

	生成されたパターン数	1種類のライセンスに マッチしたパターン数	割合
BIDEが出力したパターン	1498	375	25%
ライセンス記述パターン	371	276	74%

抽出したライセンス記述パターン例

ライセンス記述パターン	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 distributed in the hope that it will be useful, but WITH[...]	LGPL	32
You should have received a copy of the GNU Lesser General Public Lic[...]	LGPL	7

系列を示す括弧を外して表示. #Lic はライセンス記述パターンが対応するライセンス種類数を示す.

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、ライセンス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
 - 編集距離を用いてフィルタリングすることにより、クラスタリング結果から単一のライセンスから成るクラスタの割合を増加させることができた
 - 系列パターンマイニングを用いて、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

- フィルタリングされたファイルが多い
 - 課題1において、1つのクラスタから複数のライセンス記述グループを取り出せるようにする
- 複数のライセンスにマッチしたライセンス記述パターンの活用
 - 同じクラスタから抽出されたその他のライセンス記述パターンと同時に用いることで、活用できないか検証する

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

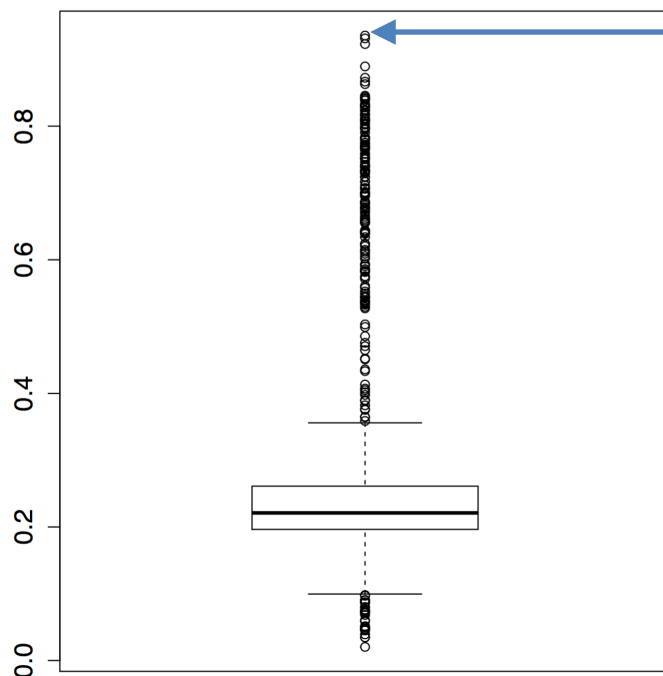
$$NLD = 1 - \frac{LD(A, B)}{\text{maxlength}(A, B)}$$

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

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