

Clustering OSS License Statements Toward Automatic Generation of License Rules

Yunosuke Higashi

Wakayama Univ.

Japan

Manabe Yuki

Kumamoto Univ.

Japan

Masao Ohira

Wakayama Univ.

Japan

Background

- Utilizing Open Source Software (OSS) is a means of reducing production costs in the modern software development
- OSS is reusable as part of a proprietary software product if it strictly complies with OSS licenses described in source files
- Identifying OSS licenses manually is a time-consuming task
- In order to help identify OSS licenses, license identification tools have been proposed

License Statements of Source Files

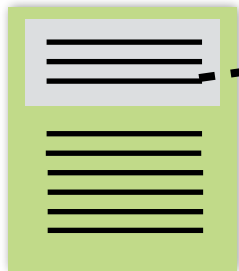
- An OSS license is declared in a header part of each source file as license statements



License Statements of GPLv3

header

code



source file

```
#####  
# This program is free software; you can redistribute it and/or modify  
# it under the terms of the GNU General Public License as published by  
# the Free Software Foundation; either version 3 of the License, or  
# (at your option) any later version.  
#  
# 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. See the  
# GNU General Public License for more details.  
#  
# #####
```

License Identification Tools

- License Identification Tools[1][2] automatically identify licenses of each source files

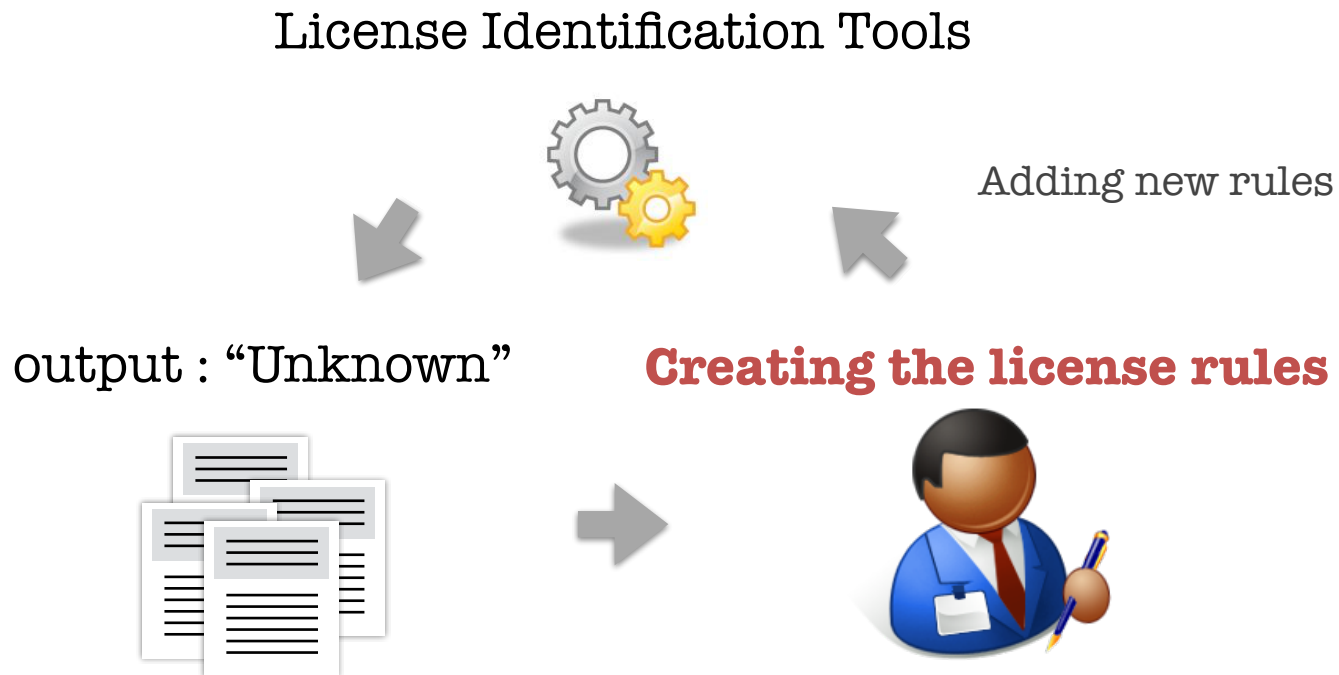


[1] **Ninka** : Daniel M German, Yuki Manabe, and Katsuro Inoue, "A Sentence-Matching Method for Automatic License Identification of Source Code Files" In Proceedings of ASE, pages 437-446, September 2010

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

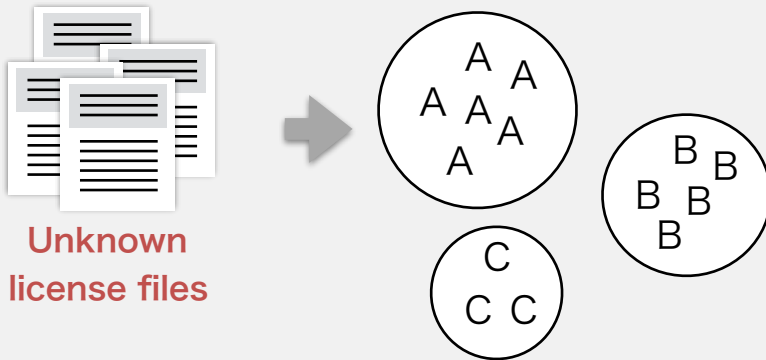
The Weakness of License Identification Tools

- New licenses and/or notation variants appear every year.
- License identification tools will require to manually and constantly add new regular expressions.



The Process of Creating License Rules

1. Grouping source files with unknown licenses

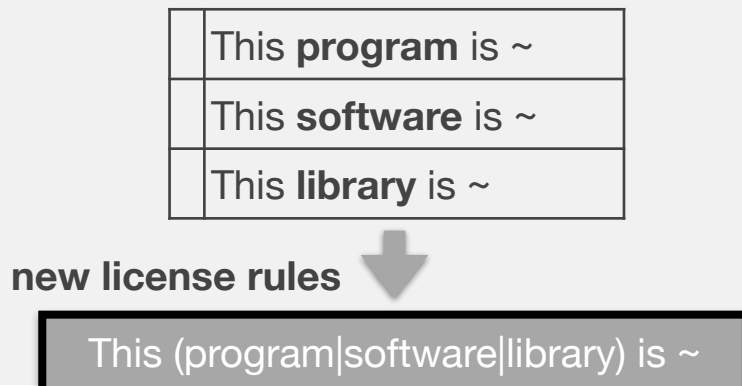


2. Checking notation variants for a single license

Notation variants for License Group A

This program is~ it under the term of the A license
This software is~ it under the term of the A license
This library is ~ it under the term of the A license

3. Creating license rules

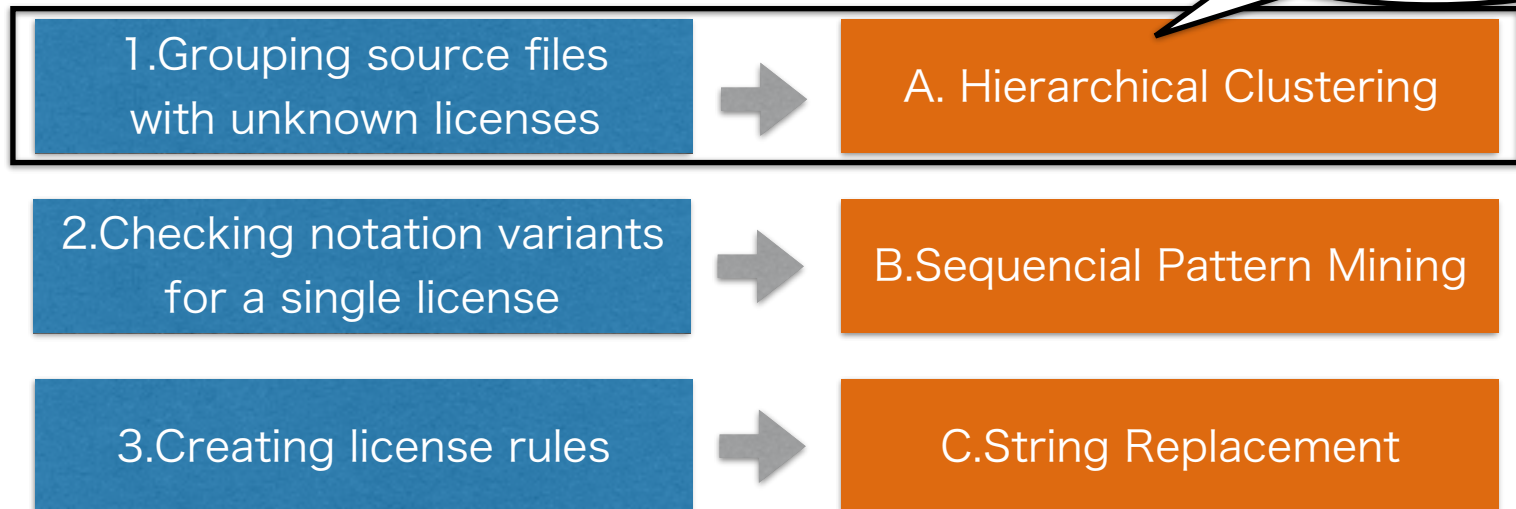


Current Problems

These tasks are time-consuming especially when the license identification tools report “**unknown**” for a large number of source files.

Our Final Goal

- Construct a method to automatically generate candidates of license rules
 - A method which consists of the following **steps**



The process of creating
license rules manually

The process of creating
license rules automatically

Two Technical Challenges

- Automating **Step 1** requires to address at least two technical challenges as follows

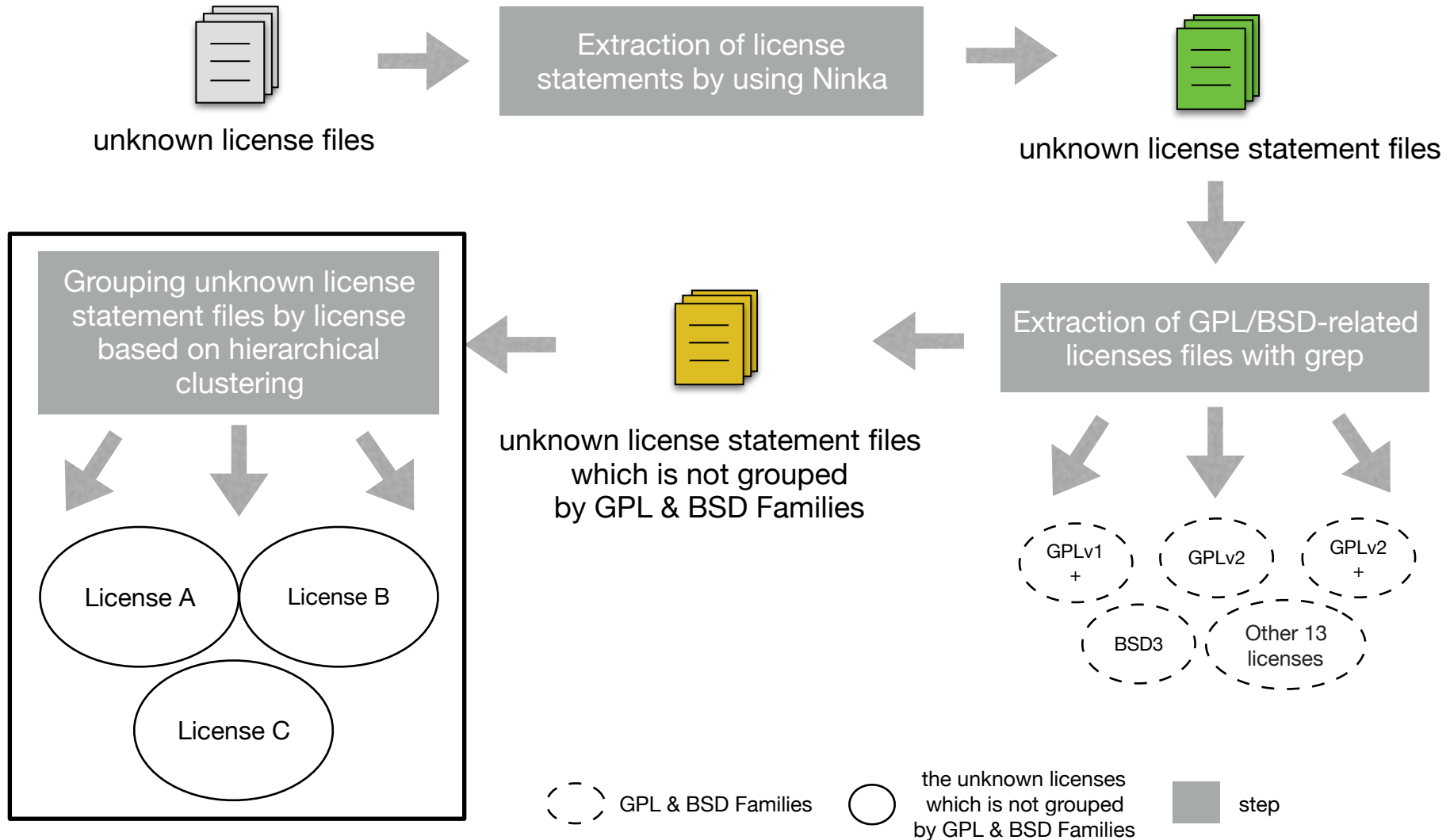
A. Discriminating GPL/BSD-related licenses

- In our pilot study, we found that Ninka judged many source files as “unknown” licenses even for very popular licenses such as GPLv2 and BSD2.
- we try to correctly discriminate GPL/BSD-related licenses

B. Creating groups of source files with unknown licenses

- The number of groups should correspond to the number of licenses existed in software components being reused

An Overview of the Proposed Method

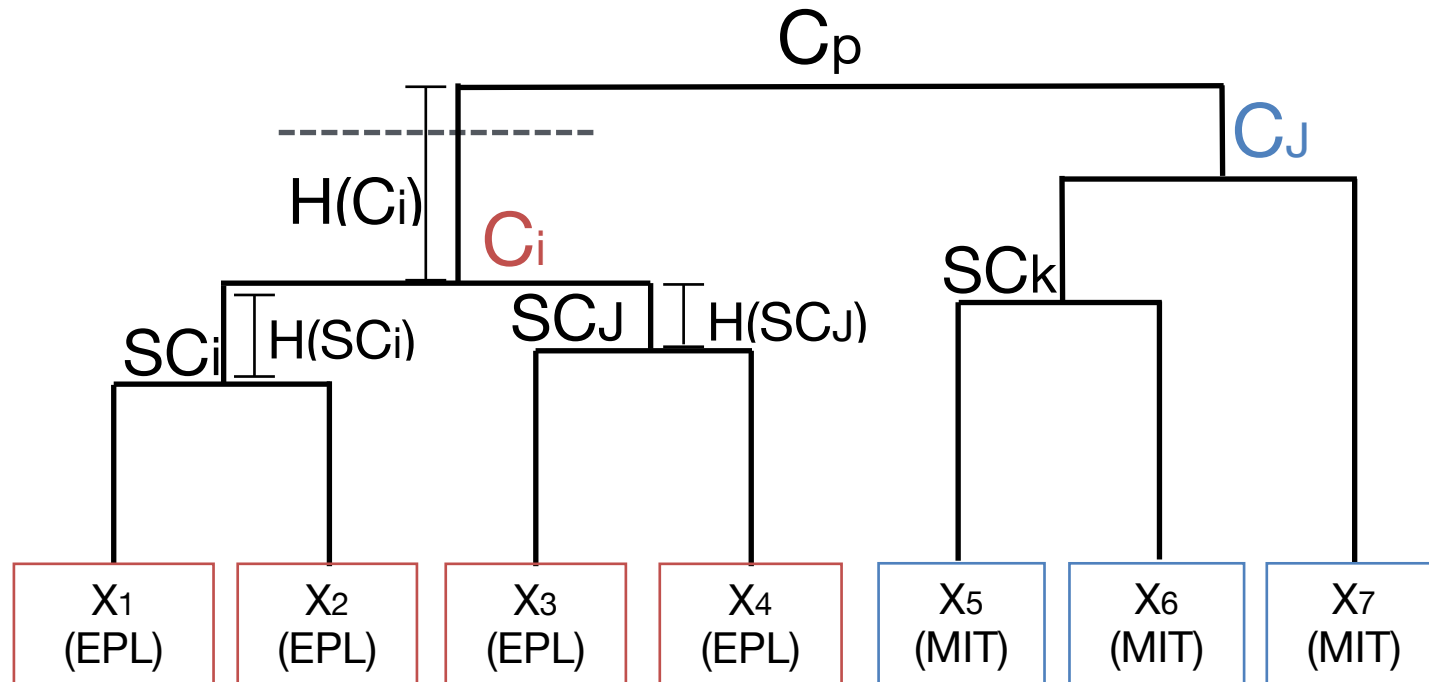


Grouping Unknown Licenses with Hierarchical Clustering



1. The unknown license statements files are converted into Bag-of-Words vectors.
2. A dendrogram is created by Ward's method from the unknown license statement files.
3. Cutting the tree (dendrogram) at a higher place would yield small clusters.
 - In this study, the same type of license statements should become a **single cluster**.
 - We introduce condition α & β for cutting the tree in a dendrogram **multiple times** at higher places.

Condition α



Condition α

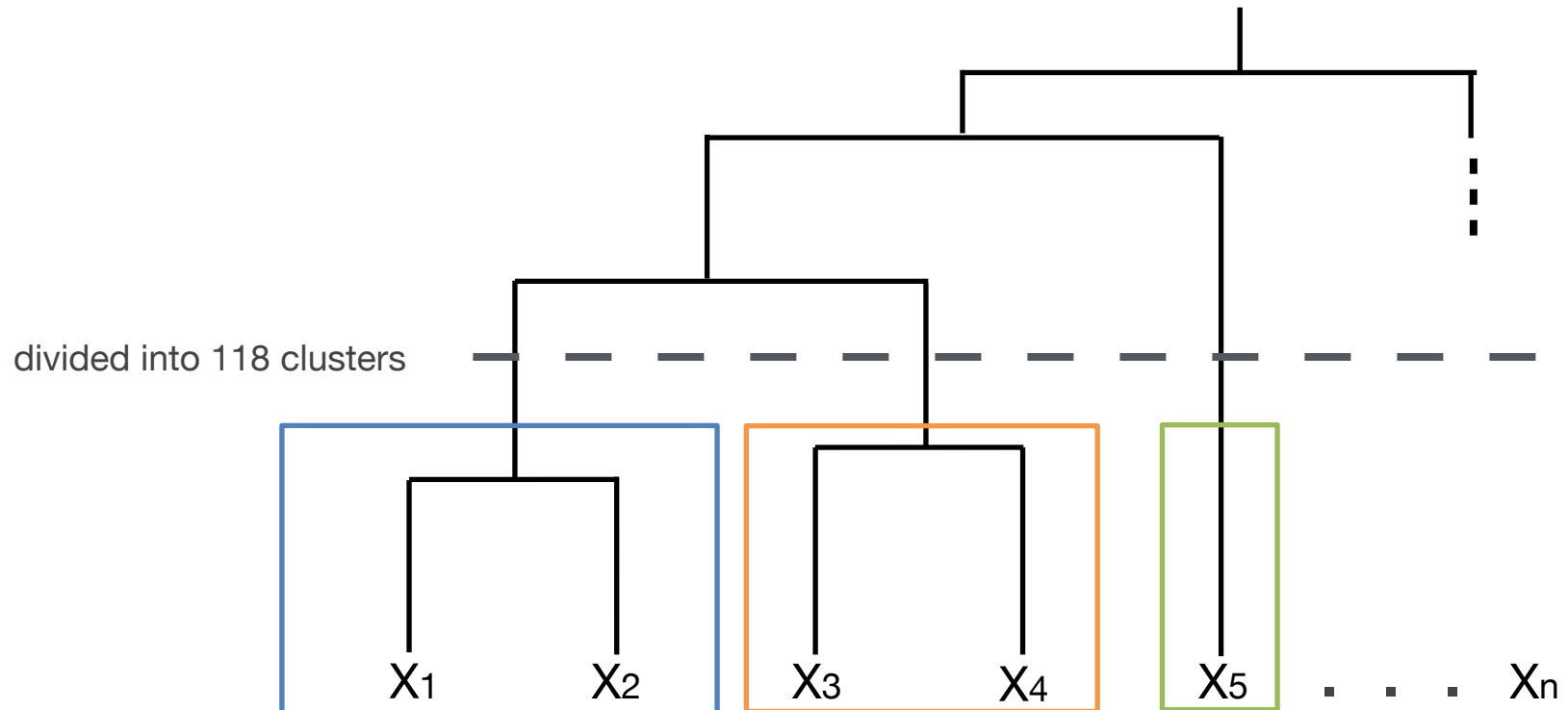
$$(H(C_i) > H(SC_i)) \text{ and } (H(C_i) > H(SC_j))$$

Case Study

- Purpose
 - We conduct a case study to investigate how well our clustering method can create clusters which could be candidates for generating license rules.
- Target files
 - **2,453 unknown license stmt. files of Debian v7.8.0**
 - Sampling 12,375 source files from each packages of Debian 7.8.0
 - Identifying 2,838 unknown license files from 12,375 source files with Ninka
 - Removing GPL/BSD related license files from 2838 files with grep
 - **118 license found in 2,453 files by reviewing manually**

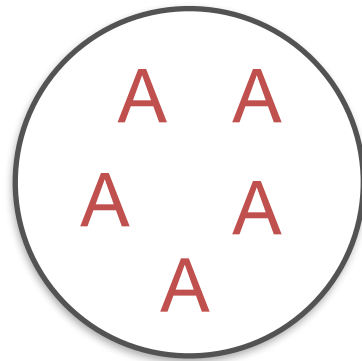
Baseline

- We prepared a baseline to evaluate clustering results using the condition α and β .
 - cut a tree as we can get 118 clusters

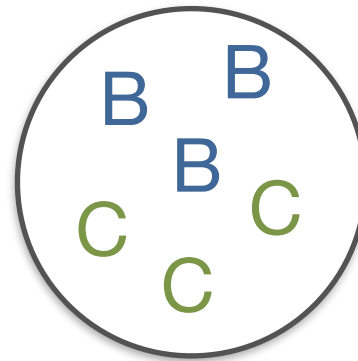


SLC (Single License Cluster)

- Clusters with a single license.
- **Creating more SLC is desired in this study**
 - We can extract license rules from SLC easily



SLC

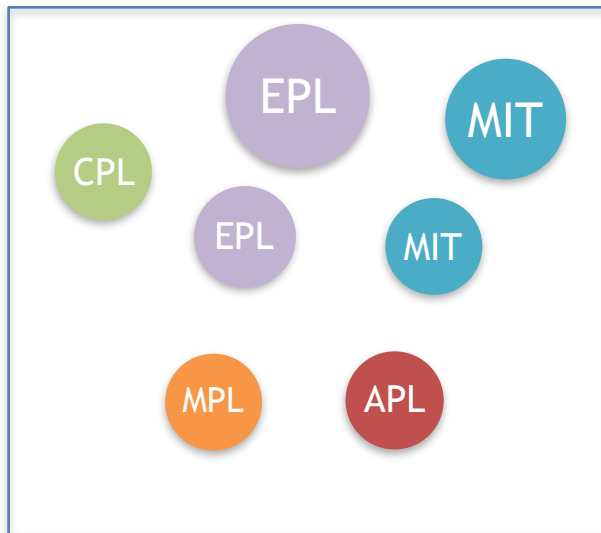


Non-SLC

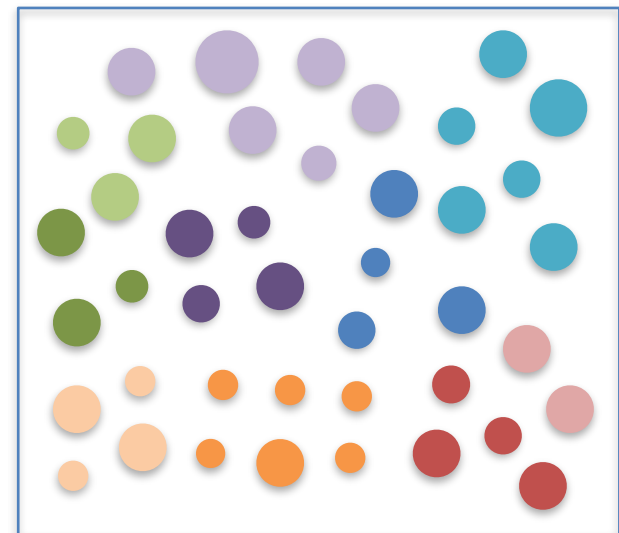
Clustering Conditions & Evaluation Metrics

	#C	#SLC	#Licenses in SLCs
cond. α	247	153	35
cond. β	719	409	42
baseline	81	31	17

baseline



condition α & β



Circle : SLC **Color** : License **Diameter** : #files in SLC

Conclusion and Future Work

- We proposed the method to classify unknown license files
 - Our method created clusters which are suitable as candidates for generating license rules automatically.
- Future Work
 - Merging SLCs with the same license
 - Creating a SLC from a Non-SLC

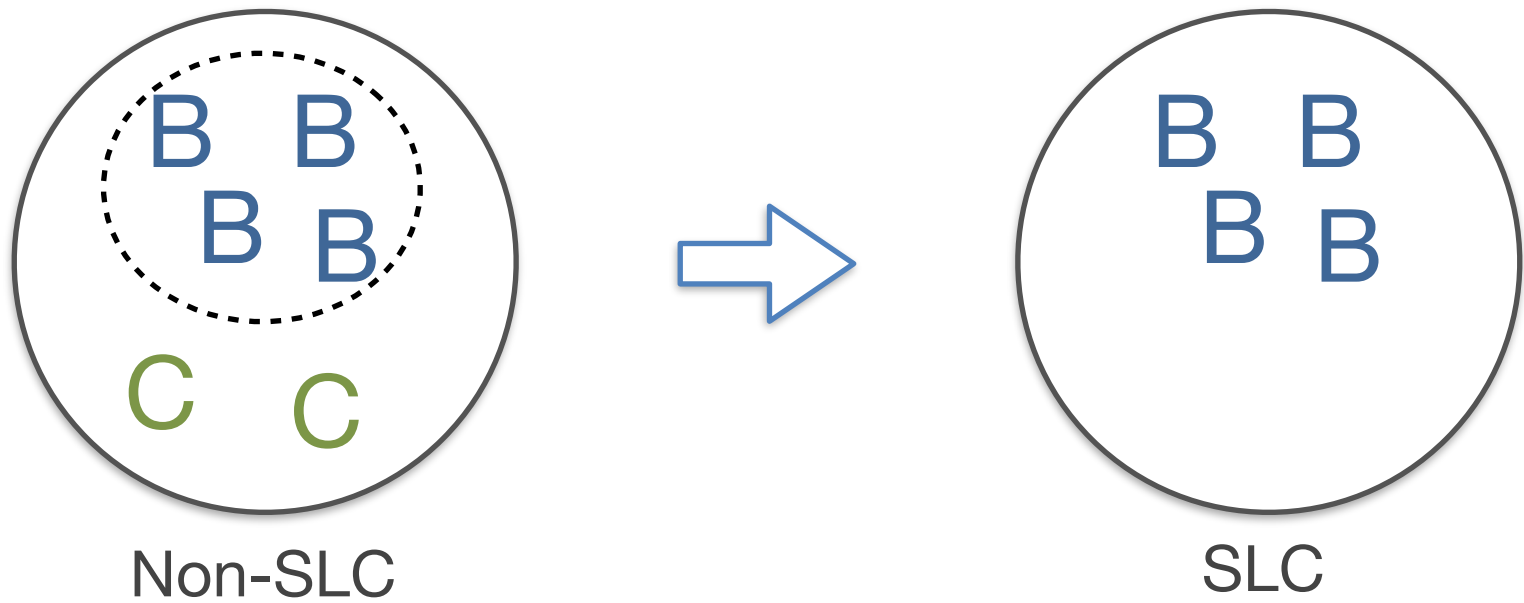
End



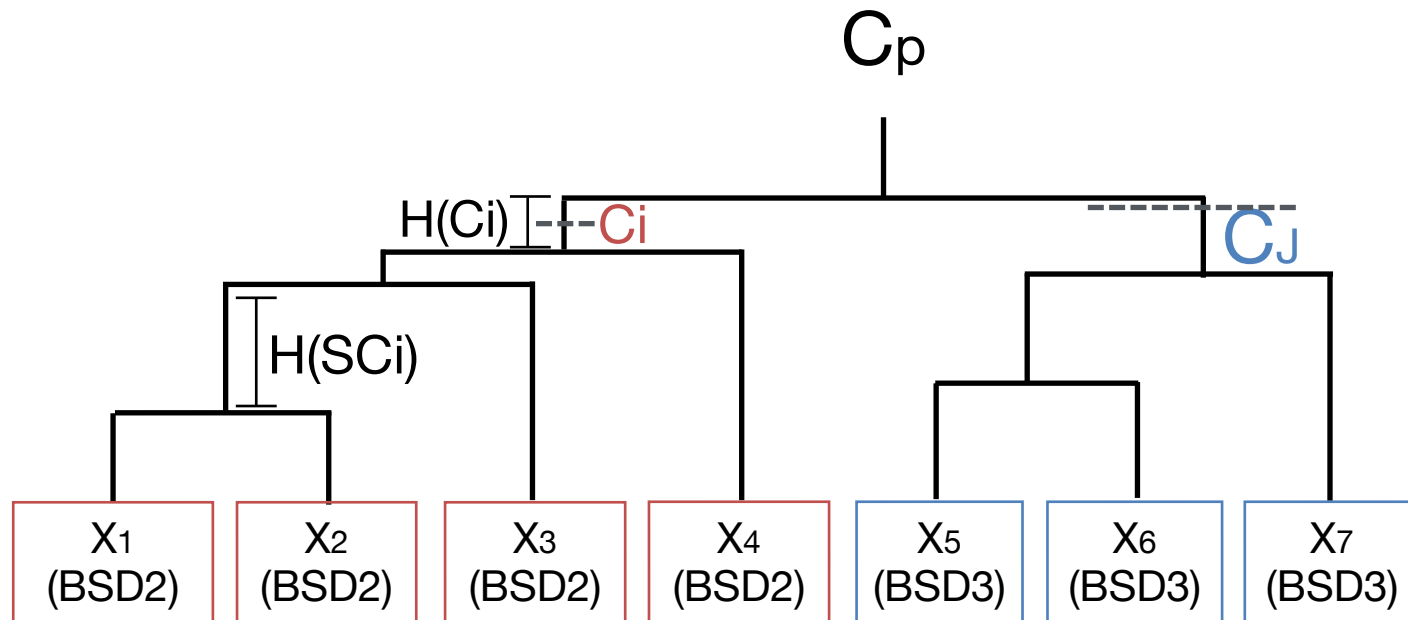


Creating a SLC from a Non-SLC

- We plan to extract the core in cluster by using the string similarity in each clusters.
 - the core in cluster : a collection of statements which similarity is high to each other in clusters.



Condition β



- Condition β
 - An extension of the condition α that the cluster is not created by two clusters.

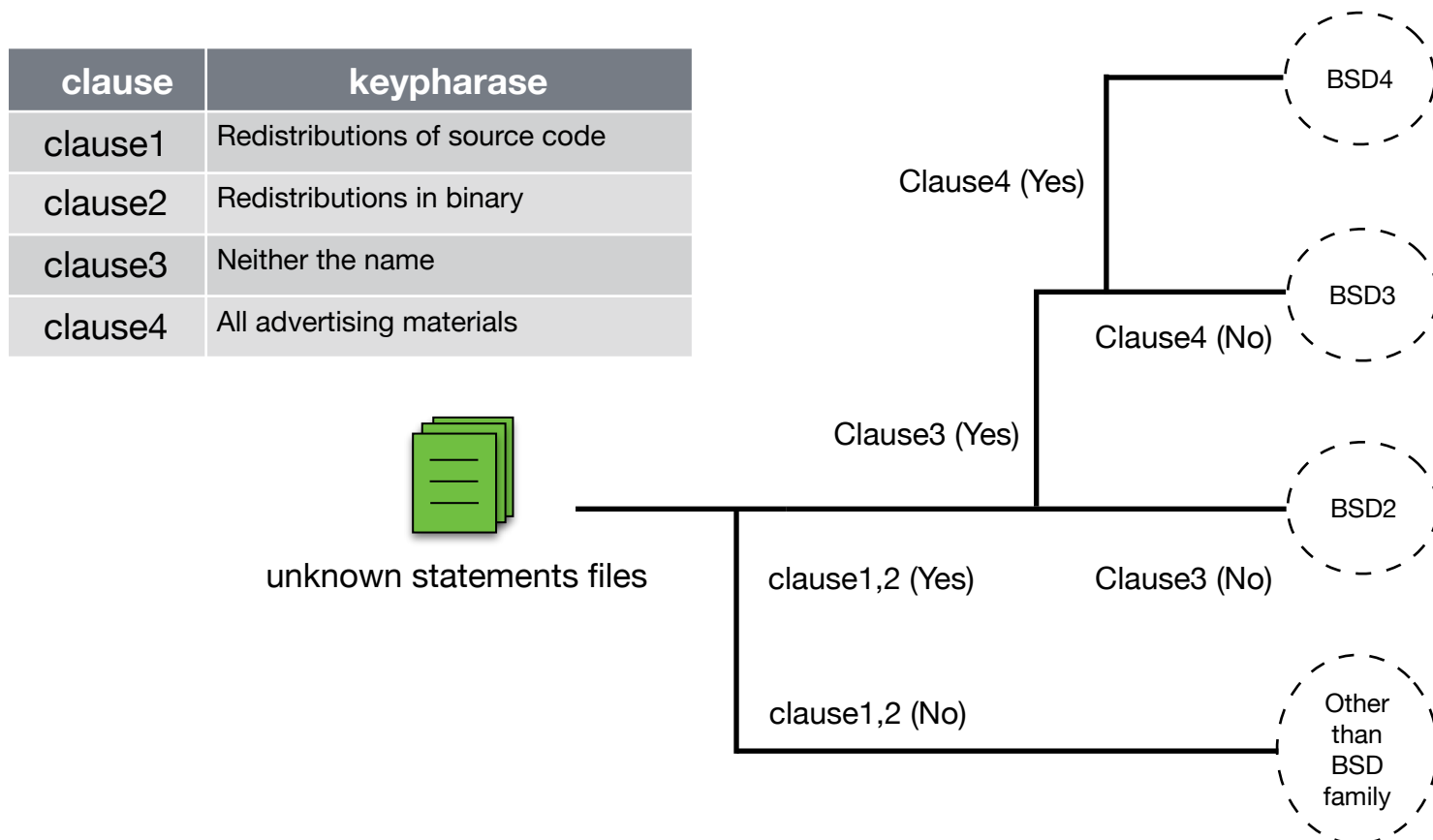
GPL/BSD-Related Lic. Extraction with grep

- We try to correctly discriminate GPL/BSD related licenses with keyphrase
 - GPL Family Licenses

GPL license family	Keyphrase	version name
GPLv1	GNU General Public License	version 1
GPLv2	GNU General Public License	version 2
GPLv3	GNU General Public License	version 3
LibraryGPLv2	GNU Library General Public License	version 2
LGPLv2.1	GNU Lesser General Public License	version 2.1
LGPLv3	GNU Lesser General Public License	version 3
AGPLv3	GNU Affero General Public License	version 3

Extraction of BSD licenses with grep

- Discriminant conditions and flow for BSD license family



Step 1: Extraction of GPL/BSD

- Evaluation
 - We evaluate that grep can correctly find GPL/BSD related licenses from the dataset with precision

- Result

Table : Accuracy of Grep For GPL/BSD Licenses.

Licenses	TP	FP	Precision
GPLv2	170	4	0.98
GPLv3	45	5	0.90
LibraryGPLv2	19	1	0.95
LGPLv2.1	31	2	0.94
LGPLv3	21	3	0.88
AGPLv3	15	0	1.00
BSD2	17	27	0.39
BSD3	16	1	0.94
BSD4	2	0	1.00
all	342	43	0.89