

判定できないライセンス記述に おける既知ライセンス文抽出手法の 有効性の調査

松浦 遊也

背景

- ソフトウェア開発において、コスト削減のために、OSSは頻繁に再利用されている
- OSSを再利用するにはソースファイルに記述されているライセンスを遵守する必要がある
- ソースファイルは膨大に存在し、理解するのに時間がかかるので、コスト削減のためにライセンス判定ツールが存在する

ライセンス判定ツール(Ninka)



Ninkaのライセンス判定:例

ライセンス
記述

Redistribution and use～:

1. Redistributions of source code must ～.
2. Redistributions in binary form must reproduce～.

ライセンスルールと比較

ライセンス
ルール

ライセンスルール:BSDv2

Redistribution and use～:(一致)

1. Redistributions of source code must ～.(一致)
2. Redistributions in binary form must reproduce～.(一致)

対応するライセンスルールが存在した

“BSDv2”

Ninkaのライセンス判定:例

ライセンス
記述

Redistribution and use~:

1. May ~
2. Redistributions in binary form must reproduce~.

ライセンスルールと比較

ライセンスルール:BSDv2

Redistribution and use~:(一致)

1. Redistributions of source code must ~.(不一致)
2. Redistributions in binary form must reproduce~.(一致)

ライセンス
ルール

対応するライセンスルールが存在しない

“Unknown”

Ninkaのライセンス判定:例

ライセンス
記述

Use and Redistiribution～:

1. Redistributions of source code must ～.
2. Redistributions in binary form must reproduce～.

ライセンスルールと比較

ライセンスルール:BSDv2

Redistribution and use～:(不一致)

1. Redistributions of source code must ～.(一致)
2. Redistributions in binary form must reproduce～.(一致)

ライセンス
ルール

対応するライセンスルールが存在しない

“Unknown”

Unknownとは

- Ninkaがライセンスルールを用いてライセンス判定を行った際に、判定ができなかったライセンス

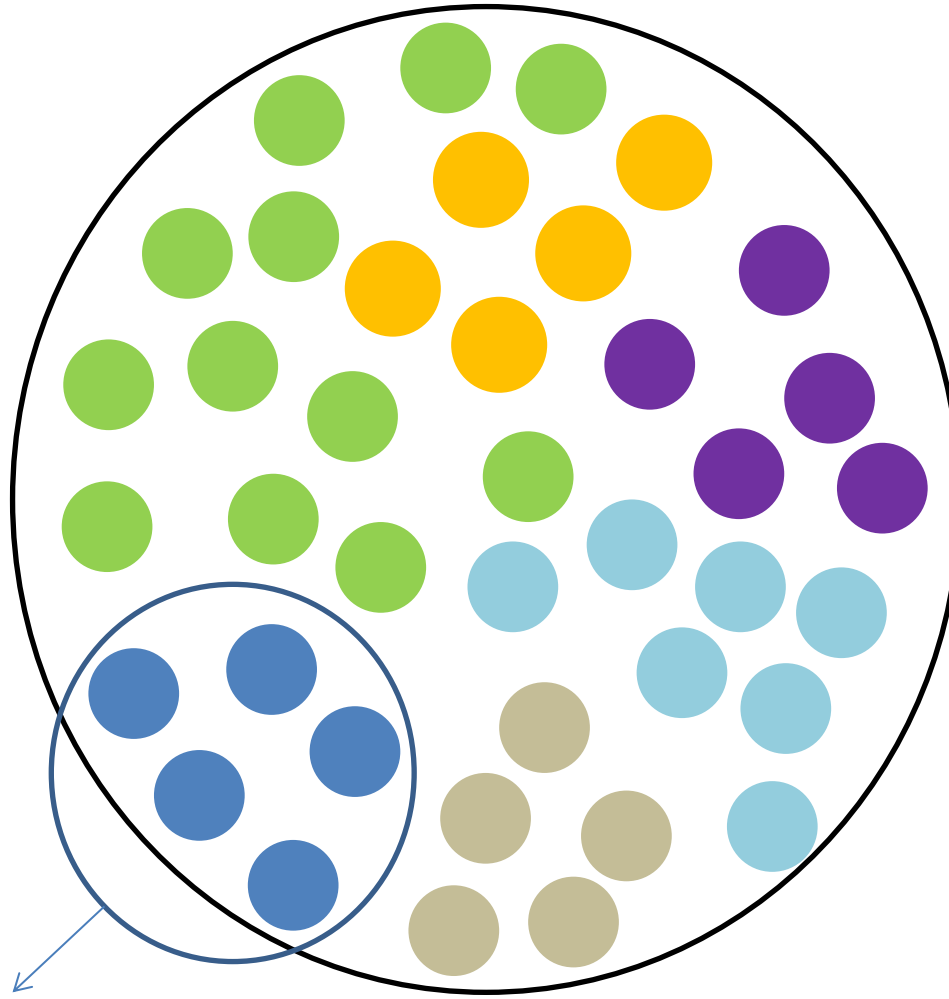
関連研究

- Unknownと判定されたライセンスから、単一のライセンスとして抽出するためのクラスタリング手法[1]
 - 目的: ライセンスルールを作成するために単一として特定できるライセンスをクラスタリングしている

未知ライセンスをルール化して、Ninkaにフィードバックするための研究

[1]Yunosuke Higashi and Yuki Manabe and Masao Ohira **Clustering OSS License Statements toward Automatic Generation of License Rules**, Proceedings - 7th International Workshop on Empirical Software Engineering in Practice, IWESEP 2016,pp30-35

関連研究の図



一つのクラスタ
として抽出

Unknown集合

関連研究の対象

- 関連研究の対象

- Debianなどは非常にファイル数が多いので、Unknownも非常に多く、ライセンスルールを作成して、Unknownファイルを減らす方がよい

- 関連研究の対象外

- ファイル数が多いOSSは、Unknownも少なく、目視でライセンスを判定可能

研究目的

- 目視で判定できる規模のUnknownライセンスを判定しやすくすること

提案手法

- 未知のライセンス記述から、既知のライセンス文を抽出する

Ninkaのライセンス判定:例

ライセンス
記述

Redistribution and use~:

1. May ~
2. Redistributions in binary form must reproduce~.

ライセンスルールと比較

ライセンスルール:BSDv2

Redistribution and use~:(一致)

1. Redistributions of source code must ~.(不一致)
2. Redistributions in binary form must reproduce~.(一致)

ライセンス
ルール

対応するライセンスルールなし

“Unknown”

Ninkaのライセンス判定:例

ライセンス
記述

Use and Redistiribution～:

1. Redistributions of source code must ～.
2. Redistributions in binary form must reproduce～.

ライセンスルールと比較

ライセンスルール:BSDv2

Redistribution and use～:(不一致)

1. Redistributions of source code must ～.(一致)
2. Redistributions in binary form must reproduce～.(一致)

ライセンス
ルール

対応するライセンスルールが存在しない

“Unknown”

現在行ったこと

- Debianからランダムに2838のソースファイルからライセンス記述を抽出、行ごとに分割した（合計行数9487行）
- 行毎に分割したデータセットから完全に未知のライセンスと判定されたライセンス記述を抽出（合計行数5164行）

これから行うこと

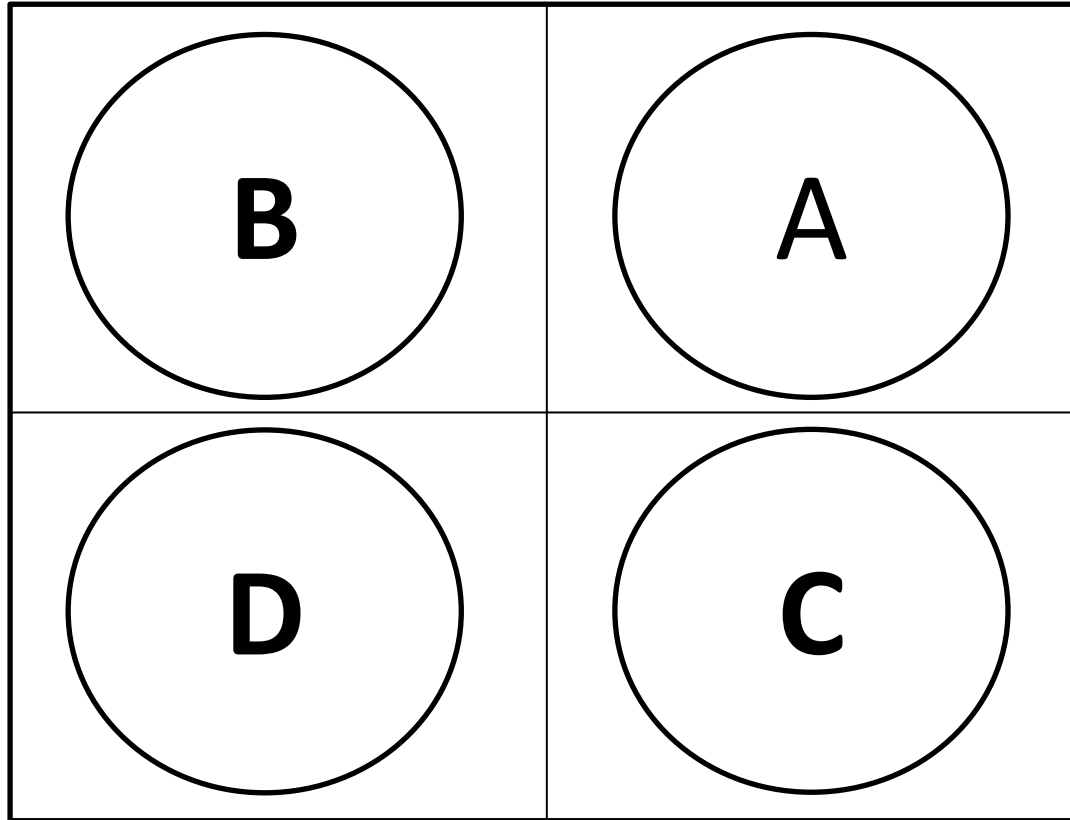
- 行数毎に抽出した未知のライセンス記述をタイプ毎に分割していく
- そのためにRを使い、未知のライセンス記述とライセンスルールの類似度を算出する

タイプ

- タイプA
 - 意味的にも、文章的にも既知ライセンスと同じ
- タイプB
 - 文章的には、異なっているが意味的に既知ライセンスと同じ
- タイプC
 - 文章的には、同じだが意味的に既知ライセンスと異なっている
- タイプD
 - 文章的にも、意味的にも既知ライセンスと異なっている

ライセンス文のタイプ

意味的一致



文字列不一致

文字列一致

意味的不一致

類似度には何を用いるか

- 類似度はレーベンシュタイン距離を用いる
 - 理由: ライセンスルールを修正する際は既存のライセンスルールを編集して行うので、編集距離が短いものを類似度が高いとするレーベンシュタイン距離を用いる