Predicting Licenses in Changed Source Code

Xiaoyu Liu¹, LiGuo Huang¹, Jidong Ge², Vincent Ng³

¹ Dept. of Computer Science, Southern Methodist University, Dallas, TX
² State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
³ Human Language Technology Research Institute, University of Texas and Dallas, TX

Emails: {xiaoyul, lghuang}@smu.edu  gjd@nju.edu.cn  vince@hlt.utdallas.edu
After software changes are made ...
Penalties for Software License Violation

Whenever piracy of software is detected, the individual or organization responsible for paying for the software licenses with others, they can face severe consequences.

Just like illegally downloading music, the legal penalties may go beyond just a fine. Depending on the software involved, they may be required to pay fines as high as $250,000 and may face up to five years in prison. In addition, a permanent felony will be added to their record.

Predict license to resolve violation as soon as possible!
Challenges of Software Licensing

- Time and labor effort consuming

- Difficulty: requires lots of experience and expertise
Software License Prediction

Changed imports in XMLPacker.java
XMLPacker.java was licensed with MPL v1.1

LogEntry adopts GPL v3+

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or

(read MPL v1.1 restrictions)
Goal

*Automatically* predict software license for a *changed source code file*
Related Work on File-level License Prediction

• **Ninka:** [German et.al., 2010]
  
  • state-of-the-art, leveraging a set of pre-defined regular expressions built upon these common terms to detect the presence of the license copyrights and terms

• **Caller-Callee (CC):**
  
  • license based on code imports’ licenses

Not taking into account the license compatibility issues of the different license restrictions
Automatic License Predictor (ALP)

• Novel Contributions
  • Propose ALP, the first learning-based approach for license prediction on changed source code file
  • Conflict resolution leverage additional knowledge resources

• Framework

A Basic ALP System → Modeling the Previous License → Adding New Knowledge Sources → Modeling conflicts
Basic ALP System

"OpenEMRConnect is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY"

OpenEMRConnect is distributed; is-in; distributed-the; in-hope; ...

code-inline text features

Logistic Regression

diff features

changed source code file
Modeling the Previous License (ALP2)

Previous version

Current version
Adding New Knowledge Resources

• New resource added!

---

MIT License

Copyright (c) 2019 Avery O'Banion

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Adding New Knowledge Resources

• New resource added!
Adding New Knowledge Resources

- ALP2+Doc+Co

![Diagram](image)

Software documents

- Document-text features

- Code-inline text features

- Diff features

Changed source code file

- Code-inline text features

- Diff features

Co-changed source code files

Conditional Random Field

\[ y_1, y_2, y_3, \ldots, y_n \]

\[ x_1, x_2, x_3, \ldots, x_n \]
An Example of License Prediction Problem

Changed source code file

Software documents

co-changed source code files

License A

License B

License C
Modeling Conflicts (ALP2-Ranker)

• Training

changed source code file

software documents

code-inline text features
diff features
document-text features

Learning to Rank

Learns the most representative features from each knowledge source for the appropriate license adoption
Resolve Software License Conflicts (ALP2-Ranker)

1. Software document
2. Changed file
3. Co-changed file

Changed source code file → code-inline text features → diff features

Software documents → document-text features

Co-changed source code file → code-inline text features → diff features
Empirical Evaluation

- Datasets: Open Source Java Projects from GitHub
  - 700 Java projects from GitHub

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># of systems</td>
<td>700</td>
</tr>
<tr>
<td># of commits</td>
<td>8128</td>
</tr>
<tr>
<td># of changed files</td>
<td>57450</td>
</tr>
</tbody>
</table>

(a) Overall statistics
Procedure of Annotating Licenses for Each File

• We looked at the **terms and conditions for copying, distribution and modifications** of each license of each resource

• **Coders**: two PhD students who have extensive experience in industry as developers

• **Agreement ratio**: 73.7%

• **Disagreements are resolved by open discussion**
Results of Data Annotation

- 25 unique licenses

<table>
<thead>
<tr>
<th>Licenses</th>
<th># of changed files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache v2</td>
<td>18770 (32.7%)</td>
</tr>
<tr>
<td>GPL v2</td>
<td>9458 (16.5%)</td>
</tr>
<tr>
<td>GPL v3+</td>
<td>5943 (10.3%)</td>
</tr>
<tr>
<td>MIT</td>
<td>3125 (5.4%)</td>
</tr>
<tr>
<td>LGPL v3+</td>
<td>2609 (4.5%)</td>
</tr>
<tr>
<td>LGPL v2.1+</td>
<td>1542 (2.7%)</td>
</tr>
<tr>
<td>BSD</td>
<td>1404 (2.4%)</td>
</tr>
<tr>
<td>EPL v1</td>
<td>1276 (2.2%)</td>
</tr>
<tr>
<td>Other</td>
<td>4960 (8.6%)</td>
</tr>
<tr>
<td>Non-licensed</td>
<td>8363 (14.6%)</td>
</tr>
</tbody>
</table>

(b) Per-license frequencies
### Changed file

**(Apache v2)** … Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form…

### Software document

**(GPL v2)** … Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it …

### Co-changed file

**(LGPL v2.1+)** … linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the Library". The executable is therefore covered by this License... …
Empirical Evaluation

- **Datasets: Open Source Java Projects from GitHub**
  - 700 Java projects from GitHub, 25 unique licenses

- **Baseline Systems**
  - *Ninka*
  - *Caller-Callee (CC)*
  - *Previous Version (Prev):* predicts license to be the same as previous version
Empirical Evaluation

• **Datasets: Open Source Java Projects from GitHub**
  - 700 Java projects from GitHub, 25 unique licenses

• **Baseline Systems**
  - *Ninka*
  - *Caller-Callee (CC)*
  - *Previous Version (Prev):* predicts license to be the same as previous version

• **Metrics**
  - Macro F1: treat each license as equally important
  - Micro F1: treat most frequent licenses as more important
  - Five-Fold Cross Validation
Overall Performance

**RQ1.** Which license prediction system performs the best?

- **ALP2**: Basic ALP + Previous version modeling
- **ALP2-Ranker**: ALP2 + conflict resolver
- **ALP2+Doc+Co**: ALP2 with more knowledge resources

<table>
<thead>
<tr>
<th>System</th>
<th>Macro-F1</th>
<th>Micro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ninka</td>
<td>38.3</td>
<td>73.5</td>
</tr>
<tr>
<td>CC</td>
<td>17.3</td>
<td>39.6</td>
</tr>
<tr>
<td>Prev</td>
<td>30.6</td>
<td>66.3</td>
</tr>
<tr>
<td>Basic-ALP</td>
<td>38.9</td>
<td>82.2</td>
</tr>
<tr>
<td>ALP2</td>
<td>46.4</td>
<td>88.3</td>
</tr>
<tr>
<td>ALP2+Doc+Co</td>
<td>48.3</td>
<td>90.9</td>
</tr>
<tr>
<td>ALP2-Ranker</td>
<td>79.2</td>
<td>92.5</td>
</tr>
</tbody>
</table>
**RQ2.** How do the systems perform on the easy and difficult tasks?

![Graph showing performance metrics for Ninka-det (62%) and Ninka-undet (38%)]
**RQ2.** How do the systems perform on the easy, difficult, and conflict instances?

**Ninka-det**
- Macro-F1: 97.2, 88.9, 98.1, 98.7
- Micro-F1: 38.1, 41.7

**Ninka-undet**
- Macro-F1: 45
- Micro-F1: 9.6, 28.6

**Conflict**
- Macro-F1: 32.3, 15.6, 56.9, 86.1
- Micro-F1: 49.5, 37
Goal

**Automatically predict software license for a changed source code file**

---

**Automatic License Predictor (ALP)**

- **Novel Contributions**
  - Propose ALP, The first learning-based approach for license prediction on changed source code files
  - Conflict resolution leverage additional knowledge resources

- **Framework**

---

**Overall Performance**

**RQ1. Which license prediction system performs the best?**

ALP: Basic ALP; Prev: previous version modeling; ALP2+Doc+Co: ALP2 + implicit conflict resolver; ALP2-Ranker: ALP2

---

**Predicting Licenses in Changed Source Code**

Xiaoyu Liu¹, LiGuo Huang¹, Jidong Ge², Vincent Ng³

1 Dept. of Computer Science, Southern Methodist University, Dallas, TX
2 State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
3 Human Language Technology Research Institute, University of Texas and Dallas, TX

Emails: {xiaoyul, lguoau}@smu.edu, gid@nju.edu.cn, vince@hit.utdallas.edu