

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](mailto:xiaoyul@smu.edu), [lghuang](mailto:lghuang@smu.edu)}@smu.edu gjd@nju.edu.cn vince@hlt.utdallas.edu



11/14/2019



SMU



南京大學
NANJING UNIVERSITY

After software changes are made ...



Penalties for Software License Violation

Whenever piracy of software is responsible for part of a crime with others, they can be charged with piracy.

Just like illegally downloading software may go beyond just a civil offense. Depending on the circumstances, it can be a criminal offense.

The legal penalties for software piracy can be severe.

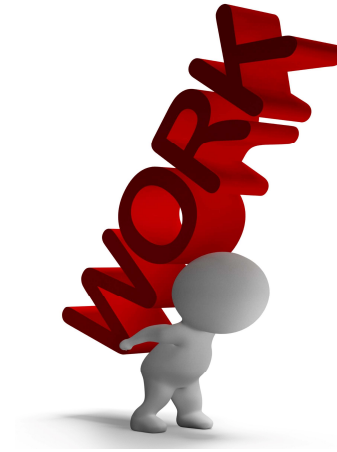
They may be required to pay fines as high as \$250,000 and may face up to five years in prison. In addition, a permanent felony will be on their record.



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

```
22 +import com.sun.org.apache.xalan.internal.xsltc.DOM;
23 +import ke.go.moh.oec.LogEntry;
24 +import java.io.ByteArrayInputStream;
25 +import java.io.IOException;
26 +import java.io.InputStream;
27 +import java.io.StringReader;
28 +import java.io.StringWriter;
29 +import java.text.ParseException;
30 +import java.text.SimpleDateFormat;
31 +import java.util.logging.Level;
32 +import java.util.logging.Logger;
33 +import javax.xml.parsers.DocumentBuilder;
34 +import javax.xml.parsers.DocumentBuilderFactory;
35 +import javax.xml.parsers.ParserConfigurationException;
36 +import javax.xml.transform.TransformerConfiguration;
37 +import javax.xml.transform.TransformerException;
```

LogEntry adopts
GPL v3+

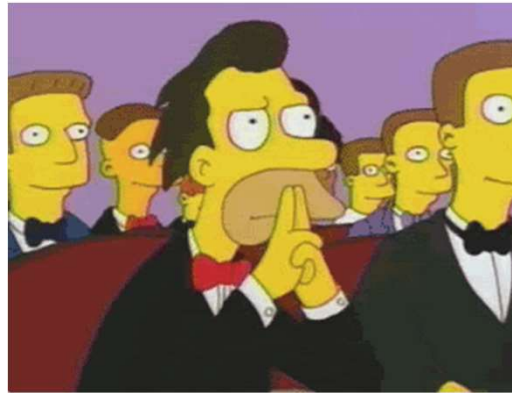
You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or

(read MPL v1.1 restrictions)

and other provisions required by the GPL or the LGPL. If you do not delete the provisions above, a recipient may use your version of this file under the terms of any one of the MPL, the GPL or the LGPL.

Goal

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

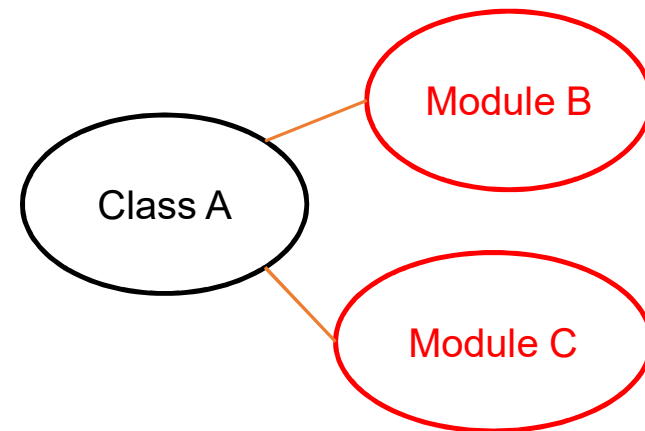


Related Work on File-level License Prediction

- **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

```
...
/\^[Reg]ular[Ex]pression$/
...

```



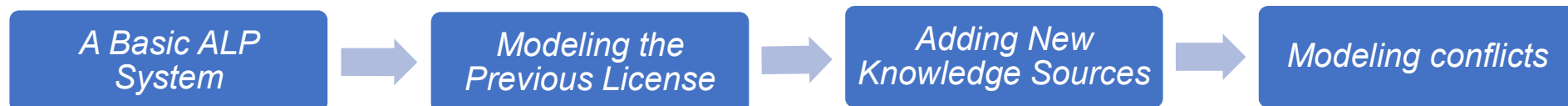
Not taking into account the license compatibility issues of the different license restrictions

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



Basic ALP System

```

43 43      * Gmail SMTP/IMAP port for mail sending/receiving.
44 44      */
45 45      private final int port;
46 46      //
47 47      * Cache for (Glink #toString())
48 48      */
49 49      private StringBuilder toString;
50 50
51 51      //
52 52      * Constructor with Gmail user, host and port
53 53      *
54 54      @Override
55 55      public int getPort() {
56 56          return port;
57 57      }
58 58
59 59      @Override
60 60      public String toString() {
61 61          if (toString != null) {
62 62              return toString.toString();
63 63          }
64 64          try {
65 65              toString = new StringBuilder();
66 66              toString.append("ConnectionInfo: (user:").append(getUser())
67 67                  .append("; host:").append(getHost())
68 68                  .append("; port:").append(getPort())
69 69          }
70 70          return toString.toString();
71 71      }
72 72
73 73      //
74 74      * Constructor with Gmail user, host and port
75 75      *
76 76      @Override
77 77      public int getPort() {
78 78          return port;
79 79      }
80 80
81 81      @Override
82 82      public String toString() {
83 83          if (toString != null) {
84 84              return toString.toString();
85 85          }
86 86          try {
87 87              toString = new StringBuilder();
88 88              toString.append("ConnectionInfo: (user:").append(getUser())
89 89                  .append("; host:").append(getHost())
90 90                  .append("; port:").append(getPort())
91 91          }
92 92          return toString.toString();
93 93      }

```

changed source code file

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

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

code-inline text features

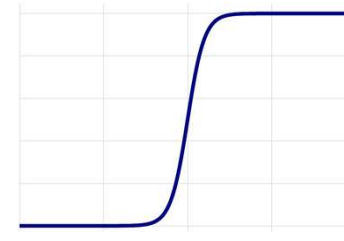
```

1 1  /*
2 2  * Copyright (c) 2008-2009 Tomas Varaneckas
3 3  * http://www.varaneckas.com
4 4  *
5 5  * Licensed under the Apache License, Version 2.0 (the "License");
6 6  * you may not use this file except in compliance with the License.
7 7  * You may obtain a copy of the License at
8 8  *
9 9  * http://www.apache.org/licenses/LICENSE-2.0
10 10  *
11 11  * Unless required by applicable law or agreed to in writing, software
12 12  * distributed under the License is distributed on an "AS IS" BASIS,
13 13  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 14  * See the License for the specific language governing permissions and
15 15  * limitations under the License.
16 16  */

```

*you-not;
may-use;
not-this;
use-file; ...*

diff features



Logistic Regression



Previous version

Diagram illustrating a Conditional Random Field (CRF) structure. The hidden states $y_1, y_2, y_3, \dots, y_n$ are connected sequentially by a horizontal line. Each hidden state y_i is connected to an observed state x_i by a vertical line. The observed states $x_1, x_2, x_3, \dots, x_n$ are represented by black circles, while the hidden states y_i are white circles. The diagram is flanked by blue triangles pointing towards the center.



Current version

Adding New Knowledge Resources

- New resource added!



Software documents

MIT License

Copyright (c) 2019 Avery O'Banion

LICENSE file

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!

Showing 6 changed files with 298 additions and 10 deletions.

+ src/main/java/com/googlecode/gmail4j/javamail/ConnectionInfo.java

+ src/main/java/com/googlecode/gmail4j/javamail/ImapConnectionHandler.java

src/main/java/com/googlecode/gmail4j/javamail/ImapGmailClient.java

src/main/java/com/googlecode/gmail4j/javamail/ImapGmailConnection.java

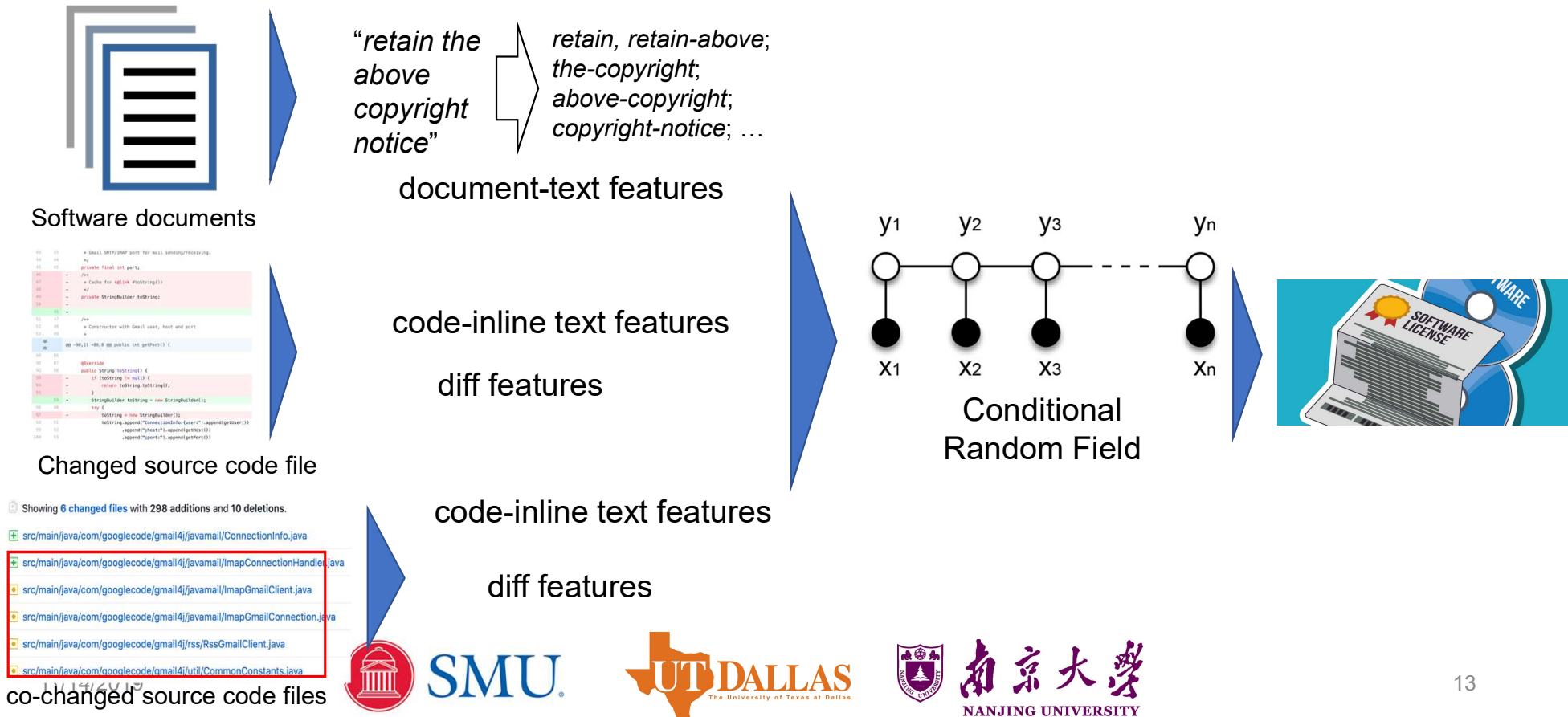
src/main/java/com/googlecode/gmail4j/rss/RssGmailClient.java

src/main/java/com/googlecode/gmail4j/util/CommonConstants.java

Co-changed files with
ConnectionInfo.java

Adding New Knowledge Resources

- ALP2+Doc+Co

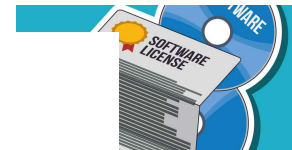


An Example of License Prediction Problem

Changed source
code file

```
126 126 @Override
127 127 public void send(final GmailMessage message) {
128 128 if (message instanceof JavaMailGmailMessage) {
129 129     Transport transport = null;
130 130     try {
131 131         final JavaMailGmailMessage msg = (JavaMailGmailMessage) message;
132 132         transport = getGmailTransport();
133 133         transport.sendMessage(
134 134             msg.getMessage(),
135 135             msg.getMessage().getAllRecipients());
136 136     } catch (Exception e) {
137 137         throw new RuntimeException(e);
138 138     }
139 139 }
```

Software documents



icense A



icense B

co-changed
source code files

Showing 6 changed files

- src/main/java/com/googlecode/gmail4j/javamail/imapGmailClient.java
- src/main/java/com/googlecode/gmail4j/javamail/imapGmailConnection.java
- src/main/java/com/googlecode/gmail4j/rssGmailClient.java
- src/main/java/com/googlecode/gmail4j/util/CommonConstants.java



License C

Modeling Conflicts (ALP2-Ranker)

- Training

changed source
code file

```
126 126 @Override
127 127 public void send(GmailMessage message) {
128 128 if (message instanceof JavaMailGmailMessage) {
129 129 Transport transport = null;
130 130 try {
131 131 final JavaMailGmailMessage msg = (JavaMailGmailMessage) message;
132 132 getGmailTransport().sendMessage(
133 133 transport = getGmailTransport();
134 134 transport.sendMessage(
135 135 msg.getMessage(),
136 136 msg.getMessage().getAttachments());
137 137 } catch (final Exception e) {
138 138 throw new GmailException("Failed sending message: " + message, e);
139 139 }
140 140 finally {
141 141 if (transport.isConnected())
142 142 {
143 143 try {
144 144 transport.close();
145 145 } catch (final Exception e) {
146 146 LOG.warn("Cannot close JmapGmailConnection: " + transport, e);
147 147 }
148 148 }
149 149 } else {
150 150 throw new GmailException("JmapGmailClient requires JavaMailGmailMessage");
151 151 }
152 152 }
```

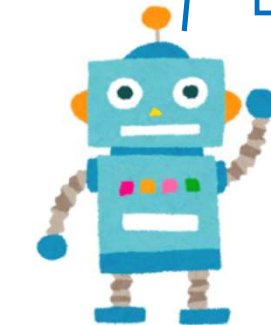
code-inline text features

diff features

software documents



document-text features



Learning to Rank

Learns the most
representative
features from each
knowledge source
for the appropriate
license adoption

co-changed
source code files

```
Showing 6 changed files with 298 additions and 10 deletions.
src/main/java/com/googlecode/gmail4j/javamail/ConnectionInfo.java
src/main/java/com/googlecode/gmail4j/javamail/ImapConnectionHandler.java
src/main/java/com/googlecode/gmail4j/javamail/ImapGmailClient.java
src/main/java/com/googlecode/gmail4j/javamail/ImapGmailConnection.java
src/main/java/com/googlecode/gmail4j/rss/RssGmailClient.java
src/main/java/com/googlecode/gmail4j/util/CommonConstants.java
```

code-inline text features

diff features

Resolve Software License Conflicts (ALP2-Ranker)

Changed source code file

```

125 126 @Override
127 127 public void sendFinal(GmailMessage message) {
128 128 if (message instanceof javax.mail.Message) {
129 129 Transport transport = null;
130 130 try {
131 131 final javax.mail.Message msg = (javax.mail.Message) message;
132 132 getMailTransport().sendMessage(
133 133 transport = getMailTransport();
134 134 transport.sendMessage(
135 135 msg.getMessage(),
136 136 msg.getAllRecipients());
137 137 } catch (final Exception e) {
138 138 throw new GmailException("failed sending message: " + message, e);
139 139 }
140 140 } finally {
141 141 if (transport.isConnected())
142 142 {
143 143 try {
144 144 transport.close();
145 145 } catch (final Exception e) {
146 146 LOG.warn("Cannot Close IMAPConnection: " + transport, e);
147 147 }
148 148 }
149 149 } else {
150 150 throw new GmailException("IMAPClient requires javax.mail.Message!");
151 151 }
152 152 }
    
```

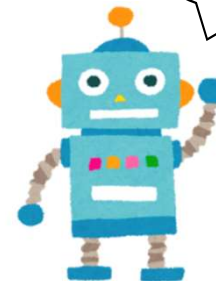
code-inline text features

diff features

Software documents

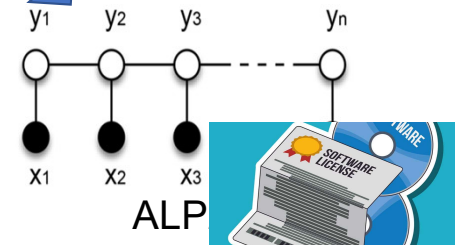


document-text features



Learning to Rank

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



Co-changed source code file

```

Showing 6 changed files with 298 additions and 10 deletions.
src/main/java/com/googlecode/gmail4j/javamail/ConnectionInfo.java
src/main/java/com/googlecode/gmail4j/javamail/ImapConnectionHandler.java
src/main/java/com/googlecode/gmail4j/javamail/ImapGmailClient.java
src/main/java/com/googlecode/gmail4j/javamail/ImapGmailConnection.java
src/main/java/com/googlecode/gmail4j/rss/RssGmailClient.java
src/main/java/com/googlecode/gmail4j/util/CommonConstants.java
    
```

code-inline text features

diff features

Empirical Evaluation

- **Datasets: Open Source Java Projects from GitHub**

- ❖ 700 Java projects from GitHub

# of systems	700
# of commits	8128
# of changed files	57450

(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

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

(b) Per-license frequencies

Changed file	Software document	Co-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...	(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 ...	(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

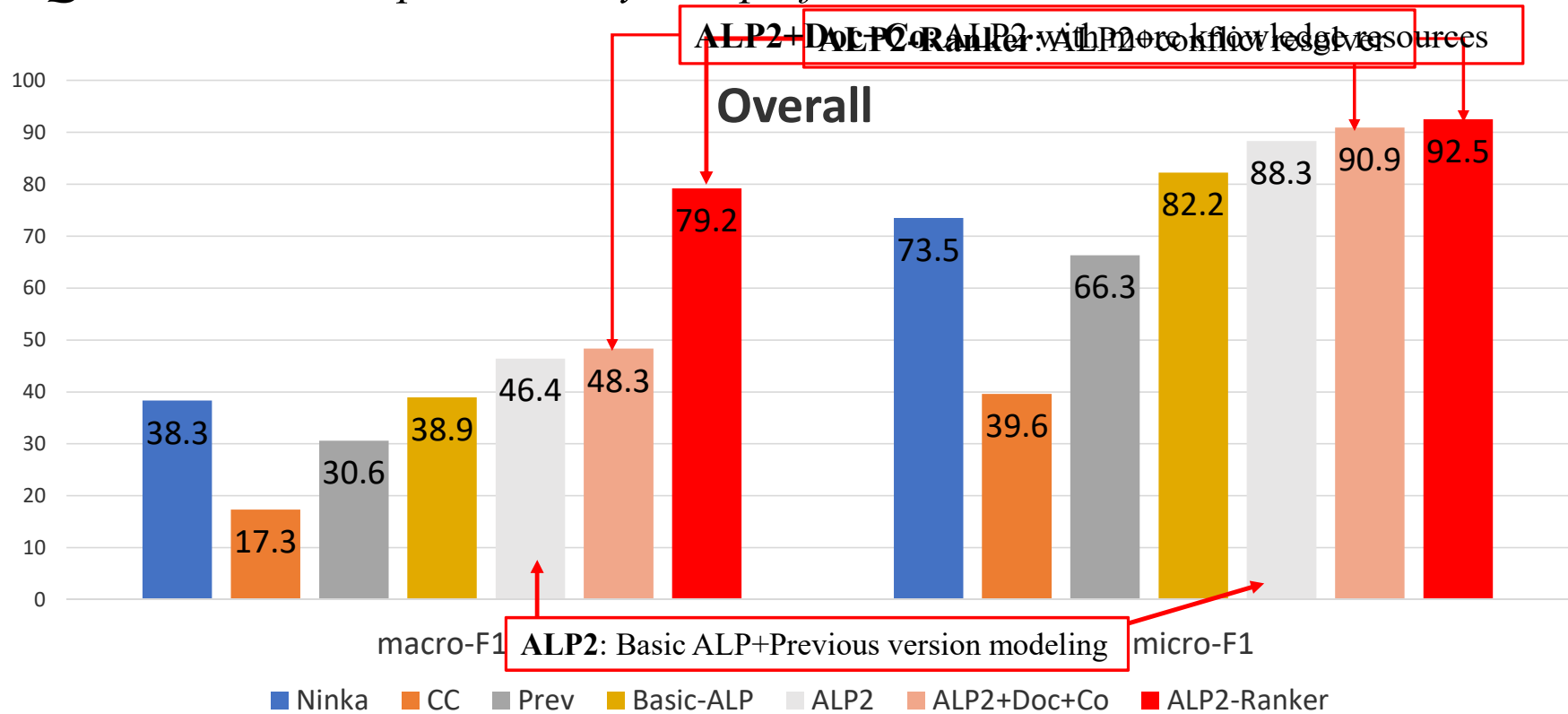
- **Datasets: Open Source Java Projects from GitHub**
 - ❖ 700 Java projects from GitHub, 25 unique licenses
- **Baseline Systems**
 - ❖ *Ninka*
 - ❖ *Caller-Callee (CC)*
 - ❖ *Previous Version (Prev)*: predicts license to be the same as previous version

Empirical Evaluation

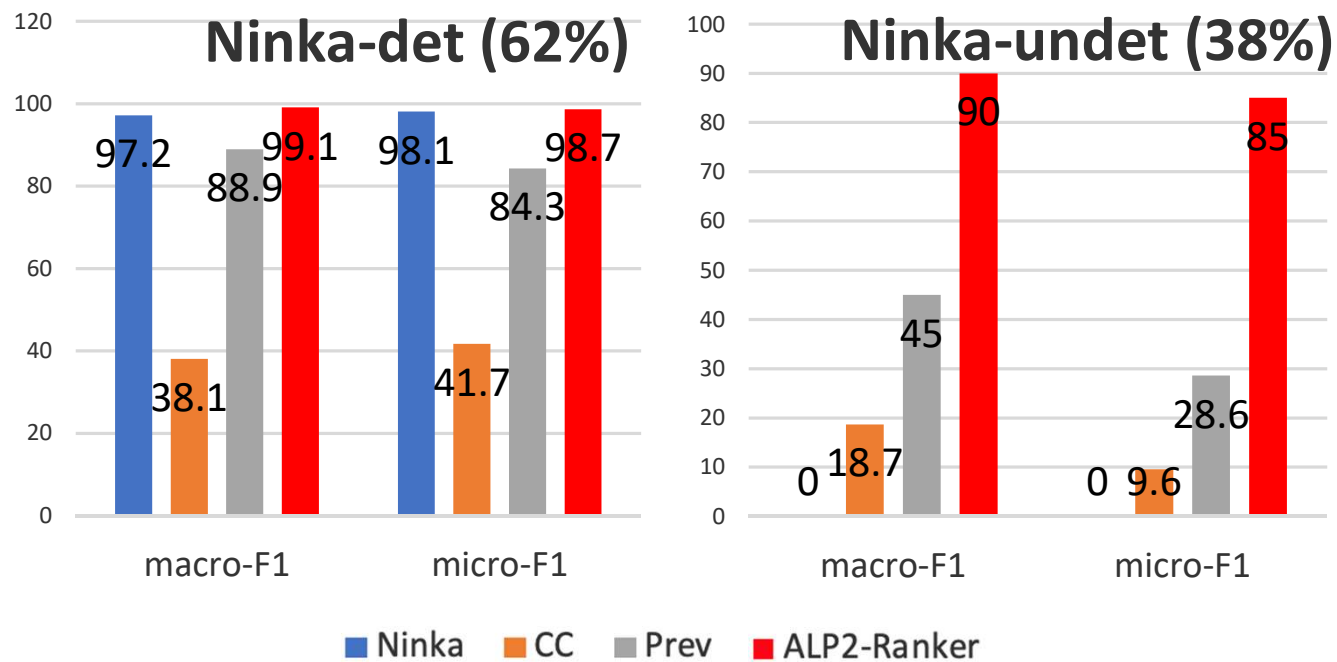
- **Datasets: Open Source Java Projects from GitHub**
 - ❖ 700 Java projects from GitHub, 25 unique licenses
- **Baseline Systems**
 - ❖ *Ninka*
 - ❖ *Caller-Callee (CC)*
 - ❖ *Previous Version (Prev)*: predicts license to be the same as previous version
- **Metrics**
 - ❖ Macro F1: treat each license as equally important
 - ❖ Micro F1: treat most frequent licenses as more important
 - ❖ Five-Fold Cross Validation

Overall Performance

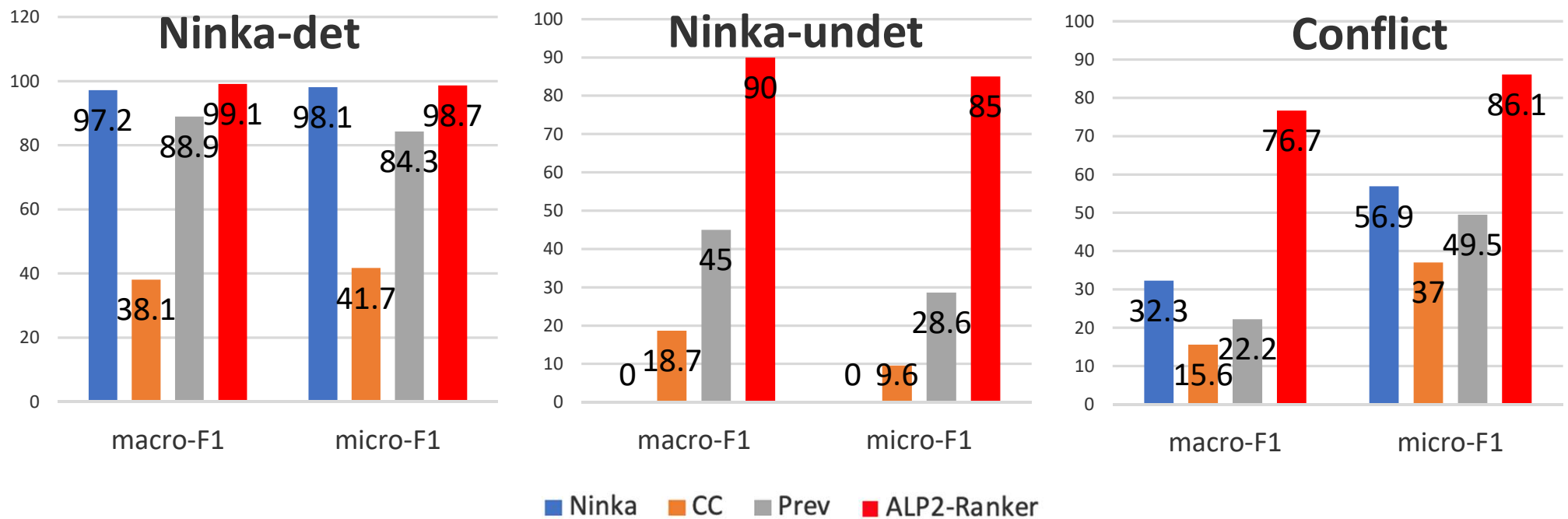
RQ1. Which license prediction system performs the best?



RQ2. *How do the systems perform on the easy and difficult tasks?*



RQ2. *How do the systems perform on the easy, difficult, and conflict instances?*



Goal

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



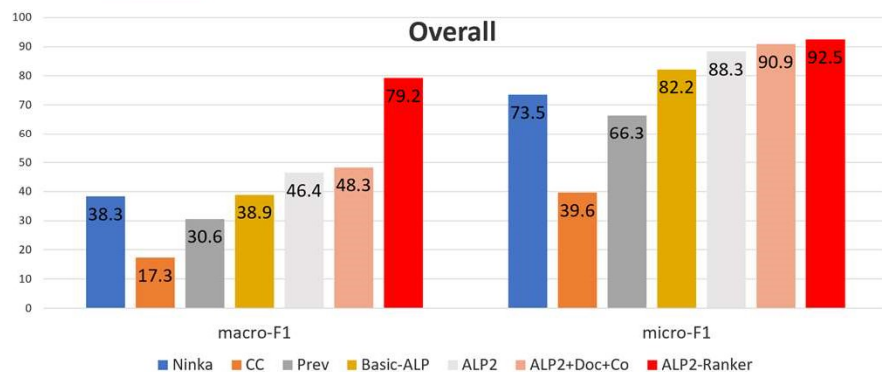
11/14/2019



Overall Performance

RQ1. Which license prediction system performs the best?

ALP2: Basic ALP+Previous version modeling; ALP2+Doc+Co: ALP2+Implicit conflict resolver; ALP2-Ranker: ALP2



11/14/2019



25

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



6

11/14/2019



10



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](mailto:xiaoyul@smu.edu), [lghuang](mailto:lghuang@smu.edu)}@smu.edu gjd@nju.edu.cn vince@hlt.utdallas.edu



1