Duplicate bug report detection through machine learning techniques

Irving Muller Rodrigues
December 10, 2018

Prof. Daniel Aloise and Prof. Michel Dagenais

Polytechnique Montréal
Laboratoire DORSAL
Introduction
Introduction

Java was started but returned exit code=13
-Xms40m
-Xmx384m
-XX:MaxPermSize=256m
-Djava.class.path=C:\Users\LINUX
PLX\Desktop\eclipse\plugins\org.eclipse.equinox.launcher_1.1.1.R36x_v20101122_1400.jar
-os win32
-ws win32
-arch x86
-showsplash C:\Users\LINUX
PLX\Desktop\eclipse\plugins\org.eclipse.platform_3.6.2.v201102101200\splash.bmp
-launcher C:\Users\LINUX PLX\Desktop\eclipse\eclipse.exe
-name Eclipse
--launcher.library C:\Users\LINUX
PLX\Desktop\eclipse\plugins\org.eclipse.equinox.launcher.win32.win32.x86_1.1.2.R36x_v20101222\eclipse_1312.dll
-startup C:\Users\LINUX
PLX\Desktop\eclipse\plugins\org.eclipse.equinox.launcher_1.1.1.R36x_v20101122_1400.jar
-vm C:\Program Files (x86)\Java\jre6\bin\client\jvm.dll
-vmargs
-Xms40m
-Xmx384m
-XX:MaxPermSize=256m
-Djava.class.path=C:\Users\LINUX
PLX\Desktop\eclipse\plugins\org.eclipse.equinox.launcher_1.1.1.R36x_v20101122_1400.jar
Bug Tracking System

Welcome to Bugzilla

File a Bug    Search    Log In    Documentation

Enter a bug # or some search terms

Common Queries:

Bugs reported in the last 24 hours
Bugs changed in the last 24 hours

Home | New | Browse | Search | [?] | Reports | Requests | Log In | Terms of Use | Copyright Agent
Bug Tracking System

User
Tester
Developer

Report

Bug Report
Bug Tracking System

User
Tester
Developer

Report

Bug Report

- Incomplete Bugs
- Invalid Bugs
- Duplicate Bugs
Bug Tracking System

User 
Tester 
Developer

Report

Bug Report

Triage Process

Valid Bug
Invalid Bug
Incomplete Bug
Duplicate Bug
Bug Tracking System

Triage Process

- Manual checking
- Time and money consuming
- Large user base project: Firefox ~300 new reports per day
Objective

- Increase software quality and save resource
  - Decrease triage team overload
  - Avoid two or more developers fixing the same bug
  - Avoid to fix a bug already solved
Duplicate bug report detection

- Detect whether a bug is duplicate or not
- Master set
  - Master report
  - Duplicate reports
  - Every report is in a master set
- Three approaches
  - Decision-making approach
  - Binary classification approach
  - Ranking approach
Decision-making approach

- Pairs of bug reports (Training and Evaluation)
- Drawbacks
  - Too Easy
  - High probability to create easy non-duplicate pairs
  - Far from the real scenario
    - Compare new bug with a set of bugs in the dataset
Binary classification approach

- Automatic prediction of the report as duplicate or not
  - General information extracted from the database and the new bug reports
- False negative can have a great impact
- Really difficult task
Ranking approach

- Recommend a similarity list
- A person check the list and label the report as duplicate or not
  - Decrease the decision time
- The most used approach in the literature
- Metric: Recall Rate
  - Rate of reports whose the lists have at least one bug report from the same master set
Ranking approach

- Two methodologies: Deshmukh et al. 2017 and Sun et al. 2011
  - Deshmukh et al. 2017
    - Training, validation and test datasets are randomly generated
    - Evaluation: similarity list are created using bug from the test dataset
    - Unrealistic scenario
    - It makes the problem easier
      - Decrease number of comparisons
      - Concept Drift mitigation
  - Sun et al. 2011
    - Reports are sorted by creation date
    - Training, validation and test are generate by period of time
    - New bug report is compared with all previous bug reports
    - More realistic scenario
Our Solution

- Ranking approach + Sun’s Methodology
- Only textual data
  - Summary and description
- Baseline: TF-IDF
- Model: Word Embeddings + Convolution Neural Network
TF-IDF

Document

adapter creation gets broken

Content

Term | Value
--- | ---
adapter | $w_1$
gets | $w_2$
broken | $w_3$
creation | $w_4$
TF-IDF

Term | Value
--- | ---
adapter | $w_1$
gets | $w_2$
broken | $w_3$
creation | $w_4$

$w_4 = \text{Term Frequency} \times \text{Inverse Document Frequency}$
TF-IDF

Term |
--- |
adapter |
gets |
broken |
creation |

| Value |
--- |
$w_1$ |
$w_2$ |
$w_3$ |
$w_4$ |

$w_4 = \text{Term Frequency} \times \text{Inverse Document Frequency}$
TF-IDF

Document

adapter creation gets broken

Content

<table>
<thead>
<tr>
<th>Term</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>adapter</td>
<td>$w_1$</td>
</tr>
<tr>
<td>gets</td>
<td>$w_2$</td>
</tr>
<tr>
<td>broken</td>
<td>$w_3$</td>
</tr>
<tr>
<td>creation</td>
<td>$w_4$</td>
</tr>
</tbody>
</table>

$w_4 = 1 \times$ Inverse Document Frequency
**TF-IDF**

A document contains the terms: **adapter creation gets broken**.

<table>
<thead>
<tr>
<th>Term</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>adapter</td>
<td>( w_1 )</td>
</tr>
<tr>
<td>gets</td>
<td>( w_2 )</td>
</tr>
<tr>
<td>broken</td>
<td>( w_3 )</td>
</tr>
<tr>
<td>creation</td>
<td>( w_4 )</td>
</tr>
</tbody>
</table>

\[ w_4 = 1 \times \text{Inverse Document Frequency} \]

\[ \log \left( \frac{\text{Number of documents}}{\text{Document Frequency}} \right) \]
TF-IDF

Document

adapter creation gets broken

Content

Term | Value
---|---
adapter | $w_1$
gets | $w_2$
broken | $w_3$
creation | $w_4$

$w_4 = 1$ x Inverse Document Frequency

\[
\log \left( \frac{10}{8} \right)
\]
**TF-IDF**

Document content:
- adapter creation
- gets broken

The content is processed to extract terms and their associated values.

<table>
<thead>
<tr>
<th>Term</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>adapter</td>
<td>$w_1$</td>
</tr>
<tr>
<td>gets</td>
<td>$w_2$</td>
</tr>
<tr>
<td>broken</td>
<td>$w_3$</td>
</tr>
<tr>
<td>creation</td>
<td>0.09</td>
</tr>
</tbody>
</table>
Represent word as vector

- **Word Embedding**
  - Dense vectors with real numbers
  - More compact representation
  - Semantic and syntactic information

<table>
<thead>
<tr>
<th>Word</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>adapter</td>
<td>[0.5, 0.6]</td>
</tr>
<tr>
<td>broken</td>
<td>[0.3, 0.2]</td>
</tr>
<tr>
<td>gets</td>
<td>[0.1, 0.7]</td>
</tr>
<tr>
<td>creation</td>
<td>[0.6, 0.3]</td>
</tr>
</tbody>
</table>
Convolution Neural Network for NLP

<table>
<thead>
<tr>
<th></th>
<th>Input</th>
<th>Filters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Filter 1</td>
</tr>
<tr>
<td>adapter</td>
<td>0.5 0.6</td>
<td>1. 2.</td>
</tr>
<tr>
<td>creation</td>
<td>0.6 0.3</td>
<td>5. 1.</td>
</tr>
<tr>
<td>gets</td>
<td>0.1 0.7</td>
<td>2. 3.</td>
</tr>
<tr>
<td>broken</td>
<td>0.6 0.3</td>
<td></td>
</tr>
</tbody>
</table>
Convolution Neural Network for NLP

<table>
<thead>
<tr>
<th>Input</th>
<th>Filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>adapter</td>
<td>Filter 1</td>
</tr>
<tr>
<td>creation</td>
<td>Filter 2</td>
</tr>
<tr>
<td>gets</td>
<td></td>
</tr>
<tr>
<td>broken</td>
<td></td>
</tr>
</tbody>
</table>

Apply:

<table>
<thead>
<tr>
<th>adapter</th>
<th>Filter 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1.2</td>
</tr>
<tr>
<td>0.6</td>
<td>5.1</td>
</tr>
<tr>
<td>0.1</td>
<td>2.3</td>
</tr>
<tr>
<td>0.6</td>
<td>1.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>creation</th>
<th>Filter 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>2.5</td>
</tr>
<tr>
<td>0.3</td>
<td>1.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gets</th>
<th>Filter 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1.1</td>
</tr>
<tr>
<td>0.7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>broken</th>
<th>Filter 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>1.1</td>
</tr>
<tr>
<td>0.3</td>
<td></td>
</tr>
</tbody>
</table>
Convolution Neural Network for NLP

Input

adapter 0.5 0.6
creation 0.6 0.3
gets 0.1 0.7
broken 0.6 0.3

Filter 1

\[
\begin{bmatrix}
1. & 2. \\
5. & 1. \\
2. & 3. \\
\end{bmatrix}
\]

\[
\text{sum} \left( \begin{bmatrix}
0.5 & 1.2 \\
3.0 & 0.3 \\
0.2 & 2.1 \\
\end{bmatrix} \right) = 7.3
\]
Convolution Neural Network for NLP

- **Input**
  - **adapter**: 0.5, 0.6
  - **creation**: 0.6, 0.3
  - **gets**: 0.1, 0.7
  - **broken**: 0.6, 0.3

- **Filter 1**
  
  \[
  \begin{array}{cc}
  1 & 2 \\
  5 & 1 \\
  2 & 3 \\
  \end{array}
  \]

  \[= \text{sum} \left( \begin{array}{cc}
  0.6 & 1.6 \\
  0.5 & 0.7 \\
  1.2 & 0.9 \\
  \end{array} \right) = 5.5 \]
Convolution Neural Network for NLP

<table>
<thead>
<tr>
<th>Input</th>
<th>adapter</th>
<th>creation</th>
<th>gets</th>
<th>broken</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5 0.6</td>
<td>0.6 0.3</td>
<td>0.1 0.7</td>
<td>0.6 0.3</td>
</tr>
</tbody>
</table>

Apply

<table>
<thead>
<tr>
<th>Filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter 1</td>
</tr>
<tr>
<td>1. 2.</td>
</tr>
<tr>
<td>5. 1.</td>
</tr>
<tr>
<td>2. 3.</td>
</tr>
</tbody>
</table>

7.3 5.5 5.7 4.4
Convolution Neural Network for NLP

<table>
<thead>
<tr>
<th>Input</th>
<th>adapter</th>
<th>0.5</th>
<th>0.6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>creation</td>
<td>0.6</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>gets</td>
<td>0.1</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>broken</td>
<td>0.6</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Apply Filters

<table>
<thead>
<tr>
<th>Filter 1</th>
<th>Filter 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 2.</td>
<td>2. 5.</td>
</tr>
<tr>
<td>5. 1.</td>
<td>1. 1.</td>
</tr>
<tr>
<td>2. 3.</td>
<td></td>
</tr>
</tbody>
</table>

Max-pooling

Output 7.3 5.7
Our Deep Learning Model

- **Encoder**
  - Represent the report as vector
Our Deep Learning Model

P(D)

Output Layer

Hidden Layer

Hidden Layer

Encoder

Encoder

Bug Report 18042

Bug Report 137861

v^1

v^2

|v^1 - v^2|

v^1 \otimes v^2
Our Deep Learning Model

Cross Entropy

\[ y \times \log(P(D)) + (1 - y) \log(1 - P(D)) \]
## Preliminary Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-5</th>
<th>Top-10</th>
<th>Top-15</th>
<th>Top-20</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF</td>
<td>44.80%</td>
<td>51.27%</td>
<td>54.97%</td>
<td>57.88%</td>
</tr>
<tr>
<td>DL Model</td>
<td>37.11%</td>
<td>43.95%</td>
<td>48.61%</td>
<td>52.03%</td>
</tr>
</tbody>
</table>
Our Deep Learning Model

- **Challenge:**
  - Generate relevant non-duplicate pairs (negative) can be difficult
  - Most non-duplicate pairs are easy
  - \( \sim n^2 \) different combinations
  - \( n = 174,002 \Rightarrow n^2 \approx 30 \times 10^9 \)

- **Solution:** Random subsample negative examples each epoch
  - Constraint: loss has to be greater than 0
  - Keep rate between positive and negative examples
Preliminary Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-5</th>
<th>Top-10</th>
<th>Top-15</th>
<th>Top-20</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF</td>
<td>44.80%</td>
<td>51.27%</td>
<td>54.97%</td>
<td>57.88%</td>
</tr>
<tr>
<td>DL Model</td>
<td>37.11%</td>
<td>43.95%</td>
<td>48.61%</td>
<td>52.03%</td>
</tr>
<tr>
<td>DL Model - subsampling by epoch</td>
<td>44.02%</td>
<td>51.03%</td>
<td>55.49%</td>
<td>58.43%</td>
</tr>
</tbody>
</table>
## Preliminary Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-5</th>
<th>Top-10</th>
<th>Top-15</th>
<th>Top-20</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF</td>
<td>44.80%</td>
<td>51.27%</td>
<td>54.97%</td>
<td>57.88%</td>
</tr>
<tr>
<td>DL Model</td>
<td>37.11%</td>
<td>43.95%</td>
<td>48.61%</td>
<td>52.03%</td>
</tr>
<tr>
<td>DL Model - subsampling by epoch</td>
<td>44.02%</td>
<td>51.03%</td>
<td>55.49%</td>
<td>58.43%</td>
</tr>
</tbody>
</table>

**6.40%**
Future Work

- **Bottleneck: select negative pairs**
  - Try different approaches
- **Encoder receives information from the first bug**
  - Attention
- **Combine different information sources**
  - Categorical information, stack trace, tracing
- **Use our solution to help our partners**
  - Partner data
Thank you for your attention!
Questions?

Irving Muller Rodrigues
irving.muller-rodrigues@polymtl.ca
References


References


References


Represent word as vector

- One hot encoding
  - Binary Vectors
  - Vector Size = Vocabulary Size
  - Curse of Dimensionality

<table>
<thead>
<tr>
<th>Word</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>adapter</td>
<td>[1,0,0,0]</td>
</tr>
<tr>
<td>broken</td>
<td>[0,1,0,0]</td>
</tr>
<tr>
<td>gets</td>
<td>[0,0,1,0]</td>
</tr>
<tr>
<td>creation</td>
<td>[0,0,0,1]</td>
</tr>
</tbody>
</table>
TF-IDF

adaptation gets broken

<table>
<thead>
<tr>
<th>Term</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>adapter</td>
<td>$w_1$</td>
</tr>
<tr>
<td>gets</td>
<td>$w_2$</td>
</tr>
<tr>
<td>broken</td>
<td>$w_3$</td>
</tr>
<tr>
<td>creation</td>
<td>$w_4$</td>
</tr>
</tbody>
</table>

$w_4 = \text{Term Frequency} \times \text{Inverse Document Frequency}$

$$\log \left( \frac{\text{Number of documents}}{\text{Document Frequency}} \right)$$