Using Semantic Similarity in Crawling-based Web Application Testing

Jun-Wei Lin (UC-Irvine)
Farn Wang (National Taiwan Univ.)
Paul Chu (QNAP, Inc)
Crawling-based Web App Testing

- the web app under test as a black-box
- interacting with the app interface
  - DOMs in browsers
- Usage
  - Model-based testing
  - Invariant detection
  - Cross-browser compatibility testing

Crawling-based Web App Testing

Challenges:

• **Input value selection**
  – topic identification

• **GUI state comparison**

Present approaches:

• **Manual labor intensive**

• **application-specific**

• **string-matching based**
  – Written by human

*J.-W. Lin, F. Wang, P. Chu (ICST 2017)*
Present approaches (1/4)

Input Value Selection (Topic Identification)

```javascript
input.id("last_name").setValue("James");
```

In Browser:

Last Name

The DOM Element:

```html
<tr>
  <th align="right"><span>Last Name</span></th>
  <td><input type="text" name="last_name" id="last_name" maxlength="35"></td>
</tr>
```
Present approaches (2/4)

String-matching Based Rules
1. Map the feature string to a topic
2. Select a value from the dataset for the topic

```javascript
input.id("last_name").setValue("James");
```
String-matching Based Rules

```javascript
input.id("last_name").setValue("James");
```

Drawbacks:

- "last name", "family name", "surname", or even randomly generated id?
- id mapped to multiple topics?

E.g., "tel" → telephone
- "ln" → last_name
- "aycreateln" → ?
GUI State Abstraction

• Distinguish newly discovered GUI states from explored ones
• Abstract the states by DOM content filtering
• Application-specific
Observations

• Human interacts with web applications through the text in natural language
  – but not the DOM structures or attributes

• In markup language (e.g. HTML and XML), the reserved words for DOM attributes are limited
  – id, name, type...

• While the words used in text and attributes for input fields of the same topic may be different among web applications, they are usually semantically similar
  – “last name”, “surname”, “family name”
Our Proposal

Inference with Semantic Similarity

Inference with Semantic Similarity

**Running Example**

Training data

<table>
<thead>
<tr>
<th>Last Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td></td>
</tr>
<tr>
<td>Verify Password</td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td></td>
</tr>
</tbody>
</table>

The input field to be inferred

| Family Name |  |

Inference with Semantic Similarity

Feature Extraction

<table>
<thead>
<tr>
<th>Last Name</th>
<th>&lt;th align=&quot;right&quot;&gt; &lt;span&gt;Last Name&lt;/span&gt; &lt;/th&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>&lt;td&gt;&lt;input type=&quot;text&quot; name=&quot;last_name&quot; id=&quot;last_name&quot; maxlength=&quot;35&quot;&gt;&lt;/td&gt;</td>
</tr>
<tr>
<td>Verify Password</td>
<td>&lt;td&gt;&lt;input type=&quot;password&quot; name=&quot;password&quot; id=&quot;password&quot; maxlength=&quot;25&quot;&gt;&lt;/td&gt;</td>
</tr>
<tr>
<td>Email</td>
<td>&lt;td&gt;&lt;input type=&quot;text&quot; name=&quot;email&quot; id=&quot;email&quot; maxlength=&quot;35&quot;&gt;&lt;/td&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;td&gt;&lt;input type=&quot;password&quot; name=&quot;password&quot; id=&quot;password&quot; maxlength=&quot;25&quot;&gt;&lt;/td&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;td&gt;&lt;input type=&quot;password&quot; name=&quot;password&quot; id=&quot;password&quot; maxlength=&quot;25&quot;&gt; check', 'password', 'check', '25'</td>
</tr>
</tbody>
</table>

Inference with Semantic Similarity

Vector Transformation

Bag-of-Words:

\[
X = \begin{pmatrix}
\text{text} & 1 & 1 & 0 & 0 \\
\text{last} & 3 & 0 & 0 & 0 \\
\text{name} & 3 & 0 & 0 & 0 \\
35 & 1 & 1 & 0 & 0 \\
\text{email} & 0 & 3 & 0 & 0 \\
25 & 0 & 0 & 1 & 1 \\
\text{password} & 0 & 0 & 4 & 4 \\
\text{verify} & 0 & 0 & 0 & 1 \\
\text{check} & 0 & 0 & 0 & 2 \\
\end{pmatrix}
\]

Inference with Semantic Similarity

**Vector Transformation**

**Tf-idf:** \[ f_{\text{password}, d3} \log_2 \left( \frac{N}{n_{\text{password}}} \right) = 4 \]  
(Term frequency with inverse document frequency)

\[
X = \begin{pmatrix}
\text{text} & 1 & 1 & 0 & 0 \\
\text{last} & 3 & 0 & 0 & 0 \\
\text{name} & 3 & 0 & 0 & 0 \\
35 & 1 & 1 & 0 & 0 \\
\text{email} & 0 & 3 & 0 & 0 \\
25 & 0 & 0 & 1 & 1 \\
\text{password} & 0 & 0 & 4 & 4 \\
\text{verify} & 0 & 0 & 0 & 1 \\
\text{check} & 0 & 0 & 0 & 2 \\
\end{pmatrix}
\]

\[
\begin{align*}
\text{text} & : 0.1162 & 0.1622 & 0 & 0 \\
\text{last} & : 0.6975 & 0 & 0 & 0 \\
\text{name} & : 0.6975 & 0 & 0 & 0 \\
35 & : 0.1162 & 0.1622 & 0 & 0 \\
\text{email} & : 0 & 0.9733 & 0 & 0 \\
25 & : 0 & 0 & 0.2425 & 0.1644 \\
\text{password} & : 0 & 0 & 0.9701 & 0.6576 \\
\text{verify} & : 0 & 0 & 0 & 0.3288 \\
\text{check} & : 0 & 0 & 0 & 0.6576 \\
\end{align*}
\]

*J.-W. Lin, F. Wang, P. Chu (ICST 2017)*
Inference with Semantic Similarity

Vector Transformation

Latent Semantic Indexing

• Singular Value Decomposition: \( X = U \Sigma V^T \)
  – \( U \): latent concepts in the documents
  – \( \Sigma \): importance of each latent concept
  – \( V^T \): Coordinates of the documents in the latent vector space

• In our experiment, we use genism library.

• Also see [http://www.bluebit.gr/matrix-calculator/](http://www.bluebit.gr/matrix-calculator/)

Inference with Semantic Similarity

**Similarity Calculation**

- With the $U$, $\Sigma$ and $V^T$, we can transform a document $q$ into the latent vector space in which its coordinates $q' = \Sigma^{-1}U^T q$

- Similarity of $q$ to the training documents = Cosine similarity of $q'$ to vectors in $V^T$

*J.-W. Lin, F. Wang, P. Chu (ICST 2017)*
### Inference with Similarity

<table>
<thead>
<tr>
<th>Family Name</th>
<th>Last Name</th>
<th>Password</th>
<th>Verify Password</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.9976</td>
<td>0.0697</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

*J.-W. Lin, F. Wang, P. Chu (ICST 2017)*
Experiment 1

*Input Topic Identification*

- 100 real-world forms of graduate program registration
- Totally 985 input fields

<table>
<thead>
<tr>
<th>Topic</th>
<th>#</th>
<th>Topic</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>password</td>
<td>188</td>
<td>validation_action</td>
<td>1</td>
</tr>
<tr>
<td>email</td>
<td>151</td>
<td>digit-16</td>
<td>1</td>
</tr>
<tr>
<td>last_name</td>
<td>105</td>
<td>ssn-middle</td>
<td>1</td>
</tr>
<tr>
<td>first_name</td>
<td>105</td>
<td>secure_q</td>
<td>1</td>
</tr>
<tr>
<td>username</td>
<td>48</td>
<td>job_title</td>
<td>1</td>
</tr>
<tr>
<td>middle_name</td>
<td>46</td>
<td>ssn-swiss-postfix-2</td>
<td>1</td>
</tr>
<tr>
<td>phone</td>
<td>46</td>
<td>date-yyyy-mm-dd</td>
<td>1</td>
</tr>
<tr>
<td>date-mm/dd/yyyy</td>
<td>43</td>
<td>unknown_hidden</td>
<td>1</td>
</tr>
<tr>
<td>zipcode</td>
<td>41</td>
<td>ssn-postfix</td>
<td>1</td>
</tr>
<tr>
<td>date-mm/yyyy</td>
<td>28</td>
<td>user_status</td>
<td>1</td>
</tr>
<tr>
<td>city</td>
<td>25</td>
<td>visa_number</td>
<td>1</td>
</tr>
<tr>
<td>street-line-2</td>
<td>13</td>
<td>ssn-prefix</td>
<td>1</td>
</tr>
<tr>
<td>street-line-1</td>
<td>13</td>
<td><strong>Total</strong></td>
<td>985</td>
</tr>
</tbody>
</table>

Experiment 1

Input Topic Identification

Steps

• Randomly choose x% of the forms as training data (corpus)
  – x = 10, 20, 30, 40, 50, 60, 70

• Generate rules (i.e. mappings from feature strings to topics) using the training forms

• Infer the rest forms with:
  – The proposed approach (NL)
  – Rule-based approach (RB)
  – RB+NL-n (no-match)
  – RB+NL-m (multiple-topic)
  – RB+NL-b (both)

• Repeat 1000 times

Experiment 1

Input Topic Identification

Result

**TABLE V.** AVERAGE ACCURACIES ACHIEVED BY DIFFERENT METHODS WHEN THE CONSIDERED PERCENTAGES ARE USED AS TRAINING DATA.

<table>
<thead>
<tr>
<th>% training</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NL</td>
</tr>
<tr>
<td>10%</td>
<td>70.42</td>
</tr>
<tr>
<td>20%</td>
<td>72.48</td>
</tr>
<tr>
<td>30%</td>
<td>72.66</td>
</tr>
<tr>
<td>40%</td>
<td>72.67</td>
</tr>
<tr>
<td>50%</td>
<td>73.26</td>
</tr>
<tr>
<td>60%</td>
<td>73.29</td>
</tr>
<tr>
<td>70%</td>
<td>74.05</td>
</tr>
</tbody>
</table>

Experiment 2

GUI State Abstraction

• A real-world web app and its test cases
• The states are manually examined and clustered by an engineer in the company

<table>
<thead>
<tr>
<th>Test Suite</th>
<th>Description</th>
<th># Test Cases</th>
<th># GUI States</th>
<th># Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>install_wiz</td>
<td>Installation wizard</td>
<td>6</td>
<td>60</td>
<td>22</td>
</tr>
<tr>
<td>rule</td>
<td>NAS Rule management</td>
<td>3</td>
<td>237</td>
<td>80</td>
</tr>
<tr>
<td>server_add</td>
<td>NAS addition and removal</td>
<td>6</td>
<td>200</td>
<td>88</td>
</tr>
<tr>
<td>server_app</td>
<td>App management and config backup</td>
<td>7</td>
<td>207</td>
<td>42</td>
</tr>
<tr>
<td>settings</td>
<td>Account and server settings management</td>
<td>8</td>
<td>451</td>
<td>84</td>
</tr>
</tbody>
</table>
Experiment 2

Gui State Abstraction

Abstraction Methods

- **WS (White Space)**
  - Replace all line breaks and tabs with white space
  - Collapse white space

- **TagAttrWD**
  - Keep only tag names and important attributes
  - Remove timestamps
  - WS abstraction

- **NL**
  - Use enclosed text in visible DOM elements
  - A similarity threshold to determine equivalence
Experiment 2

GUI State Abstraction

Result

TABLE IX. F-MEASURE OF THE CLUSTERING RESULTS

<table>
<thead>
<tr>
<th>Test Suite</th>
<th>F-measure</th>
<th>F-measure</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WS</td>
<td>TagAttrWD</td>
<td>NL</td>
</tr>
<tr>
<td>install_wiz</td>
<td>0.7817</td>
<td>0.8194</td>
<td>0.7826</td>
</tr>
<tr>
<td>rule</td>
<td>0.3241</td>
<td>0.3599</td>
<td>0.4443</td>
</tr>
<tr>
<td>server_add</td>
<td>0.4281</td>
<td>0.4751</td>
<td>0.6776</td>
</tr>
<tr>
<td>server_app</td>
<td>0.1532</td>
<td>0.1809</td>
<td>0.3559</td>
</tr>
<tr>
<td>settings</td>
<td>0.1180</td>
<td>0.4512</td>
<td>0.4156</td>
</tr>
</tbody>
</table>
Contribution

• Natural language techniques for automating crawling-based web application testing
  – Input topic identification and value selection
  – State equivalence checking

• Experiments

Future Work

• The impact overall crawling efficacy with more data and other topic model alternatives such as LDA
• Information retrieval from, e.g., comments, of DOMs
• Mobile apps?