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**

Present approaches (1/4)

Input Value Selection (Topic Identification)

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

In Browser:

Last Name

The DOM Element:

```
<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

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

Present approaches (3/4)

String-matching Based Rules

```
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" → ?

Present approaches (4/4)

GUI State Abstraction

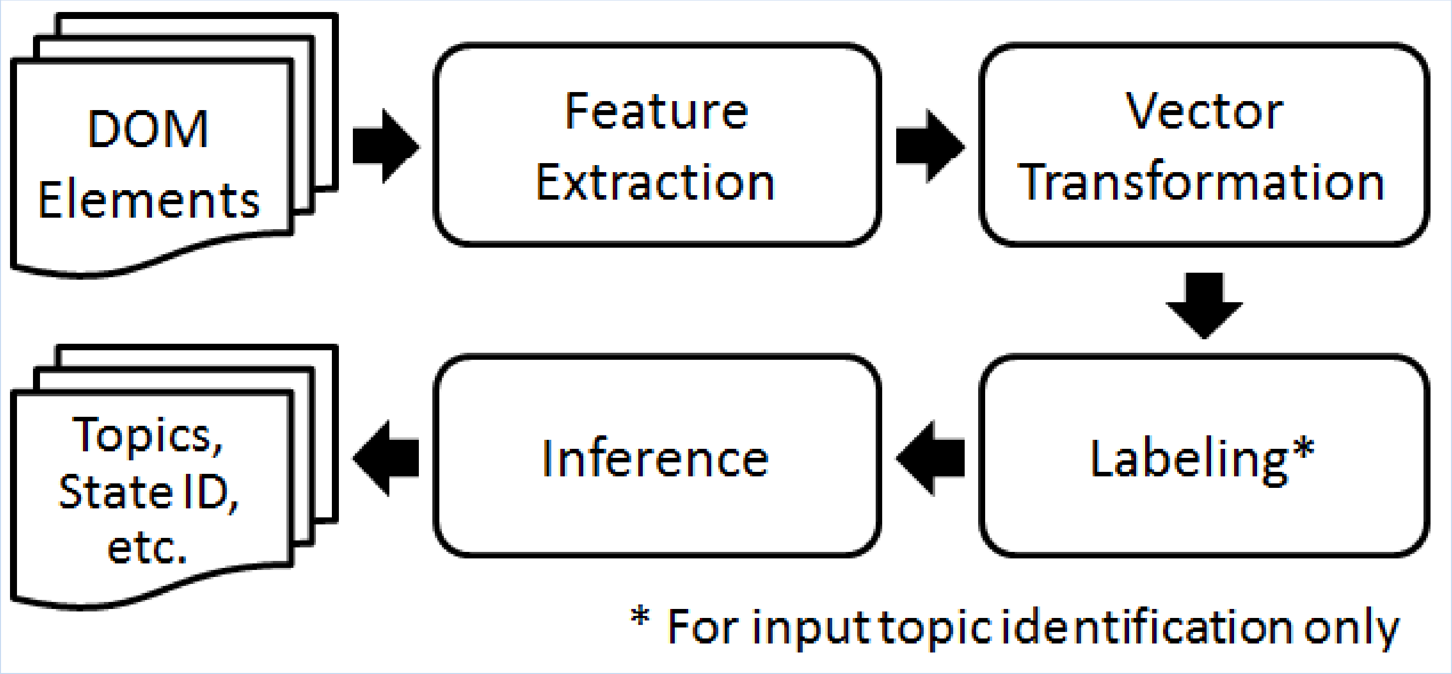
- Distinguish newly discovered GUI states from explored ones
- Abstract the states by DOM content filtering
- Application-specific

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

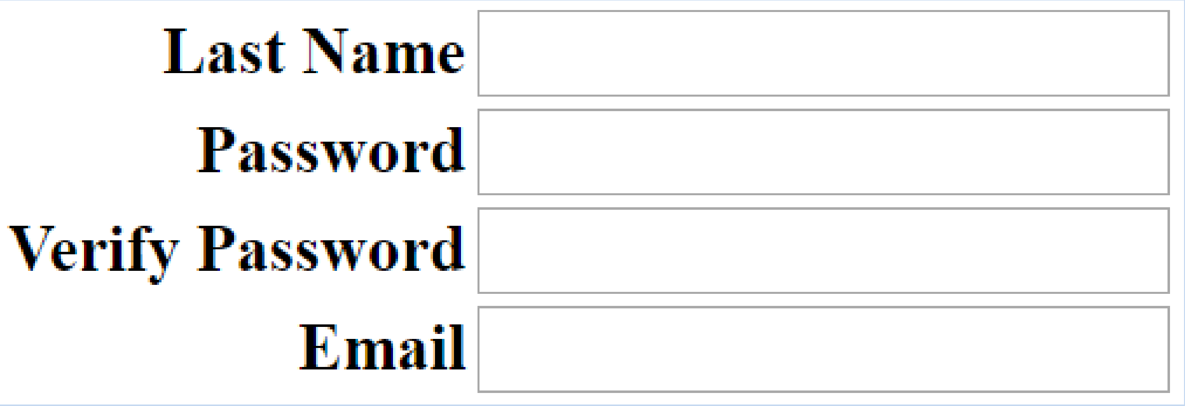


J.-W. Lin, F. Wang, P. Chu (ICST 2017)

Inference with Semantic Similarity

Running Example

Training data



The input field to be inferred

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

Inference with Semantic Similarity

Feature Extraction

Last Name
Password
Verify Password
Email

```
<tr>
  <th align="right"><span>Last Name</span></th>
  <td><input type="text" name="last_name"
       id="last_name" maxlength="35"></td></tr>
<tr>
  <th align="right"><span>Password</span></th>
  <td><input type="password" name="password"
       id="password" maxlength="25"></td></tr>
```

```
['last', 'name', 'text', 'last', 'name', 'last', 'name', '35']
['email', 'text', 'email', 'email', '35']
['password', 'password', 'password', 'password', '25']
['verify', 'password', 'password', 'check', 'password',
 'check', '25']
```

```
       maxlength="35"></td></tr>
```

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

Inference with Semantic Similarity

Vector Transformation

```
['last', 'name', 'text', 'last', 'name', 'last', 'name', '35']
['email', 'text', 'email', 'email', '35']
['password', 'passwor
['verify', 'password',
 'check', '25']
```

Bag-of-Words:

$$X = \begin{array}{cc} & \begin{array}{cccc} d_1 & d_2 & d_3 & d_4 \end{array} \\ \begin{array}{l} text \\ last \\ name \\ 35 \\ email \\ 25 \\ password \\ verify \\ check \end{array} & \begin{pmatrix} 1 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 \end{pmatrix} \end{array}$$

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

Inference with Semantic Similarity

Vector Transformation

Tf-idf: $f_{\text{"password"},d3} \log_2(N/n_{\text{"password"}})=4$

(Term frequency with inverse document frequency)

$$
X = \begin{array}{l} \\ text \\ last \\ name \\ 35 \\ email \\ 25 \\ password \\ verify \\ check \end{array}
\begin{array}{c} \begin{array}{cccc} d_1 & d_2 & d_3 & d_4 \end{array} \\
\begin{pmatrix}
1 & 1 & 0 & 0 \\
3 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 3 & 0 & 0 \\
0 & 0 & 1 & 1 \\
0 & 0 & 4 & 4 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 2
\end{pmatrix} \end{array}
$$

$$
\begin{array}{l} \\ text \\ last \\ name \\ 35 \\ email \\ 25 \\ passwoda \\ verify \\ check \end{array}
\begin{array}{c} \begin{array}{cccc} d_1 & d_2 & d_3 & d_4 \end{array} \\
\begin{pmatrix}
0.1162 & 0.1622 & 0 & 0 \\
0.6975 & 0 & 0 & 0 \\
0.6975 & 0 & 0 & 0 \\
0.1162 & 0.1622 & 0 & 0 \\
0 & 0.9733 & 0 & 0 \\
0 & 0 & 0.2425 & 0.1644 \\
0 & 0 & 0.9701 & 0.6576 \\
0 & 0 & 0 & 0.3288 \\
0 & 0 & 0 & 0.6576
\end{pmatrix} \end{array}
$$

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

Inference with Semantic Similarity
Vector Transformation

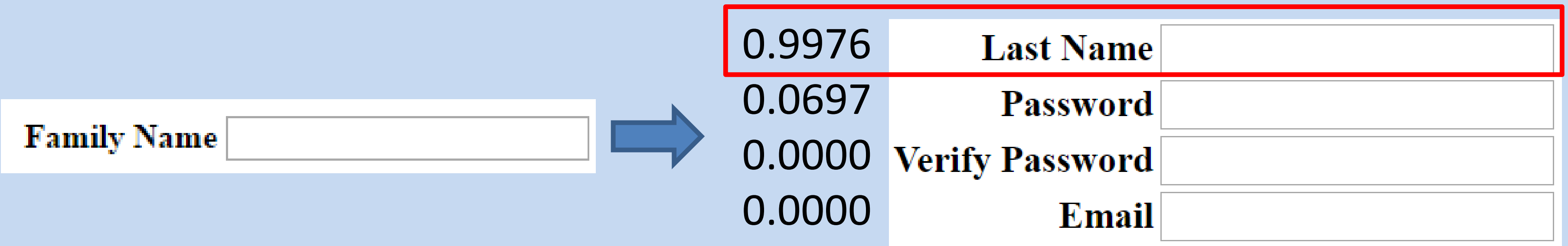
Latent Semantic Indexing

- Singular Value Decomposition: $X = U\Sigma V^T$
  - U: latent concepts in the documents
  - Σ: 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/

Inference with Semantic Similarity
Similarity Calculation

- With the U, Σ 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

Inference with Similarity




```
<th align="right"><span>Family Name</span></th>
<td><input type="text" id="textfield-1029-inputEl"
    name="1023000000003017"></td>
```

Experiment 1
Input Topic Identification

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

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

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

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.

% training	Accuracy (%)				
	NL	RB	RB+ NL-n	RB+ NL-m	RB+ NL-b
10%	70.42	75.60	75.70	82.13	82.23
20%	72.48	75.81	75.85	85.00	85.04
30%	72.66	75.04	75.05	86.18	86.19
40%	72.67	74.14	74.14	86.86	86.86
50%	73.26	73.50	73.50	87.47	87.47
60%	73.29	72.64	72.64	87.54	87.54
70%	74.05	72.44	72.44	88.39	88.39

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

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

Test Suite	Description	# Test Cases	# GUI States	# Clusters
install_wiz	Installation wizard	6	60	22
rule	NAS Rule management	3	237	80
server_add	NAS addition and removal	6	200	88
server_app	App management and config backup	7	207	42
settings	Account and server settings management	8	451	84

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

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

Test Suite	F-measure		
	WS	TagAttrWD	NL
install_wiz	0.7817	**0.8194**	0.7826
rule	0.3241	0.3599	**0.4443**
server_add	0.4281	0.4751	**0.6776**
server_app	0.1532	0.1809	**0.3559**
settings	0.1180	**0.4512**	0.4156

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

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 ?