NLP2TestableCode: Optimising the Fit of Stack Overflow Code Snippets into Existing Code

Brittany Reid, Christoph Treude & Markus Wagner
The University of Adelaide
Motivational Example

- Code reuse is a time consuming process
  - Many snippets to evaluate
- Only 8.41% of code snippets on Stack Overflow compile\(^1\)

Motivational Example

Alternatively, you can use an `Ints` method from the Guava library, which in combination with Java 8's `Optional`, makes for a powerful and concise way to convert a string into an int:

```java
import com.google.common.primitives.Ints;

int foo = Optional.ofNullable(myString)
    .map(Ints::tryParse)
    .orElse(0);
```

https://stackoverflow.com/a/5585800
public class Main{
    public static void main(String[] args){

        import com.google.common.primitives.Ints;
        int foo = Optional.ofNullable(myString)
        .map(Ints::tryParse)
        .orElse(0)

    }
}
NLP2TestableCode

https://www.youtube.com/watch?v=jI2uqfyJaM
Process

- Natural language task
**Process**

- Natural language task
- Search for code snippets from offline SO database
Process

- Natural language task
- Search for code snippets from offline SO database
- For each snippet:
  - Compile
  - Count errors
  - Add compiling snippets to final set
**Process**

- Natural language task
- Search for code snippets from offline SO database
- For each snippet:
  - Compile
  - Count errors
  - Add compiling snippets to final set
- If snippets don’t compile, try to correct
- Insert best snippet

**Flowchart:***

1. Natural Language Task
2. Search Stack Overflow Database
3. Code Snippets
4. For Each Snippet:
   - Compile
   - Has Errors?
     - YES: Targeted Fixes
     - NO: Has Errors?
       - YES: Line Deletion
       - NO: Add to Final Set of Snippets
5. Final Set of Snippets
6. Best Snippet
7. CODE CORRECTION:
   - Integrate
   - Has Errors?
     - YES: Targeted Fixes
     - NO: Add to Final Set of Snippets
Using Natural Language Tasks to Find Relevant Snippets

- Offline SO database
  - 1.5 million questions and 2.5 million answers
- Extract code snippets
- Match tasks to question titles
- Task suggestions extracted from Stack Overflow\[^1\] using TaskNav\[^2\]

---

Using Natural Language Tasks to Find Relevant Snippets

- Question titles and queries are processed for keywords
- Title keywords are associated with a list of threads

```
“Get data from programmatically added view in Java”
  convert to lowercase, remove stop words

“Get data programmatically added view”
  split by whitespace into array of words

get  data  programmatically  added  view

get  datum  programmatically  add  view

```

<table>
<thead>
<tr>
<th>ID</th>
<th>TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“get index of substring”</td>
</tr>
<tr>
<td>2</td>
<td>“try get catch”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{“get”, “index”, “of”, “substring”}</td>
</tr>
<tr>
<td>2</td>
<td>{“try”, “get”, “catch”}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WORD</th>
<th>THREAD IDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>{1, 2}</td>
</tr>
<tr>
<td>index</td>
<td>{1}</td>
</tr>
<tr>
<td>of</td>
<td>{1}</td>
</tr>
<tr>
<td>substring</td>
<td>{1}</td>
</tr>
<tr>
<td>try</td>
<td>{2}</td>
</tr>
<tr>
<td>catch</td>
<td>{2}</td>
</tr>
</tbody>
</table>
How many snippets can we retrieve?

- For 47 sample tasks:
  - These tasks are sourced from a user study for NLP2Code\(^1\)

<table>
<thead>
<tr>
<th>Omit Stop Words?</th>
<th>No Processing</th>
<th>Stemming with Porter Stemming</th>
<th>Lemmatisation with CoreNLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>2832</td>
<td>4100</td>
<td>5091</td>
</tr>
<tr>
<td>Yes</td>
<td>3464</td>
<td>5646</td>
<td>6954</td>
</tr>
</tbody>
</table>

Evaluating Code Quality

- Code snippet is combined with the users code

```
public class Main{
    public static void main(String[] args){
        int result = Integer.parseInt(args[0]);
    }
}
```

- Compile this code in memory, then count errors
  - By not creating files on the disk, we speed up compilation time
How many code snippets compile?

- 327 out of 6954 snippets (4.7%)
Code Correction

- Automatic Integration
- Targeted Fixes
- Line Deletion
Integration

- Moving import statements
- Removing nested functions and classes
Integration

- Moving import statements
- Removing nested functions and classes

```java
public class Main{
    public static void main(String[] args){
        int result = Integer.parseInt(args[0]);
        }
```

```java
public class Main{
    public static void main(String[] args){
        int result = Integer.parseInt(args[0]);
        }
```
Integration

- Increases compilable code snippets from 327 to 470 (43.7%).
Targeted Fixes

- Target common compiler errors
  - Missing semicolons
  - Missing import statements
  - Missing variable declarations

- Increases compilable snippets to 968 (106%).
Line Deletion

- Local search algorithm
  - Delete a line
  - Compile for error count
  - Accept deletion if it doesn’t increase errors
  - Move to next line
  - Continue looping over snippet until no more changes can be made

- Small changes to find ‘optimal’ snippet
- Increases snippets to 2,037 (110%)
- Total increase in compilable snippets using fixes is 522.9%
Automated Testing

- Suggesting Argument and Return Types
- Building a testable function
- Running tests
Suggesting Argument and Return Types

- Look at variable declarations for arguments
- Look at variable assignments for return type
- Use JavaParser to analyse code (https://javaparser.org/)

Input

```java
String myString = "empty";
int foo = 0;
foo = Optional.ofNullable(myString)
    .map(Ints::tryParse)
    .orElse(0);
```

Output

**Argument:** String  
**Return:** int
Generating a JUnit test case

- Using suggested or user supplied type information, we generate a customizable JUnit test case
- Use JavaParser to generate ‘default’ values for types

```java
@Test
public void JUnitTest(){
    assertEquals(snippet("empty"), 0);
}
```
Generating a testable function

● Convert a snippet into a function with input and output
● Similar to searching for argument and return types
  ○ This time, we only look for variables of specific types

```java
public static int snippet(String myString) {
    int foo = 0;
    foo = Optional.ofNullable(myString)
        .map(Ints::tryParse)
        .orElse(0);
    return foo;
}
```
Running tests

- Use JUnit to run test cases
- Tests are run in a separate process, and can be timed out
- For each code snippet, we record if the test passes
- Code snippets that pass tests are ranked above other snippets
Code on Github:
https://github.com/Brittany-Reid/nlp2testablecode

Email:
brittany.reid@adelaide.edu.au