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

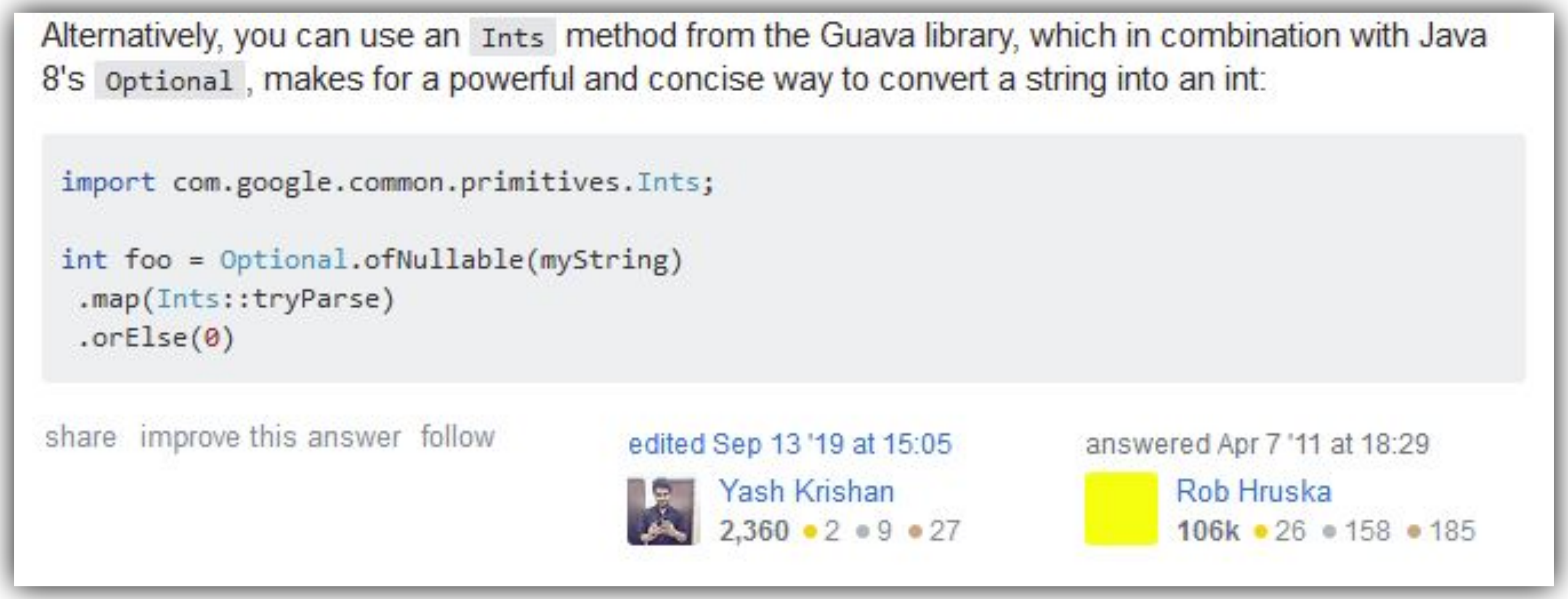
Brittany Reid, Christoph Treude & Markus Wagner
The University of Adelaide

Motivational Example

- Code reuse is a time consuming process
  - Many snippets to evaluate
- Only 8.41% of code snippets on Stack Overflow compile[1]

1. Valerio Terragni, Yepang Liu, and Shing-Chi Cheung. 2016. CSNIPPEX: Automated Synthesis of Compilable Code Snippets from Q&A Sites. InProceedings ofthe 2016 International Symposium on Software Testing and Analysis, ISSTA 2016.ACM, 118–129.

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:

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

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

share improve this answer follow

edited Sep 13 '19 at 15:05
Yash Krishan
2,360 ● 2 ● 9 ● 27

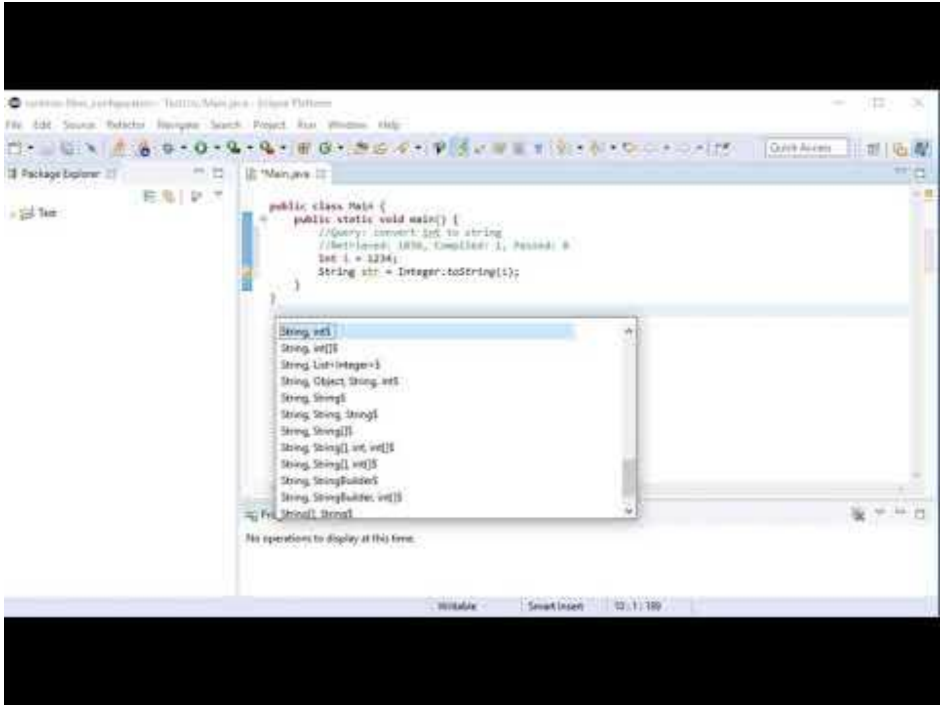
answered Apr 7 '11 at 18:29
Rob Hruska
106k ● 26 ● 158 ● 185

https://stackoverflow.com/a/5585800

Motivational Example

```
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



File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
Test
*Main.java
Writable
Smart Insert

```
public class Main {
    public static void main() {
        //Query: convert int to string
        //Retrieved: 1030, Compiled: 1, Passed: 0
        int i = 1234;
        String str = Integer.toString(i);
    }
}
```

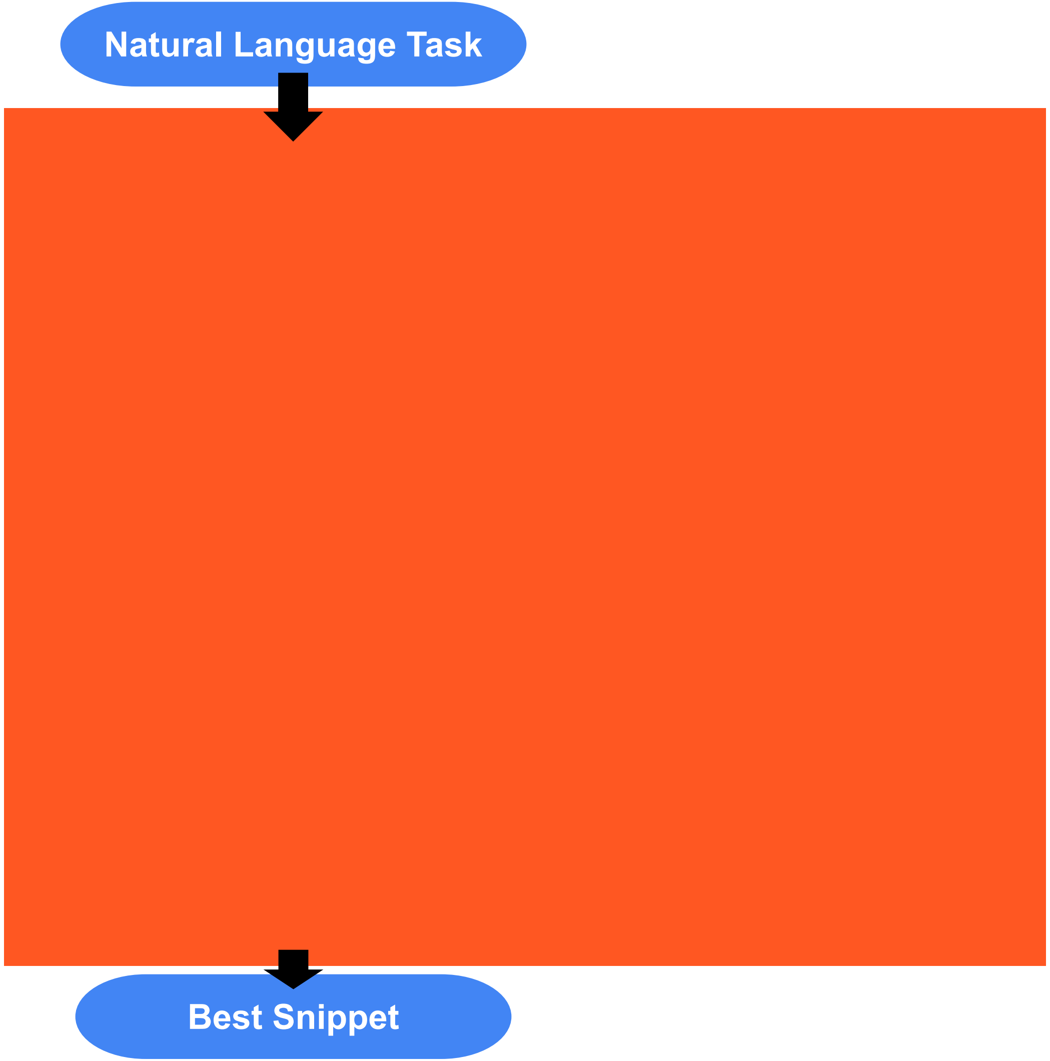
```
String, int$
String, int[]$
String, List<Integer>$
String, Object, String, int$
String, String$
String, String, String$
String, String[]$
String, String[], int, int[]$
String, String[], int[]$
String, StringBuilder$
String, StringBuilder, int[]$
String[], String$
```

No operations to display at this time.

https://www.youtube.com/watch?v=jIA2uqfyJaM

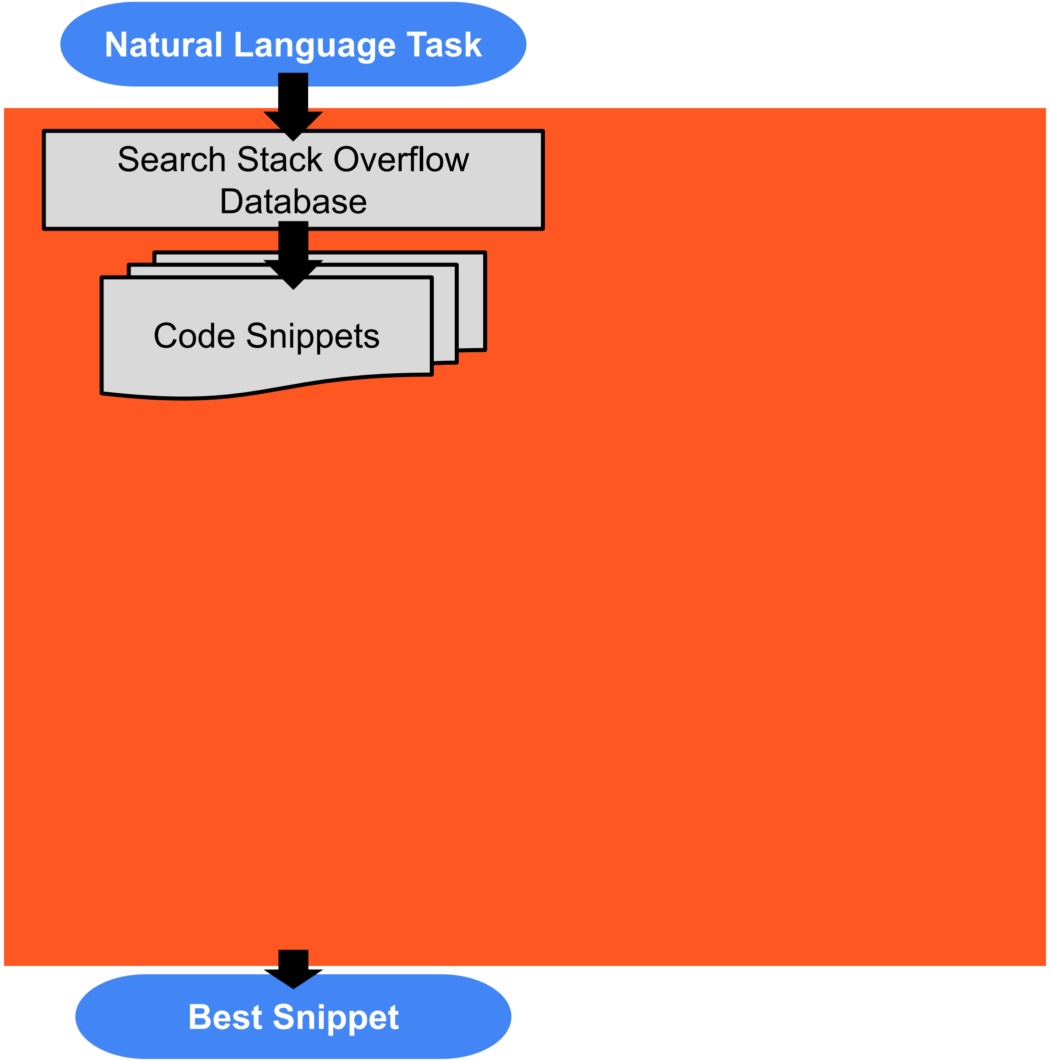
Process

- Natural language task



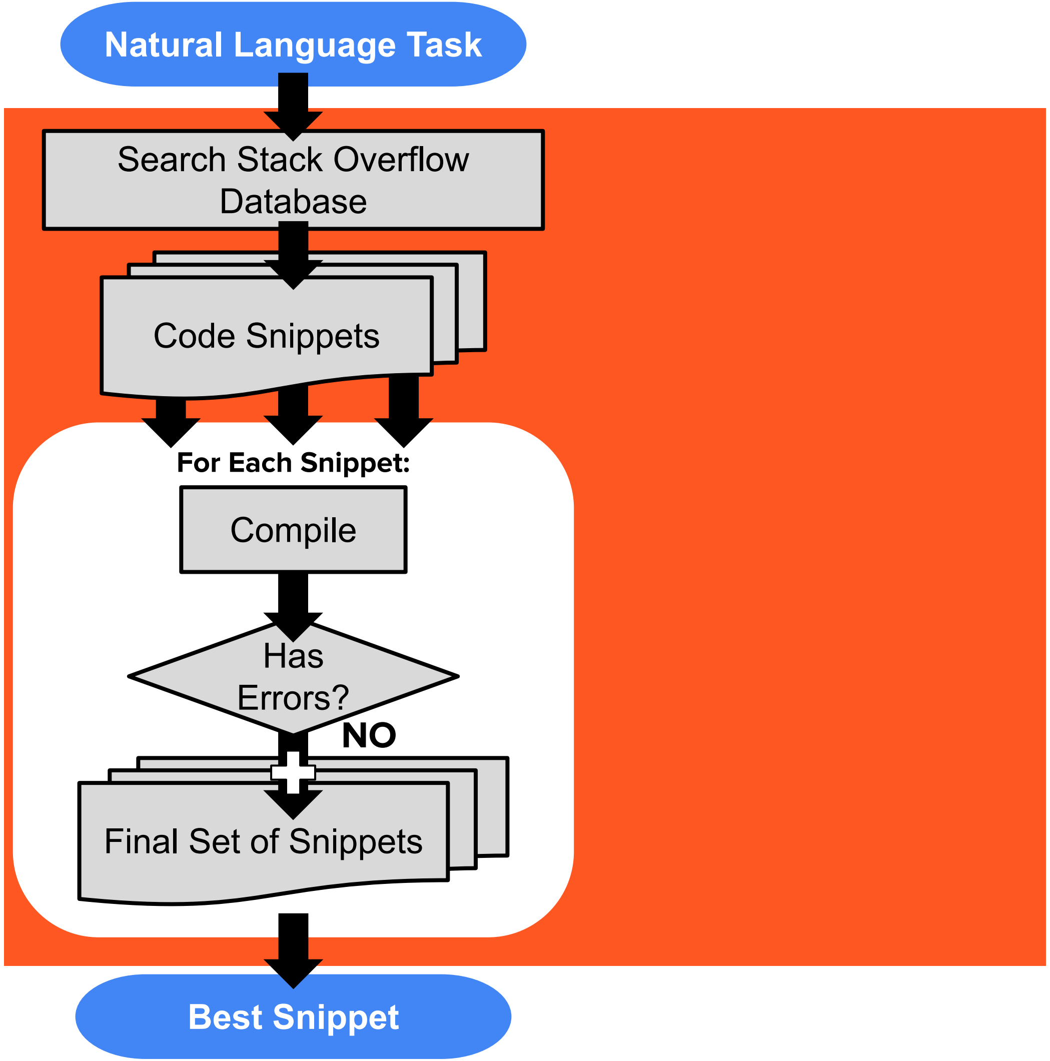
Process

- 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



Natural Language Task

Search Stack Overflow Database

Code Snippets

For Each Snippet:

Compile

Has Errors?

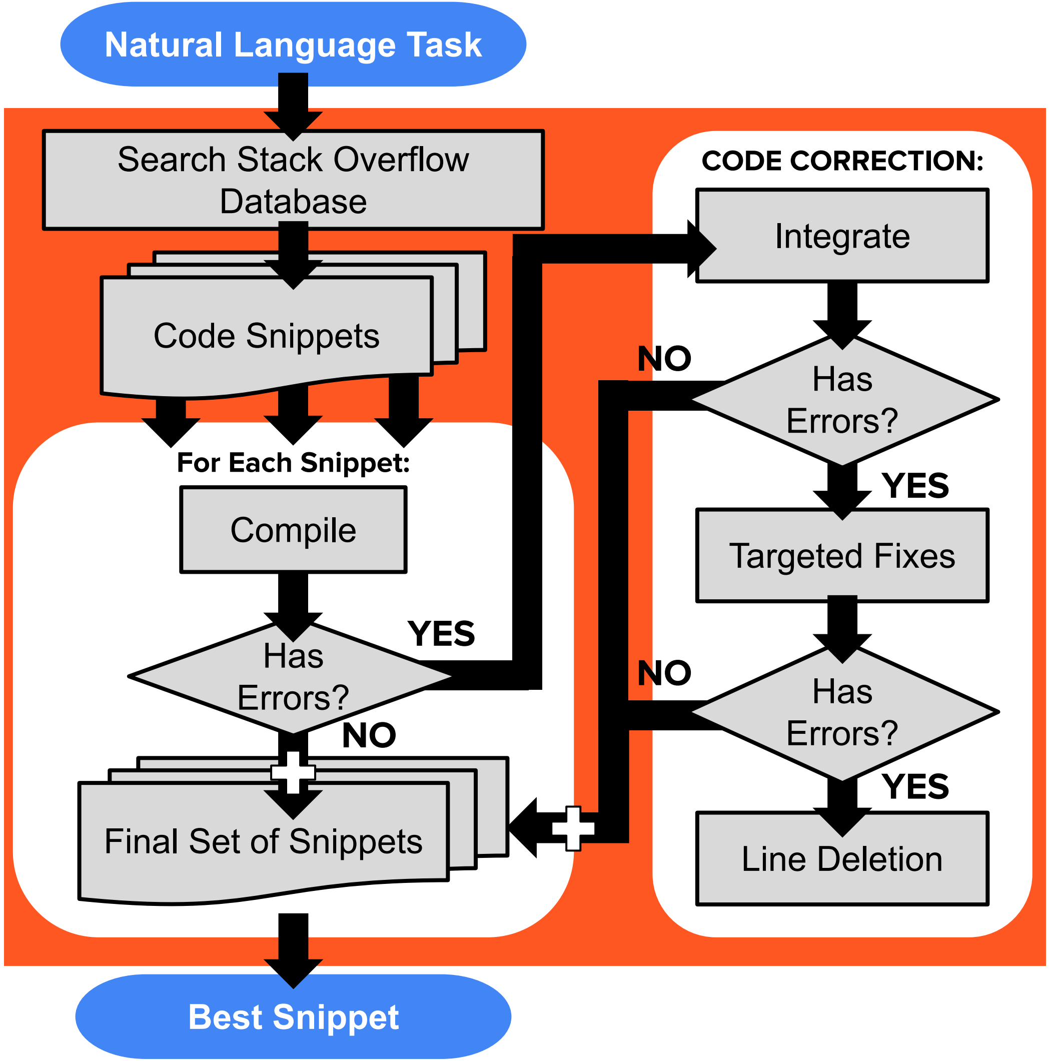
NO

Final Set of Snippets

Best Snippet

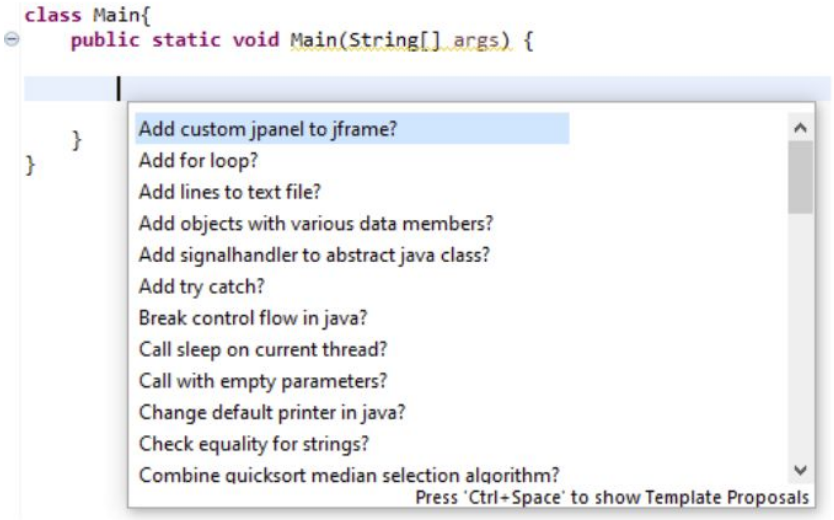
Process

- 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



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]



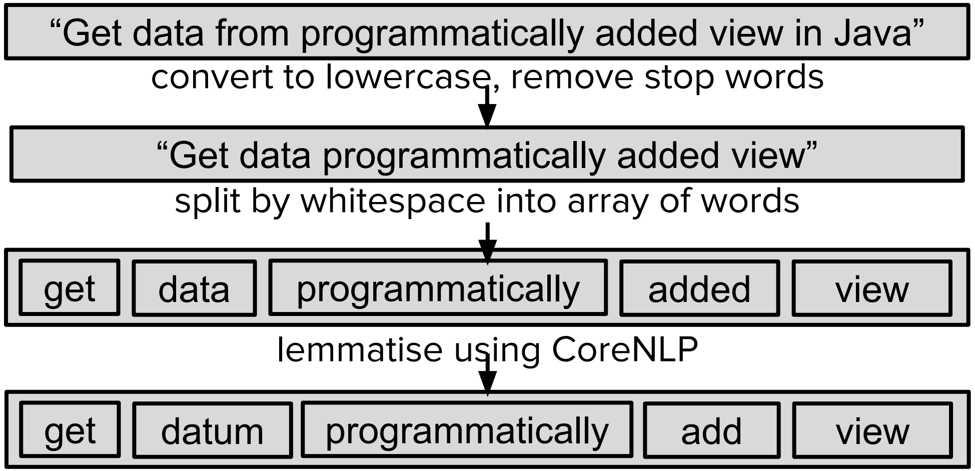
```
class Main{
    public static void Main(String[] args) {

    }
}
```

Add custom jpanel to jframe?
Add for loop?
Add lines to text file?
Add objects with various data members?
Add signalhandler to abstract java class?
Add try catch?
Break control flow in java?
Call sleep on current thread?
Call with empty parameters?
Change default printer in java?
Check equality for strings?
Combine quicksort median selection algorithm?
Press 'Ctrl+Space' to show Template Proposals

1. Tasks from: Campbell, Brock Angus and Christoph Treude. "NLP2Code: Code Snippet Content Assist via Natural Language Tasks." *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (2017): 628-632.
2. Task extraction tool: C. Treude, M. P. Robillard and B. Dagenais, "Extracting Development Tasks to Navigate Software Documentation," in *IEEE Transactions on Software Engineering*, vol. 41, no. 6, pp. 565-581, 1 June 2015, doi: 10.1109/TSE.2014.2387172.

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

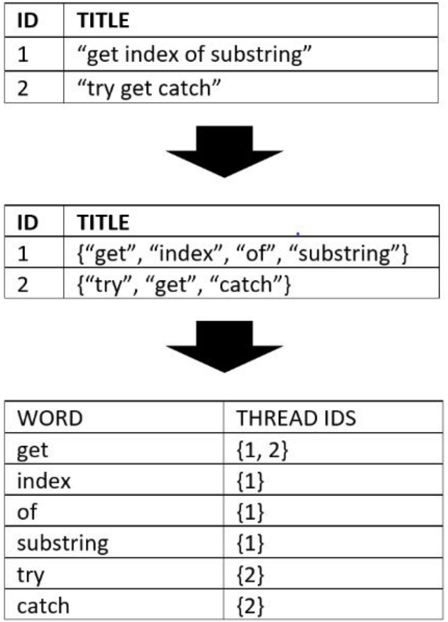
“Get data programmatically added view”

split by whitespace into array of words

get	data	programmatically	added	view

lemmatise using CoreNLP

get	datum	programmatically	add	view



ID	TITLE
1	“get index of substring”
2	“try get catch”

ID	TITLE
1	{“get”, “index”, “of”, “substring”}
2	{“try”, “get”, “catch”}

WORD	THREAD IDS
get	{1, 2}
index	{1}
of	{1}
substring	{1}
try	{2}
catch	{2}

How many snippets can we retrieve?

- For 47 sample tasks:
    - These tasks are sourced from a user study for NLP2Code[1]

Omit Stop Words?	No Processing	Stemming with Porter Stemming	Lemmatisation with CoreNLP
No	2832	4100	5091
Yes	3464	5646	6954

[1] Campbell, Brock Angus and Christoph Treude. “NLP2Code: Code Snippet Content Assist via Natural Language Tasks.” *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (2017): 628-632.

Evaluating Code Quality

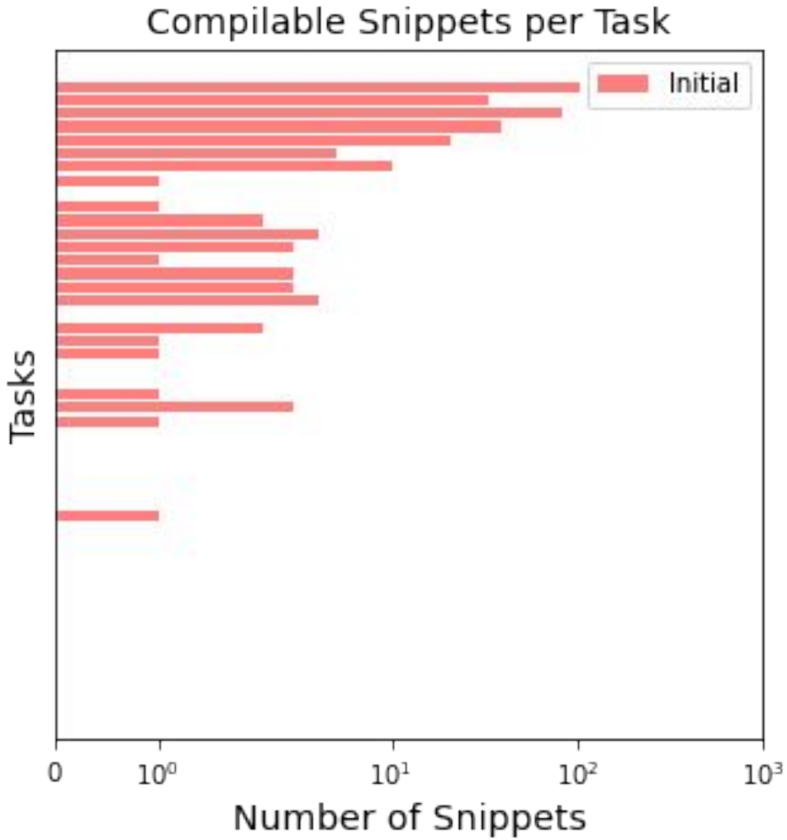
- Code snippet is combined with the users code

```
public class Main{
      public static void main(String[] args){

      + 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%)

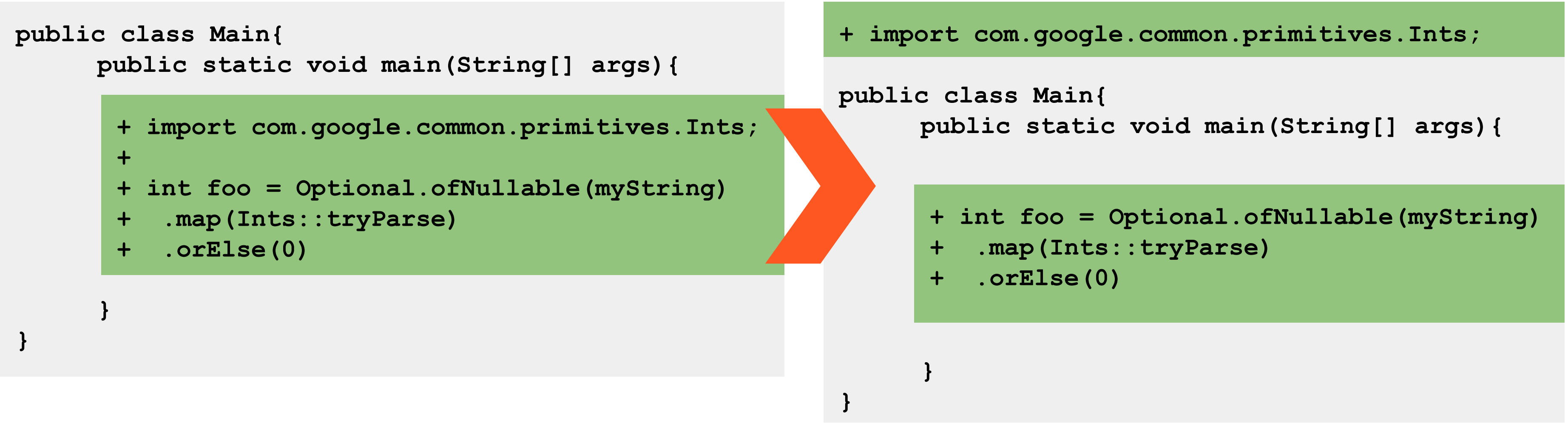


Code Correction

- Automatic Integration
- Targeted Fixes
- Line Deletion

Integration

- Moving import statements
- Removing nested functions and classes

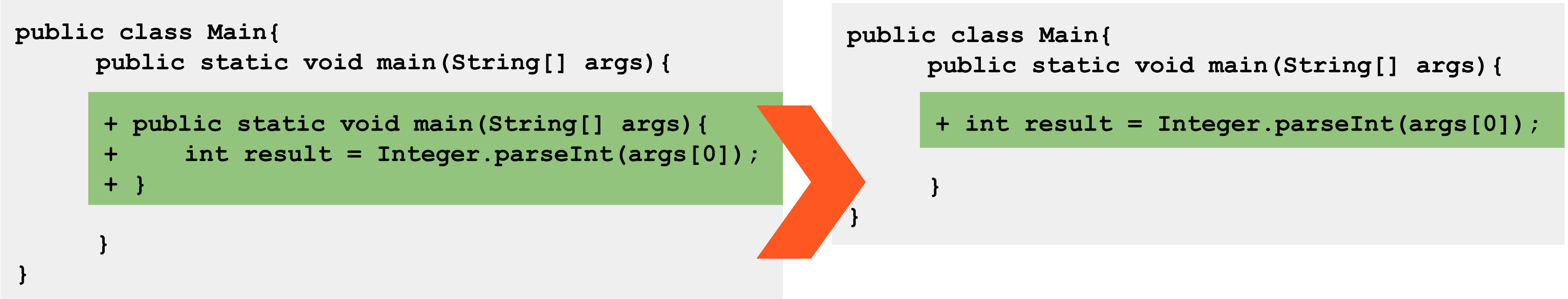


```
public class Main{
      public static void main(String[] args){

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

      }
}
```

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

public class Main{
      public static void main(String[] args){


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


      }
}
```

Integration

- Moving import statements
- Removing nested functions and classes

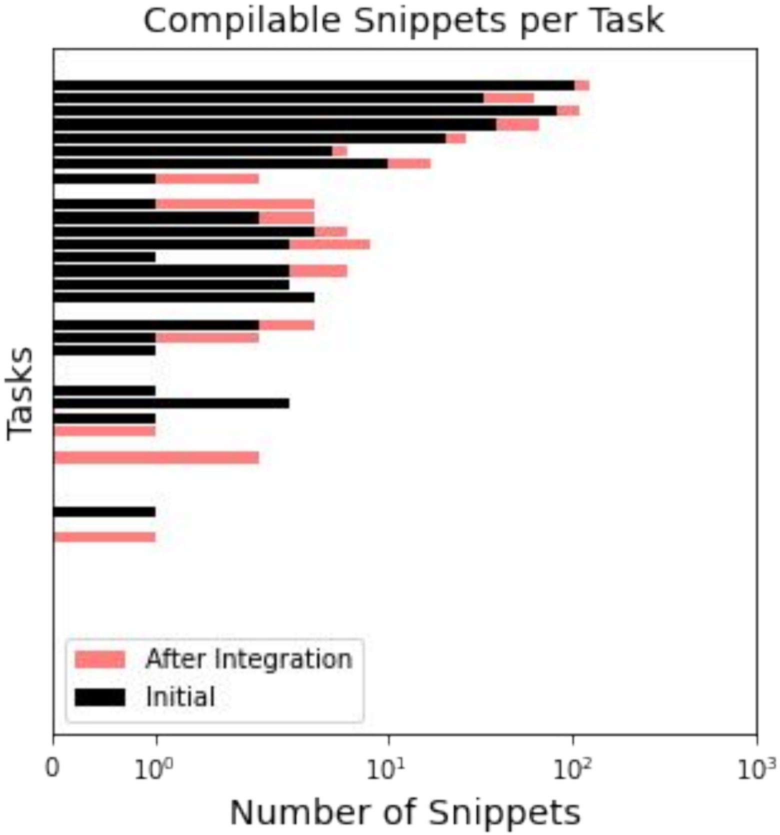


```
public class Main{
      public static void main(String[] args){

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

      }
}
```

```
public class Main{
      public static void main(String[] args){

          + int result = Integer.parseInt(args[0]);

      }
}
```

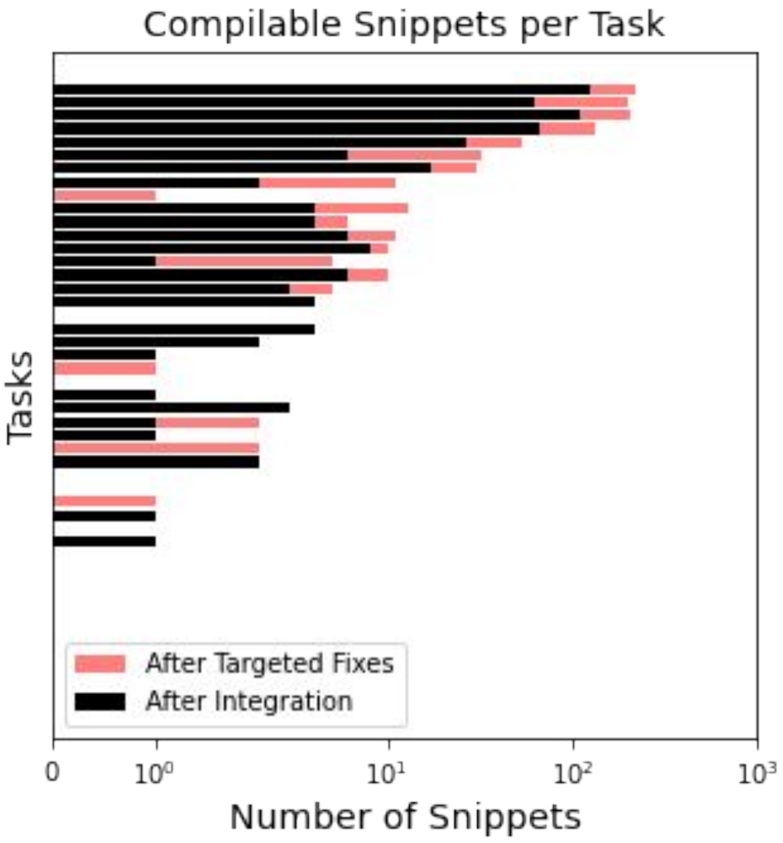
Integration

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



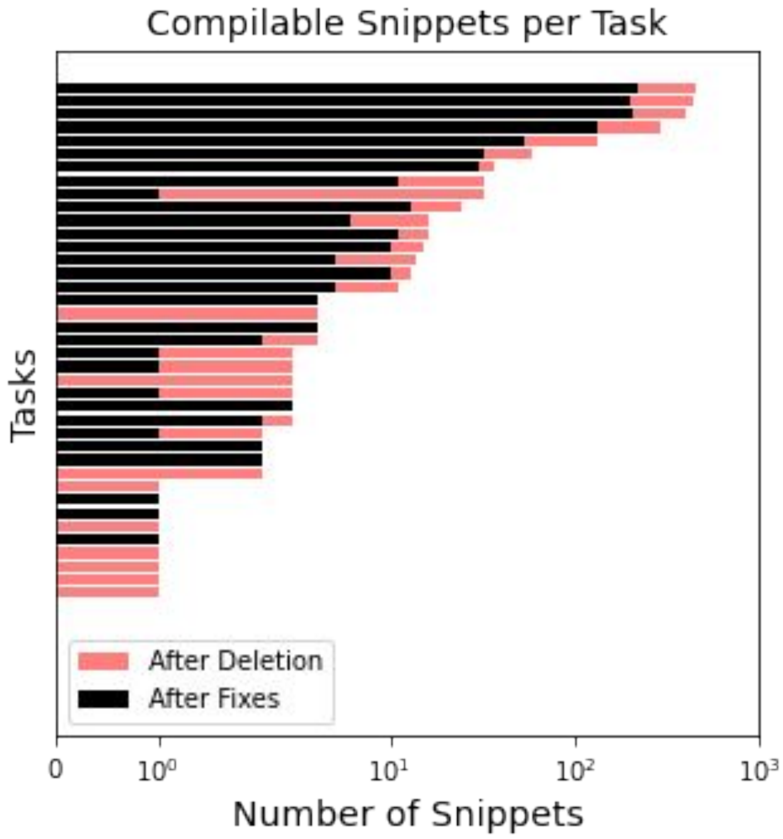
Targeted Fixes

- 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%

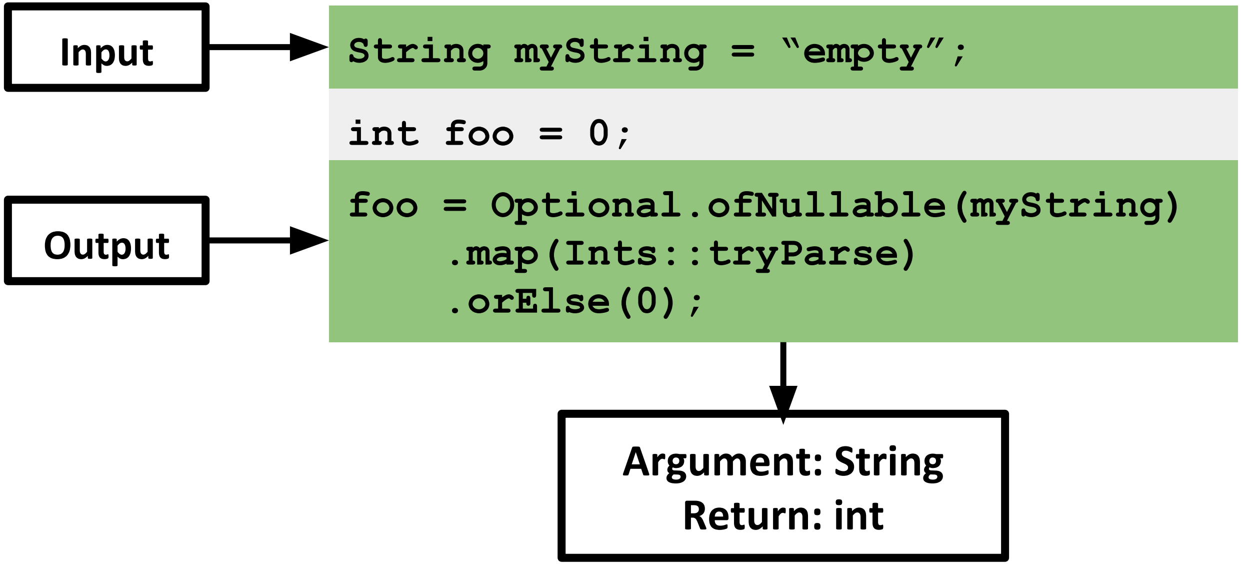


Automated Testing

- 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

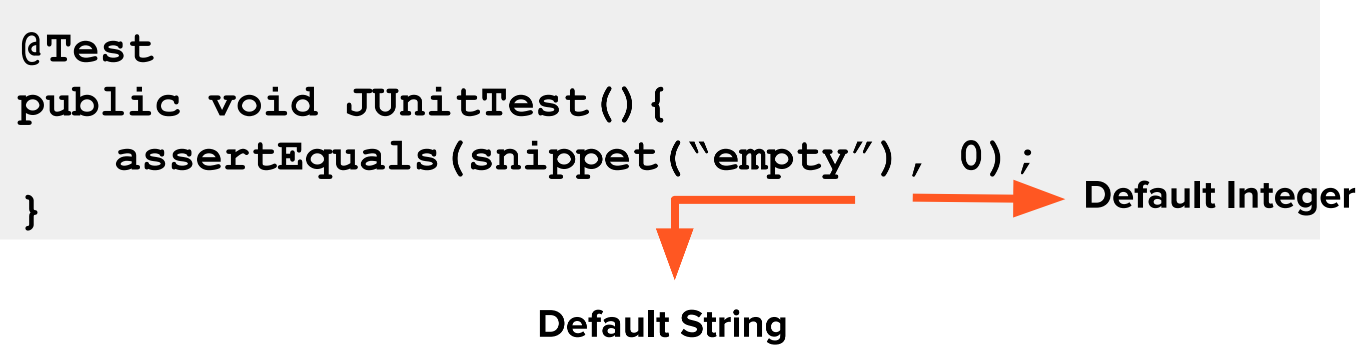
Output

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

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



```
@Test
public void JUnitTest(){
    assertEquals(snippet(“empty”), 0);
}
```

Default Integer

Default String

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

```
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