Analyzing Software using Deep Learning

Sequence-to-Sequence Networks and their Applications (Part 2)

Prof. Dr. Michael Pradel
Software Lab, University of Stuttgart
Summer 2020
Overview

- **Sequence-to-sequence networks**
- **API usage sequences for natural language queries**
  Based on "Deep API learning" by Gu et al., 2016
- **Interpreting Python programs**
  Based on "Learning to execute" by Zaremba and Sutskever, 2014
Motivation

**APIs are difficult to use**
- Which *methods* to call?
- In what *order* to call them?

Developers **ask questions**, e.g., on stackoverflow.com
- Human effort required to answer them

**Goal:** Automatically **suggest API usages** based on natural language query
Idea

Formulate the problem as a translation problem

- Input: Sequence of natural language words
- Output: Sequence of API method calls
- Train and query sequence-to-sequence neural network
Example

Natural language query:
”match regular expressions”

Sequence of API calls expected as (possible) answer:
Pattern.compile, Pattern.matcher, Matcher.group
Training Data

- Analyze 443,000 Java projects from GitHub
- Focus on JDK = APIs of Java standard library
- Extract pairs of annotation and call sequence
- About 7 million extracted pairs
- Use 10,000 for testing and others for training
Example

```java
/**
 * Copies bytes from a large (over 2GB) InputStream to an
 * OutputStream. This method uses the provided buffer, so
 * there is no need to use a BufferedInputStream.
 * @param input the InputStream to read from
 * ...*/

public static long copyLarge(final InputStream input,
final OutputStream output, final byte[] buffer)
    throws IOException {
    long count = 0;
    int n;
    while (EOF != (n = input.read(buffer))) {
        output.write(buffer, 0, n);
        count += n;
    }
    return count;
}
```
Example

```java
/**
 * Copies bytes from a large (over 2GB) InputStream to an OutputStream. This method uses the provided buffer, so
 * there is no need to use a BufferedInputStream.
 * @param input the InputStream to read from
 * @param output the OutputStream to write to
 * @param buffer the buffer to use
 * @return the number of bytes copied
 */
public static long copyLarge(final InputStream input, final OutputStream output, final byte[] buffer)
    throws IOException {
    long count = 0;
    int n;
    while (EOF != (n = input.read(buffer))) {
        output.write(buffer, 0, n);
        count += n;
    }
    return count;
}
```

Annotation:
"copies bytes from a large inputstream to an outputstream"

Call sequence:
InputStream.read, OutputStream.write
Extracting Annotations

- Extract **JavaDoc** of each method
- Extract **first sentence**
- Ignore methods without JavaDoc
- Ignore annotations with "irregular" comments, e.g., **TODO:: . . .**
Extracting Call Sequences

- Goal: Lightweight analysis that scales to millions of code files
- Static, AST-based analysis with type bindings
- Example:

```java
list.add(23);
```

```
ExpressionStatement
  MethodInvocation
    expression
      Simple-Name: list
      Simple-Name: add
      arguments
        expression
          Simple-Name: token
            identifier: 23
```
AST-based Extraction (1)

- **Constructor call:**
  \[
  \text{new } C() \rightarrow C.\text{new} \quad \text{(if } C \text{ is JDK class)}
  \]

- **Method call:**
  \[
  \text{obj.m()} \rightarrow C.m \quad \text{(if type of } \text{obj is JDK class)}
  \]

- **Call expressions as arguments:**
  \[
  \text{o1.m1(o2.m2())} \rightarrow C2.m2, C1.m1
  \]
AST-based Extraction (2)

- **Sequence of statements:**
  \[ o1.m1(); \ o2.m2(); \rightarrow C1.m1, C2.m2 \]

- **Conditionals:**
  \[
  \text{if}(o1.m1()) \{
  \hspace{1cm} o2.m2();
  \}
  \text{else} \{
  \hspace{1cm} o3.m3();
  \}
  \rightarrow C1.m1, C2.m2, C3.m3
  \]

- **Loops:**
  \[
  \text{while}(o1.m1()) \{ o2.m2(); \}
  \rightarrow C1.m1, C2.m2
  \]
Putting Everything Together

443k projects → Static analysis →

Annotation, call sequence pairs → Sequence of API calls

Encoder RNN → Context vector → Decoder RNN

Words in annotation → Developer

→ during training

→ API prediction
Examples

■ "generate md5 hash code"
  └ MessageDigest.getInstance,
      MessageDigest.update, MessageDigest.digest

■ "convert int to string"
  └ Integer.toString

■ "get files in folder"
  └ File.new, File.list, File.new, File.isDirectory