Program by Natural Language: Translation between Pseudocode and Code

Huan Sun

January 12, 2016

Overview

Natural language instruction <=> Python code

- Source code:

```
def fizzbuzz(n):
  if not isinstance(n, int):
    raise TypeError('n is not an integer')
  if n % 3 == 0:
    return 'fizzbuzz' if n % 5 == 0 else 'fizz'
  elif n % 5 == 0:
    return 'buzz'
  else:
    return str(n)
```

- Natural language instruction/annotation:

```
define the function fizzbuzz with an argument n.
  if n is not an integer value,
    throw a TypeError exception with a message ...
  if n is divisible by 3,
    return 'fizzbuzz' if n is divisible by 5, or 'fizz' if not.
  if not, and n is divisible by 5,
    return the string 'buzz'.
  otherwise,
    return the string representation of n.
```

Motivation: From pseudocode to code

- Less need to remember syntax and grammars
- Get a job in Google, as long as you know algorithms! ;)

" Define the function f with an argument n " => " def f(n): "
" Check if n is greater than 0 " => "if n>0: "

Why from PSEUDOCODE to code?
Directly generating code for a high-level task (e.g., the task in page 3) might be too hard (for now)!

Motivation: From code to pseudocode

Or, automatic code annotation

- For easier code understanding
- To facilitate retrieving relevant code snippets

This is a per-line annotation strategy, rather than a high-level one-sentence description.

Why?
Letting the algorithm directly give a general and precise description of a long code snippet might be too hard (for now)!

The State-of-the-Art

- From code to pseudocode:
  ASE'15: using Statistical Machine Translation
  Tool and dataset: http://ahclab.naist.jp/pseudogen/
  Example in page 3: code translated to pseudocode

- From pseudocode to code:
  OOPSLA'15: Synthesizing Java Expressions from Free-form Queries
  Related, but they focus on API-related query, rather than a general instruction as in page 3,
  e.g., "copy file A to B" => FileUtils.copyFile (new File (A), new File (B))

Datasets and Tentative Models

We need <code, pseudocode> pairs:

- ASE'15: using Statistical Machine Translation
  Tool and dataset: http://ahclab.naist.jp/pseudogen/

  Use this tool to generate pseudocode for more code snippets, and form (noisy) pairs

- Wikipedia algorithms, Python tutorial, Rosetta code

A bi-directional modeling approach to simultaneously deal with the two tasks: pseudocode <=> code

Doubts

- How specific should the pseudocode be?
  Balance between understandability and precision/specifity?

- Incorporate code snippets retrieval (in a real IDE tool)?
  For some algorithms, e.g., sorting, available code snippets exist.

The End