A Survey of Methods to Convert Code to Natural Language

Graham Neubig
Nara Institute of Science and Technology (NAIST)
2015-10-27

With many thanks to Dawn Lawrie and Hideaki Hata

The Problem

Summary

```
# func(t) returns the sum of the squares of all natural
# numbers less than t
```

```
def func(t):                          # Define function func with variable t
  my_list = range(1,t)                # Create a list my_list of all natural
                                      # numbers less than t
  my_val = 0                          # Set my_val equal to zero
  for x in my_list:                   # For every value x in my_list
    my_val += x * x                   # Add the square of x to my_val
  return my_val                       # Return my_val
```

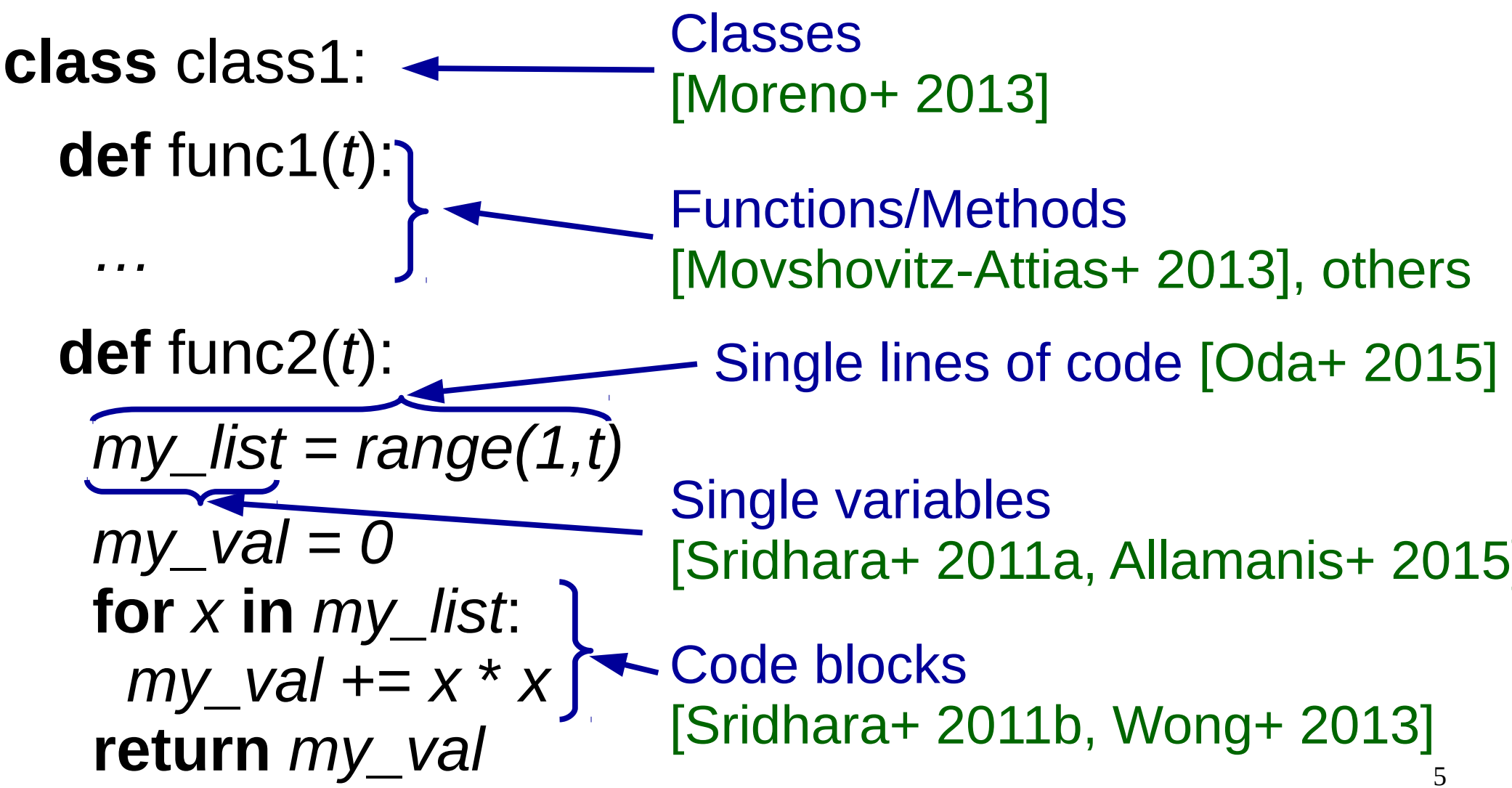
Pseudo-code

Outline

- What do we describe?
- How do we generate?
- How do we choose salient content for summaries?
- How do we evaluate?
- How do we create training data?

What Do We Generate?

What Do We Describe?



Content-Specialized Generation

- Code changes [Buse+ 2010, Cortes-Coy+ 2014]
- Exceptions [Buse+, 2008]
- Test cases [Zhang+ 2011, Kamimura+ 2013]
- Code snippets [Allamanis+ 2014]

How Do We Generate?

Rule-based Methods

- Manual rules to convert code into natural language (e.g. getLastWindow → get the last window) [Hill+ 2009, Sridhara+ 2010]
- Identify multi-word block [Sridhara+ 2011] or class [Moreno+ 2013] “stereotypes” and use templates
- Use other info such as usage context [McBurney+ 2014] or execution paths [Buse+ 2008, Zhang+ 2011]

Content Retrieval/Selection Methods

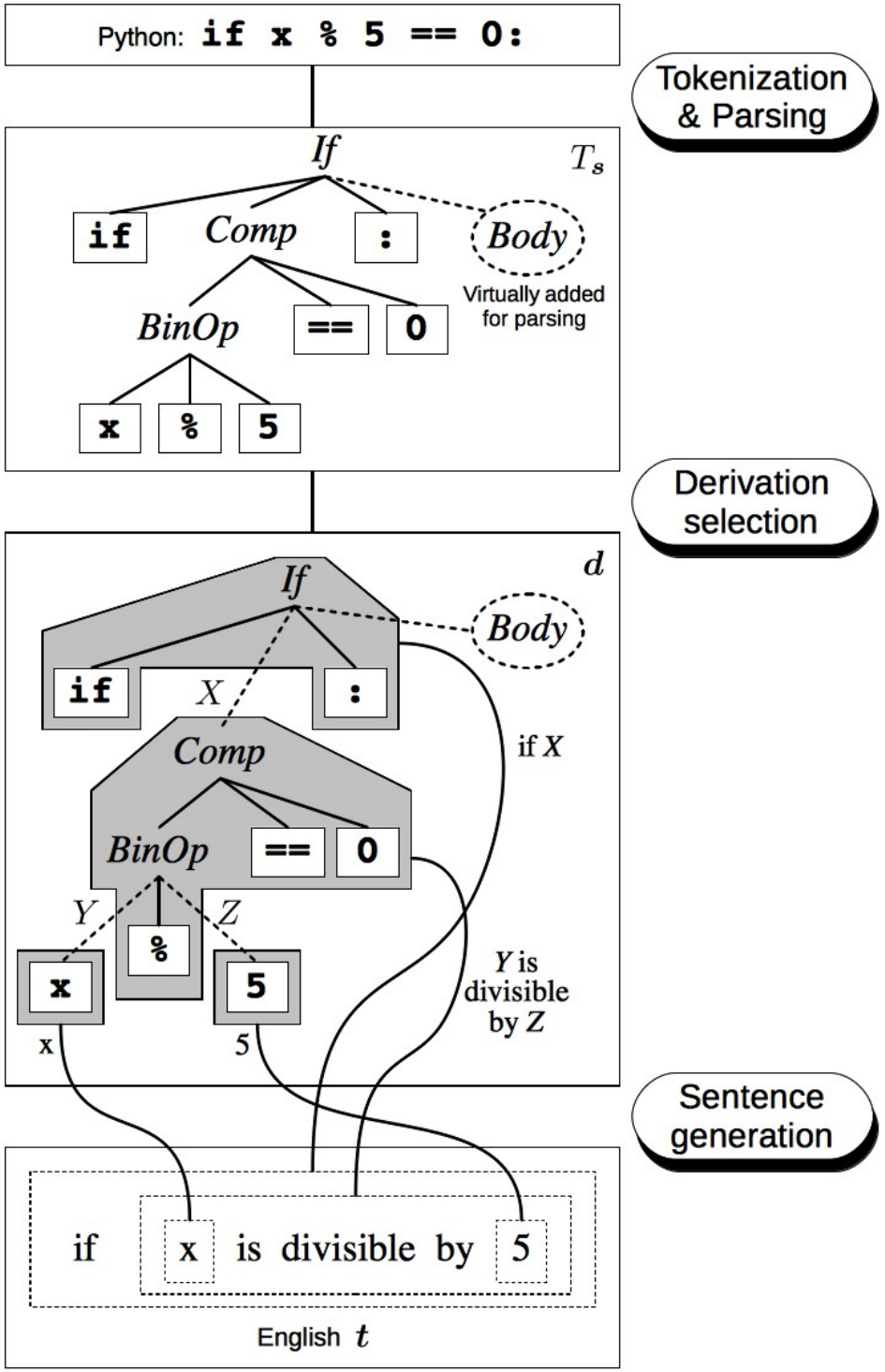
- Keyword lists: from the comments/identifiers in code, extract salient keywords using LSA, topic models, etc. [Haiduc+ 2010, De Lucia+ 2012, McBurney+ 2014]
- Comment retrieval: Create code snippet/comment pairs, and retrieve the most relevant one using edit distance [Wong+ 2013] or probabilistic models [Allamanis+ 2015]

Word-by Word Predictive Models

- Create a predictive model, and use it to predict the next word one-by-one
- Use n-gram or topic models to predict comments [Movshovitz-Attias+ 2013]
- Use bilinear language models to predict sub-tokens of identifier names [Allamanis+ 2015]

Statistical Machine Translation

- Convert from code syntax to natural language using tree-based machine translation [Oda+ 2015]



How Do We Choose Salient Content?

Summarization Criteria

- Keyword selection using LSA [Haiduc+ 2010] or topic models [McBurney+ 2014]
- Classes affected by content [Cortes-Coy+ 2014]
- Invocation frequency [Kamimura+ 2013]
- Centrality in a code ontology [Rastkar+ 2011]
- Eye tracking [Rodeghero, 2014]

How Do We Evaluate?

Intrinsic Evaluation

- Accuracy/Adequacy: Content is correct, and appropriately reflect the content of the corresponding code [Sridhara + 2010, Oda+ 2015]
- Conciseness: Whether the comments avoid saying anything that is not necessary [Sridhara+ 2010]
- Preference: Whether one type of summary is preferred over the other [Cortes-Coy+ 2014] or better than language already existing in the code [Buse+ 2008]
- Commitability: Whether comments can be committed, or actually are committed [Wong+ 2015]
- Automatic Measures: Retrieval accuracy [Haiduc+ 2010], similarity to reference using BLEU [Oda+ 2015], etc.

Extrinsic Evaluation

- Comprehension: Automatically generated natural language can help code understanding [McBurney+ 14, Oda+ 15]
- Reading Speed: Reading time is actually longer when using auto-generated pseudo code [Oda+ 15]
- Reduces Comment Typing Time: Reduction in the number of characters typed when writing comments [Movshovitz-Attias+ 2013]

How Do We Create Training Data?

Training Data Creation

- None Needed: Many methods use manual heuristics
- Manual Creation: It is possible to create data for training manually, such as 18,000 annotated lines of Python code [Oda+ 2015]
- Communications: Mine data from developer mailing lists [Panichella+ 2012], or now Stack Overflow [Wong+ 2013]
- Comments: Comments corresponding to specific methods [Movshovitz-Attias+ 2013] or blocks of code [Wang 2015, Wong+ 2015]

Thank You