Leveraging a Corpus of Natural Language Descriptions for Program Similarity

Meital Zilberstein & Eran Yahav
Technion – Israel Institute of Technology
Onward! 2016
Lots of snippets out there

>19M users
>38M repositories

>5.9M registered users
>12M questions
>19M answers

And also.. Google code, programming blogs, documentation sites, requirements documents, comments, identifier, commits, etc.

Sep ‘16
Similarity: Images VS. Programs

- Code is not organized
- Cannot accomplish even simple tasks (which are increasingly improving in other domains)
Similarity: Images VS. Programs

- Images already have some solutions
- Find somewhere on the web

Lago di Canzolino, Italy
Similarity: Images VS. Programs

With code we still don’t know what to do
Why are Programs Hard?

- A program is a data transformer
- “infinite data” $\gg$ “big data”
  - Potentially infinite number of runtime behaviors
  - Depends on inputs

```python
from subprocess import call
cmd_to_run = raw_input()
call(cmd_to_run.split())
```

Infinite code
Why are Programs Hard?

- Print the exact same value
- Both written in Java
- Syntactic difference

```
int scale = 100000;
double x = (double)Math.round(8.912384 * scale) / scale;
System.out.println(x);

DecimalFormat df = new DecimalFormat("#0.00000");
System.out.println(df.format(8.912384));
```
Syntactic Similarity is not Sufficient

- Two approaches for similarity
- Textual diff
- There's more than one way to do it -Perl slogan
Syntactic Similarity is not Sufficient

```python
import os
if os.path.exist(filename):
    print(exist)
else:
    print(no such file)
```

```python
try:
    fh = open(f)
    print "exist"
except:
    print "no such file"
```
Syntactic Similarity is not Sufficient

- Textual diff
- Abstract Syntax Tree diff

```python
from itertools import permutations
permutations(["a", "b"])

from subprocess import call
call(["ls", "-l"])
```
Cross Language Similarity

Generation of all possible permutations of a string

- Different algorithms
- Similar functionality

```python
def p(head, tail=''):  
    if len(head) == 0:
        print tail
    else:
        for i in range(len(head)):
            p(head[0:i] + head[i+1:],
               tail + head[i])
```

```c
void permute(const char *s, char *out,  C
int *used, int len, int lev){
    if (len == lev) {
        out[lev] = '\0';
        puts(out);
        return;
    }
    int i;
    for (i = 0; i < len; ++i) {
        if (used[i])
            continue;
        used[i] = 1;
        out[lev] = s[i];
        permute(s,out,used,len,lev+1);
        used[i] = 0;
    }
    return;
}
```
Our approach (simplified)

LEVERAGING A CORPUS OF NATURAL LANGUAGE DESCRIPTIONS FOR PROGRAM SIMILARITY - MEITAL ZILBERSTEIN & ERAN YAHAV
Semantic Relatedness

- First appeared in the NLP domain
  - finer case of Semantic Similarity (is-a)
  - Can be established across different parts of speech
- Based on functionality
- Quantitative similarity
- Semantic relatedness
  - Inclusion, Reversal

```java
import random
print random.randint(min, max)

public static int getRandom(int min, int max){
    Random rn = new Random();
    int range = max - min + 1;
    return rn.nextInt(range) + min;
}
```

Equivalent? NO!
Code Similarity Applications

- Code similarity is a central challenge in many programming related applications, such as:
  - Semantic Code Search
  - Automatic Translation
  - Education

```java
Date d1 = new Date();
Date d2 = new Date();
d2.setTime(d1.getTime() + 1*24*60*60*1000);
```

```php
define(DATETIME_FORMAT, 'y-m-d H:i');
$time = date(DATETIME_FORMAT, strtotime("+1 day", $time));
```

PHP though..
Automatic Tagging of Snippets

- Predict a set of textual labels
- Semantics of the code fragment
- **Long-term goal:** produce natural-language summaries for code snippets

```java
int foo = Integer.parseInt ( "1234" );
```

string  int  converting
Overview

Fragments with mapping to textual description (in database)

Textual description in database

sim(p1, p2)
Leveraging Collective Knowledge

- Stackoverflow
- Community question-answering site
- Programming related questions
- Each question is associated with a title, content and tags
- Implicit mapping between code fragments and their descriptions
How do I sort a list of strings in Python?

What is the best way of creating an alphabetically sorted list in Python?

protected by Community • Jun 12 at 8:34

This question is protected to prevent "thanks!", "me too!", or spam answers by new users. To answer it, you must have earned at least 10 reputation on this site.

1 Use `locale` and its string collation methods to sort naturally according to current locale — u0b34a0f5ae
Sep 8 '09 at 18:21

Basic answer:

```
mylist = ['b', 'c', 'A']
mylist.sort()
```

This modifies your original list (i.e. sorts in-place). To get a sorted copy of the list, without changing the original, use the `sorted()` function:

```
for x in sorted(mylist):
    print x
```

However, the examples above are a bit naïve, because they don't take locale into account, and perform a case-sensitive sorting. You can take advantage of the optional parameters, `key` to specify...
Know your limits!

- This work presents a radical departure from common approaches
- Challenge: find representatives in the pre-computed database
- The results are biased by the quality of the database
- We show that this approach is feasible for snippets that serve a common purpose
The Importance of Data

% Matches

Stackoverflow grows in More than 15K answers a day!

log₂(DB Size)
Data Coverage

“Although the number of legal statements in the language is theoretically infinite, the number of practically useful statements is much smaller, and potentially finite.”

-- Study of the uniqueness of source Code, Gabel et al.

- Software is usually an aggregation of much smaller parts
- Code is repetitive and predictable
- Syntactic similarity
Going Back to our Example

<table>
<thead>
<tr>
<th>Title</th>
<th>How to generate all permutations of a list in Python?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>How do you generate all the permutations of a list in Python, independently of the type of elements in that list? For example: <code>&lt;some code&gt;</code></td>
</tr>
<tr>
<td>Tags</td>
<td>python, algorithm, permutation, combinatorics, python-2.5</td>
</tr>
<tr>
<td>Votes</td>
<td>171</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>Generating all permutations of a given string.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>What is an elegant way to find all the permutations of a string. E.g. <code>ba</code>, would be <code>ba</code> and <code>ab</code>, but what about <code>abcdefg</code>? Is there any example</td>
</tr>
<tr>
<td>Tags</td>
<td>algorithm</td>
</tr>
<tr>
<td>Votes</td>
<td>124</td>
</tr>
</tbody>
</table>
Text Similarity

- Python code partial description:
  - “How to generate all permutations of a list in Python?”

- C code partial description:
  - “Generating list of all possible permutations of a string”

- Similarity score ≈ 0.8
Text Processing

Removing stop-words & punctuation

Lemmatization

Vector Space Model

LEVERAGING A CORPUS OF NATURAL LANGUAGE DESCRIPTIONS FOR PROGRAM SIMILARITY - MEITAL ZILBERSTEIN & ERAN YAHAV
Models – tf.idf

\[ tf.idf_{t,d} = tf_{t,d} \cdot idf_t \]

- Term Frequency Inverse Document Frequency

- Each cell term is:
  - Higher when the term occurs many times
  - Lower when the term occurs in many documents

<table>
<thead>
<tr>
<th>Doc 1</th>
<th>Doc 2</th>
<th>Wanted document</th>
</tr>
</thead>
<tbody>
<tr>
<td>term</td>
<td>count</td>
<td>term</td>
</tr>
<tr>
<td>list</td>
<td>1</td>
<td>sort</td>
</tr>
<tr>
<td>permutation</td>
<td>1</td>
<td>list</td>
</tr>
<tr>
<td>generate</td>
<td>2</td>
<td>permutation</td>
</tr>
<tr>
<td>string</td>
<td>1</td>
<td>string</td>
</tr>
</tbody>
</table>

Train set

\( \times \) Smoothing

LEVERAGING A CORPUS OF NATURAL LANGUAGE DESCRIPTIONS FOR PROGRAM SIMILARITY - MEITAL ZILBERSTEIN & ERAN YAHAV
Models – Latent Semantic Analysis

“There is some underlying latent semantic structure in the data that is obscured by the randomness of word choice.”

[Deerwester et al.]

- Words that are used in the same contexts tend to have similar meanings
- Mapping words and documents into a “concept” space
- Finding the underlying meaning
  - Synonyms

Create string ≈ Generate text
Models – Latent Semantic Analysis

- Singular Value Decomposition (SVD)
- Finding a reduced dimensional representation that emphasizes the strongest relationships
- Compute similarities between entities in the semantic space

For titles we use ADW with query expansion
Vectors Similarity

- **Cosine Similarity**
- Normalizes the vectors to unit length
- Prevent bias originating from different text sizes

\[ \text{cosine}(v_1, v_2) = \frac{0 \cdot 0.2 + 0 \cdot 0 + 0.3 \cdot 0.8 + 0.9 \cdot 2 + 0 \cdot 0}{\sqrt{0.3^2 + 0.9^2} \cdot \sqrt{0.2^2 + 0.8^2 + 2^2}} = 0.21 \]
static string ByteToHex(byte[] bytes) {
    char[] c = new char[bytes.Length * 2];
    int b;
    for (int i = 0; i < bytes.Length; i++) {
        b = bytes[i] >> 4;
        c[i * 2] = (char) (55 + b + (((b - 10) >> 31) & -7));
        b = bytes[i] & 0xF;
        c[i * 2 + 1] = (char) (55 + b + (((b - 10) >> 31) & -7));
    }
    return new string(c);
}

import javax.xml.bind.annotation.adapters.HexBinaryAdapter;
public byte[] hexToBytes(String hStr) {
    HexBinaryAdapter adapter = new HexBinaryAdapter();
    byte[] bytes = adapter.unmarshal(hStr);
    return bytes;
}
Snippets Analysis Challenges

- A code snippet
- Might not be compilable (in static languages)
- Might lack important information
- Not a full program
- Inputs and outputs might be implicit
- Different programming languages
import urllib2
res = urllib2.urlopen('http://www.example.com/)
html = res.read()
Recap

Fragments with mapping to textual description (in database)

Textual description in database

Feature Extraction & Comparison

sim(p1, p2)
Syntactic Similarity

- Need: Search a code within a massive database
  - Contains more than 1M code fragments
  - Many programming languages
- Restriction: the output needs to be syntactically similar
  - Same flow, same order of function calls, etc.
- Solution 1: keyword matching followed by alignment of the common tokens
  - Global pairwise sequence alignment
  - Generic, works for any PL
Syntactic Similarity - Solution 2

- Language specific
- Based on AST structure
- Compare only important data
  - No identifiers or concrete values

```python
import urllib2
res = urllib2.urlopen('http://www.example.com/')
html = res.read()
```
Labeling System

- For our evaluation we created a large corpus of program pairs, tagged by similarity level
- Determine the similarity between a vast group of pairs
  - This task requires human input
- Contains 6500 labeled pairs
- Crowd-source web application like2drops

www.like2drops.com
Help me to decide whether two code snippets are similar

Language: java

r = new util.Random();
long.toString (r.nextLong (), 36);

Language: javascript

Math.random().toString(36).substring();

I don’t know  Totally different  Pretty different  Related, but not similar  Pretty similar  Very similar

Current helping status: 14 pairs!

The similarity level is dependent on the functionality of the code, two code snippets are considered similar if they solve the same problem (even if the parameter values are different). Moreover, printing and logging shouldn’t be taken into account.
Program Pairs Corpus

- Based on more than 10,000 user tags
  - > 40 users!
- The possible tags are: Very Similar, Pretty Similar, Related but not Similar, Pretty Different and Totally Different
- Trust test
- Majority
- In some cases, the answers varied greatly
  - around 6%
  - no conclusive decision is possible
  - omitted these pairs from our experiment
Evaluation - Similarity Classifier

- Pairs are assigned a quantitative score from 1 to 5
- Our output is quantitative \([0, 1]\)
- We saw that the overall direction of different users is often the same
  - e.g., similar or not
- However, the specific tags are not
  - e.g., very similar and pretty similar
Results

- 4,000 program pairs
- The results show that 87.3% of our labels are consistent with the users’ labels
- Precision is 86.2%
- Recall is 85%
- AUC is 0.9391
- Can we do better?
Results, Different Configurations

<table>
<thead>
<tr>
<th>#</th>
<th>Configuration</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Full approach</td>
<td>86.2%</td>
<td>85%</td>
<td>87.3%</td>
<td>0.9391</td>
</tr>
<tr>
<td>2</td>
<td>Full, without type signatures</td>
<td>85.8%</td>
<td>85%</td>
<td>87.1%</td>
<td>0.9386</td>
</tr>
<tr>
<td>3</td>
<td>Full, without software specialized vocabulary</td>
<td>85.1%</td>
<td>83.7%</td>
<td>86.3%</td>
<td>0.9358</td>
</tr>
<tr>
<td>4</td>
<td>Full, without ADW and software specialized vocabulary</td>
<td>83.3%</td>
<td>83.1%</td>
<td>85.1%</td>
<td>0.9335</td>
</tr>
<tr>
<td>5</td>
<td>Full, without LSA</td>
<td>83.6%</td>
<td>82.4%</td>
<td>85.1%</td>
<td>0.9343</td>
</tr>
<tr>
<td>6</td>
<td>Only titles (ADW and software specialized vocabulary)</td>
<td>83.8%</td>
<td>73.7%</td>
<td>82%</td>
<td>0.9011</td>
</tr>
<tr>
<td>7</td>
<td>Only content (LSA)</td>
<td>81.1%</td>
<td>73.9%</td>
<td>80.8%</td>
<td>0.8726</td>
</tr>
<tr>
<td>8</td>
<td>Tokenized-code, TF.IDF [11]</td>
<td>77.3%</td>
<td>76.5%</td>
<td>79.6%</td>
<td>0.8342</td>
</tr>
<tr>
<td>9</td>
<td>Tokenized-code, LSA</td>
<td>79.8%</td>
<td>69.6%</td>
<td>78.7%</td>
<td>0.825</td>
</tr>
<tr>
<td>10</td>
<td>Random</td>
<td>45.6%</td>
<td>47.4%</td>
<td>50.7%</td>
<td>0.4993</td>
</tr>
</tbody>
</table>

- **McNemar’s test**: Significant difference
- The goal of our approach is to handle the case in which the given code snippets are not syntactically similar.
Similarity is not Conclusive

Similar? Not?

```java
int x = Integer.parseInt("8");
char c = '1';
int i = c - '0';
// i is now equal to 1, not '1'
```
Examples

```
HashSet hs = new HashSet();
hs.addAll(al);
al.clear();
al.addAll(hs);
```

```
def f7(seq):
    seen = set()
    seen_add = seen.add
    return [x for x in seq if x not in seen and not seen_add(x)]
```

```
from itertools import groupby
[ key for key, _ in groupby(sortedList)]
```

```
List<Type> liIDs =
liIDs.Distinct().ToList<Type>();
```
```python
def Rand7():
    while True:
        x = (Rand5() - 1) * 5 + (Rand5() - 1)
        if x < 21: return x/3 + 1
```

```python
int rand7 (void){
    return 4;} // this number has been calculated using // rand5() and is in the range 1..7
```

Examples
Conclusion

- Measuring semantic relatedness between code fragments based on their corresponding textual descriptions and their type graphs
- We used the crowd to collect labeled data, which may be of interest by itself
- We combined an open world approach, text similarity techniques, and lightweight type analysis, and showed that it leads to promising results

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. [615688]