Code Similarity via Natural Language Descriptions

Meital Ben Sinai & Eran Yahav

Technion - Israel Institute of Technology

Off the Beaten Track, Jan 2015

Lots of snippets out there

>7M users
>17M repositories

3M registered users
>8M questions
>14M answers

Dec '14

Google code, programming blogs, documentation sites…

Similarity: Images VS. Programs

- The 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

The Grand Canal, Venice, Italy

Similarity: Images VS. Programs

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

Google image search

≈

The Grand Canal, Venice, Italy

Similarity: Images VS. Programs

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

Google image search

The Grand Canal, Venice, Italy

Similarity: Images VS. Programs

- With code we still don't know what to do

```
Program P
```

≈

Why are Programs Hard?

- A program is a **data transformer**
- "infinite data" ≫ "big data"
  - Potentially **infinite** number of **runtime behaviors**
  - Depends on inputs

```
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

- Textual diff

There's more than one way to do it

-Perl slogan

Syntactic Similarity is not Sufficient

- Textual diff

```
try:
    fh = open(f)
    print “exist”
except:
    print “no such file”
```

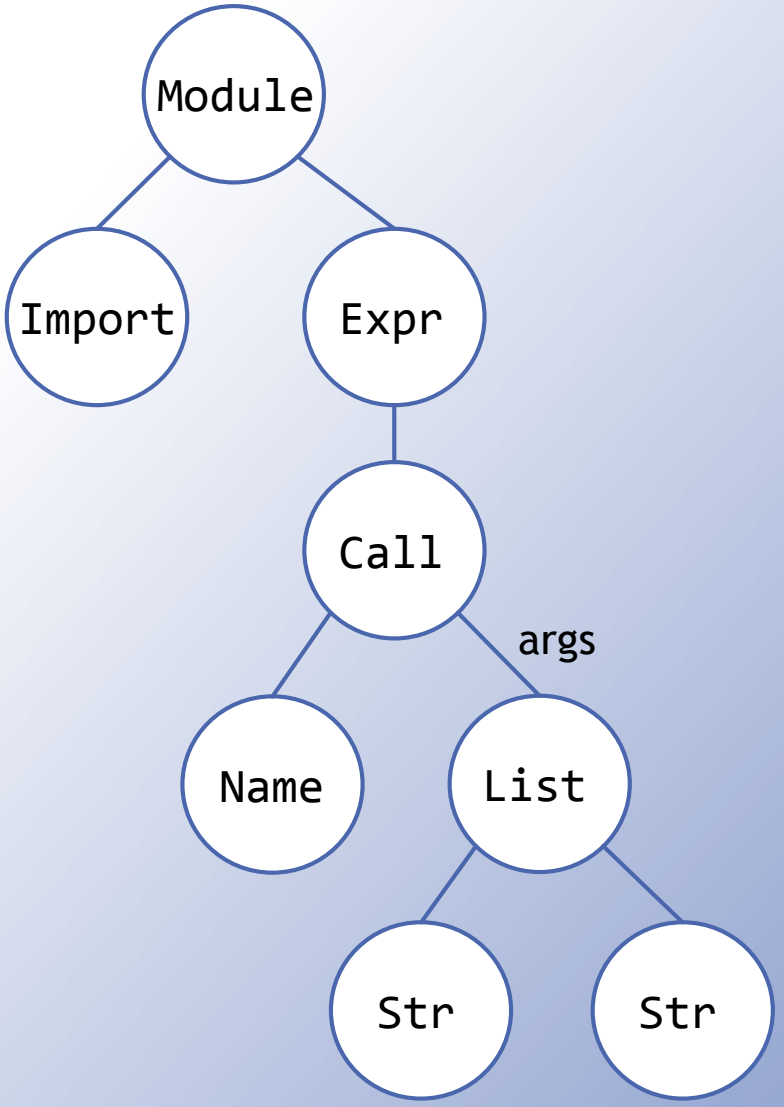
```
import os
if os.path.exist(filename):
    print(exist)
else:
    print(no such file)
```

Syntactic Similarity is not Sufficient

- Textual diff
- Abstract Syntax Tree diff

```
from itertools import permutations
permutations([“a”, “b”])
```

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



The Cross Language Challenge

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])
```

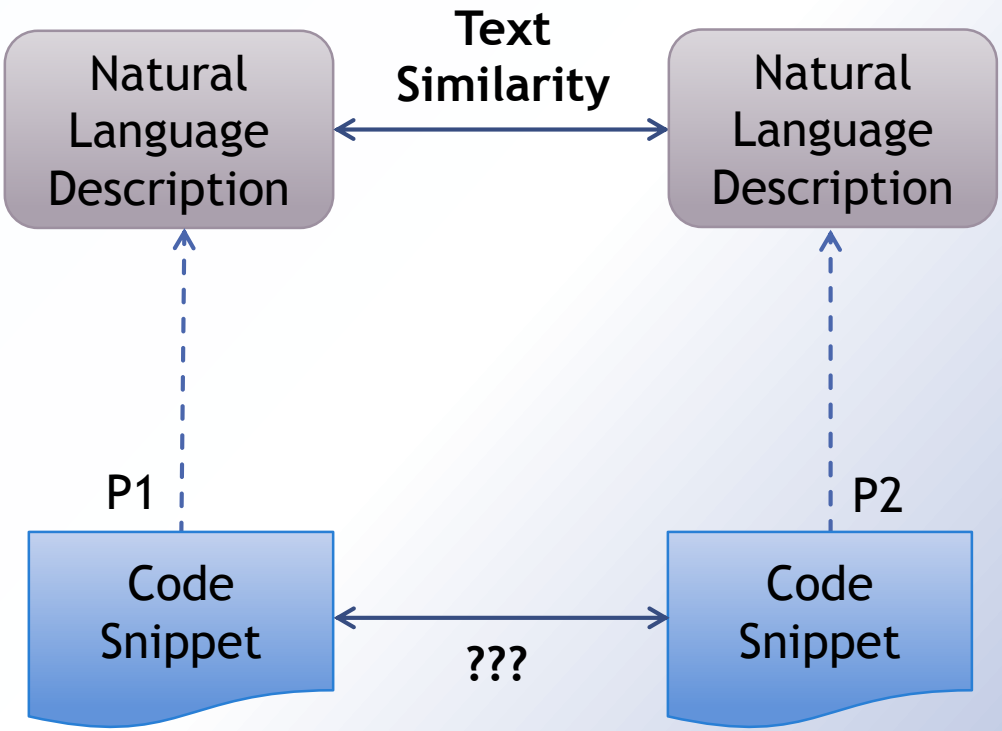
?

≈

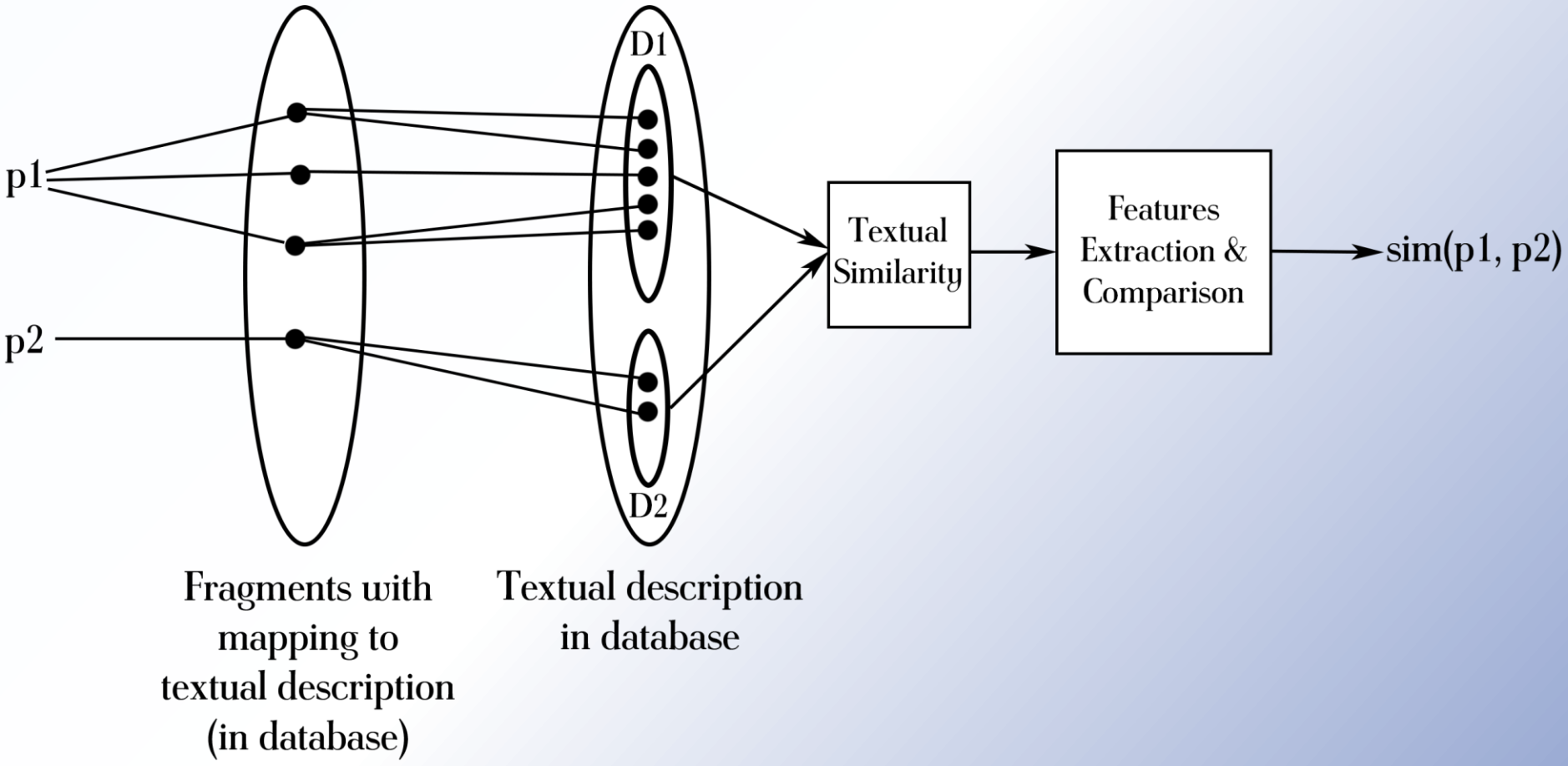
C

```
void permute(const char *s, char *out,
             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



Overview


p1
p2
D1
D2
Textual Similarity
Features Extraction & Comparison
sim(p1, p2)
Fragments with mapping to textual description (in database)
Textual description in database

Equivalence, Similarity, Relatedness..

```
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!

- Semantics
  - Functionality
- Quantitative similarity
- Semantic relatedness
  - Inclusion, Reversal, Closeness

Similarity Applications

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

duolingo

PHP though..

I know how to get tomorrow's data in JAVA, it's easy!

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

Similarity Applications

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

PHP though..

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

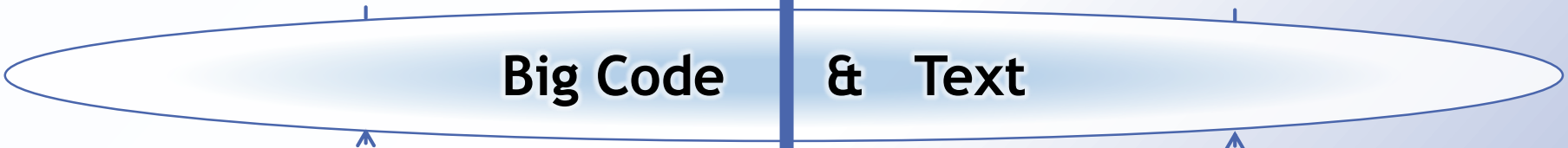
Related work

- PEPM'15 – **Source Code Examples from Unstructured Knowledge Sources** [Vinayakaro, Purandare, Nori]
- Onward'14 – **Approach based on mapping language structure** [Karaivanov, Raychev, Vechev]

Go Back to our Example

"How to generate all permutations of a list in Python"

"Generating list of all possible permutations of a string in c?"

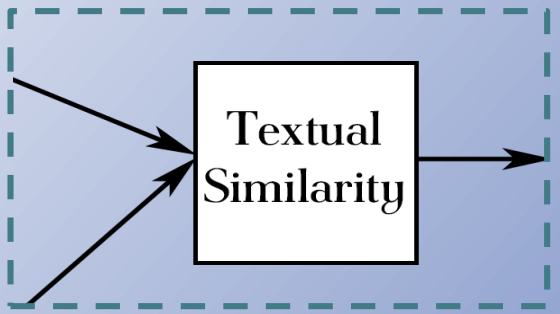



```python
def p (head, tail=''):
    if len(head) == 0:
        print tail
    else:
        for i in range(len(head)):
            p(head[0:i] + head[i+1:],
                        tail + head[i])
```

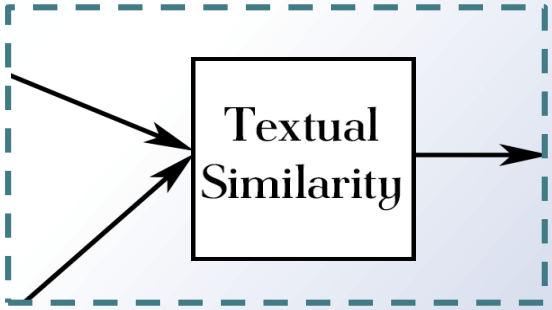
```c
void permute(const char *s, char *out,
             int *used, int len, int lev){
    if (len == lev) {
    out[lev] = '\0';
    puts(out);
    return;
    }
    int i;
    for (i = 0; i < len; ++i) {
        if (used[i])
            continue;
    used[i] = 1;
    out[lev] = s[i];
    permute(s,out,used,len,lev+1);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

The Text Similarity Magic

- 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 in c?"*
- Similarity score = **0.72**



Text Processing



generating	list	of	all	possible	permutations	of	a	string	in	c	?

Removing stop-words & punctuation

generating	list			possible	permutations			string			

Lemmatization

generate	list	possible	permutation	string

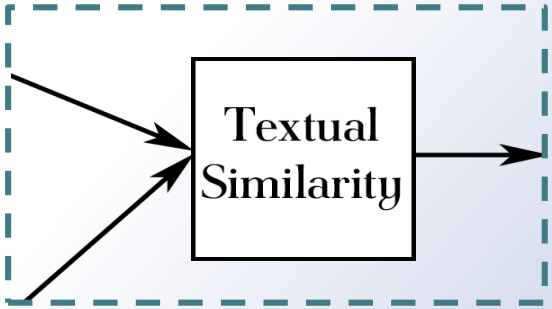
1M docs

Trained model

Vector Space Model

w(1)	w(2)	w(3)	...	w(n-1)	w(n)

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

Doc 1

term	count
list	1
permutation	1
generate	2
string	1

Doc 2

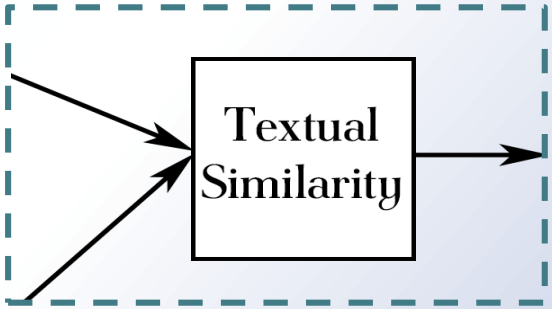
term	count
sort	3
list	1
string	1

Train set

term	idf
list	0
string	0
permutation	~0.3
generate	~0.3
sort	~0.3

Smoothing

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

0	0	0.3	0.9	0
list	string	generate	permutation	sort

=

term	idf
list	0
string	0
permutation	~0.3
generate	~0.3
sort	~0.3

×

Wanted document

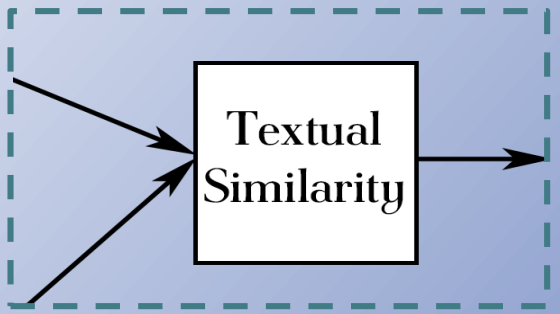
term	count
list	2
string	1
generate	1
set	1
permutation	3

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.]

Create string ≈ Generate text

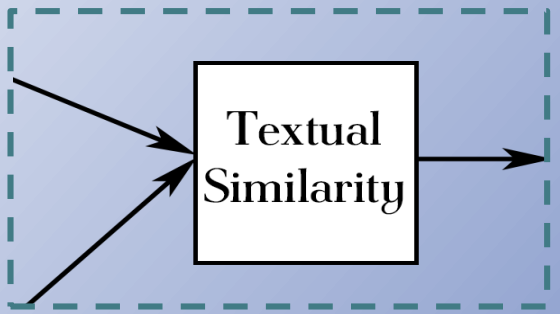
- Words that are used in the same contexts tend to have similar meanings
- Mapping words and documents into a "concept" space
- Finding the underlying meaning
  - Synonyms



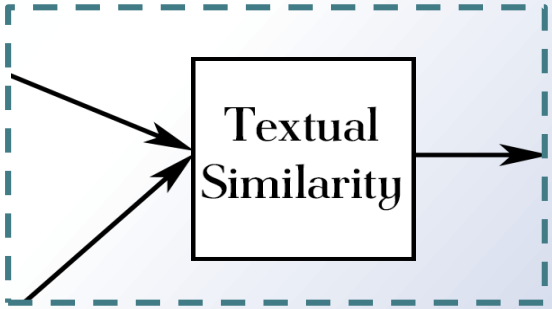
Models – Latent Semantic Analysis

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

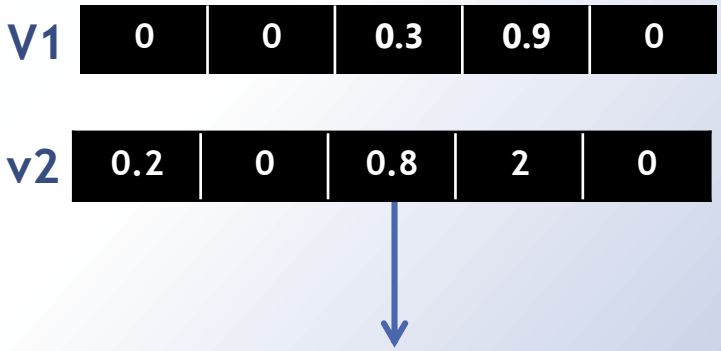
$$tf.idf(sort, order) = 0$$
$$LSA(sort, order) \sim 0.5$$



Vectors Similarity



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



$$cosine(v1, v2) = \frac{0 \cdot 0.2 + 0 \cdot 0 + 0.3 \cdot 0.8 + 0.9 \cdot 2 + 0 \cdot 0}{\sqrt{0.3^2 + 0.9^2} \cdot \sqrt{0.2^2 + 0.8^2 + 2^2}} = 0.21$$

Why Text is not Enough?

How do you convert byte array to hex String

```
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);
}
```

byte[] → String

Convert a string representation of a hex to a byte array

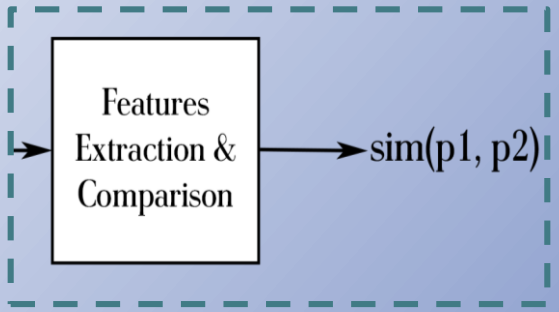
```
import javax.xml.bind.annotation.
           adapters.HexBinaryAdapter;

public byte[] hexToBytes(String hStr){
    HexBinaryAdapter adapter =
           new HexBinaryAdapter();
    byte[] bytes =
          adapter.unmarshal(hStr);
    return bytes;
}
```

String→ byte[]

Snippets Analysis Challenges

A code snippet

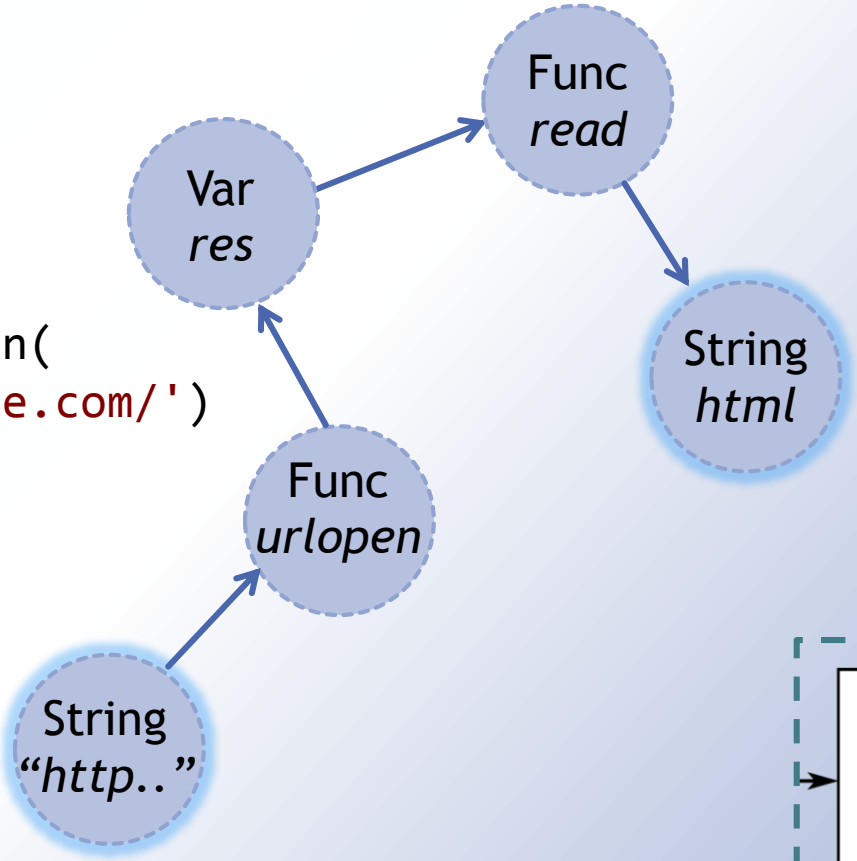
- Might not be compilable (in static languages)
- Might lack important information
- Not a full program
- **Inputs and outputs** might be implicit
- Different programming languages



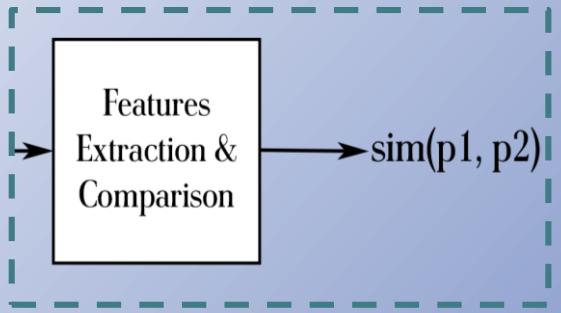
Snippets Type analysis

Code ⟶ Graph ⟶ Types signature

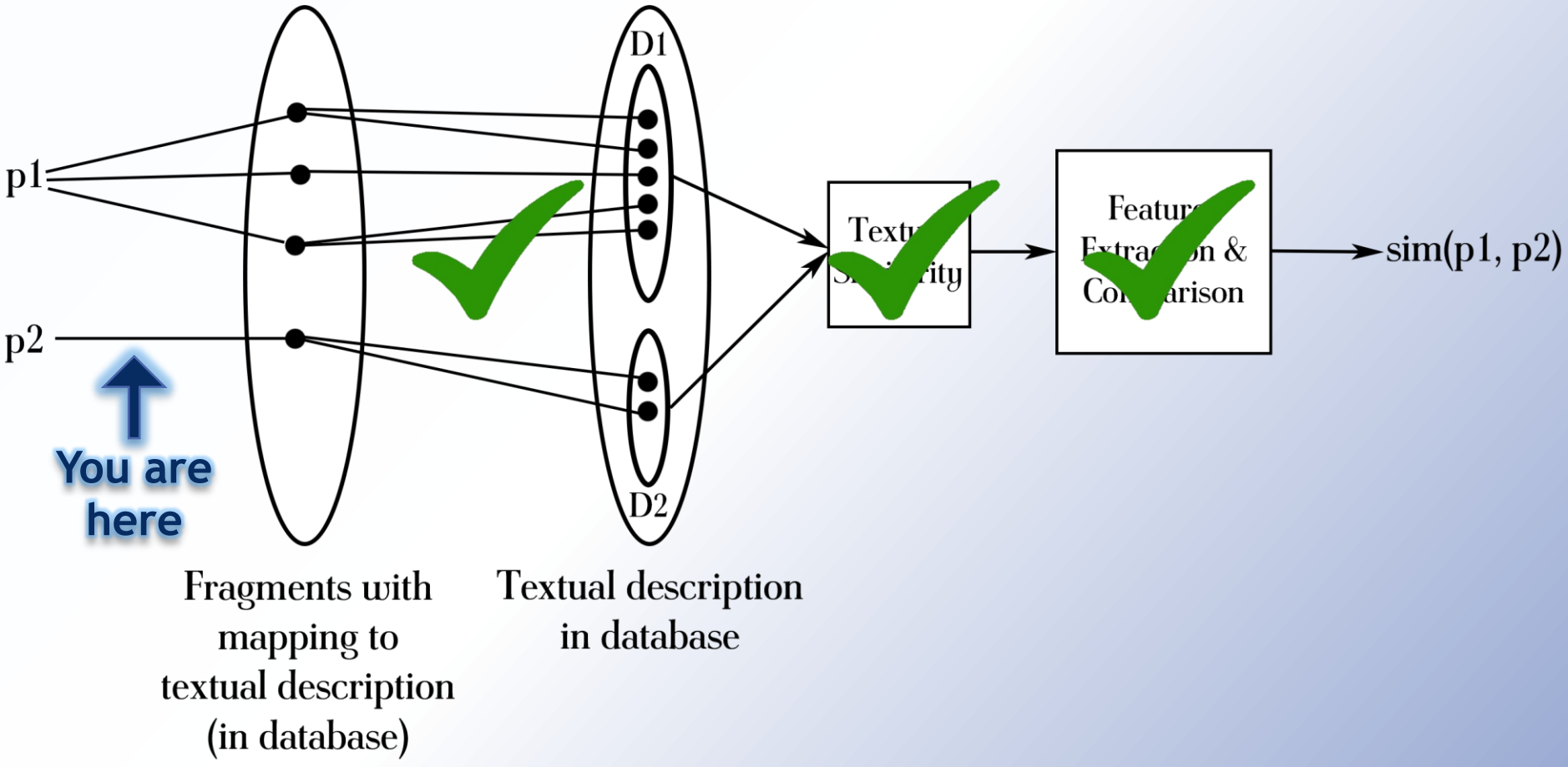
```
import urllib2
res = urllib2.urlopen(
   'http://www.example.com/')
html = res.read()
```



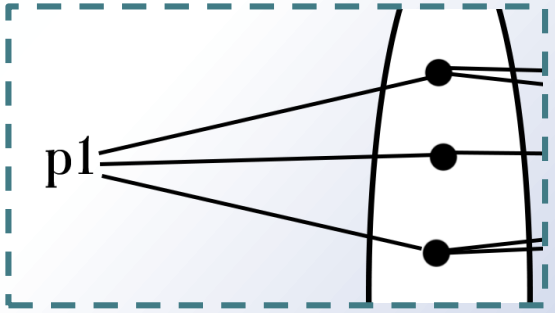
String → String



Recap


p1
p2
You are here
D1
D2
Fragments with mapping to textual description (in database)
Textual description in database
sim(p1, p2)

Query the Mapping



- 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: keyword matching followed by alignment of the common tokens
  - Global pairwise sequence alignment

Preliminary Experience

- Implementation based on 

- Code to description mapping $> 1M$
- 6500 pairs database
- Crowd-source web application like2drops
  - www.like2drops.com

http://like2drops.com

Like two drops of water?!

Help me to decide whether two code snippets are similar :)

About page

```java
import javax.xml.bind.annotation.adapters.HexBinaryAdapter;

public byte[] hexToBytes(String hexString) {
     HexBinaryAdapter adapter = new HexBinaryAdapter();
     byte[] bytes = adapter.unmarshal(hexString);
     return bytes;
}
```

language: java

```csharp
static string ByteToHexBitFiddle(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);
}
```

language: c#

Current helping status: 42 pairs!

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

The similarity level is dependent on the aim 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.

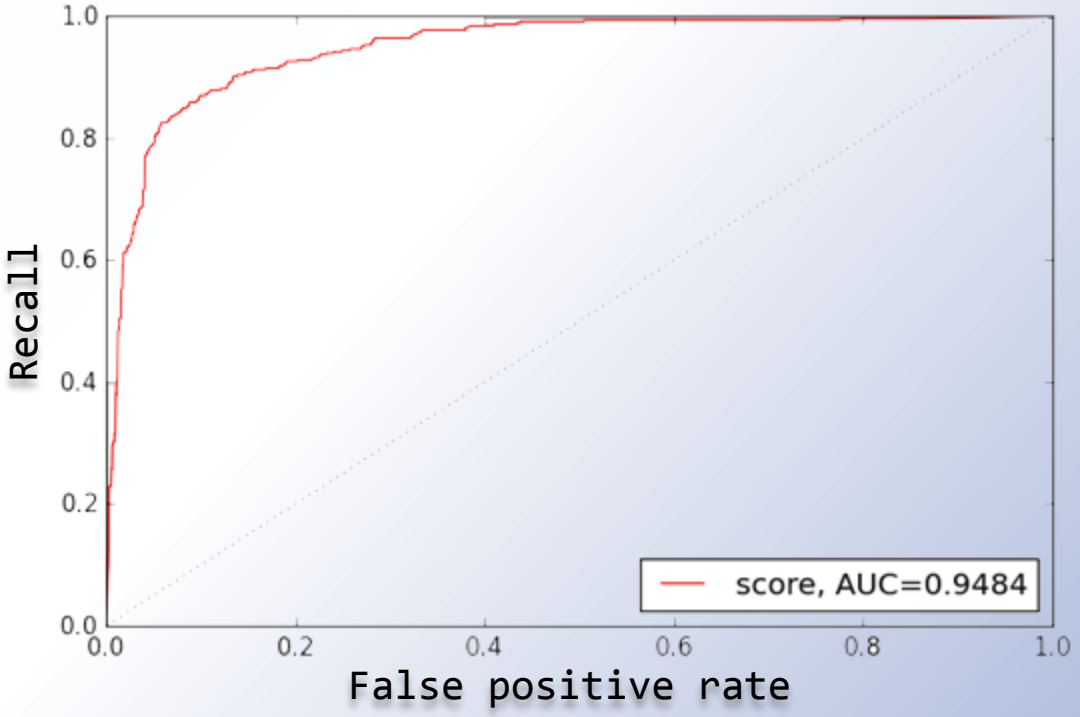
Evaluation

- The experimental database contains more than 1500 pairs of code fragments
- The preliminary results show that more than **85%** of our labels are consistent with the users' labels
- We gain around 80% precision and 75% recall, and demonstrate the promise of this approach

Accuracy, recall, precision

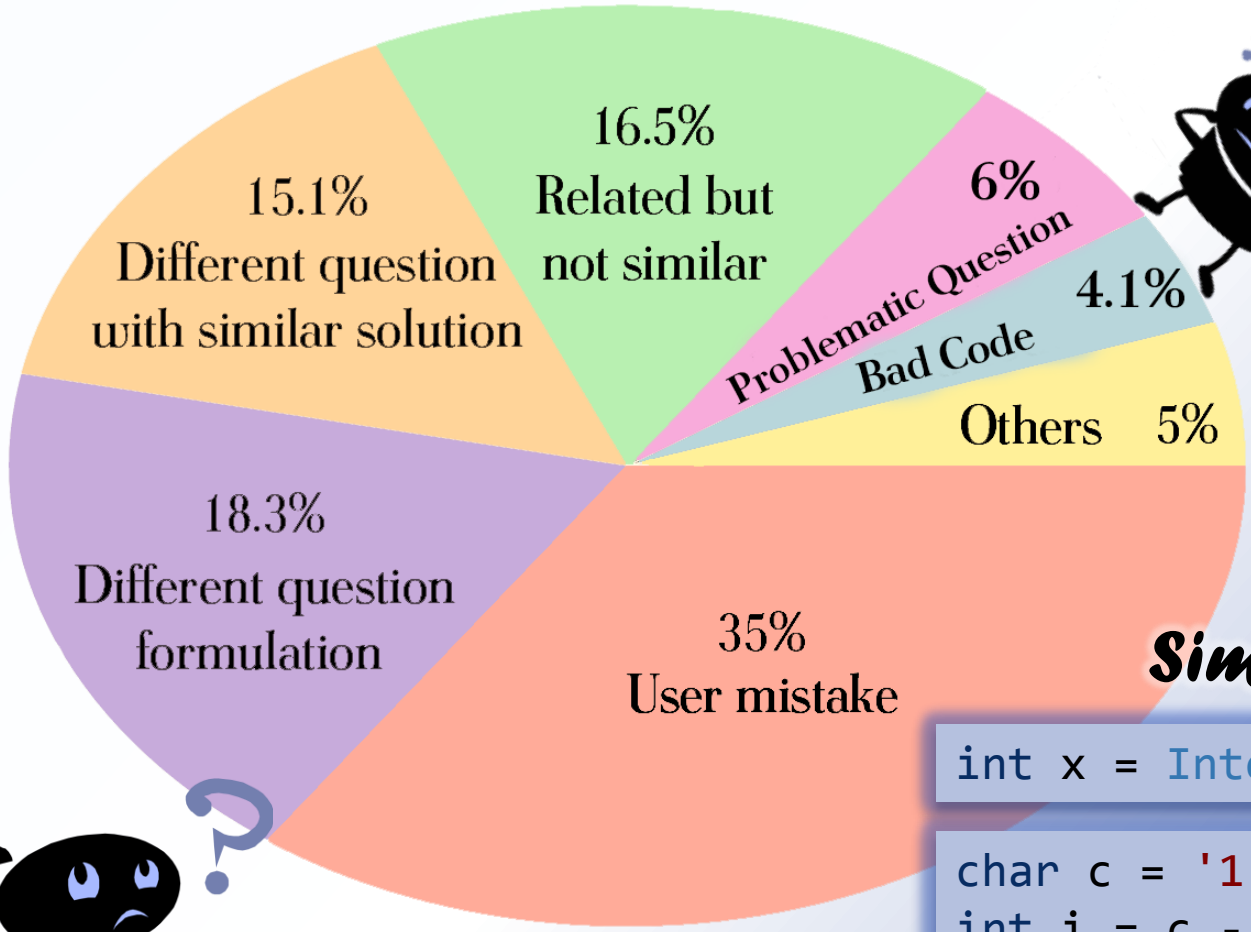
ROC - Trying all Thresholds

- ROC curves captures accuracy
- Receiver operating characteristic
- Try every threshold



AUC=0.95

Similarity is not Conclusive



Similar? Not?

```
int x = Integer.parseInt("8");
```

```
char c = '1';
int i = c - '0';
// i is now equal to 1, not '1'
```

Manually analyzed all 200 incorrect classification results

Ongoing & The Future

- Extract descriptions directly from the code
- Enrich code analysis with new code features
- Different text similarity techniques
  - ESA
  - Phrase based similarity
  - Ontologies, Freebase

Conclusion

http://like2drops.com

- Measuring semantic relatedness between code fragments based on their corresponding textual descriptions and their types graph
- Using simple techniques across large scale databases
- Combine text similarity techniques with code analysis 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]