Source Code Search
An Overview

Venkatesh Vinayakarao
venkateshv@cmi.ac.in
http://vvtesh.co.in

The increased amounts of freely available high-quality programs in code repositories such as GitHub creates a unique opportunity for new kinds of programming tools. – Veselin Raychev et al., ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, 2015.
About Me

BE (Computer Science)
MS (Software Engg.)
Software Engineer
Search Engine Dev
Yahoo and Microsoft
PhD (IR + SE)
Research & Teach
IR, SE

Code Variants and Their Retrieval

How to find better code?

Code Variants

How to write better code?

2000
BE (Computer Science)

2002
Software Engineer

2003
MS (Software Engg.)

2009
Software Engineer

2013
Search Engine Dev
Yahoo and Microsoft

2018
PhD (IR + SE)
Research & Teach
IR, SE
Growing Information & Information Needs

Several retrieval systems: Lycos, Altavista, MSN, Baidu, Yahoo!, Ask.com, etc.,

Royal Library of Alexandria 300 BC.

Bibliothèque nationale de France 1463

British Library 1970’s
170+ Million Collection

Digital Libraries 1970’s

Universal Digital Library, Project Gutenberg, etc.

Google 1998
30 Trillion documents
Content Types

Classical program analysis techniques
• do not scale well to large programs.
• have limitations while working on partial programs.

Why Code Search?

1. Limitations exist in IDE features for code search and web-scale code search engines, especially, with natural language queries.

2. Several software engineering problems which are otherwise difficult to solve can be modeled as code search problems.
Research interest in Big Code is growing.

CoaCor: Code Annotation for Code Retrieval with Reinforcement Learning

Ziyu Yao, The Ohio State University, USA, yao.470@osu.edu
Jayavardhan Reddy Peddamall, The Ohio State University, Columbus, USA, peddamall.1@osu.edu
Huan Sun, The Ohio State University, USA, sun.397@osu.edu
How to implement “factorial” in Java?
Multiple Implementation Choices

**Simple-Iterative**

```java
public static long factorial(int a) {
    if(a<1)return 1;
    long result=1;
    long x=a;
    while(x>1)    {
        result*=x;
        x--;
    }
    return result;
}
```

**Fast-Restrictive**

```java
public int factorial ( int n ) {
    switch(n){
    case 0: return 1;
    case 1: return 1;
    case 2: return 2;
    case 3: return 6;
    case 4: return 24;
    case 5: return 120;
    case 6: return 720;
    case 7: return 5040;
    case 8: return 40320;
    case 9: return 362880;
    case 10: return 3628800;
    case 11: return 39916800;
    case 12: return 479001600;
    default : throw new IllegalArgumentException();
    }
```  

**Recursive**

```java
public static int fact(int x){
    if (x==1 | x==0)
        return 1;
    return fact(x-1) * x;
}
```
Developers Discuss Choices

Is there a method that calculates a factorial in Java?

I didn't find it, yet. Did I miss something? I know a factorial method is a common example program for beginners. But wouldn't it be useful to have a standard implementation for this one to reuse? I could use such a method with standard types (int, long...) and with BigInteger / BigDecimal, too.

Statistics

<table>
<thead>
<tr>
<th>Description</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total No. of Posts</td>
<td>19 M</td>
</tr>
<tr>
<td>Posts tagged as Java</td>
<td>1.87 M</td>
</tr>
<tr>
<td>Java posts with one Java method definition</td>
<td>129 K</td>
</tr>
<tr>
<td>Java posts with “Factorial” in post</td>
<td>132</td>
</tr>
<tr>
<td>Java posts with “Factorial” in title</td>
<td>63</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bucket</th>
<th>Count (title)</th>
<th>Count (post)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computes factorial</td>
<td>53</td>
<td>94</td>
</tr>
<tr>
<td>Calls API</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Application [such as Sin(x) calculation]</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>Others</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Irrelevant</td>
<td>1</td>
<td>16</td>
</tr>
</tbody>
</table>

1Sim et al. TOSEM '11, Sadowski et al., FSE '15
2From the Sep 2014 archive of Stack Overflow.
Search on Source Code

Searchcode.com

Eclipse Find/Replace

opensearch.krugle.org
A Retrieval-based Application

Prompter

Near real-time recommendations from stack overflow to help developer code faster and easier.

[MSR 2014] Mining StackOverflow to turn the IDE into a self-confident programming prompter.
Source Code Retrieval is Hard

Source code, unlike text, does not lend well to existing IR techniques

Expressing queries is hard.

Code contexts are hard to extract.

Indexing is hard.

No surface forms for several concepts.

Binary search, increment, ...

Array: [

Declare: int var = ...

for (int i=1; i<=n; i++)
result = result * i;
Information Retrieval System

Legend: VSM = Vector Space Model, LDA = Latent Dirichlet Allocation, BM25 = Best Match 25
Can we model the source code in query and content such that existing retrieval models can be used with minimum modification?
### 13 areas, 116 Papers Surveyed*

<table>
<thead>
<tr>
<th>SE Tasks</th>
<th>Count of Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Search</td>
<td>37</td>
</tr>
<tr>
<td><em>Program Comprehension</em></td>
<td>19</td>
</tr>
<tr>
<td>Maintenance</td>
<td>12</td>
</tr>
<tr>
<td>Bug Detection</td>
<td>9</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>8</td>
</tr>
<tr>
<td>API Usage</td>
<td>7</td>
</tr>
<tr>
<td>Traceability</td>
<td>5</td>
</tr>
<tr>
<td>Clone Detection</td>
<td>5</td>
</tr>
<tr>
<td>Code Completion</td>
<td>4</td>
</tr>
<tr>
<td>Summarization</td>
<td>3</td>
</tr>
<tr>
<td>Code Quality</td>
<td>3</td>
</tr>
<tr>
<td>Education</td>
<td>3</td>
</tr>
<tr>
<td>Program Analysis</td>
<td>1</td>
</tr>
<tr>
<td><strong>Grand Total</strong></td>
<td><strong>116</strong></td>
</tr>
</tbody>
</table>

*Source code retrieval-based publications from 2000 to 2016 in ICSE, FSE, ASE, TSE, TOSEM, ESE, ICSME, WCRE.*
Program Comprehension

Distributed Systems.
Edwards et al., JSS ’06 [5].

Using Logic (Intelligent Agents).
Biggerstaff, ICSE ’93 [6].

Semantic properties (FCA).
Query-time analysis,
Platform Dependent.

Feature Location

Annotated CFG + Model Checking.
Ouellet et al., ICSE ’12 [7].

Dynamic Test-based.
Koschke et al., ASE ’05 [8].

PA

IR

Structural Properties
(Execution Trace),
Access to runtime information
may not be available.
Liu et al., ASE ’07 [9].

Structural properties
(Call Graph).
Description of all features
required for the system
Zhao et al., TOSEM ’06 [10].

Others

Semantic properties (FCA).
Query-time analysis,
Platform Dependent.
Poshyvanyk et al.,

*PA = Program Analysis, IR = Information Retrieval
FCA + LSI Approach: A Discussion

• Query: Natural Language (For eg., “print page”)
• Content: Method x Attribute. Each method has a corresponding vector of terms.
• Indexing: LSI (based on VSM)
Extracting Concepts

<table>
<thead>
<tr>
<th></th>
<th>printer</th>
<th>print</th>
<th>page</th>
<th>job</th>
<th>device</th>
<th>paper</th>
<th>rendering</th>
</tr>
</thead>
<tbody>
<tr>
<td>startJob</td>
<td></td>
<td>×</td>
<td></td>
<td>×</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>endJob</td>
<td></td>
<td>×</td>
<td></td>
<td>×</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cancelJob</td>
<td></td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>startPage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>endPage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>getBounds</td>
<td></td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

**Concepts**

C1 = \{\{startJob, endJob, cancelJob\}, \{print, job\}\}
C2 = \{\{startPage, endPage\}, \{page, paper, rendering\}\}
C3 = \{\{getBounds\}, \{printer, device, paper\}\}
C4 = \{\{startPage, endPage, getBounds\}, \{paper\}\}
C5 = \{\{startJob, endJob, cancelJob, startPage, endPage, getBounds\}, \{\}\}
C6 = \{\{\}, \{printer, print, page, job, device, paper, render\}\}

There exist sub-concepts! For example, C2 is a sub-concept of C1.
FCA - Lattice Representation

startPage, endPage, getBounds, startJob, endJob, cancelJob

paper

device, printer

startPage, endPage, getBounds

rendering, page

startPage, endPage

job, print

getBounds

printer, print, page, job, device, paper, rendering
Reduced Lattice
Education

SAT Solvers for Automated Program Feedback.
Singh et al., PLDI ’13 [2].

OverCode: Clustering Assignments for Feedback.
Glassman, UIST ’14 [1].

Structural Properties: (AST).
Schleimer et al., SIGMOD ’03 [3].

Winnowing in Stanford MOSS,
Plagiarism Detection.
Schleimer et al., SIGMOD ’03 [3].

Structural Properties: (Document Finger Printing).
Winnowing in Stanford MOSS,
Plagiarism Detection.
Schleimer et al., SIGMOD ’03 [3].

Assignment Feedback.
Nguyen et al., WWW ’14 [4].

Others

S/w Engg. Education

IR

PA
Opportunities

1. Index-time instead of query-time.
2. Platform independence.
3. Support for partial programs.
4. Principled approaches for indexing source code structure and semantics.
5. Scalability (to Big Code)
Missing Natural Language (NL) Terms

A Code Search Problem
The Problem: Missing NL Terms in Code

• Developers ask:
  • Get me the lines where variables are assigned a value.
  • Where are arrays declared?
  • Which lines either increment or decrement a variable?

Natural Language (NL) terms found in the queries are absent in source code!
Entity Retrieval in Code Snippets

Discussion Forum Data

Entity List
array declare ...

Entity Discovery

Entity Profile Preparation

Entity Profiles
array -> {{[]}}
declare -> {([int =])}

Entity Profile KB
<<uses>>

Index

Query
Where is array declared?

Developers
... Tools
Exemplar Prompter Sourcerer ...

Users

Source code

public class FindLargest {
    public static void main(String[] args) {
        int[] numbers = {...}; //declare, array
        ...
    }
}

Tagged source code

public class FindLargest {
    public static void main(String[] args) {
        int[] numbers = {...};
        ...
    }
}
Discovery

Input: Set of Seed Entities:
{collection, datatype, ...}

Ordered PoS Pattern Sequences:
- NN IN DT ENTITY IN NNS:70
- NN IN DT ENTITY IN NNS IN:51
- DT NN IN DT ENTITY:45
- NN IN DT ENTITY IN DT NN:44
- ...

PoS Type Detection

PoS Pattern Mining

Candidate Entity Mining

PoS Types:
- collection --> [NNP, NN],
- datatype --> [VB, JJ, NNP, FW, NN],
- ...

Output: Set of Discovered Entities:
{array, pointer, type, ...}
Profile Construction

• n-gram TF-IDF over source code

<table>
<thead>
<tr>
<th>Uni-gram Pattern</th>
<th>Normalized Frequency</th>
<th>n-gram Pattern</th>
<th>Normalized Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>1.00</td>
<td>if ( ) {</td>
<td>1.00</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>( )</td>
<td>0.50</td>
</tr>
<tr>
<td>if</td>
<td>0.65</td>
<td>= ( ( ) )</td>
<td>0.25</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>( new ( ) {</td>
<td>0.25</td>
</tr>
<tr>
<td>while</td>
<td>0.10</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td>0.06</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Patterns for “conditional”

What is the intuition?
http://irpower.herokuapp.com/
Linking

Input: Code Snippet
```java
public enum Planet{
    MERCURY (3.7),
    VENUS   (8.872),
    EARTH   (9.78),
}
```

Output: Annotated Code
```java
public enum Planet{
    MERCURY (3.7), //parameter
    VENUS   (8.872), // parameter
    EARTH   (9.78),  // parameter
}
```
Results

• Entity Discovery
  • 77.5% Precision

• Entity Profile Knowledge-base Construction
  • 71.8% Precision

Vinayakarao et al., WSDM ’17.
Future Work
Can we query for desired properties?

Query, q = “<code context>, <desired property>”

Retrieval Model
{jSense, VSM, LDA, BM25, …}

Result = ??

Indexed Content
code snippets

Snippet1

String input = "Be in present";
StringBuilder input1 = new StringBuilder();
input1.append(input);
input1=input1.reverse();

Snippet2

String input = "Be in present";
byte [] strAsByteArray = input.getBytes();
byte [] result = new byte [strAsByteArray.length];
for(int i = 0; i<strAsByteArray.length; i++){
    result[i] = strAsByteArray[strAsByteArray.length-i-1];
}
System.out.println( new String(result));

System: If you are looking for a readable snippet, which one will you prefer?
References

• Ridhi Jain, Sai Prathik Saba Bama, Venkatesh Vinayakarao and Rahul Purandare. A Search System for Mathematical Expressions on Software Binaries. In the Proceedings of The 15th International Conference on Mining Software Repositories (MSR 2018), Sweden.


Thank You

venkateshv@cmi.ac.in

This slide deck is available at http://vvtesh.co.in/.