CHENNAI MATHEMATICAL INSTITUTE

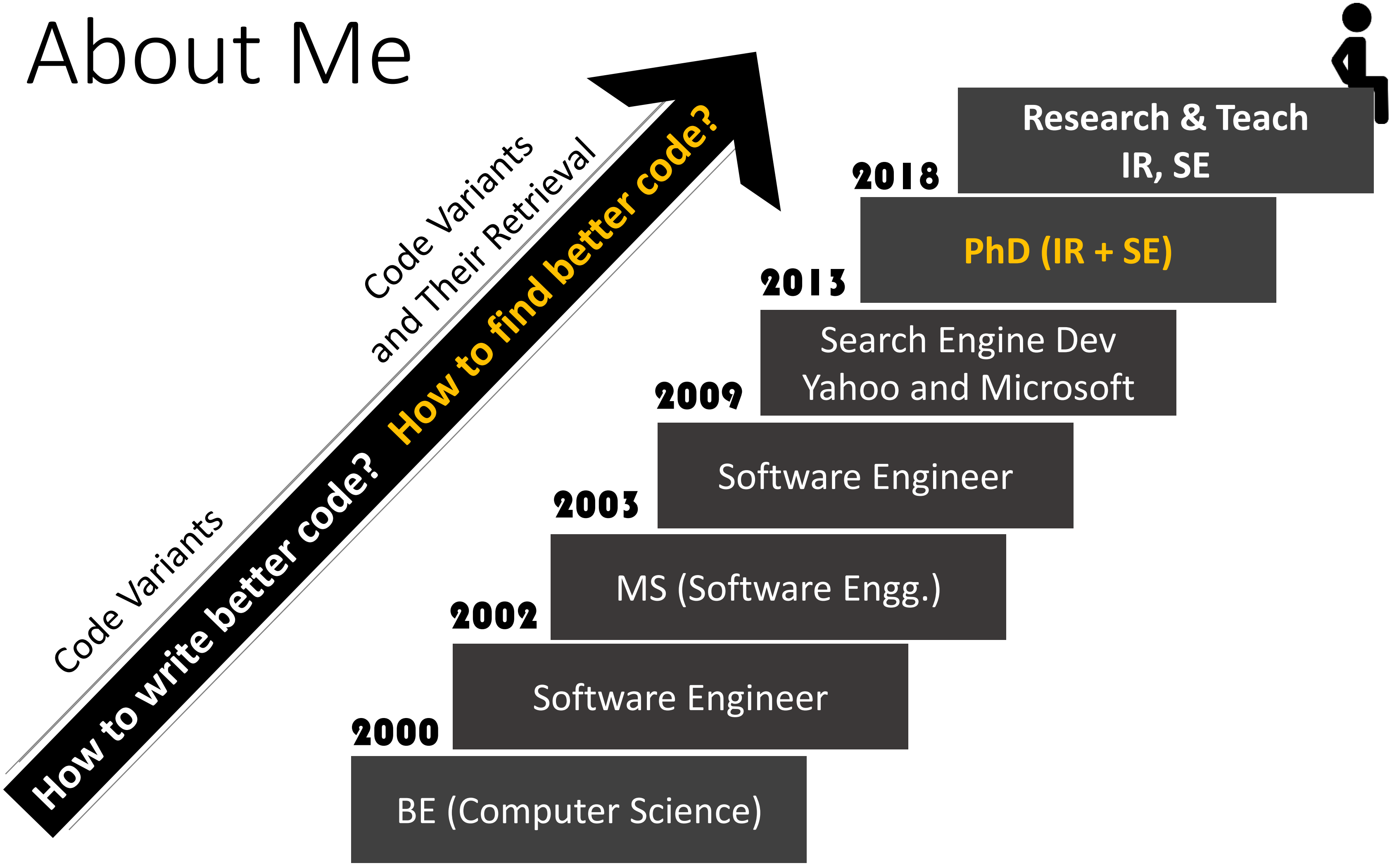
Source Code Search

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)

2000

Software Engineer

2002

MS (Software Engg.)

2003

Software Engineer

2009

Search Engine Dev
Yahoo and Microsoft

2013

PhD (IR + SE)

2018

Research & Teach
IR, SE

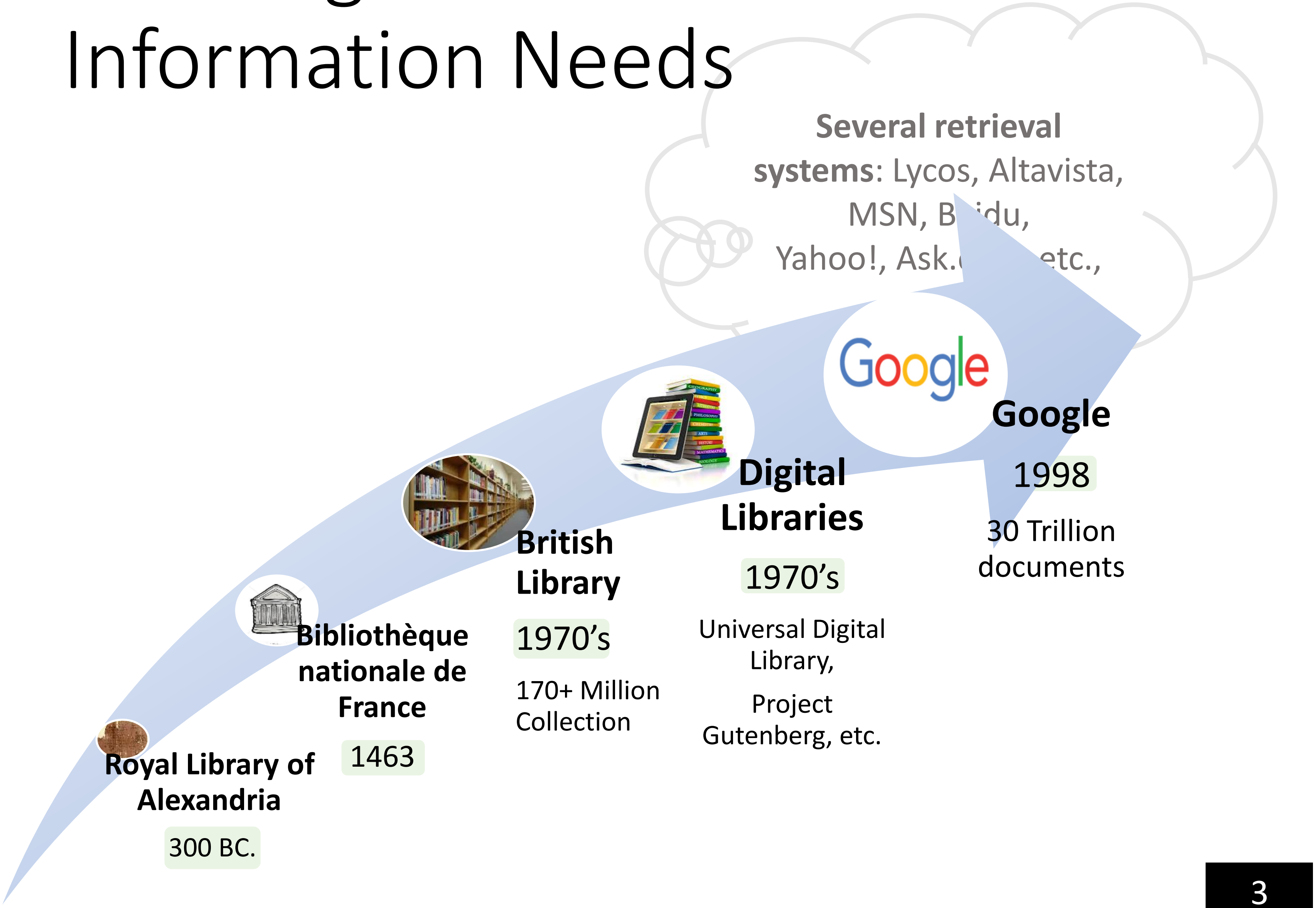
How to write better code?

How to find better code?

Code Variants

Code Variants
and Their Retrieval

Growing Information & Information Needs



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.

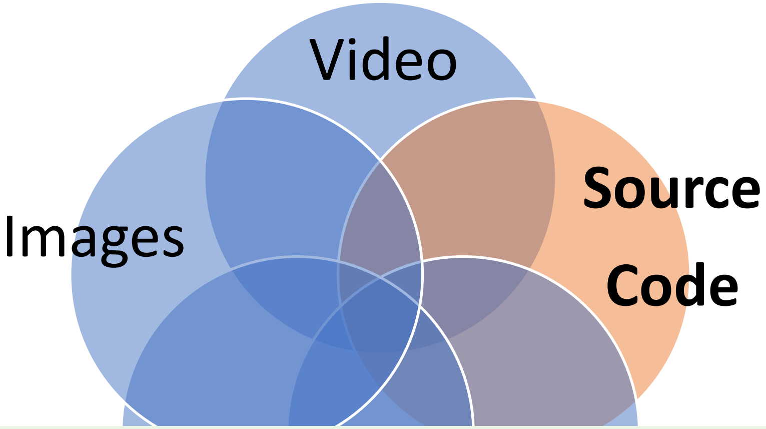
Google

1998

30 Trillion documents

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

Content Types



Research Focus

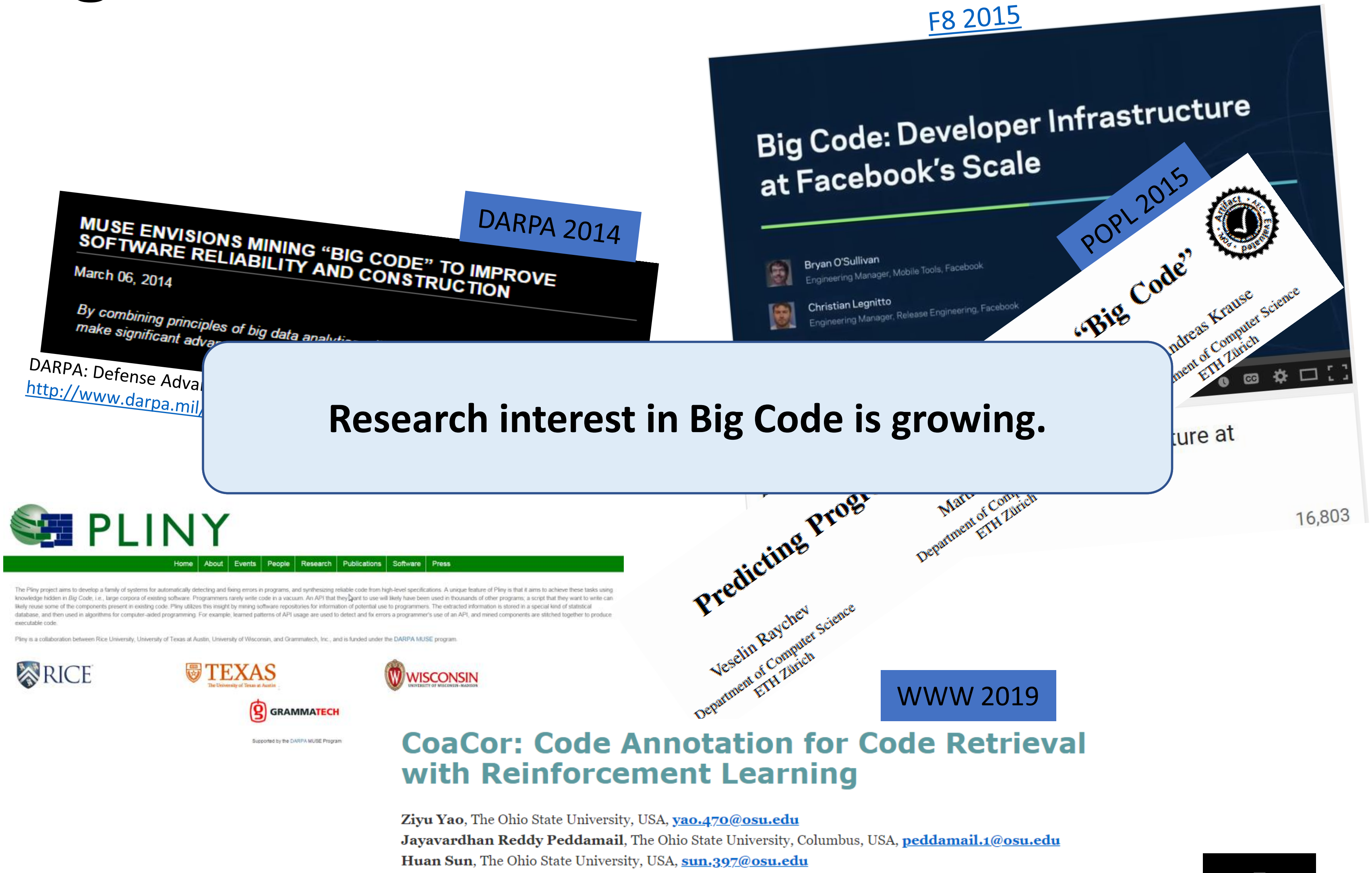
Why Code Search?

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

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

3 Several software engineering problems which are otherwise difficult to solve can be modeled as code search problems.

Big Code in the News!



F8 2015

Big Code: Developer Infrastructure at Facebook's Scale

Bryan O'Sullivan
Engineering Manager, Mobile Tools, Facebook

Christian Legnitto
Engineering Manager, Release Engineering, Facebook

ture at

16,803

DARPA 2014

MUSE ENVISIONS MINING "BIG CODE" TO IMPROVE SOFTWARE RELIABILITY AND CONSTRUCTION

March 06, 2014

By combining principles of big data analyti
make significant adva

DARPA: Defense Adva
http://www.darpa.mil/

POPL 2015

"Big Code"

ndreas Krause
ment of Computer Science
ETH Zürich

Predicting Prog

Veselin Raychev
Department of Computer Science
ETH Zürich

Mart
Department of Com
ETH Zürich

Research interest in Big Code is growing.

PLINY

The Pliny project aims to develop a family of systems for automatically detecting and fixing errors in programs, and synthesizing reliable code from high-level specifications. A unique feature of Pliny is that it aims to achieve these tasks using knowledge hidden in *Big Code*, i.e., large corpora of existing software. Programmers rarely write code in a vacuum. An API that they want to use will likely have been used in thousands of other programs; a script that they want to write can likely reuse some of the components present in existing code. Pliny utilizes this insight by mining software repositories for information of potential use to programmers. The extracted information is stored in a special kind of statistical database, and then used in algorithms for computer-aided programming. For example, learned patterns of API usage are used to detect and fix errors a programmer's use of an API, and mined components are stitched together to produce executable code.

Pliny is a collaboration between Rice University, University of Texas at Austin, University of Wisconsin, and Grammatech, Inc., and is funded under the DARPA MUSE program.

RICE TEXAS The University of Texas at Austin WISCONSIN UNIVERSITY OF WISCONSIN–MADISON GRAMMATECH

Supported by the DARPA MUSE Program

WWW 2019

CoaCor: Code Annotation for Code Retrieval with Reinforcement Learning

Ziyu Yao, The Ohio State University, USA, yao.470@osu.edu
Jayavardhan Reddy Peddamail, The Ohio State University, Columbus, USA, peddamail.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

```
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;
}
```

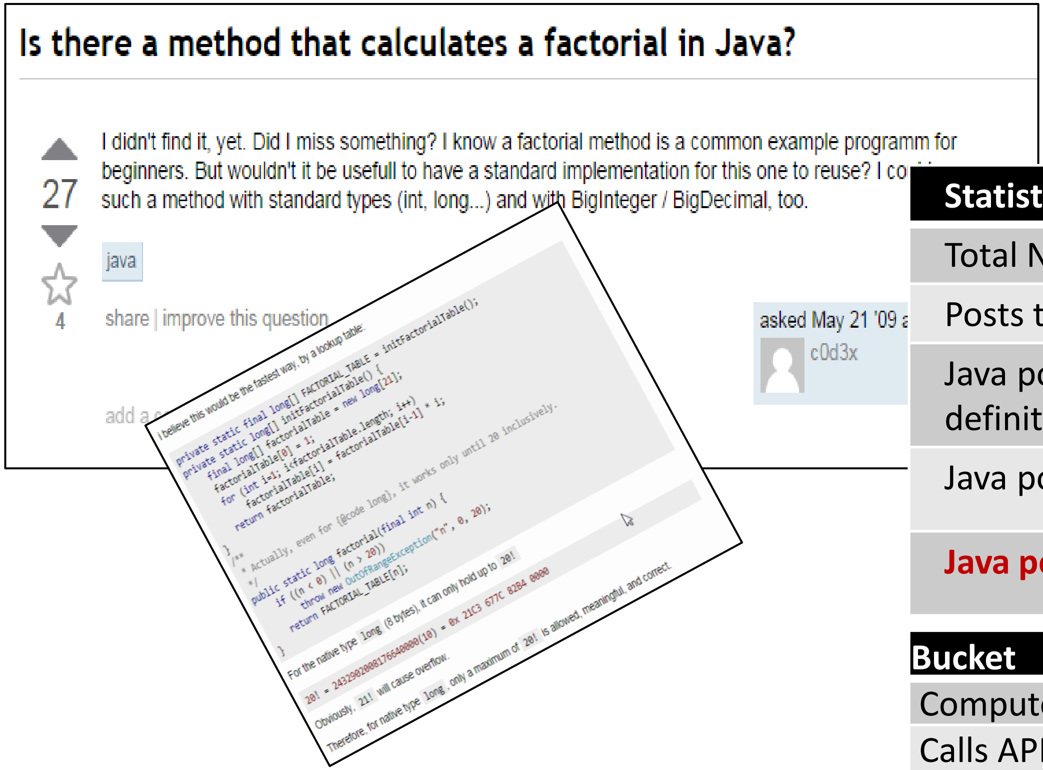
Fast-Restrictive

```
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

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

Developers Discuss Choices[1]



Is there a method that calculates a factorial in Java?

27

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

java

4

share | improve this question

asked May 21 '09 a

c0d3x

add a c

I believe this would be the fastest way, by a lookup table:

```java
private static final long[] FACTORIAL_TABLE = initFactorialTable();
private static long[] initFactorialTable() {
    final long[] factorialTable = new long[21];
    factorialTable[0] = 1;
    for (int i=1; i<factorialTable.length; i++)
        factorialTable[i] = factorialTable[i-1] * i;
    return factorialTable;
}
/**
 * Actually, even for {@code long}, it works only until 20 inclusively.
 */
public static long factorial(final int n) {
    if ((n < 0) || (n > 20))
        throw new OutOfRangeException("n", 0, 20);
    return FACTORIAL_TABLE[n];
}
```

For the native type `long` (8 bytes), it can only hold up to `20!`

`20! = 2432902008176640000(10) = 0x 21C3 677C 82B4 0000`

Obviously, `21!` will cause overflow.

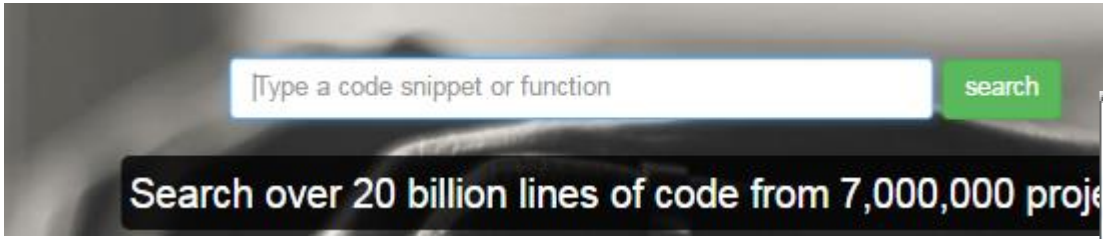
Therefore, for native type `long`, only a maximum of `20!` is allowed, meaningful, and correct.

Statistics[2]	Count
Total No. of Posts	19 M
Posts tagged as Java	1.87 M
Java posts with one Java method definition	129 K
Java posts with "Factorial" in post	132
Java posts with "Factorial" in title	**63**

Bucket	Count (title)	Count (post)
Computes factorial	53	94
Calls API	2	3
Application [such as Sin(x) calculation]	5	15
Others	2	5
Irrelevant	1	16

[1] Sim et al. TOSEM '11, Sadowski et al., FSE '15
[2] From the Sep 2014 archive of Stack Overflow.

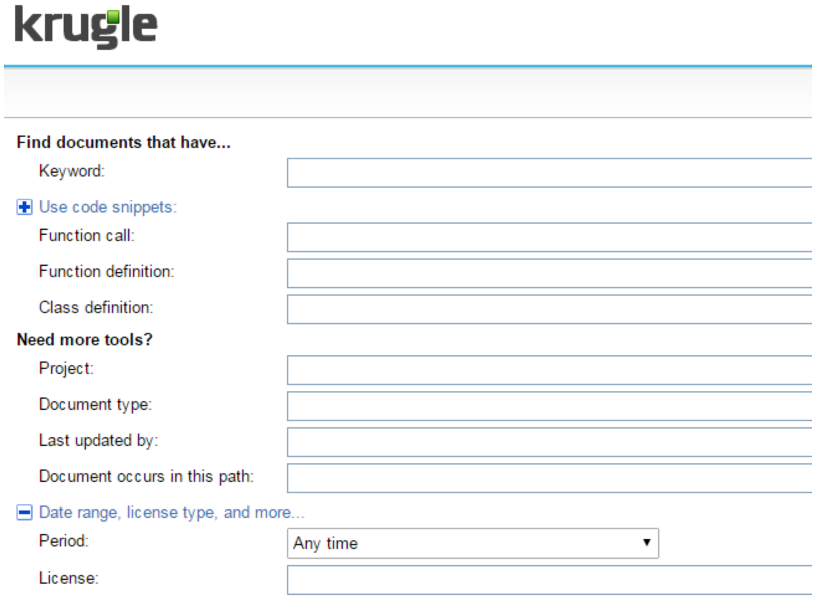
Search on Source Code



Type a code snippet or function

Search over 20 billion lines of code from 7,000,000 proje

Searchcode.com



Find documents that have...

Keyword:

Use code snippets:

Function call:

Function definition:

Class definition:

Need more tools?

Project:

Document type:

Last updated by:

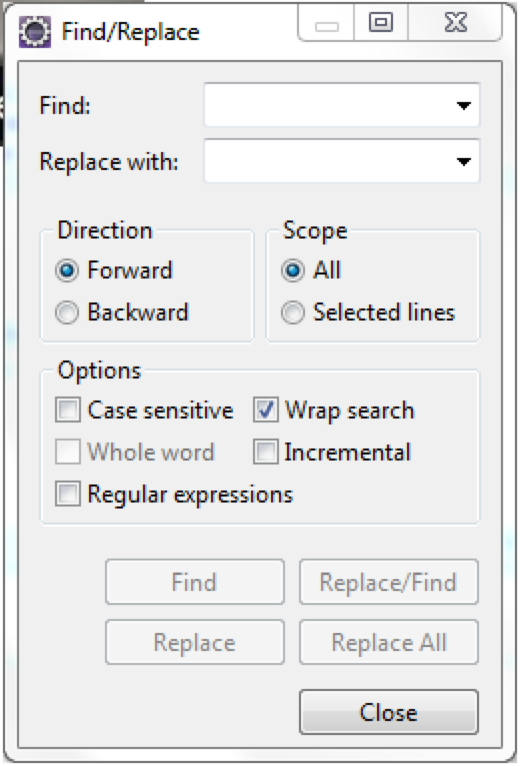
Document occurs in this path:

Date range, license type, and more...

Period: Any time

License:

opensearch.krugle.org



Find/Replace

Find:

Replace with:

Direction

● Forward

○ Backward

Scope

● All

○ Selected lines

Options

□ Case sensitive ☑ Wrap search

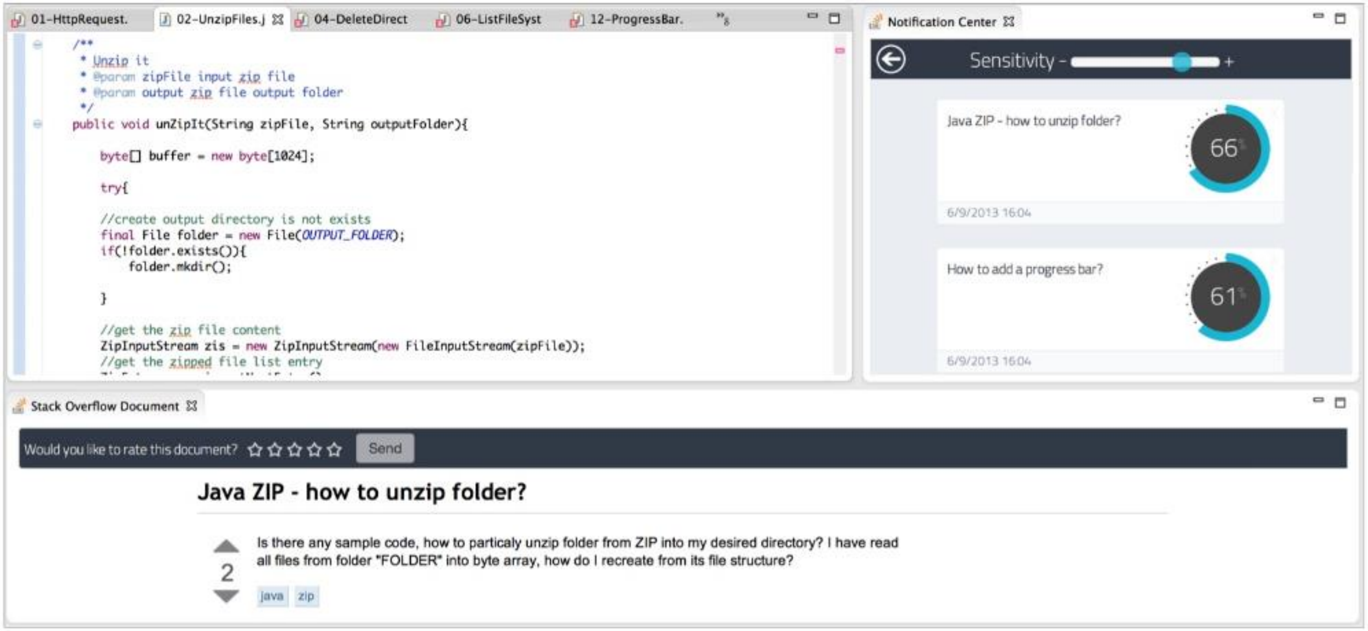
□ Whole word □ Incremental

□ Regular expressions

Eclipse Find/Replace

A Retrieval-based Application

Prompter



```
/**
 * Unzip it
 * @param zipFile input zip file
 * @param output zip file output folder
 */
public void unZipIt(String zipFile, String outputFolder){

    byte[] buffer = new byte[1024];

    try{

    //create output directory is not exists
    final File folder = new File(OUTPUT_FOLDER);
    if(!folder.exists()){
        folder.mkdir();

    }

    //get the zip file content
    ZipInputStream zis = new ZipInputStream(new FileInputStream(zipFile));
    //get the zipped file list entry
```

Java ZIP - how to unzip folder? 66%

6/9/2013 16:04

How to add a progress bar? 61%

6/9/2013 16:04

Would you like to rate this document? ☆ ☆ ☆ ☆ ☆

Java ZIP - how to unzip folder?

2

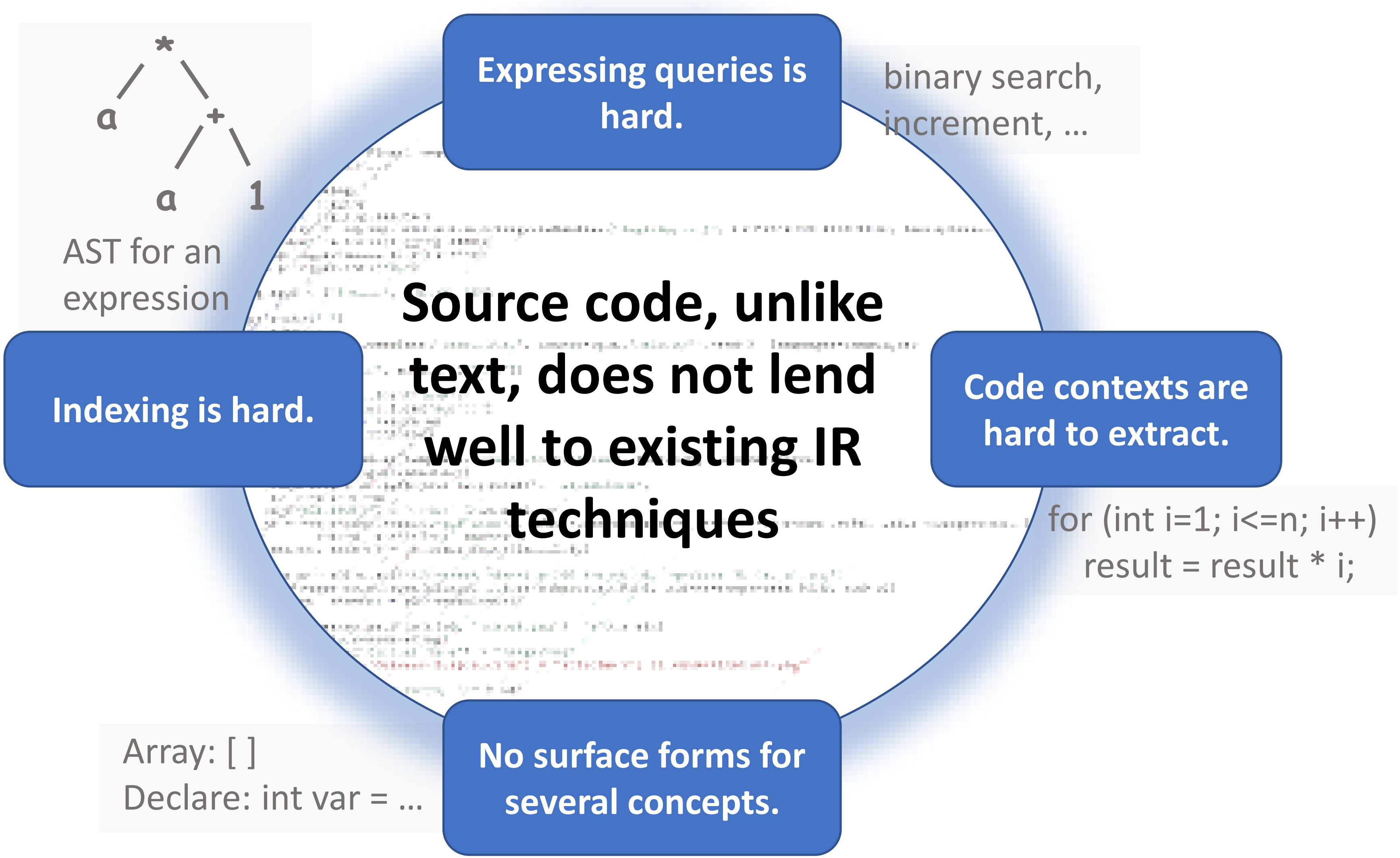
Is there any sample code, how to particaly unzip folder from ZIP into my desired directory? I have read all files from folder "FOLDER" into byte array, how do I recreate from its file structure?

java zip

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



AST for an expression

Expressing queries is hard.

binary search, increment, …

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

Indexing is hard.

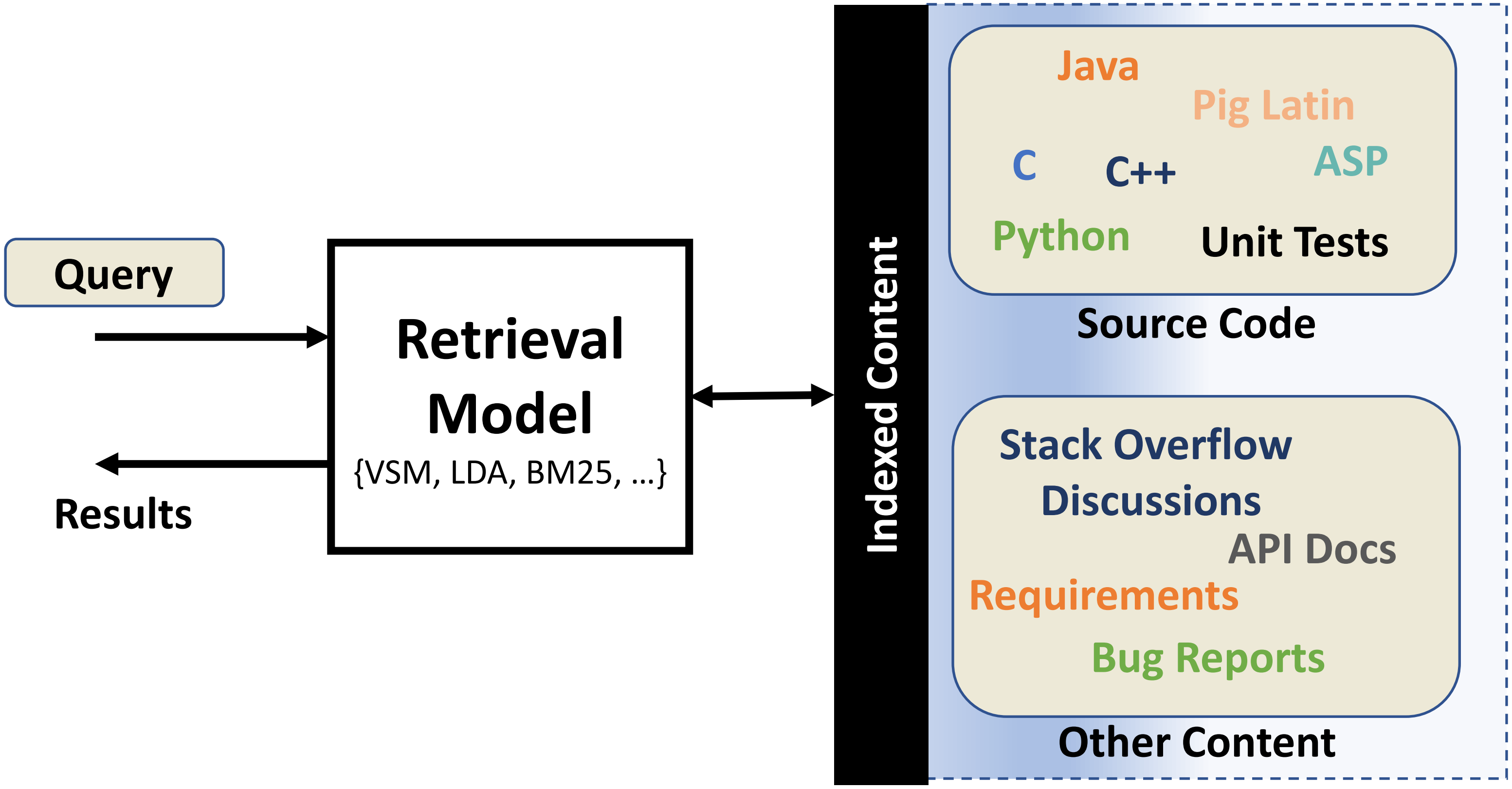
Code contexts are hard to extract.

```
for (int i=1; i<=n; i++)
    result = result * i;
```

No surface forms for several concepts.

Array: [ ]
Declare: int var = …

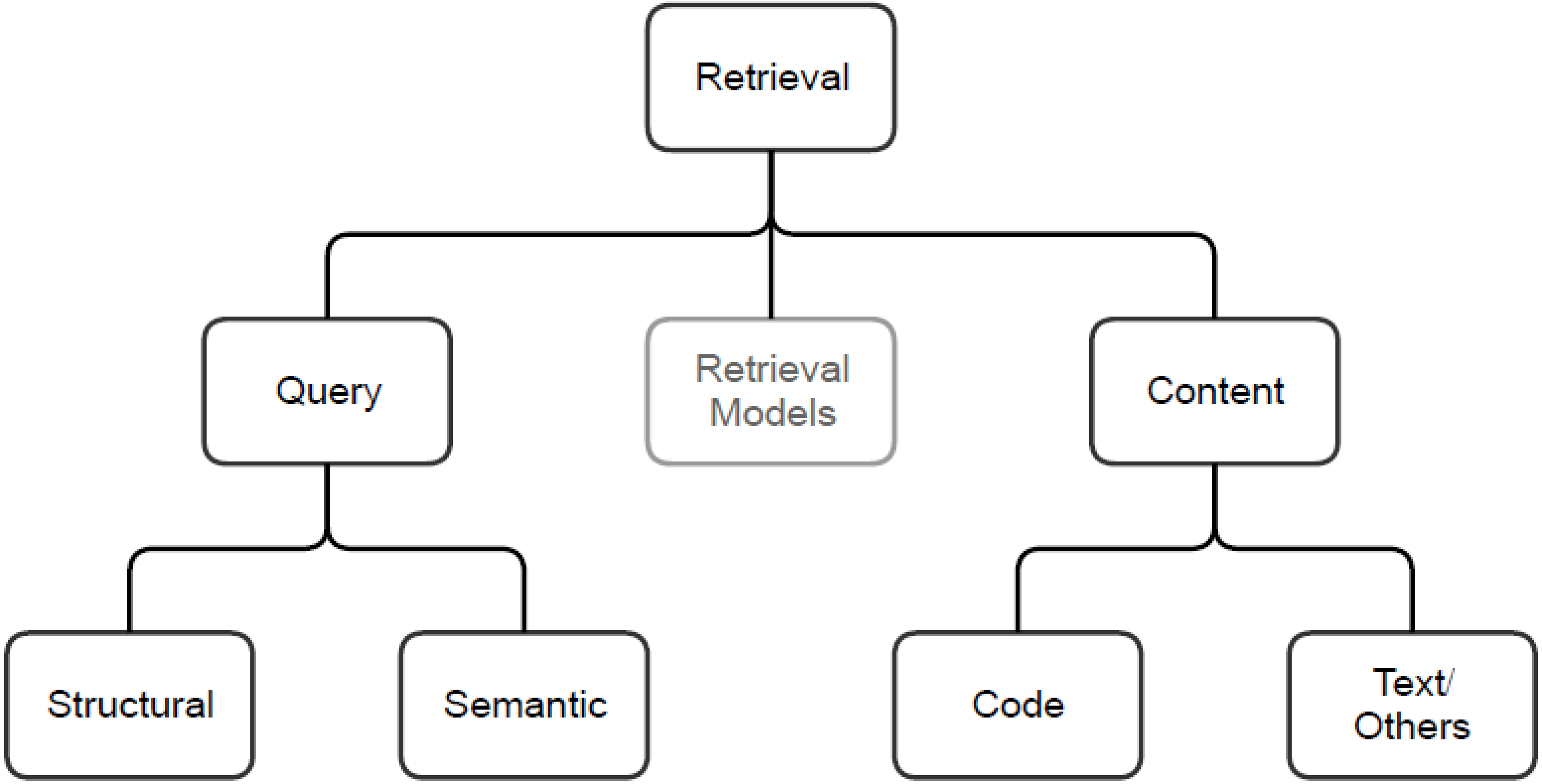
Information Retrieval System



Legend: VSM = Vector Space Model, LDA = Latent Dirichlet Allocation, BM25 = Best Match 25

Retrieval Taxonomy

Can we model the source code in query and content such that existing retrieval models can be used with minimum modification?



Survey

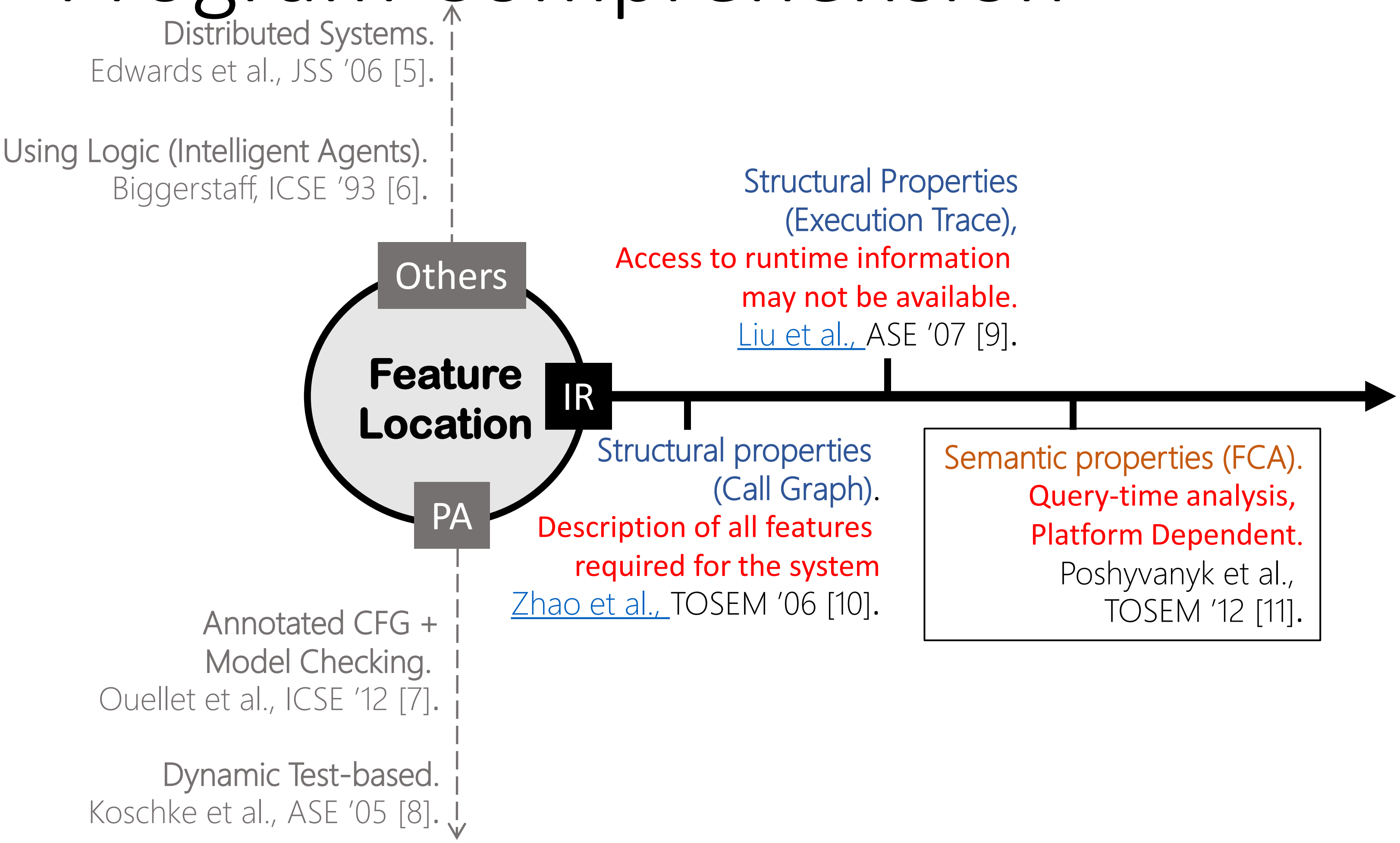
13 areas, 116 Papers Surveyed*

SE Tasks	Count of Papers
Code Search	37
Program Comprehension	19
Maintenance	12
Bug Detection	9
Miscellaneous	8
API Usage	7
Traceability	5
Clone Detection	5
Code Completion	4
Summarization	3
Code Quality	3
Education	3
Program Analysis	1
Grand Total	**116**

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

List of papers: Microsoft Excel Worksheet

Program Comprehension



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

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

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].

Semantic properties (FCA).
Query-time analysis,
Platform Dependent.
Poshyvanyk et al.,
TOSEM '12 [11].

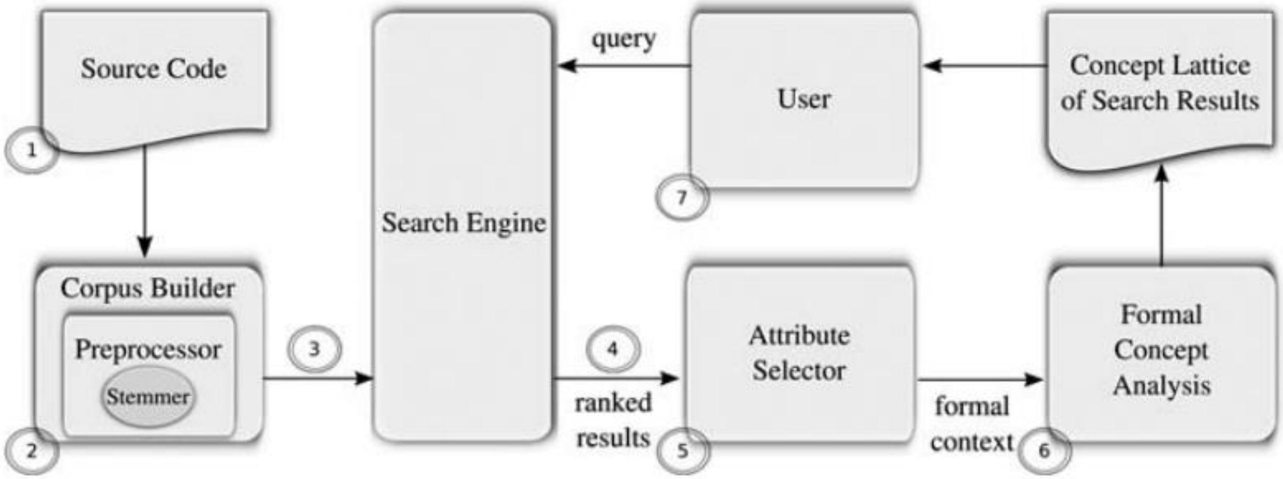
Annotated CFG +
Model Checking.
Ouellet et al., ICSE '12 [7].

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

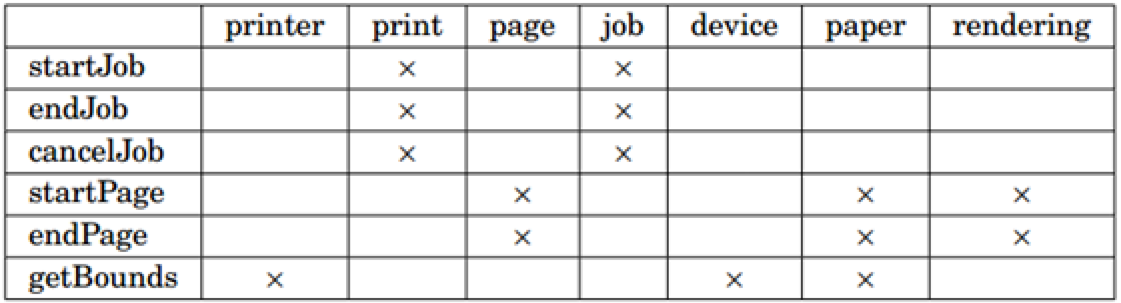
*PA = Program Analysis, IR = Information Retrieval

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



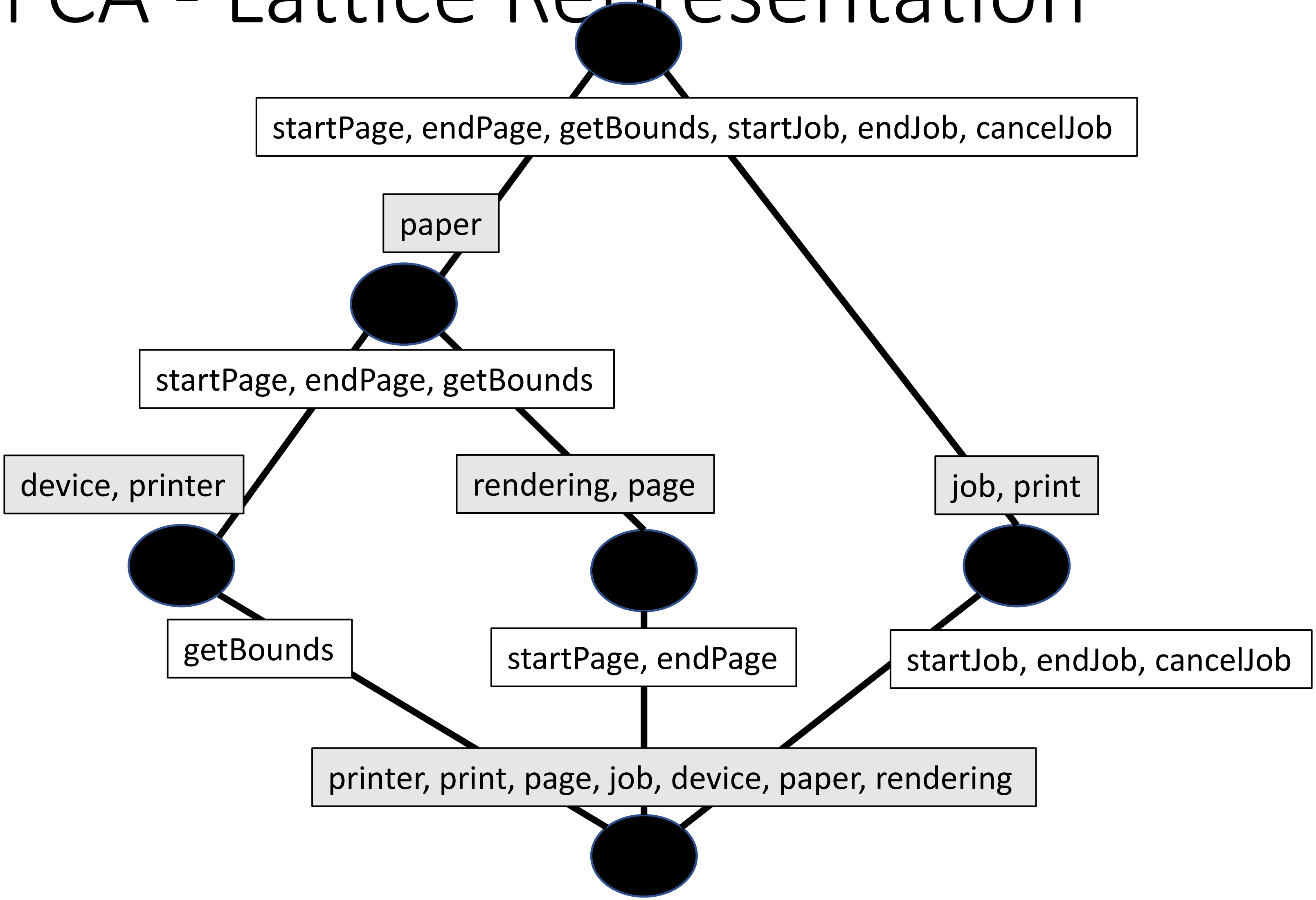
	printer	print	page	job	device	paper	rendering
startJob		×		×			
endJob		×		×			
cancelJob		×		×			
startPage			×			×	×
endPage			×			×	×
getBounds	×				×	×	

Concepts

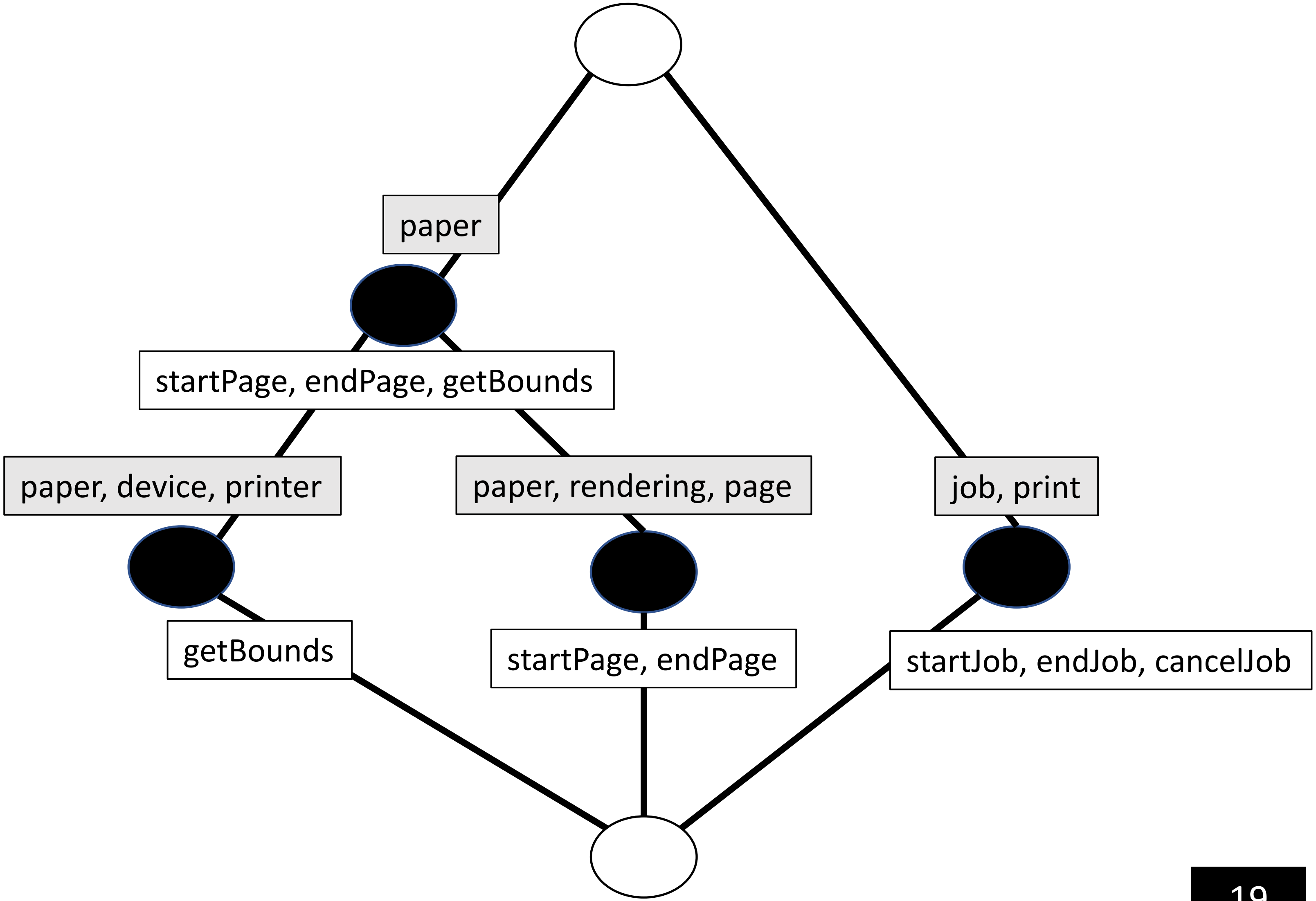
C1= ({startJob, endJob, cancelJob}, {print, job})
C2= ({startPage, endPage}, {page, paper, rendering})
C3= ({getBounds},{printer, device, paper})
C4= ({startPage, endPage, getBounds}, {paper})
C5= ({startJob, endJob, cancelJob, startPage, endPage, getBounds}, {})
C6= ({}, {printer, print, page, job,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

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

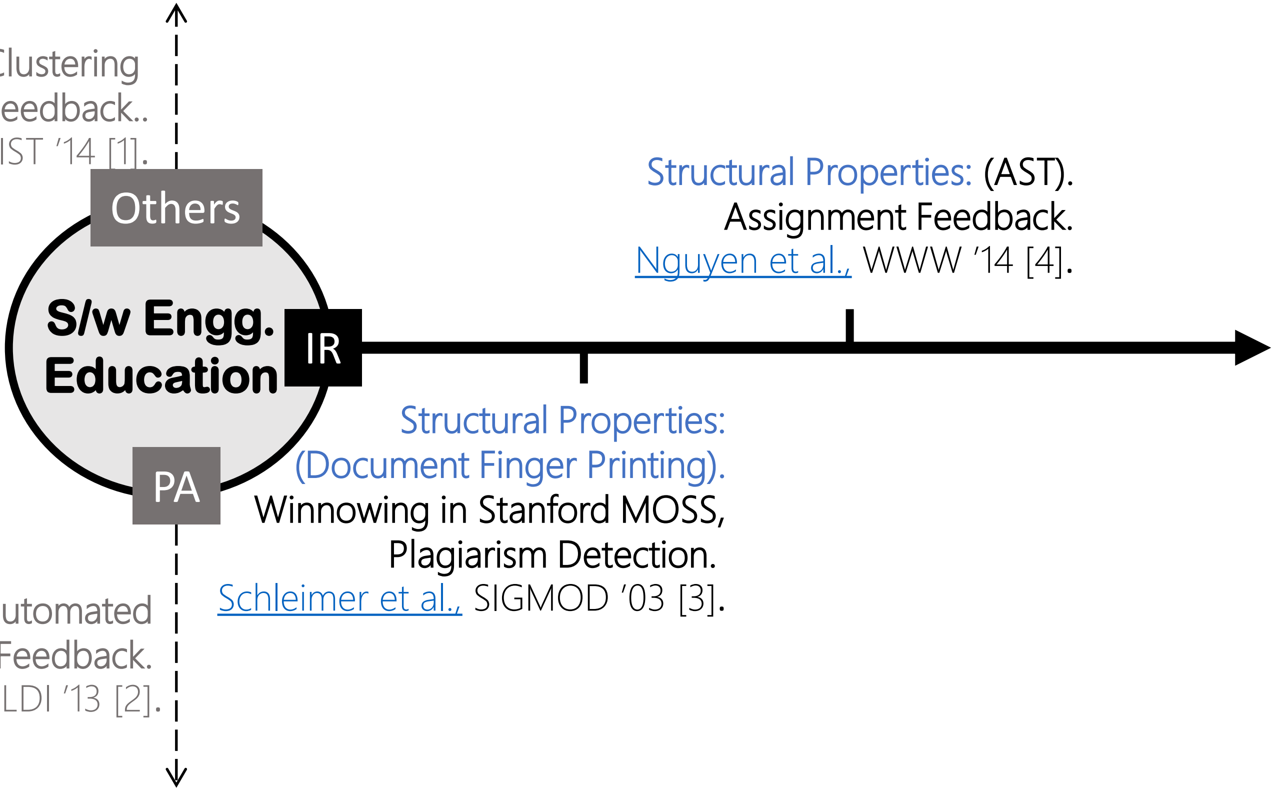
FCA - Lattice Representation


startPage, endPage, getBounds, startJob, endJob, cancelJob
paper
startPage, endPage, getBounds
device, printer
rendering, page
job, print
getBounds
startPage, endPage
startJob, endJob, cancelJob
printer, print, page, job, device, paper, rendering

Reduced Lattice



Education



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

Structural Properties: (AST).
Assignment Feedback.
Nguyen et al., WWW ’14 [4].

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

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

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

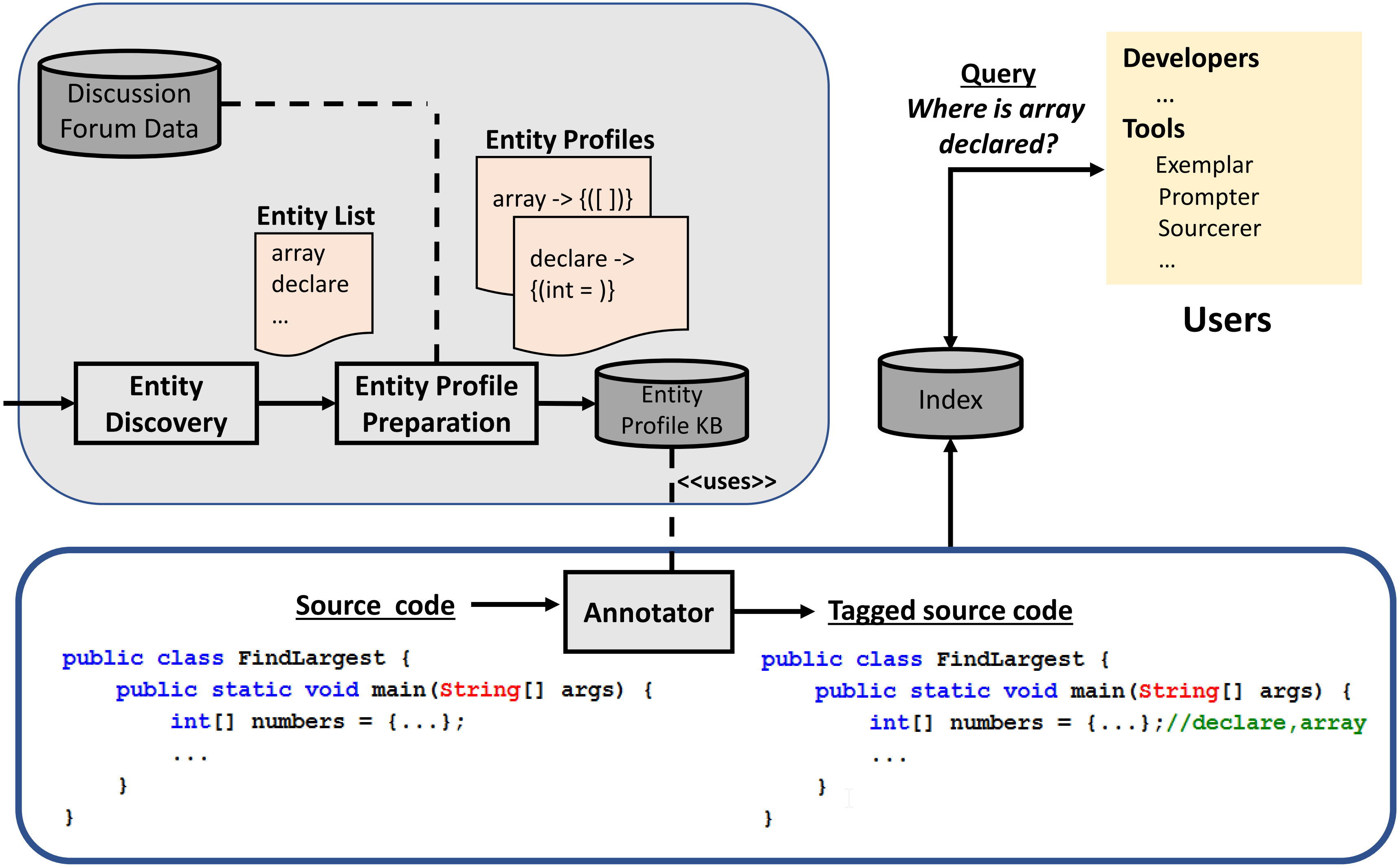
Image Source https://commons.wikimedia.org/wiki/File:Fotothek_df_n-10_0000117.jpg

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!

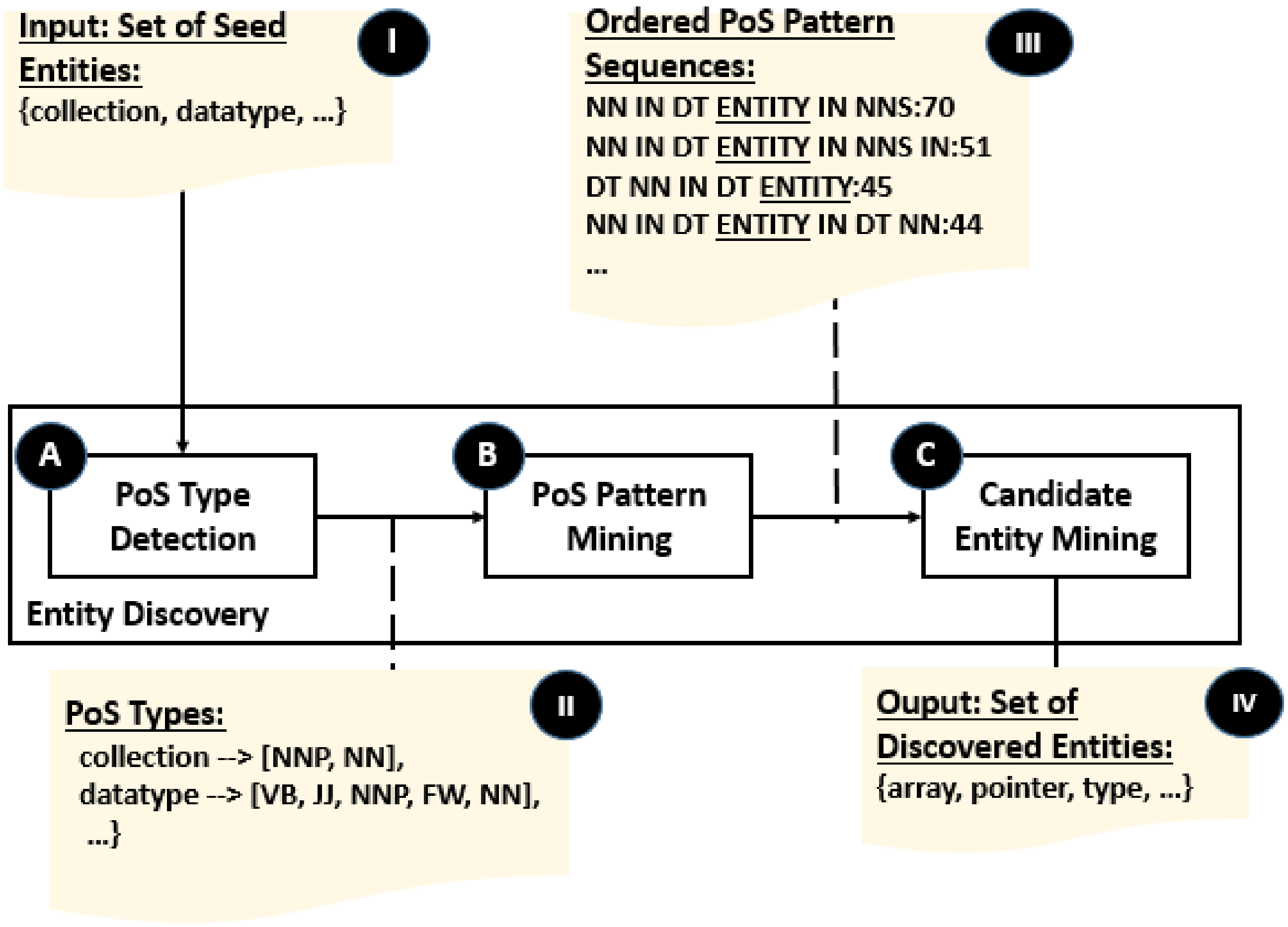
Entity Retrieval in Code Snippets



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

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

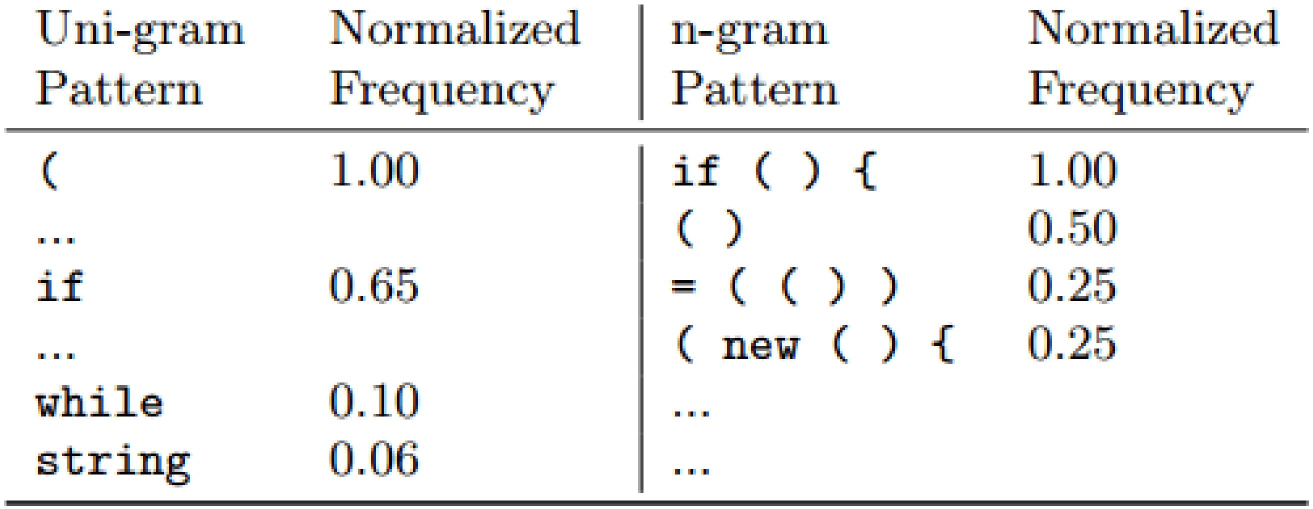
Discovery



Input: Set of Seed Entities:
{collection, datatype, ...}
I
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
...
III
A
PoS Type Detection
B
PoS Pattern Mining
C
Candidate Entity Mining
Entity Discovery
PoS Types:
collection --> [NNP, NN],
datatype --> [VB, JJ, NNP, FW, NN],
...}
II
Ouput: Set of Discovered Entities:
{array, pointer, type, ...}
IV

Profile Construction

- n-gram TF-IDF over source code



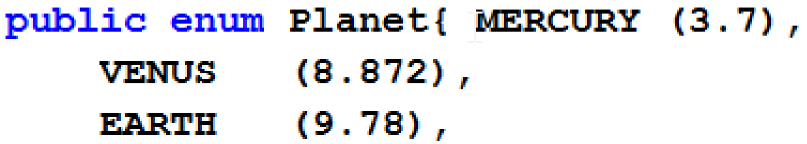
Uni-gram	Normalized	n-gram	Normalized
Pattern	Frequency	Pattern	Frequency
`(`	1.00	`if ( ) {`	1.00
...		`( )`	0.50
`if`	0.65	`= ( ( ) )`	0.25
...		`( new ( ) {`	0.25
`while`	0.10	...	
`string`	0.06	...	

Patterns for “conditional”

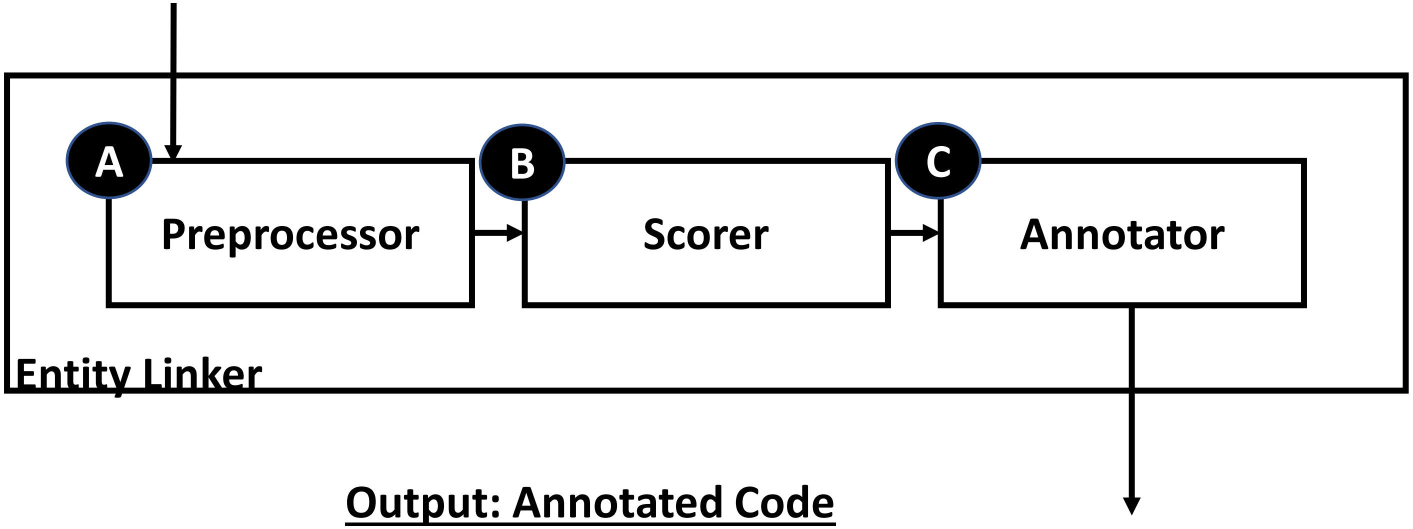
What is the intuition?
http://irpower.herokuapp.com/

Linking

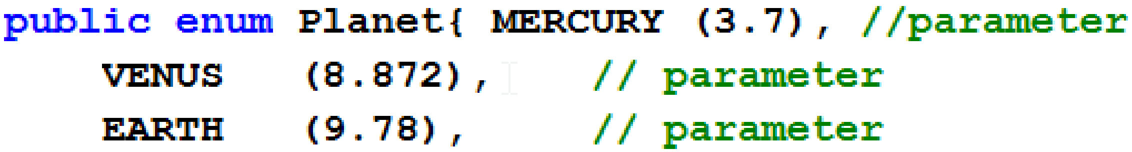
Input: Code Snippet



```
public enum Planet{ MERCURY (3.7),
     VENUS    (8.872),
     EARTH    (9.78),
```



Output: Annotated Code



```
public enum Planet{ MERCURY (3.7), //parameter
     VENUS    (8.872),     // parameter
     EARTH    (9.78),      // parameter
```

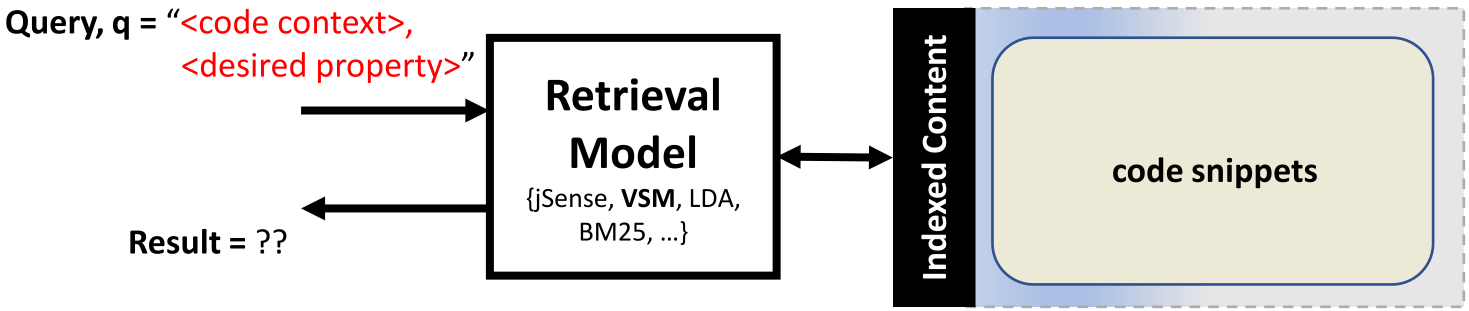
Results

- 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>”

Result = ??

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

Indexed Content

code snippets

Snippet1

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

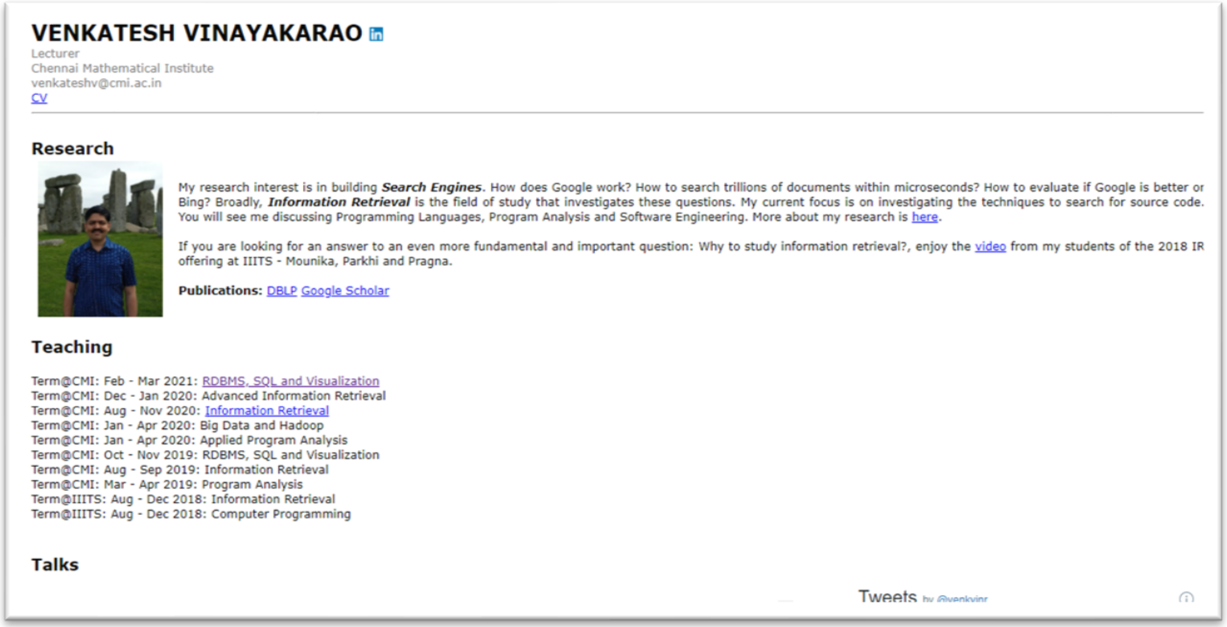
Snippet2

```
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
- Venkatesh Vinayakarao, Anita Sarma, Rahul Purandare, Shuktika Jain and Saumya Jain. ANNE: Improving Source Code Search using Entity Retrieval Approach. In the Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (**WSDM 2017**), UK.
- Venkatesh Vinayakarao. Spotting familiar code snippet structures for program comprehension. In the Proceedings of the 2015 10th Meeting on Foundations of Software Engineering, (**ESEC/FSE 2015**), Italy.



VENKATESH VINAYAKARAO

Lecturer
Chennai Mathematical Institute
venkateshv@cmi.ac.in
CV

Research

My research interest is in building ***Search Engines***. How does Google work? How to search trillions of documents within microseconds? How to evaluate if Google is better or Bing? Broadly, ***Information Retrieval*** is the field of study that investigates these questions. My current focus is on investigating the techniques to search for source code. You will see me discussing Programming Languages, Program Analysis and Software Engineering. More about my research is here.

If you are looking for an answer to an even more fundamental and important question: Why to study information retrieval?, enjoy the video from my students of the 2018 IR offering at IIITS - Mounika, Parkhi and Pragna.

Publications: DBLP Google Scholar

Teaching

Term@CMI: Feb - Mar 2021: RDBMS, SQL and Visualization
Term@CMI: Dec - Jan 2020: Advanced Information Retrieval
Term@CMI: Aug - Nov 2020: Information Retrieval
Term@CMI: Jan - Apr 2020: Big Data and Hadoop
Term@CMI: Jan - Apr 2020: Applied Program Analysis
Term@CMI: Oct - Nov 2019: RDBMS, SQL and Visualization
Term@CMI: Aug - Sep 2019: Information Retrieval
Term@CMI: Mar - Apr 2019: Program Analysis
Term@IIITS: Aug - Dec 2018: Information Retrieval
Term@IIITS: Aug - Dec 2018: Computer Programming

Talks

Thank You

venkateshv@cmi.ac.in

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