SOURCERER: MINING AND SEARCHING INTERNET-SCALE SOFTWARE REPOSITORIES

Introduction to
Information Retrieval
CS 150
Donald J. Patterson

This content based on the paper located here:
http://dx.doi.org/10.1007/s10618-008-0118-x
and slides located http://bit.ly/9CEaaT

Sourcerer: mining and searching internet-scale software repositories

Data Min Knowl Disc (2009) 18:300–336
DOI 10.1007/s10618-008-0118-x

Sourcerer: mining and searching internet-scale software repositories

Erik Linstead · Sushil Bajracharya · Trung Ngo ·
Paul Rigor · Cristina Lopes · Pierre Baldi

Received: 16 September 2007 / Accepted: 18 September 2008 / Published online: 15 October 2008

Abstract Large repositories of source code available over the Internet, or within large organizations, create new challenges and opportunities for data mining and statistical machine learning. Here we first develop Sourcerer, an infrastructure for the automated crawling, parsing, fingerprinting, and database storage of open source software on an Internet-scale. In one experiment, we gather 4,632 Java projects from SourceForge and Apache totaling over 38 million lines of code from 9,250 developers. Simple statistical analyses of the data first reveal robust power-law behavior for package, method call, and lexical containment distributions. We then develop and apply unsupervised, probabilistic, topic and author-topic (AT) models to automatically

Responsible editor: Eamonn Keogh.

Erik Linstead, Sushil Bajracharya, and Trung Ngo have contributed equally to this work.

E. Linstead · S. Bajracharya · T. Ngo · P. Rigor · C. Lopes · P. Baldi (✉)
Donald Bren School of Information and Computer Sciences, University of California, Irvine, USA
e-mail: pfbaldi@ics.uci.edu

E. Linstead
e-mail: elinstea@ics.uci.edu

S. Bajracharya
e-mail: sbajrach@ics.uci.edu

T. Ngo
e-mail: trungcn@ics.uci.edu

P. Rigor
e-mail: prigor@ics.uci.edu

C. Lopes
e-mail: lopes@ics.uci.edu

Springer

Sourcerer: mining and searching internet-scale software repositories

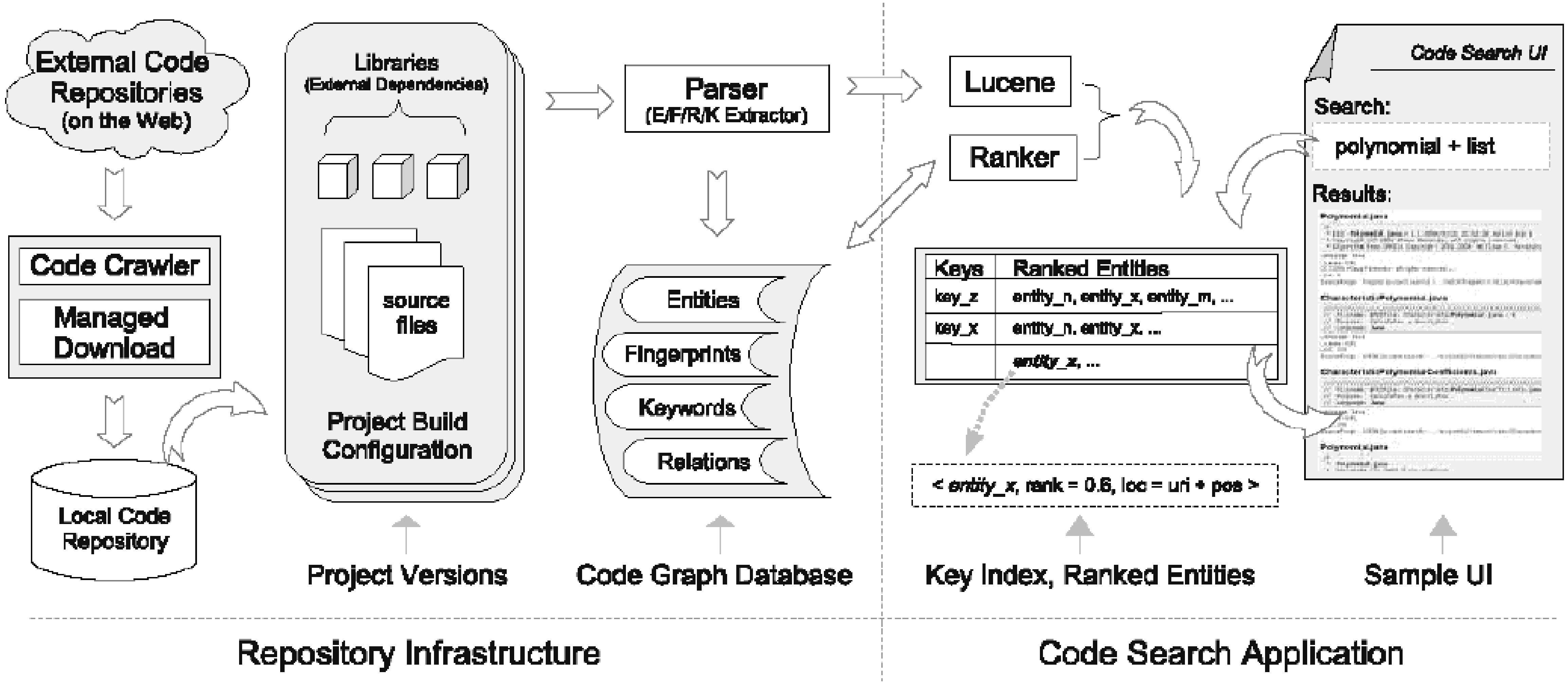
- Why mine source code?
  - to understand engineering and development
  - to understand complexity
  - to improve code reuse
  - to identify relationships between humans and their code
- Code should not be treated as text
  - There is a lot of structure that can be exploited

Sourcerer: mining and searching internet-scale software repositories

- Sourcerer is
  - a crawler of software repositories
  - a parser and feature extractor for code
  - a fingerprinter
  - a database
  - a web search interface
- for Java Source code

Sourcerer: mining and searching internet-scale software repositories

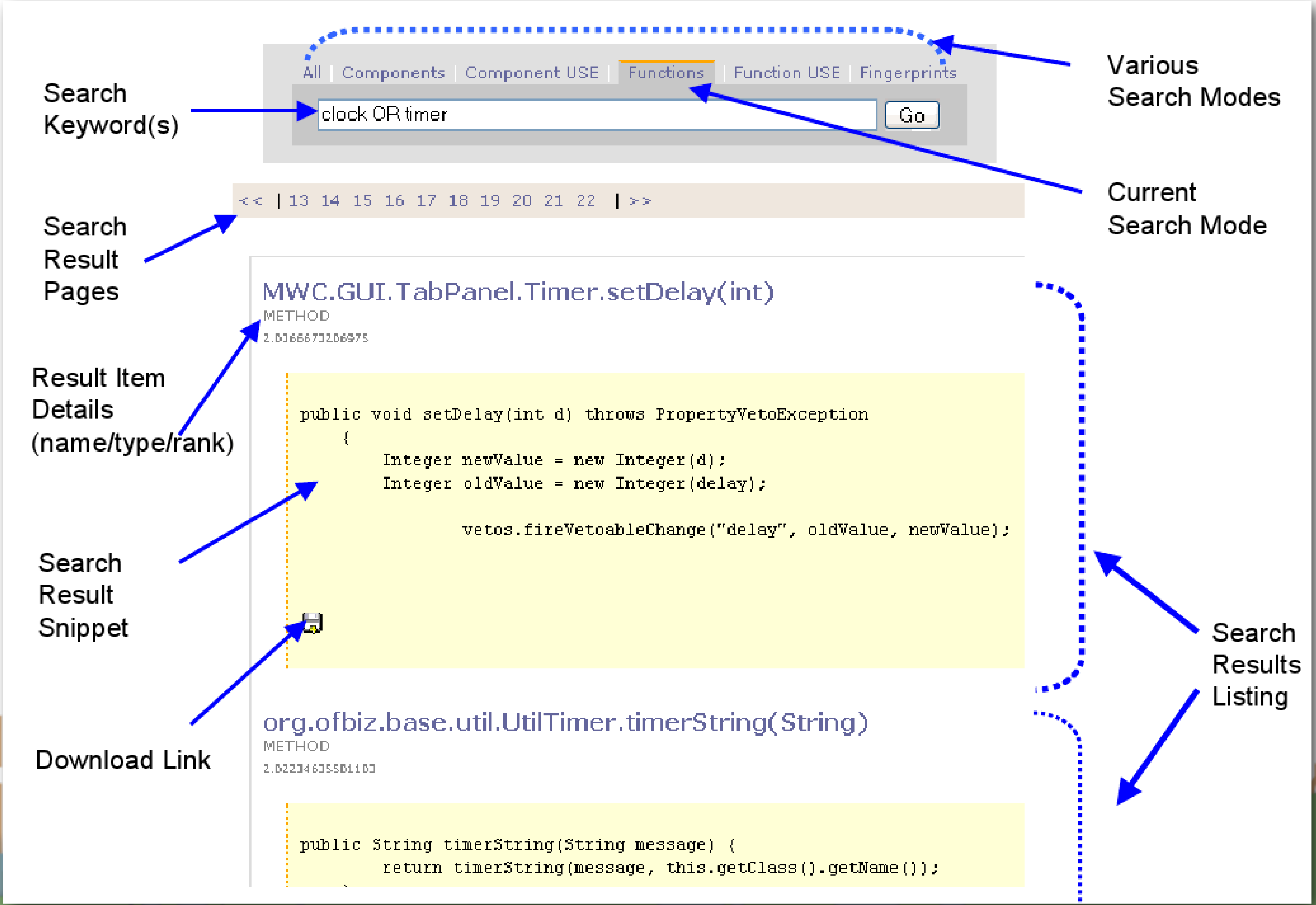
- Sourcerer architecture



Keys	Ranked Entities
key_z	entity_n, entity_x, entity_m, ...
key_x	entity_n, entity_x, ...
	entity_x, ...

Sourcerer: mining and searching internet-scale software repositories

- Sourcerer search interface



clock OR timer

MWC.GUI.TabPanel.Timer.setDelay(int)

METHOD

2.0366673206975

```
public void setDelay(int d) throws PropertyVetoException
     {
          Integer newValue = new Integer(d);
          Integer oldValue = new Integer(delay);

                    vetos.fireVetoableChange("delay", oldValue, newValue);
```

org.ofbiz.base.util.UtilTimer.timerString(String)

METHOD

2.02234635501103

```
public String timerString(String message) {
          return timerString(message, this.getClass().getName());
```

Search Keyword(s)

Search Result Pages

Result Item Details (name/type/rank)

Search Result Snippet

Download Link

Various Search Modes

Current Search Mode

Search Results Listing

Sourcerer: mining and searching internet-scale software repositories

- Parsing
  - Entities

    Package
    Class
    Method
    Field
    Constructor
    Static initializer

  - Relations

Inside	Lexical encapsulation of one entity inside another
Use	One relation uses another to achieve functionality
Extends	One class subclasses another
Implements	A class implements a given interface
Calls	One method calls another
Throws	One entity throws another as an exception
Returns	A method returns an entity
Overrides	A class overrides a method
Overload	One entity overloads a method
Instantiates	One entity instantiates another via the ‘new’ keyword
Assigned	A method call assigns a value to a field
Holds	A field holds an entity of a given type
Receives	A method receives an entity as an input parameter
Accesses	An entity reads a field

SORCERER

CRAWLING

- When the crawler finds source code, it extracts the entity and stores:
    - Fully qualified domain name (FQN)
    - Document, repository and version where it was found
    - Position and length of the entity in the source

- How does this compare to source crawling?

SORCERER

CRAWLING

- When the crawler finds source code, it extracts the entity and stores:
  - Fully qualified domain name (FQN)
  - Document, repository and version where it was found
  - Position and length of the entity in the source

- How does this compare to source crawling?
  - Like (url, term, count) tuples

SORCERER

CRAWLING

- When the crawler finds source code, it extracts the entity and stores:
  - Relations between entities

- How does this compare to source crawling?

SORCERER

CRAWLING

- When the crawler finds source code, it extracts the entity and stores:
  - Relations between entities

- How does this compare to source crawling?
  - Like keeping track of links between pages

CRAWLING

- When the crawler finds source code, it extracts the entity and stores:
    - Keywords in the entity
    - Fingerprints of the entity

- How does this compare to source crawling?

CRAWLING

- When the crawler finds source code, it extracts the entity and stores:
  - Keywords in the entity
  - Fingerprints of the entity
- How does this compare to source crawling?
  - Like the data needed for creating snippets

Sourcerer: mining and searching internet-scale software repositories

- Keyword Extraction
  - Comments
  - Splits on Case
    - QuickSort -> “Quick” “Sort”
  - Mapped to entities

Sourcerer: mining and searching internet-scale software repositories

- Fingerprinting Source Code
  - Structure-based search requires a compact representation of code characteristics
  - “Fingerprints” are vectors whose elements denote the occurrence of specific programming constructs
  - Easily lends itself to the vector model of standard information retrieval
  - Fingerprints must balance efficiency and expressiveness
    - Feature set must be rich enough to be meaningful
    - Superfluous features add to computational overhead

Sourcerer: mining and searching internet-scale software repositories

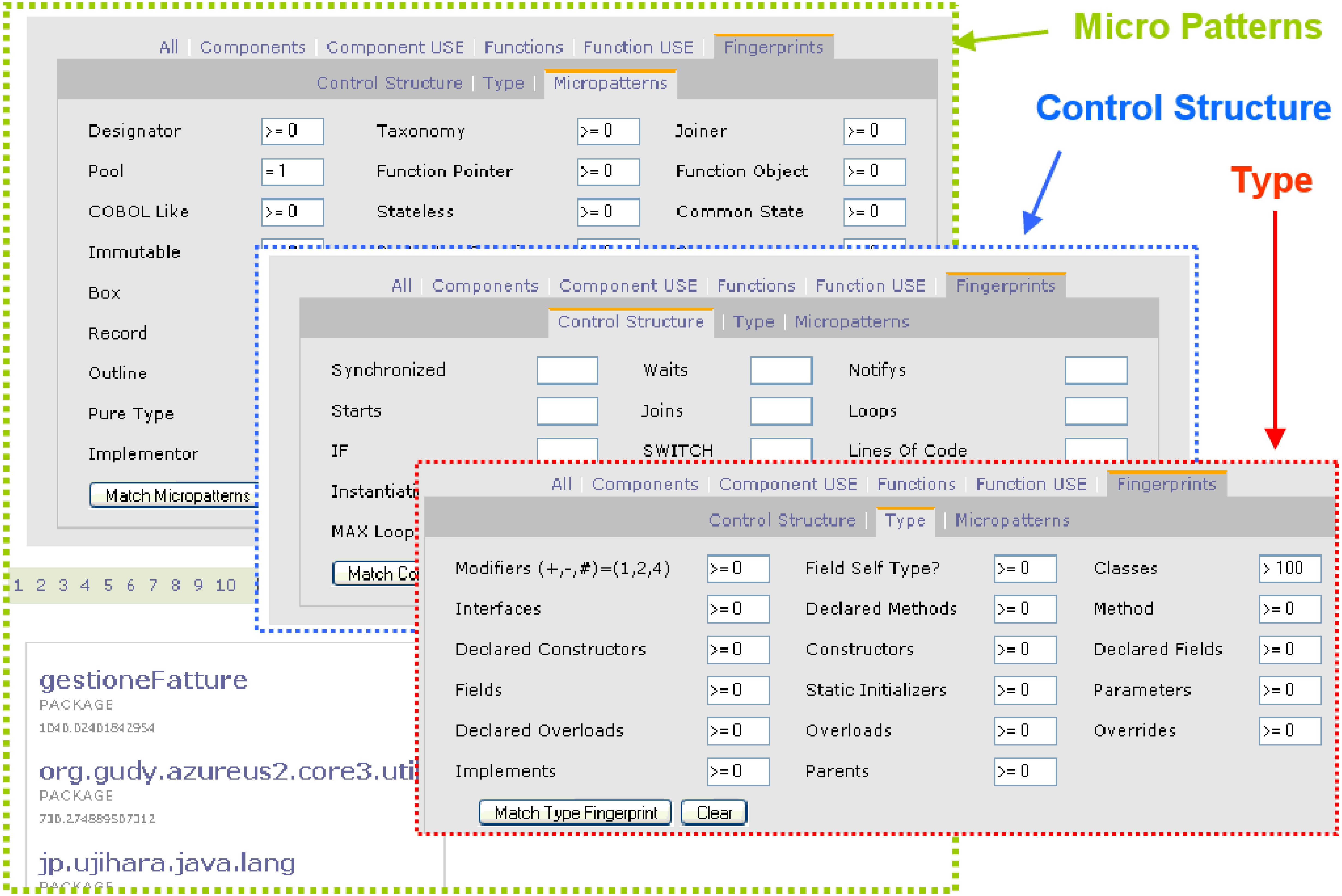
- Fingerprinting types
  - Control Structure Prints
    - Provides information about concurrency, iteration, and conditional constructs
    - Useful for identifying benchmark datasets
  - Java Type Prints
    - Captures information about object-oriented constructs (classes, methods, fields, constructors, etc)
    - Provides capability for general entity structure search

Sourcerer: mining and searching internet-scale software repositories

- Fingerprinting types
  - Micro Pattern Prints
    - Bit vector indicating occurrence of simple design patterns in code entities
    - Allows for structure-based search based on commonly occurring design practices

Sourcerer: mining and searching internet-scale software repositories

- Fingerprint Search



Micro Patterns

Control Structure

Type

All | Components | Component USE | Functions | Function USE | Fingerprints
Control Structure | Type | Micropatterns

Designator	>= 0	Taxonomy	>= 0	Joiner	>= 0
Pool	= 1	Function Pointer	>= 0	Function Object	>= 0
COBOL Like	>= 0	Stateless	>= 0	Common State	>= 0
Immutable					
Box					
Record					
Outline					
Pure Type					
Implementor					

Match Micropatterns
1 2 3 4 5 6 7 8 9 10

gestioneFatture
PACKAGE
1040.0240184 2954

org.gudy.azureus2.core3.uti
PACKAGE
710.27488958 7312

jp.ujihara.java.lang
PACKAGE

All | Components | Component USE | Functions | Function USE | Fingerprints
Control Structure | Type | Micropatterns

Synchronized		Waits		Notifys	
Starts		Joins		Loops	
IF		SWITCH		Lines Of Code	
Instantiati					
MAX Loop					

All | Components | Component USE | Functions | Function USE | Fingerprints
Control Structure | Type | Micropatterns

Modifiers (+,-,#)=(1,2,4)	>= 0	Field Self Type?	>= 0	Classes	> 100
Interfaces	>= 0	Declared Methods	>= 0	Method	>= 0
Declared Constructors	>= 0	Constructors	>= 0	Declared Fields	>= 0
Fields	>= 0	Static Initializers	>= 0	Parameters	>= 0
Declared Overloads	>= 0	Overloads	>= 0	Overrides	>= 0
Implements	>= 0	Parents	>= 0		

Sourcerer: mining and searching internet-scale software repositories

- Ranking
  - Return code that is
    - keyword relevant
    - structure relevant
    - frequently used
    - robust
  - Determine importance of entities by applying PageRank
    - to source code
    - probabilistic framework for ranking

Sourcerer: mining and searching internet-scale software repositories

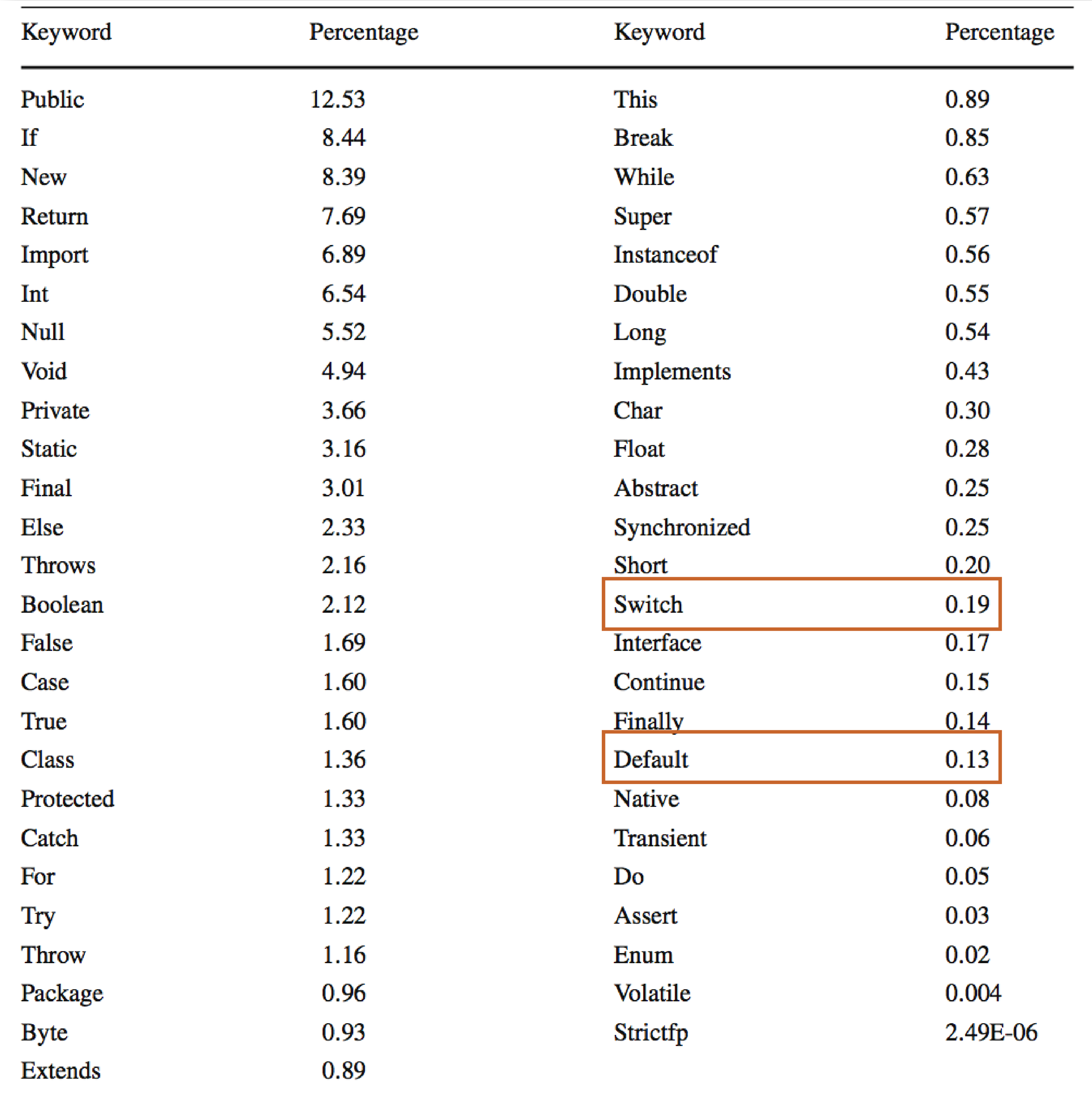
- Ranking
  - CodeRank can be tuned for
    - Project local ranking
    - Project global ranking
    - Relationship-specific Ranking
      - Increasing the weight of relevant edges in dependency graph

CURRENT SOURCERER STATISTICS

- Repository
  - Total number of projects (with source): 4632
  - Total source files: 244,342
  - Total lines of code: 38,700,000
  - Number of developers: 9,250
  - Number of entities: 5,000,000
    - 47,640 packages
    - 560,669 classes
    - 3,205,741 methods
    - 23,400,000 relations

Sourcerer: mining and searching internet-scale software repositories

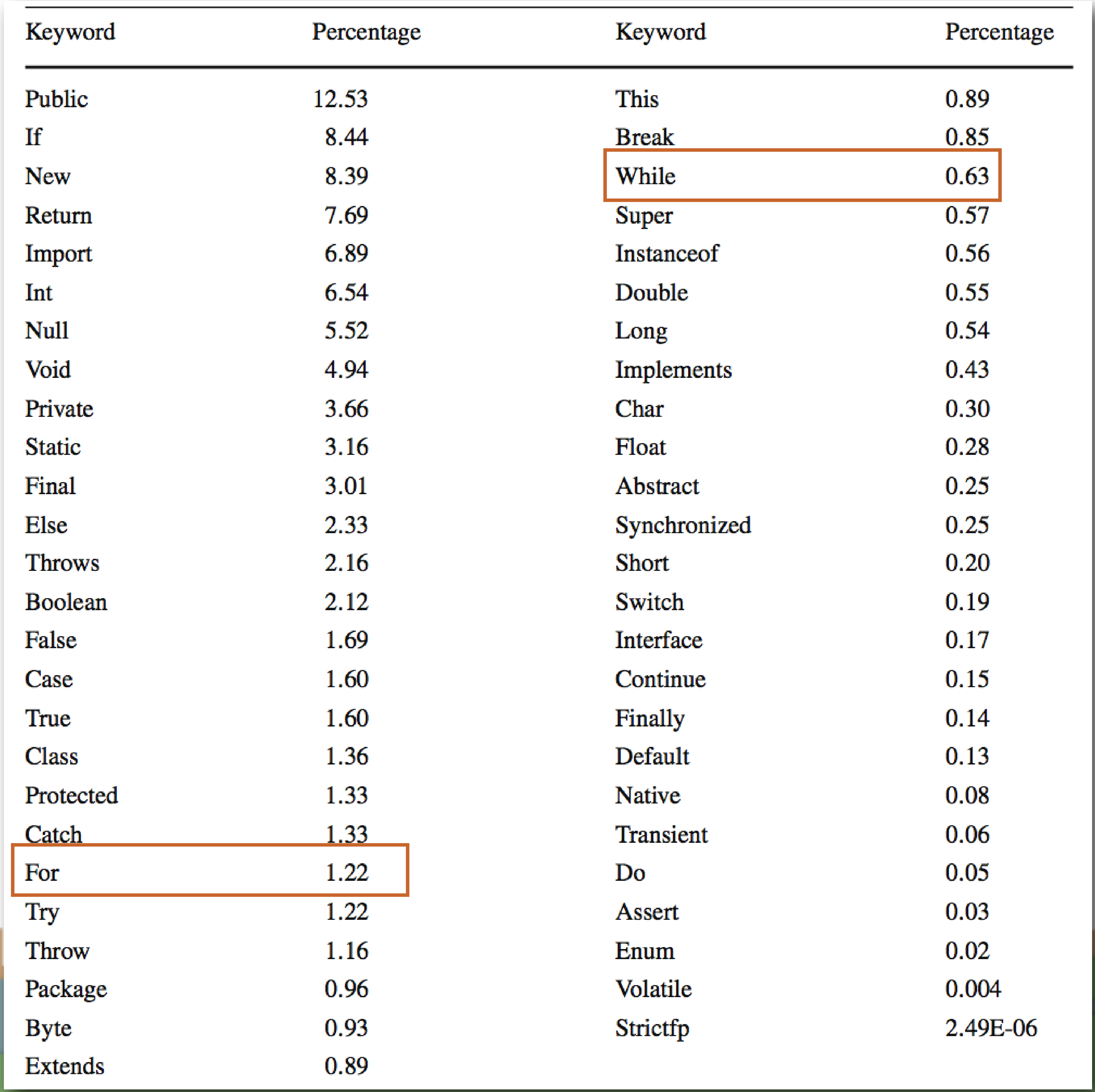
- Keyword frequency (%)



Keyword	Percentage	Keyword	Percentage
Public	12.53	This	0.89
If	8.44	Break	0.85
New	8.39	While	0.63
Return	7.69	Super	0.57
Import	6.89	Instanceof	0.56
Int	6.54	Double	0.55
Null	5.52	Long	0.54
Void	4.94	Implements	0.43
Private	3.66	Char	0.30
Static	3.16	Float	0.28
Final	3.01	Abstract	0.25
Else	2.33	Synchronized	0.25
Throws	2.16	Short	0.20
Boolean	2.12	Switch	0.19
False	1.69	Interface	0.17
Case	1.60	Continue	0.15
True	1.60	Finally	0.14
Class	1.36	Default	0.13
Protected	1.33	Native	0.08
Catch	1.33	Transient	0.06
For	1.22	Do	0.05
Try	1.22	Assert	0.03
Throw	1.16	Enum	0.02
Package	0.96	Volatile	0.004
Byte	0.93	Strictfp	2.49E-06
Extends	0.89		

Sourcerer: mining and searching internet-scale software repositories

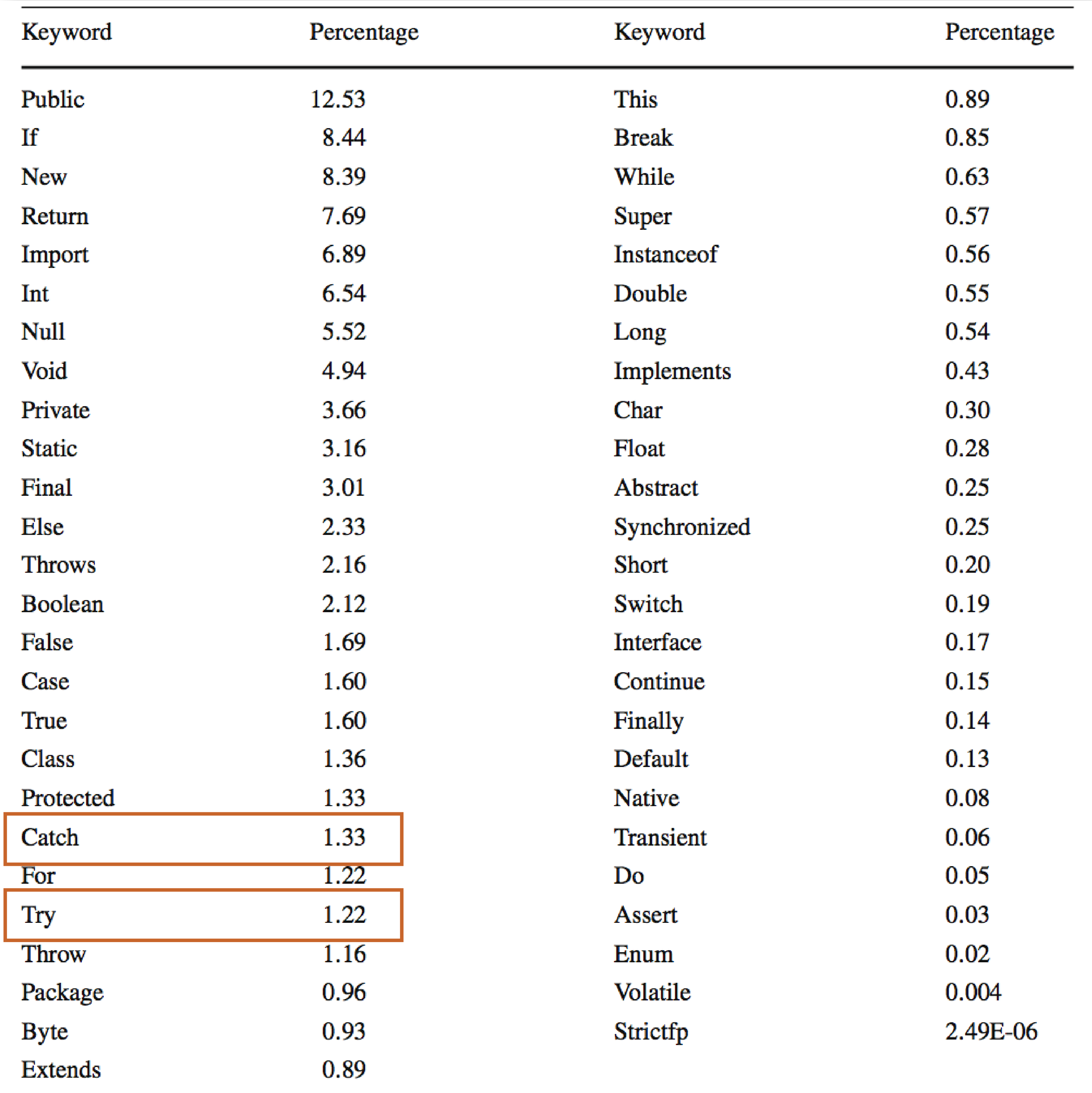
- Keyword frequency (%)



Keyword	Percentage	Keyword	Percentage
Public	12.53	This	0.89
If	8.44	Break	0.85
New	8.39	While	0.63
Return	7.69	Super	0.57
Import	6.89	Instanceof	0.56
Int	6.54	Double	0.55
Null	5.52	Long	0.54
Void	4.94	Implements	0.43
Private	3.66	Char	0.30
Static	3.16	Float	0.28
Final	3.01	Abstract	0.25
Else	2.33	Synchronized	0.25
Throws	2.16	Short	0.20
Boolean	2.12	Switch	0.19
False	1.69	Interface	0.17
Case	1.60	Continue	0.15
True	1.60	Finally	0.14
Class	1.36	Default	0.13
Protected	1.33	Native	0.08
Catch	1.33	Transient	0.06
For	1.22	Do	0.05
Try	1.22	Assert	0.03
Throw	1.16	Enum	0.02
Package	0.96	Volatile	0.004
Byte	0.93	Strictfp	2.49E-06
Extends	0.89		

Sourcerer: mining and searching internet-scale software repositories

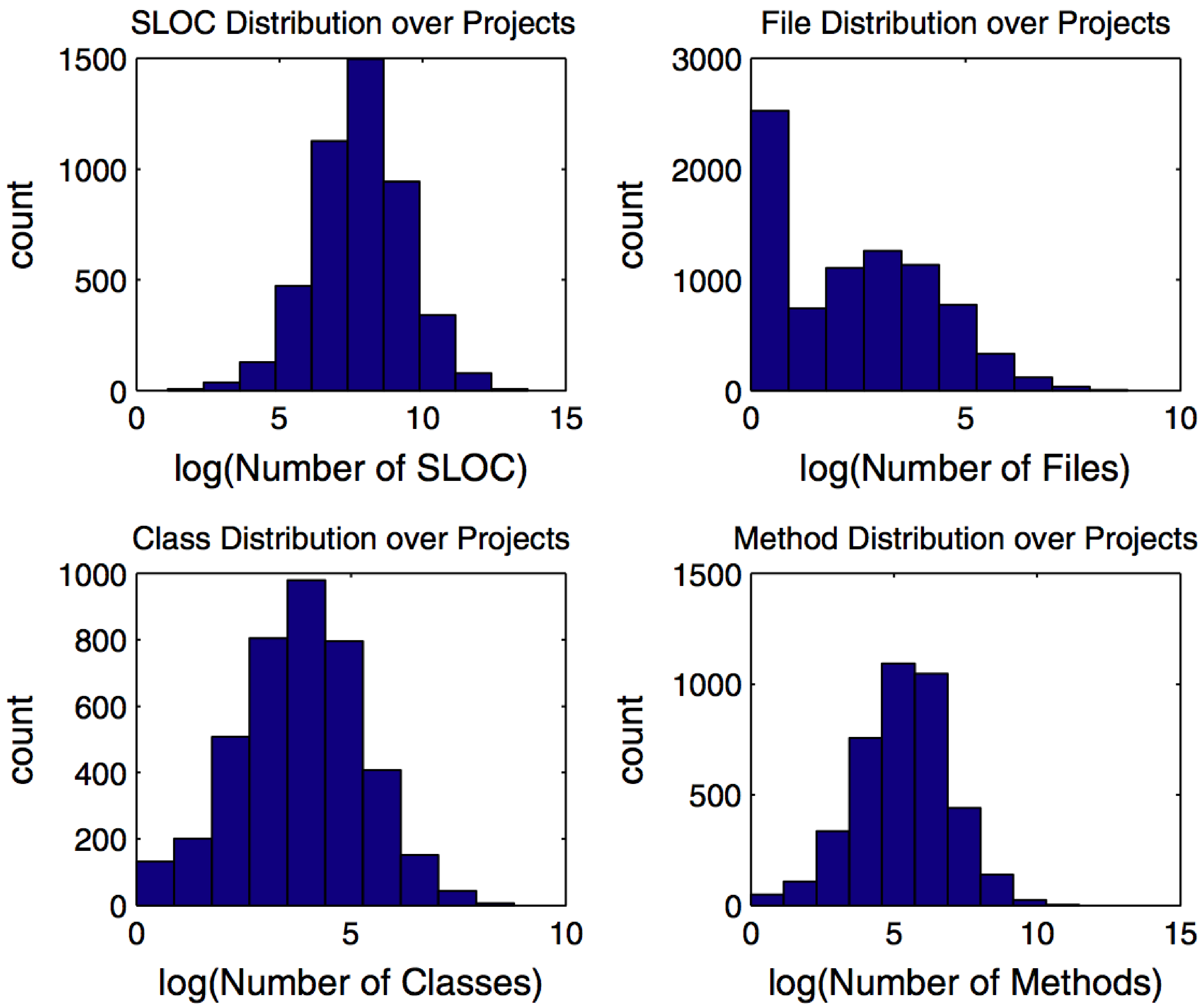
- Keyword frequency (%)



Keyword	Percentage	Keyword	Percentage
Public	12.53	This	0.89
If	8.44	Break	0.85
New	8.39	While	0.63
Return	7.69	Super	0.57
Import	6.89	Instanceof	0.56
Int	6.54	Double	0.55
Null	5.52	Long	0.54
Void	4.94	Implements	0.43
Private	3.66	Char	0.30
Static	3.16	Float	0.28
Final	3.01	Abstract	0.25
Else	2.33	Synchronized	0.25
Throws	2.16	Short	0.20
Boolean	2.12	Switch	0.19
False	1.69	Interface	0.17
Case	1.60	Continue	0.15
True	1.60	Finally	0.14
Class	1.36	Default	0.13
Protected	1.33	Native	0.08
Catch	1.33	Transient	0.06
For	1.22	Do	0.05
Try	1.22	Assert	0.03
Throw	1.16	Enum	0.02
Package	0.96	Volatile	0.004
Byte	0.93	Strictfp	2.49E-06
Extends	0.89		

Sourcerer: mining and searching internet-scale software repositories

- Code characteristics



SOURCERER

AUTHOR-TOPIC MODELS

$$P(d|\Theta, \Phi, \mathcal{A}) = \prod_{i=1}^{N_d} \frac{1}{A_d} \sum_{a} \sum_{t=1}^{T} \phi_{w_i t} \theta_{ta}$$

SOURCERER

AUTHOR-TOPIC MODELS

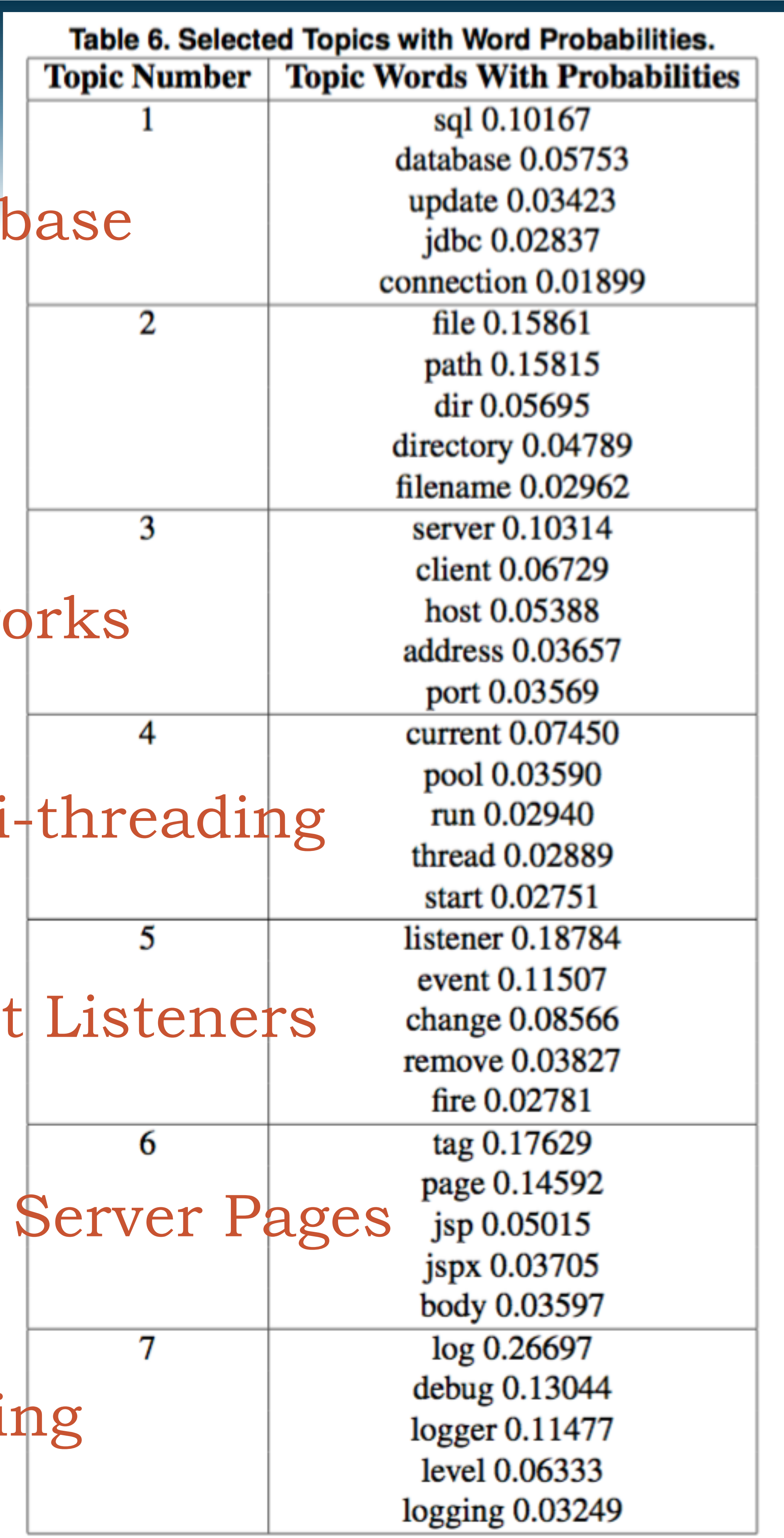


Table 6. Selected Topics with Word Probabilities.

Topic Number	Topic Words With Probabilities
1	sql 0.10167 database 0.05753 update 0.03423 jdbc 0.02837 connection 0.01899
2	file 0.15861 path 0.15815 dir 0.05695 directory 0.04789 filename 0.02962
3	server 0.10314 client 0.06729 host 0.05388 address 0.03657 port 0.03569
4	current 0.07450 pool 0.03590 run 0.02940 thread 0.02889 start 0.02751
5	listener 0.18784 event 0.11507 change 0.08566 remove 0.03827 fire 0.02781
6	tag 0.17629 page 0.14592 jsp 0.05015 jspx 0.03705 body 0.03597
7	log 0.26697 debug 0.13044 logger 0.11477 level 0.06333 logging 0.03249

Database

Files

Networks

Multi-threading

Event Listeners

Java Server Pages

Logging

SOURCERER

AUTHOR-TOPIC MODELS

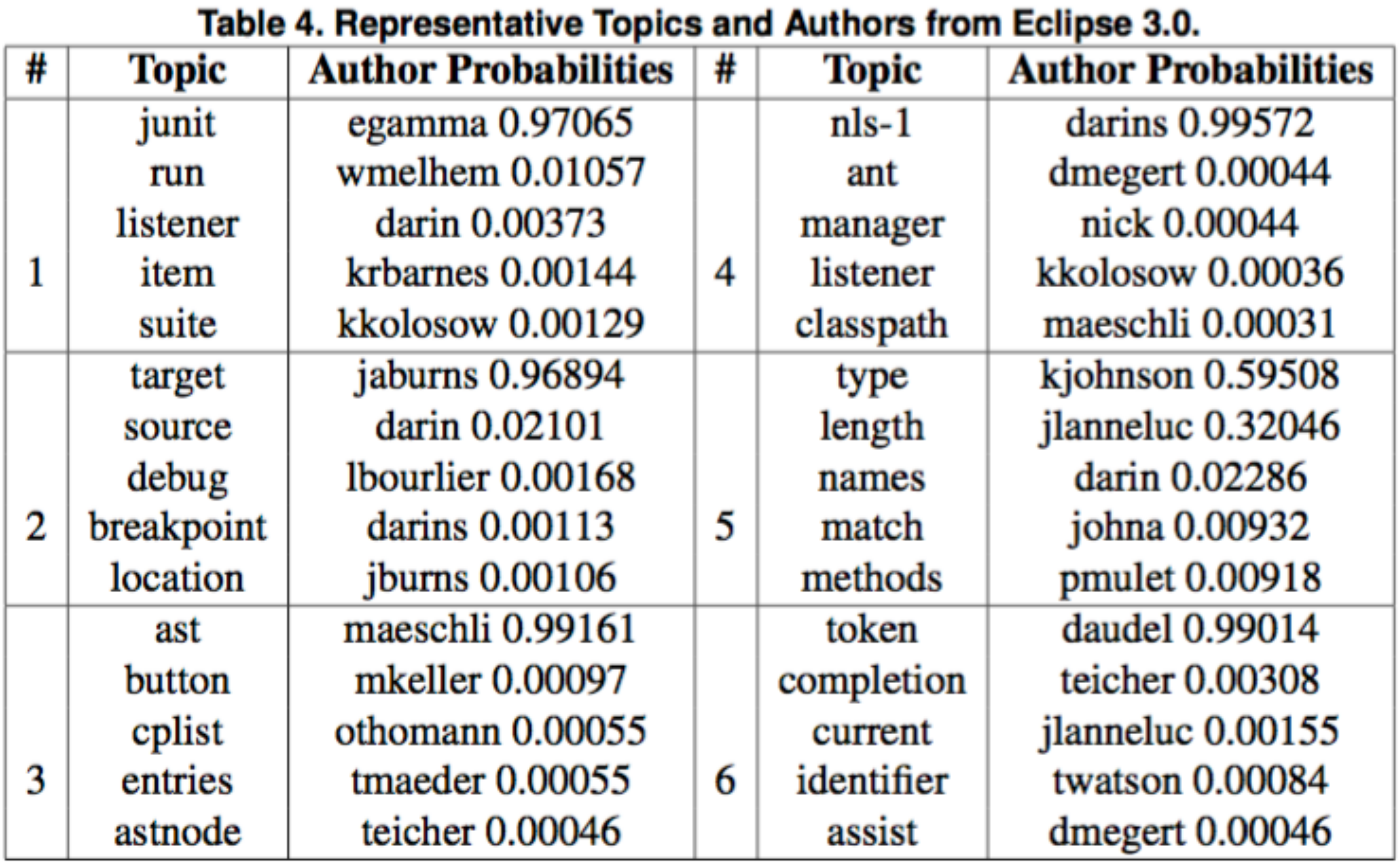


Table 4. Representative Topics and Authors from Eclipse 3.0.

#	Topic	Author Probabilities	#	Topic	Author Probabilities
1	junit	egamma 0.97065	4	nls-1	darins 0.99572
	run	wmelhem 0.01057		ant	dmegert 0.00044
	listener	darin 0.00373		manager	nick 0.00044
	item	krbarnes 0.00144		listener	kkolosow 0.00036
	suite	kkolosow 0.00129		classpath	maeschli 0.00031
2	target	jaburns 0.96894	5	type	kjohnson 0.59508
	source	darin 0.02101		length	jlanneluc 0.32046
	debug	lbourlier 0.00168		names	darin 0.02286
	breakpoint	darins 0.00113		match	johna 0.00932
	location	jburns 0.00106		methods	pmulet 0.00918
3	ast	maeschli 0.99161	6	token	daudel 0.99014
	button	mkeller 0.00097		completion	teicher 0.00308
	cplist	othomann 0.00055		current	jlanneluc 0.00155
	entries	tmaeder 0.00055		identifier	twatson 0.00084
	astnode	teicher 0.00046		assist	dmegert 0.00046

SOURCERER

AUTHOR-TOPIC MODELS

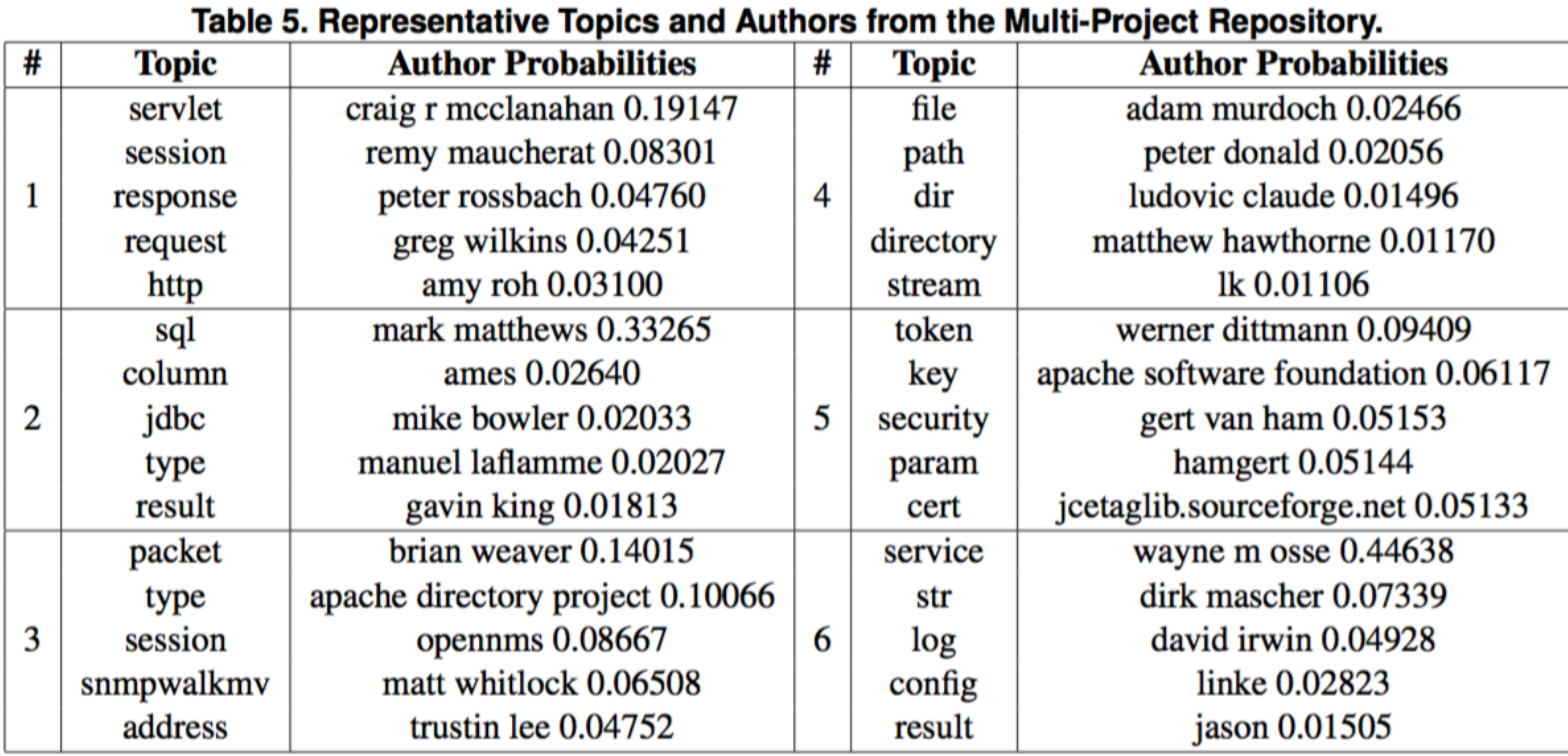


Table 5. Representative Topics and Authors from the Multi-Project Repository.

#	Topic	Author Probabilities	#	Topic	Author Probabilities
1	servlet	craig r mcclanahan 0.19147	4	file	adam murdoch 0.02466
	session	remy maucherat 0.08301		path	peter donald 0.02056
	response	peter rossbach 0.04760		dir	ludovic claude 0.01496
	request	greg wilkins 0.04251		directory	matthew hawthorne 0.01170
	http	amy roh 0.03100		stream	lk 0.01106
2	sql	mark matthews 0.33265	5	token	werner dittmann 0.09409
	column	ames 0.02640		key	apache software foundation 0.06117
	jdbc	mike bowler 0.02033		security	gert van ham 0.05153
	type	manuel laflamme 0.02027		param	hamgert 0.05144
	result	gavin king 0.01813		cert	jcetaglib.sourceforge.net 0.05133
3	packet	brian weaver 0.14015	6	service	wayne m osse 0.44638
	type	apache directory project 0.10066		str	dirk mascher 0.07339
	session	opennms 0.08667		log	david irwin 0.04928
	snmpwalkmv	matt whitlock 0.06508		config	linke 0.02823
	address	trustin lee 0.04752		result	jason 0.01505

SOURCERER

AUTHOR-TOPIC MODELS

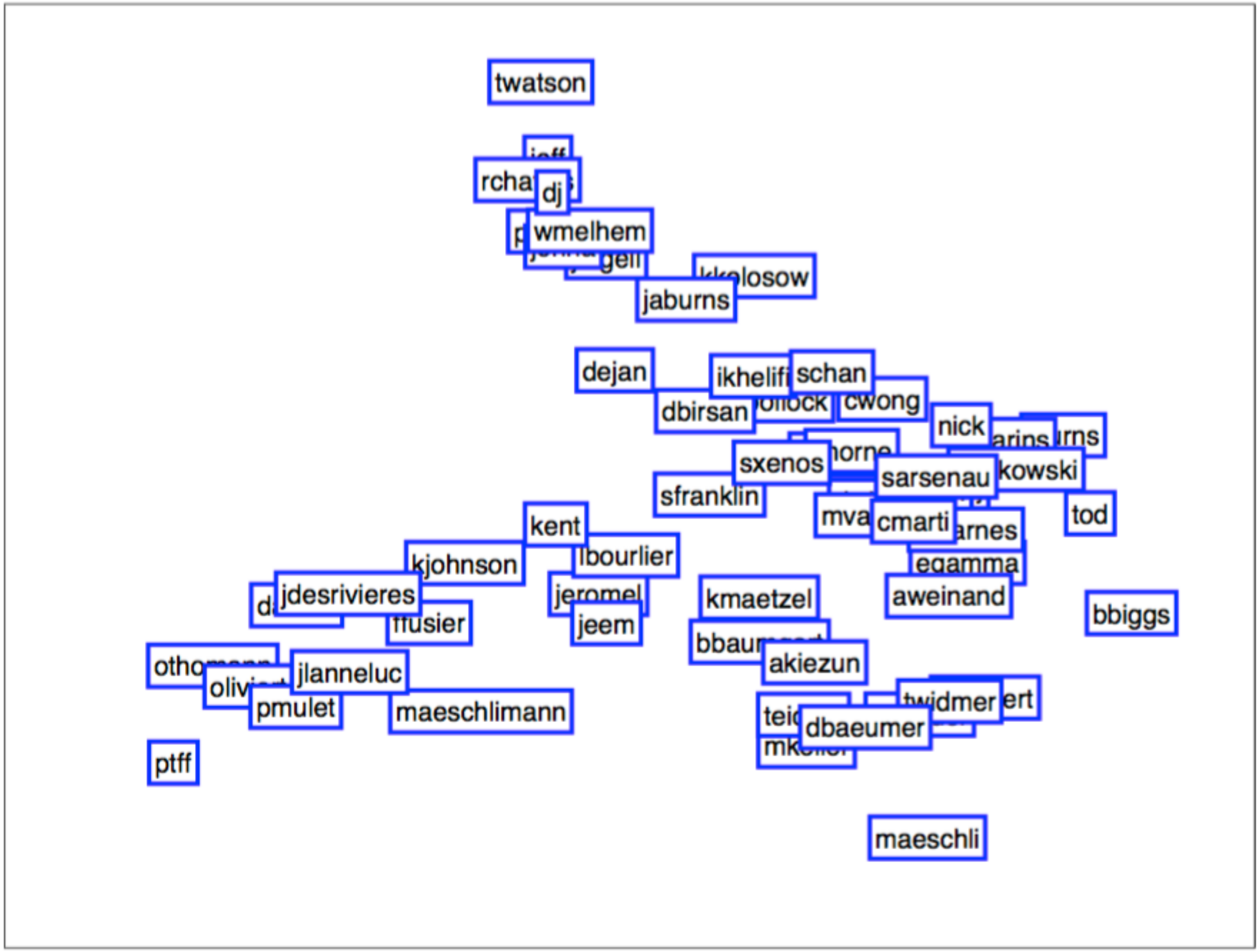


Figure 6. All 59 Eclipse 3.0 Authors Clustered by KL Divergence.

SOURCERER

EVALUATION

- Manually curated benchmark
  - queries (n = 25)
  - manual ranks (n = 3)
    - content corresponds to search intent
    - result is complete
    - reputation of source is high
    - ease of reuse is high
- Metrics
  - Precision
  - Recall
  - ROC/AUC

SOURCERER

EVALUATION

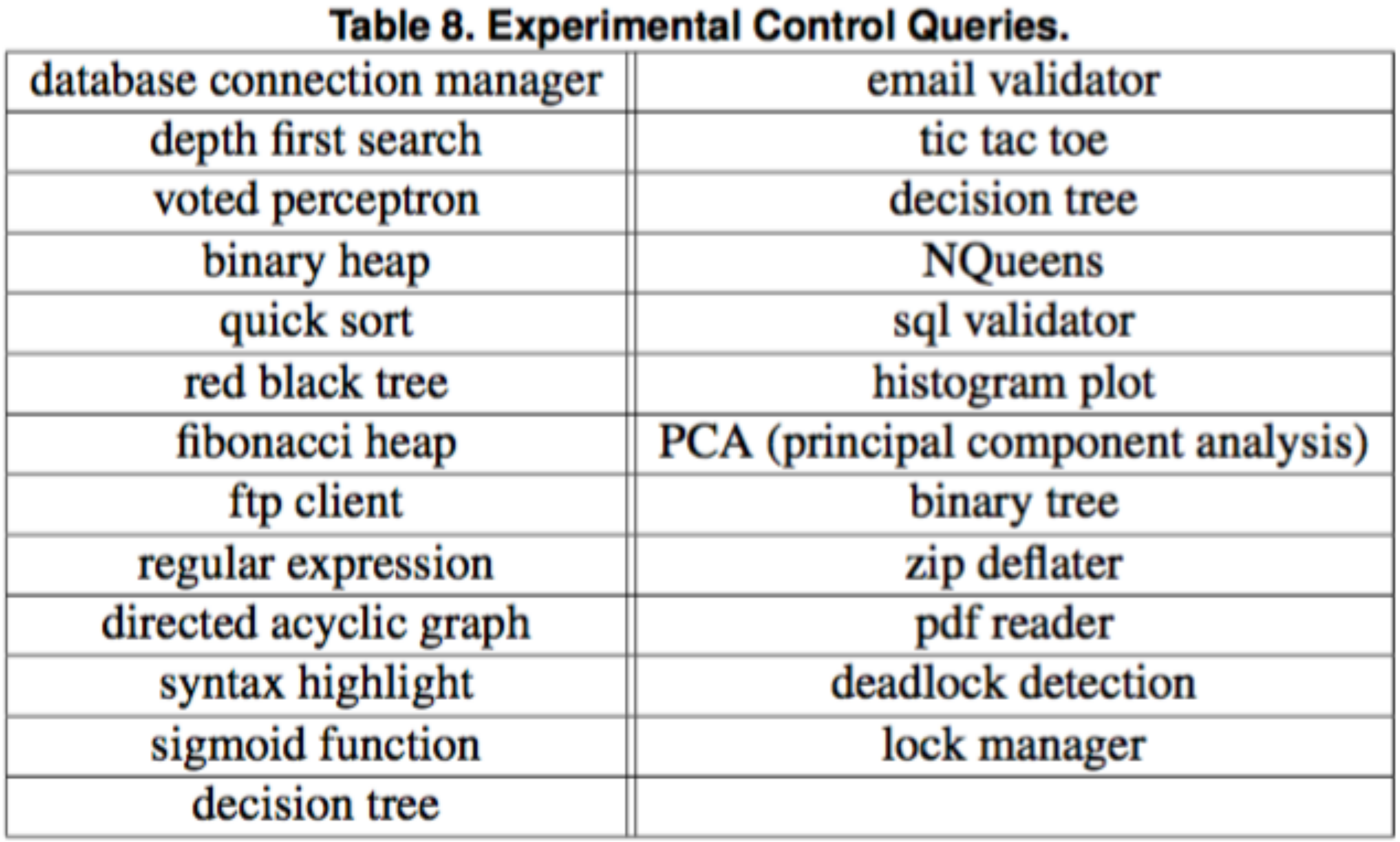
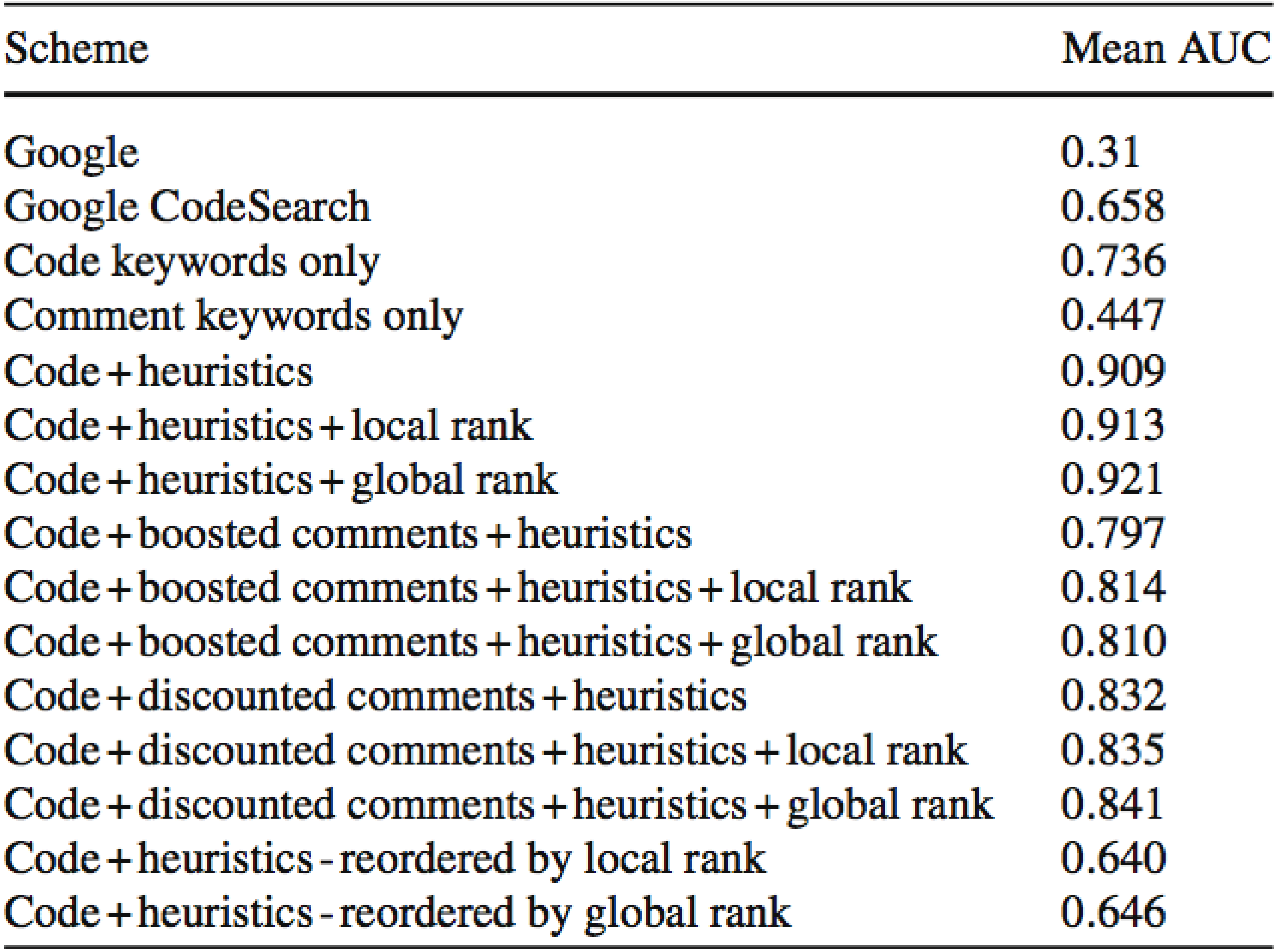


Table 8. Experimental Control Queries.

database connection manager	email validator
depth first search	tic tac toe
voted perceptron	decision tree
binary heap	NQueens
quick sort	sql validator
red black tree	histogram plot
fibonacci heap	PCA (principal component analysis)
ftp client	binary tree
regular expression	zip deflater
directed acyclic graph	pdf reader
syntax highlight	deadlock detection
sigmoid function	lock manager
decision tree	

Sourcerer: mining and searching internet-scale software repositories

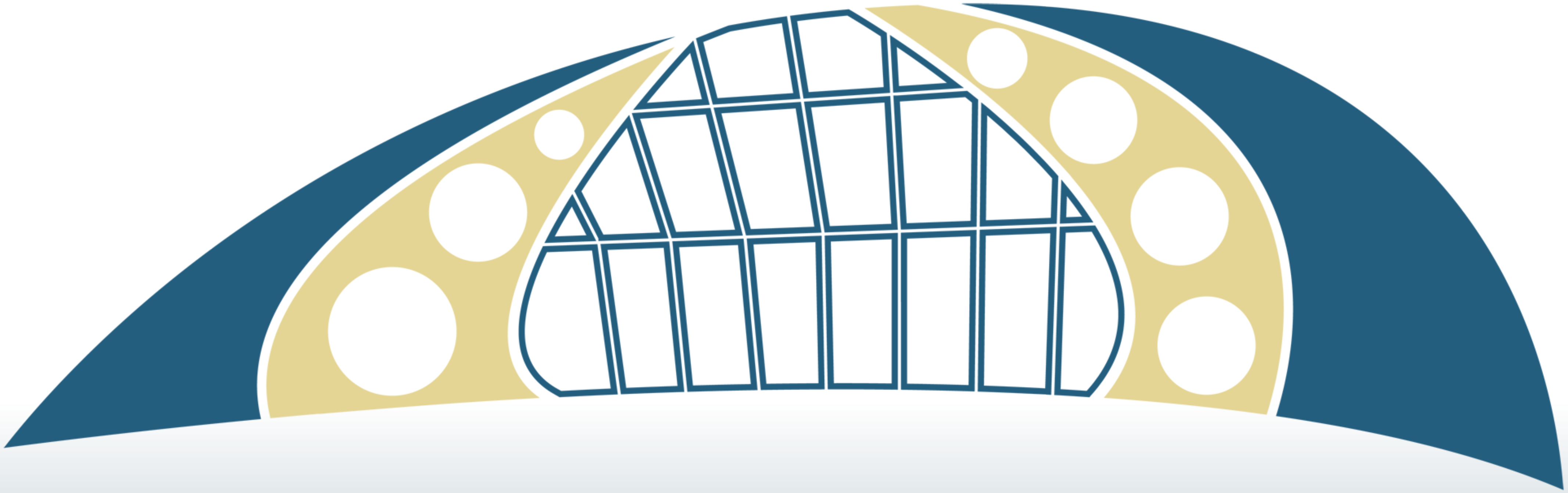
- Effectiveness of Search based on various code features



Scheme	Mean AUC
Google	0.31
Google CodeSearch	0.658
Code keywords only	0.736
Comment keywords only	0.447
Code + heuristics	0.909
Code + heuristics + local rank	0.913
Code + heuristics + global rank	0.921
Code + boosted comments + heuristics	0.797
Code + boosted comments + heuristics + local rank	0.814
Code + boosted comments + heuristics + global rank	0.810
Code + discounted comments + heuristics	0.832
Code + discounted comments + heuristics + local rank	0.835
Code + discounted comments + heuristics + global rank	0.841
Code + heuristics - reordered by local rank	0.640
Code + heuristics - reordered by global rank	0.646

CONCLUSION

- There is a lot of publicly available code
- It seems like it should help you write new code
- Finding applicable code is hard
- Sourcerer presents one way of doing it that is
  - scalable
  - innovative



WESTMONT INSPIRED

COMPUTING LAB