Sourcerer: mining and searching internet-scale software repositories

Introduction to Information Retrieval
CS 221
Donald J. Patterson

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

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

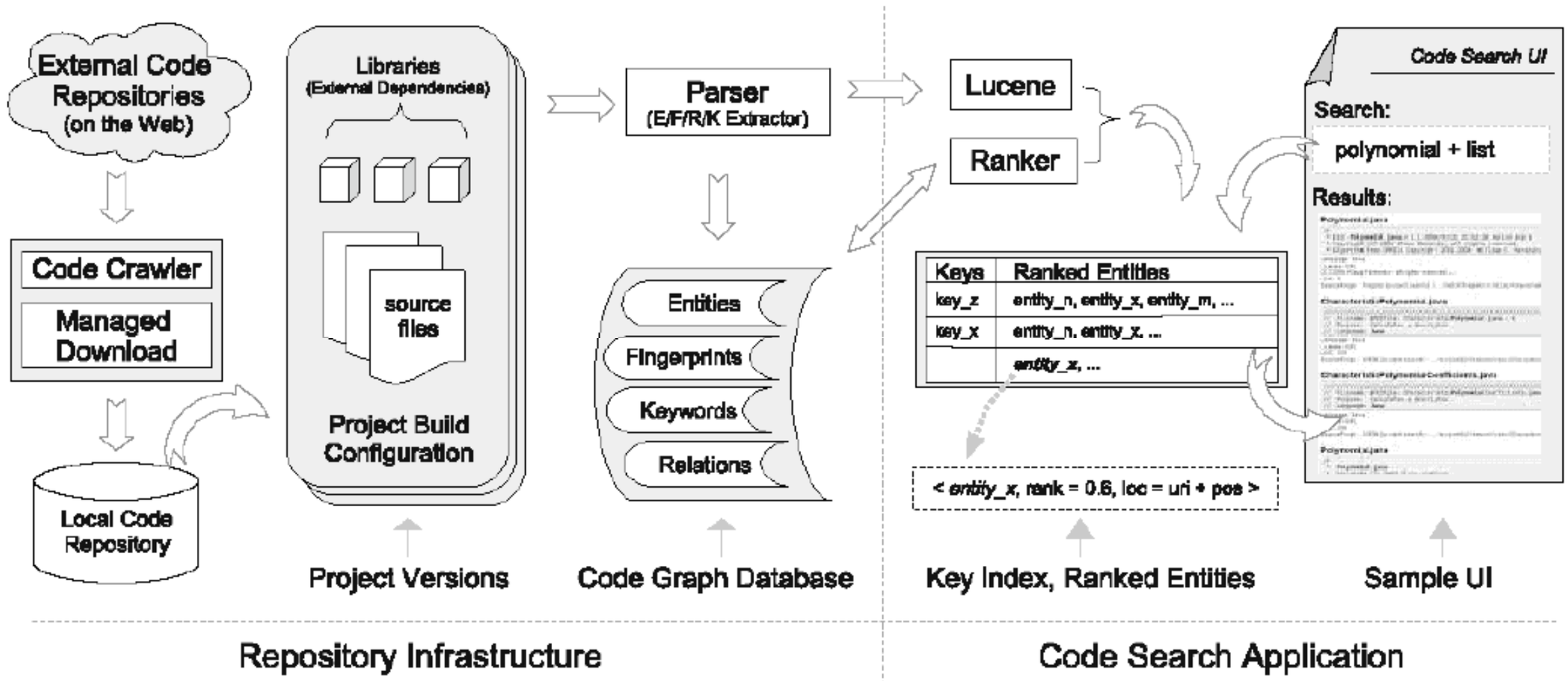
- Why mine source code?
  - to understand engineering and development
  - to understand complexity
  - to improve code reuse
  - to identify relationships between humans and their code
- Code should not be treated as text
  - There is a lot of structure that can be exploited

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

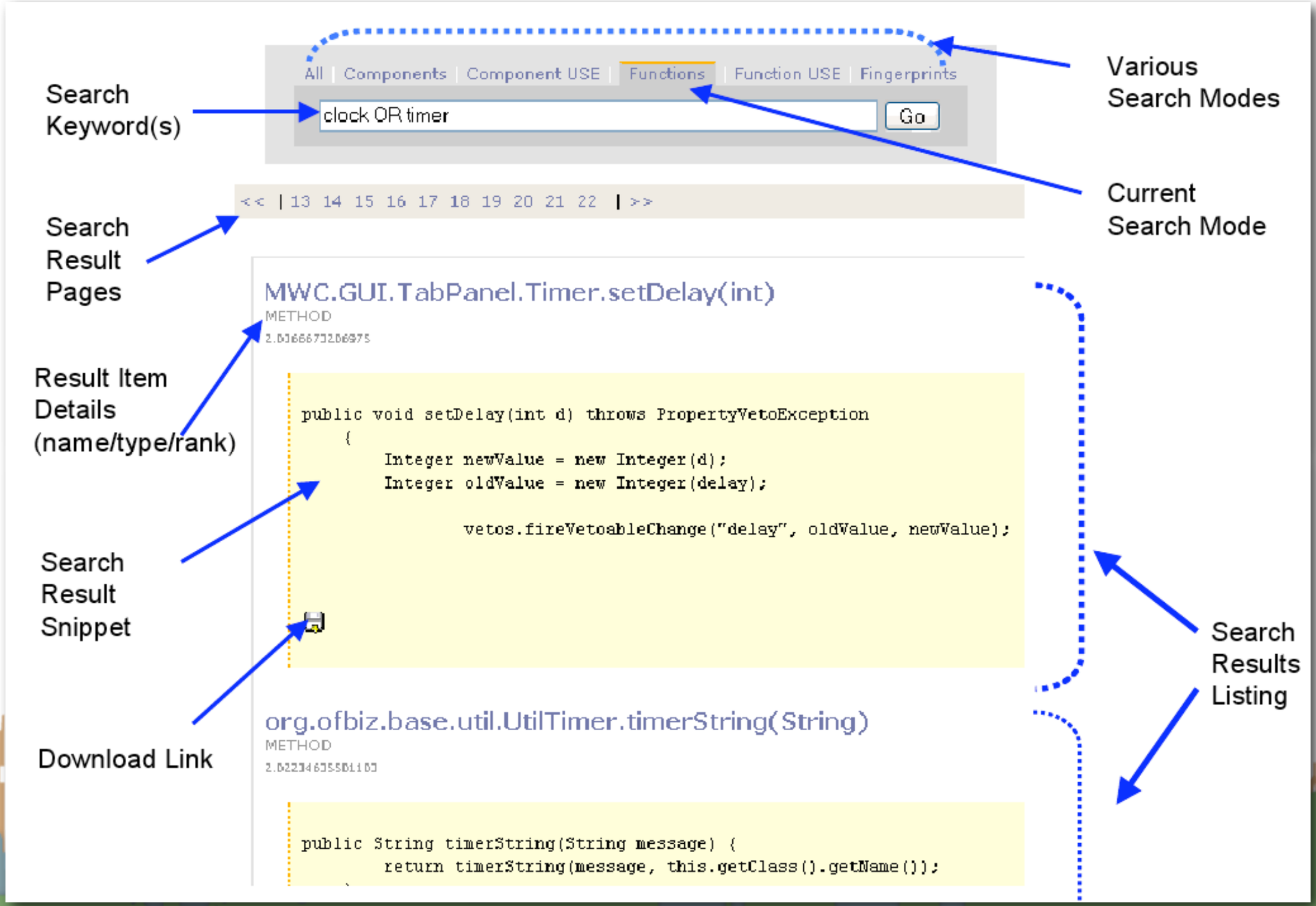
Sourcerer: mining and searching internet-scale software repositories

- Sourcerer architecture



Sourcerer: mining and searching internet-scale software repositories

- Sourcerer search interface



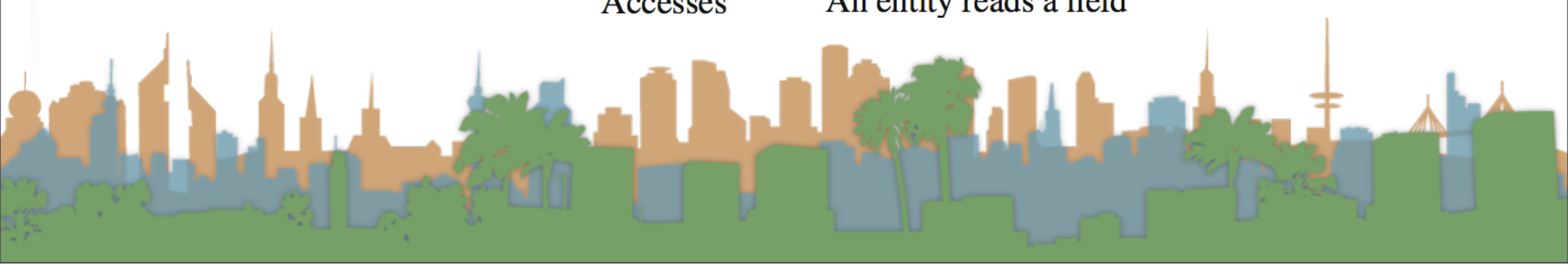
Sourcerer: mining and searching internet-scale software repositories

- 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



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

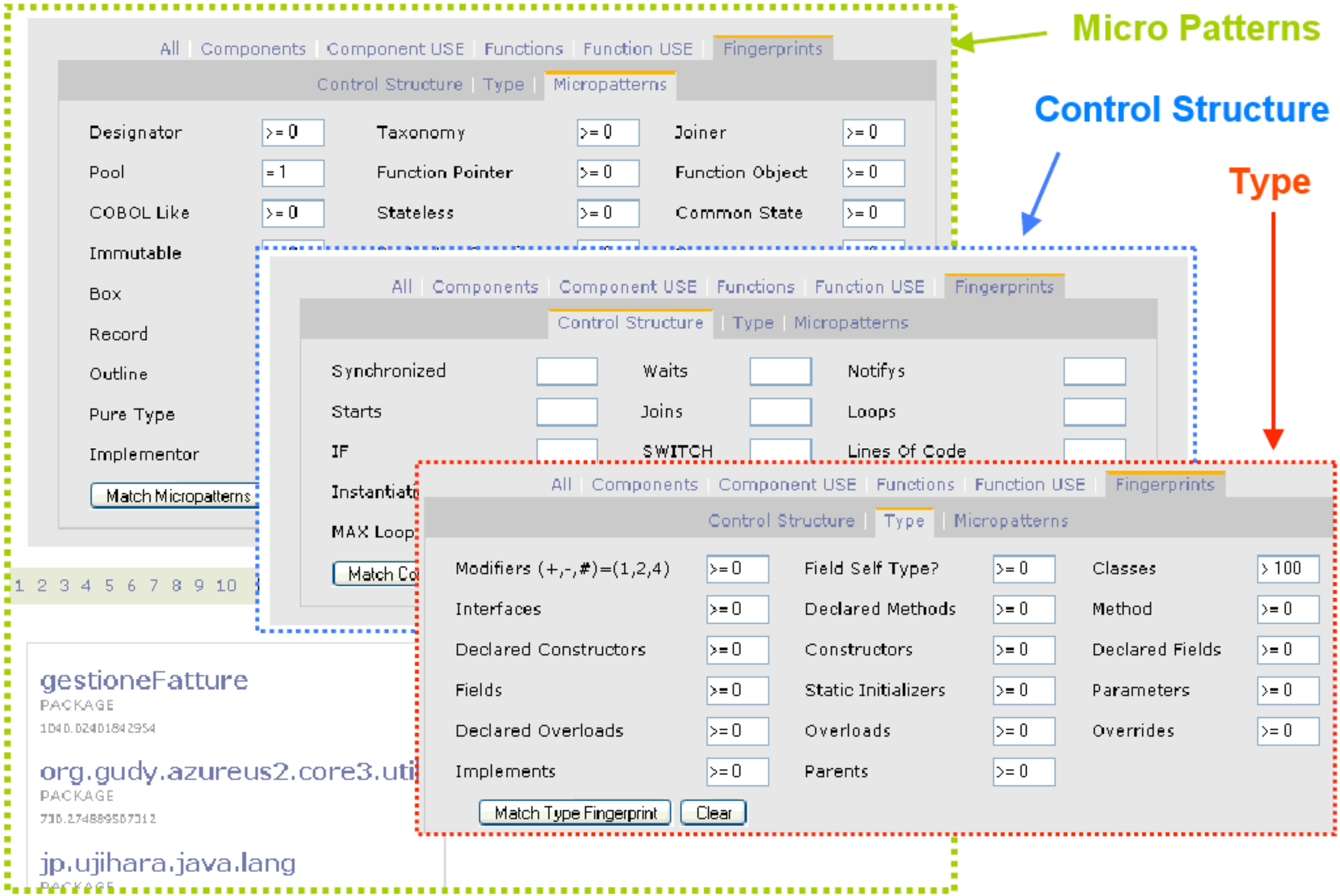
- Fingerprinting types
  - Control Structure Prints
    - Provides information about concurrency, iteration, and conditional constructs
    - Useful for identifying benchmark datasets
  - Java Type Prints
    - Captures information about object-oriented constructs (classes, methods, fields, constructors,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



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

Sourcerer: mining and searching internet-scale software repositories

- Current Sourcerer Statistics
  - Repository
    - Total number of projects (with source): 1555
    - Total source files: 185,439
    - Total lines of code: 18,804,522
    - Size of repository: 1.17 GB

Sourcerer: mining and searching internet-scale software repositories

- Current Sourcerer Statistics
  - Database
    - Total number of java packages: 47,898
    - Total number of java classes: 254,049
    - Total number of methods: 1,516,212
    - Total number of fields: 732,764
    - Total number of relations: 11,345,077

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		

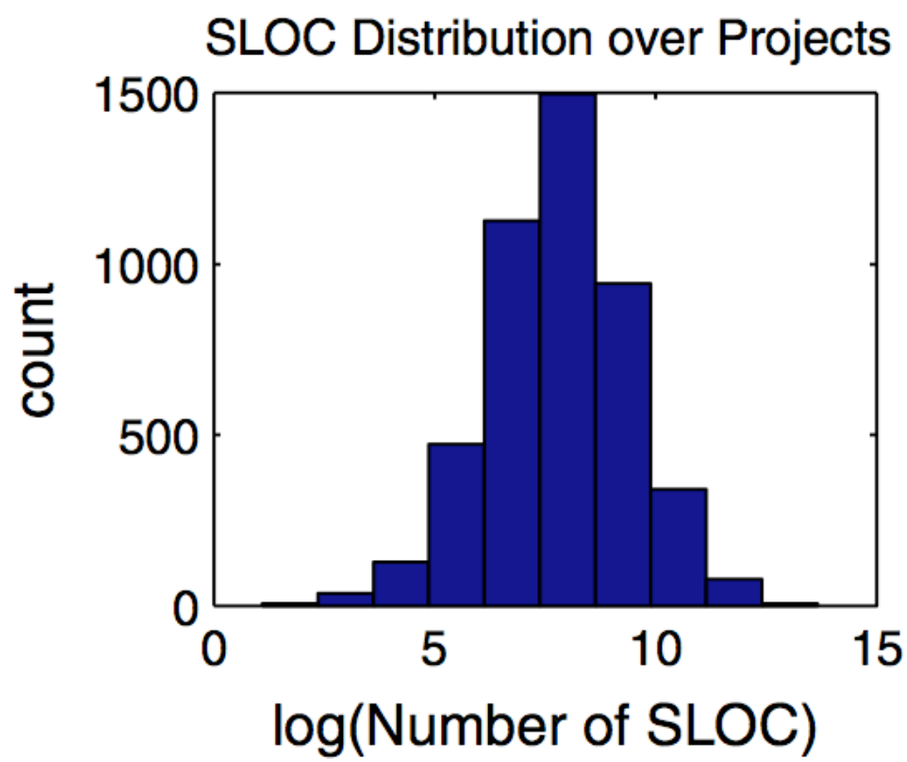
Sourcerer: mining and searching internet-scale software repositories

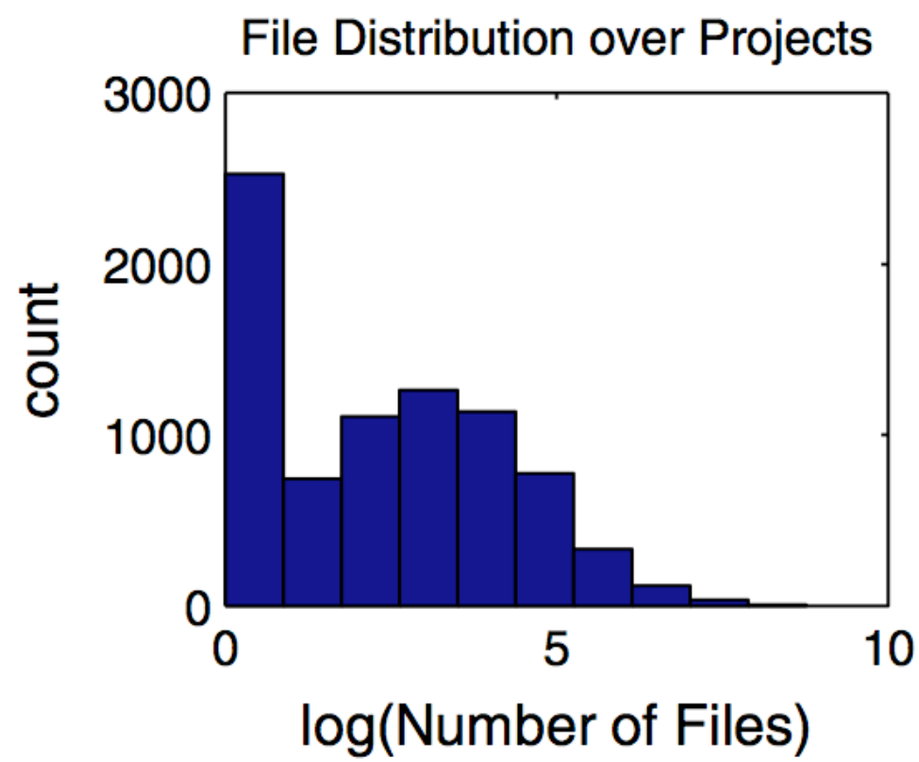
- Keyword frequency (%)

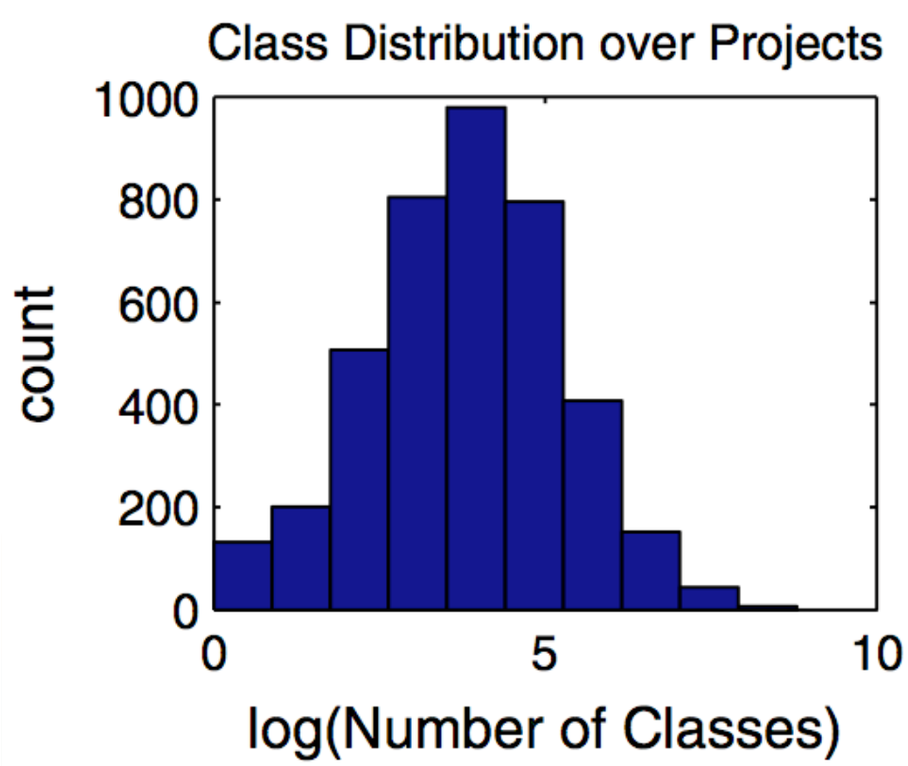
Keyword	Percentage	Keyword	Percentage
Public	12.53	This	0.89
If	8.44	Break	0.85
New	8.39	While	0.63
Return	7.69	Super	0.57
Import	6.89	Instanceof	0.56
Int	6.54	Double	0.55
Null	5.52	Long	0.54
Void	4.94	Implements	0.43
Private	3.66	Char	0.30
Static	3.16	Float	0.28
Final	3.01	Abstract	0.25
Else	2.33	Synchronized	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		

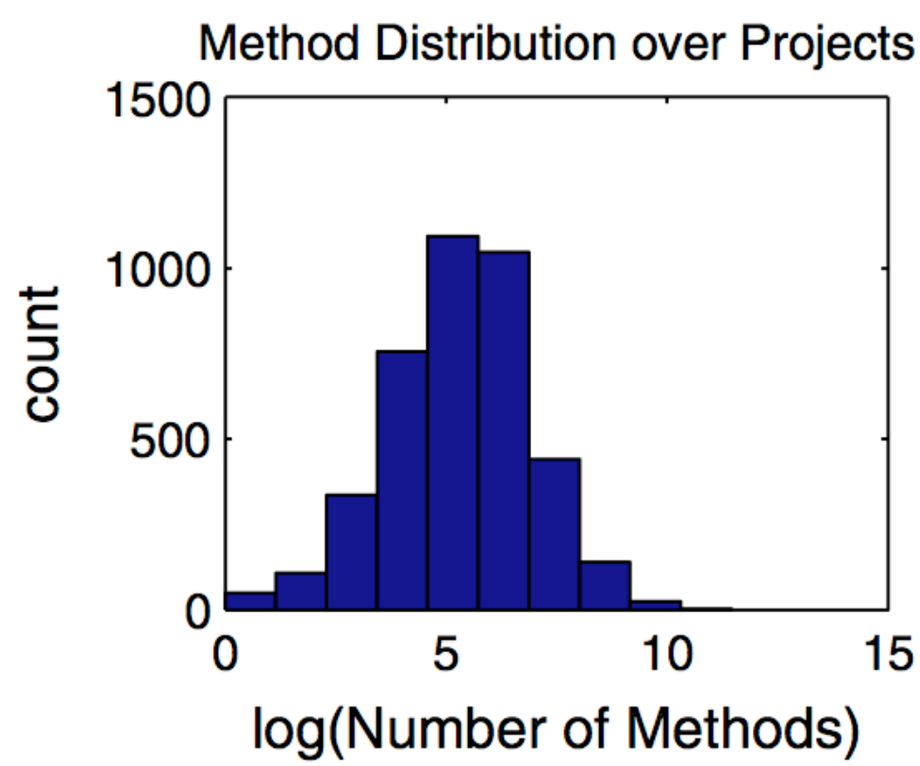
Sourcerer: mining and searching internet-scale software repositories

- Code characteristics









Sourcerer: 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