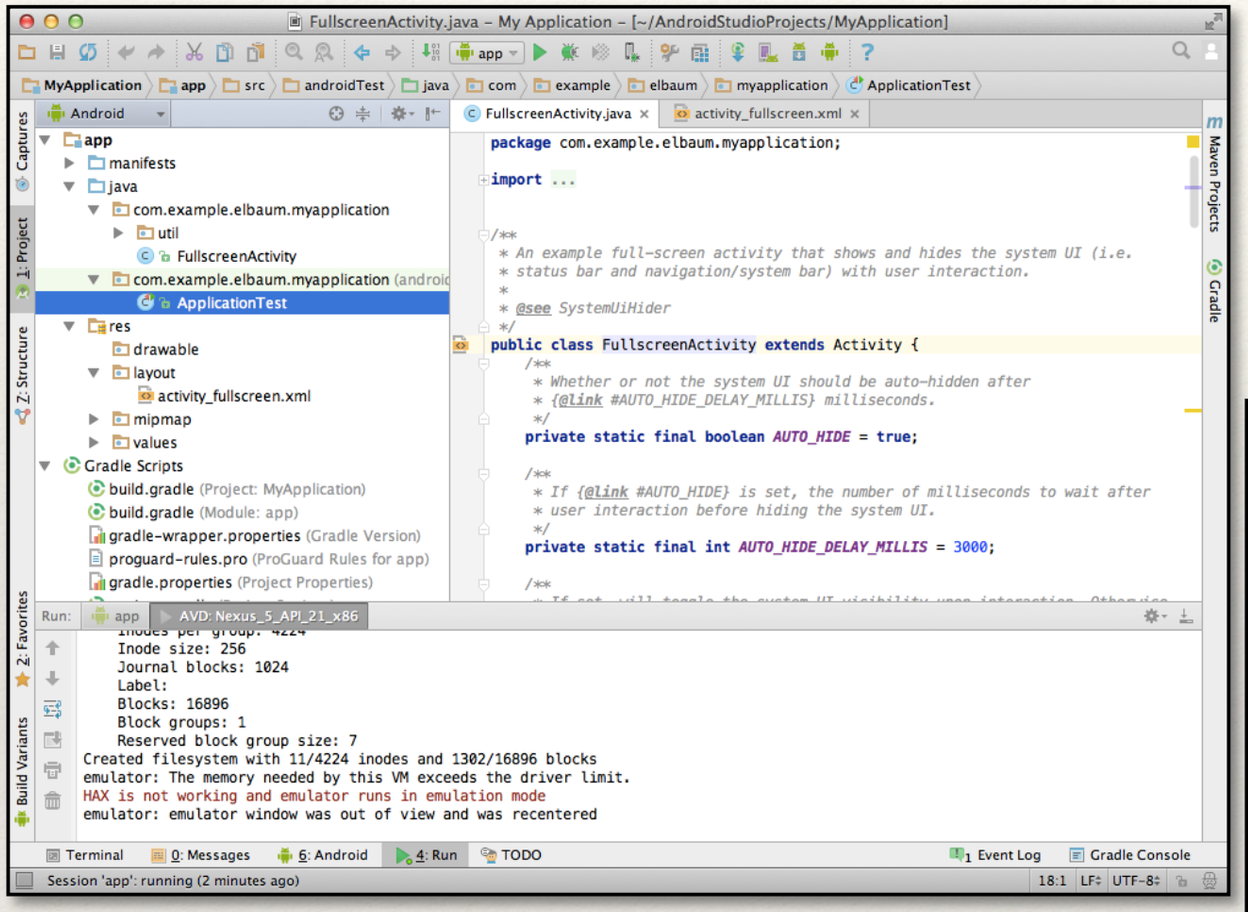
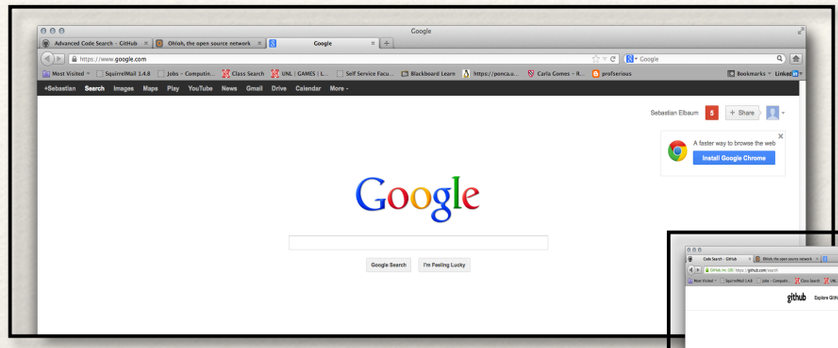
Making software development easier, one search at a time

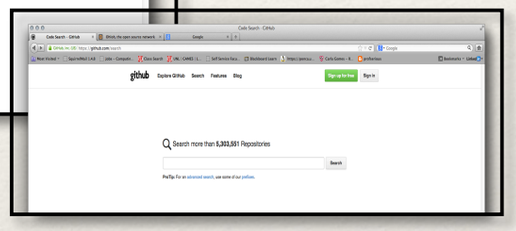
Dr. Kathryn Stolee
Assistant Professor
North Carolina State University

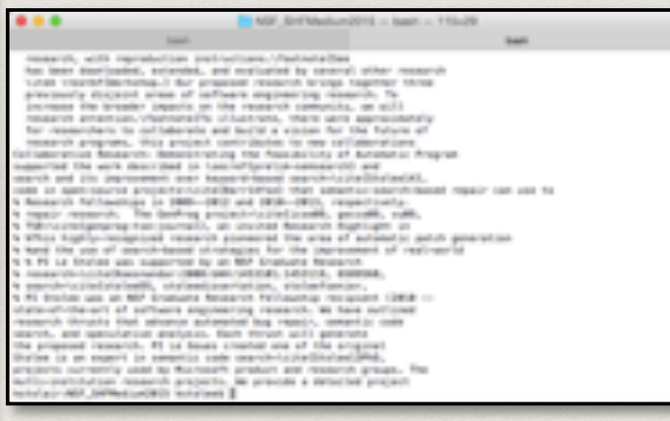
May 4, 2018

Motivation: Development and Search









What makes search for code different than search for information?

Evaluating How Developers Use General-Purpose Web-Search for Code Retrieval

Md Masudur Rahman, Jed Barson, Sydney Paul , Joshua Kayani , Federico Andrés Lois , Sebastián Fernandez Quezada , Chris Parnin, Kathryn Stolee, Baishakhi Ray

MSR 2018





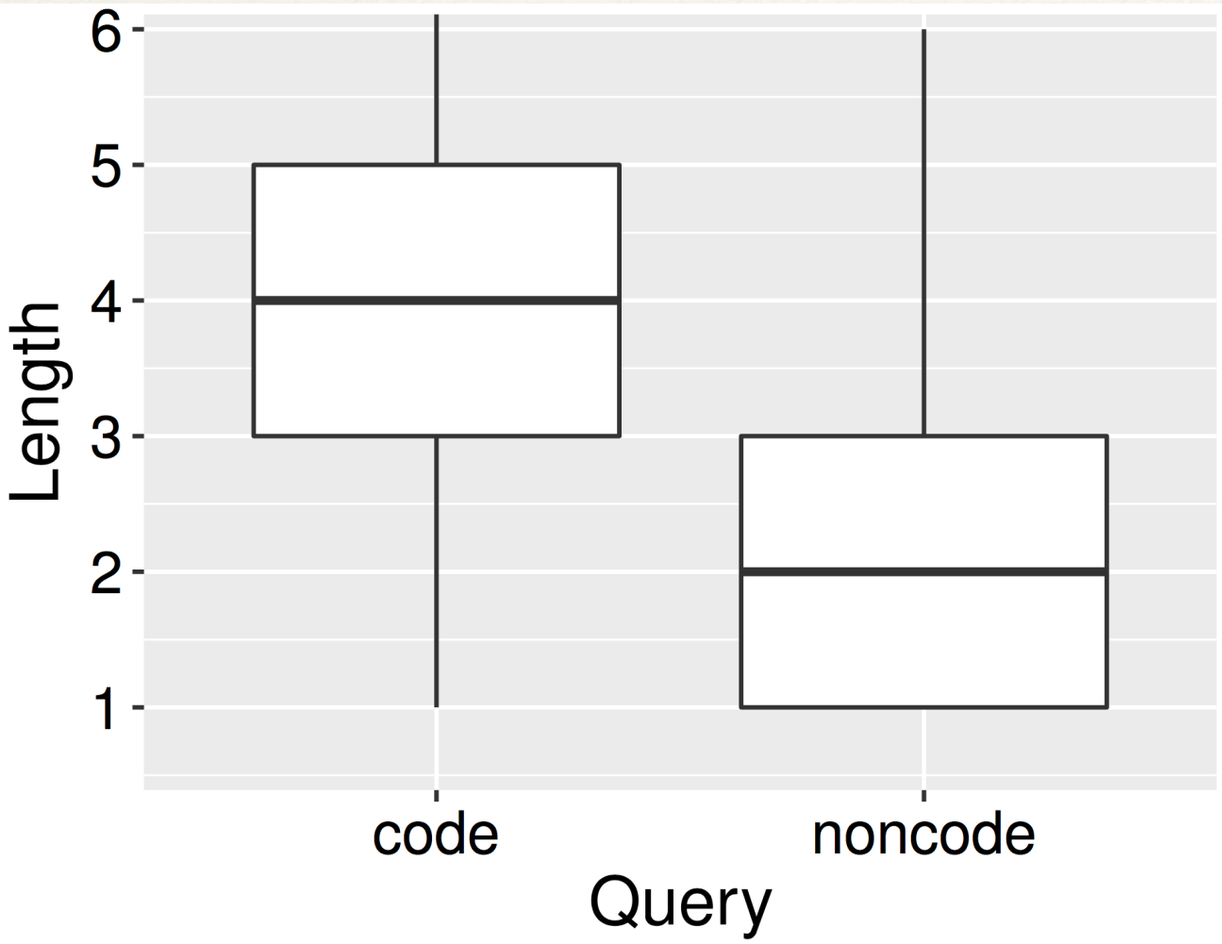
We got some data

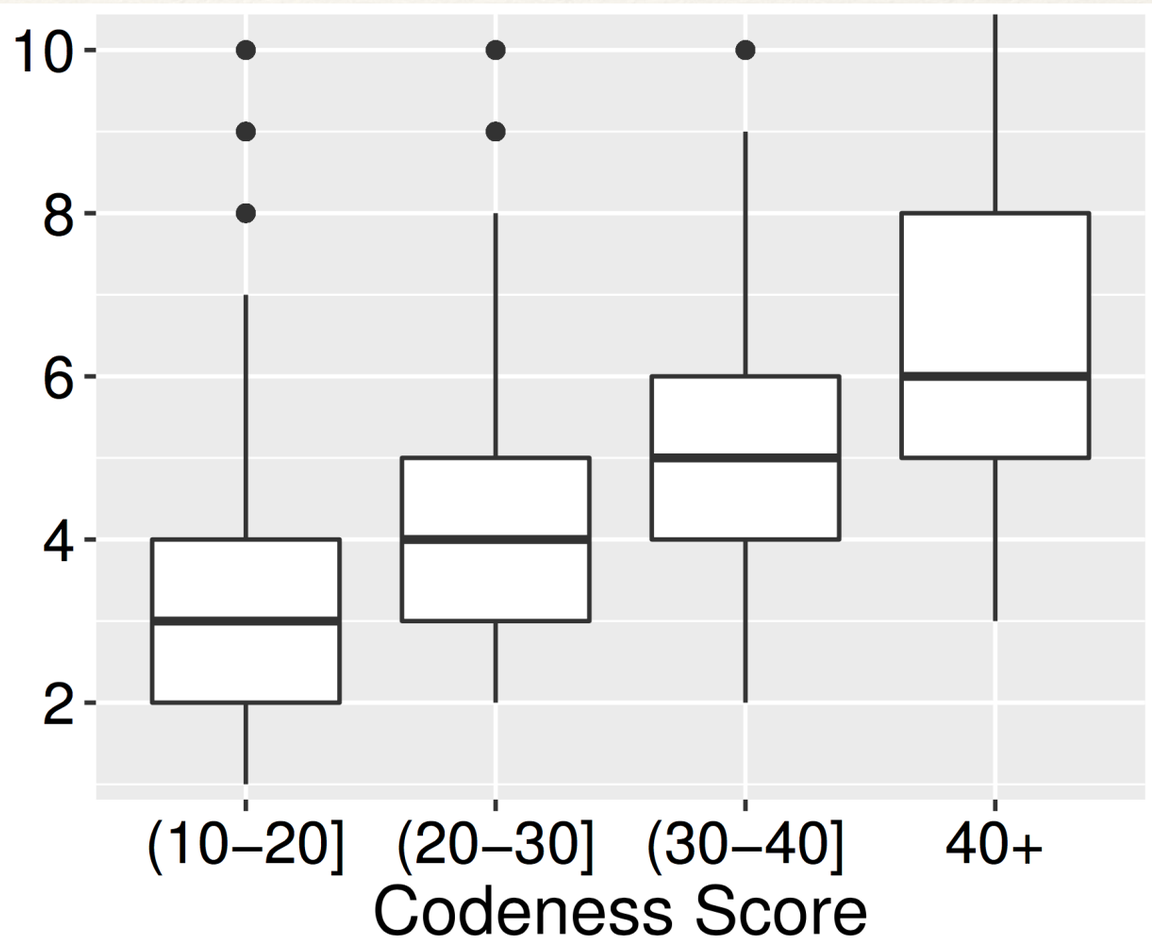
- 14-months of search history from 310 industry programmers / technical leads
- Collected *only* while the IDE was being used (assume: this reflects activity during software development)
- 149,610 search queries

We classified the data

	Query	Code score	Type	
1	Javascript mp3 play time	40.71	Code	88,577
2	How to perform xml serialization for parameterless constructor in c#	67.33	Code	
3	Javascript get track length from metadata	48.57	Code	
4	May the 4th be With You*	7.07	Noncode	61,033
5	Messi curly goal	2.58	Noncode	

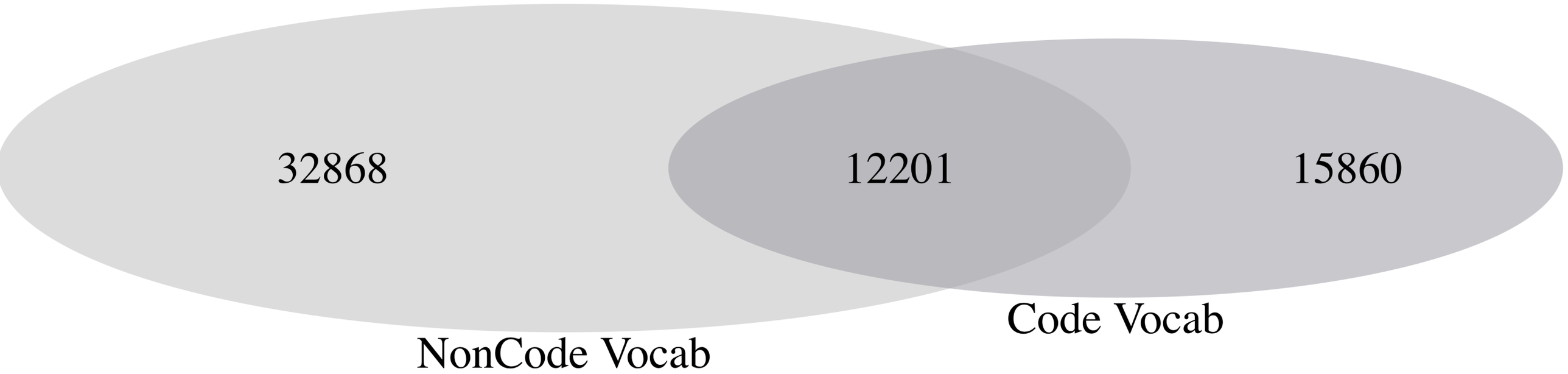
What we found





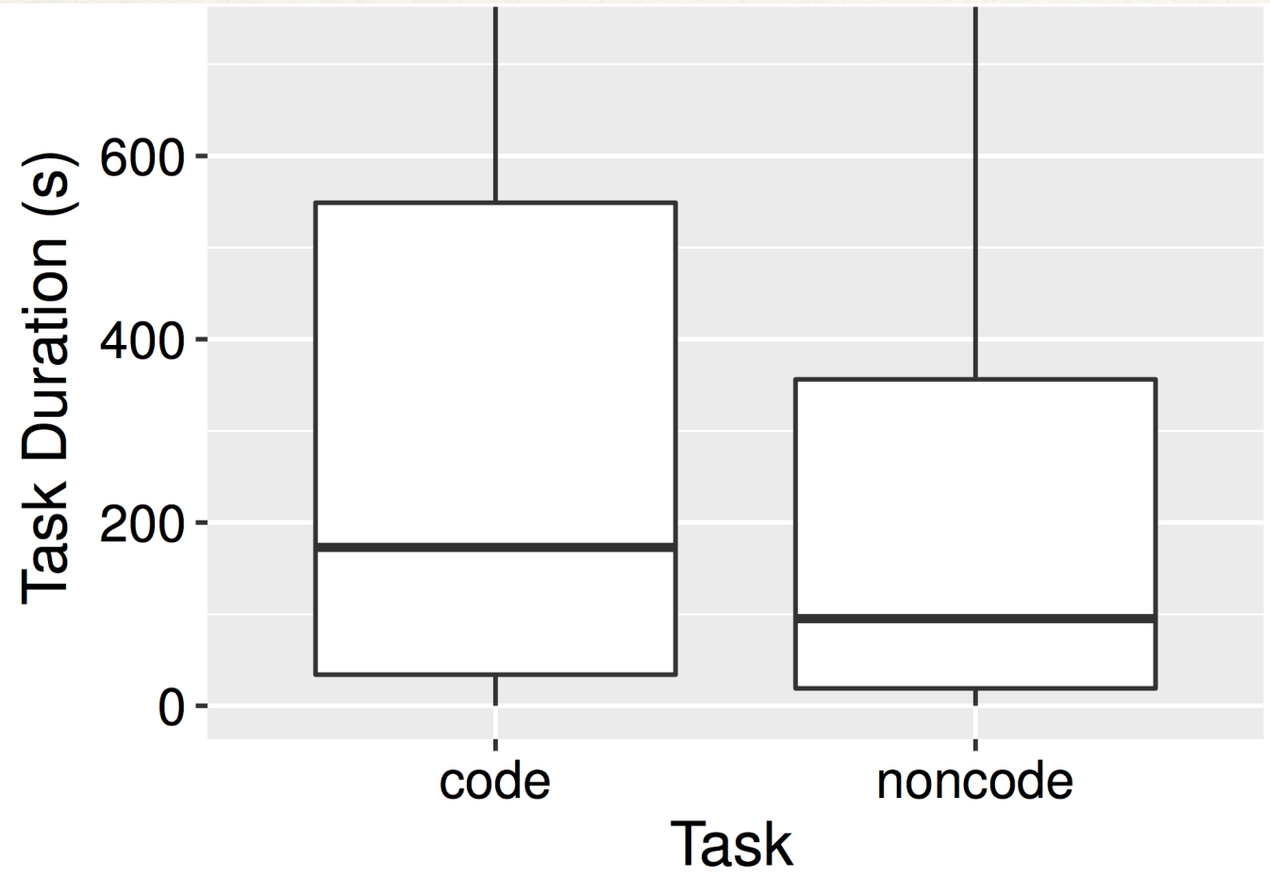
The more code-related, the longer the query.

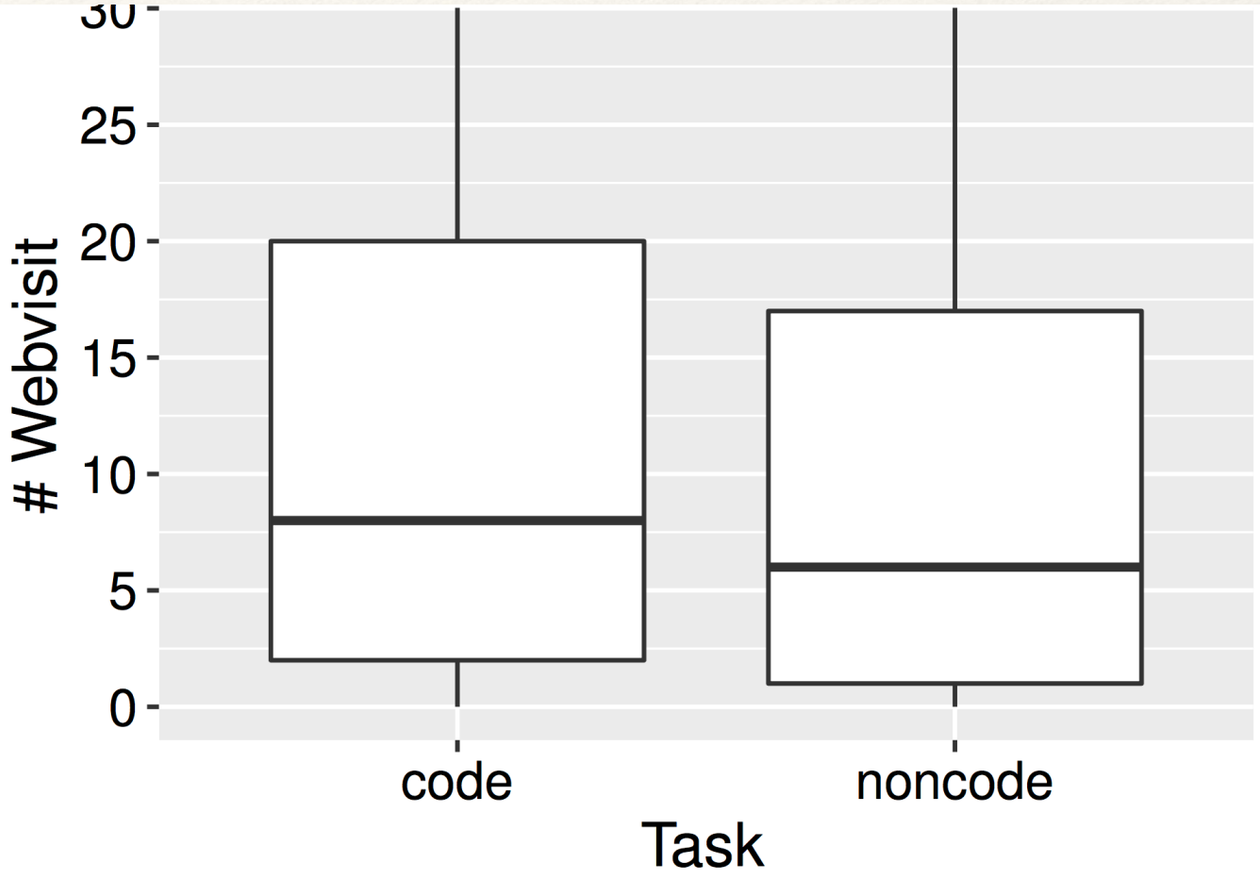
What we found



The code-query vocabulary is smaller!

What we found





Code-related tasks take longer and require more website visits.

What it means

- Code search queries are *linguistically* different
- Code search queries *take more effort* for task completion

What can we do?

- Better *previews* in the search results to answer questions more quickly and require fewer clicks
- Linguistic model that *tags parts of code-speech* for query expansion and results ranking

In what contexts are developers searching for code?

How Developers Search for Code: A Case Study (at Google)

Caitlin Sadowski, Kathryn T. Stolee, and Sebastian Elbaum

FSE 2015

Code Search at Google

chromium

An open-source browser to help move the web forward.

Search projects

Project Home | Downloads | Wiki | Issues | **Code Search**

Search code

regular expressions

Search Code

Search Options

Language: Any langu

File Path:

Class:

Function:

Symbol:

Case Sensitive: No

Exact: No

Results **1** - **10** of **596** (**0.597** seconds) View style

[chromium] **src/v8/src/ast.h** Show 10 matches

```
v8::internal
  949: class ExpressionStatement final : public Statement {
  950:  public:
  951:   DECLARE_NODE_TYPE(ExpressionStatement)

v8::internal::ExpressionStatement
  957: protected:
  958:  ExpressionStatement(Zone* zone, Expression* expression, int pos)
  959:      : Statement(zone, pos), expression_(expression) { }

v8::internal
 2844: // ----------------------------------------------------------------------
 2845: // Regular expressions
```

[chromium] **src/v8/src/ast.cc** Show 6 matches

```
v8::internal
  794: void AstVisitor::VisitExpressions(ZoneList<Expression*>* expressions) {
  795:   for (int i = 0; i < expressions->length(); i++) {
  796:     // The variable statement visiting code may pass NULL expressions
  797:     // to this code. Maybe this should be handled by introducing an

  806: // ----------------------------------------------------------------------
```

Study Design

Logging + Survey

rawTime	DayHourMin	isFileRe	isSugge	isSearcl	isQuery	isDirect	isAnnot	isImplic	session_id	click_type	hashed_quer	hashed_query	hashedUserName
1.38635E+15	340:09:49:05	1	0	0	0	0	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:05	1	0	0	0	0	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:06	0	0	0	0	1	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:06	0	0	0	0	1	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:06	0	0	0	0	0	0	0	1.38635E+12	browser_sta	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:06	0	0	0	0	0	0	0	1.38635E+12	browser_sta	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:06	0	0	0	0	0	0	0	1.38635E+12	source_visibl	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:06	0	0	0	0	0	0	0	1.38635E+12	source_visibl	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:06	0	0	0	0	0	0	0	1.38635E+12	file_requeste	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:06	0	0	0	0	0	0	0	1.38635E+12	file_requeste	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:52	1	0	0	0	0	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:52	1	0	0	0	0	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:53	0	0	0	0	1	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:53	0	0	0	0	1	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:53	0	0	0	0	0	0	0	1.38635E+12	browser_sta	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:53	0	0	0	0	0	0	0	1.38635E+12	browser_sta	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:53	0	0	0	0	0	0	0	1.38635E+12	source_visibl	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:53	0	0	0	0	0	0	0	1.38635E+12	source_visibl	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:53	0	0	0	0	0	0	0	1.38635E+12	file_requeste	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:53	0	0	0	0	0	0	0	1.38635E+12	file_requeste	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:58	1	0	0	0	0	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:58	1	0	0	0	0	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:58	0	0	0	0	0	0	0	1.38635E+12	source_visibl	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:58	0	0	0	0	0	0	0	1.38635E+12	source_visibl	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:58	0	0	0	0	0	0	0	1.38635E+12	file_requeste	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:09:49:58	0	0	0	0	0	0	0	1.38635E+12	file_requeste	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:02:27	0	0	0	0	0	0	0	1.38635E+12	browser_sta	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:02:27	0	0	1	0	0	0	0	1.38635E+12	NULL	65803f90b53	65803f90b5371dc3b88b60a06713354b	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:02:27	0	0	0	0	0	0	0	1.38635E+12	browser_sta	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:02:27	0	0	1	0	0	0	0	1.38635E+12	NULL	65803f90b53	65803f90b5371dc3b88b60a06713354b	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:02:30	0	0	0	0	0	0	0	1.38635E+12	search_done	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:02:30	0	0	0	0	0	0	0	1.38635E+12	search_done	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:02:38	0	0	1	0	0	0	0	1.38635E+12	NULL	65803f90b53	41c70400c32f9fa8840747ae9dc4f2ea	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:02:38	0	0	1	0	0	0	0	1.38635E+12	NULL	65803f90b53	41c70400c32f9fa8840747ae9dc4f2ea	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:02:44	0	0	0	0	0	0	0	1.38635E+12	search_done	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:02:44	0	0	0	0	0	0	0	1.38635E+12	search_done	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:00	1	0	0	0	0	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:00	1	0	0	0	0	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:01	0	0	0	0	0	0	0	1.38635E+12	rc	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:01	0	0	0	0	0	0	0	1.38635E+12	rc	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:01	0	0	0	0	1	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:01	0	0	0	0	1	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:03	0	0	0	0	0	1	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:03	0	0	0	0	0	1	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:03	0	0	0	0	0	1	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:03	0	0	0	0	0	1	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:03	0	0	0	0	0	1	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:03	0	0	0	0	0	1	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:03	0	0	0	0	0	1	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:03	0	0	0	0	0	1	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:04	0	0	0	0	1	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:04	0	0	0	0	1	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:04	0	0	0	0	0	0	0	1.38635E+12	source_visibl	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:04	0	0	0	0	0	0	0	1.38635E+12	source_visibl	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:04	0	0	0	0	0	0	0	1.38635E+12	file_requeste	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:03:04	0	0	0	0	0	0	0	1.38635E+12	file_requeste	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:08:25	0	0	1	0	0	0	0	1.38635E+12	NULL	file:3911d45	a5a6680dc83ef6ce89db78d36b83a901	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:08:25	0	0	1	0	0	0	0	1.38635E+12	NULL	file:3911d45	a5a6680dc83ef6ce89db78d36b83a901	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:08:25	0	0	0	0	0	0	0	1.38635E+12	browser_sta	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:08:25	0	0	0	0	0	0	0	1.38635E+12	browser_sta	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:08:26	0	0	0	0	0	0	0	1.38635E+12	search_done	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:08:26	0	0	0	0	0	0	0	1.38635E+12	search_done	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:08:30	0	0	1	0	0	0	0	1.38635E+12	NULL	file:3911d45	96fea05cc3c9aebf6868950b7f2817e7	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:08:30	0	0	1	0	0	0	0	1.38635E+12	NULL	file:3911d45	96fea05cc3c9aebf6868950b7f2817e7	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:08:31	0	0	0	0	0	0	0	1.38635E+12	search_done	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:08:31	0	0	0	0	0	0	0	1.38635E+12	search_done	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:08:33	1	0	0	0	0	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471
1.38635E+15	340:10:08:33	1	0	0	0	0	0	0	1.38635E+12	NULL	NULL	6c3e226b4d4795d518ab341b0824ec29	71eb53619b468957238af3f6cef5f471

What are you doing?

- ☐ Exploring
- ☐ Working on a CL
- ☐ Triaging a problem
- ☐ Reviewing a CL
- ☐ Designing a new feature
- ☐ Other: ______

How familiar are you with the code you are searching for?

- ☐ Very familiar (e.g., I wrote it)
- ☐ Somewhat familiar (e.g., I've seen it before)
- ☐ Not familiar
- ☐ Not sure

What are you trying to learn?

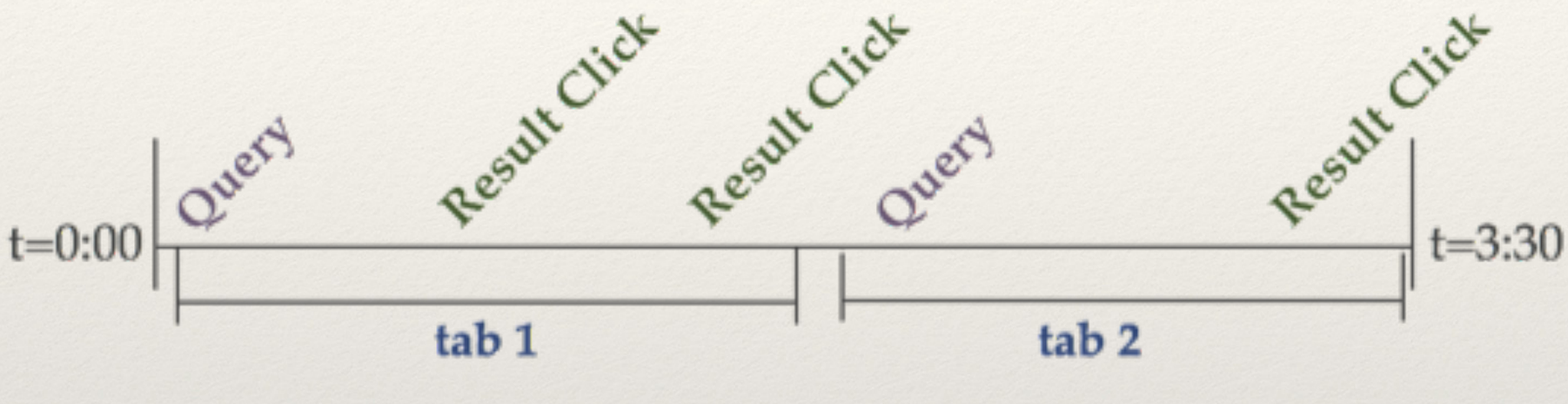
- ☐ Code attributes (e.g., location, owner, size)
- ☐ How code changed
- ☐ How code works
- ☐ How to use code
- ☐ Other: ______

What questions are you trying to answer?

Study Context

- 27 developers
  - 5 female, 22 male
  - Avg. 9.75 years development (3.4 at Google)
- 2 weeks of data collection

RQ4: What does a typical search session entail?



~12 sessions / day = 42 minutes per day searching!

Why do programmers search?

Analyzed the free-text survey responses

Category - "example question"	Percent
Example Code Needed - *"How to write a hash function?"*	33.5%
Exploring or Reading Code - *"What does this script do?"*	26%
Code Localization - *"Where a class is instantiated?"*	16%
Determine Impact - *"Why is this failing?"*	16%
Metadata - *"Who last touched some code?"*	8.5%

High-Level Results

- Code search tools are becoming entrenched
- Developers are searching more, reformulating often
- Search patterns vary greatly among contexts

Search Context	Search Pattern	
Discovering a library for a task	Search+	73%
What is the side effect of a proposed change?	Search+ Click+	75%
Very familiar with code I am searching for	Search Click	28%

Open Question => Next Steps

- How do we know a search is successful?
  => Query developers directly?
- How do these results generalize to other devs?
  => Replicate with another company, students, etc.

Are keywords expressive enough to describe code-related concepts?

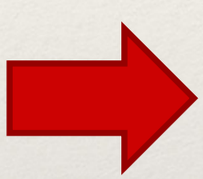
Solving the Search for Source Code

Kathryn T. Stolee, Sebastian Elbaum, and Daniel Dobos
TOSEM 2014

Search by Example

Search

Get email alias in Java

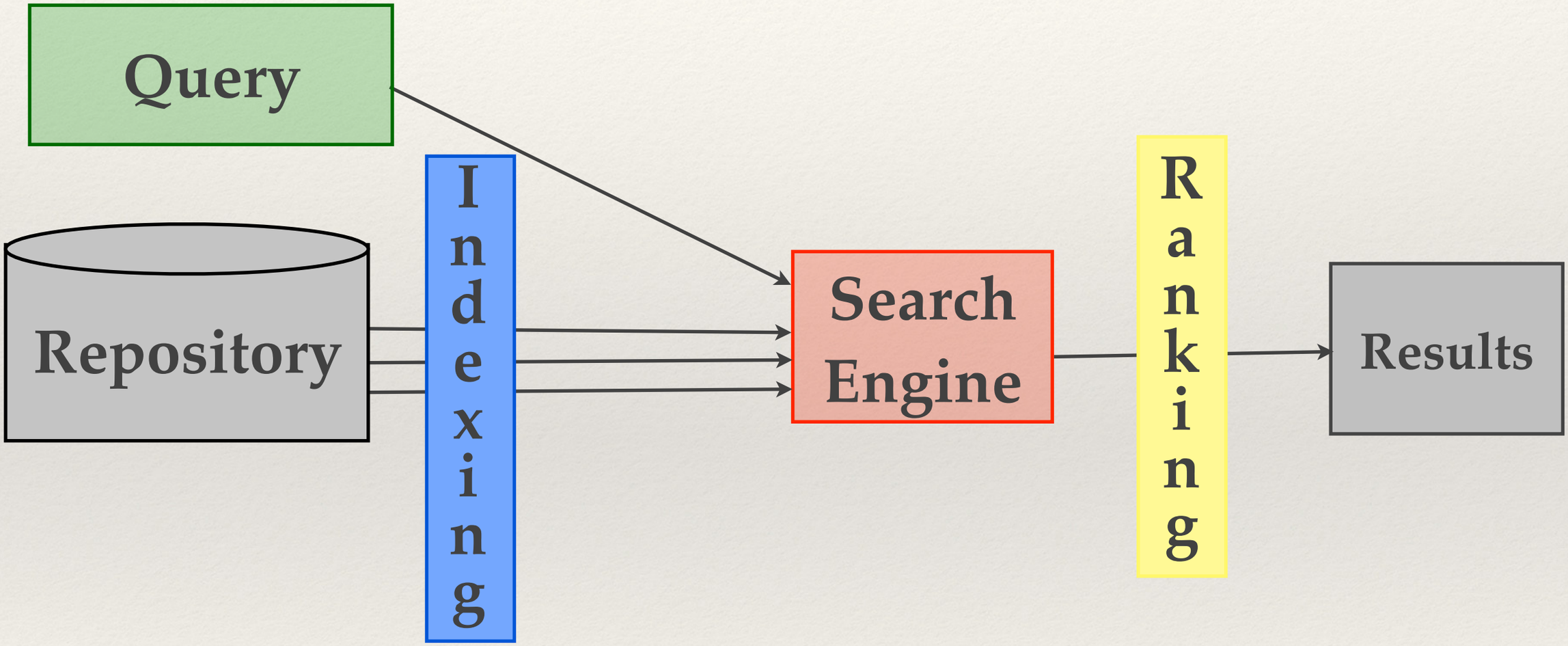


Satsy

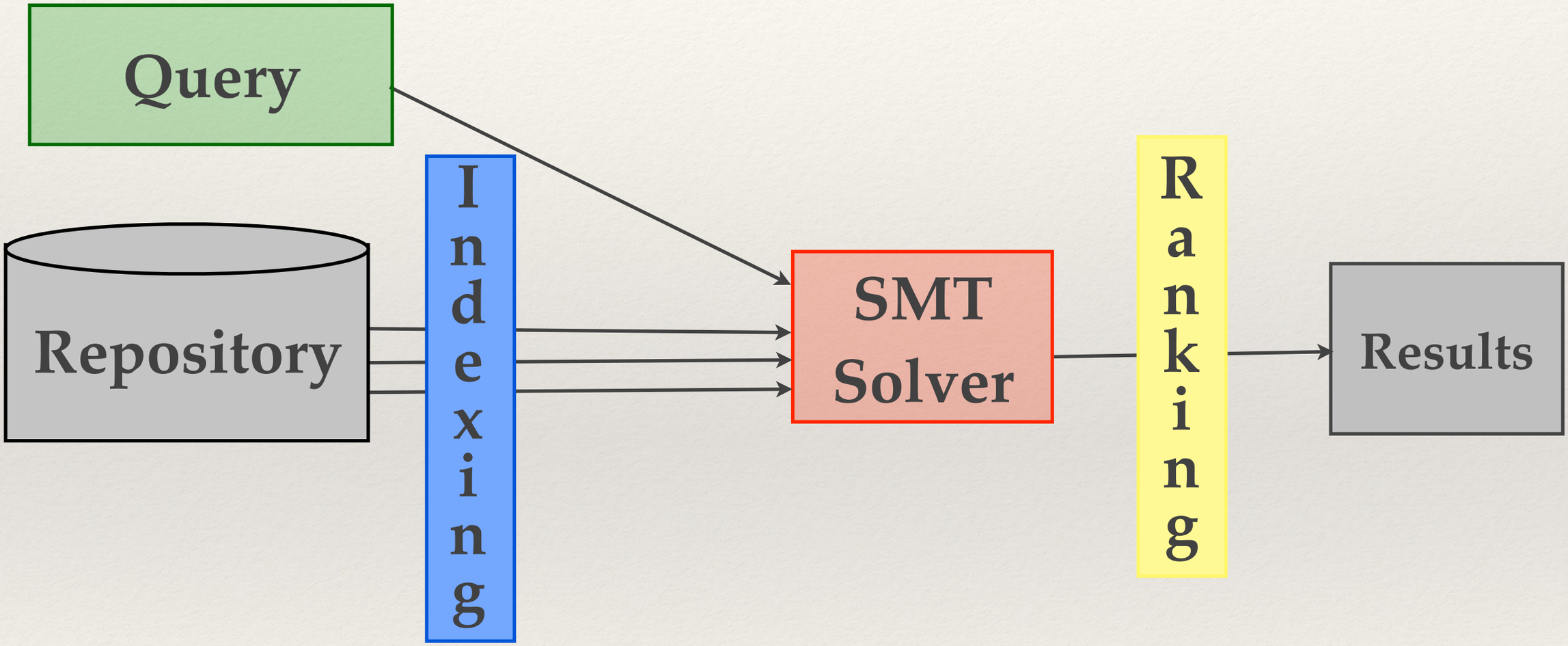
Input: elsa@mail.com

Output: elsa

How Does Search Work?



How Does Search Work?



SMT Solvers

Satisfiability **M**odulo **T**heory solvers determine if a logical formula is satisfiable

Facts	Assertions
`a >= 0`	`(assert (>= a 0))`
`b = 2`	`(assert (= b 2))`
`c = 2`	`(assert (= c 2))`
`c = a * b`	`(assert (= ( * a b) c))`

Result: **sat** $a \mapsto 1$

SMT Solvers

Satisfiability **M**odulo **T**heory solvers determine if a logical formula is satisfiable

Facts	Assertions
`a >= 0`	`(assert (>= a 0))`
`b = ?`	**`(assert (= b ?))`**
`c = 2`	`(assert (= c 2))`
`c = a * b`	`(assert (= ( * a b) c))`

Result: **sat** $a \mapsto \mathbb{2} \wedge b \mapsto \mathbb{2}$

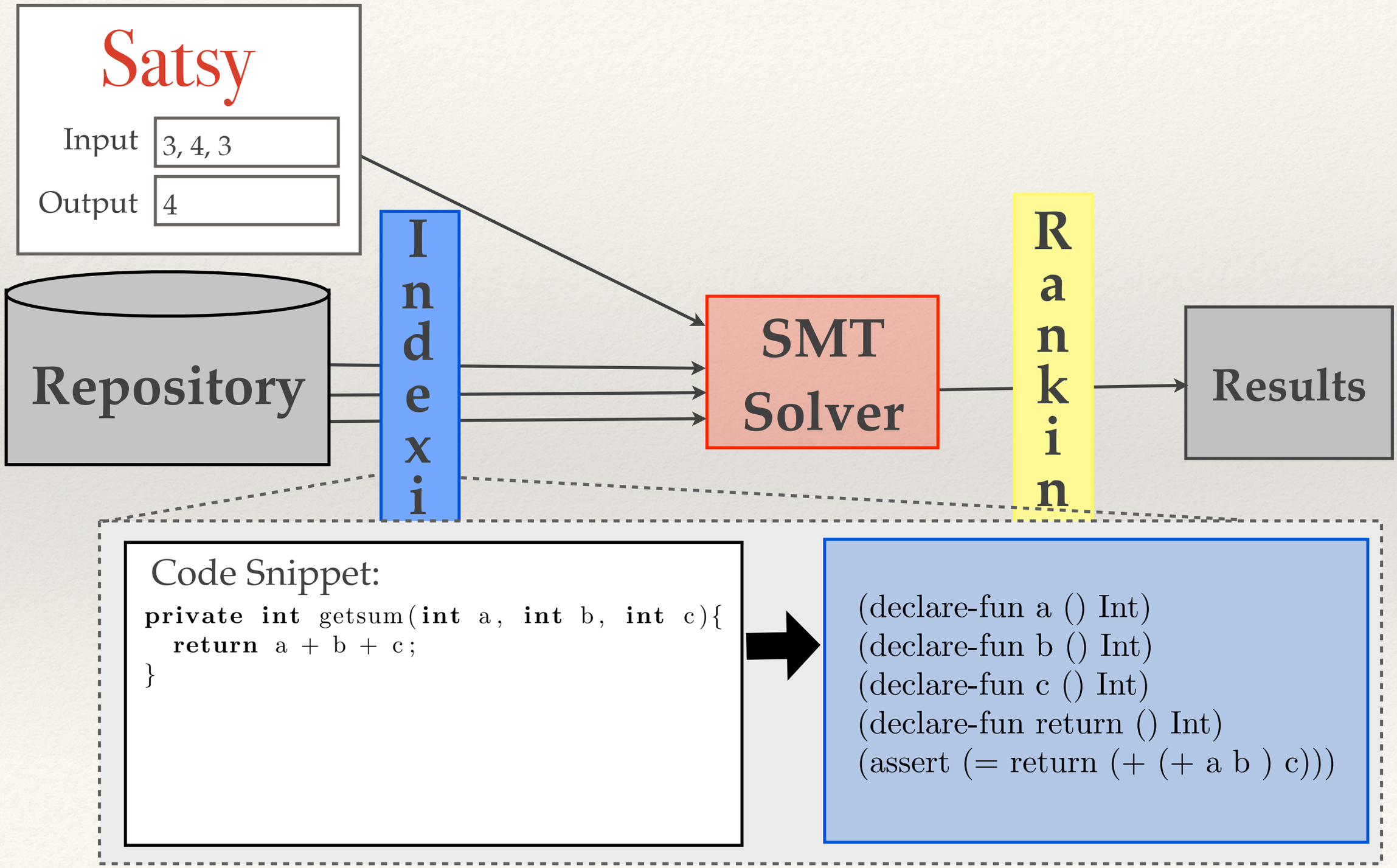
SMT Solvers

Satisfiability **M**odulo **T**heory solvers determine if a logical formula is satisfiable

Facts	Assertions
`a = 0`	`(assert (= a 0))`
`b = ?`	`(assert (= b ?))`
`c = 2`	`(assert (= c 2))`
`c = a * b`	`(assert (= ( * a b) c))`

Result: **unsat**

Search by Example



Language Support

⟨*statements*⟩ ::= (⟨*statement*⟩)*

⟨*statement*⟩ ::= ⟨*variableDeclaration*⟩ | ⟨*assignment*⟩ | ⟨*ifStatement*⟩ | ⟨*return*⟩

⟨*variableDeclaration*⟩ ::= ⟨*type*⟩ ⟨*ident*⟩ '`;`'

⟨*ifStatement*⟩ ::= '`if`' '`(`' ⟨*expr*⟩ '`)`' '`{`' ⟨*statements*⟩ '`}`'
('`else`' '`if`' '`(`' ⟨*expr*⟩ '`)`' '`{`' ⟨*statements*⟩ '`}`')*
('`else`' '`{`' ⟨*statements*⟩ '`}`')?

⟨*assignment*⟩ ::= ⟨*type*⟩? ⟨*ident*⟩ '`=`' ⟨*expr*⟩ '`;`'

⟨*return*⟩ ::= '`return`' ⟨*expr*⟩ '`;`'

⟨*expr*⟩ ::= ⟨*logicalOrExpr*⟩

⟨*logicalOrExpr*⟩ ::= ⟨*logicalAndExpr*⟩ ('`||`' ⟨*logicalAndExpr*⟩)*

⟨*logicalAndExpr*⟩ ::= ⟨*equalityExpr*⟩ ('`&&`' ⟨*equalityExpr*⟩)?

⟨*equalityExpr*⟩ ::= ⟨*relationalExpr*⟩ (('`!=`' | '`==`') ⟨*relationalExpr*⟩)?

⟨*relationalExpr*⟩ ::= ⟨*addExpr*⟩ (('`<`' | '`<=`' | '`>`' | '`>=`') ⟨*addExpr*⟩)?

⟨*addExpr*⟩ ::= ⟨*multExpr*⟩ (('`+`' | '`-`') ⟨*multExpr*⟩)?

⟨*multExpr*⟩ ::= ⟨*primaryExpr*⟩ (('`*`' | '`/`' | '`%`') ⟨*primaryExpr*⟩)?

⟨*primaryExpr*⟩ ::= ⟨*invokeExpr*⟩ | ⟨*stringLit*⟩ | ⟨*charLit*⟩ | ⟨*integer*⟩ | ⟨*booleanLit*⟩
| '`(`' ⟨*expr*⟩ '`)`' | ⟨*ident*⟩

⟨*invokeExpr*⟩ ::= (⟨*ident*⟩ | ⟨*stringLit*⟩) (('`.`' ⟨*noarg*⟩ '`(`' '`)`')
| (⟨*onearg*⟩ '`(`' ⟨*expr*⟩ '`)`')
| (⟨*twoarg*⟩ '`(`' ⟨*expr*⟩ ('`,`' ⟨*expr*⟩)? '`)`'))+

⟨*noarg*⟩ ::= '`length`' | '`toLowerCase`' | '`toUpperCase`' | '`trim`'

⟨*onearg*⟩ ::= '`charAt`' | '`concat`' | '`contains`' | '`endsWith`' | '`equals`' | '`equalsIgnoreCase`'
| '`indexOf`' | '`lastIndexOf`' | '`startsWith`' | '`substring`'

⟨*twoarg*⟩ ::= '`indexOf`' | '`lastIndexOf`' | '`replace`'

⟨*type*⟩ ::= '`char`' | '`String`' | '`int`' | '`boolean`'

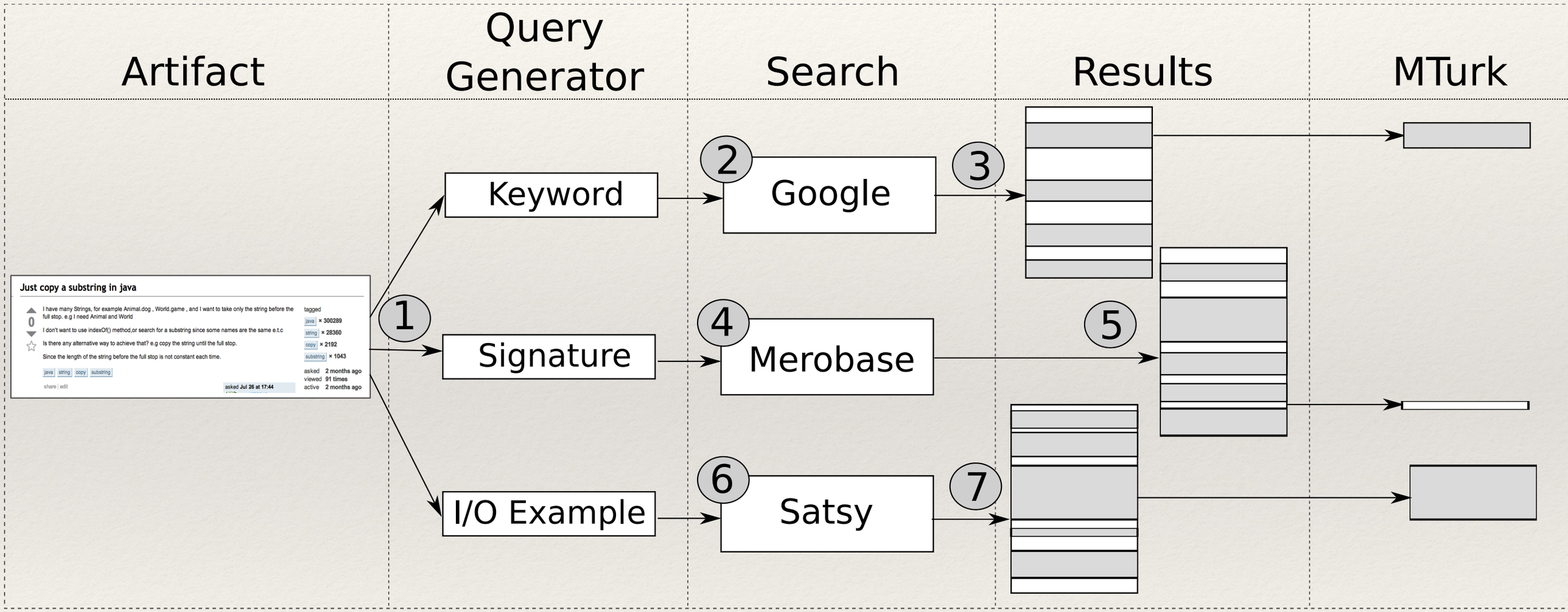
⟨*booleanLit*⟩ ::= '`true`' | '`false`'

⟨*integer*⟩ ::= ('`0`' | ('`1`'..'`9`') ('`0`'..'`9`')*);

Evaluation

How do **semantic** search results compare to those found using **Google** and those found with the code-specific search engine, **Merobase**, *from the perspective of a programmer*?

Setup


Artifact
Query Generator
Search
Results
MTurk
Just copy a substring in java
I have many Strings, for example Animal.dog , World.game , and I want to take only the string before the full stop. e.g I need Animal and World
I don't want to use indexOf() method,or search for a substring since some names are the same e.t.c
Is there any alternative way to achieve that? e.g copy the string until the full stop.
Since the length of the string before the full stop is not constant each time.
java string copy substring
share | edit
asked Jul 26 at 17:44
tagged
java × 300289
string × 28360
copy × 2192
substring × 1043
asked 2 months ago
viewed 91 times
active 2 months ago
1
Keyword
Signature
I/O Example
2
Google
3
4
Merobase
5
6
Satsy
7

Results

Semantic search returns more relevant results than Merobase and results at least as relevant as Google

Open Questions

- How can we extend the encoding? (e.g., C language, regular expressions, objects)
- How can we measure code similarity to enable efficient search?

How can semantic search be used?

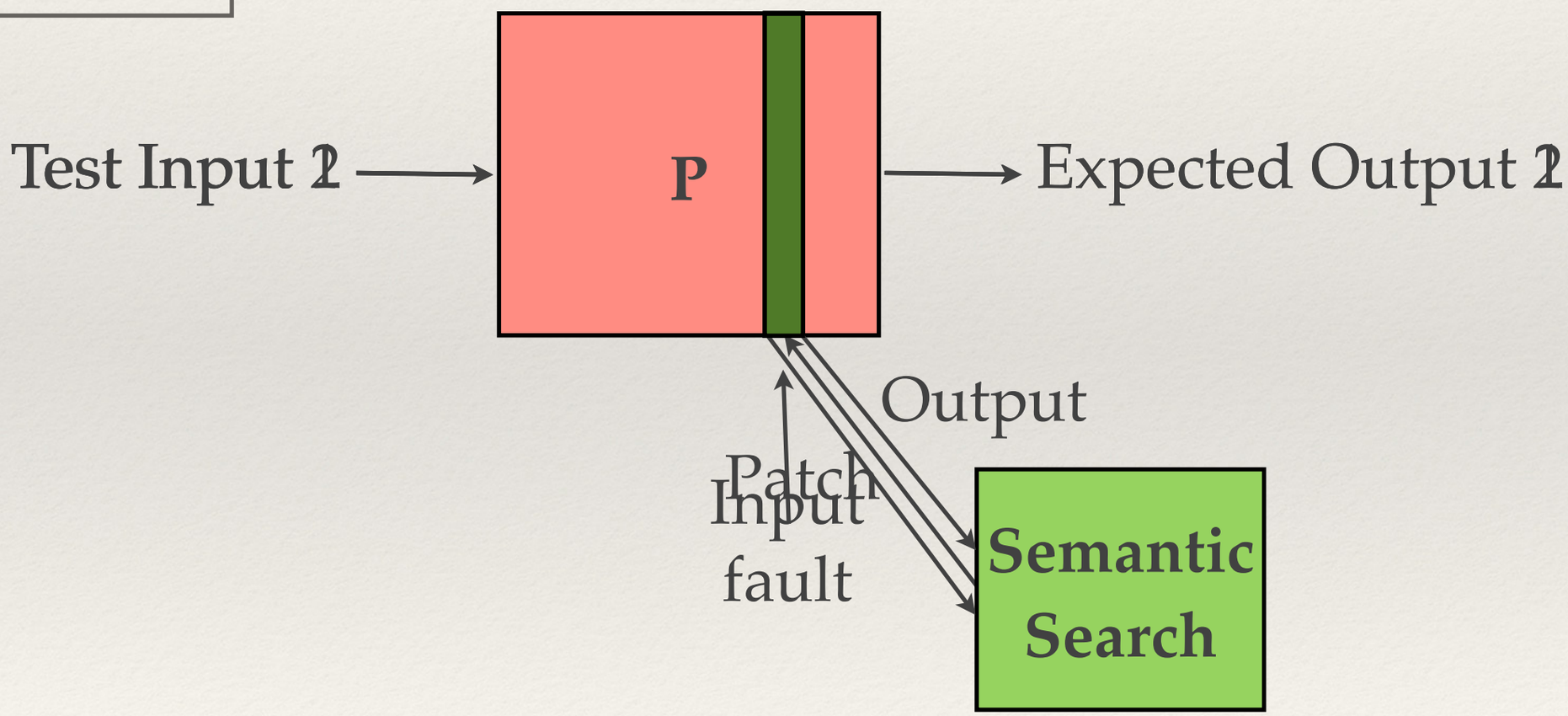
Semantic-Search Driven Automated Program Repair

Yalin Ke, Kathryn T. Stolee, Claire Le Goues and Yuriy Brun

ASE 2015

Semantic Code Search + Automated Program Repair

Test Suite
Test Case 1 ✓
Test Case 2 ✗



Example

```
1    int main() {
2         int a,b,c,median=0;
3         printf("Please enter 3 numbers separated by spaces >");
4         scanf("%d%d%d", &a, &b, &c);
5         if ((a<=b && a>=c) || (a>=b && a<=c))
6              median = a;
7         else if ((b<=a && b>=c) || (b>=a && b<=c))
8              median = b;
9         else if ((c<=b && a>=c) || (c>=b && a<=c))
10             median = c;
11        printf("%d is med\n", median);
12        return 0;
13   }
```

Fault!

Example

```
1    int main() {
2         int a,b,c,median=0;
3         printf("Please enter 3 numbers separated by spaces >");
4         scanf("%d%d%d", &a, &b, &c);
5         if ((a<=b && a>=c) || (a>=b && a<=c))
6              median = a;
7         else if ((b<=a && b>=c) || (b>=a && b<=c))
8              median = b;
9         else if ((c<=b && a>=c) || (c>=b && a<=c))
10             median = c;
11        printf("%d is med\n", median);
12        return 0;
13   }
```

Fault Region

Example

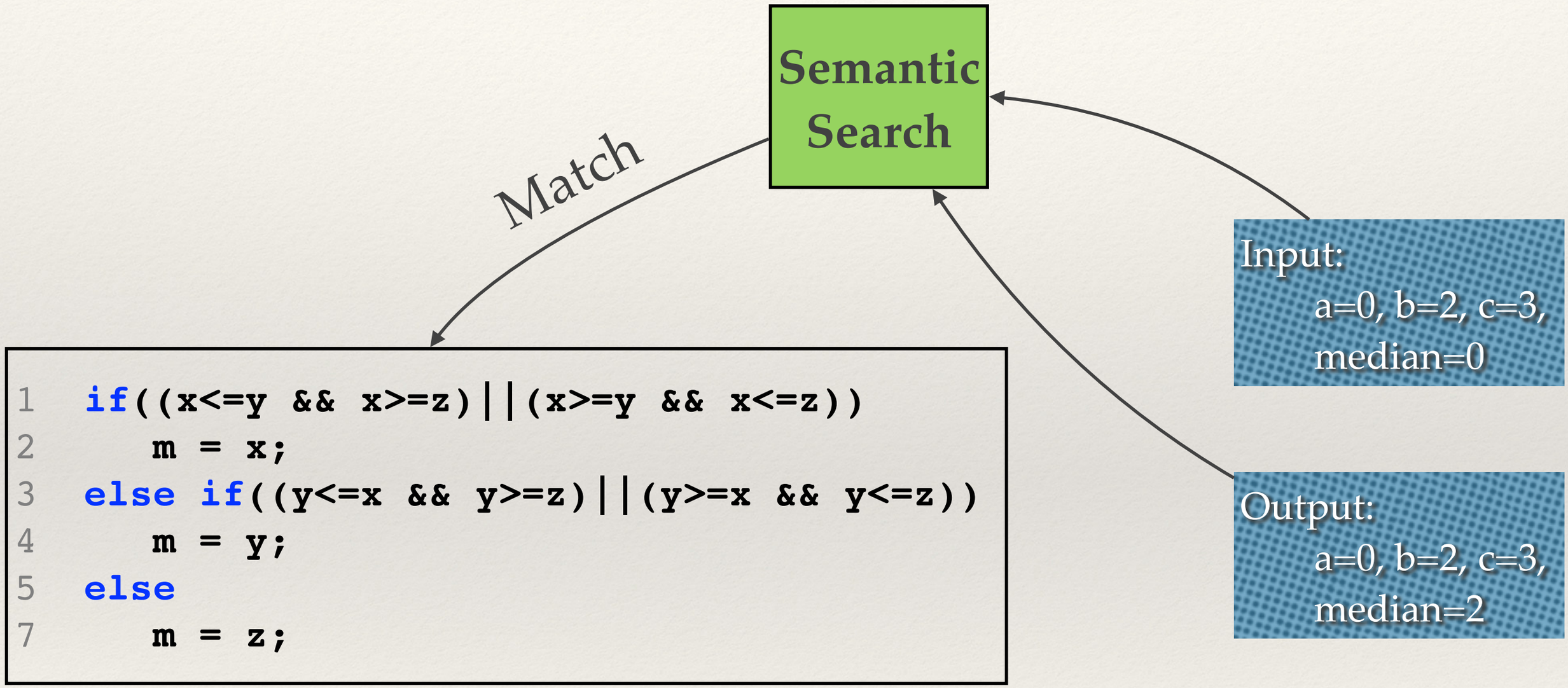
Input	Expected	Actual	Result
0 2 3	"2 is med"	"2 is med"	✓
0 2 1	"1 is med"	"0 is med"	X
8 2 6	"6 is med"	"0 is med"	X

```
1    int main() {
2         int a,b,c,median=0;
3         printf("Please enter 3 numbers separated by spaces >");
4         scanf("%d%d%d", &a, &b, &c);
5         if ((a<=b && a>=c) || (a>=b && a<=c))
6              median = a;
7         else if ((b<=a && b>=c) || (b>=a && b<=c))
8              median = b;
9         else if ((c<=b && a>=c) || (c>=b && a<=c))
10             median = c;
11        printf("%d is med\n", median);
12        return 0;
13   }
```

Input:
a=0, b=2, c=3,
median=0

Output:
a=0, b=2, c=3,
median=2

Example


Semantic Search
Match
Input:
a=0, b=2, c=3,
median=0
Output:
a=0, b=2, c=3,
median=2
1 if((x<=y && x>=z)||(x>=y && x<=z))
2 m = x;
3 else if((y<=x && y>=z)||(y>=x && y<=z))
4 m = y;
5 else
7 m = z;

Example

Input	Expected	Actual	Result
0 2 3	“2 is med”	“2 is med”	✓
0 2 1	“1 is med”	“0 is med”	X
8 2 6	“6 is med”	“0 is med”	X

```
1    int main() {
2        int a,b,c,median=0;
3        printf("Please enter 3 numbers separated by spaces >");
4        scanf("%d%d%d", &a, &b, &c);
5        if((a<=b && a>=c)||(a>=b && a<=c))
6            median = a;
7        else if((b<=a && b>=c)||(b>=a && b<=c))
8            median = b;
9        else
10           median = c;
11       printf("%d is med\n", median);
12       return 0;
13   }
```

Example

Input	Expected	Actual	Result
0 2 3	“2 is med”	“2 is med”	✓
0 2 1	“1 is med”	“0 is med”	✓
8 2 6	“6 is med”	“0 is med”	✓

Full Repair!

```
1    int main() {
2        int a,b,c,median=0;
3        printf("Please enter 3 numbers separated by spaces >");
4        scanf("%d%d%d", &a, &b, &c);
5        if((a<=b && a>=c)||(a>=b && a<=c))
6            median = a;
7        else if((b<=a && b>=c)||(b>=a && b<=c))
8            median = b;
9        else
10           median = c;
11       printf("%d is med\n", median);
12       return 0;
13   }
```

Example

Input	Expected	Actual	Result
0 2 3	“2 is med”	“2 is med”	✓
0 2 1	“1 is med”	“0 is med”	✓
8 2 6	“6 is med”	“0 is med”	X

Partial Repair

```
1    int main() {
2         int a,b,c,median=0;
3         printf("Please enter 3 numbers separated by spaces >");
4         scanf("%d%d%d", &a, &b, &c);
5         if((a<=b && a>=c)||(a>=b && a<=c))
6            median = a;
7         else if((b<=a && b>=c)||(b>=a && b<=c))
8            median = b;
9         else if((c<=b && a<=c)||(c>=b && a<=c))
10           median = c;
11        printf("%d is med\n", median);
12        return 0;
13   }
```

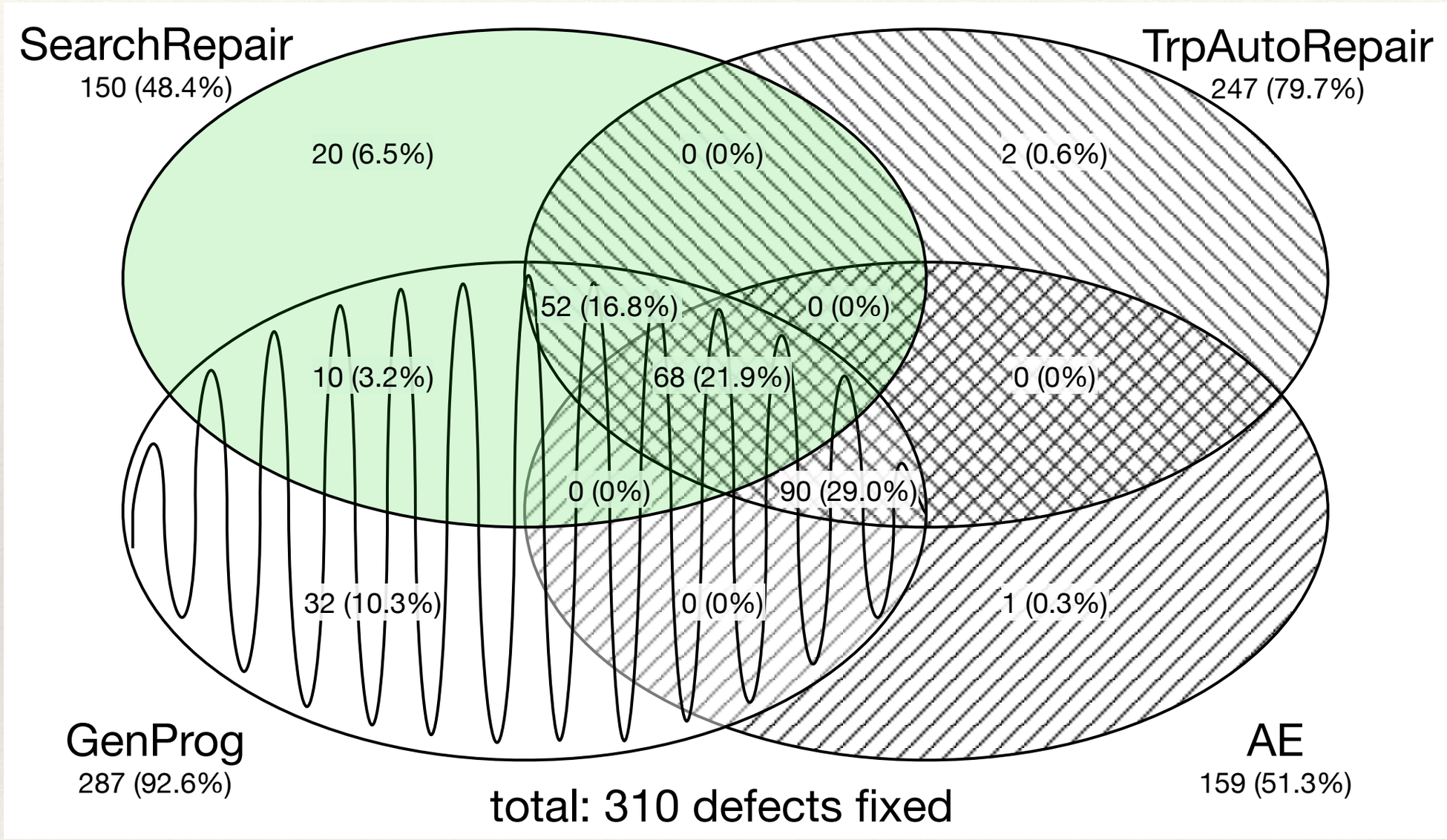
Evaluation

- 778 total versions for six small C programs
- Compared to state-of-the-art program repair tools: GenProg, TrpAutoRepair and AE

IntroClass Data Set

Program	Defects	Tests	Description
checksum	29	6	check sum of a string
digits	91	6	digits of a number
grade	226	9	grade from score
median	168	7	median of three numbers
smallest	155	8	smallest of four numbers
syllables	109	6	count vowels in string
Total	778	42	

Defects Fixed



Hypothesis:

using human-written code to construct patches and performing repair at a higher granularity level leads to better patches.

Independent Test Suite

- Generated from KLEE run over oracle programs for branch coverage
- Measures how well patches adhere to an unwritten specification — repair quality

SearchRepair	GenProg	TrpAutoRepair	AE
97.2%	68.7%	72.1%	64.2%

Take-Home Message

- SearchRepair can repair faults not repairable by state-of-the-art *generate-and-validate* repair techniques
- Patches are less prone to overfitting

Open Questions

- What is the best level of granularity for the patches?
- Can this scale to larger/complex programs?
- Can we lose precision in the matching and still find useful patches?

Making software development easier, one search at a time

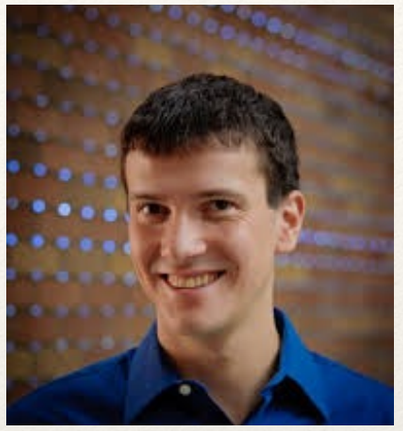
- We've explored:
  - Linguistic properties of code searches
  - Contexts of code searches
  - An example-based approach to code search
  - Code-search-enabled program repair
- And there's still a lot more to be done!

Acknowledgements

- **Funding**:
  - [2018-2023] NSF CAREER #1749936: On the Foundations of Semantic Code Search
  - [2016-2020] NSF SHF Medium #1645136: Collaborative Research: Semi and Fully Automated Program Repair and Synthesis via Semantic Code Search
  - [2014-2016] NSF SHF EAGER #1446932: Collaborative Research: Demonstrating the Feasibility of Automatic Program Repair Guided by Semantic Code Search.

Acknowledgements



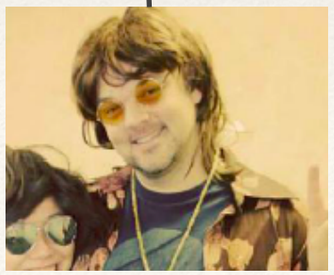



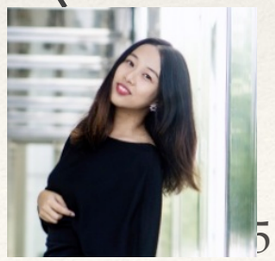
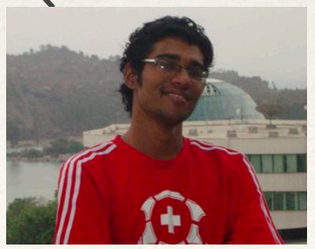








Questions?

(P.S. come to grad school at NCSU!)