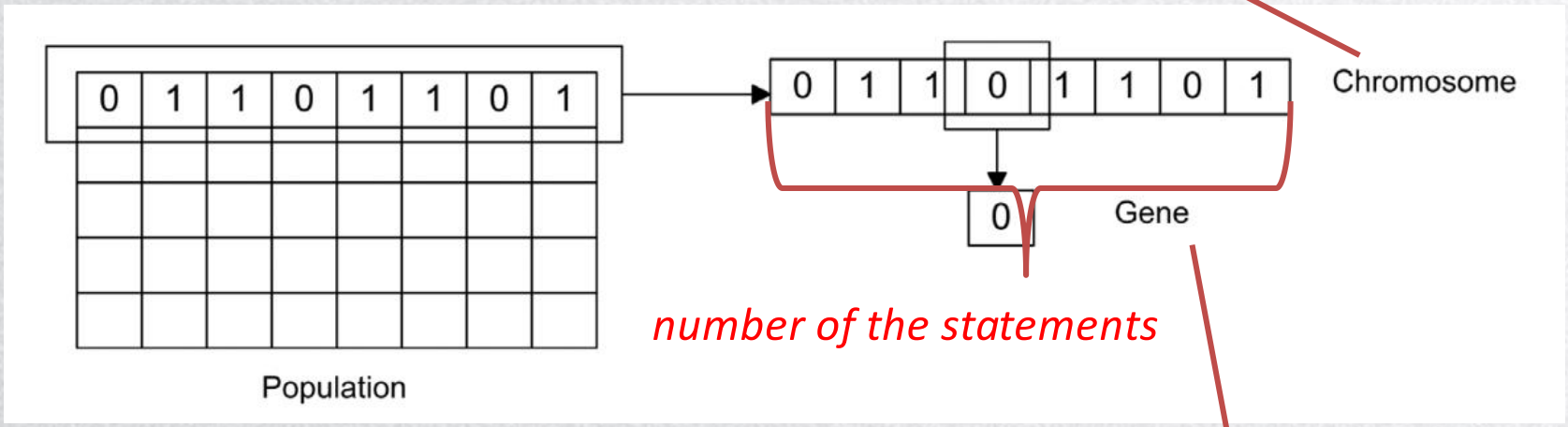
《Spectrum-based multi-fault localization using Chaotic Genetic Algorithm》	2021	IST
《Practical program repair via bytecode mutation》	2019	ISSTA
《Improving Feature Location by Enhancing Source Code with Stereotypes》	2013	ICSM

Spectrum-based multi-fault localization using Chaotic Genetic Algorithm

2021 IST

the list of all statements present in the given program



number of the statements

each statement's suspiciousness

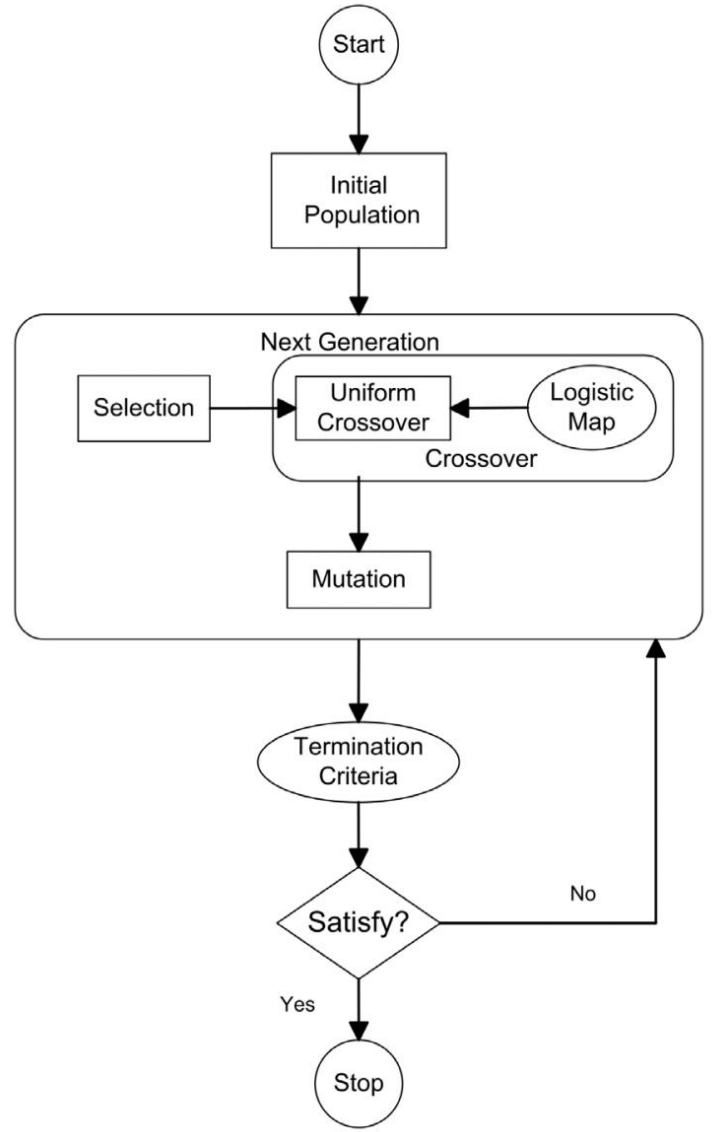


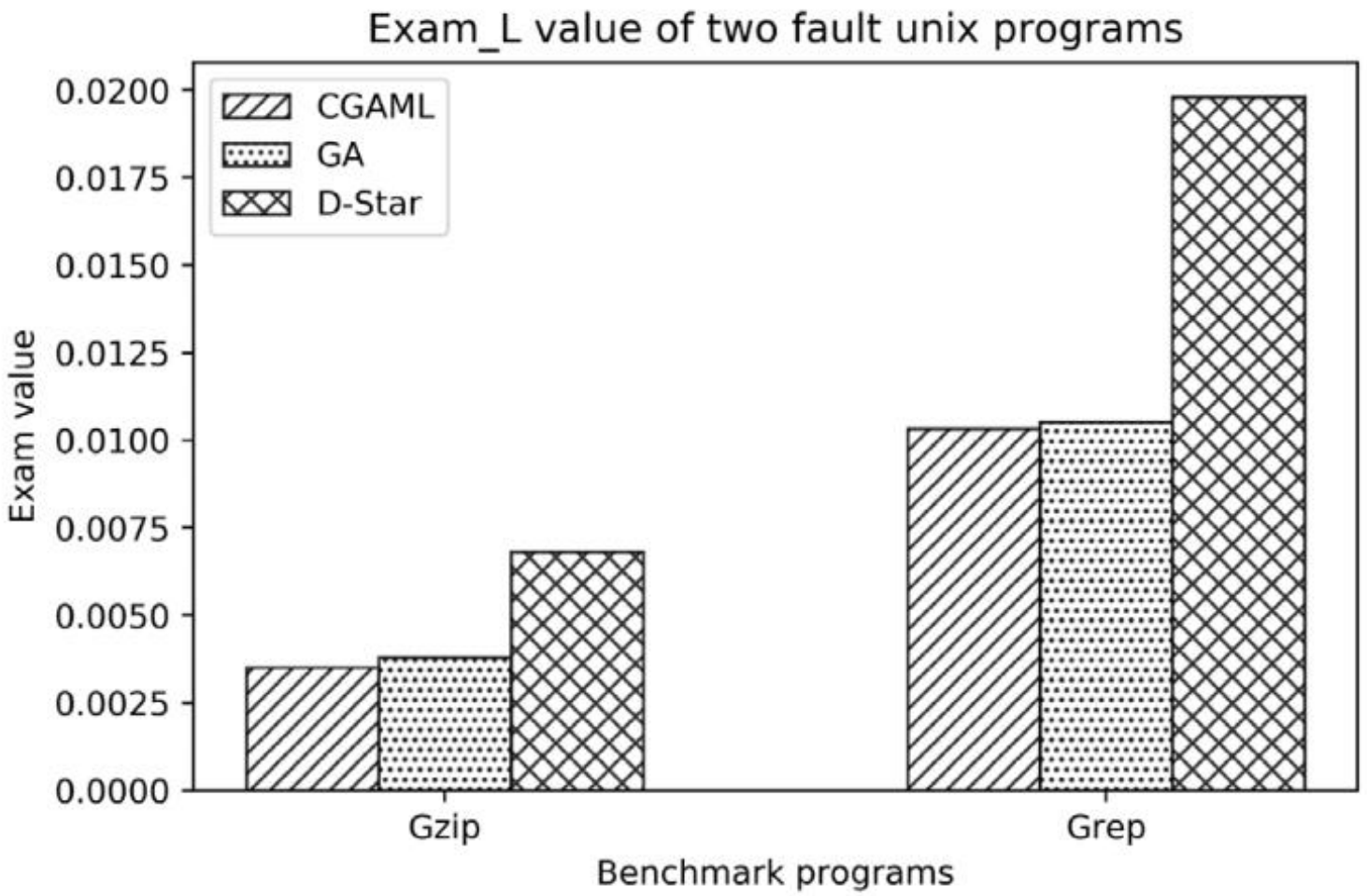
Fig. 5. Flow-chart of CGAML.

Spectrum-based multi-fault localization using Chaotic Genetic Algorithm

2021 IST

Table 5
An examp

Find secc											] FSMFL [21]	CGAML
1.int find												
2.int max												
3.if(a<=b											10	6
4. max=b											2	12
5. min=a											3	13
6.}else {												
7. max=a											12	7
8. min=b											13	1
9.}												
10.if(c>m											11	10
11. temp											14	14
12. max=											15	5
13. } else											7	9
14.											5	4
15. min=											6	11
16. } else												
17.											4	3
18. }												
19. if d<											9	15
20. retu											16	16
21. } els											8	8
22. retu											17	17
23. } else												
24. retur											1	2
25. }												
26. }												
Test Case	Result	P	P	F	F	P	P	P	P	P	P	



(b) EXAM_L score of two-fault Unix programs

Practical Program Repair via Bytecode Mutation

2019 ISSTA

Algorithm 1: PraPR

Input: Original buggy program $\mathcal{P}$, failing tests $\mathcal{T}_f$, passing tests $\mathcal{T}_p$

Output: Plausible patch set $\mathbb{P}$,

```
1  begin
2      ℒ
3      ℙ ←
       /*                                                    */
4      for
5
6
7
       /*                                                    */
8
9
10
11
12
       /*                                                    */
13
14
15
16
17
18         ℙ✓ ← ℙ✓ ∪ {P'} // Store current plausible patch
19     return ℙ✓ // Return the resulting patch set
```

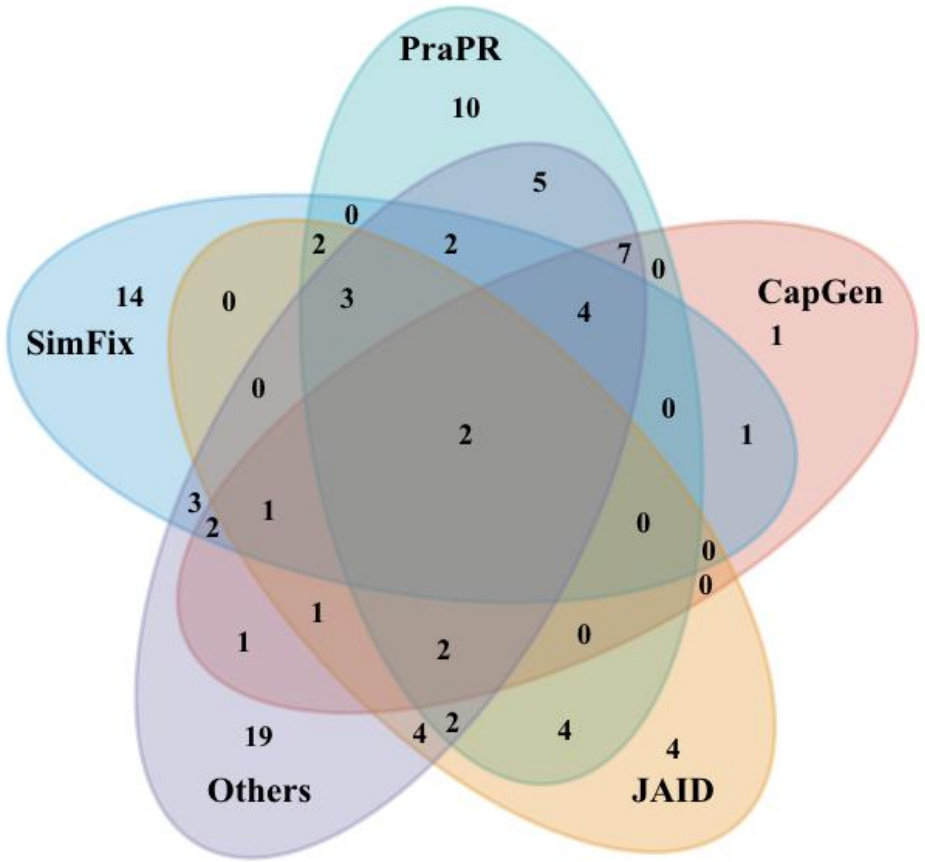


Figure 6: Fixed bug dist.

PraMutator

Computes and ranks the suspicious program locations

Generates candidate patches

Iterate to find plausible patches

Improving Feature Location by Enhancing Source Code with Stereotypes

2013 ICSM

Author：Nouh Alhindawi, Natalia Dragan, Michael L. Collard, Jonathan I. Maletic

汇报人：陈冰婷　　导师：邹卫琴

目录
Contents

Motivation

Hypothesis: *the stereotype information is relevant and will improve the results in the context of feature location.*

Stereotypes *are terms that describe the abstract role of a method*

Stereotype

“Stereotypes are a concise abstraction of a method's role and responsibility in a class and system.”

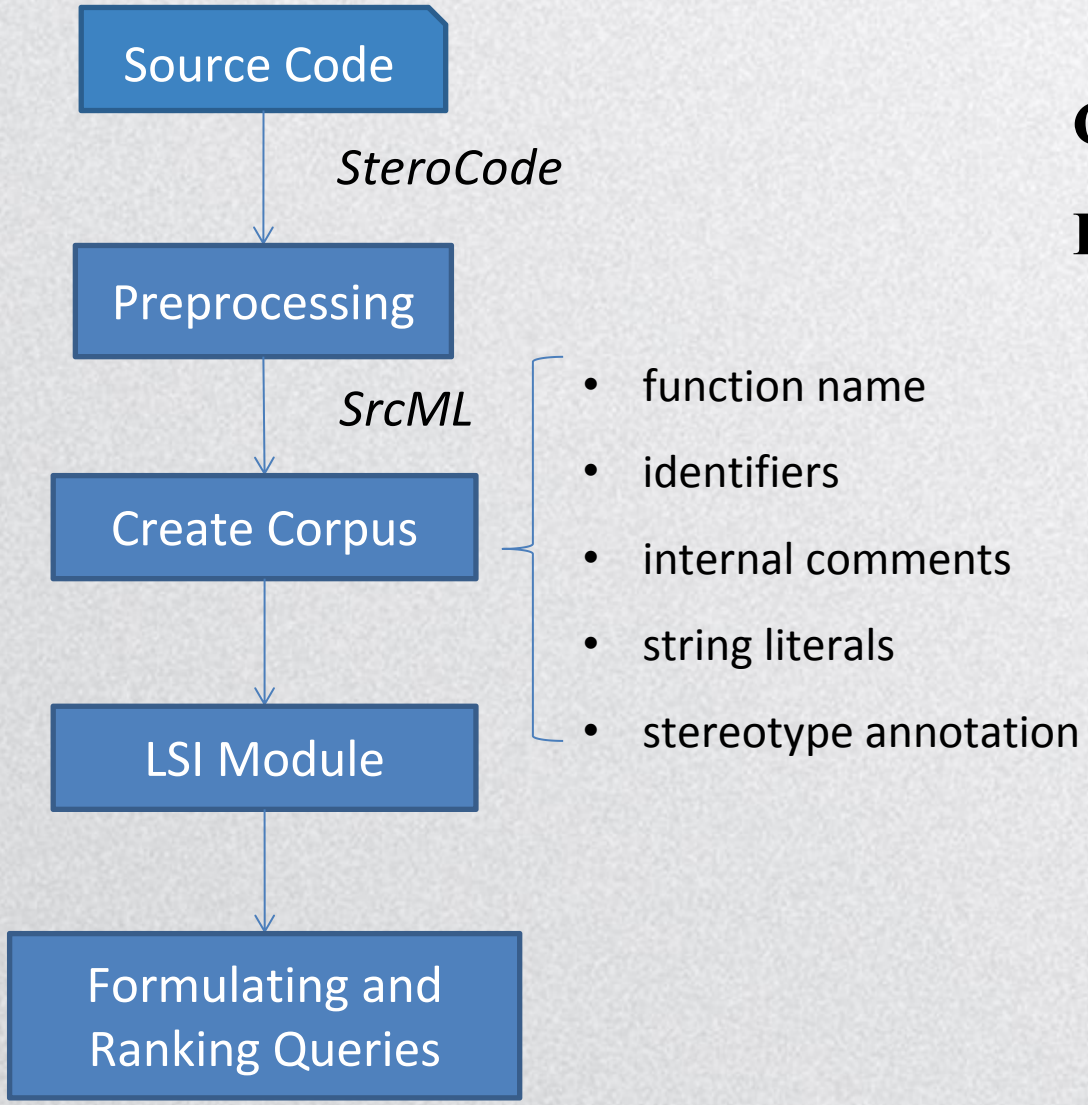
```
class  DataSource :public Observable
{
…
public:
  /** @stereotype get */
  const string& getName() const;

  /** @stereotype predicate */
  bool isValidLabel(const string& label) const;

  /** @stereotype command */
  virtual void reserve(int count );
…
};
```

Figure 1. A code snippit of the HippoDraw C++ Class DataSource after re-documenting with the method stereotypes.

TABLE 1 TAXONOMY OF METHOD STEREOTYPES AS GIVEN IN [20]. THE TAXONOMY IS MAINLY FOCUSED ON THE C++ PROGRAMMING LANGAUGE. METHODS MAY BE LABELED WITH ONE OR MORE STEREOTYPES.

Stereotype Category	Stereotype	Description
Structural ***Accessor***	get	Returns a data member.
	predicate	Returns Boolean value which is not a data member.
	property	Returns information about data members.
	void-accessor	Returns information through a parameter.
Structural ***Mutator***	set	Sets a data member.
	command	Performs a complex change to the object’s state *(this)*.
	non-void-command	
Creational	constructor, copy-const, destructor, factory	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects (parameter, local variable and return object).
	controller	Changes an external object’s state (not *this*).
Degenerate	incidental	Does not read/change the object’s state.
	empty	Has no statements.

Implement



Corpus: HippoDraw、Qt

Evaluation Metric： Recall and Precision

- $\sum EM$: Total Effort Measurement (number of methods the developer needs to investigate to find all relevant documents).
- PFR: Position of first relevant document.
- PLR: Position of last relevant document.

TABLE 2. DETAILS OF THE CORPUS USED AS INPUT TO LSI FOR EACH OF THE TWO SYSTEMS USED IN THE EXPERIMENTAL STUDY.

	HippoDraw 1.21.3	Qt 4.4.3
Vocabulary Size	6,803	91,187
Number of Parsed Documents/Methods	3,706	70,871
Dimensionality Used	200	300

Evaluation

HippoDraw

TABLE 3. HIPPODRAW FEATURE DESCRIPTION, APPLIED QUERY, AND THE NUMBER OF RELEVANT METHODS FOR EACH FEATURE.

Feature	Query	Number Relevant Methods
1. change font size	*change font size weight set*	10
2. change font style	*change font style italic*	18
3. update zoom mode	*update zoom mode zoomin zoomout*	9
4. reset printer settings	*reset change printer settings*	8
5. add item	*insert add item canvas*	7
6. remove item	*Delete remove item canvas*	7
7. change mouse property	*Option change mouse property*	9
8. change cut color	*change cut color set*	7
9. change representation color	*change representation color set*	7
10. make new display	*make new display add make*	12
11. update axis modeling	*update axis modeling reset*	8

TABLE 6. RESULT OF QT FOR THREE MEASUREMENTS; TOTAL EFFORT MEASUREMENT (EM), POSITION OF FIRST RELEVANT DOCUMENT (PFR), AND POSITION OF LAST RELEVANT DOCUMENT (PLR).

Feature	Total Effort Measurement (Σ EM)		First Relevant Document (PFR)		Last Relevant Document (PLR)	
	LSI	LSI+S	LSI	LSI+S	LSI	LSI+S
1	2208	1846	2	1	1054	332
2	1900	928	1	1	520	467
3	1668	1192	4	1	684	443
4	1760	996	4	1	710	359
5	112	100	19	8	59	40
6	2792	1667	2	1	830	451
7	251	149	1	1	101	94
8	1239	701	3	1	815	456
9	359	185	1	1	153	100
10	1078	599	2	1	184	150
11	1641	566	1	1	1321	450

Evaluation

HippoDraw

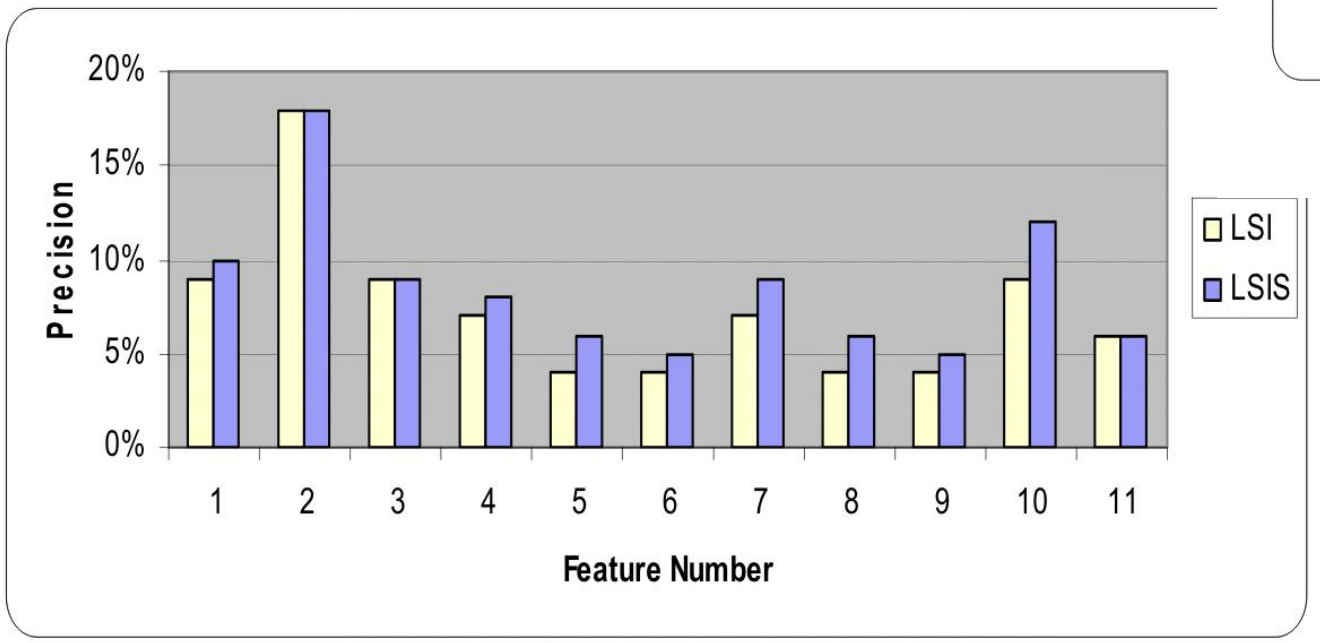


Figure 2. Precision results for the HippoDraw.

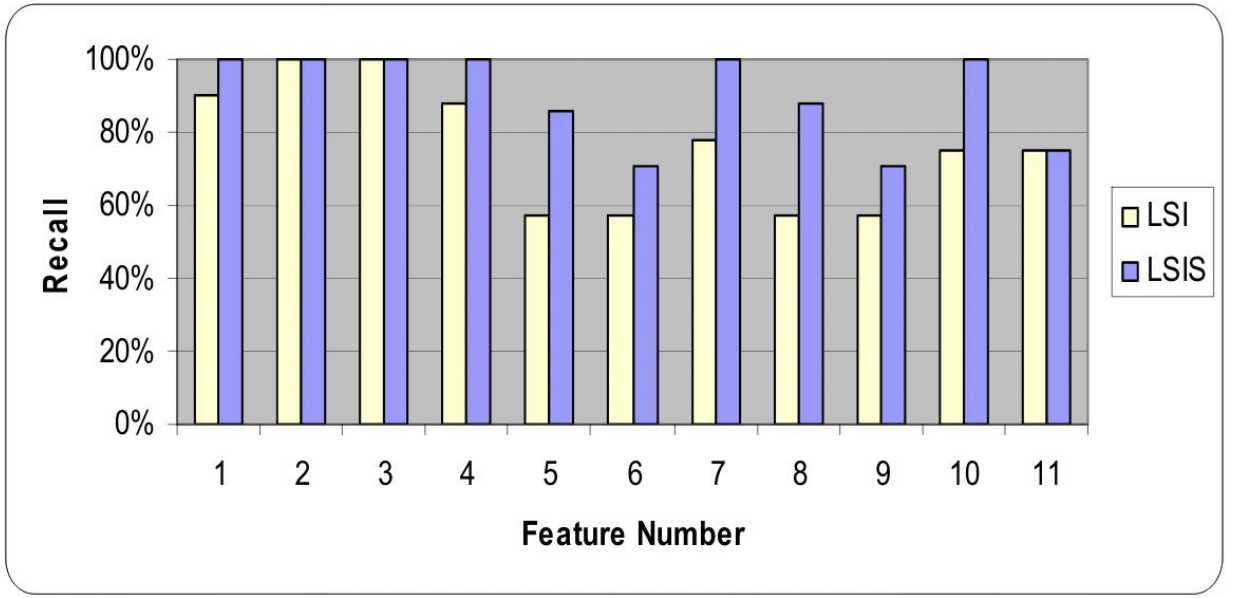


Figure 3. Recall results for the HippoDraw .

Evaluation

Qt

TABLE 5. QT FEATURE DESCRIPTIONS; FEATURE NAME, APPLIED QUERY, AND NUMBER OF RELEVANT METHODS TO EACH FEATURE.

Feature	Query	Number Relevant Methods
1. update font settings	*font update options settings reset*	21
2. create new font	*create new font*	24
3. change font size	*size font change*	23
4. set password	*set password change*	12
5. set RGB	*update RGB color RGBA RGBF*	7
6. add menu	*add create new menu insert menubar*	15
7. remove menu	*menu remove delete*	7
8. add action	*insert action add new*	11
9. remove action	*action delete remove*	9
10. search	*index search searching searcher indexing find*	12
11. draw polygon	*points polygon draw lines polyline*	7

TABLE 6. RESULT OF QT FOR THREE MEASUREMENTS; TOTAL EFFORT MEASUREMENT (EM), POSITION OF FIRST RELEVANT DOCUMENT (PFR), AND POSITION OF LAST RELEVANT DOCUMENT (PLR).

Feature	Total Effort Measurement (Σ EM)		First Relevant Document (PFR)		Last Relevant Document (PLR)	
	LSI	LSI+S	LSI	LSI+S	LSI	LSI+S
1	2208	1846	2	1	1054	332
2	1900	928	1	1	520	467
3	1668	1192	4	1	684	443
4	1760	996	4	1	710	359
5	112	100	19	8	59	40
6	2792	1667	2	1	830	451
7	251	149	1	1	101	94
8	1239	701	3	1	815	456
9	359	185	1	1	153	100
10	1078	599	2	1	184	150
11	1641	566	1	1	1321	450

Evaluation

Qt

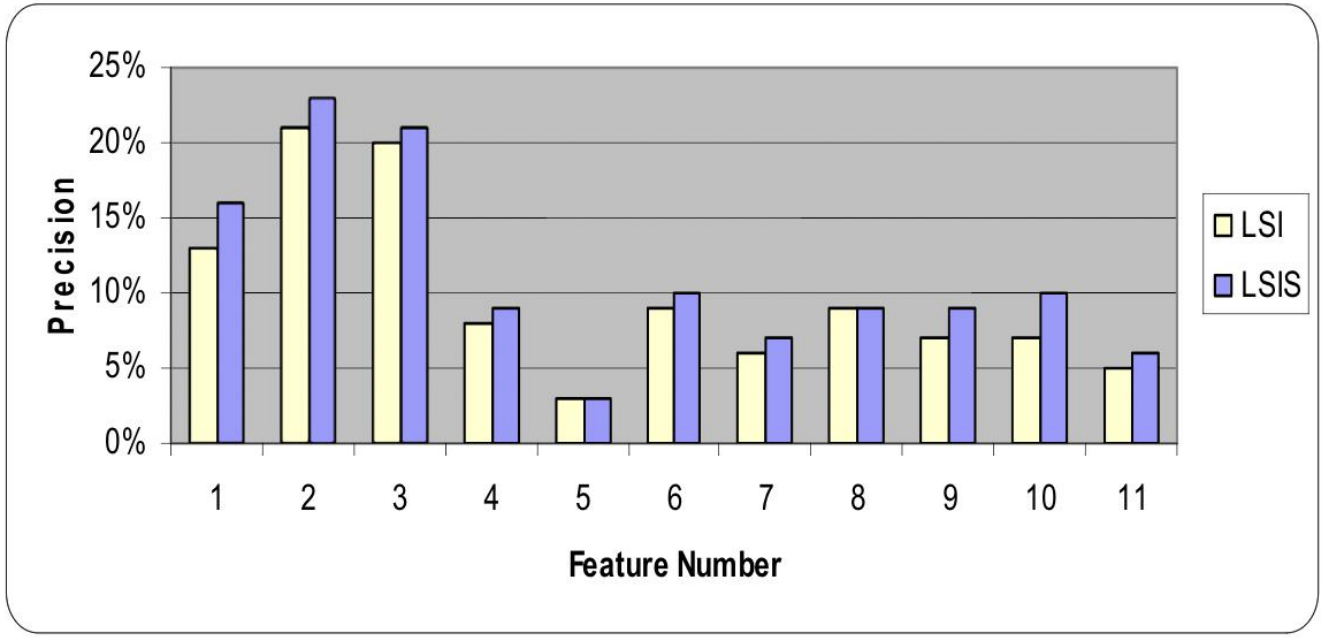


Figure 4. Precision for the 11 features from Qt.

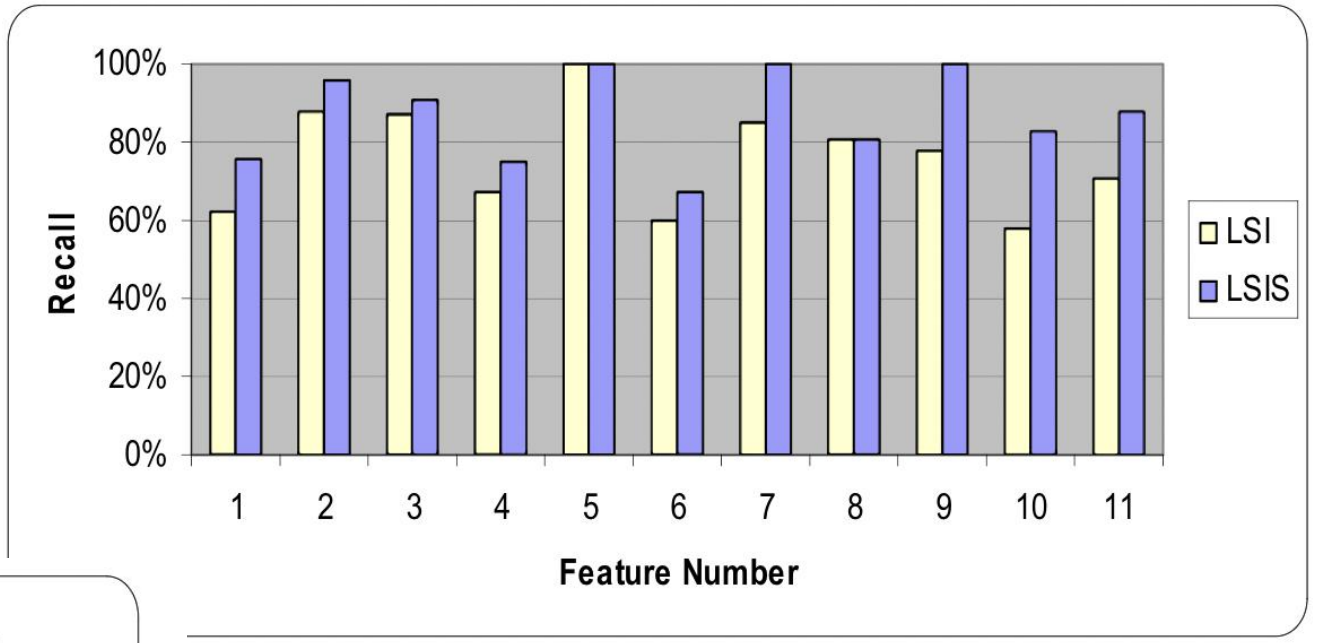


Figure 5. Recall for the 11 features from Qt .

THANK YOU FOR YOUR LISTENING.

谢谢您的聆听