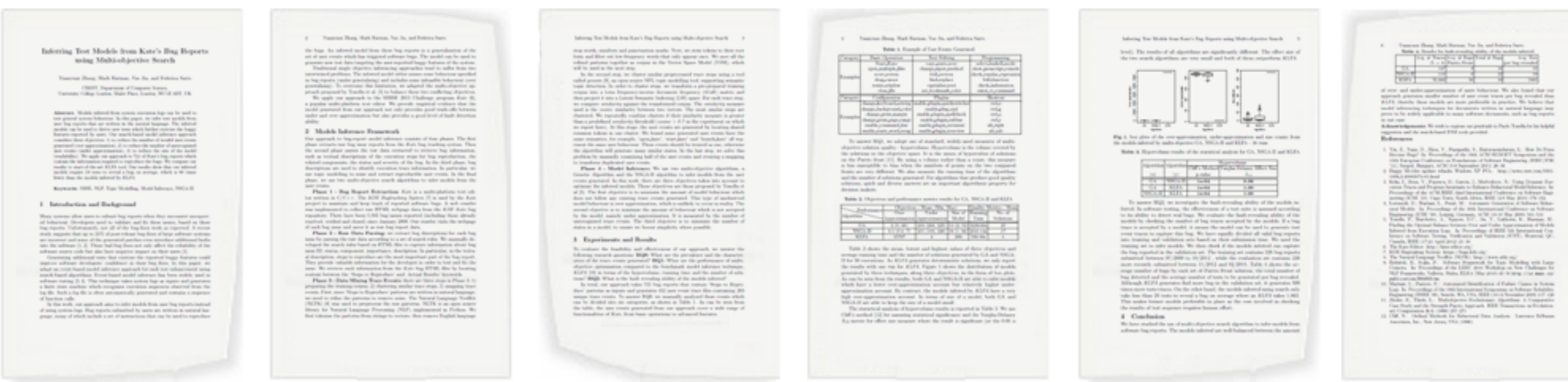
Inferring Test Models from Kate's Bug Reports using Multi-objective Search

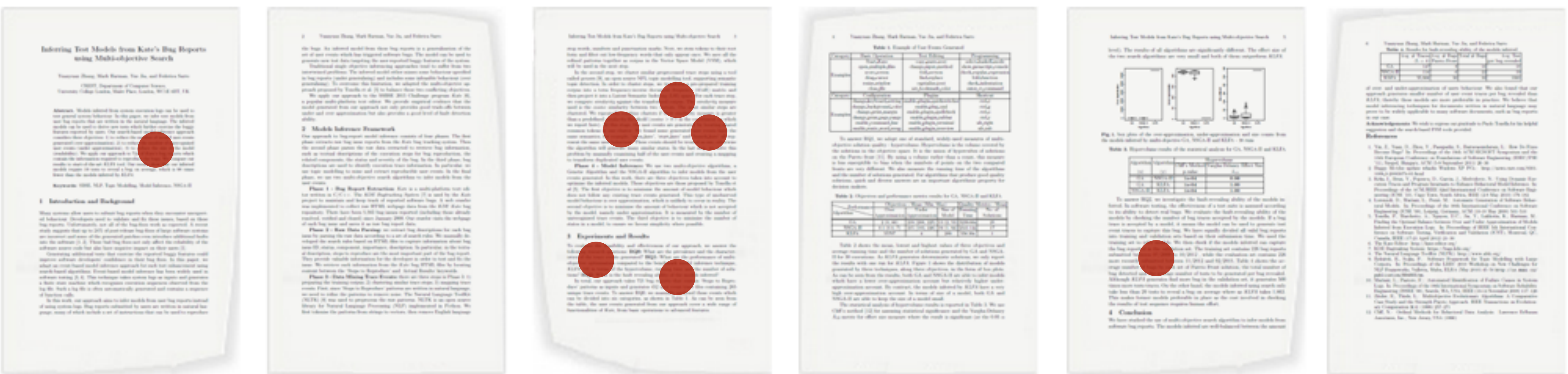
Yuanyuan Zhang Mark Harman Yue Jia and Federica Sarro

University College London

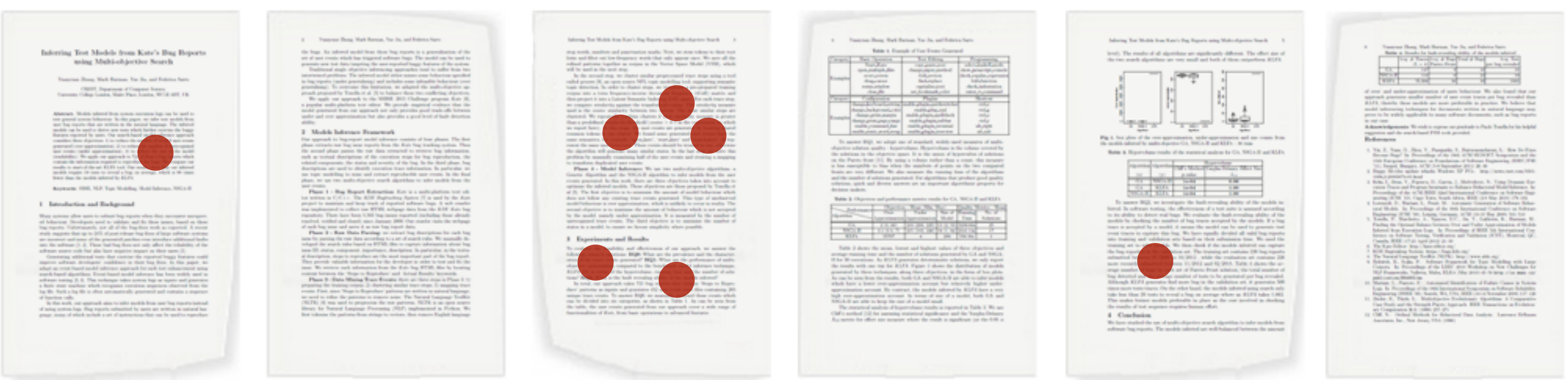
SSBSE 2015 Challenge Track Paper



Inferring Test Models from Kate's Bug Reports using Multi-objective Search


Inferring Test Models from Kate's Bug Reports using Multi-objective Search

Which page should we put more effort to proofread?



Motivation

ID ▲	Product	Comp	Assignee ▲	Status ▲	Resolution ▼	Summary	Changed ▼
349015	kate	general	kwrite-bugs-null	UNCO	---	Upgrading from kate4 does not provide any upgrading instructions, loses all data	07:48:33
349006	kate	kwrite	kwrite-bugs-null	UNCO	---	KWrite and Kate have the same configuration	23:35:03
348977	kate	general	kwrite-bugs-null	UNCO	---	Kate crashes trying to open file with <SPC><LF>	Wed 13:49
344341	kate	syntax	kwrite-bugs-null	UNCO	---	Kate ignores custom syntax highlighting xml files	Wed 12:29
348317	kate	syntax	kwrite-bugs-null	UNCO	---	[PATCH] Katepart syntax highlighting should recognize \u0123 style escapes for JavaScript	Tue 20:53
205447	kate	part	kwrite-bugs-null	UNCO	---	[BiDi/Unicode] Non-BMP characters are incorrectly handled	Tue 19:00
348934	kate	kwrite	kwrite-bugs-null	UNCO	---	Kwrite hangs while opening non-empty files via sftp	Tue 16:28
[illegible]	kate	session	kwrite-bugs-null	UNCO	[illegible]	[illegible]	[illegible]
347147	kate	syntax	kwrite-bugs-null	UNCO	---	keyword missing in csound	Mon 20:23
[illegible]	[illegible]	general	kwrite-bugs-null	UNCO	---	[illegible]	[illegible]
348846	kate	Vi Input	kwrite-bugs-null	UNCO	---	cannot use the CTRL+W+Direction shortcut anymore	Sun 15:32
348845	kate	Vi Input	kwrite-bugs-null	UNCO	---	split and vsplit commands are reversed	Sun 15:25
348843	framewor	general	kwrite-bugs-null	UNCO	---	Editor highlights floating point number 1.039 in red in C++ code	Sun 15:15
348765	kate	syntax	kwrite-bugs-null	UNCO	---	Perl syntax highlighting is wrong when using scalar references (backslash quote)	Sat 00:12

Up to **24%** of post-release bug fixes of large software systems are **incorrect** and even **introduce additional faults**

Event-based Model Inference

Input: an abstraction of the observed sequences

- log files (contain a sequence of execution traces function calls)

<println, formatter, close, println>

- bug reports (written in the natural language)

Output: an inferred model

- a FSM (accepts more traces than the observed ones and might not accept some of the observed traces)

Event-based Model Inference Challenges

- bug reports (written in the natural language)

a "profane" user upgrading from kate4 to kate5 gets is development settings completely wiped out as a result of the upgrade.

A description of a bug

Reproducible: Always

Reproducible or not?

Steps to Reproduce:
1. Install KDE4/kate4.x
2. Save your settings/data
3.Upgrade to KDE5/kate5.0.0

Steps to Reproduce

Actual Results:
- All the session data is lost
- All the custom syntax files are lost
- Probably more settings i didn't use are lost, everything saved into ~/.kde/* in general

Actual Results

Expected Results:
Kate5 showing a BIG warr the user data is lost and what is changed and instructing the user on how to recover it:
- ~/.kde is now ~/.local
- copy ~/.kde/share/apps/kate as ~/.local/share/kate

Expected Results

Event-based Model Inference Challenges

the generalisation capability of a model

- a FSM (accepts more traces than the observed ones and might not accept some of the observed traces)

introduce infeasible behaviours

exclude possible behaviours

over-generalisation
over-approximation

under-generalisation
under-approximation

Model Inference Framework

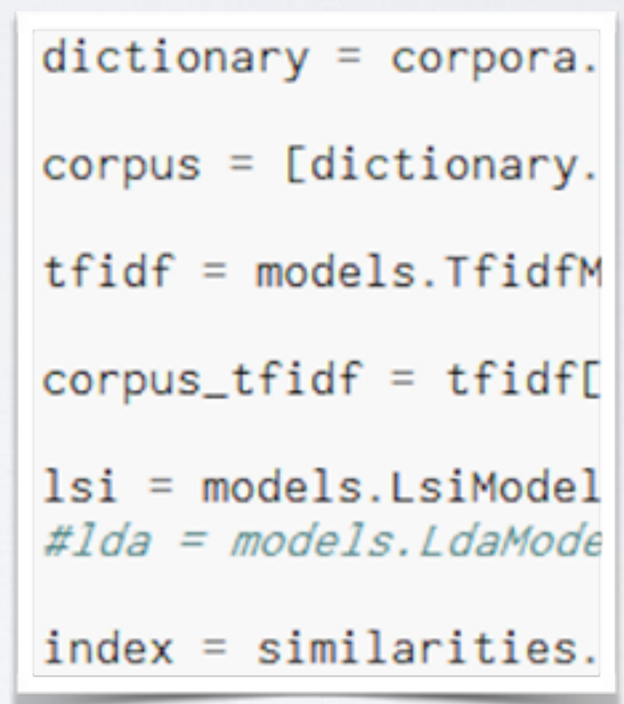
Extraction

1

Parsing

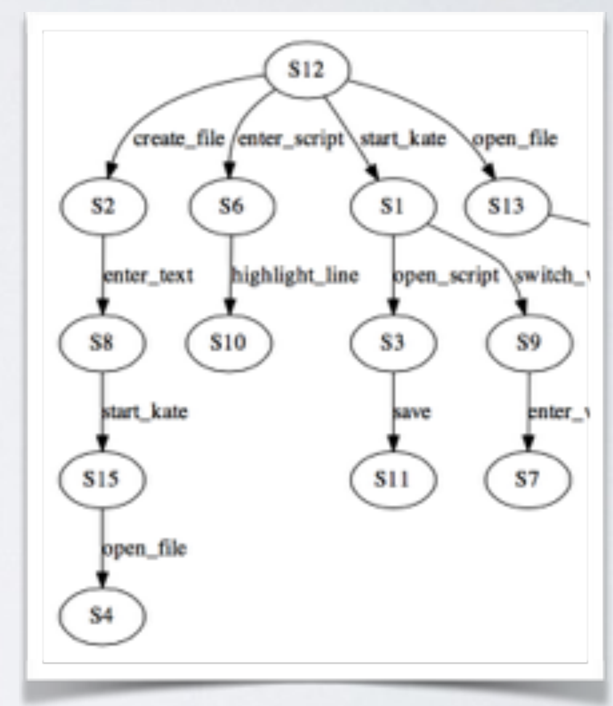
2

Data Mining



3

Seach-Based Model Generation



4

Model Inference Framework

Extraction

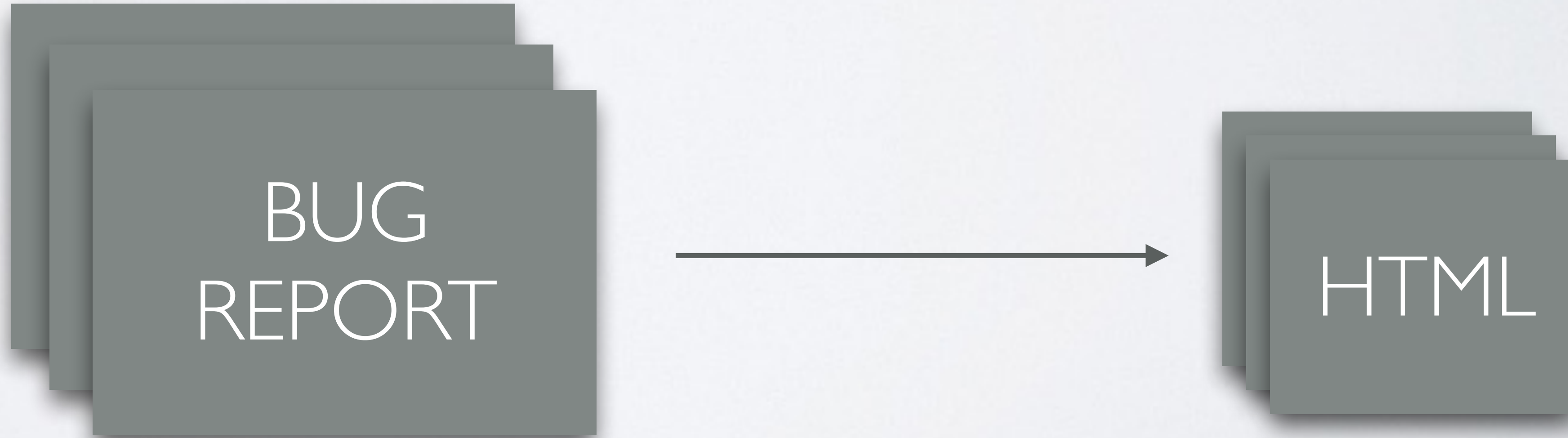
1

I. Filter Bug Reports

filter terms: product; bug status; bug severity

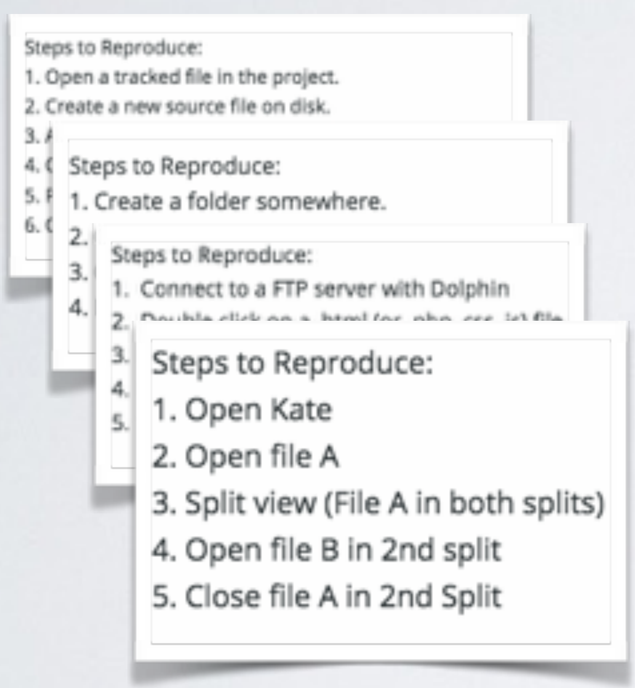
II. Extract All Valid Bug Report IDs

III. Crawl Bug Reports



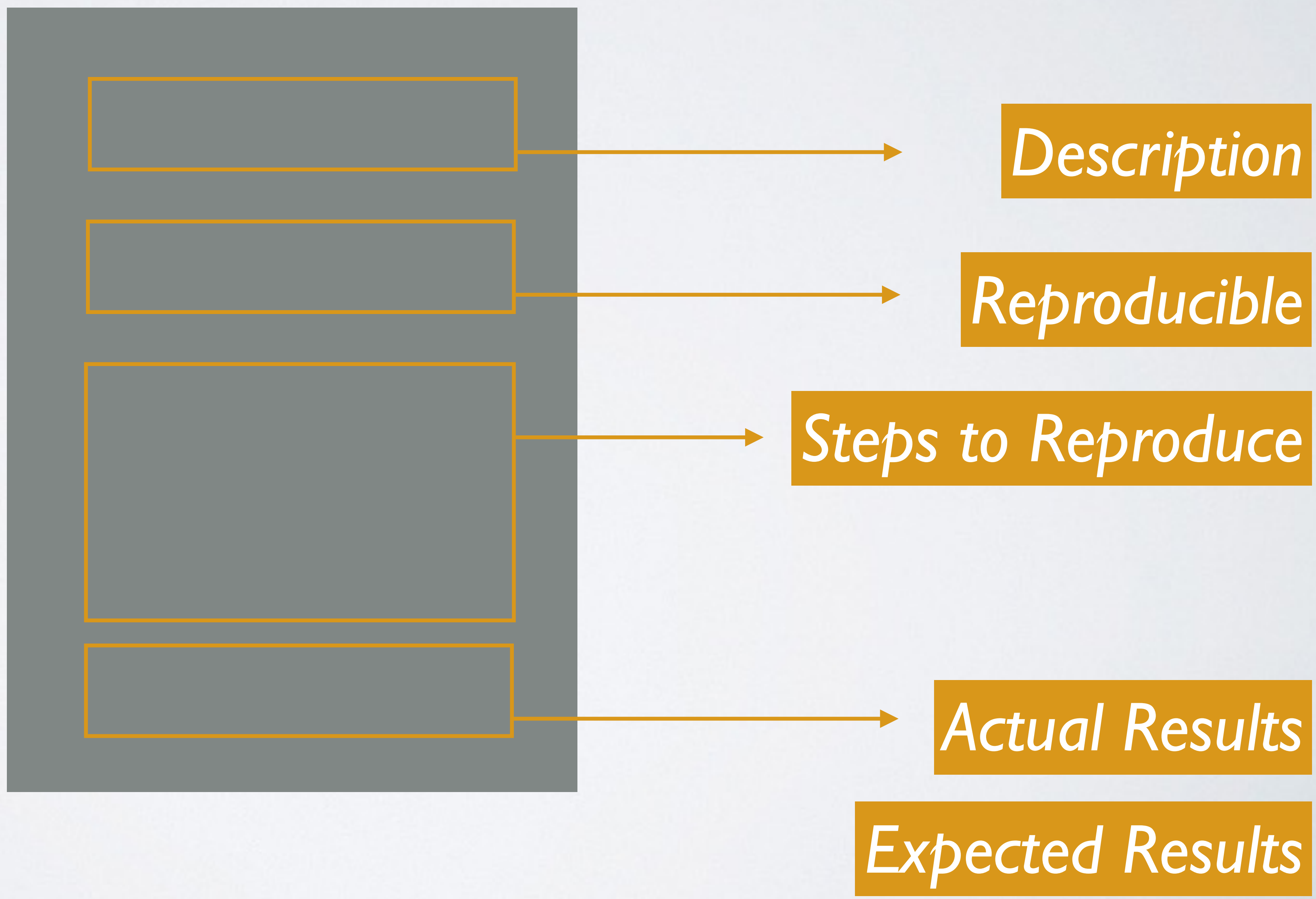
Model Inference Framework

Parsing



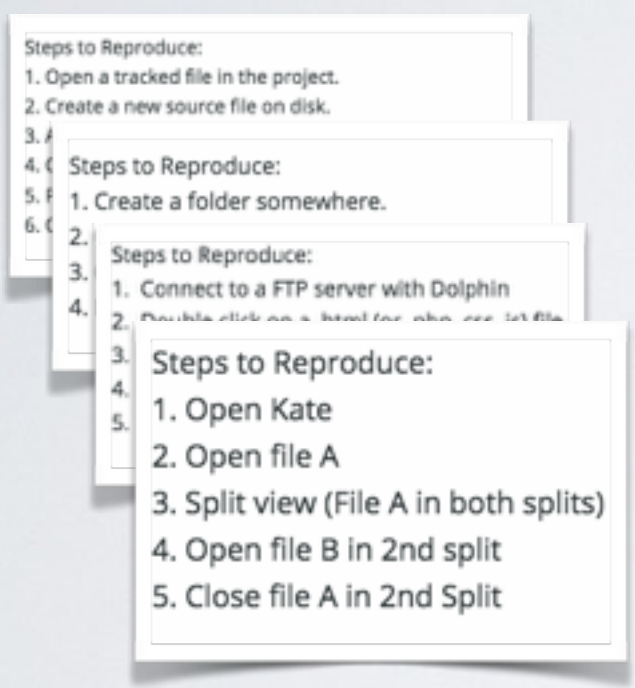
2

1. Filter Bug Reports



Model Inference Framework

Parsing


Steps to Reproduce:
1. Open a tracked file in the project.
2. Create a new source file on disk.

Steps to Reproduce:
1. Create a folder somewhere.

Steps to Reproduce:
1. Connect to a FTP server with Dolphin

Steps to Reproduce:
1. Open Kate
2. Open file A
3. Split view (File A in both splits)
4. Open file B in 2nd split
5. Close file A in 2nd Split

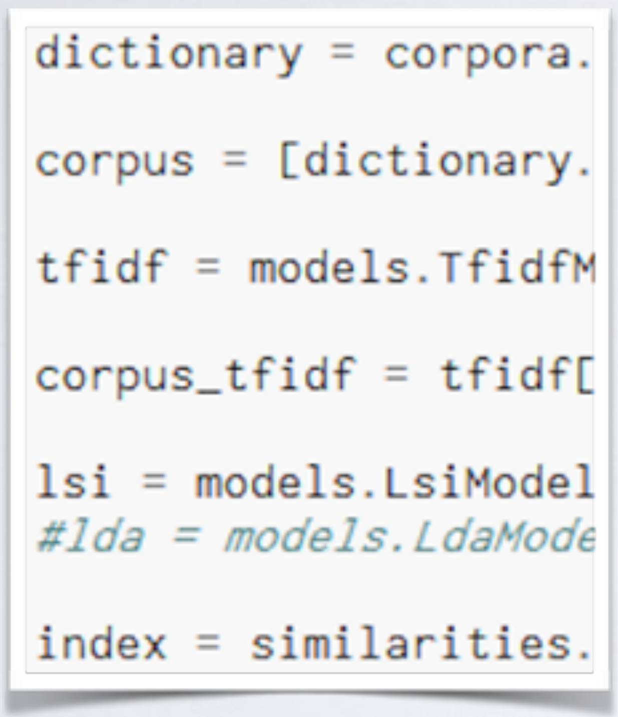
2

I. Filter Bug Reports

II. OUTPUT .str Files

Model Inference Framework

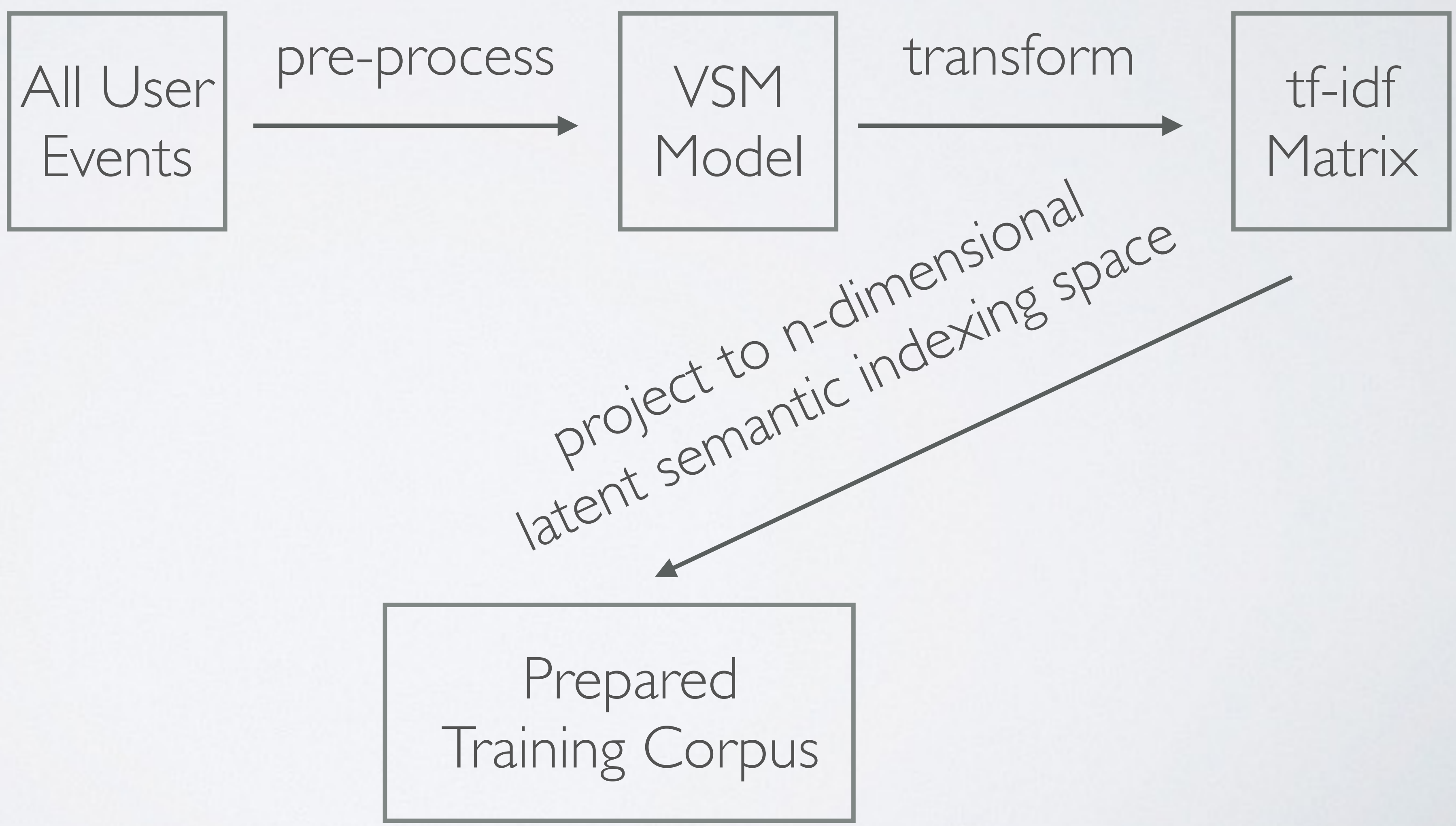
Data Mining



```
dictionary = corpora.
corpus = [dictionary.
tfidf = models.TfidfM
corpus_tfidf = tfidf[
lsi = models.LsiModel
#lda = models.LdaMode
index = similarities.
```

3

1. Preparing Training Corpus



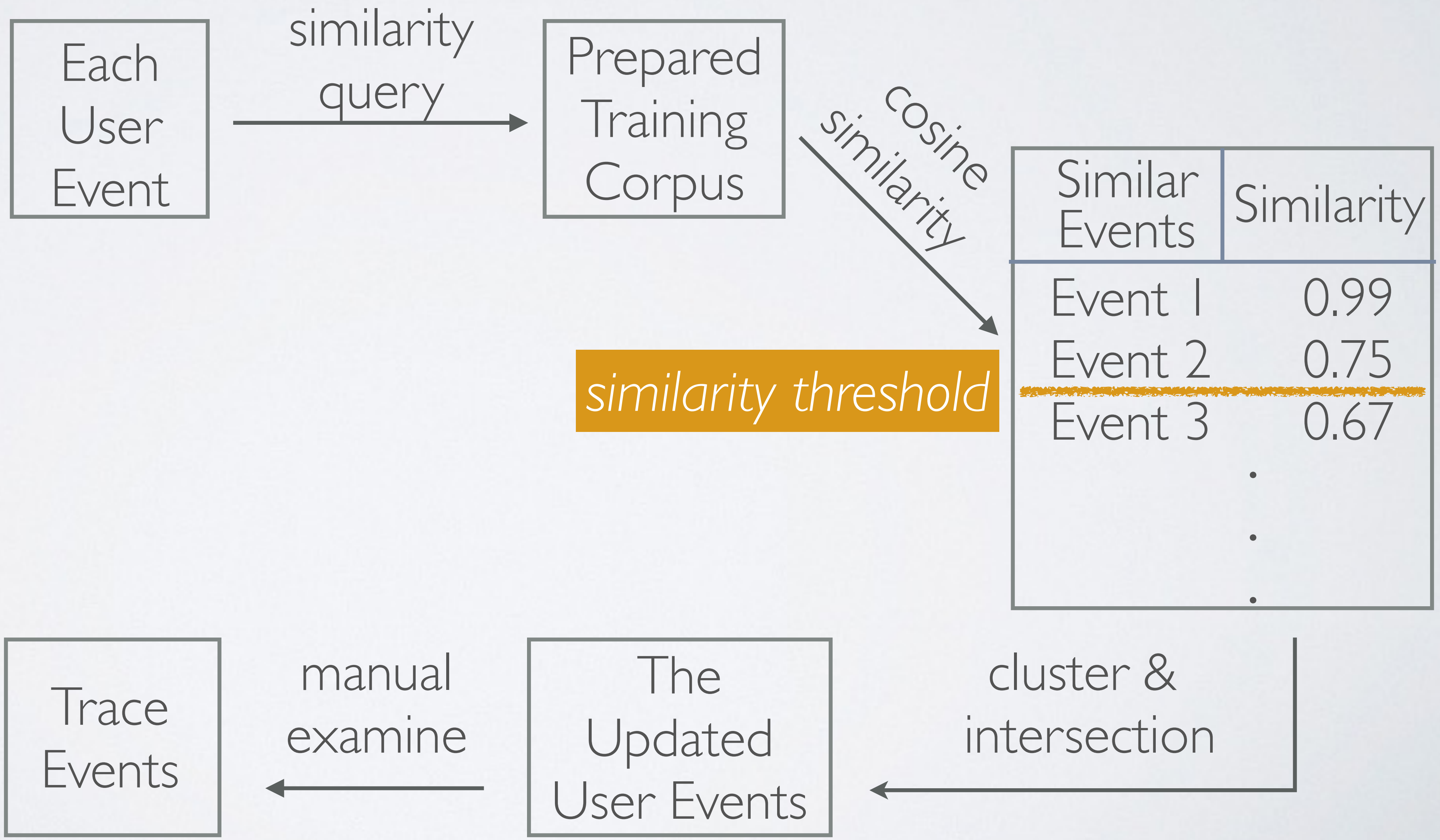
Model Inference Framework

Data Mining

```
dictionary = corpora.
corpus = [dictionary.
tfidf = models.TfidfM
corpus_tfidf = tfidf[
lsi = models.LsiModel
#lda = models.LdaMode
index = similarities.
```

3

II. Clustering Similar User Events



Model Inference Framework

Data Mining

```
dictionary = corpora.

corpus = [dictionary.

tfidf = models.TfidfM

corpus_tfidf = tfidf[

lsi = models.LsiModel
#lda = models.LdaMode

index = similarities.
```

3

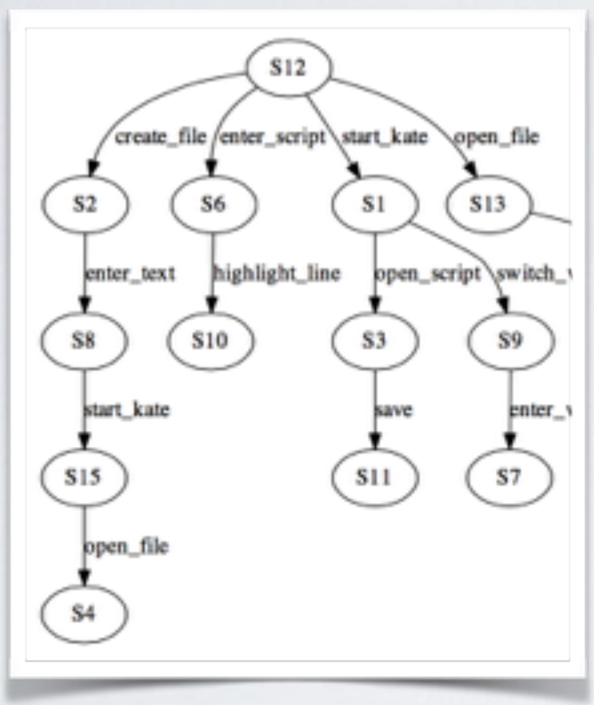
I. Preparing Training Corpus

II. Clustering Similar User Events

III. Mapping Trace Events

Model Inference Framework

Seach-Based
Model Generation



Multi-objective Fitness Functions

1. ***Minimise over-approximation***

   minimise the number of trace sequences generated from a model that do not correspond to any existing execution traces

2. ***Minimise under-approximation***

   minimise the number of trace sequences that are excluded from a model

3. ***Minimise the number of states in a model***

Tonella, P., Marchetto, A., Nguyen, D.C., Jia, Y., Lakhotia, K., Harman, M.: Finding the Optimal Balance between Over and Under Approximation of Models Inferred from Execution Logs. In: Proceedings of IEEE 5th International Con- ference on Software Testing, Verification and Validation (ICST), Montreal, QC, Canada, IEEE (17-21 April 2012) 21–30

Experimental Setting

Subject: Kate bug reports - KDE Bugtracking system

5583 bug issues reported since 2000 - 1/8 721 STR pattern

Search algorithms: multi-objective GA, NSGA-II

Benchmark tool: KLFA

Three Research Questions

RQ0: What are the prevalence and the characteristics of the trace events generated?

RQ1: What are the performance of multi-objective optimisation compared to the benchmark model inference technique, *KLFA*, in terms of the hypervolume, running time and the number of solutions?

RQ2: What is the fault revealing ability of the models inferred?

RQ0 Example of Trace Events Generated

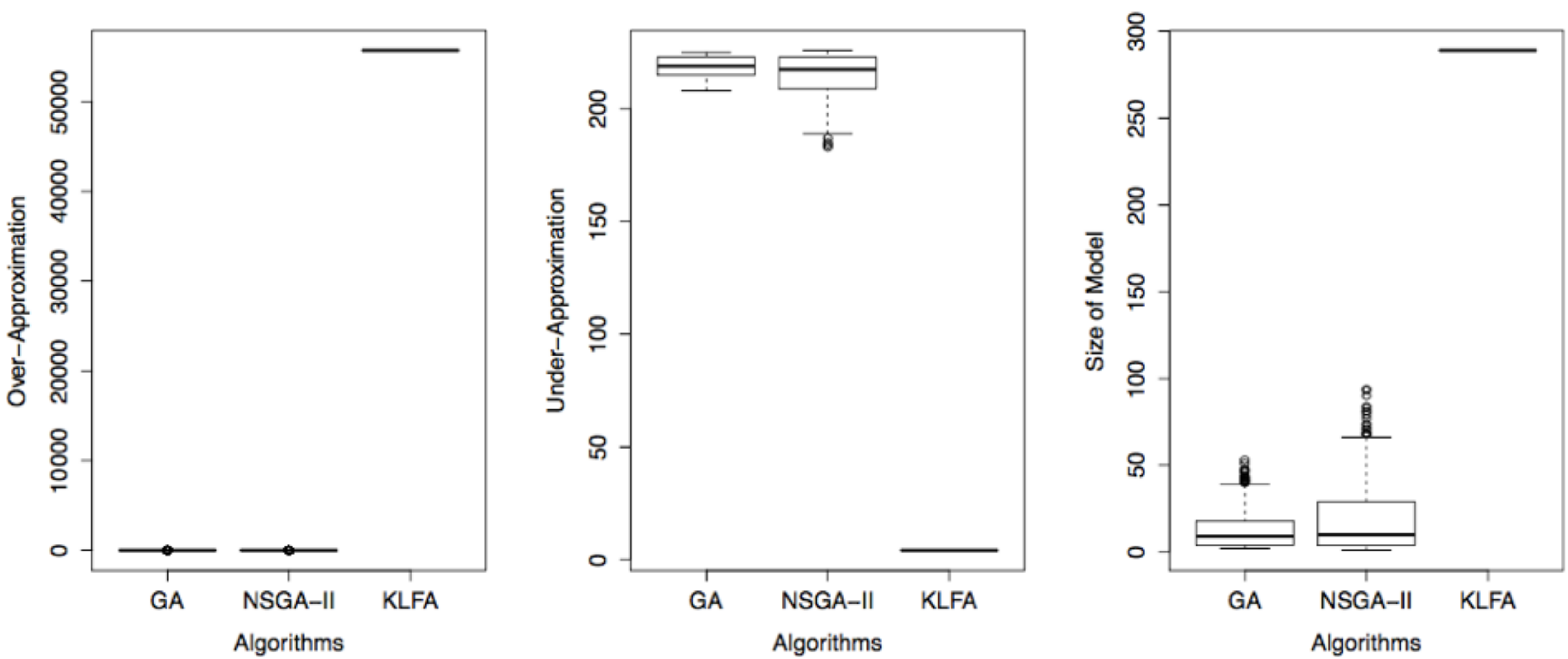
721 bug reports → 452 user events files → 265 unique trace events

Category	Basic Operation	Text Editing	Programming
Examples	Start_Kate	copy_paste_text	select_haskell_mode
	open_multiple_files	change_input_method	show_javascript_console
	score_screen	fold_section	check_regular_expression
	drag_cursor	find_replace	fold_function
	resize_window	captialize_text	check_indentation
	close_file	set_bookmark_color	enter_vi_command

Category	Configuration	Plugins	Shortcut
Examples	change_keyboard_setting	enable_plugin_quickswitcher	ctrl_1
	change_background_color	enable_plug_xml	ctrl_g
	change_print_margin	enable_plugin_spellcheck	ctrl_o
	change_print_page_range	enable_plugin_tabbar	ctrl_r
	enable_command_line	enable_plugin_terminal	alt_right
	enable_static_word_wrap	enable_plugin_treeview	alt_tab

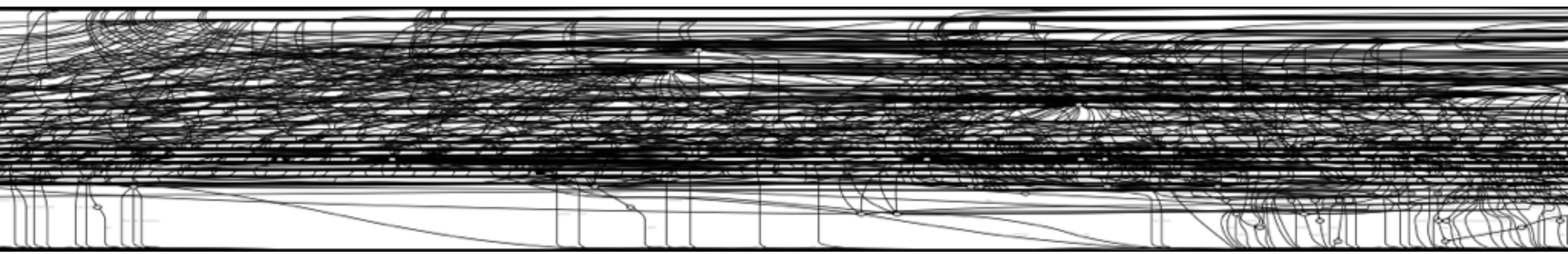
RQ1 Performance of the Algorithms - Three Objectives

Algorithm \ Performance	Objectives - Mean (Min, Max)		
	Over Approximation	Under Approximation	Size of Model
GA	2 (0, 66)	219 (208, 225)	13 (2, 53)
NSGA-II	0.1 (0.0, 7)	215 (183, 226)	19 (1, 94)
KLFA	55707	4	289

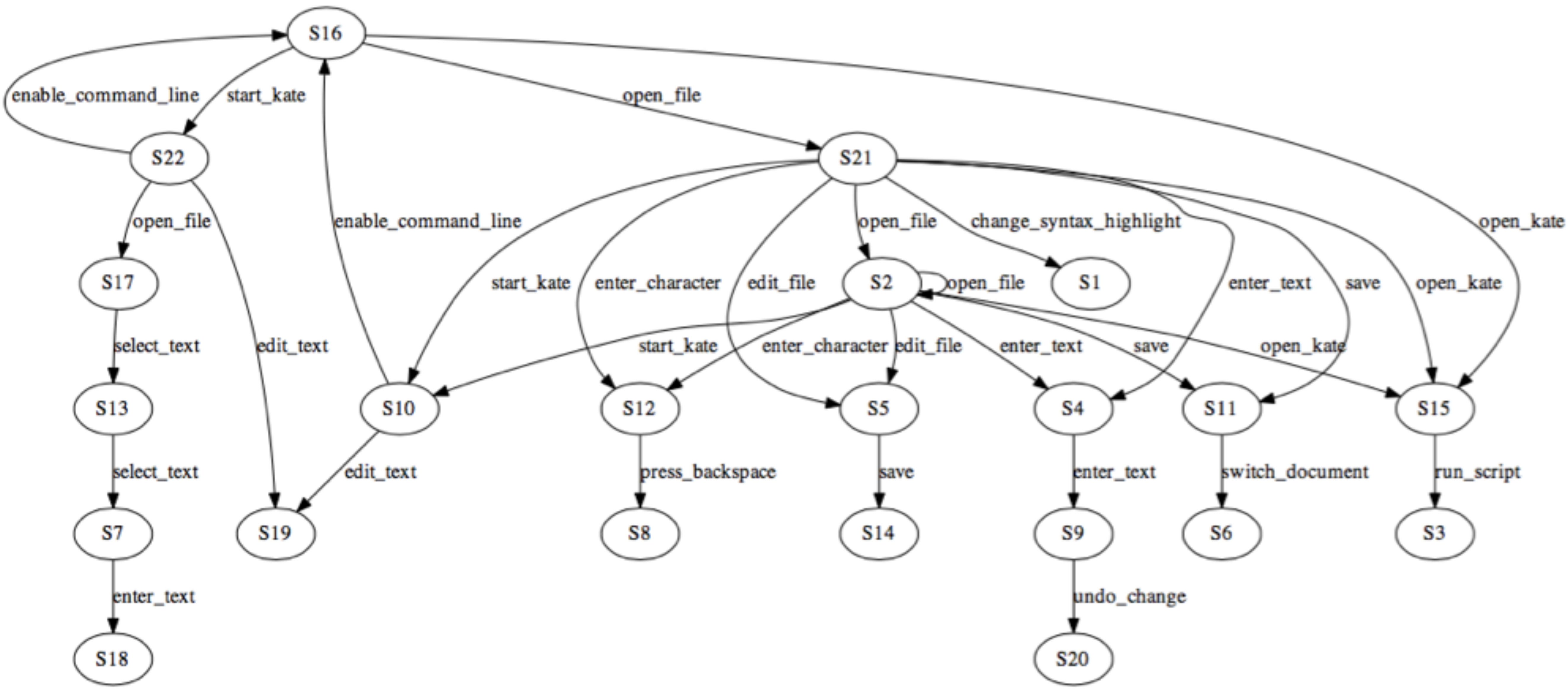


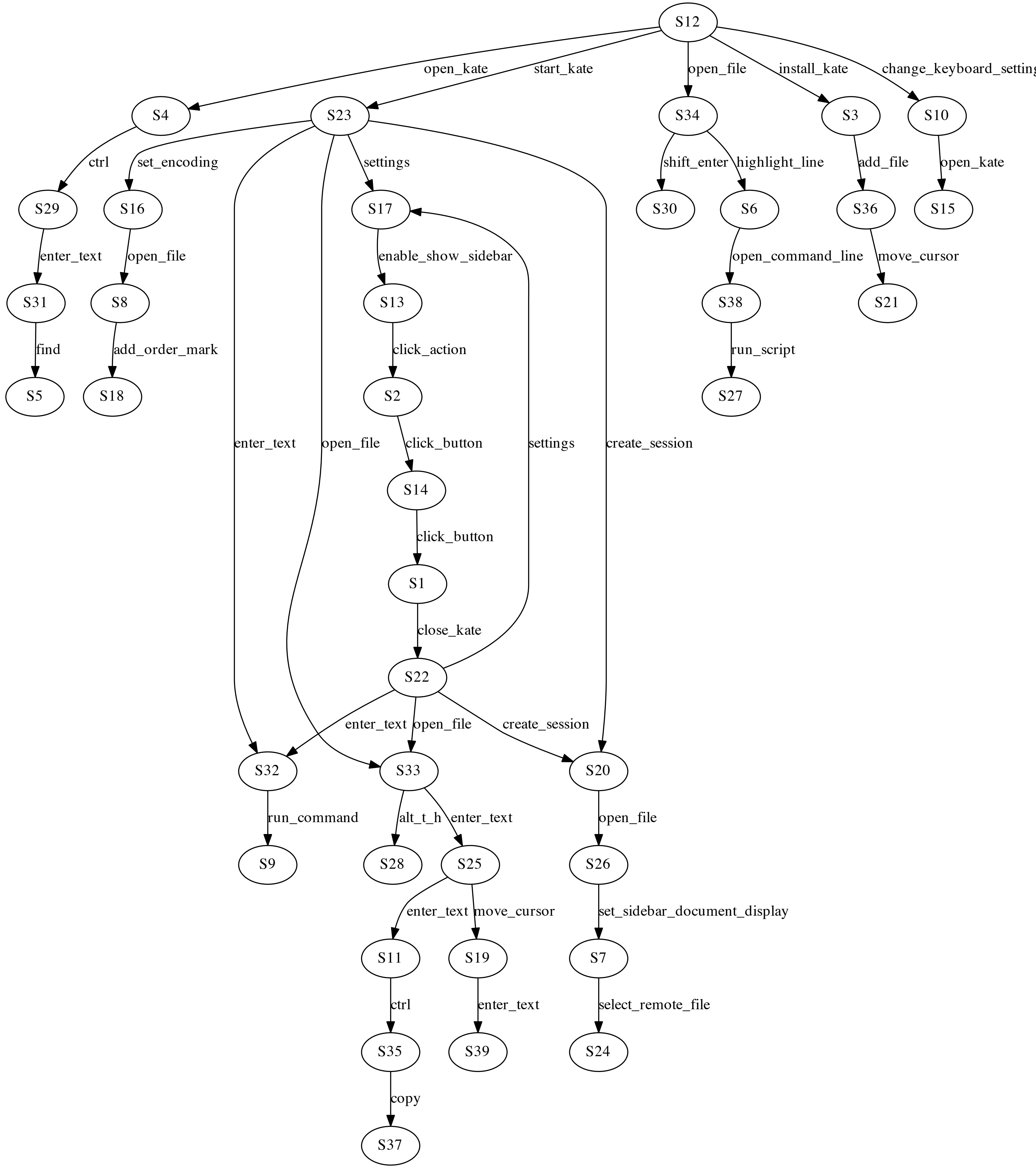
KLFA

KLFA



KLFA


S16
enable_command_line
start_kate
open_file
S22
S21
open_file
edit_text
enable_command_line
start_kate
enter_character
edit_file
open_file
change_syntax_highlight
enter_text
save
open_kate
open_kate
S17
S2
open_file
S1
select_text
start_kate
enter_character
edit_file
enter_text
save
open_kate
S13
S10
S12
S5
S4
S11
S15
select_text
edit_text
press_backspace
save
enter_text
switch_document
run_script
S7
S19
S8
S14
S9
S6
S3
enter_text
undo_change
S18
S20



S12
open_kate
start_kate
open_file
install_kate
change_keyboard_setting
S4
S23
S34
S3
S10
ctrl
set_encoding
settings
shift_enter
highlight_line
add_file
open_kate
S29
S16
S17
S30
S6
S36
S15
enter_text
open_file
enable_show_sidebar
open_command_line
move_cursor
S31
S8
S13
S38
S21
find
add_order_mark
click_action
run_script
S5
S18
S2
S27
enter_text
open_file
click_button
settings
create_session
S14
click_button
S1
close_kate
S22
enter_text
open_file
create_session
S32
S33
S20
run_command
alt_t_h
enter_text
open_file
S9
S28
S25
S26
enter_text
move_cursor
set_sidebar_document_display
S11
S19
S7
ctrl
enter_text
select_remote_file
S35
S39
S24
copy
S37

RQ1 Performance of the Algorithms - Quality Metrics

Performance / Algorithm	Quality Metrics - Mean	
	Running Time	No. of Solutions
GA	3239.66s	25
NSGA-II	2341.14s	17
KLFA	556.30s	1

Algorithm (x)	Algorithm (y)	Hypervolume	
		Cliff's Method *p-value*	Vargha-Delaney Effect Size $\hat{A}_{12}$
GA	NSGA-II	**1e-04**	**0.06**
GA	KLFA	**1e-04**	**1.00**
NSGA-II	KLFA	**1e-04**	**1.00**

RQ2 Fault-Revealing Ability of the Model

Divided 452 user events files into training and test sets based on submission time.
Training set: 226 user events files from July 2009 to Oct. 2012
Test set: 226 user events files from Nov. 2012 to Feb. 2015

checking the number of trace events, which are in the test set, accepted by the models generated by the training set

If a bug trace event is accepted by a model, the model can be used to generate test trace sequence to capture this bug

	Avg. # Traces (L = 4)	Avg. # Bugs Pareto Front	Total # Bugs	Avg. Test per bug revealed
GA	147	8	16	18
NSGA-II	116	6	22	19
KLFA	55,906	30	30	1863

Inferring Test Models from Kate's Bug Reports using Multi-objective Search

Model Inference Framework

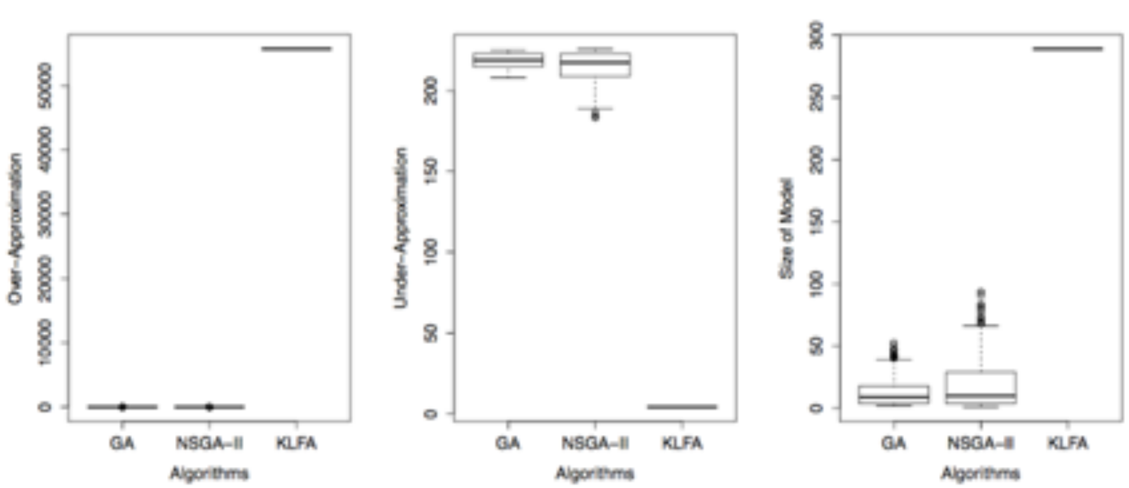
RQ0 Example of Trace Events Generated

721 bug reports → 452 user events files → 265 unique trace events

Category	Basic Operation	Text Editing	Programming
Examples	Start_Kate open_multiple_files score_screen drag_cursor resize_window close_file	copy_paste_text change_input_method fold_section find_replace captialize_text set_bookmark_color	select_haskell_mode show_javascript_console check_regular_expression fold_function check_indentation enter_vi_command
Category	**Configuration**	**Plugins**	**Shortcut**
Examples	change_keyboard_setting change_background_color change_print_margin change_print_page_range enable_command_line enable_static_word_wrap	enable_plugin_quickswitcher enable_plug_xml enable_plugin_spellcheck enable_plugin_tabbar enable_plugin_terminal enable_plugin_treeview	ctrl_1 ctrl_g ctrl_o ctrl_r alt_right alt_tab

RQ1 Performance of the Algorithms - Three Objectives

Performance / Algorithm	Objectives - Mean (Min, Max)		
	Over Approximation	Under Approximation	Size of Model
GA	2 (0, 66)	219 (208, 225)	13 (2, 53)
NSGA-II	0.1 (0.0, 7)	215 (183, 226)	19 (1, 94)
KLFA	55707	4	289



RQ2 Fault-Revealing Ability of the Model

Divided 452 user events files into training and test sets based on submission time.

Training set: 226 user events files from July 2009 to Oct. 2012

Test set: 226 user events files from Nov. 2012 to Feb. 2015

checking the number of trace events, which are in the test set, accepted by the models generated by the training set

If a bug trace event is accepted by a model, the model can be used to generate test trace sequence to capture this bug

	Avg. # Traces (L = 4)	Avg. # Bugs Pareto Front	Total # Bugs	Avg. Test per bug revealed
GA	147	8	16	18
NSGA-II	116	6	22	19
KLFA	55,906	30	30	1863