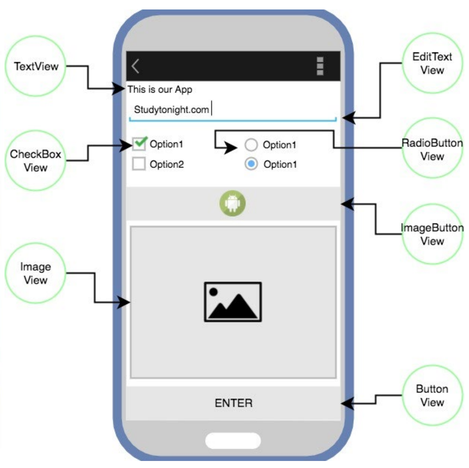
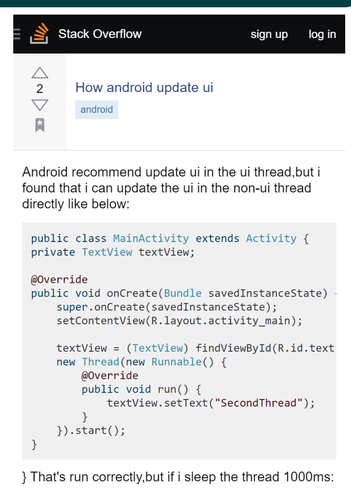
Bugine: a bug report recommendation system for Android apps

Ziqiang Li, Shin Hwei Tan*
Southern University of Science and Technology
*Corresponding author

Motivation

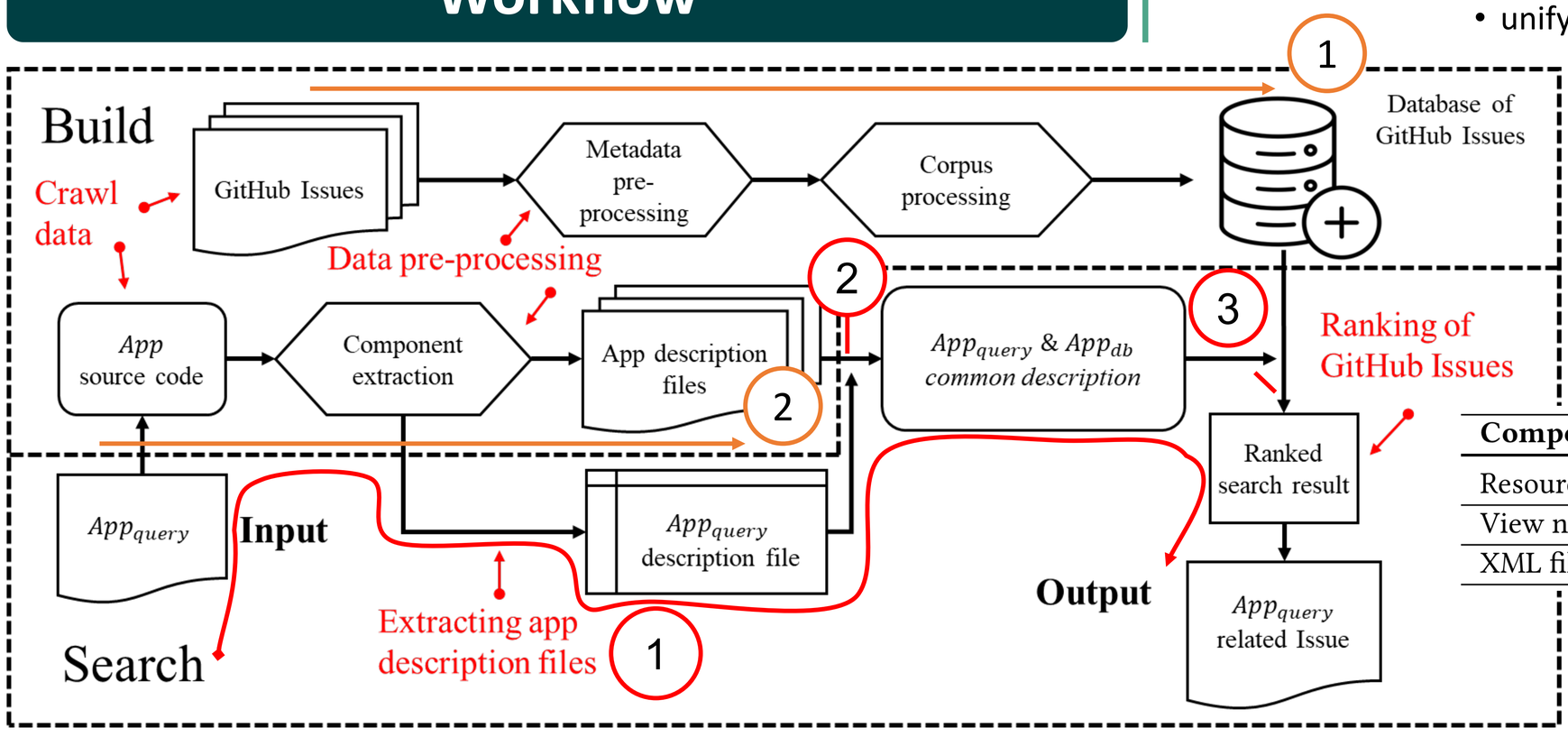




1. Query is not good enough to find a solution – unrelated, unhelpful result
2. Textual data may not contain enough info. – make use of UI information
3. Code snippets may be confusing. The naïve developer may not know where to insert them.

→ a solution in a similar complete project may be better

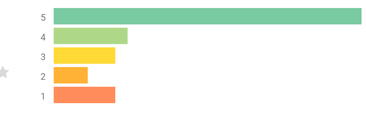
Workflow



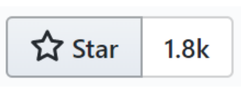
Build 1 – Metadata

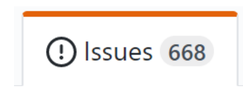
Select Apps by

1. [Quality]: user rating in the App store



2. [Actively maintained]: # dev discussions
3. [Popularity]: # star, # issues





4. [Diversity]category of App

Parse Metadata

- issue title, author
- # issue comments, issue labels, issue state
- issue body
- commit SHA (pull request / fix patch), etc.

Build 2 – Code - UI

Code - UI

- unify the naming conventions
  - `snake_case` or `CamelCase`
    → [“snake”, “case”] and [“camel”, “case”]
  - Extract UI components
    - By Android definition, XML files of UI layout are placed in `src/main/res`

Component	Example
Resource name	`android:id="@+id/`**`my_btn`**`"`
View name	`<`**`Button`** `android:id="@+id/my_btn" />`
XML file name	**`main_layout`**`.xml`

For each XML file, convert each view and each resource in the form:

XML file name ∧ View Name ∧ Resource name

Given an App_{query}

Search 1

Extracts its UI components to its app description file.
(like “Build 2 – Code - UI”)

Rank by correlation + usefulness

[Correlation] Similarity metrics:
Issue Title / Body
overlap coefficient
UI Components
n-gram

Search 2

Use similarities between app description file for App_{query} and app description files in database to search for similar apps

Search 3

Ranking the related issues by combine metadata and UI similarities.
(by correlation and usefulness)

[Usefulness] Metadata:
1. the length of the text body of ***issue***
2. the status of ***issue*** (closed / opened)
3. if ***issue*** contains any bug-fixing commit
4. the number of replies that ***issue*** received
5. the overlap coefficient between the search keywords and corpus
6. if ***issue*** contains all keywords in the corpus
7. if ***issue*** contains any important keyword (e.g., reproduce, defect).

Result = 34 new bugs / 5 Apps

App Name	Category	KLOC	#Downloads	Rating	Version No.	#GitHub Stars	#GitHub Issue (closed)	# Bugs Found (new,old)
Camera-Roll	Gallery	26.00	100,000+	4.2	1.0.6	420	227(133)	(11, 0)
PocketHub	GitHub client	31.35	10,000+	3.3	0.5.1	9429	644(526)	(12, 2)
Simple File Manager	Explorer	5.84	50,000+	4.5	6.3.4	378	189(130)	(6, 2)
Zapp	Broadcast	8.41	N.A.	N.A.	3.2.0	60	151(137)	(2, 7)
Simpletask	Reminder	24.80	10,000+	4.7	10.3.0	349	821(583)	(3, 2)