User & Usage Feedback Mining

Towards Data-Driven Design and Evolution of Software

W. Maalej, 2023

Continuously & quickly prioritize, decide what to build

Cope with changing and highly personalized needs

Understand how software is actually being used

Continuously & quickly prioritize, decide what to build

Cope with changing and highly personalized needs

Understand how software is actually being used

Understand frustrated users and win them back

I Hate Lotus Notes | The Global Lotus Notes Support Group

http://www.ihatelotusnotes.com/

I HATE LOTUS NOTES

The Global Lotus Notes Support Group

Hello there! If you are new here, you might want to **subscribe to the RSS feed** for updates on this topic.

Ads by Google | Lotus Sametime | Lotus Notes | Lotus Domino | IBM Lotus Symphony

This website is dedicated to my fellow sufferers who day in day out are forced to use Lotus Notes, causing them to struggle with email communications, squirm at the thought of planning another day and generally fighting for their will to live. Don't despair, don't be broken, don't hang yourself, stand up and be counted.

The Healing Process

Notes 5 July 2007 3,748 Comments

+1 Like Tweet 538

The healing process is started by sharing your thoughts. Release your pain by sharing your experiences of Lotus Notes with me and the others here using the form below. Constructive criticism is of course encouraged (maybe we can stumble over some good ideas that IBM actually want to adopt?!?), but i know some good old fashion bitchin goes a long way to making me feel better, so feel free. Just keep it clean…..ish. Post your error messages, stories of woe (or successes) both will provide either hope or solace.

Now….. You ready?….. Vent away.

3,748 Responses on "The Healing Process"

Sponsors

Advertise Here £10 a week! Click for more information

Advertise Here £10 a week! Click for more information

Share pain, get the gear.

dreckstool.de/hitlist.do

www.dreckstool.de

www.dreckstool.de

[HITLIST] [HISTORY] [NEWS] [LINKS] [SHOTS] [ABOUT] [NOMINATE] [ADMIN]

Hitlist

Willkommen bei der basisdemokratisch ermittelten Hitliste der schlechtesten Tools.

Sonstige Development System Enterprise Internet Office

Platz	Programmname	Punkte	Vote for it !
1	Lotus Notes	27364	Dreckstool !
2	Microsoft Project	11856	Dreckstool !
3	Update Marketing AG Marketing Manager	11362	Dreckstool !
4	Microsoft Outlook	10879	Dreckstool !
5	SAP R/3	8257	Dreckstool !
6	IBM DB2	8010	Dreckstool !
7	HP OpenView ServiceCenter	7828	Dreckstool !
8	Remedy Action Request System	7713	Dreckstool !
9	Lotus Sametime Connect	7604	Dreckstool !
10	Lotus SmartSuite	7603	Dreckstool !
11	Oracle Database	3071	Dreckstool !
12	Tobit Software David-Server / Tobit Info-Center	1557	Dreckstool !
13	IBM Rational Software Architect	1350	Dreckstool !
14	Citrix ICA Client	1349	Dreckstool !
15	Lotus Domino	1292	Dreckstool !

Apple App Store

~1.4 million reviews

Google Play

~4.7 million reviews

Twitter @SpotifyCares

~2.4 million tweets

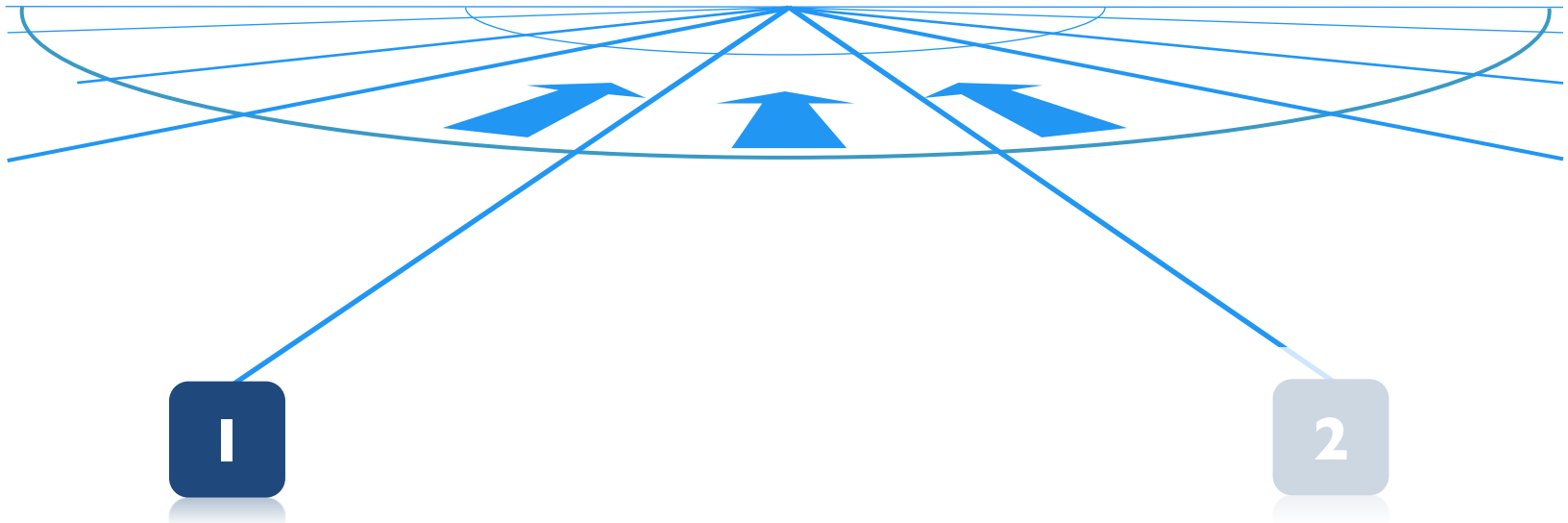
Community Forum

~0.3 million posts

How can we make user feedback useful for software teams?

[Pagano & Maalej RE'13] [Guzman & Maalej RE'14] [Maalej et al. REJ'16] [Martens & Maalej EMSE'19], [Martens & Maalej RE'19], [Martens & Maalej Software'19] [Stanick et al RE'20], [Haering et al. ICSE'21]…

Continuously and systematically gather & process user data to inform RE/SE decisions



What user say

User Feedback

+

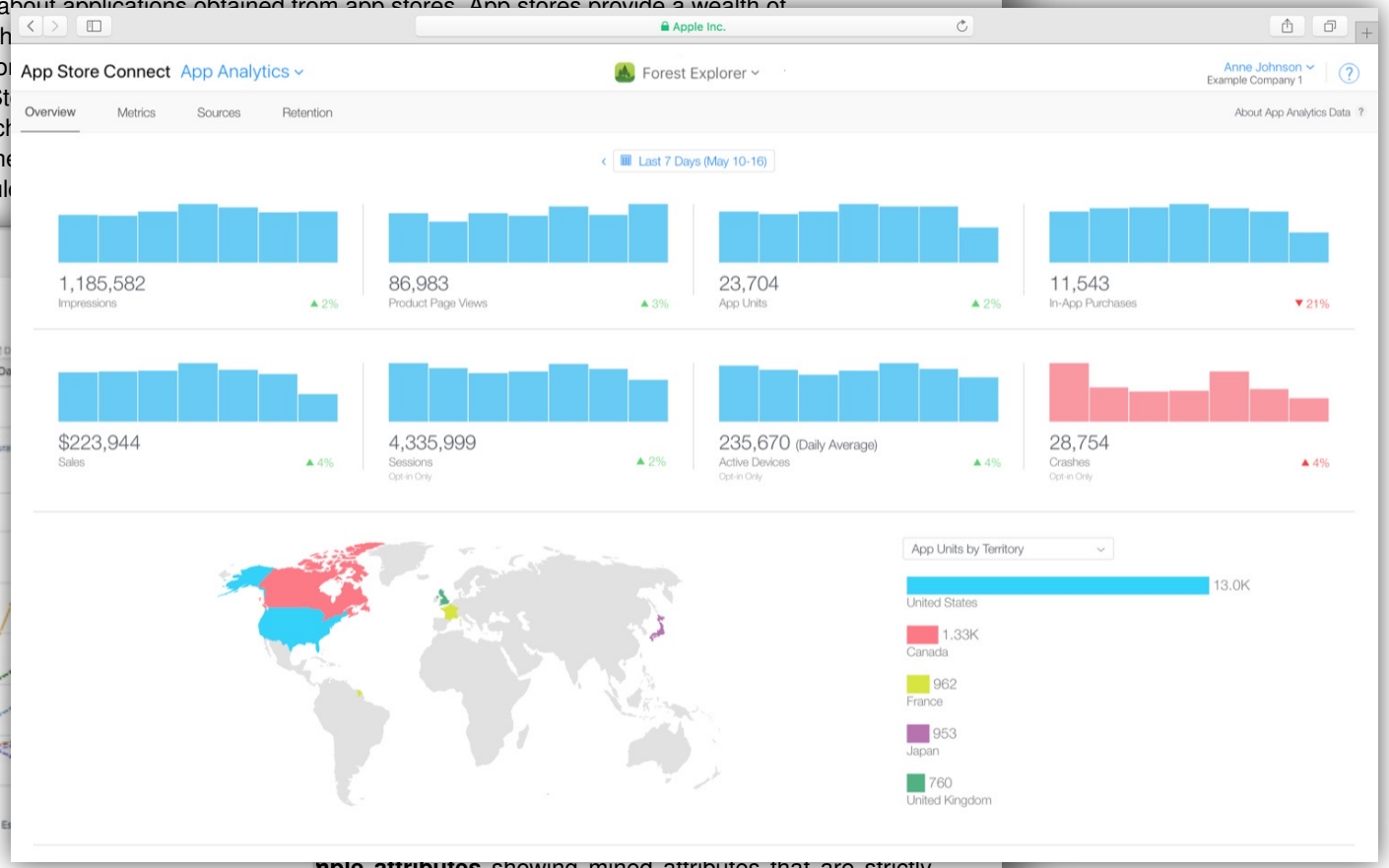
What user do

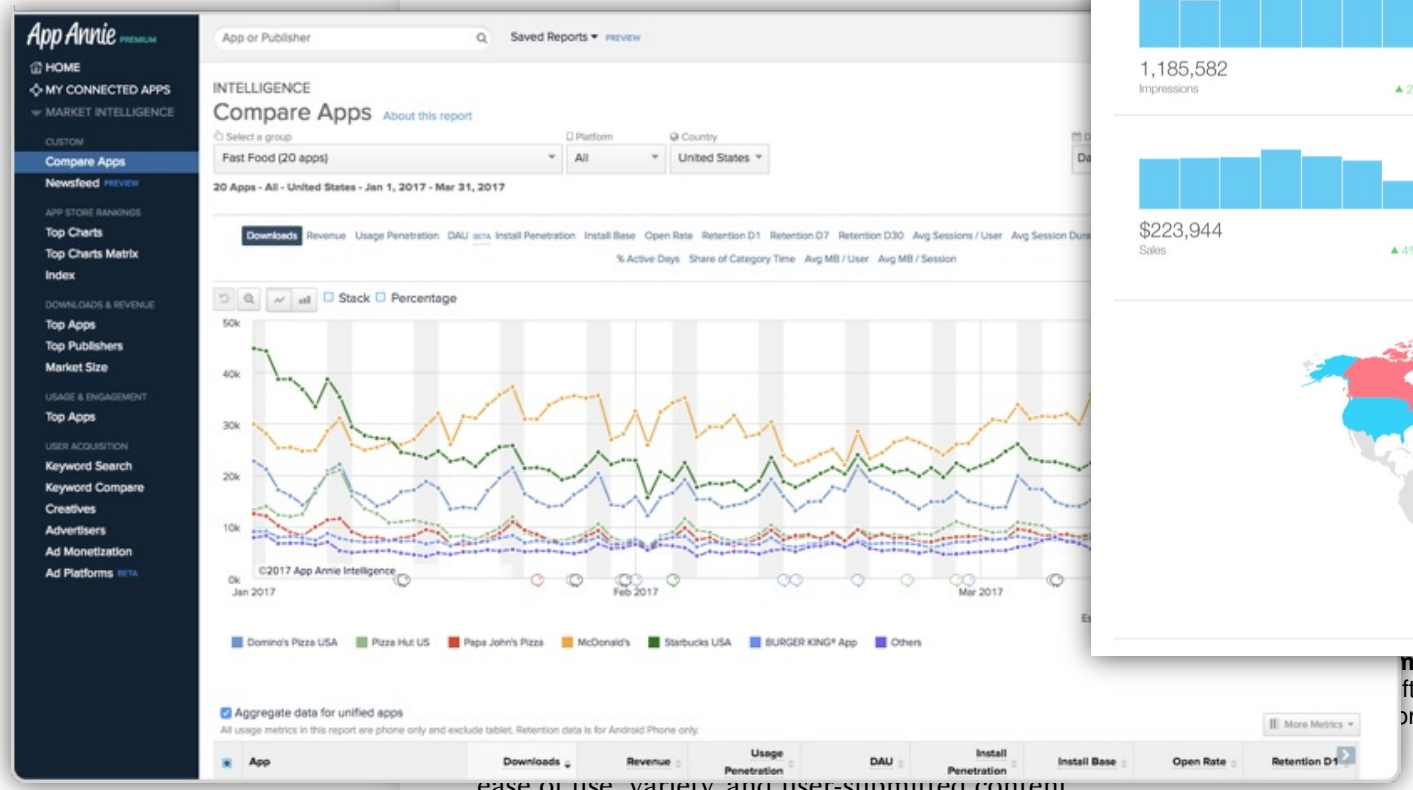
Usage Data

A Survey of App Store Analysis for Software Engineering

William Martin, Federica Sarro, Yue Jia, Yuanyuan Zhang and Mark Harman

Abstract—App Store Analysis studies information about applications obtained from app stores. App stores provide a wealth of information derived from users that would not exist h
App Store Analysis combines this non-technical infor
forms of software repositories. Findings from App St
develop software for app stores, and have led to tech
and testing. This survey describes and compares the
aspects, trends and directions future research shoul





nple attributes showing mined attributes that are strictly
ft) or non-technical (right), and attributes that may be in
ory (centre in box).

ease of use, variety, and user-submitted content.

It is the user-submitted content that fundamentally dis-

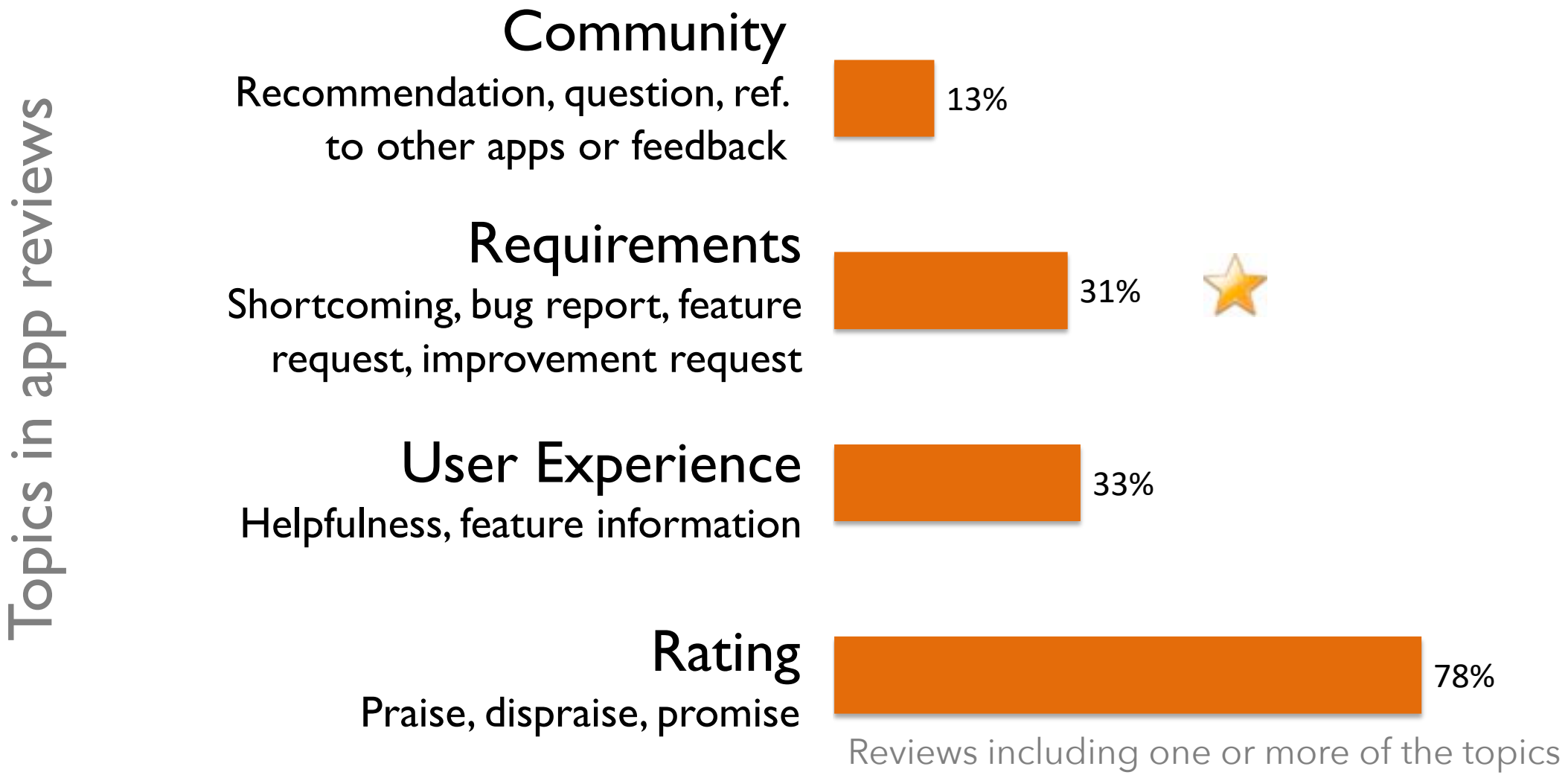
for App Store Analysis research. **ii)** We study the growth

1 Lots of continuous feedback with varying topics and quality

E.g. in App Store users submit on average 22 reviews per app per day

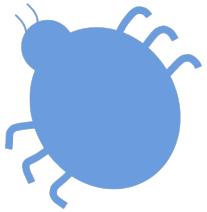
[Pagano & Maalej RE'13] [Finkelstein et al. 2014] [Martens & Johann SEMotion'16]…

Topics included in user feedback



[Pagano & Maalej, RE'13]

Automated feedback classification


Bug reports


Feature request


User experience


irrelevant

Document classification
BoW, With/out tf-idf, Embeddings / End-to-End Learning

Natural language processing
Stop words, Lemmatization, Bigrams…

Top accuracy: 80-95%

Review metadata
Star rating, length, tense, time

Sentiment analysis
Multiple scores, e.g. with SentiStrength

[Maalej & Nabil RE'15] [Maalej et al. REJ'16] [Dhinakaran et al. RE'18] [Stanik et al. AIRE'19]

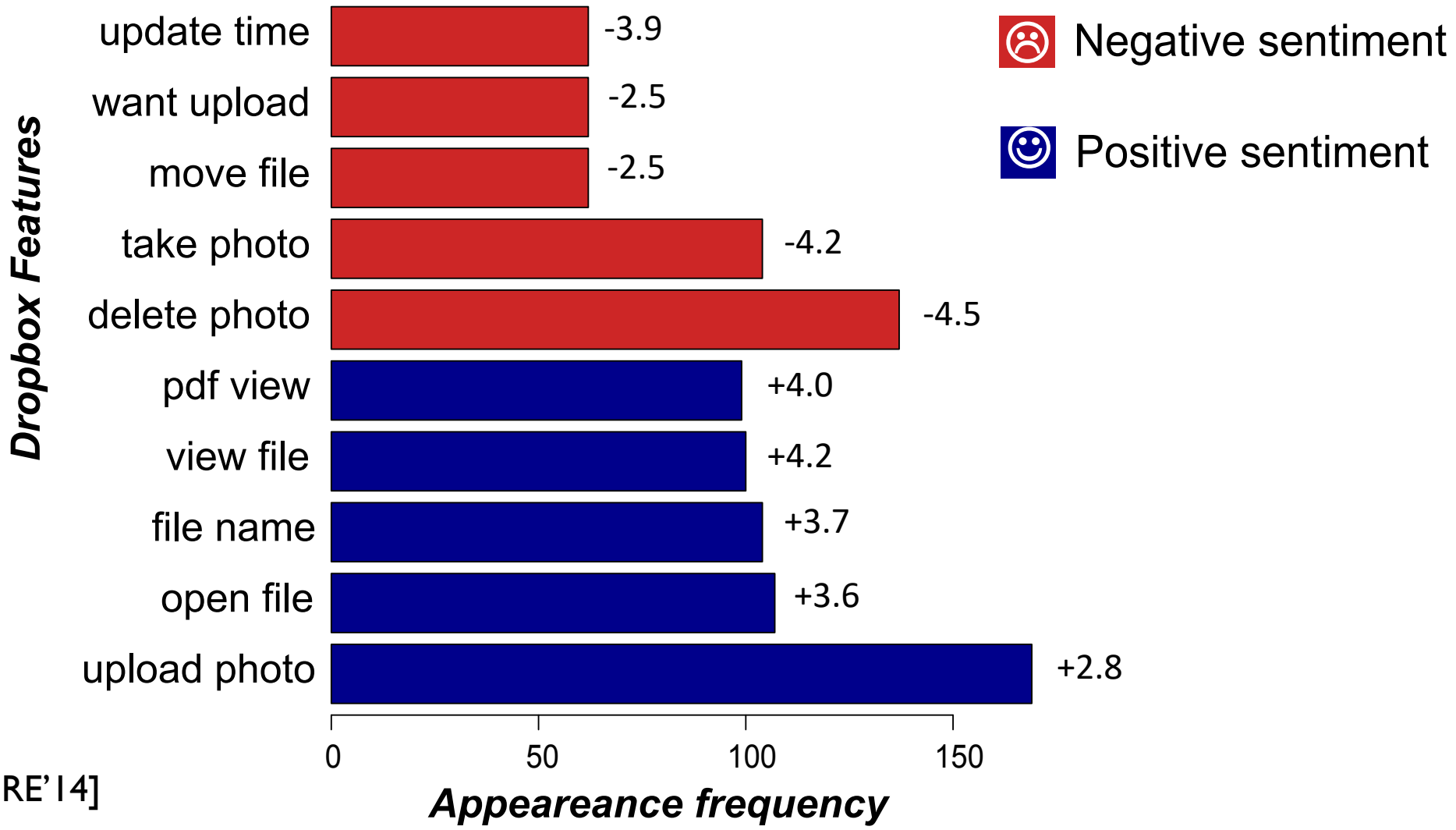
2 Feedback refers to multiple features with mixed sentiments

I love this app, it's the best of its kind! the **pdf viewer** is great. But I hate that the **files** take so long to **synch**.

Sharing files is great, but the **white background** is horrible. Please put the background from before!

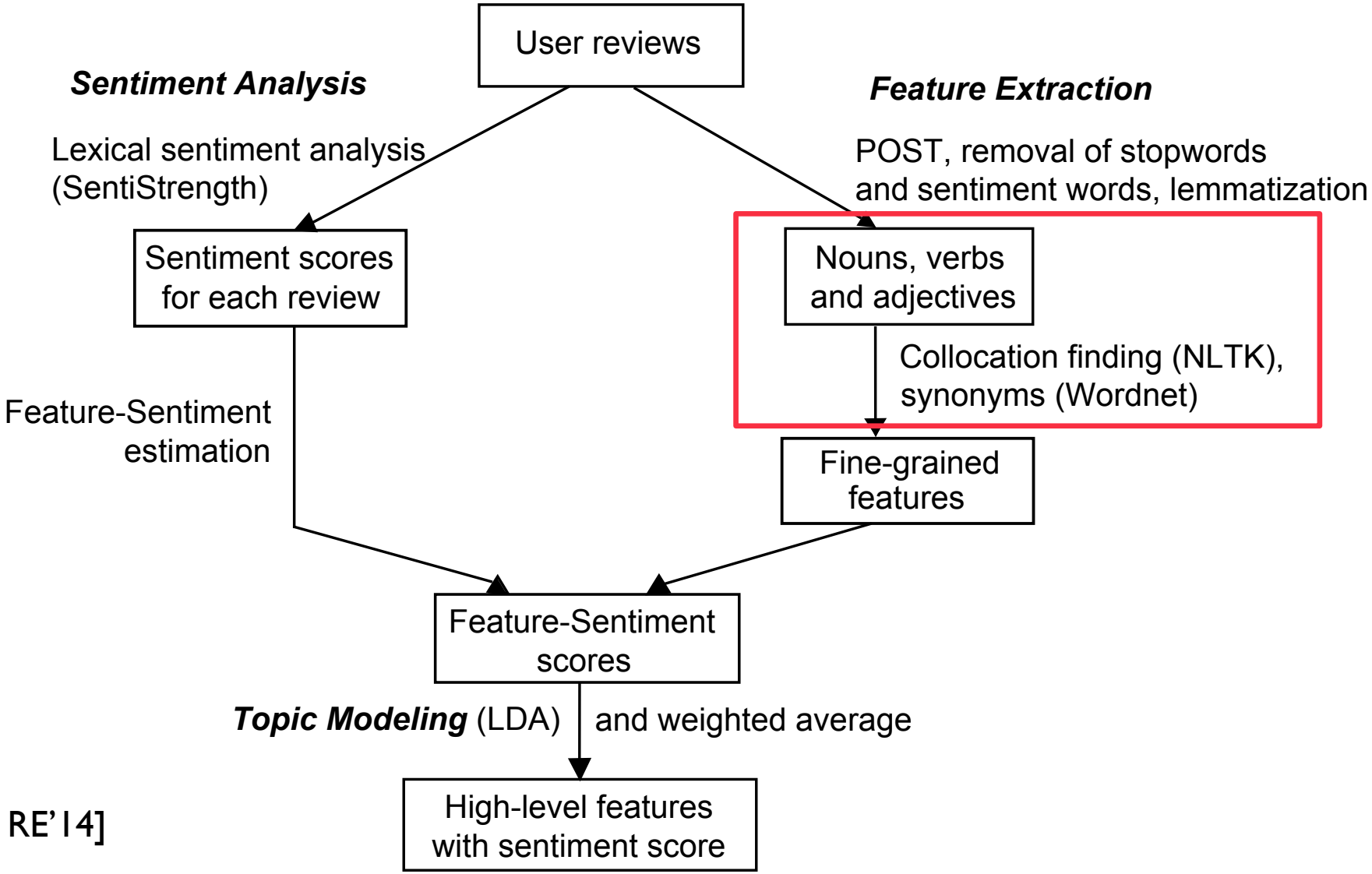
[Guzman and Maalej, RE'14]

Automated extraction of features and summarization of the sentiments



[Guzman and Maalej, RE'14]

Approach for feature extraction und sentiment analysis

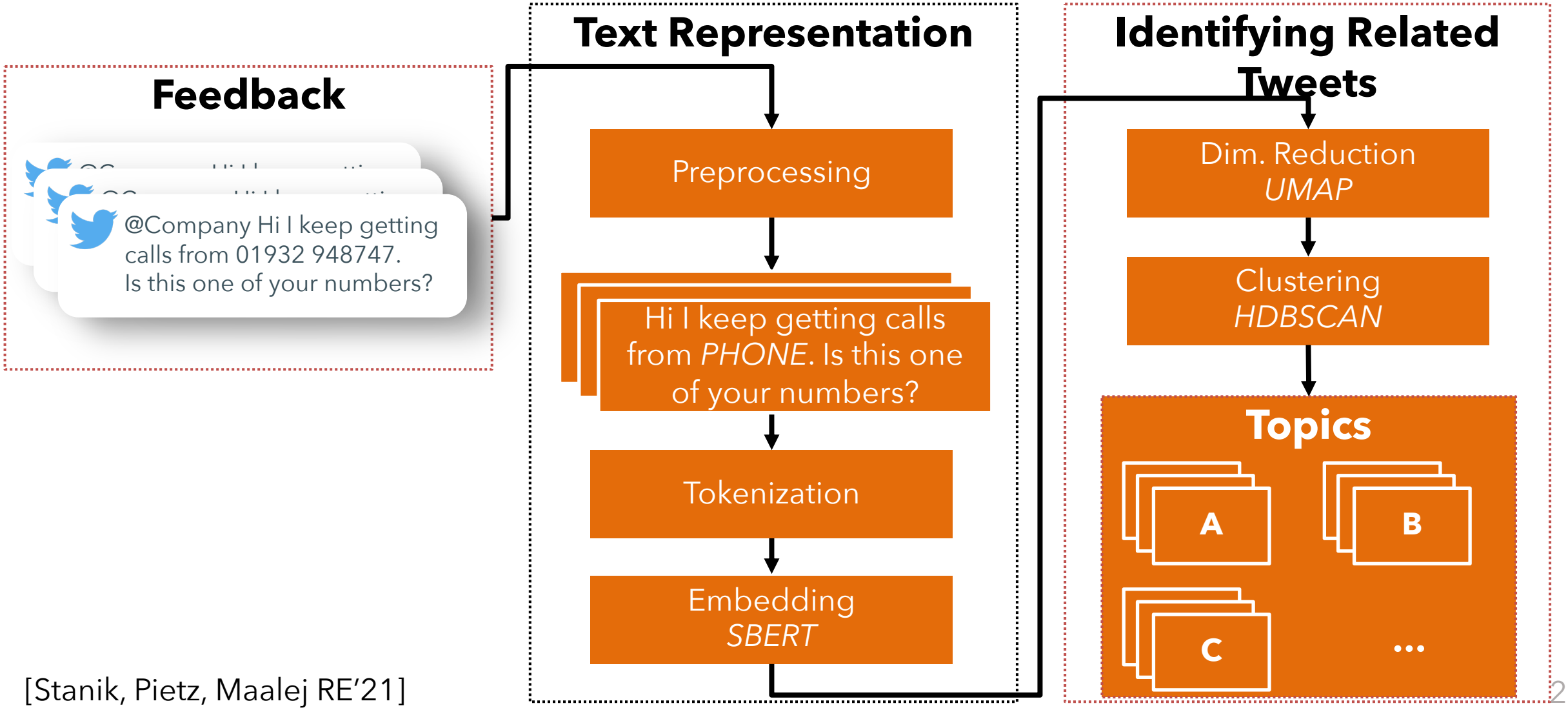


[Guzman & Maalej, RE'14]

Evaluation of the feature extraction

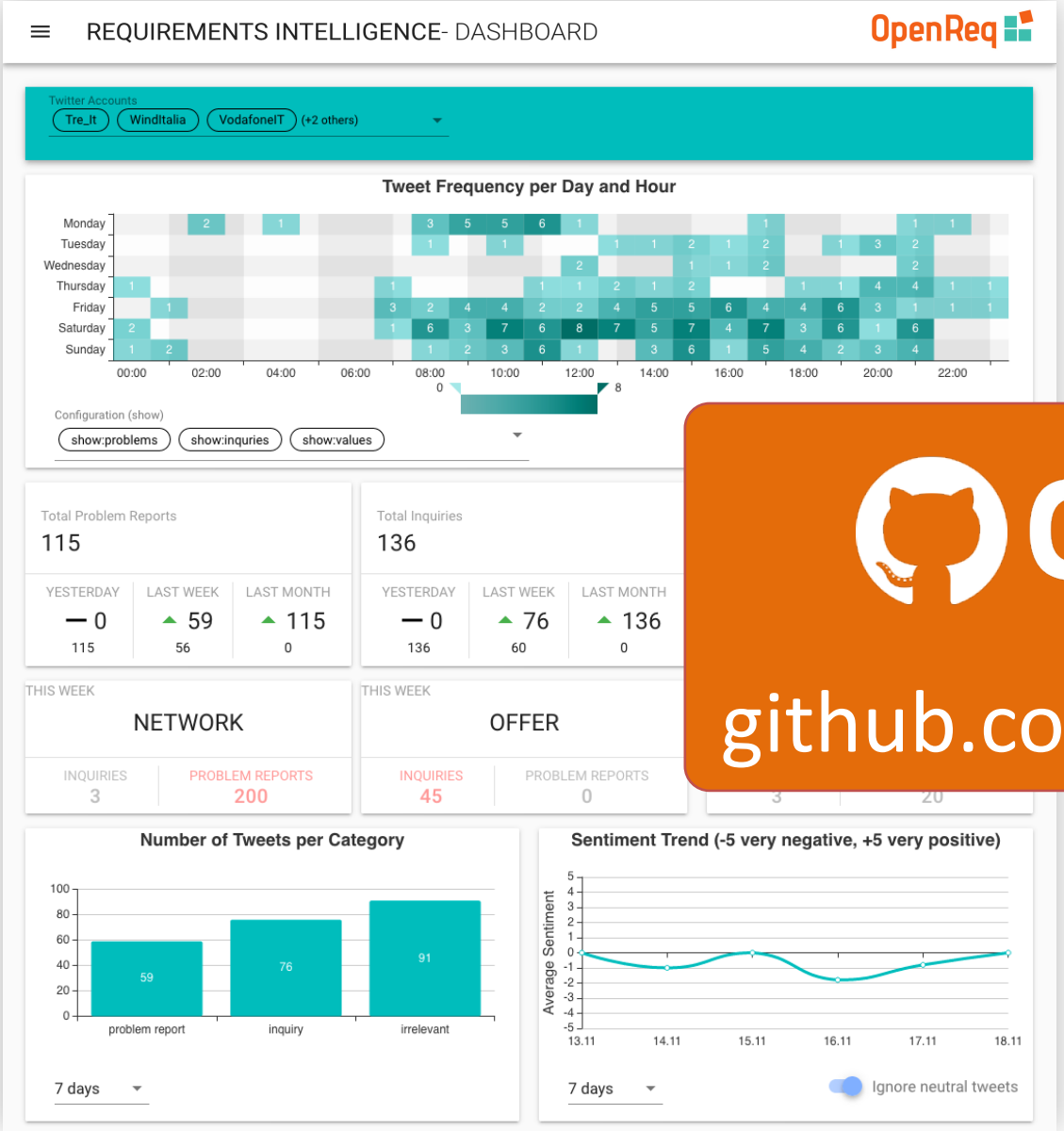
App	Precision	Recall	F-Measure
Angrybird	0.368	0.321	0.343
Dropbox	0.603	0.473	0.531
Evernote	0.451	0.389	0.418
Tripadvisor	0.403	0.370	0.386
Picsart	0.815	0.661	0.730
Pinterest	0.658	0.592	0.623
Whatsapp	**0.910**	**0.734**	**0.813**
Average	0.601	0.506	0.549

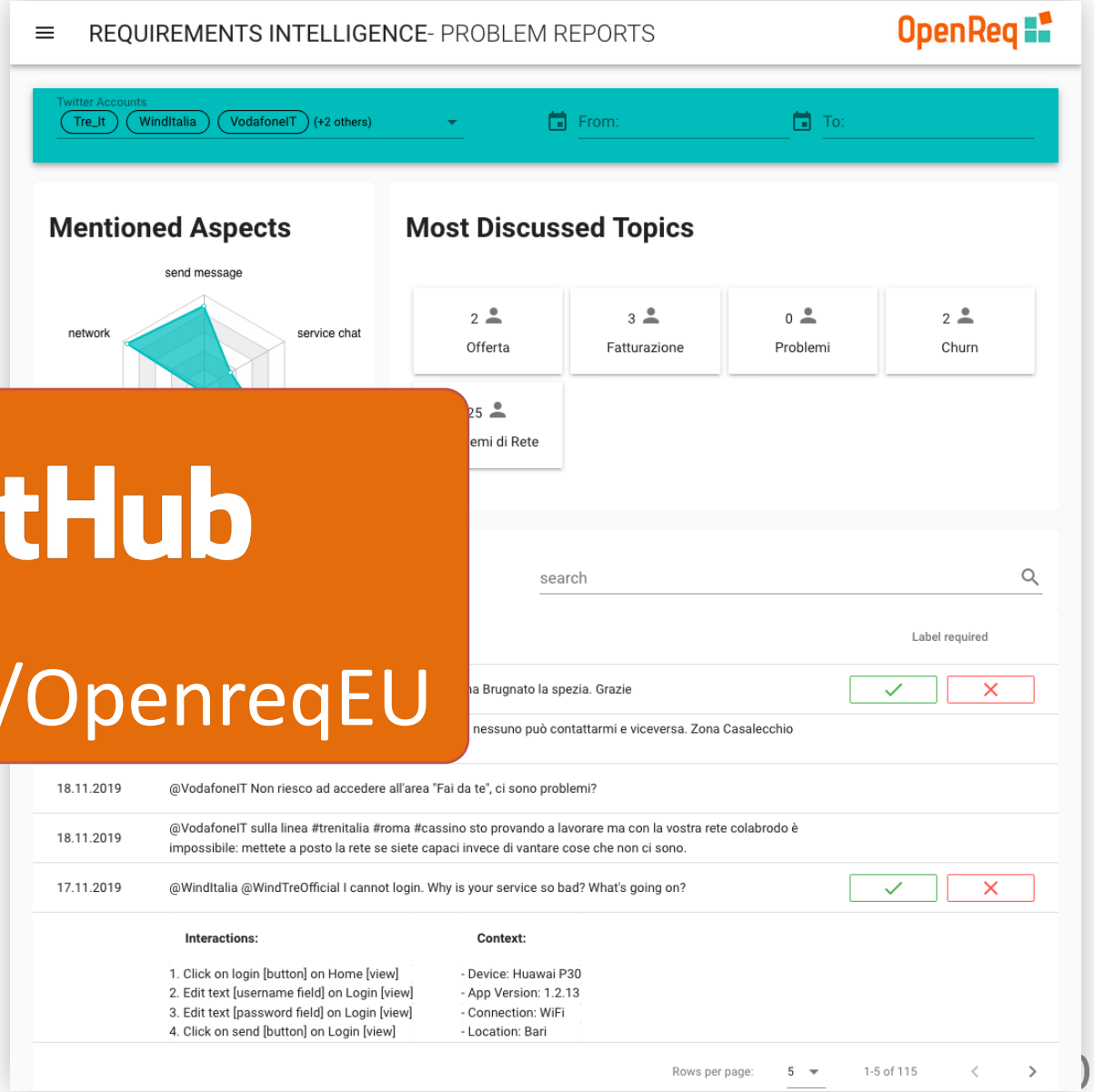
Deep Learning / Clustering Approach



[Stanik, Pietz, Maalej RE'21]

Requirements Intelligence Prototype





GitHub

github.com/OpenreqEU

3 Developers “live in a parallel universe”

GitLab Projects Groups More Search or jump to...

B Blog front-end

GitLab > Blog front-end > Jira issues

Create new issue in Jira

rches Search or filter results... Last updated

uld support longer titles
pened 2 weeks ago by Sean Smith in Jira Jun 4, 2020 Jira Story In progress updated just now

tag fails to load page
pened 1 week ago by Sean Smith in Jira Jun 6, 2020 Jira Story In progress updated just now

ges are missing a back to home link
pened 2 week ago Sean Smith in Jira Jun 6, 2020 Jira Task Backlog updated 2 days ago

re not working
pened 1 month ago Sean Smith in Jira Jun 6, 2020 Jira Bug Backlog updated 3 weeks ago

eeds to have an update page template
pened 1 month ago Sean Smith in Jira Jun 6, 2020 Jira Task Backlog updated 1 month ago

Jira Your work Projects Filters Dashboards People Apps Create Search

Blog Posts Management
Classic business project

Summary
Board
Issues
Reports
Business projects basics
Project performance
Find attachments
Employee Ratings
Checklist
Add item
Project settings

Projects / Blog Posts Management / BPM-1

Manage issues in Jira Cloud like a pro!

Attach Link issue Add Checklist

Description

Write a blog post that will feature the most interesting apps for managing issues in Jira Cloud.

Checklist 6 / 12

Add ToDo item or header text here...

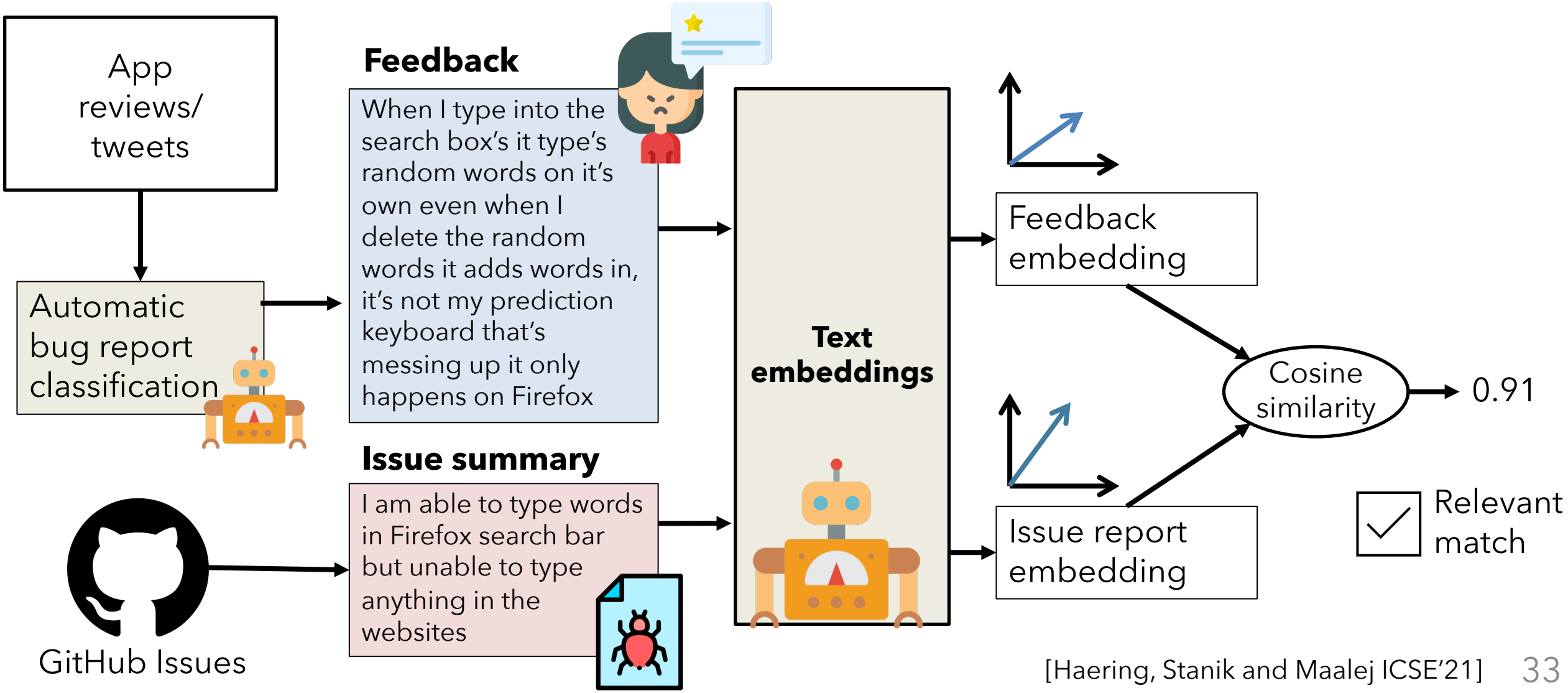
- [x] ~~Find the apps that manage issues in Jira Cloud~~
- [x] ~~Select the most popular apps with powerful capabilities~~
- [x] ~~Evaluate every app before writing a blog post~~
- [x] ~~Find the pros/cons of every app~~
- [x] ~~Prepare the description of each app~~
- [x] ~~Prepare eye-catching screenshots~~
- [ ] Add the links to products
- [ ] Review the blog post 2020-08-09
- [ ] Share the blog post on LinkedIn @NATALIE PARAMONOVA
- [ ] Share the blog post on Twitter @NATALIE PARAMONOVA
- [ ] Share the blog post on Facebook @NATALIE PARAMONOVA
- [ ] Send the newsletter to the blog subscribers @NATALIE PARAMONOVA

Draft

Assignee: Larry Stenfer
Reporter: Vadim Rutkevich
Due date: None
Priority: High
Content type: blog
Publication date: 2020/08/11
User Profile Details: View User Profiles
Smart Attachments: Open Smart Attachments

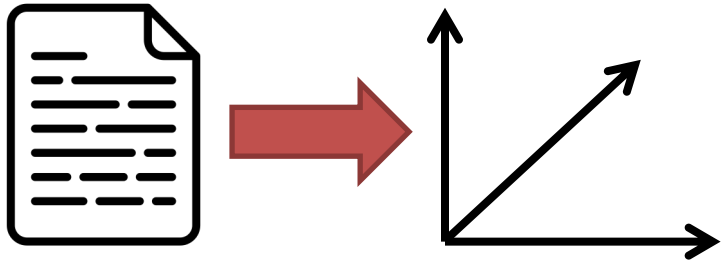
Show 3 more fields
Labels, Original Estimate and Time tracking

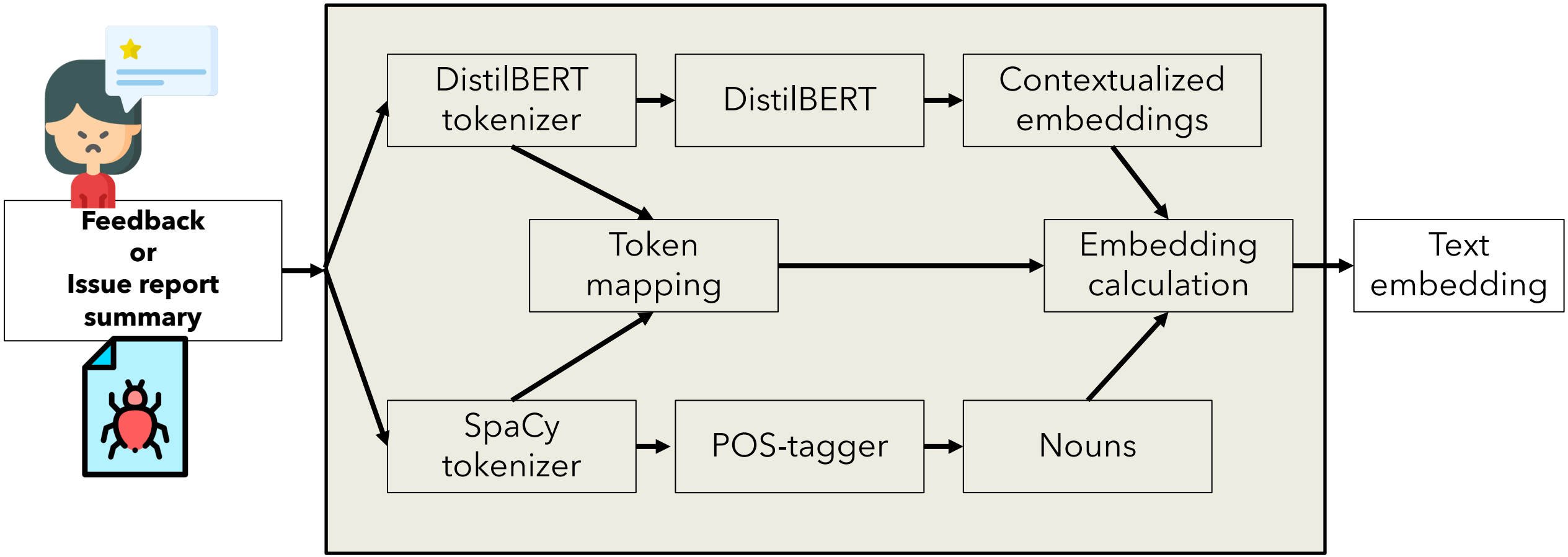
DeepMatcher: automatically match issue reports with related user feedback



[Haering, Stanik and Maalej ICSE'21]

Text embedding





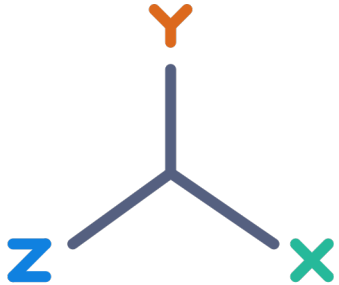
Evaluation Results

	Hit ratio (1 / 3 suggestions)	Mean average precision (1 / 3 suggestions)
Firefox (Bugzilla)	0.50 / 0.74	0.50 / 0.58
VLC Media Player (Trac)	0.32 / 0.51	0.32 / 0.40
Signal Messenger (GitHub)	0.38 / 0.68	0.38 / 0.50
Nextcloud (GitHub)	0.62 / 0.89	0.62 / 0.73
∅	0.46 / **0.71**	∅ 0.45 / **0.55**

Major findings

Manual analysis revealed 47 of 91 potentially **missing issue reports**

In 35 of 167 relevant matches, **users reported the bugs before** the developers



Strength of **contextual embedding** similarity

"synchronization" ≈ "upload"

"download" ≈ "save"

"consuming" ≈ "draining"

"update" ≈ "version"

Language gap leads to fewer matches

- VLC has the largest language gap (lowest similarity)
- Nextcloud has the smallest language gap (highest similarity)

[Haering, Stanik and Maalej ICSE'21]

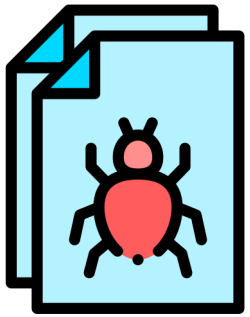
Use cases

Detect bugs earlier

- Continuously monitor user feedback
- Discover and create bug reports

Enhance bug reports

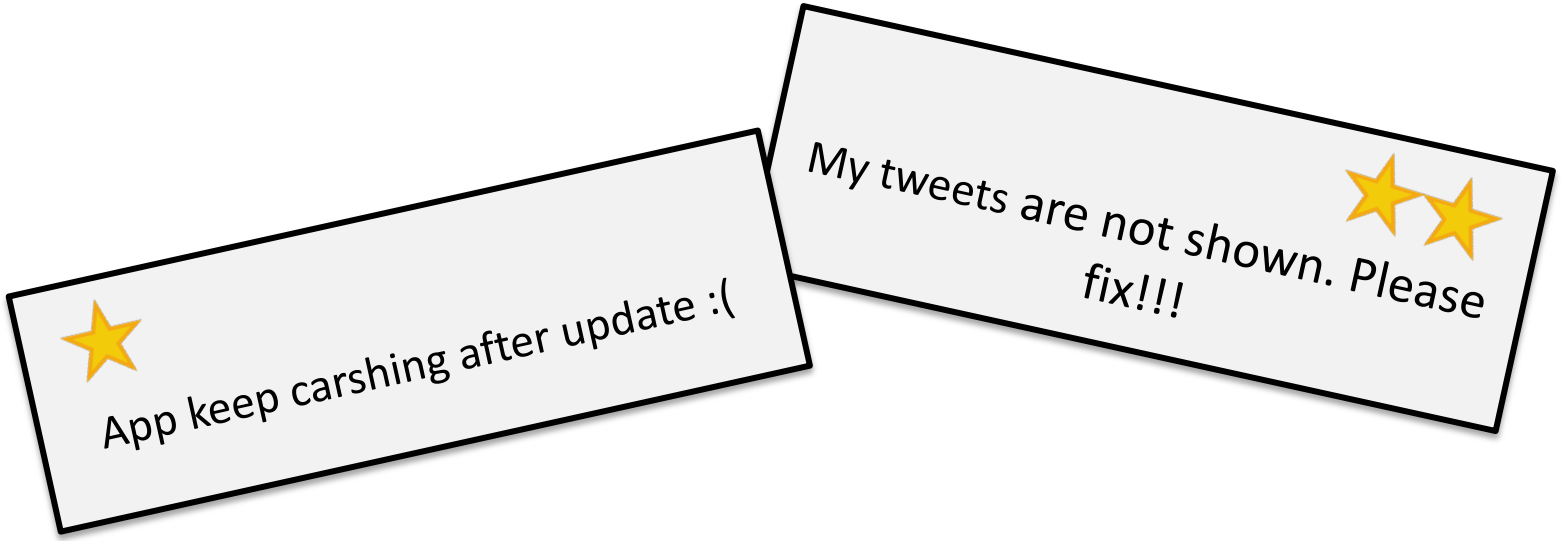
- Crowd-based severity level of bugs
- Extract context-information from app reviews



Find similar bugs

- Find duplicated bugs
- Find recurring bugs

4 Negative feedback often lack relevant details for developers



[Pagano & Maalej RE'13]

Ratings distribution across the topics

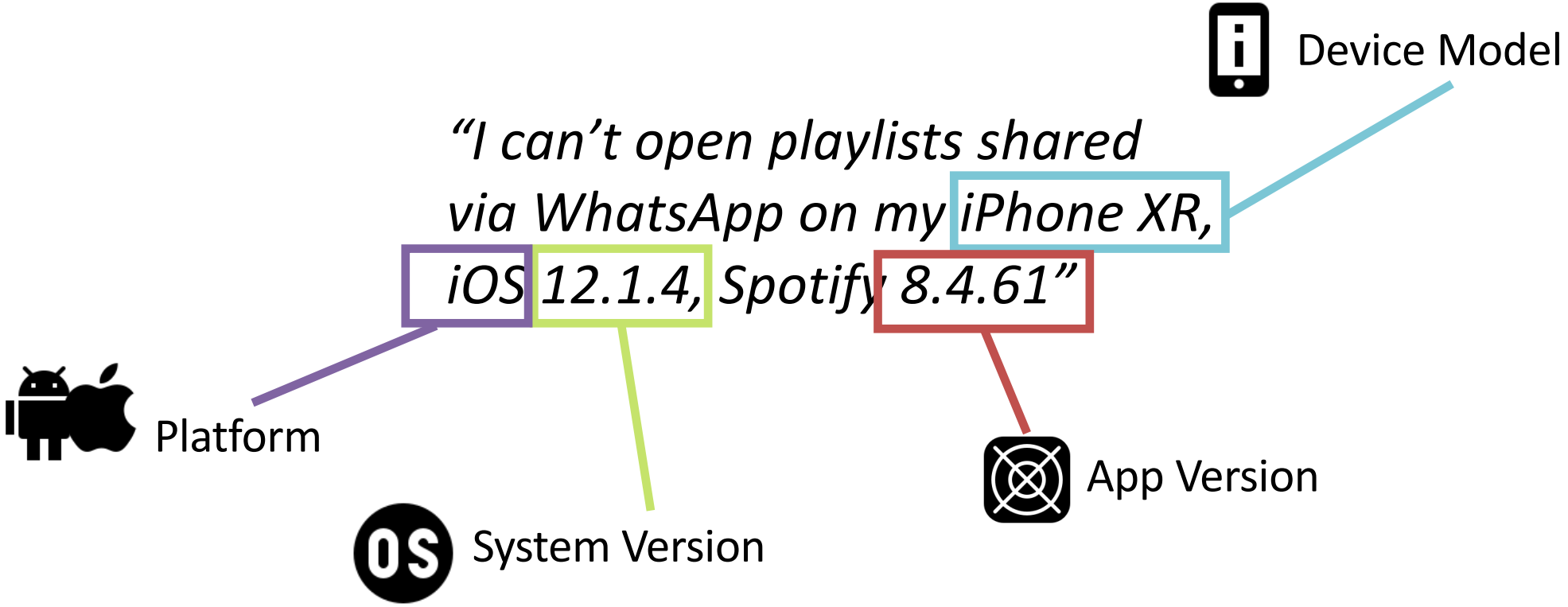
topic	Ø rating	rating distribution
recommendation	4.88	
helpfulness	4.85	
feature info.	4.81	
howto	4.80	
praise	4.78	
content request	4.25	
Ø sample rating	*4.08*	
impr. request	3.92	
other app	3.91	
feature request	3.89	
noise	3.67	
other feedback	3.67	
question	2.86	
promise	2.27	
shortcoming	2.10	
bug report	1.84	
dispraise	1.69	
dissuasion	1.39	

Legend

- *5 stars*
- *4 stars*
- *3 stars*
- *2 stars*
- *1 stars*

Best case

Tweet includes complete and correct basic context information



Reality case

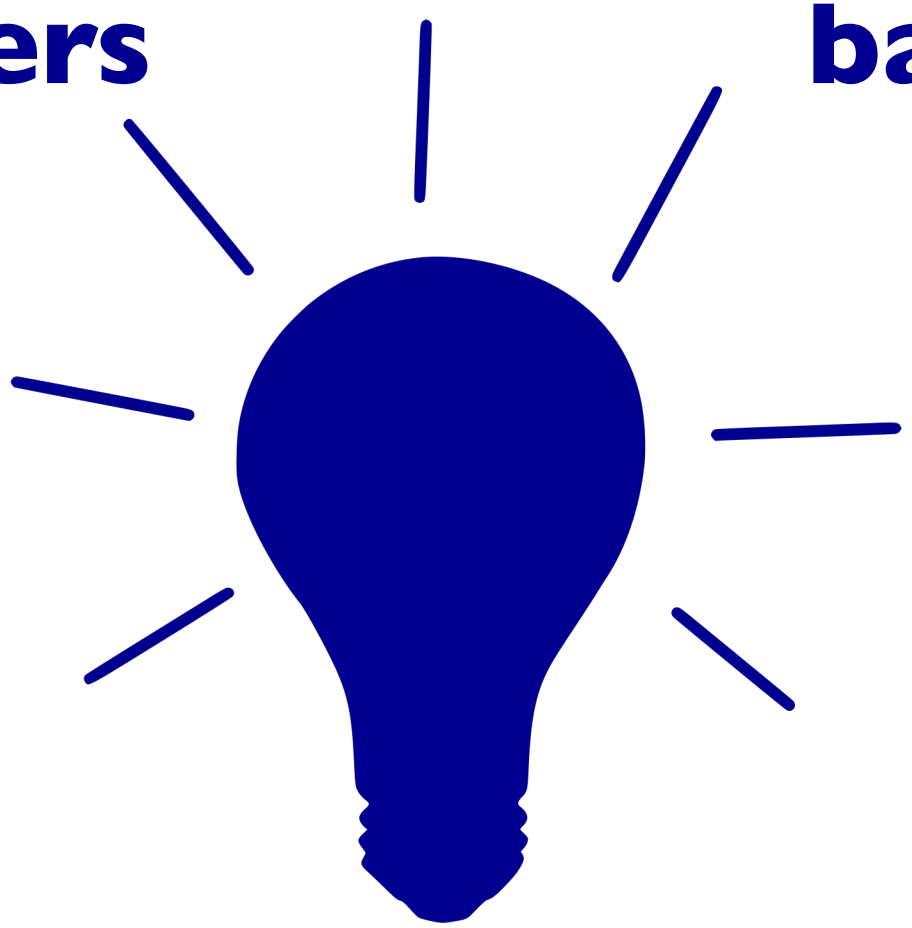
- **Missing information**
  - Limited time, willingness, and ability of users
  - Unstructured feedback processes
- **Unreproducible issues**
- **Manual clarification efforts**

[Martens and Maalej RE'19]

~40% of tweets by support teams to clarify missing information

I. Clarify in discussions with the users

II. Collect data implicitly (in background)



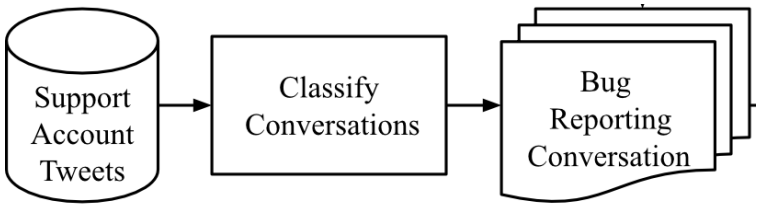
I. Bot-based approach to exchange context information

platform device app version and system version

[Martens and Maalej RE'19, Martens PhD thesis 2020]

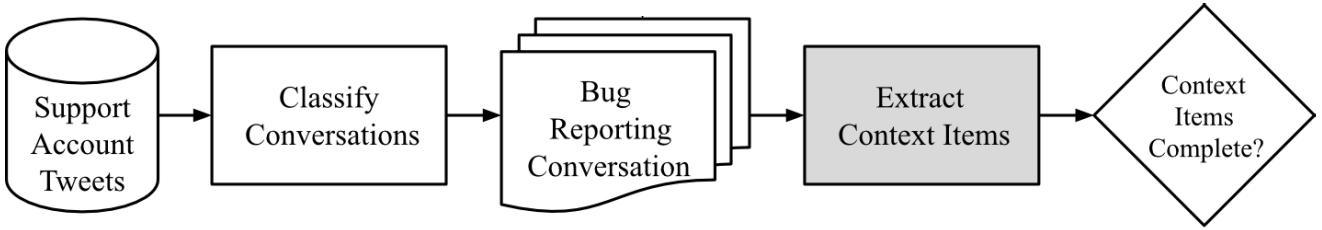
Phase 1: Tweet classification

- Only reports which require context information to be understandable and reproducible (e.g. bug reports or enhancement requests)



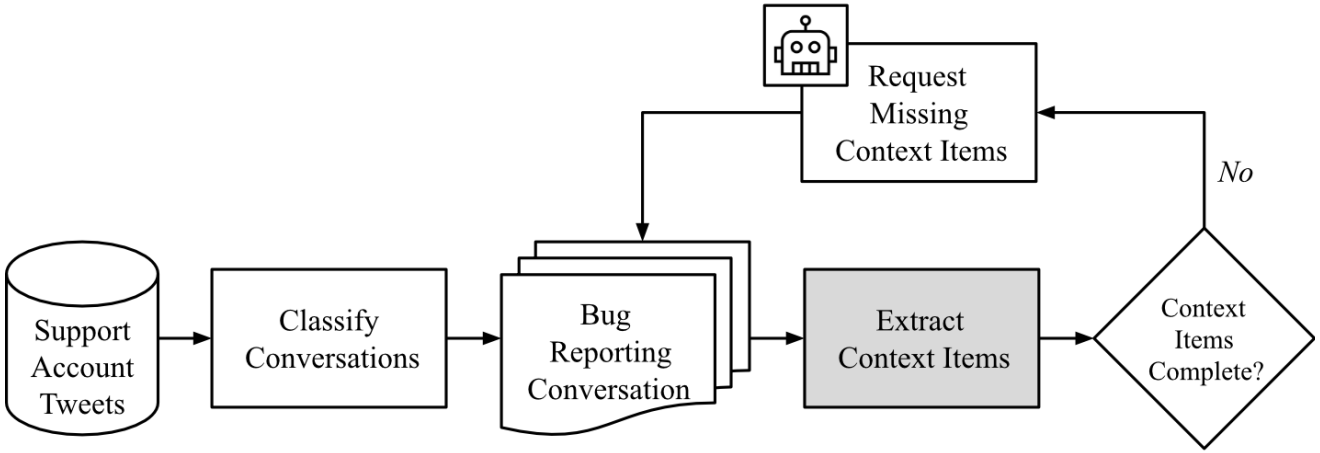
Phase 2: Context extraction

- **Example:** "The app widget has died and is now a. rectangular black hole. **Xperia xz3** running **Android**"
  - Missing: App Version, System Version



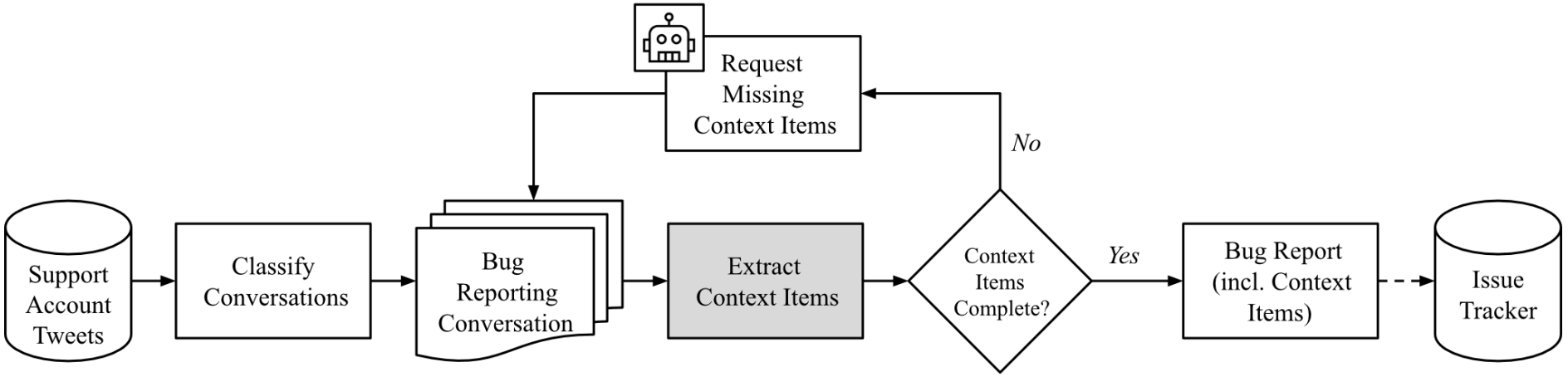
Phase 3: Context clarification

- Chatbot requests missing context items from reporting user
- Example Reply: "Hey, help's here! Can you let us know the **app version** you're running, as well as the **system version** installed?"



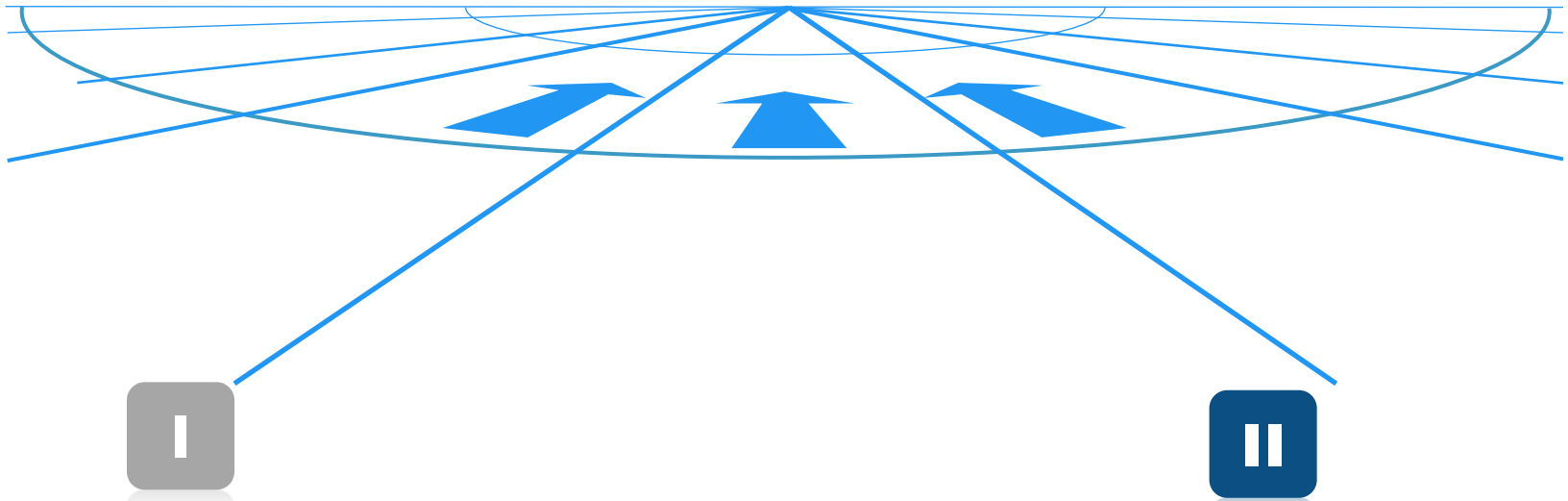
Phase 4: Issue creation

- Once basic context items are present, these are used to create structured bug reports within issue trackers



[Martens and Maalej RE'19, Martens PhD thesis 2020, Wolfinger et al. RE'22]

Continuously and systematically gather & process user data to inform RE/SE decisions



What user say

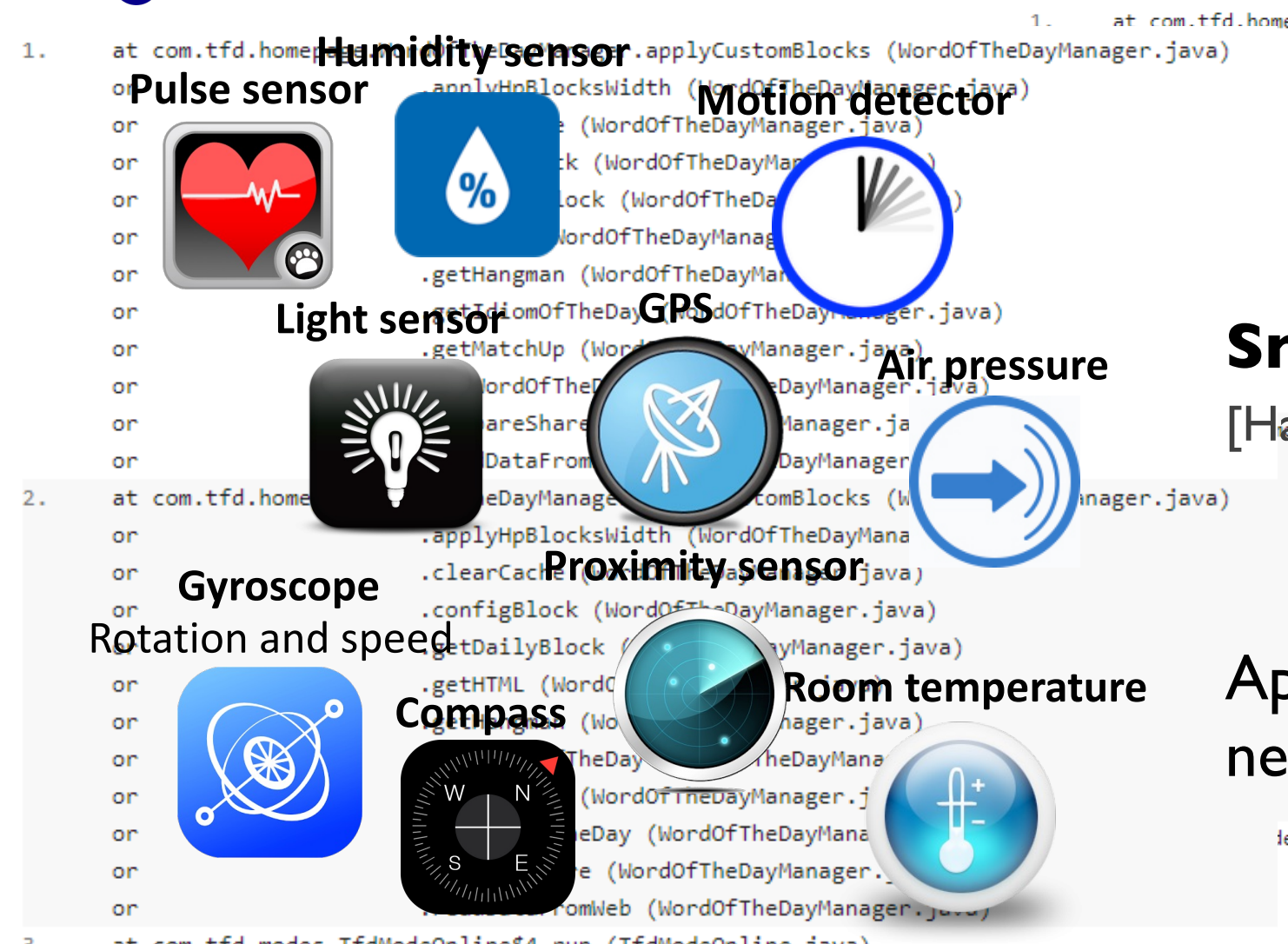
User Feedback

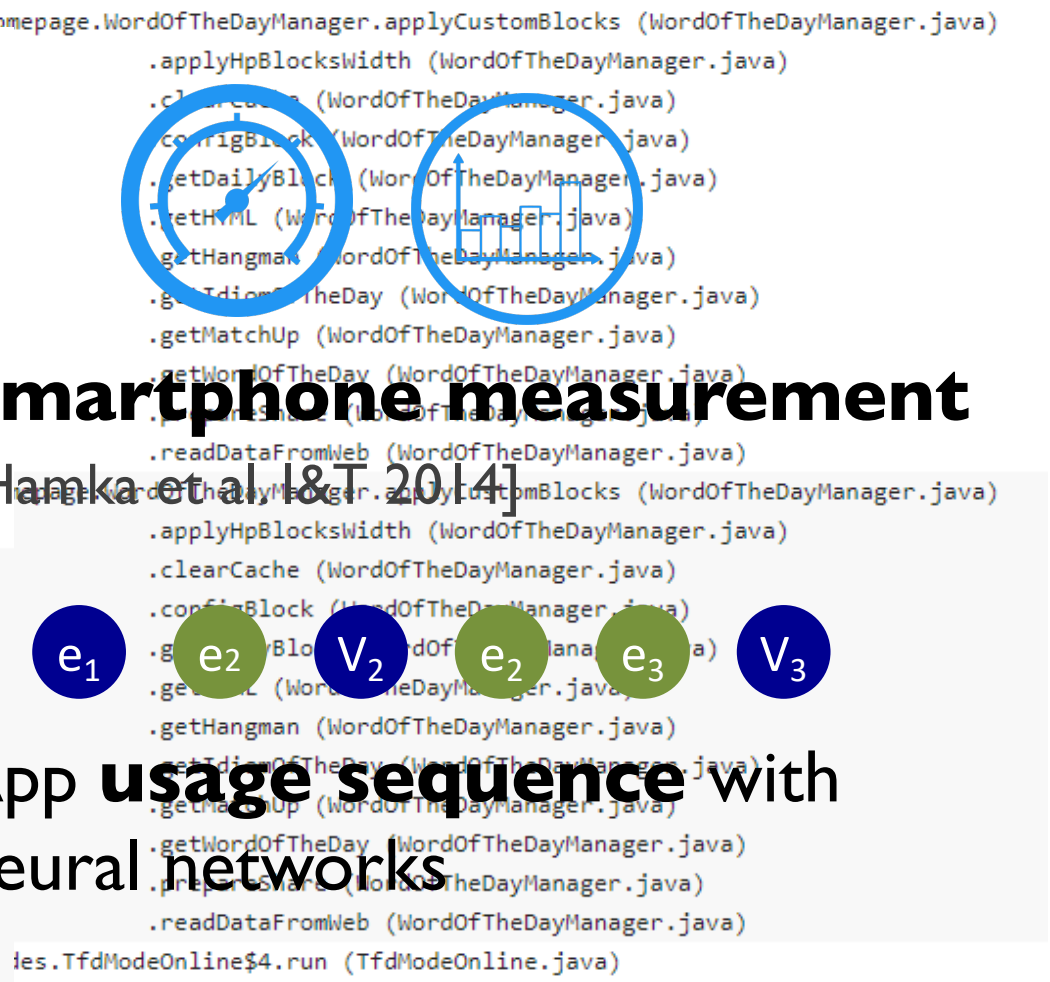
+

What user do

Usage Data / Implicit Feedabck

5 Usage data is (partly) sensitive heterogeneous and include much noise





Smartphone measurement

[Hamka et al. I&T 2014]

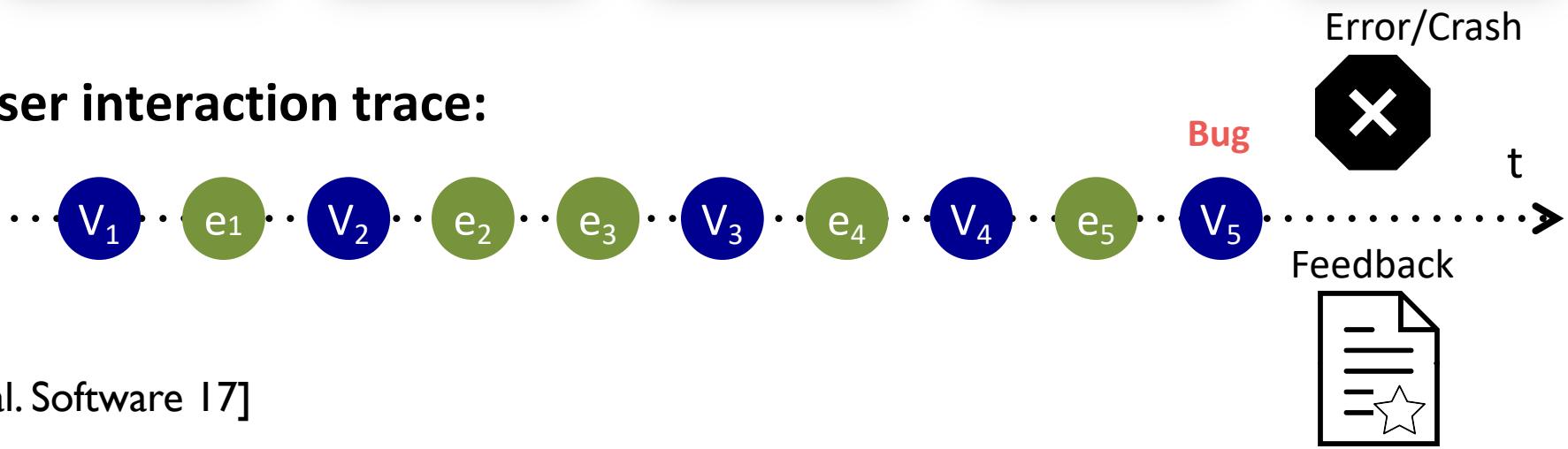
e_1 e_2 V_2 e_2 e_3 V_3

App **usage sequence** with neural networks

[Jesdebodi & Maalej UbiComp'2015]

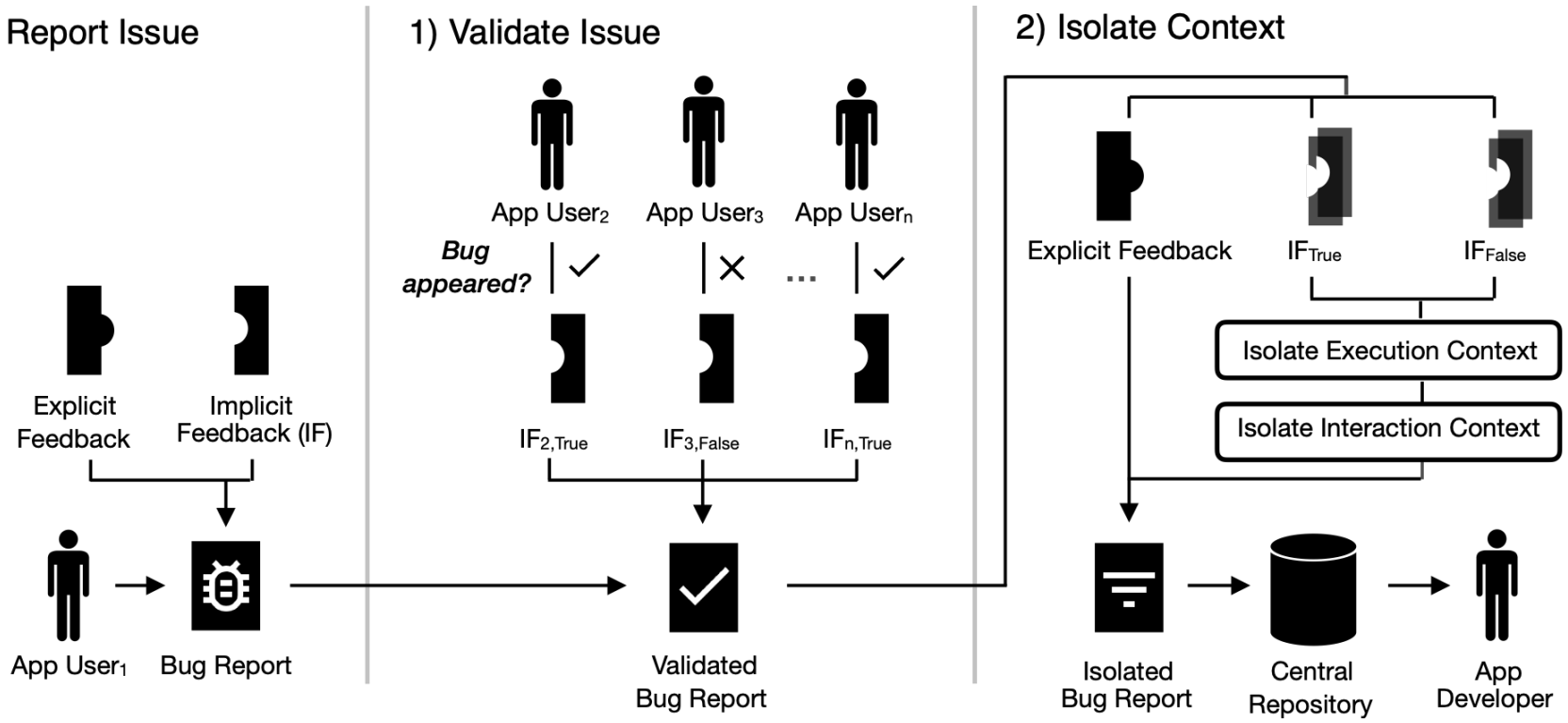
Focus on user interaction context for non-crashing bugs

User interaction trace:



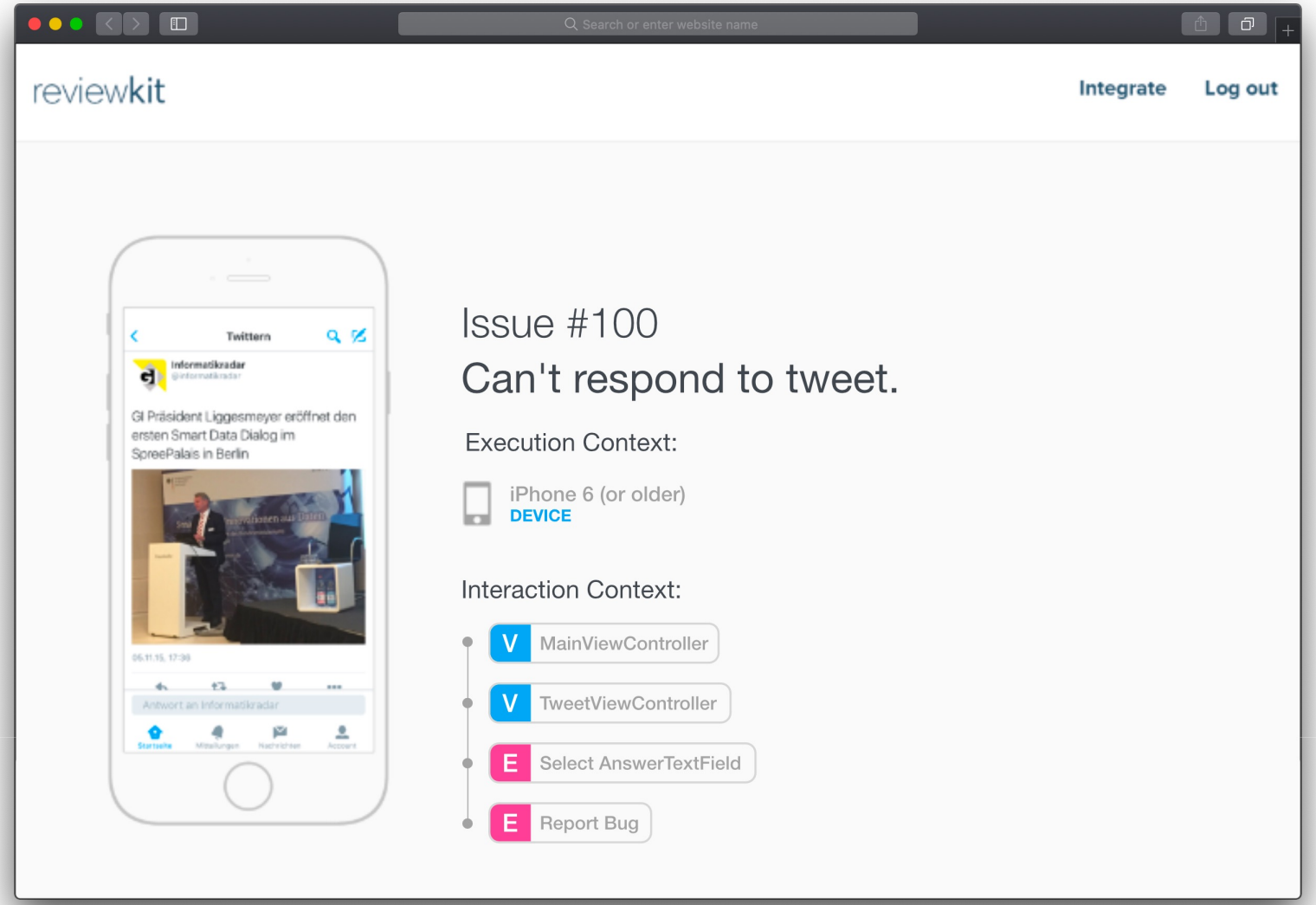
[Gomez et al. Software 17]

Crowdsourced isolation of context information



[Martens PhD Thesis 2020]

Isolated issue report in reviewkit



Bug reproduction experiments

- Does **isolated context information** help developers understand and reproduce **non-crashing bugs**?
    - 14 iOS developers + 2 software testers
    - 4 open-source iOS apps: DuckDuckGo, Firefox, Wikipedia, and Wordpress
    - Real bug data from app reviews, issue trackers, and fixed bugs in unreleased commits of the apps
- Developers needed **30% to 70% less time** and fewer interactions to understand and reproduce non-crashing bugs when context information is used

[Martens PhD Thesis 2020]

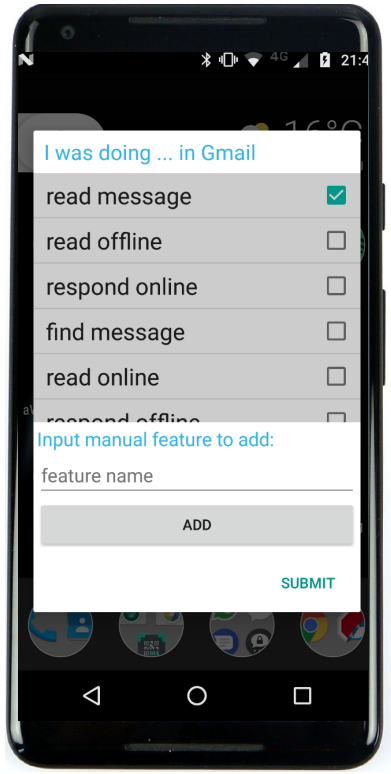
6 A granularity gap between observable and useful context data

Learning <u>app features</u> and their usages from context data

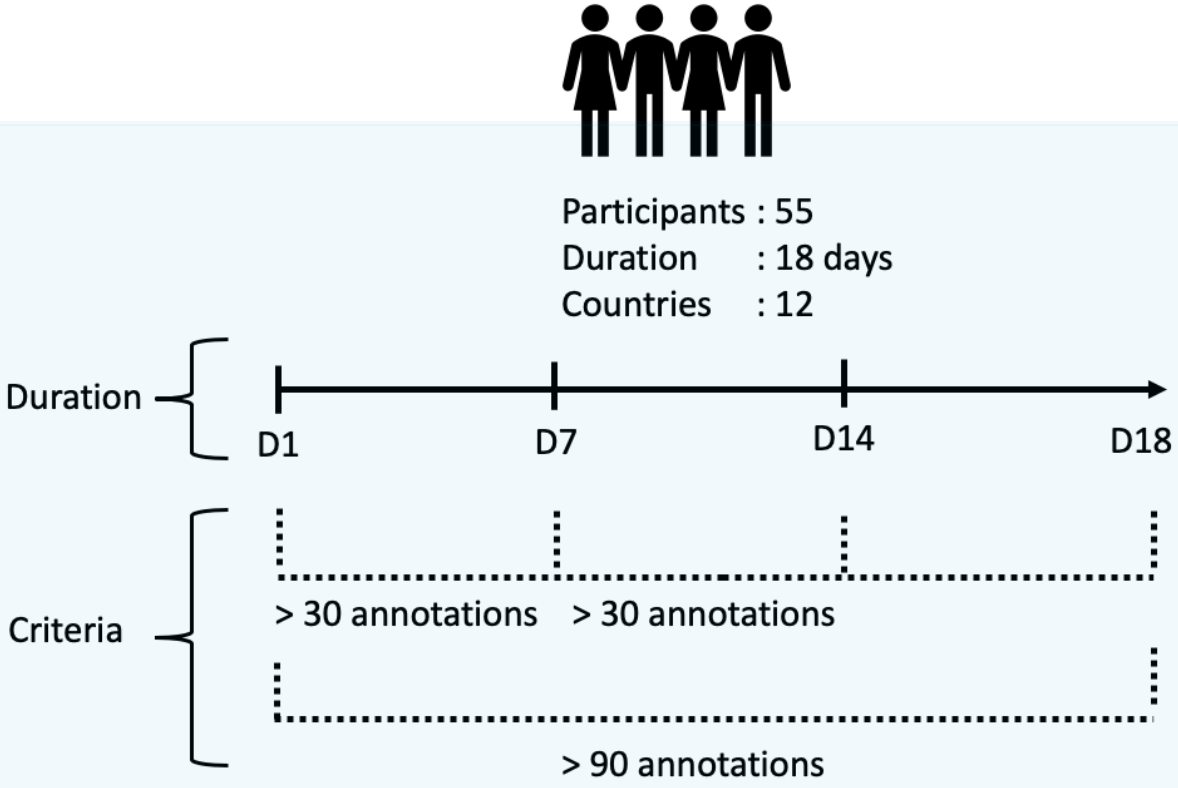
A large crowdsourcing study

Background interaction data collection

9 event types like: click, scroll, edit text, …



Session labeling



[Stanik et al. RE'20]

55 GB
Raw Data

170
Unique Apps

53
App Categories

5,815
Labeled Session

Data Modeling

Goals

- Transform interaction events to vectors
- Respect privacy
- Keep it simple

✕ We excluded:

- Context data
- Artifact data
- Location

✓ We included:

- 9 interaction events
- The app name

Data transformation

1. **Extract app sessions**

AS_1

AS_2

packagename	Type	Internet	event_type	longitude	latitude	Data sent
com.facebook.orca	MOBILE	TRUE	change content	-97.54606246	35.42715311	13401779
com.lge.launcher3	WIFI	TRUE	change content	-94.20432142	45.56927969	0
com.facebook.katana	WIFI	TRUE	change content	-80.65248462	35.50420719	581203
com.android.chrome	WIFI	TRUE	change content	-76.30209865	36.89856881	0
com.facebook.katana	WIFI	TRUE	scroll view	-80.65241874	35.50404454	581203
com.facebook.orca	MOBILE	TRUE	change content	-97.54606246	35.42715311	13401779
com.lge.launcher3	WIFI	TRUE	change content	-94.20432142	45.56927969	0
com.facebook.katana	WIFI	TRUE	change content	-80.65248462	35.50420719	581203
com.android.chrome	WIFI	TRUE	change content	-76.30209865	36.89856881	0
com.facebook.katana	WIFI	TRUE	scroll view	-80.65241874	35.50404454	581203

2. **Create ML feature vectors**

AS_1 E_1 E_2 ... E_{10}

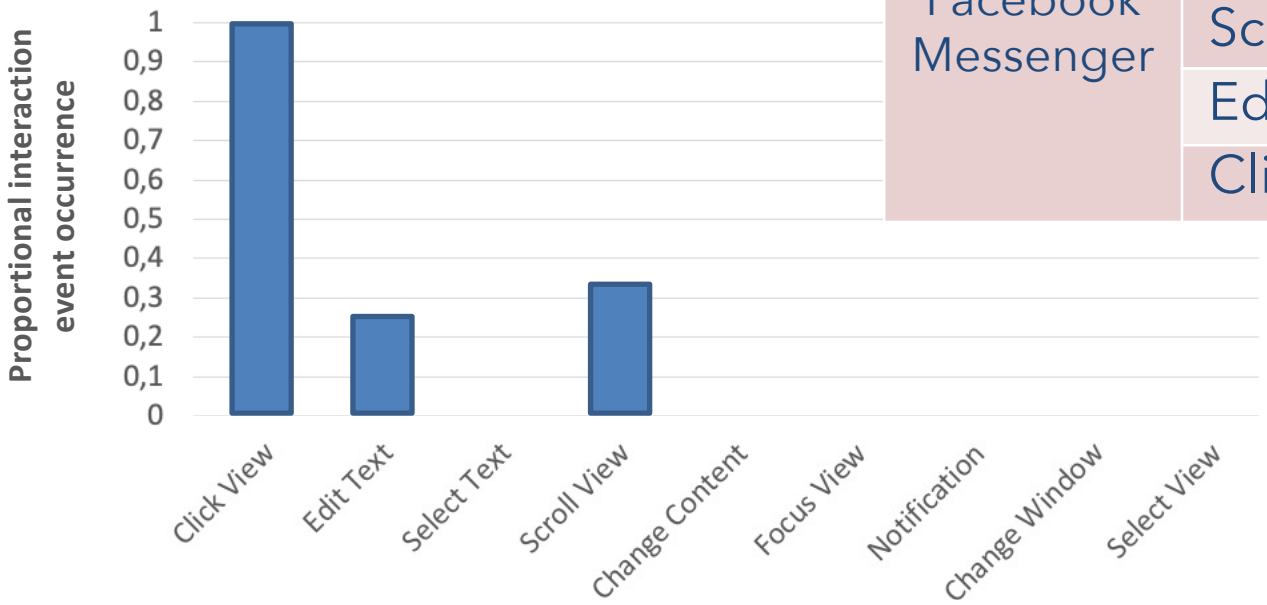
AS_2 E_1 E_2 ... E_{10}

⋮

AS_n E_1 E_2 ... E_{10}

Data modeling – Data transformation

App	Interaction Type	App Feature
Facebook Messenger	Click View	Send Message
	Click View	
	Scroll View	
	Edit Text	
	Click View	



ML Feature Vector:

0.6	0.2	0.0	0.2	0.0	0.0	0.0	0.0	0.0

[Stanik et al. RE'20]

Within-apps analysis

Goal

Learn all app feature usages in one app.

How we trained the model

App	App feature	F1
FB Messenger	Send message	**.88**
	Play game	**.80**
	Read message	**.73**

- We can **learn app features from interaction events**.
- ML feature significance hints toward understandable decisions.

Between-apps analysis

Goal

Learn all app feature usages across apps.

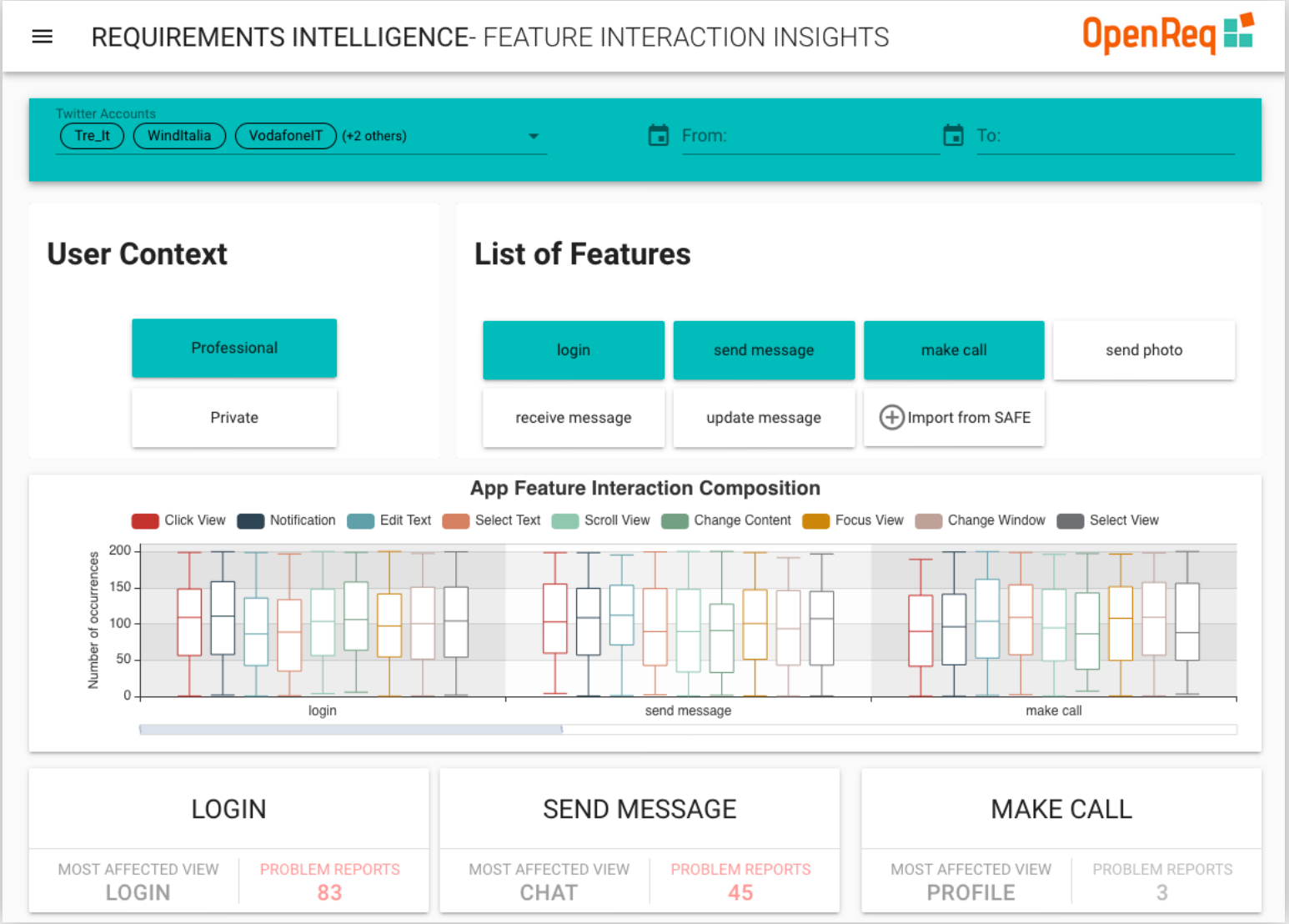
How we trained the model

App feature	F1	Part of *n* apps
Listen to music	**.86**	**4**
Delete	**.83**	**4**
Play game	**.75**	**10**

Promising results for introducing a **system-wide ML model**.

User **interactions across apps are alike for the similar features**.

Understanding app and feature usage

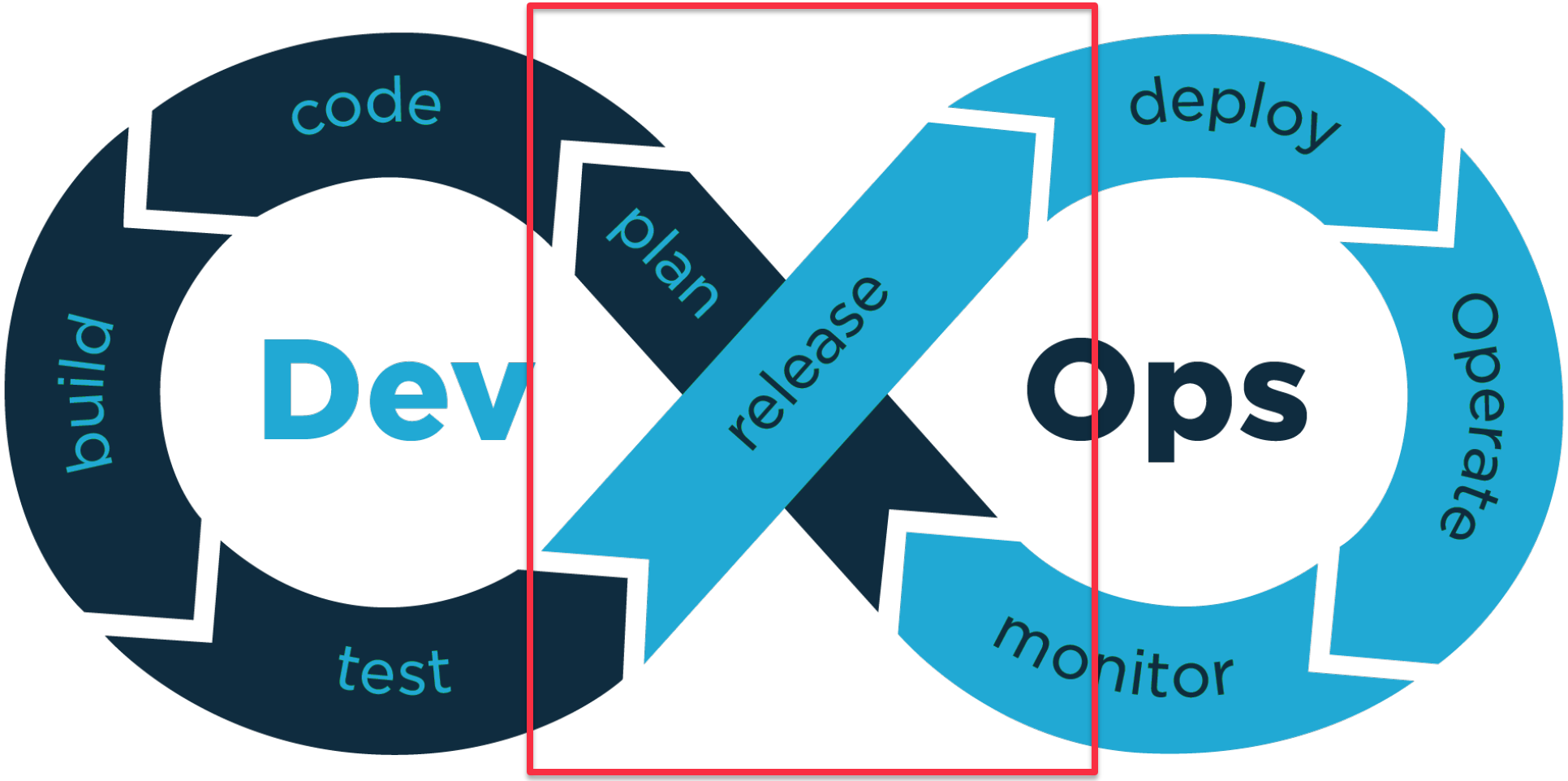


[Stanik PhD 2020]

Outline of my talk

- ☑ Motivation
- ☑ User feedback
- ☑ Usage data
- 4 **Summary and future directions**

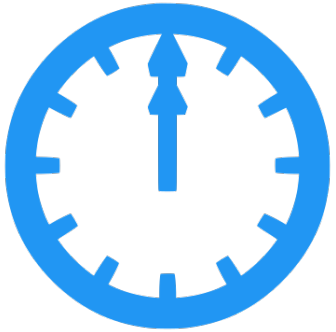
User feedback and usage data analytics tap the full potential of DevOps



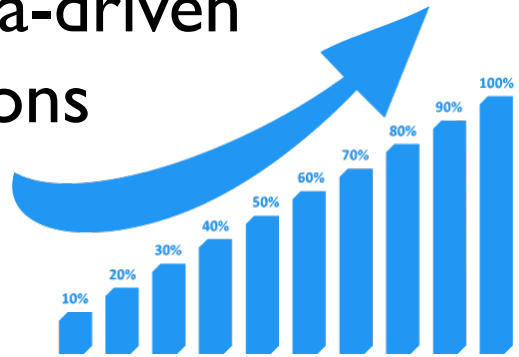
Impact on RE/SE decision making

Who: involve more stakeholders and users

When: real time and even proactive



How: Shift from intuition and experience to data-driven explainable decisions



What: Features can evolve and die

[Maalej et al Software 2016]

„The **hardest** single part of building a software system is **deciding** precisely **what** to build."

Fred Brooks
(Turing Award Winner) 1987

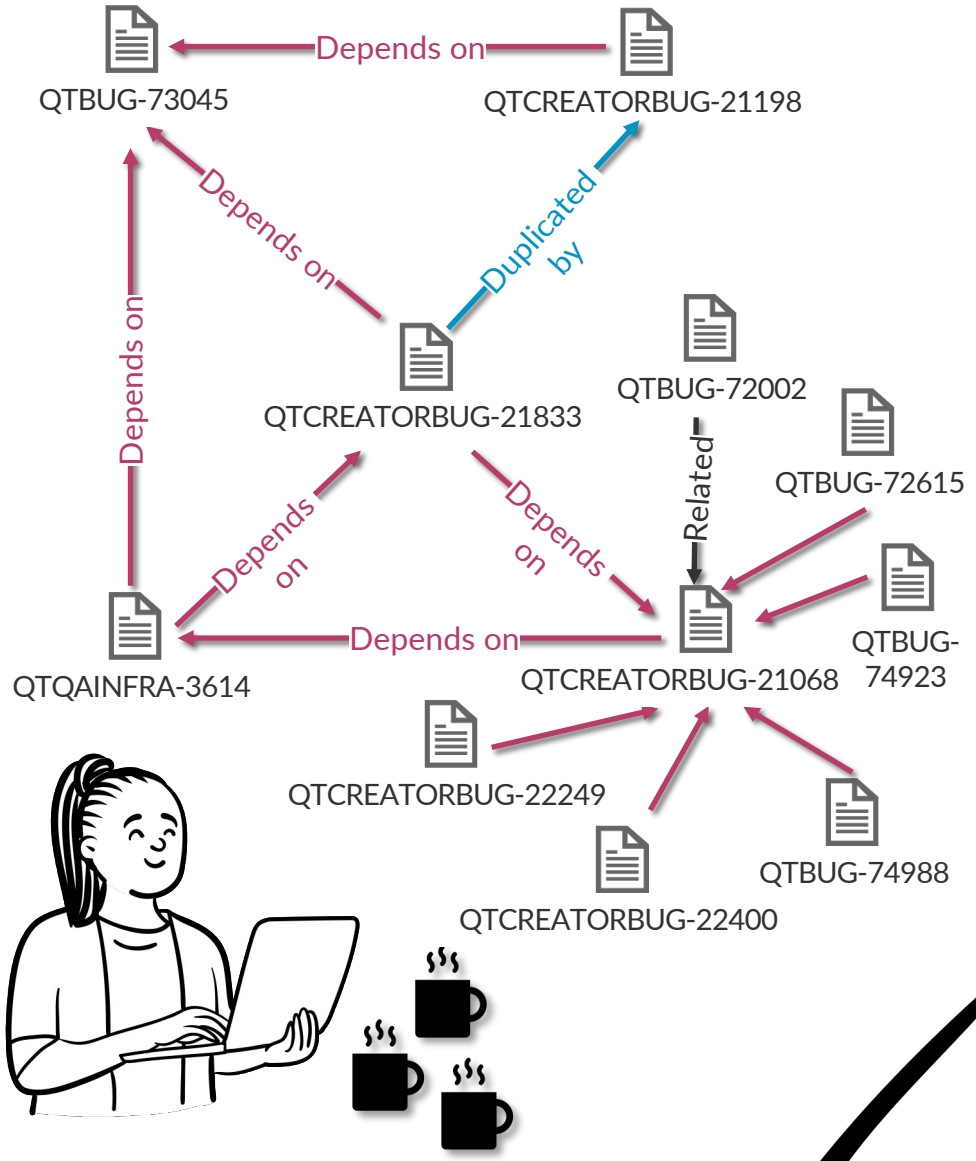
7

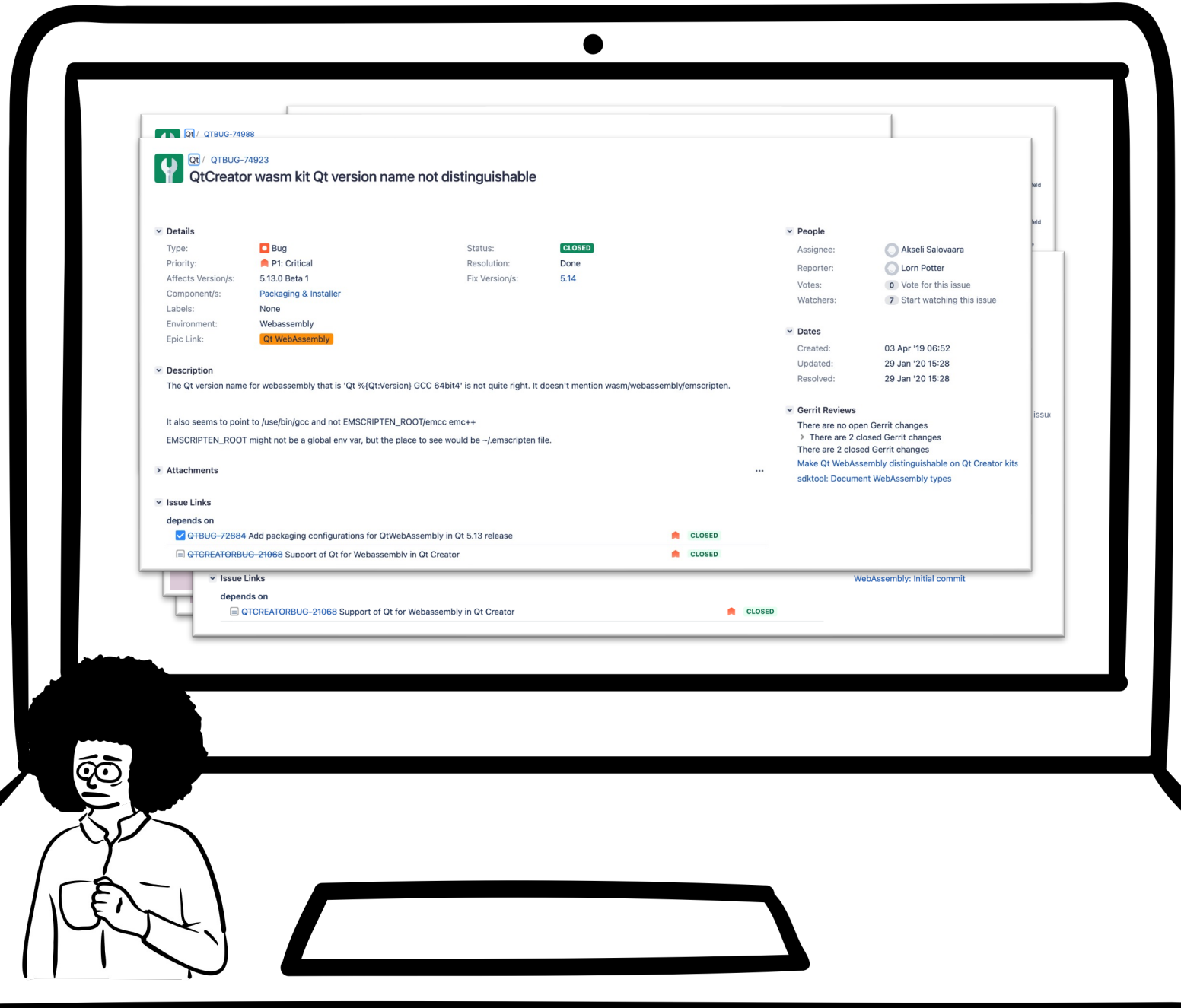
User data can make modern software engineering a "little less hard"

References

- C. Stanik, T. Pietz, W. Maalej. Unsupervised Topic Discovery in User Comments. In IEEE 29th International Requirements Engineering Conference (RE), 150-161, 2021
- JS. Andersen, O. Zukunft, W. Maalej, REM: Efficient Semi-Automated Real-Time Moderation of Online Forums. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, ACL 2021
- M. Haering, C. Stanik, W. Maalej, Automatically Matching Bug Reports With Related App Reviews. In Proceedings of the 43rd Int. Conf. on Software Engineering, 2021
- C. Stanik, M. Haering, C. Jesdabodi, W. Maalej. Which App Features Are Being Used? Learning App Feature Usages from Interaction Data. In Proceedings of the 28th IEEE International Requirements Engineering Conference (RE'20), 2020.
- D. Martens, W. Maalej. Release early, release often, and watch your users' emotions, IEEE Software, 2019.
- D. Martens, W. Maalej. Towards Understanding and Detecting Fake Reviews in App Stores, Empirical Software Engineering Journal, 2019
- D. Martens, W. Maalej. Extracting and Analyzing Context Information in User-Support Conversations on Twitter Conference, in Proceedings of the 27th IEEE International Requirements Engineering Conference (RE'19), 2019.
- Z. Kurtanović, W. Maalej. On user rationale in software engineering. In Requirements Engineering Journal, 2018.
- M. Häring, W. Loosen, W. Maalej, Who is Addressed in This Comment? Automatically Classifying Meta-Comments in News Comments Journal Article: Proceedings of the ACM on Human-Computer Interaction - CSCW, 2 (CSCW), pp. 67:1–67:20, 2018, ISSN: 2573-0142
- T. Johann, C. Stanik, A. M. Alizadeh, W. Maalej, SAFE: A Simple Approach for Feature Extraction from App Descriptions and App Reviews. In IEEE 25th International Requirements Engineering Conference (RE), 2017
- Z. Kurtanović, W. Maalej, Mining User Rationale from Software Reviews, In IEEE 25th International Requirements Engineering Conference (RE), 2017
- Z. Kurtanović, W. Maalej, Automatically Classifying Functional and Non-Functional Requirements Using Supervised Machine Learning, In IEEE 25th International Requirements Engineering Conference (RE), 2017
- M. Gomez, B. Adams, W. Maalej, M. Monperrus, R. Rouvoy, App Store 2.0: From Crowd Information to Actionable Feedback in Mobile Ecosystems. In IEEE Software, 2016
- W. Maalej, Z. Kurtanović, H. Nabil, C. Stanik. On the automatic classification of app reviews. In Requirements Engineering Journal, 21, 3, Page(s): 311-331, 2016
- W. Maalej, M. Nayebi, T. Johann, G. Ruhe. Toward Data-Driven Requirements Engineering. In IEEE Software: Special Issue on the Future of Software Engineering, 33, 1, Page(s): 48-54, ISSN: 0740-7459, 2016
- C. Jesdabodi and W. Maalej, Understanding Usage States on Mobile Devices, In proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2015), 2015
- T. Johann and W. Maalej, Democratic Mass Participation of Users in Requirements Engineering? In Proceedings of the 23rd IEEE International Requirements Engineering Conference, IEEE, 2015
- W. Maalej and H. Nabil, Bug Report, Feature Request, or Just Rating? On Automatically Classifying User Reviews, In Proceeding of the 23rd International Requirements Engineering Conference, 2015
- E. Guzman and W. Maalej, How Do Users Like this Feature? A Fine Grained Sentiment Analysis of App Reviews. In Proceedings of the 22nd IEEE International Requirements Engineering Conference 2014
- D. Pagano and W. Maalej, User Feedback in the AppStore: An Empirical Study, In Proceedings of the 21st. IEEE International Conference on Requirements Engineering, IEEE, 2013
- W. Maalej and D. Pagano, On the Socialness of Software, In Proceedings of the International Conference on Social Computing and its Applications, IEEE 2011
- W. Maalej, H-J. Happel, A. Rashid, When Users Become Collaborators: Towards Continuous and Context-Aware User Input. In OOPSLA'09, ACM, 2009

Mining [Links in] Issue Trackers

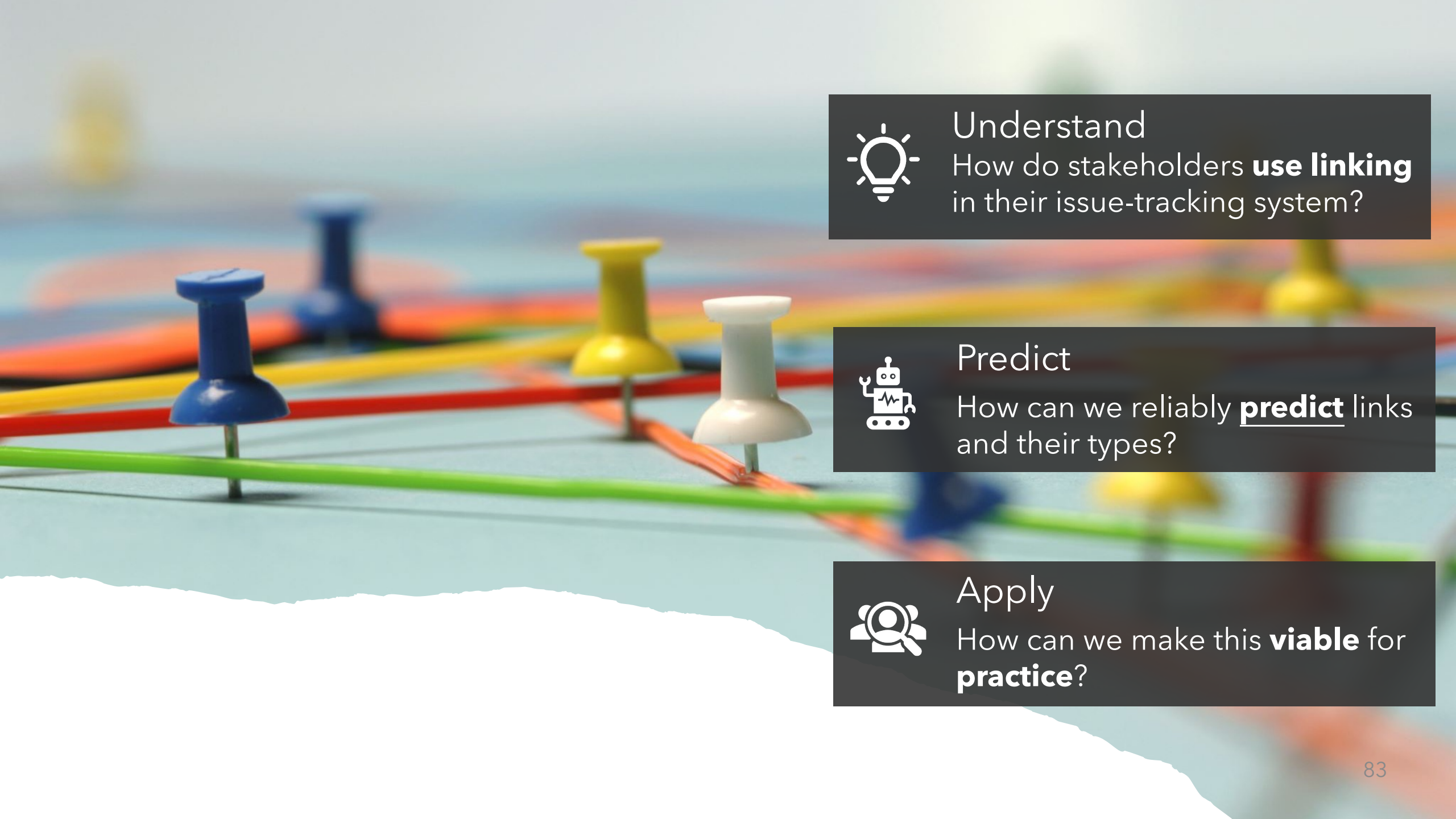

QTBUG-73045
Depends on
QTCREATORBUG-21198
Depends on
Duplicated by
Depends on
QTCREATORBUG-21833
QTBUG-72002
Related
QTBUG-72615
Depends on
Depends on
QTQAINFRA-3614
Depends on
QTCREATORBUG-21068
QTBUG-74923
QTCREATORBUG-22249
QTBUG-74988
QTCREATORBUG-22400


Qt / QTBUG-74988
Qt / QTBUG-74923
QtCreator wasm kit Qt version name not distinguishable
Details
Type: Bug
Priority: P1: Critical
Affects Version/s: 5.13.0 Beta 1
Component/s: Packaging & Installer
Labels: None
Environment: Webassembly
Epic Link: Qt WebAssembly
Status: CLOSED
Resolution: Done
Fix Version/s: 5.14
People
Assignee: Akseli Salovaara
Reporter: Lorn Potter
Votes: 0 Vote for this issue
Watchers: 7 Start watching this issue
Dates
Created: 03 Apr '19 06:52
Updated: 29 Jan '20 15:28
Resolved: 29 Jan '20 15:28
Description
The Qt version name for webassembly that is 'Qt %{Qt:Version} GCC 64bit4' is not quite right. It doesn't mention wasm/webassembly/emscripten.
It also seems to point to /use/bin/gcc and not EMSCRIPTEN_ROOT/emcc emc++
EMSCRIPTEN_ROOT might not be a global env var, but the place to see would be ~/.emscripten file.
Gerrit Reviews
There are no open Gerrit changes
There are 2 closed Gerrit changes
There are 2 closed Gerrit changes
Make Qt WebAssembly distinguishable on Qt Creator kits
sdktool: Document WebAssembly types
Attachments
Issue Links
depends on
QTBUG-72884 Add packaging configurations for QtWebAssembly in Qt 5.13 release CLOSED
QTCREATORBUG-21068 Support of Qt for Webassembly in Qt Creator CLOSED
Issue Links
depends on
QTCREATORBUG-21068 Support of Qt for Webassembly in Qt Creator CLOSED
WebAssembly: Initial commit

Dataset

- 16 JIRA Repositories
- 2.7M Issues
- 940K Links
- 8.5M Comments
- 1.276 Projects

Repository	Year	#Issues	#Links	#Comments	#Projects
Apache	2000	1,014,637	264,076	4,608,221	646
Hyperledger	2016	28,146	16,846	44,590	32
IntelDAOS	2016	9,474	2,667	32,203	2
JFrog	2006	15,535	3,303	13,152	10
Jira	2002	274,545	110,507	779,104	30
JiraEcosystem	2004	14,950	12,422	68,387	101
MariaDB	2009	31,229	14,906	-	11
Mindville	2015	2,134	46	-	7
Mojang	2012	420,819	215,821	933,348	8
MongoDB	2009	137,172	92,362	368,976	27
Qt	2003	148,579	41,402	41,426	21
RedHat	2001	353,000	127,721	859,880	241
Sakai	2004	50,550	20,292	180,191	53
SecondLife	2007	1,867	674	15,728	2
Sonatype	2008	87,284	4,975	339,127	5
Spring	2003	69,156	14,716	186,077	80
Total	-	2,659,077	942,736	8,470,410	1,276
Mean	2007.2	166,192.30	58,921.00	605,029.30	79.8
Median	2006.5	59,853.00	15,876.00	183,134.00	24
St. Dev	5.3	261,128.40	82,003.70	1,197,962.10	162.6



Understand

How do stakeholders **use linking** in their issue-tracking system?

Predict

How can we reliably **predict** links and their types?

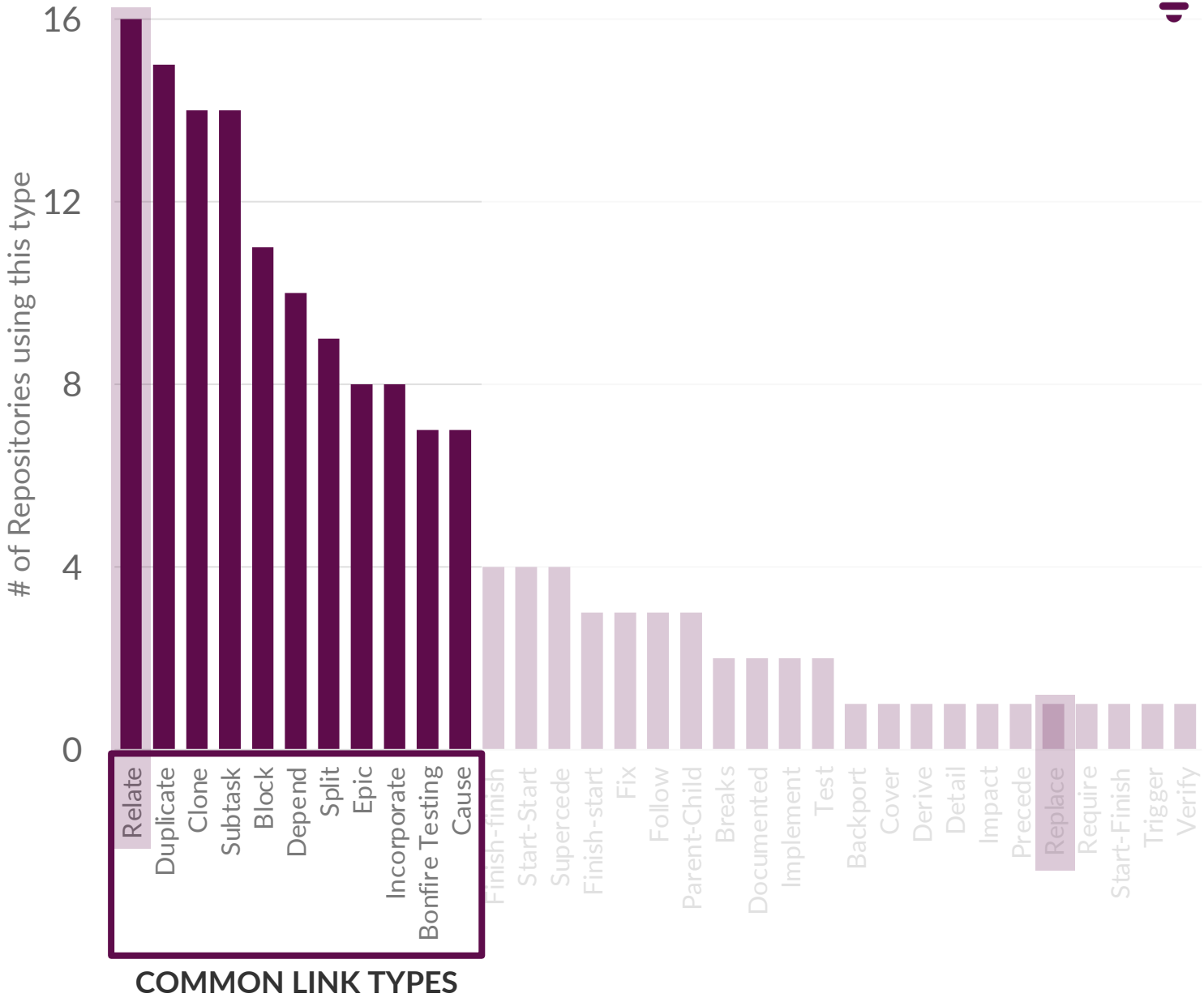
Apply

How can we make this **viable** for **practice**?

The Link Types

90 unique link types

33 cleaned link types



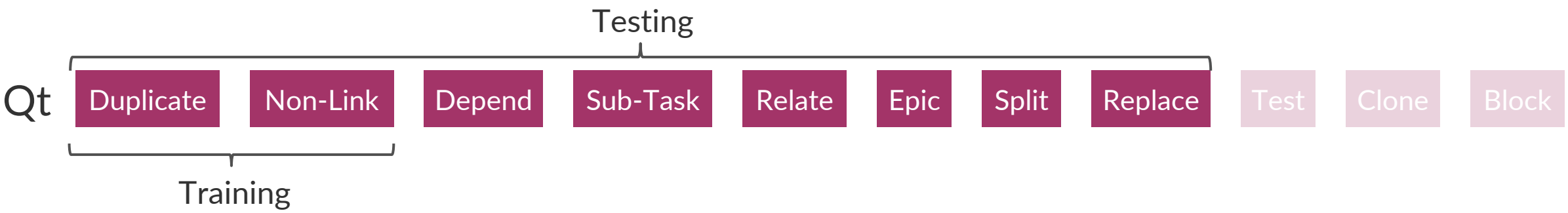
Lüders, Buraffa and Maalej, "Beyond duplicates: towards understanding and predicting link types in issue tracking systems", MSR2022

Repository	Rel.	Dup.	Sub.	Clo.	Blo.	Dep.	Epic	Spl.	Inc.	Bon.	Cau.	Cov.
Apache	28.3	10.1	32.8	1.7	6.1	5.1	4.9	0.0	4.1	0.0	1.2	94.3
Hyperledger	17.2	3.9	27.6	2.9	8.2	-	39.6	0.5	-	0.0	-	100.0
IntelDAOS	39.3	9.7	10.5	8.2	25.6	-	-	-	-	-		93.3
JFrog	27.4	19.9	36.0	0.8	-	7.9	-	-	1.4	-	-	93.5
Jira	63.8	21.7	2.5	2.9	1.0	0.2	-	0.2	2.5	0.2	1.8	96.6
JiraEcosystem	22.9	15.3	20.0	1.8	5.9	1.1	24.3	1.2	1.8	0.9	3.9	99.1
MariaDB	51.1	9.4	6.1	-	13.0	-	6.4	0.2	7.9	-	6.0	100.0
Mindville	43.2	38.6	-	15.9	2.3	-	-	-	-	-		100.0
Mojang	9.5	90.0	-	0.3	0.1	-	-	-	-	0.1	-	100.0
MongoDB	39.9	13.5	1.4	0.3	-	22.9	15.9	1.2	-	-	1.7	96.7
Qt	22.4	10.6	24.4	0.1	0.0	15.6	13.5	6.7	-	-	-	93.4
RedHat	25.9	4.9	20.8	15.4	15.2	-	-	0.1	8.9	-	2.6	94.0
Sakai	49.0	9.3	17.0	4.8	0.0	13.0	-	-	6.7	0.0	-	100.0
SecondLife	29.5	-	49.8	7.6	-	4.4	-	-	2.2	-	-	93.5
Sonatype	40.0	7.7	30.1	-	-	3.6	0.2	0.1	-	8.1	5.3	95.0
Spring	47.7	12.1	13.4	0.1	-	12.1	11.3	-	-	-	-	96.7
Mean	34.8	18.4	20.9	4.5	7.0	8.6	14.5	1.1	4.4	1.3	3.2	96.6
Median	34.4	10.6	20.4	2.4	5.9	6.5	12.4	0.2	3.3	0.1	2.6	96.7
St.Dev	14.3	21.5	13.8	5.4	8.1	7.2	12.5	2.1	3.0	3.0	1.9	2.8

Repositories differ in the link type shares

Link Prediction Models

Duplicate Detection Set-Up



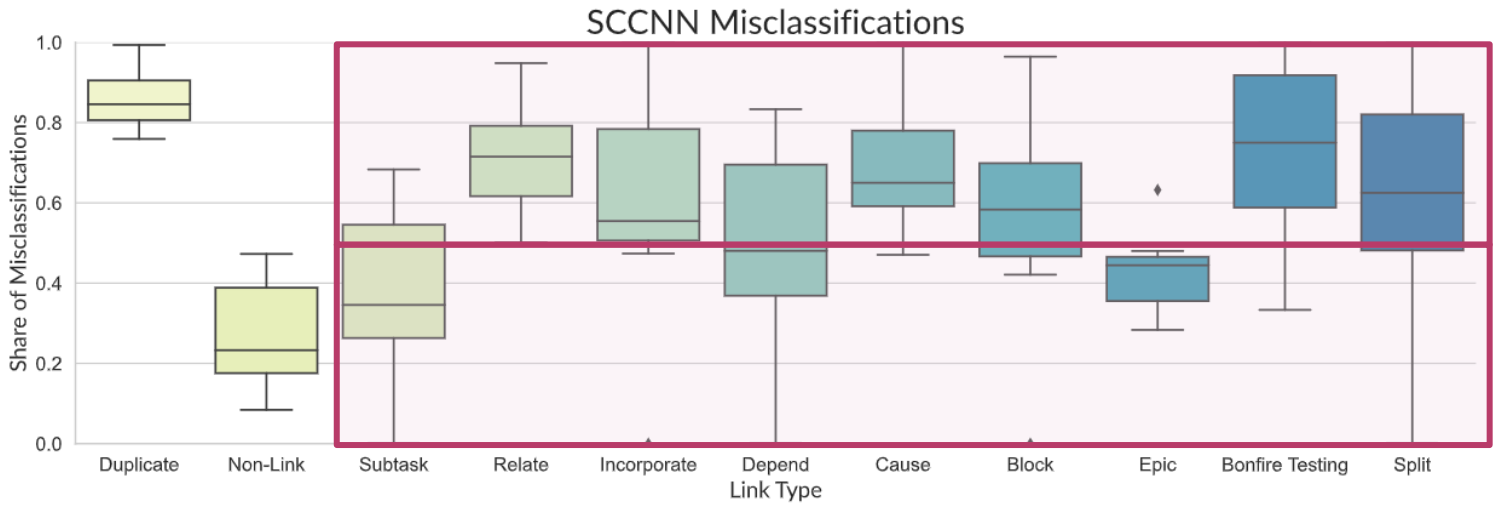
- Creation of Non-Links by randomly selecting pairs that do not have resolution *Duplicate**
- How robust are traditional trained SotA models when presented with other link types?

**Lazar et al. "Generating Duplicate Bug Datasets". MSR2014*

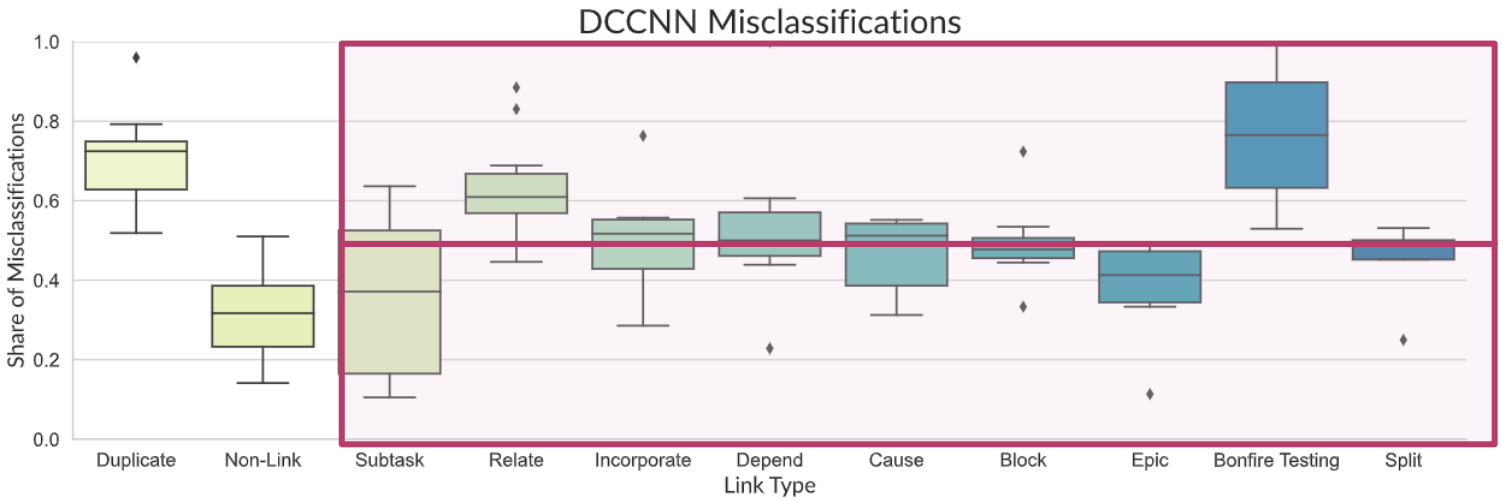
Can State-of-the-Art models **reliably** detect other link types as **Non-Duplicates**?

No, they can't.

Single-Channel

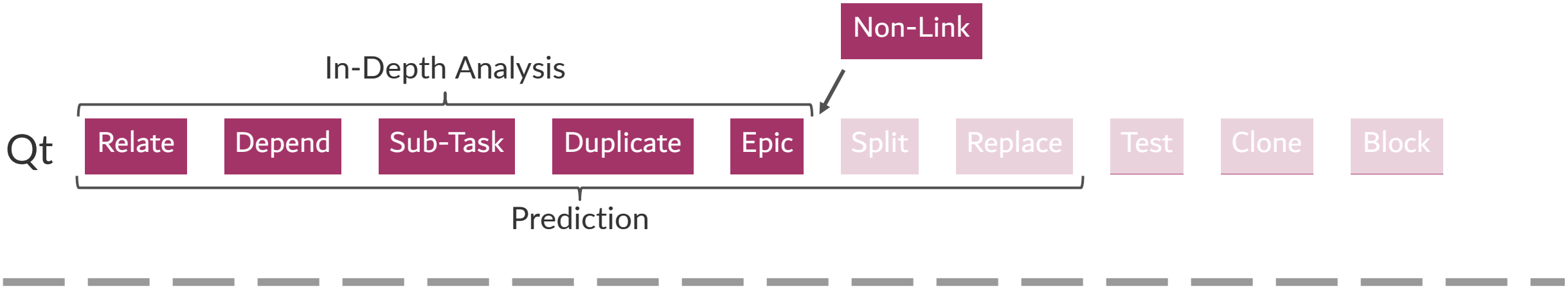


Dual-Channel



Lüders et al., "Beyond duplicates: towards understanding and predicting link types in issue tracking systems", MSR2022

Link Type Detection Set-Up

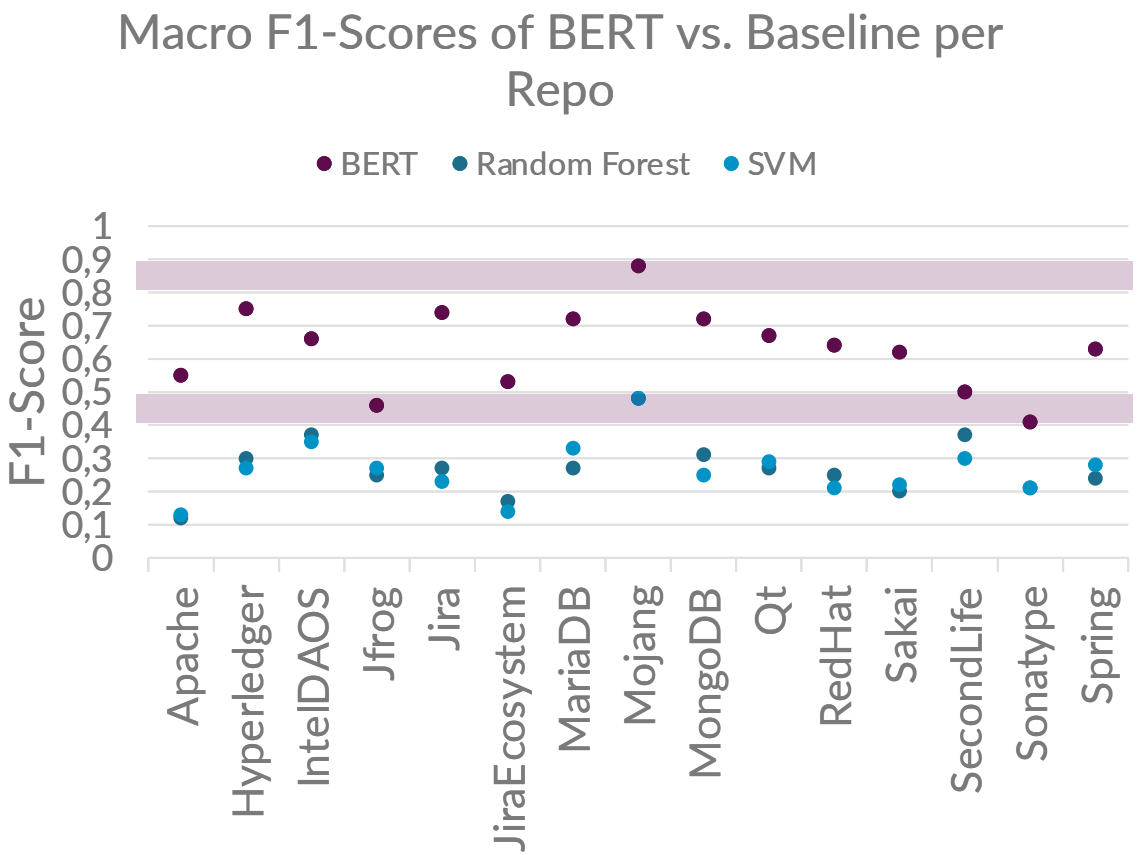


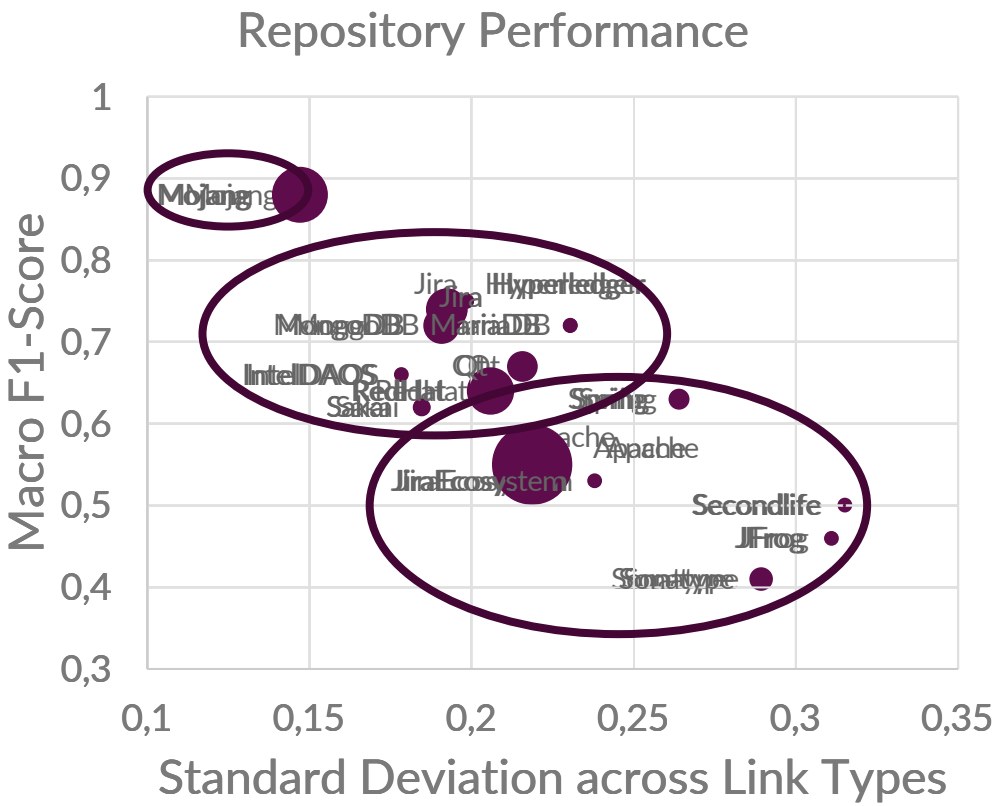
Deep Learning
- CNN with Word2Vec/FastText
- BERT & distillBERT

Traditional Models
- SVM
- Random Forest

BERT outperforms DCCNN and SCCNN

Link Type Detection Results



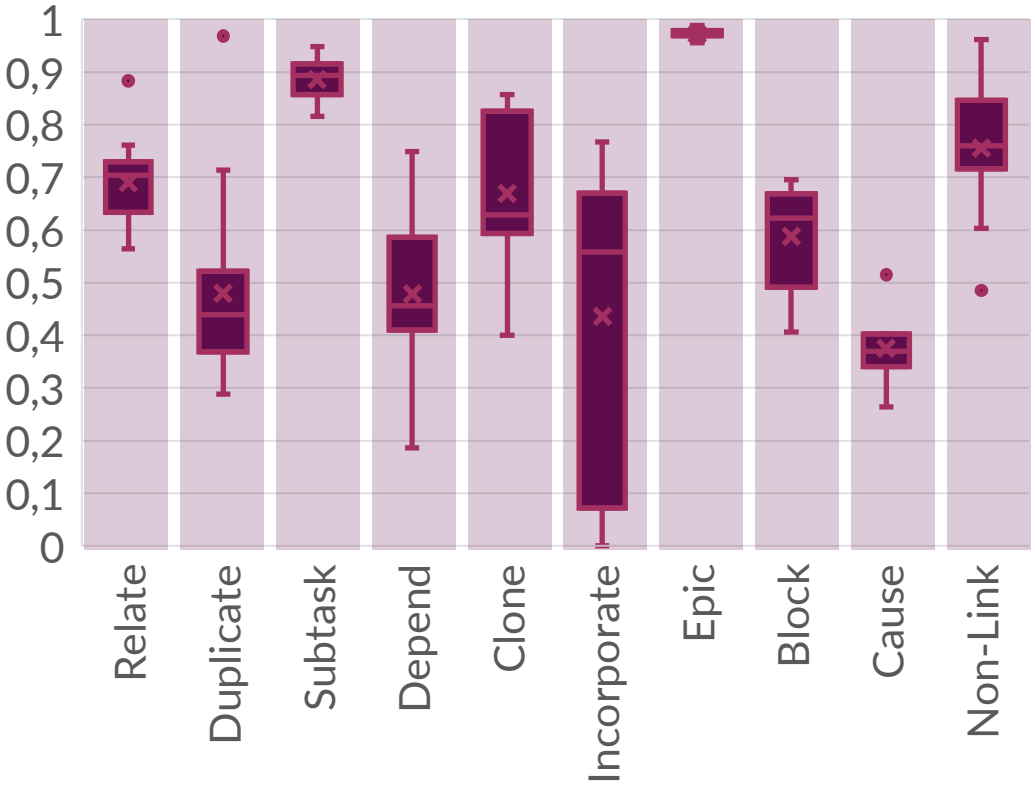


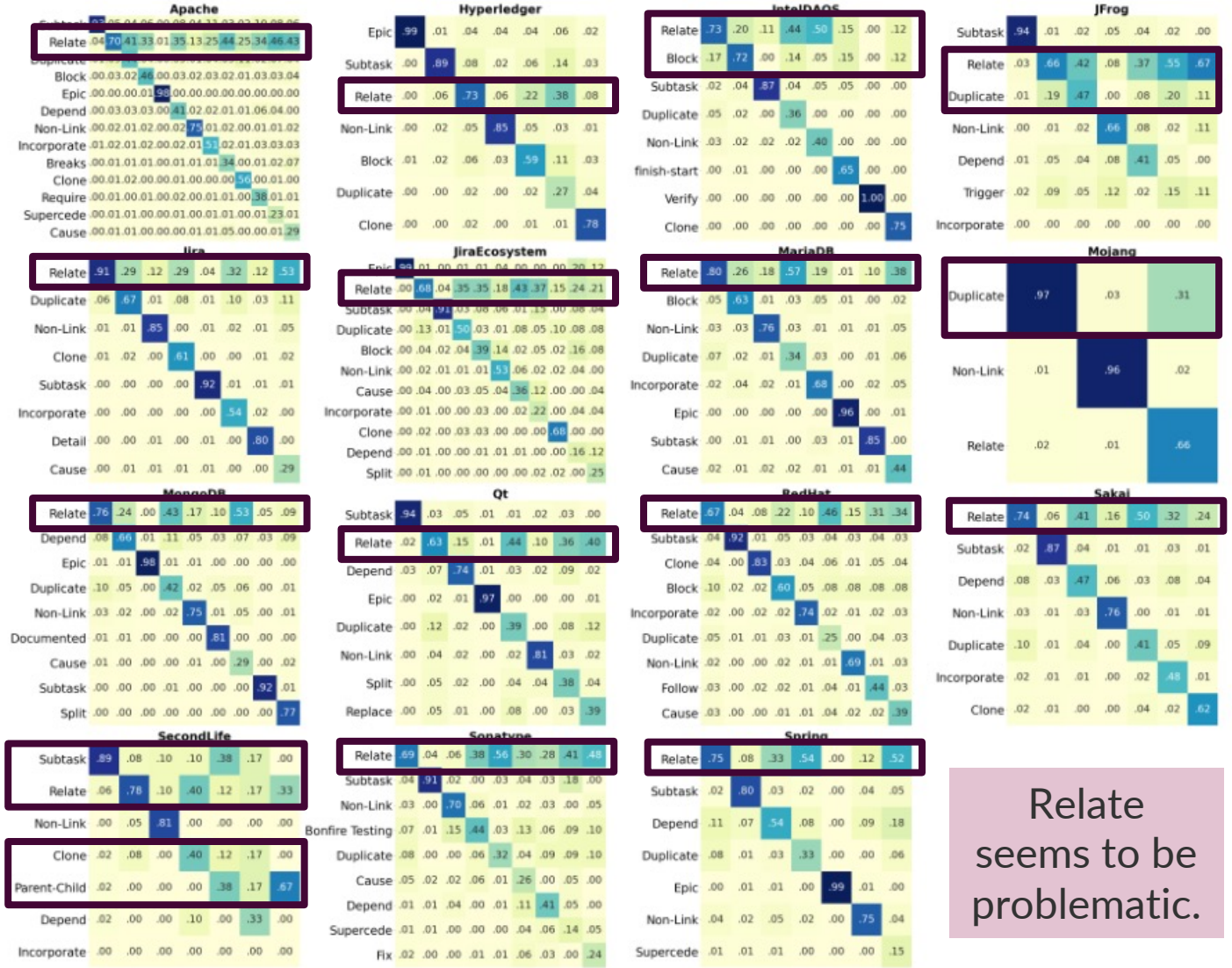
Significant Correlations

	Characteristic	Correlation
Basic	Coverage	0.85
	Common Link Type Coverage	0.52
	Duplicates	0.55
Link Types	Subtasks	-0.75
	% Isolated Issues	-0.79
	Assortativity	0.52
	Median Comp. Size	0.60
	Mean Comp. Size	0.71
Structure	St. Dev. Comp. Size	0.70
Communication	Comment Link Correlation	0.64
	Links Per Project	0.62
	Issues Per Maintainer	0.57
	Links Per Maintainer	0.62
	Issues Created Per Maintainer	0.60
	Mean Link Edits	0.62
	Median Link Edits	0.58
Collaboration	Links Edited Per Project	0.63
	Duplicate Issues without Links	-0.70
	Duplicate Issues without Duplicate Links	-0.62
Quality/Smells	Ref. Comment without Link	0.84

Link Type Analysis

F1-Score of Link Types across Repositories





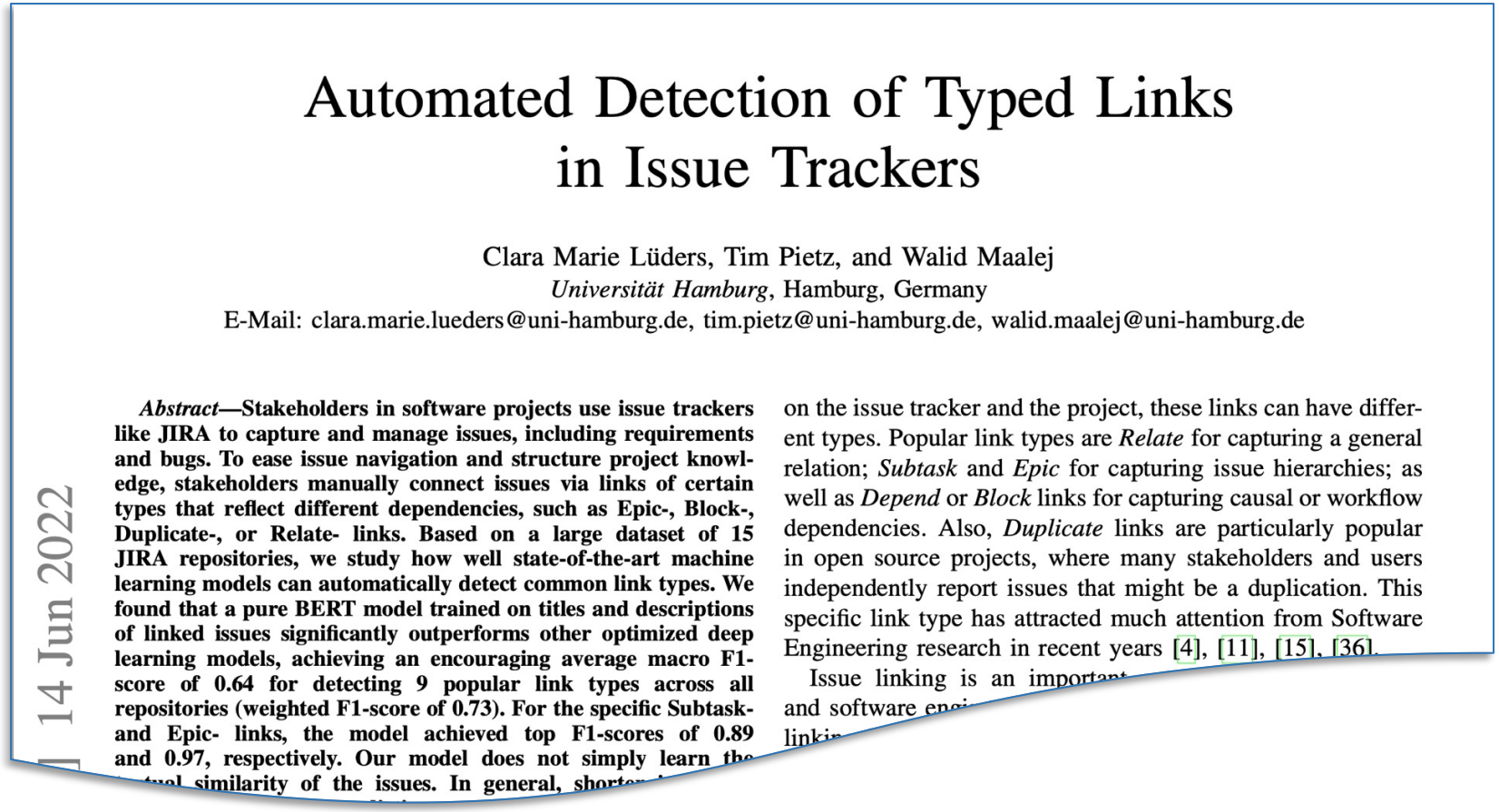
Relate seems to be problematic.

Prediction Improvement Strategies

Repository	Baseline	No Relate		Only Link Types		Categories		Only Linked	
Apache	0.55	0.62	0.08	0.62	0.07	0.66	0.11	0.97	0.42
Hyperledger	0.77	0.79	0.02	0.80	0.03	0.71	-0.06	0.95	0.18
IntelDAOS	0.66	0.75	0.09	0.79	0.13	0.68	0.02	0.89	0.23
JFrog	0.46	0.55	0.09	0.49	0.03	0.55	0.09	0.93	0.47
Jira	0.74	0.80	0.06	0.78	0.04	0.72	-0.02	0.98	0.24
JiraEcosystem	0.53	0.58	0.05	0.57	0.04	0.63	0.10	0.94	0.41
MariaDB	0.72	0.82	0.10	0.79	0.07	0.73	0.01	0.93	0.21
Mojang	0.88	0.98	0.10	-	-	0.88	+/-0	0.99	0.11
MongoDB	0.72	0.80	0.08	0.82	0.10	0.73	0.01	0.95	0.23
Qt	0.67	0.75	0.08	0.74	0.07	0.63	-0.04	0.96	0.29
RedHat	0.64	0.70	0.06	0.69	0.05	0.72	0.08	0.96	0.32
Sakai	0.62	0.74	0.12	0.73	0.11	0.66	0.04	0.94	0.32
SecondLife	0.50	0.36	-0.14	0.41	-0.09	0.57	0.07	0.90	0.40
Sonatype	0.41	0.58	0.17	0.71	0.30	0.53	0.12	0.95	0.55
Spring	0.63	0.73	0.10	0.73	0.10	0.12	-0.41	0.94	0.31
Mean	0.63	0.70	0.07	0.69	0.08	0.63	0.01	0.95	0.31
Median	0.64	0.74	0.08	0.73	0.07	0.66	0.02	0.95	0.31
St. Dev	0.13	0.15	0.07	0.12	0.08	0.17	0.13	0.03	0.12

C-M Lueders, T Pietz, W Maalej: “On understanding and predicting issue links” REJ 2023

Read more…



Automated Detection of Typed Links in Issue Trackers

Clara Marie Lüders, Tim Pietz, and Walid Maalej
Universität Hamburg, Hamburg, Germany
E-Mail: clara.marie.lueders@uni-hamburg.de, tim.pietz@uni-hamburg.de, walid.maalej@uni-hamburg.de

***Abstract*—Stakeholders in software projects use issue trackers like JIRA to capture and manage issues, including requirements and bugs. To ease issue navigation and structure project knowledge, stakeholders manually connect issues via links of certain types that reflect different dependencies, such as Epic-, Block-, Duplicate-, or Relate- links. Based on a large dataset of 15 JIRA repositories, we study how well state-of-the-art machine learning models can automatically detect common link types. We found that a pure BERT model trained on titles and descriptions of linked issues significantly outperforms other optimized deep learning models, achieving an encouraging average macro F1-score of 0.64 for detecting 9 popular link types across all repositories (weighted F1-score of 0.73). For the specific Subtask- and Epic- links, the model achieved top F1-scores of 0.89 and 0.97, respectively. Our model does not simply learn the**

on the issue tracker and the project, these links can have different types. Popular link types are *Relate* for capturing a general relation; *Subtask* and *Epic* for capturing issue hierarchies; as well as *Depend* or *Block* links for capturing causal or workflow dependencies. Also, *Duplicate* links are particularly popular in open source projects, where many stakeholders and users independently report issues that might be a duplication. This specific link type has attracted much attention from Software Engineering research in recent years [4], [11], [15], [36].

Issue linking is an important

and software engi

linkin

https://arxiv.org/abs/2206.07182

Extended version:

https://link.springer.com/article/10.1007/s00766-023-00406-x