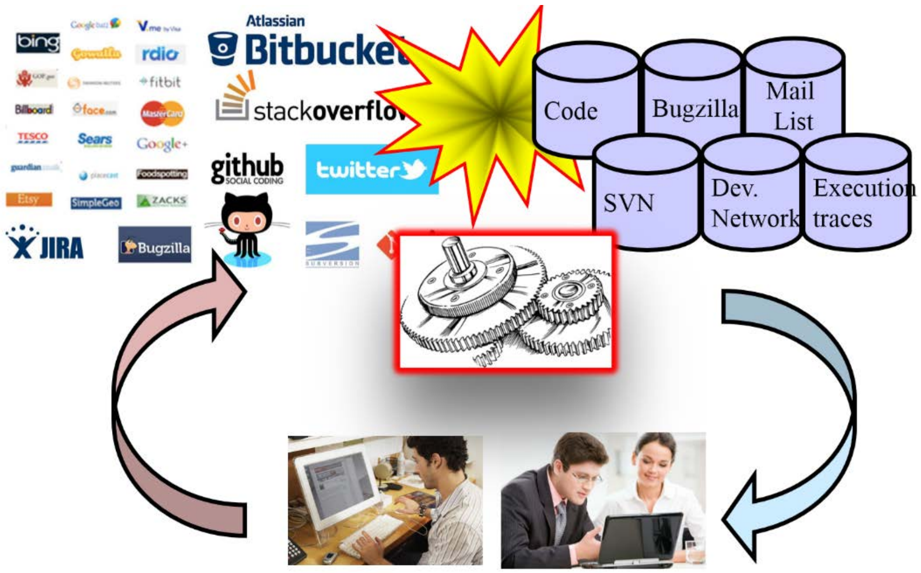
武漢大學 人工智能与软件工程暑期学校

2020-08

演讲人：夏 鑫

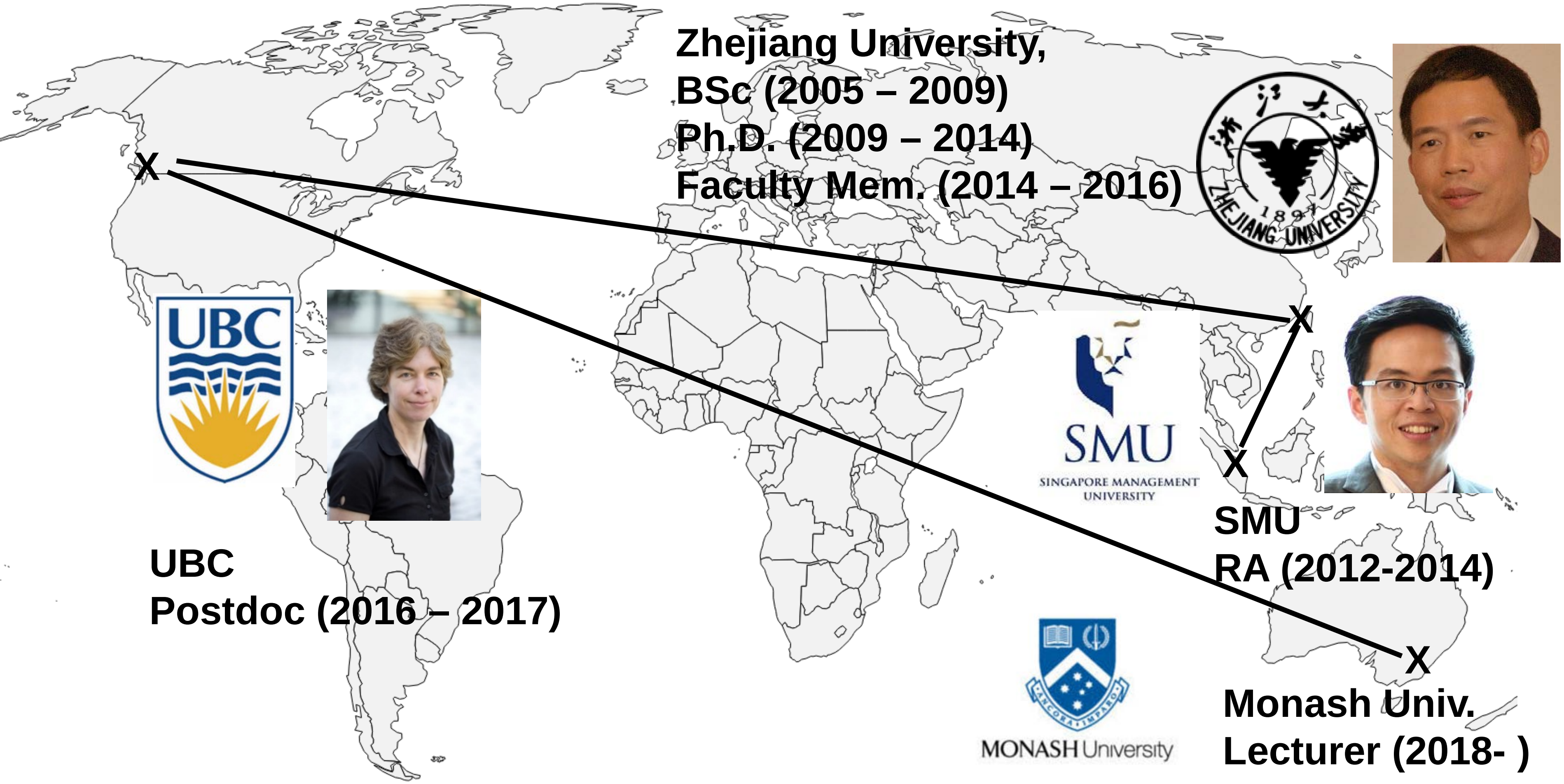
澳大利亚蒙纳士大学ARC DECRA Fellow和讲师。研究方向是软件仓库挖掘和软件解析学。至今发表了169篇期刊和会议论文，其中包括42篇CCF A类期刊和会议长文（包括22篇TSE，8篇ASE），57篇CCF B类期刊和会议长文。部分论文获得国际会议最佳/杰出论文奖项，包括4篇ACM SIGSOFT Distinguished Paper Award（ASE 2018和2019，ICPC 2018和2020），ESEC/FSE 2019 Best Tool Demo Award。此外他担任了MSR和SANER会议的Steering Committee，多个国际会议的PC (ICSE，ESEC/FSE, ASE等)，以及参与组织了多个国际会议（ASE 2020，ICSME 2020, SANER 2019等）。更多信息在https://xin-xia.github.io/

Bridging the Gap Between AI and Software Engineering



Xin Xia
ARC DECRA Fellow
Monash University
Xin.Xia@monash.edu

About Myself



Zhejiang University,
BSc (2005 – 2009)
Ph.D. (2009 – 2014)
Faculty Mem. (2014 – 2016)

UBC
Postdoc (2016 – 2017)

SMU
RA (2012-2014)

Monash Univ.
Lecturer (2018-)

Background

- **Software Engineering Data**

14M+ Questions,
19M+ Answers

63M+ Projects,
23M+ Developers

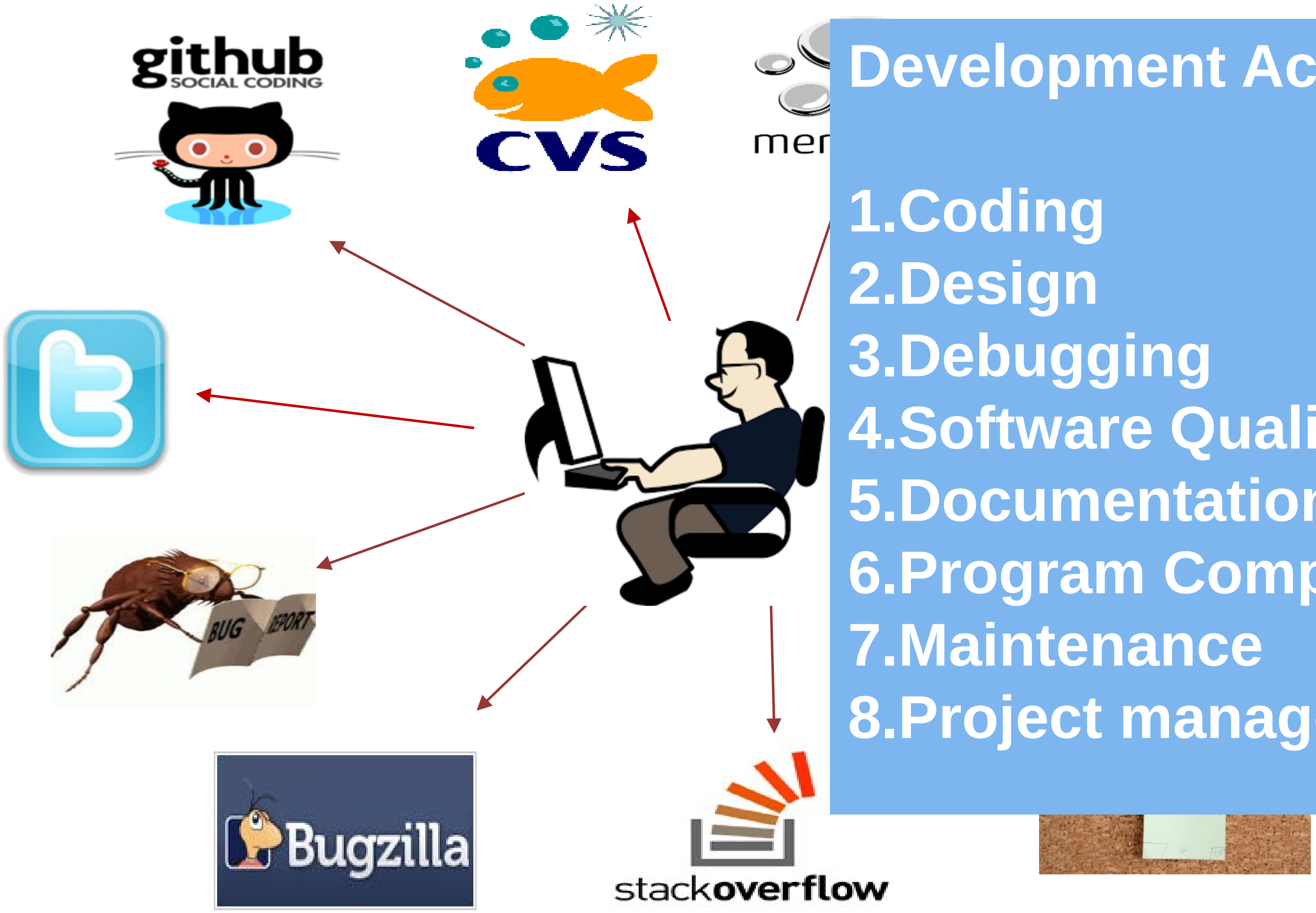
- # Heterogeneous Software Artifacts

Analyzing SE Data is Complex!

Source Code	• Structure, Execution Trace, Comments
Development History	• Natural Language, Patch, Time, Developer
Bug Reports	• Meta Data, Natural Language, Code, Discussions
Code Reviews, Pull Requests	• Natural Language, Patch, Discussions
Software Forums	• Natural Language, Code Snippets, Comments

- # What Do Software Engineers Do

Software Development is Complex!



Development Activities

1. Coding
2. Design
3. Debugging
4. Software Quality
5. Documentation
6. Program Comprehension
7. Maintenance
8. Project management

Development Activities

1.Coding
2.Design
3.Debugging
4.Software Qual
5.Docum
6.
ension
t management

How to leverage software data to handle the complexity of software development?

Software Engineer

State of AI Today

Data Scientist for AI People

Intelligent Techniques:

1. Statistical analysis
2. Data Mining
3. Machine Learning
4. Natural Language Processing
5. Deep Learning

…

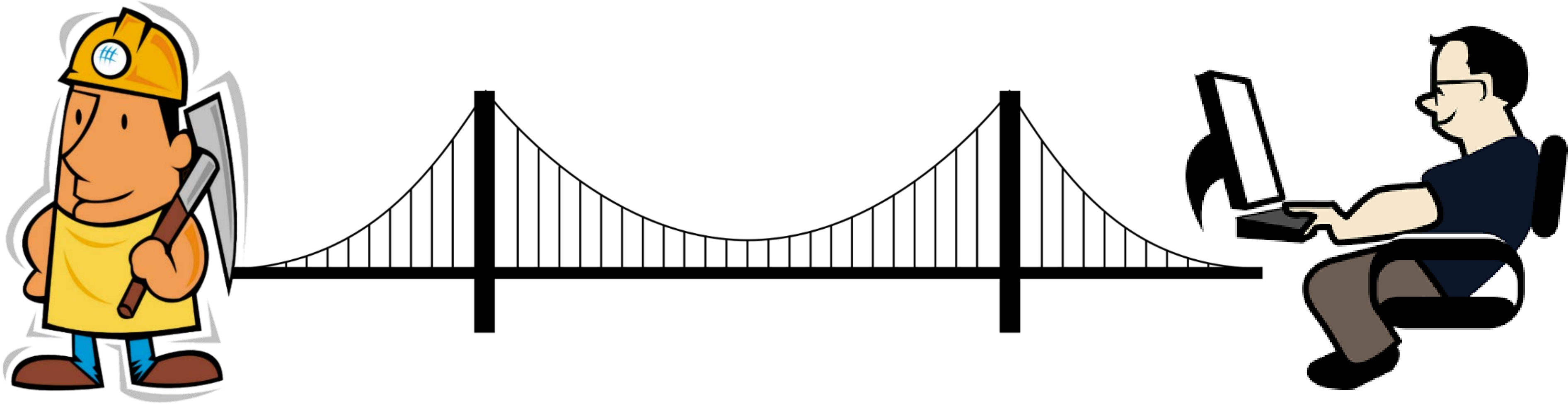
⋮

Data in Domain N

Gaps Between AI and Software Engineering

	Knowledge of Domains	Knowledge of Techniques
Data Scientist	**Low**	**High**
Software Engineers	**High**	**Low**

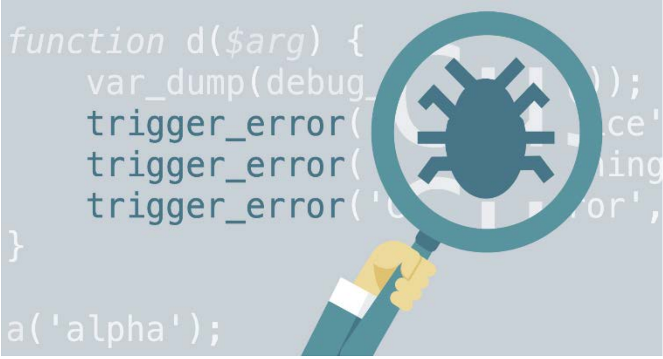
My Research



Build automated tools by mining and analyzing the rich data in software repositories, to handle the complexity of software development

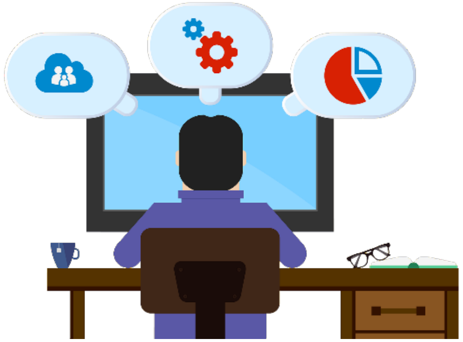
Research Topics

Bug Localization & Repair

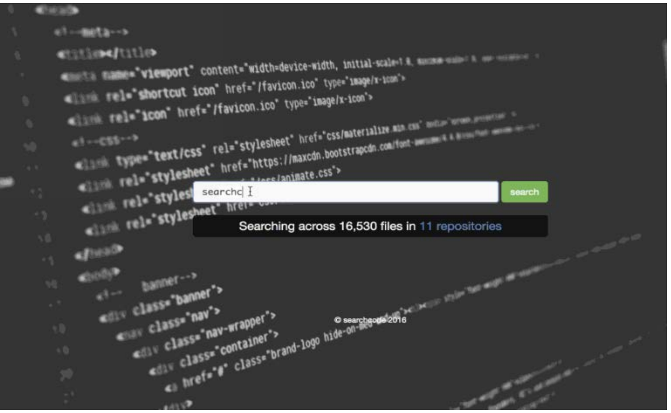


Software Artifacts Generation

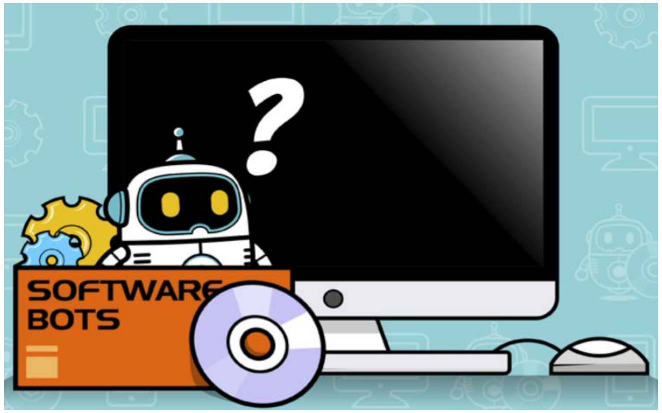
Human-centric SE



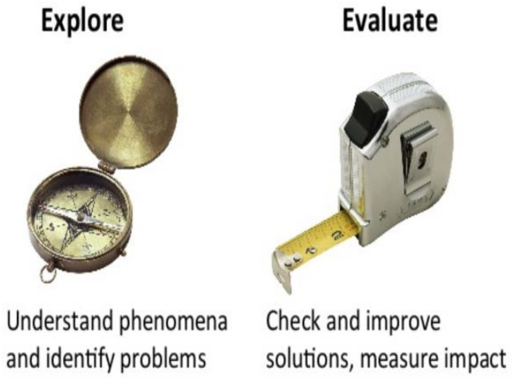
Intelligent Code Recommender



Software Bots



Empirical SE


Explore
Evaluate
Understand phenomena and identify problems
Check and improve solutions, measure impact

Example Research

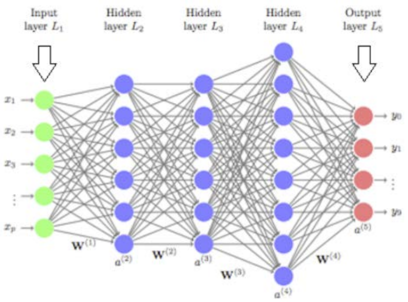
- Automatic **Generation of Pull Request Descriptions**
- **API Method Recommendation** without Worrying About the Task-API Knowledge Gap
- Automating Change-Level **Self-Admitted Technical Debt Determination**
- **Chatbot4QR**: Interactive Query Refinement for Technical Question Retrieval
- **VT-Revolution**: Interactive Programming Video Tutorial Authoring and Watching System

Example Research 1

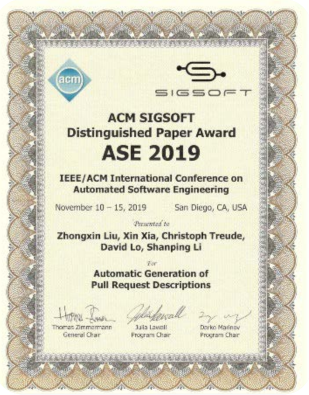
Automatic Generation of Pull Request Descriptions

ACM SIGSOFT Distinguished Paper Award

GitHub



TXT



SIGSOFT

ACM SIGSOFT
Distinguished Paper Award
ASE 2019

IEEE/ACM International Conference on
Automated Software Engineering

November 10 – 15, 2019 San Diego, CA, USA

Presented to

Zhongxin Liu, Xin Xia, Christoph Treude,
David Lo, Shanping Li

For

Automatic Generation of
Pull Request Descriptions

Thomas Zimmermann
General Chair

Julia Lawall
Program Chair

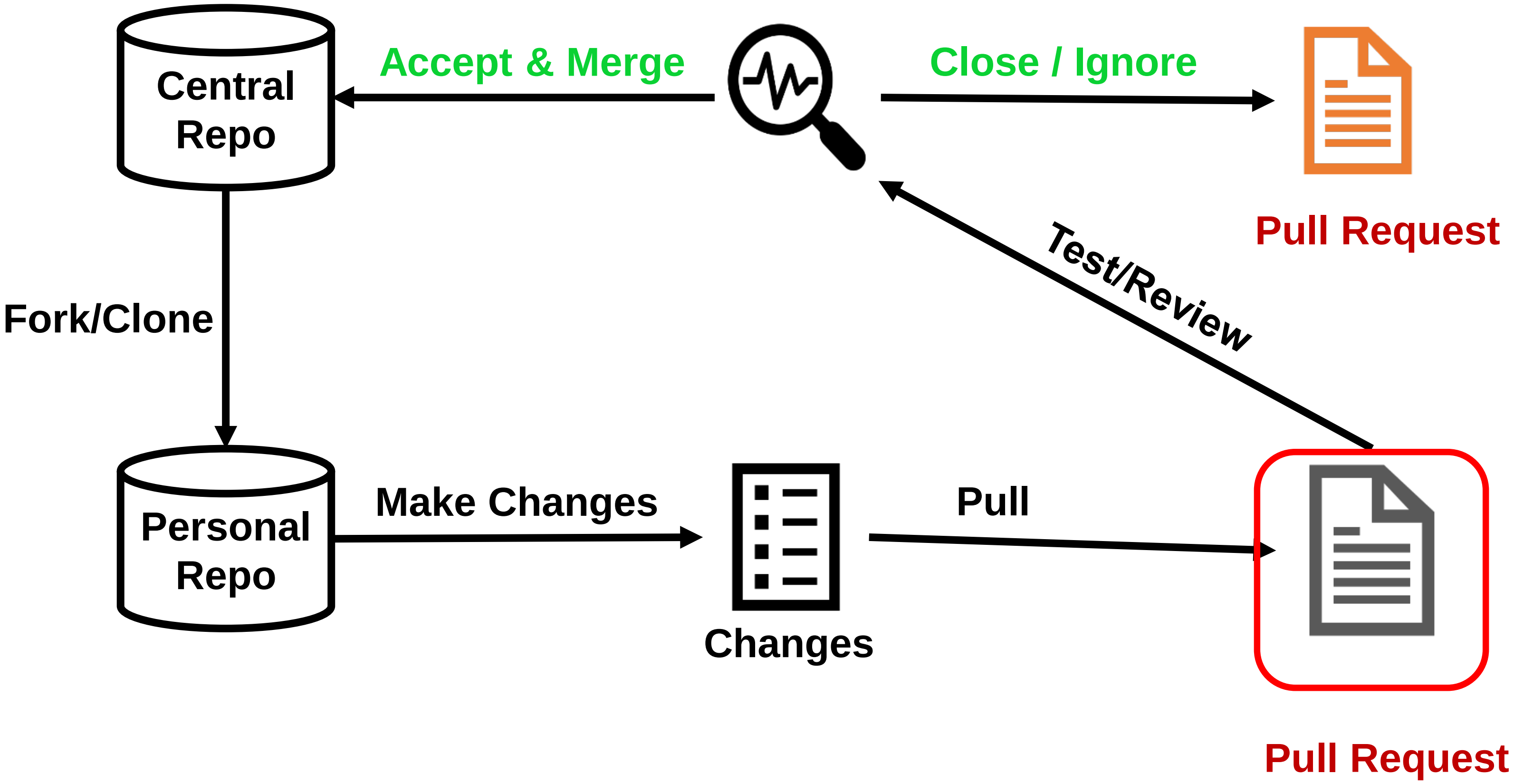
Darko Marinov
Program Chair

Deep Learning

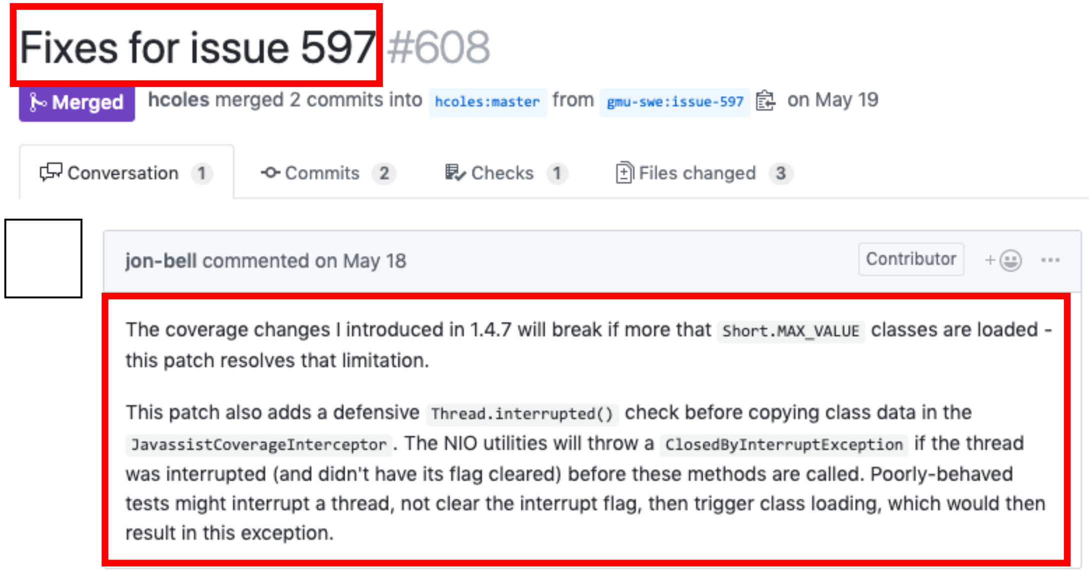
Pull Requests

Liu, **Xia**, et al. ASE 2019

The Pull-Based Development



Pull Request (PR) Description



Fixes for issue 597 #608

Merged hcoles merged 2 commits into `hcoles:master` from `gmu-swe:issue-597` on May 19

jon-bell commented on May 18

The coverage changes I introduced in 1.4.7 will break if more that `Short.MAX_VALUE` classes are loaded - this patch resolves that limitation.

This patch also adds a defensive `Thread.interrupted()` check before copying class data in the `JavassistCoverageInterceptor`. The NIO utilities will throw a `ClosedByInterruptException` if the thread was interrupted (and didn't have its flag cleared) before these methods are called. Poorly-behaved tests might interrupt a thread, not clear the interrupt flag, then trigger class loading, which would then result in this exception.

PR title

PR Description

Sometimes neglected by Devs

34% of 333K PRs

Reduce probability of being ignored or rejected

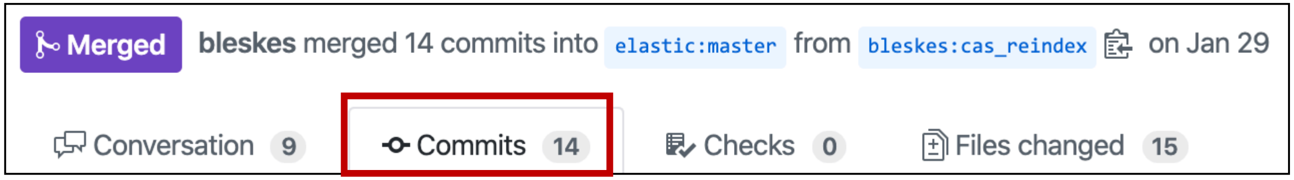
Help assign tags

Help identify related PRs

Automated PR Description Generation

Automatic Generation of PR Descriptions

- A PR often contains multiple commits



bleskes merged 14 commits into `elastic:master` from `bleskes:cas_reindex` on Jan 29

- It’s challenging to even summarize a single commit.
  - Jiang et al (ASE 2017). Liu, **Xia** et al (ASE 2018).

Motivating Example

A PR in the Pitest Project
PR Description:
Added an option to ignore failing tests from coverage, activated from maven plugin
Commit 1:
CM: Added skipFailingTests option from maven plugin ***Added Comments:*** When set will Ignore failing tests when computing coverage. Otherwise, the run will fail. If parseSurefireConfig is true, will be overridden from surefire configuration property testFailureIgnore
Commit 2:
CM: Simplified surefire testFailureIgnore value retrieval ***Added Comments:*** N/A

Generating PR Desc by summarizing commit msgs and code comments

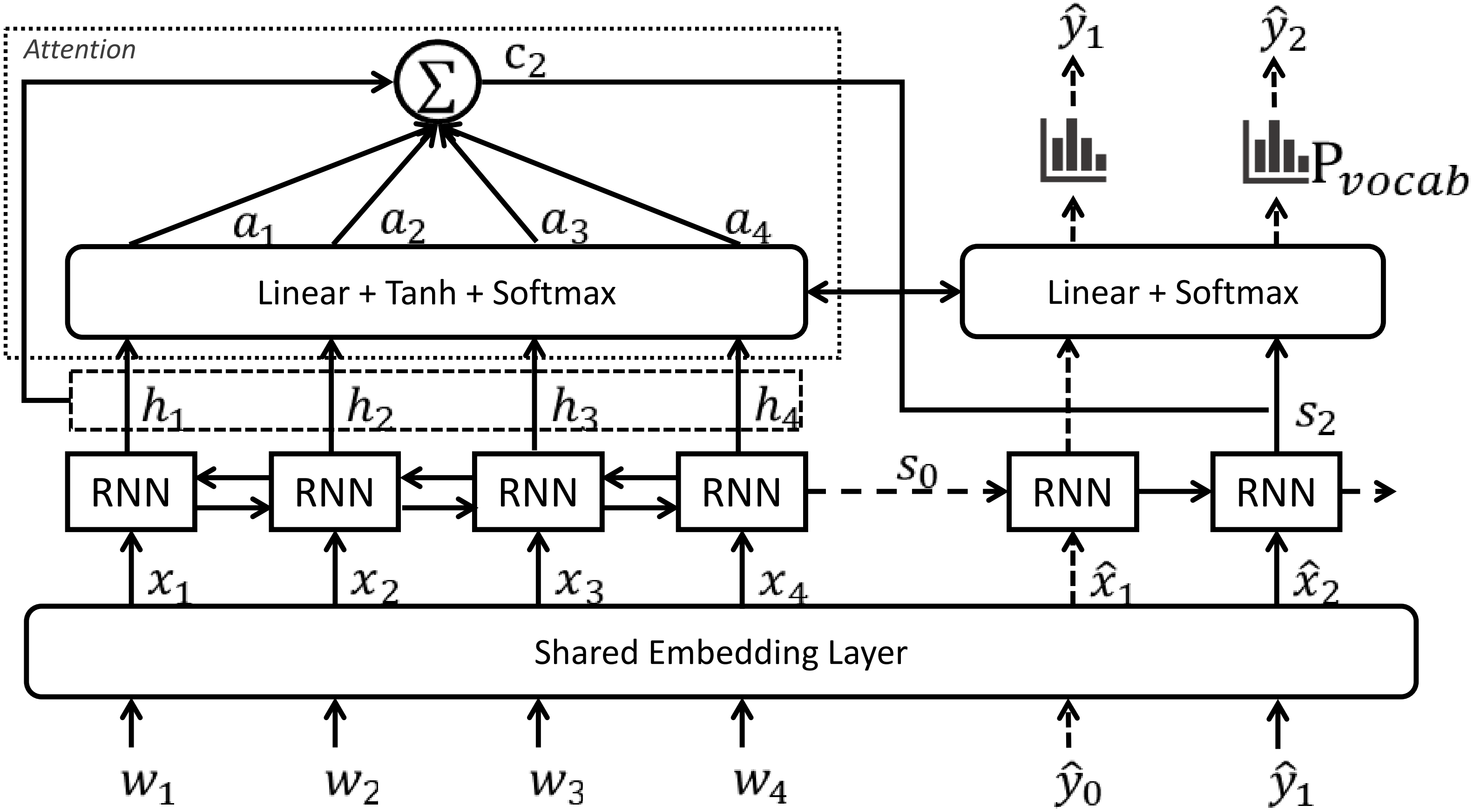
PR Description Generation through Summarization

- Regard the automatic generation of PR descriptions as a text summarization problem:
  - Summary: PR Description
  - Article: Commit Msgs + Added Code Comments

- Further formulated as a seq2seq learning problem:
  - Commit Msgs + Code Comments: $w = (w_1, w_2, \dots, w_{|w|})$
  - PR Description: $y = (y_1, y_2, \dots, y_{|y|})$
  - Target: Find a function f, so that $f(w) = y$

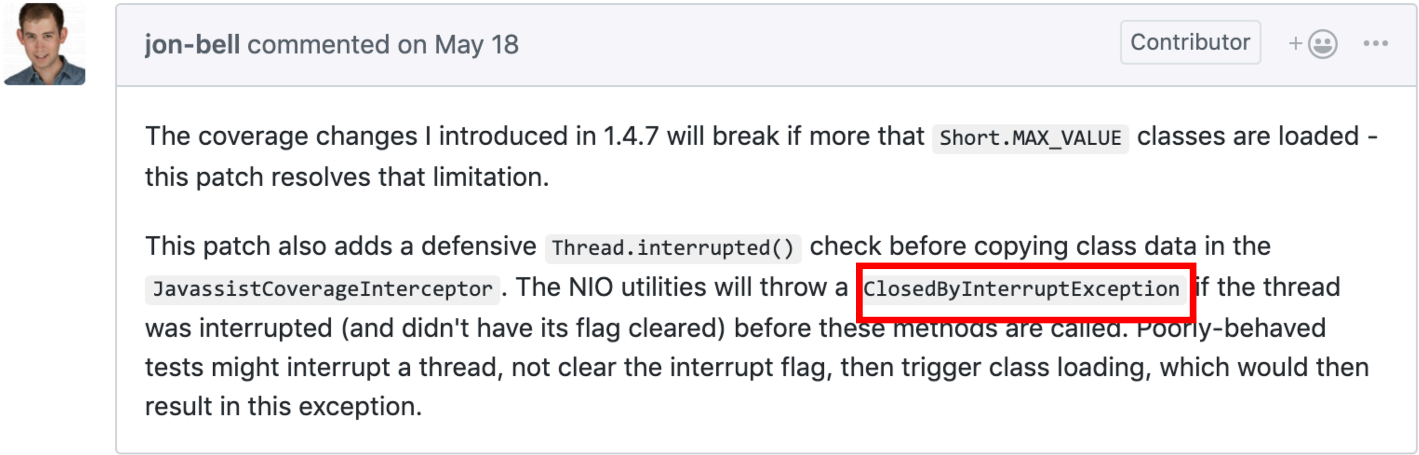
Approach – Attentional Encoder-Decoder Model

- Our approach is based on the Attentional Encoder-Decoder Model
  - A popular and effective model for seq2seq learning problems



Challenge: Out-of-Vocab Words

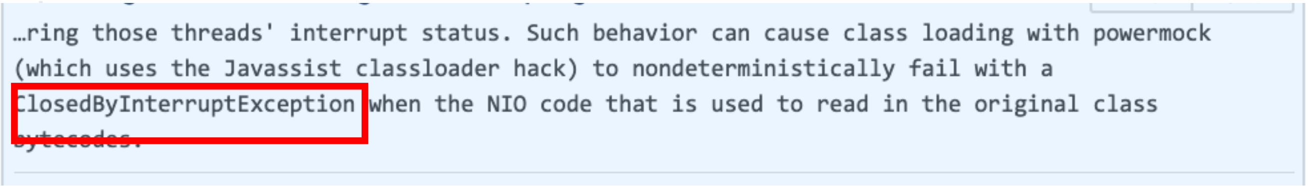
- Out-of-vocabulary (OOV) words are ubiquitous in software artifacts due to developer-named identifiers
  - e.g., *ClosedByInterruptException* may not appear in the training set



The coverage changes I introduced in 1.4.7 will break if more that `Short.MAX_VALUE` classes are loaded - this patch resolves that limitation.

This patch also adds a defensive `Thread.interrupted()` check before copying class data in the `JavassistCoverageInterceptor`. The NIO utilities will throw a `ClosedByInterruptException` if the thread was interrupted (and didn't have its flag cleared) before these methods are called. Poorly-behaved tests might interrupt a thread, not clear the interrupt flag, then trigger class loading, which would then result in this exception.

PR Description



```
…ring those threads' interrupt status. Such behavior can cause class loading with powermock
(which uses the Javassist classloader hack) to nondeterministically fail with a
ClosedByInterruptException when the NIO code that is used to read in the original class
bytecodes.
```

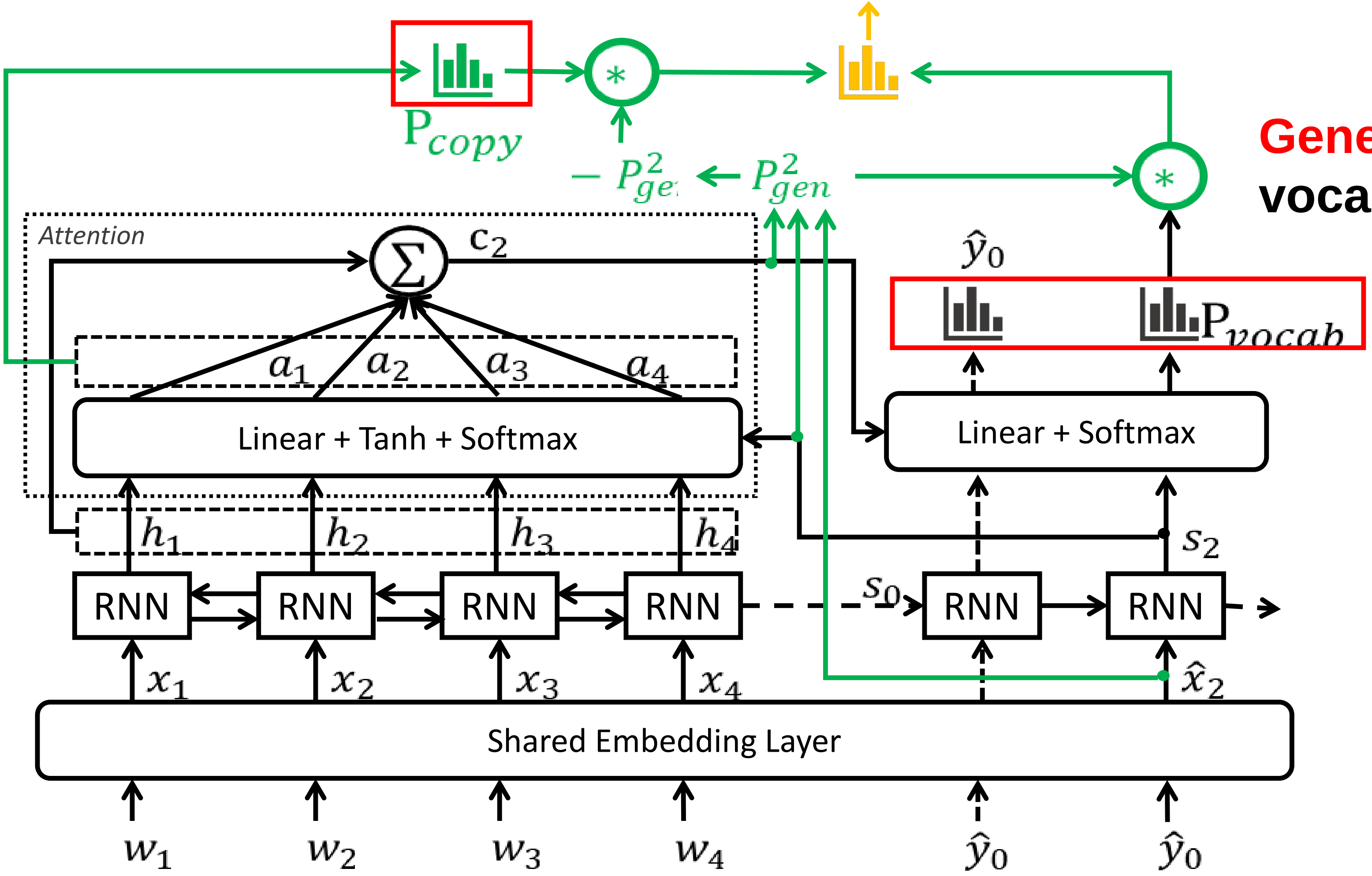
A Commit Msg

OOV words can usually be found in the input!

Solution: Pointer Generator

Copy from input

Generate from vocabulary



$$P(y_j | y_0, \ldots y_{j-1}, w) = p_{gen}^j P_{vocab}(y_j) + \left(1 - p_{gen}^j\right) P_{copy}(y_j)$$

Challenge: Gap between ML Loss and Human Eval

- The Maximum-likelihood (ML) Loss

$$loss_{ml} = -\frac{1}{|\boldsymbol{y}|}\sum_{j=1}^{|\boldsymbol{y}|}\log p(y_j|\hat{y}_0,\ldots,\hat{y}_{j-1},\boldsymbol{w})$$

- Gap between $loss_{ml}$ and Human Evaluation

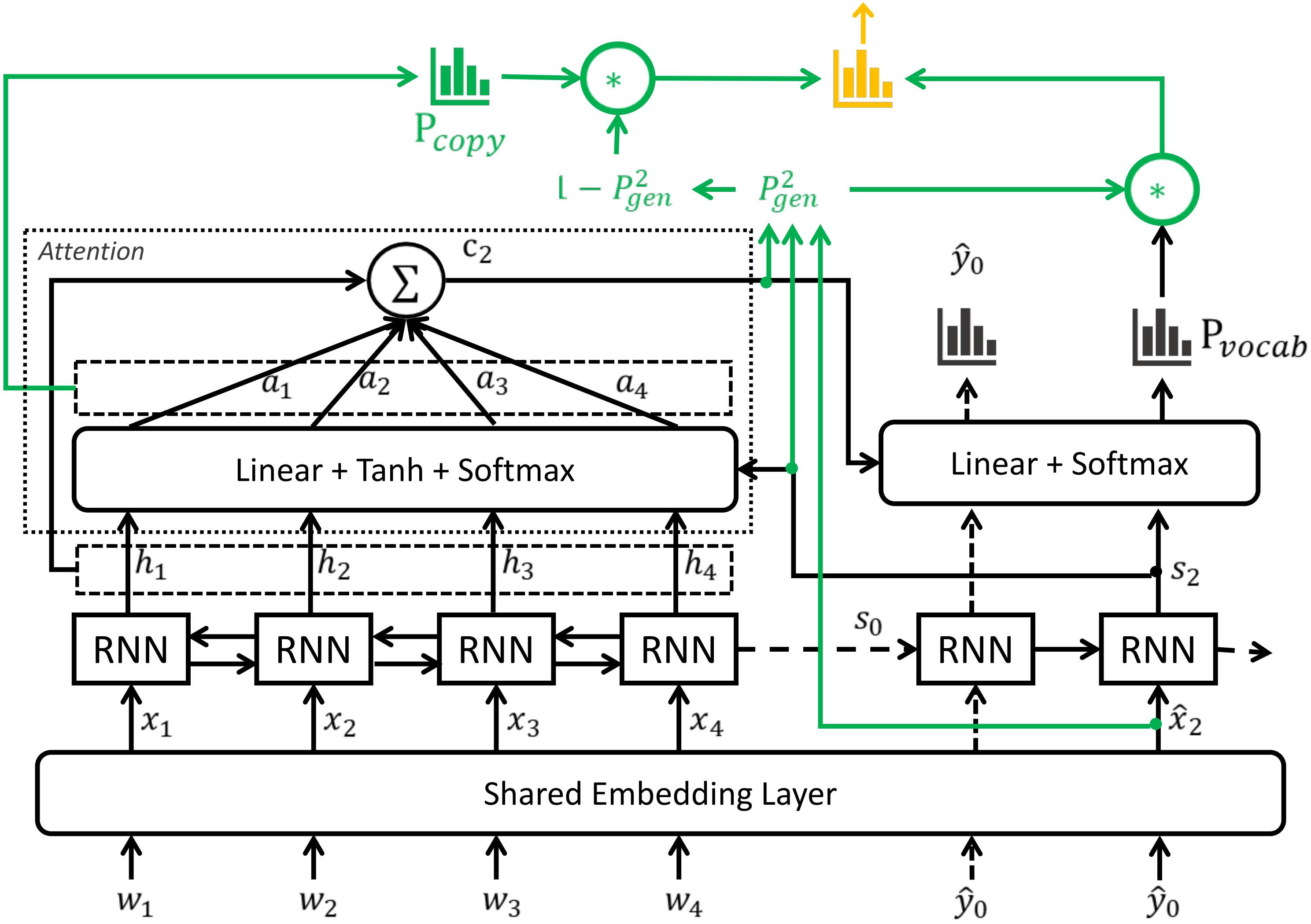
The cat sat on the mat **Human Ev**: the same meaning

On the mat sat the cat **ML Loss is high** e" matches

GAPS

- We adopt the self-critical sequence training (SCST) and a special loss function named RL loss ($loss_{rl}$).

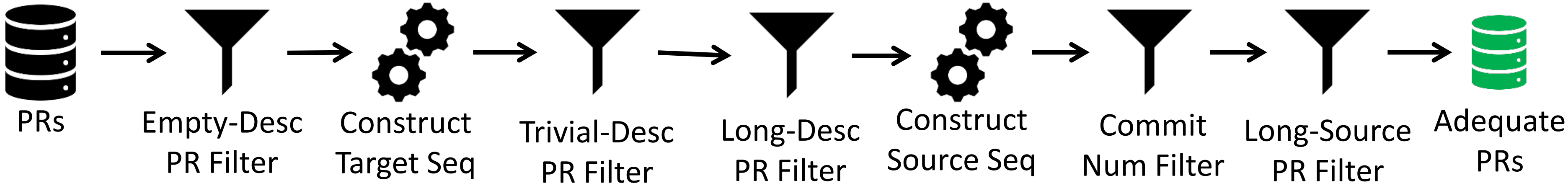
Overview of Our Approach



$$loss = \gamma loss_{ml} + (1 - \gamma) loss_{rl}$$

Dataset

- Collect 333K merged PRs from the top 1K Java projects on GitHub


PRs → Empty-Desc PR Filter → Construct Target Seq → Trivial-Desc PR Filter → Long-Desc PR Filter → Construct Source Seq → Commit Num Filter → Long-Source PR Filter → Adequate PRs

- Obtain 41.8K adequate PRs
  - Train, Valid, Test: 80%, 10%, 10%

Evaluation

- Evaluation metric: ROUGE
  - ROUGE-N (N=1,2) and ROUGE-L
  - Widely used to evaluate text summarization systems
- Baselines
  - LeadCM: take the first 25 tokens of the commit message paragraph as output
    - 25: median length of the PRs in our dataset
  - LexRank: sort input according to relative sentence importance, take the first 25 tokens

The Effectiveness of Our Approach

Approach	**Avg. Length**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
LexRank	24.21	24.11	11.40	22.42
LeadCM	24.37	30.61	17.85	28.89
Attn+PG+RL	19.21	**34.15**	**22.38**	**32.41**
Attn+PG+RL vs. LexRank	-5.00	**+41.65%**	**+96.33%**	**+44.52%**
Attn+PG+RL vs. LeadCM	-5.16	**+11.57%**	**+25.40%**	**12.18%**

☛ **Our approach outperforms the two baselines in terms of ROUGE-1,2 and L.**

The Effects of Main Components

Approach	Avg. Length	ROUGE-1	ROUGE-2	ROUGE-L
Attn	13.95	22.92	12.74	21.95
Attn+PG	14.02	31.27	21.15	29.82
Attn+PG+RL	19.21	**34.15**	**22.38**	**32.41**
PG	+0.07	**+36.47%**	**+66.10%**	**+35.87%**
RL	-5.19	**+9.21%**	**+5.81%**	**8.68%**

☛ **Our approach outperforms Attn and Attn+PG.**

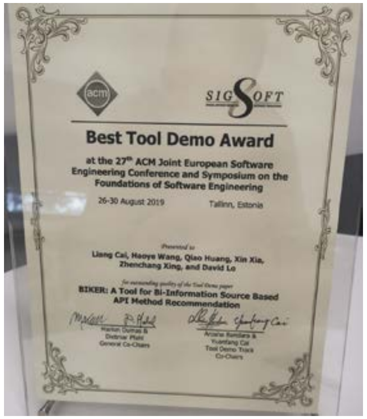
☛ **The pointer generator and the RL loss are effective and helpful for boosting the effectiveness of our approach.**

Example Research 2

API Method Recommendation without Worrying About the Task-API Knowledge Gap

ESEC/FSE 2019 Best Tool Demo Award

stackoverflow



ACM SIGSOFT

Best Tool Demo Award

at the 27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering

26-30 August 2019 Tallinn, Estonia

Presented to

Liang Cai, Haoye Wang, Qiao Huang, Xin Xia, Zhenchang Xing, and David Lo

for outstanding quality of the Tool Demo paper

BIKER: A Tool for Bi-Information Source Based API Method Recommendation

Marlon Dumas & Dietmar Pfahl
General Co-Chairs

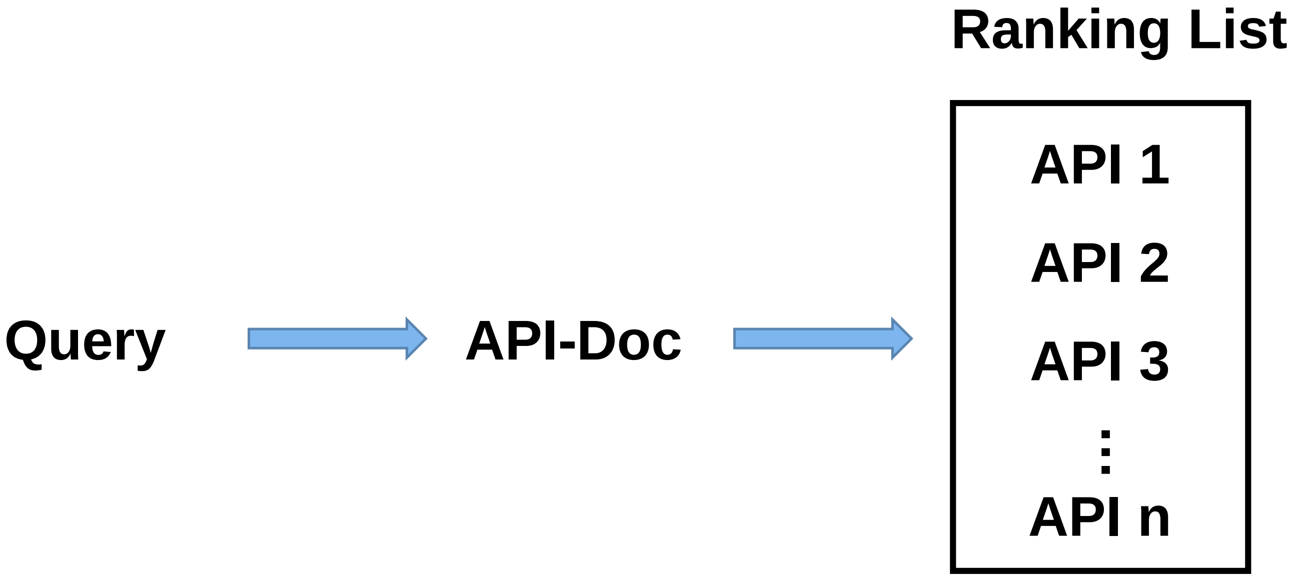
Arosha Bandara & Yuanfang Cai
Tool Demo Track Co-Chairs

Qiao, **Xia**, et al. ASE 2018, ESEC/FSE 2019 Tool

Background

- Too many APIs in a large library
  - Java SE 8 API, 4K classes, 31K methods
- A practical scenario
  - I have a programming task
  - I don’t even know which API is worth for investigation

A Straightforward Solution



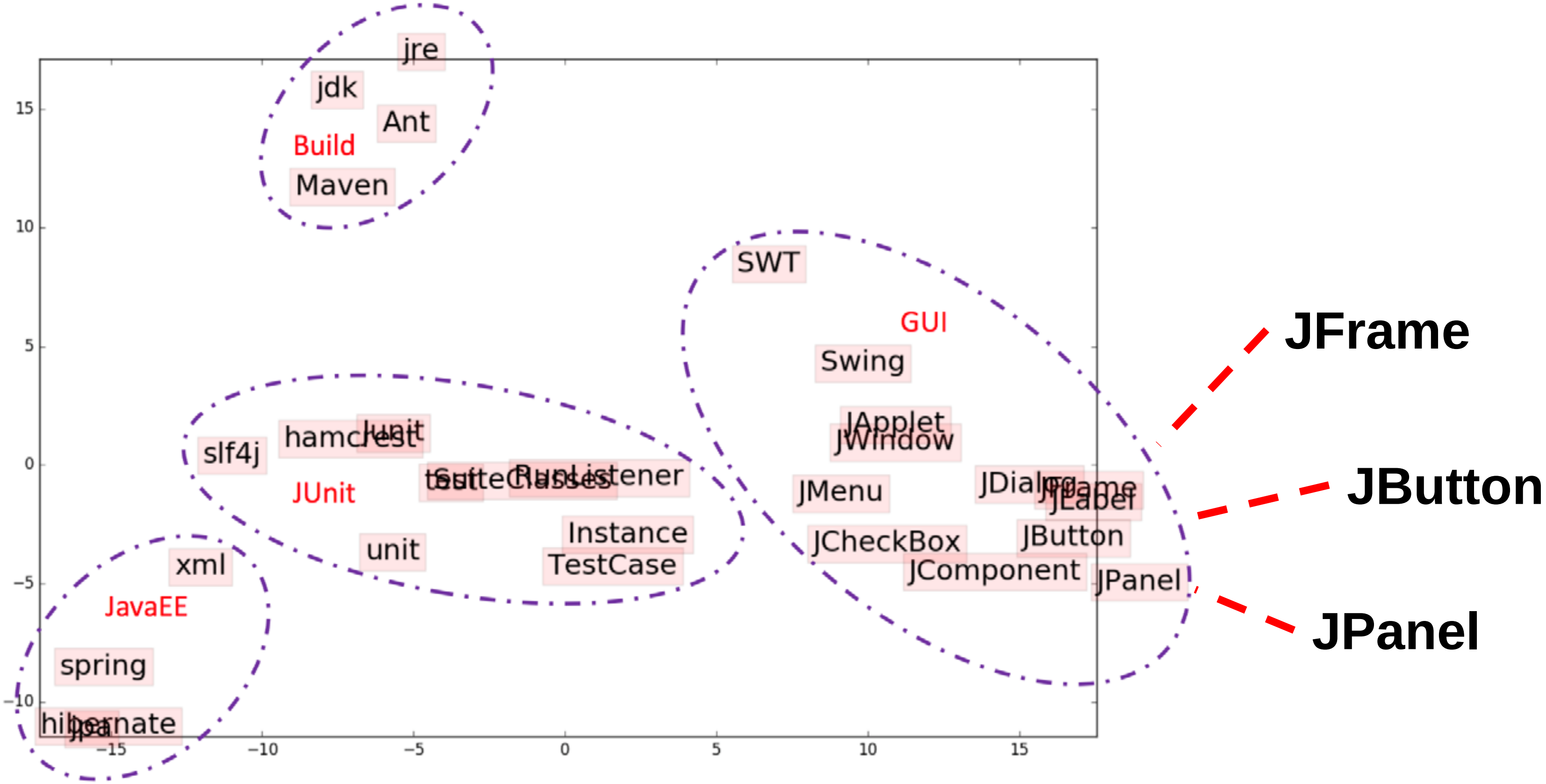
Query → API-Doc → Ranking List

- API 1
- API 2
- API 3
- ⋮
- API n

Challenge 1: Lexical Gap

- How to initialize all values in an array to false?
- Correct API: *Arrays.fill*
  - Assigns the specified boolean value to each element of the specified array of booleans.
- Lexical Gap:
  - Initialize <-> Assign
  - Boolean <-> False

Solution: Word Embedding



JFrame

JButton

JPanel

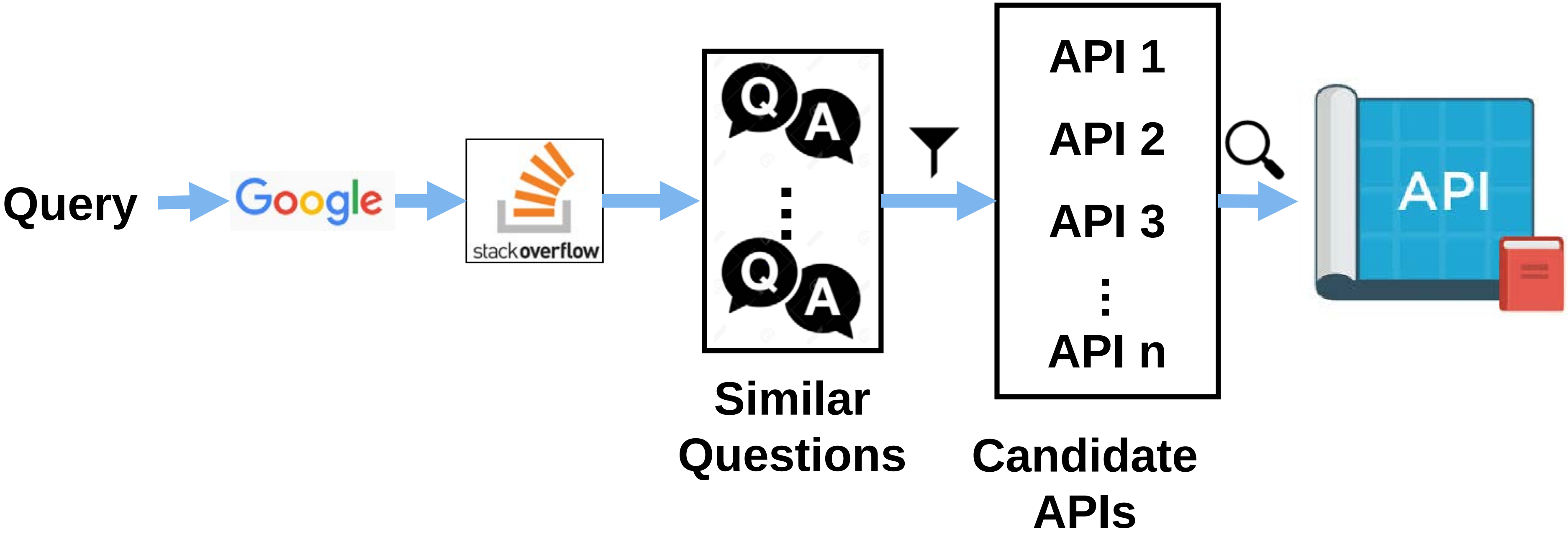
Challenge 2: Task-API Knowledge Gap

- **How to check whether a class exists?**
- **Wrong API: *org.omg.CORBA.Object.is_a* (score = 0.669)**
  - **class that**
    interface

API-Doc describes functionality and structure, but lacks concepts or purposes

- **Correct API: *java.lang.Class.forName* (score = 0.377)**
  - **Returns the Class object associated with the class with the given string name**

How Do Developers Search for APIs?



Query

Similar Questions

API 1

API 2

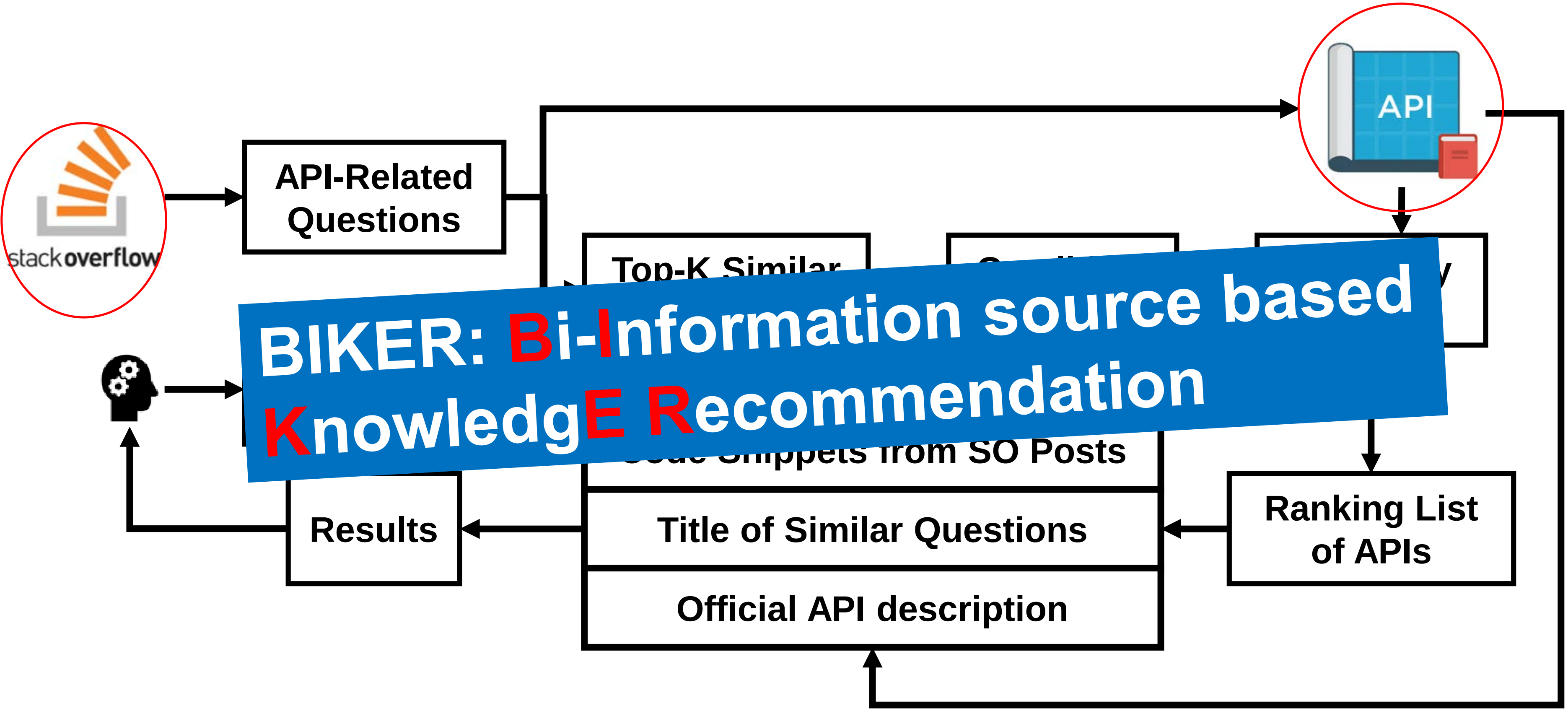
API 3

⋮

API n

Candidate APIs

Overall Framework of Our Approach



API-Related Questions

Top-K Similar

Code Snippets from SO Posts

Title of Similar Questions

Official API description

Ranking List of APIs

Results

BIKER: Bi-Information source based KnowledgE Recommendation

Similarity Scores between Query and API

- **SimSO: Based on StackOverflow posts**

$$SimSO(API, Q) = min(1, \frac{\sum_{i=1}^{n} sim(T_i, Q)}{n} \times \log_2 n)$$

Text sim between query and question title

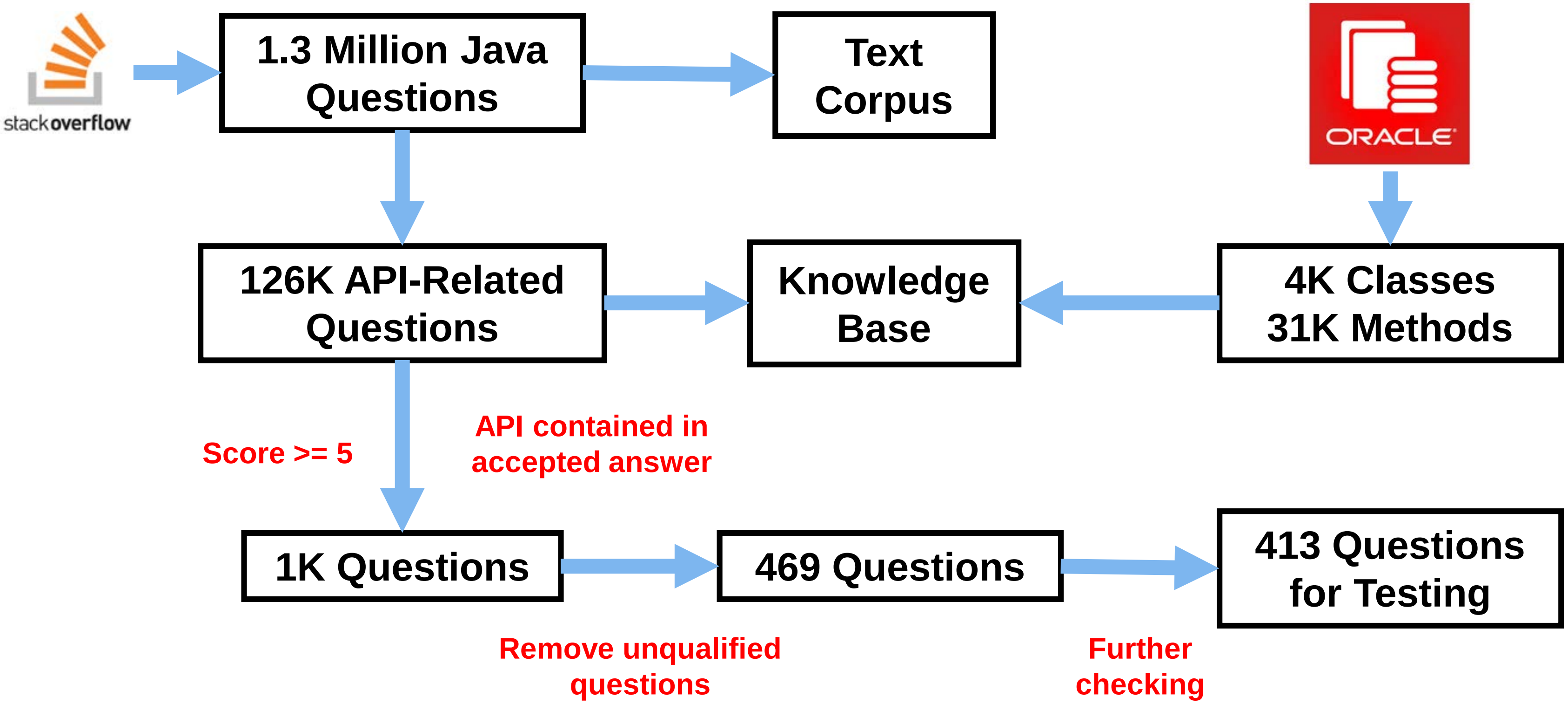
- **SimDoc: Based on API documentation**
  - **Calculating text sim between query and API description**
  - **The formula is based on Ye et al.’s work in ICSE 2016**

Ye, Xin, et al. "From word embeddings to document similarities for improved information retrieval in software engineering." ***Proceedings of the 38th international conference on software engineering*****. ACM, 2016.**

An Example of API Summary

- Query: *Run linux command in java code*
- Top-1 API: *java.lang.Runtime.exec*
- Doc: *Executes the specified string command in a separate process*
- *Similar Questions*
  - 1. *Run cmd commands through java*
  - 2. *use cmd commands in java program*
  - 3. *Unable to execute Unix command through Java code*
- *Code Snippets*
  - 1. *Process p = Runtime.getRuntime().exec(command);*
  - 2. *Runtime.exec(-whatever cmd command you need to execute-)*

Data Collection


stackoverflow
1.3 Million Java Questions
Text Corpus
ORACLE
126K API-Related Questions
Knowledge Base
4K Classes 31K Methods
Score >= 5
API contained in accepted answer
1K Questions
469 Questions
413 Questions for Testing
Remove unqualified questions
Further checking

Baselines

- RACK (Rahman et al., SANER 2016)
  - Using SO posts to build a keyword-API mapping database
  - Only support class-level
  - Published a dataset (150 questions from Java tutorial sites)
- DeepAPI (Gu et al., FSE 2016)
  - Based on deep neural network (seq2seq)
  - Training with annotated API sequences from code repositories.
  - Natural language query -> API sequence

RQ1: Effectiveness of Our Approach

- **Class-Level, compared with RACK and DeepAPI**
  - **Our dataset: MRR 0.69 (50%), MAP 0.66 (57%)**
  - **RACK’s dataset: MRR 0.43 (42%), MAP 0.27 (58%)**
- **Method-Level, compared with DeepAPI**
  - **Our dataset: MRR 0.57 (205%), MAP 0.52 (241%)**

RQ2: Effectiveness of Information Sources

Info Source	Class-Level		Method-Level	
	MRR	MAP	MRR	MAP
SO Posts	0.56	0.53	0.52	0.48
Java Doc	0.29	0.27	0.10	0.08
Both	0.69	0.66	0.57	0.52
Improve.SO	24%	25%	9%	9%
Improve.Doc	141%	149%	491%	559%

RQ3: Time Cost of Our Approach

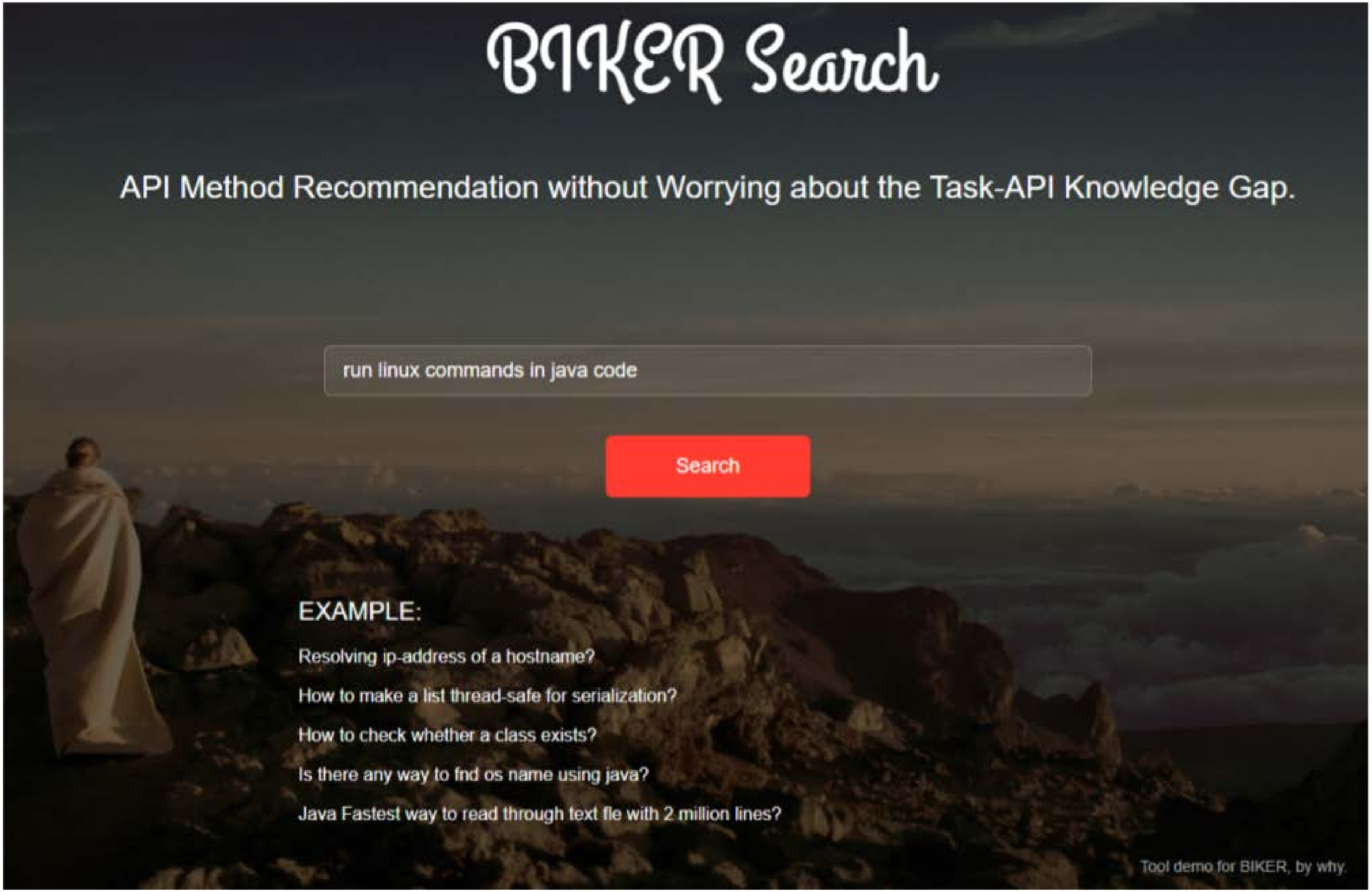
Approach	Model Training Time	Query Processing Time
Our Approach	36 minutes	2.8s / query
DeepAPI	240 hours	2.6s / query
RACK	unknown	12.8s / query

User Study

- 28 Java developers, 4 groups, 10 questions
  - WSO, DeepAPI, Ours-Simple, Ours-Full
- Evaluation Metrics
  - Correctness and Completion Time
- Results

Group	WSO	DeepAPI	Ours-Simple	Ours-Full
Correctness	0.79	0.87	0.86	0.97 (11%)
Time	84s	65s	60s	43s (28%)

Tool



BIKER Search

API Method Recommendation without Worrying about the Task-API Knowledge Gap.

run linux commands in java code

EXAMPLE:

Resolving ip-address of a hostname?

How to make a list thread-safe for serialization?

How to check whether a class exists?

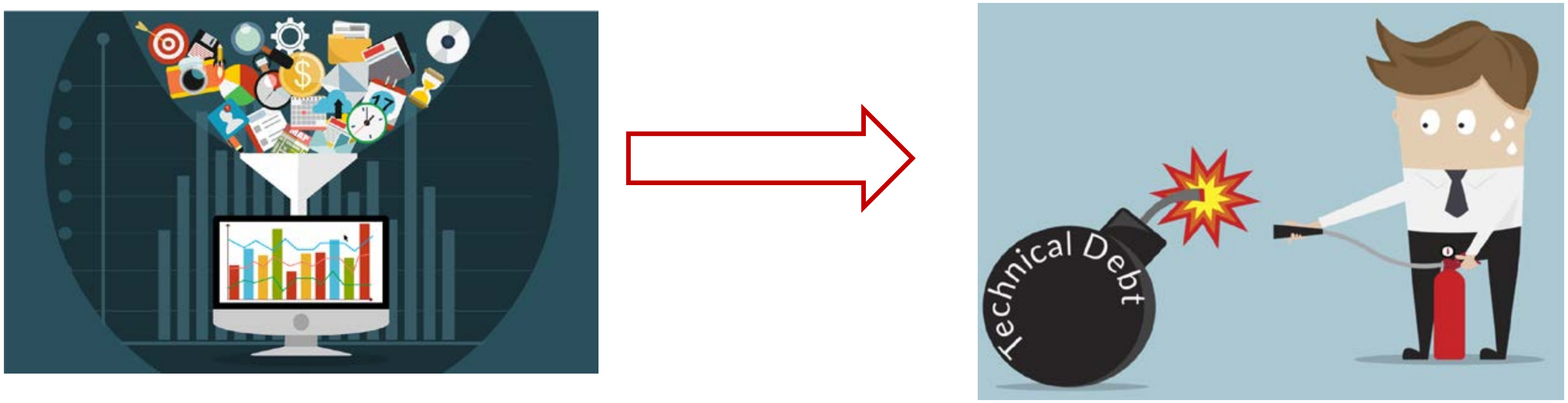
Is there any way to find os name using java?

Java Fastest way to read through text file with 2 million lines?

Tool demo for BIKER, by why.

Example Research 3

Automating Change-Level Self-Admitted Technical Debt Determination



Yan, **Xia**, et al. TSE 2019

Technical Debt (TD)

In order to achieve short-term goals, suboptimal solutions are introduced in a software. This increases effort to maintain the software in long-term.

Metaphor

↔

Technical Debt

Financial Debt

Examples of indicating TD

It’s ok for now but we’ll refactor it later!

Don’t worry about the documentation for now!

Todo/Fixme: this should be fixed before release.

Impact of TD

😉 TD Benefits

- Higher productivity
- Lower cost

⚠ **Current release!**

😟 TD Costs

- Introducing risk
  (debt out of control)
- Increasing interest
  (Effort to pay back)

⚠ **Don’t underestimate the danger!**

How to identify TD in a cost-effective way?

How to identify TD?

Method 1: Identifying TD through source code metrics or code smells

God class

A class knows too much or does too much!

Lines Of Code
Number of Methods
Complexity
Cohesion and Coupling

http://www.jdeodorant.org

Code rules

Rules which enforce a specific coding style.

Dollar Signs when naming;
Boolean Get Method Name;
At Least One Constructor;

DON'T SHOOT THE MESSENGER

https://pmd.github.io/

False positive rate

Heavy code analysis tasks

How to identify TD?

// Is There a Correct Way to Comment Your Code?

Method 2: **Identifying TD through source code comments**

Examples: (in Tomcat project)

//FIXME handle EVT_GET_ALL_SESSIONS later

//TODO: Need some format checking here!!!

→

SATD:

Self-**A**dmitted **T**echnical **D**ebt

😉

More reliable

More lightweight

Identify TD from code comments

Manual summarization

62 Patterns: todo; fixme; temporary crutch; this isn’t very solid; fix this crap; remove me before production

[Shihab et al. ICSME 2014]

Automatically identification

NLP; Classification;

[Shihab et al. TSE 2017]

Ensemble learning;

[Huang et al. EMSE 2017]

Issues remaining unanswered

However, all of the current identification methods are file-level

Issues remaining unanswered:

TD-introducing change?

Characteristics of TD-introducing changes?

How/Why the TD is introduced?

File-level detection cannot describe TD-introducing context.

(e.g., TD related to multiple files)

Our motivation

Can we identify TD at change-level? i.e.,

Can we determine whether a change introduces TD?

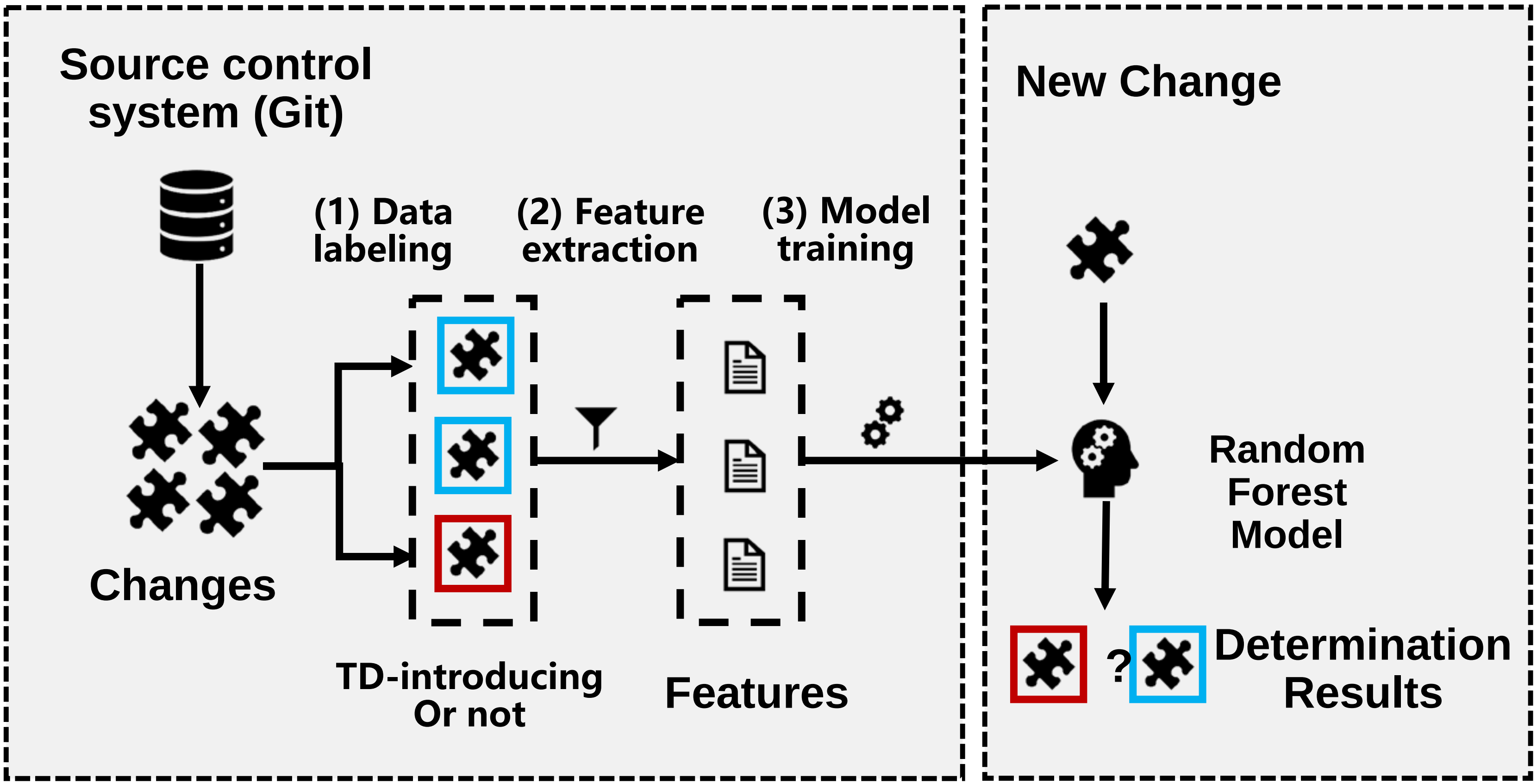
☺ **Benefits:**

Characterizing TD-introducing change.

Understand the TD-introducing context.

Identify TD just-in-time.

Overview of our approach



Source control system (Git)

(1) Data labeling

(2) Feature extraction

(3) Model training

Changes

TD-introducing Or not

Features

New Change

Random Forest Model

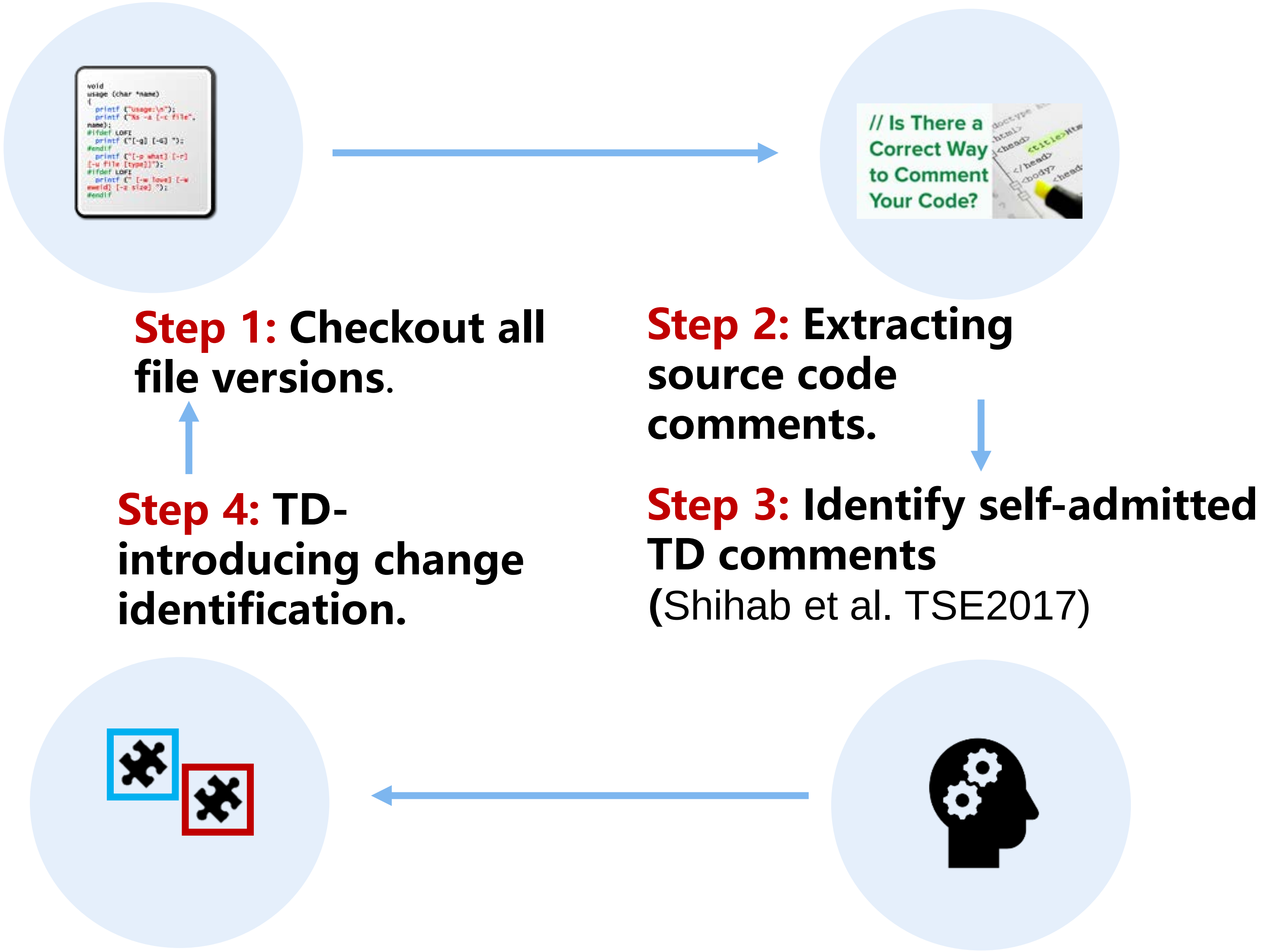
?

Determination Results

Model Building Phase

Model application Phase

(1) Data labeling



Step 1: **Checkout all file versions.**

Step 2: **Extracting source code comments.**

Step 3: **Identify self-admitted TD comments** (Shihab et al. TSE2017)

Step 4: **TD-introducing change identification.**

(2) Feature extraction

Three dimensions with 25 features:

Diffusion:
Capture the distribution of the change
e.g., Size, #Directories, #Files

History:
Capture the historical information
e.g., NDEV, EXP, NUC

Message:
Capture the commit log
e.g., activity type (bug, feature)

[Kamei et al. TSE13]

(3) Experimental setup

Dataset:
7 Open source Java,
100,011 Changes
2.7% TD-introducing

Measures:
AUC
Cost-effectiveness

Classifier:
Random-forest

Validation:
10*10 stratified
cross-validation

Cost-effectiveness: Recall of TD-introducing changes when using 20% of the entire effort required to inspect all changes to inspect the top ranked changes.

Research questions

RQ1: Can we effectively determine the changes that introduce TD?

RQ2: Which dimension of features are most important in determining TD-introducing changes?

RQ1: Baselines

Random guess (RG)

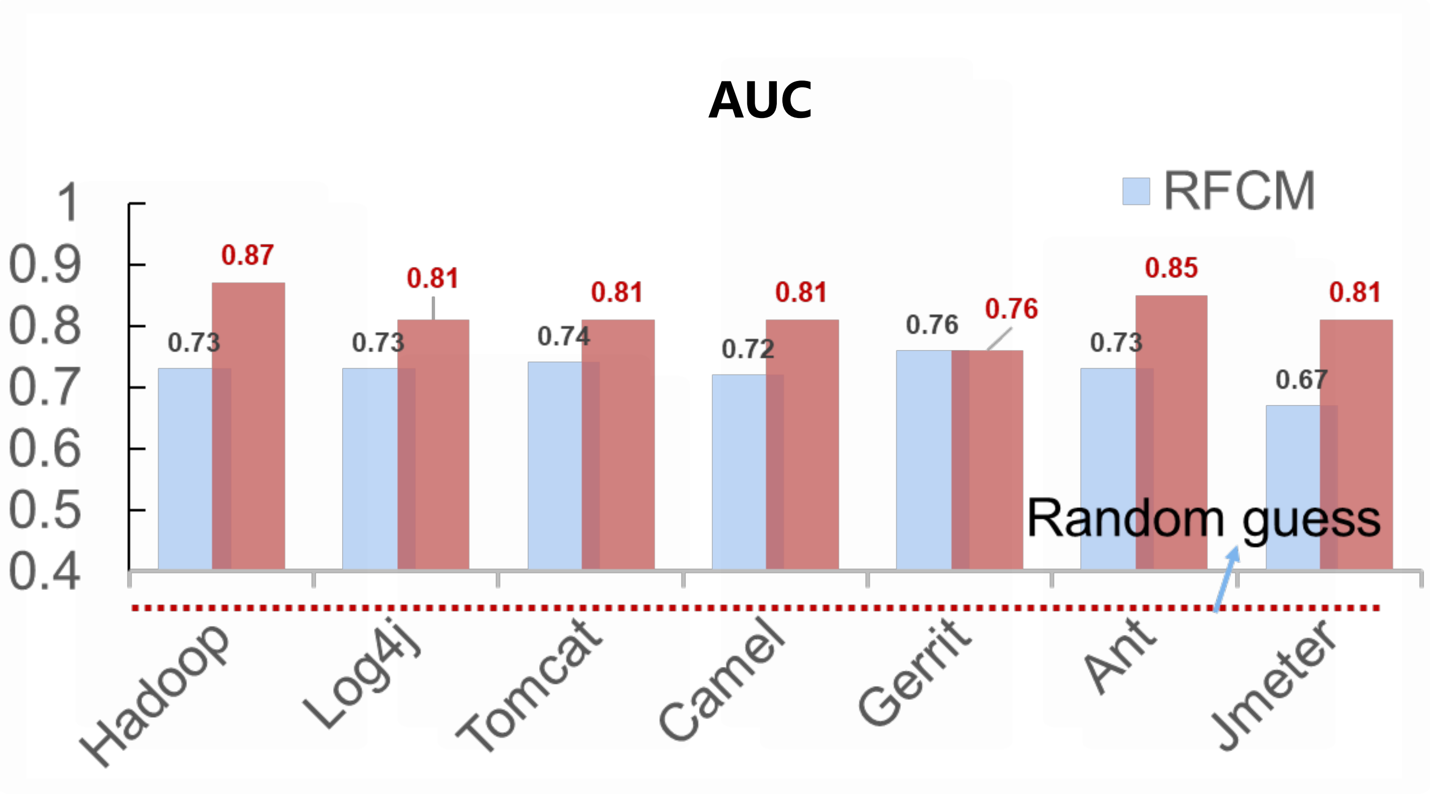
Random determination 10 times to get the average performance.

Text classifiers based on change message

Naive Bayes, Naive Bayes Multinomial and Random Forest

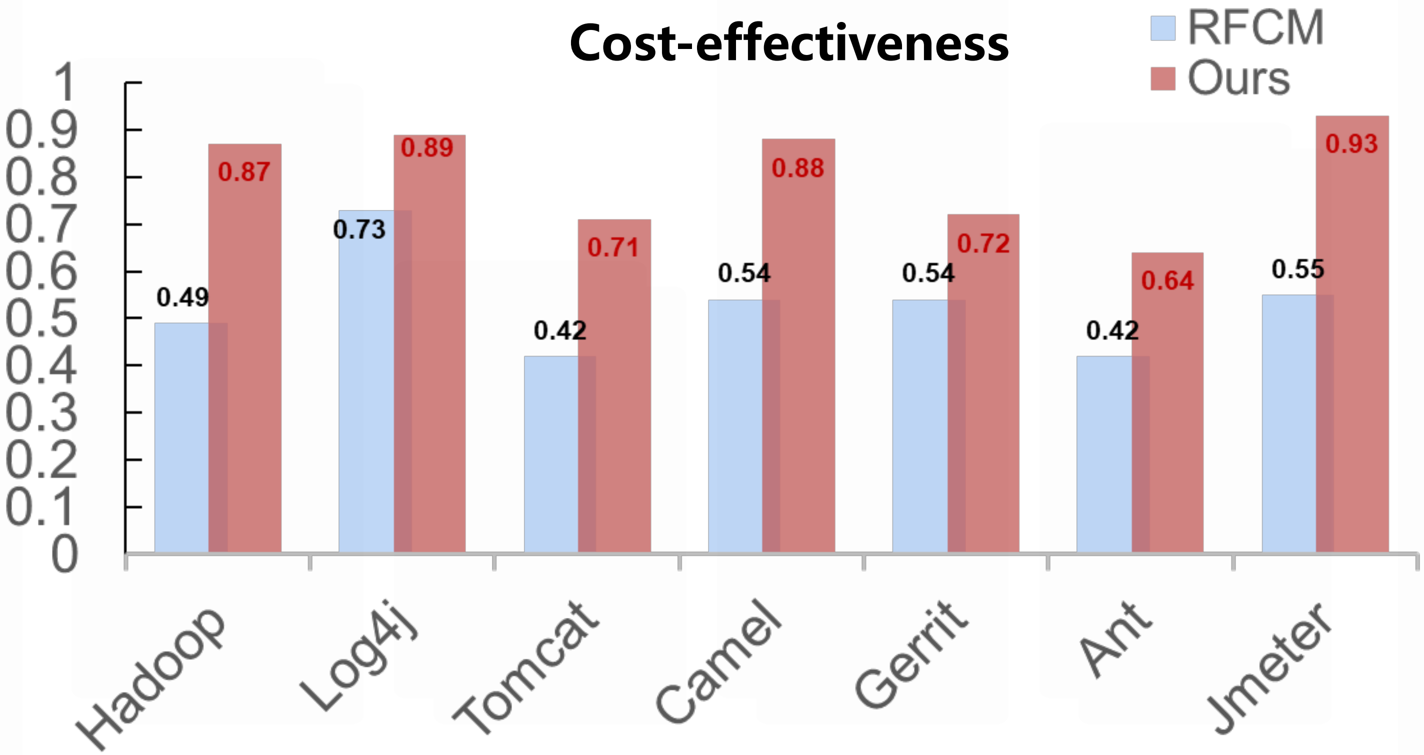
Four baselines: RG, NBCM, NBMCM, RFCM

RQ1: Performance of AUC



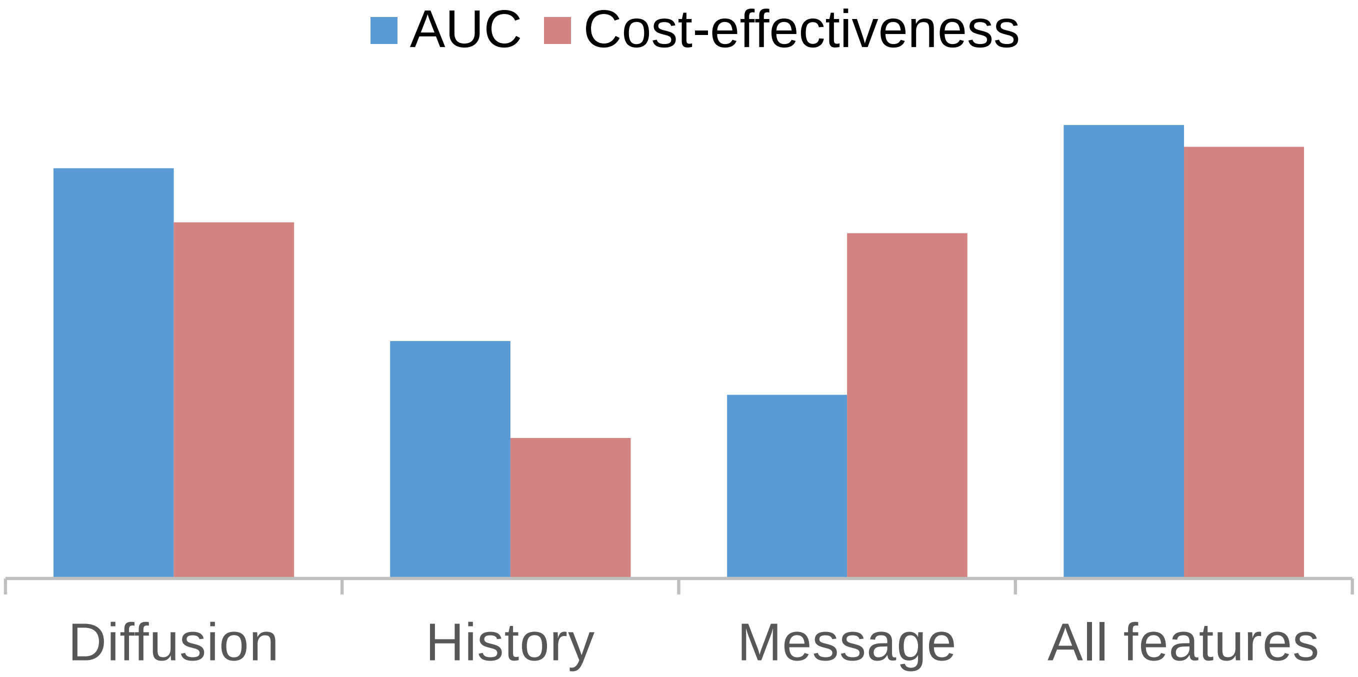
On average, our model improves four baselines by a substantial margin, with a statistical significance and large effect size in most cases.

RQ1: Performance of Cost-effectiveness



On average, our model improves four baselines by a substantial margin, with a statistical significance and large effect size in all cases.

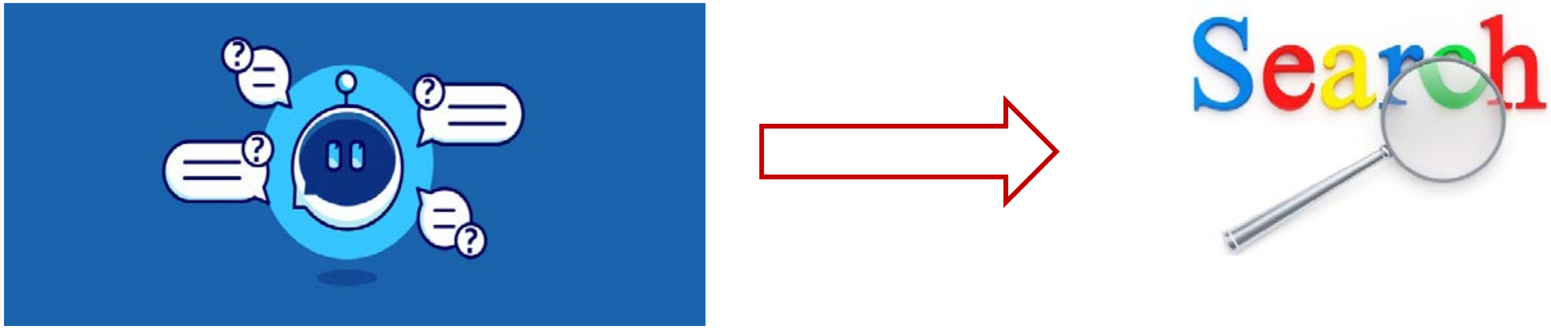
RQ2: Performance of dimensions



Diffusion is the most discriminative dimension.
Using all dimensions of features is better.

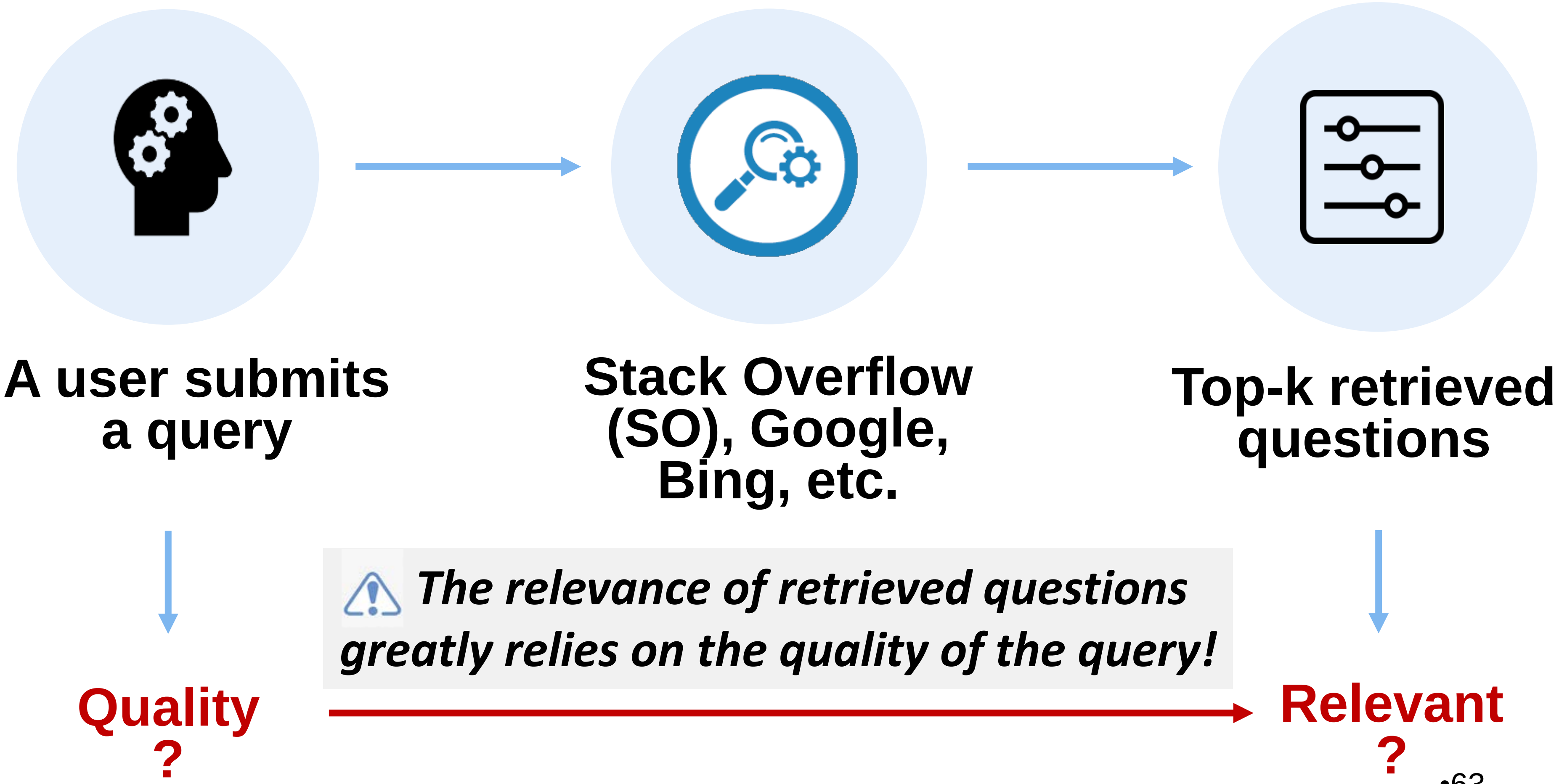
Example Research 4

Chatbot4QR: Interactive Query Refinement for Technical Question Retrieval



Zhang, Qiao, **Xia**, et al. TSE 2020

Background: Question Retrieval (QR)



A user submits a query

Stack Overflow (SO), Google, Bing, etc.

Top-k retrieved questions

⚠ ***The relevance of retrieved questions greatly relies on the quality of the query!***

Quality ?

Relevant ?

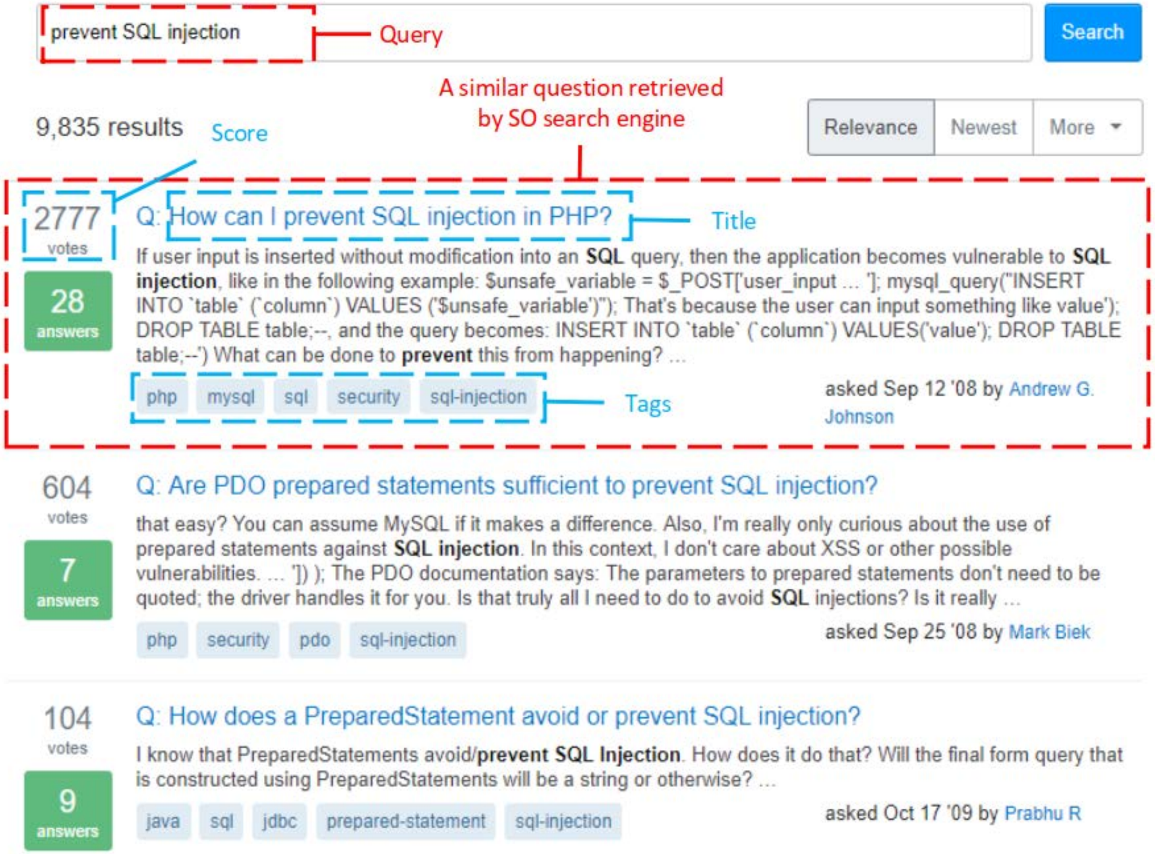
Two Challenging Issues in QR

It is not always an easy task for users to formulate a good query. – *[SANER’15, TSC’16, ASE’17, MSR’18]*

[MSR’ 18]: it is **common** for users to **miss some important technical keywords in queries** when performing code search on Google.

Users may probably have different preferred questions for a query, depending on their personalized technical background or contexts.

Motivating Example



prevent SQL injection

9,835 results

2777 votes 28 answers

Q: How can I prevent SQL injection in PHP?

If user input is inserted without modification into an **SQL** query, then the application becomes vulnerable to **SQL injection**, like in the following example: $unsafe_variable = $_POST['user_input ... ']; mysql_query("INSERT INTO `table` (`column`) VALUES ('$unsafe_variable')"); That's because the user can input something like value'); DROP TABLE table;--, and the query becomes: INSERT INTO `table` (`column`) VALUES('value'); DROP TABLE table;--') What can be done to **prevent** this from happening? ...

php mysql sql security sql-injection

asked Sep 12 '08 by Andrew G. Johnson

604 votes 7 answers

Q: Are PDO prepared statements sufficient to prevent SQL injection?

that easy? You can assume MySQL if it makes a difference. Also, I'm really only curious about the use of prepared statements against **SQL injection**. In this context, I don't care about XSS or other possible vulnerabilities. ... '])); The PDO documentation says: The parameters to prepared statements don't need to be quoted; the driver handles it for you. Is that truly all I need to do to avoid **SQL** injections? Is it really ...

php security pdo sql-injection

asked Sep 25 '08 by Mark Biek

104 votes 9 answers

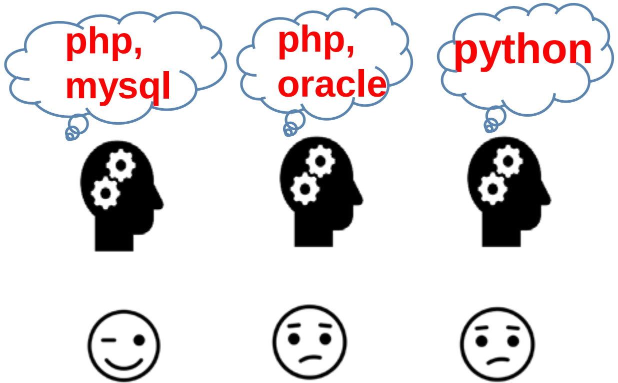
Q: How does a PreparedStatement avoid or prevent SQL injection?

I know that PreparedStatements avoid/**prevent SQL Injection**. How does it do that? Will the final form query that is constructed using PreparedStatements will be a string or otherwise? ...

java sql jdbc prepared-statement sql-injection

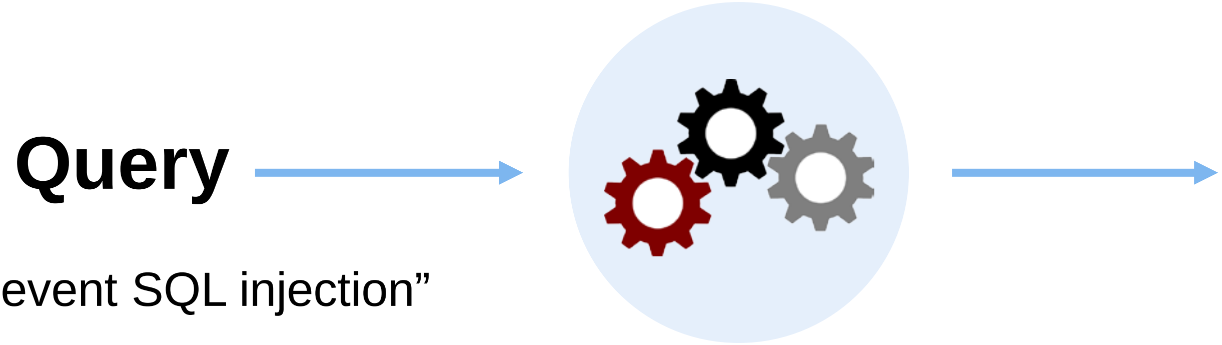
asked Oct 17 '09 by Prabhu R

Are the retrieved questions desired by all users?



Key Ideas of Our Chatbot4QR

KI-1: automatically detect the missing technical context in a query.



Query

“prevent SQL injection”

Detected missing types of technical context:

Type1: a **programming language**, e.g., php, python, etc.

Type2: a **database**, e.g., mysql, oracle, etc.

Type3: …

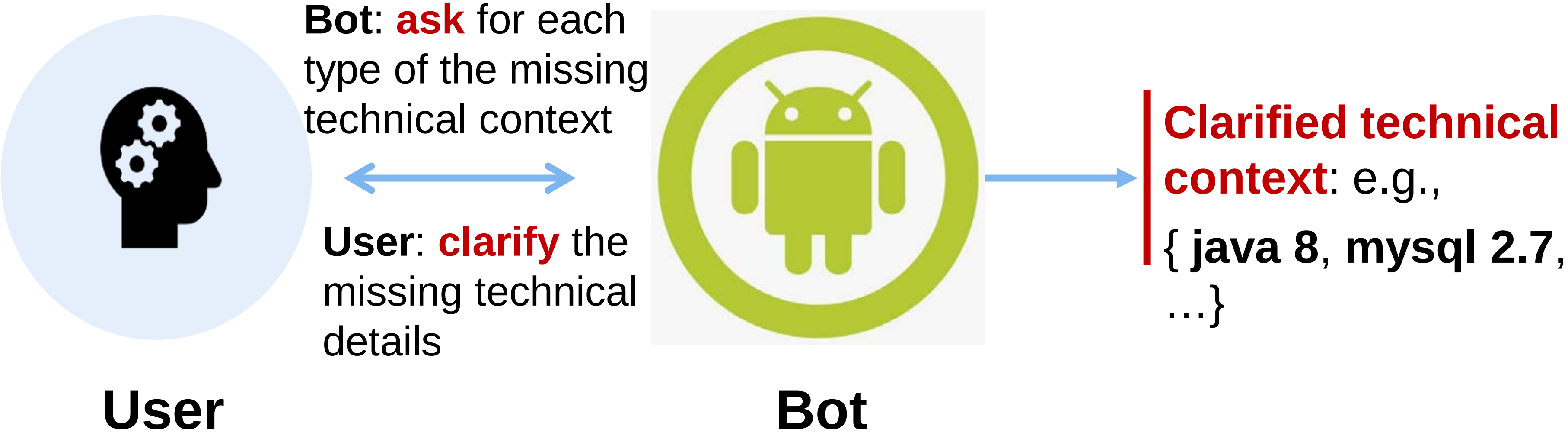
Need to detect:

1. What kinds of technical details are likely to be missed in the query?

2. What are the most relevant techniques of each missing type?

Key Ideas of Our Chatbot4QR

KI-2: interactively assist users in refining the query based on the detected missing technical context using a bot.



Bot: **ask** for each type of the missing technical context

User: **clarify** the missing technical details

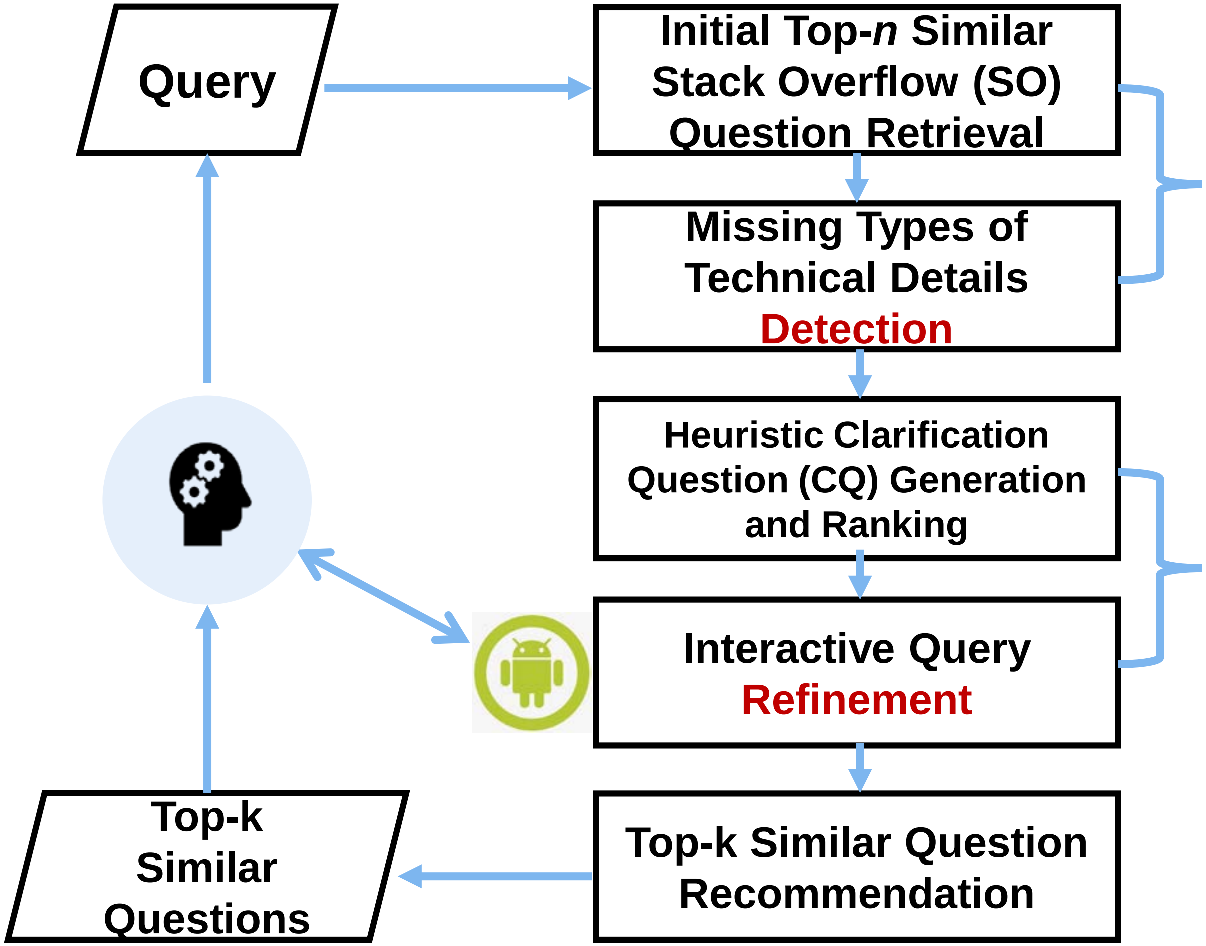
Clarified technical context: e.g.,

{ **java 8**, **mysql 2.7**, …}

User

Bot

Chatbot4QR: Approach Overview



Query

Initial Top-*n* Similar Stack Overflow (SO) Question Retrieval

Missing Types of Technical Details Detection

Heuristic Clarification Question (CQ) Generation and Ranking

Interactive Query Refinement

Top-k Similar Question Recommendation

Top-k Similar Questions

Two steps for the key idea "KI-1"

Two steps for the key idea "KI-2"

(1) Initial Top-n Similar Question Retrieval

Lucene:

- 😉 **Efficient**
- 😟 **Cannot bridge the lexical gaps**

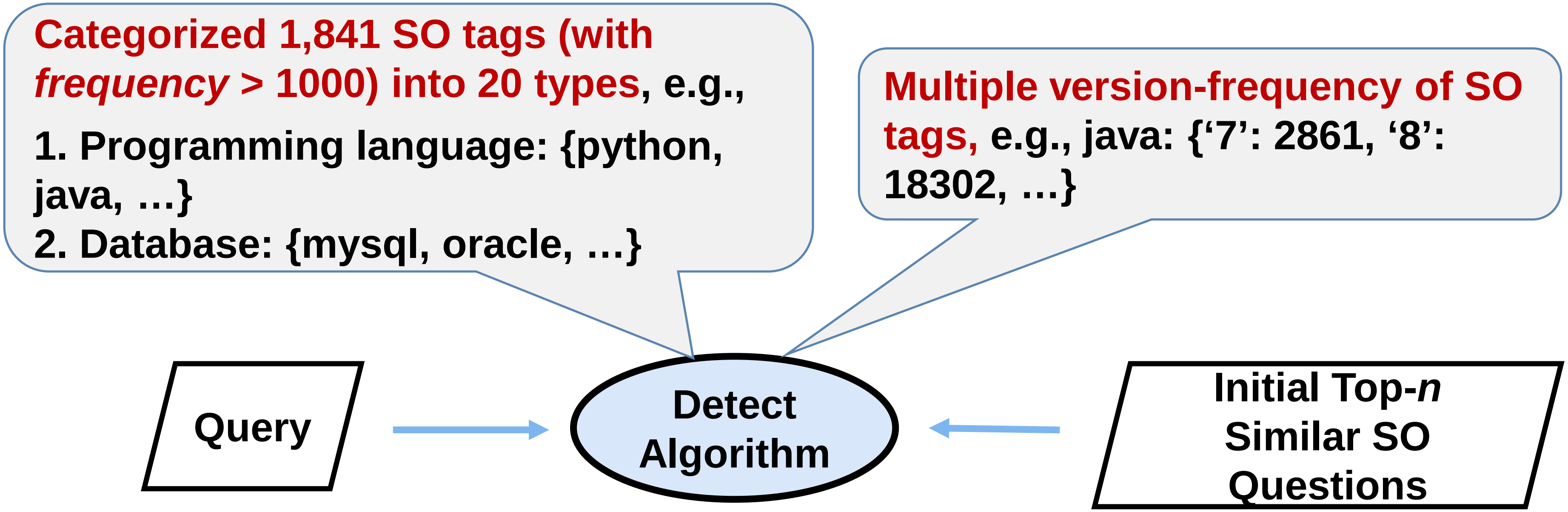
Word Embedding:

- 😉 **Can bridge the lexical gaps**
- 😟 **Inefficient to deal with large-scale data**

A Two-Phase Similar Question Retrieval Method:

1)Retrieve the **top-*N*** (e.g., N=10,000) similar SO questions using **Lucene**
2)Retrieve the **top-*n*** (e.g., k=15 << N) similar SO questions using a **Word Embedding** method (*ASE'18*)

(2) Missing Types of Technical Details Detection



Categorized 1,841 SO tags (with *frequency* > 1000) into 20 types, e.g.,

1. Programming language: {python, java, …}
2. Database: {mysql, oracle, …}

Multiple version-frequency of SO tags, e.g., java: {‘7’: 2861, ‘8’: 18302, …}

Query

Detect Algorithm

Initial Top-*n* Similar SO Questions

Identify the **types of technical details** that are **not specified in the query** but **appear in the initial top-*n* similar SO questions**

Detection Example

Query: “prevent SQL injection”

Top 3 Similar SO Questions:

Title: How can I prevent SQL injection in PHP?
Tags: *php, mysql, sql, security, sql-injection*

Title: Are PDO prepared statements to prevent SQL injection?
Tags: *php, security, pdo, sql-injection*

Title: How does a PreparedStatement avoid or prevent SQL injection?
Tags: *java, sql, jdbc, prepared-statement, sql-injection*

Detect Algorithm

Detected types of missing technical details in the

Type	Relevant SO Tags
Programming Language	{ php: [‘7’, ‘5.3’] java: [‘8’, ‘7’] }
Database	{ mysql: [‘2’, ‘5.7’] }
Framework	{ .net: [‘4.0’, ‘3.5’] }
Library	{ jdbc: [] }
Class	{ pdo: [] }

(3) Heuristic Clarification Question (CQ) Generation & Ranking

Three heuristic rules for generating a set of CQs that ask for three kinds of missing technical details.

Rule 1: Generate a **version-related CQ** that asks for a specific version of a technique.

Rule 2: Generate a **selection-related CQ** that asks for a specific technique from a candidate set of relevant techniques.

Rule 3: Generate a **confirmation-related CQ** that asks for whether using a specific technique.

CQ Generation Example

Detected types of missing technical details in the

Type	Relevant SO Tags
Programming Language	{ php: ['7', '5.3'] java: ['8', '7'] }
Database	{ mysql: ['2', '5.7'] }
Framework	{ .net: ['4.0', '3.5'] }
Library	{ jdbc: [] }
Class	{ pdo: [] }

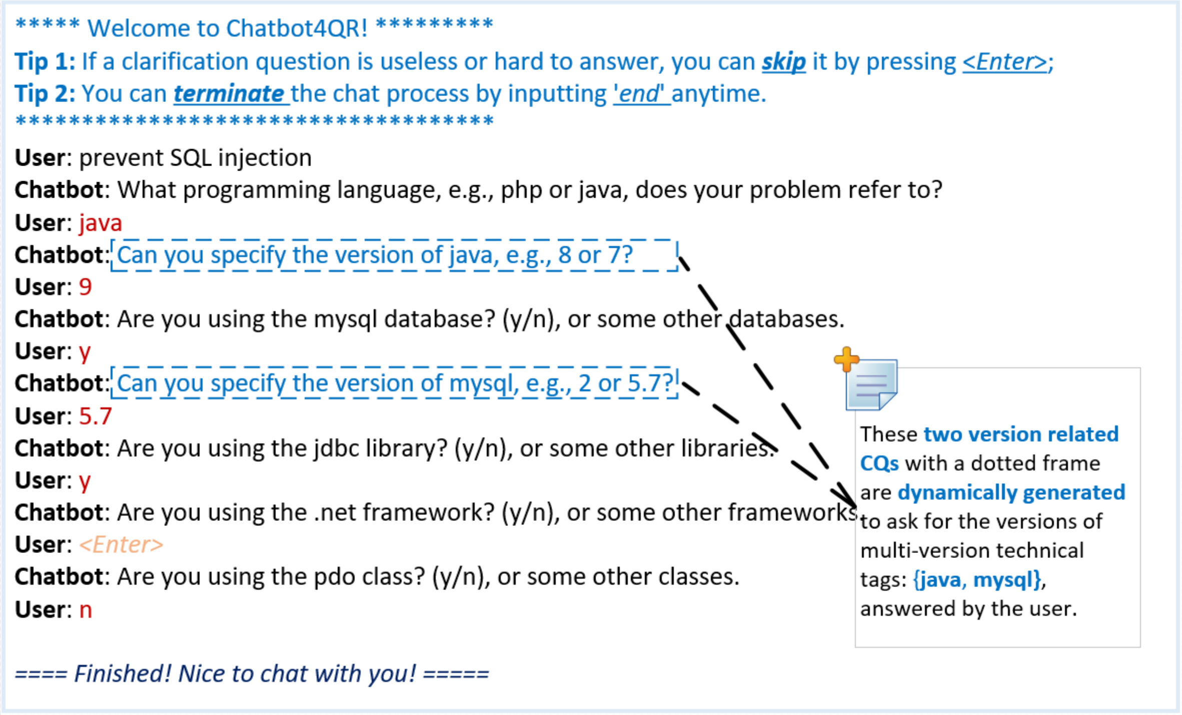
Initially generated CQs:

Rank	CQ
1	What programming language, e.g., php or java, does your problem refer to?
2	Are you using the mysql database? (y/n), or some other databases.
3	Are you using the jdbc library? (y/n), or some other libraries.
4	Are you using the .net framework? (y/n), or some other frameworks
5	Are you using the pdo class? (y/n), or some other classes.

CQs are ranked by the sum of similarities of the questions that contain any SO tags of the involved *type*.

(4) Interactive Query Refinement

Interact with the user by asking each generated CQ, and gather the user's feedback to the CQs.



***** Welcome to Chatbot4QR! *********
Tip 1: If a clarification question is useless or hard to answer, you can ***skip*** it by pressing *<Enter>*;
Tip 2: You can ***terminate*** the chat process by inputting *'end'* anytime.
****

User: prevent SQL injection
Chatbot: What programming language, e.g., php or java, does your problem refer to?
User: java
Chatbot: Can you specify the version of java, e.g., 8 or 7?
User: 9
Chatbot: Are you using the mysql database? (y/n), or some other databases.
User: y
Chatbot: Can you specify the version of mysql, e.g., 2 or 5.7?
User: 5.7
Chatbot: Are you using the jdbc library? (y/n), or some other libraries.
User: y
Chatbot: Are you using the .net framework? (y/n), or some other frameworks.
User: *<Enter>*
Chatbot: Are you using the pdo class? (y/n), or some other classes.
User: n

==== Finished! Nice to chat with you! =====

These **two version related CQs** with a dotted frame are **dynamically generated** to ask for the versions of multi-version technical tags: **{java, mysql}**, answered by the user.

Two kinds of the user's feedback to CQs:

1. **Positive feedback (*pfb*)**: { java 8, mysql 5.7, jdbc }
2. **Negative feedback (*nfb*)**: { pdo }

(5) Top-k Similar Question Recommendation

Recommend the top-k similar questions by **leveraging the user’s feedback to CQs** to adjust the semantic similarities of the top-N questions.

a weight coefficient of the feedback

$$sim(q,Q) = sim(q,Q) \times (1 + \eta \times (\textstyle\sum_{\underline{e \in pfb(q)}} md(e,Q) - \textstyle\sum_{\underline{e \in nfb(q)}} md(e,Q))) \qquad (1)$$

Two kinds of user’s feedback to CQs

$$md(e,Q) = \begin{cases} 1.5, & \text{if } e.v \text{ exists and both} \\ & e.t \text{ and } e.v \text{ are matched by } Q \\ 1.0, & \text{if only } e.t \text{ is matched by } Q \\ 0, & \text{otherwise.} \end{cases} \qquad (2)$$

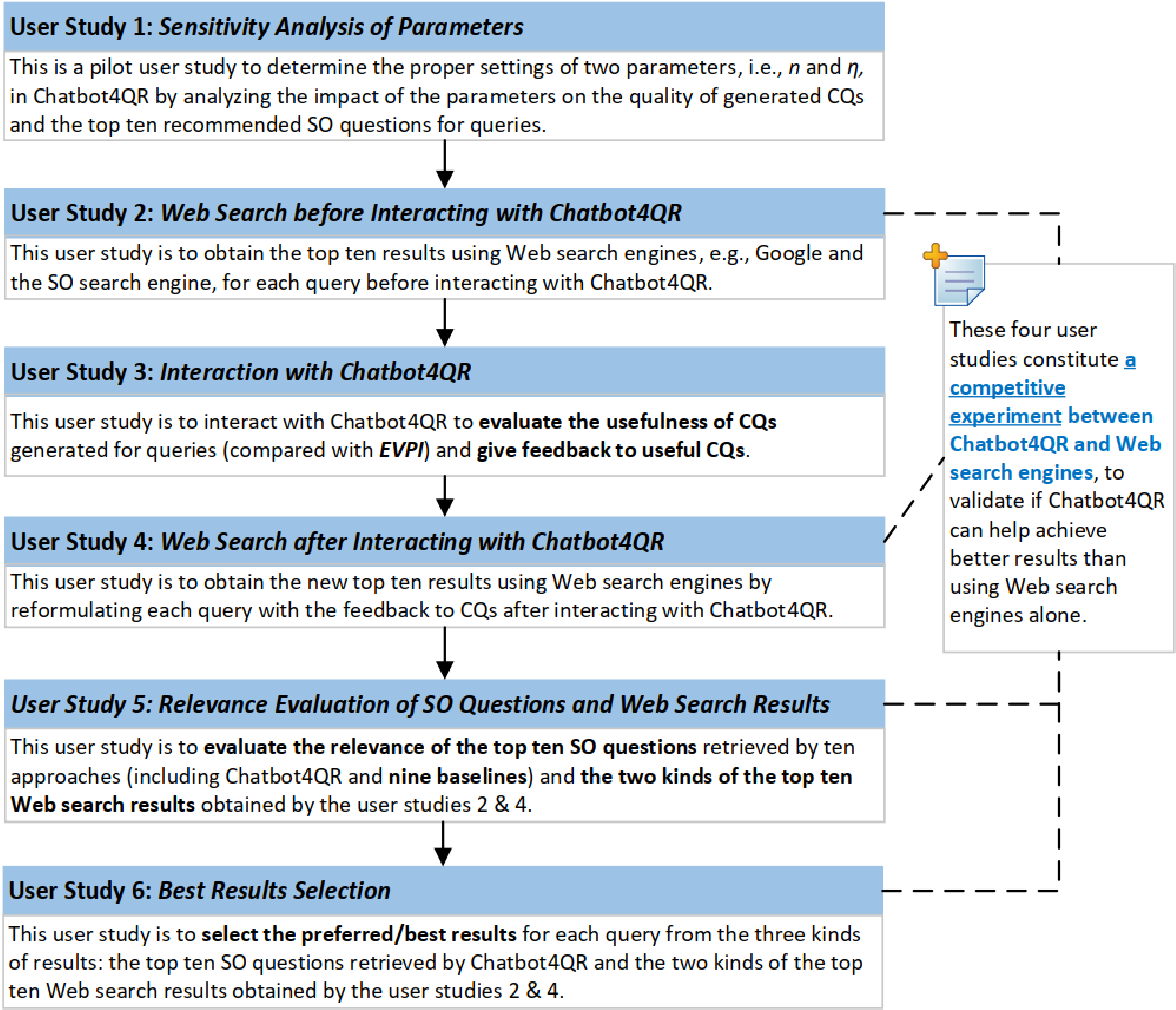
Experimental Setup

- **A repository of 188,0269 SO questions**
- **50 queries** built from the titles of SO questions outside the repository
- **25 participants**
- **6 user studies**

Whether a CQ can help the participants recognize some missing technical details in a query.

- **Metrics: *Usefulness of CQs, Pre@k, NDCG@k***

Flow of Our Six User Studies



User Study 1: ***Sensitivity Analysis of Parameters***

This is a pilot user study to determine the proper settings of two parameters, i.e., *n* and *η*, in Chatbot4QR by analyzing the impact of the parameters on the quality of generated CQs and the top ten recommended SO questions for queries.

User Study 2: ***Web Search before Interacting with Chatbot4QR***

This user study is to obtain the top ten results using Web search engines, e.g., Google and the SO search engine, for each query before interacting with Chatbot4QR.

User Study 3: ***Interaction with Chatbot4QR***

This user study is to interact with Chatbot4QR to **evaluate the usefulness of CQs** generated for queries (compared with ***EVPI***) and **give feedback to useful CQs**.

User Study 4: ***Web Search after Interacting with Chatbot4QR***

This user study is to obtain the new top ten results using Web search engines by reformulating each query with the feedback to CQs after interacting with Chatbot4QR.

User Study 5: Relevance Evaluation of SO Questions and Web Search Results

This user study is to **evaluate the relevance of the top ten SO questions** retrieved by ten approaches (including Chatbot4QR and **nine baselines**) and **the two kinds of the top ten Web search results** obtained by the user studies 2 & 4.

User Study 6: ***Best Results Selection***

This user study is to **select the preferred/best results** for each query from the three kinds of results: the top ten SO questions retrieved by Chatbot4QR and the two kinds of the top ten Web search results obtained by the user studies 2 & 4.

These four user studies constitute **a competitive experiment between Chatbot4QR and Web search engines**, to validate if Chatbot4QR can help achieve better results than using Web search engines alone.

Five Research Questions

RQ1: What are the **proper settings** of the **parameters** ***n* and *η*** in Chatbot4QR?

RQ2: How **effective** can Chatbot4QR generate CQs?

RQ3: Can Chatbot4QR retrieve more relevant SO questions than the **state-of-the-art question retrieval and query expansion approaches**?

RQ4: How **efficient** is Chatbot4QR?

RQ5: Can Chatbot4QR help obtain better results than using **Web search engines alone**?

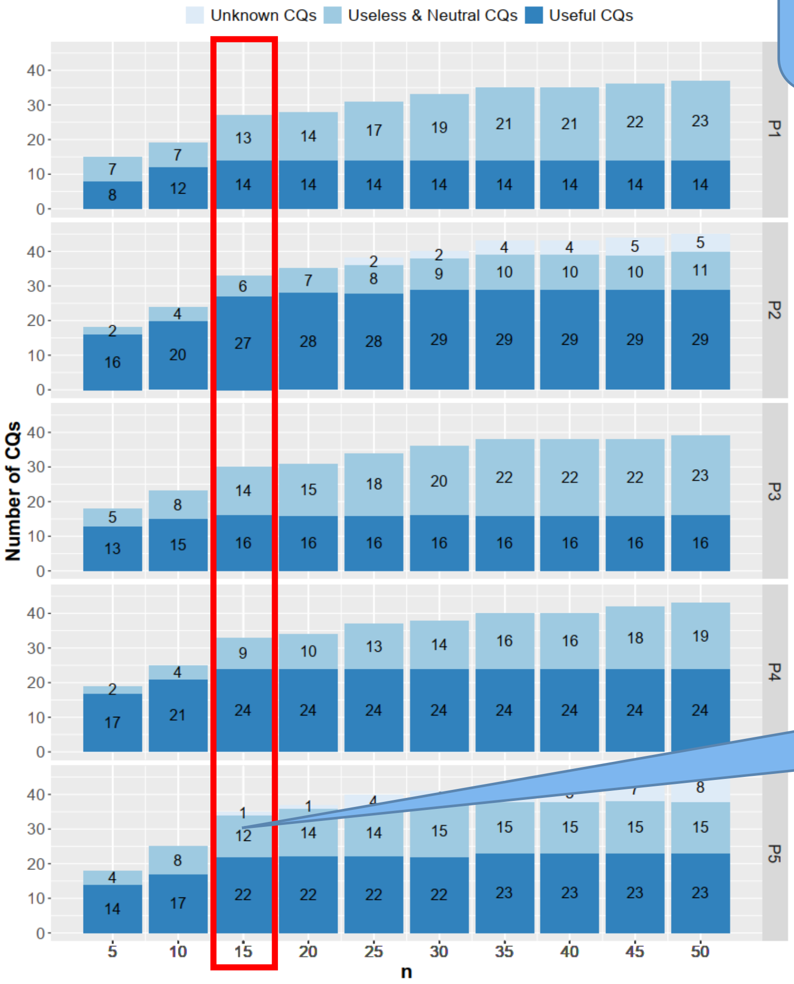
RQ1: What are the proper settings of the parameters n and η in Chatbot4QR?

- We conducted **a pilot user study** with 5 participants on 10 randomly selected queries.

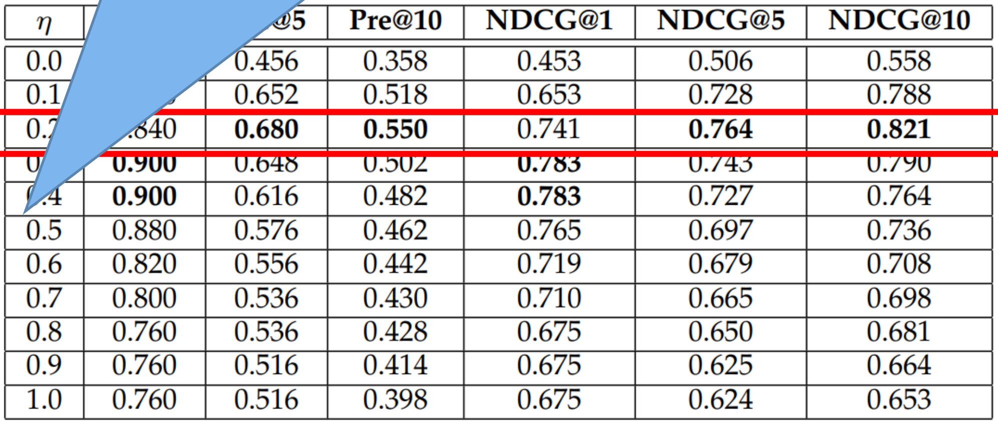
- Tested settings:
  - **n: from 5 to 50**
  - **η : from 0.0 to 1.0**

- The participants **performed:**
  - Evaluated the **usefulness of the CQs**
  - **Gave feedback to useful CQs**
  - Evaluated the **relevance of the recommended top-k SO questions**

RQ1: What are the proper settings of the parameters n and η in Chatbot4QR?



When *η* = 0.2, Chatbot4QR achieved the optimal values on most of the Pre@k and NDCG@k metrics.



η		@5	Pre@10	NDCG@1	NDCG@5	NDCG@10
0.0		0.456	0.358	0.453	0.506	0.558
0.1		0.652	0.518	0.653	0.728	0.788
0.	.840	**0.680**	**0.550**	0.741	**0.764**	**0.821**
0	**0.900**	0.648	0.502	**0.783**	0.743	0.790
.4	**0.900**	0.616	0.482	**0.783**	0.727	0.764
0.5	0.880	0.576	0.462	0.765	0.697	0.736
0.6	0.820	0.556	0.442	0.719	0.679	0.708
0.7	0.800	0.536	0.430	0.710	0.665	0.698
0.8	0.760	0.536	0.428	0.675	0.650	0.681
0.9	0.760	0.516	0.414	0.675	0.625	0.664
1.0	0.760	0.516	0.398	0.675	0.624	0.653

When *n* = 15, more than 93% CQs are useful for a query.

Proper settings: *n* = 15, *η* = 0.

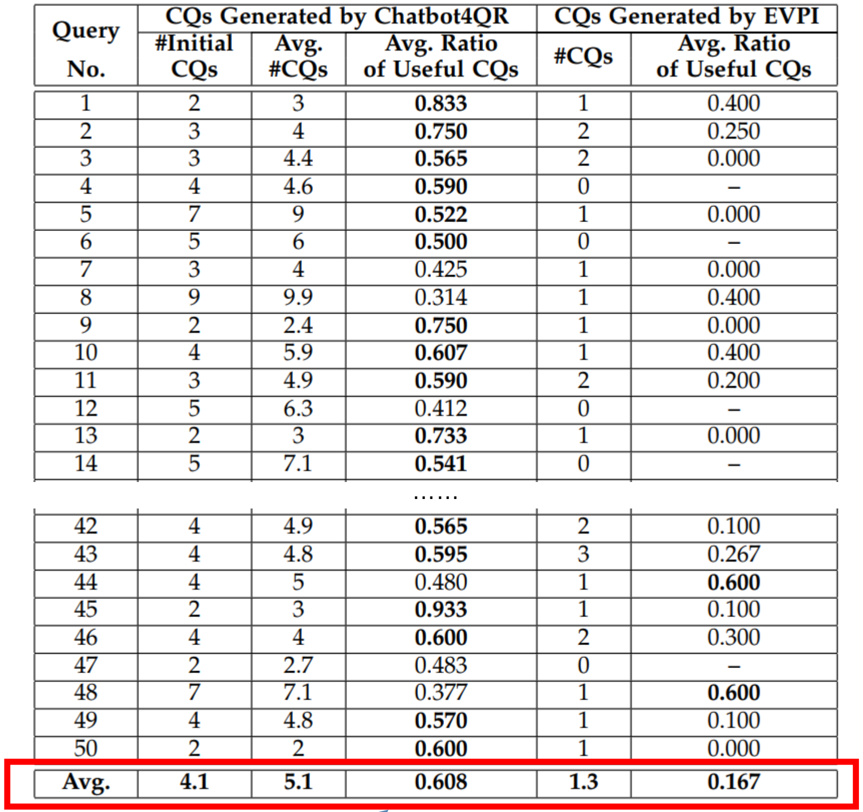
RQ2: How effective can Chatbot4QR generate CQs?

- We conducted **a user study** with 20 participants on 50 queries.

- **Baseline**:
  - **EVPI** [ACL'18]: a neural network based approach to generating CQs for asking good technical questions in Q&A sites.

- The participants **performed**:
  - Evaluated the **usefulness of CQs** generated for queries by Chatbot4QR and EVPI
  - **Gave feedback to the useful CQs**

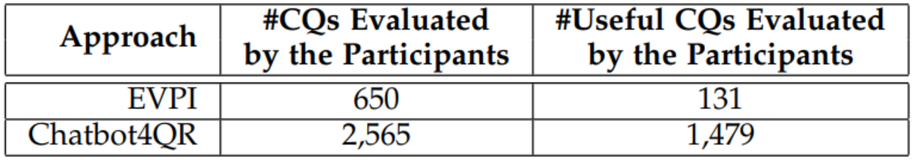
RQ2: How effective can Chatbot4QR generate CQs?



Query No.	CQs Generated by Chatbot4QR			CQs Generated by EVPI	
	#Initial CQs	**Avg. #CQs**	**Avg. Ratio of Useful CQs**	**#CQs**	**Avg. Ratio of Useful CQs**
1	2	3	**0.833**	1	0.400
2	3	4	**0.750**	2	0.250
3	3	4.4	**0.565**	2	0.000
4	4	4.6	**0.590**	0	–
5	7	9	**0.522**	1	0.000
6	5	6	**0.500**	0	–
7	3	4	0.425	1	0.000
8	9	9.9	0.314	1	0.400
9	2	2.4	**0.750**	1	0.000
10	4	5.9	**0.607**	1	0.400
11	3	4.9	**0.590**	2	0.200
12	5	6.3	0.412	0	–
13	2	3	**0.733**	1	0.000
14	5	7.1	**0.541**	0	–
			……		
42	4	4.9	**0.565**	2	0.100
43	4	4.8	**0.595**	3	0.267
44	4	5	0.480	1	**0.600**
45	2	3	**0.933**	1	0.100
46	4	4	**0.600**	2	0.300
47	2	2.7	0.483	0	–
48	7	7.1	0.377	1	**0.600**
49	4	4.8	**0.570**	1	0.100
50	2	2	**0.600**	1	0.000
Avg.	**4.1**	**5.1**	**0.608**	**1.3**	**0.167**

The overall performance of Chatbot4QR and EVPI on 50 queries.

Statistics of the CQs and useful CQs generated for 50 queries by both approaches.



Approach	#CQs Evaluated by the Participants	#Useful CQs Evaluated by the Participants
EVPI	650	131
Chatbot4QR	2,565	1,479

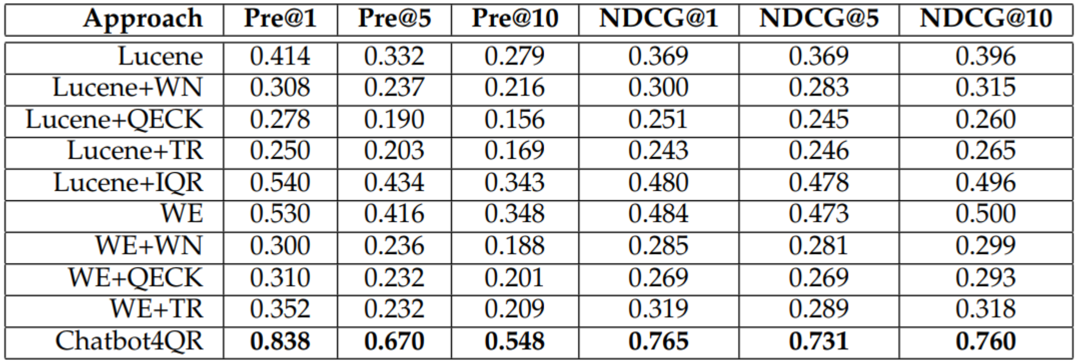
On average, Chatbot4QR generate 5.1 CQs for a query and 60.8% are useful, which outperforms *EVPI*.

RQ3: Can Chatbot4QR retrieve more relevant SO questions than the state-of-the-art question retrieval and query expansion approaches?

- We conducted a user study with 20 participants on 50 queries.

- **Nine Baselines**:
  - **Two popular retrieval methods**: **Lucene**, **Word Embedding (WE)** [ASE’18]
  - **Four query expansion methods**: **WordNet** (WN) [SANER’15], **QECK** (a SO based) [TSC’16], **TR** (a tag recommendation based) [ASEJ’18], **IQR** (i.e., our interactive query refinement method)

RQ3: Can Chatbot4QR retrieve more relevant SO questions than the state-of-the-art question retrieval and query expansion approaches?

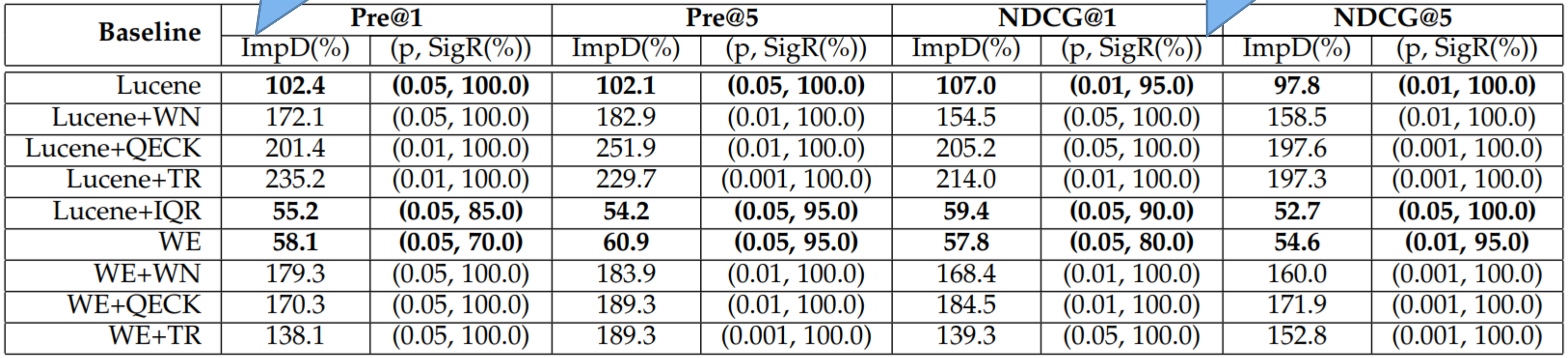


Approach	Pre@1	Pre@5	Pre@10	NDCG@1	NDCG@5	NDCG@10
Lucene	0.414	0.332	0.279	0.369	0.369	0.396
Lucene+WN	0.308	0.237	0.216	0.300	0.283	0.315
Lucene+QECK	0.278	0.190	0.156	0.251	0.245	0.260
Lucene+TR	0.250	0.203	0.169	0.243	0.246	0.265
Lucene+IQR	0.540	0.434	0.343	0.480	0.478	0.496
WE	0.530	0.416	0.348	0.484	0.473	0.500
WE+WN	0.300	0.236	0.188	0.285	0.281	0.299
WE+QECK	0.310	0.232	0.201	0.269	0.269	0.293
WE+TR	0.352	0.232	0.209	0.319	0.289	0.318
Chatbot4QR	**0.838**	**0.670**	**0.548**	**0.765**	**0.731**	**0.760**

Chatbot4QR improves the baselines by at least 54.6%, and the improvement is statistically significant for >=70% participants.

Improvement Degree of Chatbot4QR over baselines

statistically significant ratio of the improvement



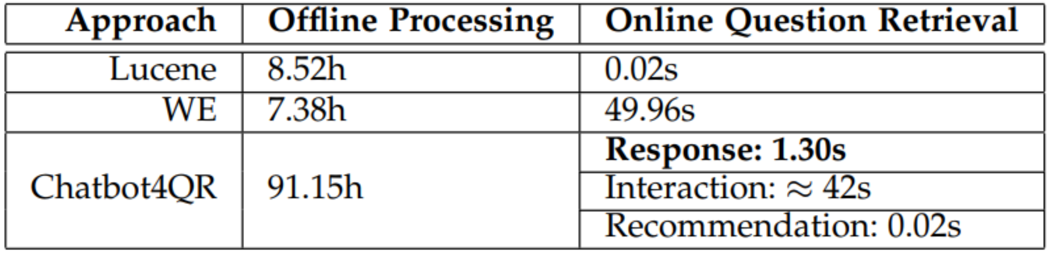
Baseline	Pre@1		Pre@5		NDCG@1		NDCG@5	
	ImpD(%)	(p, SigR(%))	ImpD(%)	(p, SigR(%))	ImpD(%)	(p, SigR(%))	ImpD(%)	(p, SigR(%))
Lucene	**102.4**	**(0.05, 100.0)**	**102.1**	**(0.05, 100.0)**	**107.0**	**(0.01, 95.0)**	**97.8**	**(0.01, 100.0)**
Lucene+WN	172.1	(0.05, 100.0)	182.9	(0.01, 100.0)	154.5	(0.05, 100.0)	158.5	(0.01, 100.0)
Lucene+QECK	201.4	(0.01, 100.0)	251.9	(0.01, 100.0)	205.2	(0.05, 100.0)	197.6	(0.001, 100.0)
Lucene+TR	235.2	(0.01, 100.0)	229.7	(0.001, 100.0)	214.0	(0.01, 100.0)	197.3	(0.001, 100.0)
Lucene+IQR	**55.2**	**(0.05, 85.0)**	**54.2**	**(0.05, 95.0)**	**59.4**	**(0.05, 90.0)**	**52.7**	**(0.05, 100.0)**
WE	**58.1**	**(0.05, 70.0)**	**60.9**	**(0.05, 95.0)**	**57.8**	**(0.05, 80.0)**	**54.6**	**(0.01, 95.0)**
WE+WN	179.3	(0.05, 100.0)	183.9	(0.01, 100.0)	168.4	(0.01, 100.0)	160.0	(0.001, 100.0)
WE+QECK	170.3	(0.05, 100.0)	189.3	(0.01, 100.0)	184.5	(0.01, 100.0)	171.9	(0.001, 100.0)
WE+TR	138.1	(0.05, 100.0)	189.3	(0.001, 100.0)	139.3	(0.05, 100.0)	152.8	(0.001, 100.0)

RQ4: How efficient is Chatbot4QR?

- We recorded the **time costs of three representative approaches**: Chatbot4QR, Lucene, and WE.

- For Chatbot4QR, we recorded three kinds of the amount of times:
  - **Respond**: the amount of time required by Chatbot4QR to respond to a participant after receiving a query.
  - **Interaction**: the amount of time required by a participant to interact with Chatbot4QR.
  - **Recommendation**: the amount of time required to produce the top-k recommended questions.

RQ4: How efficient is Chatbot4QR?



Approach	Offline Processing	Online Question Retrieval
Lucene	8.52h	0.02s
WE	7.38h	49.96s
Chatbot4QR	91.15h	**Response: 1.30s** Interaction: $\approx$ 42s Recommendation: 0.02s

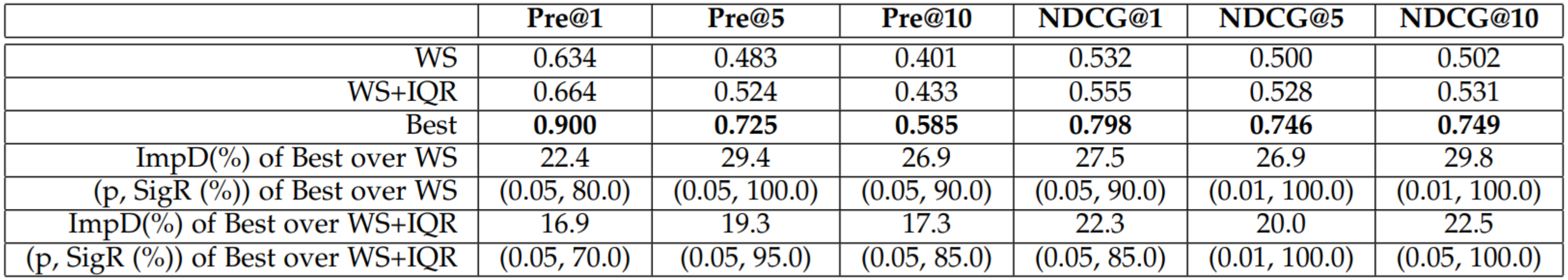
Chatbot4QR takes approximately 1.3s to respond to a user, which is acceptable for practical uses, as confirmed by the participant

RQ5: Can Chatbot4QR help obtain better results than using Web search engines alone?

- We conducted **four user studies** with 20 participants on 50 queries.

- The participants **performed**:
  - **WS**: Search the top-k results for queries using Web search engines (e.g., Google, SO, etc.) **before interacting with Chatbot4QR**.
  - **WS+IQR**: Search a new top-k results for queries using Web search engines **after interacting with Chatbot4QR**.
  - Evaluated the relevance of search results.
  - Chose the **Best** results from three kinds of results: WS, WS+IQR, and the top-k SO questions retrieved by Chatbot4QR.

RQ5: Can Chatbot4QR help obtain better results than using Web search engines alone?



	Pre@1	Pre@5	Pre@10	NDCG@1	NDCG@5	NDCG@10
WS	0.634	0.483	0.401	0.532	0.500	0.502
WS+IQR	0.664	0.524	0.433	0.555	0.528	0.531
Best	**0.900**	**0.725**	**0.585**	**0.798**	**0.746**	**0.749**
ImpD(%) of Best over WS	22.4	29.4	26.9	27.5	26.9	29.8
(p, SigR (%)) of Best over WS	(0.05, 80.0)	(0.05, 100.0)	(0.05, 90.0)	(0.05, 90.0)	(0.01, 100.0)	(0.01, 100.0)
ImpD(%) of Best over WS+IQR	16.9	19.3	17.3	22.3	20.0	22.5
(p, SigR (%)) of Best over WS+IQR	(0.05, 70.0)	(0.05, 95.0)	(0.05, 85.0)	(0.05, 85.0)	(0.01, 100.0)	(0.05, 100.0)

Chatbot4QR helps the participants obtain better results than using the Web search engines alone. The improvement of Best over WS is by at least 22.4%, and is statistically significant for >= 80% participants.

Future Work

- **Improve Chatbot4QR** by mining and incorporating the **knowledge of the relationships among SO tags**

- **Implement Chatbot4QR as a browser plugin** to help developers in question retrieval from Google, SO, etc.

Example Research 5

VT-Revolution: Interactive Programming Video Tutorial Authoring and Watching System

Bao, Xing, **Xia**, et al. TSE 2018

Background

Programming

Concept Knowledge

- Knowledge about concepts and APIs in the task

Procedural Knowledge

- Actions and manipulations that apply conceptual knowledge in the task

Programming Tutorials

Text

Video

More Valuable

Programming videos can serve as a reasonable approximation of watching a developer's live coding practice.

Background

☛ Limitations of programming videos

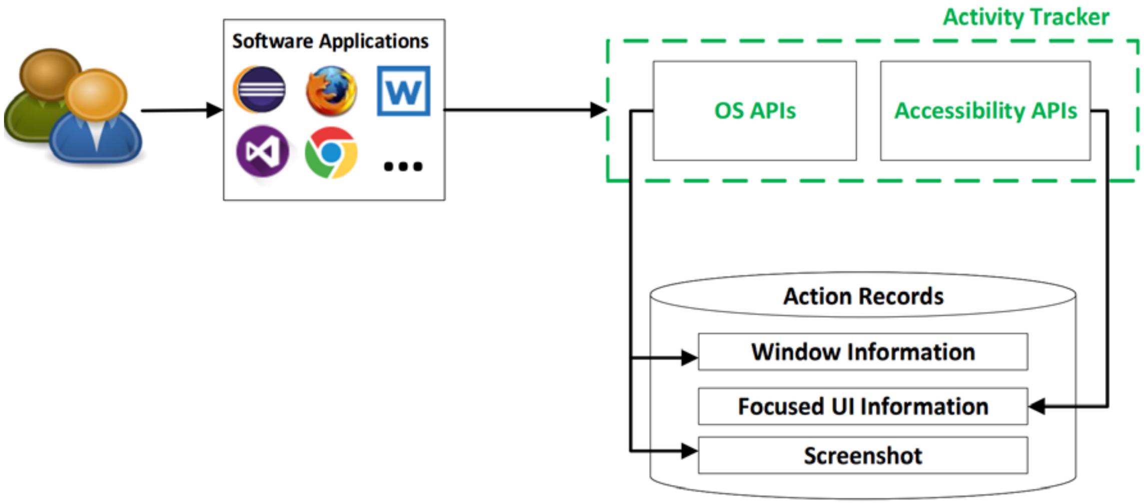
- Lack of a high-level overview of the workflow
- No effective navigation support of workflow and tutorial content
- Inconvenience in linking to supplementary resources

☛ Goal of our work

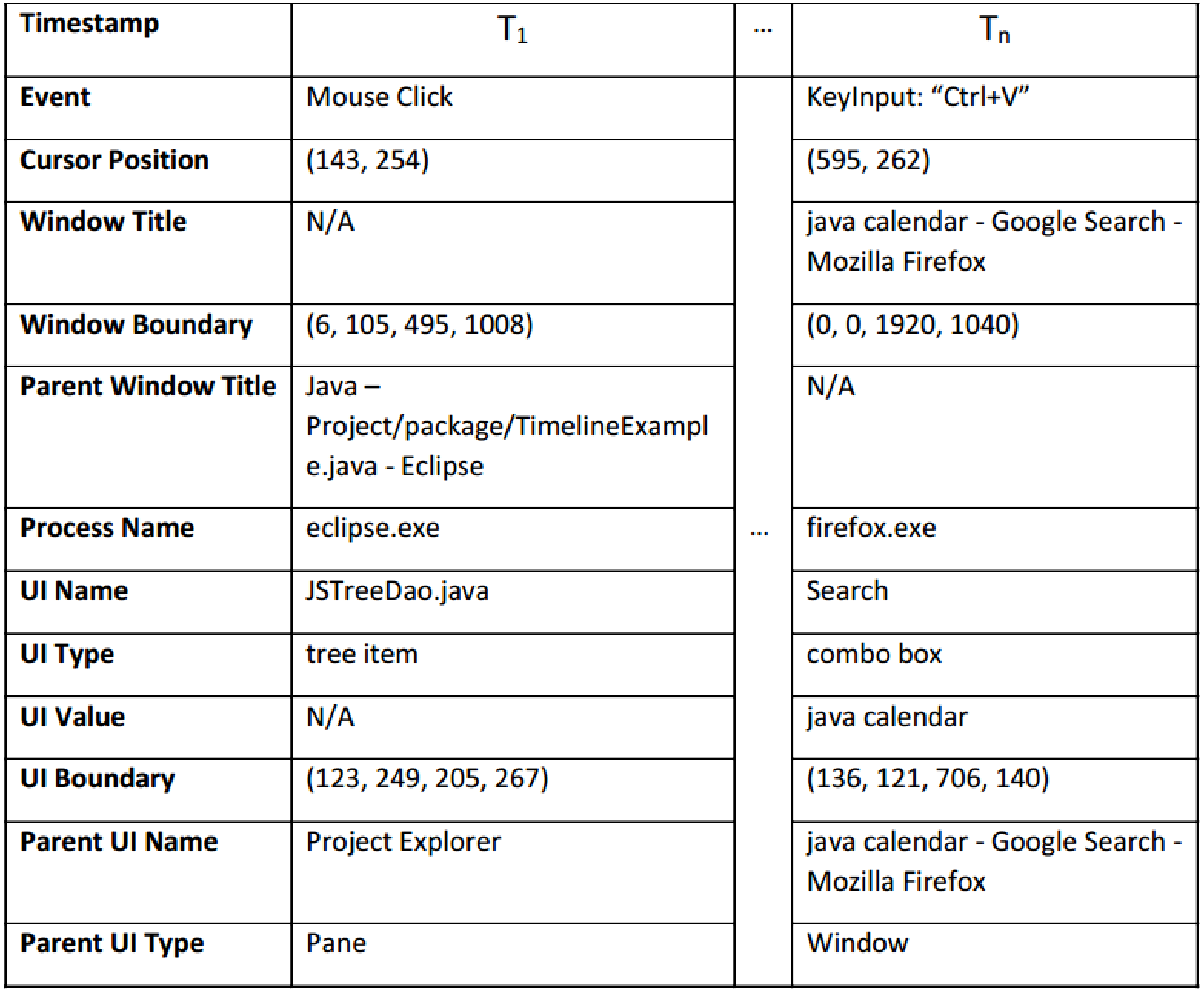
to make programming video tutorials **interactive**

- tutorial watchers can freely explore the workflow of a programming task in the video
- Interact with files, code and program output in the video in a similar way to the IDE interaction

ActivitySpace: A Framework to Support the Recording of Interapplication Interactions



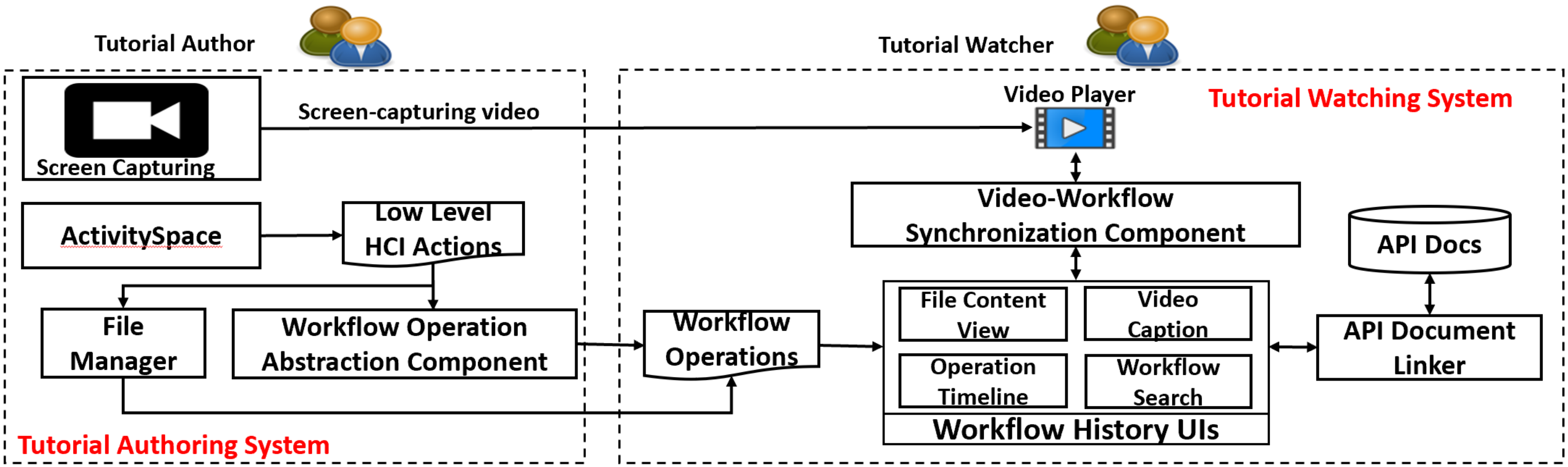
Low-level Interaction Data



Timestamp	T_1	…	T_n
Event	Mouse Click		KeyInput: “Ctrl+V”
Cursor Position	(143, 254)		(595, 262)
Window Title	N/A		java calendar - Google Search - Mozilla Firefox
Window Boundary	(6, 105, 495, 1008)		(0, 0, 1920, 1040)
Parent Window Title	Java – Project/package/TimelineExample.java - Eclipse		N/A
Process Name	eclipse.exe	…	firefox.exe
UI Name	JSTreeDao.java		Search
UI Type	tree item		combo box
UI Value	N/A		java calendar
UI Boundary	(123, 249, 205, 267)		(136, 121, 706, 140)
Parent UI Name	Project Explorer		java calendar - Google Search - Mozilla Firefox
Parent UI Type	Pane		Window

Is it possible to use ActivitySpace to make video tutorial interactive?

Our System: VT-Revolution



VT-Revolution: Tutorial Authoring System

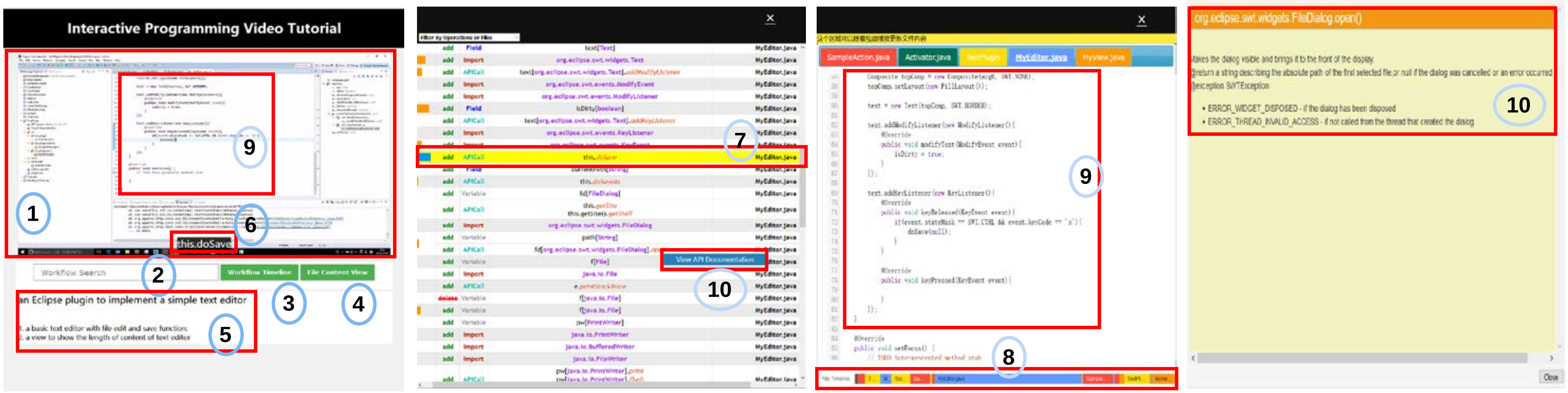
☛ Workflow operation abstraction

Operation Category	Operation Type	Notion
File	Edit	FileOpen<t_i, *name*>
	View	FileSwitch<t_i, *origin, target*>
Exception	Inspect	Inspect<t_{i-1}, t_i, *exception*>
Code Element	Add	Add<t_{i-1}, t_i, *type, info*>
	Delete	Delete<t_{i-1}, t_i, *type, info*>
Text Content	Edit	Edit<t_{i-1}, t_i, *file, change*>

ASTNode	Info
Import statement	Package name in the import statement
Field Declaration	Field name, Field datatype name
Variable Declaration	Variable name, Variable datatype name
Method Call	Method identifier, Object and its datatype on which a method is called

VT-Revolution: Tutorial Watching System

☛ **Screenshots of our prototype**



an Eclipse plugin to implement a simple text editor

1. a basic text editor with file edit and save function;
2. a view to show the length of content of text editor

org.eclipse.swt.widgets.FileDialog.open()

Makes the dialog visible and brings it to the front of the display.

@return a string describing the absolute path of the first selected file or null if the dialog was cancelled or an error occurred

@exception SWTException

- ERROR_WIDGET_DISPOSED - if the dialog has been disposed
- ERROR_THREAD_INVALID_ACCESS - if not called from the thread that created the dialog

(a) Main

(b) Workflow Operation Timeline

(c) File Content View

(d) API Document Linking

- Prototype website: http://baolingfeng.xyz:8080/VTRevolution/

Experiment

☛ Research Questions

- **RQ1.** How well and efficiently does our *VT-Revolution* system help developers search relevant information in video tutorials, compared with developers using the OCR prototype and regular video player?

- **RQ2**. Are the participants using *VT-Revolution* more satisfied with the learning experience of the video tutorials than those using the OCR prototype and regular video player?

- **RQ3**. Which feature(s) of *VT-Revolution* are most useful?

Experiment Setup

☛ Programming Tutorials

Tutorial	Programming Task	LOC	#File	Duration
email	A simple program to send email	75	2	08:39
mysql	a program to illustrate some MySql Database operations	175	1	11:06
plugin	a Eclipse plugin	309	5	19:19

☛ Baseline tools

- Regular video player
- A prototype with OCRed-text based search and navigation

Experiment Setup

☛ Participants

- **135** professional developers who do not use Java as main programming language
- **Nine** comparable groups:
  for each tutorial, one experimental group (VT-Revolution) and two control groups (regular video players and OCR prototype)

Project	Year	#Dev.	Pro.	#Participant
A	6	136	C#	40
B	4	90	C#	25
C	4	18	C#	12
D	3	48	C#	15
E	2	10	Python	4
F	4	28	Python	12
G	2	32	C/C++	12
H	6	68	C/C++	15

Experiment Setup

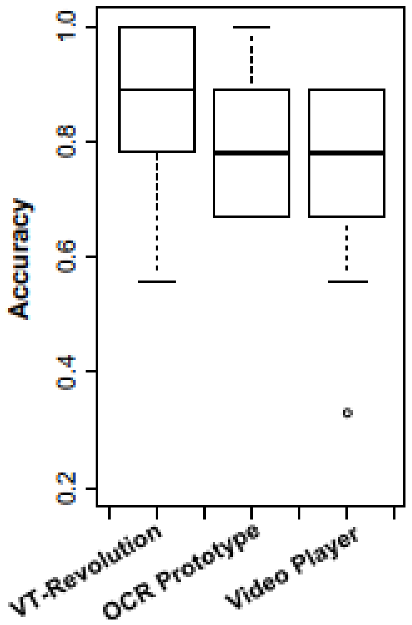
☛ Questionnaire Design

- **API Usage**
  - In tutorial
  - API documentation
- **Workflow**
- **Output**
- **File Content**

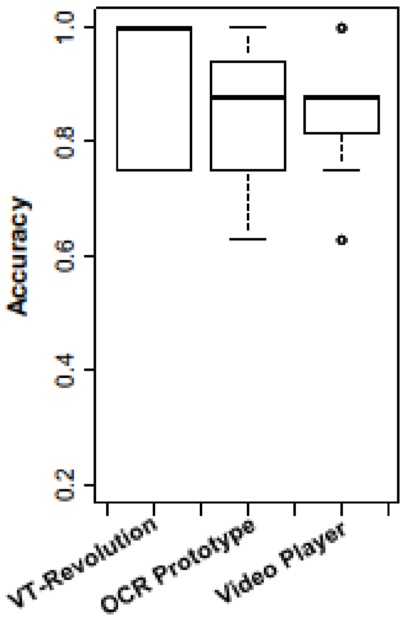
Question Category	email	mysql	plugin
API Usage	4	3	4
Workflow	2	3	3
Output	1	2	1
File Content	2	1	3

Experiment Results

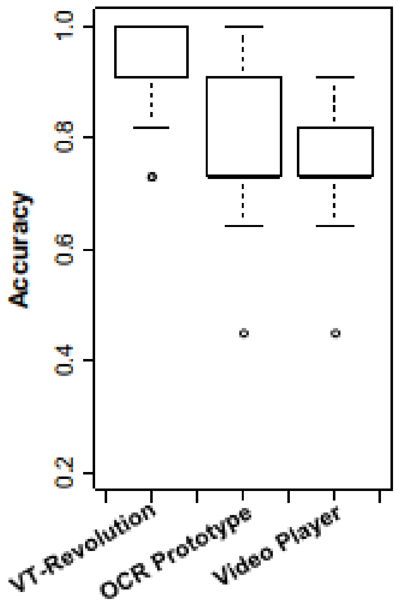
☛ **RQ1 – Accuracy of answers to questions**



email



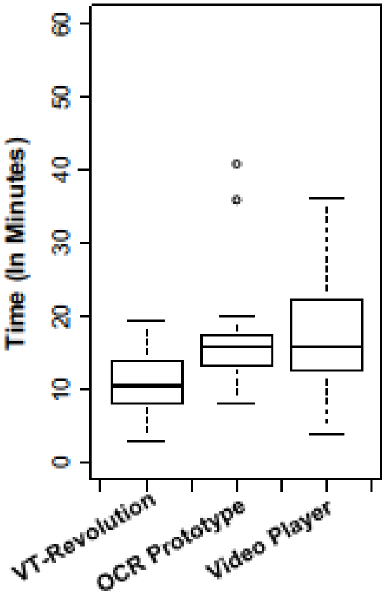
mysql



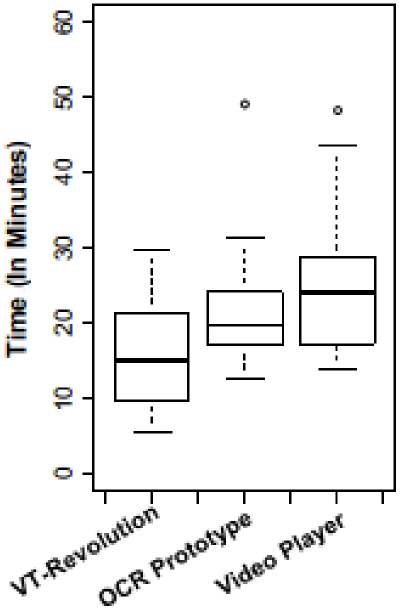
plugin

Experiment Results

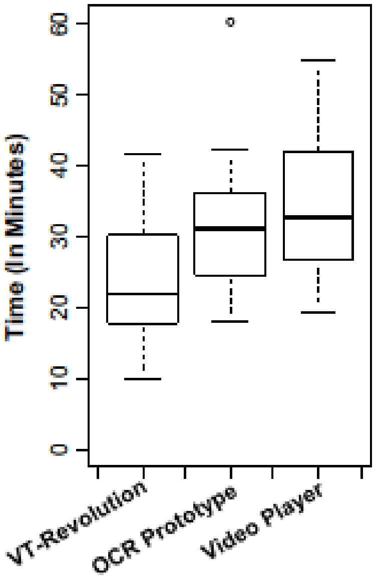
☛ RQ1 – Time of completing questionnaires



email



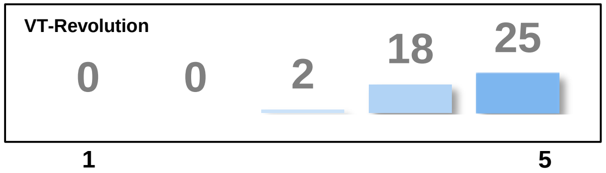
mysql



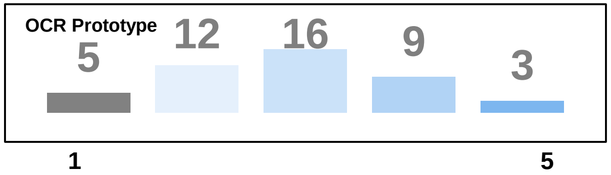
plugin

Experiment Results

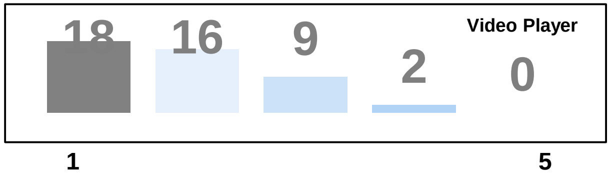
RQ2 – Satisfaction



“The code in text format is more familiar to me than the code in video. I can copy the code fragment from the video tutorial using this tool. Very cool! ”



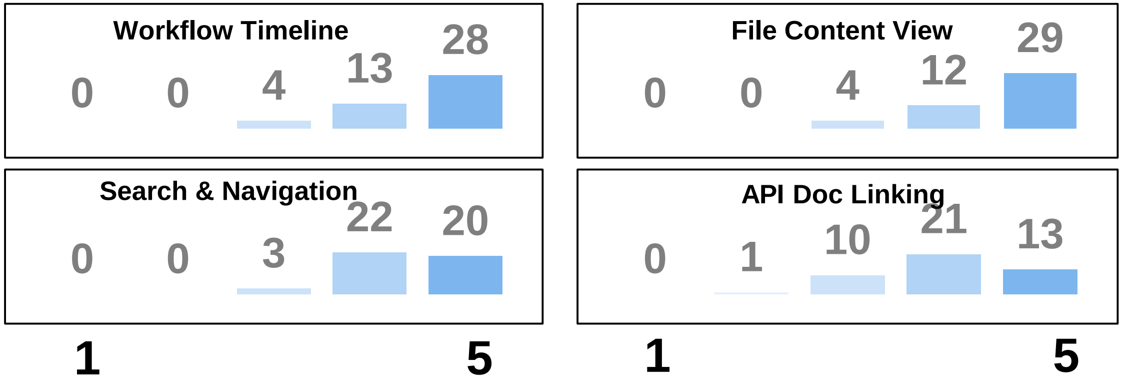
“I can use this tool to navigate the video tutorial, but for some questions in the questionnaire that require the context and programming process, that’s not enough. I have to spend more time to look into the tutorial.”



“Even though I can locate the information in the video, I often need to watch this fragment of the video repeatedly so that I can find out what really happen.”

Experiment Results

RQ3 – Usefulness of different features



- *“I can know the whole **workflow** more clearly using this timeline and use it to navigate video more easily.”*

- *“**File content view** gives me an overview of the program in the video, and it is easy to know the code change by comparing the code content at two different times.”*

- *“I like the **synchronization between the video and the workflow**. I can easily find the needed information and jump to that video part.”*

- *“I do not know the usage of many classes and APIs in the video tutorial since I never write Eclipse plugin programs. I can understand the video better using **API documentation**, just like what I can do in the IDE”*

Discussion

- **Working environment as a tutorial system**
- **Bridging conceptual and procedural knowledge in software engineering**
- **Making existing video tutorials interactive**

Example Research

- Automatic **Generation of Pull Request Descriptions**
- **API Method Recommendation** without Worrying About the Task-API Knowledge Gap
- Automating Change-Level **Self-Admitted Technical Debt Determination**
- **Chatbot4QR**: Interactive Query Refinement for Technical Question Retrieval
- **VT-Revolution**: Interactive Programming Video Tutorial Authoring and Watching System

Three Challenges

- **Easy over hard**: simple solution first

- **Strong performance of techniques is not sufficient**, instead a deeper understanding of the domain is essential

- Results should be presented in a **domain-centric context**

Challenge 1

Easy over hard: simple solution first

Neural-Machine-Translation-Based Commit Message Generation: How Far Are We?

ACM SIGSOFT Distinguished Paper Award

Liu, **Xia** et al. ASE 2018

ASE 2018

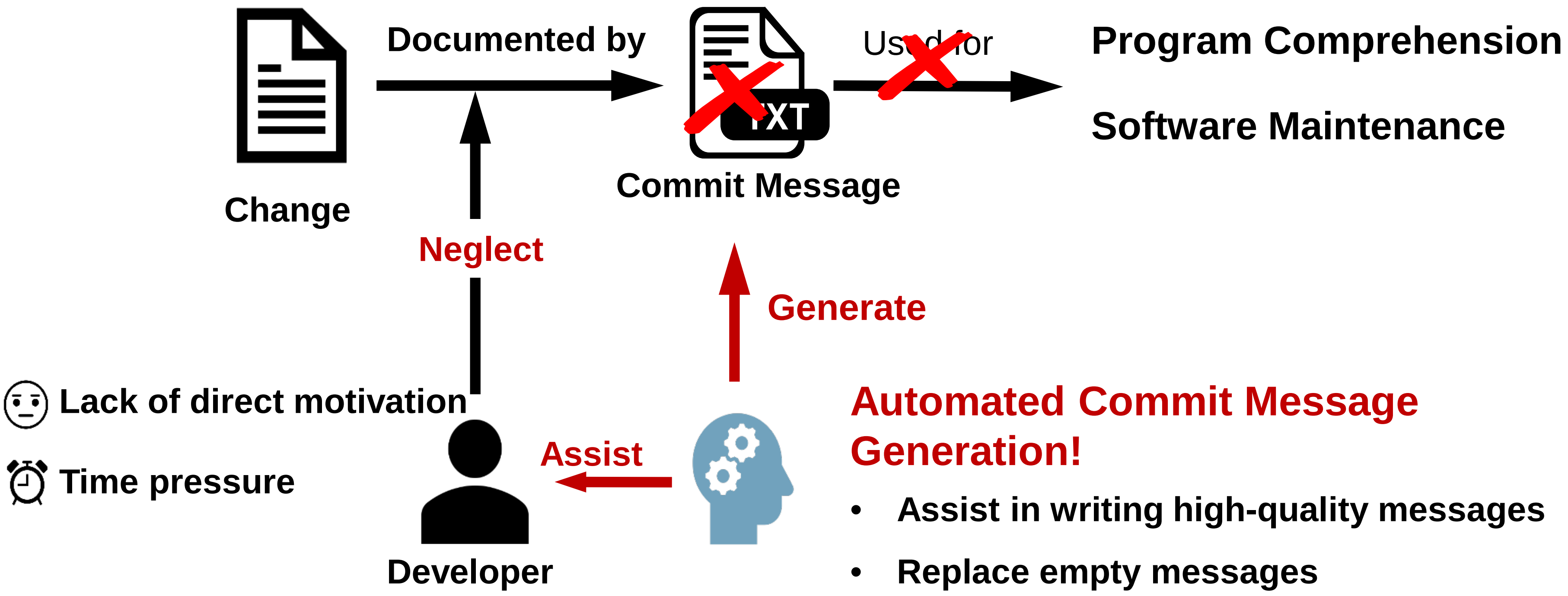
Certificate

of the

ACM SIGSOFT Distinguished PAPER AWARD

33rd IEEE/ACM International Conference
on Automated Software Engineering
September, 3 to 7 2018 - Montpellier, France

Commit Messages



Change

Documented by

Commit Message

Used for

Program Comprehension

Software Maintenance

Neglect

Generate

😑 Lack of direct motivation

⏰ Time pressure

Assist

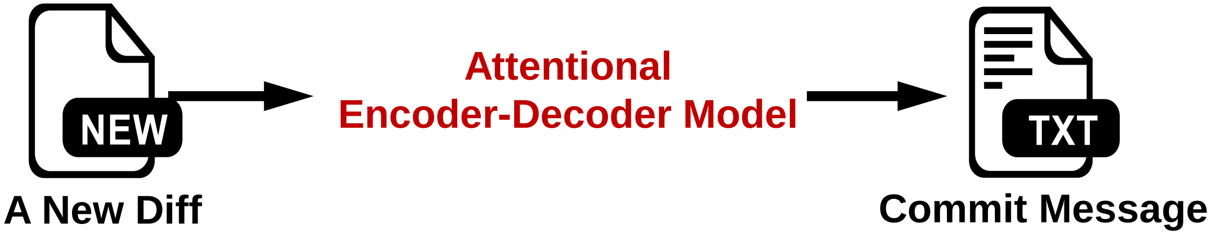
Developer

Automated Commit Message Generation!

- Assist in writing high-quality messages
- Replace empty messages

NMT-Based Commit Message Generation

- Recently, Jiang et al. proposed an approach, which uses a Neural Machine Translation (NMT) algorithm to generate one-sentence commit messages from diffs. [Jiang et al. ASE 2017]



Evaluation of NMT

- Jiang et al. evaluated *NMT* using the BLEU-4 score:
  - an accuracy measure that is widely used to evaluate machine translation systems

Model	Task	BLEU-4
NMT	diff -> commit msg	31.92

Model	Task	BLEU-4
Transformer[1]	En -> Fr	41.0
	En -> De	28.4

[1] Vaswani, Ashish, et al. "Attention is all you need." *Advances in Neural Information Processing Systems*. 2017.

However …

- Jiang et al. did not investigate the reasons behind *NMT*'s good performance.

RQ1: Why does NMT perform so well?

- *NMT* is complicated and slow!
  - Attentional RNN encoder-decoder model
  - 38 hours for training on a GPU

RQ2: Can a simpler and faster approach outperform *NMT?*

RQ1: Why does NMT perform so well?

Analyze *NMT Messages*

- ***NMT messages****:* commit messages generated by *NMT*

200 Test Commits

Randomly select 200 commits from Jiang et al.'s test set

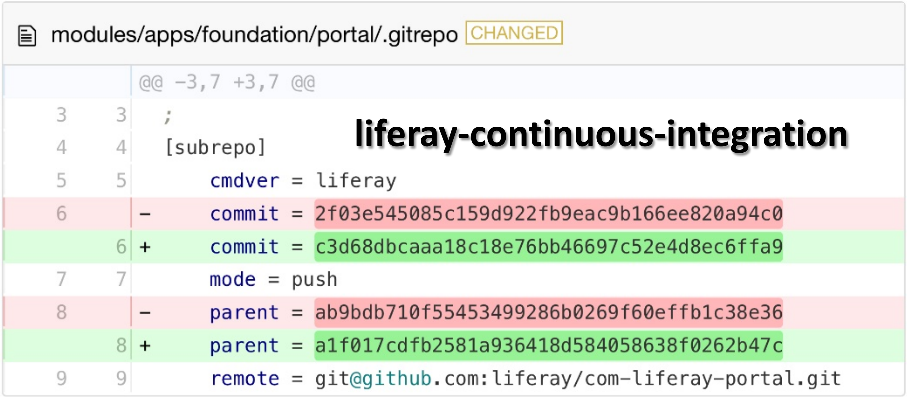
Human Evaluation

- Scores range from 0-7
- Pick out *NMT messages* of score of 6 or 7 (*good messages*)

Manually Analyze
Good Messages

Noisy Messages

- (37%) of the *reference messages* of these *good messages* are noisy.
- Two types of noisy messages:



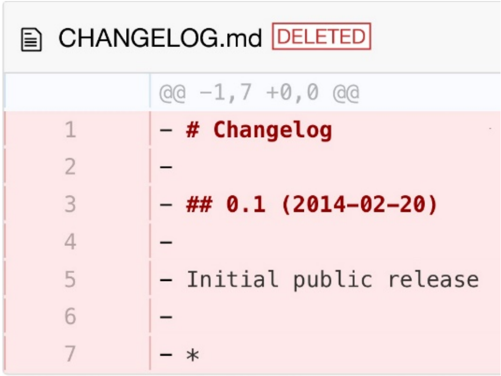
modules/apps/foundation/portal/.gitrepo CHANGED

liferay-continuous-integration

```
@@ -3,7 +3,7 @@
  ;
  [subrepo]
      cmdver = liferay
-     commit = 2f03e545085c159d922fb9eac9b166ee820a94c0
+     commit = c3d68dbcaaa18c18e76bb46697c52e4d8ec6ffa9
      mode = push
-     parent = ab9bdb710f55453499286b0269f60effb1c38e36
+     parent = a1f017cdfb2581a936418d584058638f0262b47c
      remote = git@github.com:liferay/com-liferay-portal.git
```

Reference Message:
ignore Update ' modules / apps / foundation / portal / .

Message Generated by NMT:
Ignore Update ' modules / apps / foundation / portal / .



CHANGELOG.md DELETED

```
@@ -1,7 +0,0 @@
- # Changelog
-
- ## 0.1 (2014-02-20)
-
- Initial public release
-
- *
```

Reference Message:
update changelog

Message Generated by NMT:
Updated changelog

Bot Message

Automatically generated by other dev tools

Trivial Message

Contains little and redundant information

The Impact of Noisy Commits

- Identify Noisy Messages in Jiang et al.’s Dataset

Dataset	Bot	Trivial	Total
Training	**12.6%**	**3.1%**	**15.6%**
Validation	**13.4%**	**2.9%**	**16.3%**
Test	**12.8%**	**3.2%**	**16.0%**

Noisy messages are common in Jiang et al.'s dataset!

- Train and test *NMT* on the cleaned dataset.

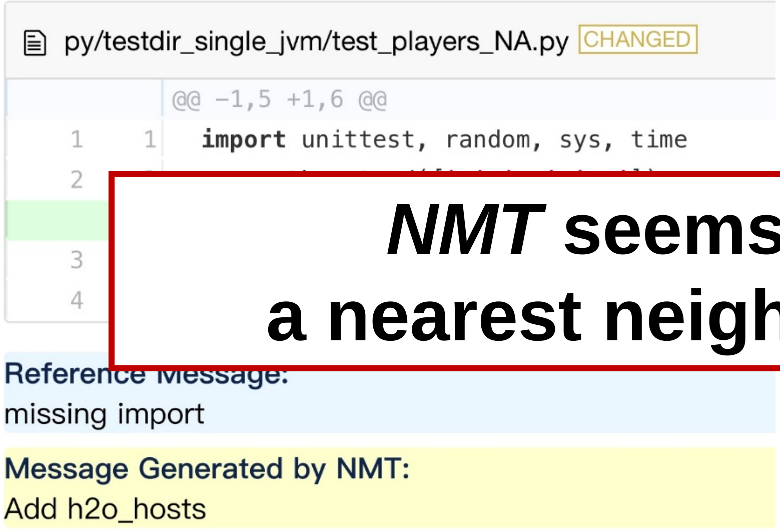
Dataset	BLEU-4
JIANG	**31.92**
Cleaned	**14.19**

Performance declines by a large amount!

RQ2 : Can a simpler and faster approach outperform *NMT*?

Another Finding of Our Analysis

- For nearly every (70/71) *good message*, we can find out one or more similar training commits:

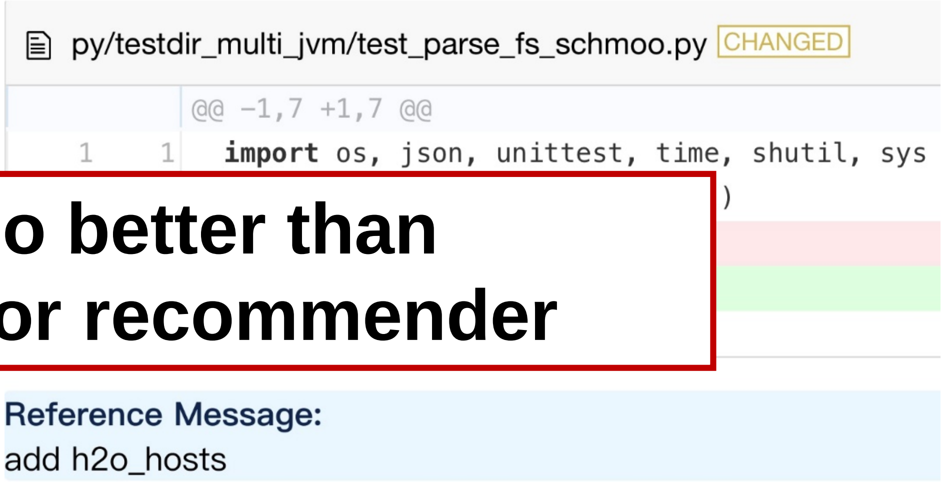


py/testdir_single_jvm/test_players_NA.py CHANGED

```
@@ -1,5 +1,6 @@
import unittest, random, sys, time
```

Reference Message:
missing import

Message Generated by NMT:
Add h2o_hosts



py/testdir_multi_jvm/test_parse_fs_schmoo.py CHANGED

```
@@ -1,7 +1,7 @@
import os, json, unittest, time, shutil, sys
```

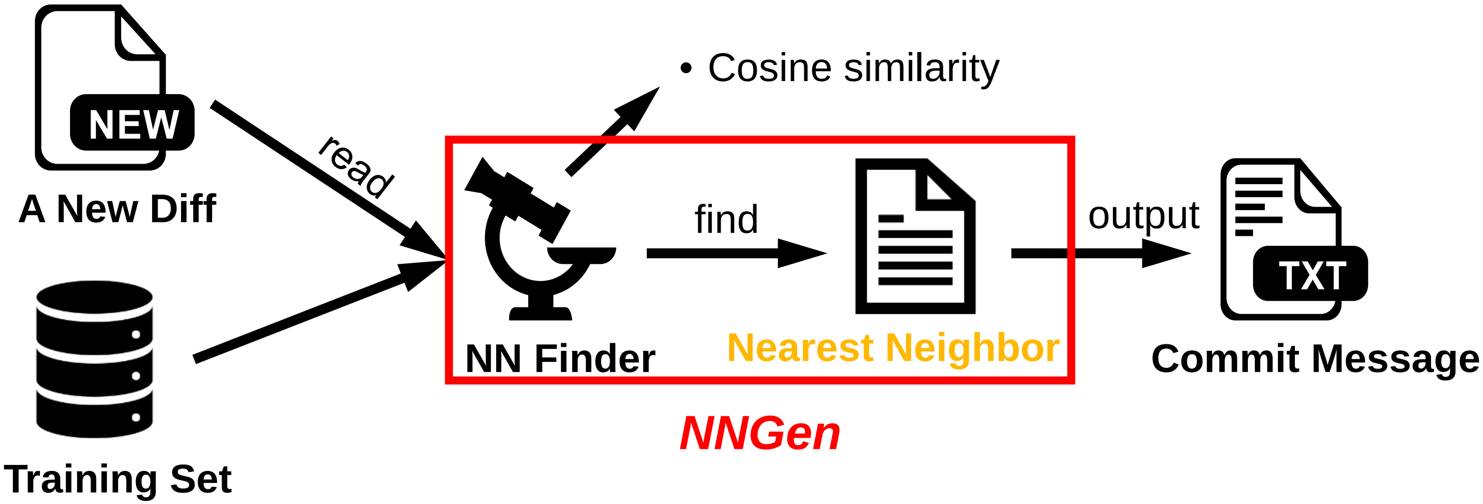
Reference Message:
add h2o_hosts

NMT seems no better than
a nearest neighbor recommender

A Test Commit

A Similar Training Commit

Nearest Neighbor Generator (NNGen)



Automatic Evaluation & Time Costs

Dataset	Approach	BLEU-4	
JIANG	*NMT*	31.92	
	NNGen	38.55	↑ 21%
Cleaned	*NMT*	14.19	
	NNGen	16.42	↑ 16%

Dataset	Approach	Device	Train	Test
JIANG	*NMT*	GTX 1070	38 hours	4.5 mins
	NMT	GTX 1080	34 hours	17 mins
	NNGen	CPU	N/A	30 secs
Cleaned	*NMT*	GTX 1080	24 hours	13 mins
	NNGen	CPU	N/A	23 secs

- **GTX 1070: Nvidia GTX 1070 GPU, time costs reported by Jiang et al.**
- **GTX 1080: Nvidia GTX 1080 GPU, time costs on our server**
- **CPU: Intel i5 2.6GHz**

Take-Away Message

- **Clean up the data carefully.**
  - Noisy commits will affect performance.

- **Consider simple approaches first.**
  - Specifically, consider the nearest neighbor algorithm first for diff-msg “translation” tasks.
  - Little effort to understand data, sometimes leads to better performance

Challenge 2

Strong performance of techniques is not sufficient

Measuring Program Comprehension:
A Large-Scale Field Study with Professionals.

Xia et al. TSE 2018

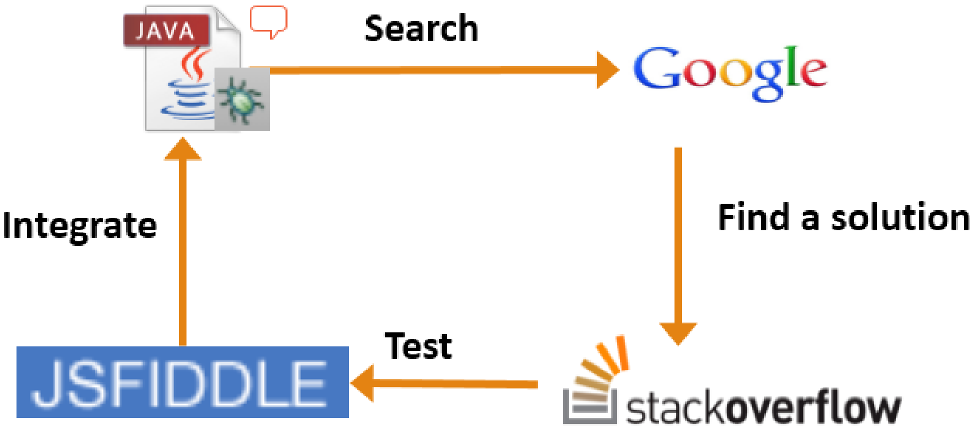
How Much Time Do Developers Spend on Program Comprehension Activities?

- Program comprehension is an essential and time-consuming activity in software maintenance
- Zelkowitz et al. : more than half of a time
- Minelli et al.: 70% of time

We want to validate a well-known assumption:
program comprehension is time consuming

Program comprehension activities might happen **across many applications**

Software Engineer



Search

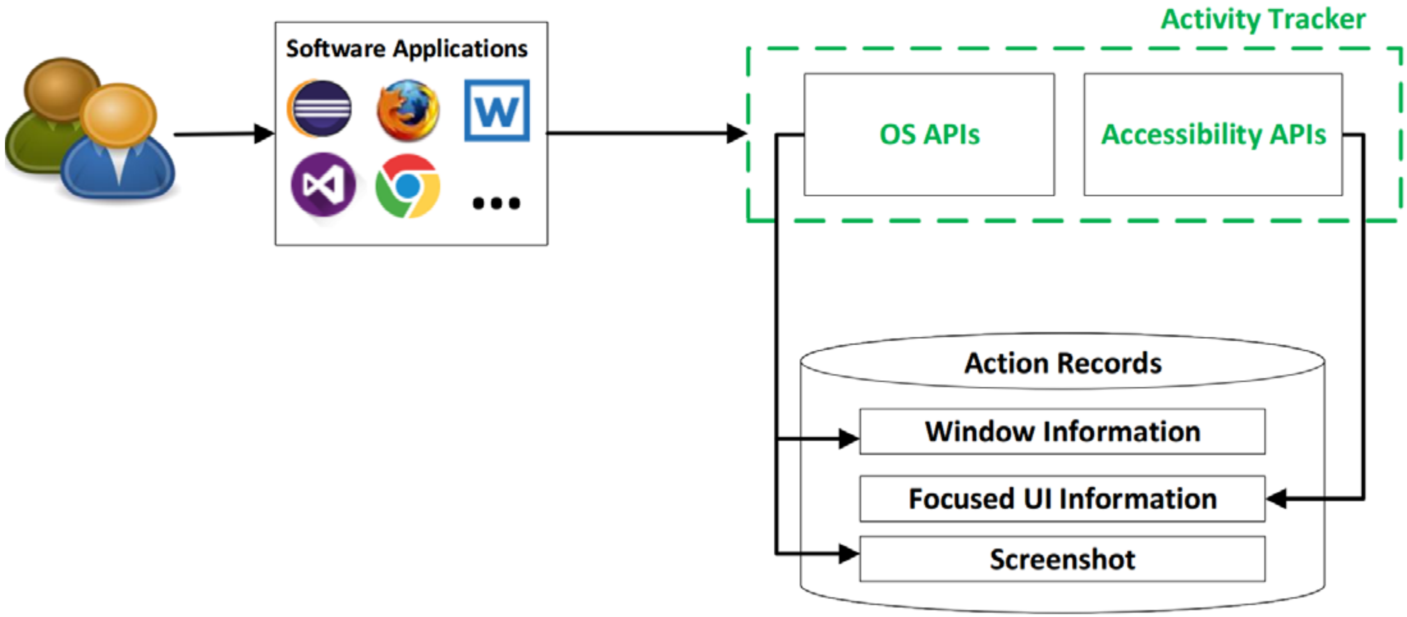
Find a solution

Test

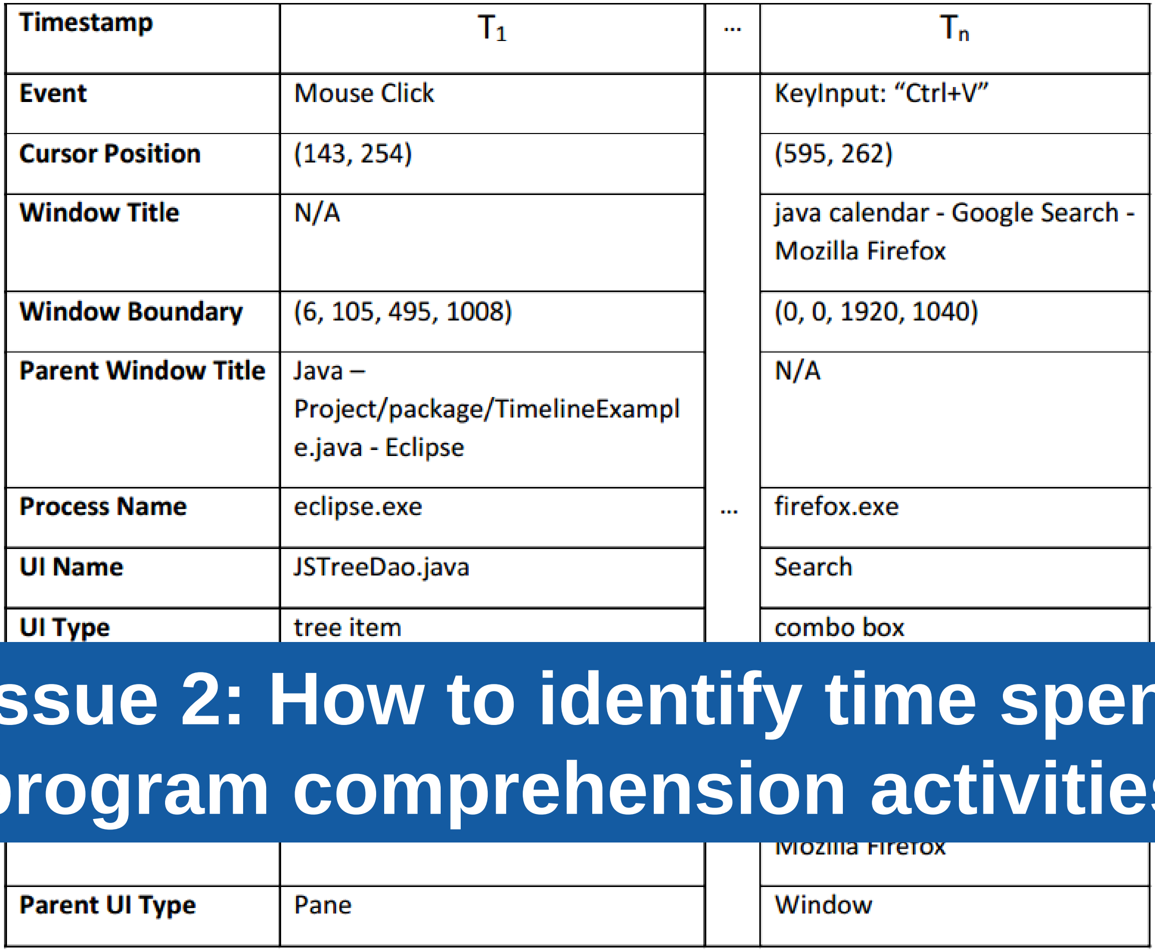
Integrate

Issue 1: How to collect interaction data across multiple applications?

ActivitySpace: A Framework to Support the Recording of Interapplication Interactions



Low-level Interaction Data



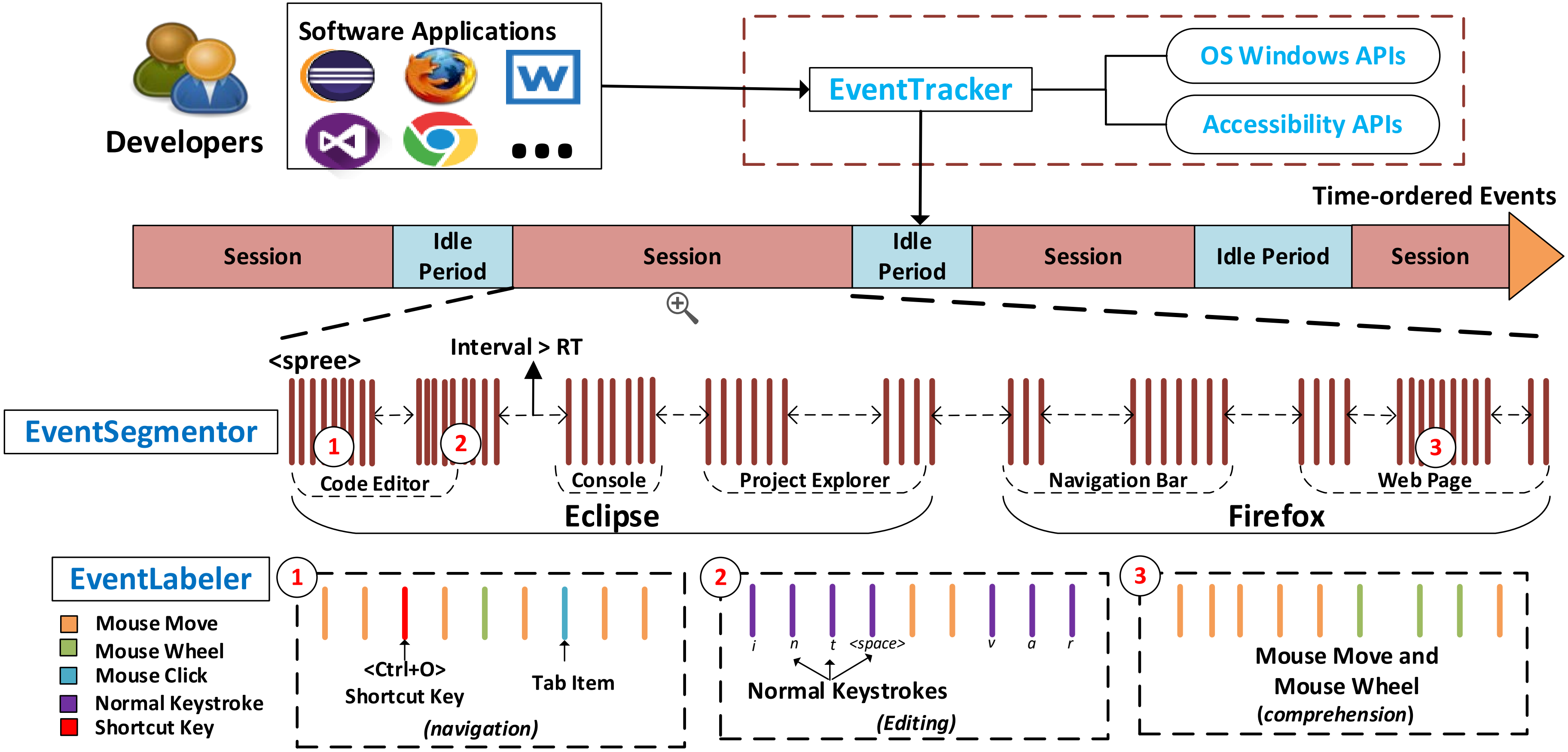
Timestamp	T_1	…	T_n
Event	Mouse Click		KeyInput: “Ctrl+V”
Cursor Position	(143, 254)		(595, 262)
Window Title	N/A		java calendar - Google Search - Mozilla Firefox
Window Boundary	(6, 105, 495, 1008)		(0, 0, 1920, 1040)
Parent Window Title	Java – Project/package/TimelineExample.java - Eclipse		N/A
Process Name	eclipse.exe	…	firefox.exe
UI Name	JSTreeDao.java		Search
UI Type	tree item		combo box
			Mozilla Firefox
Parent UI Type	Pane		Window

Issue 2: How to identify time spent on program comprehension activities?

Psychology: Reaction Time

- Time that elapses between the end of a physical action sequence (e.g., typing, moving the mouse, etc.) and the beginning of concrete mental processes (e.g., reflecting, or planning).

Data collection and analysis



Main Findings

Program comprehension takes up ~58% of developers' time

Besides IDEs, developers frequently use web browsers and document editors during program comprehension

Java developers > C# developers

Senior developers < Junior developers

Maintenance projects > new development projects

We can stop here, and conclude the whole study.

But why does it happen? Why do developers spend so much time on program comprehension?

Improved Study Process

- Data Collection and Analysis
- Interview
- Observation Study
- In-depth Data Analysis
- Post-study Survey

Data Scientist

Interview

- Interview 10 participants
- Open-ended questions, e.g., importance, challenges, and difficulties in program comprehension
- Topic discussion, e.g., impact of different programming languages and project phases on program comprehension

Observation Study

- Randomly choose 200 sessions which have long program comprehension times
- Identify the root causes

Long-Duration Program Comprehension Activities

- No comments or insufficient comments
- Meaningless classes/methods/variables names
- Large number of LOC in a class/method
- Inconsistent coding styles
- Navigating inheritance hierarchies
- Query refinement, and browsing a number of search results/links
- Lack of documents, and ambiguous/incomplete document content
- Searching for the relevant documents
- Unfamiliarity with business logic

Post-Study Survey

- Send the results section along with the abstract and introduction to ten interviewees
- Ask them for feedback about our findings

A Deeper Understanding of the Domain is Essential

Perform both quantitative and qualitative analysis

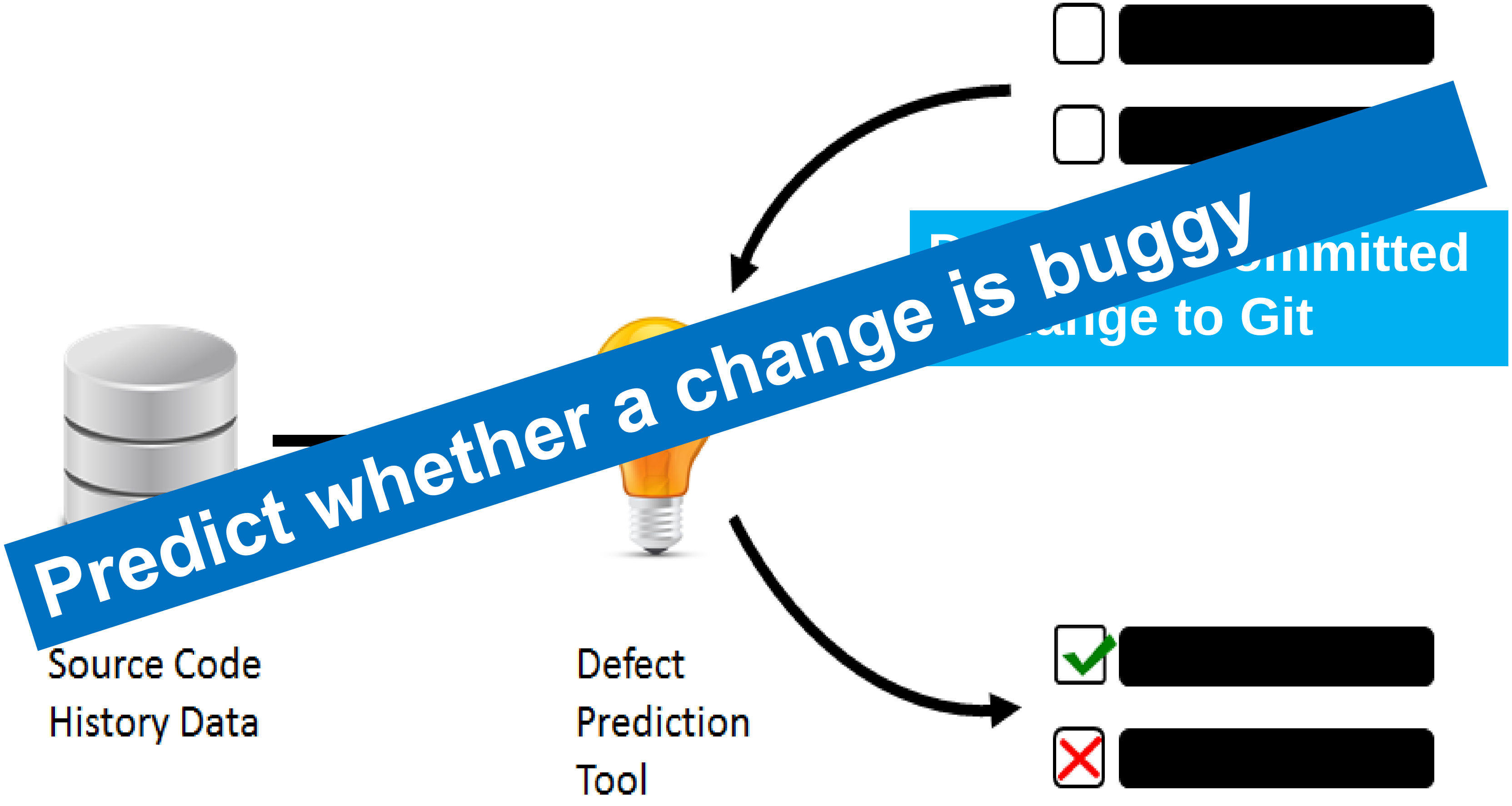
Challenge 3

Results should be presented in a domain-centric context

Supervised vs Unsupervised Models:
A Holistic Look at Effort-Aware Just-in-Time Defect Prediction.

Huang, **Xia**, Lo. ICSME 2017, EMSE 2018

Just-in-Time (JIT) Defect Prediction



D ommitted
ange to Git

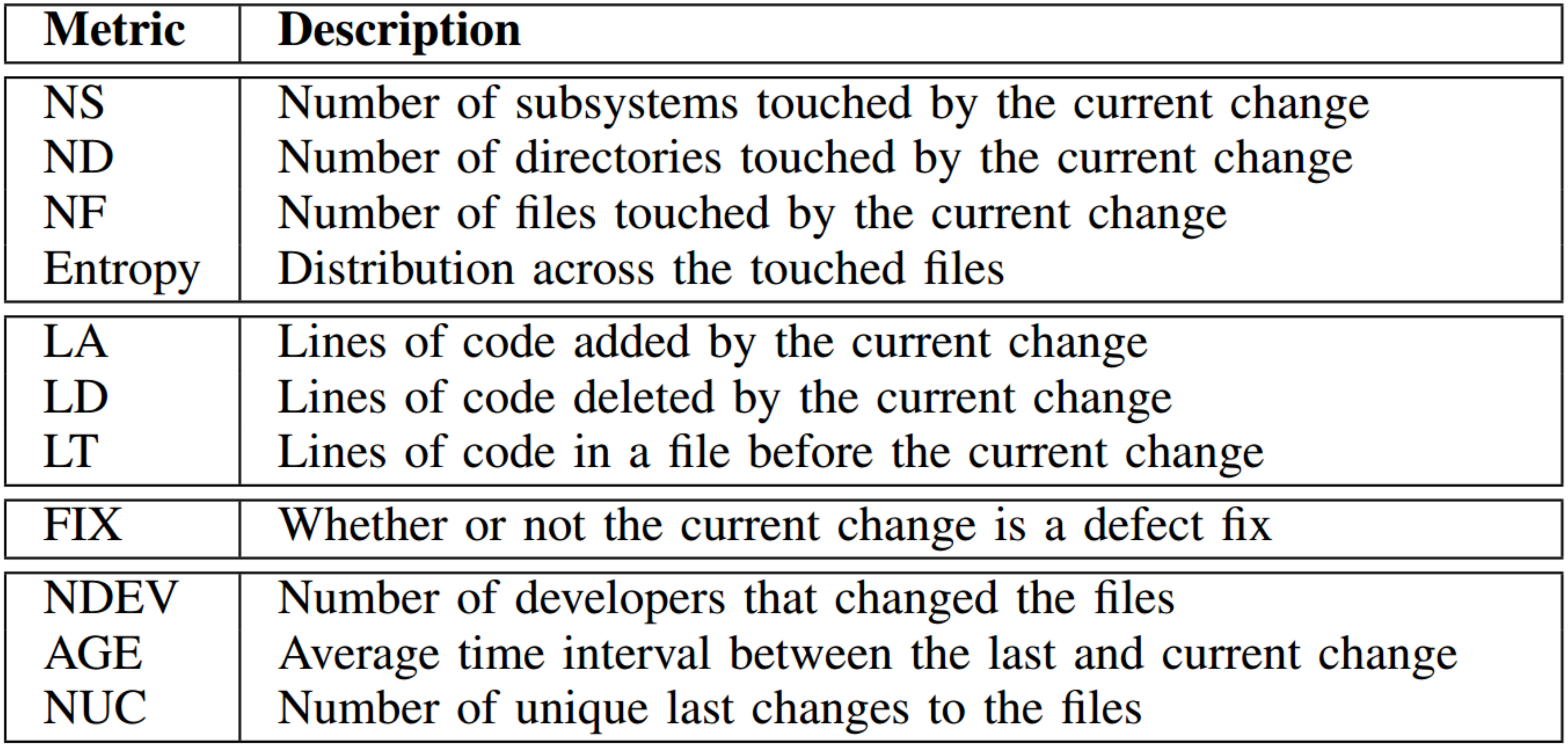
Predict whether a change is buggy

Source Code
History Data

Defect
Prediction
Tool

Supervised JIT Defect Prediction

- We extract a number of metrics from the historical changes with known defective information
- We build a prediction model on these metrics



Metric	**Description**
NS	Number of subsystems touched by the current change
ND	Number of directories touched by the current change
NF	Number of files touched by the current change
Entropy	Distribution across the touched files
LA	Lines of code added by the current change
LD	Lines of code deleted by the current change
LT	Lines of code in a file before the current change
FIX	Whether or not the current change is a defect fix
NDEV	Number of developers that changed the files
AGE	Average time interval between the last and current change
NUC	Number of unique last changes to the files

Unsupervised JIT Defect Prediction Model

- Yang et al. proposed a simple unsupervised defect prediction model
- Simply sort the changes by one metric
  - LT: Lines of code in a file before the current change
- These unsupervised models can detect 30% to 74% more defect-inducing changes than the best supervised model when inspecting 20% LOC

Effort-aware just-in-time defect prediction: simple unsupervised models could be better than supervised models. FSE 2016

Advantages of the unsupervised model:

1. Easy to implement and understand
2. No need to label the data
3. Under the same inspection cost (i.e., [illegible]), it can find more defects

Previous studies on defect prediction made a simple problem too complex!!!

But why do unsupervised defect prediction models perform so well?

Cost Effectiveness

- Given a limited budget (e.g., 20% LOC) in a release, how many bugs one can identify

But none of previous studies report how many changes we need to inspect when inspecting 20% LOC

Number of Changes to Inspect when Inspecting 20% LOC

Approaches	% Bugs	% Chang
Supervised	**31%**	[illegible]
Unsupervised	[illegible]	[illegible]

Unsupervised models need to inspect ~twice as many changes as supervised models.

Do unsupervised models perform better than supervised models?

Domain-Centric Evaluation Measure

- **PCI@20%:** Proportion of Changes Inspected when 20% LOC modified by all changes are inspected
- **IFA:** Number of Initial False Alarms encountered before we find the first defect

The lower the values, the better the performance

Results when Inspecting 20% LOC

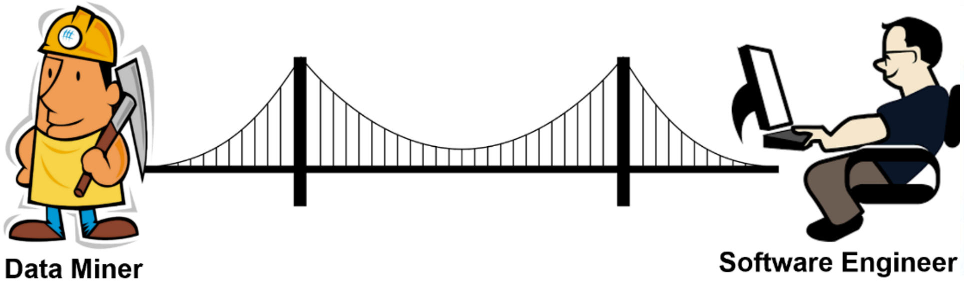
	Supervised	Unsupervised
PCI@20%	**0.33**	**0.60**
IFA	**4**	**70**

Every evaluation measure has its own bias

- We need to design domain centric evaluation measures
- Understanding why we get strong results is more important than how to get strong results

Summary

My Research



Build automated tools by mining and analyzing the rich data in software repositories, to handle the **complexity of software development**

Research Topics

Bug Localization & Repair

Software Artifacts Generation

Human-centric SE

Intelligent Code Recommender

Software Bots

Empirical SE

Example Research

- Automatic **Generation of Pull Request Descriptions**
- **API Method Recommendation** without Worrying About the Task-API Knowledge Gap
- Automating Change-Level **Self-Admitted Technical Debt Determination**
- **Chatbot4QR**: Interactive Query Refinement for Technical Question Retrieval
- **VT-Revolution**: Interactive Programming Video Tutorial Authoring and Watching System

Three Challenges

- **Easy over hard**: simple solution first
- **Strong performance of techniques is not sufficient,** instead a deeper understanding of the domain is essential
- Results should be presented in a **domain-centric context**