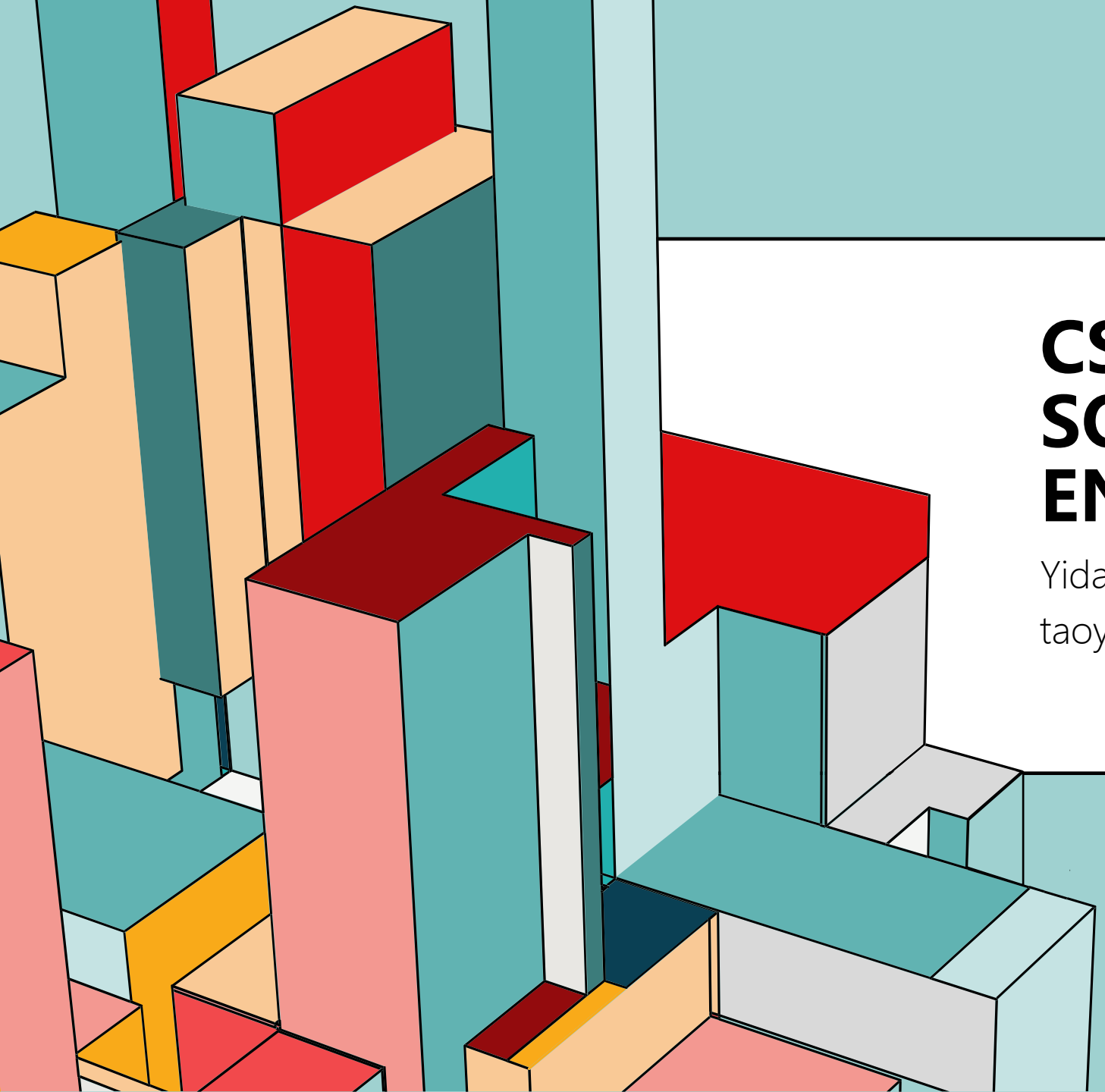


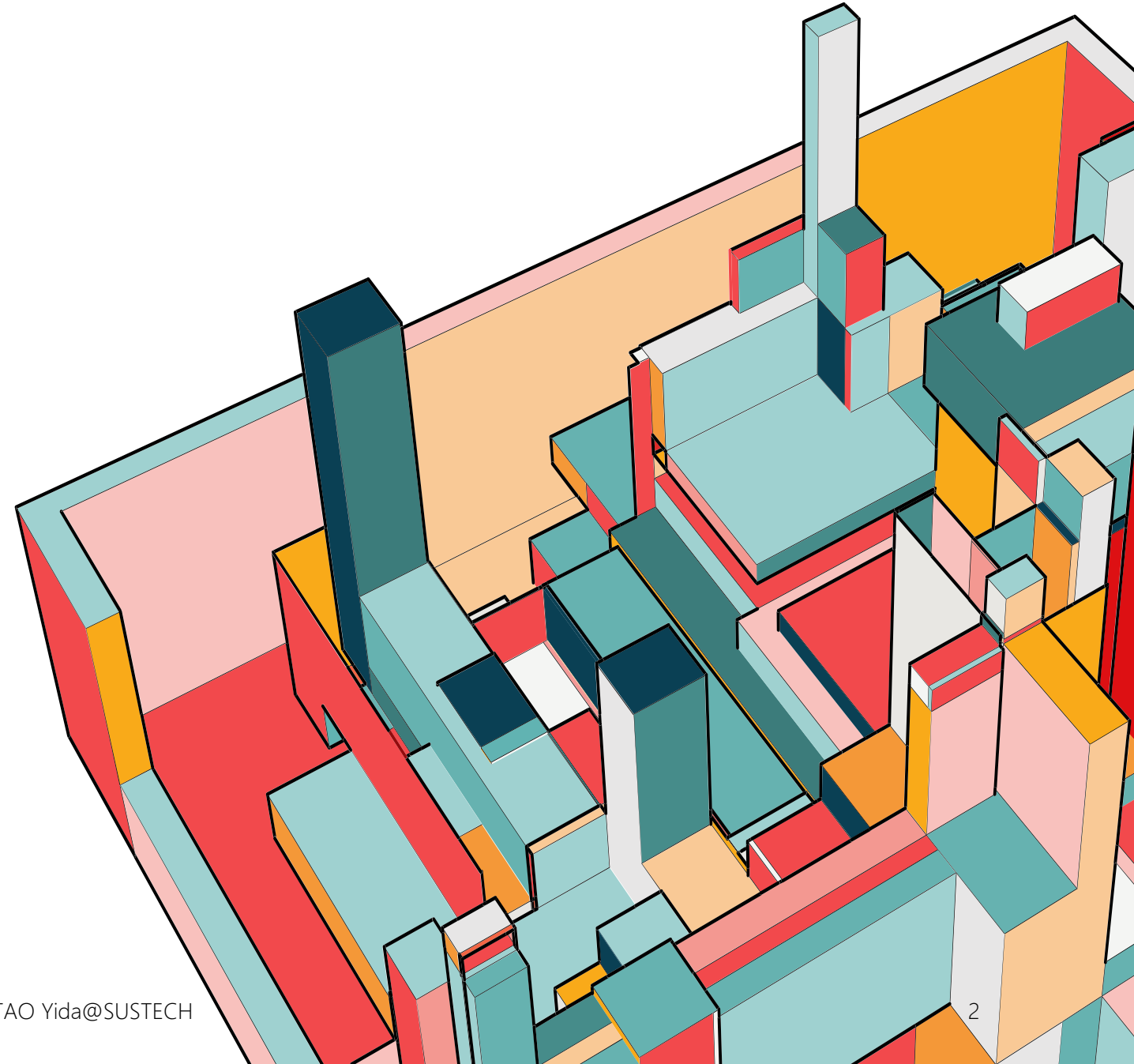
CS304 SOFTWARE ENGINEERING

Yida Tao

taoyd@sustech.edu.cn

LECTURE 13

- AI Overview
- Data & tasks in SE



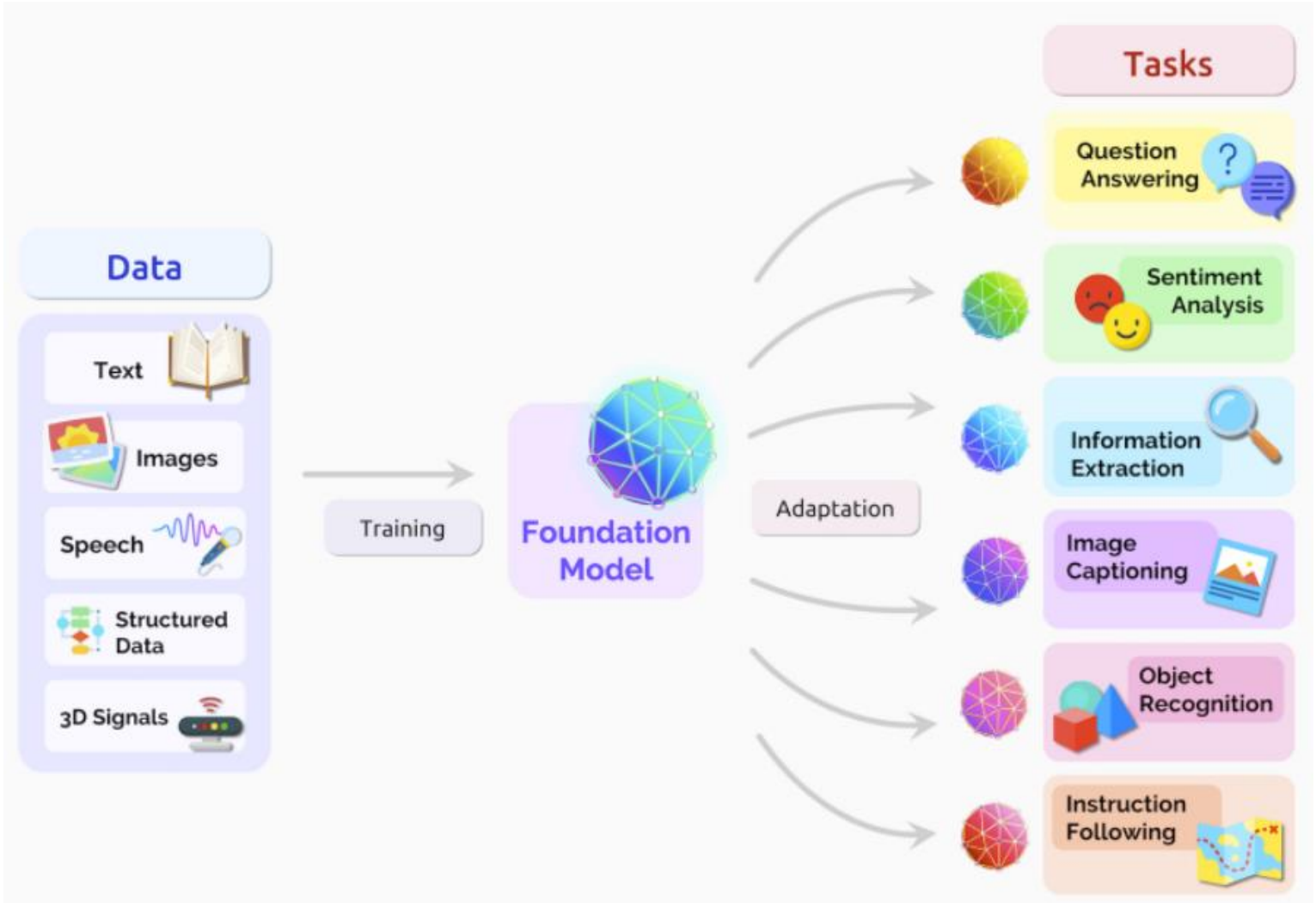
AI OVERVIEW

Documentation
Bug/issue report

Bug snapshots

Code metrics

… Other artifacts



https://research.aimultiple.com/large-language-models/

Bug prediction

Bug fixing

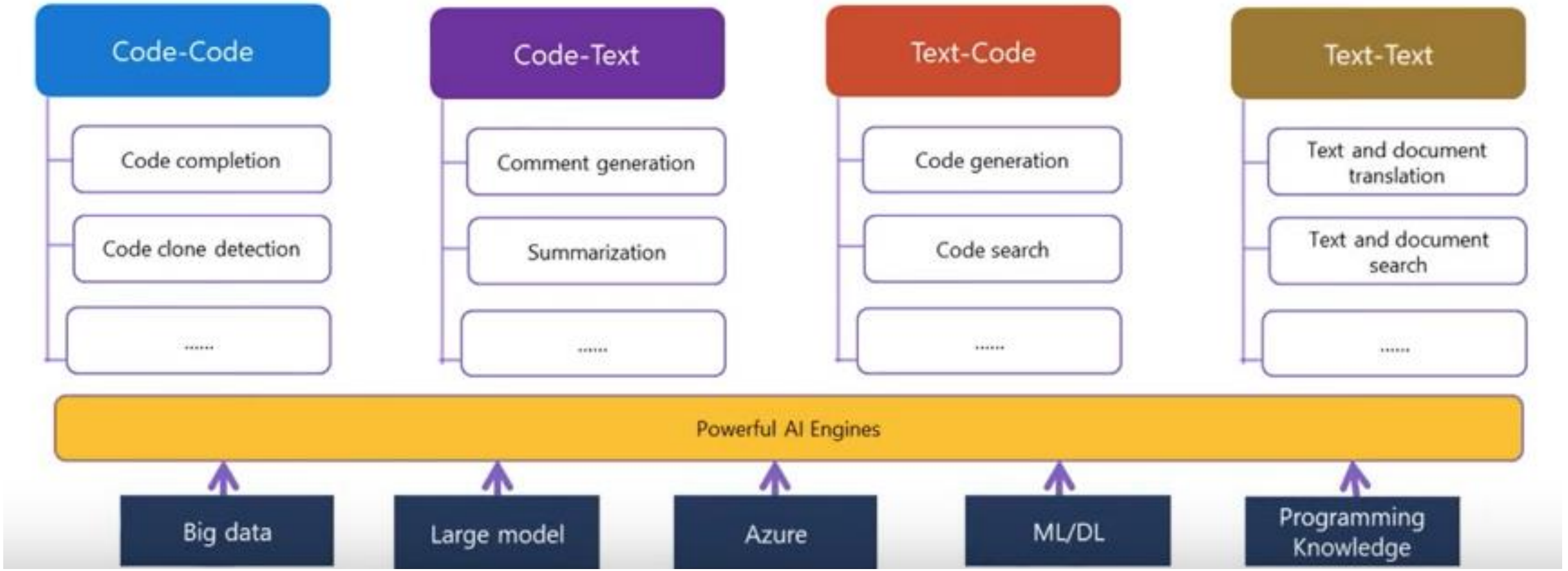
Documentation generation

Code evolution

Test case generation

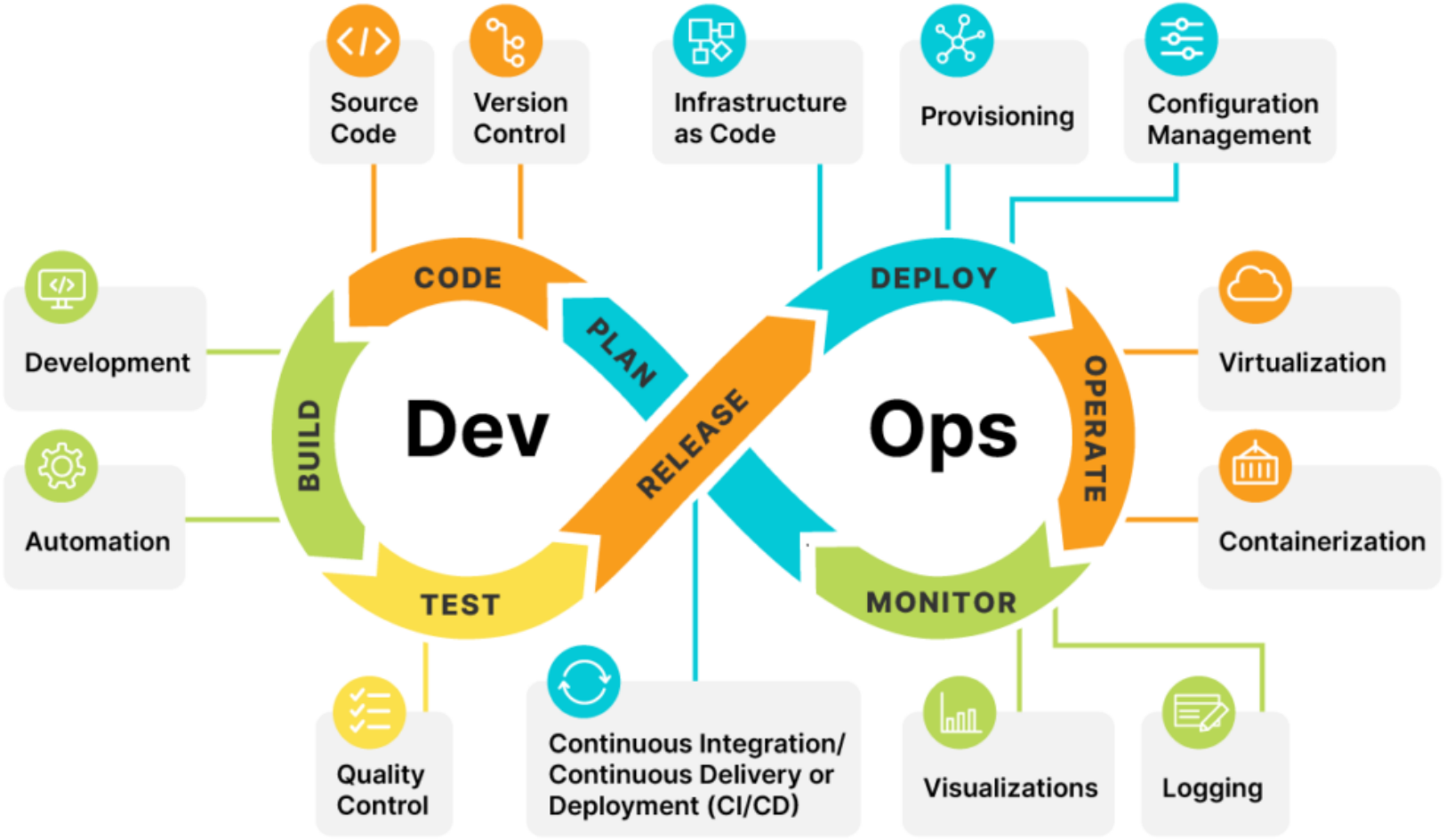
…….

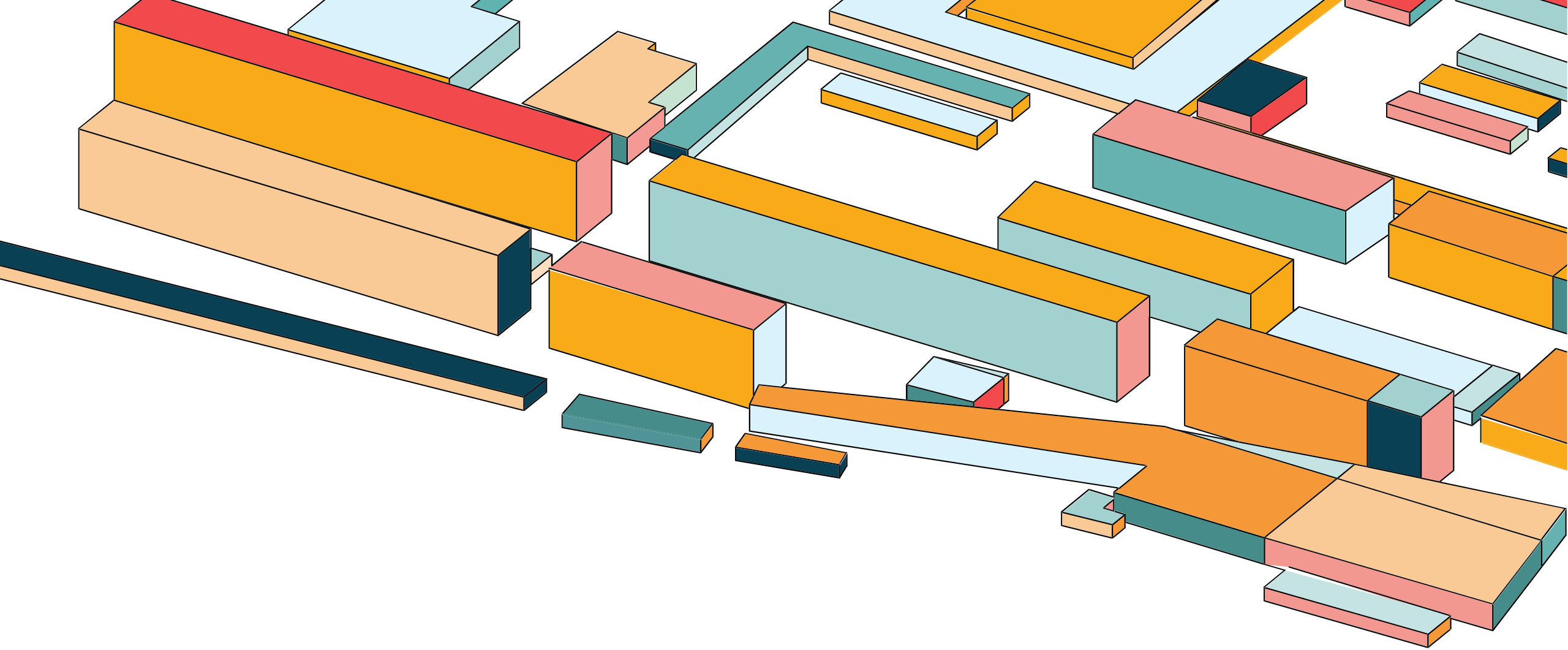
AI FOR SE OVERVIEW



https://indiaai.gov.in/article/understanding-codebert

HOW AI COULD HELP EACH DEVOPS PHASE?





PLANNING, CODDING, AND DEBUGGING

PLAN & DESIGN

Natural language specification -> UML

Towards automatically extracting UML class diagrams from natural language specifications

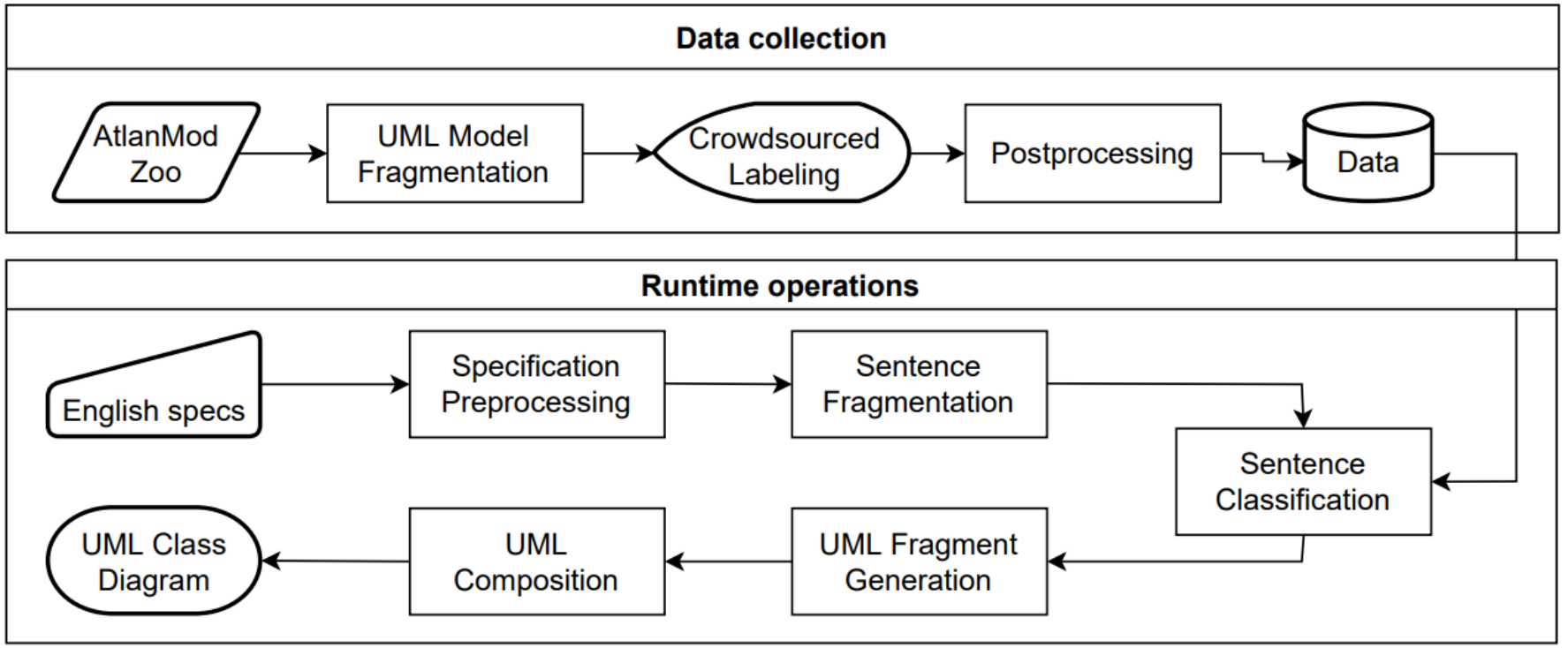
Authors: Song Yang, Houari Sahraoui Authors Info & Claims

MODELS '22: Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings • October 2022 • Pages 396–403 • https://doi.org/10.1145/3550356.3561592

https://dl.acm.org/doi/abs/10.1145/3550356.3561592

PLAN & DESIGN

Natural language specification -> UML



https://dl.acm.org/doi/abs/10.1145/3550356.3561592

CODING AND DEBUGGING

CodeBERT: A Pre-Trained Model for Programming and Natural Languages

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, Ming Zhou

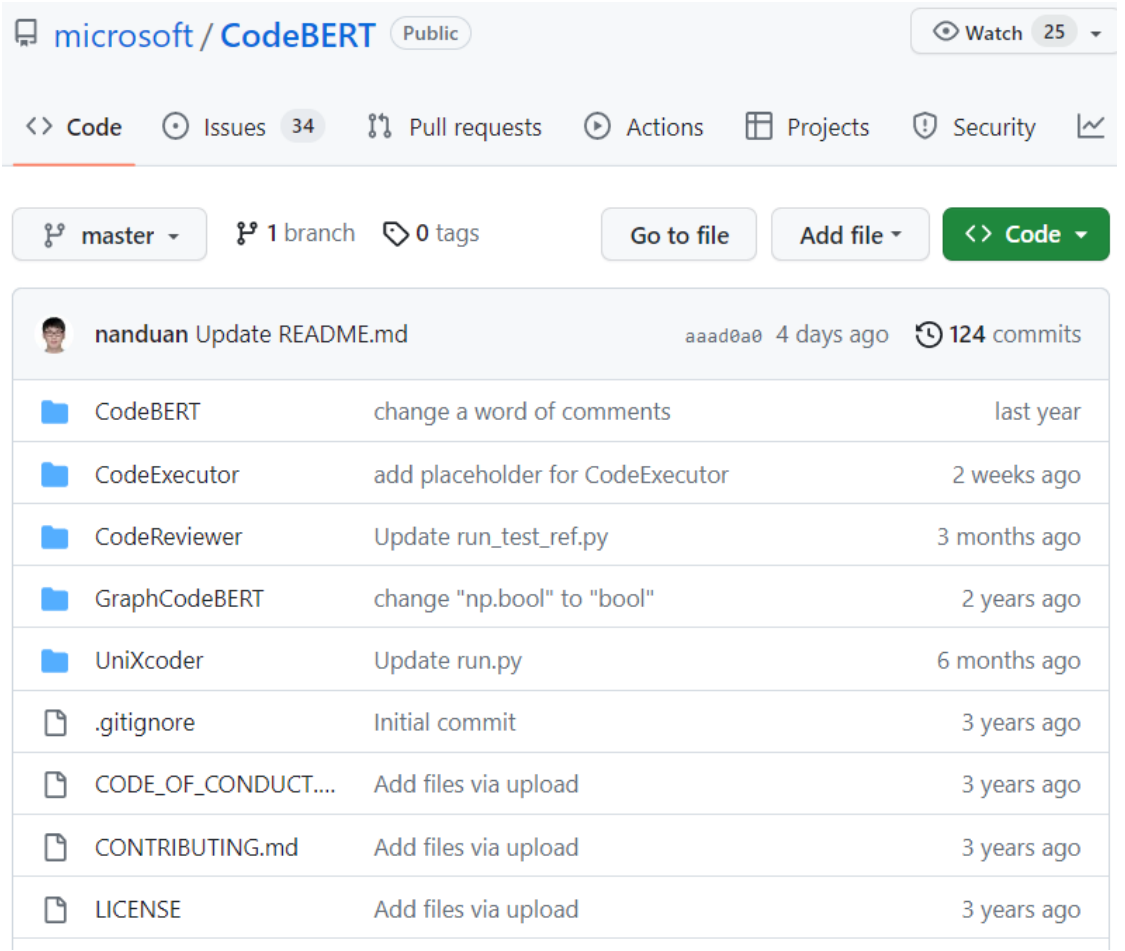
We present CodeBERT, a bimodal pre-trained model for programming language (PL) and nat-ural language (NL). CodeBERT learns general-purpose represen
such as natural language codesearch, code documentation generation, etc. We develop CodeBERT with Transformer-based neural architecture, and train it wi
pre-training task of replaced token detection, which is to detect plausible alternatives sampled from generators. This enables us to utilize both bimodal data of N
provides input tokens for model training while the latter helps to learn better generators. We evaluate CodeBERT on two NL-PL applications by fine-tuning mod
achieves state-of-the-art performance on both natural language code search and code documentation generation tasks. Furthermore, to investigate what type
a dataset for NL-PL probing, and evaluate in a zero-shot setting where parameters of pre-trained models are fixed. Results show that CodeBERT performs bett

https://arxiv.org/abs/2002.08155

CODEBERT

- Training data: GitHub repos. Code paired with NLP documentation
- Model architecture: RoBERTa-base (multi-layer bi-directional transformer)
- Downstream tasks:
  - Code-to-text: generate documentation by code
  - Text-to-code: search code by text

https://arxiv.org/abs/2002.08155



CODEBERT *Text -> Code*

Query

create file and write something

Search Results (top2)

https://github.com/darknessomi/musicbox/blob/master/NEMbox/utils.py#L37-L40

```
def create_file(path, default="\n"):
        if not os.path.exists(path):
            with open(path, "w") as f:
                f.write(default)
```

---

https://github.com/datakortet/yamldirs/blob/master/yamldirs/filemaker.py#L114-L118

```
def make_file(self, filename, content):
        """Create a new file with name ``filename`` and content ``content``.
        """
        with open(filename, 'w') as fp:
            fp.write(content)
```

Given NL, search PL

CODEBERT *Code -> Text*

```
public void addWriteErrorResult(final BulkWriteError writeError,
        final IndexMap indexMap) {
        notNull("writeError", writeError);
        mergeWriteErrors(asList(writeError), indexMap);
    }
```

Gold: Add a write error result

CodeBERT: Add a write error result .

PRE-TRAIN W/ CODEONLY : Merges the given write error .

Roberta: Add a write operation to the map .

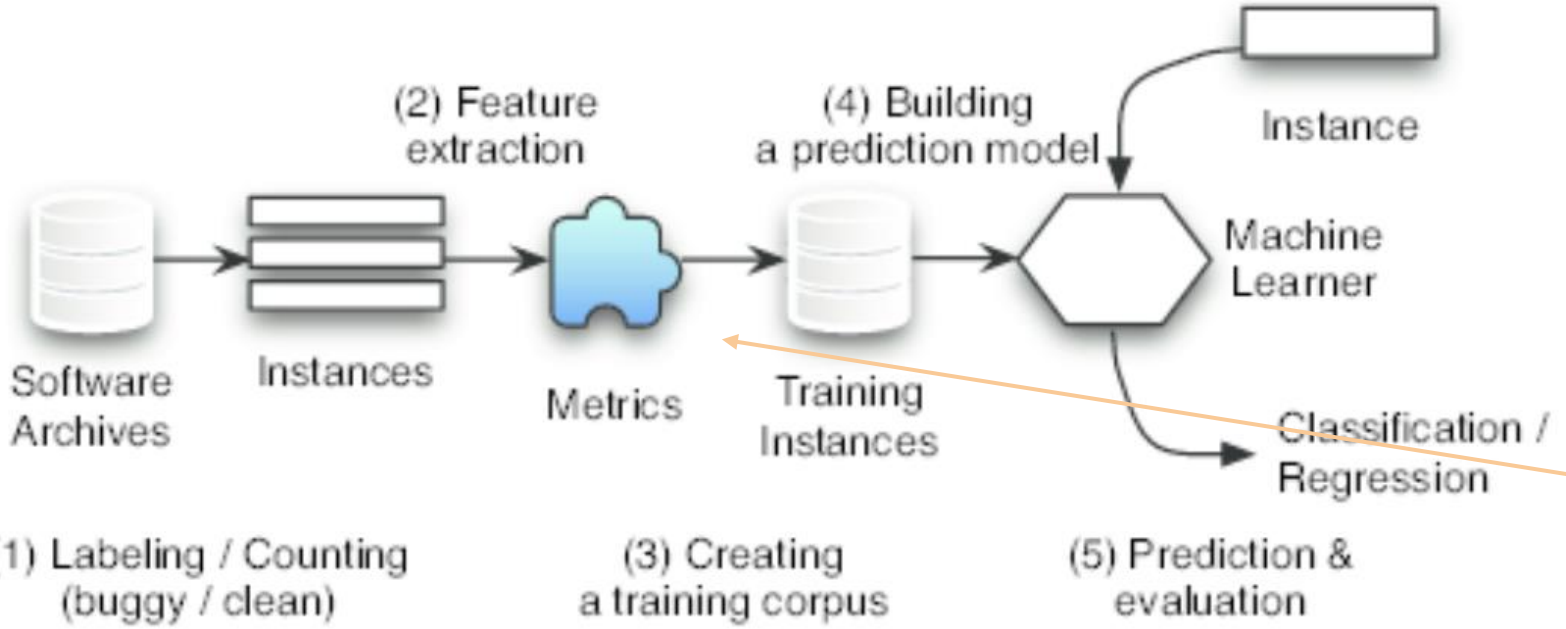
Transformer: Adds an error to the write map .

RNN: Add an error map .

Figure 6: Java code documentation generation output example.

Given PL, get NL

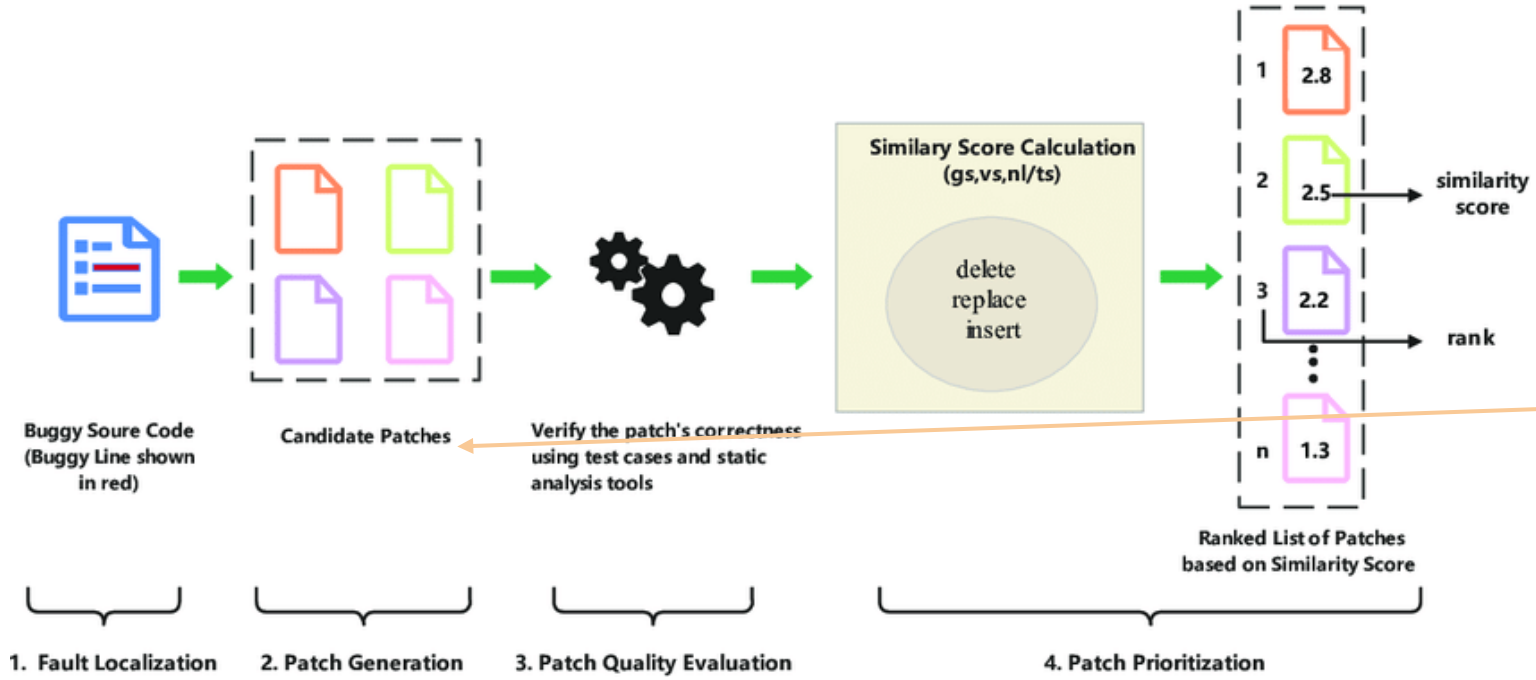
(TRADITIONAL) BUG PREDICTION



Code metrics we've learned!

Overall steps of bug prediction process https://www.researchgate.net/publication/221560578_Micro_interaction_metrics_for_defect_prediction

(TRADITIONAL) AUTOMATED PROGRAM REPAIR (APR)



Patch generation

- Generic algorithm
- Bug/fix Patterns
- ML/DL models

https://www.researchgate.net/publication/362868552_Quality_Evaluation_Method_of_Automatic_Software_Repair_Using_Syntax_Distance_Metrics

AUTOMATED PROGRAM REPAIR... WITH GPT

ChatGPT is Now Fixing Bugs in Code

AI AND DATA SCIENCE NEWS CHATGPT posted by ODSC Team January 31, 2023

ChatGPT 8

Since its public introduction, ChatGPT has become a gar
of the worker, research associate, or both, OpenAI's cha
a new and interesting use for ChatGPT has emerged. Re

arXiv > cs > arXiv:2301.08653

Computer Science > Software Engineering

[Submitted on 20 Jan 2023]

An Analysis of the Automatic Bug Fixing Performance of ChatGPT

Dominik Sobania, Martin Briesch, Carol Hanna, Justyna Petke

To support software developers in finding and fixing software bugs, several automated program repair techniques have been introduced. G
navigate a search space of software edits to find test-suite passing variants. Recent program repair methods are based on deep learning a
for automated program repair, but is still suitable for it, is ChatGPT. The bug fixing performance of ChatGPT, however, is so far unclear. The
benchmark set, QuixBugs, and compare the performance with the results of several other approaches reported in the literature. We find tha
learning approaches CoCoNut and Codex and notably better than the results reported for the standard program repair approaches. In cont

https://arxiv.org/abs/2301.08653

QuixBugs Benchmark 

The QuixBugs benchmark consists of 40 programs from the Quixey Challenge translated into both Python and Java.

Each contains a one-line defect, along with passing (when possible) and failing testcases.

Defects fall into one of 14 defect classes. Corrected Python programs are also supplied.

Defect class	Count
Incorrect assignment operator	1
Incorrect variable	5
Incorect comparison operator	5
Missing condition	2
Missing/added +1	4
Variable swap	6
Incorrect array slice	2
Variable prepend	2
Incorrect data structure constant	2
Incorrect method called	1
Incorrect field dereference	1
Missing arithmetic expression	1
Missing function call	4
Missing line	4

Lin, D., Koppel, J., Chen, A., & Solar-Lezama, A. (2017). QuixBugs: a multi-lingual program repair benchmark set based on the quixey challenge.

COMPARE WITH SOTA REPAIR TECHNIQUES

Benchmark problem	ChatGPT	Codex [15]	CoCoNut [14]	Standard APR [13]
bitcount	✗ (0 / 4)	✓	✓	✗
breadth-first-search	✓ (2 / 4)	✗	✓	✗
bucketsort	✓ (4 / 4)	✓	✓	✗
depth-first-search	✗ (0 / 4)	✓	✗	✗
detect-cycle	✗ (0 / 4)	✗	✗	✓
find-first-in-sorted	✓ (2 / 4)	✓	✓	✗
find-in-sorted	✓ (3 / 4)	✗	✗	✗
flatten	✓ (4 / 4)	✓	✓	✗
gcd	✗ (0 / 4)	✓	✗	✗
.....				
Σ (Solved)	**19**	**21**	**19**	**7**

Reference: https://arxiv.org/abs/2301.08653

PROVIDE MORE INFO

- Exception message

Request (1):

Does this program have a bug? How to fix it?

```
def gcd(a, b):
    if b == 0:
        return a
    else:
        return gcd(a % b, b)
```

Response (1):

This is the Euclidean algorithm for finding the greatest common divisor (GCD) of two integers, a and b. It appears to be implemented correctly. If you are experiencing issues with this program, it is likely that the problem lies elsewhere in your code. Without more information about the bug you are experiencing, it is difficult to say how to fix it.

Request (2):

No, the function is not correct. E.g., for the input a = 4 and b = 5 I get the following error message: "RecursionError: maximum recursion depth exceeded in comparison".

Response (2):

The issue is that the function calls itself recursively, and in some cases, it never reaches the base case (b == 0), causing it to recurse indefinitely. The base case of the gcd algorithm is when the remainder is zero, but in this implementation, it's when b is zero. For example, when a=4 and b=5, gcd(4, 5) is called, which then calls gcd(4 % 5, 5) which is gcd(4, 5). This causes an infinite loop.

You can fix this by swapping the parameters in the recursive call.

```
def gcd(a, b):
    if a == 0:
        return b
    else:
        return gcd(b % a, a)
```

This way, if a is zero, the function returns b (which is the gcd of a and b), otherwise it calls itself with the new values of b % a and a.

Reference: https://arxiv.org/abs/2301.08653

PROVIDE MORE INFO

- Exception message
- Function description
- Buggy behaviors

Request (1):
Does this program have a bug? How to fix it?

```
def bitcount(n):
    count = 0
    while n:
        n ^= n - 1
        count += 1
    return count
```

Response (1):
I'm unable to tell if the program has a bug without more information on the expected behavior and the input that is causing the issue. [...]

Request (2):
The program should count the number of 1-bits in the binary encoding of n.

Response (2):
The function bitcount(n) appears to be counting the number of 1-bits [...]

To confirm this, you can test the function with some test cases to check if it produces the expected output:

```
test_cases = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
expected_output = [0, 1, 1, 2, 1, 2, 2, 3, 1, 2, 2, 3, 2, 3, 3, 4]
```

Request (3):
The function does not work. For input 1 it seems to run into an infinite loop.

Response (3):
You are correct that the function will run into an infinite loop when the input is 1. [...]

Reference: https://arxiv.org/abs/2301.08653

PROVIDE MORE INFO

- Exception message
- Function description
- Buggy behaviors
- ……

By providing such hints to ChatGPT, its success rate can be further increased, fixing 31 out of 40 bugs, outperforming state-of-the-art.

Reference: https://arxiv.org/abs/2301.08653

TABLE II: Results achieved by ChatGPT with additional information given in a follow-up request for the unsolved benchmark problems (see Table I).

Benchmark problem	ChatGPT
bitcount	✓
depth-first-search	✓*
detect-cycle	✓*
gcd	✓
hanoi	✓
kheapsort	✗
kth	✓
lcs-length	✗
levenshtein	✓
lis	✗
longest-common-subsequence	✗
max-sublist-sum	✓
minimum-spanning-tree	✓
next-permutation	✓
powerset	✓
rpn-eval	✗
shortest-path-lengths	✗
sieve	✓*
to-base	✗
topological-ordering	✗
wrap	✗
Σ (Solved)	**9 (12)**

AUTOMATED PROGRAM REPAIR... WITH GPT

arXiv > cs > arXiv:2304.00385

Computer Science > Software Engineering

[Submitted on 1 Apr 2023]

Keep the Conversation Going: Fixing 162 out of 337 bugs for $0.42 each using ChatGPT

Chunqiu Steven Xia, Lingming Zhang

Automated Program Repair (APR) aims to automatically generate patches for buggy programs. Recent APR work has been focused on leveraging modern Large Lang
first constructing an input prompt built using the original buggy code and then queries the LLM to generate patches. While the LLM-based APR tools are able to achiev
generating lots of patches and then validating each one afterwards. This not only leads to many repeated patches that are incorrect but also miss the crucial informatic
To address these limitations, we propose ChatRepair, the first fully automated conversation-driven APR approach that interleaves patch generation with instant feedba
failure information to start with, and then learns from both failures and successes of earlier patching attempts of the same bug for more powerful APR. For earlier patcl
test failure information to construct a new prompt for the LLM to generate the next patch. In this way, we can avoid making the same mistakes. For earlier patches that
plausible patches. In this way, we can further build on and learn from earlier successes to generate more plausible patches to increase the chance of having correct pa
based LLM -- ChatGPT. By calculating the cost of accessing ChatGPT, we can fix 162 out of 337 bugs for $0.42 each!

Subjects: **Software Engineering (cs.SE)**; Machine Learning (cs.LG)
Cite as: arXiv:2304.00385 **[cs.SE]**
(or arXiv:2304.00385v1 **[cs.SE]** for this version)
https://doi.org/10.48550/arXiv.2304.00385

TESTS ARE INFORMATIVE..... FOR LLMS

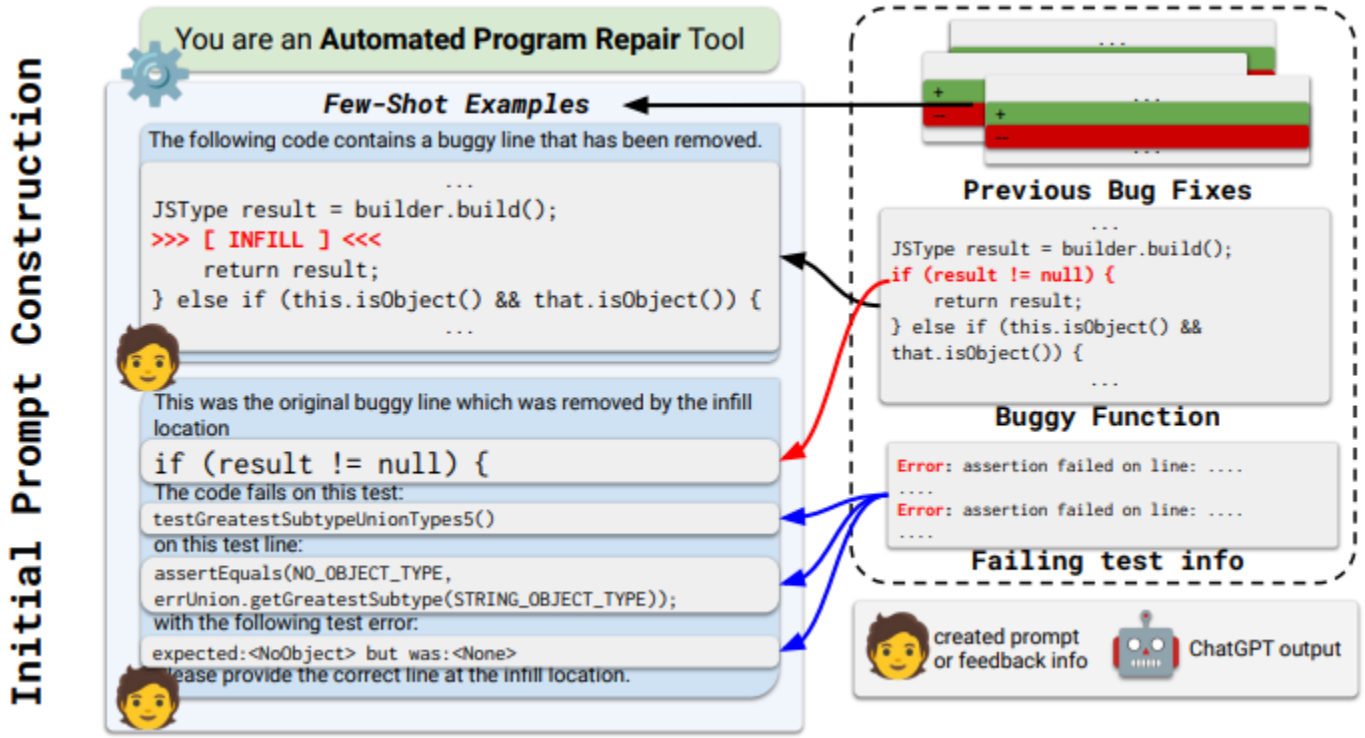
Testname: `testZero()`
Failure Line: `assertPrint("var x ='\\0';", "var x=\"\\000\"");`
Error Message: `expected:<var x="\0[00]"> but was:<var x="\0[]">`

```
   switch (c) {
-      case '\0': sb.append("\\0"); break;
+      case '\0': sb.append("\\000"); break;
       case '\n': sb.append("\\n"); break;
```

Test name, failure line, and error message offer valuable info to LLMs like ChatGPT

Xia, Chunqiu Steven, and Lingming Zhang. "Keep the Conversation Going: Fixing 162 out of 337 bugs for $0.42 each using ChatGPT." arXiv preprint arXiv:2304.00385 (2023).

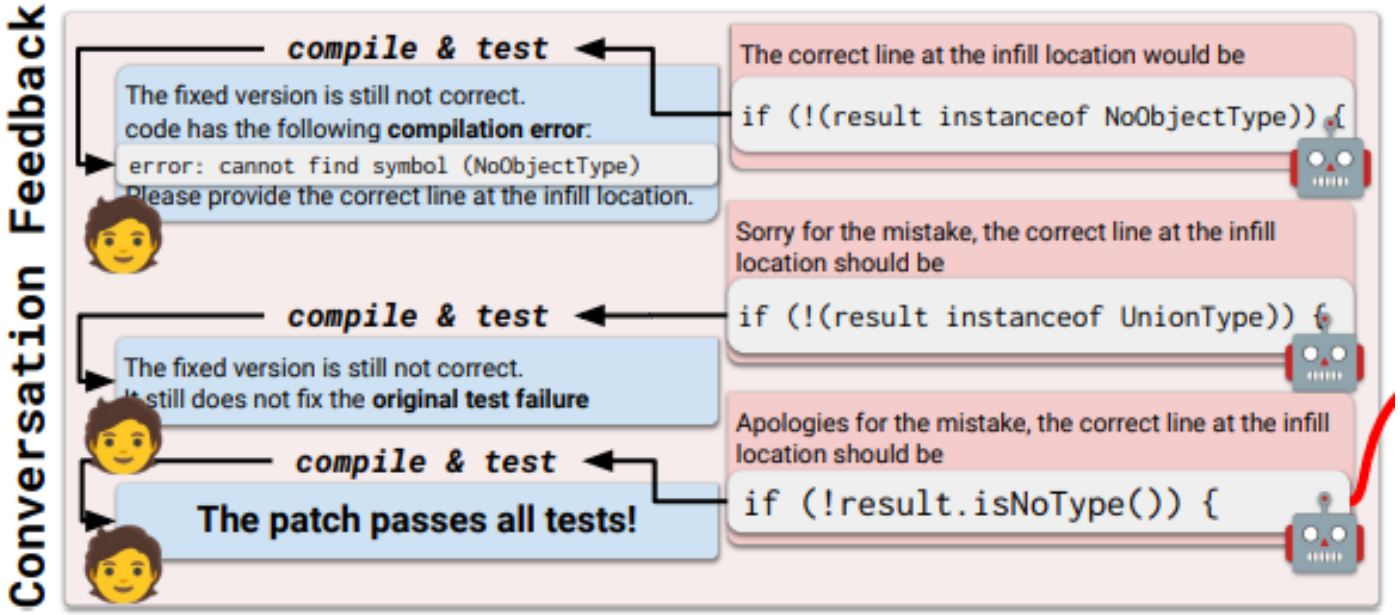
INITIAL INPUT



Ask LLM to fix a bug with the same formatted input, including test information

Xia, Chunqiu Steven, and Lingming Zhang. "Keep the Conversation Going: Fixing 162 out of 337 bugs for $0.42 each using ChatGPT." arXiv preprint arXiv:2304.00385 (2023).

CONVERSATIONAL REPAIR

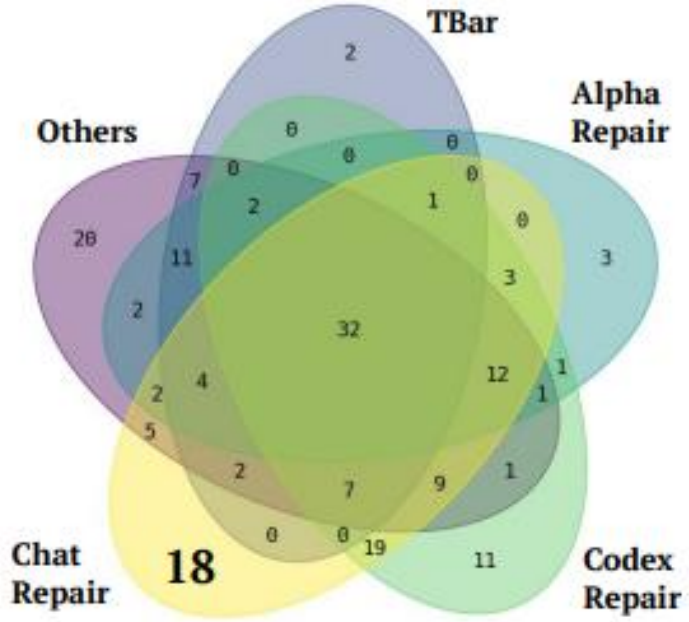


Iteratively

- Take the repair generated by LLM
- Compile & run test suite
- Generate test execution feedback to LLM

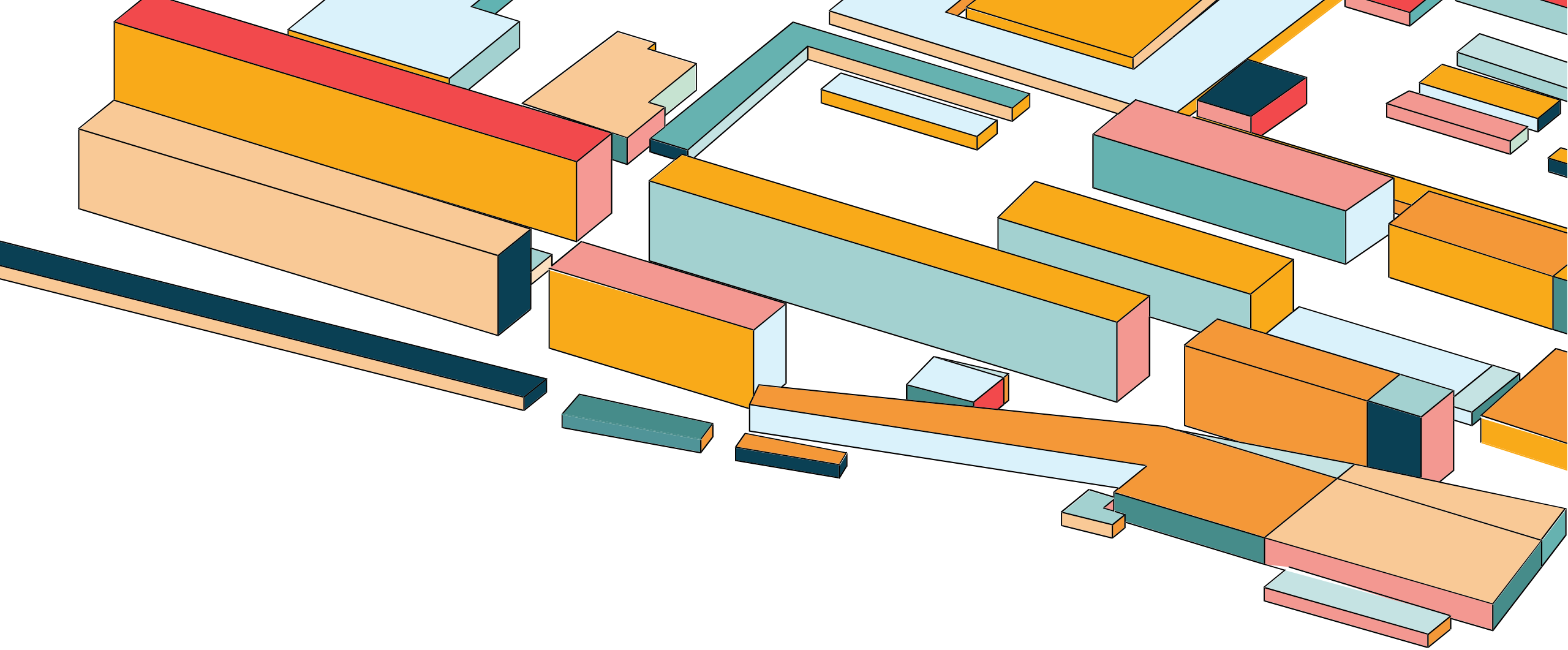
Xia, Chunqiu Steven, and Lingming Zhang. "Keep the Conversation Going: Fixing 162 out of 337 bugs for $0.42 each using ChatGPT." arXiv preprint arXiv:2304.00385 (2023).

RESULTS



Dataset	ChatRepair	BaseChatGPT
Chart	15	9
Closure	37	23
Lang	21	15
Math	32	25
Mockito	6	6
Time	3	2
D4J 1.2	**114**	80
D4J 2.0	**48**	25

Xia, Chunqiu Steven, and Lingming Zhang. "Keep the Conversation Going: Fixing 162 out of 337 bugs for $0.42 each using ChatGPT." arXiv preprint arXiv:2304.00385 (2023).



EVOLUTION AND MAINTENANCE

API MIGRATION - MOTIVATION

- A software product is often required to support a variety of devices and environments.
- This requires **developing the software product in one language and manually porting it to other languages**.
- The procedure is tedious, time-consuming, and error-prone

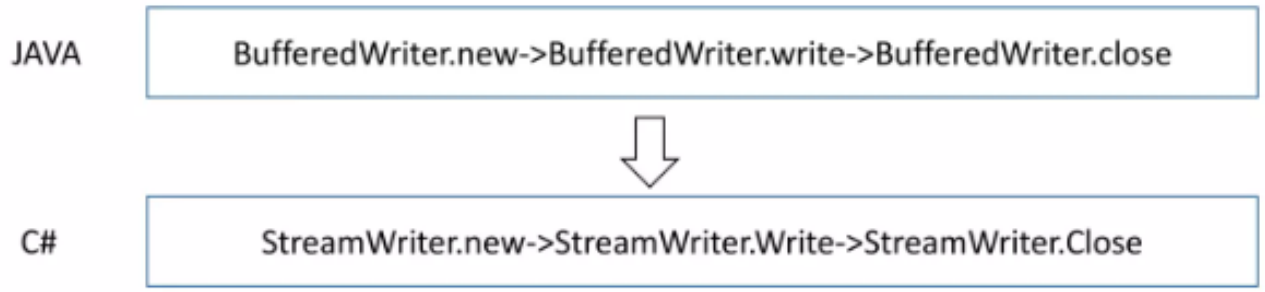
Reference: Gu Xiaodong, Hongyu Zhang, Dongmei Zhang, and Sunghun Kim. "DeepAM: Migrate APIs with multi-modal sequence to sequence learning." arXiv preprint arXiv:1704.07734 (2017).

API MIGRATION - MOTIVATION

```
BufferedWriter bw=new BufferedWriter();
bw.write();
bw.close();
```

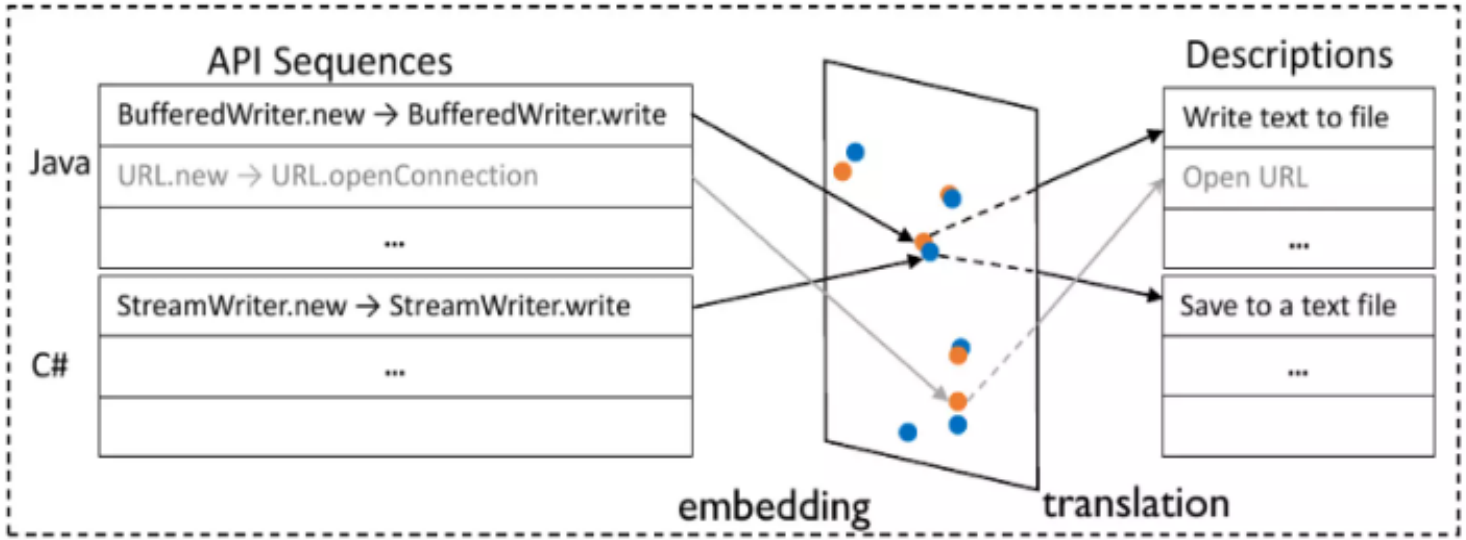
⇨

```
StreamWriter sw=new StreamWriter();
sw.Write();
sw.Close();
```



Reference: Gu Xiaodong, Hongyu Zhang, Dongmei Zhang, and Sunghun Kim. "DeepAM: Migrate APIs with multi-modal sequence to sequence learning." arXiv preprint arXiv:1704.07734 (2017).

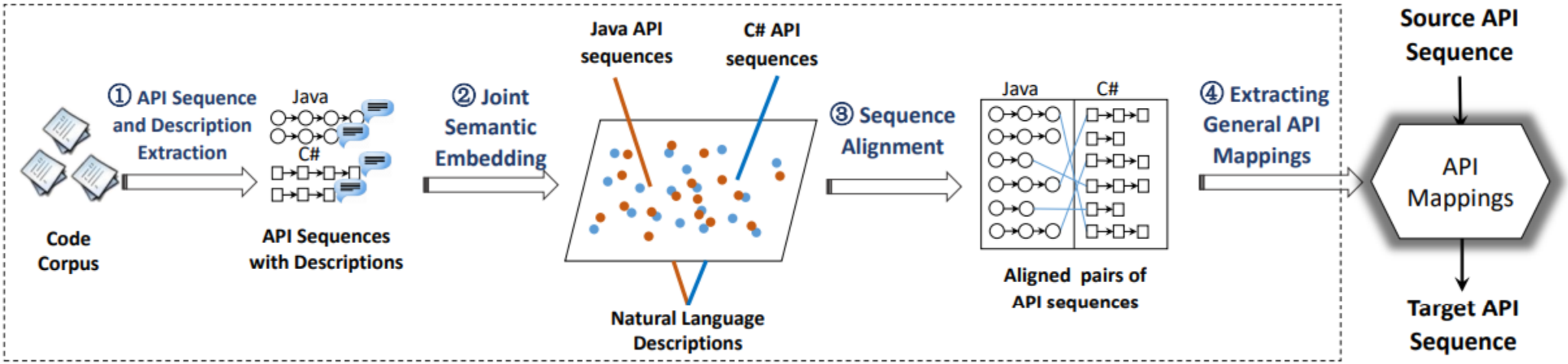
API MIGRATION - APPROACH



DEEPAM assigns to each API sequence a continuous vector in a high-dimensional semantic space in such a way that API sequences with similar vectors, or “embeddings”, tend to have similar natural language descriptions.

Reference: Gu Xiaodong, Hongyu Zhang, Dongmei Zhang, and Sunghun Kim. "DeepAM: Migrate APIs with multi-modal sequence to sequence learning." arXiv preprint arXiv:1704.07734 (2017).

API MIGRATION - APPROACH



Reference: Gu Xiaodong, Hongyu Zhang, Dongmei Zhang, and Sunghun Kim. "DeepAM: Migrate APIs with multi-modal sequence to sequence learning." arXiv preprint arXiv:1704.07734 (2017).

API MIGRATION - RESULTS

parse datetime from string	
SimpleDateFormat.new→SimpleDateFormat.parse	DateTimeFormatInfo.new→ DateTime.parseExact → DateTime.parse
open a url	
URL.new→ URL.openConnection	WebRequest.create → Uri.new → HttpWebRequest.getRequestStream
get files in folder	
File.new→File.list→File.new→File.isDirectory	DirectoryInfo.new→ DirectoryInfo.getDirectories
create a directory	
File.new→File.exists→File.createNewFile	FileInfo.new→Directory.exists→Directory.createDirectory

Reference: Gu Xiaodong, Hongyu Zhang, Dongmei Zhang, and Sunghun Kim. "DeepAM: Migrate APIs with multi-modal sequence to sequence learning." arXiv preprint arXiv:1704.07734 (2017).

AUTO-UPDATE LIBRARY DEPENDENCIES

Table 1: Migration path of the *org.apache.hadoop:hadoop-auth* repository.

Client version	lib_1	lib_2	lib_3	lib_4	Timestamp
2.0.2-alpha	1.2.17	1.6.1	0	1.4	2012-10-02T00:44:04
2.3.0	1.2.17	1.7.5	4.2.5	1.4	2014-02-11T13:55:58
2.4.1	0	1.7.5	4.2.5	1.4	2014-06-21T06:08:34
2.5.1	0	0	4.2.5	0	2014-09-05T23:05:15
2.6.0	1.2.17	0	4.3.1	0	2014-11-13T22:35:37
2.7.2	1.2.17	1.7.10	4.2.5	1.4	2016-01-14T21:32:14
3.0.0-alpha3	1.2.17	1.7.10	4.5.2	1.4	2017-05-26T20:39:35
*	?	?	?	?	

- A software may depend on multiple 3rd-party libraries (TPL)
- Cannot simply upgrade to the next or latest version
- May involve library removal or downgrading
- It's complex to upgrade TPL versions without introducing compatibility issues

Nguyen, Phuong T., et al. "DeepLib: Machine translation techniques to recommend upgrades for third-party libraries." Expert Systems with Applications 202 (2022): 117267.

AUTO-UPDATE LIBRARY DEPENDENCIES

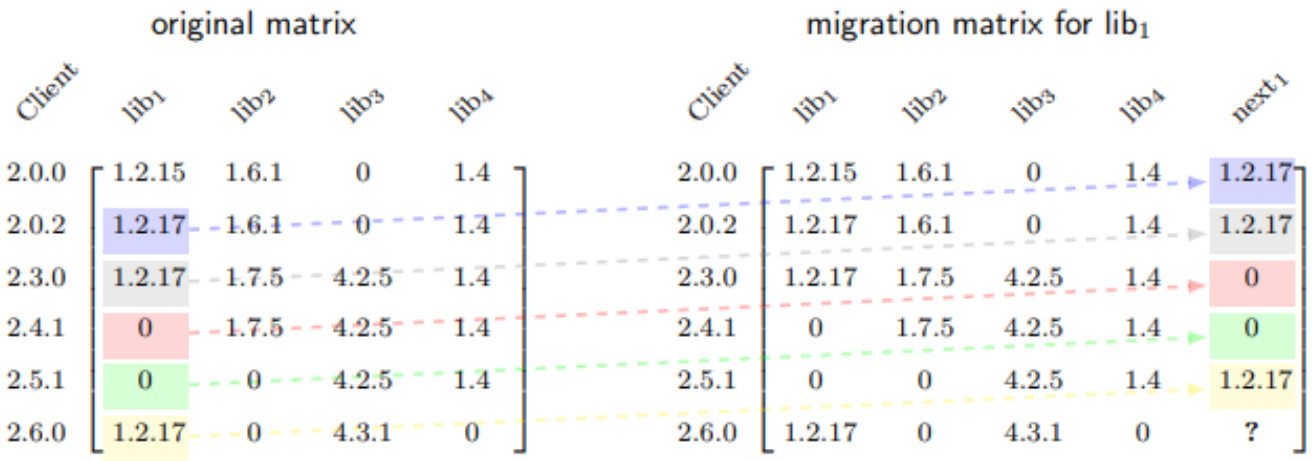
Table 1: Migration path of the *org.apache.hadoop:hadoop-auth* repository.

Client version	lib_1	lib_2	lib_3	lib_4	Timestamp
2.0.2-alpha	1.2.17	1.6.1	0	1.4	2012-10-02T00:44:04
2.3.0	1.2.17	1.7.5	4.2.5	1.4	2014-02-11T13:55:58
2.4.1	0	1.7.5	4.2.5	1.4	2014-06-21T06:08:34
2.5.1	0	0	4.2.5	0	2014-09-05T23:05:15
2.6.0	1.2.17	0	4.3.1	0	2014-11-13T22:35:37
2.7.2	1.2.17	1.7.10	4.2.5	1.4	2016-01-14T21:32:14
3.0.0-alpha3	1.2.17	1.7.10	4.5.2	1.4	2017-05-26T20:39:35
*	?	?	?	?	

- A DL model can learn upgrades for TPLs that have been performed by similar clients.
- Such upgrades are considered safe, i.e., they do not trigger any conflict, since in the training clients, the libraries already coexist without causing any compatibility or dependency issues

Nguyen, Phuong T., et al. "DeepLib: Machine translation techniques to recommend upgrades for third-party libraries." Expert Systems with Applications 202 (2022): 117267.

AUTO-UPDATE LIBRARY DEPENDENCIES



- Data is collected from GitHub and Maven repository
- Migration matrices are built, normalized, and used as training data
- “next” is the label to be predicted
- LSTM model is used

Nguyen, Phuong T., et al. "DeepLib: Machine translation techniques to recommend upgrades for third-party libraries." Expert Systems with Applications 202 (2022): 117267.

AUTO-UPDATE LIBRARY DEPENDENCIES

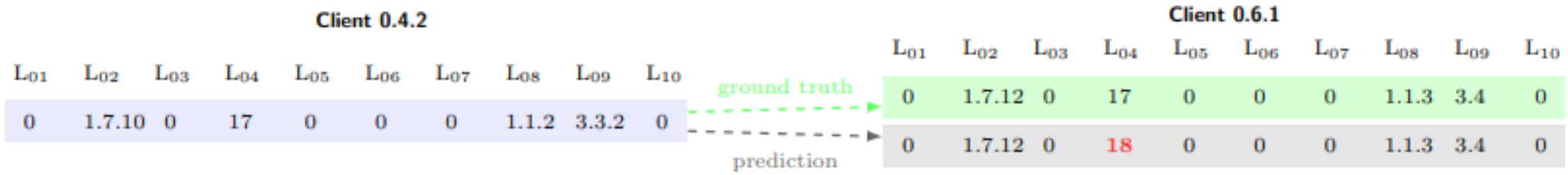
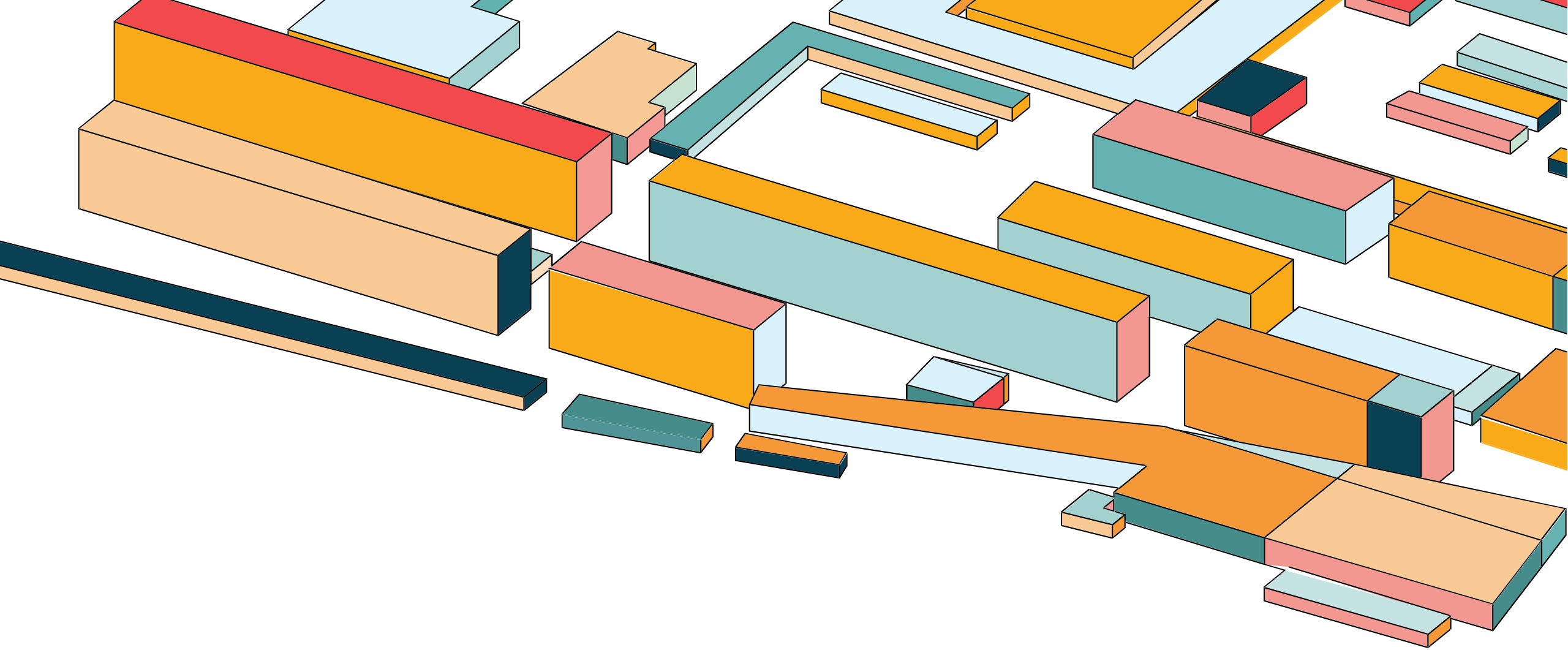


Figure 12: Recommendation for the *com.hubspot:SingularityService* repository.

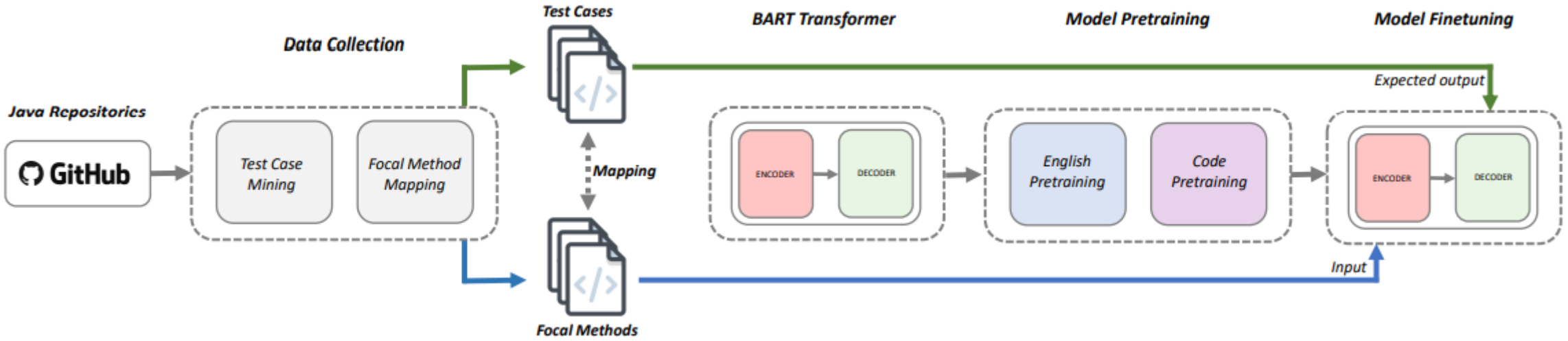
Nguyen, Phuong T., et al. "DeepLib: Machine translation techniques to recommend upgrades for third-party libraries." Expert Systems with Applications 202 (2022): 117267.



SOFTWARE TESTING

AUTO-GENERATING UNIT TESTS

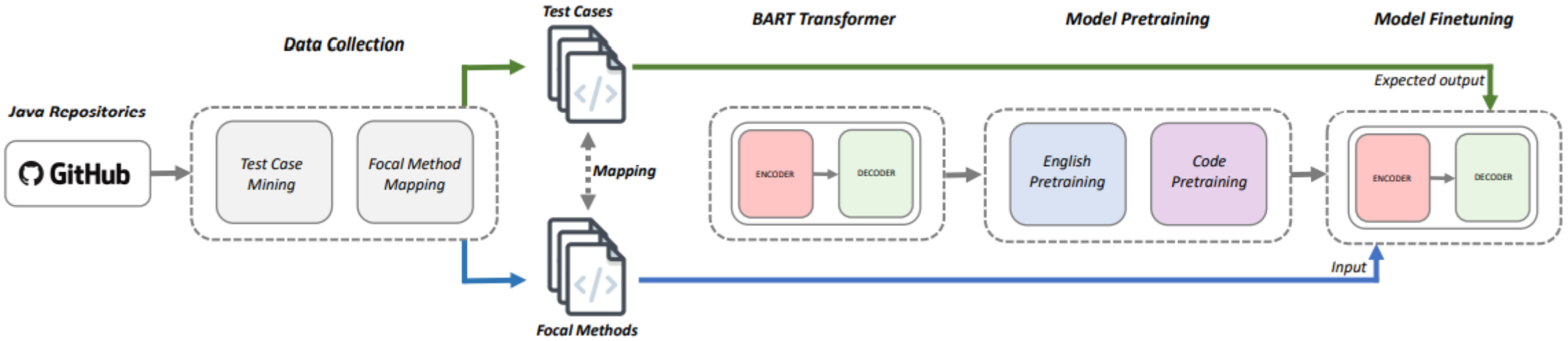
- Task: given a focal method (the method to be tested by the test case), generate a set of test cases
- Idea: learn from large test cases and corresponding focal methods



Tufano, Michele, et al. "Unit test case generation with transformers and focal context." arXiv preprint arXiv:2009.05617 (2020).

AUTO-GENERATING UNIT TESTS

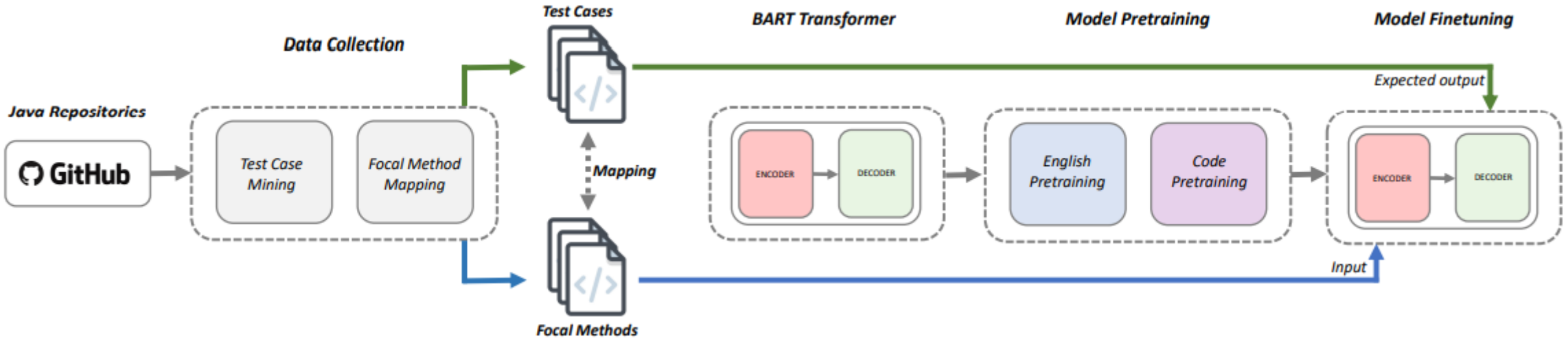
- Data: 91K Java projects from GitHub
- Identify and pair tests & focal methods using heuristics such as name matching



Tufano, Michele, et al. "Unit test case generation with transformers and focal context." arXiv preprint arXiv:2009.05617 (2020).

AUTO-GENERATING UNIT TESTS

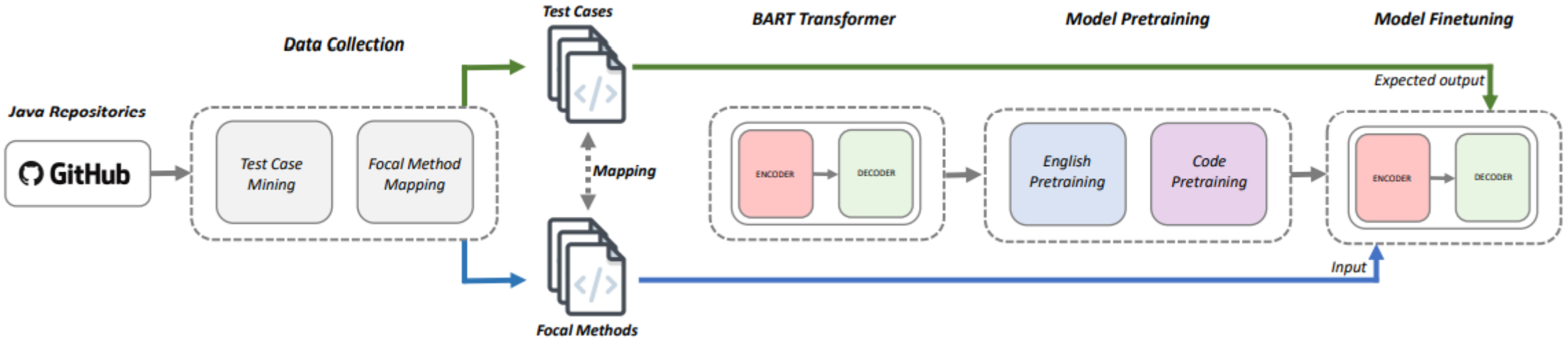
- BART transformer model
- English pretraining: 160GB English text from books, Wikipedia, and news articles
- Code pretraining: 25GB Java projects from GitHub



Tufano, Michele, et al. "Unit test case generation with transformers and focal context." arXiv preprint arXiv:2009.05617 (2020).

AUTO-GENERATING UNIT TESTS

- Finetuning: the source is a focal method, and the target is the corresponding test case originally written by a software developer.



Tufano, Michele, et al. "Unit test case generation with transformers and focal context." arXiv preprint arXiv:2009.05617 (2020).

AUTO-GENERATING UNIT TESTS

- AthenaTest outperforms GPT-3 w.r.t. code coverage, and outperforms EvoSuite w.r.t. readability

Focal Method

```
public static boolean isDigits(final String str) {
        if (StringUtils.isEmpty(str)) {
            return false;
        }
        for (int i = 0; i < str.length(); i++) {
            if (!Character.isDigit(str.charAt(i))) {
                return false;
            }
        }
        return true;
}
```

GPT-3

```
@Test public void testIsDigits() {
    assertTrue(NumberUtils.isDigits("100"));
    assertFalse(NumberUtils.isDigits("1a"));
}
```

ATHENATEST

```
@Test public void isDigits() {
     assertTrue(NumberUtils.isDigits("123"));
     assertFalse(NumberUtils.isDigits("-123"));
     assertFalse(NumberUtils.isDigits("1a2"));
     assertFalse(NumberUtils.isDigits("a0b-2"));
     assertFalse(NumberUtils.isDigits("abcax"));
     assertFalse(NumberUtils.isDigits("1,2,3"));
}
```

Tufano, Michele, et al. "Unit test case generation with transformers and focal context." arXiv preprint arXiv:2009.05617 (2020).

GPT-POWERED UI AUTOMATION FOR ANDROID

Given a natural language description, DroidBot-GPT can automatically generate and execute actions that navigate the Android app to complete the task (Useful in automated testing)

Wen, Hao, et al. "DroidBot-GPT: GPT-powered UI Automation for Android." arXiv preprint arXiv:2304.07061 (2023).

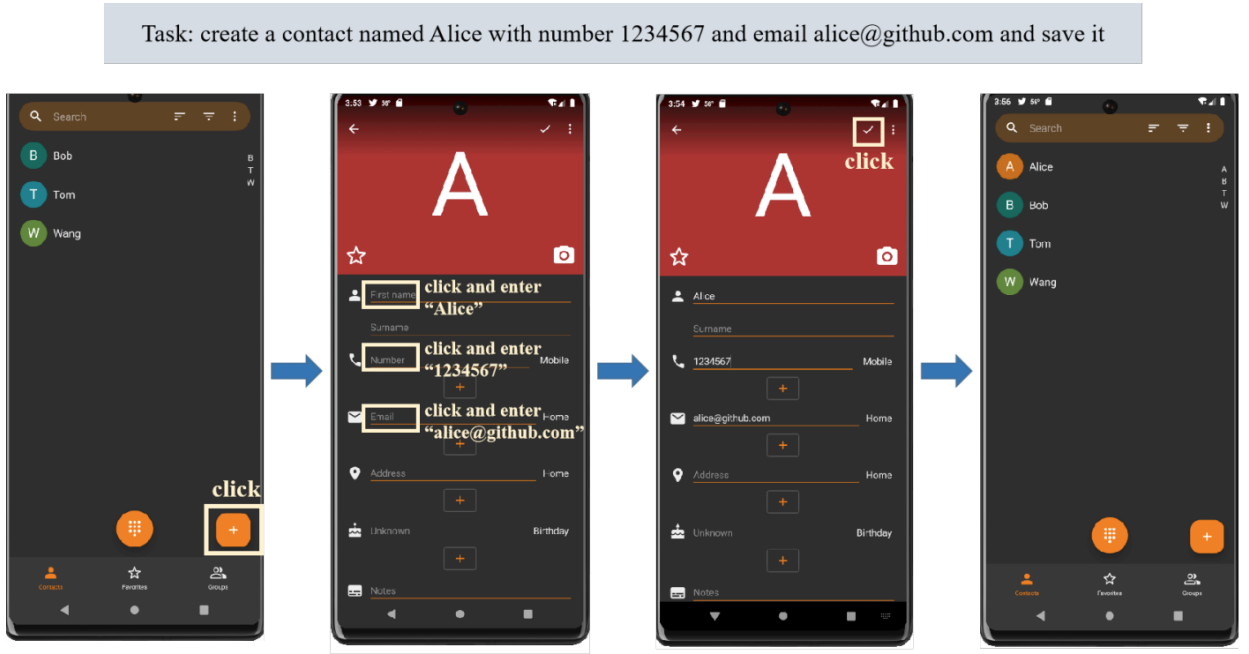


Figure 1: An illustration of task automation with DroidBot-GPT. The task is to use the "Simple Contacts Pro" app to complete the task "create a contact named Alice with number 1234567 and email alice@github.com".

GPT-POWERED UI AUTOMATION FOR ANDROID

1. Given an Android app and a task described by the user, DroidBot-GPT first fetches the state of the app and describes it in natural language.
2. Then it combines the state information, the action history, and the task into a prompt, and sends it to ChatGPT.
3. ChatGPT generates the sends back the proper action
4. DroidBot-GPT sends operations to be executed on the phone.

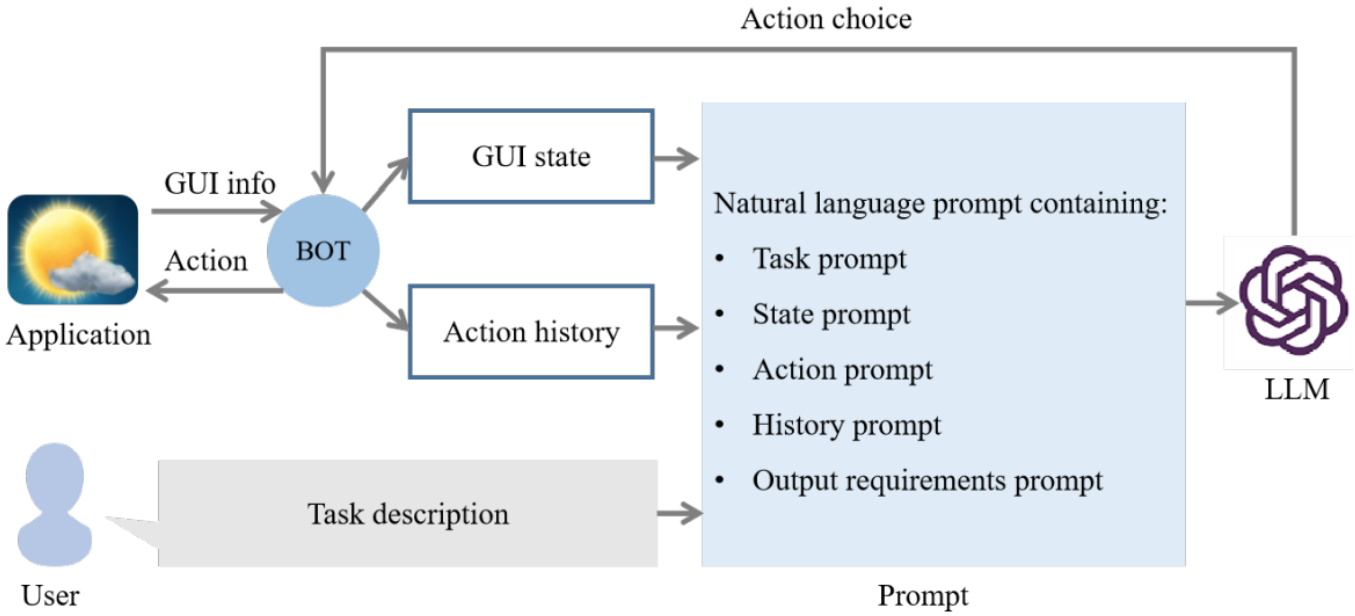
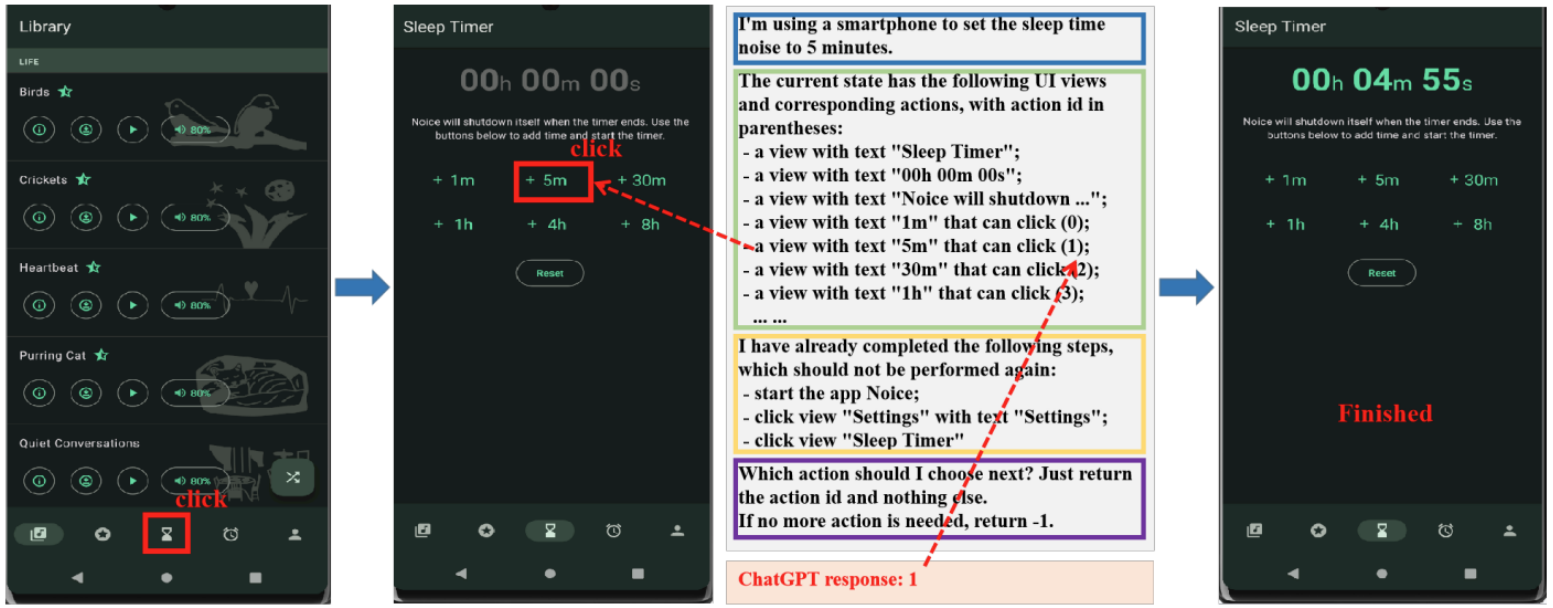


Figure 2: The workflow of DroidBot-GPT.

Wen, Hao, et al. "DroidBot-GPT: GPT-powered UI Automation for Android." arXiv preprint arXiv:2304.07061 (2023).

GPT-POWERED UI AUTOMATION FOR ANDROID



- (Blue) Task
- (Yellow) Action history
- (Green) GUI elements
- (Purple) Output requirements

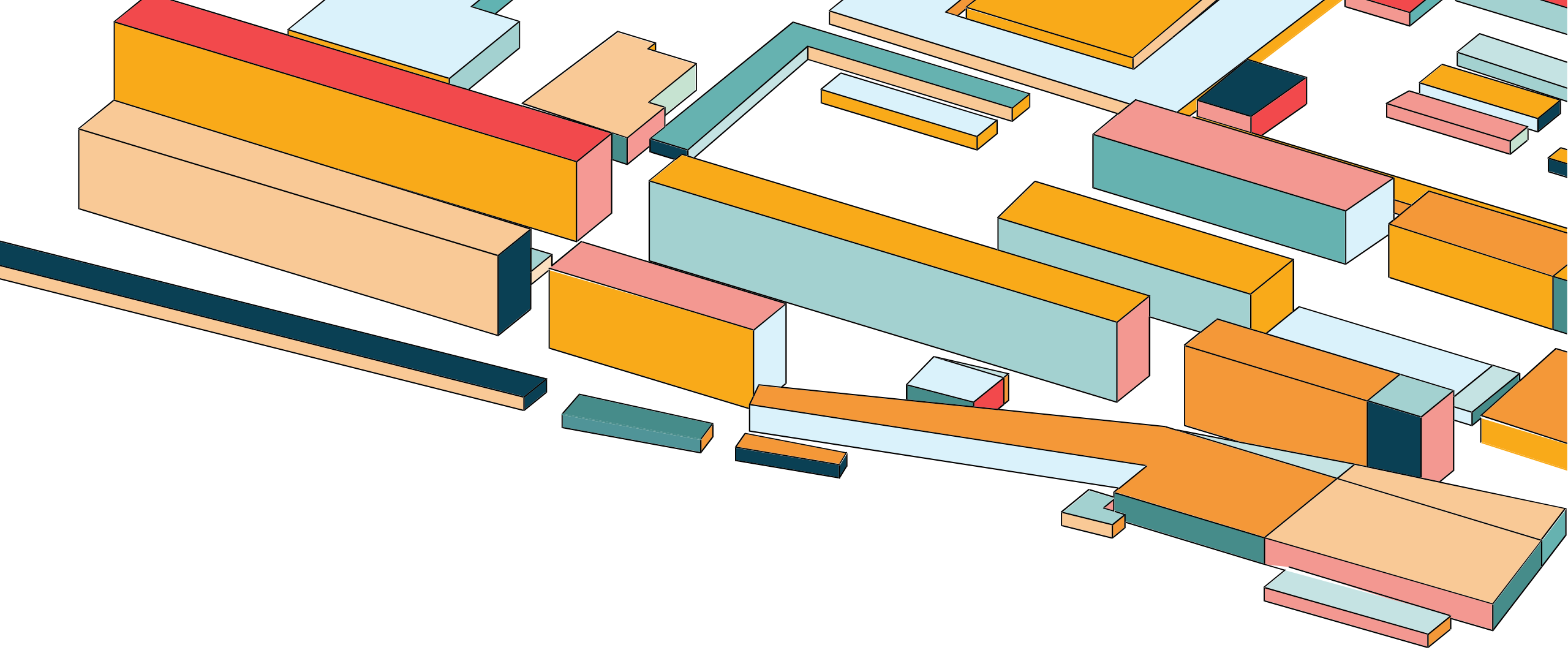
Wen, Hao, et al. "DroidBot-GPT: GPT-powered UI Automation for Android." arXiv preprint arXiv:2304.07061 (2023).

GPT-POWERED UI AUTOMATION FOR ANDROID

Task	Actions
Check the current temperature of London	start the app World Weather -> click view with text "London"
Add Beijing and check its current temperature	start the app World Weather -> click view 'Add city' -> click view with text "city, country" -> enter "Beijing China" into view with text "city, country" -> click view 'search' -> click view with text "Beijing, CN (39.91,..." -> click view with text "Current weather"
Close the wind direction display	start the app World Weather -> click view 'Extras' -> click view with text "Settings" -> click view with text "Wind direction display"

Task complexity	Number	Average completion progress	Fully completion rate
2∼3 steps	10	73.33%	60.00%
4∼5 steps	13	63.46%	38.46%
6∼13 steps	10	66.97%	20.00%
Total	33	66.76%	39.39%

Wen, Hao, et al. "DroidBot-GPT: GPT-powered UI Automation for Android." arXiv preprint arXiv:2304.07061 (2023).



SOFTWARE DEPLOYMENT

ANSWERING SOFTWARE DEPLOYMENT QUESTIONS VIA DEEP LEARNING

Motivation

- Software deployment dominates the time and cost of product delivery and system administration
- As software systems continue to grow in complexity and scale, deploying and delivering them becomes increasingly difficult

Guanjie Qiu, Diwei Chen, Shuai Zhang, Yitian Chai, Xiaodong Gu, and Beijun Shen. Answering Software Deployment Questions via Neural Machine Reading at Scale. ASE'22

ANSWERING SOFTWARE DEPLOYMENT QUESTIONS VIA DEEP LEARNING

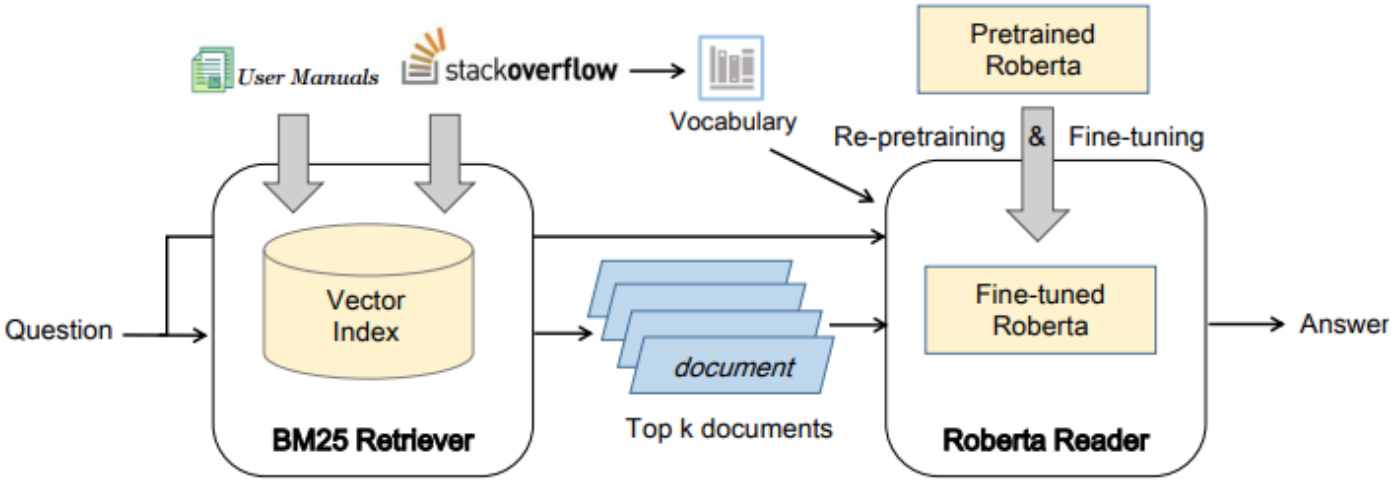


Figure 1: An overview of *DeployQA*.

Adaptation on Vocabulary

- Extending the vocab of RoBERTa with domain specific tokens from User manual and StackOverflow
- Domain-specific tokens: Kubernetes, K8s, etc.

Guanjie Qiu, Diwei Chen, Shuai Zhang, Yitian Chai, Xiaodong Gu, and Beijun Shen. Answering Software Deployment Questions via Neural Machine Reading at Scale. ASE'22

ANSWERING SOFTWARE DEPLOYMENT QUESTIONS VIA DEEP LEARNING

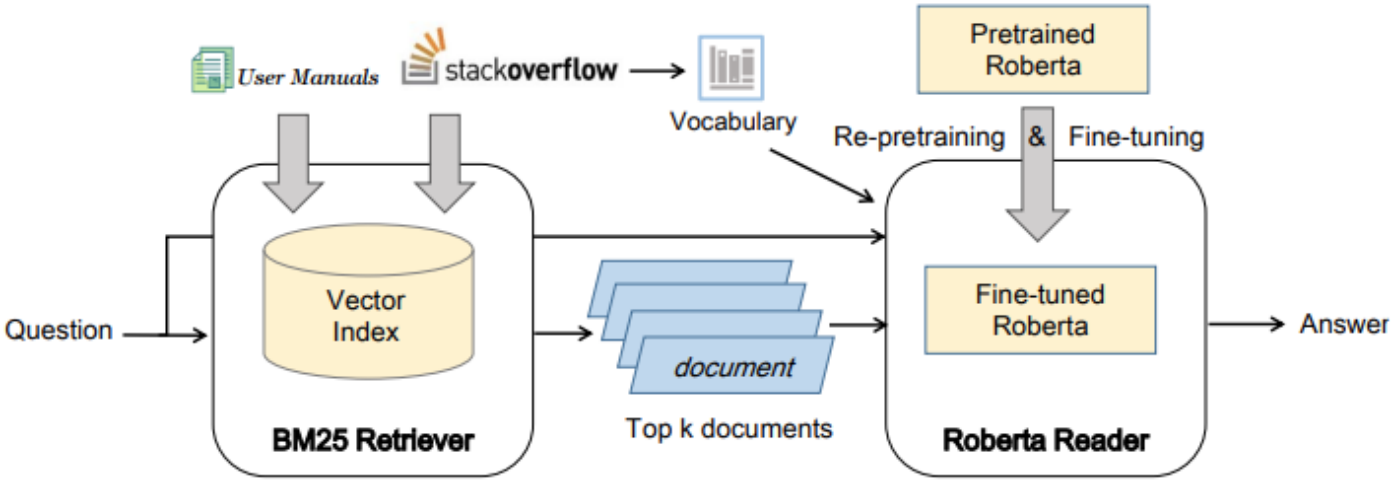


Figure 1: An overview of *DeployQA*.

Adaptation on Pre-training

- RoBERTa was pretrained on natural-language corpus as such Wikipedia and news articles
- Continuously pre-train RoBERTa on a Stack Overflow corpus to adapt it to the software and technical domain

Guanjie Qiu, Diwei Chen, Shuai Zhang, Yitian Chai, Xiaodong Gu, and Beijun Shen. Answering Software Deployment Questions via Neural Machine Reading at Scale. ASE'22

ANSWERING SOFTWARE DEPLOYMENT QUESTIONS VIA DEEP LEARNING

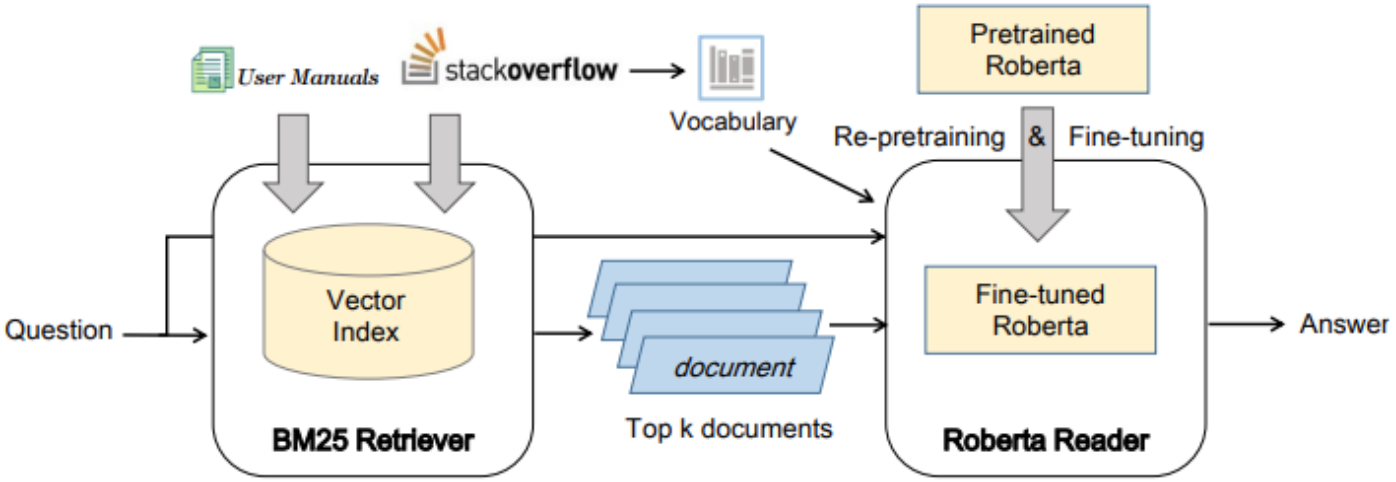


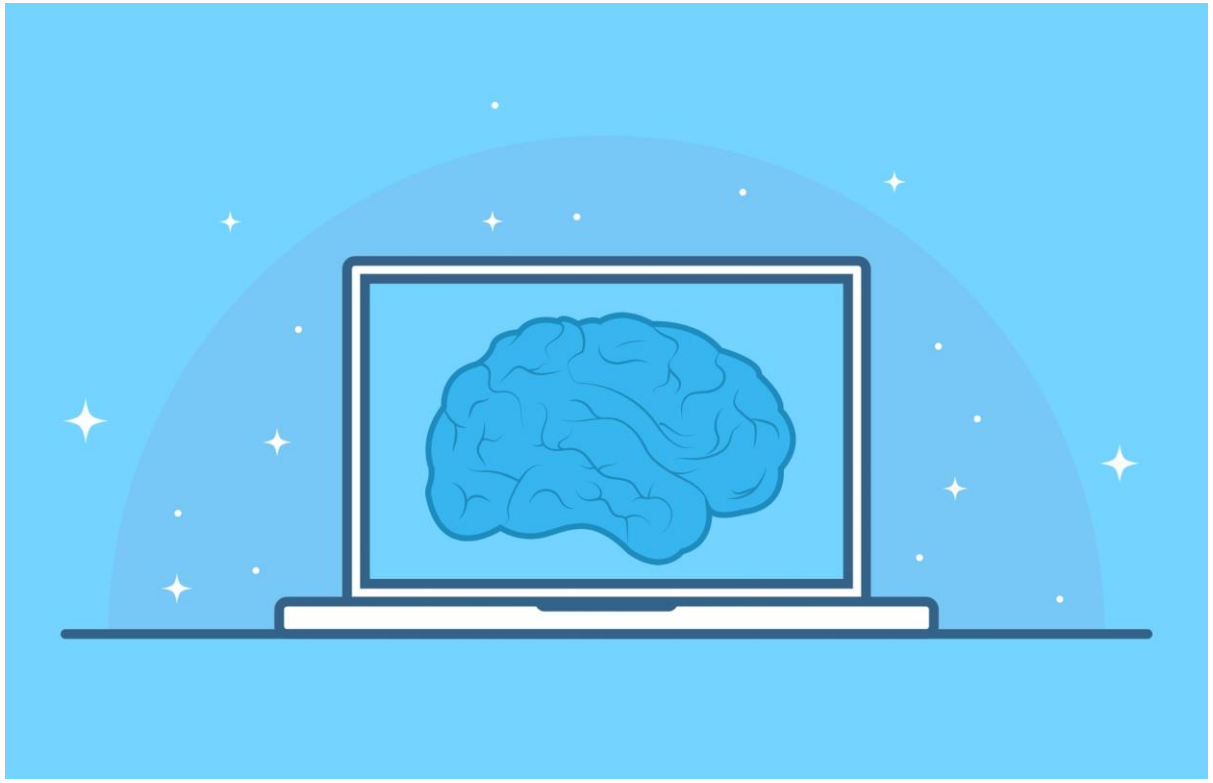
Figure 1: An overview of *DeployQA*.

Fine-tuning

- Fine-tune RoBERTa on Stack Overflow Q&A and deploy Q&A corpus to adapt it to the software deployment task

Guanjie Qiu, Diwei Chen, Shuai Zhang, Yitian Chai, Xiaodong Gu, and Beijun Shen. Answering Software Deployment Questions via Neural Machine Reading at Scale. ASE'22

CHALLENGES?



NEXT

- Individual academic talk
- Project demo
- Course review