

# ACK 2024

논문집

Annual Conference of KIPS 2024



신진학자 워크숍

---

## 자연어 처리 및 딥러닝 기반 소프트웨어 자동 유지보수

김미수 교수  
(전남대학교)

한국정보처리학회 ACK2024

# 자연어 처리 및 딥러닝 기반 소프트웨어 자동 유지보수

전남대학교 AI융합대학 인공지능학부  
김미수 (misoo.kim@jnu.ac.kr)

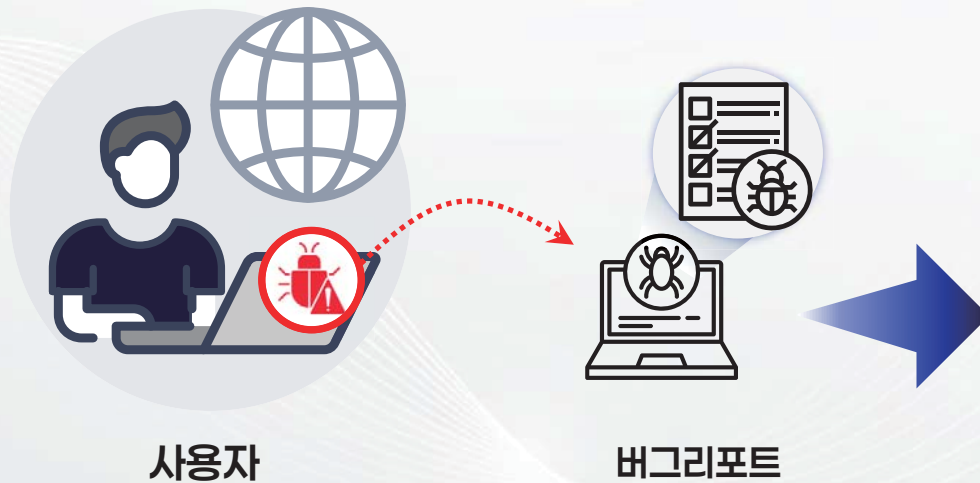


## 발표자 소개

이 름	김미수
최종 학위	성균관대학교 컴퓨터공학박사 정보검색기반 버그 추적 기술을 위한 자연어 처리 및 딥러닝 기반 프레임워크
연 락 처	misoo.kim@jnu.ac.kr
연 구 실	지능형 소프트웨어 진화 및 유지보수 연구실 (SEMI Lab) <a href="https://sites.google.com/view/semi-jnu/">https://sites.google.com/view/semi-jnu/</a>



## 연구 분야 - 버그 해결 자동화



<https://github.com/WordPress/gutenberg/issues/65825>

### Global Styles: Unnecessary vertical scrollbar appears #65825

Closed

#65875

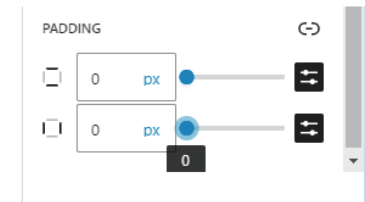
t-hamano opened last week

#### Description

Note: This issue cannot be reproduced in WP6.7 Beta 1. It can be reproduced in the latest Gutenberg.

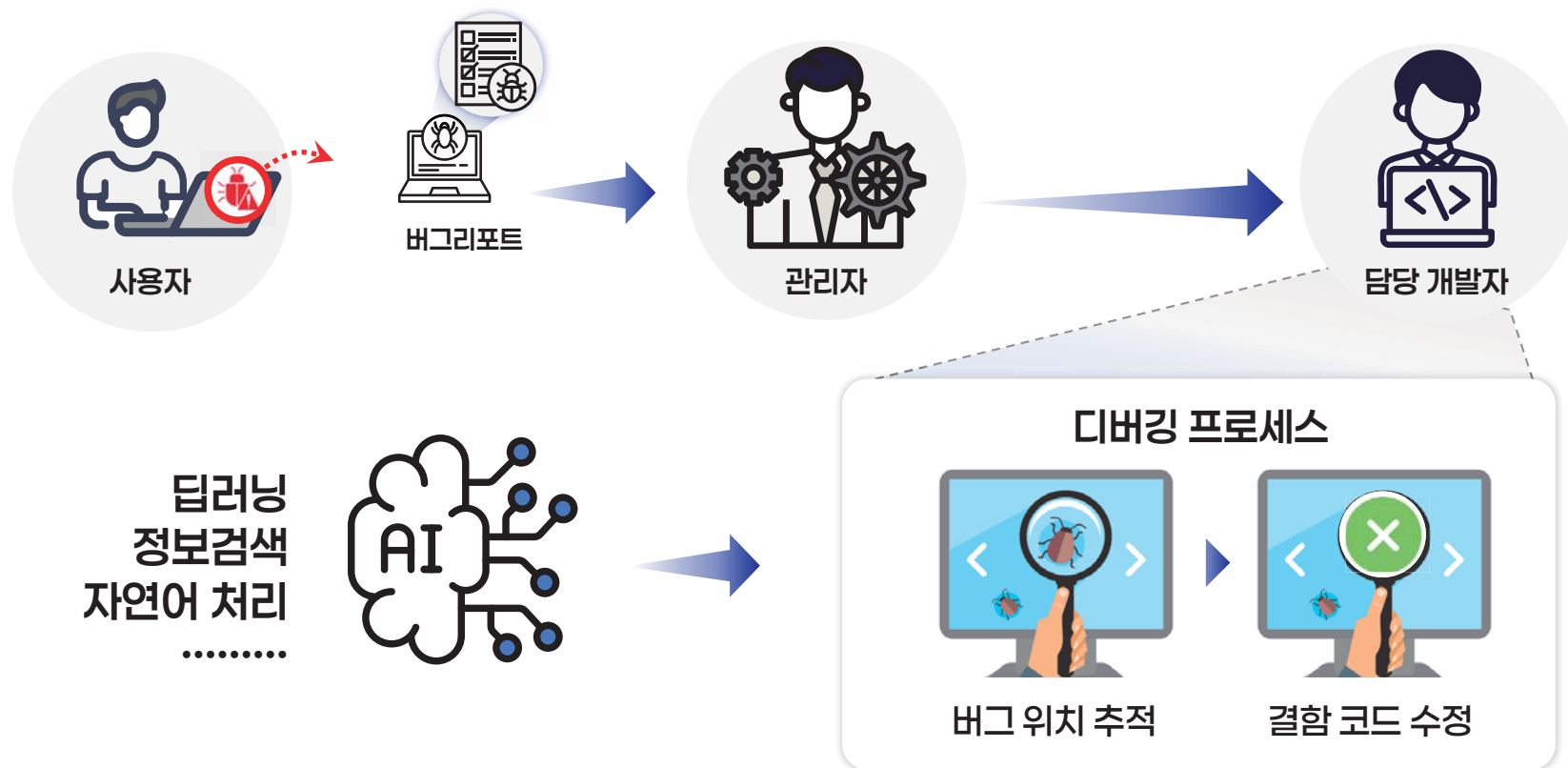
A vertical scrollbar may appear in global styles even when the content does not need to scroll vertically.

This issue seems to occur when there is a `RangeControl` at the end of the content. You can see that the tooltip that appears when operating the slider overflows:



The tooltip is there even though it is not visually visible, which is why the overflow appears to occur.

## 연구 분야 - 버그 해결 자동화



# 정보검색 기반 버그 추적 자동화 연구

## IRBL : Information retrieval-based bug localization

### 버그리포트 (쿼리)

Global Styles: Unnecessary vertical scrollbar appears #65825

Closed #65875

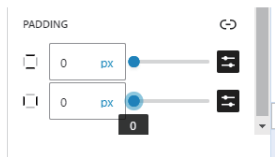
t-hamano opened last week

#### Description

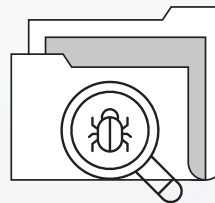
Note: This issue cannot be reproduced in WP6.7 Beta 1. It can be reproduced in the latest Gutenberg.

A vertical scrollbar may appear in global styles even when the content does not need to scroll vertically.

This issue seems to occur when there is a `RangeControl` at the end of the content. You can see that the tooltip that appears when operating the slider overflows:



The tooltip is there even though it is not visually visible, which is why the overflow appears to occur.



IRBL + NLP

### 결함 메소드

gutenberg / packages / components / src / range-control / tooltip.tsx

Code Blame 77 lines (66 loc) · 1.58 KB · ①

```

18
19 export default function SimpleTooltip(
20   props: WordPressComponentProps< TooltipProps, 'span' >
21 ) {
22   const {
23     className,
24     inputRef,
25     tooltipPosition,
26     show = false,
27     style = {},
28     value = 0,
29     renderTooltipContent = ( v ) => v,
30     zIndex = 100,
31     ...restProps
32   } = props;
33   const position = useTooltipPosition( { inputRef, tooltipPosition } );
34   const classes = clsx( 'components-simple-tooltip', className );
35   const styles = {
36     ...style,
37     zIndex,
38   };
39
40   return (
41     <Tooltip
42       { ...restProps }

```

# 정보검색 기반 버그 추적 자동화 연구

IRBL : Information retrieval-based bug localization

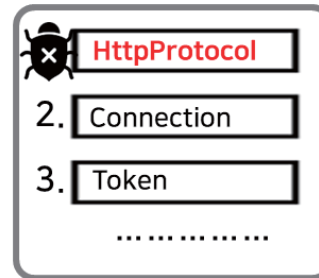
버그리포트 (쿼리)



소스 파일



IRBL



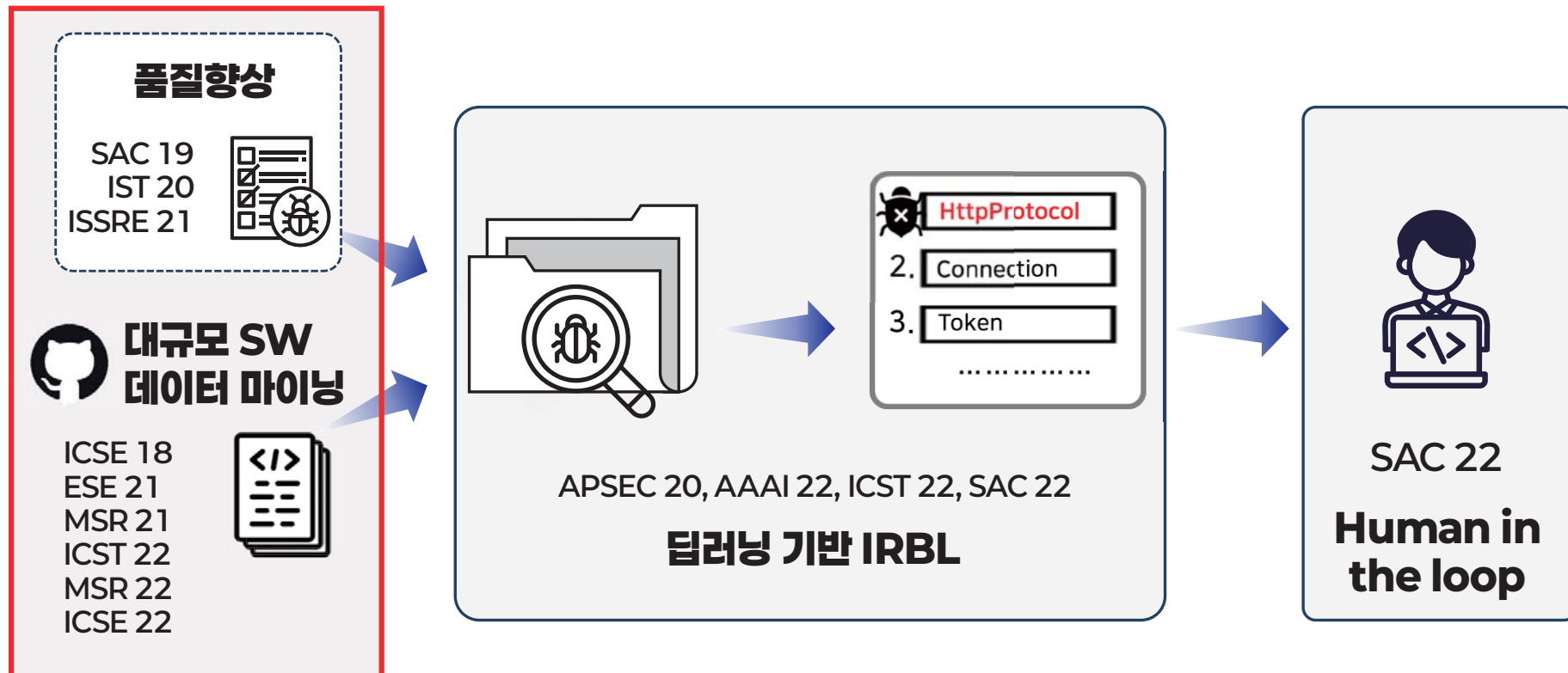
결함 위치 순위 목록



담당 개발자

# 정보검색 기반 버그 추적 자동화 연구

IRBL : Information retrieval-based bug localization





# 정보검색 기반 버그 추적 자동화 연구

## 버그리포트 품질 이슈

잡음 텍스트가  
많은 버그리포트



Camel / CAMEL-16032

[camel-main] autoconfiguration does not bind  
dataformat in the registry

▼ Description

Configuring a dataformat through properties does not work because after setting the properties, the dataformat bean is not bound in the registry and when the route starts, the dataformat is recreated: <https://pastebin.com/raw/Tf05i7FK>

Attached a reproducer 816.zip<sup>8</sup> with FHIR dataformat, but it's the same for every dataformat;

텍스트가 불충분한  
버그리포트



Commons Collections / COLLECTIONS-603

Small improvements for generics, conditional  
statements, and warnings suppressions

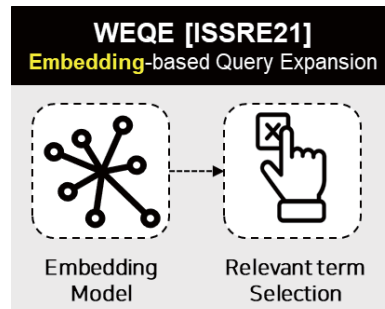
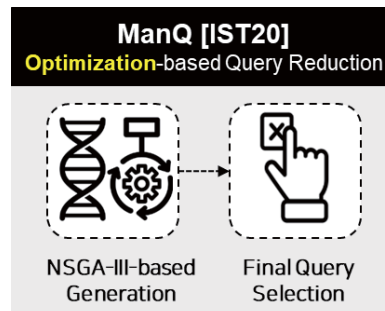
▼ Description

Place holder for pull request #17

# 정보검색 기반 버그 추적 자동화 연구

## 버그리포트 품질 향상을 위한 자동 쿼리 재구성 기법 제안

- 최적화 기반 쿼리 축소 (ManQ), 워드 임베딩 기반 쿼리 확장 (WEQE)



ManQ: ▲ 4~8%

WEQE: ▲ 6~16%

## 정보검색 기반 버그 추적 자동화 연구

### 대규모 SW 데이터 마이닝 for IRBL

- OSS 32개의 약 40,000 개 버그 리포트 및 해결 이력 수집
- 버그리포트와 소스파일 구분 필요성 제시 (프로덕션/테스트)
- 버그 종류를 간과하여 **73% 버그에 대한 평가 정확도를 저해함**

II

버그 자동 분류 시, 73% 성능 개선 됨을 발견



도메인 지향 딥 전이학습 기반  
버그리포트 분류 모델 개발

### 딥러닝 소프트웨어 버그 마이닝 및 분석

- 딥러닝 소프트웨어 버그 마이닝
- Denchmark 배포 :  
딥러닝 소프트웨어 버그 데이터셋

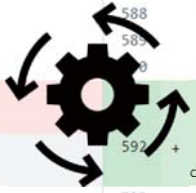


딥러닝 소프트웨어에서 가장 좋은 IRBL 제시

  
**Denchmark**

# 결함 코드 수정 자동화 연구

## APR (Automated Program Repair)



File Path	Commit Hash
third_party/xla/xla/hlo/Utils/hlo_sharding_util.cc	https://github.com/tensorflow/tensorflow/commit/ea0b19

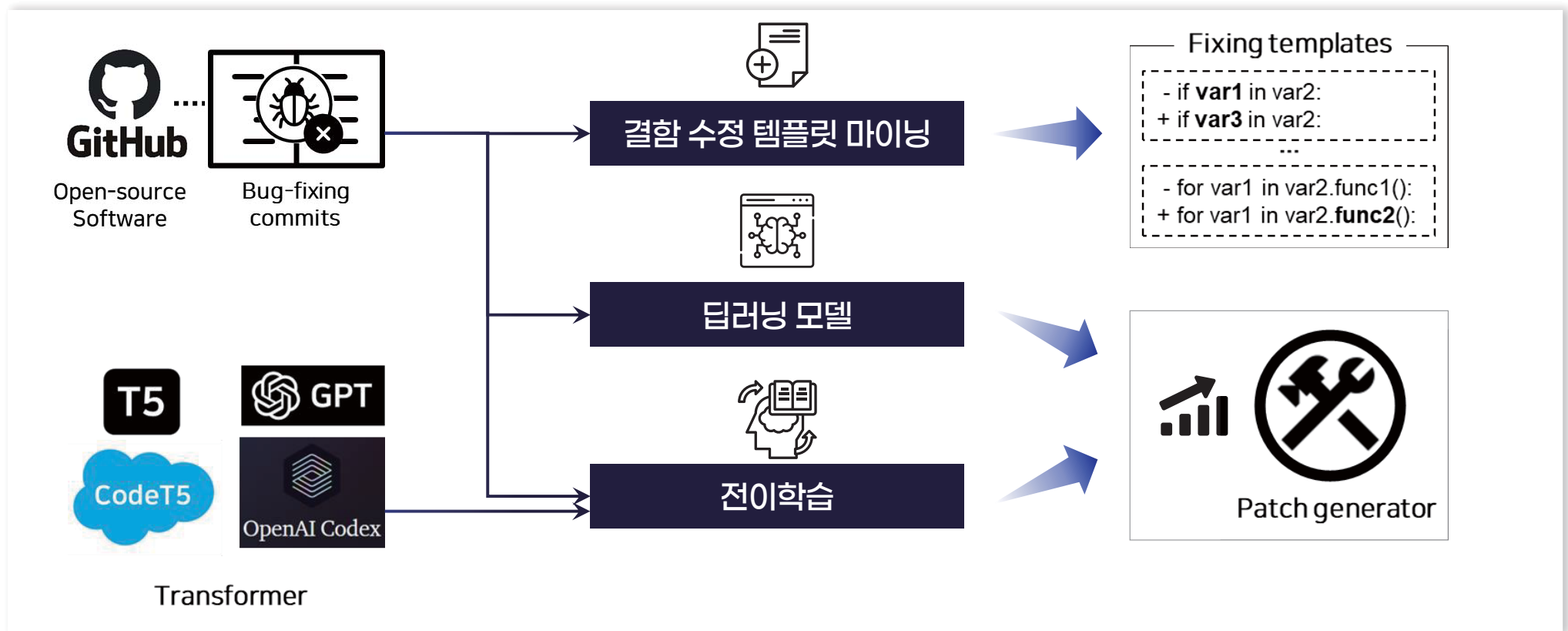
  

Line	Original Code	Modified Code
582	<code>return count &gt; 0 ? std::optional&lt;int64_t&gt;(device) :</code>	<code>return count &gt; 0 ? std::optional&lt;int64_t&gt;(device) :</code>
583	<code>std::optional&lt;int64_t&gt;();</code>	<code>std::optional&lt;int64_t&gt;();</code>
584	<code>}</code>	<code>}</code>
585	<code>- HloSharding FindCommonSharding(absl::Span&lt;const HloSharding&gt; shardings) {</code>	<code>+ HloSharding FindCommonSharding(absl::Span&lt;const HloSharding&gt; shardings,</code>
586	<code></code>	<code>+ std::optional&lt;HloSharding&gt; default_sharding)</code>
587	<code>{</code>	<code>{</code>
588	<code>CHECK(!shardings.empty());</code>	<code>CHECK(!shardings.empty());</code>
589	<code>bool all_compatible = true;</code>	<code>bool all_compatible = true;</code>
590	<code>HloSharding common_sharding = shardings[0];</code>	<code>HloSharding common_sharding = shardings[0];</code>
591	<code>for (int i = 1; i != shardings.size(); ++i) {</code>	<code>for (int i = 1; i != shardings.size(); ++i) {</code>
592	<code>if (!MergeShardingIfCompatible(shardings[i],</code>	<code>if (common_sharding != shardings[i] &amp;&amp;</code>
593	<code>common_sharding.NumTiles(),</code>	<code>!MergeShardingIfCompatible(shardings[i],</code>
594	<code>&amp;common_sharding)) {</code>	<code>common_sharding.NumTiles(),</code>
595	<code>all_compatible = false;</code>	<code>&amp;common_sharding)) {</code>
596	<code>break;</code>	<code>all_compatible = false;</code>
597	<code></code>	<code>break;</code>
598	<code>@@ -599,7 +601,7 @@ HloSharding FindCommonSharding(absl::Span&lt;const HloSharding&gt; shardings) {</code>	<code>599 // TODO(tongfei): instead of return the first sharding in case not all</code>
599	<code>// shardings are compatible, we should find a sharding that's compatible</code>	<code>600 // shardings are compatible, we should find a sharding that's compatible</code>
600	<code>with</code>	<code>with</code>
601	<code>// the most number of shardings instead.</code>	<code>603 // the most number of shardings instead.</code>
602	<code>- return shardings[0];</code>	<code>604 + return default_sharding.has_value() ? default_sharding.value() :</code>
603	<code>}</code>	<code>shardings[0];</code>
		<code>605 }</code>



# 결함 코드 수정 자동화 연구

## 연구 대상



# 결함 코드 수정 자동화 연구

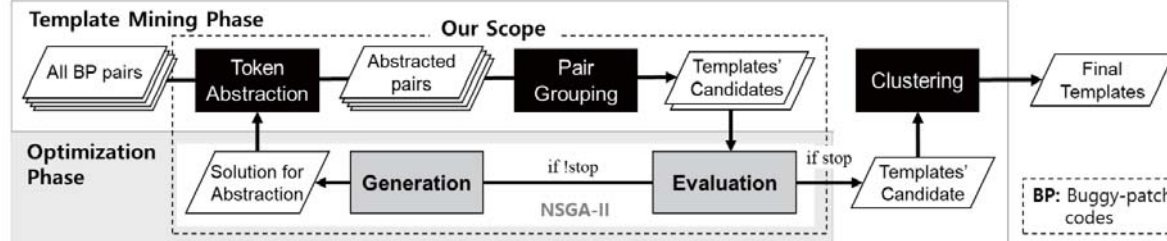
## 정적 분석 및 다목적 최적화 기반 템플릿 마이닝

- 목적함수 설계 : fixability (기준), applicability, efficiency → 유전알고리즘 기반 최적화
- 실험 결과 : : fixability를 유지하며 applicability와 efficiency를 각각 7%, 146% 향상

Table 1: Motivating Examples (Blue bold: different abstraction compared with previous, Underline: changed codes)

Type	Line	Method	Buggy Code	Patch Code	App.	Fix.		#Ingredients	Efficiency	Search space
if stmt.	1	Target	if d == 0	if d == 0 && k >1						
	2	DevReplay	if \$0 == \$1	if \$0 == \$1 && k >1	L	?	H	0	-	
	3	Yang et al.	if V1 == 'N'	if V1 == 'N' && V2 >'N'	M	M	L	2 (V2, 'N')		( V  - 1) *  N
	4	COND Abstraction	if <b>COND1</b>	if <b>COND1</b> && <b>COND2</b>	H	L	M	1 (COND2)		COND  - 1
assign. & func. call stmt.	5	Target	a = calc(x=x_v, yz=y_v)    b = calc(x=x_v, yz=(y_v * values.z_v))							
	6	DevReplay	a = calc(\$0=\$1, \$2=\$3)    b = calc(\$0=\$1, \$2=(\$3 * values.z_v))		L	H	?	0	-	
	7	Yang et al.	V1 = F1(KA1=V2, KA2=V3)    V4 = F1(KA1=V2, KA2=(V3 * V5.F2))		M	H	L	3 (V4, V5, F2)		( V  - 3) <sup>2</sup> * ( FUA  - 1)
	8	ATTR Abstraction	V1 = F1(KA1=V2, KA2=V3)    V4 = F1(KA1=V2, KA2=(V3 * V5.A1))		M	H+	M	3 (V4, V5, A1)		( V  - 3) <sup>2</sup> *  A
	9	MATH Abstraction	V1 = F1(KA1=V2, KA2=V3)    V4 = F1(KA1=V2, KA2= <b>MATH1</b> )		H	M	H	2 (V4, MATH1)		( V  - 3) * ( MATH )
	10	ARG Abstraction	V1 = F1( <b>KP1</b> , <b>KP2</b> )    V4 = F1( <b>KP1</b> , <b>KP3</b> )		H+	L	H	2 (V4, KP3)		( KP  - 1) <sup>2</sup>

V: variable, N: number, F: function, KA: keyword argument, KP: keyword argument pair, COND: conditional expression, A/ATTR: attribute, MATH: mathematical expression, ARG: argument, L: low, M: medium, H: high, App:Applicability, Fix.: fixability





## 결함 코드 수정 자동화 연구

### 딥러닝 기반 APR에서 인스턴스 품질의 영향 분석

- 발견1. 49.3%의 결함 코드 프롬프트에 APR 성능을 저해하는 잡음 토큰이 있음
- 발견2. 잡음 토큰 제거 시 APR 성능을 88% 개선시킬 수 있음을 보임
- 발견3. 잡음 토큰들이 전체 훈련 인스턴스에서 IDF가 낮은 경향을 확인



Input Instance	fix no-invalid-this Unexpected 'this' s(\${this.s}), r(\${this.r}) : transform:[s(\${this.s}), r(\${this.r}), t(\${this.t})]
Wrong Output	transform:[s(\${this.s}), r(\${r}), t(\${this.t})] <span style="color: red; font-size: 2em;">✗</span>
<div style="text-align: center; color: red; font-weight: bold;">Noise token</div> <div style="text-align: center;"> </div>	
Input Instance	fix no-invalid-this Unexpected 'this' s(\${this.s}), r(\${this.r}) : transform:s(\${this.s}), r(\${this.r}), t(\${this.t})
Correct Output	transform:[s(\${this.s}), t(\${this.t})] <span style="color: orange; font-size: 2em;">○</span>

# 결함 코드 수정 자동화 연구

## Transformer와 전이학습 기반 APR의 실증적 분석

- 요구사항 : 삼성전자 모바일 사업부에서 SonarQube로 식별한 **Kotlin 결함**의 자동 수정

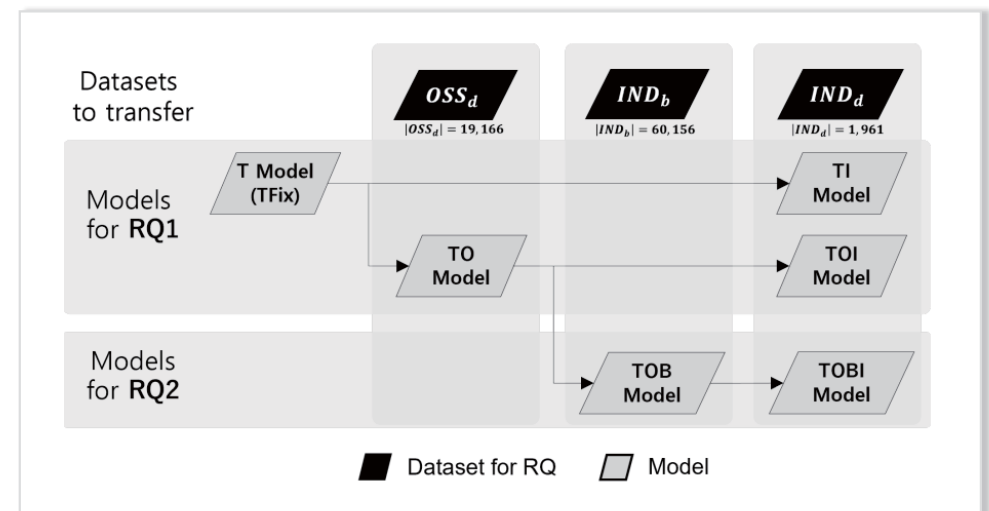
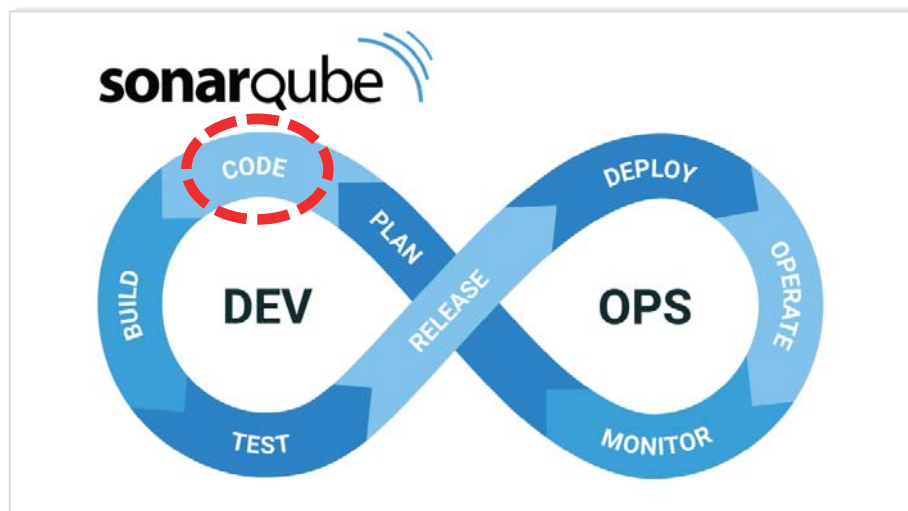
- 제안 및 실험 결과

1. Transformer + Transfer learning + bug history

▶ 30%의 결함 수정

2. Clustering & embedding-based instance selection

▶ 26% 더 많은 결함 수정





## 현재 진행 중인 연구 (1/4)

### 한국어 자연어 처리 기반 한국어 버그 추적 & 수정 자동화

- 문제: 기존 연구들은 대부분 **영어 리포트와 영어 기반 소스코드**를 기반으로 함
- RQ1: 한국어 텍스트로 결함 코드를 어떻게 찾을 것인가?
- RQ2: 결함 코드를 수정 할 때 한국어 버그리포트를 어떻게 활용할 것인가?

### ▶ 한국어 언어 모델 활용

#### Global Styles: Unnecessary vertical scrollbar appears #65825

Global Styles: Unnecessary vertical scrollbar appears #65825

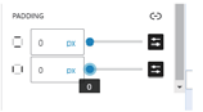
1-hamano opened last week

Description

Note: This issue cannot be reproduced in WP6.7 Beta 1. It can be reproduced in the latest Gutenberg.

A vertical scrollbar may appear in global styles even when the content does not need to scroll vertically.

This issue seems to occur when there is a `RangeControl` at the end of the content. You can see that the tooltip that appears when operating the slider overflows:



The tooltip is there even though it is not visually visible, which is why the overflow appears to occur.

#### 일정 추가 시 null값 에러 수정 #21

일정 추가 시 null값 에러 수정 #21

choyeseol opened this issue on Sep 9 · 0 comments · Fixed by #22

choyeseol commented on Sep 9

설명

일정 추가 시, id를 제외한 모든 값들이 null로 들어갑니다.

맥락

1. 일정을 추가한다
2. 일정이 정렬되어 저장되는 것이 아닌, id를 제외한 모든 값들이 null로 뜬다.

로그

덧붙임 말

controller에서도 안받아지니, 코드 자체를 잘못 생각한 것 같습니다. map으로 다시 해봅시다



```

@@ -11,7 +11,9 @@ public record PlanResponseDto(
11     String activity,
12     String location,
13     String storeName,
14 -    String placeUrl
14 +    String placeUrl,
15     String title,
16     String userId
17 ) {
18     public static PlanResponseDto from(final Plan plan) {
19         return new PlanResponseDto(
20             plan.getActivity(),
21             plan.getLocation(),
22             plan.getStoreName(),
23             plan.getPlaceUrl(),
24             plan.getTitle(),
25             plan.getUserId()
26         );
27     }
28 }

```

<https://github.com/WordPress/gutenberg/issues/65825>

<https://github.com/HASH-SQUAD/Harp-Server/issues/21>

<https://github.com/HASH-SQUAD/Harp-Server/pull/22/files>

## 현재 진행 중인 연구 (2/4)

### 프롬프트 최적화 기반 LLM 기반 APR

- 최근 LLM 기반 APR 연구들이 크게 증가되고 있음 (ICSE 2023 : 3 / 8 APR papers, ICSE 2024 : 5 / 9 APR papers)
- 이슈: 어떻게 LLM을 효과적으로 활용할 것인가?

#### 프로세스 측면



ChatGPT,  
반복된 접근

#### 사용성 측면



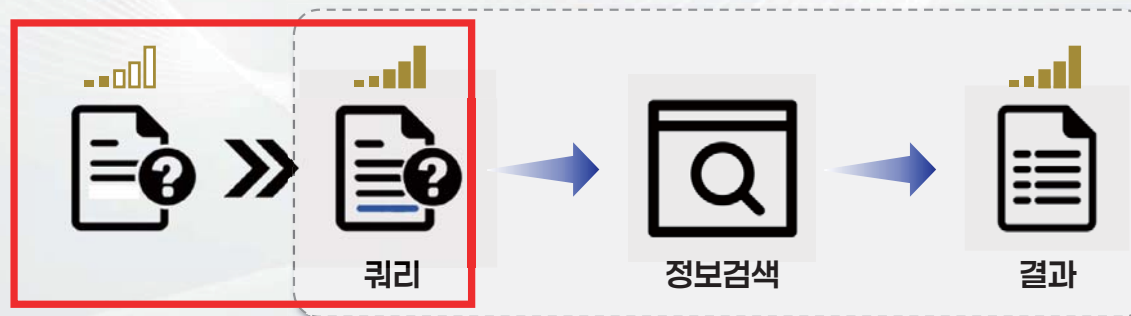
버그 유형에 맞는  
APR 활용 (양상블)

#### 입력 데이터 측면



프롬프트  
최적화

- 주 연구 분야: IRBL에서 **쿼리 (입력 데이터) 품질 개선**



## 현재 진행 중인 연구 (3/4)

### 프롬프트 최적화 기반 LLM 기반 APR

- 최근 LLM 기반 APR 연구들이 크게 증가되고 있음 (ICSE 2023 : 3 / 8 APR papers, ICSE 2024 : 5 / 9 APR papers)

- 이슈: 어떻게 LLM을 효과적으로 활용할 것인가?

➤ 패치 코드 생성에 도움이 되는 정보는?

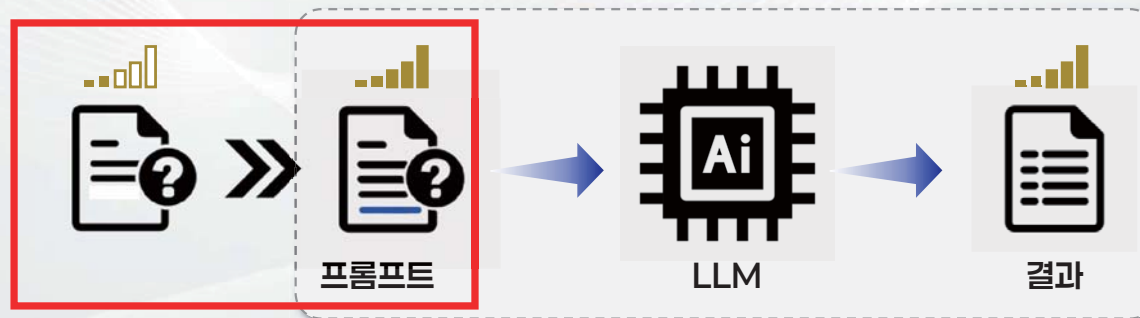
➤ 고품질 자연어와 고품질 프롬프트는 같은가?

➤ 패치 코드 생성을 방해하는 정보는?

➤ LLM은 버그 관련 프롬프트를 어떻게 이해하는가?

- 주 연구 분야: IRBL에서 쿼리 (입력 데이터) 품질 개선

### ▶ 프롬프트 최적화 문제

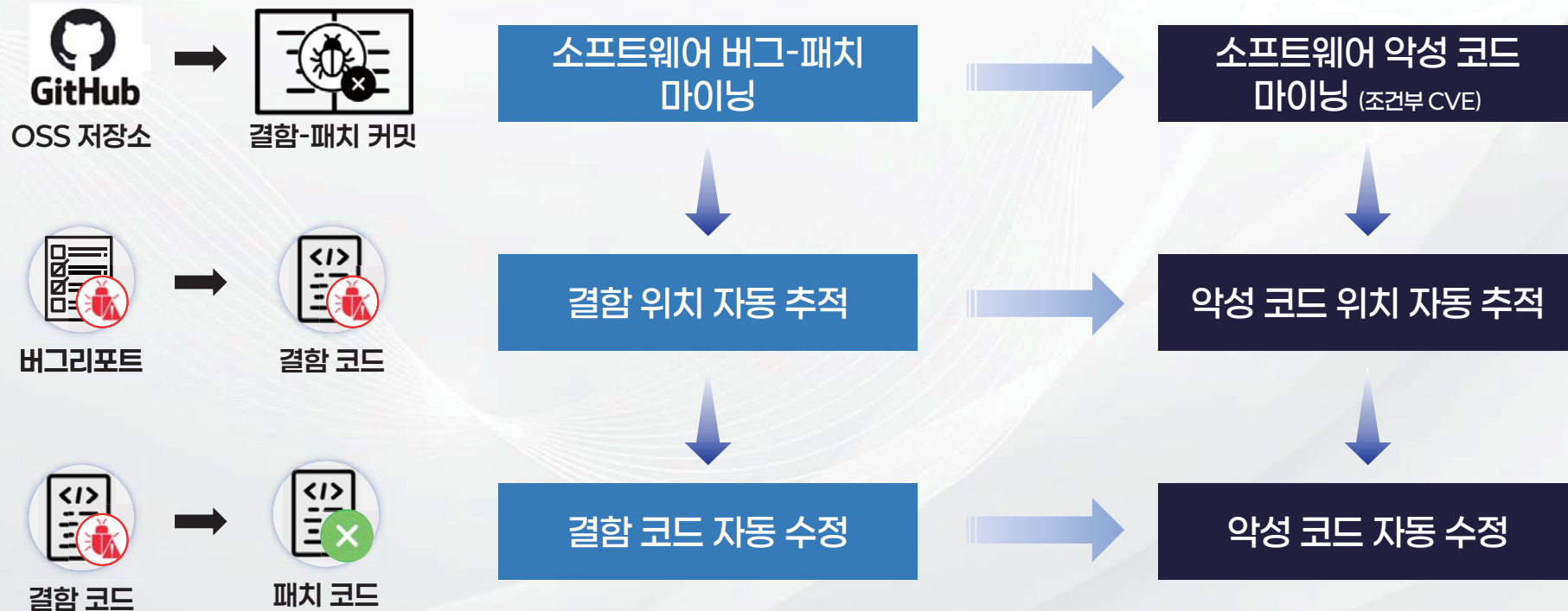


## 현재 진행 중인 연구 (4/4)

### DevSecOps를 위한 악성 패키지 탐지 및 악성 코드 자동 수정

- 주 연구분야: 소프트웨어 디버깅 자동화

▶ 악성 코드 위치 식별 및 수정 자동화



「**감사합니다.**」  
misoo.kim@jnu.ac.kr