Beyond the Prompt:
An Empirical Study of Cursor Rules

Shaokang Jiang
shj@uci.edu

Daye Nam
dnam1@uci.edu

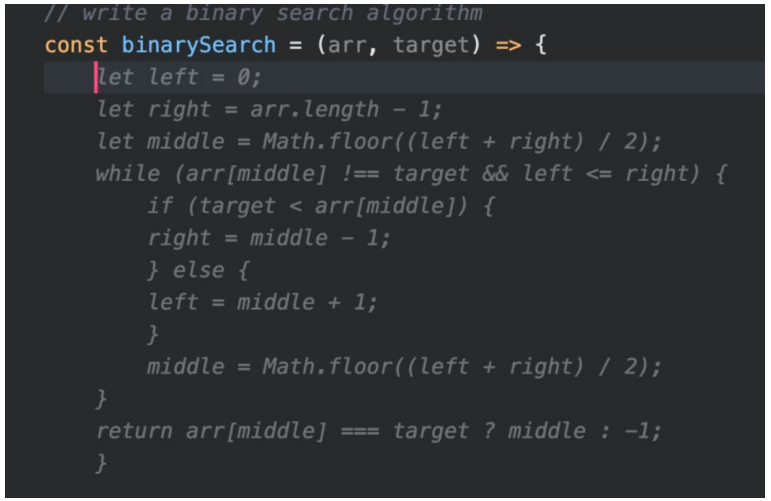
UC Irvine Donald Bren School of
Information & Computer Sciences

From Autocompletion to Persistent Rules


```
// write a binary search algorithm
const binarySearch = (arr, target) => {
    let left = 0;
    let right = arr.length - 1;
    let middle = Math.floor((left + right) / 2);
    while (arr[middle] !== target && left <= right) {
        if (target < arr[middle]) {
        right = middle - 1;
        } else {
        left = middle + 1;
        }
        middle = Math.floor((left + right) / 2);
    }
    return arr[middle] === target ? middle : -1;
    }
```

Autocompletion

From Autocompletion to Persistent Rules



```
// write a binary search algorithm
const binarySearch = (arr, target) => {
    let left = 0;
    let right = arr.length - 1;
    let middle = Math.floor((left + right) / 2);
    while (arr[middle] !== target && left <= right) {
        if (target < arr[middle]) {
        right = middle - 1;
        } else {
        left = middle + 1;
        }
        middle = Math.floor((left + right) / 2);
    }
    return arr[middle] === target ? middle : -1;
    }
```

Autocompletion

Prompt Engineering

From Autocompletion to Persistent Rules

GitHub Copilot

```
// write a binary search algorithm
const binarySearch = (arr, target) => {
    let left = 0;
    let right = arr.length - 1;
    let middle = Math.floor((left + right) / 2);
    while (arr[middle] !== target && left <= right) {
        if (target < arr[middle]) {
        right = middle - 1;
        } else {
        left = middle + 1;
        }
        middle = Math.floor((left + right) / 2);
    }
    return arr[middle] === target ? middle : -1;
    }
```

Autocompletion

H you are a python developer tasked with writing an application…

Ask LLM…
+ ⌥

Prompt Engineering

CURSOR

CLAUDE CODE

rule.mdc / SKILL.md

Always use X for styling
Conduct a comprehensive security review of this code
…

Rules/skills

From Autocompletion to Persistent Rules

GitHub Copilot

```
// write a binary search algorithm
const binarySearch = (arr, target) => {
    let left = 0;
    let right = arr.length - 1;
    let middle = Math.floor((left + right) / 2);
    while (arr[middle] !== target && left <= right) {
        if (target < arr[middle]) {
        right = middle - 1;
        } else {
        left = middle + 1;
        }
        middle = Math.floor((left + right) / 2);
    }
    return arr[middle] === target ? middle : -1;
    }
```

Autocompletion

H you are a python developer tasked with writing an application…

Ask LLM…
+ ⌥

Prompt Engineering

CURSOR

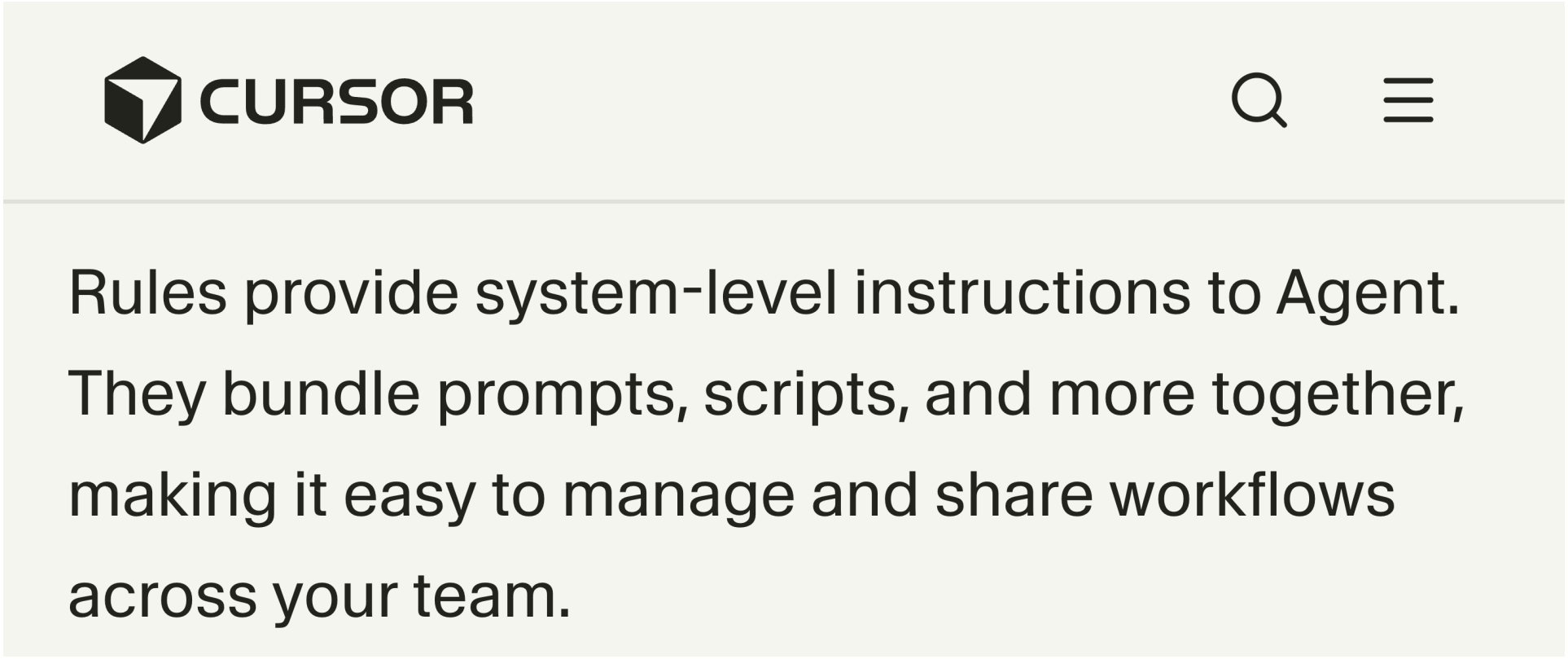
CLAUDE CODE

rule.mdc / SKILL.md

Always use X for styling
Conduct a comprehensive security review of this code
…

Rules/skills New!

Cursor Rule



https://cursor.com/docs/rules

Cursor Rule Example

frontend-development.mdc

This rule provides standards for frontend components:

- `setup/`: Setting up the environment.
- Use CamelCase
- Maintain module separation of concerns.
- Always ask clarifying questions if context is missing
- …

Cursor Rule Example

frontend-development.mdc

This rule provides standards for frontend components:

- `setup/`: Setting up the environment. — **Project**
- Use CamelCase — **Convention**
- Maintain module separation of concerns. — **Guideline**
- Always ask clarifying questions if context is missing — **LLM**
- …

Qualitative Analysis

Data Collection

Sourcegraph for Public Code

Search repo
repo:has.file(path:.mdc$)

GitHub

Fetch repo and metadata
Github API

Qualitative Analysis

Data Collection

Sourcegraph for Public Code

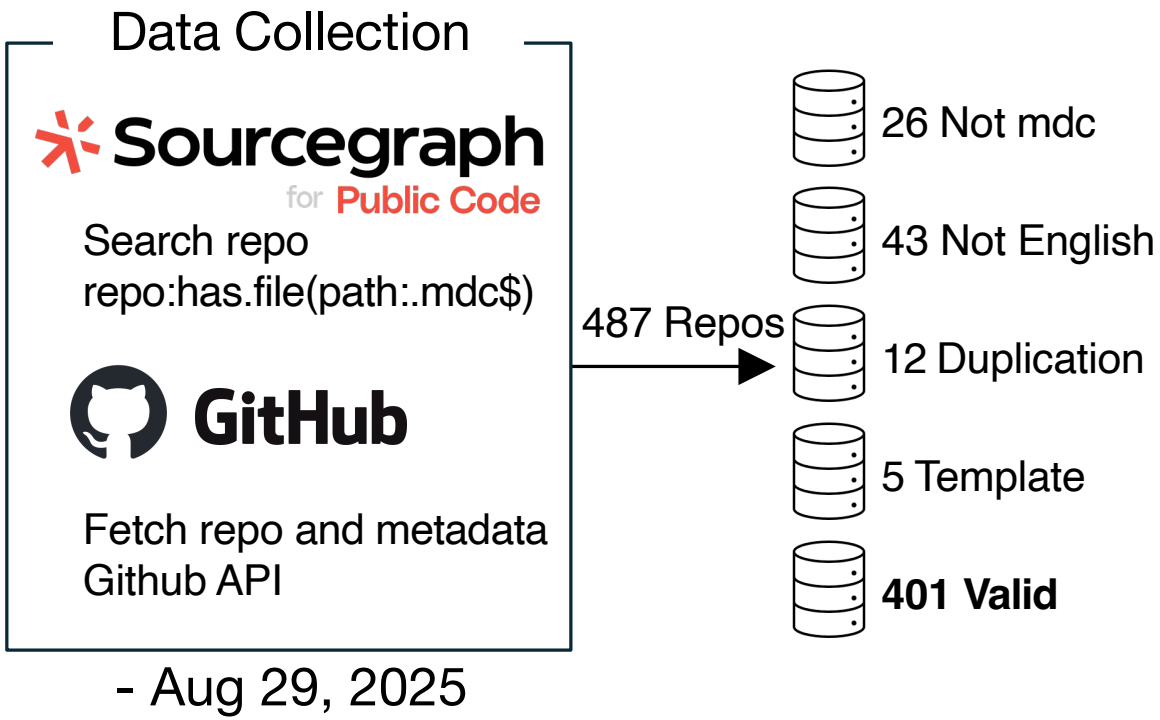
Search repo
repo:has.file(path:.mdc$)

GitHub

Fetch repo and metadata
Github API

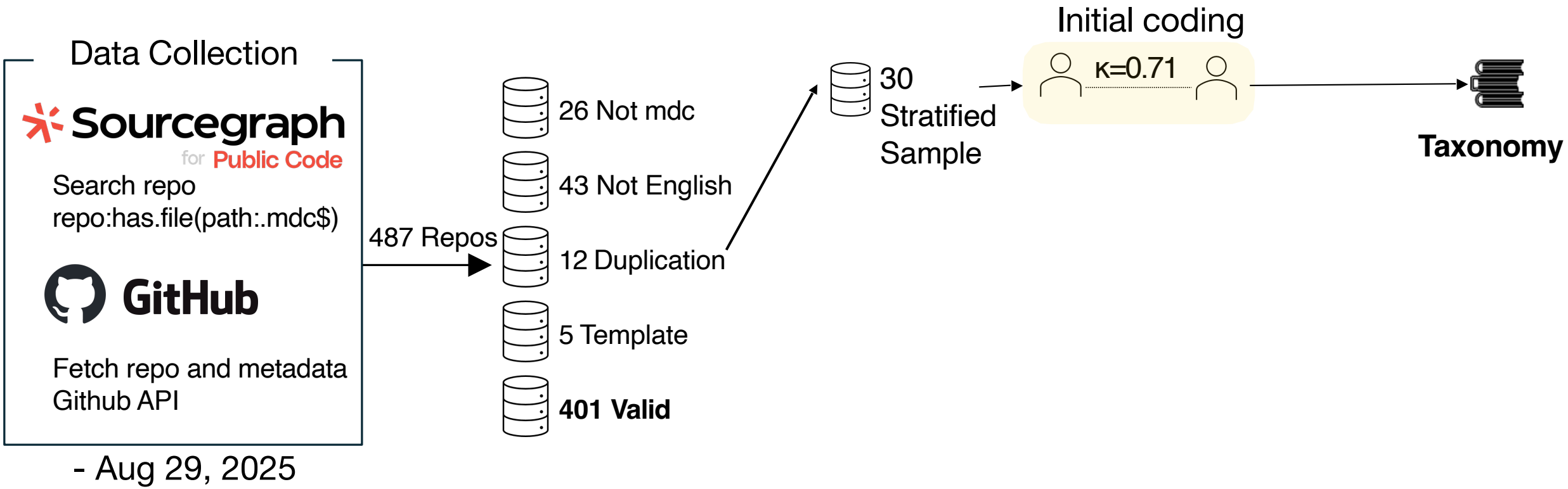
- Aug 29, 2025

Qualitative Analysis

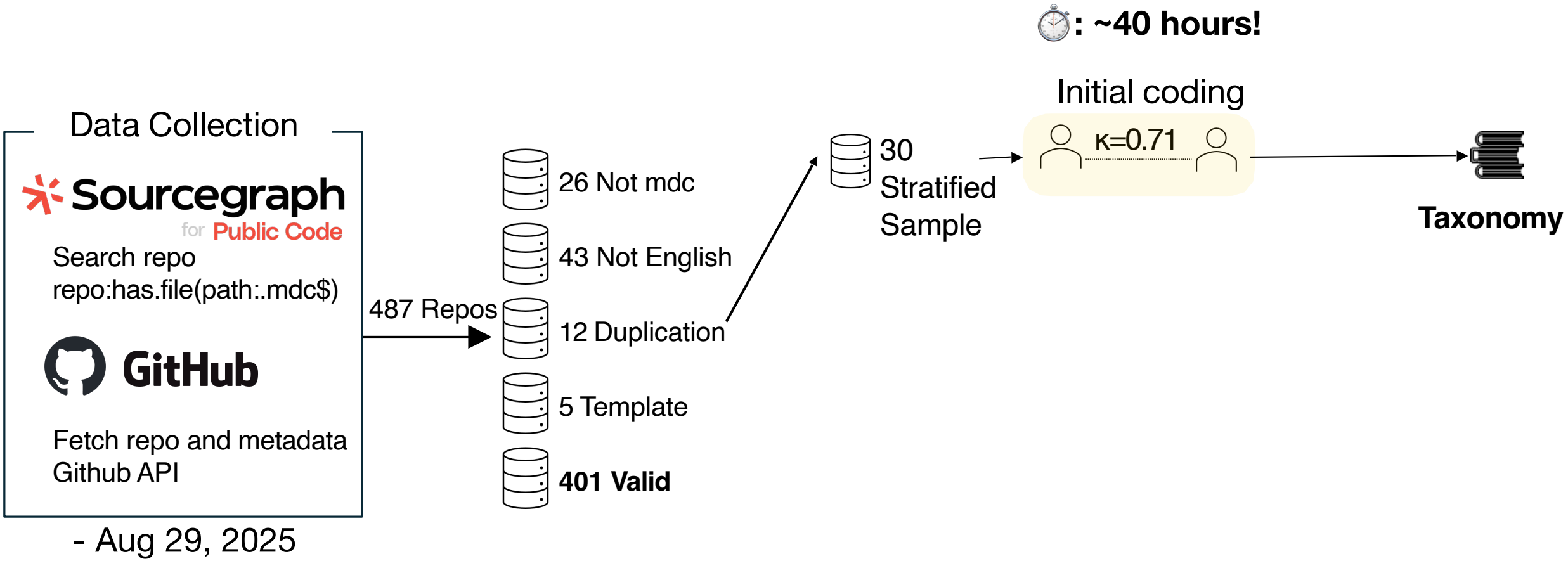


- Aug 29, 2025

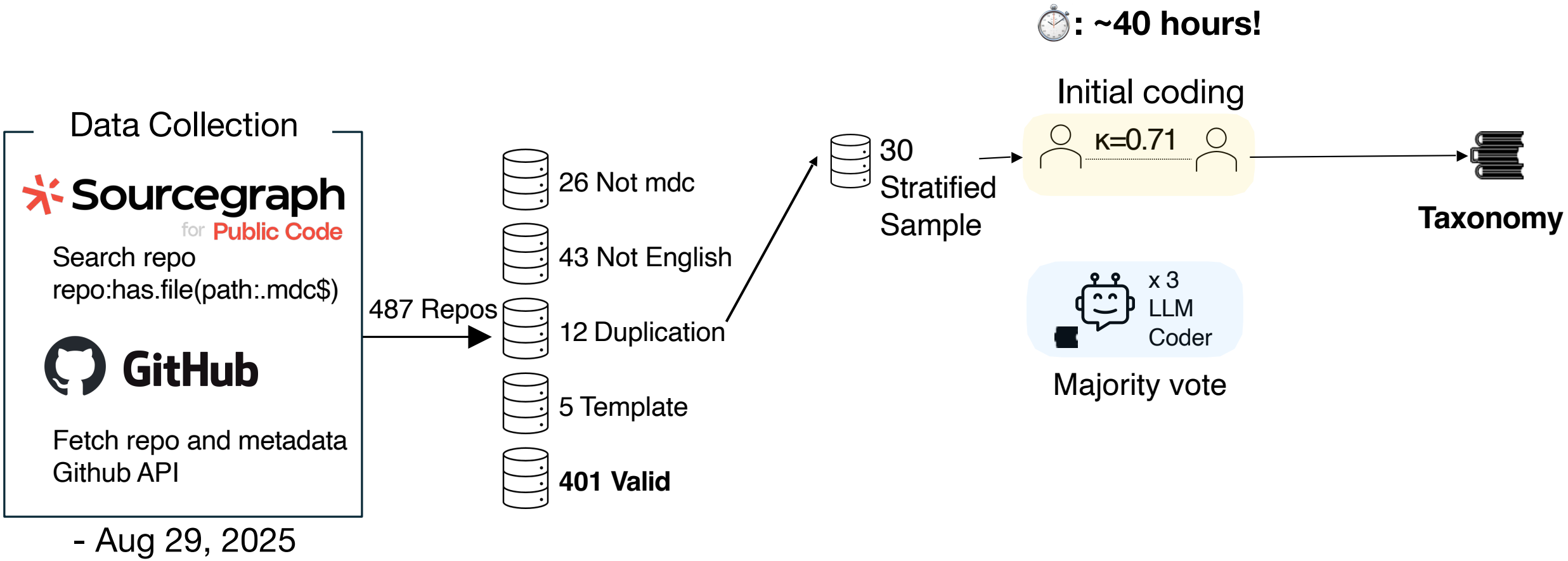
Qualitative Analysis



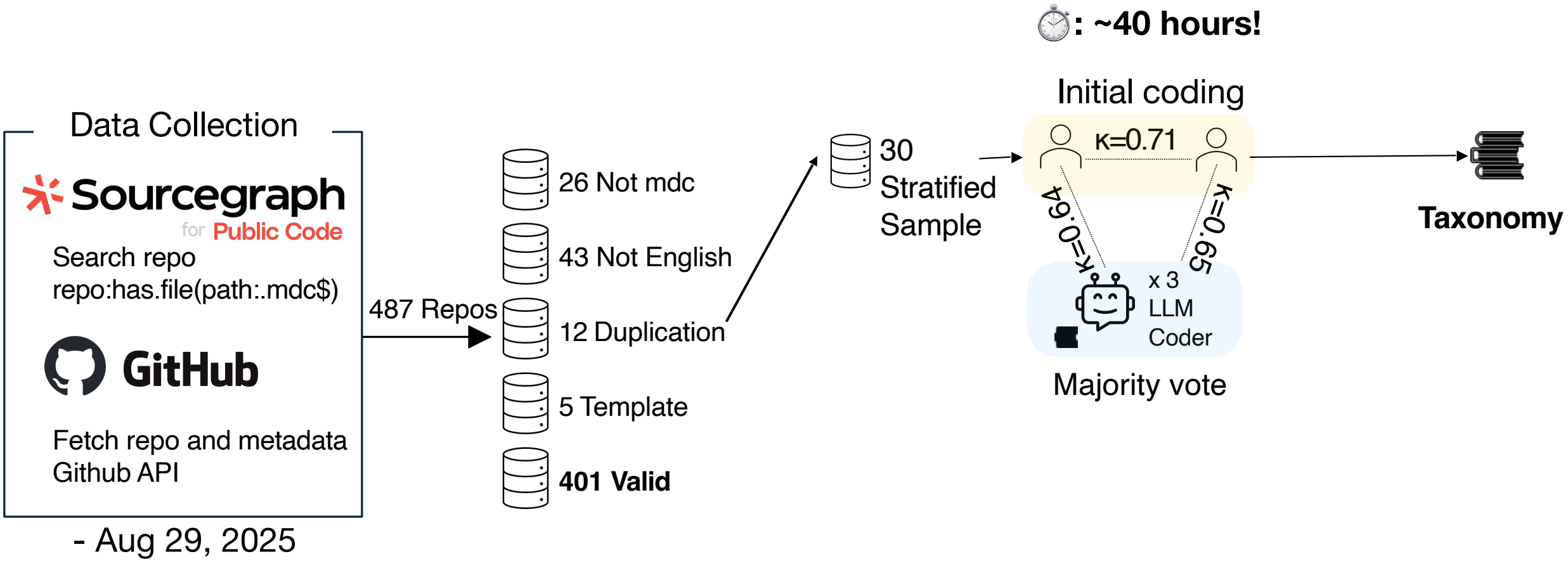
Qualitative Analysis



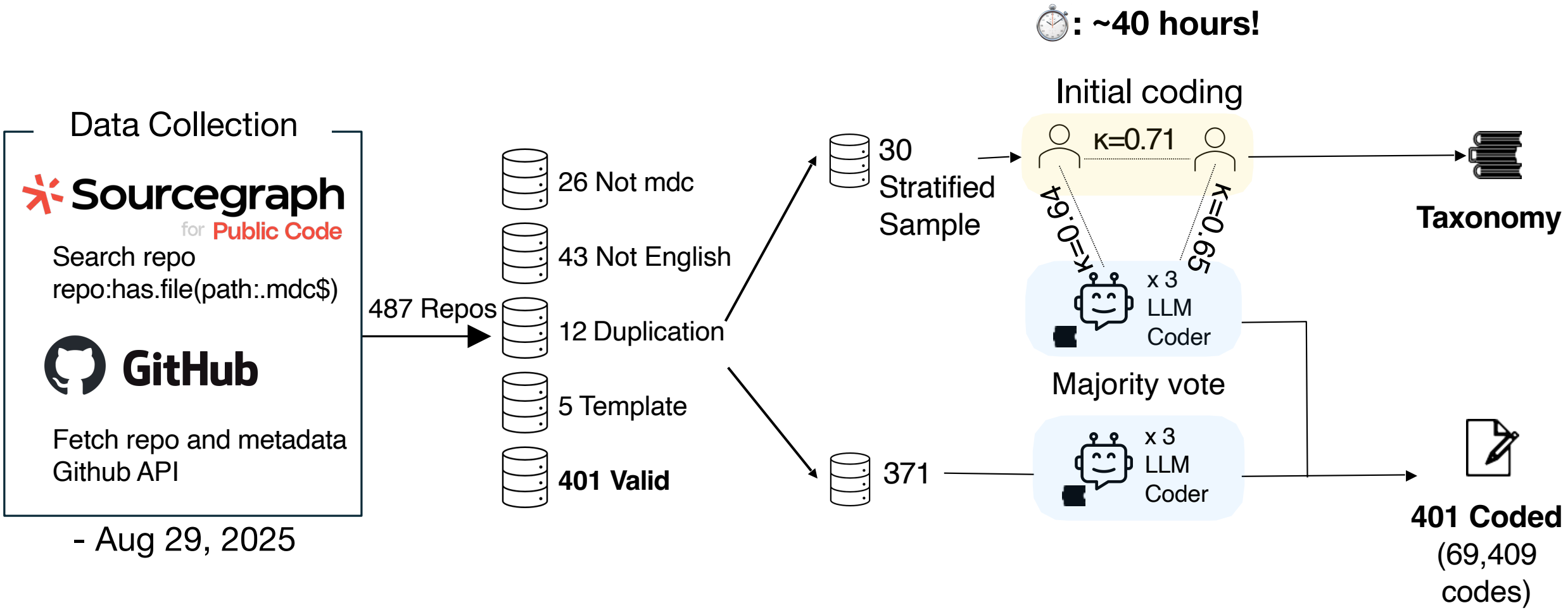
Qualitative Analysis



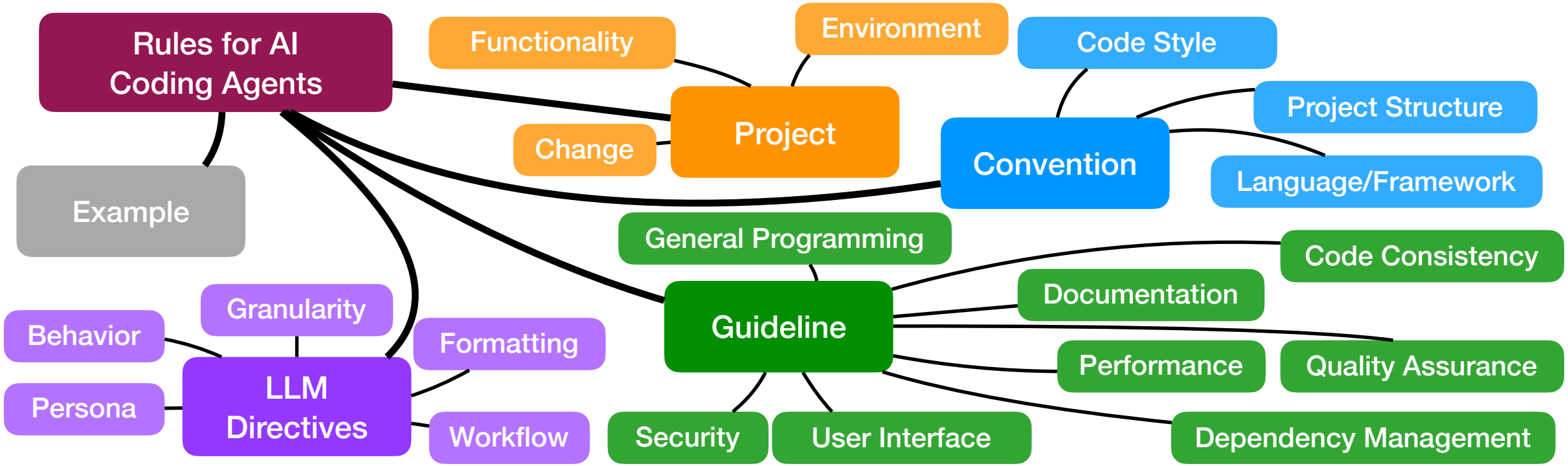
Qualitative Analysis



Qualitative Analysis



Taxonomy of Cursor Rule Information Types



Different Types of Information in Cursor Rules

This rule provides standards for frontend components:

- `setup/`: Setting up the environment.
- Use CamelCase
- Maintain module separation of concerns.
- Always ask clarifying questions if context is missing
- …

Different Types of Information in Cursor Rules

This rule provides standards for frontend components:

- `setup/`: Setting up the environment.
- Use CamelCase
- Maintain module separation of concerns.
- Always ask clarifying questions if context is missing
- …

Project

Describes the software project of the repositories.

Different Types of Information in Cursor Rules

This rule provides standards for frontend components:

- `setup/`: Setting up the environment. **Project**
- Use CamelCase **Convention**
- Maintain module separation of concerns.
- Always ask clarifying questions if context is missing
- …

How code should be written

Different Types of Information in Cursor Rules

This rule provides standards for frontend components:

- `setup/`: Setting up the environment. — **Project**
- Use CamelCase — **Convention**
- Maintain module separation of concerns. — **Guideline**

Rules for practices and pitfalls

- Always ask clarifying questions if context is missing
- …

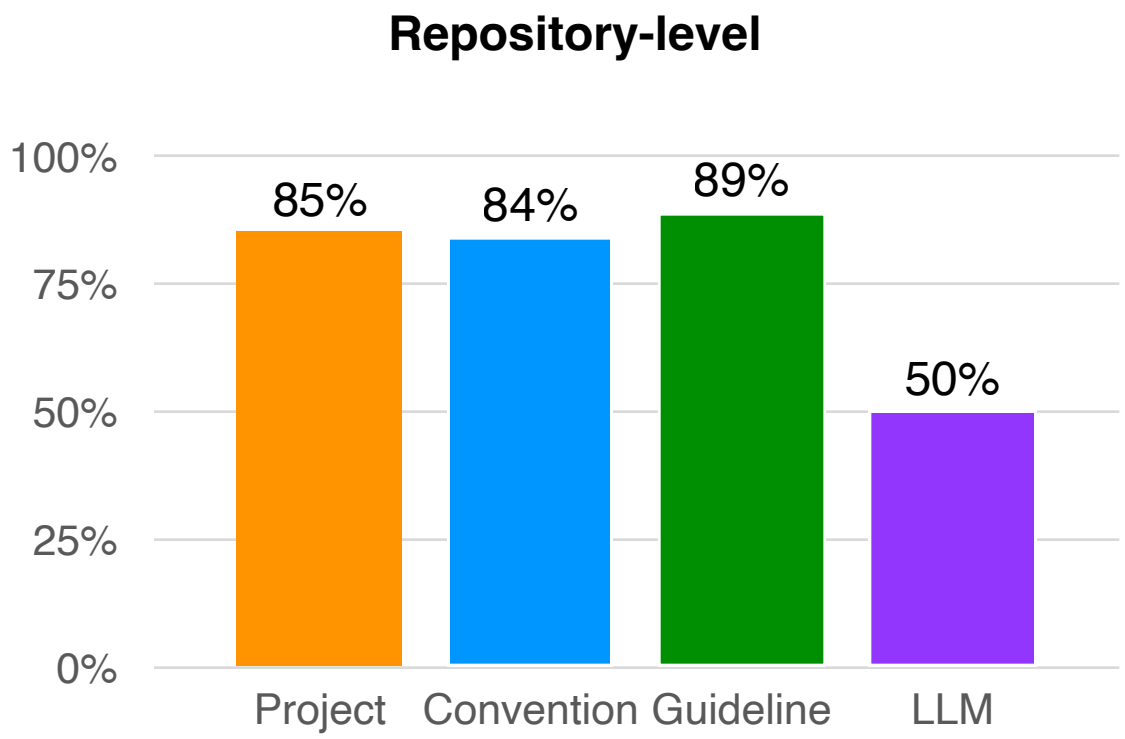
Different Types of Information in Cursor Rules

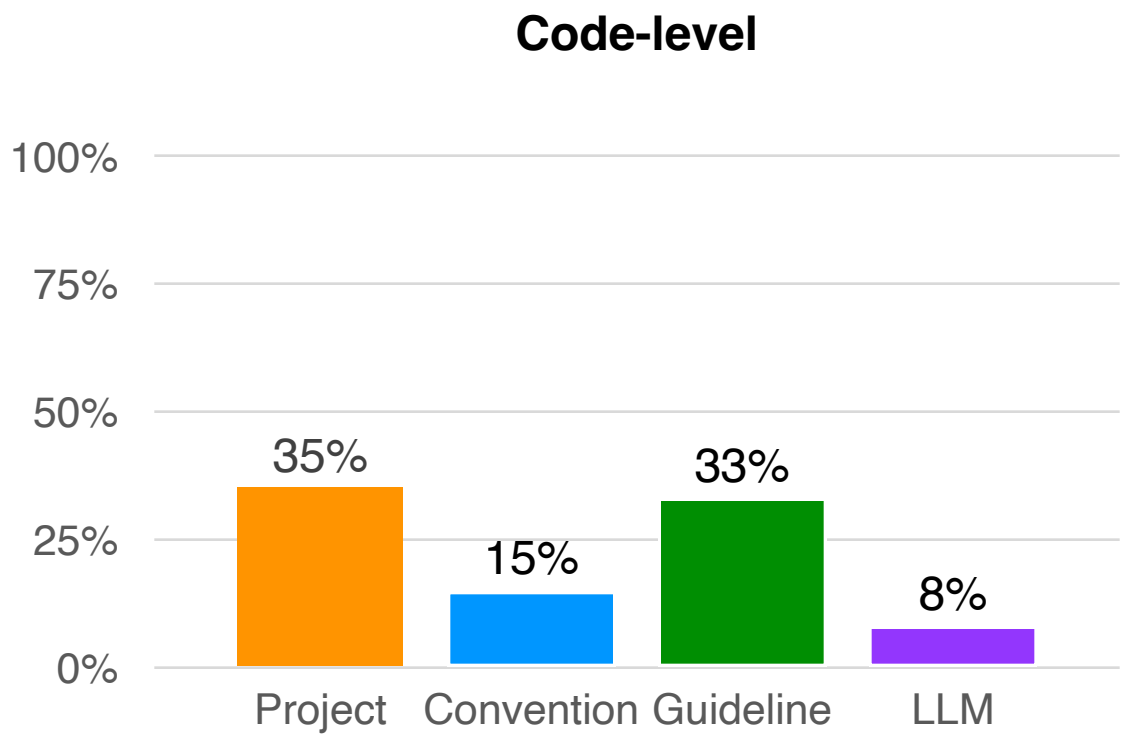
This rule provides standards for frontend components:

- `setup/`: Setting up the environment. — **Project**
- Use CamelCase — **Convention**
- Maintain module separation of concerns. — **Guideline**
- Always ask clarifying questions if context is missing — **LLM**
- …

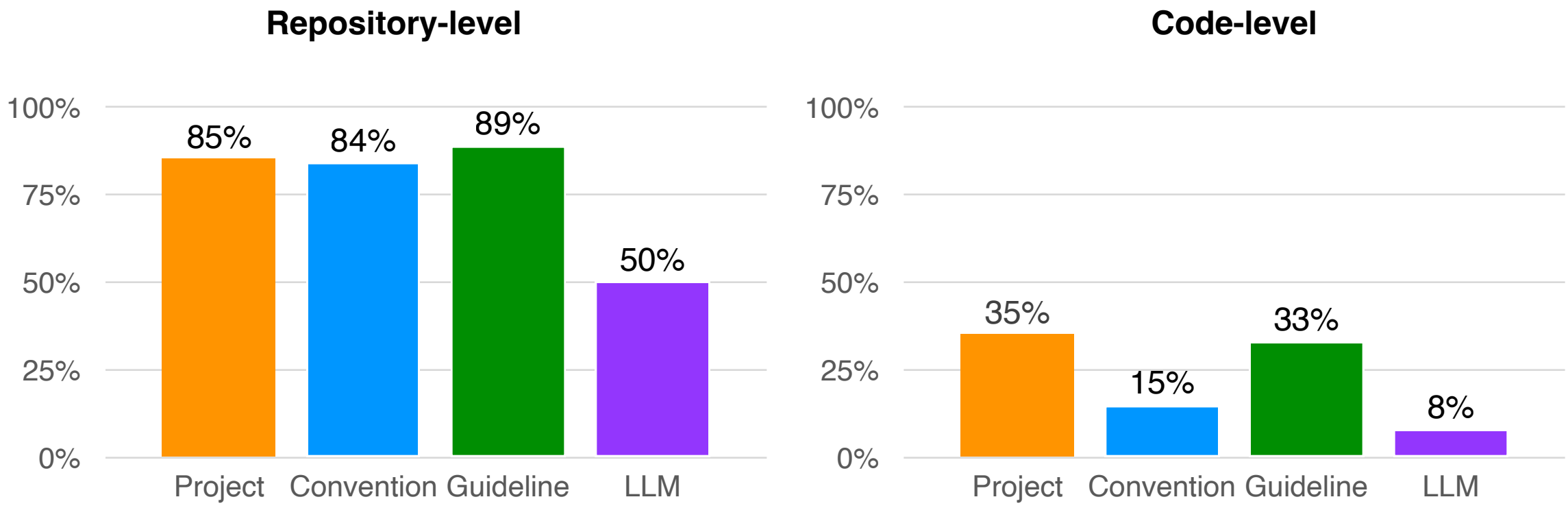
For LLMs on how to generate responses

Distribution of Information Types



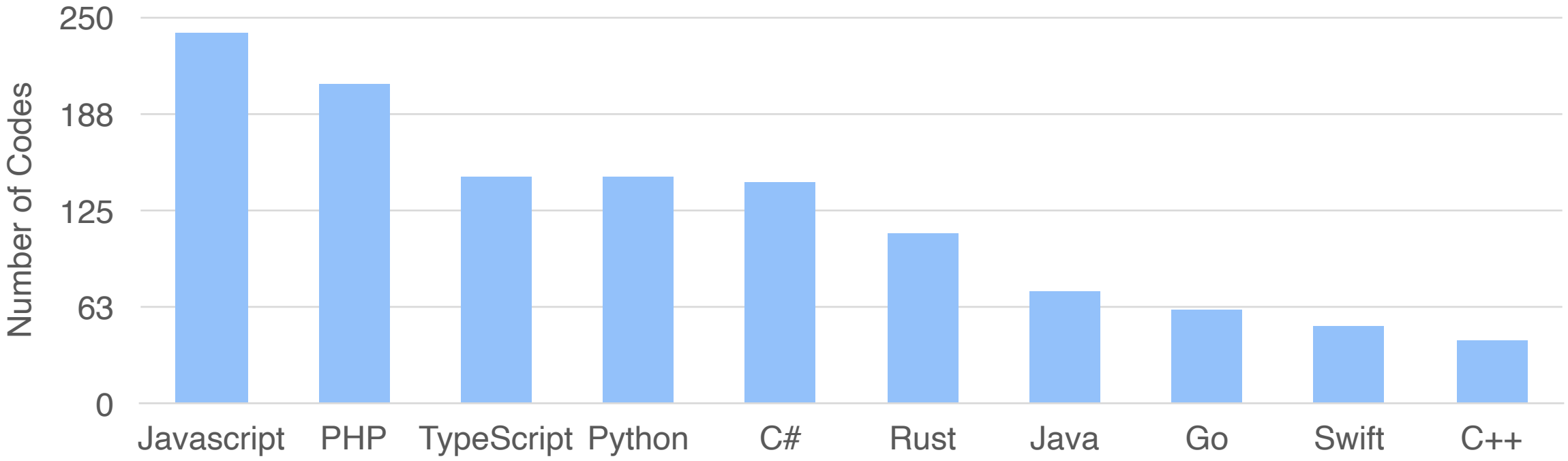


Distribution of Information Types

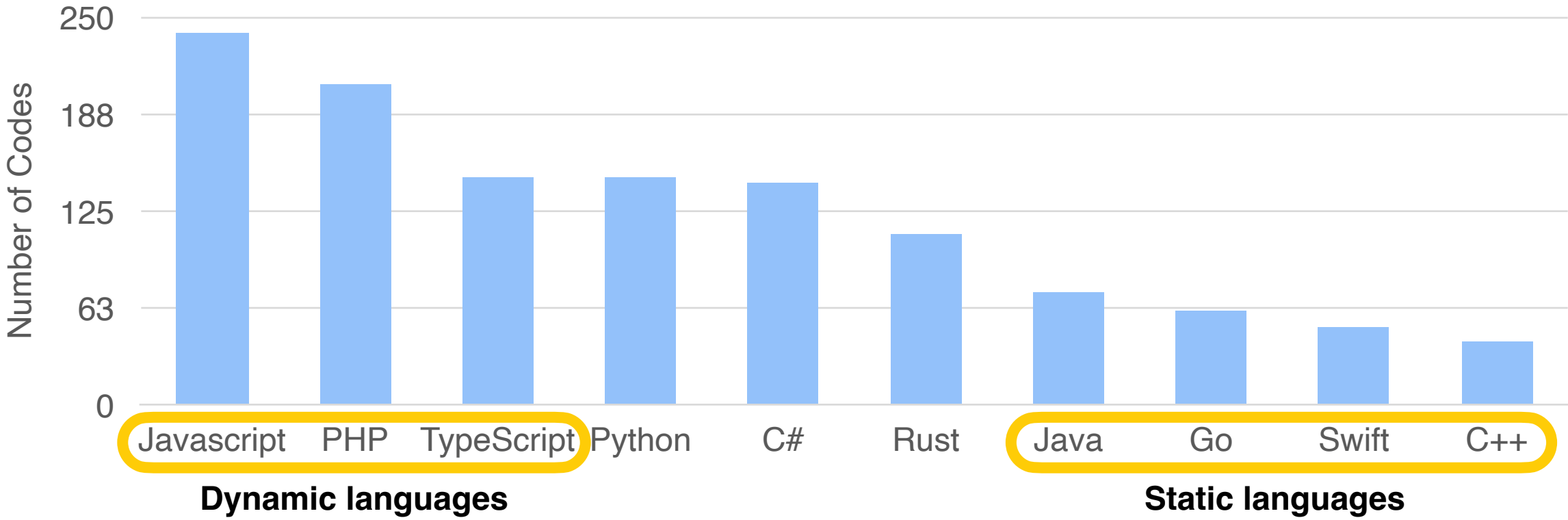


Cursor rules still mirror traditional project documentation, possibly because developers default to familiar documentation practices.

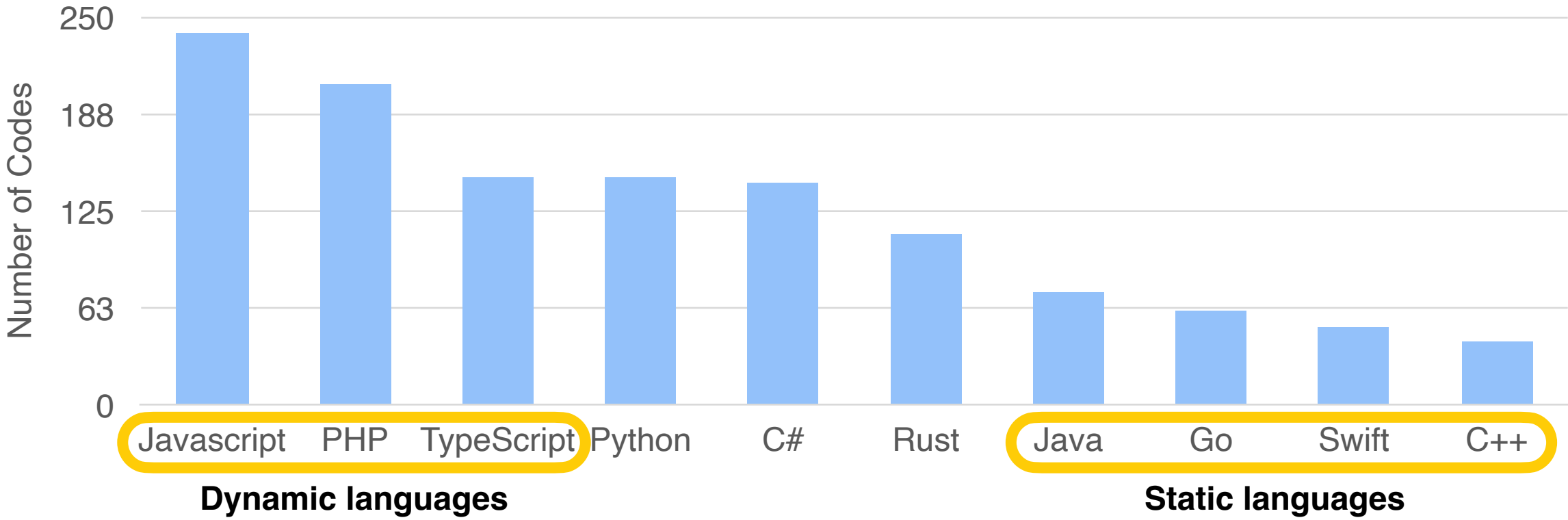
Variation in Rules Across PLs



Variation in Rules Across PLs



Variation in Rules Across PLs



Rule/skill generation tools should tailor their output to the project's language and framework, rather than relying on generic templates

Developers Misunderstand Rule Semantics

performance-optimizer.mdc

```
triggers:
  - file.modified
  - pull_request.created
  - pull_request.updated
  - command: "optimize_performance"
```

Performance Analysis Logic

```javascript
// Main function to analyze code for performance issues
function analyzePerformance(files, codebase) {
  const issues = [];
  const suggestions = [];
  let score = 100; // Start with perfect score
```

Developers Misunderstand Rule Semantics

performance-optimizer.mdc

```
triggers:
  - file.modified
  - pull_request.created
  - pull_request.updated
  - command: "optimize_performance"
```

Performance Analysis Logic

```javascript
// Main function to analyze code for performance issues
function analyzePerformance(files, codebase) {
  const issues = [];
  const suggestions = [];
  let score = 100; // Start with perfect score
```

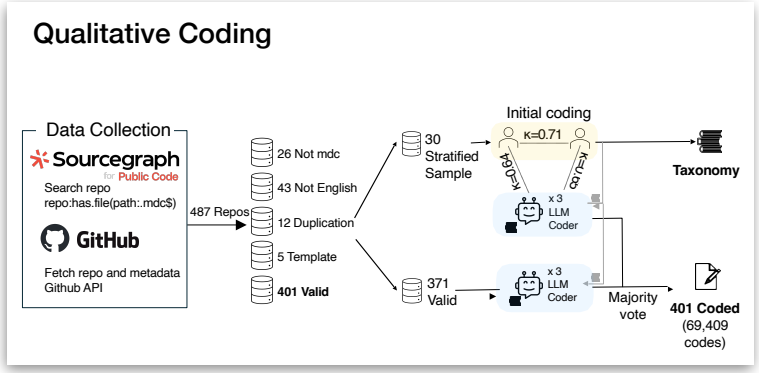
Tools should make transparent how rules are used and which rules are active for each task

Beyond the Prompt:
An Empirical Study of Cursor Rules

Shaokang Jiang (shj@uci.edu) Daye Nam (daye.nam@uci.edu)

UC Irvine Donald Bren School of Information & Computer Sciences

Beyond the Prompt:
An Empirical Study of Cursor Rules

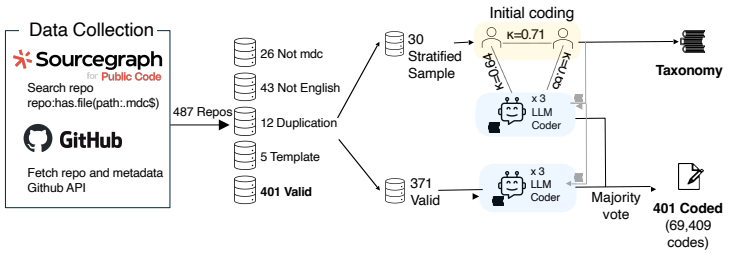


Shaokang Jiang (shj@uci.edu) **Daye Nam** (daye.nam@uci.edu)

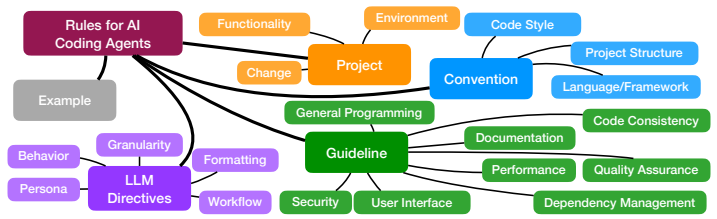
UC Irvine Donald Bren School of Information & Computer Sciences

Beyond the Prompt:
An Empirical Study of Cursor Rules

Qualitative Coding



Taxonomy of Cursor Rule Content Type

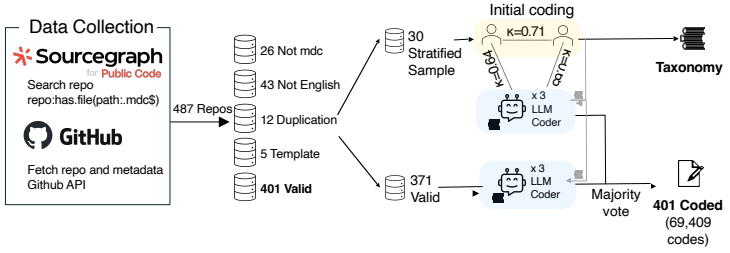


Shaokang Jiang (shj@uci.edu) **Daye Nam** (daye.nam@uci.edu)

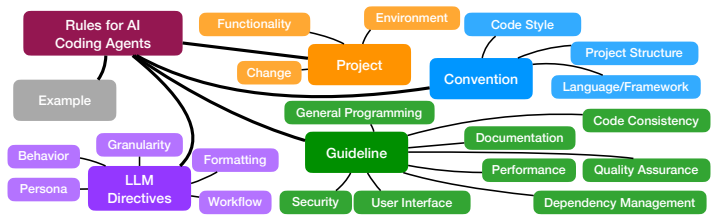
UC Irvine Donald Bren School of Information & Computer Sciences

Beyond the Prompt:
An Empirical Study of Cursor Rules

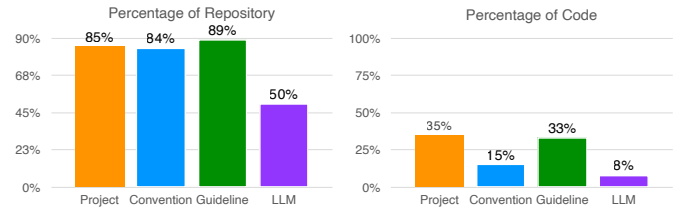
Qualitative Coding



Taxonomy of Cursor Rule Content Type

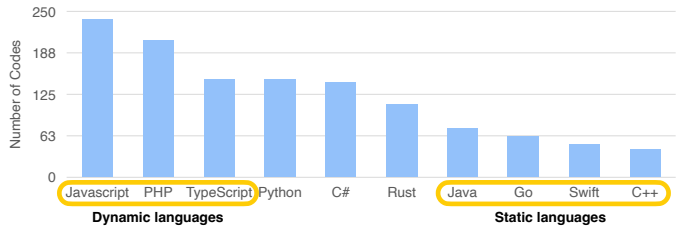


Distribution of Context Types



Cursor rules still mirror traditional project documentation, possibly because developers default to familiar documentation practices.

Variation in Context Types Across PLs



Rule/skill generation tools should tailor their output to the project's language and framework, rather than relying on generic templates

Developers Misunderstand Rule Semantics

performance-optimizer.mdc

```
## Rule Definition

This rule analyzes code changes for potential performance
   issues and suggests optimizations.

## Performance Analysis Logic

```javascript
// Main function to analyze code for performance issues
function analyzePerformance(files, codebase) {
  const issues = [];
  const suggestions = [];
  let score = 100; // Start with perfect score
```

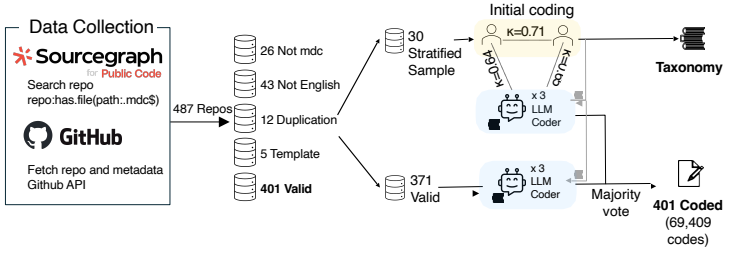
1) Future work should investigate which rule aspects are most effective,
2) Tools should make transparent which rules are active

Shaokang Jiang (shj@uci.edu) **Daye Nam** (daye.nam@uci.edu)

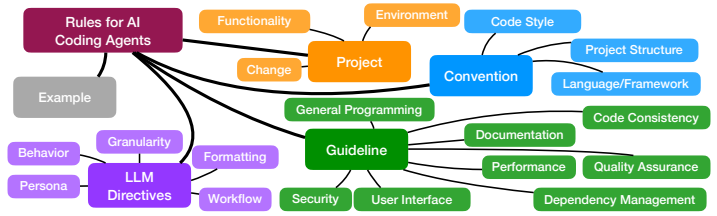
UC Irvine Donald Bren School of Information & Computer Sciences
```

Beyond the Prompt:
An Empirical Study of Cursor Rules

Qualitative Coding

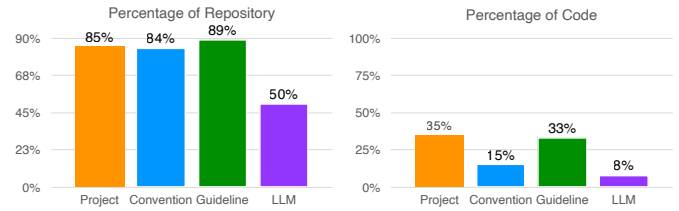


Taxonomy of Cursor Rule Content Type



Full Paper

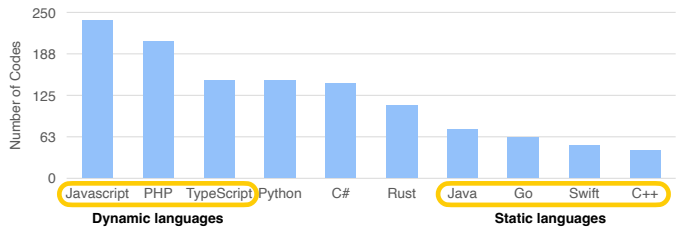
Distribution of Context Types



Cursor rules still mirror traditional project documentation, possibly because developers default to familiar documentation practices.

9

Variation in Context Types Across PLs



Rule/skill generation tools should tailor their output to the project's language and framework, rather than relying on generic templates

10

Developers Misunderstand Rule Semantics

performance-optimizer.mdc

```
## Rule Definition

This rule analyzes code changes for potential performance
   issues and suggests optimizations.

## Performance Analysis Logic

```javascript
// Main function to analyze code for performance issues
function analyzePerformance(files, codebase) {
  const issues = [];
  const suggestions = [];
  let score = 100; // Start with perfect score
```

1) Future work should investigate which rule aspects are most effective,
2) Tools should make transparent which rules are active

11

Shaokang Jiang (shj@uci.edu) **Daye Nam** (daye.nam@uci.edu)

UC Irvine Donald Bren School of Information & Computer Sciences
```