LERO
Software for
a better world

UCD DUBLIN

Taighde Éireann
Research Ireland

# SpareCodeSearch

## Searching for Code Contexts When You Have No Spare GPUs


***Minh Nguyen***

*University College Dublin*


Seoul - 18th November 2025

# A little bit of background information...

2022-2024
R&D team
FPT Software
Vietnam


CODETEXT

Creating AST-based tools for mining and processing big code [1]

CODE VISTA

docify

Building in-house AI Coding Assistant [2]

- Code Completion
- RAG-based QA
- Docstring generation

[1] The Vault: A Comprehensive Multilingual Dataset for Advancing Code Understanding and Generation (Nguyen et al., EMNLP 2023)

[2] Envisioning the Next-Generation AI Coding Assistants: Insights & Proposals. (Nghiem, et al 2024., IDE '24)

# Repository-level Code Completion???

Learning from competitors and how they do it ....

Source Graph

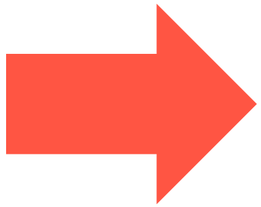
Cody (Now Amp)

[3] Cody is cheating, 2023

[4] Open-sourcing Cody, 2023

[5] The life-cycle of AI code completion, 2023

[6] How Cody understand your codebase, 2024



- Tree-sitter for parsing
- Cursor location for syntactic trigger
- **Zoekt** for lexical (key-word based) code search

## From an unfinished project...

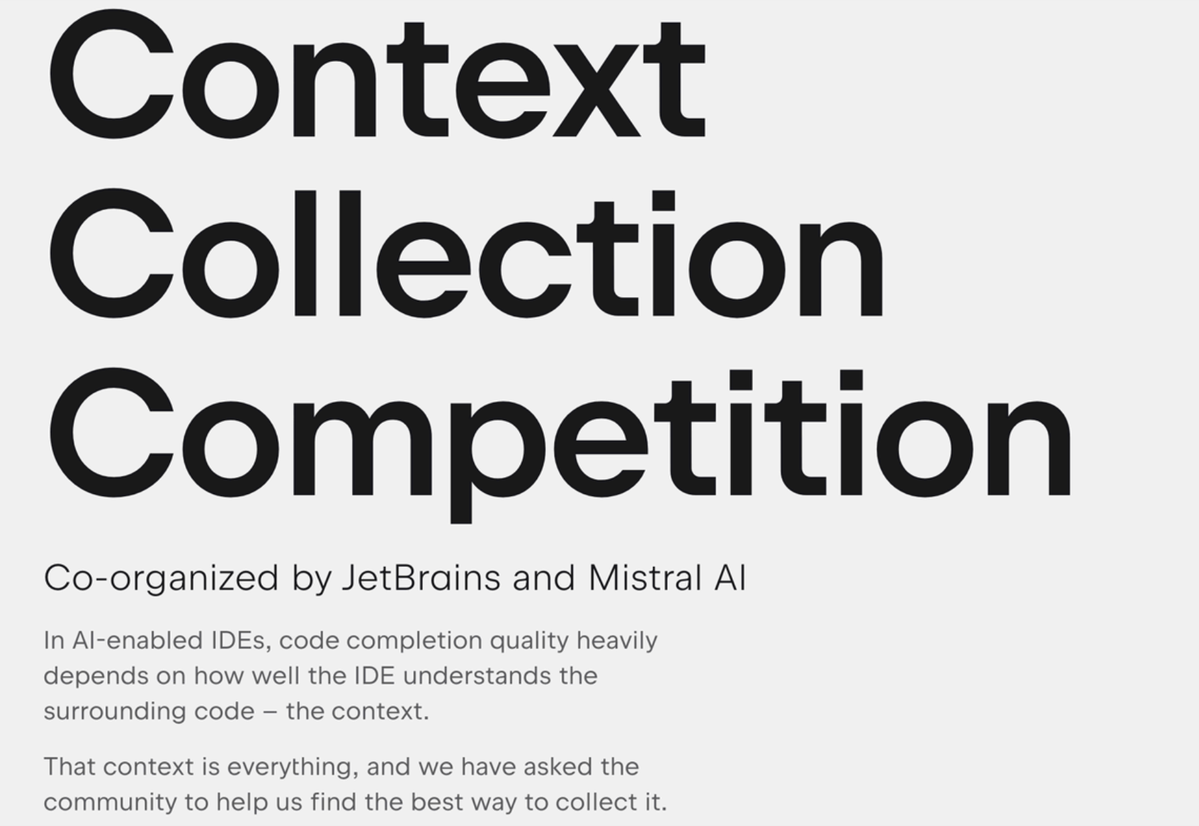
Minh Nguyen

PhD student in Computer Science

Khanh Nghiem

MSc student in Information Study

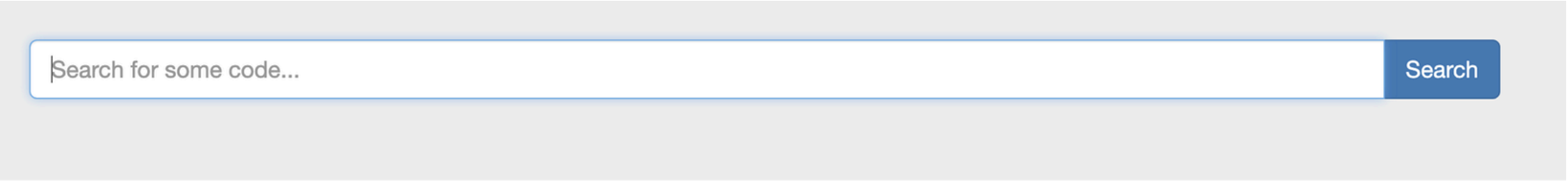
## ... to SpareCodeSearch in Code Context Competition




**Motivation**: To prove that lexical code search is efficient for in-IDE repository-level code completion

**Requirement**: Lightweight, extensible, and polyglot(generalizable to different programming languages), and model agnostic

# Something interesting about Zoekt...




**Search examples:**

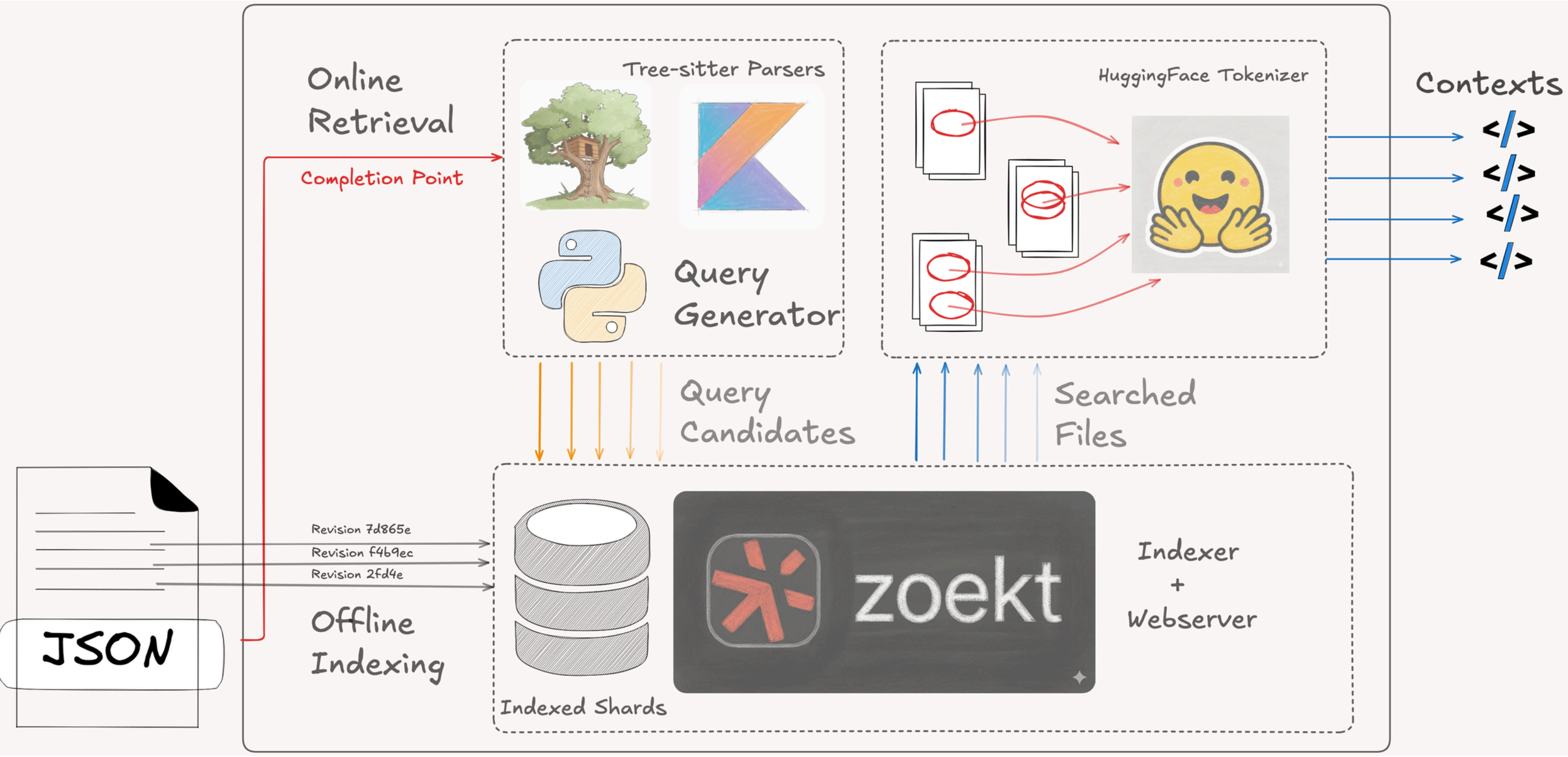
| | |
|---|---|
| needle | search for "needle" |
| thread or needle | search for either "thread" or "needle" |
| class needle | search for files containing both "class" and "needle" |
| class Needle | search for files containing both "class" (case insensitive) and "Needle" (case sensitive) |
| class Needle case:yes | search for files containing "class" and "Needle", both case sensitively |
| "class Needle" | search for files with the phrase "class Needle" |
| needle -hay | search for files with the word "needle" but not the word "hay" |
| path file:java | search for the word "path" in files whose name contains "java" |
| needle lang:python | search for "needle" in Python source code |
| f:\.c$ | search for files whose name ends with ".c" |
| path -file:java | search for the word "path" excluding files whose name contains "java" |
| foo.*bar | search for the regular expression "foo.*bar" |
| -(Path File) Stream | search "Stream", but exclude files containing both "Path" and "File" |
| -Path\ file Stream | search "Stream", but exclude files containing "Path File" |
| sym:data | search for symbol definitions containing "data" |
| phone r:droid | search for "phone" in repositories whose name contains "droid" |
| phone archived:no | search for "phone" in repositories that are not archived |
| phone fork:no | search for "phone" in repositories that are not forks |
| phone public:no | search for "phone" in repositories that are not public |
| phone b:master | for Git repos, find "phone" in files in branches whose name contains "master". |
| phone b:HEAD | for Git repos, find "phone" in the default ('HEAD') branch. |

**To list repositories, try:**

| | |
|---|---|
| r:droid | list repositories whose name contains "droid". |
| r:go -r:google | list repositories whose name contains "go" but not "google". |

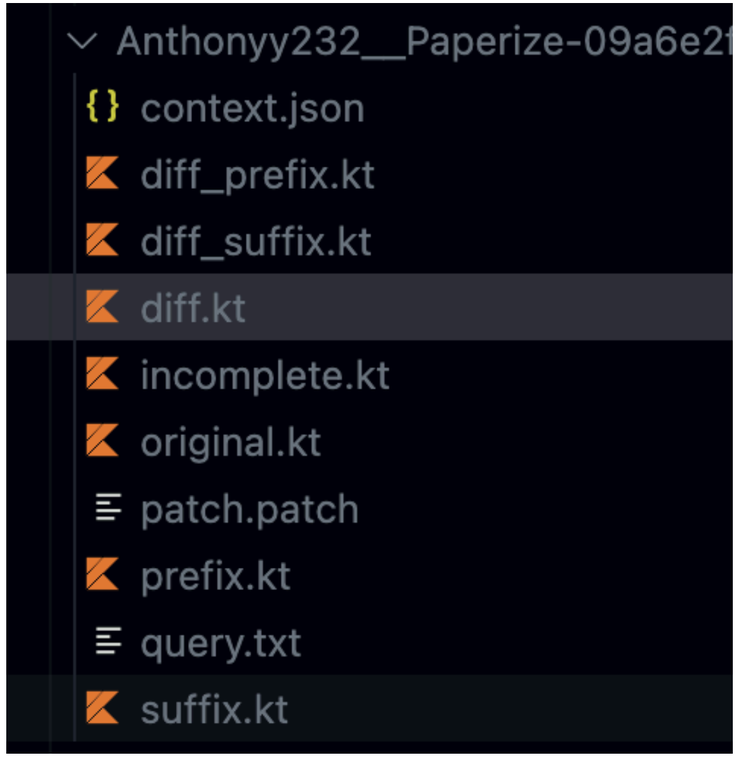
- Friendly GUI
- Extensible Search API
- Support multi-repo indexing
- **Ctags** for universal symbols extraction
- Scalable and Deployable (verified by Google, Gitlab Sourcegraph)

# Overview Architecture



# Initial (and chaotic) development on Jupyter-notebook

# Locating Completion point




```
                DropdownMenu(
                    expanded = expanded,
                    onDismissRequest = {expanded = false},
                ) {
                    DropdownMenuItem(
                        text = { Text(stringResource(R.string.
                        dropdownmenu_settings)) },
                        onClick = {
                            navController.navigate(SettingsNavScreens.
                            Settings.route) {
                                p
/**<FIM>*/
                            }
                        }
                    )
                }

```

Concatenate prefix + suffix → Generate Diff → Locate the Completion Point location within the Diff

# Symbols Gathering with Tree Sitter

```
;;; Class definitions

;; Regular classes, data classes, sealed classes, abstract
    classes
(class_declaration
      (type_identifier) @class.name)

;; Object declarations (singletons)
(object_declaration
   (type_identifier) @object.name)

;; Companion objects
(class_body
   (companion_object
      (type_identifier)? @companion.name) ; name is optional
   )
```

```
;; Navigation Expression
(navigation_expression) @navigation_expression
```

```
;; Wild Identifiers extraction
(simple_identifier) @identifier
            (type_identifier) @type_identifier
```

```
DropdownMenu(
    expanded = expanded,
    onDismissRequest = {expanded = false},
) {
    DropdownMenuItem(
        text = { Text(stringResource(R.string.
        dropdownmenu_settings)) },
        onClick = {
            navController.navigate(SettingsNavScreens.
            Settings.route) {
                p

            }
        }
    )
}
```

Scheme queries to extract identifiers from the generated diffs

# Query Constructing

```
DropdownMenu(
    expanded = expanded,
    onDismissRequest = {expanded = false},
) {
    DropdownMenuItem(
        text = { Text(stringResource(R.string.
        dropdownmenu_settings)) },
        onClick = {
            navController.navigate(SettingsNavScreens.
            Settings.route) {
                p

            }
        }
    )
}
```

navigate Text stringResource DropdownMenuItem DropdownMenu

navController.navigate SettingsNavScreens.Settings.route SettingsNavScreens.
Settings R.string.dropdownmenu.settings R.string

Text Settings dropdownmenu_settings R navController text

Experiments with AND logic (natively join all the identifiers) → A total **failure**!!!
(Most queries return **empty** search result!)

# Making things easier ...

Using OR logic to join the identifiers in the Query

| | |
|---|---|
| functions_classes_naive | `navigate Text stringResource DropdownMenuItem DropdownMenu` |
| functions_classes_or | `navigate or Text or stringResource or DropdownMenuItem or DropdownMenu` |
| functions_classes_top5 | `navigate Text stringResource DropdownMenuItem DropdownMenu` |
| functions_classes_top4 | `navigate Text stringResource DropdownMenuItem` |
| functions_classes_top3 | `navigate Text stringResource` |
| functions_classes_regex | `DropdownMenu.*navigate` |

Regular Expression for fuzzy search

Lower number of search terms

# Making things easier ...

| Query Variation | Kotlin | Python |
|---|---|---|
| functions_classes_naive | 371 | 237 |
| functions_classes_or | 371 | 237 |
| functions_classes_top5 | 214 | 201 |
| functions_classes_top4 | 250 | 209 |
| functions_classes_top3 | 275 | 218 |
| functions_classes_regex | 337 | 230 |
| navigation_naive | 346 | 219 |
| navigation_unpacked | 346 | 219 |
| navigation_unpacked_or | 346 | 219 |
| navigation_unpacked_top5 | 293 | 154 |
| navigation_unpacked_top4 | 307 | 176 |
| navigation_unpacked_top3 | 320 | 191 |
| navigation_regex | 346 | 219 |
| identifiers_naive | 396 | 244 |
| identifiers_or | 396 | 244 |
| identifiers_top5 | 386 | 226 |
| identifiers_top4 | 387 | 230 |
| identifiers_top3 | 388 | 236 |
| identifiers_regex | 396 | 244 |

Number of queries generated for each query construction strategy, for Kotlin and Python public datasets

Iteratively send all requests to Zoekt server until a succesful search results is returned

# Cross-shard searching also helped ...

# Preliminary results on Public phase

| | Single-shard Query | Cross-shard Query |
|---|---|---|
| Python Hit | 213 | 239 |
| Python Miss | 34 | 8 |
| Kotlin Hit | 344 | 390 |
| Kotlin Miss | 56 | 10 |

## Zoekt query hit rate

| Rank | Participant team | Average ChrF (↑) | Mellum ChrF (↑) | Codestral ChrF (↑) | Qwen-Coder ChrF (↑) |
|---|---|---|---|---|---|
| 1 | spareCodeComplete (heuristic-mf3-mt6-pd) | 0.7125 | 0.6791 | 0.7442 | 0.7143 |

## Kotlin track

| Rank | Participant team | Average ChrF (↑) | Mellum ChrF (↑) | Codestral ChrF (↑) | Qwen-Coder ChrF (↑) |
|---|---|---|---|---|---|
| 1 | NoMoreActimel | 0.7469 | 0.6950 | 0.8034 | 0.7424 |
| 2 | spareCodeComplete (heuristic-mf3-mt6-pd.jsonl) | 0.6152 | 0.5563 | 0.6568 | 0.6324 |

## Python track

Contexts are generated on a consumer Macbook Air M3 with only 16Gb RAM

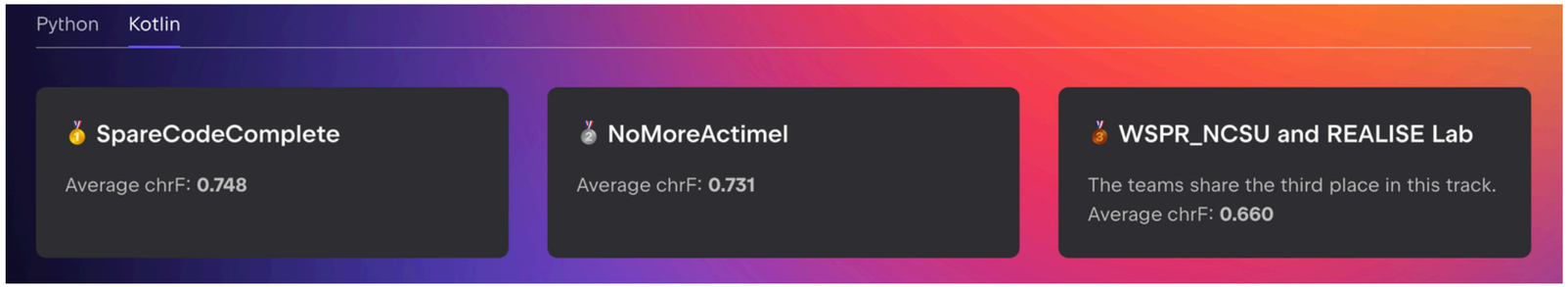
# Difficulties faced...

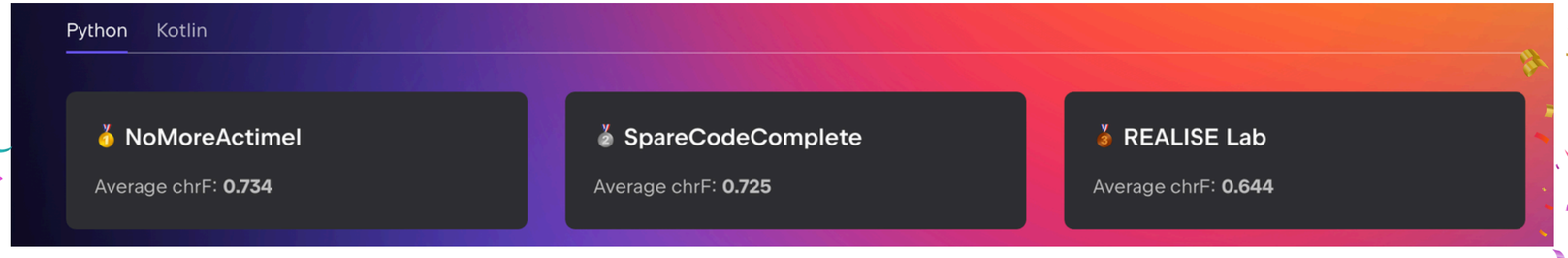
- Starting with Jupyter notebooks is great for initial experiments, but **terrible** when modularizing and connecting everything later
- Setting and running-up Zoekt took quite an effort:
  - Instructions unclear, had to rewrite the whole Docker compose
  - Lack necessary Go skills to extend the base image for handling rate limit and server keep-alive.

# The private phase...


**After countless of times Egor have to re-run the Dockerized solution...**

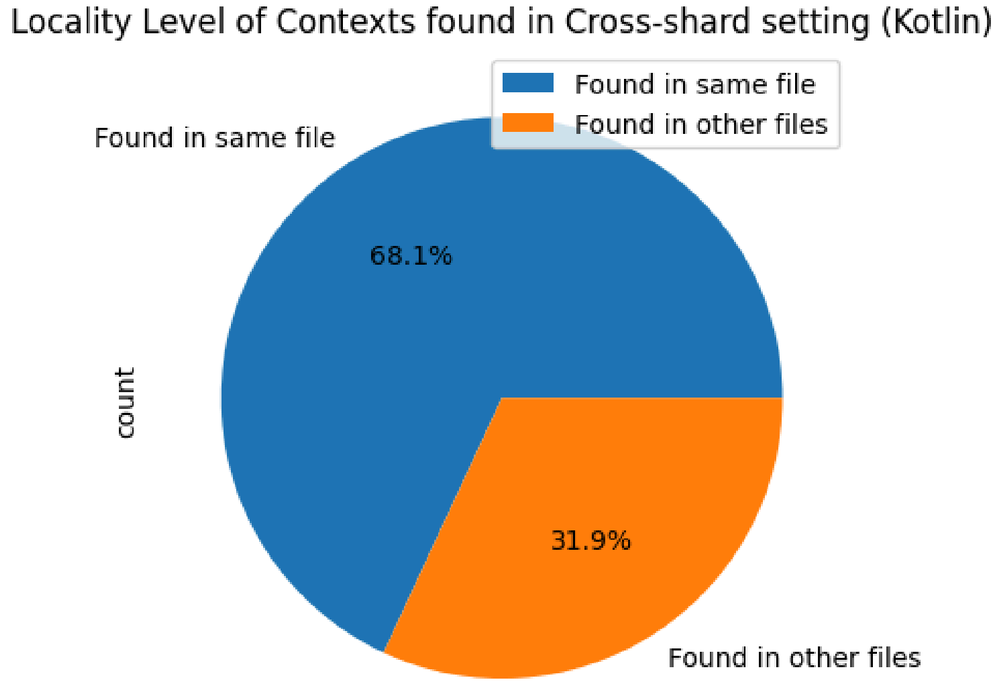
# It worked!

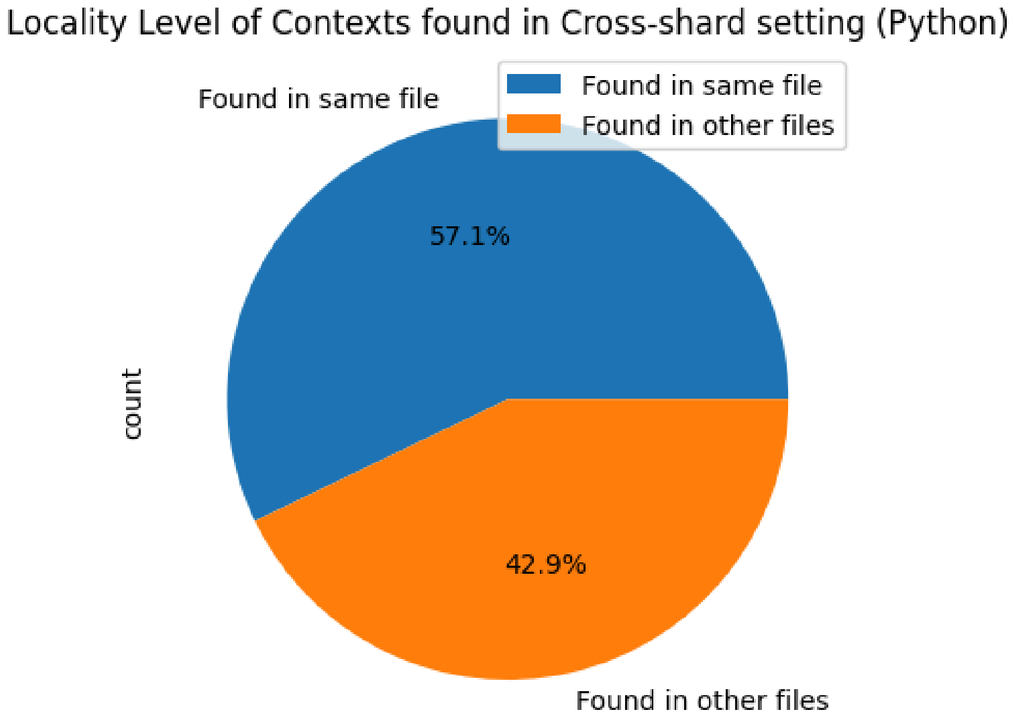

Python
Kotlin
SpareCodeComplete
Average chrF: 0.748
NoMoreActimel
Average chrF: 0.731
WSPR_NCSU and REALISE Lab
The teams share the third place in this track.
Average chrF: 0.660


Python
Kotlin
NoMoreActimel
Average chrF: 0.734
SpareCodeComplete
Average chrF: 0.725
REALISE Lab
Average chrF: 0.644

# However, threats of validity...

- Possible data leakage in Cross-shard settings: code snippet from a **later** revision might be retrieved to complete a code snippet from an **earlier** revision



# Needs for further ablation study

- Existing experiments only  analyze the relationship: hit rate -> final chRF
- Ablation experiments can are needed to understand the  different **symbol extraction strategies** and **query constructs** affect  hit rate the most and the **completion quality**.

| Query Variation |
|---|
| functions_classes_naive |
| functions_classes_or |
| functions_classes_top5 |
| functions_classes_top4 |
| functions_classes_top3 |
| functions_classes_regex |
| navigation_naive |
| navigation_unpacked |
| navigation_unpacked_or |
| navigation_unpacked_top5 |
| navigation_unpacked_top4 |
| navigation_unpacked_top3 |
| navigation_regex |
| identifiers_naive |
| identifiers_or |
| identifiers_top5 |
| identifiers_top4 |
| identifiers_top3 |
| identifiers_regex |

???

???

```
;;; Class definitions

;; Regular classes, data classes, sealed classes, abstract
    classes
(class_declaration
      (type_identifier) @class.name)

;; Object declarations (singletons)
(object_declaration
   (type_identifier) @object.name)

;; Companion objects
(class_body
   (companion_object
      (type_identifier)? @companion.name) ; name is optional
   )
```

```
;; Navigation Expression
(navigation_expression) @navigation_expression
```

```
;; Wild Identifiers extraction
(simple_identifier) @identifier
            (type_identifier) @type_identifier
```

# Possible future extensions

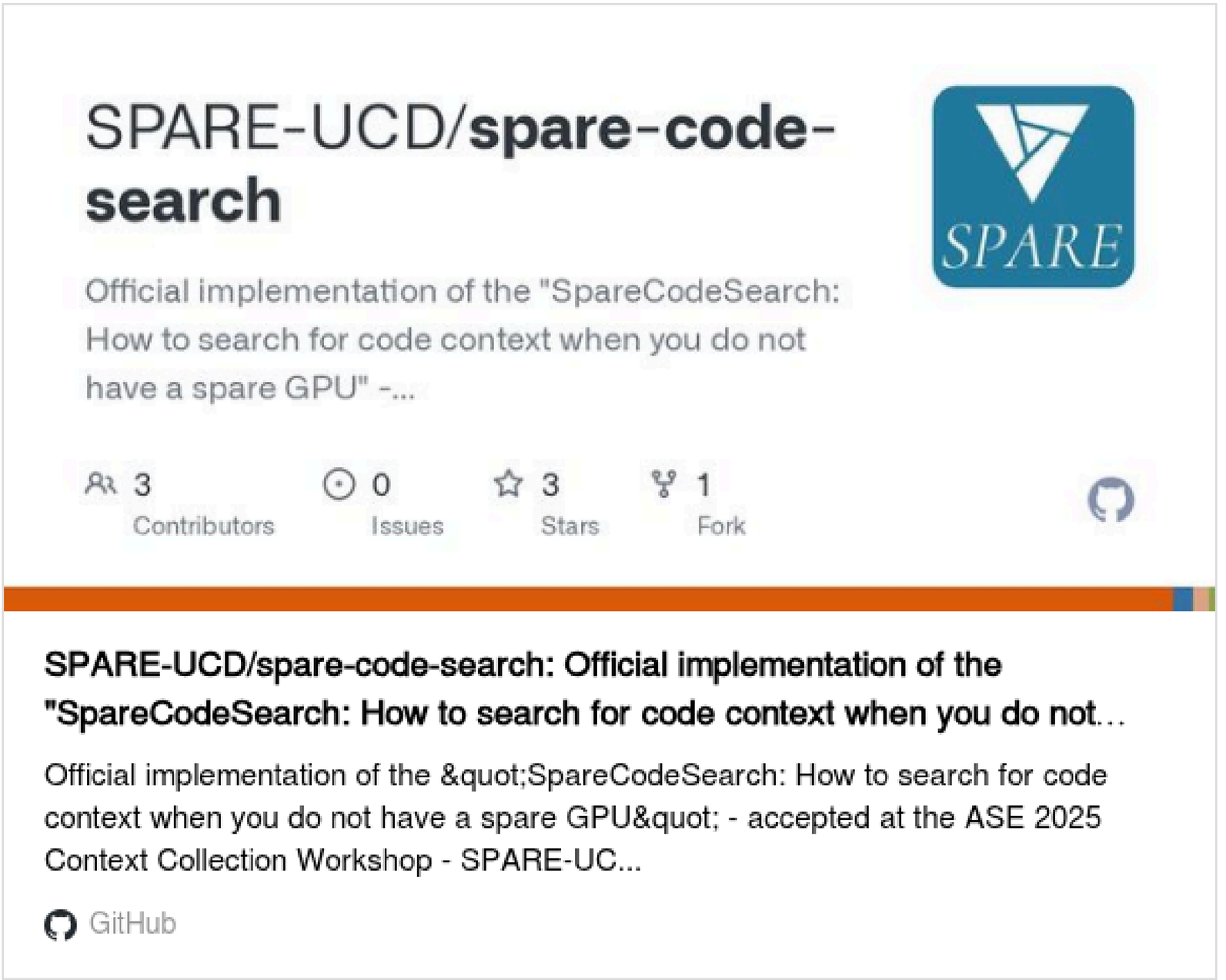
- A possible PoC with an existing in-IDE code completion tool?
- Zoekt indexer has the option to be deployed on-cloud → Centralized indexer for enterprise-scale code completion?
- Context-sensitive secure code generation)?

???

???

# In conclusion, SpareCodeSearch successed due to

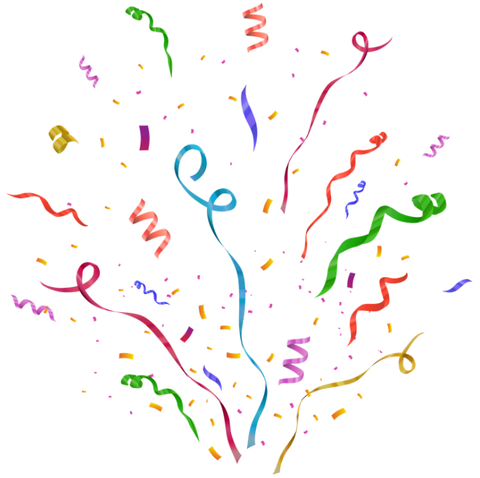
- Han-Wen Nienhuys, ex-Google, creator of Zoekt
- Supports from my supervisors in UCD: Liliana Pasquale and Alzubair Hassan
- Inspirations from Khanh Nghiem in UWash Ischool
- All the helps from Jetbrains and MistralAI folks
- And possibly my incompetencies to handle commit hashes properly




**SPARE-UCD/spare-code-search: Official implementation of the "SpareCodeSearch: How to search for code context when you do not…**

Official implementation of the "SpareCodeSearch: How to search for code context when you do not have a spare GPU" - accepted at the ASE 2025 Context Collection Workshop - SPARE-UC...

GitHub



# Thank you!

# Q & A