

From Fuzzy to Fluent

Semantic Code Search in Vim/Neovim

By OKABE Gota (@sirasagi62)



I'm **Sirasagi62**



Sendai, Japan



Tohoku Univ.



TS/Go/Python/C++ ...



About 10 years



Finder

また

*Fuzzy Finder*か!

Yet Another

Fuzzy Finder AGAIN!?

No, It's

Fluent.

Lingur

Motivation

Finding code is **difficult**

"Searching with different verb with **same meaning**"

"Can't think of **what words** to search"

Motivation

Finding code is **difficult**

"Searching with different verb with **same meaning**"

"Can't think of **what words** to search"

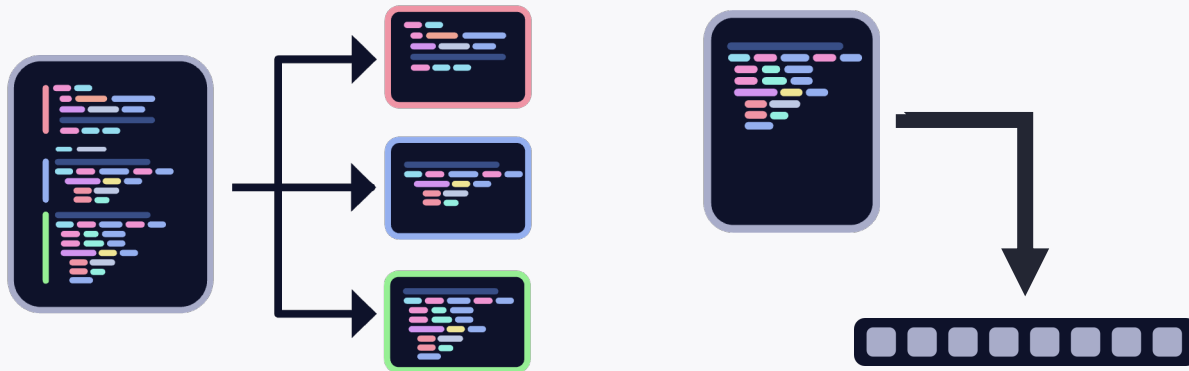
→ Cannot use **synonym** or **sentences**

What is Fluent Finder and flf-vim



flf-vim finds code **semantically**

→ AI **understands** the meaning of the code



Example 1: FlfDir



 flf-vim

```
1 *
データベースを閉じる
src/buffercore.ts#deinit
src/dircore.ts#deinit
src/buffercore.ts#[Symbol.dispose]
src/dircore.ts#[Symbol.dispose]
src/index.ts#destroyTTY
src/dircore.ts#db
src/buffercore.ts#db
src/index.ts#bindKeys
src/index.ts#screen
src/index.ts#options
src/index.ts#input
src/dircore.ts#executeQuery
src/index.ts#preview
src/index.ts#createScreen
src/dircore.ts#isJsonOutput
src/dircore.ts#core
src/index.ts#list
src/dircore.ts#formatResult
src/dircore.ts#init

deinit() {
    if (this.db) this.db.close()
}

TERMINA
-- TERMINAL --
```

Problem



 [flf-vim](#)

Problem

":FlfDir" takes a **very long** time

Example: Analyze vim source code
→ It takes



 flf-vim

Problem

":FlfDir" takes a **very long** time

Example: Analyze vim source code

→ It takes **over 1 hour**



 flf-vim

Problem

//

Not Fluent

de



 flf-vim

Example 2: FlfBuf



 flf-vim

```
charset.c vimrun.c
src > charset.c
1 文字列 サイズ
2 src/charset.c#int vim_strsize(ch
3 src/charset.c#int linetabsize_st int
4 src/charset.c#int win_chartabsiz vim_strsize(char_u *s)
5 src/charset.c#int win_linetabsiz {
6 src/charset.c#int linetabsize(wi return vim_strnsiz(s,
7 src/charset.c#int vim_strnsiz(c (int)MAXCOL);
8 src/charset.c#int chartabsiz(ch }
9 src/charset.c#int linetabsize_eo
10 src/charset.c#void clear_chartab
11 src/charset.c#int linetabsize_no
12 src/charset.c#int linetabsize_co
13 src/charset.c#char_u *
14 src/charset.c#int getwhitecols(c
15 src/charset.c#void win_linetabsi
16 src/charset.c#int init_chartab(v
17 src/charset.c#char_u *
18 src/charset.c#int vim_iswordp(ch
19 src/charset.c#int lbr_chartabsiz
20 src/charset.c#static int

TERMINAL
-- TERMINAL --
```

Embedding Model

Use SLM(Small Language Model)

- Safe
- Cost-Effective
- Fast



 flf-vim

Embedding Model

"ruri-v3-70m-code-v-0.1"

- **Finetuned "ruri-v3"**



 **flf-vim**

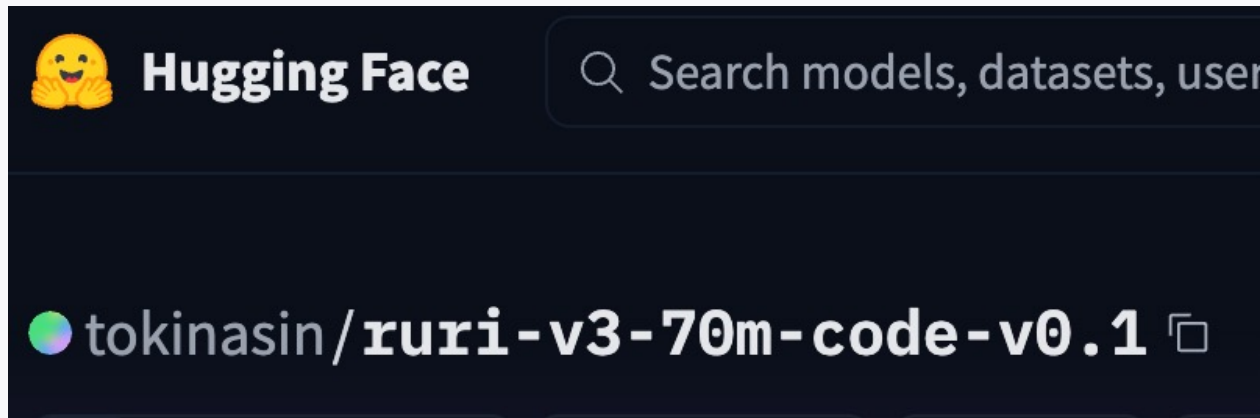
Embedding Model

"ruri-v3-70m-code-v-0.1"

- Finetuned "ruri-v3"



 flf-vim



Conclusion



 flf-vim

flf-vim allows searching
with natural languages

More things to talk...



 flf-vim

- Unclear behavior in Vim's ``:buffer``
- A bug I found during development that crashes Bun.js in just 4 lines
- Hacks to support both Vim and Neovim

Thank you!