ICLR

CLARC: C/C++ Benchmark for Robust Code Search

Kaicheng Wang*, Liyan Huang*, Weike Fang, Weihang Wang

University of Southern California

USC University of Southern California

USC Viterbi

Thomas Lord Department of Computer Science

Motivation

- SOTA code search models' overreliance on lexical features instead of semantic understanding.

- Recent code search benchmarks focus on Python, and frequently miss dependencies.

Ground Truth Code

```
void sum_reduce(void* e1, void* e2,
void* res)
{
    int i = *(int*)e1;
    int j = *(int*)e2;

    *(int*)res = i + j;
}
```

Irrelevant Code

```
void vecswap2(unsigned char **a,
unsigned char **b, int n)
{   while (n-- > 0) {
        unsigned char *t = *a;
        *a++ = *b;
        *b++ = t;
    }
}
```

Sim. with Query 0.6195 > 0.4487

Query

This function accepts three pointer arguments referring to memory locations. The first two pointers supply integer values, which are retrieved by dereferencing. The function then computes the sum of these integers and writes the result into the memory location indicated by the third pointer. No value is returned by the function, as the computed sum is stored directly via the provided output pointer.

Sim. with Query 0.4030 < 0.4903

Ground Truth Code

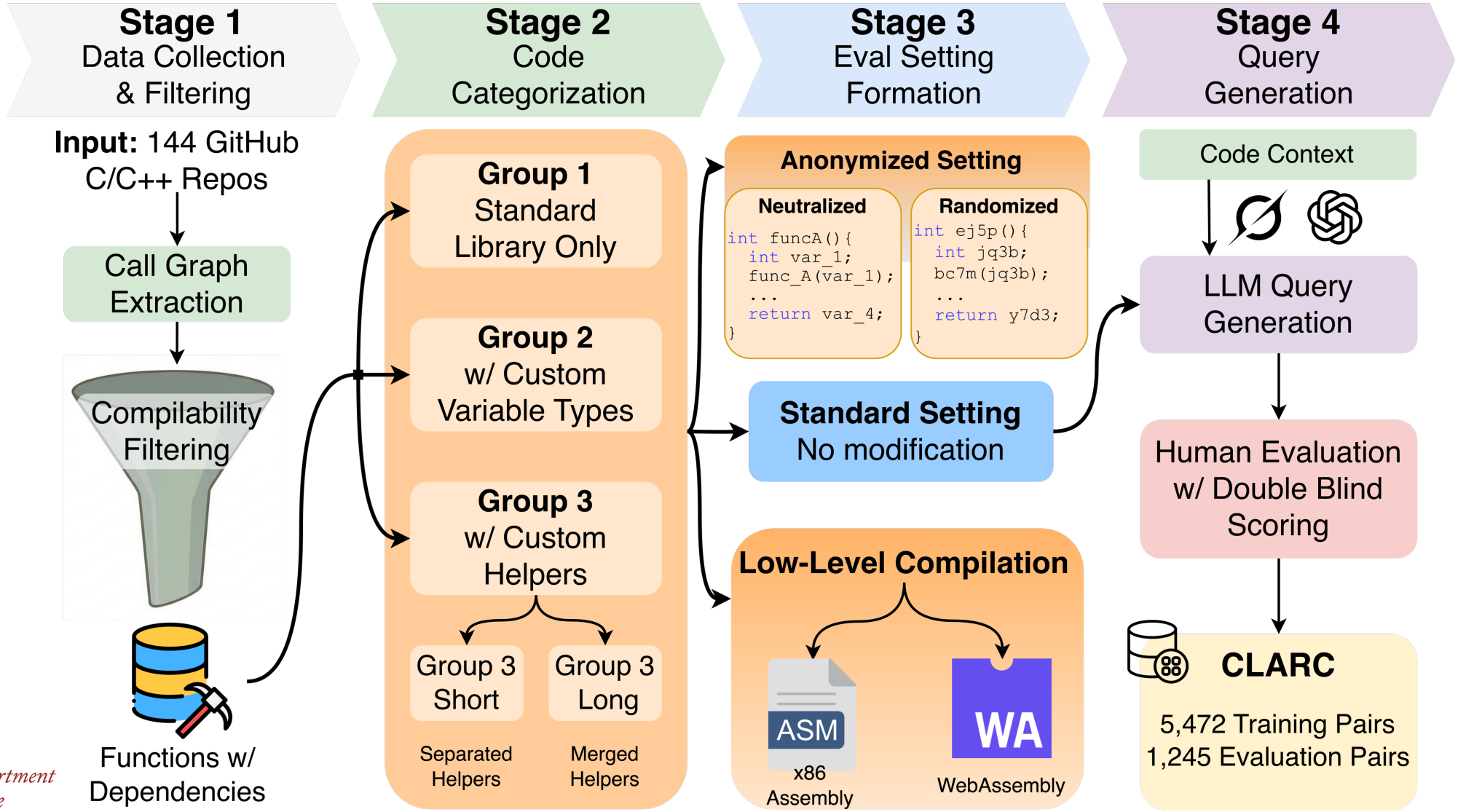
```
void func_0(void* var_0, void*
var_1, void* var_2)
{
    int var_3 = *(int*)var_0;
    int var_4 = *(int*)var_1;

    *(int*)var_2 = var_3 + var_4;
}
```

Irrelevant Code

```
void func_0(unsigned char **var_0,
unsigned char **var_1, int var_2)
{   while (var_2-- > 0) {
        unsigned char *var_3 = *var_0;
        *var_0++ = *var_1;
        *var_1++ = var_3;
    }
}
```

USC Viterbi
Thomas Lord Department of Computer Science

Benchmark Construction



USC Viterbi
Thomas Lord Department of Computer Science

Dataset Summary

Code Categorization

- **Group 1**: Only relies on standard library dependencies.
- **Group2**: Includes custom-defined variable types but no helper functions.
- **Group3**: Invokes user-defined helper functions.
  - **Group 3 Short**: Helpers are treated as separate relevant snippets.
  - **Group 3 Long**: Helpers are combined with the ground truth as one large snippet with contexts.

Evaluation Settings

- **Standard**: The original, unaltered C/C++ source code.
- **Neutralized / Randomized**: Identifiers are replaced with generic / random strings to test models' lexical reliance.
- **Assembly / WebAssembly**: Snippets are compiled to x86 Assembly or WebAssembly to test low-level understanding.

USC Viterbi
Thomas Lord Department of Computer Science

Experiments

Models

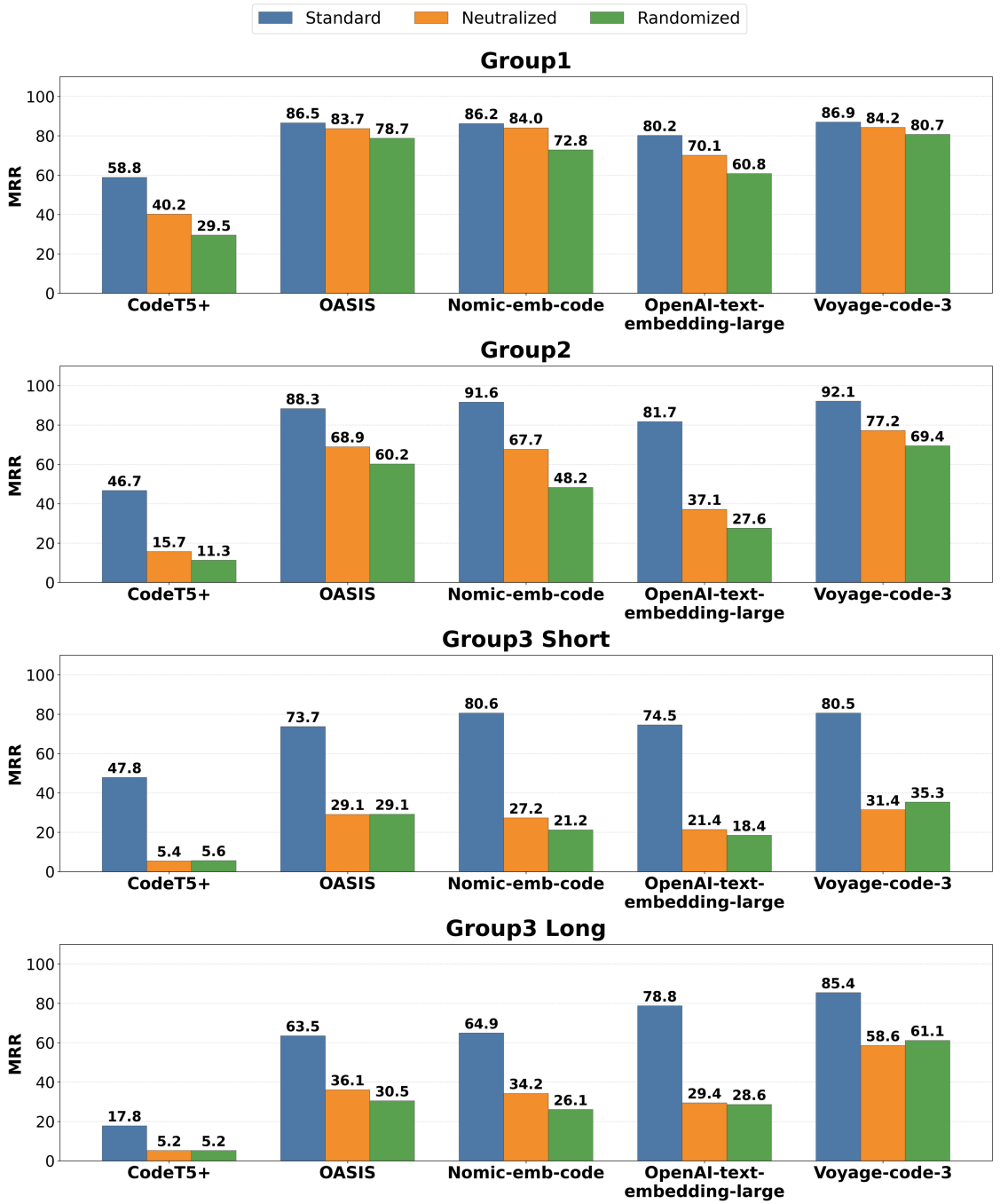
- CodeT5+,
- OASIS,
- Nomic-emb-code,
- Voyage-code-3,
- OpenAI-text-embedding-large

Metrics

- MRR
- NDCG
- MAP
- Recall @ k

Evaluation

- **Overreliance on Lexical Cues**: Retrieval performance universally and substantially declines when identifier names are neutralized or randomized.
- **Complexity Exacerbates the Gap**: The performance drop is most dramatic in complex code (Group 2&3), indicating a heavy reliance on textual cues to infer intricate logic.



USC Viterbi
Thomas Lord Department of Computer Science

Evaluation

The general-purpose code search models are still confused by low-level languages.

Group	Model	Setting	NDCG	MRR	MAP	R@1
Group 1	OpenAI-text-embedding-3-large	Asm	11.50 (-72.07)	8.61 (-71.55)	10.12 (-70.33)	4.25 (-67.42)
		Wasm	8.89 (-74.68)	6.60 (-73.56)	8.29 (-72.16)	3.22 (-68.45)
	Voyage-code-3	Asm	34.12 (-54.87)	29.02 (-57.91)	30.21 (-56.97)	19.88 (-61.11)
		Wasm	31.40 (-57.59)	27.19 (-59.74)	28.81 (-58.37)	20.17 (-60.82)
Group 2	OpenAI-text-embedding-3-large	Asm	6.86 (-79.01)	5.15 (-76.51)	6.70 (-75.03)	2.40 (-69.46)
		Wasm	10.85 (-75.02)	8.14 (-73.52)	10.11 (-71.62)	4.20 (-67.66)
	Voyage-code-3	Asm	35.28 (-58.78)	28.77 (-63.33)	30.19 (-61.92)	17.43 (-68.50)
		Wasm	30.56 (-63.50)	24.36 (-67.74)	25.96 (-66.15)	14.81 (-71.12)
Group 3	OpenAI-text-embedding-3-large	Asm	4.79 (-79.01)	3.46 (-75.30)	5.04 (-73.79)	1.60 (-64.80)
		Wasm	8.90 (-74.90)	5.86 (-72.90)	8.14 (-70.69)	2.63 (-63.77)
	Voyage-code-3	Asm	18.77 (-70.36)	15.20 (-70.23)	17.02 (-68.41)	9.20 (-65.20)
		Wasm	23.17 (-65.96)	19.26 (-66.17)	21.20 (-64.23)	13.16 (-61.24)

USC Viterbi
Thomas Lord Department of Computer Science

Thank you

Check our poster at Poster Session 4 (Apr. 24th 3:15PM – 4:15PM)

Paper

Benchmark

USC Viterbi
Thomas Lord Department of Computer Science