Ask KG Agent: A Multi-Agent Framework for Code Localization Using Code and Knowledge Graphs

▼ BDILab ▼ GitHub

Chanyoung Chung[1], **Jeemin Kim**[1], and **Joyce Jiyoung Whang**[2*]

[1] School of Computing, KAIST [2] Department of AI Computing, KAIST

The 40th Annual Conference on Neural Information Processing Systems (NeurIPS 2026)

KAIST BDILab BIG DATA INTELLIGENCE

Motivation

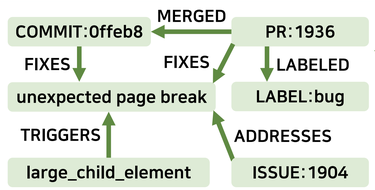
- **Code Localization** is the task of identifying code segments that are **relevant to a given software issue description**.

DataArray.where() can truncate strings with `<U` dtypes.
I want to replace all `"="` occurrences called `sign` with `"<="`.

Function 1: xarray/core/dtypes.py:preprocess_scalar_types()
Function 2: xarray/core/dtypes.py:result_type()

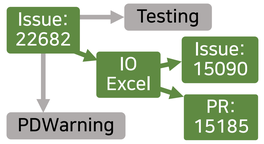
- Limitations of Prior Works
  - Relying solely on functional dependencies within the codebase, **overlooking developer's intent and design rationales**.
  - Treating development traces in **isolation for simple inspection**, **obscuring broader rationales across multiple records**.
- Proposed Approach: **Co-Locator**
  - Leverages **code and knowledge graphs** to integrate **functional dependencies with insights from development history**.

Codebase Knowledge Graphs (KGs)



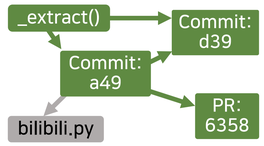
- A **knowledge graph** constructed by extracting triplets that capture **design rationales and prior fixes across development history**.
- **Building an Issue-centric Backbone**
  - Retrieve the **most relevant historical records** to the target issue.
  - Collect records **mentioned within the retrieved records**.
- **Augmenting Contextual Knowledge** to the Issue-centric Backbone
  - For each record, **generate a structured summary**. (LLM Call #1)
  - **Identify key entities** using the generated summary. (LLM Call #2)
  - **Extract triplets** based on the summary and entities. (LLM Call #3)
- **Use Cases of Codebase Knowledge Graphs**

Why xlrd is replaced with openpyxl?



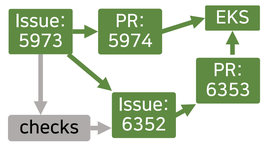
xlrd is unmaintained ... Deprecate warnings ...

Recent code changes around _extract()



Enhance downloads by utilizing subtitles ...

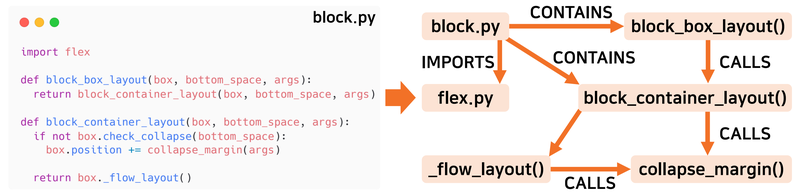
Past PRs related to AWS EKS versions.

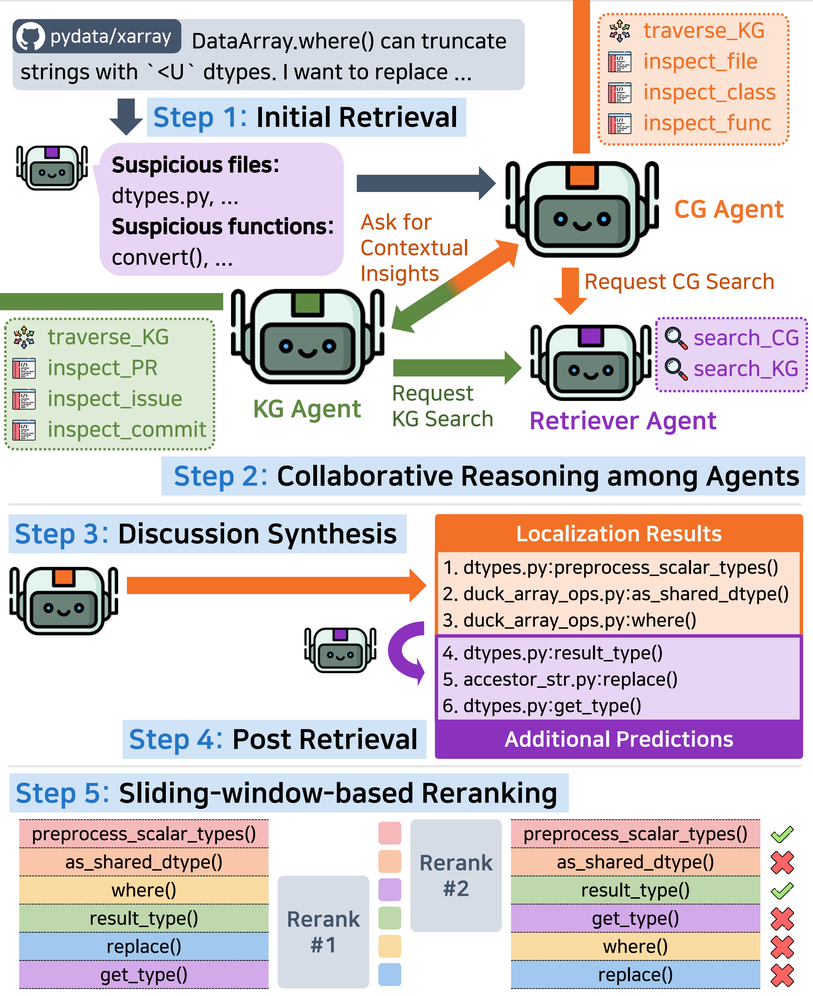


PRs 5974 and 6358 focus on versions ...

Code Graphs (CGs)

- A graph capturing **syntax-level dependencies** among code elements.
  - Nodes: Directory, File, Class, Function, Variable, Library, ...
  - Edges: Contains, Defines, Calls, Imports, Inherits, References
- Constructed via **AST parsing on the source code** of the codebase.





Code Localization Results

- **Co-Locator outperforms 12 state-of-the-art baseline methods.**

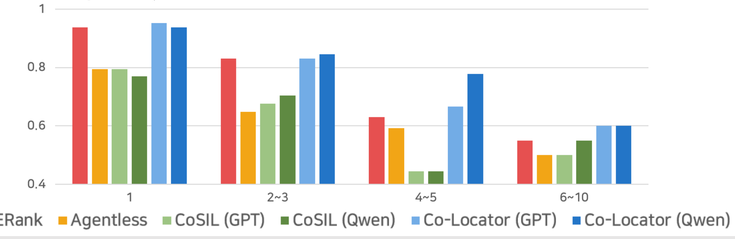
	SWE-bench-Live (Lite)			LocBench		
	File@5	Mod@10	Func@10	File@5	Mod@10	Func@10
Best Baseline	0.787	0.754	0.586	0.811	0.729	0.616
Co-Locator	0.813	0.779	0.610	0.841	0.751	0.648

- **Codebase knowledge graph is beneficial for code localization.**

	SWE-bench-Live (Lite)			LocBench		
	File@5	Mod@10	Func@10	File@5	Mod@10	Func@10
w/o KG	0.798	0.749	0.591	0.833	0.742	0.627
Co-Locator	0.813	0.779	0.610	0.841	0.751	0.648
p-value	0.024	0.012	0.003	0.010	0.029	0.002

Performance by Issue Complexity

- Define complexity levels based on the number of functions to fix.



Impacts on Code Generation

- For each localization method, **generate patches with Agentless using the predicted code locations**.
  - **Oracle**: Provides **ground-truth locations** to Agentless.

	SWERank	Agentless	CoSIL	Co-Locator	Oracle
Resolve Rate (%)	11.1	10.6	9.4	13.1	14.8

Conclusion

- Define **Codebase Knowledge Graphs** that capture **design rationales and developer intents under development traces** of the codebase.
- Propose **Co-Locator**, a collaborative multi-agent framework for code localization **using code and codebase knowledge graphs**.
  - Consider both **functional dependencies and historical context**.
- Co-Locator achieves **state-of-the-art** code localization performance.
  - Shows the **benefits of using historical data in a form of KG**.

* Corresponding Author (jjwhang@kaist.ac.kr) | GitHub: https://github.com/bdi-lab/Co-Locator | Lab Homepage: https://bdi-lab.kaist.ac.kr