Judging an LLM on judging LLM-generated Microservice Documentation: A Supplemental Study and Observations

Dawn Mai, 25R50028
Supervised by Takashi Kobayashi, School of Computing

Agenda

01

Introduction

Background, Motivation

02

Methods

Proposed framework, working with repositories

03

Results

Observation notes

04

Conclusion

Summary of the study

05

References

Related works

06

Q&A

Questions?

Background

Research goal

- Improve the pre-existing, original code from the provided thesis to correctly evaluate and rank the generated documentation for a new, partially-derived dataset from the original repositories.

- Enforce the ideal ranking:
  - KG/GNN → 3+
  - RAG → 1 - 3
  - Baseline → < 2

01

Microservices

Small, independent services that work well together

02

Issues Found

Incorrect documentation generation and scoring with original code

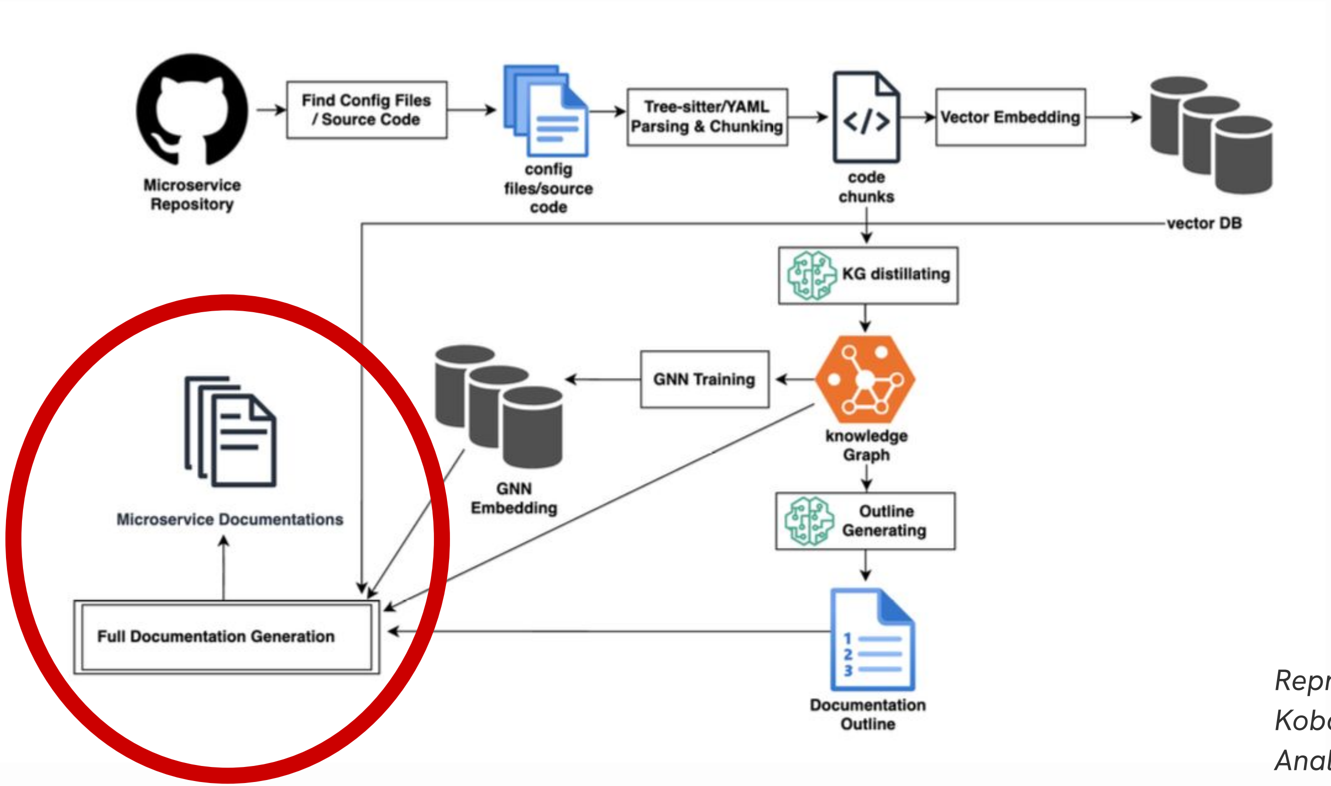
03

New data(?)

How do these issues persist in various microservices repositories?

Motivation

Current Framework



Reproduced from QIAO Lin, Kobayashi Software Analytics Group [5]

Issues (to fix in original code)

1. Introduction

- The README file describes a food delivery app built using a microservice architecture.
- The technologies listed include Spring Boot, Spring Cloud, NestJS, Laravel, MySQL, PostgreSQL, MongoDB, Kafka, Zipkin, and Docker.
- The README does not explicitly mention any services named `gateway`, `ms_order`, `ms_payments`, or `service_discovery`.
- The project is hosted on GitHub under the repository `Jonas56/food-delivery-app`.
- The project can be run using Docker, with instructions provided for using `docker-compose`.
- Contributions follow a "fork-and-pull" Git workflow.
- The project is distributed under the MIT License.

2. High-Level Architecture

No info found (Graph Disconnected).

3. Service Details

`gateway`

No info found (Graph Disconnected).

No info found (Graph Disconnected).

No info found (Graph Disconnected).

📊 EVALUATION SUMMARY TABLE

filename	overall_score
documentation_rag_k5_a0.5.md	2.4
documentation_gnn_k5_a0.5.md	2.2
documentation_kg_ablation_kg_first_h2-1_k5.md	2
doc_0_baseline.md	1.4

Experiment

Fix the code!

Run it and manually debug + use Cursor to fix and explain why issues occurred and to adjust the LLM-as-a-judge prompts for more critical judgment

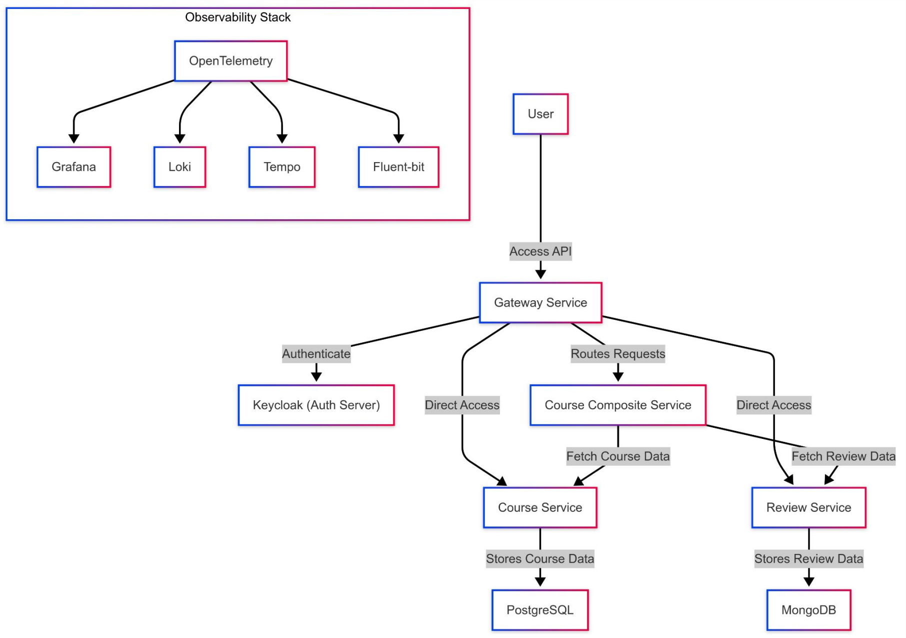
Repository analysis

Three repositories from original thesis experiment + four new ones to further diversity the new dataset

Document generation

Average scores of generated documentation was obtained by running the LLM prompt about 6 times; documentation of each attempt per repository was recorded on Google Sheets

Experiment



Architecture diagram of [Nasruddin/spring-boot-based-microservices]

Changes made:

"No info found (Graph Disconnected)" error messages in the KG-ablation document: this was fixed by forcing the framework to rely on semantic-based fallback and locate the graph instead of relying on an empty list.

Low scores (scores of less than 3) given to KG- and GNN-based LLM-generated documents: this was in part due to LLM hallucination of the presence/absence of services, which can result in documents saying a certain microservice was/wasn't present when the opposite was the case. This was fixed by tightening the “universal architect” prompt to penalize such contradictions, with a lighter penalty if the document in question was still useful. In addition, the “universal architect” prompts were edited to use a more strict, specific criteria, since we were looking for a certain ranking.

RAG-based LLM-generated documents scoring higher than KG- and GNN-based documents: the fixes in 3. were also applied here. No manual boosts were used to interfere with the data and neither the code generating these documents were altered; scoring only used the material of the documents provided and the repository they reported on.

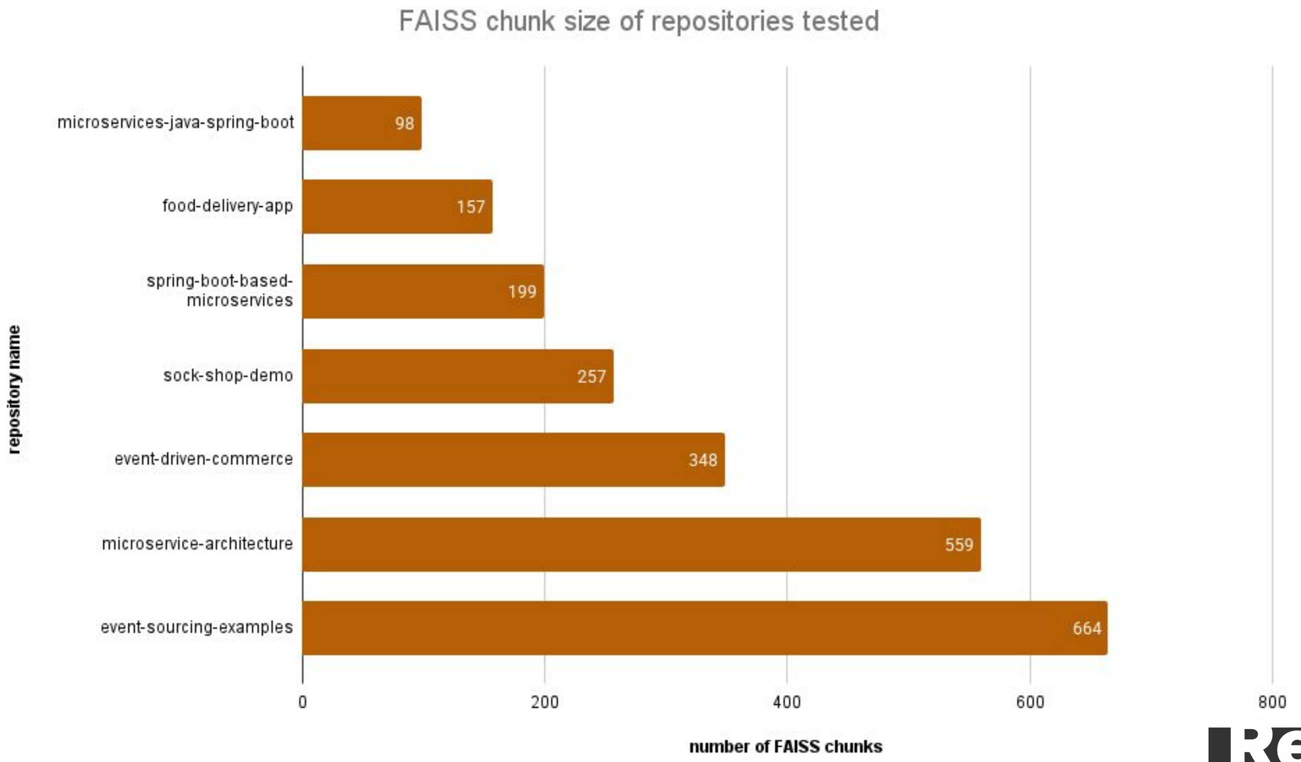
Improvement!

📊 EVALUATION SUMMARY TABLE

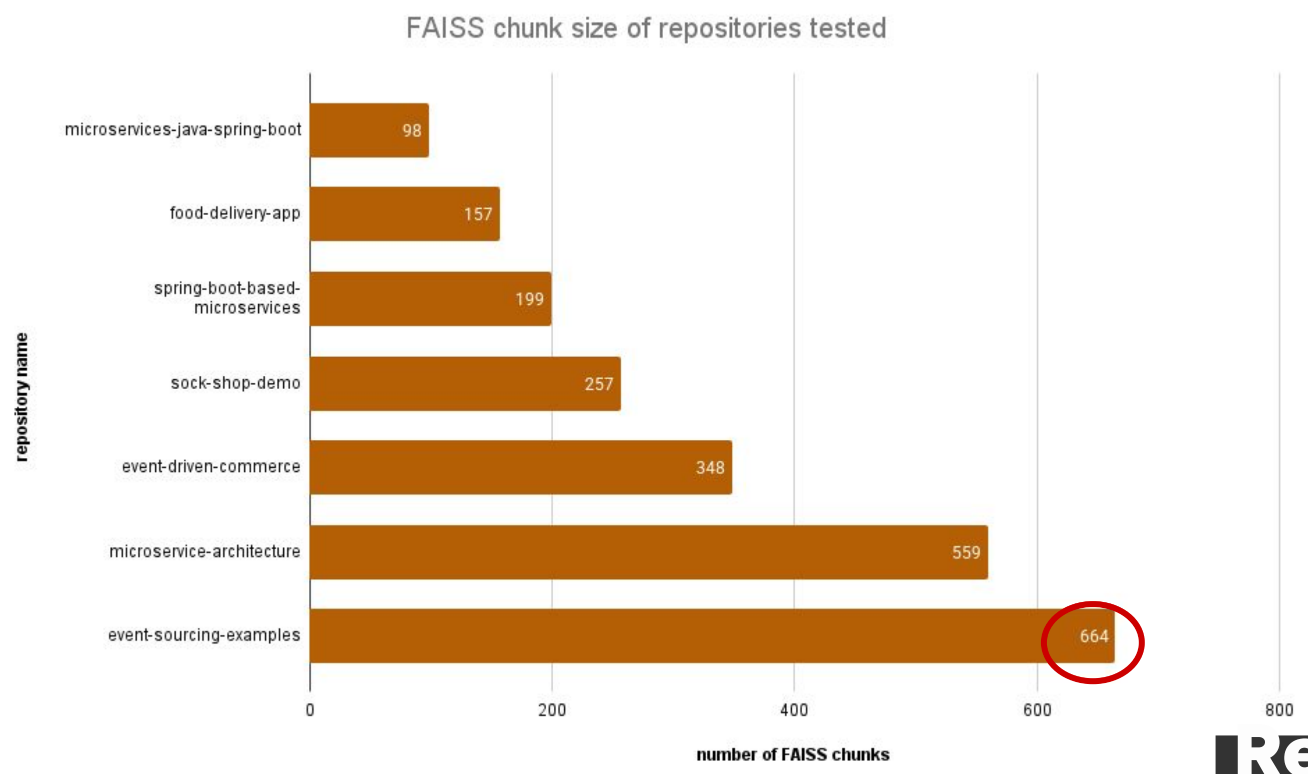
filename	overall_score
documentation_rag_k5_a0.5.md	2.4
documentation_gnn_k5_a0.5.md	2.2
documentation_kg_ablation_kg_first_h2-1_k5.md	2
doc_0_baseline.md	1.4

📊 EVALUATION SUMMARY TABLE

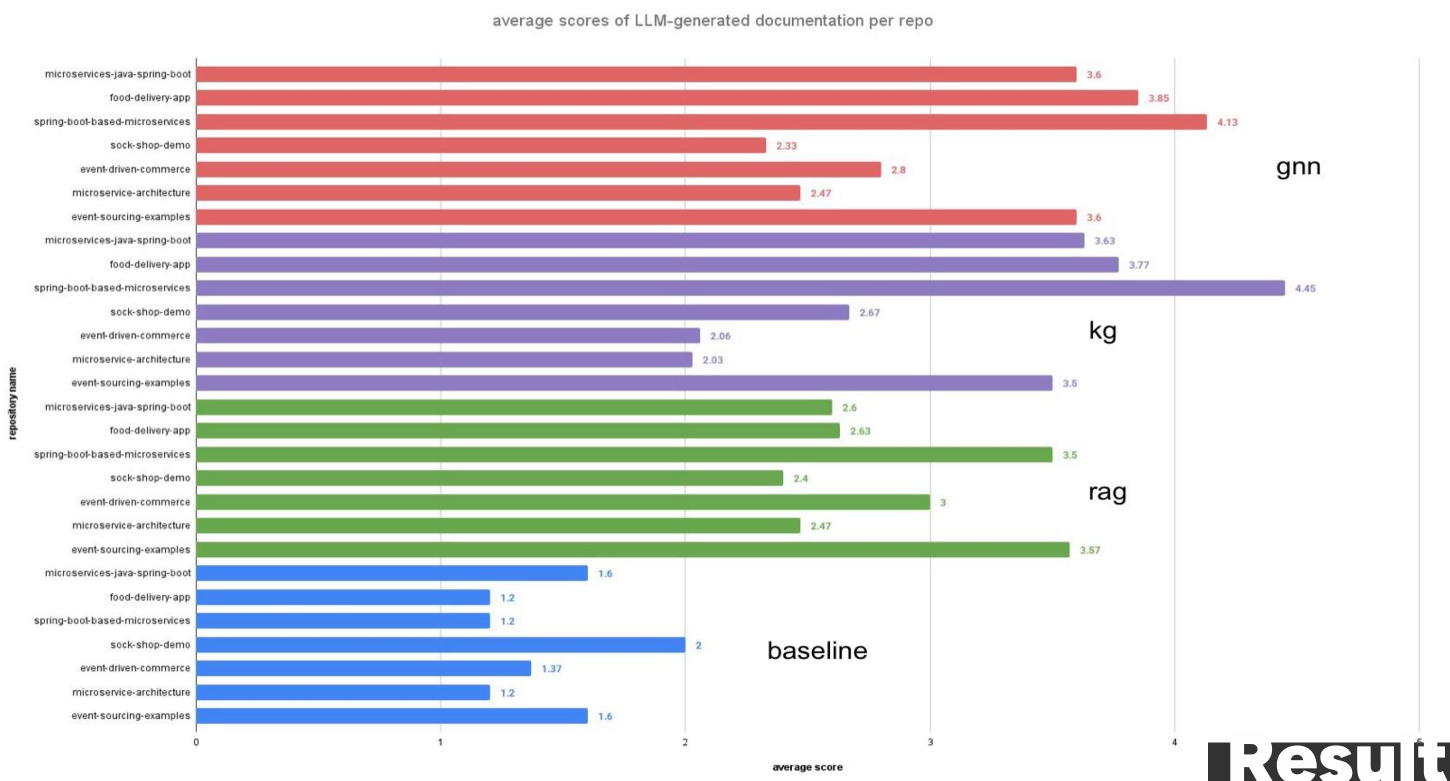
filename	overall_score
documentation_gnn_k5_a0.5.md	4.6
documentation_kg_ablation_kg_first_h2-1_k5.md	4.4
documentation_rag_k5_a0.5.md	4.2
doc_0_baseline.md	1.2


FAISS chunk size of repositories tested
repository name
microservices-java-spring-boot
food-delivery-app
spring-boot-based-microservices
sock-shop-demo
event-driven-commerce
microservice-architecture
event-sourcing-examples
98
157
199
257
348
559
664
0
200
400
600
800
number of FAISS chunks

Results

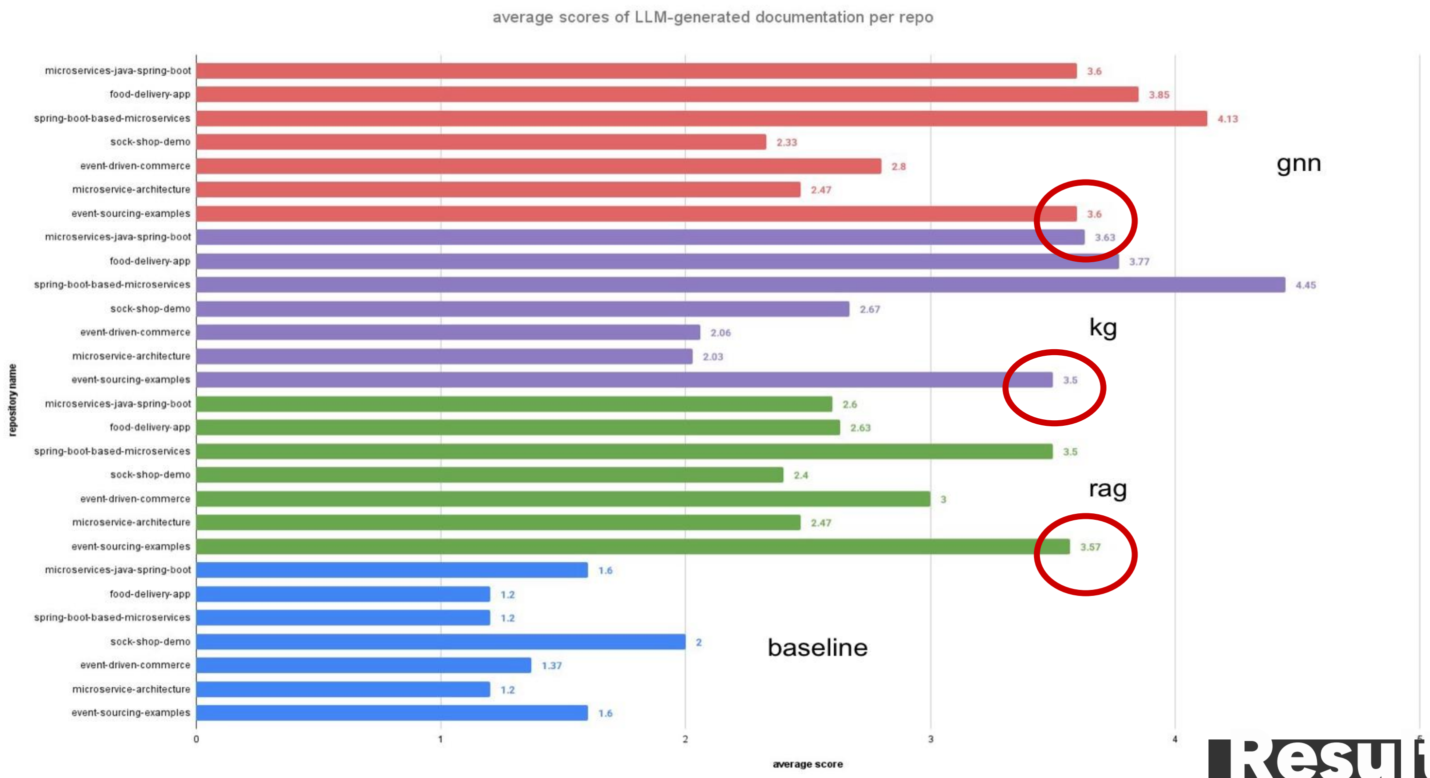

FAISS chunk size of repositories tested
repository name
microservices-java-spring-boot
98
food-delivery-app
157
spring-boot-based-microservices
199
sock-shop-demo
257
event-driven-commerce
348
microservice-architecture
559
event-sourcing-examples
664
0
200
400
600
800
number of FAISS chunks

Results



average scores of LLM-generated documentation per repo
microservices-java-spring-boot
food-delivery-app
spring-boot-based-microservices
sock-shop-demo
event-driven-commerce
microservice-architecture
event-sourcing-examples
3.6
3.85
4.13
2.33
2.8
2.47
3.6
gnn
3.63
3.77
4.45
2.67
2.06
2.03
3.5
kg
2.6
2.63
3.5
2.4
3
2.47
3.57
rag
1.6
1.2
1.2
2
1.37
1.2
1.6
baseline
repository name
average score
0
1
2
3
4

Results


average scores of LLM-generated documentation per repo
repository name
average score
microservices-java-spring-boot
food-delivery-app
spring-boot-based-microservices
sock-shop-demo
event-driven-commerce
microservice-architecture
event-sourcing-examples
3.6
3.85
4.13
2.33
2.8
2.47
3.6
gnn
3.63
3.77
4.45
2.67
2.06
2.03
3.5
kg
2.6
2.63
3.5
2.4
3
2.47
3.57
rag
1.6
1.2
1.2
2
1.37
1.2
1.6
baseline
0
1
2
3
4
5

Results

Conclusion

Observations

Did we improve the code? Yes!
But we found that it works better on smaller microservices repositories.

Results

More code chunks → lower, less accurate scores to documentation; high scores for RAG documentation

Limitations

Cost per OpenAI LLM call + one Neo4j instance = unable to test our framework on massive industrial scale repositories, LLM-as-a-judge bias

Future Work

Validate proposed method on more massive larger scale repositories + improve how ground-truth microservice architecture topology is extracted from a target repository [5].

Works Cited

1. N. Dragoni et al., “Microservices: yesterday, today, and tomorrow,” 2017.
2. J. Bogner, et al., “Assuring the evolvability of microservices: Insights into industry practices and challenges,” Sept. 2019.
3. J. Bogner et al., “Microservices in industry: Insights into technologies, characteristics, and software quality,” in 2019 IEEE International Conference on Software Architecture Companion (ICSA-C), pp. 187–195, 2019.
4. P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive nlp tasks,” 2021.
5. L. Qiao, “Structure-Aware Enhanced LLMs via Knowledge Graphs for Microservice Architecture Documentation”, 2026. Master’s Thesis, School of Computing, Institute of Science Tokyo.

Questions?

Thanks!

dawnmai@cs.washington.edu
https://github.com/dawnmaii

www.linkedin.com/in/dawnmai