Large Language Models for Legacy Software Systems
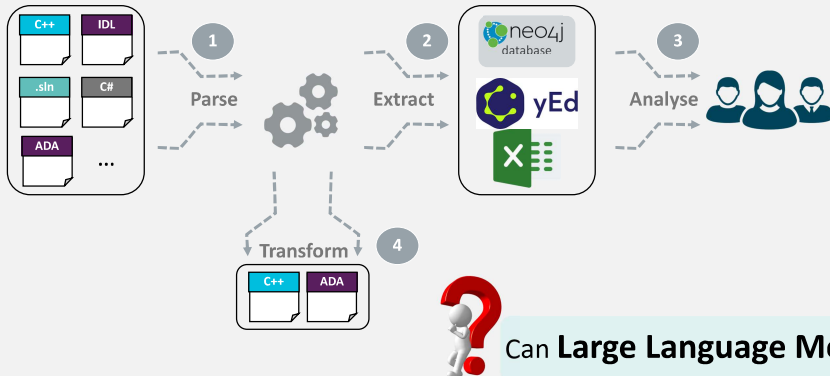
Joe Reynolds, Nan Yang, Rosilde Corvino, Anca Lichiardopol, Joost van Zwam

Software legacy is, at the same time, **an inevitable burden and a valuable heritage**.
Maintenance is estimated at
75% to 90% of software development life cycle cost (Galorath, Koskinen).

Renaissance: ESI methodology for legacy software

1 Parse **2** Extract **3** Analyse **4** Transform

These tools make possible rejuvenations that would otherwise be impossible

Reduction of the complexity in specifying the solution

The complexity of the problem at stake remains high

Can **Large Language Models (LLMs)** help with addressing the complexity?

Research Questions

Methodology
- Can LLMs be used as **an alternative to Renaissance** methodology?
- Can LLMs **boost Renaissance** methodology?

Industrial embedding
- What is the **best setup** to apply LLMs **in an industrial environment**?
- How to **make LLM based solutions adopted** by software engineers?

Literature

- LLMs have shown **initial competencies** in understanding code syntax, but still **struggle with code semantics** (Ma 2024).
- **No existing work** helps developers comprehend the **architecture of large-scale systems** (Zhang 2023).

Experiments

- LLMs ability to **extract architectural diagrams from C++ source code** was tested with a vector database.
- LLMs ability to **generate English explanations from C++ source code** was tested with a vector database.

Observations

- Evaluating generated architecture diagrams against Renaissance methodology determined that **Renaissance was superior to out-of-box LLMs** for providing **consistent and accurate** results.
- The Renaissance methodology lacks the **capabilities of generating English explanations,** and after evaluation against human benchmarks, **LLMs potential was clearly demonstrated**.
- Preliminary results show the **complementary nature of LLMs and Renaissance**.
- Our ongoing experiments show the benefit of **combining these two techniques, using vector and graph databases**.

Retrieval Augmented Generation (RAG): Promising LLM setup for industrial applications

Database setups

Domain knowledge is incorporated as a database.

Trustworthiness is increased by tracing answers back to the code.

C++ repo

Renaissance

Embeddings model

Completed experiment

Ongoing experiment

Graph

Vector

neo4j database

Security and privacy concerns are mitigated by using open-source LLMs deployed on trusted infrastructure.

Structured data improves the quality of results.

Cypher query | Matching subgraph | Semantic query | Matching vectors

Context

LLM

User query

Chatbot

Answer