Hệ thống tạo sinh và tóm tắt mã nguồn dựa trên truy xuất tăng cường – REDCODER

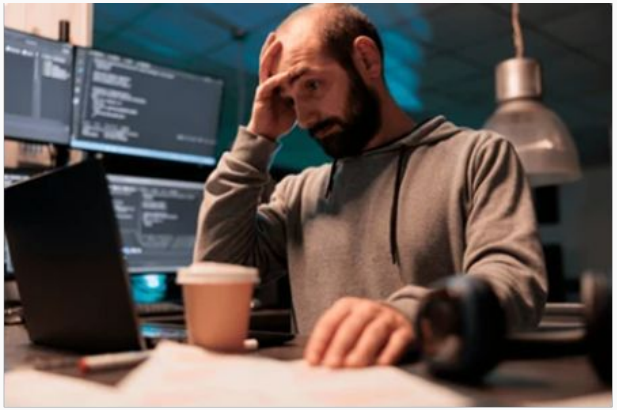
Lê Minh Thanh Tú - 230101032

Tóm tắt

- Lớp: CS2205.CH190
- Link Github của nhóm: tulmt-work/CS2205.CH190
- Link YouTube video: Phương pháp nghiên cứu khoa học - CS2205.CH190 - Lê Minh Thanh Tú

Lê Minh Thanh Tú

Giới thiệu



Viết code là một chuyện, hiểu lại đoạn code cũ còn đau đầu hơn.

- Giải pháp đề xuất: REDCODER (**R**etrieval augment**ED COD**e g**E**neration and summa**Ri**zation) – hệ thống hỗ trợ lập trình viên **tạo sinh** và **tóm tắt mã nguồn** một cách **chính xác** và **ngữ cảnh hóa**, nhờ vào kỹ thuật **truy xuất tăng cường (RAG)** kết hợp mô hình ngôn ngữ mạnh như **PLBART**.

Mục tiêu

- Xây dựng hệ thống hỗ trợ sinh mã từ mô tả và tóm tắt mã thành văn bản dễ hiểu.
- Kết hợp kỹ thuật truy xuất thông tin (**RAG**) và mô hình tạo sinh ngôn ngữ (**PLBART**) trong một framework thống nhất (**REDCODER**).
- Cung cấp một hệ thống hỗ trợ cho công việc của lập trình viên một cách nhanh chóng, chính xác.

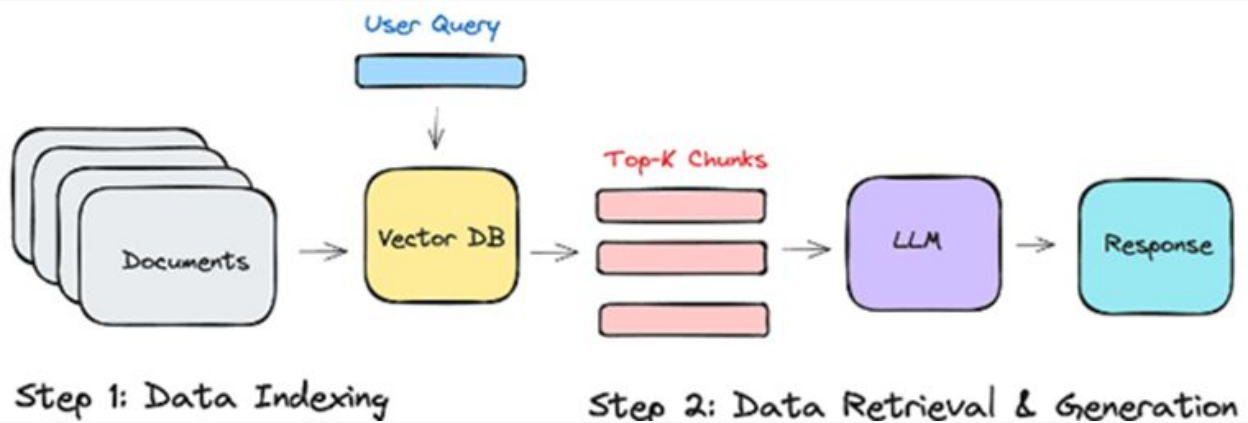
Nội dung và Phương pháp

- Xây dựng cơ sở dữ liệu: từ các bộ như CodeSearchNet, CodeXGLUE, Concode.
  - Bộ dữ liệu này được thu thập từ mã nguồn của các developer thật trên các repository mã nguồn mở trên GitHub.
  - Trải qua nhiều bước tiền xử lý như loại bỏ trùng lặp, trích xuất các đoạn tóm tắt có chất lượng cao. Kết quả thu được là bộ dữ liệu khoảng 1.1M đoạn tóm tắt với 20% trong số đó có thể ghép cặp với đoạn code tương ứng bằng ngôn ngữ Java và Python.

Nội dung và Phương pháp

RAG (Retrieval-Augmented Generation)

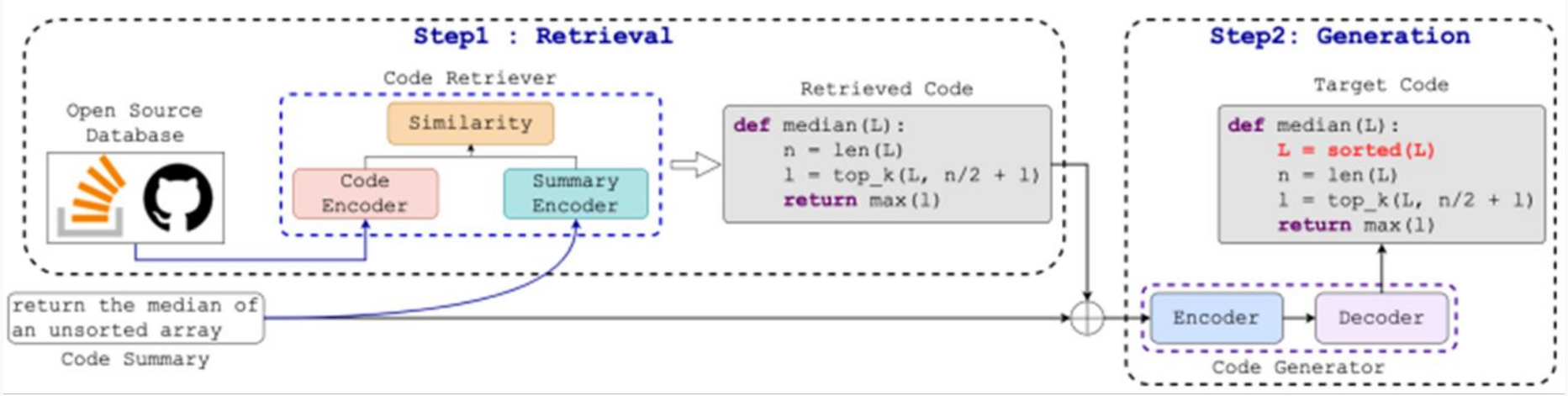
- Là kỹ thuật nâng cao độ chính xác cho câu trả lời của mô hình gen-AI với dữ liệu được truy xuất từ các nguồn dữ liệu bên ngoài.
- Giải quyết nhiều vấn đề mà các LLM thường hay gặp phải (ảo giác - hallucination; độ tin cậy; giảm chi phí, khối lượng công việc,..).



Nội dung và Phương pháp

- Mô hình truy xuất SCODE-R: dựa trên Dense Passage Retriever (DPR).
- Mô hình tạo sinh SCODE-G: sử dụng PLBART (mô hình seq2seq pre-trained).
- Kết hợp dữ liệu truy xuất và đầu vào để tạo kết quả tăng cường (RAG).

Nội dung và Phương pháp



Pipeline cơ bản của REDCODER với tác vụ sinh mã

Kết quả dự kiến

- Xây dựng dataset mã nguồn chất lượng cao có cả mã và mô tả.
- Triển khai mô hình REDCODER hoạt động hiệu quả trên hai tác vụ chính.
- Kết quả đầu ra sát với ngữ cảnh, có thể áp dụng cho IDE hoặc công cụ hỗ trợ lập trình.

Tài liệu tham khảo

[1]. Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 440–450, Vancouver, Canada. Association for Computational Linguistics.
[2]. Xiaodong Gu, Hongyu Zhang, Dongmei Zhang, and Sunghun Kim. 2016. Deep api learning. In Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, pages 631–642.
[3]. Wasi Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2020. A transformer-based approach for source code summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4998–5007, Online. Association for Computational Linguistics.
[4]. Bolin Wei, Ge Li, Xin Xia, Zhiyi Fu, and Zhi Jin. 2019. Code generation as a dual task of code summarization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Ad vances in Neural Information Processing Systems 32, pages 6563–6573. Curran Associates, Inc.