Применение нейросетевой повторной сортировки в системах дополненной поиском генерации

Романюк Полина Дмитриевна

Студент

Факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия

E-mail: s02240333@gse.cs.msu.ru

Научный руководитель *— Захарова Татьяна Валерьевна*

Решается задача автоматической классификации уязвимостей программного кода и генерации расширенных отчетов с рекомендациями по их устранению методом генерации с дополненной выборкой (Retrieval Augmented Generation, RAG). Источником данных об уязвимостях послужил локальный набор данных, собранный путем автоматизированной обработки актуального реестра Common Weakness Enumeration (CWE) [6] и обогащенный примерами реальных уязвимостей (CVE) с открытых репозиториев.

В качестве технологической основы применяется базовая архитектура большой языковой модели [7], механика работы которой описана в [1]. Данная система осуществляет векторизацию образцов уязвимого кода и использует ансамблевый (гибридный) поиск, где для хранения и индексации векторов применяется база данных FAISS (Facebook AI Similarity Search) [2].

Одной из ключевых проблем применения RAG для генерации отчетов является сложность оценки качества работы модели. Система создает значительный объем текста, что затрудняет как ручную проверку, так и использование стандартных метрик, которые плохо отражают фактологическую точность.

Для решения проблемы оценки качества предлагается использовать раздел «Рекомендации» из создаваемого отчета. Выбор данного раздела обусловлен тем, что рекомендации представляют собой короткие текстовые сообщения, несущие конкретную смысловую нагрузку, что упрощает их формализованное сравнение с эталоном.

Способом оценки выбрано создание метрики семантического подобия. Для этого применяется векторизация текстовых сообщений с помощью модели Sentence-BERT (S-BERT) [3], которая преобразует предложения в векторы фиксированной размерности.

Сравнение созданных рекомендаций с эталонными производится с помощью расчета косинусного сходства между векторами. Тогда сходство эталонной ($\vec{u}$) и созданной ($\vec{v}$) рекомендации вычисляется

по формуле:

$$Q = \cos(\theta) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\|\|\vec{v}\|} \quad (1)$$

В работе [7] авторы предложили внедрить в процесс ранжирования этап нейросетевой повторной сортировки (re-ranking). Цель данного этапа — заново оценить релевантность документов, найденных базовым поиском FAISS, и решить проблему потери контекста. Предложены легковесная модель ms-marco-MiniLM-L-6-v2 [4] и более сложная модель BAAI/bge-reranker-base [5].

В настоящем докладе приведены результаты сравнения эффективности использования указанных нейросетевых моделей повторной сортировки в применении к задаче классификации кода и создания рекомендаций по устранению уязвимостей в небезопасном коде.

Литература

1. Архитектура LLM для финансового сектора и ИБ. `https://habr.com/ru/articles/963482`
2. Johnson J., Douze M., Jégou H. Billion-scale similarity search with GPUs // IEEE Transactions on Big Data. – 2019. – Vol. 7. – №. 3. – P. 535-547. `https://arxiv.org/abs/1702.08734`
3. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. – 2019. – P. 3982-3992. `https://arxiv.org/abs/1908.10084`
4. Hugging Face: ms-marco-MiniLM-L-6-v2. `https://huggingface.co/cross-encoder/ms-marco-MiniLM-L6-v2`
5. Hugging Face: BAAI/bge-reranker-base. `https://huggingface.co/BAAI/bge-reranker-base`
6. MITRE Common Weakness Enumeration (CWE) Downloads. `https://cwe.mitre.org/data/downloads.html`
7. Репозиторий проекта AI for Finance Hack 2025. `https://github.com/Runoi/ai-for-finance-hack-2025`