УДК 004.89

РАЗРАБОТКА СИСТЕМЫ СЕМАНТИЧЕСКОГО КЭШИРОВАНИЯ ОТВЕТОВ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ НА ОСНОВЕ ГРАФОВОГО ПРЕДСТАВЛЕНИЯ ДИАЛОГОВ

Литвинов И.А.
Университет ИТМО, Санкт-Петербург
Научный руководитель — Курилов Фёдор Константинович, преподаватель практики, ФИТиП

Введение. Широкое внедрение больших языковых моделей (LLM) в B2B-сервисы порождает проблему высоких операционных затрат на инференс и значительной латентности ответов. На практике существенная доля пользовательских запросов семантически повторяется, однако классические методы кэширования, основанные на точном совпадении строк, не способны выявлять смысловую близость формулировок. Существующие подходы к семантическому кэшированию, такие как GPTCache, используют flat-хранилища эмбеддингов без учёта контекста диалога, что приводит к ошибочным cache hit для одинаковых запросов, требующих различных ответов в зависимости от предшествующей беседы. Таким образом, актуальной является задача разработки системы кэширования, учитывающей траекторию диалога для корректного определения семантической эквивалентности запросов.

Основная часть. В работе предложена архитектура middleware-сервиса семантического кэширования, встраиваемого между RAG-системой и внешним LLM API. Ключевой идеей является представление накопленных диалогов в виде направленного графа, где каждый узел содержит текст запроса, набор вариантов ответов и предвычисленные эмбеддинги траектории от корня до данного узла. Рёбра графа отражают последовательность реплик в рамках сессии.

При поступлении нового запроса система выполняет поиск по графу, начиная с корневых узлов и проходя по траектории текущей сессии. На каждом уровне сравниваются эмбеддинги траекторий с использованием моделей sentence-transformers. Вычисляется показатель уверенности: при достижении настраиваемого порога возвращается один из кэшированных вариантов ответа без обращения к LLM. При резкой смене темы в середине диалога система направляет запрос напрямую к языковой модели, минимизируя риск некорректного ответа.

Система реализует мультитенантность: каждый клиент платформы работает с изолированным деревом диалогов, идентифицируемым по паре «версия системного промпта — версия базы знаний». При обновлении промпта или базы знаний старый граф инвалидируется и начинается накопление нового. Компонент предоставляет REST API, совместимый с существующим LLM Gateway, что обеспечивает прозрачную интеграцию в инфраструктуру платформы.

Стек технологий включает Python 3.11, FastAPI, PostgreSQL для хранения структуры графа, Qdrant для векторного поиска и Redis для кэширования активных сессий. Экспериментальная оценка выполнена на анонимизированном датасете диалогов платформы ООО «Нейросеть» и включает измерение hit rate, BERTScore F1 между кэшированными и эталонными ответами, а также нагрузочное тестирование при 100–1000 запросах в секунду.

Выводы. Предложенный подход на основе графового представления диалогов позволяет корректно различать семантически идентичные по формулировке запросы, требующие разных ответов в зависимости от контекста беседы. Разработанный middleware-сервис обеспечивает снижение операционных затрат на инференс LLM и уменьшение латентности ответов за счёт повторного использования ранее сгенерированных ответов. Практическая значимость работы подтверждается

интеграцией компонента в staging-окружение платформы ООО «Нейросеть», обслуживающей B2B-клиентов.

Список использованных источников

1. Bang, F. GPTCache: An Open-Source Semantic Cache for LLM Applications Enabling Faster Answers and Cost Savings / F. Bang // Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI. — 2023. — P. 212–218.
2. Reimers, N. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks / N. Reimers, I. Gurevych // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. — 2019. — P. 3982–3992.
3. Lewis, P. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks / P. Lewis, E. Perez, A. Piktus [et al.] // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 9459–9474.