

РАЗРАБОТКА МУЛЬТИАГЕНТНОЙ СИСТЕМЫ ГЕНЕРАЦИИ КОНТЕНТА КАК ПРОГРАММНОГО КОМПЛЕКСА ИНТЕЛЛЕКТУАЛЬНОЙ ОБРАБОТКИ ДАННЫХ

Вайтович Д. В.¹

Научный руководитель – старший преподаватель Жаранова А. О.¹

¹СПбГУТ им. проф. М. А. Бонч-Бруевича

vaitovitchd@yandex.ru

Введение

Современные интеллектуальные приложения, использующие большие языковые модели и генеративные алгоритмы, предъявляют повышенные требования к архитектуре программных систем. Основной задачей разработки становится обеспечение масштабируемости, независимости вычислительных компонентов и устойчивости к изменениям внешних сервисов искусственного интеллекта.

Основной проблемой является высокая связанность программных решений с конкретными языковыми моделями и API, что затрудняет сопровождение и развитие программного продукта. В связи с этим актуальным направлением является применение модульных и мультиагентных архитектур, позволяющих разделить вычислительные процессы на независимые компоненты с управляемым взаимодействием.

В работе представляется разработка программного комплекса интеллектуальной обработки данных, основанного на принципах модульности, оркестрация агентов и независимости используемых LLM-моделей. Архитектура системы ориентирована на динамическую маршрутизацию задач между агентами, поддержку локальных и облачных моделей, а также использование Retrieval-Augmented Generation (RAG) как отдельного слоя обработки знаний [1].

Предложенный подход демонстрирует возможность применения прикладных методов для построения расширяемых интеллектуальных систем, адаптируемых к изменениям информационной среды и требованиям обработки больших объемов данных.

Основная часть

Разработка системы выполнялась с уклоном на архитектурное проектирование программного комплекса, где основным объектом исследования и разработки являются способы организации взаимодействия вычислительных компонентов.

Архитектура системы построена на принципах модульности и слабой связанности. Каждый функциональный элемент реализован в виде независимого программного модуля, взаимодействующего через унифицированные интерфейсы и типы. Такой подход позволяет гибко изменять компоненты без влияния на остальные части системы и обеспечивает возможность масштабирования.

Основным архитектурным решением является применение мультиагентной модели обработки данных. Агенты являются не просто модулями, а являются автономными объектами, взаимодействующие через управляющий слой. В отличие от традиционной микросервисной архитектуры, агенты обладают логикой выполнения и могут динамически переключать вычислительные процессы в зависимости от состояния каждого агента.

Координация работы комплекса реализована через механизм оркестрации, выполняющий маршрутизацию задач между агентами. Управляющий модуль анализирует параметры запроса, параметры ответа, доступность моделей и текущую

нагрузку, после чего формирует последовательность обработки данных. Такой подход позволяет реализовать адаптивный поток выполнения.

Особое внимание уделено независимости системы от конкретных языковых моделей. Для этого введен слой с облачной LLM. Адаптеры нормализуют входные и выходные данные моделей, что позволяет использовать облачные сервисы и локально развернутые нейросети в рамках единого интерфейса. В результате достигается модельная архитектура, где адаптеры объединяют логику работы с LLM, а инфраструктурный модуль RAG дополняет их. Выделение RAG в самостоятельный компонент позволяет повторно использовать контекстные данные, уменьшить нагрузку на генеративные модели и повысить результаты обработки [2].

Для повышения устойчивости системы применена каскадная модель выполнения задач. Каждый агент поддерживает использование основной и резервной модели, а механизм автоматического переключения обеспечивает продолжение обработки при возникновении ошибок или превышении времени ответа. Каскадная обработка снижает зависимость от внешних сервисов и повышает надёжность системы.

Прикладная реализация системы включает веб-клиентский интерфейс, серверный уровень и слой интеграцию нейросетей. Использование современных веб-технологий и API-ориентированного взаимодействия обеспечивает расширяемость и возможность интеграции новых агентов без изменения архитектурной логики.

Выводы

В результате работы был разработан программный комплекс интеллектуальной обработки данных, архитектура основана на принципах модульности, мультиагентного взаимодействия и оркестрации процессов.

Ключевым фактором устойчивости интеллектуальных приложений является независимость от конкретных моделей искусственного интеллекта и перенос управления логики на уровень архитектуры системы. Использование слоя адаптеров, RAG-модуля и каскадной обработки позволило обеспечить отказоустойчивость, масштабируемость и гибкость программного решения.

Описанный подход показывает, как на практике можно применять современные методы разработки для создания интеллектуальных систем, которые работают с большими объемами данных. А главное, эти наработки легко использовать при проектировании веб-сервисов, которые будет просто расширять и сопровождать.

Литература

1. Тарасов А. И. Искусственный интеллект и автоматизация создания медиаконтента // Информационные технологии и системы. 2023. № 4. С. 51–57.
2. Руденко Н. В. Методы интеллектуального анализа новостных потоков. – СПб: Санкт-Петербургский государственный университет телекоммуникаций. 2022. 164 с.