УДК 004.42
DOI 10.18822/byusu20250340-47

МУЛЬТИМОДАЛЬНАЯ НЕЙРОСЕТЕВАЯ ОБРАБОТКА ВИДЕОЛЕКЦИИ ПОСРЕДСТВОМ МУЛЬТИАГЕНТНЫХ СИСТЕМ

Исмагулов Милан Ерикович
аспирант 3 года обучения направления
«Системный анализ, управление и обработка
информации, статистика»
Инженерной школы цифровых технологий,
Югорский государственный университет,
Ханты-Мансийск, Россия
E-mail: m_ismagulov@ugrasu.ru

Предмет исследования: мультимодальная обработка видеолекций с использованием мультиагентных систем. Статья фокусируется на промежуточных результатах исследования, включая обзор понятий мультимодальности, мультиагентности и многомодельных систем, а также на разработке подходов к обработке видеоданных из лекций.

Цель исследования: преобразование всей релевантной информации из видеолекции в текстовый документ для формирования сопровождающего конспекта лекции. Цель – разработать эффективный цикл обработки данных, учитывая различия в форматах видеолекций.

Методы исследования: выбор паттерна «Оркестратор-исполнитель» (Orchestrator-Worker Pattern) с большой языковой моделью (LLM) в роли оркестратора. Обзор альтернативных подходов, а именно одноранговый децентрализованный паттерн и гибридный паттерн, с обоснованием выбора оркестраторного подхода для обеспечения последовательной обработки и отказоустойчивости. Интеграция конвейерной обработки видеопотока в мультиагентную систему (гибридный подход).

Объекты исследования в данной статье представляют собой видеолекции трех основных типов, служащие источниками мультимодальных данных для анализа и обработки. Первый тип – «Лектор и презентация» – включает видеозаписи, где лектор располагается слева или справа от сопровождающей презентации, с акцентом на визуальное сочетание человеческой фигуры и слайдов. Второй тип – «Презентация и закадровый голос» – фокусируется на теоретическом материале, представленном на слайдах презентации, с объяснением за кадром через аудиодорожку. Третий тип – «Лектор и доска» – охватывает записи, где лектор пишет материал на классической меловой или маркерной доске, подчеркивая рукописный ввод информации.

Основные результаты исследования: разработана и обоснована архитектура мультиагентной системы на основе паттерна «Оркестратор-исполнитель» с гибридным подходом, интегрирующим конвейерную обработку видео в мультиагентную среду для эффективного распределения задач и управления нагрузкой. Выбраны и описаны модели и инструменты, а именно оркестраторы, модели аудиообработки, OCR, с учетом типов лекций для адаптивных конвейеров. Описано функционирование агентов, инициализация, взаимодействие с оркестратором, параллельная обработка аудио/видео, агрегация результатов в текстовый документ с возможностью скачивания/печати.

Ключевые слова: мультимодальность, мультиагентность, паттерн «Оркестратор-исполнитель», большая языковая модель, взаимодействие агентов, конвейер обработки данных, многомодельные подходы, машинное обучение, одноранговая децентрализованная архитектура, обработка видеолекций.

MULTIMODAL NEURAL NETWORK PROCESSING OF VIDEO LECTURES USING MULTI-AGENT SYSTEMS

Milan E. Ismagulov
Postgraduate student,
Engineering School of Digital Technologies,
Yugra State University,
Khanty-Mansiysk, Russia
E-mail: m_ismagulov@ugrasu.ru

Subject of research: multimodal processing of video lectures using multi-agent systems. The article focuses on intermediate results of the research, including an overview of the concepts of multimodality, multi-agent systems, and multi-model systems, as well as the development of approaches to processing video data from lectures.

Purpose of research: transformation of all relevant information from a video lecture into a text document to form an accompanying lecture summary. The goal is to develop an effective data processing cycle, taking into account differences in video lecture formats.

Research methods: selection of the «Orchestrator-Performer» pattern (Orchestrator-Worker Pattern) with a large language model (LLM) in the role of the orchestrator. Overview of alternative approaches, namely the peer-to-peer decentralized pattern and the hybrid pattern, with justification for choosing the orchestrator approach to ensure consistent processing and fault tolerance. Integration of pipeline video stream processing into a multi-agent system (hybrid approach).

The objects of research in this article are video lectures of three main types, serving as sources of multimodal data for analysis and processing. The first type – «Lecturer and Presentation» – includes video recordings where the lecturer is positioned to the left or right of the accompanying presentation, with an emphasis on the visual combination of the human figure and slides. The second type – «Presentation and Voiceover» – focuses on theoretical material presented on the presentation slides, with explanation off-screen through the audio track. The third type – «Lecturer and Blackboard» – covers recordings where the lecturer writes material on a classic chalk or marker board, emphasizing handwritten input of information.

Research findings: An architecture for a multi-agent system has been developed and justified based on the «Orchestrator-Performer» pattern with a hybrid approach, integrating pipeline video processing into a multi-agent environment for effective task distribution and load management. Models and tools have been selected and described, namely orchestrators, audio processing models, OCR, taking into account lecture types for adaptive pipelines. The functioning of agents is described, including initialization, interaction with the orchestrator, parallel audio/video processing, and aggregation of results into a text document with the possibility of downloading/printing.

Keywords: multimodality, multi-agent systems, Orchestrator-Performer pattern, large language model, agent interaction, data processing pipeline, multi-model approaches, machine learning, peer-to-peer decentralized architecture, video lecture processing.

ВВЕДЕНИЕ

Мультимодальность в информационных науках и машинном обучении – это концепция, связанная с обработкой данных различных форм из разнородных источников, что особенно актуально для обработки такого типа данных, как видеолекция, так как при декомпозиции на первом этапе получается 4 источника данных, а именно видеоряд, аудиодорожка, возможно наличие субтитров и метаданных, а на втором этапе возможно разложение видеоряда на последовательность кадров. На данном этапе развития методов искусственного интеллекта и машинного обучения для обработки видеоряда активно применяются трансформерные нейросетевые модели, например дообученные модели ViT (Visual Transformers), OpenAI CLIP или LLaMa Vision [1; 2]. Для обработки аудио применяются модели speech-to-text, OpenAI Whisper, Alphacep-vosk и т. д. [3]. Модели на основе трансформеров способны эффективно извлекать контекст и обрабатывать данные, однако требуют хорошо размеченных мультимодальных данных, больших вычислительных ресурсов для дообучения и инференса [4].

Также активно начинают разрабатываться многомодельные методы обработки данных. Многомодельность – это подход, при котором данные обрабатываются последовательно или параллельно несколькими моделями, делается это для преобразования данных, для улучшения их обработки или для обеих этих целей [5]. Например, аудиодорожка преобразуется моделью в текст и подается на вход большой языковой модели для исправления синтаксических ошибок [6].

Мультиагентность – это подход к решению сложных задач, при котором задача декомпозируется на более простые подзадачи. Эти подзадачи распределяются между автономными агентами, обладающими своей компетенцией (возможно, реализованной разными моделями или алгоритмами) и действующими на основе собственных целей. Агенты функционируют децентрализованно, или взаимодействуют друг с другом, или через центральную модель-оркестратор в процессе решения (включая обмен данными, координацию или переговоры). Конечный результат формируется путем агрегации выходов агентов или является следствием совместной обработки задачи [7].

Постановка задачи. Основной задачей в исследовании является преобразование всей релевантной информации из видеолекции в текстовый документ посредством мультимодальной обработки. Это необходимо для формирования сопровождающего конспекта лекции для видео. В исследовании рассматривается возможность преобразования видеолекций 3 видов:

1. Лекция «Лектор и презентация» представляет собой видеозапись, в которой лектор находится слева или справа от сопровождающей его презентации, подробнее на рисунке 1.

2. Лекция «Презентация и закадровый голос» представляет собой видеозапись, в которой основной упор делается на теоретический материал, представленный на презентации, а закадровый голос объясняет этот материал.

3. Лекция «Лектор и доска» представляет собой видеозапись, в которой лектор пишет материал на классической меловой или маркерной доске.

Поскольку видеоформат во всех лекциях разный, можно сделать вывод, что и алгоритмы обработки этих видов будут различаться, так как обработка и абстрагирование данных будут разными [8].

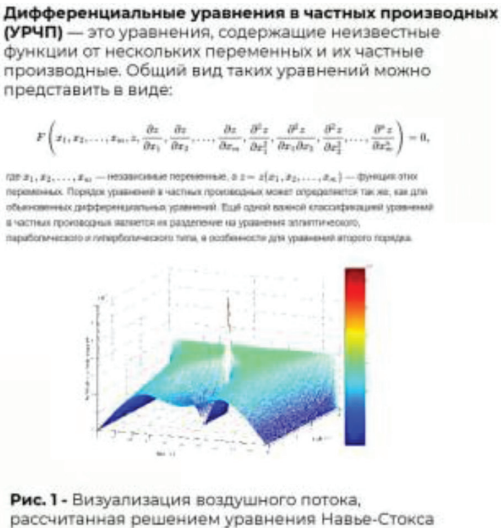

Рис. 1 - Визуализация воздушного потока, рассчитанная решением уравнения Навье-Стокса

Рисунок 1. Иллюстрация видеолекции «Лектор и презентация»

Выбранный метод решения поставленной задачи. Из определения мультиагентности, представленного в данной статье, известно, что подобные методы хорошо справляются с задачами, где требуется обработка сложных данных. Поскольку видеолекция достаточно просто декомпозируется на составляющие (видеоряд в последовательность кадров, аудиодорожку, метаданные), возможно построить мультиагентную систему, способную извлекать релевантные данные и преобразовывать их в текстовый документ.

Существует 3 базовых метода построения мультиагентных систем:

1. Мультиагентная система с одноранговыми децентрализованными агентами, в которой агенты находятся на одной иерархии и обмениваются сообщениями на равных; могут быть полносвязанными и неполносвязанными [9; 10; 11]. Схема данного паттерна изображена на рисунке 2.

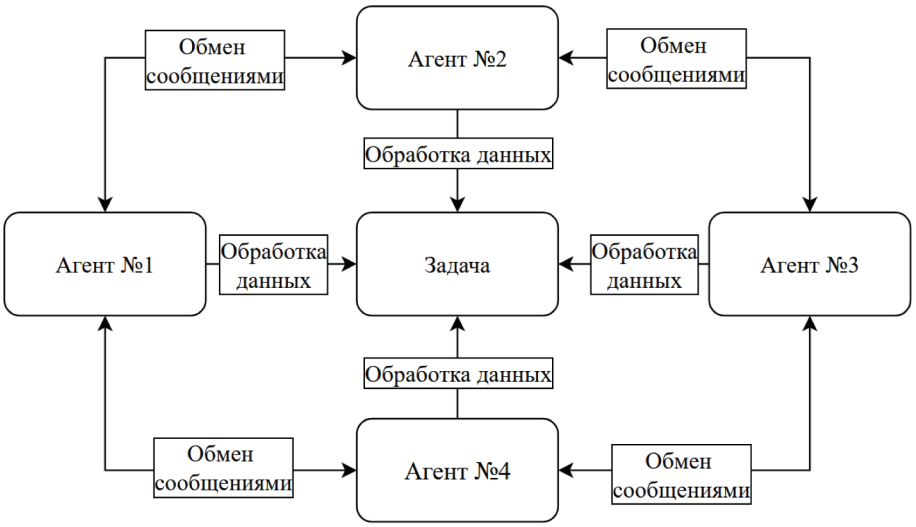

Рисунок 2. Схема мультиагентного паттерна с одноранговыми децентрализованными агентами с неполной связью

2. Мультиагентная система, построенная по принципу «Оркестратор-исполнитель» (Orchestrator-Worker Pattern). В разной литературе оркестратор может называться дирижером, менеджером задач или агентом-контроллером, а исполнитель – агентом-рабочим. Построена по принципу иерархии, где оркестратор раздает задачи агентам и принимает от них результат выполнения задания. Агенты не имеют горизонтальных связей [12]. Схема данного паттерна изображена на рисунке 3.

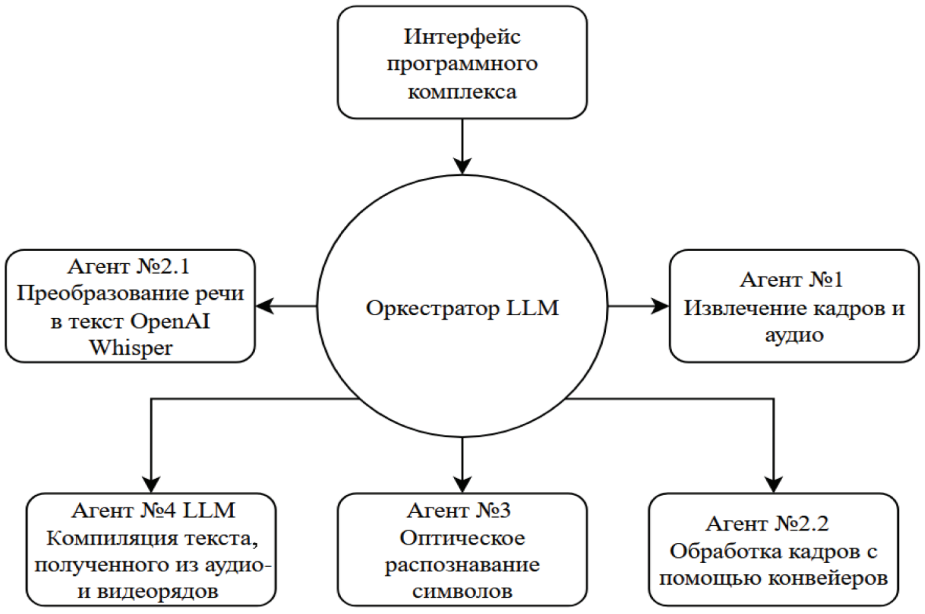

Рисунок 3. Схема мультиагентного паттерна «Orchestrator-Worker Pattern»

3. Гибридный метод сочетает в себе первые 2 метода, например через реализацию сложных агентов-рабочих, которые могут обмениваться сообщениями друг с другом, или нескольких оркестраторов, которые соединены как одноранговые агенты, а также нескольких оркестраторов, которые иерархически подчиняются другому оркестратору [13]. Схема данного паттерна изображена на рисунке 4.

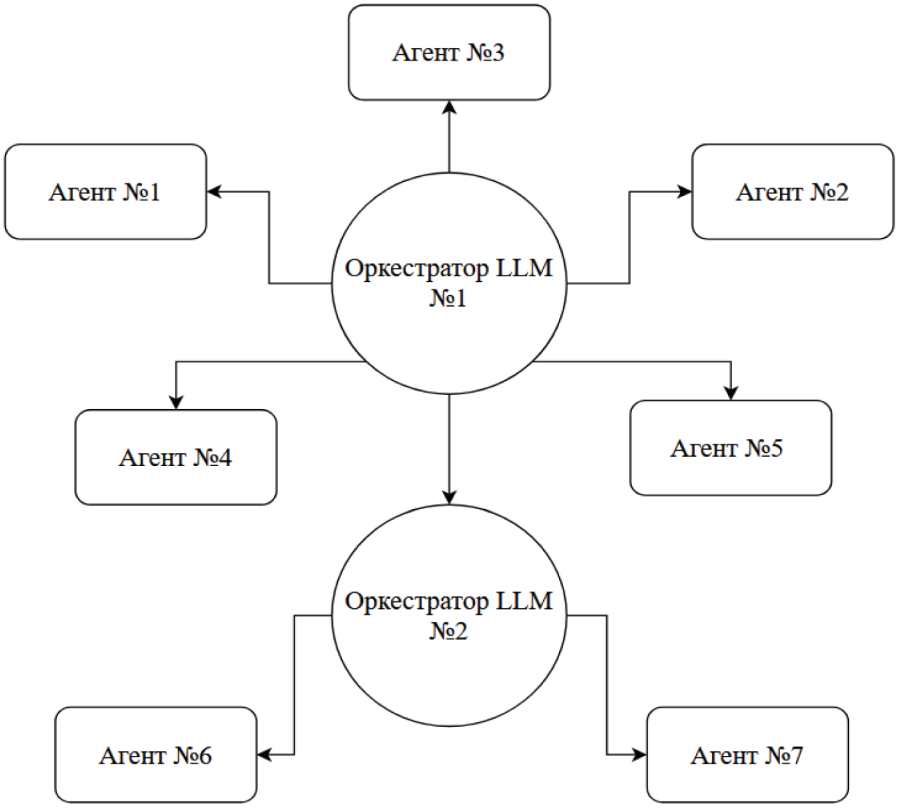

Рисунок 4. Схема мультиагентного паттерна с гибридным подходом

Как видно из схемы, присутствует 4 одноранговых агента, работающих над одной задачей (P2P-сеть), и в процессе работы агенты обмениваются сообщениями о своих состояниях, для обработки видео подобный паттерн не подходит по ряду причин. Первое – видеолекции характеризуются высоким объемом данных (HD/4K, длительная продолжительность). В P2P-сетях каждый узел должен ретранслировать данные другим участникам, что создает экспоненциальный рост сетевой нагрузки. При числе узлов N количество соединений достигает $O(N^2)$, приводя к перегрузке каналов даже в средних группах. Например, протокол передачи данных RTMP, оптимизированный для низкой задержки, эффективен только в модели «один-ко-многим», но не «многие-ко-многим».

Обработка видео требует конвейерных операций: декодирование, анализ кадров, распознавание текста/объектов.

В децентрализованной P2P-среде:

- невозможно гарантировать порядок выполнения этапов из-за равной иерархии агентов и отсутствия явного планировщика работы;
- зависимые задачи (например, распознавание речи) требуют сложных механизмов синхронизации;
- динамическая балансировка нагрузки затруднена из-за отсутствия глобального планировщика.

Современные исследования (например, на примере FANET для дронов) подтверждают, что сильная децентрализация оправдана только для задач с низким объемом данных и высокой динамичностью узлов.

РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

В качестве архитектуры для решения поставленной задачи был выбран паттерн независимых агентов-рабочих с оркестратором «Orchestrator-Worker Pattern», рисунок 3. Видеолекции требуют сложной многоэтапной обработки (декодирование, анализ кадров, распознавание речи, генерация субтитров). В паттерне Orchestrator-Worker центральный координатор (Orchestrator) динамически разбивает задачу на параллельно выполняемые подзадачи, которые распределяются между

специализированными Worker-агентами. Это исключает дублирование функций и оптимизирует загрузку вычислительных ресурсов (например, GPU для нейросетевых задач). Эксперименты показывают до 40 % сокращение времени обработки по сравнению с одноранговыми моделями [14].

При отказе Worker-агента Orchestrator автоматически перераспределяет его подзадачу другому агенту, сохраняя прогресс выполнения. В event-driven-реализациях (например, с использованием Kafka) это обеспечивается механизмами репликации и повторной обработки событий [15]. Для видеолекций длительностью 60+ минут такая отказоустойчивость критична, тогда как в P2P восстановление после сбоев требует ручной координации.

Выбранные модели, алгоритмы и инструменты для реализации предложенного метода. В первую очередь пользователь через специальный элемент должен указать путь до файла видеолекции и выбрать тип лекции (рисунок 5) из представленных в пункте «Постановка задачи». В качестве оркестратора рассматриваются следующие модели – Qwen 72b-Instruct и Mistral Large-2 Instruct. Приписка Instruct в названии модели говорит о том, что модель лучше подходит для задач, связанных с выполнением четких инструкций.

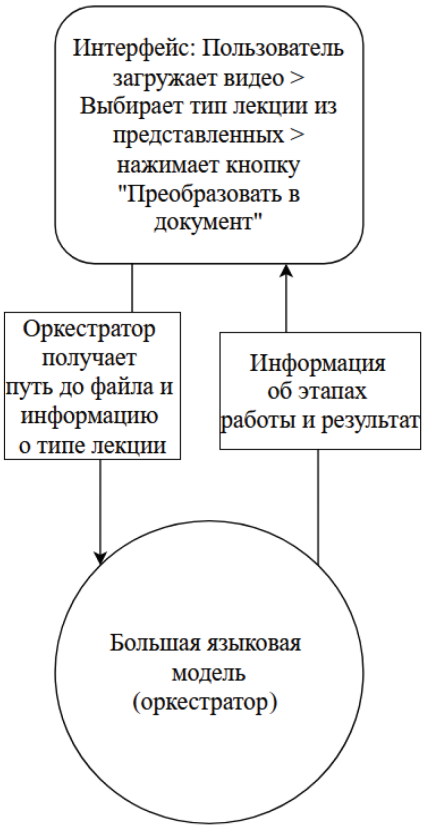

Рисунок 5. Схема взаимодействия пользователя и оркестратора

От оркестратора пользователь получает коллбек-информацию о ходе выполнения обработки видеолекции. Следующим этапом предполагается использование инициализации агентов-исполнителей, оркестратор опрашивает агентов о готовности к работе, проверяет доступ API-ключей. Если агенты не отвечают на запрос, происходит информирование пользователя и переключение на другого агента со схожей функциональностью. Для оркестрации возможно использование следующих библиотек: LangChain, LangGraph, CrewAI. В данных библиотеках существуют следующие типы агентов:

1. Интеллектуальный автономный (может быть ИИ-агентом) – чаще всего это агент с функциями обучения, дообучения или адаптации.

2. Оркестратор – в других источниках также может обозначаться как дирижер, мастер (часто встречается в англоязычной литературе), менеджер. Особый вид агента, который распределяет задачи между агентами, и координирует их действия, и, возможно, агрегирует финальный результат.

3. Инструменты – по своей сути это обычные алгоритмы, утилиты и функции, написанные на каком-либо языке программирования;

необходимы для обработки, проверки и иных действий с данными.

Первым агентом-обработчиком является агент, извлекающий кадры видео и аудиодорожку, подробнее на рисунке 6. В качестве ответа этот агент отправляет оркестратору сообщение о выполнении операции и путь до файлов. В случае неудачи отправляет код ошибки.

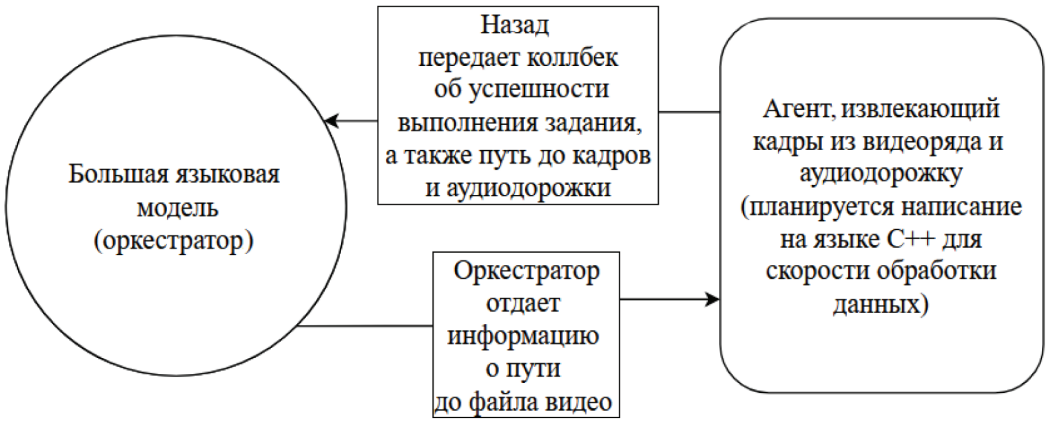

Рисунок 6. Схема взаимодействия оркестратора и агента, декомпозирующего видеолекцию

Следующим этапом обработки является параллельная обработка аудиоданных и последовательности кадров. Для этого разработан AI-агент с моделью OpenAI Whisper-Medium, преобразующей аудио в текст формата JSON-нотаций с таймингами. Интеллектуальные агенты, обрабатывающие кадры, должны извлекать уникальные кадры, содержащие в себе изображения с текстовой информацией, и в зависимости от типа лекций оркестратор подберет наиболее подходящий конвейер обработки видеоряда. Схема данного процесса представлена на рисунке 7.

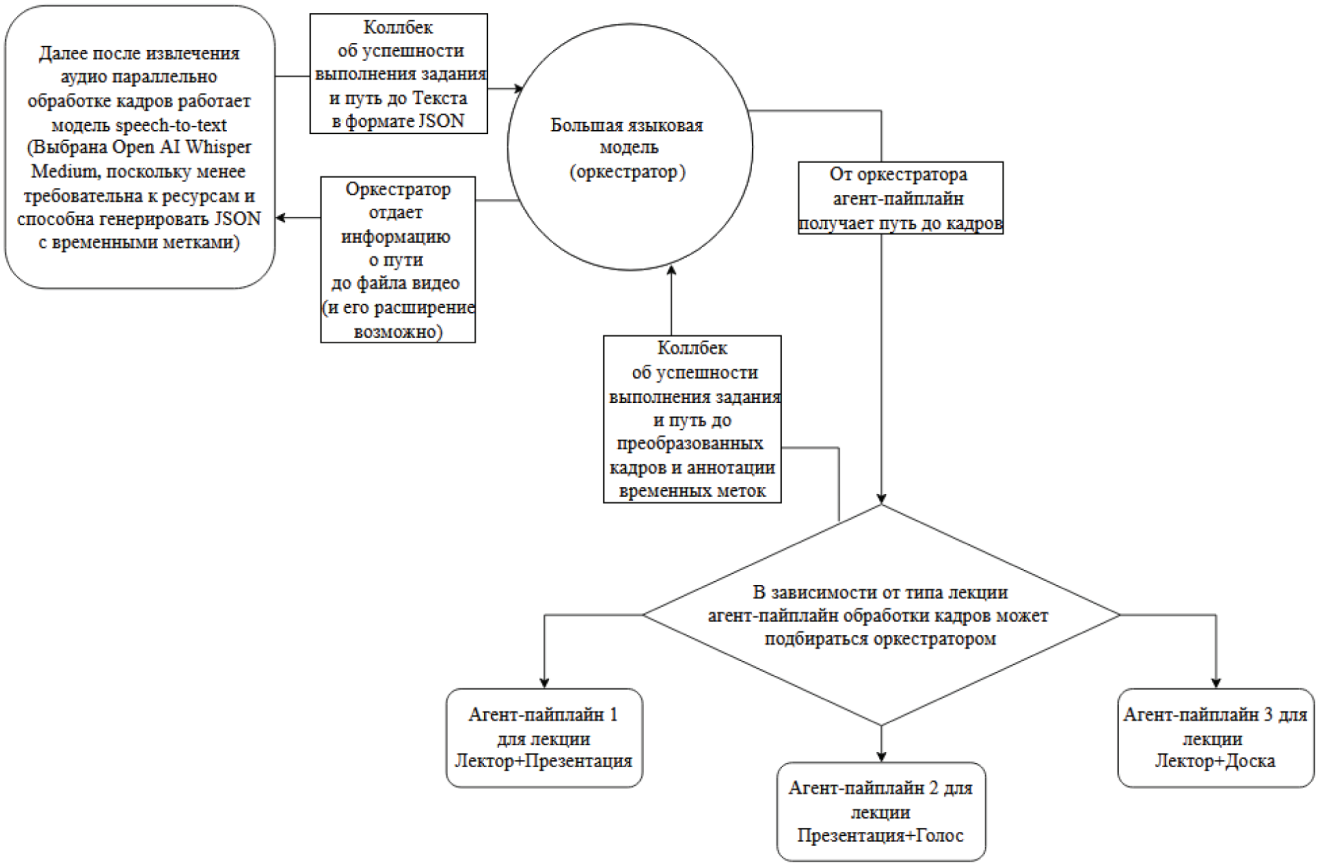

Рисунок 7. Схема взаимодействия оркестратора и агентов обработки аудио- и видеорядов

Последним этапом происходит оптическое распознавание символов и агрегация текстовых файлов в формате JSON в единый текстовый документ. Для этого используются ИИ-агенты, основанные на моделях Mistral OCR, LeChat, Qwen и Google Gemini Flash 2.0. Финальный документ передается в интерфейс пользователя для ознакомления с последующей возможностью скачивания или печати, схема процесса представлена на рисунке 8.

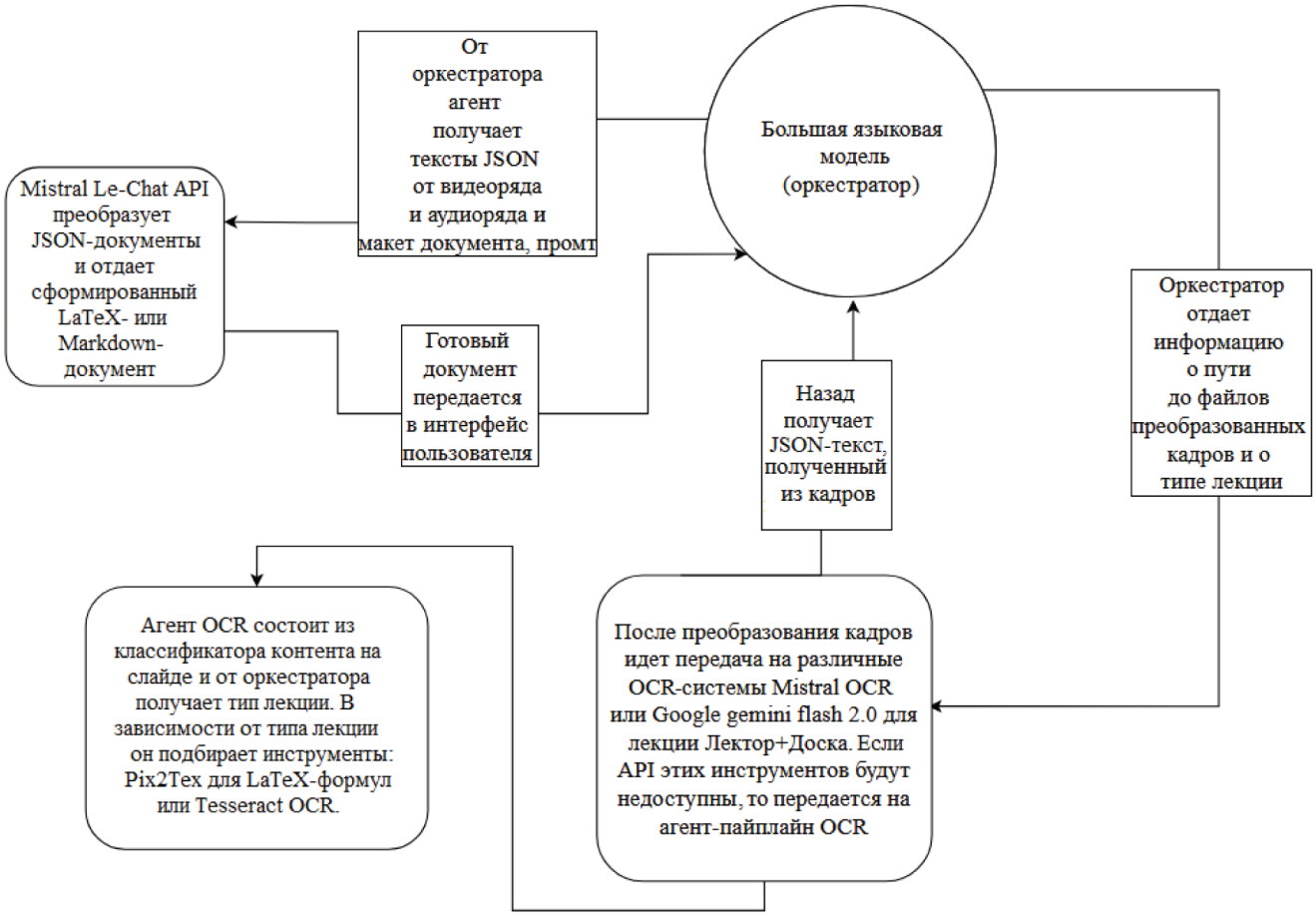

Рисунок 8. Схема взаимодействия оркестратора и агентов оптического распознавания символов и агрегаторов текстового документа

Особенности реализации предложенного метода. Ключевым элементом представленной работы является предложенный гибридный подход, интегрирующий конвейерную обработку видеопотока в мультиагентную систему. Этот подход формирует основное архитектурное решение, где этапы извлечения и предварительной обработки видеоданных из лекций выстраиваются в последовательный конвейер, результаты работы которого затем используются специализированными агентами для решения конкретных задач. Подобная интеграция позволяет эффективно управлять сложностью обработки видео и распределять вычислительную нагрузку между агентами, развернутыми как на локальных серверах, так и взаимодействующими с внешними API. Поскольку типы лекций разные, то и для каждого уникального случая возможно выстраивать свой конвейер обработки, тем самым гипотетически обрабатывать все виды видеолекций.

ЗАКЛЮЧЕНИЕ И ВЫВОДЫ

Промежуточная стадия работы была сосредоточена на разработке и обосновании архитектуры мультимодальной мультиагентной системы и выборе специализированных моделей для каждого этапа конвейера обработки видеолекций. В качестве оркестратора предложены LLM-модели Qwen 72b-Instruct и Mistral Large-2 Instruct, для преобразования аудиодорожки в текст выбрана модель OpenAI Whisper-Medium, для извлечения и анализа ключевых кадров – пайплайн-алгоритмы с нейросетевыми моделями, а для оптического распознавания текста – Mistral OCR и Google Gemini Flash 2.0. Выбранная гибридная архитектура «Оркестратор-Исполнитель» обеспечивает динамическое разбиение задач,

параллельную обработку и автоматическое перераспределение при сбоях, а уникальные виды обработки (селекция уникальных кадров, способы агрегации JSON-аннотаций, гибкая настройка конвейерных сегментов под разные форматы лекций) заложены в основу представленной системы. Далее предстоит реализовать прототип и оценить эффективность предложенной архитектуры на реальных видеолекциях.

СПИСОК ЛИТЕРАТУРЫ

1. Zhao, B. Hierarchical multimodal transformer for long video generation / B. Zhao, M. Gong, X. Li. – DOI 10.1016/j.neucom.2021.10.039 // Neurocomputing. – 2022. – Vol. 471. – P. 36–43.
2. VDTR: Video Deblurring with Transformer / M. Cao, Y. Fan, Y. Zhang [et al.]. – DOI 10.1109/TCSVT.2022.3201045 // IEEE Transactions on Circuits and Systems for Video Technology. – 2022. – Vol. 33. – P. 160–171.
3. Efficient Training of Audio Transformers with Patchout / K. Koutini, J. Schlüter, H. Eghbal-zadeh, G. Widmer. – DOI 10.21437/Interspeech.2022-227 // Interspeech. – 2022. – P. 2753–2757.
4. Comprehensive Survey on Applications of Transformers for Deep Learning Tasks / S. Islam, H. Elmekki, A. Elsebai [et al.]. – DOI 10.48550/arXiv.2306.07303 // ArXiv. – URL: https://arxiv.org/html/2306.07303 (date of application: 21.06.2025).
5. Large Language Model Should Understand Pinyin for Chinese ASR Error Correction / Y. Li, X. Qiao, X. Zhao [et al.] // ArXiv. – URL: https://arxiv.org/abs/2409.13262 (date of application: 21.06.2025).
6. AudioPaLM: A Large Language Model That Can Speak and Listen / P. K. Rubenstein, C. Asawaroengchai, A. Bapna [et al.] // ArXiv. – URL: https://arxiv.org/abs/2306.12925 (date of application: 21.06.2025).
7. Gutowska, A. What is a multiagent system? / A. Gutowska // IBM сайт. – URL: https://www.ibm.com/think/topics/multiagent-system/ (date of application: 21.06.2025).
8. Ismagulov, M. E. Methods and Algorithms for Multimodal Conversion of Video Lectures / M. E. Ismagulov // Proceedings of the XXIV International Conference on Information Technologies and Mathematical Modelling (ITMM-2024) (Tomsk, 2024). – Tomsk : Tomsk State University, 2024. – P. 605–607. – URL: https://www.researchgate.net/publication/391833448_1_Conference_proceedings_with_your_article_Ismagulov_M_E_Methods_and_Algorithms_for_Multimodal_Conversion_of_Video_Lectures (date of application: 17.05.2025).
9. Лекция 10. Распределенные интеллектуальные системы на основе агентов // Ronl. – URL: https://ronl.org/lektsii/informatika/882253/ (дата обращения: 21.06.2025).
10. A decentralized optimization approach for scalable agent-based energy dispatch and congestion management / M. Kilthau, V. Henkel, L. P. Wagner [et al.]. – DOI 10.1016/j.apenergy.2024.124659 // Applied Energy. – 2025. – Vol. 377, Part C. – URL: https://www.sciencedirect.com/science/article/pii/S0306261924020427?via%3Dihub (date of application: 17.05.2025).
11. Zhang, H. L. Classification of Intelligent Agent Network Topologies and a New Topological Description Language for Agent Networks / H. L. Zhang, C. H. C. Leung, G. K. Raikundalia. – DOI 10.1007/978-0-387-44641-7_3 // Intelligent Information Processing III : Proceedings of the IFIP International Conference. – Boston : Springer, 2006. – P. 21–31.
12. Mwifunyi, R. J. Distributed approach in fault localisation and service restoration: State-of-the-Art and future direction / R. J. Mwifunyi, M. M. Kissaka, N. H. Mvungi. – DOI 10.1080/23311916.2019.1628424 // Cogent Engineering. – 2019. – Vol. 6. – P. 1–20. – URL: https://www.researchgate.net/publication/344738267_Distributed_approach_in_fault_localisation_and_service_restoration_State-of-the-Art_and_future_direction (date of application: 08.06.2025).
13. Finio, M. What is AI agent orchestration? / M. Finio, A. Downie // IBM. – URL: https://www.ibm.com/think/topics/ai-agent-orchestration (date of application: 21.06.2025).
14. Falconer, S. The orchestrator-worker pattern is a well-known design pattern for structuring multi-agent systems / S. Falconer // LinkedIn. – URL: https://www.linkedin.com/posts/seanf_the-orchestrator-worker-pattern-is-a-well-known-activity-7294775230353313792-_zFL (date of application: 21.06.2025).
15. Orchestrator-Workers Workflow // Java AI Dev. – URL: https://javaaidev.com/docs/agentic-patterns/patterns/orchestrator-workers-workflow/ (date of application: 21.06.2025).