УДК 004.89
DOI: 10.26102/2310-6018/2026.58.7.012

Мультимодальные нейросети и локально обученные модели в задачах извлечения структурированных фактов

Н.Г. Аснина[1]✉, М.О. Ухин[2], Е.В. Нетесов[1]

[1]Воронежский государственный технический университет, Воронеж, Российская Федерация
[2]Воронежский государственный университет, Воронеж, Российская Федерация

Резюме. Социальный капитал стал значимым профессиональным ресурсом, однако удержание в памяти большого массива контекстных данных затруднено. Проблему решают интеллектуальные боты Personal CRM, способные извлекать полезные факты из текста и речи. При проектировании таких систем ключевой дилеммой становится выбор архитектуры NLP в условиях строгих требований к конфиденциальности. В работе проведено сравнение производительности и ресурсоемкости четырех практических подходов. Наивысшую скорость работы показала изолированная векторная модель, расходующая около 1,9 ГБ оперативной памяти при обработке выборки за 14 секунд. Связка распознавания речи с компактной LLM (Qwen 2.5-0.6B) увеличивает задержку отклика до 25 секунд. Данной конфигурации требуется 2,5 ГБ ОЗУ, причем усложнение промптов резко повышает расход памяти, а пиковая нагрузка на процессор достигает 600 %. Гибридная схема требует около 3,0 ГБ ОЗУ при сопоставимом времени работы в 24 секунды, но снижает вычислительную нагрузку на CPU до 495 %. Оптимизация достигается за счет точечного подключения тяжеловесной модели. Запуск цельной мультимодальной сети оказался технически нецелесообразным: инференс превысил 40 минут при потреблении 5,8 ГБ памяти. Для локальных систем целесообразно использовать либо полностью автономный самообученный конвейер, обеспечивающий максимальную скорость, либо гибридную маршрутизацию. В гибридном варианте легкая модель фильтрует повседневные задачи, а языковая сеть привлекается исключительно для разрешения семантических неоднозначностей.

Ключевые слова: мультимодальные нейросети, LLM, самообученные модели, NLP, извлечение фактов, Personal CRM.

Для цитирования: Аснина Н.Г., Ухин М.О., Нетесов Е.В. Мультимодальные нейросети и локально обученные модели в задачах извлечения структурированных фактов. *Моделирование, оптимизация и информационные технологии.* 2026;14(7). URL: https://moitvivt.ru/ru/journal/article?id=2470 DOI: 10.26102/2310-6018/2026.58.7.012

Multimodal neural networks and locally trained models in structured fact extraction tasks

N.G. Asnina[1]✉, M.O. Ukhin[2], E.V. Netesov[1]

[1]Voronezh Technical State University, Voronezh, the Russian Federation
[2]Voronezh State University, Voronezh, the Russian Federation

Abstract. Social capital has become a significant professional resource; however, retaining a large volume of contextual data in memory is challenging. This problem is addressed by intelligent Personal CRM bots capable of extracting useful facts from text and speech. When designing such systems, a key dilemma is selecting an NLP architecture under strict confidentiality requirements. This paper compares the performance and resource consumption of four practical approaches. The isolated vector model demonstrated the highest processing speed, consuming approximately 1.9 GB of RAM with a sample processing time of 14 seconds. The combination of speech recognition with a compact LLM (Qwen

2.5-0.6B) increases the response latency to 25 seconds. This configuration requires 2.5 GB of RAM; furthermore, complicating the prompts sharply increases memory consumption, while the peak CPU load reaches 600 %. The hybrid scheme requires about 3.0 GB of RAM with a comparable processing time of 24 seconds, but it reduces the computational load on the CPU to 495 %. This optimization is achieved through the selective activation of the heavy-weight model. The deployment of a unified multimodal network proved technically unfeasible: inference exceeded 40 minutes while consuming 5.8 GB of memory. For local systems, it is advisable to use either a fully autonomous self-trained pipeline, which ensures maximum speed, or hybrid routing. In the hybrid variant, a lightweight model filters routine tasks, and the language network is engaged exclusively to resolve semantic ambiguities.

Keywords: multimodal neural networks, LLM, self-trained models, NLP, fact extraction, Personal CRM.

For citation: Asnina N.G., Ukhin M.O., Netesov E.V. Multimodal neural networks and locally trained models in structured fact extraction tasks. *Modeling, Optimization and Information Technology.* 2026;14(7). (In Russ.). URL: https://moitvivt.ru/ru/journal/article?id=2470 DOI: 10.26102/2310-6018/2026.58.7.012

Введение

В современном информационном обществе социальный капитал, выраженный в совокупности межличностных связей, доверия и взаимной поддержки, становится одним из ключевых стратегических ресурсов для профессионального и личностного развития [1]. Динамичное развитие цифровых коммуникаций привело к экспоненциальному росту числа ежедневных социальных контактов, однако когнитивные возможности человеческой памяти по-прежнему ограничены естественными биологическими факторами [2, 3]. Поддержание долгосрочных и качественных связей в таких условиях требует постоянного удержания в памяти огромного массива контекстной информации: деталей прошлых диалогов, профессиональных интересов собеседников, личных договоренностей и совместных планов. Для преодоления данного когнитивного барьера сегодня активно развиваются интеллектуальные персональные системы управления связями, известные как Personal CRM. Подобные программные комплексы, ориентированные на конечного пользователя, чаще всего проектируются в виде интеллектуальных ассистентов, способных в режиме реального времени анализировать входящие неструктурированные текстовые и голосовые данные, автоматически выделяя, категоризируя и сохраняя из них полезные факты.

Ключевой технической дилеммой, определяющей архитектурный облик и жизнеспособность таких систем, является выбор оптимального подхода к обработке естественного языка и извлечению структурированной информации из спонтанной речи. В современной индустрии искусственного интеллекта наметилось два полярных направления решения этой задачи. Первое направление предполагает использование мощных универсальных больших языковых и мультимодальных моделей. Данный подход привлекает высокой степенью обобщения и способностью нейросетей качественно решать задачи анализа сложных семантических конструкций «из коробки», без необходимости предварительного дообучения на специфических наборах данных. Но у внедрения масштабных моделей есть два критических барьера. Главная проблема – это угроза информационной безопасности. Чтобы полноценно задействовать такие нейросети, чаще всего приходится отправлять чувствительную персональную информацию на удаленные облачные серверы через сторонние API. К тому же, для их работы требуются колоссальные вычислительные мощности.

В качестве альтернативы мы предлагаем другой путь: с нуля обучить и развернуть собственную узкоспециализированную модель. Она должна быть легковесной и запускаться непосредственно на локальном железе пользователя или на приватном

сервере. Что дает такая архитектура? Прежде всего, она гарантирует стопроцентную конфиденциальность обрабатываемых данных. Кроме того, система перестает зависеть от внешних поставщиков услуг, а требования к аппаратному обеспечению снижаются в разы. Однако реализация этого направления требует проектирования сложных специфических конвейеров для предобработки акустических и текстовых данных, их векторизации и последующей классификации, уступая при этом крупным универсальным моделям в гибкости восприятия широкого контекста.

Компромиссным решением между ресурсоемкими большими нейросетями и простыми локальными алгоритмами выступает гибридный подход, в основе которого лежит механизм динамической маршрутизации запросов в зависимости от их семантической сложности. В рамках этой архитектуры все поступающие сообщения изначально обрабатывает бинарный классификатор, выступающий в роли первичного фильтра для отделения конкретных целевых поручений от обычной фоновой беседы. Если классификатор демонстрирует высокую степень уверенности в своем решении, система мгновенно фиксирует результат с минимальной нагрузкой на процессор, однако при смысловой неоднозначности запрос автоматически перенаправляется к резервной компактной языковой модели для более глубокого анализа контекста. В результате такой каскадный конвейер позволяет быстро и с существенной экономией оперативной памяти обрабатывать основную массу рутинных сообщений, точечно подключая ресурсоемкий искусственный интеллект лишь для разрешения спорных ситуаций, что гарантирует высокую точность извлечения фактов при полном сохранении конфиденциальности данных пользователя.

На сегодняшний день использование крупных мультимодальных систем и больших языковых моделей стало фактической нормой и индустриальным стандартом при решении большинства задач обработки естественного языка. Однако прямой перенос этого массового тренда в разработку персональных интеллектуальных ассистентов порождает явное научно-практическое противоречие. С одной стороны, системы управления социальными связями объективно нуждаются в глубоком семантическом анализе естественной речи, который с легкостью обеспечивают современные ресурсоемкие нейросети. С другой стороны, практическая реализация таких систем для повседневного использования жестко обусловлена необходимостью обеспечения высокой скорости обработки запросов (в режиме реального времени) и строгой экономии системных ресурсов, в частности, процессорного времени и оперативной памяти. В связи с этим возникает необходимость проведения комплексного исследования, направленного на практическое сравнение производительности тяжеловесных индустриальных архитектур и локально обученных специализированных конвейеров. В данной статье рассматривается техническая целесообразность обоих подходов на базе разработанного экспериментального программного комплекса с целью определения наиболее оптимальной архитектуры для извлечения фактов в условиях бескомпромиссных требований к скорости отклика и минимизации аппаратных затрат.

Материалы и методы

Фундаментальным требованием при проектировании архитектуры Personal CRM являлось обеспечение высокой скорости обработки запросов и жесткая оптимизация потребляемых вычислительных ресурсов (CPU и RAM). Подобные аппаратные ограничения, равно как и использование среды Docker для удобного развертывания, продиктованы целевой концепцией продукта: разрабатываемая система ориентирована либо на запуск в формате стартапа, где критически важна минимизация издержек на серверное оборудование, либо в виде полностью автономного self-hosted решения,

требующего гарантированно стабильной работы на ограниченных мощностях конечных пользователей. Отказ от использования облачных коммерческих API и передачи трафика на сторонние серверы обусловил выбор в пользу полностью локального развертывания нейросетевых конвейеров. Для проведения экспериментов был разработан программный стенд на базе микросервисной архитектуры [4, 5]. Взаимодействие между компонентами системы (точкой входа, хранилищем и вычислительными узлами) осуществлялось асинхронно через брокер сообщений, что позволило разделить потоки данных и точно измерить производительность каждого изолированного узла. В рамках исследования были спроектированы и протестированы четыре независимые реализации модуля извлечения фактов (processor), развернутые в контейнеризированной среде Docker (Рисунок 1).

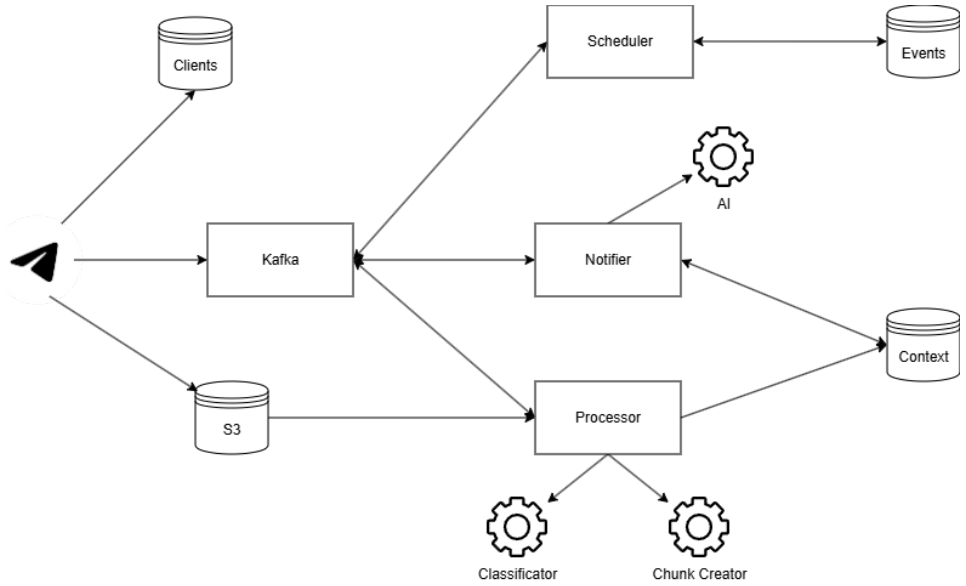

Рисунок 1 –Архитектура системы
Figure 1 – System architecture

Первый из исследуемых подходов предполагал построение собственного легковесного NLP-конвейера в виде изолированной самообученной векторной модели. Архитектура модуля состояла из последовательных этапов: локального распознавания входящего аудиопотока, предобработки транскрипции и смыслового анализа. Полученный текст проходил стадию токенизации с использованием библиотеки razdel, оптимизированной для русскоязычного сегмента. Ядром модуля выступал ансамбль классификаторов интентов и экстракторов именованных сущностей (NER), работающий на векторных представлениях слов [6, 7]. Подобные алгоритмы на базе глубокого обучения гибко адаптируются и демонстрируют высокую точность при автоматическом извлечении сущностей из различных видов специализированного контента [8, 9]. Кроме того, методы NER успешно применяются для автоматического поиска данных о мероприятиях [10], что напрямую отвечает задачам извлечения фактов из повседневных коммуникаций. Ввиду отсутствия готовых открытых датасетов для систем Personal CRM был разработан собственный пайплайн синтеза данных. С помощью автоматизированных скриптов генерировались тренировочные выборки, имитирующие специфику повседневных диалогов, договоренностей и фактологических утверждений. В результате модель запускалась в легковесном контейнере, демонстрируя минимальный объем аллоцируемой оперативной памяти и высокую скорость отклика (Рисунок 2).

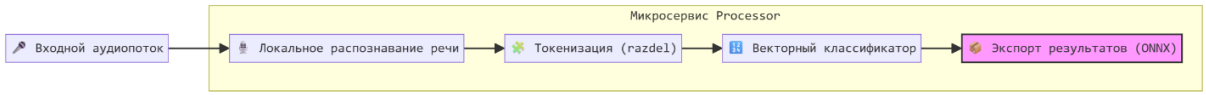

Рисунок 2 – Внутренний конвейер самообученной модели
Figure 2 – Internal pipeline of self-trained model

Вторая реализация базировалась на интеграции с легковесными текстовыми LLM, представляя собой связку модуля распознавания речи и локально развернутой компактной большой языковой модели (на базе архитектуры Qwen 2.5-0.6B). Обработка данных осуществлялась путем передачи транскрибированного текста в контекстное окно LLM. Основным методом извлечения фактов выступал промпт-инжиниринг: от модели требовался строгий возврат данных в формализованном виде (структурированный JSON), пригодном для записи в базу данных. В ходе тестирования выявилась критическая проблема масштабируемости данного подхода. Как только мы добавляли в системный промпт новое логическое ограничение или пример, расход ресурсов резко подскакивал. Практика показала следующее: каждое усложнение инструкций «съедает» в среднем около 400 МБ оперативки на один микросервис. В условиях, когда аппаратные мощности жестко лимитированы, это делает работу модели крайне нестабильной.

Третий путь – это компромиссная гибридная архитектура. В ее основе лежит динамическая семантическая маршрутизация. Мы объединили автономный векторный конвейер и языковую модель, чтобы распределять вычислительную нагрузку каскадом. В роли первичного диспетчера мы задействовали локальный бинарный классификатор. Данная модель выполняет предварительный скоринг текста. Когда расчетная вероятность попадает в пограничный интервал (примерно от 0,2 до 0,8), запрос классифицируется как семантически неоднозначный. При таком сценарии текст автоматически уходит на обработку в компактную LLM. Языковая сеть берет на себя глубокий анализ контекста, после чего выдает структурированный ответ. Напротив, при высокой уверенности базового алгоритма весь вычислительный процесс мгновенно завершается локально. Согласно результатам мониторинга, подобная маршрутизация эффективно снимает проблему нехватки аппаратных мощностей. Средняя нагрузка на центральный процессор существенно падает. Снижение обусловлено точечным подключением ресурсоемкой генеративной нейросети: мы используем ее исключительно для разбора сложных или нетривиальных формулировок. Такой подход позволяет соблюсти строгий баланс между качеством извлечения фактов и стабильной работой оборудования.

Технически наиболее сложным оказался четвертый вариант. Мы протестировали единую мультимодальную сеть формата end-to-end на базе Qwen 2.5-7B. Изначальная гипотеза заключалась в полном отказе от каскадной архитектуры. Ожидалось, что модель будет напрямую принимать сырой аудиопоток, сама находить нужные смысловые блоки и отдавать бизнес-контекст. То есть, промежуточный этап транскрипции выпадал совсем. Но на практике мы столкнулись с непреодолимыми системными барьерами. Мультимодальные трансформеры устроены так, что для обработки кросс-модальных связей им требуется загрузка массивных графов вычислений. При запуске в локальной среде развертывания со строго заданными лимитами Docker-контейнера модель не смогла пройти этап инициализации и загрузки весов в память («холодный старт»).

Сбор метрик производительности проводился в строго идентичных аппаратных условиях. Во избежание искажений от внешних систем мониторинга применялось внутреннее профилирование непосредственно на уровне кода микросервисов. Для каждой архитектуры система автоматически логировала ключевые показатели:

утилизацию CPU, пиковое потребление и прирост оперативной памяти (Delta), время обработки тестовой выборки, а также длительность «холодного старта» (загрузки весов модели).

Результаты

Важным методологическим условием проведения экспериментов и сбора метрик являлось строгое соблюдение микросервисной парадигмы. Все компоненты системы для обработки данных, вплоть до тяжеловесных мультимодальных сетей, разворачивались, запускались и тестировались исключительно в изолированных Docker-контейнерах. Такой подход позволил не только смоделировать реальные условия эксплуатации программного комплекса, но и получить максимально объективные данные о потреблении системных ресурсов каждым из архитектурных решений в строго заданных граничных рамках. Сравнительный анализ производительности проводился по пяти ключевым параметрам, критически важным для систем класса Personal CRM: пиковое потребление оперативной памяти (RAM), динамический прирост потребления памяти в процессе работы (Delta), утилизация процессорного времени (CPU), время обработки запроса и время холодного старта микросервиса. Полученные в ходе тестирования количественные показатели представлены в Таблице 1.

Таблица 1 – Сравнительный анализ производительности
Table 1 – Comparative performance analysis

	Потребление CPU, %	**Пиковое потребление RAM, GB**	**Прирост RAM (Delta), MB**	**Время обработки, с**	**Время холодного старта, с**
Локальная самообученная модель	~ 380	~ 1,9	~ 250	~ 14,1	~ 19,8
Гибридная маршрутизация	~ 495	~ 3,0	~ 1225	~ 24	~ 20,2
Распознавание речи + Qwen 2.5-0.6B	~ 600	~ 2,5	~ 925	~ 25	~ 23,6
Qwen 2.5-7B	~ 150	~ 5,8	~ 4800	~ 2483,2	~ 22,4

Анализ зафиксированных метрик наглядно демонстрирует, что использование локальной самообученной модели на базе легковесных векторных представлений обеспечивает наилучшие показатели ресурсоэффективности среди всех протестированных архитектур. При запуске в изолированном Docker-контейнере пиковое потребление оперативной памяти данной моделью составило всего около 1,9 ГБ при минимальном динамическом приросте (Delta) порядка 250 МБ в момент активного инференса. В этом заключается ее главное преимущество. Она обходит и гибридную схему, и тяжеловесные языковые сети, которым, как известно, нужно выделять куда больше памяти. Процессор загружается примерно на 380 %. О чем это говорит? Матричные вычисления отлично распараллеливаются на доступных ядрах. При этом операционная система не испытывает критической нагрузки (а ведь это классическая проблема при запуске полного цикла LLM). Если говорить о скорости, то здесь самообучаемая модель тоже в лидерах. Тестовую выборку она обрабатывает в среднем за 14 с небольшим секунд. Это почти вдвое быстрее, чем работают текстовые сети или гибридный вариант. Про тяжелые мультимодальные решения и говорить не приходится – отрыв колоссальный. Холодный старт занимает чуть меньше 20 секунд. Все это делает

базовую архитектуру весьма стабильной. Ее удобно динамически масштабировать на лету. В итоге выбор специализированного инструмента полностью себя оправдывает.

Гибридная архитектура построена на динамической семантической маршрутизации. Замеры показали, что это отличный компромиссный вариант. Как работает каскадное распределение нагрузки, хорошо видно по цифрам. В пике расходуется около 3 ГБ оперативки, а динамический скачок (Delta) держится в районе 1200 МБ. Откуда такие аппетиты? Системе приходится держать в памяти сразу два компонента: базовый классификатор и резервную языковую модель. Но есть и огромный плюс – процессор разгружается весьма ощутимо. Нагрузка на CPU не превышает 495 %. Для полного цикла LLM это недостижимый результат. Секрет кроется в точечном использовании тяжелой генеративной сети. Она включается только тогда, когда смысл запроса неоднозначен. Всю остальную рутину мгновенно отсекает первичный фильтр. Тестовая выборка обрабатывалась 24 секунды. Да, время сопоставимо с работой обычной монолитной текстовой нейросети. Зато гибрид ведет себя гораздо предсказуемее. Операционной системе больше не грозит потоковое голодание. Получается очень сбалансированный подход. Он идеально подходит для ситуаций, когда можно пожертвовать лишней памятью ради высокой точности извлечения фактов, сохраняя при этом стабильность железа.

Теперь разберем связку из локального распознавателя речи и небольшой LLM (Qwen 2.5-0.6B). Как и ожидалось, модель блестяще справляется со сложным контекстом. Но за эту гибкость приходится платить высокими требованиями к инфраструктуре. Контейнер съедает в пике примерно 2,5 ГБ памяти. Во время генерации ответа наблюдается прирост (Delta) почти на 900 МБ. Процессор при этом стабильно загружен под 600 %. Удивляться тут нечему: механизма маршрутизации нет, поэтому ресурсоемкая нейросеть лопатит абсолютно весь поток данных, включая самые примитивные запросы. Полный цикл обработки занял 25 секунд. Поскольку весь массив текста идет через тяжелую модель, использовать этот метод стоит только для фоновых асинхронных задач. Если нужен отклик в реальном времени, логичнее взять что-то полегче. Правда, холодный старт порадовал – порядка 23,5 секунд. Это вполне приемлемый показатель, на уровне гибридной архитектуры, так что с потенциальным автомасштабированием микросервиса критических проблем не будет.

Исследование единой мультимодальной сети (Qwen 2.5-7B) представляет особый интерес. Главная фишка здесь в концепции end-to-end обработки. Мы хотели анализировать аудио напрямую, минуя текстовую транскрипцию. Первичный холодный старт оказался на удивление быстрым – чуть больше 22 секунд. Но архитектура мультимодальных трансформеров диктует свои суровые правила. Им критически важна экстремальная пропускная способность оперативки. Мы развернули систему локально, опираясь исключительно на мощности центрального процессора. Итог? Пиковый расход ОЗУ пробил отметку в 5,8 ГБ, а динамический скачок составил почти 4800 МБ. Инференс тестовой выборки растянулся на 41 с лишним минуту. Зато процессор откровенно простаивал: утилизация едва доходила до 150 %. Налицо классическое «бутылочное горлышко». Память просто не справляется без графических ускорителей (GPU). Идея мультимодальных сетей звучит очень мощно, но давайте смотреть правде в глаза. Без специализированного аппаратного обеспечения внедрять их в локальные Personal CRM сейчас нецелесообразно.

Обсуждение

Если обобщить метрики, вывод напрашивается сам собой. В условиях нехватки вычислительных ресурсов специализированные локальные конвейеры и гибридные

схемы безнадежно обходят универсальные генеративные модели полного цикла. Мы успешно внедрили самообучаемую векторную модель. Практика доказала: если нужно просто вытащить структурированные факты для Personal CRM, старые добрые методы классификации интентов и NER (выделение именованных сущностей) остаются самым прагматичным выбором. Почему модель потребляет всего 1,9 ГБ памяти и выдает результат за 14 секунд? Ответ кроется в детерминированном инференсе. Сам граф вычислений очень легкий. За счет этого система ведет себя максимально предсказуемо. Логичным развитием этой идеи выступает гибридная маршрутизация. Базовая ресурсоэффективность сохраняется, но в запутанных семантических кейсах система привлекает на помощь LLM.

Теперь о главных проблемах. Наши эмпирические данные наглядно подтверждают: прямое монолитное развертывание языковых или мультимодальных моделей сугубо на CPU сопряжено с жесткими технологическими барьерами. На обычных потребительских машинах нейросети часто выдают низкую фактологическую точность. Чтобы выжать из компактной LLM нормальный ответ, приходится глубоко уходить в промпт-инжиниринг. В системный запрос добавляются жесткие ограничения, многоуровневые правила и тяжеловесные примеры контекста. Помогают ли такие корректировки? Не всегда. Расширение промпта не гарантирует повышения качества контента, зато провоцирует колоссальный рост потребления ОЗУ и перегружает аппаратную часть. Как следствие, возникает технический тупик. На данном этапе невозможно одновременно обеспечить достоверность данных и жесткие стандарты отказоустойчивости, которые нужны для работы интерфейса в реальном времени.

Напротив, смещение фокуса в сторону использования легковесных самообученных моделей и гибридных диспетчеров является стратегическим плюсом как для конечных пользователей, так и для бизнеса. Для пользователей self-hosted решений это гарантирует быстрый отклик приложения и исключает необходимость приобретения дорогостоящего оборудования со специализированными графическими ускорителями (GPU). Для бизнес-модели и инфраструктуры разработчика (в формате стартапа) внедрение таких компактных конвейеров открывает возможности кратного горизонтального масштабирования. Минимальные требования к вычислительной мощности позволяют безболезненно разворачивать существенно большее количество независимых инстансов микросервиса на одних и тех же аппаратных узлах. Это значительно увеличивает общую пропускную способность системы, позволяя успешно обрабатывать пиковые объемы входящего трафика при минимальном потреблении ресурсов и низких операционных затратах.

Заключение

Результаты экспериментов продемонстрировали, что прямой перенос индустриальных трендов по монолитному использованию ресурсоемких LLM и мультимодальных сетей в сферу приватных персональных ассистентов сопряжен с существенными технологическими барьерами. Если запускать генеративные модели исключительно на центральных процессорах (без использования видеокарт), сразу начинаются проблемы. Стоит только немного усложнить промпт, как расход оперативки стремительно ползет вверх. Вдобавок вычислительная нагрузка становится просто предельной. Время отклика при этом заметно проседает. Как результат, применять такие сети напрямую в реальном времени на обычном домашнем железе крайне затруднительно. Синхронные сценарии здесь, по сути, не работают.

Совершенно иную картину показал наш специализированный локальный конвейер. Как оказалось, самообучаемая модель расходует ресурсы куда эффективнее.

Мы внедрили оптимизированные векторные представления. Что это дало? Во-первых, стабильно высокую скорость инференса. Во-вторых, весьма скромное потребление памяти на старте – где-то в районе 1,9 ГБ. Практическая значимость данного подхода заключается в том, что он гарантирует отказоустойчивую работу системы, а для бизнеса открывает возможности кратного горизонтального масштабирования инфраструктуры с минимальными операционными затратами при развертывании множества независимых инстансов.

На основании полученных данных делается вывод, что наиболее оптимальным и стратегически оправданным вектором развития интеллектуальных агентов, оперирующих чувствительными социальными данными, является применение архитектур гибридной маршрутизации. В рамках такой парадигмы легковесный самообученный конвейер берет на себя роль энергоэффективного ядра, выступая первичным интеллектуальным диспетчером для мгновенной обработки основной массы рутинных запросов. При этом тяжеловесные языковые модели привлекаются исключительно точечно – для разрешения семантических неоднозначностей и сложных аналитических коллизий. Подобный каскадный подход обеспечивает идеальный баланс между высокой точностью извлечения фактов, абсолютной конфиденциальностью данных пользователя и гарантированной аппаратной стабильностью системы.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Гвоздиков Д.С. Онлайн-сети и развитие сетевых взаимодействий. *Вестник СПбГУ. Сер. 12.* 2015;(2):100–107.
2. Лызь Н.А., Гладкая Е.В. Влияние цифровизации на когнитивные процессы: обзор причин и последствий. *Вестник Пермского университета. Философия. Психология. Социология.* 2025;(3):396–405. https://doi.org/10.17072/2078-7898/2025-3-396-405
   Lyz N.A., Gladkaya E.V. The impact of digitalization on cognitive processes: a review of causes and consequences. *Perm University Herald. Philosophy. Psychology. Sociology.* 2025;(3):396–405. (In Russ.). https://doi.org/10.17072/2078-7898/2025-3-396-405
3. Шмаков А.В. Мягкое регулирование в условиях цифровой трансформации: поведенческие основания. *Journal of Institutional Studies.* 2021;13(3):102–116. https://doi.org/10.17835/2076-6297.2021.13.3.102-116
   Shmakov A.V. Nudge in the conditions of digital transformation: Behavioral basis. *Journal of Institutional Studies.* 2021;13(3):102–116. (In Russ.). https://doi.org/10.17835/2076-6297.2021.13.3.102-116
4. Кравченко Д.А. Микросервисная архитектура. Интерактивная наука. 2022:43–44 https://doi.org/10.21661/r-556565
5. Абдирашитулы А. Микросервисная архитектура в NLP платформах: проектирование и масштабируемость. Вестник науки. 2026;1(5):609–619.
6. Рахманов А, Исхакова Н, Абдувалиева З. Представление слов в векторном пространстве применяя модель word2vec. *Eurasian Journal of Mathematical Theory and Computer Sciences.* 2025:5(5)54–59. https://doi.org/10.5281/zenodo.15524744
   Rakhmanov A., Iskhakova N., Abduvalieva Z. Word representation in vector space using word2vec model. *Eurasian Journal of Mathematical Theory and Computer Sciences.* 2025:5(5)54–59. (In Russ.). https://doi.org/10.5281/zenodo.15524744
7. Лыченко Н.М., Сороковая А.В. Сравнение эффективности методов векторного представления слов для определения тональности текстов. *Математические*

структуры и моделирование. 2019;(4):97–110. https://doi.org/10.24147/2222-8772.2019.4.97-110
Lychenko N.M., Sorokovaja A.V. Comparison of effectiveness of word representations methods in vector space for the text sentiment analysis. *Mathematical Structures and Modeling.* 2019;(4):97–110. (In Russ.). https://doi.org/10.24147/2222-8772.2019.4.97-110

8. Титов Ю.П., Кильмишкин Н.В., Кубраков Д.Д., Иванова П.М. Поиск именованных сущностей в инструкциях по медицинскому применению лекарственных средств с использованием глубокого обучения и методов обработки естественного языка. *Труды ИСП РАН.* 2025;37(2):217–236.
Titov Y.P., Kilmishkin N.V., Kubrakov D.D., Ivanova P.M. Use of deep learning and natural language processing techniques for searching named entities in the medical instructions for use of drugs. *Proceedings of ISP RAS.* 2025;37(2):217–236. (In Russ.).
9. Качанов В.В., Хитрова А.С., Марков С.И. Извлечение именованных сущностей из рецензий к исходному коду. *Труды ИСП РАН.* 2023;35(5):193–214.
Kachanov V.V., Khitrova A.S., Markov S.I. Named entity recognition for code review comments. *Proceedings of ISP RAS.* 2023;35(5):193–214. (In Russ.).
10. Ильин Г.А. Поиск и извлечение информации о мероприятиях с использованием методов Named Entity Recognition (NER). Вестник науки. 2025; 3(3):572–576.

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Аснина Наталия Георгиевна, кандидат технических наук, доцент, заведующая кафедрой систем управления и информационных технологий в строительстве, Воронежский государственный технический университет, Воронеж, Российская Федерация.
e-mail: andrey050569@yandex.ru

Asnina G. Natalia, Candidate of Engineering Sciences, Docent, Head of the Department of Management Systems and Information Technologies in Construction, Voronezh State Technical University, Voronezh, the Russian Federation.

Ухин Максим Олегович, бакалавр, Воронежский государственный университет, Воронеж, Российская Федерация.
e-mail: ukhin.maxim@gmail.com

Maxim O. Ukhin, Bachelor, Voronezh State University, Voronezh, the Russian Federation

Нетесов Евгений Владимирович, аспирант, Воронежский государственный технический университет, Воронеж, Российская Федерация.
e-mail: arshavin.07@mail.ru

Evgeniy V. Netesov, Postgraduate, Voronezh State Technical University, Voronezh, the Russian Federation.

Статья поступила в редакцию 01.06.2026; одобрена после рецензирования 10.07.2026; принята к публикации 17.07.2026.

The article was submitted 01.06.2026; approved after reviewing 10.07.2026; accepted for publication 17.07.2026.