

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления
Кафедра интеллектуальных информационных технологий

К защите допустить:

Заведующий кафедрой ИИТ

_____ Д. В. Шункевич

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
по дисциплине «Проектирование баз знаний»
на тему:

**Интеллектуальный ассистент по источнику данных на
основании RAG. Социальные сети**

БГУИР КП6 1-40 03 01 022 ПЗ

Студент гр. 221703
Руководитель

Д. В. Демидовец
М. Е. Садовский

Минск 2025

СОДЕРЖАНИЕ

Перечень условных обозначений	7
Введение	8
1 Анализ подходов к проектированию интеллектуального ассистента по социальным сетям	10
1.1 Характеристика предметной области социальных сетей	10
1.2 Анализ аналогов	12
1.2.1 Jasper AI	12
1.2.2 BrandBastion Lite	12
1.2.3 Brand24	13
1.2.4 FeedHive	14
1.2.5 UpGrow	15
1.3 Подходы к решению задачи	15
1.3.1 Поточковая обработка событий и активных правил	16
1.3.2 Машинное обучение	17
1.3.3 Дообучение	17
1.3.4 Retrieval-Augmented Generation	18
1.4 Вывод	19
2 Проектирование интеллектуального ассистента по социальным сетям	20
2.1 Постановка задачи	20
2.2 Описание пользователей интеллектуального ассистента по социальным сетям	21
2.3 Архитектура системы	22
2.4 Алгоритмы	24
2.4.1 Алгоритм обработки пользовательского ввода	24
2.4.2 Алгоритм краткого пересказа сообщений в диалоге	24
2.4.3 Алгоритм анализа постов, вызвавших наибольшую реакцию	25
2.4.4 Алгоритм анализа подписок для составления описания личности пользователя	25
2.4.5 Алгоритм извлечения часто задаваемых вопросов под постом пользователя или сообщества	25
2.4.6 Алгоритм выявления тем, вызывающих яркие обсуждения в диалоге	26
2.5 Вывод	26
3 Разработка интеллектуального ассистента по социальным сетям	28
3.1 Средства разработки	28
3.1.1 Python	28
3.1.2 C++	28

3.1.3 VK API	28
3.1.4 Langchain	29
3.1.5 Ollama	29
3.1.6 YandexGPT	29
3.1.7 Github	29
3.2 Детали разработки	29
3.3 Личный вклад	32
3.4 Демонстрация основного функционала системы	32
3.5 Вывод	35
Заключение	37
Список использованных источников	39

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

В курсовом проекте используются следующие условные обозначения:

ПрО — Предметная область;

RAG (от англ. Retrieval Augmented Generation) — Генерация с дополненной выборкой;

LLM(от англ. Large Language Model) — Большая Языковая Модель;

OSTIS(от англ. Open Semantic Technology for Intelligent Systems) — Открытые Семантические Технологии для Интеллектуальных Систем;

SC — Semantic Code;

SCg — Semantic Code graphical;

SCn — Semantic Code natural;

SCs — Semantic Code string.

ВВЕДЕНИЕ

Социальные сети — одна из ключевых сфер цифровой жизни современного человека, непрерывно развивающаяся и меняющая взаимодействие между пользователями. С каждым годом появляются новые платформы, форматы контента и алгоритмы, определяющие, каким образом люди общаются, потребляют информацию и выражают себя в Интернете.

Социальные сети охватили большую часть всех пользователей Интернета и значительную часть населения Земли. Объем информации, которую на сегодняшний день можно получить из социальных сетей, колоссален и с каждым днем он увеличивается, как растет и количество пользователей этих ресурсов[1]. По этой причине пользователи социальных сетей часто сталкиваются с проблемой организации и анализа своего цифрового следа: огромный поток информации, ленты новостей, комментарии и личные переписки делают поиск нужных данных сложным и затрудненным.

Однако существующие инструменты предоставляют либо ограниченный набор возможностей, либо требуют самостоятельного анализа, что затрудняет быстрый доступ к информации. В условиях, когда объем создаваемого контента только растёт, а алгоритмы социальных сетей усложняются, пользователям необходимы интеллектуальные системы, способные анализировать их активность, предоставлять персонализированные рекомендации и помогать в управлении взаимодействием.

Для решения этой задачи требуется разработка интеллектуального ассистента, способного обрабатывать данные из социальных сетей, анализировать пользовательские предпочтения, отвечать на заданные вопросы пользователя, а также предлагать аналитические отчёты и рекомендации.

Такой ассистент сможет не только помогать пользователю быстро находить нужную информацию в его ленте или переписке, но и выявлять закономерности в активности, предлагать оптимальные стратегии ведения профиля, а также помогать в поиске интересных тем и обсуждений.

Цель данного курсового проекта — разработка интеллектуального ассистента для работы с социальными сетями, использующего подход Retrieval-Augmented Generation (RAG) для анализа и генерации ответов на основе пользовательских данных. Ассистент должен обеспечивать извлечение ключевой информации из переписок, постов и комментариев, формирование персонализированных рекомендаций и содействие в эффективном взаимодействии пользователя с цифровым пространством.

Для достижения заданных целей были поставлены следующие задачи:

- исследование и анализ потребностей пользователей в области социальных сетей;
- определение ключевых сценариев использования интеллектуального ассистента;
- определение доступных источников данных для использования в запросах пользователей;
- анализ доступных больших языковых моделей (LLM);
- разработка алгоритмов для анализа пользовательских запросов.

1 АНАЛИЗ ПОДХОДОВ К ПРОЕКТИРОВАНИЮ ИНТЕЛЛЕКТУАЛЬНОГО АССИСТЕНТА ПО СОЦИАЛЬНЫМ СЕТЯМ

1.1 Характеристика предметной области социальных сетей

Социальная сеть - интерактивный многопользовательский веб-сайт, контент (т.е. содержание) которого наполняется самими участниками сети[2].

Данный сайт представляет собой автоматизированную, а в перспективе, возможно, и автоматическую социальную среду, позволяющую общаться группе пользователей, объединённых общим интересом.

Развитие социальных сетей в настоящее время носит взрывной характер. Они все в большей степени оказывают влияние не только на повседневную жизнь, но и на политику, и на сам характер общения между людьми. Данный феномен - социальные сети - не просто вторгается в жизнь человека, но и, в определенной степени, начинает трансформировать её под себя[3].

По числу пользователей абсолютными лидерами являются следующие, прежде всего, изначально англоязычные сервисы:

- Facebook - 2.375 млрд пользователей;
- YouTube - 2 млрд пользователей;
- Instagram - 1 млрд пользователей;
- TikTok - 500 млн пользователей;
- ВКонтакте - 8.2 млн пользователей.

Современные социальные сети делятся на несколько категорий в зависимости от их функционала и целевой аудитории. Например, платформы общего назначения, такие как Facebook, ВКонтакте и Twitter (X), ориентированы на широкий круг пользователей и позволяют публиковать разнообразный контент. Видеохостинговые сервисы, такие как YouTube и TikTok, фокусируются на распространении видеоконтента. Профессиональные сети, такие как LinkedIn, предназначены для делового общения и поиска работы, а мессенджеры с элементами социальных сетей, например Telegram и WhatsApp, совмещают возможности личной переписки и публичных каналов.

Одним из ключевых аспектов функционирования социальных сетей является алгоритмическая обработка контента. Использование искусственного интеллекта и методов машинного обучения позволяет платформам анализировать интересы пользователей и персонализировать ленты новостей, выдавая наиболее релевантную информацию. Это способствует увеличению вовлеченности аудитории, но также может приводить к эффекту

«информационного пузыря».

«Информационный пузырь» - негативная сторона персонализированного поиска, явление, при котором веб-сайты определяют, какую информацию пользователь хотел бы увидеть, основываясь на его «цифровом следе»: месторасположении, прошлых поисковых запросах, предпочтениях[4]. Интеллектуальный ассистент может значительно облегчить взаимодействие пользователей с платформами, предлагая персонализированные рекомендации, фильтруя контент и помогая бороться с информационным шумом. Он способен анализировать интересы пользователя и предлагать наиболее релевантные новости, посты, видео и обсуждения, снижая влияние «информационного пузыря».

Интеллектуальный ассистент может помочь исследователям и аналитикам выявлять тенденции, настроения и закономерности в данных социальных сетей, предоставляя ценную информацию о поведении и мнениях в Интернете. Одним из главных преимуществ ИИ при анализе социальных сетей является его способность работать с неструктурированными данными, такими как текстовые сообщения, фотографии и видео, что позволяет анализировать контент, который трудно классифицировать и интерпретировать вручную. В целом, потенциал ИИ при анализе социальных сетей огромен, и его использование может привести к более точному и эффективному принятию решений в различных областях бизнеса и маркетинга.

При анализе социальных сетей могут участвовать различные объекты, представляющие собой элементы взаимодействия пользователей и контента. Основным объектом анализа являются пользователи, которые могут классифицироваться по разным характеристикам, таким как активность, интересы, подписки, связи с другими участниками платформы. Их можно анализировать как индивидуально, так и в группах, выявляя динамику общения. Еще одним важным объектом являются социальные связи, включающие подписки, дружеские связи, группы, чаты и упоминания, так как они помогают выявлять взаимодействия между различными пользователями.

Контент, создаваемый и распространяемый в социальных сетях, представлен в виде постов, комментариев, репостов, изображений и видео. Их анализ позволяет оценивать популярные темы, выявлять тренды и исследовать реакцию аудитории.

Реакции пользователей, включая лайки, дизлайки, репосты, просмотры и комментарии, служат показателями вовлеченности и позволяют оценить успешность публикаций, интерес аудитории и динамику распространения контента.

Таким образом, имеется обширная область информации, которая требует тщательной обработки и анализа для предоставления данных заинтересованным пользователям, для чего и требуется создание интеллектуального ассистента по социальным сетям.

1.2 Анализ аналогов

Разработка для социальных сетей интеллектуального ассистента, способного анализировать данные и отвечать на вопросы пользователей, является уникальной задачей, однако на рынке уже существуют различные решения, которые частично перекрывают функциональность предлагаемой системы.

1.2.1 Jasper AI

Описание: Jasper AI(рисунок 1.1) представляет собой инструмент для маркетинга в социальных сетях, предназначенный для ускорения создания контента с помощью ИИ[5].

Достоинства:

- создание подписей в социальных сетях на основе искусственного интеллекта;
- более 50 специализированных шаблонов контента AI;
- отзывы от пользователей;
- мобильная совместимость.

Недостатки:

- ориентированность в первую очередь на генерацию контента;
- зависимость от шаблонов;
- отсутствие персонализации.

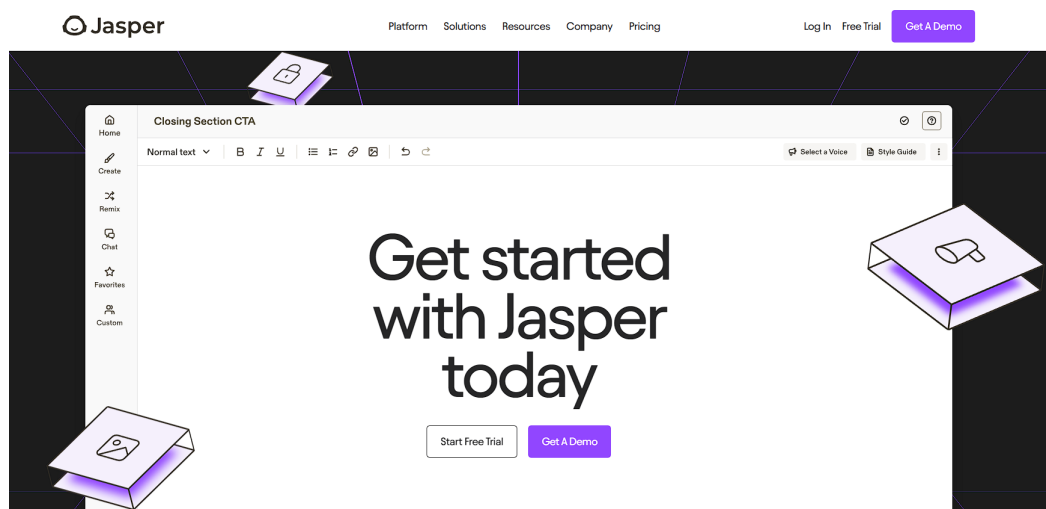


Рисунок 1.1 – Пример Jasper AI

1.2.2 BrandBastion Lite

Описание: BrandBastion Lite(рисунок 1.2) представляет собой нейросеть для управления всеми социальными медиа из одного места.[6].

Достоинства:

- возможность выявления трендов и понимания настроений вашей аудитории;
- использование ИИ для поиска и скрывания потенциально вредных комментариев;
- единый социальный ящик для сообщений из разных платформ.

Недостатки:

- фокус на бренды и компании;
- отсутствие инструментов для генерации постов или ответов;
- минимальная стоимость подписки – 229\$ в месяц.

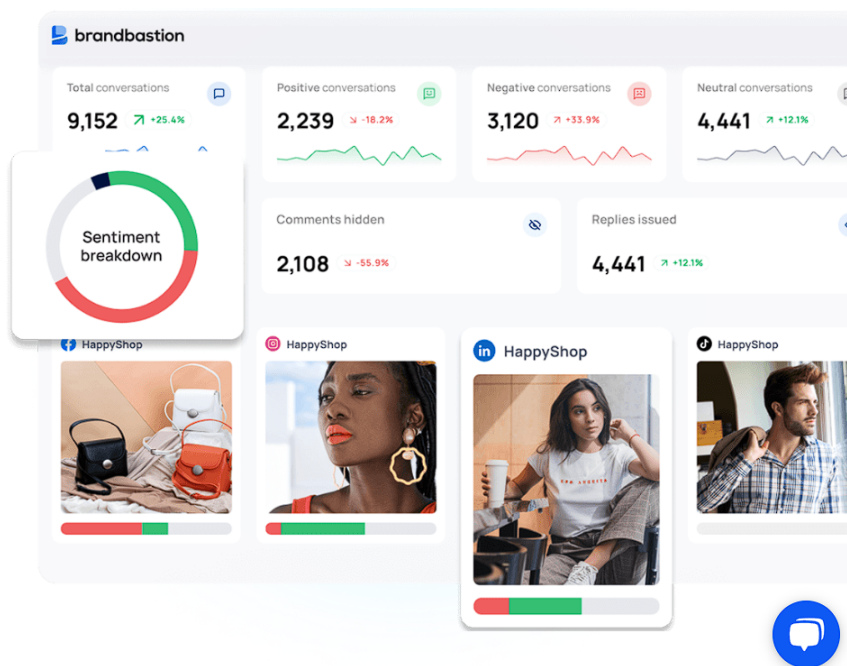


Рисунок 1.2 – Пример BrandBastion Lite

1.2.3 Brand24

Описание: Инструмент Brand24(рисунок 1.3) для прослушивания социальных сетей с искусственным интеллектом предназначен для управления репутацией бренда и сбора информации о клиентах[7].

Достоинства:

- расширенный анализ настроений клиентов;
- мощные показатели для измерения узнаваемости, охвата и вовлеченности бренда;
- отслеживание хэштегов и идентификация влиятельных лиц.

Недостатки:

- фокус на маркетинговые задачи;
- отсутствие функций для создания контента;
- стоимость подписки.

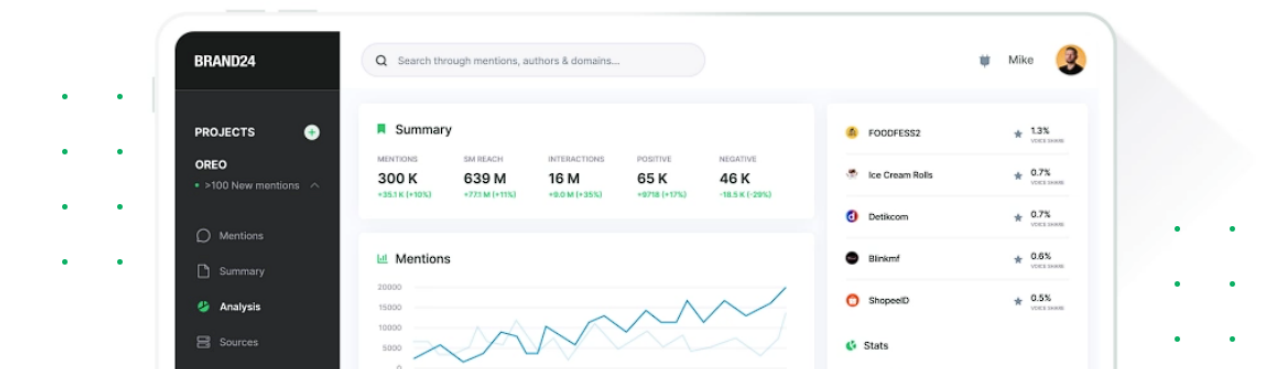


Рисунок 1.3 – Пример Brand24

1.2.4 FeedHive

Описание: FeedHive(рисунок 1.4) — это инструмент для управления социальными медиа, использующий искусственный интеллект для создания, планирования, публикации и управления своим контентом в социальных сетях в большом масштабе[8].

Достоинства:

- автоматизация управления социальными сетями;
- предсказания производительности постов;
- добавление соответствующих посту хештегов;
- простой интерфейс.

Недостатки:

- ограниченная аналитика;
- платная подписка.

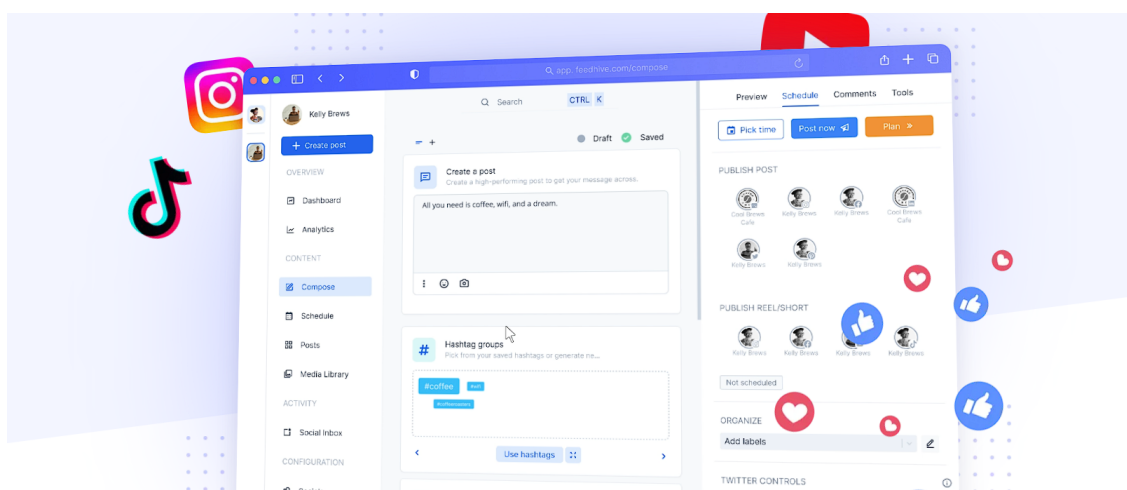


Рисунок 1.4 – Пример FeedHive

1.2.5 UpGrow

Описание: UpGrow(рисунок 1.5) является инструментом для роста в Instagram с помощью искусственного интеллекта[9].

Достоинства:

- привлечение реальных подписчиков;
- рекомендации по улучшению профиля;
- настраиваемый таргетинг с фильтрами по возрасту, полу, локации, интересам и хэштегам.

Недостатки:

- ориентированность только на Instagram;
- риск нарушения политики Instagram из-за автоматизированного привлечения подписчиков;
- стоимость подписки.

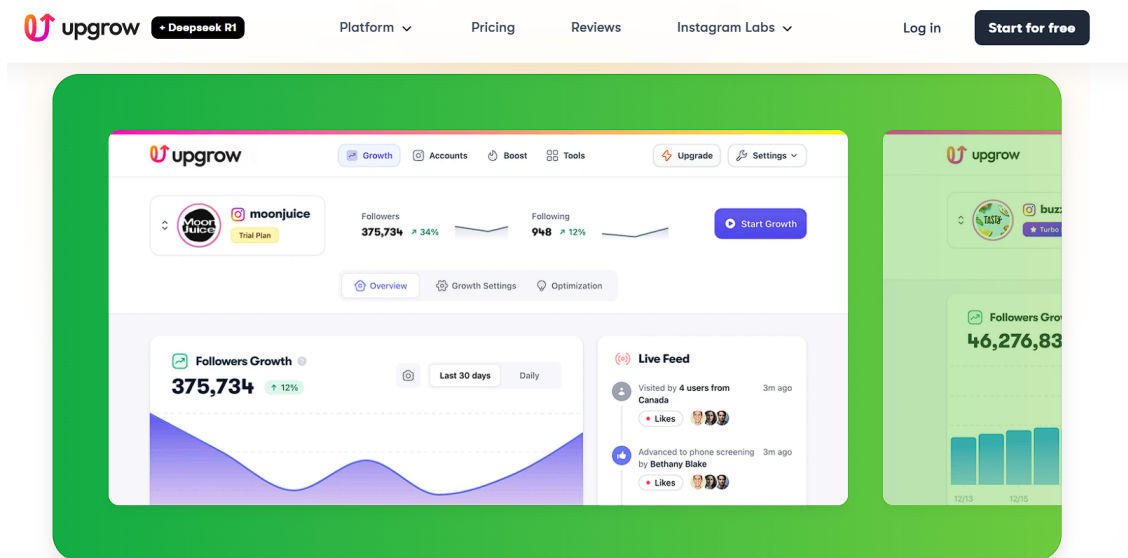


Рисунок 1.5 – Пример UpGrow

1.3 Подходы к решению задачи

Разработка интеллектуального ассистента для социальных сетей требует выбора эффективного подхода к обработке информации, анализу данных и взаимодействию с пользователем. Современные технологии позволяют создавать системы, способные не только генерировать ответы, но и анализировать тренды, извлекать актуальную информацию, давать персонализированные рекомендации и даже адаптироваться к стилю общения пользователя. В данном пункте рассматриваются ключевые подходы, используемые при создании интеллектуальных ассистентов, включая машинное обучение и дообучение, обработку событий и т.д.

1.3.1 Поточковая обработка событий и активных правил

Зачастую данные, получаемые из социальных сетей, генерируются в режиме реального времени, что затрудняет их обработку традиционными методами. Поточковая обработка данных позволяет проводить анализ и обработку данных по мере их возникновения, обнаруживать различные события в реальном масштабе времени. Механизм активных правил позволяет оперативно реагировать на те или иные события.

Сервис потоковой обработки событий получает данные, соответствующие элементарным событиям, от внешних наблюдаемых систем. На основе полученных данных с применением соответствующих алгоритмов обнаруживаются сложные, агрегированные и составные события. Механизм активных правил позволяет соответствующим образом реагировать на события и их контекст.

В сервисе потоковой обработки событий метаданные - это информация, описывающая свойства и характеристики событий или активных правил. Эта информация может включать такие сведения, как тип события, временная метка, источник или любая другая дополнительная информация, которая имеет отношение к обработке события или исполнения правила[10].

При потоковом получении событий основной набор метаданных приходит в связке с самим событием. Одним из способов использования метаданных в сервисе является исполнение активных правил. Активные правила - это предопределенные действия, которые являются реакцией на возникновение заданных событий. Активные правила реализованы на основе парадигмы ЕСА (event-condition-action): события (event), которое поступает от сторонних наблюдаемых систем; условия (condition), в зависимости от выполнения которого будет принято решение о запуске действия; действия (action), которое будет запущено, если событие соответствует условиям.

Использование потоковой обработки данных с активными правилами может значительно повысить скорость реагирования интеллектуального ассистента на события в социальных сетях, но требует внимательной настройки и учета особенностей работы с большими объемами информации и динамическими данными.

Преимущества:

- обработка данных в реальном времени;
- оперативная реакция на события;
- агрегирование и обнаружение сложных событий.

Недостатки:

- высокие требования к вычислительным ресурсам;
- трудности с обработкой неоднозначных данных.

1.3.2 Машинное обучение

Машинное обучение - это категория алгоритмов, которая позволяет программным приложениям быть более точными в прогнозировании результатов без явного программирования[11]. Основная предпосылка машинного обучения заключается в создании алгоритмов, которые могут принимать входные данные и использовать статистический анализ для прогнозирования выходных данных по мере появления новых данных.

Самой популярной задачей машинного обучения является классификация. Цель этого метода - классифицировать объекты по заранее известному признаку, например, документы по языкам, музыку по жанрам и т.д. Классическим примером классификации является алгоритм «Наивный Байес». Раньше он применялся в спам-фильтрах, сейчас этим занимаются деревья решений. Другой пример классификации - метод опорных векторов, он является самым популярным методом классификации.

Регрессия используется в задачах, когда надо предсказать результат. Регрессия задается в виде уравнения, которое описывает функцию, показывающую взаимосвязь между входными и выходными данными.

Классификация может помочь определить, к какой категории или типу относится тот или иной контент, а регрессия будет использоваться для предсказания того, как пользователи будут взаимодействовать с этим контентом. В комбинации эти методы могут улучшить персонализацию взаимодействий ассистента с пользователями, например, на основе классификации предпочтений пользователей и предсказания их дальнейших действий.

Преимущества:

- автоматизация и улучшение качества ответов;
- быстрая обработка больших объемов данных;
- предсказания трендов и аналитика.

Недостатки:

- необходимость в больших объемах данных для обучения;
- зависимость от качества данных;
- возможность переобучения.

1.3.3 Дообучение

Файнтюнинг (fine-tuning) — это метод дообучения, который позволяет адаптировать уже обученные модели для новых задач, избегая начала обучения с нуля. Обучение «с нуля» требует много данных и вычислительных мощностей, что делает его длительным и ресурсозатратным. Файнтюнинг, в отличие от этого, позволяет использовать уже обученные модели, что снижает затраты[12].

При подготовке модели важно правильно выбрать модель и настроить её параметры и гиперпараметры. Далее определяются цели и требования к производительности, после чего необходимо собрать релевантные данные и разделить их на обучающие и тестовые выборки. Настройка и запуск включают оптимизацию гиперпараметров и тренировку модели, затем происходит оценка результатов производительности и проводится итеративное дообучение для улучшения.

Файнтюнинг может значительно улучшить работу интеллектуального ассистента для социальных сетей, так как позволяет адаптировать предобученные языковые модели под конкретные задачи и специфику общения пользователей. Используя этот подход, можно дообучить модель на данных из различных социальных сетей, что поможет ей лучше понимать специфический контекст. Этот подход поможет повысить точность и релевантность ответов ассистента, так как модель будет лучше понимать контекст диалогов и выдавать более осмысленные и уместные ответы.

Преимущества:

- экономия ресурсов и времени;
- гибкость;
- высокая точность адаптации к конкретной задаче.

Недостатки:

- снижение эффективности при использовании плохо подготовленных данных;
- обязательна тщательная настройка гиперпараметров;
- риск переобучения.

1.3.4 Retrieval-Augmented Generation

Генерация с дополненной выборкой (RAG) – это процесс оптимизации выходных данных большой языковой модели, в котором перед получением ответа обращение идёт к надёжной базе знаний за пределами источников обучающих данных[13]. Большие языковые модели (LLM) обучаются на огромных объёмах данных и используют миллиарды параметров для получения оригинальных результатов для таких задач, как ответы на вопросы, перевод с различных языков и составление предложений. RAG расширяет и без того обширные возможности LLM в определённых предметных областях или внутренних базах знаний без необходимости переобучать модели.

При использовании метода RAG большая языковая модель дополняется механизмом поиска, который позволяет ей получать доступ к внешним источникам знаний. Этот механизм поиска позволяет модели извлекать соответствующую информацию из базы знаний или корпуса документов перед генерированием ответа. Метод RAG используется в задаче улучшения способности модели генерировать релевантные, информативные и фактически

точные ответы с помощью тех данных, на которых она не была обучена[14].

Использование RAG в интеллектуальном ассистенте для социальных сетей делает его более осведомлённым, гибким и полезным для пользователей. Это дает возможность ассистенту анализировать актуальные тренды, комментарии и события, а не просто выдавать статические ответы. Кроме того, этот метод помогает обходить ограничения языковых моделей, предоставляя точную и проверенную информацию в реальном времени.

Преимущества:

- обновляемость данных;
- гибкость;
- экономия ресурсов.

Недостатки:

- зависимость от качества базы данных;
- сложность работы с другими типами данных(изображения, видео, аудио).

1.4 Вывод

В ходе анализа предметной области социальных сетей для создания интеллектуального ассистента было выявлено, что существующие системы имеют свои преимущества и недостатки, однако не предоставляют в полной мере всю требуемую функциональность, из-за чего и существует необходимость в разработке собственного ассистента.

Наиболее подходящим под поставленную задачу был выбран RAG-подход, что позволит создать гибкую и масштабируемую систему, которая сможет эффективно взаимодействовать с внешними источниками данных и обеспечивать персонализированные рекомендации. Интеграция с существующими социальными платформами и ресурсами обеспечит доступ к обширному массиву информации, что позволит ассистенту выполнять анализ трендов, классификацию контента и обработку пользовательских запросов в режиме реального времени.

Выбранные методы и архитектурные решения создадут надежную основу для дальнейшего развития интеллектуального ассистента, обеспечивая его гибкость, масштабируемость и удобство использования. Это позволит системе адаптироваться к изменяющимся требованиям и расширять функциональность, улучшая взаимодействие с пользователями и повышая качество предоставляемых возможностей ассистента.

2 ПРОЕКТИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНОГО АССИСТЕНТА ПО СОЦИАЛЬНЫМ СЕТЯМ

2.1 Постановка задачи

Для работы над данным курсовым проектом были поставлены следующие задачи:

Сбор информации и анализ существующих решений. Необходимо исследовать существующие платформы с открытым API для определения потенциальных источников данных.

Выбор фреймворка для RAG-цепочки. Было проведено исследование современных методов построения интеллектуальных ассистентов для социальных сетей, включая RAG-подход

Формирование структуры системы. Это включает проектирование архитектуры интеллектуального ассистента, включая модули взаимодействия с API и базами знаний.

Разработка механизма обработки запросов. Создание логики анализа пользовательских вопросов, поиска релевантной информации и генерации ответов.

Оценка эффективности решения. Требуется тестирование работы системы на реальных данных, выявление сильных и слабых сторон, формирование выводов и рекомендаций.

Обеспечение доступности и обновляемости. Учет данных и использование механизмов для возможности регулярного обновления системы, добавления новой информации и корректировки существующих данных.

Соответствующие требования, предъявленные к разрабатываемому ассистенту:

- Полнота и актуальность. Предоставляемые данные должны быть максимально свежими и релевантными.
- Надежность и достоверность. Ассистент должен корректно обрабатывать запросы, минимизировать ошибки, а также проверять полученную информацию на соответствие реальности.
- Масштабируемость. Архитектура должна позволять расширение функционала, добавление новых источников данных и интеграцию с дополнительными сервисами.
- Безопасность. Ассистент должен соблюдать правила конфиденциальности, не раскрывать личные данные пользователей и защищать систему от несанкционированного доступа.

Проект создания ассистента по социальным сетям представляет собой комплексную задачу, требующую тщательной проработки структуры дан-

ных, разработки алгоритмов для анализа и обеспечения высоких стандартов надежности и доступности. Ключевая цель проекта заключается в создании инструмента, который сделает взаимодействие любого пользователя со своими аккаунтами в социальных сетях более интерактивным, ускоренным и содержательным.

2.2 Описание пользователей интеллектуального ассистента по социальным сетям

Интеллектуальный ассистент для социальных сетей предоставляет разнообразные возможности для различных категорий пользователей, включая SMM-специалистов, маркетологов, аналитиков, журналистов, исследователей, владельцев бизнесов, а также обычных пользователей, интересующихся трендами и актуальными обсуждениями. Каждой группе пользователей доступны специализированные функции, которые помогают им достигать своих целей, связанных с анализом данных, мониторингом информации и исследованием социальных процессов (рисунок 2.1).

SMM-специалисты и маркетологи могут использовать интеллектуального ассистента для анализа популярности контента, выявления трендов и мониторинга реакции аудитории на публикации. Система поможет им анализировать метрики вовлеченности, отслеживать ключевые темы обсуждений и оценивать эффективность рекламных кампаний. Также ассистент сможет предоставлять рекомендации по оптимальному времени публикации и формату контента для достижения наилучших результатов.

Аналитики и исследователи могут применять систему для глубинного анализа социальных процессов и пользовательских настроений. Ассистент способен обрабатывать большие объемы данных из социальных сетей, выявлять закономерности в поведении пользователей, анализировать изменения общественного мнения и предсказывать развитие трендов. Это особенно полезно для социологических и маркетинговых исследований.

Журналисты могут использовать ассистента для мониторинга новостей, поиска актуальных тем и анализа общественного мнения по резонансным событиям. Система поможет оперативно находить ключевые дискуссии, выявлять мнение лидеров общественного мнения и анализировать распространение информации.

Владельцы бизнеса и бренд-менеджеры могут применять систему для отслеживания репутации компании в социальных сетях, анализа отзывов клиентов и работы с обратной связью. Ассистент поможет выявлять негативные упоминания, анализировать потребности аудитории и находить точки роста для улучшения качества обслуживания и позиционирования бренда.

Обычные пользователи могут использовать интеллектуального assi-

стента для поиска интересных обсуждений, получения персонализированной ленты новостей и анализа своей активности в социальных сетях.

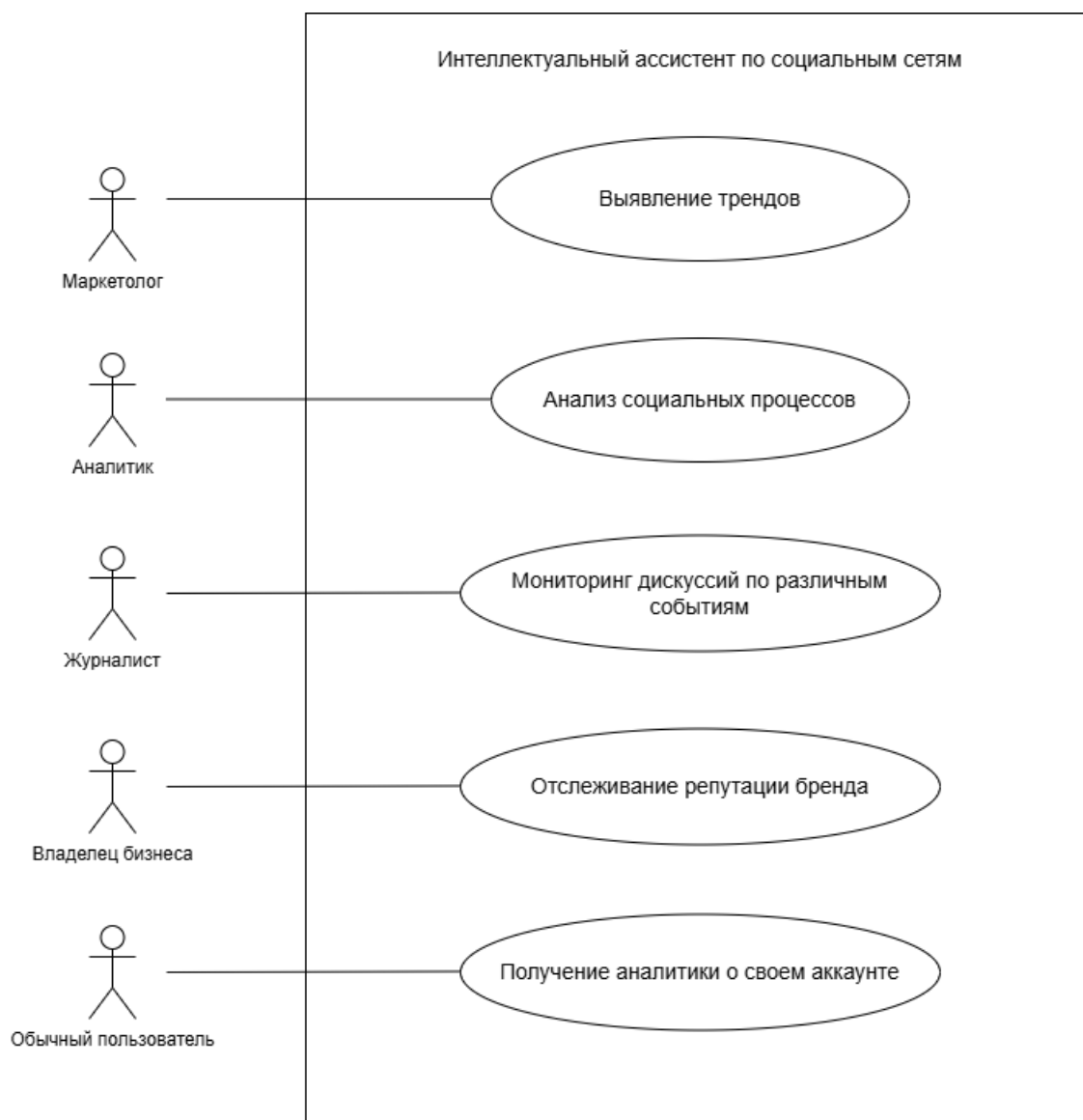


Рисунок 2.1 – Сценарии использования пользователями интеллектуального ассистента по социальным сетям

2.3 Архитектура системы

Архитектура системы играет ключевую роль в обеспечении стабильной работы ассистента и его способности масштабироваться под новые требования. В данном разделе представлена модульная структура системы, описаны взаимосвязи между компонентами системы и обоснованы ключевые архитектурные решения.

Так как интеллектуальный ассистент разрабатывается на основе метода RAG, архитектура ассистента включает несколько ключевых компонентов, взаимодействующих между собой. Процесс взаимодействия компонентов выглядит следующим образом: пользователь взаимодействует с

системой через интерфейс, отправляя запрос, который передается на сервер. Сервер выполняет первичную обработку запроса и отправляет его в классификатор, который определяет, какой метод обработки необходимо использовать. Классификатор возвращает серверу идентификатор функции, соответствующей данному запросу.

Если запрос требует получения данных из социальной сети, сервер вызывает соответствующую функцию, которая обращается к API нужной социальной сети. После успешного получения информации сервер формирует пром프트, который передается в языковую модель (LLM). LLM, используя предоставленный контекст, генерирует осмысленный ответ, который затем возвращается на сервер и передается в пользовательский интерфейс.

Кроме того, ассистент может отвечать на более общие вопросы о социальных сетях, не требующие обращения к API конкретной социальной сети. В таких случаях сервер выполняет поиск релевантных фрагментов в заранее собранном документе, разбитом на чанки по смыслу. Подходящие чанки используются для формирования промпта перед отправкой в LLM, что позволяет модели выдавать достоверные и точные ответы, основанные на предварительно подготовленной информации.

Для наглядного представления внутренней структуры и взаимосвязей компонентов ассистента по социальным сетям используется диаграмма последовательности (рисунок 2.2). Она позволяет проанализировать основные сущности системы, их свойства и методы, а также продемонстрировать взаимосвязи между объектами.

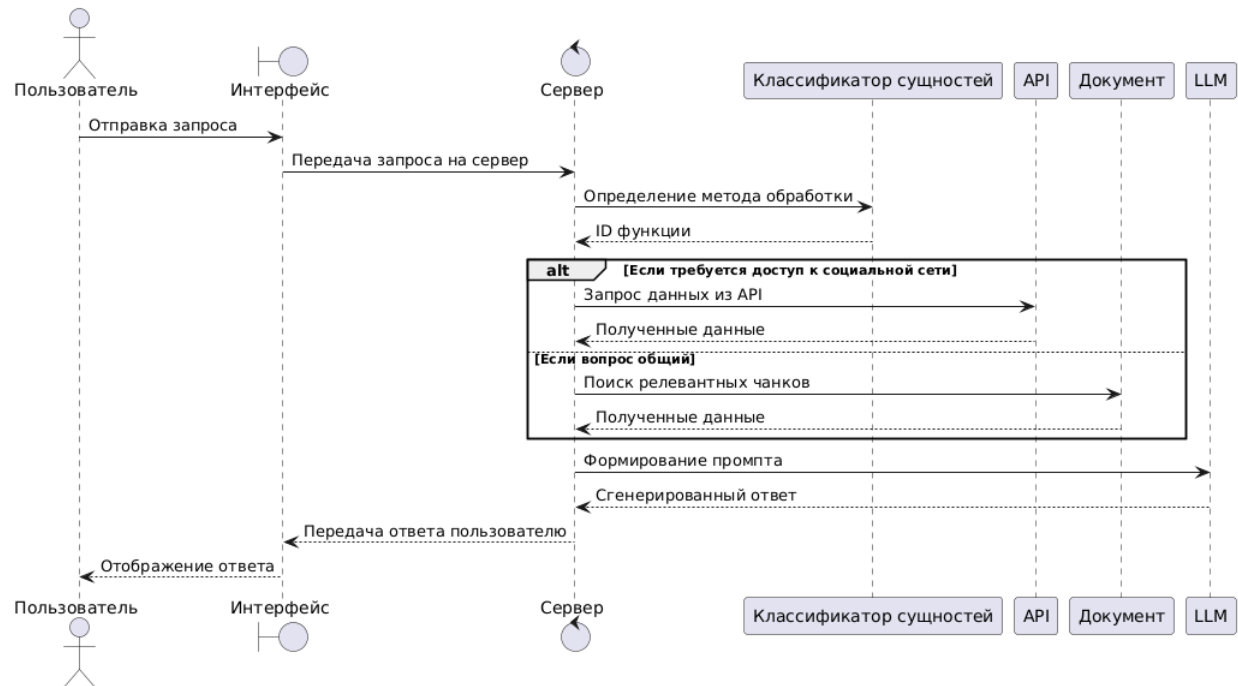


Рисунок 2.2 – Диаграмма последовательности для интеллектуального ассистента по социальным сетям

Такой подход обеспечивает гибкость системы, позволяя ассистенту как извлекать актуальные данные из социальной сети, так и отвечать на общие вопросы, используя структурированную базу знаний.

2.4 Алгоритмы

2.4.1 Алгоритм обработки пользовательского ввода

- пользователь отправляет сообщение через интерфейс;
- сообщение перехватывается и передаётся для обработки;
- полученный текстовый запрос отправляется классификатору сущностей для определения метода;
- классификатор возвращает ответ, который включает распознанные сущности;
- по полученным сущностям определяется соответствующий обработчик;
- если запрос относится к общим вопросам, выполняется поиск релевантных данных в подготовленном документе либо генерация ответа без дополнительного поиска;
- если запрос требует обращения к социальной сети, сервер вызывает API и получает нужные данные.

2.4.2 Алгоритм краткого пересказа сообщений в диалоге

- определяются параметры запроса, такие как идентификатор чата, количество сообщений и дополнительные настройки;
- устанавливаются данные аутентификации для доступа к API социальной сети;
- запрос на получение истории сообщений передается в API социальной сети;
- производится обработка запроса, если запрос выполнен успешно, полученные данные обрабатываются и преобразуются в удобный формат;
- извлеченные сообщения передаются в метод для обработки диалогов;
- анализируется структура диалога, выделяются ключевые моменты: имя пользователя и его сообщения;
- на основе обработанных сообщений создается текстовый запрос для языковой модели с просьбой пересказать диалог;
- итоговый запрос передается для дальнейшей обработки в LLM.

2.4.3 Алгоритм анализа постов, вызвавших наибольшую реакцию

- задаются параметры запроса: идентификатор страницы пользователя или сообщества, количество постов, фильтры и версия API;
- устанавливаются данные аутентификации для доступа к API социальной сети;
- выполняется запрос к API для получения списка постов на стене пользователя или сообщества;
- если запрос выполнен успешно, полученные данные о постах обрабатываются и извлекаются ключевые характеристики: число лайков, репостов, комментариев и просмотров;
- на основе извлеченных данных формируется текстовый запрос к языковой модели с формулировкой: "Какие посты вызвали наибольший отклик?";
- сформированный запрос передается в LLM для генерации итогового ответа.

2.4.4 Алгоритм анализа подписок для составления описания личности пользователя

- задаются параметры запроса: идентификатор пользователя, количество сообществ, дополнительные поля (описание, название, численность участников);
- устанавливаются данные аутентификации для доступа к API социальной сети;
- выполняется запрос к API для получения списка сообществ, на которые подписан пользователь;
- если запрос выполнен успешно, извлекаются сведения о сообществах: тематика, численность, описание и прочее;
- полученные данные передаются в метод обработки подписок;
- анализируются типы сообществ и их характеристики, формируются выводы о личных интересах пользователя;
- составляется текстовый запрос к языковой модели с формулировкой: "Опиши личность пользователя по его подпискам";
- сформированный запрос передается в LLM для генерации результата.

2.4.5 Алгоритм извлечения часто задаваемых вопросов под постом пользователя или сообщества

- задаются параметры запроса: идентификатор владельца и поста, количество комментариев, порядок сортировки и дополнительные настройки;

ки;

- устанавливаются данные аутентификации для доступа к API социальной сети;
- выполняется запрос к API для получения комментариев к конкретному посту;
- если запрос выполнен успешно, извлекаются тексты комментариев и дополнительная информация (лайки, авторы);
- комментарии передаются в метод обработки, который выделяет повторяющиеся или схожие вопросы;
- проводится группировка комментариев по смыслу для определения наиболее частых тем;
- составляется текстовый запрос к языковой модели с формулировкой: "Создай список часто задаваемых вопросов и предложи возможные ответы";
- итоговый запрос передаётся в LLM для генерации результата.

2.4.6 Алгоритм выявления тем, вызывающих яркие обсуждения в диалоге

- задаются параметры запроса: идентификатор чата, количество сообщений, дополнительные параметры;
- устанавливаются данные аутентификации для доступа к API социальной сети;
- выполняется запрос к API для получения истории переписки;
- при успешном выполнении запроса полученные сообщения обрабатываются и структурируются;
- извлечённые сообщения передаются в метод анализа диалогов;
- выделяются сообщения с наибольшей активностью, эмоциональной окраской или повторяющимися темами;
- составляется текстовый запрос к языковой модели с формулировкой: "Какие темы вызывают наиболее яркие обсуждения?";
- сформированный запрос передаётся в LLM для анализа и генерации ответа.

2.5 Вывод

В ходе проектирования интеллектуального ассистента по социальным сетям был проведен детальный анализ предметной области для определения сценариев использования на основании потребностей пользователей, а также поиск технологий, связанных с получением данных пользователя из конкретной социальной сети и их обработкой.

Кроме того, в рамках проектирования были уточнены основные задачи, решаемые системой, описаны потенциальные пользователи, а также рас-

смотрены типовые запросы и сценарии взаимодействия с системой. Особое внимание было уделено проектированию архитектуры системы, описанию её компонентов и определению ключевых требований к реализации.

Таким образом, была заложена основа для реализации ассистента, включающего функциональный интерфейс и гибкую архитектуру. Описанные решения обеспечивают перспективу создания эффективного инструмента, который сможет удовлетворить запросы пользователей и предоставить дополнительные возможности.

3 РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОГО АССИСТЕНТА ПО СОЦИАЛЬНЫМ СЕТЯМ

В данном разделе подробно рассмотрен процесс создания ассистента, следующий после проектирования и включающий в себя такие аспекты, как разработка и реализация основных компонентов. В разделе также указаны выбранные технические инструменты, способы разработки и некоторые детали реализации интерфейса.

3.1 Средства разработки

3.1.1 Python

Python является отличным выбором для разработки благодаря его гибкости и большому количеству инструментов для обработки текста и анализа данных. Одним из главных преимуществ Python является наличие мощных библиотек, таких как NLTK, spaCy и прочих, которые могут эффективно анализировать тональность комментариев, выявлять ключевые темы обсуждений и классифицировать сообщения по настроению или тематике. Благодаря интуитивно понятному синтаксису и быстрому прототипированию, Python значительно ускоряет разработку сложных алгоритмов для анализа контекста сообщений и иных объектов профиля[15].

3.1.2 C++

C++, с другой стороны, используется для разработки производительных компонентов ассистента, требующих высокой эффективности и работы с низкоуровневыми системами. C++ обеспечивает более высокую производительность и управление ресурсами, что делает его подходящим для создания ядра системы, ответственного за обработку больших объёмов данных или моделирование сложных взаимодействий в реальном времени. C++ особенно полезен, если нужно обеспечить быструю работу алгоритмов анализа, где требуется оптимизация по памяти и скорости выполнения[16].

3.1.3 VK API

VK API – это программный интерфейс ВКонтакте, который предоставляет разработчикам доступ к данным социальной сети, таким как посты, комментарии, профили пользователей, группы, сообщения и другая информация. Это мощный инструмент для анализа контента, мониторинга активности и создания интеллектуальных сервисов, включая чат-ботов и ассистентов.[17].

3.1.4 Langchain

Langchain – это фреймворк для построения цепочек обработки запросов с использованием больших языковых моделей. Он позволяет гибко интегрировать внешние источники данных, инструменты и модели, обеспечивая удобную реализацию систем Retrieval-Augmented Generation, агентов и интеллектуальных ассистентов. В проекте Langchain используется для организации взаимодействия между полученными из социальной сети данными и языковой моделью, формируя итоговые ответы на пользовательские запросы.[18]

3.1.5 Ollama

Ollama – локальный фреймворк для запуска больших языковых моделей на персональном компьютере. Он обеспечивает быстрый и автономный доступ к LLM без необходимости обращения к облачным API. В рамках проекта Ollama используется в связке с Langchain как модель генерации, отвечающая на обработанные запросы пользователя на основе контекста из социальной сети.[19]

3.1.6 YandexGPT

YandexGPT[20] — это большая языковая модель, разработанная Яндексом на основе трансформерной архитектуры. Она предназначена для генерации и обработки естественного языка и может применяться в задачах диалогов, суммаризации, анализа текста и других NLP-задач, в том числе для генерации ответов на основе данных из социальной сети, предоставляя высокое качество генерации в отдельных сценариях.

3.1.7 Github

Github[21] позволяет команде синхронизировать работу над проектом, устраняя риски конфликтов в коде. GitHub поддерживает коллективную работу через Pull Requests — участники команды могут предлагать новые функции или улучшения системы, которые будут рецензироваться перед добавлением в основную версию проекта. Каждая задача назначается отдельному участнику проекта, что позволяет отслеживать прогресс и координировать действия команды. Таким образом, GitHub обеспечивает прозрачный процесс обсуждения и улучшений.

3.2 Детали разработки

Для создания RAG-цепочки, обеспечивающей генерацию ответов на основе как внешнего источника знаний, так и истории запросов, предназна-

чен метод `get_rag_chain`(рисунок 3.1).

При вызове функции инициализируется большая языковая модель с использованием Ollama и создаётся ретривер, учитывающий контекст диалога. Затем формируется цепочка обработки документов на основе подхода "stuff"— все релевантные документы объединяются и передаются в модель вместе с вопросом. На финальном этапе создаётся основная RAG-цепочка, объединяющая ретривер и генеративную модель в единый рабочий процесс, после чего она возвращается из функции.

```
def get_rag_chain(model="denisavetisyan/saiga_yandexgpt_8b_gguf_q5_k_m:latest"):
    llm = Ollama(model=model)
    history_aware_retriever = create_history_aware_retriever(llm, retriever, contextualize_q_prompt)
    question_answer_chain = create_stuff_documents_chain(llm, qa_prompt)
    rag_chain = create_retrieval_chain(history_aware_retriever, question_answer_chain)
    return rag_chain
```

Рисунок 3.1 – Метод `get_rag_chain`

Метод `get_vk_chat_history` предназначен для получения истории сообщений из определённого чата во «ВКонтакте» (рисунок 3.2). В начале формируется URL-адрес и параметры строки запроса, включая версию API и идентификатор клиента. Далее подготавливаются данные для запроса, в которых указываются такие параметры, как идентификатор чата (`peer_id`), количество сообщений (`count`), а также параметры расширенного ответа и список возвращаемых полей пользователей. В запрос также включается токен доступа пользователя (`access_token`) для авторизации.

После этого формируются заголовки запроса, чтобы имитировать поведение браузера и корректно взаимодействовать с API, и выполняется POST-запрос к методу `messages.getHistory` API VK с использованием библиотеки `requests`. Если запрос выполняется успешно, результат преобразуется в формат JSON и возвращается. В случае возникновения ошибки она перехватывается и выводится сообщение об ошибке, после чего функция возвращает `None`.

Схожим образом написаны и другие методы для обработки данных из API VK, в том числе методы для получения информации о подписках пользователя, постах на странице пользователя или сообщества и т.д. Для каждого метода различаются параметры, которые передаются в POST-запросе, а также отличаются получаемые данные в JSON-формате.

```

# Метод для пересказа диалога
def get_vk_chat_history(peer_id, access_token):
    # URL и параметры строки запроса
    url = "https://api.vk.com/method/messages.getHistory"
    query_params = {
        "v": "5.246",
        "client_id": "6287487"
    }

    # Данные для тела запроса
    form_data = {
        "peer_id": peer_id, # Идентификатор чата
        "start_cmuid": 339782,
        "count": 32,
        "offset": -1,
        "extended": 1,
        "group_id": 0,
        "fwd_extended": 1,
        "fields": "id,first_name,last_name",
        "access_token": access_token # Токен доступа
    }

```

Рисунок 3.2 – Метод `get_vk_chat_history`

Метод `vkchat` (рисунок 3.3) — это обработчик Telegram-бота, который позволяет пользователю запросить анализ переписки из ВКонтакте. Пользователь вводит команду с инструкцией (например, "перескажи диалог"), после чего бот загружает историю сообщений через VK API, обрабатывает их, формирует запрос и передаёт его в RAG-цепочку на базе LLM. Ответ модели возвращается пользователю в чат. В случае ошибки бот уведомляет об этом.

```

async def vkchat(update: Update, context: CallbackContext):
    user_prompt = " ".join(context.args).strip()

    if not user_prompt:
        await update.message.reply_text(" ! Пожалуйста, укажи, что нужно сделать с перепиской.\n\nПример:\n`/vkchat обобщи переписку`", parse_mode="MarkdownV2")
        return

    try:
        await update.message.reply_text(" 📥 Загружаю переписку из ВКонтакте...")

        vk_data = get_vk_chat_history(VK_PEER_ID, VK_ACCESS_TOKEN)
        if not vk_data:
            await update.message.reply_text(" ❌ Не удалось получить переписку из VK.")
            return

        convo = parse_vk_messages(vk_data)
        prompt = conversation_to_prompt(convo, user_prompt)

        response = rag_chain.invoke({"input": prompt, "chat_history": []})
        await update.message.reply_text(f" 📤 Ответ: \n{response['answer']}")
    except Exception as e:
        await update.message.reply_text(f" ❌ Ошибка при обработке: {str(e)}")

```

Рисунок 3.3 – Метод `vkchat`

3.3 Личный вклад

В рамках разработки ассистента была выполнена интеграция с VK API и проведена настройка доступа, включая получение и конфигурацию токена, реализацию запросов к необходимым методам API, а также обработку возвращаемых данных. Особое внимание уделено корректному извлечению информации из личных сообщений, постов и связанных с ними метаданных. Полученные данные были структурированы в формат, пригодный для последующей обработки в RAG-цепочке. Это обеспечило надёжный канал получения актуальной информации из социальной сети ВКонтакте и её включение в логику работы интеллектуального ассистента.

3.4 Демонстрация основного функционала системы

Для демонстрации работы интеллектуального ассистента был создан Telegram-бот, выступающий в роли основного пользовательского интерфейса системы. Через него осуществляется отправка текстовых команд, которые обрабатываются цепочкой RAG с использованием API социальной сети ВКонтакте и языковой модели YandexGPT.

Рисунок 3.4 демонстрирует способность ассистента пересказать сообщения из указанного пользователем диалога. Полученные от VK API данные преобразуются в удобный формат и отправляются LLM с запросом «Перескажи данный диалог», после чего ответ приходит пользователю в Telegram-бот.

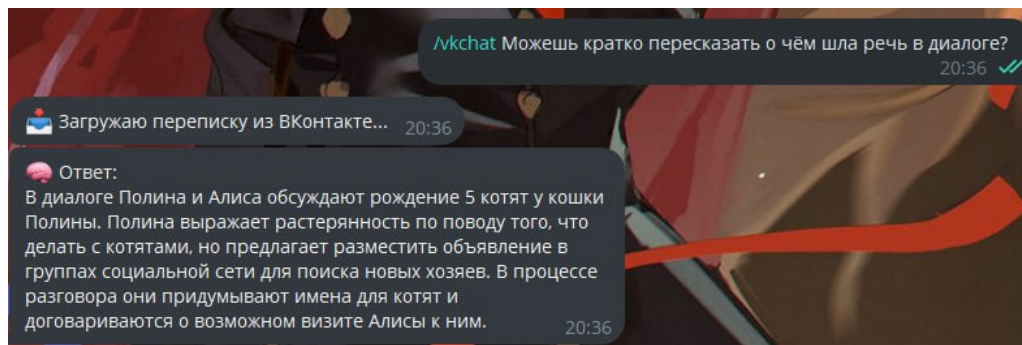


Рисунок 3.4 – Результат пересказа сообщений из чата

Рисунок 3.5 демонстрирует способность ассистента находить на странице указанного пользователя или сообщества самые популярные посты. Полученные от VK API данные преобразуются в удобный формат и отправляются LLM с запросом «Какие посты вызвали наибольшую реакцию?», после чего ответ приходит пользователю в Telegram-бот.

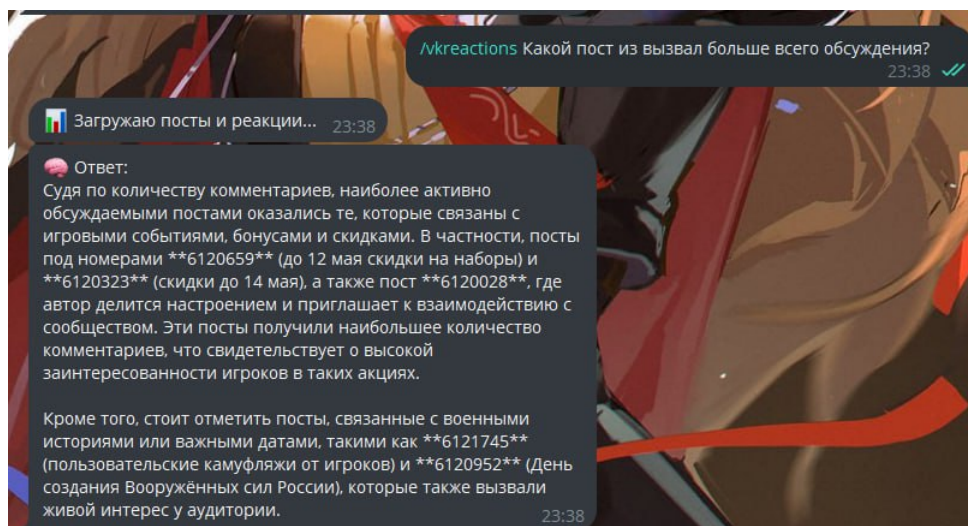


Рисунок 3.5 – Результат получения наиболее популярных постов

Рисунок 3.6 демонстрирует способность ассистента анализировать подписки указанного пользователя на сообщества и, исходя из этого, составлять примерное описание его личности и интересов. Полученные от VK API данные преобразуются в удобный формат и отправляются LLM с запросом «Опиши личность пользователя по его подпискам на вышеперечисленные сообщества», после чего ответ приходит пользователю в Telegram-бот.

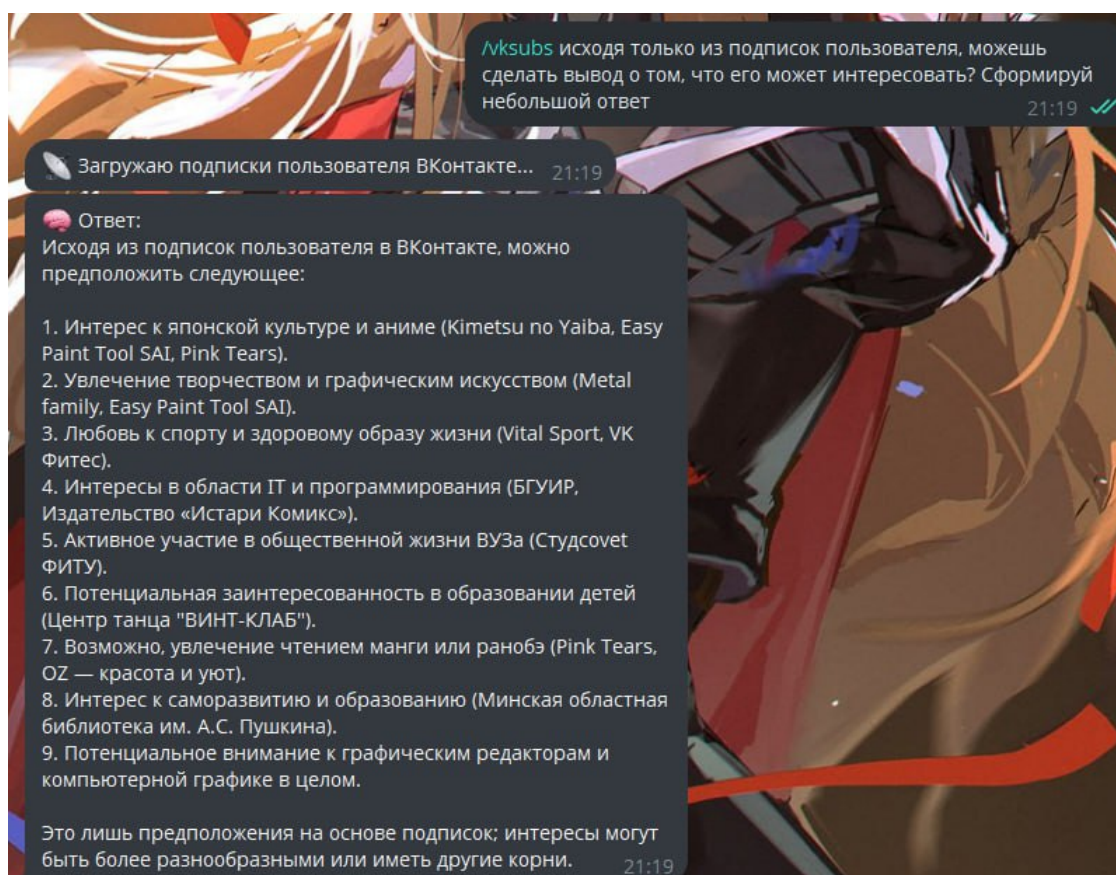


Рисунок 3.6 – Результат описания личности пользователя по его подпискам

Рисунок 3.7 демонстрирует способность ассистента анализировать комментарии под указанным постом на странице пользователя или сообщества, чтобы составить список часто задаваемых вопросов и предложить на них ответы. Полученные от VK API данные преобразуются в удобный формат и отправляются LLM с запросом «Создай список часто задаваемых вопросов в комментариях и предложи возможные ответы», после чего ответ приходит пользователю в Telegram-бот.

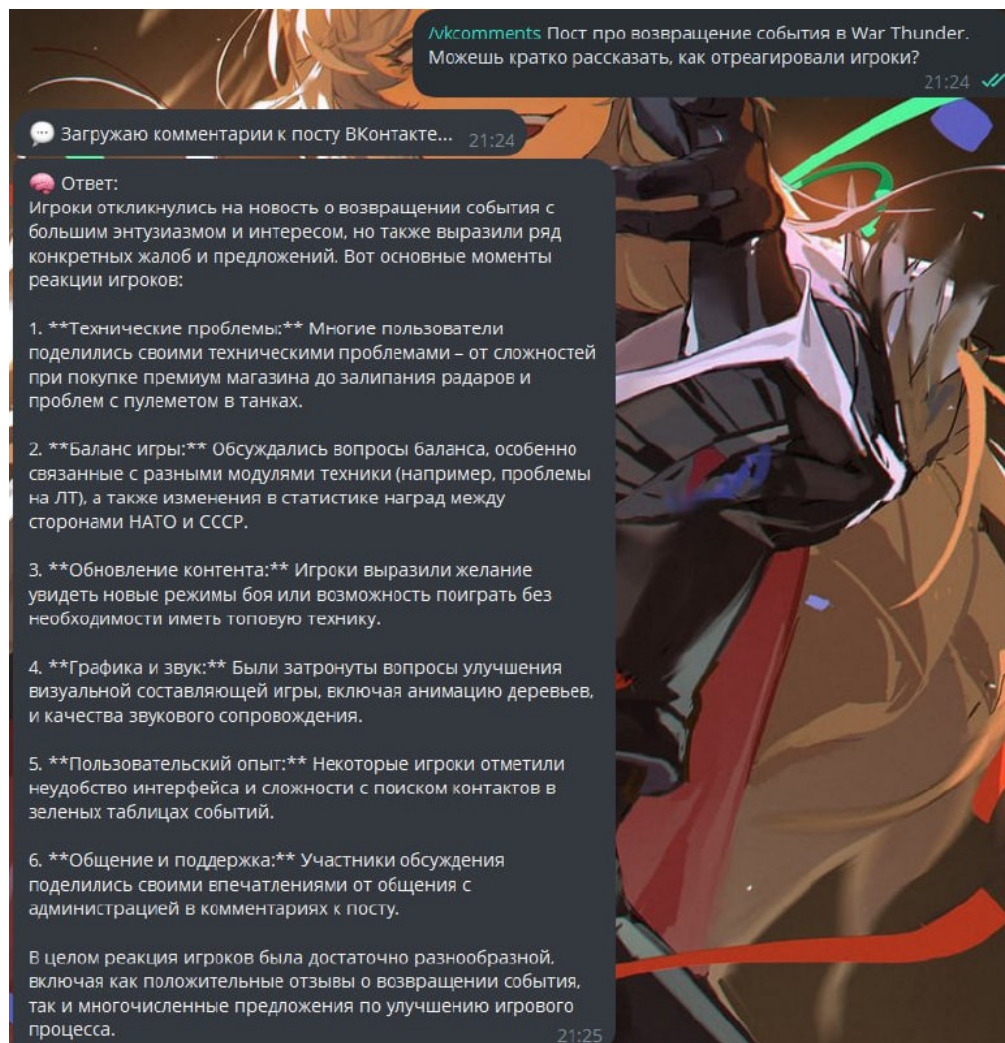


Рисунок 3.7 – Результат получения часто задаваемых вопросов под конкретным постом

Рисунок 3.8 демонстрирует способность ассистента отвечать на общие вопросы, не связанные с конкретным пользователем. Сведения о социальных сетях, их истории развития, статистике использования и прочей информации были представлены в виде отдельных документов, погруженных в большую языковую модель.

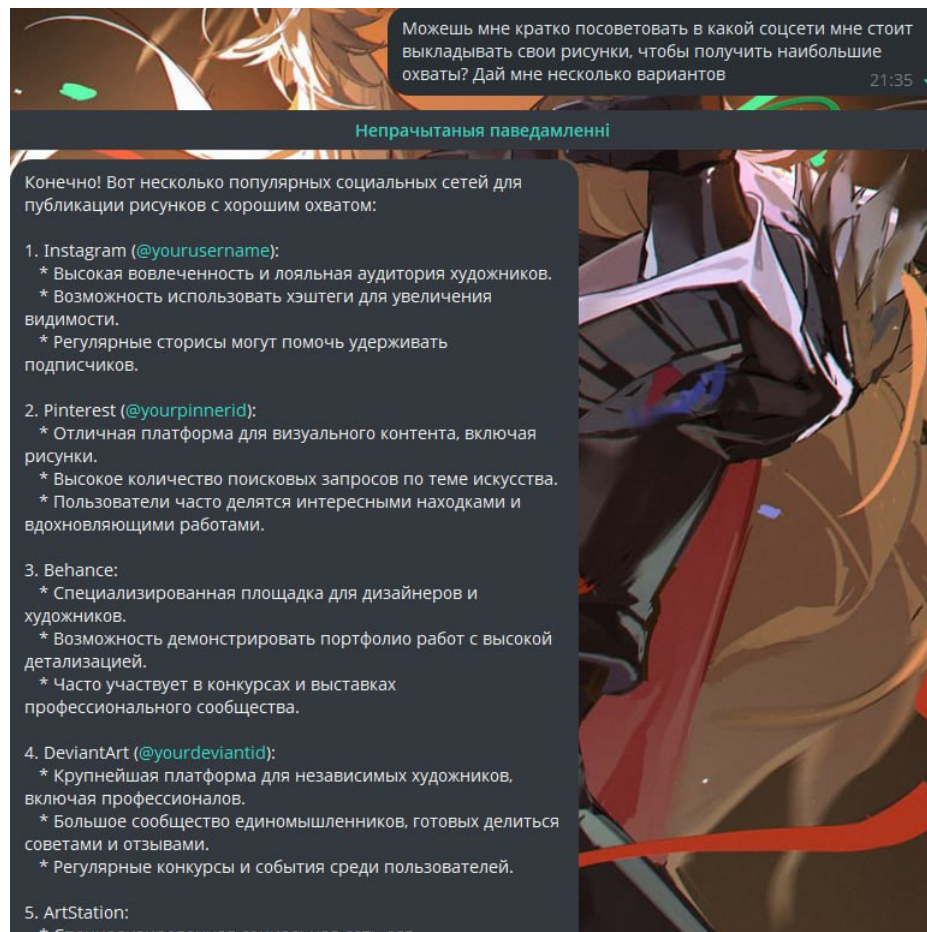


Рисунок 3.8 – Результат получения ответа на общий вопрос

Рисунок 3.9 иллюстрирует, как бот справляется с неоднозначными запросами, на которых не было найдено ответа. Ассистент даёт пользователю более точно выразить свои потребности, чтобы подобрать наиболее подходящий ответ на его запрос.

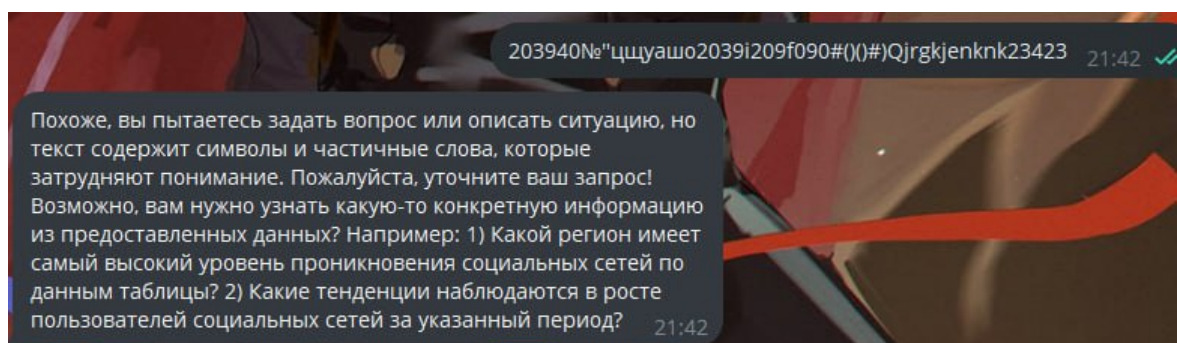


Рисунок 3.9 – Результат работы при некорректном запросе пользователя

3.5 Вывод

В результате курсового проектирования был реализован интеллектуальный ассистент для работы с данными из социальных сетей, основанный на подходе Retrieval-Augmented Generation и интегрированный с

мессенджером Telegram. Разработка охватила все этапы, необходимые для полноценного функционирования системы, а именно:

- реализована цепочка RAG с использованием фреймворка Langchain и моделей Ollama и YandexGPT для генерации ответов на основе внешних источников;
- обеспечено подключение к VK API для получения данных из диалогов, постов и комментариев;
- разработаны методы анализа и обработки контента из социальных сетей;
- все компоненты системы интегрированы в Telegram-бота, предоставляющего пользователю удобный интерфейс взаимодействия с ассистентом.

Таким образом, был создан функциональный ассистент, способный эффективно обрабатывать запросы пользователей и предоставлять информативные ответы на основе данных из социальных сетей.

ЗАКЛЮЧЕНИЕ

В рамках данного курсового проекта был разработан интеллектуальный ассистент, предназначенный для обработки и анализа информации из социальных сетей, с последующей генерацией ответов на запросы пользователей. Ассистент реализован с использованием архитектуры Retrieval-Augmented Generation, что позволило эффективно объединить возможности извлечения фактических данных и генеративных моделей.

В процессе проектирования и реализации системы были решены ключевые задачи, связанные с получением и обработкой информации из API социальной сети ВКонтакте, включая сообщения, посты и комментарии, разработаны методы преобразования полученных данных в формат, пригодный для передачи в языковую модель. Для генерации ответов использовались модели Ollama и YandexGPT, интегрированные в цепочку обработки с помощью фреймворка Langchain.

Особое внимание было уделено созданию модульной архитектуры, обеспечивающей гибкость и расширяемость системы. Важным элементом проекта стало подключение Telegram-бота, через которого осуществляется взаимодействие с пользователем. Таким образом, ассистент стал доступен в привычной среде общения и получил удобный интерфейс для отправки запросов и получения ответов.

На каждом этапе разработки проводилось тестирование корректности работы реализованных функций, что обеспечило надёжность и устойчивость системы. Все компоненты были интегрированы в единую цепочку, позволяющую обрабатывать пользовательские запросы от получения информации до генерации финального ответа.

Ожидаемый эффект от созданной системы заключается в автоматизации анализа контента социальных сетей, что может быть полезно как в личных целях, так и в профессиональной деятельности для мониторинга активности, выявления ключевых тем и реакции аудитории.

В дальнейшем возможны следующие направления развития:

- расширение функциональности ассистента за счёт поддержки других социальных платформ;
- внедрение более продвинутых алгоритмов анализа контента и сценариев использования ассистента;
- улучшение качества генерации ответов за счёт использования более мощных моделей или дополнительного обучения;
- интеграция визуальных компонентов и средств визуализации данных для более наглядного анализа;
- разработка собственного графического интерфейса и отдельного приложения.

Таким образом, созданный интеллектуальный ассистент представляет

собой эффективный инструмент для обработки пользовательских данных из социальных сетей, демонстрируя практическое применение современных технологий генерации текста и анализа информации в рамках реальных задач.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Роль социальных сетей в жизни современного человека [Электронный ресурс]. — 2022. — Режим доступа: <https://dipmall.ru/stati/rol-sotsialnykh-setej-v-zhizni-sovremennogo-cheloveka>. — Дата доступа: 3.03.2025.

[2] Коваленко, Г.А. Социальная сеть всемирной паутины Интернет как потенциальная модель обучения / Г.А Коваленко, Г.А. Хаертдинова // Современные проблемы науки и образования. — 2018. — май. — Режим доступа: <https://science-education.ru/ru/article/view?id=13229>. — Дата доступа: 5.03.2025.

[3] Горбатов, А.В. Социальные сети [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/sotsialnye-seti-3> — Дата доступа: 5.03.2025.

[4] Новокшинова, П.Н. «Информационный пузырь» и медиапотребление [Электронный ресурс]. — 2021. — Режим доступа: <https://cyberleninka.ru/article/n/informatsionnyy-puzyr-i-mediapotreblenie> — Дата доступа: 5.03.2025.

[5] Jasper AI for social media [Электронный ресурс]. — Режим доступа: <https://www.jasper.ai/use-cases/social-media-marketing?fpr=uniteai> — Дата доступа: 5.03.2025.

[6] BrandBastion Lite [Электронный ресурс]. — Режим доступа: <https://www.brandbastion.com/p/platform?ref=ailibricom> — Дата доступа: 6.03.2025.

[7] Brand24 [Электронный ресурс]. — Режим доступа: <https://brand24.com/> — Дата доступа: 6.03.2025.

[8] FeedHive [Электронный ресурс]. — Режим доступа: <https://www.feedhive.com/> — Дата доступа: 6.03.2025.

[9] UpGrow Platform [Электронный ресурс]. — Режим доступа: <https://www.upgrow.com/instagram-growth-service> — Дата доступа: 6.03.2025.

[10] Шибанов, С.В. Архитектура метаданных сервиса потоковой обработки событий и исполнения активных правил / С.В. Шибанов, А.С. Гусаров // Вестник Пензенского государственного университета. — 2023. — август. — Режим доступа: <https://cyberleninka.ru/article/n/arhitektura-metadannyh-servisa-potokovoy-obrabotki-sobytiy-i-ispolneniya-aktivnyh-pravil> — Дата доступа: 6.03.2025.

- [11] Акжолов, Р.К. Машинное обучение / Р.К. Акжолов // Вестник науки. — 2019. — декабрь. — Режим доступа: <https://cyberleninka.ru/article/n/mashinnoe-obuchenie-1>. — Дата доступа: 6.03.2025.
- [12] Что такое файттюнинг (fine-tuning)? [Электронный ресурс]. — Режим доступа: <https://dzen.ru/a/Z5961vVWjEdCyzNG> — Дата доступа: 6.03.2025.
- [13] Что такое RAG (генерация с дополненной выборкой)? [Электронный ресурс]. — Режим доступа: <https://aws.amazon.com/ru/what-is/retrieval-augmented-generation/> — Дата доступа: 6.03.2025.
- [14] Оболенский, Д.М. Использование метода RAG и больших языковых моделей в интеллектуальных образовательных экосистемах [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/ispolzovanie-metoda-rag-i-bolshih-yazykovyh-modeley-v-intellektualnyh-obrazovatelnyh-ekosistemah> — Дата доступа: 6.03.2025.
- [15] Python Documentation [Электронный ресурс]. — Режим доступа: <https://www.python.org/doc/> — Дата доступа: 7.03.2025.
- [16] C++ Documentation [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/en-us/cpp/cpp/?view=msvc-170> — Дата доступа: 10.10.2024.
- [17] VK API для разработчиков [Электронный ресурс]. — Режим доступа: <https://dev.vk.com/ru/method> — Дата доступа: 7.03.2025.
- [18] Langchain Documentation [Электронный ресурс]. — Режим доступа: <https://github.com/langchain-ai/langchain> — Дата доступа: 11.04.2025.
- [19] Ollama Documentation [Электронный ресурс]. — Режим доступа: <https://github.com/ollama/ollama> — Дата доступа: 11.04.2025.
- [20] Нейросети Яндекса [Электронный ресурс]. — Режим доступа: <https://ya.ru/ai/index> — Дата доступа: 18.04.2025.
- [21] Документация по GitHub [Электронный ресурс]. — Режим доступа: <https://docs.github.com/ru> — Дата доступа: 7.03.2025.