Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

КУРСОВАЯ РАБОТА
по дисциплине «Основы систем искусственного интеллекта»

Тема Определение акторов, действия и временные характеристики их осуществления в текстах с использованием возможностей больших языковых моделей

Студент Батуев А.Г.

Группа ИУ7-46Б

Преподаватели Строганов Ю.В.

Москва, 2025

Содержание

ВВЕДЕНИЕ

Современный этап цифровой трансформации характеризуется экспоненциальным ростом объёмов неструктурированных текстовых данных — от новостных лент и юридических документов до пользовательских отзывов и публикаций в социальных медиа [1]. Такой информационный поток создаёт критическую потребность в инструментах автоматического анализа, способных идентифицировать ключевые сущности (акторы), действия и временные метки. Актуальность разработки программных решений на основе искусственного интеллекта (ИИ) обусловлена их возможностью трансформировать рутинные процессы обработки данных в масштабируемые и интеллектуальные системы, отвечающие современным требованиям бизнеса и государственных структур.

Во-первых, автоматизация извлечения информации стала неотъемлемым элементом цифровой экономики. В сфере медиаанализа системы на базе ИИ позволяют в режиме реального времени отслеживать нарративы и выявлять тренды общественного мнения. В области кибербезопасности алгоритмы распознавания подозрительных активностей в текстовых логах способствуют предотвращению атак ещё до их реализации. Юридические компании активно применяют подобные технологии для анализа тысяч судебных прецедентов, что позволяет сократить время подготовки к делам на 40-60% [2].

Во-вторых, объёмы текстовых данных ежегодно увеличиваются на 55-60%, а их ручная обработка становится экономически нецелесообразной. Например, крупные корпорации могут тратить до 30% рабочего времени сотрудников на поиск и структурирование информации в документах [3]. Эти тенденции подчёркивают необходимость внедрения технологий машинного обучения (МО), способных обрабатывать петабайты данных с минимальными затратами человеческих ресурсов.

Целью данной курсовой работы является разработка программного решения на основе искусственного интеллекта, способного эффективно определять акторов, действия и временные характеристики их осуществления в текстах с использованием возможностей больших языковых моделей (LLM). Для достижения этой цели предполагается решить следующие задачи:

1) провести анализ существующих методов автоматического извлечения информации из текстов;

2) изучить возможности и ограничения больших языковых моделей в контексте извлечения семантических компонентов текстовых данных;

3) разработать прототип программного решения, реализующего алгоритмы идентификации акторов, действий и временных меток на основе выбранной модели LLM;

4) провести экспериментальную оценку разработанного решения на различных наборах данных для определения его точности и эффективности;

5) подвести итоги исследования и сформулировать рекомендации по дальнейшему совершенствованию методов.

1 Аналитическая часть

1.1 Введение в большие языковые модели

Большие языковые модели (Large Language Models, LLM) представляют собой один из наиболее значимых прорывов в области обработки естественного языка (Natural Language Processing, NLP) за последние годы. Они используются для решения широкого спектра задач: от генерации текстов и перевода до извлечения информации и анализа тональности. В этой части мы подробно рассмотрим, что такое LLM, как они работают и какие модели получили наибольшее признание в научном и практическом сообществе.

1.1.1 Определение LLM

Большие языковые модели — это нейронные сети, обученные на огромных объёмах текстовой информации, способные понимать, генерировать и обрабатывать естественный язык. Ключевой особенностью этих моделей является их масштаб: количество параметров (веса, которые настраиваются во время обучения) может достигать сотен миллиардов, что позволяет моделям захватывать сложные языковые закономерности.

Формально, можно представить LLM как функцию:

$$f_\theta : X \to Y,$$

где X – входная последовательность текста (например, набор слов или токенов), Y – выход модели, который может представлять собой продолжение текста, предсказание следующего слова или метки для классификации, θ – вектор параметров модели, оптимизируемый в процессе обучения.

Одной из самых популярных архитектур, лежащих в основе современных LLM, является трансформер. Архитектура трансформера позволяет моделям обрабатывать все входные данные параллельно, что существенно ускоряет обучение и повышает качество обработки длинных последовательностей.

Ключевые компоненты архитектуры трансформера включают:

— механизм внимания (attention): позволяет модели взвешивать вклад каждого элемента входной последовательности при формировании выходного представления. Простейший вариант – механизм самовнимания (self-attention), где каждый токен оценивается относительно всех остальных токенов;

— многоголовочное внимание (Multi-head Attention): позволяет модели одновременно фокусироваться на различных аспектах информации, что улучшает её способность учитывать сложные зависимости между токенами;

— нормализация и позиционные кодировки: поскольку трансформеры не обладают врождённой способностью учитывать порядок слов, вводятся специальные позиционные

кодировки, которые позволяют модели учитывать последовательность входных данных.

1.2 Принципы работы LLM

Чтобы понять, как работают большие языковые модели, важно ознакомиться с несколькими базовыми понятиями.

1.2.1 Токенизация

Токенизация – это процесс разбиения входного текста на более мелкие единицы, называемые токенами. Токен может быть словом, частью слова или даже символом. Например, фраза «Программное обеспечение» может быть разбита на два токена «Программное» и «обеспечение», либо на более мелкие единицы (например, корни, приставки, суффиксы) в зависимости от выбранного метода токенизации.

1.2.2 Эмбеддинги

После токенизации каждому токену присваивается эмбеддинг — вектор фиксированной размерности, который численно представляет смысл и контекст токена. Эмбеддинги позволяют модели работать с текстовыми данными, преобразовывая их в числовую форму, пригодную для дальнейшей обработки нейронной сетью. Обычно эмбеддинги обучаются одновременно с остальными параметрами модели или инициализируются с помощью предварительно обученных векторных представлений (например, Word2Vec [4]).

1.2.3 Механизм внимания и векторы контекста

Основой работы трансформеров является механизм внимания. Его суть заключается в том, что при обработке каждого токена модель оценивает, насколько важен каждый другой токен в последовательности для определения его значения. Это позволяет учитывать долгосрочные зависимости, что особенно важно для понимания сложных синтаксических конструкций и контекстуальных взаимосвязей.

Процесс самовнимания можно формализовать следующим образом. Пусть Q (query), K (key) и V (value) – это матрицы, полученные из эмбеддингов токенов посредством линейных преобразований. Тогда механизм внимания рассчитывается по формуле:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

где T - операция транспонирования, d_k – размерность векторов ключей, softmax обеспечивает нормировку весов внимания, так что сумма их значений равна 1 [5].

Таким образом, каждая позиция в последовательности получает взвешенное представление, учитывающее информацию со всех остальных позиций, что и формирует вектор контекста.

1.2.4 Генерация текста и предсказание следующего слова

Одна из основных задач больших языковых моделей (LLM) – предсказание следующего слова в последовательности. Для этого модель сначала вычисляет логиты — необработанные оценки для каждого слова из словаря, а затем преобразует их в вероятностное распределение, используя функцию softmax. Это позволяет определить вероятность появления каждого слова в качестве следующего. Функция softmax записывается следующим образом:

$$softmax(z)_i = \frac{\exp(\mathbf{z}_i)}{\sum_j \exp(\mathbf{z}_i)},$$

где z_i – логит (необработанная оценка) для i-го слова в словаре, $\exp(z_i)$ – экспоненциальное преобразование логита, которое гарантирует положительные значения, $\sum_j \exp(\mathbf{z}_i)$ – сумма экспоненциальных значений всех логитов, служащая для нормализации, так что сумма вероятностей по всем словам равна 1 [6].

1.2.5 Prompt Engineering

Prompt engineering — это искусство создания эффективных входных запросов (промтов) для LLM, чтобы направить их на выполнение конкретных задач. Промты могут включать инструкции, примеры, а также специфическую терминологию, которая помогает модели сконцентрироваться на нужном аспекте задачи.

1.3 Методы и алгоритмы извлечения информации

Извлечение информации (Information Extraction, IE) из текстовых данных представляет собой процесс автоматизированного структурирования неформатированного текста путём выделения значимых элементов. К таким элементам относятся именованные сущности (например, персоналии, организации, локации), связи между ними, события и временные характеристики.

1.3.1 Обзор задач извлечения информации

Ключевые задачи IE ориентированы на преобразование текста в структурированные форматы, что критически важно для последующего анализа. Основные направления включают:

— распознавание именованных сущностей (Named Entity Recognition, NER): идентификация и классификация объектов текста по предопределённым категориям. Например, в предложении «Компания Tesla начала поставки электромобилей в Европу в марте 2023 года» система NER выделит «Tesla» (организация) и «Европу» (географический объект). Точность NER влияет на качество последующих этапов;

— извлечение отношений (Relation Extraction, RE): определяется семантическая связь между сущностями. Например, между «Tesla» и «Европой» устанавливается отношение, указывающее на направление поставок;

— извлечение временной информации (Temporal Information Extraction, TIE): распознаются временные выражения и приводятся к конкретному формату. Здесь временной указатель «в марте 2023 года» позволяет определить дату события.

— извлечение событий (Event Extraction, EE): выявляется событие и его компоненты. Из этого предложения можно извлечь событие начала поставок, где субъектом выступает «Компания Tesla», действием — «начала поставки», а объектом — «электромобили», с указанием времени «в марте 2023 года»;

1.3.2 NER модели

Модели распознавания именованных сущностей (NER) являются одним из центральных компонентов систем извлечения информации. Их основная задача — автоматическая идентификация и классификация объектов текста (персоны, организации, локации и т.д.) по заранее определённым категориям. Существуют различные подходы к построению NER-моделей: основанные на правилах, использующие лингвистические правила и словари; основанные на машинном обучении (включая классические методы, такие как скрытые марковские модели и условные случайные поля, а также нейросетевые модели); и гибридные, комбинирующие оба подхода.

Среди нейросетевых моделей, демонстрирующих наиболее высокие результаты, выделяют рекуррентные нейронные сети (RNN, в частности, LSTM и GRU), сверточные нейронные сети (CNN) и трансформеры (BERT, RoBERTa, XLNet, ELECTRA и др.). Модели на основе правил отличаются высокой точностью при условии качественной проработки правил, но ограничены в обработке неоднозначности и новых слов. Модели машинного обучения, особенно нейросетевые, лучше справляются с неоднозначностью, способны к автоматическому извлечению признаков и обучению на больших объемах данных, однако требуют значительных вычислительных ресурсов и, как правило, менее интерпретируемы. Гибридные модели позволяют использовать преимущества обоих подходов.

1.3.3 Современные подходы на основе LLM

Большие языковые модели, предобученные на корпусах в масштабе терабайтов, кардинально изменили подходы к IE. Их ключевое преимущество — способность к контекстуальному пониманию, что критично для многозначных слов и имплицитных связей.

Стратегии применения LLM

— Zero-shot вывод: модель выполняет задачу, используя только текстовое описание в промпте (например, «Извлеки все организации из текста»). Это удобно для быстрого прототипирования, но точность зависит от способности модели декодировать неявные инструкции. Например, LLM может спутать формат даты «05/06/2023» (5 июня на 6 мая)

без явных указаний.

— Few-shot вывод: модель получает несколько примеров ввода-вывода, что особенно полезно для задач с нестандартными форматами.

— Дообучение (Fine-tuning): предобученная модель (например, BERT) адаптируется к конкретной задаче на размеченных данных. Для NER последний слой BERT заменяется на классификатор меток токенов.

— Специализированные LLM: модели вроде LUKE (обучена на связях между сущностями) или SPECTER (для научных текстов) используют доменно-специфичное предобучение.

1.4 Существующие продукты на основе ИИ для анализа акторов, действий и временных характеристик

Современные решения для извлечения акторов, действий и временных параметров из текста охватывают широкий спектр технологий. Ниже представлен анализ 30 ключевых продуктов, демонстрирующих разнообразие подходов в этой области.

1.4.1 Популярные LLM

GPT-4 (OpenAI)

GPT-4 (Generative Pre-trained Transformer 4) [7] представляет собой дальнейшее развитие авторегрессионной архитектуры OpenAI. Модель отличается улучшенной способностью к пониманию контекста, более глубоким семантическим анализом и генерацией текста повышенного качества. GPT-4 распространяется по закрытой коммерческой лицензии и доступна через API OpenAI, что требует оплаты и соблюдения установленных правил использования.

BERT (Google)

BERT (Bidirectional Encoder Representations from Transformers) [8] — модель, разработанная компанией Google, которая базируется на двунаправленной архитектуре трансформера. В отличие от авторегрессионных моделей, BERT обучается предсказывать маскированные слова в предложении с учетом контекста с обеих сторон, что делает её эффективной для задач, требующих глубокого понимания семантики текста. Модель является открытой и доступна для свободного использования и модификации, что способствует её широкому применению в академических и практических проектах. Благодаря двунаправленному обучению, BERT достигает высокой точности в таких задачах, как распознавание именованных сущностей (NER), извлечение отношений (RE), построение вопросно-ответных систем и классификация текста.

LaMDA (Google)

LaMDA (Language Model for Dialogue Applications) [9] — специализированная языковая модель, разработанная Google для создания диалоговых систем. Модель оптимизирована для ведения естественных и осмысленных бесед, поддержания контекста диалога и генерации релевантных ответов. Доступ к LaMDA ограничен и предоставляется преимущественно в рамках внутренних проектов Google или через партнёрские программы, что обусловлено стратегическими соображениями и необходимостью контроля за использованием модели. LaMDA обладает способностью поддерживать продолжительные и связные диалоги, адаптироваться к стилю общения собеседника и генерировать ответы с учётом предыдущих реплик, демонстрируя высокий уровень эмпатии.

Gemini (Google)

Gemini [10] — это семейство мультимодальных больших языковых моделей (LLM), разработанных Google. Способны обрабатывать и генерировать не только текст, но и изображения, аудио и видео. Gemini представлен в нескольких версиях (Ultra, Pro, Nano), оптимизированных для разных задач и устройств, от мощных серверов до мобильных телефонов. Gemini интегрирован в различные продукты Google, такие как поисковая система, рекламные сервисы и инструменты для разработчиков. Доступ к Gemini осуществляется через API и платформу Google AI Studio.

Claude 2 (Anthropic)

Claude 2 [11] — это модель, разработанная с использованием подхода «конституционального ИИ», направленного на обеспечение безопасного, полезного и этически корректного взаимодействия с пользователями. Модель доступна через API и оснащена механизмами фильтрации контента, предотвращающими генерацию нежелательных или опасных ответов.

PaLM (Google)

PaLM (Pathways Language Model) [12] — масштабируемая языковая модель от Google, отличающаяся высокой производительностью и способностью обрабатывать огромные объёмы текстовых данных с высокой точностью. Модель использует архитектуру Pathways, которая эффективно распределяет вычислительную нагрузку между различными компонентами системы. PaLM распространяется по закрытой коммерческой лицензии и используется преимущественно в рамках внутренних проектов Google или через платные сервисы. Благодаря своей универсальности, PaLM успешно решает широкий спектр задач, включая обработку естественного языка, машинное обучение и анализ данных, демонстрируя выдающиеся результаты в задачах, требующих глубокого понимания контекста.

DeepSeek

DeepSeek [13] представляет собой семейство языковых моделей, специализированных на задачах семантического поиска и извлечения информации. Модели DeepSeek способны обрабатывать запросы с учётом глубокого контекстного анализа, что делает их эффективными для построения поисковых систем и систем рекомендаций. DeepSeek распространяется с открытым исходным кодом, что позволяет свободно интегрировать её в исследовательские и коммерческие проекты без значительных финансовых затрат. Ключевым преимуществом DeepSeek является высокая точность в задачах семантического поиска и анализа текстовых данных, достигаемая за счёт применения современных методов глубокого обучения.

Sonar

Sonar [14] — языковая модель, разработанная для извлечения и анализа информации из текстовых данных. Особое внимание в Sonar уделяется идентификации именованных сущностей (NER), извлечению отношений между ними (RE) и определению временных характеристик событий. Модель предоставляется через API на основе платной подписки, что делает её доступной для интеграции в коммерческие продукты при соблюдении лицензионных условий. Sonar отличается высокой эффективностью при обработке разнородных текстовых данных, включая новостные статьи, научные публикации и сообщения в социальных сетях, и оптимизирована для задач информационного мониторинга.

Grok (xAI)

Grok [15] — это чат-бот с искусственным интеллектом, разработанный компанией xAI Илона Маска. Grok отличается способностью получать информацию в режиме реального времени с платформы X (ранее Twitter), а также способностью отвечать на «острые» вопросы, которые могут быть отклонены другими системами ИИ. Grok все еще находится на стадии бета-тестирования. Доступ к Grok-3 осуществляется через подписку X Premium+.

YandexGPT (Yandex)

YandexGPT [16] — российская языковая модель, специально адаптированная для работы с русскоязычным контентом. Благодаря обучению на большом объёме локальных данных, модель обеспечивает высокую точность в понимании и генерации текста на русском языке, что делает её незаменимой для интеграции в сервисы Яндекса, такие как голосовой помощник Алиса, поисковые системы и другие приложения.

1.4.2 Локальные LLM

Mistral

Mistral [17] — семейство высокоэффективных языковых моделей с открытыми весами, оптимизированных для локального запуска на потребительском GPU. Модели Mistral 7B и Mistral 8x22B используют группированные запросы (Grouped-Query Attention) и скользящее окно внимания (Sliding Window Attention), что позволяет обрабатывать длинные контексты (до 32k токенов) с минимальными ресурсами. Распространяются под лицензией Apache 2.0. Отличительная черта — способность точно определять временные последовательности в тексте благодаря обучению на синтетических данных с временными метками.

Qwen (Alibaba Cloud)

Qwen [18] — серия многоязычных моделей от Alibaba (1.8B-72B параметров) с поддержкой контекстного окна 32k токенов. Модели доступны в двух вариантах: базовые (Qwen) и chat-оптимизированные (Qwen-Chat) с квантованными версиями для CPU. Лицензия позволяет коммерческое использование с обязательной атрибуцией. Эффективна для анализа событийных цепочек в юридических документах и медиа-контенте.

Llama (Meta)

Llama [19] от Meta — открытые модели (7B-70B параметров), ставшие стандартом для локального развёртывания. Версия Llama-3 добавила архитектурные улучшения: динамическую загрузку весов и гибридное внимание (локальное и глобальное). Для временного анализа часто используется вместе с адаптерами типа Chronos-Llama, обучающимися на временных рядах текстовых событий. Распространяется под специальной коммерческой лицензией с запретом на обучение конкурентных моделей. Популярна в исследовательских проектах благодаря балансу между размером и качеством.

Falcon (TII)

Falcon-40B/180B [20] от Technology Innovation Institute (ОАЭ) — модель с революционной архитектурой, где 95% параметров выделено под внимание (FlashAttention-optimized). Поддерживает мультимодальный ввод, включая временные метаданные. Лицензия Apache 2.0 разрешает коммерческое использование без ограничений. Для временного анализа применяют метод временной анкеровки (Temporal Anchoring), связывающий события с абсолютными временными точками. Особенно эффективна в финансовой аналитике для обработки отчётов с временными зависимостями.

Zephyr (Hugging Face)

Zephyr-7B [21] — доработанная версия Mistral, оптимизированная для диалоговых сценариев с акцентом на извлечение временных паттернов. Использует технику прямого распределённого обучения (Direct Preference Optimization) без RLHF (Reinforcement learning from human feedback). Распространяется через Hugging Face Hub с открытой лицензией MIT. Интегрирует временные графы (Temporal Graph Networks) для визуализации событийных последовательностей.

Phi-3 (Microsoft)

Phi-3-mini (3.8B параметров) [22] — компактная модель Microsoft, обучавшаяся на синтетических данных с усиленными временными зависимостями. Использует инновационный подход «учебник с подсказками» (Textbooks Are All You Need) для улучшения понимания хронологий. Поддерживает квантование до 4 бит без потери точности. Лицензия MIT позволяет встраивать в мобильные приложения. Эффективна для анализа логов серверов и временной агрегации событий в реальном времени.

OLMo (Allen Institute)

OLMo (1B-7B) [23] — полностью открытая модель (веса, данные, код обучения) с архитектурой, оптимизированной для временного причинно-следственного анализа. Включает специальные токены для маркировки временных интервалов ([<start>], [<end>]). Обучена на датированных корпусах (PubMed, arXiv, News). Лицензия Apache 2.0 разрешает модификации. Особенность — встроенная проверка временной согласованности событий через механизм временных ограничений (Temporal Constraint Checking).

1.4.3 Специализированные отраслевые решения

Kira Systems

Kira Systems [24] Анализ юридических документов с выделением сторон соглашения, обязательств и сроков. ML-модели с доменной адаптацией.

ABBYY Timeline

ABBYY Timeline [25] — специализированное решение для извлечения хронологических последовательностей из технической документации. Система использует технологии оптического распознавания текста (OCR) и интеграцию с PDF для автоматического создания временных линий событий, что облегчает анализ документации и отслеживание эволюции технических процессов.

Retresco

Retresco [26] — платформа для автоматизации журналистики, которая использует алгоритмы анализа текста для генерации спортивных отчетов с точной хронологией событий. Система способна анализировать новостной поток в режиме реального времени, автоматически структурируя данные о событиях и обеспечивая оперативное создание качественных журналистских материалов.

Eigen Technologies

Eigen Technologies [27] — Платформа для извлечения данных из финансовых и юридических документов. Eigen использует NLP и машинное обучение для идентификации ключевых сущностей, отношений и временных характеристик, что позволяет автоматизировать анализ сложных документов и сократить время, затрачиваемое на ручную обработку.

1.4.4 Сервисы машинного перевода с ИИ

DeepL

DeepL [28] – нейросетевой переводчик с поддержкой 30+ языков, использующий трансформерные модели для контекстного анализа. Особое внимание уделяется передаче временных маркеров (например, «в течение двух дней») и идентификации участников действий. Доступен через API с помесячной тарификацией.

Yandex Translate

Yandex Translate [29] – многоязычный сервис от Яндекса на основе RNN и трансформеров, оптимизированный для восточноевропейских языков. Определяет временные конструкции (даты, периоды) и роли субъектов в предложении для повышения точности перевода. Интегрируется в сторонние приложения через REST API.

Google Translate

Google Translate [30] – Переводческая система с гибридной архитектурой (Transformer + BERT), автоматически аннотирующая временные выражения и именованные сущности для контекстно-чувствительного перевода. Поддерживает 134 языка, доступен через Cloud Translation API с оплатой за количество символов.

1.4.5 Чат-боты и голосовые ассистенты

Алиса (Яндекс)

Алиса [31] – Умная колонка с голосовым помощником, анализирующим временные запросы («напомни в 18:00»), идентифицирующая участников действий («закажи такси маме») и распознающая контекст многошаговых диалогов. Интегрирована с экосистемой Яндекс.Услуг (Такси, Еда).

Marusia (Mail.ru Group)

Marusia [32] — российский голосовой ассистент от Mail.ru Group, обладающий способностью точно распознавать намерения пользователей и обрабатывать временные запросы, такие как напоминания через заданные интервалы времени. Используя современные алгоритмы распознавания речи и обработки естественного языка, Marusia обеспечивает интуитивное и оперативное взаимодействие с пользователями, способствуя улучшению пользовательского опыта.

Amazon Alexa

Amazon Alexa [33] – Глобальный голосовой ассистент с поддержкой навыков (Skills), способный определять временные параметры («поставь таймер на 10 минут») и связи между событиями («напомни о встрече, когда я буду дома»). Использует трансферное обучение для адаптации к акцентам.

Dialogflow (Google)

Dialogflow [34] — платформа для создания интеллектуальных ботов, которая автоматически извлекает сущности (например, даты, имена) из пользовательских запросов. Она позволяет разработчикам создавать диалоговые системы с поддержкой естественного языка, интегрируемые в различные каналы связи, что упрощает создание интерактивных и адаптивных сервисов.

1.4.6 Другие решения

Reppify

Reppify [35] — рекрутинговая платформа, которая анализирует резюме кандидатов, выделяя ключевые данные об опыте работы, включая должности и даты, с использованием комбинации модели BERT и правил. Это позволяет работодателям быстро оценивать квалификацию кандидатов и оптимизировать процесс найма.

Clarabridge CX Analytics

Clarabridge CX Analytics [36] — платформа для управления клиентским опытом, которая автоматически обнаруживает инциденты и отслеживает время их эскалации в обращениях клиентов. Используя современные алгоритмы анализа текста и временных характеристик, система помогает организациям оперативно реагировать на проблемы, повышая удовлетворенность клиентов и улучшая качество обслуживания.

1.5 Выбор и обоснование оптимального подхода

В данном разделе рассматриваются ключевые факторы, влияющие на выбор метода извлечения акторов, действий и временных характеристик из текстовых данных, а также обосновывается применение конкретной модели. Особое внимание уделяется требованиям открытости и воспроизводимости, что обеспечивает прозрачность методологии и возможность её дальнейшей адаптации в исследовательских проектах.

1.5.1 Ключевые критерии выбора метода

При выборе оптимального подхода учитываются как технические, так и практические и методологические требования. Основные критерии включают:

1) открытость и воспроизводимость: применение моделей с открытым исходным кодом является приоритетом в академических исследованиях, поскольку это позволяет проводить независимую верификацию результатов, модифицировать архитектуру под специфику задачи и избегать зависимости от проприетарных решений. Открытые реализации способствуют легкой интеграции предобученных весов в аналитические конвейеры;

2) точность и надёжность: избранная модель должна демонстрировать стабильные показатели по метрикам Precision (доля корректно извлечённых сущностей от общего числа извлечённых) и Recall (доля извлечённых сущностей от общего числа релевантных).

3) адаптивность к доменным особенностям: метод должен предусматривать механизмы адаптации к специфике текстов из различных областей (например, юридической или медицинской), где характер лексики и синтаксиса существенно отличается от общего языка. В таких случаях использование методов дообучения позволяет модели учитывать доменно-специфические особенности (например, особую интерпретацию аббревиатур в технической документации);

4) эффективность использования вычислительных ресурсов: при обработке больших объемов данных (например, архивов новостных сообщений) важна оптимизация расхода памяти и времени инференса. Модели с чрезмерным числом параметров могут оказаться непрактичными при ограниченных вычислительных ресурсах;

5) устойчивость к вариативности входных данных: решение должно корректно обрабатывать тексты с орфографическими ошибками, неформальной лексикой (например,

сообщения в соцсетях) и мультиязычными вставками;

6) поддержка русского языка: крайне важно, чтобы выбранная LLM могла работать с русским языком, учитывала особенности русской морфологии и его богатую систему словообразования.

Сравнительная таблица

Модель	Доступность	Русский	Адаптивность	Кол-во параметров
GPT-4	Через API	Да	Высокая	∼1.76T
BERT	Открытая	Да	Средняя/Высокая	110M - 340M
LaMDA	Закрытая	Да	Высокая	137B
Gemini	Через API	Да	Высокая	Разные версии
Claude 2	Через API	Да	Высокая	Неизвестно
PaLM 2	Через API	Да	Высокая	340B
DeepSeek	Открытая	Да	Высокая	До 67B
Sonar	Открытая	Да	Высокая	Неизвестно
Grok	Через API	Да	Хорошая	33B - 414B
YandexGPT	Через API	Да	Высокая	Разные версии
Mistral	Открытая	Да	Хорошая	7B - 70B
Qwen	Открытая	Да	Хорошая	До 110B+
Llama	Открытая	Да	Хорошая	7B - 70B
Falcon	Открытая	Да	Средняя	7B - 180B
Zephyr	Открытая	Да	Средняя	7B
Phi-3	Открытая	Да	Средняя	3.8B
OLMo	Открытая	Нет	Средняя	7B, 65B

Таблица 1.1 — Сравнительная таблица языковых моделей 1

Модель	Точность	Устойчивость
GPT-4	Очень высокая	Высокая
BERT	Высокая	Хорошая
LaMDA	Высокая	Высокая
Gemini	Очень высокая	Высокая
Claude 2	Очень высокая	Высокая
PaLM 2	Высокая	Высокая
DeepSeek	Хорошая	Хорошая
Sonar	Высокая	Хорошая
Grok	Хорошая	Хорошая
YandexGPT	Высокая	Высокая
Mistral	Хорошая	Хорошая
Qwen	Хорошая	Средняя
Llama	Хорошая	Хорошая
Falcon	Средняя	Средняя
Zephyr	Средняя	Средняя
Phi-3	Средняя	Средняя
OLMo	Хорошая	Средняя

Таблица 1.2 — Сравнительная таблица языковых моделей 2

1.5.2 Mistral 7B

Модель Mistral 7B представляет собой современное решение в области обработки естественного языка, сочетающее в себе высокую точность, адаптивность и эффективное использование вычислительных ресурсов, что делает её особенно привлекательной для задач извлечения акторов, действий и временных характеристик в текстах. Одним из основных преимуществ данной модели является её открытость и воспроизводимость: использование открытых архитектур и предобученных весов позволяет проводить независимую верификацию результатов, а также адаптировать модель под специфические требования исследовательских задач. В свою очередь, базовая (vanilla) версия Mistral 7b демонстрирует высокую универсальность при обработке широкого спектра текстовых данных, тогда как instruct-версия, дообученная на примерах инструктивного взаимодействия, обеспечивает улучшенную интерпретацию запросов и более точное соблюдение заданных инструкций, что может быть критически важно для узконаправленных приложений.

Особое внимание в реализации Mistral 7B уделено технологии квантования, в частности использованию 4-битного квантования. Применение данного метода позволяет существенно снизить требования к объёму оперативной памяти и ускорить время инференса без значительной деградации качества предсказаний. 4-битное квантование обеспечивает оптимальное соотношение между степенью сжатия и сохранением информативности параметров модели, что особенно важно при работе с большими корпусами данных, где критически важна эффективность обработки. При этом модель сохраняет стабильные показатели по метрикам Precision, Recall и F1-score даже при наличии доменно-специфических особенностей, неформальной лек-

сики и орфографических ошибок во входных текстах [37].

Вывод

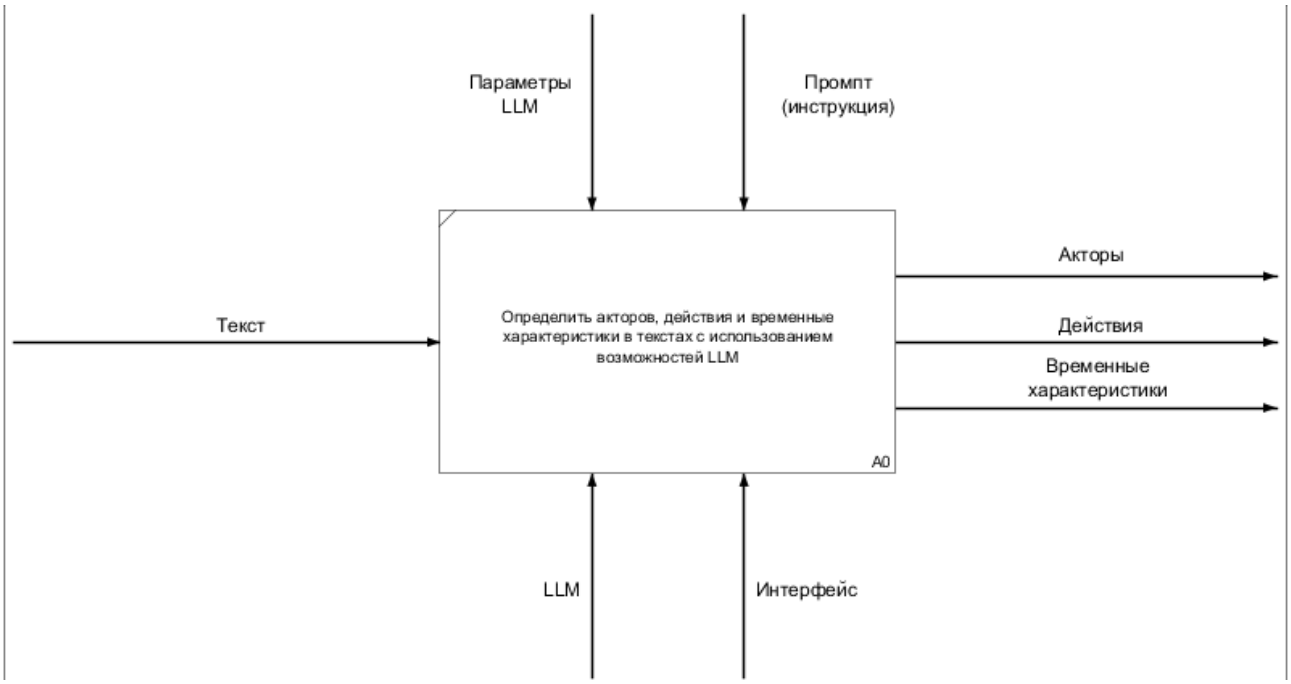

Рисунок 1.1 — IDEF0 диаграмма

В рамках представленной аналитической части был проведен комплексный обзор современного состояния области извлечения информации из неструктурированных текстовых данных, с акцентом на идентификацию акторов, действий и временных характеристик

В разделе были подробно рассмотрены ключевые концепции, лежащие в основе LLM. был проведен детальный анализ основных задач извлечения информации: распознавания именованных сущностей (NER), извлечения отношений (RE), извлечения временной информации (TIE) и извлечения событий (EE). Обширный раздел был посвящен обзору существующих программных продуктов и платформ, реализующих функциональность извлечения информации

В совокупности, выбор модели Mistral 7b с применением 4q квантования обусловлен её способностью удовлетворять основным критериям выбора. При этом возможность применения instruct-версии модели позволяет дополнительно адаптировать её под специфические задачи, требующие точного соблюдения инструктивных параметров, что делает данное решение оптимальным для широкого спектра исследовательских и прикладных задач в области анализа текстов.

2 Конструкторская часть

В аналитической части работы была представлена концептуальная модель процесса извлечения информации, визуализированная с помощью IDEF0 диаграммы. На основе этого анализа в конструкторской части рассматриваются детали процессов, происходящих внутри, которые и выявляют необъходимые сущности с использованием больших языковых моделей (LLM).

2.1 Детали алгоритма

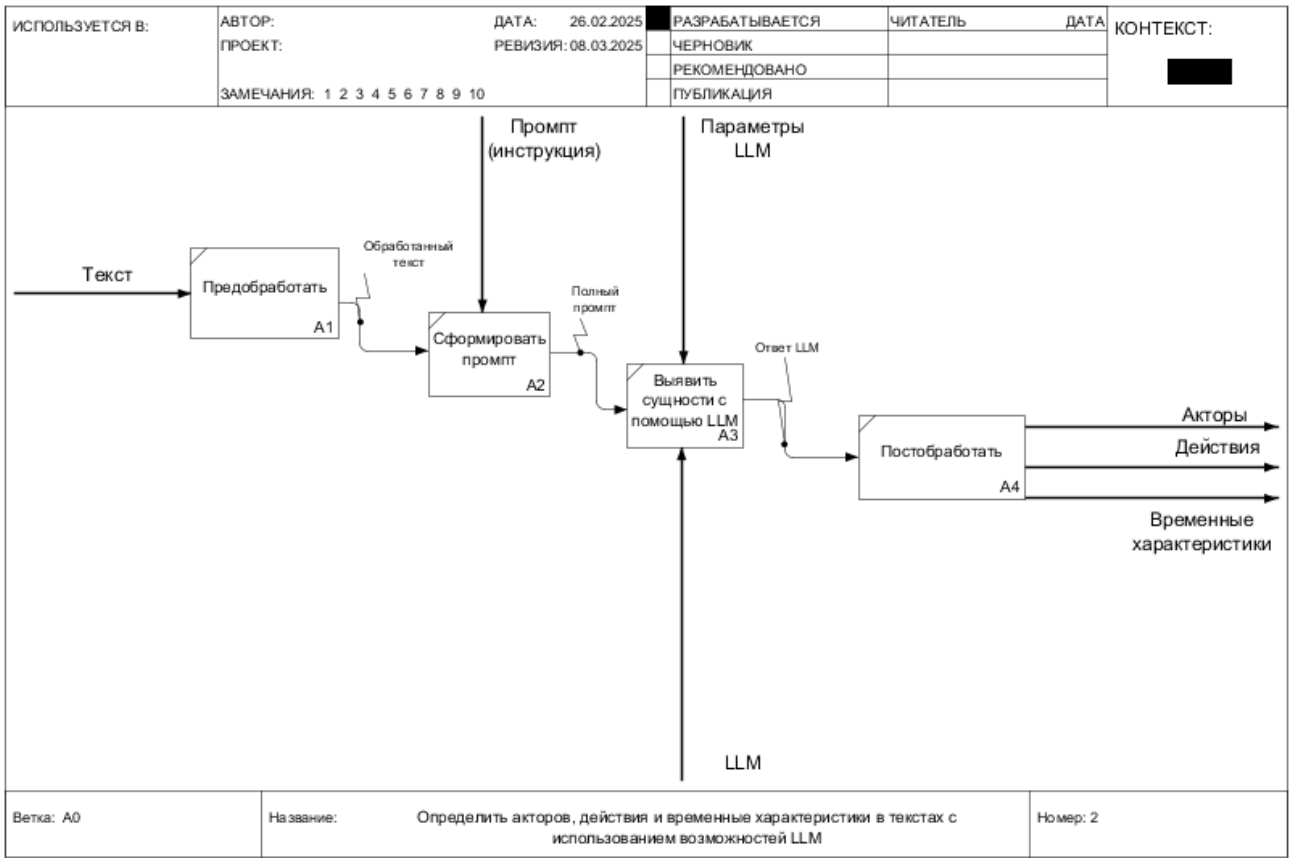

Рисунок 2.1 — Алгоритм определения акторов, действий и временных характеристик

Диаграмма (рис. 2.1) представляет IDEF0-модель процесса извлечения сущностей, в которой опускается этап взаимодействия с пользователем.

2.1.1 Алгоритмы обработки

Предобработка

Перед анализом текста производится его предварительная обработка, которая включает в себя:

— удаление лишних пробельных символов, переносов строк и иных символов;

— очистка от неинформативных элементов, таких как специальные символы, HTML-теги и т.д;

— разделение текста на логические фрагменты (абзацы, предложения).

Формирование промпта

Для корректной работы LLM необходимо сформировать промпт, содержащий четкую инструкцию для модели. В случае Instruct-версий моделей применяется формат явных директив.

Например, текст может быть представлен в следующем виде:

[INST] Определи акторов, их действия и временные характеристики.
Представь результат в формате JSON. [/INST]
Текст: "В 2023 году исследователь написал статью."

Такой подход уменьшает вероятности появления ошибок в неправильной интерпретации промпта, что улучшает качество ответа.

Выявление сущностей

На этом этапе происходит автоматическое определение акторов, их действий и временных характеристик на основе обученных представлений модели, повлиять на который можно только через конфигурационный файл модели.

Постобработка

После получения ответа от модели выполняется его дополнительная обработка:

— проверяется соответствие выходного формата запрашиваемому (например, если ответ должен быть в формате JSON, производится валидация структуры);

— выделяются только те элементы, которые требовалось определить, устраняя возможные избыточные данные.

2.2 Взаимодействие с системой

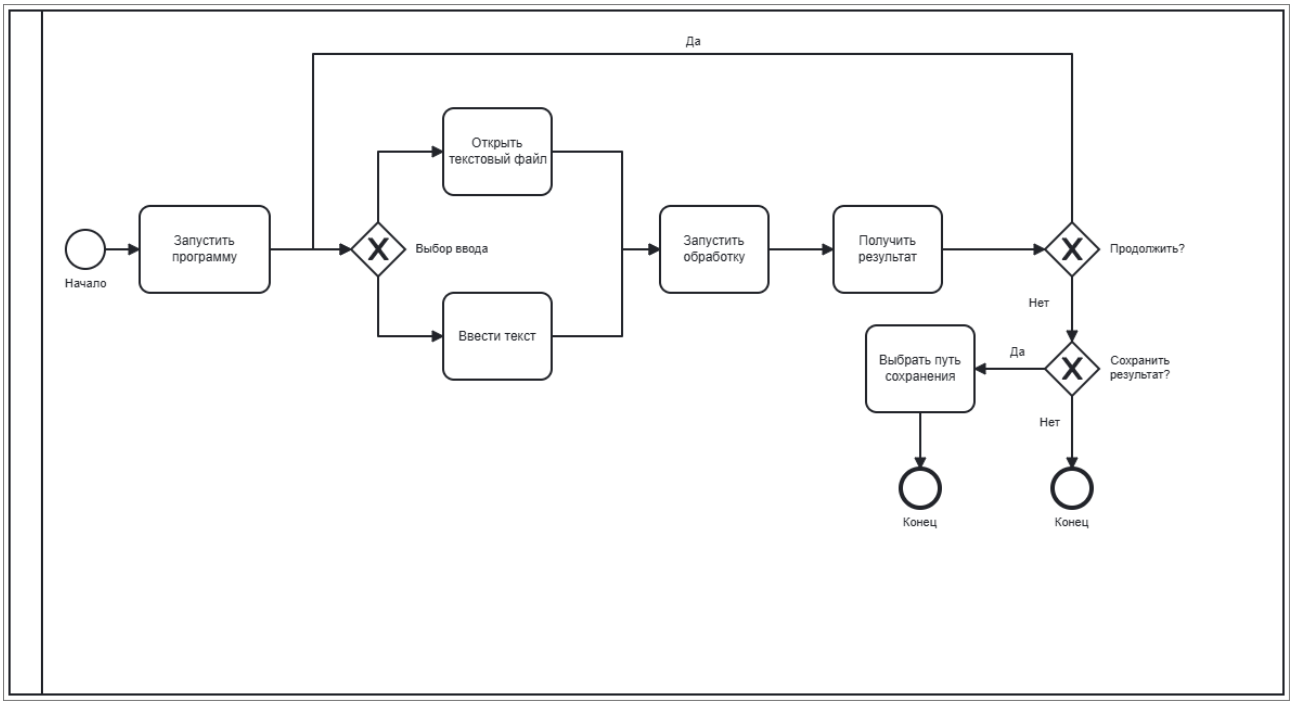

Рисунок 2.2 — Алгоритм взимодейсвия с системой

На рисунке 2.2 представлен алгоритм взаимодействие пользователя с системой посредством консольного либо графического приложения. В обоих случаях, исходный текст для анализа вводится пользователем либо непосредственно в поле ввода интерфейса, либо загружается из текстового файла. Обработка текста инициируется пользователем. Результаты анализа представляются различными способами в зависимости от типа интерфейса. В графическом приложении производится визуализация: извлеченные сущности (акторы, действия, временные характеристики) выделяются различными цветами непосредственно в тексте (например, акторы – красным цветом, действия – зеленым, временные характеристики – оранжевым). В консольном приложении вывод результатов осуществляется в текстовом формате, с явным указанием принадлежности каждого элемента к определенной категории сущностей. Дополнительно, вне зависимости от типа интерфейса, предусмотрена возможность сохранения результатов анализа в файл формата JSON для последующей обработки или интеграции с другими системами.

2.2.1 Конфигурационный файл

Для тонкой настройки процесса обработки текста и представления результатов используется конфигурационный файл в формате JSON. Данный файл позволяет пользователю задать ряд параметров, влияющих на поведение системы, такие как:

— шаблон промпта;

— формат вывода;

— цветовая схема (для графического интерфейса);

— путь к файлу с текстом по умолчанию;
— температура модели (отвечает за случайность сгенерированного текста);

Вывод

Разработанная архитектура демонстрирует модульный и масштабируемый подход к автоматизированному извлечению акторов, действий и временных характеристик из текстовых данных с использованием больших языковых моделей. Детально описаны этапы: предобработка текста, формирование четкого запроса к модели, автоматическое выявление сущностей и последующая постобработка. Предусмотрены возможные варианты взаимодействия с системой — графический интерфейс с визуальной подсветкой ключевых элементов и консольное приложение для текстового вывода. Гибкая настройка параметров через конфигурационный файл в формате JSON позволяет адаптировать систему под различные условия эксплуатации.

3 Технологическая часть

Вывод

4 Исследовательская часть

Вывод

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Текст в эпоху Больших Данных* [Электронный ресурс]. – Режим доступа: https://www.osp.ru/os/2012/06/13017063 (дата обращения 22.02.2025)

2. *Инструмент автоматизированного сбора данных для улучшения бизнес-процессов* [Электронный ресурс]. – Режим доступа: https://www.tadviser.ru/index.php/Новости:Использование_инструмента_автоматизированного_сбор процессов (дата обращения 22.02.2025)

3. *Как искусственный интеллект позволяет упростить рутинные операции* [Электронный ресурс]. – Режим доступа: https://ritg.ru/blog/kak_iskusstvennyy_intellekt_pozvolyaet_uprostit_rutinnye_operatsii/ (дата обращения 22.02.2025)

4. *word2vec* [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/text/tutorials/word2vec (дата обращения 23.02.2025)

5. Васвани А., Шазир Н., Пармар Н. [и др.] Attention is All You Need [Электронный ресурс] // arXiv : [сайт]. – 2017. – URL: https://arxiv.org/pdf/1706.03762v3 (дата обращения: 26.02.2025).

6. Глубокое обучение / пер. с анг. А. А. Слинкина. – 2-е изд., испр. – М.: ДМКПресс, 2018. – 164 с.

7. *OpenAI API* [Электронный ресурс]. – Режим доступа: https://openai.com/api/ (дата обращения 23.02.2025)

8. *bert-base-NER* [Электронный ресурс]. – Режим доступа: https://huggingface.co/dslim/bert-base-NER (дата обращения 23.02.2025)

9. *LaMDA* [Электронный ресурс]. – Режим доступа: https://blog.google/technology/ai/lamda/ (дата обращения 23.02.2025)

10. *Gemini* [Электронный ресурс]. – Режим доступа: https://aistudio.google.com (дата обращения 23.02.2025)

11. *Anthropic API* [Электронный ресурс]. – Режим доступа: https://www.anthropic.com/api (дата обращения 23.02.2025)

12. *PaLM* [Электронный ресурс]. – Режим доступа: https://research.google/blog/pathways-language-model-palm-scaling-to-540-billion-parameters-for-breakthrough-performance/ (дата обращения 23.02.2025)

13. *DeepSeek-Coder* [Электронный ресурс]. – Режим доступа: https://github.com/deepseek-ai/DeepSeek-Coder (дата обращения 23.02.2025)

14. *Sonar Perplexity* [Электронный ресурс]. – Режим доступа: https://sonar.perplexity.ai/ (дата обращения 23.02.2025)

15. *Grok* [Электронный ресурс]. – Режим доступа: https://x.ai/ (дата обращения 23.02.2025)

16. *YandexGPT* [Электронный ресурс]. – Режим доступа: https://cloud.yandex.ru/services/yandexgpt (дата обращения 23.02.2025)

17. *Mistral-7B-Instruct-v0.3* [Электронный ресурс]. – Режим доступа: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3 (дата обращения 23.02.2025)

18. *Qwen2.5-VL-7B-Instruct* [Электронный ресурс]. – Режим доступа: https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct (дата обращения 23.02.2025)

19. *Llama-3.1-8B-Instruct* [Электронный ресурс]. – Режим доступа: https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct (дата обращения 23.02.2025)

20. *Falcon3-7B-Instruct* [Электронный ресурс]. – Режим доступа: https://huggingface.co/tiiuae/Falcon3-7B-Instruct (дата обращения 23.02.2025)

21. *zephyr-7b-beta* [Электронный ресурс]. – Режим доступа: https://huggingface.co/HuggingFaceH4/zephyr-7b-beta (дата обращения 23.02.2025)

22. *Phi-3.5-mini-instruct* [Электронный ресурс]. – Режим доступа: https://huggingface.co/microsoft/Phi-3.5-mini-instruct (дата обращения 23.02.2025)

23. *OLMo* [Электронный ресурс]. – Режим доступа: https://allenai.org/olmo (дата обращения 23.02.2025)

24. *Kira Systems* [Электронный ресурс]. – Режим доступа: https://kirasystems.com/ (дата обращения 23.02.2025)

25. *ABBYY* [Электронный ресурс]. – Режим доступа: https://www.abbyy.com/ (дата обращения 23.02.2025)

26. *Retresco* [Электронный ресурс]. – Режим доступа: https://www.retresco.de/ (дата обращения 23.02.2025)

27. *Eigen Technologies* [Электронный ресурс]. – Режим доступа: https://eigentech.com/ (дата обращения 23.02.2025)

28. *DeepL API* [Электронный ресурс]. – Режим доступа: https://www.deepl.com/ru/products/api (дата обращения 24.02.2025)

29. *Yandex Translate* [Электронный ресурс]. – Режим доступа: https://yandex.cloud/en/services/translate (дата обращения 24.02.2025)

30. *Google Translate API* [Электронный ресурс]. – Режим доступа: https://developers.google.com/ml-kit/language/translation?hl=ru (дата обращения 24.02.2025)

31. *Яндекс Станция* [Электронный ресурс]. – Режим доступа: https://alice.yandex.ru/station (дата обращения 24.02.2025)

32. *Маруся* [Электронный ресурс]. – Режим доступа: https://marusia.vk.com/ (дата обращения 24.02.2025)

33. *Alexa Skills Kit* [Электронный ресурс]. – Режим доступа: https://developer.amazon.com/en-US/alexa (дата обращения 24.02.2025)

34. *Dialogflow* [Электронный ресурс]. – Режим доступа: https://cloud.google.com/products/conversational-agents (дата обращения 24.02.2025)

35. *Reppify* [Электронный ресурс]. – Режим доступа: https://wilsonhcg.atlassian.net/wiki/spaces/KH/page (дата обращения 24.02.2025)

36. *Qualtrics* [Электронный ресурс]. – Режим доступа: https://www.qualtrics.com/ (дата обращения 24.02.2025)

37. *Mistral 7B* [Электронный ресурс]. – Режим доступа: https://mistral.ai/news/announcing-mistral-7b (дата обращения 25.02.2025)