**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, $\theta$ – вектор параметров модели, оптимизируемый в процессе обучения.

Одной из самых популярных архитектур, лежащих в основе современных LLM, является трансформер. Архитектура трансформера отличается от классических рекуррентных нейронных сетей (Reccurrent Neural Networks, RNN) тем, что она 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 или GloVe).

### 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.

Таким образом, каждая позиция в последовательности получает взвешенное представле-

ние, учитывающее информацию со всех остальных позиций, что и формирует вектор контекста.

### 1.2.4 Генерация текста и предсказание следующего слова

Одна из основных задач LLM – предсказание следующего слова в последовательности. При генерации текста модель анализирует уже сгенерированные токены и на их основе предсказывает наиболее вероятный следующий токен. Это делается путём вычисления распределения вероятностей по всему словарю. Формально, вероятность появления следующего токена $w_t$ при заданной последовательности $w_1, w_2, \ldots, w_{t-1}$ определяется как:

$$P(w_t \mid w_1, \ldots, w_{t-1}) = \frac{\exp(\mathbf{z}_{w_t})}{\sum_{w \in V} \exp(\mathbf{z}_w)},$$

где $\mathbf{z}_{w_t}$ — логит (необработанный выход нейронной сети) для токена $w_t$, $V$ – словарь модели.

Функция экспоненты используется для преобразования логитов в положительные значения, что позволяет затем нормализовать их в распределённое вероятностное пространство, где сумма вероятностей по всему словарю равна 1.

### 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 с использованием LLM

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. Современные подходы к решению этих задач эволюционировали от ручного создания правил до использования самообучающихся систем на основе больших языковых моделей (LLM). В данном разделе рассматривается эволюция методов IE — от классических алгоритмов до трансформерных архитектур, анализируются их особенности и области применения.

### 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)**: иден-

тификация 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), «Европу» (географический объект) и «март 2023 года» (временная метка). Точность NER влияет на качество последующих этапов, таких как извлечение отношений;

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

— **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»;

— **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.

### 1.3.2 Эволюция методов извлечения информации

#### Rule-based системы

Ранние подходы к IE основывались на ручном создании правил и шаблонов. Правила могли включать лексические паттерны (например, регулярные выражения для дат: mm.dd.yyyy) или синтаксические конструкции (например, извлечение организаций после ключевых слов «ООО», «Inc»). Такие системы демонстрировали высокую точность в узких доменах, где языковые конструкции предсказуемы. Однако поддержка и масштабирование правил для разнородных данных (например, соцсети vs. юридические тексты) требовали значительных трудозатрат. Кроме того, правила, созданные для одного языка, часто оказывались неприменимы к другим из-за различий в грамматике.

#### Статистические методы машинного обучения

С развитием машинного обучения появились методы, способные обобщать закономерности на основе размеченных данных.

Скрытые марковские модели (Hidden Markov Models, HMM): моделируют последовательности скрытых состояний (например, типы сущностей) и наблюдаемых токенов. Формула совместной вероятности:

$$P(X, Y) = P(y_1) \prod_{t=2}^{T} P(y_t \mid y_{t-1}) \prod_{t=1}^{T} P(x_t \mid y_t),$$

где $X$ – последовательность наблюдаемых токенов (слов), $Y$ – последовательность скрытых

состояний (меток), $P(y_1)$ – априорная вероятность того, что первое слово в последовательности имеет метку $y_1$, $\prod_{t=2}^{T} P(y_t \mid y_{t-1})$ – произведение вероятностей перехода между скрытыми состояниями. Это выражение описывает, насколько вероятно, что метка на позиции $t$ является $y_t$, при условии, что на предыдущей позиции была метка $y_{t-1}$. $\prod_{t=1}^{T} P(x_t \mid y_t)$ – произведение вероятностей эмиссии. Определяет вероятность появления наблюдаемого токена $x_t$ при заданной метке $y_t$.

HMM эффективны для коротких контекстов, но игнорируют глобальные зависимости между токенами.

Условные случайные поля (Conditional Random Fields, CRF) представляют дискриминативные модели, оценивающие вероятность меток $Y$ при заданном $X$:

$$P(Y \mid X) = \frac{1}{Z(X)} \exp\left(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(y_{t-1}, y_t, X, t)\right),$$

где $f_k(y_{t-1}, y_t, X, t)$ – функция признаков, оценивающая вклад конкретной характеристики, $\lambda_k$ – весовой коэффициент для функции признаков $f_k$, $Z(X)$ – нормировочная функция, обеспечивающая суммарную вероятность всех возможных последовательностей равной 1.

CRF учитывают соседние метки, что улучшает согласованность предсказаний (например, метка «B-ORG» чаще следует за «B-ORG», чем за «I-PER»).

Эти методы требовали тщательной инженерии признаков (например, часть речи, морфологические характеристики) и большого объёма размеченных данных. Так, для CRF признаки могли включать наличие заглавной буквы, соседние слова или суффиксы.

## Нейронные сети до эпохи трансформеров

С появлением глубокого обучения стали использоваться архитектуры, автоматически извлекающие признаки из текста:

— RNN/LSTM (Long Short-Term Memory): рекуррентные сети обрабатывают текст последовательно, сохраняя скрытое состояние между токенами. LSTM решают проблему затухания градиентов через механизм вентилей, управляющих потоком информации. Например, в предложении «Компания [X], основанная в 1990 году, объявила о банкротстве» LSTM может связать «основанная в 1990 году» с упоминанием компании [X], даже если между ними есть дистанция.

— CNN (Convolutional Neural Network): свёрточные сети выделяют локальные n-граммные паттерны. Например, в задаче классификации организаций фильтры CNN могут активироваться на сочетаниях вроде «корпорация [X]» или «[X] GmbH».

Эти модели снизили зависимость от ручного создания признаков, но имели ограничения: RNN трудно параллелизовать из-за последовательной обработки, а CNN не учитывали глобальный контекст. Для определения, что «Apple» в одном предложении относится к компании, а в другом — к фрукту, требовались дополнительные механизмы внимания.

### 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, GPT-3 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. Эксперименты показывают, что дообученный RoBERTa достигает F1=92.1 на CoNLL-2003, превосходя CRF (F1=88.3).

— *Специализированные LLM*: модели вроде LUKE (обучена на связях между сущностями) или SPECTER (для научных текстов) используют доменно-специфичное предобучение. Например, PubMedBERT, обученная на медицинских статьях, точнее извлекает медицинские термины по сравнению с общей BERT.

## Сравнение с традиционными методами

LLM устраняют необходимость ручного создания правил и признаков, но требуют значительных вычислительных ресурсов для обучения. Там, где CRF обрабатывает 10K документов за минуту на CPU, inference GPT-3 требует GPU. Однако LLM демонстрируют лучшую обобщающую способность на разнородных данных: например, извлечение редко встречающихся сущностей в социальных медиа, где правила и CRF недостаточны.

Гибридные подходы (например, использование правил для предфильтрации текста с последующим применением LLM) позволяют балансировать между точностью и ресурсозатратностью. Такие решения актуальны в доменах с жёсткими требованиями к интерпретируемости, где каждое извлечение должно быть верифицируемо.

## 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 – от облачных API до специализированных отраслевых платформ. Ниже представлен анализ 30 ключевых продуктов, демонстрирующих разнообразие подходов в этой области.

### 1.4.1 Популярные LLM

#### GPT-4 / O3 (OpenAI)

GPT-4 (Generative Pre-trained Transformer 4) является преемником GPT-3 и представляет собой дальнейшее развитие авторегрессионной архитектуры OpenAI. O3 — это набор инструментов и API, предоставляемых OpenAI, которые позволяют разработчикам интегрировать возможности GPT-4 в свои приложения. Модель отличается улучшенной способностью к пониманию контекста, более глубоким семантическим анализом и генерацией текста повышенного качества. Как и её предшественница, GPT-4 распространяется по закрытой коммерческой лицензии и доступна через API OpenAI, что требует оплаты и соблюдения установленных правил использования. По сравнению с GPT-3, GPT-4 демонстрирует более высокую точность, лучшее качество генерируемого текста и способность решать более сложные многоэтапные задачи NLP, а также обладает улучшенными возможностями в области few-shot обучения.

#### BERT (Google)

BERT (Bidirectional Encoder Representations from Transformers) — 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) — специализированная языковая модель, разработанная Google для создания диалоговых систем. Модель оптимизирована для ведения естественных и осмысленных бесед, поддержания контекста диалога и генерации ре-

левантных 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 — 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.

## LLaMA (Meta)

LLaMA (Large Language Model Meta AI) представляет собой семейство языковых моделей, разработанных компанией Meta и ориентированных на исследовательские цели. Модель предлагается в различных вариантах, что позволяет выбирать оптимальный баланс между производительностью и вычислительными затратами. Распространяется LLaMA по условно открытой лицензии для исследовательского использования; доступ регулируется определёнными условиями и может требовать регистрации или соблюдения ограничений. LLaMA предоставляет возможность экспериментировать с современными архитектурами трансформеров, что позволяет проводить детальный анализ их работы и демонстрировать конкурентоспособные результаты в широком спектре задач NLP.

## PaLM (Google)

PaLM (Pathways Language Model) — 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 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 — 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 — 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 — 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 Специализированные отраслевые решения

#### Kira Systems

Анализ юридических документов с выделением сторон соглашения, обязательств и сроков. ML-модели с доменной адаптацией.

#### ABBYY Timeline

ABBYY Timeline — 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 — 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

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.

### 1.4.3 Платформы для анализа данных

#### Diffbot

Diffbot — платформа, специализирующаяся на извлечении структурированных данных из веб-страниц с использованием машинного обучения. Ее Knowledge Graph API автоматически собирает, анализирует и организует данные, включая события, даты и другие важные параметры, что облегчает построение информационных графов и интеграцию данных для аналитических задач.

### Lexalytics

Lexalytics — аналитическая платформа для анализа опросов клиентов с временной привязкой отзывов. Система использует гибридный подход, объединяя машинное обучение и правило-ориентированные алгоритмы, что позволяет быстро выявлять ключевые тенденции в клиентском опыте и улучшать качество обслуживания.

### SAS Text Miner

SAS Text Miner — аналитическое решение, интегрирующее методы анализа временных рядов с продвинутой текстовой аналитикой для прогнозного моделирования. Оно позволяет выявлять скрытые тренды и взаимосвязи в больших объемах данных, комбинируя статистические модели с технологиями NLP, что делает его незаменимым инструментом для бизнеса и научных исследований.

## 1.4.4 Сервисы машинного перевода с ИИ

Современные системы машинного перевода, основанные на нейросетевых архитектурах, демонстрируют способность к анализу семантики, распознаванию акторов и временных конструкций в тексте, что критически важно для обеспечения контекстно-зависимого перевода.

### DeepL

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

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

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 Сервисы автоматической модерации контента

#### CleanTalk

Определяет спам, мат и рекламу в пользовательских отзывах. Интегрируется с WordPress, OpenCart.

#### Perspective API (Google Jigsaw)

Оценивает токсичность текста через ML-модели. Используется Reddit и Wikipedia.

#### Sightengine

Sightengine — платформа мультимодальной модерации, которая анализирует как текстовый, так и визуальный контент для выявления агрессии, дискриминации и других негативных аспектов. Используя технологии глубокого обучения, система обеспечивает высокую точность модерации, что помогает поддерживать безопасное и дружественное сообщество.

#### Bodyguard.ai

Сервис, специализирующийся на защите от онлайн-травли и оскорблений в режиме реального времени. Использует комбинацию алгоритмов машинного обучения и правил для обнаружения и фильтрации токсичного контента в социальных сетях, чатах и на других платформах.

### 1.4.6 Чат-боты и голосовые ассистенты

#### Алиса (Яндекс)

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 — 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

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 от Google — 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.

### 1.4.7 Другие решения

### Reppify

Reppify — 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 — 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). Для задач Named Entity Recognition (NER) обычно требуется F1-score не ниже 85%, что подтверждено бенчмарками на датасетах, таких как CoNLL-2003.

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) простота интеграции в аналитические системы: наличие готовых API, поддержка популярных фреймворков (например, Hugging Face, spaCy) и подробная документация существенно сокращают время внедрения решения. Для исследовательских проектов важен также доступ к предобученным эмбеддингам.

### 1.5.2 Анализ применимости модели BERT

В качестве базовой модели выбран BERT (Bidirectional Encoder Representations from Transformers). Его трансформерная архитектура удовлетворяет приведенным критериям, обеспечивая баланс между точностью, адаптивностью и эффективностью.

Архитектурные особенности. Двунаправленный механизм внимания позволяет BERT учитывать контекст с обеих сторон от целевого токена, что существенно снижает неоднознач-

ность интерпретации.

Адаптация к специфике домена. Методика дообучения (fine-tuning) на специализированных корпусах (например, юридических или медицинских текстах) позволяет модифицировать веса модели для лучшего распознавания доменно-специфичных терминов. Например, дообученная версия BioBERT демонстрирует F1-score 92.4 на медицинских текстах по сравнению с 86.2 для базовой модели.

Баланс между точностью и расходом ресурсов. Модель BERT-base, насчитывающая около 110 миллионов параметров, обеспечивает приемлемую скорость инференса на GPU (до 200 документов в секунду).

### 1.5.3 Ограничения модели и методы их компенсации

Несмотря на очевидные преимущества, применение BERT сопровождается рядом технических и методологических сложностей:

— ограничение длины контекста: стандартная архитектура BERT обрабатывает максимум 512 токенов, что затрудняет анализ длинных документов. Для решения этой проблемы применяется иерархическая обработка: сначала извлекаются сущности из каждого сегмента, а затем выполняется постобработка для устранения дублирования и разрешения конфликтов между сегментами;

— зависимость от размеченных данных: высокая эффективность модели обусловлена наличием качественной разметки. При отсутствии доменно-специфичных размеченных данных используются методы слабого обучения (weak supervision), такие как генерация псевдоразметки на основе правил или внешних баз знаний;

## Вывод

Таким образом, модель BERT демонстрирует оптимальное соотношение точности, адаптивности и эффективности, удовлетворяя требованиям извлечения информации из разнородных текстовых источников. Открытая экосистема и активная поддержка сообщества способствуют быстрой интеграции и адаптации модели в исследовательских проектах.

# 2 Конструкторская часть

## Вывод

# 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 26.01.2025)

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 26.01.2025)

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 26.01.2025)