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 Разработка программного решения на основе искусственного интеллекта, способного эффекти

Студент Батуев А.Г.

Группа ИУ7-46Б

Преподаватели Строганов Ю.В.

Москва, 2025

Содержание

ВВЕДЕНИЕ **5**

1 Аналитическая часть **6**

1.1 Введение в большие языковые модели 6

1.1.1 Определение LLM 6

1.2 Принципы работы LLM 7

1.2.1 Токенизация 7

1.2.2 Эмбеддинги 7

1.2.3 Механизм внимания и векторы контекста 7

1.2.4 Генерация текста и предсказание следующего слова 8

1.2.5 Prompt Engineering 8

1.3 Обзор популярных LLM 8

1.3.1 GPT-3 (OpenAI) 8

1.3.2 GPT-4 (OpenAI) 9

1.3.3 BERT (Google) 9

1.3.4 LaMDA (Google) 9

1.3.5 LLaMA (Meta) 10

1.3.6 PaLM (Google) 10

1.3.7 DeepSeek 10

1.3.8 Sonar 10

1.4 Методы и алгоритмы извлечения информации с использованием LLM 11

1.4.1 Обзор задач извлечения информации 11

1.4.2 Эволюция методов извлечения информации 12

1.4.3 Современные подходы на основе LLM 13

1.5 Выбор и обоснование оптимального подхода 15

1.5.1 Ключевые критерии выбора 15

1.5.2 Анализ применимости BERT 16

1.5.3 Ограничения и методы их компенсации 16

1.5.4 Интеграция в аналитический конвейер 17

1.5.5 Заключение и переход к реализации 17

2 Конструкторская часть **18**

3 Технологическая часть **19**

4 Исследовательская часть **20**

ЗАКЛЮЧЕНИЕ **21**

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ **22**

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). Д ля достижения этой цели предполагается решить следующие задачи:

1) провести анализ существующих методов автоматического извлечения информации из текстов, выявив их преимущества и недостатки;

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 (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 отличается от классических рекуррентных нейронных сетей (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.

Таким образом, LLM представляют собой мощный инструмент, позволяющий моделировать естественный язык на основе глубокой нейронной архитектуры, способной работать с огромными объёмами 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 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: - 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 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 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.

Такой подход позволяет модели генерировать связный текст, отвечать на вопросы и даже выполнять сложные задачи, требующие понимания контекста.

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. Для новичка важно понимать, что правильная формулировка промта существенно влияет на качество ответа модели.

1.3 Обзор популярных LLM

Современный ландшафт больших языковых моделей характеризуется широким разнообразием архитектур, подходов к обучению и областей применения. В данном разделе представлен обзор наиболее значимых LLM с акцентом на их ключевые характеристики и режимы доступности, что является критически важным при выборе оптимальной модели для конкретных исследовательских и прикладных задач.

1.3.1 GPT-3 (OpenAI)

GPT-3 (Generative Pre-trained Transformer 3), разработанная компанией OpenAI, представляет собой авторегрессионную языковую модель, демонстрирующую выдающиеся способности к генерации текста и решению широкого спектра задач обработки естественного языка (NLP). Модель способна адаптироваться к различным контекстам посредством методов few-shot и zero-shot обучения. Распространяется GPT-3 по закрытой лицензии и доступна исключительно

через коммерческий API OpenAI, что подразумевает необходимость оплаты за использование и соблюдение условий лицензионного соглашения. Основные преимущества GPT-3 заключаются в высокой универсальности, гибкости настройки и способности генерировать связные, семантически корректные тексты. Модель успешно применяется в задачах генерации контента, машинного перевода, автоматического реферирования и анализа тональности.

1.3.2 GPT-4 (OpenAI)

GPT-4 (Generative Pre-trained Transformer 4) является преемником GPT-3 и представляет собой дальнейшее развитие авторегрессионной архитектуры OpenAI. 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.

1.3.3 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.

1.3.4 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 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.

1.3.5 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.

1.3.6 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.

1.3.7 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.

1.3.8 Sonar

Sonar — языковая модель, разработанная для извлечения и анализа информации из текстовых данных. Особое внимание в Sonar уделяется идентификации именованных сущно-

стей (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.

1.4 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.

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

— **Извлечение отношений (RE)**: определение семантических связей между сущностями. В предложении «Илон Маск является CEO компании SpaceX» устанавливается связь «является CEO» между «Илон Маск» и «SpaceX». Эта задача требует анализа контекста и часто зависит от результатов NER.

— **Извлечение событий**: обнаружение действий или происшествий, их участников и атрибутов. Например, в новости о кибератаке необходимо идентифицировать тип атаки (DDoS), цель (инфраструктура компании), время и используемые методы.

— **Извлечение временной информации**: распознавание временных выражений и их нормализация (например, преобразование «через две недели» в конкретную дату). Это необходимо для построения временных линий событий в аналитических системах.

Эти задачи находят применение в областях, где требуется автоматизация обработки текстов: мониторинг медиапространства, анализ юридических документов, обнаружение киберугроз. Например, в кибербезопасности извлечение IoC (Indicators of Compromise) из логов позволяет автоматизировать реагирование на инциденты.

1.4.2 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: $\backslash d\{1,2\}\backslash .\backslash d\{1,2\}\backslash .\backslash d\{4$ 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 (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 слов, Y — последовательность меток. HMM эффективны для коротких контекстов, но игнорируют глобальные зависимости между токенами.

— **Условные случайные поля (CRF)**: дискриминативные модели, оценивающие вероятность меток Y при заданном X:

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

Нормировочная функция $Z(X)$ обеспечивает 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»).

— **SVM и другие классификаторы**: применялись для независимой классификации токенов. Однако игнорирование последовательностной природы текста снижало их эф-

фективность для задач вроде NER.

Эти методы требовали тщательной инженерии признаков (например, часть речи, морфологические характеристики) и большого объёма размеченных данных. Например, 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**: 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**: 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. Например, для определения, что «Apple» в одном предложении относится к компании, а в другом — к фрукту, требовались дополнительные механизмы внимания.

1.4.3 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 vs. 6 мая) без явных указаний.

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

```
Текст: "Встреча назначена на 15:30 20 апреля."
Выход: {"время": "15:30", "дата": "2024-04-20"}
```

помогает модели корректно обрабатывать относительные временные выражения («через три дня»).

— **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.

Архитектурные особенности

Трансформерные модели применяют механизм самовнимания, вычисляющий взвешенные связи между всеми токенами:

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

где Q, K, V — матрицы запроса, ключа и значения. Это позволяет модели напрямую связывать далёкие токены, например, место события с его временем через несколько предложений.

Для IE часто используется подход Token Classification: скрытые состояния трансформера передаются в линейный слой, предсказывающий метку для каждого токена. В задачах извлечения отношений применяется схема «сущность1 + контекст + сущность2» → классификатор связи.

Сравнение с традиционными методами

LLM устраняют необходимость ручного создания правил и признаков, но требуют значительных вычислительных ресурсов для обучения. Там, где CRF обрабатывает 10К 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 (например, медицина), где каждое извлечение должно быть верифицируемо.

Ниже представлен улучшённый вариант текста с доработками для повышения логичности, последовательности изложения и ясности для специалистов, включая новичков в данной области.

—

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. Так, модель, анализирующая твит «Meeting at 5pm @cafe », должна уметь выделить временную метку, игнорируя эмодзи.

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 неоднозначность интерпретации. Так, в предложении «Apple выпустила iOS 17» модель правильно определяет «Apple» как название организации, опираясь на контекст, связанный с «iOS».

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). Для анализа длинных текстов применяется метод сегментации с перекрытием контекста, при котором документ разбивается на блоки по 512 токенов с шагом 256, что позволяет сохранить связь между сегментами.

Сравнение с альтернативными подходами

— **Системы на основе правил:** Ручное создание правил обеспечивает высокую интерпретируемость, однако разработка такого решения для многодоменных задач (например, одновременное извлечение киберугроз и юридических сущностей) требует значительных трудозатрат и времени. В этом плане BERT, обучаясь на обширных корпусах, автоматически выявляет общие паттерны, что значительно упрощает процесс.

— **Модели семейства GPT (GPT-3/4):** Несмотря на высокую генеративную способность, закрытая архитектура GPT ограничивает возможность детальной настройки и анализа внутренних механизмов модели, что противоречит требованию открытости. Кроме того, высокая стоимость API, особенно для GPT-4, затрудняет применение этой модели для анализа больших объемов данных.

— **Специализированные модели (например, spaCy):** Готовые пайплайны, предоставляемые spaCy, позволяют быстро развернуть решение, однако их точность на специализированных доменах часто ниже. Например, на юридических текстах spaCy может показывать F1-score около 78.9, в то время как дообученный BERT достигает значения

89.3.

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. Например, для извлечения названий лекарств можно использовать UMLS Metathesaurus с последующим дообучением модели.

— **Интерпретируемость результатов:** Несмотря на высокую точность, модели на основе глубокого обучения часто воспринимаются как «чёрный ящик». Для повышения интерпретируемости применяются методы, такие как LIME или SHAP, которые визуализируют вклад отдельных токенов в итоговое предсказание. Это позволяет, например, выделить слова «подписан» и «договор №» как ключевые признаки для идентификации события «заключение контракта».

1.5.4 Интеграция модели в аналитический конвейер

Выбор BERT обусловлен его совместимостью с современными NLP-стеками и возможностью легкой интеграции в существующие аналитические системы. Типичный конвейер обработки текстов включает следующие этапы:

1) **Предобработка:** Выполняется токенизация, нормализация (например, приведение дат к формату ISO) и удаление шумовых элементов, что позволяет подготовить данные к дальнейшему анализу.

2) **Инференс модели:** Тексты обрабатываются пакетно с использованием библиотек, таких как Hugging Face Transformers, что обеспечивает высокую скорость и параллельность вычислений.

3) **Постобработка:** На данном этапе осуществляется разрешение кореференции (связь местоимений с соответствующими сущностями) и валидация временных меток, что повышает качество извлеченной информации.

Для задач с жесткими требованиями к задержкам (например, мониторинг соцсетей в реальном времени) применяются оптимизированные версии BERT, такие как DistilBERT, которые

сокращают размер модели примерно на 40% при сохранении до 95% точности базовой модели.

1.5.5 Заключение и переход к реализации

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.

В следующем разделе будет подробно описана практическая реализация: подготовка датасета, дообучение модели на доменных данных и оценка качества с использованием методов кросс-валидации. Особое внимание уделяется подходам для компенсации ограничений BERT, таким как обработка длинных текстов и повышение интерпретируемости результатов.

Вывод

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)