Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет»

На правах рукописи

ГРИШИНА Любовь Сергеевна

МЕТОДЫ И АЛГОРИТМЫ ИНТЕЛЛЕКТУАЛЬНОЙ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В МЕДИЦИНСКОЙ ПРАКТИКЕ НА ОСНОВЕ ОБРАБОТКИ ЕСТЕСТВЕННЫХ ЯЗЫКОВ

2.3.1 – Системный анализ, управление и обработка информации, статистика

Диссертация на соискание ученой степени кандидата технических наук

Научный руководитель
доктор технических наук, профессор
Болодурина Ирина Павловна

Оренбург – 2024

ОГЛАВЛЕНИЕ

Введение

Актуальность темы исследования. Сердечно-сосудистые заболевания (ССЗ) возглавляют рейтинг самых главных причин смертности в мире – ежегодно умирает 17 миллионов человек. Определение рисков возникновения заболеваний и своевременная диагностика являются приоритетными направлениями развития данной отрасли. Данная проблема имеет множество инициатив по сокращению показателя смертности, главная идея которых состоит в разработке программ скрининга и ранней диагностики. Методы искусственного интеллекта (ИИ) дают обширные возможности и многофункциональные инструменты для добычи новых знаний и паттернов внутри накопленных данных медицинских информационных систем. Внедрение методов машинного обучения в существующие медико-клинические процессы позволит автоматизировать решение множества задач для обеспечения своевременной помощи пациенту.

Единая государственная информационная система в сфере здравоохранения (ЕГИСЗ) объединяет данные информационных систем различных медицинских организаций и хранит большие объемы информации. Однако, как правило, данные медицинских информационных систем (МИС) представлены слабоструктурированной информацией, потенциал которой можно использовать, опираясь исключительно на методы обработки естественных языков (Natural language processing, NLP).

Информация о посещениях пациентами поликлиник хранится в разнородных шаблонах МИС, которые адаптируются под лечащего врача, и поступают в ЕГИСЗ. Данные протоколов дополнительных обследований (кровь, ЭКГ и другие), а также данные о приеме пациентов хранятся в виде отдельных файлов и представлены в основном текстовой информацией. Извлечение только числовых показателей из них сужает возможности глубокого анализа причинно-следственных связей заболеваний, поэтому желательно использовать всю доступную информацию протоколов электронных-медицинских карт (ЭМК).

Таким образом, актуальность диссертационного исследования определяется необходимостью разработки и совершенствования методов извлечения и структурирования знаний из ЭМК для поддержки принятия решений при диагностике и лечении заболеваний.

Степень разработанности темы исследования. В области применения методов искусственного интеллекта для поддержки принятия решений в медицинской практике принято использовать подходы, продемонстрированные в работах А.Г. Хасанова, Д.А. Госмана, И.Л. Кашириной, М.В. Демченко, И. А. Мишкина, М.А. Фирюлина, М. В. Сахибгареева, Б.А. Урмашев, C. Kilgour, W. Sun, D.A. Hanauer, A. J. Graham, S. Pasha, R. Bharti и других.

Эффективность применения методов обработки естественных языков и глубокого обучения для диагностики заболеваний и анализа ЭМК подтверждена в работах Е.В. Тутубалиной, H.S. Chase, S.S. Zhao, V.M. Castro, B. Hazlehurst и

других. Однако, к настоящему моменту в современных исследованиях открытым является вопрос о разработке общего алгоритма обработки информации ЭМК пациентов для прогнозирования заболеваний и формирования рекомендаций к лечению. Для решения данной проблемы требуется, во-первых, разработать концептуальную модель анализа клинических данных и алгоритмизировать процесс извлечения текстовой информации документов МИС. Кроме того, необходимо построить модель прогнозирования заболеваний, при этом использование методов NLP и машинного обучения представляется многообещающим подходом в задачах медицинской диагностики. Далее, требуется разработать подход к автоматической генерации индивидуальных листов назначений и рекомендаций для использования результатов при автоматизации процессов заполнения документов и сокращения времени оказания медицинских услуг.

Объектом исследования является процесс формирования листов назначений и рекомендаций к лечению в медицинской практике.

Предметом исследования являются модели и алгоритмы интеллектуальной поддержки принятия решений на основе слабоструктурированных данных медицинских информационных систем.

Целью диссертационной работы является повышение эффективности принятия решений в медицинской практике на основе анализа слабоструктурированной текстовой информации электронных-медицинских карт методами обработки естественных языков.

Для достижения поставленной цели предполагается решение следующих **задач**:

1) построить концептуальную модель анализа клинических данных и поддержки принятия решений для автоматизации процессов заполнения ЭМК с учетом индивидуализации этапов оказания медицинских услуг;

2) разработать иерархическую модель данных амбулаторных карт пациентов для обеспечения семантической интероперабельности при обработке разношаблонных документов МИС;

3) разработать метод и алгоритм прогнозирования укрупненных групп заболеваний на основе слабоструктурированных текстовых данных ЭМК пациентов;

4) разработать метод и алгоритм автоматической генерации индивидуальных листов назначений и рекомендаций к лечению для автоматизации процессов заполнения документов;

5) построить прототип автоматизированного программного комплекса интеллектуальной поддержки принятия решений в медицинской практике и исследовать эффективность его применения.

Научная новизна. В диссертационной работе получены следующие результаты, характеризующиеся научной новизной:

− предложена концептуальная модель анализа клинических данных и поддержки принятия решений для автоматизации процессов заполнения ЭМК, *отличающаяся* формализацией этапов структурирования текстовых данных и

построением интеллектуальных моделей формирования рекомендаций к лечению диагностированных заболеваний;

− разработана иерархическая модель данных амбулаторных карт пациентов, *отличающаяся* возможностью обработки разношаблонных XML-документов МИС на основе рекурсивного подхода для обеспечения семантической интероперабельности;

− разработаны метод и алгоритм прогнозирования группы заболеваний пациентов на основе методов обработки естественных языков и машинного обучения, *отличающиеся* применением уникального узкоспециализированного корпуса текстов, построенного на основе слабоструктурированных текстовых данных ЭМК;

− разработаны метод и алгоритм автоматической генерации индивидуальных листов назначений и рекомендаций к лечению для автоматизации процессов заполнения документов, *отличающиеся* применением современных предобученных нейросетевых моделей трансформеров;

− построен прототип автоматизированного программного комплекса интеллектуальной поддержки принятия решений в медицинской практике, *отличающийся* применением модулей искусственного интеллекта для диагностирования заболеваний и формирования рекомендаций к лечению на основе методов обработки естественных языков.

Теоретическая значимость диссертационной работы заключается в разработке алгоритмов обработки слабоструктурированной текстовой информации разношаблонных документов информационных систем, а также построении узкоспециализированных языковых моделей, построенных на основе методов обработки естественного языка.

Практическая значимость диссертационной работы заключается в разработке программного комплекса, позволяющего производить автоматизированный анализ состояния пациента и генерацию индивидуального листа назначений и рекомендаций к лечению на основе методов глубокого обучения. Разработанные алгоритмы прошли апробацию на множестве реальных деперсонализированных данных электронных медицинских карт, полученных из базы данных медицинских организаций Оренбургской области.

Методы исследования. Для решения поставленных задач использовались методы системного анализа, обработки информации, методы машинного обучения, нейросетевые технологии, методы обработки естественного языка, методы глубокого обучения.

Основные положения, выносимые на защиту:

1. Концептуальная модель анализа клинических данных и поддержки принятия решений для автоматизации процессов заполнения ЭМК.

2. Иерархическая модель данных амбулаторных карт пациентов для сбора информации из разношаблонных XML-документов.

3. Метод прогнозирования укрупненной группы заболеваний пациентов на основе алгоритмов обработки естественных языков и модели логистической регрессии.

4. Метод автоматической генерации индивидульных шаблонов листа назначений и рекомендаций к лечению на основе алгоритмов обработки естественных языков и модели глубокого обучения GPT-3.

5. Структура интеллектуальной системы поддержки принятия решений (СППР) диагностики заболеваний и формирования рекомендаций к лечению пациентам.

Область исследования соответствует паспорту специальности 2.3.1. – «Системный анализ, управление и обработка информации, статистика», а именно: п. 2 – Формализация и постановка задач системного анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта; п. 5 – разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта; п. 12 – визуализация, трансформация и анализ информации на основе компьютерных методов обработки информации.

Внедрение результатов работы. Материалы диссертации в форме СППР внедрены в практику медицинских исследований организационно-методического отдела ГАУЗ «Оренбургской областной клинической больницы имени В.И. Войнова» и ГАУЗ «Бузулукской больницы скорой медицинской помощи им. академика Н.А. Семашко». Теоретические результаты диссертационной работы внедрены в учебный процесс ФГБОУ ВО «Оренбургского государственного медицинского университета» на кафедре «Общественного здоровья и здравоохранения №1».

Основные результаты диссертационного исследования представлялись и докладывались на научных конференциях: Всероссийская научно-методическая конференция «Университетский комплекс как региональный центр образования, науки и культуры» (Оренбург, 2023), Международная научно-техническая конференция "Перспективные информационные технологии" (Самара, 2022); Международный семинар «Вычислительные технологии и прикладная математика» (International Workshop on Computing Technologies and Applied Mathematics) (Владивосток, 2022); 2nd International Scientific and Practical Conference "Modern Management Trends and the Digital Economy: from Regional Development to Global Economic Growth" MTDE (Екатеринбург, 2020).

Публикации. Основные результаты диссертации опубликованы в 8 научных работах, в том числе 3 − в изданиях, рекомендованных ВАК РФ и отечественных изданиях, которые входят в международные базы данных и системы цитирования, а также 2 работы − в изданиях, индексируемых Scopus и Web of Science, получено 1 свидетельство о государственной регистрации программ для ЭВМ.

Личный вклад автора. В работах, опубликованных в соавторстве, лично автором получены следующие результаты: [2, 5, 7] – исследование подходов к построению моделей данных информационных систем для их интеллектуальной обработки; [3, 6] – разработка подхода к прогнозированию заболеваний на

основе методов NLP и машинного обучения; [1, 8] – исследование современных архитектур генерации русскоязычного текста и разработка алгоритма формирования медицинских рекомендаций на их основе; [4, 9] – реализация основных компонентов программного комплекса интеллектуальной поддержки принятия решений в медицинской практике.

Структура и объем диссертации. Диссертация состоит из введения, 5 глав с выводами, заключения, приложений и списка литературы из 107 наименований. Основная часть работы изложена на 114 страницах, содержит 38 рисунков, 17 таблиц и 5 приложений.

Глава 1. Исследование алгоритмического инструментария поддержки принятия решений при диагностике и лечении заболеваний на основе машинного обучения

В главе представлен обзор современных инструментов и результатов исследования и обоснования алгоритмического инструментария для решения задачи прогнозирования групп заболеваний и генерации индивидульных листов назначений и рекомендаций к лечению, описана проблематика обработки слабоструктурированных данных МИС при решении задач поддержки принятия решений, сформулированы цели и задачи исследования.

1.1. Современное состояние проблемы автоматизированного формирования протоколов ЭМК для МИС

Благодаря высокому уровню цифровизации, проникающей во все сферы жизнедеятельности, стало возможным накапливать, обрабатывать и анализировать информацию для дальнейшего извлечения принципиально новых и практически полезных знаний. С одной стороны, сопровождение документооборота различных организаций позволило автоматизировать множество процессов, сократить время обслуживания и повысить эффективность работы в целом. С другой стороны, в области здравоохранения процесс цифровизации характеризуется еще и наличием большого количества периферийных устройств (медицинских датчиков и приборов), которых с каждым годом появляется все больше и при этом они генерируют цифровую информацию сложной структуры (изображения, видеофрагменты, сигналы, текстовая информация, омиксные данные, метаданные) [1, 2]. В следствие чего, одной из главных проблем, связанных с обработкой медицинских данных, является обеспечение интеграции разнородных медицинских данных (интероперабельность) [3]. В рамках стратегии научно-технологического развития Российской Федерации (Указ Президента РФ от 28.02.2024 № 145) переход к персонализированной, предиктивной и профилактической медицине занимает важное место. Одной из основных стратегий является развитие цифровой инфраструктуры и переход к высокотехнологичному здравоохранению, включая создание единой электронной медицинской карты, обмен информацией между медицинскими учреждениями, использование телемедицины, а также развитие и внедрение искусственного интеллекта и аналитических систем. Данный подход позволит улучшить диагностику, предотвратить развитие серьезных заболеваний и приведет к повышению качества и доступности медицинской помощи.

Нормативно-правовое обеспечение данного процесса закреплено в Указе Президента РФ от 06.06.2019 г. № 254 "О Стратегии развития здравоохранения в Российской Федерации на период до 2025 года" и Приказе Минздрава России от 07.09.2020 № 947н "Об утверждении Порядка организации системы

документооборота в сфере охраны здоровья в части ведения медицинской документации в форме электронных документов". Данные нормативные акты запустили переход на электронный документооборот внутри медицинской организации без использования бумажных форм и позволили накапливать информацию. В результате реализации закрепленных нормативных актов разработана Единая государственная информационная система в сфере здравоохранения (ЕГИСЗ), которая объединяет данные информационных систем различных медицинских организаций и хранит большие объемы информации.

Одновременно с этим, стали активно развиваться технологии обработки больших данных (Big Data) и методы их интеллектуального анализа. Среди медицинских аналитических задач, которые можно решать с применением данного аппарата, можно выделить описательную аналитику [4], диагностическую аналитику [5], предиктивную аналитику [6] и предписывающую аналитику [7]. При увеличении сложности задач возрастает сложность аналитических процедур и алгоритмов, а также может потребоваться и большее количество входных источников информации - от данных объективных осмотров и жалоб пациентов из медицинских записей и биометрических данных до геномных данных и информации о наследственности. Однако, построенные модели искусственного интеллекта могут быть использованы для поддержки принятия врачебных решений и позволят оптимизировать процессы лечения и повысить эффективность работы медицинских учреждений.

Медицинские информационные системы разработаны и внедрены в действующие медицинские учреждения с целью автоматизации документооборота и позволяют проводить сбор, хранение, обработку и обмен финансовой и административной информацией, данных медицинских исследований в цифровой форме, электронных медицинских карт (ЭМК) пациентов, а также существующих систем поддержки принятия врачебных решений (СППВР).

Грамотная разработка и использование МИС позволяет создать единую базу данных о пациентах, которая включает в себя информацию о медицинской истории, диагнозах, назначениях, лекарствах и прочих медицинских процедурах. Данный подход упрощает работу врачей и медсестер, позволяет быстро получить доступ к необходимой информации и принимать клинически обоснованные решения. Кроме того, МИС позволяют оптимизировать процессы лечения, уменьшить время на оформление документов и свести к минимуму возможность ошибок при назначении лекарств. Однако, как правило, данные медицинских информационных систем представлены слабоструктурированной информацией, потенциал которой можно использовать, опираясь исключительно на методы обработки естественных языков.

Интеллектуальная система поддержки принятия решений при диагностике и лечении заболеваний – это информационная система, обеспечивающая путем анализа данных с использованием технологий искусственного интеллекта информационное сопровождение врача при

выставлении диагноза и генерации листа назначений и рекомендаций к лечению с целью снижения возможности возникновения врачебных ошибок и повышения качества оказываемой медицинской помощи.

Среди современных систем поддержки принятия решений для диагностики и лечения заболеваний можно выделить ряд разработок.

- «SberMedAi» - комплексная платформа сервисов по ППВР на основе технологий искусственного интеллекта. Позволяет проводить предварительную диагностику COVID-19 по наличию симптомов и записи голоса, диагностику кожных заболеваний на основе анализа изображений и многое другое (https://sbermed.ai/).

- «Webiomed» - платформа прогнозной аналитики и управления рисками, которая предоставляет доступ к группе моделей машинного обучения для решения задач, среди которых оценка развития ССЗ, смерти от ишемической болезни сердца и инсульта (https://webiomed.ru/).

- «MeDiCas» - система для дистанционной диагностики и мониторинга заболеваний на базе искусственного интеллекта, которая позволяет проводить оценку хронический заболеваний и жизнеугрожающих состояний, а также симптомов ССЗ и COVID-19 (http://medicase.newdiamed.ru/).

- «CoBrain-Аналитика» - система для обработки медицинской информации о головном мозге человека, которая осуществляет постановку диагнозов, разрабатывает индивидуальные схемы лечения (https://rusneuro.net/).

Однако следует отметить, что среди данных разработок отсутствуют полноценные универсальные системы диагностирования заболеваний и генерации шаблонов листов назначений, которые, во-первых, не ограничивались бы применимостью моделей к относительно структурированной информации, а во-вторых, грамотно учитывали состояние здоровья пациента и позволяли бы производить генерацию индивидуальных шаблонов рекомендаций к лечению.

Выделяется несколько основных подходов к разработке систем поддержки принятия решений для диагностирования заболеваний и генерации шаблонов листов назначений.

1. Информационно-справочные системы (ИСС) поддержки принятия решений.

Медицинские информационно-справочные системы предназначены для поиска и выдачи информации по запросу пользователя (пациента, врача и лиц, принимающих участие в административных и финансовых процессах). Как правило, базы данных ИСС представляют собой массивы медицинской справочной информации различного характера и не имеют функционала для обработки и анализа информации.

Согласно исследованию [8] Г.С. Лебедева и Ю.Ю. Мухина ИСС представляют собой банки медицинской информации для обслуживания медицинских учреждений и служб управления здравоохранением, и при этом авторы отмечают, что они являются дорогостоящими в разработке и внедрении, а также требуют постоянного обновления и поддержки.

В рамках диссертационного исследования для разработки ИСППР в рамках МИС достаточно реализовать поддержку минимального набора справочников:

- справочник международной классификации болезней (МКБ-10);
- справочник медицинских услуг (для определения порядка диагностических, лечебных и консультативных мероприятий).

2. Медицинские экспертные системы (МЭС) поддержки принятия решений.

Медицинские экспертные системы используются в медицине для помощи в принятии решений в области диагностики и лечения пациентов [9]. МЭС основаны на знаниях и опыте медицинских экспертов и позволяют автоматизировать процесс принятия решений врачами. В связи с этим, ключевыми компонентами МЭС являются база знаний и механизм логических выводов.

В настоящий момент построено множество примеров эффективного использования экспертных систем в диагностике заболеваний. Мишкин И.А. в работе [10] представил систему ранней диагностики соматических заболеваний, в исследовании [11] Госман Д.А. построил систему оценки риска туберкулёза, а в статье [12] авторами разработана нечеткая экспертная система для оценки риска развития профессиональных заболеваний. Результаты исследований демонстрируют основные достоинства МЭС – высокая точность и производительность построенных решений, а также прозрачность и интерпретируемость полученных результатов.

Однако, несмотря на опыт успешной реализации, МЭС имеют существенные недостатки:

- Ограниченность области применения: могут быть применены только в тех областях, где есть достаточное количество знаний и опыта экспертов в формализованном виде.
- Необходимость постоянного обновления: знания и опыт экспертов могут устареть со временем, поэтому МЭС требуют постоянного обновления и поддержки.
- Ограниченность вариативности решений: экспертные системы могут предлагать только те решения, которые заложены в их базу знаний, что может ограничивать вариативность решений.

Выделенные недостатки могут быть скорректированы в рамках разработки систем поддержки принятия решений, основанных на первичных данных, а не на знаниях. Как правило, МЭС используют предиктивные модели машинного обучения, которые решают задачи, исходя из анализа доступной информации и апробации подходов к решению множества схожих задач. Следовательно, не требуют накопления базы знаний экспертов и могут быть обновлены в рамках обучения на новых данных по заранее определенной стратегии.

3. Системы поддержки принятия решений, построенные на основе методов искусственного интеллекта.

Интеллектуальные системы поддержки принятия решений на базе методов искусственного интеллекта и машинного обучения проводят анализ данных и построение моделей на основе обучения на данных и обратной связи.

Инструменты искусственного интеллекта, такие как нечеткая логика, методы эволюционной оптимизации, статистические методы машинного обучения и искусственные нейронные сети могут быть интегрированы с ИСППР для диагностики и лечения в здравоохранении.

Алгоритмы машинного обучения предложены в качестве мощного инструмента для обработки данных ЭМК для различных задач, включая прогнозирование результатов, оценку риска для пациента и диагностику заболеваний [13].

Так, авторы исследования [14] рассмотрели метааналитическую методологию для оценки прогностической способности алгоритмов машинного обучения для сердечно-сосудистых заболеваний. В рамках данной работы проанализировано применение свёрточных нейронных сетей, метода опорных векторов и алгоритма градиентного бустинга для прогнозирования ишемической болезни сердца и риска инсульта. Результаты экспериментов подтвердили эффективность внедрения моделей ИИ в медицинскую практику.

В статье [15] проведен сравнительный анализ точности методов машинного обучения, таких как метод опорных векторов, алгоритм деревьев решений, алгоритм k-ближайших соседей и простейшая искуственная нейронная сеть (ИНС) для оценки риска ишемической болезни сердца и построения соответствующих моделей бинарной классификации. Наиболее высокую точность диагностики наличия заболеваний ССЗ (около 85% для задачи бинарной классификации) продемонстрировала ИНС.

Применение методов глубокого обучения для прогнозирования наличия заболеваний ССЗ проведено в работе [16] на основе янглоязычного набора данных UCI Machine Learning Heart Disease. Экспериментальные исследования показали высокую точность бинарной классификации – 94%, используя всего 14 признаков описания состояния пациента.

Эффективность методов глубокого обучения, таких как сверточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN), проанализирована в исследовании [17] в задаче прогнозирования ССЗ для пациентов с COVID-19. Авторы статьи подчеркивают эффективность применения построенных моделей для выявления ранней стадии заболевания.

Однако использование моделей машинного обучения для выполнения этих задач сопряжено с двумя серьезными проблемами. С одной стороны, характеристики структурированных данных могут различаться у разных пациентов в зависимости от их заболеваний и сроков пребывания. Кроме того, структурированные медицинские данные требуют выбора признаков в качестве механизма регуляризации данных для стандартных моделей, таких как нейронные сети, деревья решений и другие [18]. Этот процесс нормализации отсекает различные признаки и резко сокращает выборки данных.

С другой стороны, неструктурированные и слабоструктурированные данные, такие как описательные отчеты, могут иметь информационный шум и сложную структуру, в связи с чем требуют применения алгоритмов обработки естественного языка.

В связи с этим, методы NLP обработки естественного языка активно применяются для решения различных задач анализа электронных медицинских карт, в частности для распознавания рассеянного склероза [19], аксиального спондилоартрита [20], гепатоцеллюлярного рака [21], сахарного диабета [22], биполярного расстройства [23], выявления передозировок, связанных с опиоидами [24], диагностики инфекционных заболеваний [25], выявления пациентов с метастатическим раком молочной железы [26] и др.

Исследование [27] представило возможности инструментов обработки естественных языков для создания классификатора на основе англоязычной текстовой информации выписки по четырем группам критических заболеваний. Точность модели NLP по выписным документам пациентов с диагнозом острый респираторный дистресс-синдром составила 95%. Применение методов NLP также продемонстрировано в работе [28], но относительно задачи прогнозирования ранней психиатрической реадмиссии по содержанию выписки. Точность классификации на основе англоязычного текста выписки составила 78,4%, при этом модель имела повышенную специфичность.

В рамках статьи [29] представлены эффективные модели NLP для оценки наличия важных сопутствующих сердечно-сосудистых заболеваний в медицинских картах с произвольным текстом. Прогностическая способность моделей оценена в 85%, а самая высокая точность получена для состояний с большей диагностической ясностью (диабет, гипертония и др.).

При этом, хотя модели NLP продемонстрировали впечатляющую производительность в извлечении информации и представлении данных, клинические описания и отчеты ЭМК особенно сложны для данных моделей, связанных с прогнозированием заболеваний. Отчасти это обусловлено тем, что применение NLP к таким данным требует определенных стандартов, поскольку представление документов опирается на справочники и словари распространенных естественных языков [30].

Таким образом, проведенный обзор современных исследований показывает ограниченные возможности доступных моделей к масштабированию результатов для других МИС ввиду специфики выбранного языка построенных моделей и описания состояния пациента в ЭМК в соответствии с нормативной базой отдельных медицинских учреждений. В связи с этим, на сегодняшний момент является актуальной задача построения системы поддержки принятия решений при диагностике и лечении заболеваний на основе текстовой информации жалоб пациента на приеме у врача с применением методов NLP.

При этом, применение методов обработки естественного языка в медицине затруднено из-за специфики данной предметной области. Медицинский текст содержит большое количество терминов, сокращений, сложных концепций и контекстуальных нюансов, что делает его сложным для автоматического анализа

с использованием стандартных методов NLP. Кроме того, важно учитывать конфиденциальность и безопасность данных пациентов при работе с медицинской информацией, что также создает дополнительные вызовы для разработки и применения технологий искусственного интеллекта в этой области.

1.2. Исследование подходов к анализу медицинских документов и построения моделей машинного обучения на основе методов обработки естественных языков

Технологии обработки естественного языка и машинного обучения разработаны для интеллектуального анализа и понимания текстов и документов. С учетом того, что примерно 80% корпоративных данных неструктурированы, методы NLP в настоящий момент стали неотъемлемой частью для проектов цифровой трансформации.

Опишем подробно процесс построения моделей ИИ для поддержки принятия врачебных решений на основе анализа данных МИС с помощью методов NLP (рисунок 1.1). Специфика данных МИС состоит в том, что информация о посещениях пациентами поликлиник хранится в разнородных шаблонах, которые адаптируются под лечащего врача. Данные протоколов дополнительных обследований (кровь, ЭКГ и другие), а также данные о приеме пациентов хранятся в виде отдельных XML-файлов и представлены в основном текстовой информацией.

Диссертационное исследование направлено на построение модели прогнозирования укрупненных групп заболеваний на основе текстовой информации жалоб пациента на приеме у врача из ЭМК и автоматической генерации индивидуальных листов назначений и рекомендаций к лечению с применением методов NLP. В связи с этим, можно выделить следующие этапы предварительной подготовки данных для проведения моделирования:

1. Предварительный анализ документов

Необходимо проанализировать количество пациентов медицинских организаций, обращающихся за медицинским обслуживанием, и сформировать репрезентативную выборку пациентов, связанных схожими заболеваниями и/или характеризующих наиболее актуальные примеры оказания мед. помощи.

2. Выделение информативных блоков

Для предоставленных XML-документов необходимо проводить автоматическое распознавание наиболее информативных блоков, пригодных для построения моделей ИИ. Характерной проблемой данного этапа является наличие документов различной структуры – ввиду возможности коррекции врачом шаблона протоколов посещений, индивидуальных концепций заполнения информации о проводимых дополнительных обследованиях мед. лабораторий и т.д. Таким образом, необходимо выработать единый подход к обработке разношаблонных документов и их информативных блоков текстовой информации для обеспечения семантической интероперабельности.

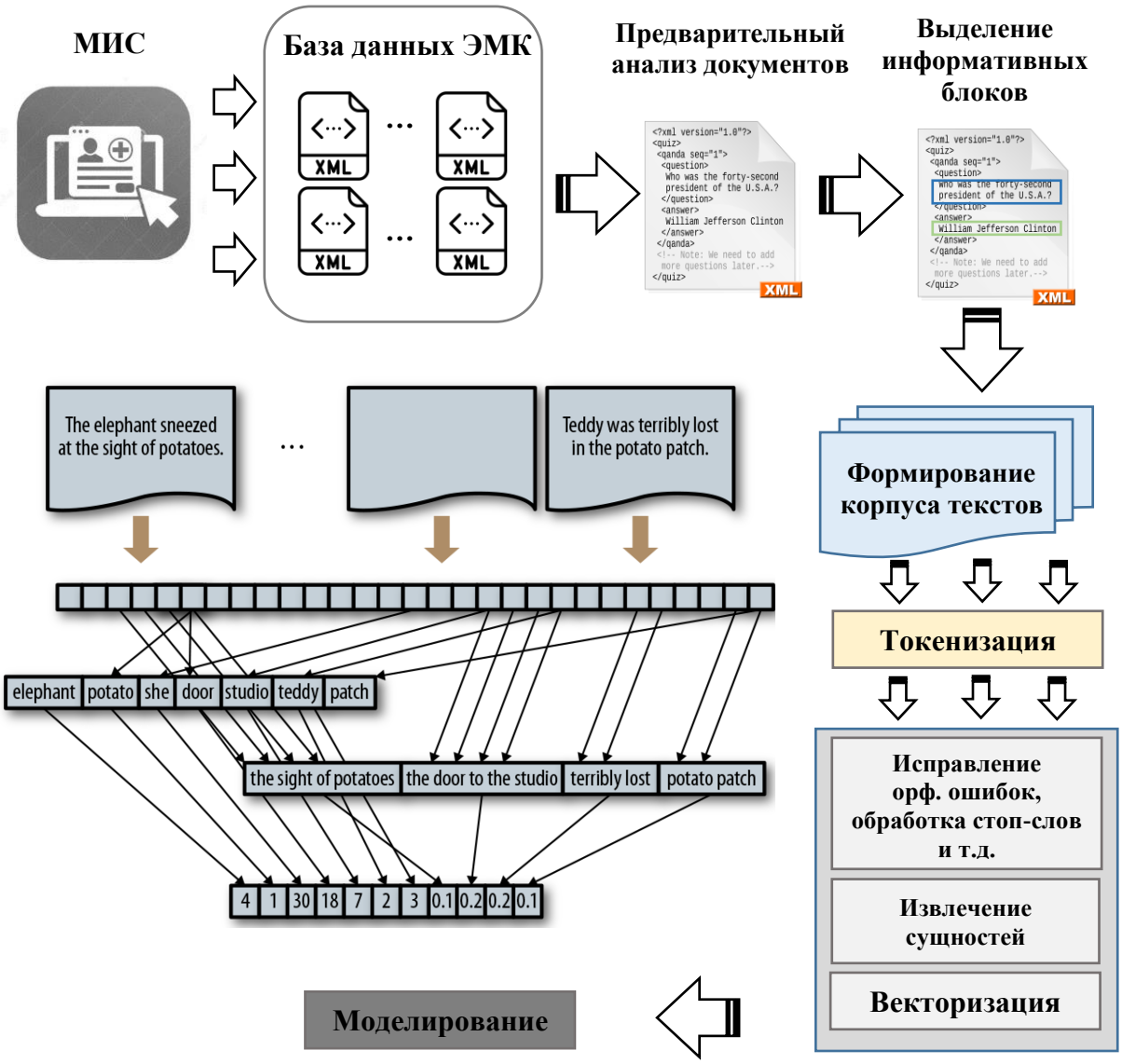

Рисунок 1.1 – Схема процесса построения моделей ИИ на основе анализа данных МИС с помощью методов NLP

3. Формирование корпуса текстов

На основе подходов к анализу и извлечению практически полезной текстовой информации из разношаблонных документов необходимо сформировать корпуса текстов для обучения моделей ИИ.

4. Токенизация текста

Токенизация текста - это процесс разделения текста на отдельные слова или токены. Данный этап необходим для анализа текста, включая его обработку и классификацию. Токенизация позволяет компьютеру лучше понимать текст и выполнять задачи, такие как поиск ключевых слов, анализ тональности и определение частей речи.

Существуют различные подходы к токенизации текста:

- Разбиение по пробелам.
- Разбиение по знакам препинания.

- Использование регулярных выражений для разбиения текста на токены.
- Использование известных словарей для разбиения текста на токены.
- Использование алгоритмов машинного обучения для автоматического разбиения текста на токены.

5. Векторизация текста

Векторизация текста – это процесс преобразования исходного текста, представленного токенами в числовые векторы, которые могут быть использованы для построения моделей ИИ. Данный процесс реализуется для того, чтобы алгоритмы машинного обучения могли работать с текстом, который представлен в виде векторов чисел.

Существуют различные подходы к векторизации текста:

- Мешок слов (Bag of words) – векторизация, которая учитывает только наличие или отсутствие слов в тексте, но не их порядок и контекст.
- TF-IDF (Term Frequency-Inverse Document Frequency) – метод векторизации, который учитывает частоту использования слов в документе и в корпусе документов.
- Word2Vec – метод векторизации, который использует нейронные сети для преобразования слов в числовые векторы, учитывая их контекст и семантическое значение.
- GloVe (Global Vectors) – метод векторизации, который использует матрицу совместной встречаемости слов для создания числовых векторов.
- FastText – метод векторизации, который учитывает не только слова, но и их подслова (n-граммы), что позволяет учитывать морфологические особенности языка.

Рассмотренные этапы предварительной подготовки данных для проведения моделирования реализованы на основе деперсонализированных данных МИС в виде протоколов посещений пациентами медицинских учреждений по Оренбургской области, которые предоставлены Медицинским информационно-аналитическим центром (МИАЦ) города Оренбурга. Исходный набор данных построен по историям болезней пациентов, имеющим ССЗ, и сформирован путем преобразования XML-файлов и извлечения в хронологическом порядке жалоб больного.

1.3. Обзор современных инструментов для прогнозирования заболевания и генерации индивидуальных листов назначений и рекомендаций

Данное диссертационное исследование направлено на повышение эффективности принятия решений в медицинской практике путем разработки и совершенствования методов извлечения и структурирования знаний из ЭМК при диагностике и лечении заболеваний. Для осуществления поддержки принятия

решений при прогнозировании заболеваний и генерации персонализированных листов назначений и рекомендаций важно отметить, что данные электронных-медицинских карт в основном представлены слабоструктурированной текстовой информацией. В связи с этим современные исследования в области медицины все более активно используют методы обработки естественного языка.

Клиническая классификация текстов является фундаментальной проблемой обработки естественного языка в медицине, потому что позволяет решать задачи прогнозирования заболеваний. Существующие исследования традиционно сосредоточены на разработке признаков на основе правил или экспертных знаний, и только в ограниченном количестве работ используются возможности машинного бучения.

Авторы исследования [31] K. R. Anantha Padmanaban и G. Parthiban представили модели машинного обучения для прогнозирования риска хронической болезни почек. В качестве базовых алгоритмов рассмотрены деревья решений и наивный байесовский классификатор, качество предсказания составило 91%. Кроме того, для повышения точности прогнозирования рассматривалось также применение ИНС и алгоритмов кластеризации данных.

Метод опорных векторов, наивный байесовский классификатор и метод модифицированного экстремального обучения для прогнозирования заболеваний на основе отчетов о радиологии и патологии представлены в работе [32]. Наиболее высокую точность продемонстрировала модель наивного байесовского классификатора по F-мере - 81%.

В работе [33] проводилось прогнозирование болезни Паркинсона на основе алгоритма k-ближайших соседей, ИНС и алгоритма адаптивного бустинга. Авторы продемонстрировали гибридный подход, который имеет точность - 91,28%.

Исследование [34] рассматривает проблему решения задачи оптимизации стратегий лечения пациентов с атеросклерозом с помощью моделей машинного обучения, который основан на выделении ключевых групп состояний пациентов через кластерный анализ.

Существует несколько подходов к формированию векторного представления текста с учетом контекстной информации, а также методов отнесения текста к некоторому классу при заданной последовательности слов.

Так, в работе [35] рассмотрена возможность применения логистической регрессии, метода опорных векторов и ИНС для англоязычных текстов отзывов пациентов о четырех лекарственных препаратах. Полученные результаты продемонстрировали точность от 60,8% до 77,6% прогнозирования положительной или отрицательной оценки лекарства по отзыву пациента, что свидетельствует о потенциальных перпективах применения методов NLP для построения СППВР.

Кроме того, высокую эффективность применения в сфере обработки медицинских текстов продемонстрировали методы глубокого обучения. Авторы исследования [36] провели адаптацию современных языковых моделей для извлечения информации о пациенте из англоязычных клинических рассказов в

свободной форме для автоматического заполнения форм. Для фильтрации нерелевантной информации получена оценка F1-score 81,1%, что является довольно значимым показателем.

С другой стороны, огромную важность при разработке и интеграции технологий высокотехнологичного здравоохранения имеет решение задачи генерации индивидуализированных листов назначений и рекомендаций. Во-первых, данный подход позволяет сократить время на подготовку документации и способствует повышению удовлетворенности от качества медицинского обслуживания. Во-вторых, автоматизированная генерация листов назначений уменьшает вероятность ошибок в выборе лекарств, дозировках и схемах лечения, что способствует безопасности пациентов. Однако, как входные, так и выходные данные протоколов приема пациентов в МИС представлены слабоструктурированной текстовой информацией. В связи с тем, что стандартные экспертные методы и статистические алгоритмы машинного обучения малоэффективны при решении задач анализа текста, необходимо рассматривать группу методов обработки естественных языков и глубокого обучения.

Современные исследования анализа текста в сфере здравоохранения сфокусированы на применении языковых моделей, основанных на трансформерах и контекстуализированных эмбеддингах. Данный подход используется для представления слов с несколькими значениями в зависимости от контекста, в котором они используются в предложении. Так как задача генерации медицинского текста осложняется наличием множества узкоспециализированных терминов, необходима реализация языковых моделей генерации, требующих использования большого объема вычислительных ресурсов. Многообещающим подходом к обучению ресурсоёмких глубоких нейронных сетей является использование ресурсов графических процессоров (GPU).

В рамках диссертационного исследования [37] рассмотрена проблема обработки естественного языка для формирования схем лечения. Демченко Е.В. предложила алгоритм отбора значимых признаков и поиска клинических взаимосвязей, а также алгоритм мониторинга состояний и назначения лечения на основе машинного обучения с подкреплением.

Rasmy и др. [38] представили модель контекстуализированных эмбеддингов Med-BERT, предварительно обученная на структурированном наборе данных, содержащим более 28 миллионов записей ЭМК пациентов. В работе [39] Li и др. провели обзор современных подходов к обработке неструктурированных медицинских текстов, которые отличаются от традиционных статистических систем и систем на основе правил. В исследовании [40] Syed и др. описывают гибридную архитектуру ИНС с комбинированными контекстуализированными эмбеддингами моделей BERT и FLAIR для решения задачи диагностики колоректального рака.

Авторами работы [41] предложен подход к автоматической разработке и ранжированию большого корпуса для генерации русских парафраз. Тексты для

обучения не имели определенной специфики, однако полученные результаты исследования могут быть распространены на другие генеративные задачи. В работе [42] авторы представили языковую модель GatorTron – с использованием более 90 миллиардов слов и систематически провели оценку по пяти клиническим задачам NLP, включая извлечение клинических понятий, извлечение медицинских отношений, семантическое текстовое сходство, вывод на естественном языке и ответы на медицинские вопросы. Результаты исследований показали, что увеличение количества параметров и размера обучающих данных может повысить эффективность решения поставленных задач.

В работе [43] исследователи представили BioBERT, которая представляет собой предварительно обученную модель языкового представления для биомедицинской области. Авторы дообучили BERT на корпусах биомедицинских доменов и сравнили эффективность обучения полученной модели BioBERT на трех распространенных задачах биомедицинского анализа текста (распознавания сущностей, извлечении соотношений и ответа на вопросы), тестируя стратегии предварительного обучения с различными комбинациями и размерами корпусов общих предметных областей и биомедицинских корпусов, и проанализировали влияние каждого корпуса на предварительное обучение.

В исследовании [44] авторы получили искусственные медицинские данные, используя современные модели генерации текста. Для сохранения семантической связности абзацев, предложение за предложением формируется с помощью ключевых фраз.

Авторы работы [45] улучшили определение суицидальных исходов с помощью обработки естественного языка, разработав методологию поиска информации из более чем 200 миллионов записей ЭМК. Суицидальные термины извлечены с помощью word2vec. В результате проверки на 200 извлеченных пациентов модель показала высокую эффективность в отношении суицидальных мыслей AUROC: 98,6.

В работе [46] авторами определены клинические и гистопатологические характеристики 14 436 пациентов с меланомой кожи с использованием алгоритма обработки естественного языка для создания группы из 2624 пациентов с меланомой в фазе вертикального роста и степенью TIL. Анализ выживаемости по Каплану-Мейеру и многопараметрический анализ показали, что активные TIL значительно связаны с улучшением общей выживаемости (преимущество 14% через 5 лет) по сравнению с отсутствием TIL.

Проводятся активные исследования в области NLP для русского языка. В статье Ялунина и др. [47] представлены модели RuBioBERT и RuBioRoBERTa для анализа биомедицинских текстов на русском языке. В статье Блинова и др. [48] описывается бенчмарк понимания русского медицинского языка, частично решая проблему отсутствия универсального медицинского датасета. В задаче интеллектуального анализа клинического текста решают проблему обнаружения отрицаний [49] и автоматической коррекции орфографии [50].

В рамках диссертации Тутубалиной Е.В. [51] представлены новые модели и методы для классификации и извлечения информации, которые включают использование мультимодальных моделей и технологий трансферного обучения. Рассмотрено применение разработанных подходов к решению задач анализа текстов отзывов пользователей на лекарства.

Таким образом, в сфере здравоохранении существуют решения для автоматизации и поддержки принятия решения на основе методов искусственного интеллекта с применением NLP. Эффективные методы генерации текстовых данных в области медицины показывают сопоставимые результаты с экспертными системами, что показывает возможность применения данных алгоритмов для задачи генерации рекомендаций пациентам.

1.4. Проблематика обработки слабоструктурированных данных МИС при решении задач поддержки принятия решений

Современные исследования показали, что для построения интеллектуальной системы поддержки принятия решений при диагностике и лечении заболеваний возможно эффективное использование методов машинного обучения и обработки естественных языков. Однако, в связи с тем, что система ЕГИСЗ автоматизации процессов управления и обеспечения качественного медицинского обслуживания имеет сложную техническую инфраструктуру, можно выделить ряд проблем, которые появляются ввиду необходимости обработки слабоструктурированных данных:

1. *Отсутствие готовых протоколов подключения к МИС для сбора обезличенных данных.* В настоящий момент медицинские организации устанавливают частные МИС, которые имеют различную архитектуру сбора и хранения данных и требуют разработку индивидуальных процедур обезличивания информации, что затрудняет построение модулей сбора данных для обучения моделей.
2. *Хранение данных протоколов оказания медицинских услуг в разношаблонных документах.* Врач вправе лично корректировать структуру протокола оказания медицинской услуги, лаборатории дополнительных обследований могут индивидуализировать форму отчета и загружать ее в МИС. В связи с этим, требуется разработка алгоритмов автоматической обработки разношаблонных документов для выделения наиболее информативных блоков. При этом, готовых открытых инструментов реализации анализа разношаблонных документов в настоящий момент нет.
3. *Сырые данные протоколов могут содержать опечатки и не корректные сокращения.* Человеческий фактор при заполнении протоколов может спровоцировать орфографически некорректное описание, что затрудняет работу методов обработки естественных языков и требует реализации подходов к выявлению и исправлению такого текста.

4. *Сырые данные протоколов могут содержать неоднозначные диагнозы.* Ввиду наличия спорных симптомов заболевания у пациента, которые могут относиться к нескольким группам заболеваний, врач, основываясь на своем опыте, может выставлять протоколу диагнозы по международному классификатору МКБ-10 неоднозначно.

5. *Многовариативность выхода моделей генерации текста.* Листы назначений и рекомендаций к лечению не имеют четкой структуры, врач описывает заключение в свободной форме, которая может отличаться даже в рамках одного диагноза. Таким образом, проблема генеративных моделей ИИ, построенных на базе сырых данных МИС заключается в многовариативности выходов моделей и сложностях оценки результатов генерации.

6. *Наличие узкоспециализированных терминов.* Базовые языковые модели не содержат в словарях информации о терминах, относящихся к узкоспециализированным предметным областям и требуют дообучения. Кроме того, некоторые термины в зависимости от контекста могут иметь различный семантический смысл, что также усложняет поставленную задачу.

7. *Необходимость использования большого объема вычислительных ресурсов.* Модели ИИ, способные обрабатывать данные на естественных языках, как правило, используют огромное число настраиваемые параметров. Обучение подобных моделей машинного обучения требует большого объема вычислительных ресурсов для проведения операций обучения (обновления параметров), что увеличивает и расчётное время, и размер модели.

Таким образом, при построении интеллектуальных систем поддержки принятия врачебных решений при диагностировании заболеваний и генерации рекомендаций необходимо разработать подходы и алгоритмы решения множества открытых проблем.

1.5. Цель и задачи исследования

Целью диссертационной работы является повышение эффективности принятия решений в медицинской практике на основе анализа слабоструктурированной текстовой информации электронных-медицинских карт методами обработки естественных языков.

Для достижения поставленной цели предполагается решение следующих задач:

1) построить концептуальную модель анализа клинических данных и поддержки принятия решений для автоматизации процессов заполнения ЭМК с учетом индивидуализации этапов оказания медицинских услуг;

2) разработать иерархическую модель данных амбулаторных карт пациентов для обеспечения семантической интероперабельности при обработке разношаблонных документов МИС;

3) разработать метод и алгоритм прогнозирования укрупненных групп заболеваний на основе слабоструктурированных текстовых данных ЭМК

пациентов;

4) разработать метод и алгоритм автоматической генерации индивидуальных листов назначений и рекомендаций к лечению для автоматизации процессов заполнения документов;

5) построить прототип автоматизированного программного комплекса интеллектуальной поддержки принятия решений в медицинской практике и исследовать эффективность его применения.

Выводы первой главы

1. Интеллектуальная система поддержки принятия решений при диагностике и лечении заболеваний – это информационная система, обеспечивающая путем анализа данных с использованием технологий искусственного интеллекта информационное сопровождение врача при выставлении диагноза и генерации листа назначений и рекомендаций к лечению с целью снижения возможности возникновения врачебных ошибок и повышения качества оказываемой медицинской помощи.

Однако следует отметить, что в настоящий момент отсутствуют полноценные универсальные системы диагностирования заболеваний и генерации шаблонов листов назначений, которые, во-первых, не ограничивались бы применимостью моделей к относительно структурированной информации, а во-вторых, грамотно учитывали состояние здоровья пациента и позволяли бы производить генерацию индивидуальных шаблонов рекомендаций к лечению.

2. К настоящему моменту не алгоритмизированы этапы выделения наиболее информативных блоков документов МИС и формирования корпуса текстов, пригодных для построения моделей ИИ. Характерной проблемой данного этапа является наличие документов различной структуры, в то время как получение репрезентативного набора входных данных является ключевым условием эффективного решения задач машинного обучения.

3. Несмотря на высокую эффективность моделей машинного обучения, построенных на структурированных данных и используемых на практике, они имеют недостатки. С одной стороны, характеристики структурированных данных могут различаться у разных пациентов в зависимости от их заболеваний и сроков пребывания. Кроме того, структурированные медицинские данные требуют выбора признаков в качестве механизма регуляризации данных для стандартных моделей, что отсекает различные признаки и резко сокращает выборки данных.

Оптимальному решению задачи анализа слабоструктурированных данных способствует использование наиболее современных методов обработки естественного языка, которые являются одним из наиболее активно развивающихся направлений искусственного интеллекта.

4. Сложность задачи обработки слабоструктурированных данных МИС и построения ИСППР заключается в отсутствии готовых протоколов

подключения к МИС для сбора обезличенных данных, хранении данных протоколов оказания медицинских услуг в разношаблонных документах, которые могут содержать опечатки и неоднозначные диагнозы. Кроме того, существует многовариантность выхода моделей генерации текста, которая осложняется наличием узкоспециализированных терминов и необходимостью использования большого объема вычислительных ресурсов. Многообещающим подходом является использование современных языковых моделей для классификации и генерации текстов и ресурсов графических процессоров (GPU).

Глава 2. Концепция интеллектуального анализа клинических данных медицинских информационных систем для поддержки принятия решений

В главе представлено описание концептуальной модели анализа клинических данных и поддержки принятия решений для автоматизации процессов заполнения ЭМК. Описана иерархическая модель структурирования данных амбулаторных карт пациентов для сбора информации из разношаблонных XML-документов. Предложены алгоритмы автоматической выгрузки данных ЭМК и извлечения информации из разнородных XML-документов.

2.1. Концептуальная модель анализа клинических данных МИС и поддержки принятия решений

Активная цифровизация медицинских процессов и внедрение информационных технологий является популярной тенденцией на протяжении последних лет. Данный вектор развития позволяет повысить качество обслуживания пациентов и упростить предоставление услуг.

В связи с высокой скоростью роста объема данных процесс получения и анализа важной информации из хранилища данных медицинской информационной системы может быть проблемой. Поэтому для управления, анализа и обработки данными в медицинской информационной структуре требуются системы, работающие с большими данными. Необходимы эффективные технологии их обработки и хранения, а также способы организации взаимодействия подсистем МИС друг с другом при обмене данными.

Процесс оказания медицинской услуги проведения приема и лечения пациента зависит от доступной для лица принимающего решения (ЛПР) информации, в соответствии с которой он формирует свои рекомендации, поэтому при разработке СППР следует, прежде всего, рассмотреть данный процесс в подобных условиях. На системном уровне это лучше всего сделать, используя методологию IDEF0 [52].

Контекстная диаграмма IDEF0 процесса оказания медицинской услуги проведения приема и лечения пациента показана на рисунке 2.1. Основной функцией выбран процесс «Формирования индивидуальных рекомендаций при управлении лечебным процессом».

Входными величинами определены классические для медицинских услуг объективные данные обследования лаборатории и измерения жизненных показателей, а также информация о жалобах пациента на приеме, которые являются отражением быстрого изменения его субъективного восприятия своего состояния.

В качестве *управляющих ограничений (условий)* выступают нормативные акты (федеральные, ведомственные документы и нормативные документы

учреждений), а также выработанные управляющие воздействия, характеризующие опыт врача, и, соответственно, методы обработки естественных языков, машинного обучения, нейросетевые методы и методы оптимизации – предложенные в рамках данного исследования.

В качестве *выходной информации* служит поставленный диагноз (укрупненная группа заболевания по МКБ-10) и сформированный лист назначений и рекомендаций к лечению.

Механизмами реализации выступают: врач и дополнительный инструментарий (средства диагностики и лаборатории).

В рамках разработанной концептуальной модели анализа клинических данных и поддержки принятия решений для автоматизации процессов заполнения ЭМК с учетом индивидуализации этапов оказания медицинских услуг рассмотрена классическая схема взаимодействия врача, пациента и лабораторий с дополнительными обследованиями. По результатам взаимодействия в МИС вносится информация, которая в последствии хранится в распределённой базе данных (БД).

Рисунок 2.1 – Контекстная диаграмма IDEF0 процесса оказания медицинской услуги проведения приема и лечения пациента

В настоящее время медицинская информационная система реализована по принципу монолита, а основное взаимодействие между врачом и системой реализовано в виде веб-сервиса, через которое специалист после авторизации имеет разграниченный доступ к информации о пациенте, его истории и т.д.

Чтобы обеспечить взаимодействие между подсистемами МИС и настроить интеграцию больших данных, они должны быть загружены из подсистемы источника данных, отформатированы в соответствии с требованиями

подсистемы хранилища и переданы в нужную подсистему для обработки или анализа. В соответствии с архитектурой интеграции данных необходимо выполнить три основных шага: согласование схемы, связывание записей и объединение данных.

Для реализации компетенций аналитической МИС необходимо решать следующие задачи:

- Сбор и хранение больших объемов данных: аналитическая система Big Data должна быть способна собирать и хранить большие объемы данных, включая данные о пациентах, медицинские записи, результаты тестов, изображения и т.д.
- Анализ данных: система должна иметь возможность анализировать данные, чтобы выявлять тенденции, паттерны и связи между различными факторами, которые могут помочь в улучшении качества здравоохранения.
- Визуализация данных: система должна иметь возможность представлять данные в удобном для восприятия формате, таком как графики, диаграммы и таблицы.
- Интеграция данных: система должна быть способна интегрировать данные из различных источников, включая медицинские информационные системы, лаборатории и другие медицинские учреждения.
- Прогнозирование: система должна иметь возможность прогнозировать будущие тенденции и результаты на основе анализа данных.
- Безопасность данных: система должна обеспечивать безопасность и конфиденциальность данных пациентов в соответствии с законодательством о защите персональных данных.
- Машинное обучение: система должна иметь возможность использовать машинное обучение для улучшения анализа данных и принятия более точных прогнозов.

Для реализации и решения поставленных задач необходимо модифицировать архитектуру МИС интегрировав в нее подсистемы извлечения, обработки и анализа данных.

Рассмотрим основные структурные элементы системы аналитической обработки больших массивов данных МИС:

1) База данных, где хранится информация истории развития заболеваний пациента, его посещений, оказанных ему услуг, а также различная справочная информация.

2) Приложение, с помощью которого врач-специалист взаимодействует с базой данных для заполнения, получения или обновления информации. В общем случае это может быть веб-сервис. Тогда процесс взаимодействия специалиста с базой данных можно представить схемой на рисунке 2.2.

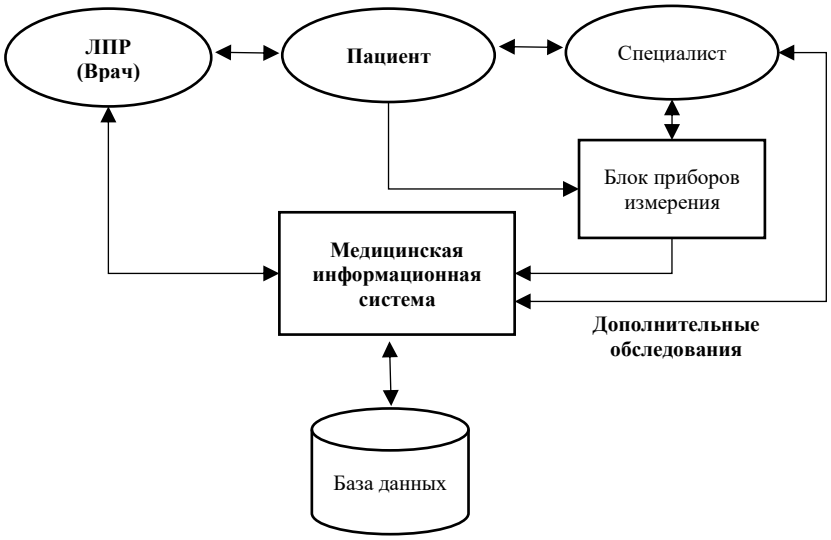

Рисунок 2.2 – Схема взаимодействия с МИС при оказании медицинских услуг

3) Одной из основных задач МИС является задача обработки данных для формирования слабоструктурированных данных таких, как XML-шаблоны выписки, листов назначений и т.д. Данная подсистема нужна для формирования наборов данных и служит поставщиком данных. Условно работу компоненты можно разделить на следующие части: модуль «XML Parser» (реализуется на стороне МИС) выгрузки данных (обезличивание, преобразование больших объемов данных, включая данные о пациентах, медицинские записи, результаты тестов, изображения дополнительных лабораторных исследований; модуль «DictParse» (реализуется на стороне внешнего сервера) преобразования неструктурированных данных в необходимый формат для последующей обработки.

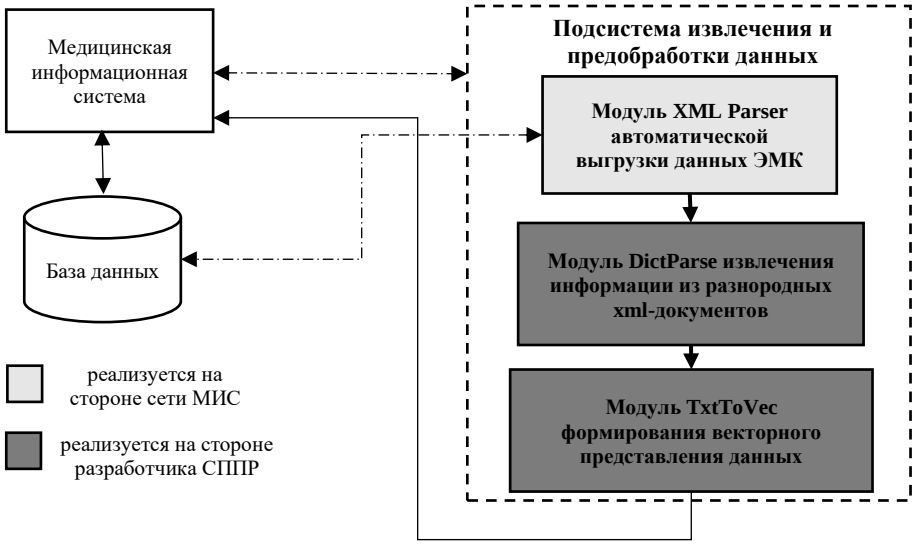

Рисунок 2.3 – Структура подсистемы извлечения и предобработки данных

Данная подсистема взаимодействует с приложением МИС и может получать пакетные данные с помощью авторизации и токенов-сессий. Для

анализа текстовой информации, полученной из МИС, необходимо ее преобразовать к векторному формату, необходимому для реализации моделей искусственного интеллекта. Поэтому встроим модуль «TxtToVec», который обучен на медицинском корпусе слов для последующей векторизации текстов. Схема подсистемы представлена на рисунке 2.3.

4) Для решения задач прогнозирования укрупненной группы заболевания по МКБ (модуль «PredictMKB») и автоматической генерации шаблонов листа назначений и рекомендаций (модуль «GenNLP») необходимо создать модели, обучить, настроить интеграцию с другими подсистемами МИС, наладить процесс обновления моделей. Данная подсистема является хранилищем выбранных алгоритмов и непосредственно связана с поставщиками данных в нее. Также необходимо автоматическое тестирование и оценка эффективности алгоритмов машинного обучения, что может включать в себя использование тестовых данных для оценки точности и полноты модели, а также проведение анализа ошибок, чтобы определить, какие параметры необходимо улучшить. Данный узел имеет интеграцию с подсистемой МИС для предоставления результатов врачу-специалисту по его запросу. Данная подсистема будет разворачиваться на внешнем сервере. Схематично подсистему обучения моделей ИИ и ее связи можно представить в виде рисунка 2.4

Рисунок 2.4 – Структура подсистемы моделей машинного обучения и ее связи с другими подсистемами.

Таким образом, концептуальную модель анализа клинических данных и интеллектуальной поддержки принятия решений, разработанную в соответствии с описанным выше подходом, можно представить в виде рисунка 2.5.

Анализируя построенную систему, можно сделать следующий вывод, что построенная система требует выполнения двух важных аспектов со стороны реализации МИС:

- Реализация интерфейса поставщика данных, необходимых для анализа и решения задач;
- Внесение изменений в интерфейс приложения для получения результатов моделей и их внедрения в систему поддержки МИС.

Построенная архитектура имеет небольшую стоимость интеграции и может быть внедрена в различные МИС при реализации этих аспектов.

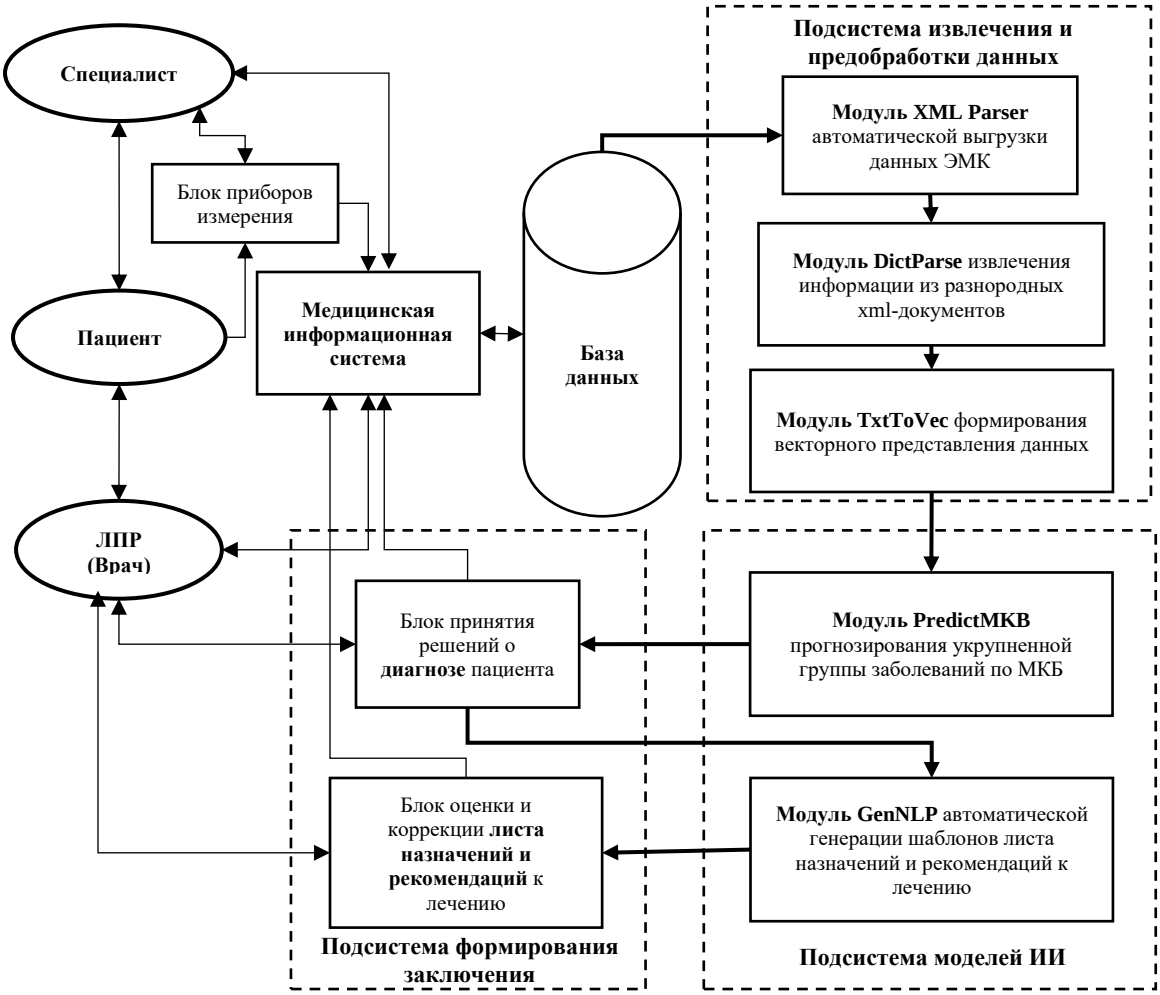

Рисунок 2.5 – Концептуальная модель анализа клинических данных и интеллектуальной поддержки принятия решений

Для определения функций процесса оказания медицинской услуги при проведении приёма и лечения пациента осуществлена декомпозиция первого уровня, представленная на рисунке 2.6. Согласно предварительному распределению функций данный процесс проходит четыре стадии: осмотр пациента и сбор информации; анализ истории болезней; выставление диагноза; назначение рекомендаций к лечению.

В рамках разработанной концептуальной модели предполагается разбиение процесса оказания медицинской услуги на 6 блоков: сбор информации и осмотр пациента; обработка информации в МИС; анализ информации и прогноз группы заболевания; принятие решений об окончательном диагнозе; формирование листа назначений и рекомендаций; коррекция результатов в соответствии с мнением ЛПР и управление лечебным процессом.

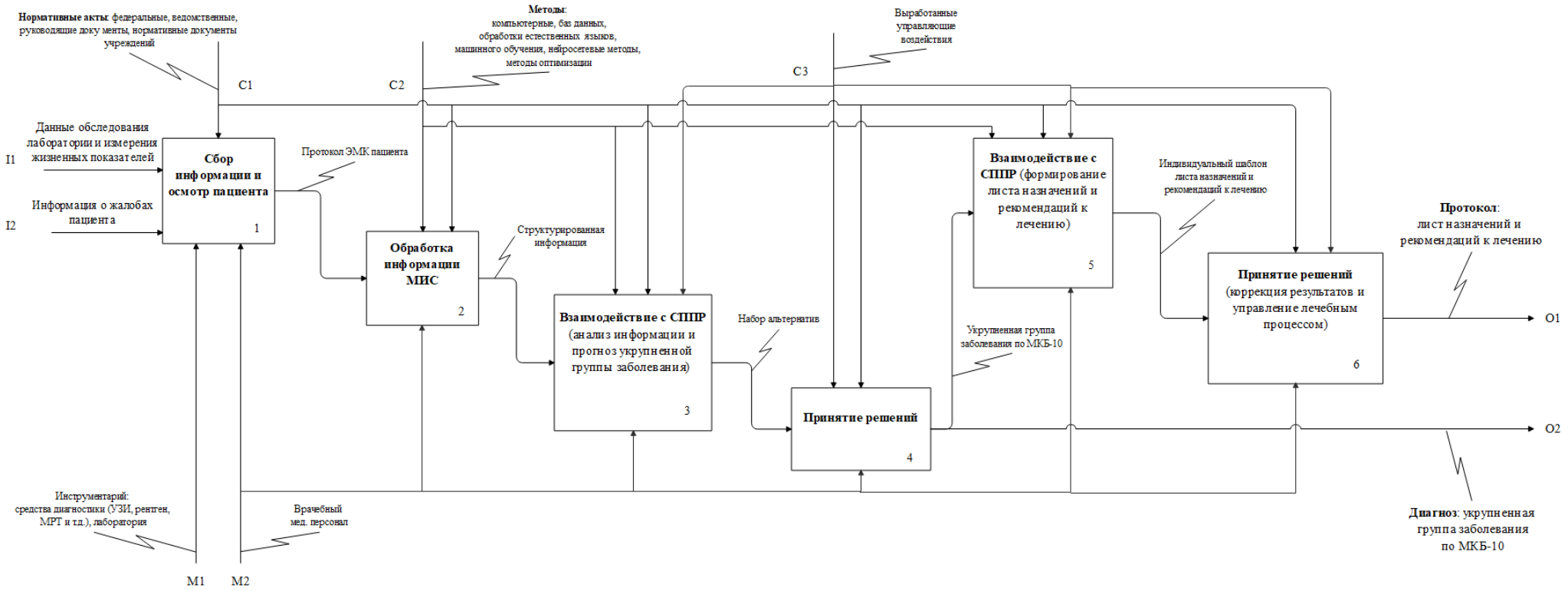

Рисунок 2.6 – Функциональная диаграмма интеллектуальной СППР в медицинской практике

Таким образом, концептуальная модель анализа клинических данных и поддержки принятия решений представляет собой систему, которая использует методы искусственного интеллекта для структурирования и анализа текстовых данных, содержащихся в медицинских записях пациентов. Данная модель включает основные этапы анализа текстовых данных и позволяет визуализировать процесс разработки и применения интеллектуальных моделей для решения диагностических и прогностических задач медицины. Построенная модель анализа клинических данных может быть интегрирована в другие системы здравоохранения и автоматизировать процессы заполнения ЭМК при диагностике заболеваний и формировании рекомендаций к лечению.

2.2. Иерархическая модель данных амбулаторных карт пациентов для обработки разношаблонных документов МИС

Применение интеллектуальных инструментов обработки данных в анализе медицинской информации позволяет получить более полное и точное представление о состоянии здоровья пациентов, улучшить качество обслуживания и позволит принимать более обоснованные решения в области здравоохранения. При этом, важно отметить, что данные медицинских информационных систем относятся к категории больших данных по ряду причин:

— *Объем данных:* Медицинские данные в МИС могут включать информацию о миллионах пациентов, результаты обследований, истории болезней, лекарствах и т.д.

— *Разнообразие данных:* Медицинская информация включает в себя различные типы данных, такие как текстовые документы, изображения (рентгенограммы, МРТ), результаты анализов, данные о приемах и т.д.

— *Сложность данных:* Медицинские данные могут быть сложными и неструктурированными, содержащими большое количество деталей и взаимосвязей между различными параметрами.

— *Скорость накопления данных:* В медицинской практике данные поступают постоянно, в реальном времени, например, результаты анализов, данные о приемах пациентов и т.д. Это приводит к быстрому накоплению большого объема информации.

Выделенные факторы определяют накопленную информацию в медицинской информационной системе большими данными и требуют специальных подходов к их обработке, анализу и защите.

Для структурирования данных амбулаторных карт пациентов в медицинской информационной системе в рамках иссследования разработана иерархическая модель данных, которая позволяет описать структурные отношения между различными сущностями в документах, такими как диагнозы, лекарства, результаты анализов и т.д.

Отметим, что между представителями различных сущностей возможна прямая связь: например, выставленный диагноз для конкретного пациента; медицинская организация, оказывающая услугу пациенту; соответствующие назначенные лекарства для данного диагноза и т.д. Разработанная иерархическая модель данных позволит использовать связи между различными сущностями для выявления паттернов и трендов.

Иерархическая модель данных значительно упрощает структурирование и анализ данных в медицинской информационной системе, что может быть полезно для улучшения качества управления больницей и оптимизации лечения пациентов.

Для построения иерархической структуры необходимо определить все сущности, участвующие в процессе оказания услуг в медицинских учреждениях. В общем случае можно выделить такие сущности, такие как «Пациент», «Медицинское учреждение», «Случай» и «Протокол».

Сущность «Пациент» описывает посетителя, который обращается за услугой в «Медицинское учреждение» (поликлинику). При обращении в поликлинику и оказании услуги формируется «Случай» (для каждого случая лечения может в истории болезни хранится несколько протоколов посещения врача, получения анализов и т.п). В зависимости от типа «Случая» внутри него может содержаться такая информация как:

- жалобы «Пациента»;
- рекомендации врача;
- результаты анализов;
- результаты осмотра врача;
- описание оказания медицинских услуг;
- первичный диагноз по МКБ;
- расширенный диагноз по МКБ;
- дата начала и окончания оказания услуги.

Каждый «Пациент» может обратиться за длительный промежуток времени в несколько медицинских учреждений, где внутри каждой поликлиники имеет свою историю посещений из «Случаев». Каждый «Случай» закреплен за «Пациентом» и «Медицинским учреждением» и связывает данные сущности между собой.

В общем случае электронная медицинская карта, которая закрепляет пациента за медицинским учреждением, является хранилищем «Случаев» и может храниться на серверах МИС.

Каждый «Случай» медицинской информационной системы г. Оренбург представляет из себя древовидную структуру в виде файла с расширением XML и может иметь различный формат в зависимости от типа «Случай» (прием-осмотр, результаты анализов и т.д.). «Пациент» может прийти для получения услуги в «Медицинское учреждение» «А», а затем его могут перевести в «Медицинское учреждение» «В» в рамках одного «Случая».

Таким образом, иерархическая модель данных МИС может быть представлена следующим образом:

$$M =< P, M, C, D > \quad (2.1)$$

где $P = \{P_1, \dots, P_n\}$ – множество пациентов, $M = \{M_1, \dots, M_k\}$ – множество медицинских организаций, зарегистрированных в МИС, $\mathrm{C} = \{\mathrm{C}_{ij} | i \in \overline{1,n}, j \in \overline{1,v_i}\}$ – множество случаев лечения (v_i – количество случаев заболеваний i-го пациента), $D = \{D_{ij}^t | i \in \overline{1,n}, j \in \overline{1,v_i}, t \in \overline{1,w_i}\}\}$ – множество протоколов посещений (w_i – количество посещений i-го пациента), причем

$$D =< D_{obj}, D_{comp}, D_{mkb}, D_{recom} >, \quad (2.2)$$

где D_{obj} – множество записей объективного осмотра пациента (данные измерения жизненных показателей и дополнительных обследований), D_{comp} – множество описаний жалоб пациента, D_{mkb} – множество групп заболеваний по МКБ-10 (диагнозы), D_{recom} – множество листов назначений и рекомендаций к лечению.

Иерархическая модель данных отношений между сущностями различных слабоструктурированных документов информационных систем, необходимых для интеллектуальной обработки больших данных, на примере электронных медицинских карт пациентов, можно представить в виде схемы, изображенной на рисунке 2.7.

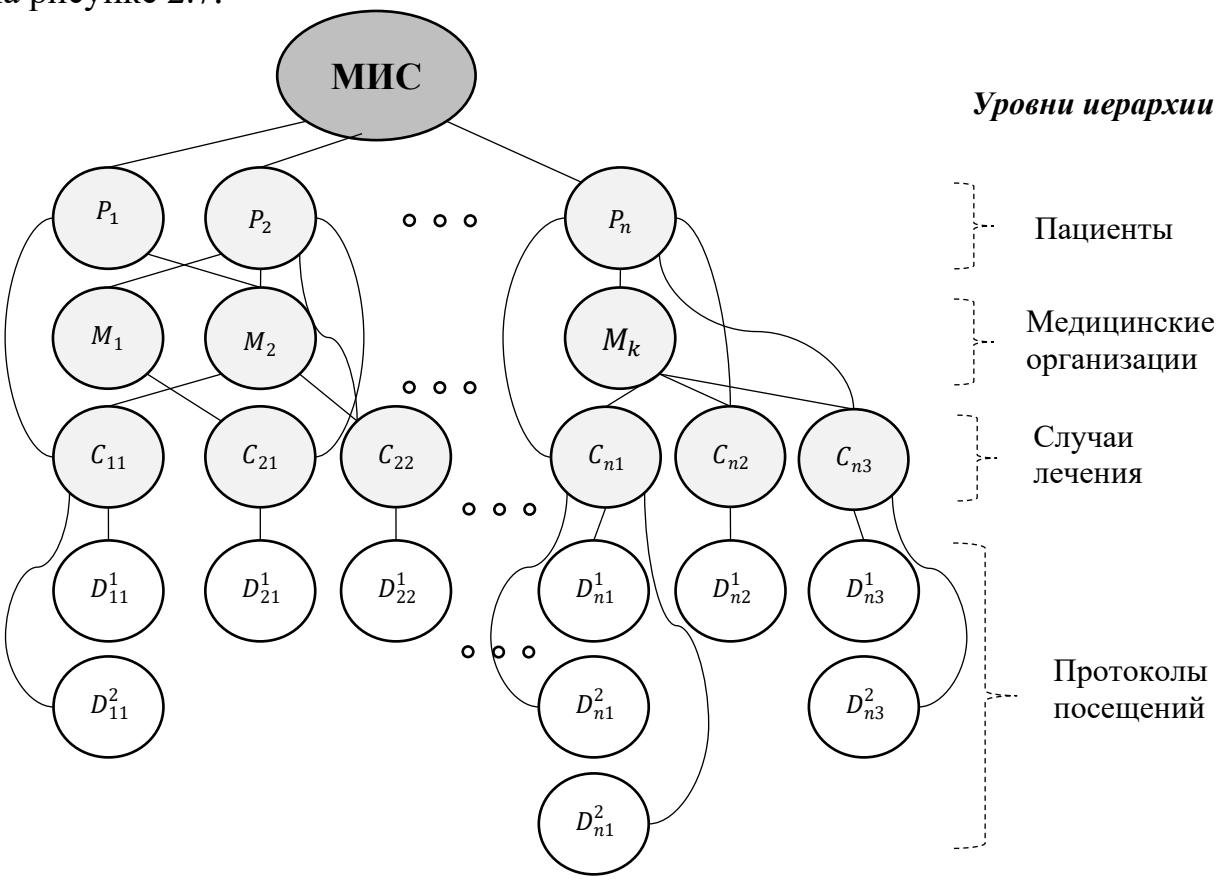

Рисунок 2.7 – Иерархическая модель структурирования данных амбулаторных карт пациентов

Рассмотренная структура МИС по описанию посещения «Пациентами» поликлиник отражает реальный процесс оказания медицинских услуг. Связь между сущностями «Случай» и «Пациент» может отражать весь процесс протекания начала заболевания, приема врача. лечения и рекомендаций врача, сдача анализов, выздоравливания и выписки. Так как «Пациент» может получать медицинскую помощью в различных медицинских учреждениях, например, стационарах, важно рассматривать историю пациента и протекания болезни независимо от конкретного «Медицинского учреждения».

В связи с тем, что документы протоколов посещений не обладают семантической интероперабельностью, множество D не может быть выгружено в заданном виде без дополнительной обработки. Для реализации функциональной совместимости в диссертационном исследовании предложен алгоритм автоматической выгрузки данных ЭМК.

2.3. Алгоритм автоматической выгрузки данных ЭМК

Для подсистемы извлечения слабоструктурированной информации из системы МИС необходим модуль, который бы позволял выгружать обезличенные протоколы электронных медицинских карт пациентов и обладал свойством интероперабельности. Для этого можно использовать специальное программное обеспечение, реализованное на стороне МИС и имеющее следующие основные характеристики:

1) Обезличивание исходных данных. Так как реализация модуля планируется в контуре медицинского информационного аналитического центра, данный модуль может обрабатывать персональную информацию.

2) Устойчивость выгрузки. Обработка и выгрузка одного протокола может занимать несколько секунд в зависимости от его размера. Поэтому возникает проблема в случае разрыва соединения или превышения времени обработки протокола, в последствии которых может произойти сбой выгрузки.

3) Масштабирование. Для повышения скорости выгрузки протоколов из МИС необходимо реализовать распараллеливание процесса.

4) Хранение результатов выгрузки. После выгрузки и обезличивания данных необходимо хранилище для файлов протоколов с информацией об обработанных данных.

Для описания модуля автоматической выгрузки «XML Parser» рассмотрим реализацию каждого пункта.

Обезличивание файлов в МИС можно осуществить путем удаления или замены конфиденциальных данных пациента, таких как ФИО, дата рождения, адрес, номер полиса ОМС и т.д. на псевдонимы или коды. Для этого используют специальные алгоритмы обезличивания, которые сохраняют целостность и структуру данных, но исключают возможность идентификации пациента. Также необходимо убедиться, что все копии файлов с обезличенными данными удалены или зашифрованы, чтобы предотвратить несанкционированный доступ

к конфиденциальной информации. Важно отметить, что обезличивание данных не должно противоречить законодательству о защите персональных данных и требованиям к хранению медицинской документации.

Конкретный список персональных данных, которые могут храниться в системе МИС, может зависеть от рассматриваемой реализации системы. Однако, как правило, в системе МИС могут храниться следующие персональные данные:

- ФИО пациента
- Дата рождения
- Данные паспорта
- Адрес проживания
- Номер телефона
- Полис медицинского страхования
- Семейное положение
- Персональные данные медицинского персонала, включая ФИО, должности, специальности и квалификации.

Для реализации функции обезличивания использованы следующие подходы:

- Замена идентифицирующих данных таких как ФИО, на уникальный идентификатор из базы данных по его полису МИС, что исключает присвоение одинакового значения двум разным пациентам с полностью одинаковыми именами. Такой метод заменяет конфиденциальные данные пациента на уникальный идентификатор, который не может быть связан с реальным человеком.

2. Хэширование данных. Данный метод преобразует конфиденциальные данные пациента в уникальный хэш-код, который не может быть обратно преобразован в исходные данные.

3. Удаление данных. Удаление конфиденциальных данных пациента из медицинских записей полностью. К таким данных можно отнести адрес проживания, номер телефона, полис медицинского страхования, семейное положение.

4. Анонимизация данных. Метод заменяет конфиденциальные данные пациента на общие характеристики, которые могут быть использованы для статистического анализа, но не могут быть связаны с конкретным человеком. Например, анонимизацию можно провести для информации о дне рождения. Данное поле можно заменить, преобразовав возраст в количество лет, а для детей меньше 3 лет – в количество месяцев со дня рождения, соответственно, добавив тип хранения возраста месяц или год.

5. Маскирование данных. Метод заменяет часть конфиденциальных данных пациента на символы или звездочки, чтобы скрыть часть информации.

После обезличивания данных их можно использовать уже на стороне внешнего сервера для построения моделей искусственного интеллекта.

Для обеспечения надежности выгрузки и повышения устойчивости модуля, необходимо реализовать систему учета выполнения выгрузки. Для этого использована база данных, в которой хранится информация о статусе задачи

выгрузки отдельного протокола. В таблице 2.1 представлена таблица с основными полями сущности «TaskCaseUpload».

Таблица 2.1 – Структура таблицы задачи выгрузки случая из подсистемы МИС «TaskCaseUpload»

Название	Тип	Описание
id	integer(10)	уникальный идентификатор задачи выгрузки протокола случая
id_case	integer(10)	уникальный идентификатор протокола случай в базе данных МИС
file_path	varchar(512)	Путь к сохранению выгруженного файла протокола в файловой системе МИС, имя которого является уникальным для каждого отдельного случая
file_size	integer	Размер выгруженного файла протокола в байтах

У каждой задачи выгрузки протокола «TaskCaseUpload» необходим текущий статус. Введем справочник возможных типов статусов задач и таблицу с допустимыми переходами задачи, чтобы исключить пропуск этапов в случае сбоя подмодуля из последовательности алгоритма.

Введем следующий справочник статусов задачи выгрузки протокола из медицинского информационного аналитического центра:

- «Задача создана». Самый первый этап выгрузки протокола. На данном этапе в базе данных создается запись, что инициирована выгрузка протокола.
- «Отправлен запрос на сервер». Второй этап выгрузки протокола, после создания записи в БД, скрипт отправляет запрос на выгрузку протокола из подсистемы МИС.
- «Ошибка отправки запроса на сервер». Данный статус относится к категории ошибок и может возникнуть в случае ошибки на этапе запроса получения данных на сервер, возможной причиной может служить разрыв соединения с сервером.
- «Успешное получение данных от сервера». Статус появляется если данные протокола успешно получены от сервера.
- «Ошибка получения данных от сервера». Статус относится к категории ошибок и может возникнуть в случае некорректного запроса или ошибки авторизации.
- «Сохранение файла протокола на диск». Статус сообщает об инициации сохранения файла протокола на диск после успешного его получения из МИС.
- «Ошибка сохранения файла на диск». Статус относится к категории ошибок и может возникнуть в таких случаях как: нехватка памяти на выделенном жестком диске, файл с таким именем протокола уже существует, отсутствует

директория для сохранения файла протокола или некорректный путь сохранения файла.

- «Успешное сохранение файла на диск». Статус задачи сообщает об окончании работы с протоколом случая и является последним индикатором, после которого можно переходить к выгрузке следующего протокола.

Структура сущности типа статуса задачи выгрузки протокола «TypeStatusTaskCaseUpload» представлена в таблице 2.2.

Таблица 2.2 – Структура таблицы типа статуса задачи выгрузки протокола «TypeStatusTaskCaseUpload»

Название	Тип	Описание
id	integer(10)	Уникальный идентификатор типа статуса задачи выгрузки протокола
name	varchar(512)	Текстовое описание статуса задачи
is_success	boolean	Флаг, является ли данный статус успешной операцией или вызвана ошибка

Для отслеживания правильной смены статуса задач необходимо определить, какой статус задач может быть после предыдущей, чтобы в случае возникновения ошибок не пропустить некоторый этап и не пропустить выгрузку протокола.

Для этого определим корректные последовательности смен статусов в виде таблицы в базе данных «TaskStatusValidChange», представленной в таблице 2.3. Для отслеживания истории статуса задачи выгрузки протокола от создания записи в БД до получения и сохранения файла определим сущность «TaskCaseStatus». Данная сущность будет хранить изменения статуса во времени. Структура сущности «TaskCaseStatus» реализованная в базе данных представлена в таблице 2.4.

Таблица 2.3 – Структура таблицы с корректными последовательностями изменения статуса задачи «TaskStatusValidChange»

Название	Тип	Описание
id	integer(10)	Уникальный идентификатор пары текущий статус, новый статус
id_current	integer(10)	Идентификатор на текущий статус задачи «TypeStatusTaskCaseUpload», поле является внешним ключом
id_next	integer(10)	Идентификатор на новый статус задачи «TypeStatusTaskCaseUpload», поле является внешним ключом

Таблица 2.4 – Структура таблицы с историей изменения статуса задачи «TaskCaseStatus»

Название	Тип	Описание
id	integer(10)	Уникальный идентификатор истории задачи
id_task	integer(10)	Идентификатор на таблицу задачи выгрузки «TaskCaseUpload», поле является внешним ключом
id_status	integer(10)	Идентификатор на статус задачи выгрузки протокола «TypeStatusTaskCaseUpload», поле является внешним ключом
id_prev	integer(10)	Идентификатор на статус предыдущую историю задачи выгрузки протокола «TaskCaseStatus», поле является внешним ключом
timestamp	datetime	Время изменения статуса задачи

Перед тем как обновить статус задачи на новый, процедура изменения проверяет, является ли данный переход задачи корректным, в соответствии с таблицей 2.5.

Таблица 2.5 – Допустимые варианты смены статуса задачи при выгрузке протоколов

Текущий статус задачи выгрузки протокола	Следующий допустимый статус задачи выгрузки протокола	Является ли смена статусов успешной операцией
Задача создана	Отправлен запрос на сервер	Да
Задача создана	Ошибка отправки запроса на сервер	Нет
Отправлен запрос на сервер	Успешное получение данных от сервера	Да
Отправлен запрос на сервер	Ошибка получения данных от сервера	Нет
Успешное получение данных от сервера	Сохранение файла протокола на диск	Да
Успешное получение данных от сервера	Ошибка сохранения файла на диск	Нет
Сохранение файла протокола на диск	Успешное сохранение файла на диск	Да

Построенная реляционная база данных позволяет отслеживать выгрузку протоколов и возникновение ошибок на различных стадиях, а также продолжать процесс с того момента, на котором произошел сбой. ER-диаграмма построенной базы данных представлена на рисунке 2.8.

Рассмотрим задачу хранения выгруженных данных. Хранение протоколов электронных медицинских карт (ЭМК) может быть организовано с помощью

специализированной системы управления данными. Для этого необходимо создать отдельный раздел в МИС, где будут храниться все выгруженные протоколы ЭМК. Каждый протокол должен быть связан с конкретным пациентом и содержать информацию и иметь запись в построенной базе данных.

Для обеспечения безопасности и конфиденциальности персональных данных пациентов, доступ к необезличенным протоколам должен быть ограничен только медицинским работникам, имеющим соответствующие права доступа.

Для обеспечения масштабирования и ускорения выгрузки файлов протоколов, необходимо иметь возможность распараллеливания работы скрипта. Существуют следующие способы распараллеливания:

- Разделение задачи на подзадачи: задача разбивается на несколько подзадач, которые могут быть выполнены параллельно.
- Разделение данных: данные разбиваются на несколько частей, которые могут быть обработаны параллельно.
- Разделение времени: задача выполняется в несколько этапов, каждый из которых может быть выполнен параллельно.
- Разделение ресурсов: задача выполняется на нескольких компьютерах или серверах, каждый из которых выполняет свою часть задачи.

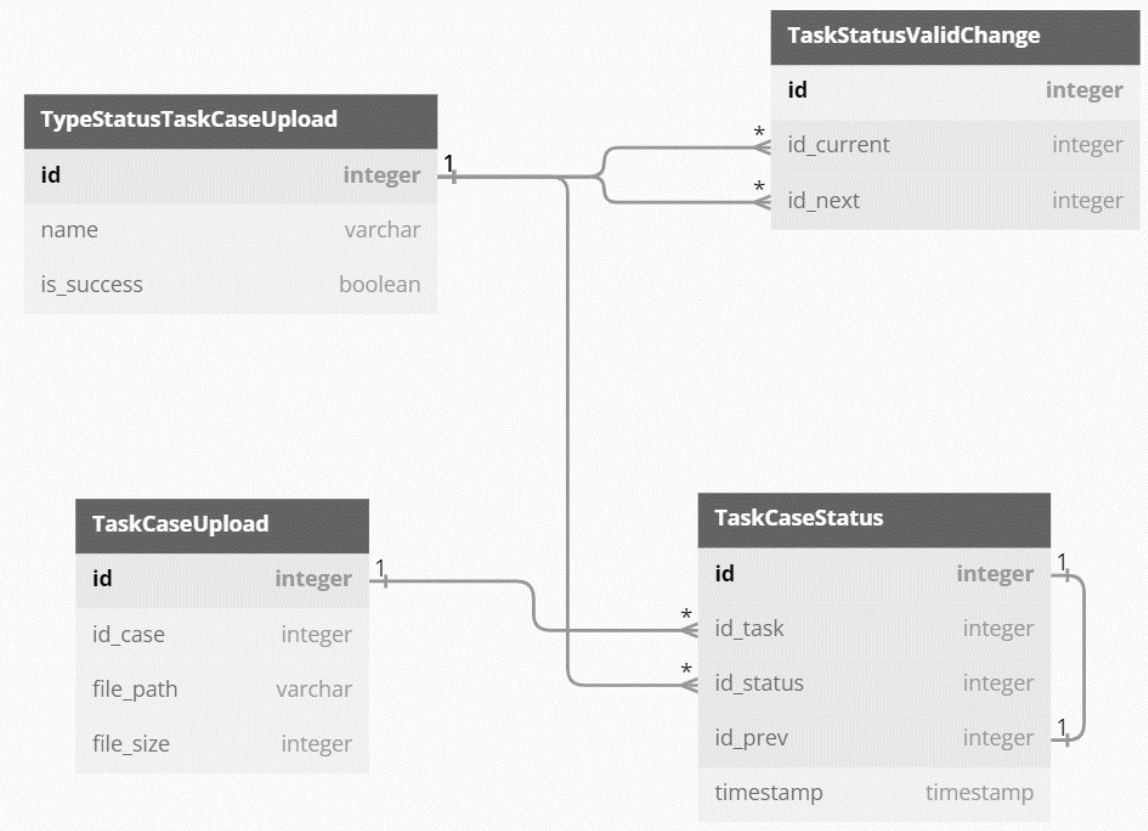

Рисунок 2.8 – ER-диаграмма для отслеживания выполнения выгрузки протоколов из МИС

- Модель акторов: задача представляется в виде набора акторов, которые могут обмениваться сообщениями и выполнять свои действия параллельно.

Самым простым и быстро интегрируемым вариантом является подход с выделением нескольким скриптам определенного непересекающегося пула протоколов случаев. Данный подход относится к описанному выше способу разделения ресурсов, при этом вместо серверов и компьютеров используются отдельно запущенные процессы. Алгоритм выполнения скрипта выгрузки пула протоколов представлен в таблице 2.6.

Таблица 2.6 – Алгоритм выгрузки файлов протоколов из медицинской информационной системы

Номер этапа	Описание этапа
1	На первом этапе необходимо сформировать пул из уникальных идентификаторов для данного скрипта, например, это может быть текстовый файл или таблица в БД. Данный пул выгружается из МИС.
2	На втором этапе в скрипт начинает обрабатывать первую запись случая, для которой в таблице «TaskCaseUpload» нет записи со значением case_id из пула. Затем в таблицу «TaskCaseStatus» происходит запись для текущей задачи со статусом «Задача создана». После чего инициируется отправка POST запроса (функция download_case(case_id)) с авторизацией к хранилищу для получения данных, при этом добавляя соответствующую запись об отправке запроса в таблицу со сменой статуса в случае успеха. В случае возникновения ошибки, алгоритм заканчивает работу с текущем протоколом, создает запись в истории обработки задачи в базе данных со статусом «Ошибка отправки запроса на сервер» и переходит к следующему протоколу.
3	В случае успешной отправки запроса на выгрузку файла, алгоритм ожидает файл от сервера МИС с расширением XML. После успешного получения данных в БД в таблице с историей обработки задачи выгрузки протокола «TaskCaseStatus» создается запись со значением «Успешное получение данных от сервера». В случае возникновения ошибки создается соответствующая запись и алгоритм переходит к обработке следующего протокола.
4	После получения данных от сервера скрипт инициирует процесс сохранения файла (функция save_file_data(case_id)) создав запись для данной задачи в истории обработки со значением «Сохранение файла протокола на диск». В случае успешного выполнения сохранения, в таблицу «TaskCaseUpload» заносится информация о пути сохранения и размере файла протокола. Для каждого файла протокола формируется уникальное имя, основанное на поле «TaskCaseUpload.id» чтобы избежать конфликта имен при сохранении.
5	После успешного сохранения протокола скрипт переходит к следующему протоколу с этапа 1.

Отметим, что в таблице 2.6 представлен общий алгоритм работы скрипта по выгрузке файлов протоколов из МИС, а его распараллеливание происходит при получении следующего ID случая в базе данных: запись с ней блокируется, чтобы данный ID не обрабатывался другим скриптом в параллельном процессе,

таким образом для каждого скрипта генерируются уникальные ID задачи, которые еще не были обработаны в БД.

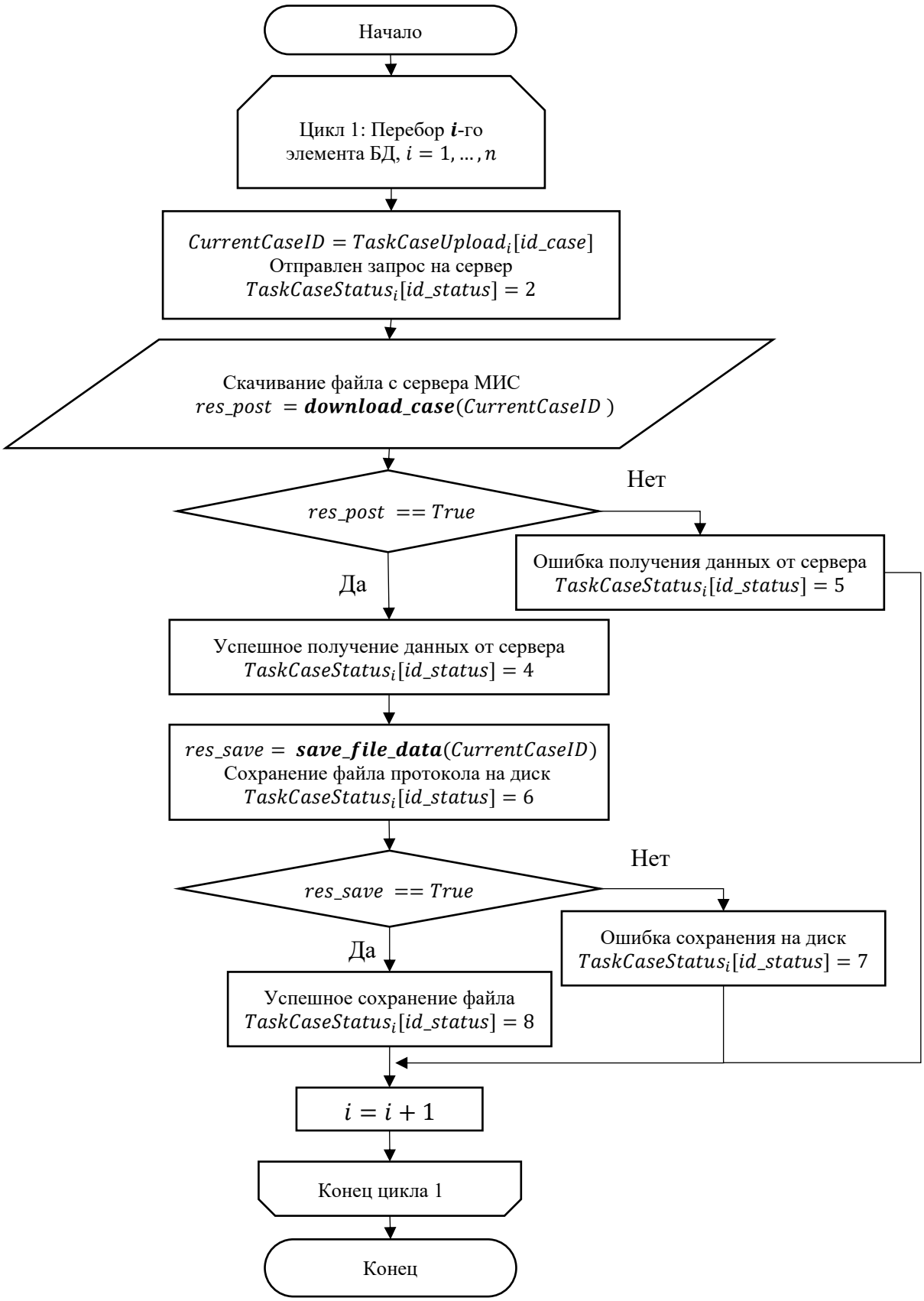

Рисунок 2.9 – Блок-схема алгоритма модуля «XML Parser»

Для реализации работы алгоритма разработан скрипт на языке программирования PHP для автоматической выгрузки. После чего данный модуль «XML Parser» запущен на уникальном пуле случаев протокола в отдельных процессах для распараллеливания.

В случае возникновения сбоев выгрузки протоколов, процесс выгрузки протоколов возобновлялся на данных протоколах заново после устранения неполадок.

Таким образом, разработанный модуль «XML Parser» автоматизированной выгрузки данных XML-протоколов, обладает функциями обезличивания, отслеживания статусов задач выгрузки и масштабированием. Данная подсистема основана на взаимодействии с веб-сервисом по API для сбора информации документов и позволяет автоматизировать процесс обработки данных МИС, распределить нагрузку между серверами и учитывает права доступа пользователя и возможные сбои при выполнении модуля.

2.4. Алгоритм извлечения информации из разнородных XML-документов

Медицинские протоколы случаев в формате XML имеют свои особенности, связанные с требованиями к защите конфиденциальности пациентов и обработке медицинских данных. Так, они включают в себя элементы для идентификации пациента и защиты его данных, а также для описания медицинских процедур и результатов исследований. Кроме того, медицинские протоколы в формате XML часто используются для обмена данными между различными системами здравоохранения, что требует соответствия стандартам и протоколам обмена данными в этой области.

Для обработки медицинского протокола случаев в формате XML можно использовать специальные программы и библиотеки, которые позволяют читать, записывать и обрабатывать данные в этом формате. Например, для работы с медицинскими протоколами в формате XML используются такие технологии, как XML Schema, XSLT, XPath и другие.

С помощью XML Schema можно определить структуру и типы данных, которые должны содержаться в медицинском протоколе. XSLT позволяет преобразовывать данные в XML-документе в другой формат, например, в HTML или PDF. XPath используется для выборки данных из XML-документа по заданным критериям.

Кроме того, для обработки медицинских протоколов в формате XML могут использоваться специализированные программные продукты, такие как системы электронной медицинской документации (ЭМД) и системы управления медицинскими данными (СУМД). Эти системы позволяют хранить, обрабатывать и обмениваться медицинскими данными в соответствии с требованиями к защите конфиденциальности информации о пациентах и стандартами обмена данными, определёнными в здравоохранении.

Структура медицинского протокола в формате XML может варьироваться в зависимости от конкретных требований и стандартов, но обычно она включает в себя следующие элементы:

- Информация о пациенте, включая ФИО, дату рождения, пол и другие персональные данные.
- Информация о проведенных медицинских процедурах, исследованиях и лечении, включая даты, названия процедур, результаты и другие детали.
- Информация о медицинских диагнозах и результатах обследований, включая данные о состоянии здоровья пациента и рекомендации по лечению.
- Информация о медицинском персонале, включая ФИО, должность и другие данные об участниках медицинского процесса.
- Другие дополнительные сведения, такие как информация о медикаментах, аллергиях, противопоказаниях и т.д.

Структура медицинского протокола определена с помощью XML Schema, что позволяет обеспечить ее стандартизацию и совместимость с другими системами и приложениями.

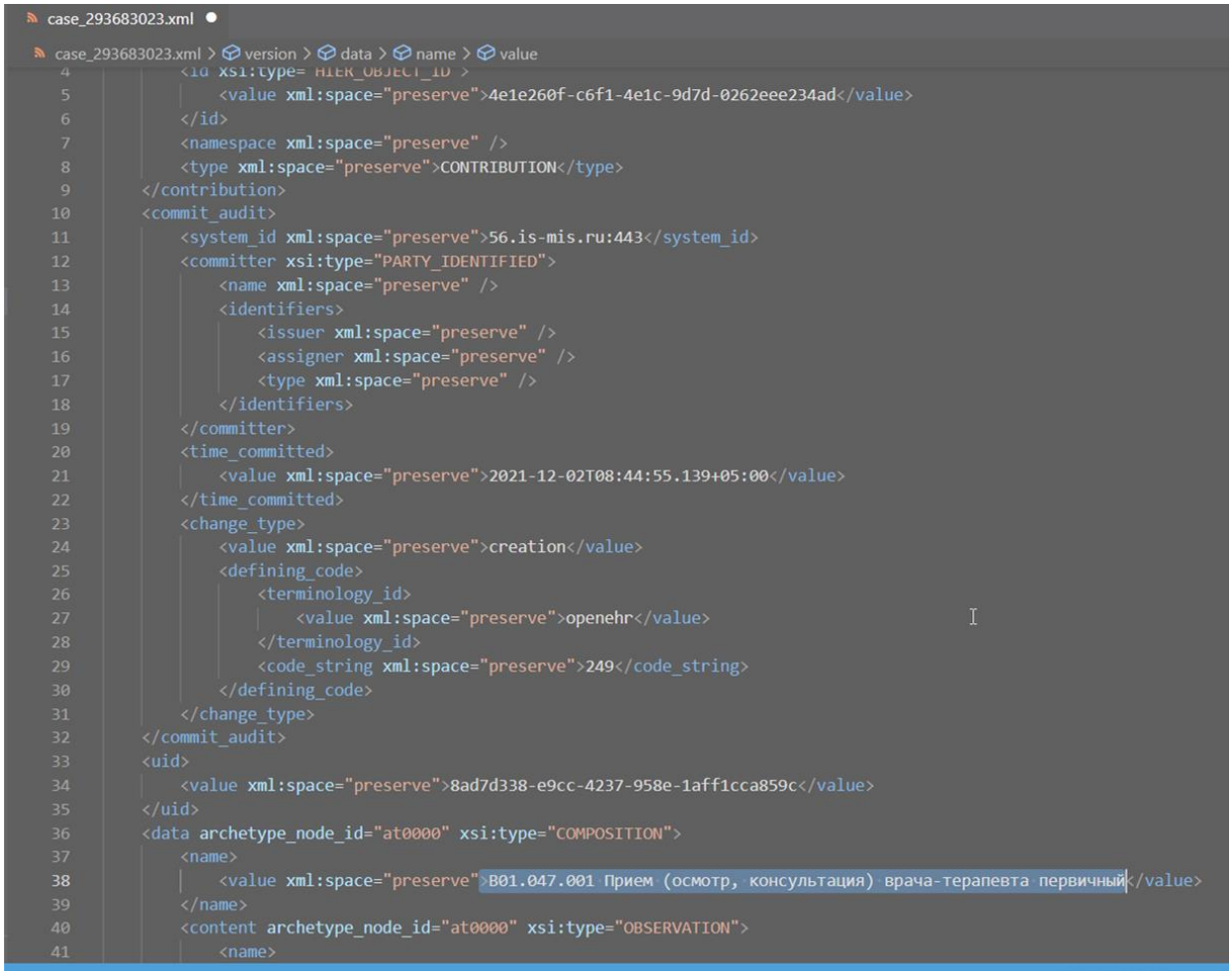

Рисунок 2.10 – Фрагмент примера XML протокола случая приема

Корневой узел содержит информацию о типе протокола и, помимо протоколов приема, в оказанных услугах также встречаются протоколы заключения ЭКГ, выписные эпикризы, анализы крови и многое другое.

Шаблоны представленных документов настолько разные, что при обработке и выделении заданных полей по определенным ключевым названиям может быть утеряна часть информации. На рисунке 2.10 представлен пример XML протокола случая приема. Выделяя основные узлы, XML-протокол может иметь следующую структуру, представленную на рисунке 2.11.

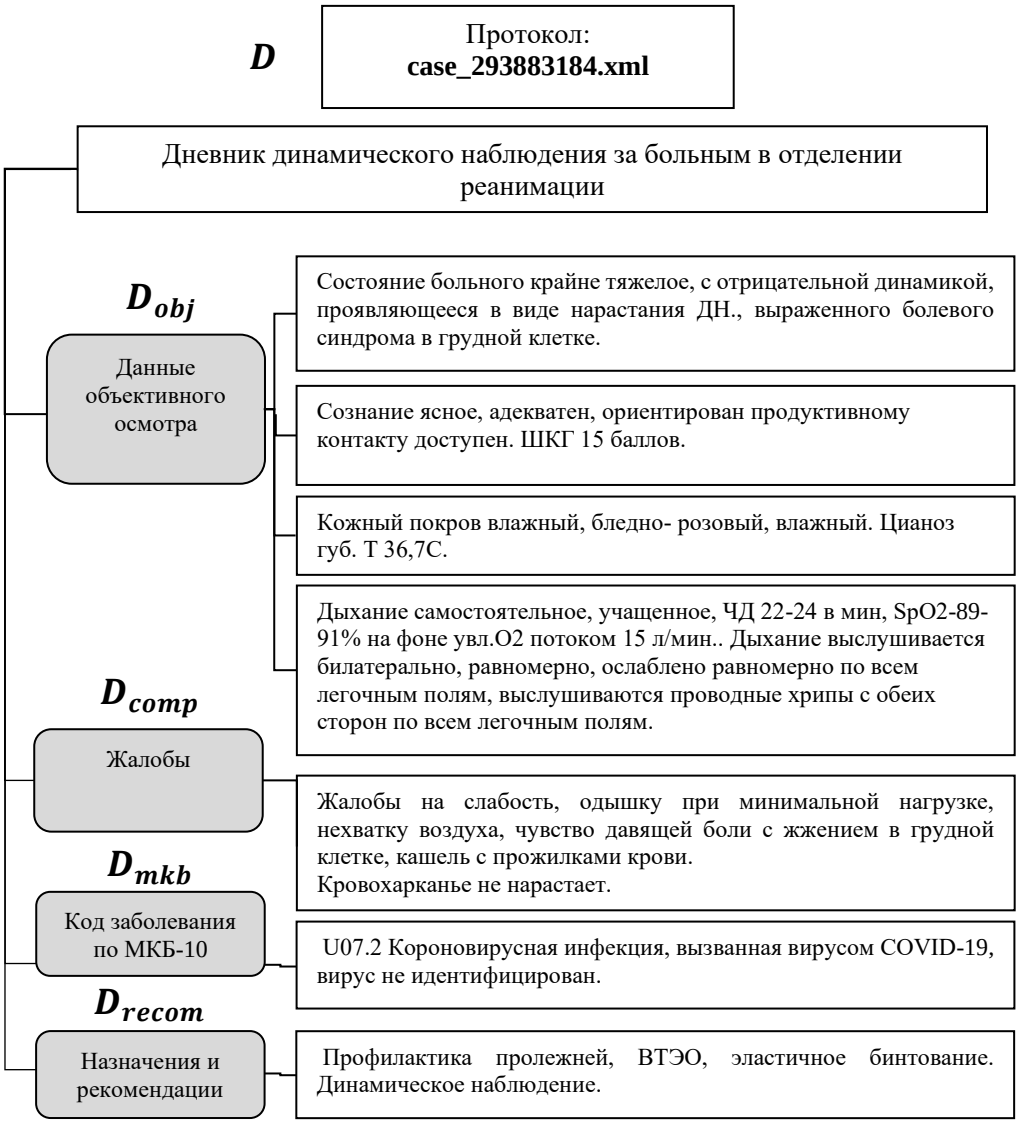

Рисунок 2.11 – Пример основных блоков XML-протокола

В связи с этим появляется необходимость разработки нетривиального модуля автоматического преобразования XML-документа и выделения основной информации посредством рекурсивного обхода и анализа древовидной структуры.

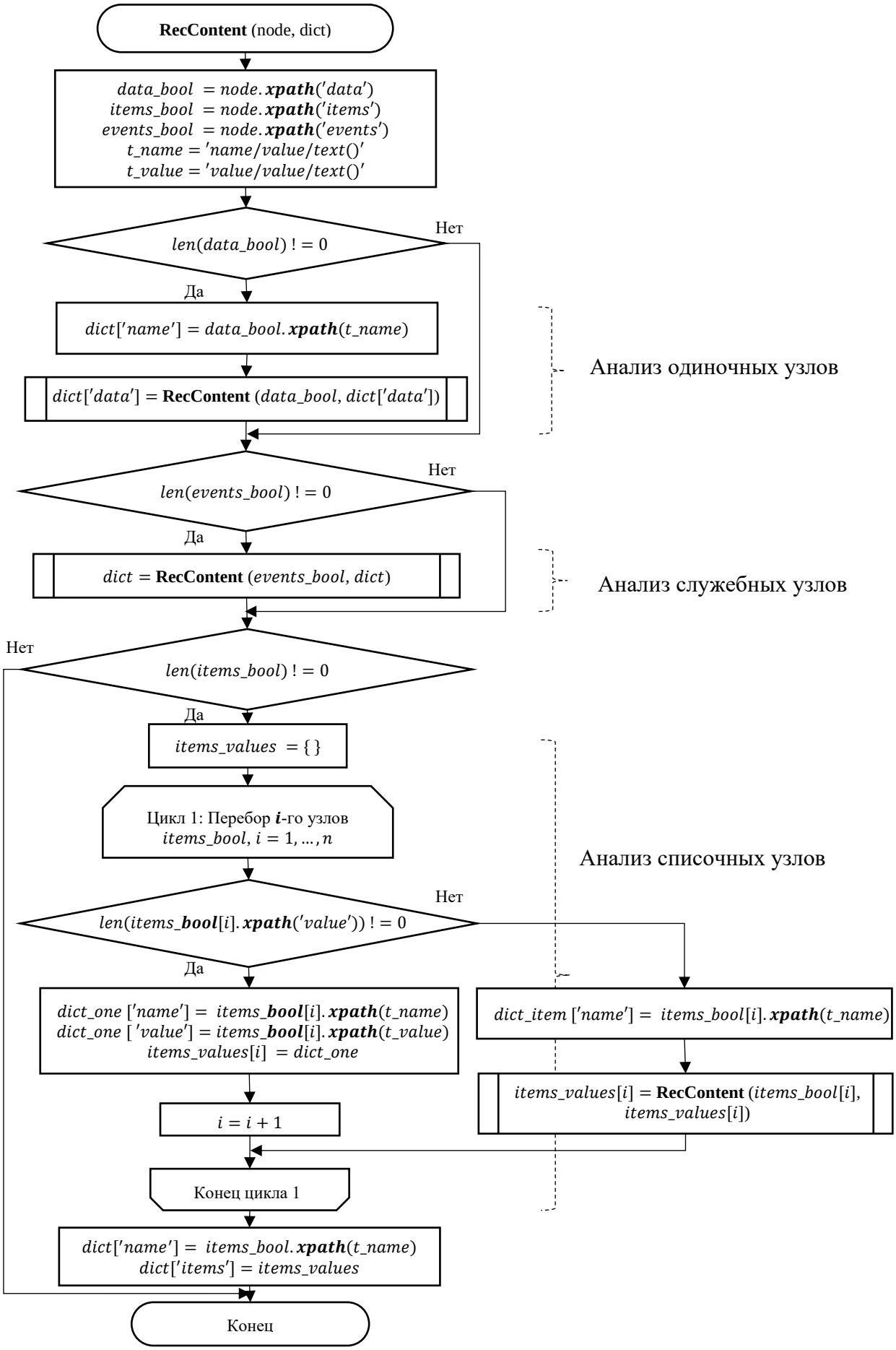

Рисунок 2.12 – Блок-схема алгоритма модуля «DictParse»

Для считывания информации из медицинского протокола в формате XML разработан рекурсивный алгоритм, который представлен на рисунке 2.12 и проходит по всем элементам документа для сохранения необходимой информации.

XML-документ протокола случая одного пациента имеет четыре основных типа узла, которые содержат информацию, необходимую для решения поставленных задач:

- ITEMS – узел, содержащий список узлов ITEM, где каждый узел может быть вложенным списком. Используется для описания жалоб, объединенных в группы.
- CONTENT – узел, хранящий в себе один элемент. Используется как корневой элемент описания типа протокола.
- DATA – узел, хранящий в себе информацию описания узла в блоке NAME и значение узла в блоке VALUE.
- EVENTS – узел, хранящий служебную информацию и может содержать дочерние элементы с узлами ITEMS и DATA.

Основной функцией в модуле «DictParse» является RecContent, которая обеспечивает рекурсивный проход по всем возможным веткам узлов ITEMS и CONTENT. Алгоритм выполняется до тех пор, пока не будут считаны все непустые поля xml-файла. При чтении и выделении данных из файла протокола важно запоминать к какому корневому узлу относится значение (запоминаются соответствующие name и value). Например, к блоку по сердечно-сосудистой системе будут относиться все измеренные параметры давления, пульса и т.д. В связи с этим реализована стратегия вложенных словарей для сохранения данных отношений признаков (в рекурсивную функцию передаются текущий узел для анализа и сохраненный к данному моменту вложенный словарь со считанными разделами и их характеристиками).

Для поиска информации в шаблоне при рекурсивном обходе документа используются следующие функции библиотек языка Python:

- len – функция определения длины массива (списка);
- xpath – функция оценивания XPath выражения с использованием некоторого экземпляра Element как контекстного узла, в случае положительного результата возвращает элемент XML-документа;

Результатом работы алгоритма после анализа документов XML являются иерархические словари – деревья записей. Деревья записей содержат следующую информацию:

- тип протокола;
- дата протокола;
- уникальный идентификатор пациента;
- список с жалобами по каждому корневому узлу;
- рекомендации врача;
- результаты анализов;
- диагноз по МКБ.

Таким образом, объединяя протоколы посещений одного пациента в единое дерево, разработанный модуль «DictParse» преобразования протоколов XML-документов позволяет построить дерево электронных медицинских карт пациентов и последовательно проанализировать содежимое историй заболеваний, учитывая взаимосвязь некоторых факторов внутри документа (жалобы пациента и анамнез болезни, результаты обследований и диагноз, назначения лечения). В отличии от извлечения исключительно числовых характеристик, данный подход позволяет комплексно оценить состояние здоровья пациента.

Выводы второй главы

1. Построена концептуальная модель анализа клинических данных и поддержки принятия решений для автоматизации процессов заполнения ЭМК, обладающая высокой степенью интегрируемости и позволяющая решать задачу внедрения моделей искусственного интеллекта в систему поддержки принятия врачебных решения. Данная модель проводит формализацию этапов структурирования текстовых данных и построения интеллектуальных моделей формирования рекомендаций к лечению диагностированных заболеваний.
2. Построена иерархическая модель данных амбулаторных карт пациентов, описывающая отношения между сущностями различных слабоструктурированных документов медицинских информационных систем, необходимых для интеллектуальной обработки больших данных и разработки систем искусственного интеллекта.
3. Разработан модуль «XML Parser» автоматизированной выгрузки данных XML-протоколов, обладающий функциями обезличивания, отслеживания статусов задач выгрузки и масштабированием. Данная подсистема основана на взаимодействии с веб-сервисом по API для сбора информации документов и позволяет автоматизировать процесс обработки данных МИС, распределить нагрузку между серверами и учитывает права доступа пользователя и возможные сбои при выполнении модуля.

Важно отметить, что разработанный модуль обеспечивает защиту персональных данных путем использования современных методов шифрования, управления доступом и мониторинга, что способствует сохранности и конфиденциальности информации.

4. Разработан модуль «DictParse» выделения информации из разношаблонных файлов протоколов случаев с расширением XML из медицинской информационной системы. В основе модуля «DictParse» лежит подход к рекурсивному перебору узлов XML, с последовательным анализом наличия содержимого и созданием дерева записи оказанной услуги в медицинской организации, позволяющей проанализировать взаимосвязь некоторых факторов внутри документа.

Глава 3. Разработка алгоритма прогнозирования укрупненных групп заболеваний на основе слабоструктурированных данных ЭМК

В третьей главе описаны подходы к прогнозированию укрупненных групп заболеваний на основе слабоструктурированных текстовых данных жалоб пациентов и данных объективного осмотра из ЭМК, приведены результаты реализации моделей прогнозирования на основе методов машинного обучения.

3.1. Формализация задачи прогнозирования укрупненных групп заболеваний на основе методов машинного обучения

Для разработки прогнозных моделей диагностирования заболеваний пациентов с ССЗ на основе алгоритмов, представленных в главе 2, реализованы модули взаимодействия с региональной МИС города Оренбурга для выгрузки данных протоколов посещений: модуль «XML Parser» загружает обезличенные протоколы через API Медицинского информационно-аналитического центра в формате XML для пациентов, имеющих диагноз ССЗ в истории болезней, а модуль «DictParse» выделяет информацию из разношаблонных XML-документов посредством рекурсивного обхода и анализа всех веток разметки. Важно отметить, что выборка пациентов за закрепленный для анализа период времени включает не только заболевания сердечно-сосудистой системы, но и другие (острые респираторные инфекции, новые диагнозы неясной этиологии – в связи с эпидемией COVID-19). Назначение лечения таким пациентам, имеющим сопутствующие ССЗ, должно учитывать совместимость препаратов и другие особенности процесса восстановления. Поэтому, все доступные протоколы лечения включены в исследование для построения модели прогнозирования укрупненных групп заболеваний по МКБ-10 на основе слабоструктурированных данных ЭМК.

С помощью разработанного программно-аппаратного комплекса обработано 364020 протоколов посещений пациентов с 01 октября по 31 декабря 2021 года. Объем XML-файлов варьировался от 3 КБ до 1008 КБ (в зависимости от степени наполненности). Объединенные данные по всем доступным протоколам сформированы в единую БД. Таким образом, получены данные амбулаторных карт пациентов в соответствии с иерархической моделью, содержащие наиболее информативные блоки слабоструктурированных документов МИС, необходимых для интеллектуальной обработки больших данных и разработки систем искусственного интеллекта.

Проблема прогнозирования диагноза заболевания в рамках теории интеллектуального анализа данных относится к задаче классификации. Для описания формальной математической постановки задачи классификации

укрупненных групп заболеваний по МКБ-10 на основе слабоструктурированных данных ЭМК, введем следующие обозначения:

$d \in D' = \{D_{comp} \cup D_{obj}\}$ - множество текстовых документов: ***жалобы пациентов и данные объективного осмотра на приеме***;

$y \in Y$ - множество меток классов: ***укрупненные группы заболеваний по МКБ-10*** пациентов с ССЗ ($|Y| = 7$), причем:

- y_1 - «**I1** Болезни, характеризующиеся повышенным кровяным давлением»;
- y_2 - «**I2** Ишемическая болезнь сердца, Легочное сердце и нарушения легочного кровообращения»;
- y_3 - «**I4** Другие болезни сердца»;
- y_4 - «**I6** Цереброваскулярные болезни»;
- y_5 - «**I8** Болезни вен, лимфатических сосудов и лимфатических узлов»;
- y_6 - «**J0** Острые респираторные инфекции верхних дыхательных путей»;
- y_7 - «**U0** Временные обозначения новых диагнозов неясной этиологии»;

$D'^{l} = (d_i, y_i)_{i=1}^{l}$- обучающая выборка;

$y_i = y(d_i), \quad y: D' \to Y$ – неизвестная зависимость.

Постановка задачи многоклассовой классификации:

Необходимо построить алгоритм $a: D' \to Y$, приближающий зависимость y на всем множестве D' и позволяющий классифицировать поступающие жалобы новых пациентов по соответствующим кодам МКБ некоторым способом с точностью eps.

Таблица 3.1 – Распределение данных по семи группам заболеваний по коду МКБ

№	Код МКБ	Название	Количество записей
1	**I1**	Болезни, характеризующиеся повышенным кровяным давлением	13 463
2	**I6**	Цереброваскулярные болезни	7 166
3	**I2**	Ишемическая болезнь сердца, Легочное сердце и нарушения легочного кровообращения	6 661
4	**I4**	Другие болезни сердца	2 244
5	**I8**	Болезни вен, лимфатических сосудов и лимфатических узлов, не классифицированные в других рубриках	1 021
6	**J0**	Острые респираторные инфекции верхних дыхательных путей	892
7	**U0**	Временные обозначения новых диагнозов неясной этиологии	784

Предобработка данных.

В связи с тем, что примеры оформления различными врачами раздела «Жалоб пациентов и данных объективного осмотра» не поддаются четкому

разделению по форме, встает вопрос в том, как находить близкие по диагнозу заболевания, ответить на который помогут методы обработки естественных языков.

Проведем некоторую предварительную обработку данных для построения прогнозных моделей. Во-первых, исключим все протоколы ЭКГ, анализа крови и т.д. (оставим только протоколы приема пациентов, которые включают жалобы). Во-вторых, удалим записи с пропущенными значениями и оценим распределение записей с жалобами пациентов по диагнозам.

В-третьих, исключим записи, в которых длина строки с жалобами пациента меньше 100 символов. Также выполнено разбиение датасета на семь основных групп по коду МКБ длины два. Для каждого тектового описания жалоб пациента в исходном массиве данных поставлен один диагноз по коду МКБ, в связи с чем в проводимом исследовании речь идет о задаче многоклассовой классификации.

Итоговый датасет для исследования содержит 32 231 запись. На рисунке 1 представлены два примера слабоструктурированных текстов жалоб пациентов. В протоколе посещения в полях с жалобами пациентов и данными объективного осмотра на приеме можно выделить пять специфических групп:

1. Опечатки и грамматические ошибки (выделены красным цветом).

Рисунок 3.1 – Примеры слабоструктурированных текстовых данных жалоб пациентов

2. Медицинские аббревиатуры и сокращения терминов на английском и русском языке (выделены желтым цветом). Некоторые аббревиатуры и термины многозначны, то есть могут иметь разные значения в зависимости от контекста (например, «САГ» может обозначать «Синдром антифосфолипидных антител», «Сердечная артерия головного мозга», «Специфический антиген гонококка» и т.д.). Такая многозначность создает сложности при автоматической интерпретации и классификации текста.

3. Числовые показатели (выделены зеленым цветом). Как правило, измеренные объективные показатели состояния здоровья пациента очень важны и всегда должны учитываться при постановке диагноза (температура, давление и т.д.) Однако, некоторые из них могут повторяться в тексте (при повторном измерении значений спустя определенный период), что приводит к проблеме корректного учета всех значений.

4. Произвольные сокращения слов (выделены голубым цветом). Например, «с-м» может означать «симптом» или «синдром».

5. Оценочная характеристика потенциальных ключевых слов и фраз (выделена фиолетовым цветом). Например, характеристики «отрицательный», «не затруднено» и другие. Данные характеристики влияют на постановку диагноза и выражаются как текстово, так и символьно.

Таким образом, для корректного распознавания и интерпретации медицинского текста необходимо учитывать контекст, предыдущую историю болезни пациента, результаты дополнительных обследований и другие факторы, определяющие специфику данной предметной области. Анализ медицинского текста осложняется наличием специфических терминов, аббревиатур, сокращений, неоднозначных понятий и сложных медицинских концепций, которые могут быть непонятны для стандартных моделей NLP. Кроме того, в открытом доступе находится только ограниченный объем размеченных данных, а для обучения моделей NLP требуются обучающие выборки больших размеров. Доступ к размеченным данным ограничен из-за конфиденциальности информации и сложности разметки медицинских текстов. Для преодоления этих проблем необходимо разработать специализированные методы и модели, учитывающие особенности медицинского текста, а также использовать техники передачи обучения (трансферное обучение, предобученные модели) и другие подходы для работы с ограниченным объемом данных.

3.2. Алгоритмы обработки естественного языка формирования векторного представления данных ЭМК

Для построения прогнозных моделей укрупненных групп заболеваний на основе слабоструктурированных текстовых данных жалоб пациентов и данных объективного осмотра из ЭМК необходимо представить текст на входе вектором признаков. Существует множество подходов к извлечению признаков, в рамках данного этапа используется модель мешка слов и TF-IDF векторизация.

- **Модель мешка слов** (Bag of Words)

Модель Bag of Words (мешок слов) - это метод представления текстовой информации в виде множества слов, без учета их порядка и связей между ними (рисунок 2.2). Данная модель используется в задачах обработки естественного языка, таких как классификация текстов, анализ тональности, извлечение ключевых слов и др.

Процесс реализации модели Bag of Words:

1. Разбиваем текст на отдельные слова и создаем словарь, где каждое слово имеет уникальный индекс.

2. Для каждого документа создаем вектор, где каждый элемент соответствует слову из словаря, а значение этого элемента равно количеству вхождений данного слова в документ.

3. Объединяем все векторы документов в матрицу, где каждая строка соответствует одному документу.

4. Используем полученную матрицу для обучения модели машинного обучения.

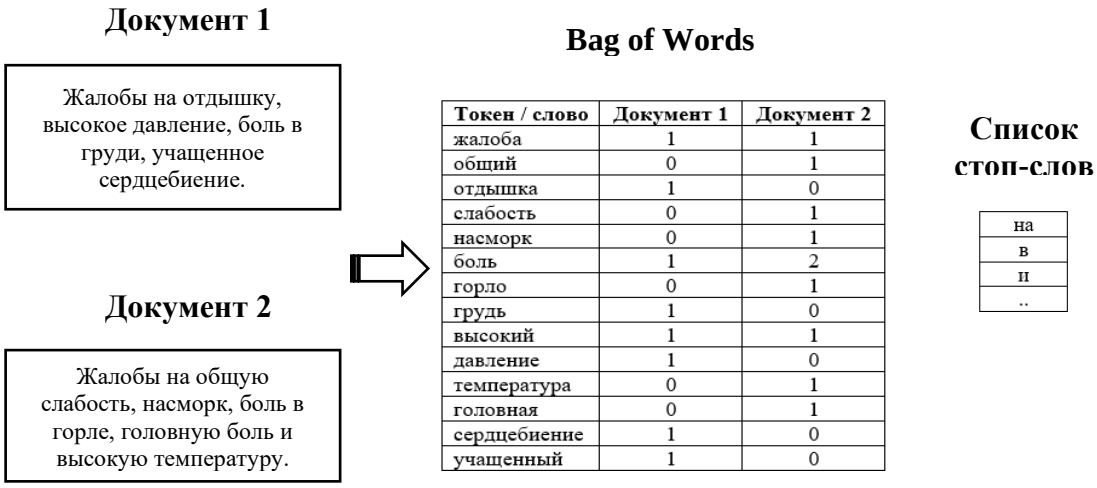

Рисунок 3.2 – Пример реализации модели мешка слов (Bag of Words)

Реализовать модель Bag of Words на языке Python можно используя метод CountVectorizer из библиотеки Sklearn.

Стоит отметить, что набор униграмм (набор токенов длинны n=1) не может фиксировать фразы и выражения из нескольких слов, игнорируя любую зависимость от порядка слов. В связи с этим, можно использовать наборы биграмм (n=2), в котором подсчитываются вхождения пар последовательных слов. В качестве альтернативы можно рассмотреть также набор n-грамм символов, представление, которое будет характеризовать устойчивые фразы.

Кроме того, в большинстве документов обычно используется очень небольшое подмножество слов, используемых в корпусе, поэтому результирующая матрица будет иметь много нулевых значений признаков (обычно более 95%). Чтобы иметь возможность хранить такую матрицу в памяти,

а также ускорить алгебраические операции с матрицей/векторами, обычно используют разреженное представление, доступное в пакете scipy.sparse.

- **TF-IDF векторизация**

В большом текстовом корпусе некоторые слова будут присутствовать в большом количестве (например, слова «данный», «это» и другие в русском языке) и при этом нести очень мало значимой информации о фактическом содержании документа. В связи с этим, необходима модификация алгоритма, чтобы очень частые термины не зашумляли частоты более редких, но более важных терминов.

TF-IDF используется для оценки важности слова в контексте документа, являющегося частью коллекции документов. Вес некоторого слова пропорционален частоте употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции.

TF (*term frequency* — частота слова) — отношение числа вхождений слова к общему числу слов в документе. Таким образом, чем чаще слово встречается в документе, тем выше его TF:

$$tf(t,d) = \frac{n_t}{\sum_k n_k}, \tag{3.1}$$

где n_t есть число вхождений слова t в документ d, а в знаменателе $\sum_k n_k$ — общее число слов в данном документе.

IDF (*inverse document frequency* — обратная частота документа) — логарифм отношения числа документов в коллекции к числу документов, содержащих данное слово. Таким образом, чем реже слово встречается в коллекции, тем выше его IDF:

$$idf(t,D) = log\frac{|D|}{|\{d_i \in D | t \in d_i\}|}, \tag{3.2}$$

где$|D|$ — число документов в коллекции; $|\{d_i \in D | t \in d_i\}|$ — число документов из коллекции D, в которых встречается t (когда $n_t \neq 0$).

Таким образом, мера TF-IDF имеет вид:

$$tf - idf(t,d,D) = tf(t,d) \times idf(t,D). \tag{3.3}$$

Мера TF-IDF позволяет выделить ключевые слова, которые наиболее характерны для данного документа или коллекции документов. Применение TF-IDF векторизации позволяет улучшить качество классификации текстовых данных, так как учитывает не только наличие слов в документе, но и их значимость для данного контекста.

Реализовать модель TF-IDF на языке Python можно используя метод TfidfVectorizer из библиотеки Sklearn. Для каждого 32 231 текста жалоб сгенерировано 56 062 признаков (TF-IDF score для unigrams, bigrams и др).

3.3. Алгоритмы машинного обучения для прогнозирования укрупненных групп заболеваний

Первый подход, реализованный для решения задачи диагностирования групп заболеваний - использование моделей машинного обучения таких как случайный лес (Random Forest), метод опорных векторов (Support Vector Machine), наивный байесовский классификатор (Naive Bayes) и логистической регрессии (Logistic regression) для классификации.

Процесс построения моделей прогнозирования укрупненных групп заболеваний схематично представлен на рисунке 3.3 и состоит из следующих этапов:

Процесс 1: Предобработка данных.

На первом этапе выполняется числовое кодирование целевой переменной – названия семи групп заболеваний по МКБ. Задается словарь стоп-слов из русскоязычного корпуса библиотеки nltk и задается минимальная и максимальная длина *n*-грам от 1 до 5.

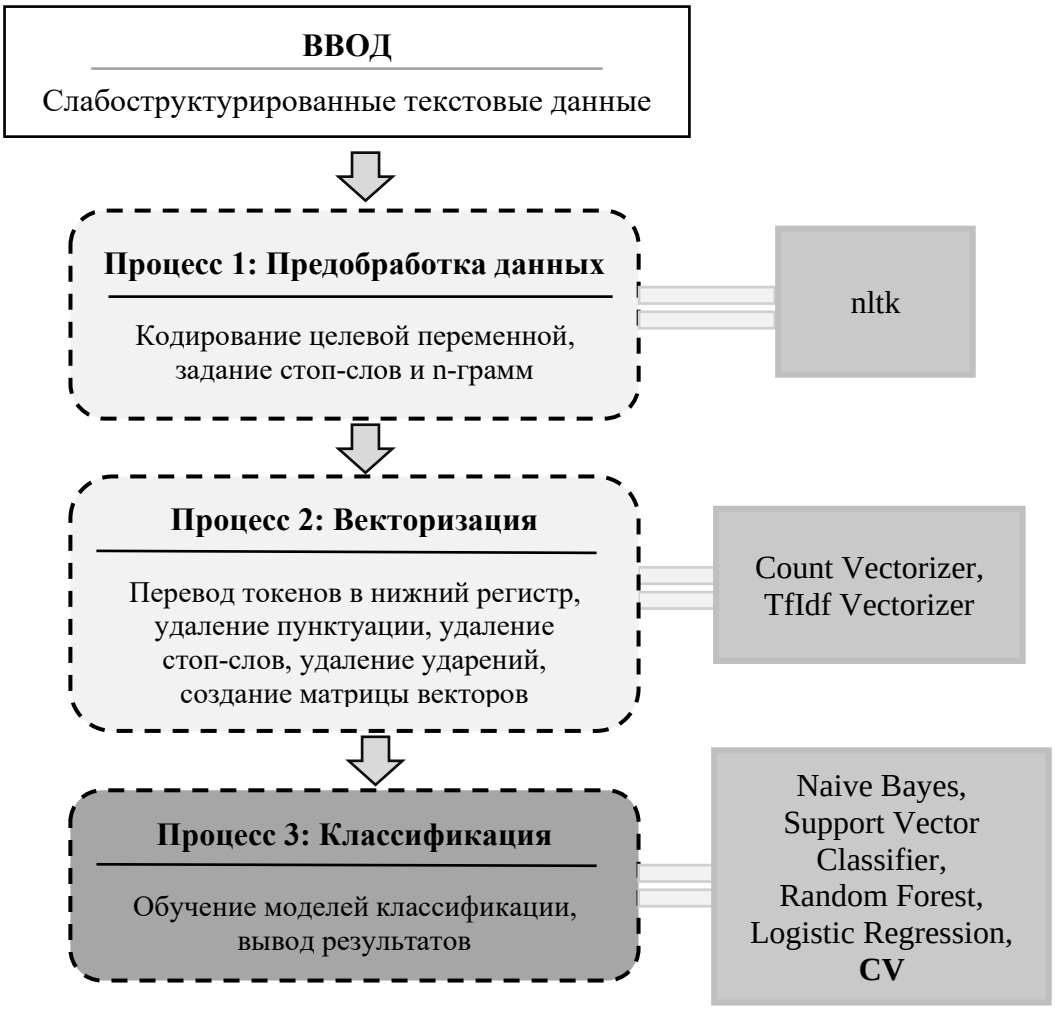

Рисунок 3.3 – Схема процесса построения моделей прогнозирования укрупненных групп заболеваний на основе методов машинного обучения

Процесс 2: Векторизация.

Предварительно выполняются операции перевода токенов в нижний регистр, удаления пунктуации, удаления стоп-слов, удаление ударений и др. Коллекция неструктурированных текстовых документов с жалобами пациентов преобразуется в матрицу векторов с помощью методов CountVectorizer и TfidfVectorizer (модель мешка слов, bag-of-words) для *n*-грам длины не более трех.

Процесс 3: Классификация.

Полученные векторные текстовые эмбеддинги разбиваются на тренировочную и тестовую выборки (train_size = 0.8, test_size = 0.2), для обучения используются классификаторы LogisticRegression, MultinomialNB, RandomForest и LinearSVC с поддержкой кросс-валидации. На следующем этапе выполняется подсчет метрик. Метод predict_proba модели LogisticRegression используется для получения предсказанной вероятности принадлежности выборки к одной из семи групп заболеваний по МКБ.

Рассмотрим основные теоретические аспекты группы алгоритмов машинного обучения, реализуемых в рамках процесса построения моделей прогнозирования укрупненных групп заболеваний.

1. Метод опорных векторов (Support Vector Machine)

Метод опорных векторов (Support Vector Machine, SVM) - это линейный классификатор, который осуществляет поиск гиперплоскости в n-мерном пространстве, которая разделяет классы объектов. Гиперплоскость выбирается таким образом, чтобы она максимизировала расстояние между ближайшими объектами разных классов, которые называются опорными векторами.

Гиперплоскость l, уравнение которой задается соотношением

$$l: \quad w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p = 0, \tag{3.4}$$

в котором коэффициенты w_0, w_1, ..., w_p определяются из оптимизационной задачи:

$$\begin{cases} \dfrac{1}{2}(\omega, \omega) \to \boldsymbol{min}_{\omega, \omega_0}, \\ y_i\big((\omega, \boldsymbol{x_i}) - \omega_0\big) \geq 1, \;\; i \in \{1, \ldots, l\}, \end{cases} \tag{3.5}$$

называется ***оптимальной разделяющей гиперплоскостью***.

Объекты обучающей выборки, для которых выполняется условие:

$$(\omega, \boldsymbol{x_i}) - \omega_0 = y_i, \tag{3.6}$$

называются ***опорными векторами*** (support vector).

В случае, если данные ***линейно неразделимы***, вводится дополнительный набор переменных $\xi_i \geq 0$, $i \in \{1, \ldots, l\}$, каждая из которых характеризует величину ошибки на обучающем объекте x_i.

Если ослабить ограничения, то получим следующую оптимизационную задачу:

$$\begin{cases} \dfrac{1}{2}(\omega,\omega) + \mathrm{C}\sum\limits_{i=1}^{l}\xi_i \to \boldsymbol{min}_{\omega,\omega_0,\xi}, \\ y_i\big((\omega,\boldsymbol{x_i}) - \omega_0\big) \geq 1 - \xi_i, \;\; i \in \{1,\ldots,l\}, \\ \xi_i \geq 0, \;\; i \in \{1,\ldots,l\}, \end{cases} \tag{3.7}$$

где C – некоторый положительный параметр.

Данная оптимизационная задачи также стремится максимизировать ширину разделяющей полосы и сумму ошибок ξ_i с некоторым положительным весом C. Параметр C является управляющим параметром модели и контролирует баланс между максимизацией разделяющей полосы и суммарной ошибкой в классификации обучающих данных.

Гиперплоскость l, в котором коэффициенты w_0, w_1, ..., w_p определяются из оптимизационной задачи (3.7), называется разделяющей **гиперплоскостью с мягким зазором** (soft margin hyperplane) или почти-разделяющей гиперплоскостью.

В случае, если данные ***нелинейной*** природы, вводится все элементы обучающей выборки вкладываются в пространство X более высокой размерности с помощью специального отображения $\varphi\colon \mathbb{R}^p \to H$. При этом отображение φ выбирается так, что в новом пространстве X выборка линейно разделима.

Пусть $\varphi\colon \mathbb{R}^p \to H$, где H – **спрямляющее пространство** со скалярным произведением $(\cdot,\cdot)_H$ - функция, переводящая объект $x_i \in \mathbb{R}^p$ в пространство более высокой размерности, где данные линейно разделимы.

Функция $K\colon \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ называется **ядром** (kernel), если она представима в виде

$$K(x,x') = \big(\varphi(x),\varphi(x')\big)_H. \tag{3.8}$$

В новом признаковом пространстве разделяющая поверхность является линейной, однако после ее проецирования на исходные пространства она окажется нелинейной.

Существует несколько стандартных ядер, которые встроены во множество пакетов и библиотек.

- *Линейное*: $K\big(x_i,x_j\big) = x_i^T x_j$;
- *Полиномиальное*: $K\big(x_i,x_j\big) = \big(1 + x_i^T x_j\big)^p$;
- *Гауссово (радиальная функция)*: $K\big(x_i,x_j\big) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$;
- *Сигмоидальная функция:* $K\big(x_i,x_j\big) = tanh\,(\beta_0 x_i^T x_j + \beta_1)$.

Таким образом, **алгоритм** работы метода опорных векторов имеет вид:

1. Подготавливаем данные для обучения, разбивая их на тренировочную и тестовую выборки.

2. Выбираем ядро, которое определяет функцию преобразования данных в более высокую размерность, где данные становятся линейно разделимыми.

3. Обучаем модель на обучающей выборке, используя метод оптимизации, который находит гиперплоскость, максимизирующую расстояние между опорными векторами разных классов.

4. Проверяем точность модели на тестовой выборке.

5. Если заданная точность не достигнута, корректируем параметры модели и повторяем шаги 3-4.

Преимущества метода SVM:

- Может работать с большим количеством признаков и объектов.
- Хорошо работает в условиях разреженных данных.
- Способен обрабатывать нелинейные зависимости между признаками.

Недостатки метода SVM:

- Не подходит для задач, где количество объектов каждого класса сильно отличается.
- Не всегда может давать интерпретируемые результаты.

2. Наивный байесовский классификатор (Naive Bayes)

Наивный байесовский классификатор (Naive Bayes Classifier) – это алгоритм машинного обучения, который используется для классификации объектов на основе их признаков. Данный классификатор основан на теореме Байеса, которая позволяет вычислить вероятность принадлежности объекта к определенному классу на основе его признаков:

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n|y)}{P(x_1, \dots, x_n)}. \quad (3.9)$$

Используя наивное предположение об условной независимости, что

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y), \qquad \forall i, \quad (3.10)$$

можно упростить исходное соотношение:

$$P(y|x_1, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \dots, x_n)}. \quad (3.11)$$

С учетом того, что вероятность $P(x_1, \dots, x_n)$ является константой для заданного входного набора, то можно использовать следующее правило классификации:

$$\hat{y} = \arg\max_y P(y) \prod\nolimits_{i=1}^{n} P(x_i|y) \quad (3.12)$$

и максимальную апостериорную оценку (MAP) для оценки $P(y)$ и $P(x_i|y)$, причем $P(y)$ — это относительная частота класса в обучающем наборе данных.

Существуют три основных типа наивного байесовского классификатора:

1. Бернулли. Классификатор используется для бинарных данных, где каждый признак может принимать только два значения (0 или 1). Примеры входных данных - текстовые документы, где каждое слово может быть либо в документе, либо нет.

2. Мультиномиальный. Классификатор используется для данных, где каждый признак может принимать несколько значений. Примеры входных данных - частоты слов в текстовых документах.

3. Гауссовский. Классификатор используется для непрерывных данных, где каждый признак имеет нормальное распределение. Примеры входных данных - рост и вес людей.

Алгоритм работы наивного байесовского классификатора:

1. Подготавливаем данные для обучения, разбивая их на тренировочную и тестовую выборки.

2. Определяем тип задачи: бинарная или многоклассовая классификация.

3. Выбираем тип наивного байесовского классификатора: Бернулли, мультиномиальный или гауссовский.

4. Обучаем модель на тренировочной выборке, вычисляя вероятности принадлежности каждого признака к каждому классу.

5. Для новых объектов вычисляем вероятности принадлежности к каждому классу на основе их признаков, используя формулу теоремы Байеса.

6. Классифицируем объекты, выбирая класс с наибольшей вероятностью.

Преимущества наивного байесовского классификатора:

- Простая реализация и высокая скорость обучения.
- Хорошо работает с большим количеством признаков и объектов.
- Способен обрабатывать разреженные данные.

Недостатки наивного байесовского классификатора:

- Предполагает независимость признаков, что не всегда соответствует реальным данным.
- Не учитывает взаимосвязи между признаками.
- Может давать низкую точность в задачах с сильными зависимостями между признаками.

3. Логистическая регрессия (Logistic Regression)

Логистическая регрессия - это метод машинного обучения для классификации данных. Он использует логистическую функцию для прогнозирования вероятности отнесения объекта к определенному классу.

В процессе обучения модели логистической регрессии, входные данные (например, TF-IDF векторизация текстовых данных) и соответствующие им метки классов (например, категории текстов) используются для нахождения оптимальных весов параметров модели. Веса определяют, как входные данные будут взвешиваться для прогнозирования вероятности отнесения объекта к определенному классу.

После обучения модель может быть использована для классификации новых данных. Для каждого объекта модель вычисляет вероятность его отнесения к каждому из возможных классов, а затем выбирает класс с наибольшей вероятностью.

Бинарная классификация методом логистической регрессии

1. Вероятность положительного класса $P_+ = P(y_i = 1|x)$:

$$P_+(x) = \frac{1}{1 + e^{-y(x,w)}}; \tag{3.13}$$

2. Для оптимизации параметров используется логистическая функция потерь с функцией регуляризации $r(w)$*:*

$$\min_w C \sum_{i=1}^{n} \left(-y_i \log\left(P(x_i)\right) - (1 - y_i) \log\left(1 - P(x_i)\right)\right) + r(w), \tag{3.14}$$

где функция регуляризации $r(w)$ определена в соответствии с таблицей 3.2.

Таблица 3.2 – Типы функций регуляризации

Штраф	$\boldsymbol{r(w)}$
None	0
ℓ_1	$\|w\|_1$
ℓ_2	$\frac{1}{2}\|w\|_2^2 = \frac{1}{2}w^T w$
ElasticNet	$\frac{1-\rho}{2}w^T w + \rho\|w\|_1$

Многоклассовая классификация методом логистической регрессии

1. Вероятность отнесения объекта x_i к классу $y_i = k$ вычисляется как

$$P(y_i = k|x_i) = \frac{e^{(x_i W_k + W_{o,k})}}{\sum_{j=0}^{K-1} e^{(x_i W_j + W_{o,j})}}; \tag{3.15}$$

2. Для оптимизации параметров используется следующая функция потерь*:*

$$\min_w -C \sum_{i=1}^{n} \sum_{k=0}^{K-1} [y_i = k] \log\left(P(y_i = k|x_i)\right) + r(W), \tag{3.16}$$

где функция $[P]$ представляет собой скобку Айверсона, которая принимает значение 0, если P является ложным, в противном случае принимает значение 1, а функция регуляризации $r(W)$ определена в соответствии с таблицей 3.3.

Таблица 3.3 – Типы функций регуляризации

Штраф	$\boldsymbol{r(W)}$
None	0
ℓ_1	$\|W\|_1 = \sum_{i=1}^{n}\sum_{k=0}^{K-1}\|W_{i,j}\|$
ℓ_2	$\frac{1}{2}\|W\|_F^2 = \frac{1}{2}\sum_{i=1}^{n}\sum_{k=0}^{K-1}W_{i,j}^2$
ElasticNet	$\frac{1-\rho}{2}\|W\|_F^2 + \rho\|W\|_1$

Логистическая регрессия широко используется в задачах классификации, таких как анализ тональности текстов, определение спама в электронной почте и других.

Преимущества:

1. Интерпретируемость: Логистическая регрессия позволяет интерпретировать вклад каждого параметра в прогноз, что может быть полезно при анализе данных.

2. Масштабируемость: метод может быть применен к большим наборам данных и быстро обучен с помощью методов оптимизации.

Недостатки:

1. Логистическая регрессия использует линейную функцию для прогнозирования вероятности класса, что может быть недостаточно для сложных задач классификации.

2. Логистическая регрессия чувствительна к выбросам в данных, что может привести к неправильным прогнозам.

3. Логистическая регрессия может терять качество от мультиколлинеарности, когда два или более параметров сильно коррелируют друг с другом, что может привести к неустойчивым весам параметров и некорректным прогнозам.

4. Алгоритм случайного леса (Random Forest)

Случайный лес (Random Forest) - это алгоритм машинного обучения, который основан на идее комбинирования нескольких решающих деревьев (decision trees) для получения более точных результатов.

Алгоритм случайного леса:

1. Из обучающей выборки случайным образом выбирается подвыборка данных.

2. На данной подвыборке строится решающее дерево с помощью алгоритма CART.

3. Шаги 1-2 повторяются несколько раз, чтобы создать несколько решающих деревьев.

4. Для каждого нового объекта, который нужно классифицировать, каждое дерево решений строит прогноз и относит его к одному из классов.

5. Результаты всех деревьев объединяются по заданному правилу агрегирования.

Алгоритм построения решающего дерева

0. Вход: Обучающий набор данных $x_1, x_2, \ldots, x_n$, состоящем из n элементов с p предикторами $X_1, X_2, \ldots, X_p$ каждый, и откликом Y.

1. Пусть на вход подается множество объектов X. Среди p предикторов выбрать тот, для которого прирост информации максимален:

$$arg \max_{Q \in \{X_1, X_2, \ldots, X_p\}} IG(Y|Q), \tag{3.17}$$

где прирост информации $IG(\Omega|\Theta) = H(\Omega) - H(\Omega|\Theta)$, а H – мера неопределенности.

2. Пусть выбран предиктор X_i, принимающий на наборе данных X ровно t уникальных значений.

Выполнить разделение набора данных X на подмножества $S_1, \ldots, S_t$ по уникальным значениям предиктора X_i.

3. Для каждого множества S_i, $i \in \{1, 2, \ldots, t\}$, если энтропия по отклику не равна нулю, повторить шаги 1 и 2.

В качестве критериев измерения неопределенности (информативности), как правило выбирают энтропию и индекс Джини.

Преимущества:

- Высокая устойчивость к переобучению благодаря случайному выбору подвыборок данных и признаков.
- Высокая точность классификации из-за комбинации нескольких решающих деревьев.

Недостатки:

- Сложность интерпретации результатов.
- Неэффективность для работы с большими объемами данных.

Метрики оценки качества классификации

Для оценки качества алгоритмов многоклассовой классификации, как правило, используют следующие метрики:

1. Accuracy (точность): отношение числа правильных прогнозов к общему числу прогнозов. Данная метрика может быть недостаточно информативной, если классы несбалансированы.

$$accuracy = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} [\hat{y}_i = y_i], \quad (3.18)$$

где $\hat{y}_i$ – предсказанный класс объекта, y_i – истинный класс объекта, а $n_{samples}$ – общее количество примеров в тестовой выборке.

2. Balanced accuracy (сбалансированная точность): среднее арифметическое точности для каждого класса. Таким образом, сбалансированная точность показывает, насколько хорошо алгоритм работает для всех классов в данных, даже если некоторые классы имеют меньшее количество объектов, чем другие.

$$balanced\_accuracy = \frac{1}{\sum \widehat{w}_i} \sum_{i=0}^{n_{samples}-1} [\hat{y}_i = y_i]\widehat{w}_i, \quad (3.19)$$

где $\widehat{w}_i$ – вес i-го класса, который является обратным отношением размера i-го класса ко всему объему выборки.

3. Precision (точность): в случае 2 классов данная метрика характеризует отношение числа правильных прогнозов положительного класса к общему числу прогнозов положительного класса. В случае 3 и более классов используют подходы к макро- и микро-усреднению.

Значение $Precision$ для класса k:

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k}, \quad (3.20)$$

где TP_k – истинно-положительные примеры класса k, FP_k - ложноположительные примеры класса k.

Макроусреднение метрики $Precision$ по всем классам:

$$MacroAvgPrecision = \frac{\sum_{k=1}^{K} \text{Precision}_k}{K}, \quad (3.21)$$

где K – общее количество классов.

Микроусреднение метрики $Precision$ по всем классам:

$$MicroAvgPrecision = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} TP_k + \sum_{k=1}^{K} FP_k}. \quad (3.22)$$

4. Recall (полнота): в случае 2 классов данная метрика характеризует отношение числа правильных прогнозов положительного класса к общему числу

объектов положительного класса в данных. В случае 3 и более классов используют подходы к макро- и микро-усреднению.

Значение $Recall$ для класса k:

$$\text{Recall}_k = \frac{TP_k}{TP_k + FN_k}, \tag{3.23}$$

где TP_k – истинно-положительные примеры класса k, FN_k - ложноотрицательные примеры класса k.

Макроусреднение метрики $Recall$ по всем классам:

$$MacroAvg\text{Recall} = \frac{\sum_{k=1}^{K} \text{Recall}_k}{K}, \tag{3.24}$$

где K – общее количество классов.

Микроусреднение метрики Recall по всем классам:

$$MicroAvg\text{Recall} = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} TP_k + \sum_{k=1}^{K} FN_k}. \tag{3.25}$$

5. *F1-score*: гармоническое среднее между precision и recall, показывает баланс между точностью и полнотой:

$$Macro\text{F1} = 2\left(\frac{MacroAvgPrecision * MacroAvg\text{Recall}}{MacroAvgPrecision^{-1} + MacroAvg\text{Recall}^{-1}}\right), \tag{3.26}$$

$$Micro\text{F1} = 2\left(\frac{MicroAvgPrecision * MicroAvg\text{Recall}}{MicroAvgPrecision^{-1} + MicroAvg\text{Recall}^{-1}}\right). \tag{3.27}$$

Кроме того, для полноты представления результатов классификации строят матрицу ошибок (*confusion matrix*): таблица, которая показывает, сколько объектов каждого класса правильно или неправильно классифицировано.

Рассмотренные в рамках диссертационного исследования модели машинного обучения для решения задачи многоклассовой классификации обучены на сформированном уникальном узкоспециализированном неразмеченном корпусе медицинских текстов. Кроме того, проведено сравнение данных моделей с использованием описанных метрик оценки качества работы алгоритмов. Результаты исследования эффективности алгоритмов прогнозирования укрупненных групп заболеваний на основе слабоструктурированных данных ЭМК представлены в пункте № 3.5.

3.4. Применение предобученных языковых моделей трансформеров для прогнозирования укрупненных групп заболеваний

Рассмотрим альтернативный подход прогнозирования укрупненных групп заболеваний с использованием русскоязычных моделей трансформеров BERT на слабоструктурированных медицинских текстах, который заключается в дообучении предварительно обученной нейронной сети с дополненными слоями классификатора на размеченном наборе данных.

BERT (Bidirectional Encoder Representations from Transformers) — это модель глубокого обучения, которая используется для решения задач обработки естественного языка, таких как классификация текстов, ответы на вопросы, генерация текстов и т.д.

Структура BERT состоит из двух основных компонентов: энкодера и декодера (рисунок 3.4). Энкодер представляет собой набор слоев трансформера, которые обрабатывают входные данные и создают их векторное представление. Декодер используется для выполнения конкретной задачи, например, классификации или генерации текста.

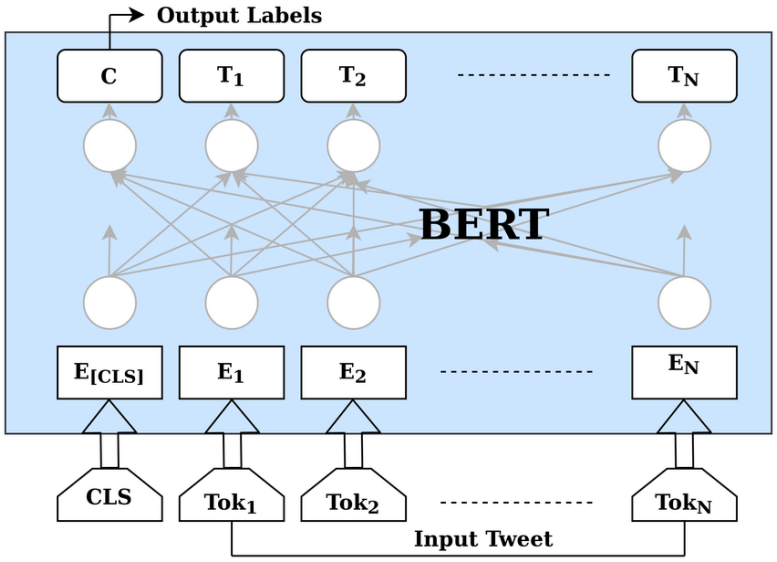

Рисунок 3.4 – Архитектура модели BERT для прогнозирования класса заболевания

Энкодер BERT состоит из 12 слоев трансформера, каждый из которых содержит множество механизмов внимания и сверточных слоев. Входные данные проходят через эти слои в обратном направлении и в прямом направлении, что позволяет модели учитывать контекст и семантику слов в предложении.

Декодер BERT может быть настроен для выполнения различных задач обработки естественного языка. Например, для задачи классификации текстов он может использовать последний слой энкодера, который содержит векторное

представление всего входного текста, и добавить несколько полносвязных слоев для классификации.

В целом, структура BERT очень гибкая и адаптируется для выполнения различных задач обработки естественного языка. Эффективность данной языковой модели доказана во множестве исследований [38,43,47]. Способность модели учитывать контекст и семантику фразы позволяет работать с медицинскими текстами, где необходимо учитывать широкий контекст и множество возможных вариантов описания симптомов и заболеваний.

Рассмотрим общую схему процесса построения модели BERT для прогнозирования укрупненных групп заболеваний на основе глубокого обучения:

Процесс 1: Предобработка данных.

На первом этапе выполняется числовое кодирование целевой переменной – названия семи групп заболеваний по МКБ. Задается максимальный размер словаря num_words = 15000 и максимальная длина сообщения max_len = 200 в токенах, происходит выравнивание предложений исходного датасета до одинаковой длины (padding='post').

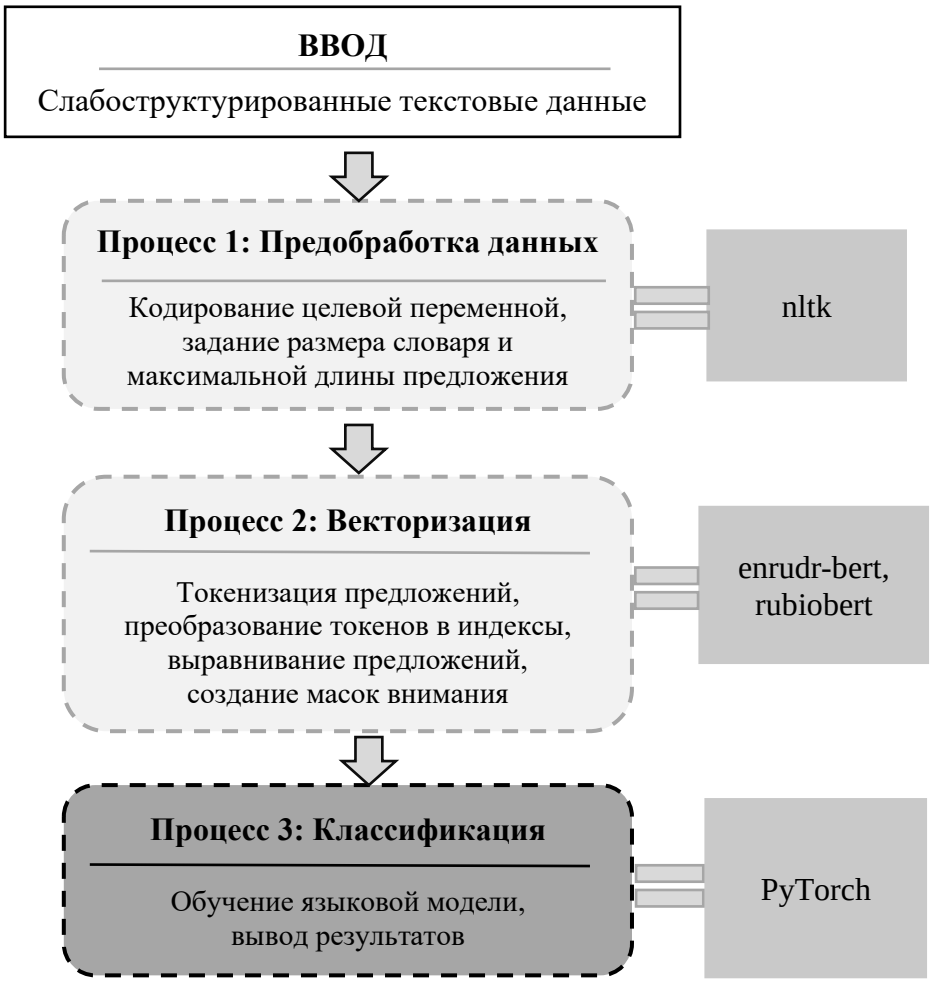

Рисунок 3.5 – Схема процесса построения модели BERT для прогнозирования укрупненных групп заболеваний на основе глубокого обучения

Процесс 2: Токенизатор.

Выполняется токенизация обучающей выборки с помощью модели EnRuDR-BERT [18], предварительно обученной на коллекции отзывов потребителей о приеме лекарств и модели RuBioBERT [47], предварительно обученной на корпусе свободно доступных текстов в области биомедицины.

Модель EnRuDR-BERT имеет общий размер словаря 119 547, включает в себя последовательность следующих блоков: входной слой вложений, который формирует 768-байтовое векторное представление токена; кодировщик, состоящий из 12 блоков трансформеров, включая слой внимания, полносвязные слои и слои нормализации; последний полносвязный слой – пулер.

Модель RuBioBERT имеет общий размер словаря 120 138. Стоит отметить, что изначально RuBERT включает выходной слой, который предсказывает замаскированные слова в тексте (Masked Word Prediction). Для решения задачи классификации группы заболеваний он заменен выходным слоем с семью выходами в соответствии с таблицей 3.1. Создается маска внимания для каждого примера обучающей выборки. Единицами заполняются те токены, которые нужно учитывать при обучении и вычислении градиентов, нулями заполняются те токены, которые следует пропустить.

Процесс 3: Тренировка модели

Векторные представления формируются с помощью входного слоя нейронной сети на основе списка словарных номеров текстовых токенов. Выполняется обучение и тестирование модели.

Таким образом, применение русскоязычной модели трансформеров BERT для анализа медицинских текстов имеет большой потенциал в решении задач диагностики и лечения заболеваний. Архитектура модели BERT способствует выявлению связей и зависимостей между словами в тексте, что позволяет ей эффективно анализировать сложные и слабоструктурированные медицинские данные, такие как отчеты врачей, истории болезней и симптоматика пациентов.

3.5. Исследование эффективности алгоритмов прогнозирования укрупненных групп заболеваний на основе слабоструктурированных данных ЭМК

В рамках данного диссертационного исследования рассмотрено применение следующих моделей машинного обучения: Random Forest, Multinomial Naive Bayes, Support Vector Machine, Logistic Regression и BERT.

1) Random Forest (RF). Для построения эффективной модели проведено исследование параметров с измерением accuracy и balanced_accuracy. Лучшая модель с параметрами n_estimators =100, max_dept=150 и criterion= 'entropy' показала точность 85,17 % и сбалансированную точность 80,91%. Наиболее эффективно модель RF определяет диагнозы «I1 Болезни, характеризующиеся повышенным кровяным давлением», «I8 Болезни вен, лимфатических сосудов

и лимфатических узлов» и «J0 Острые респираторные инфекции верхних дыхательных путей» (более 92%).

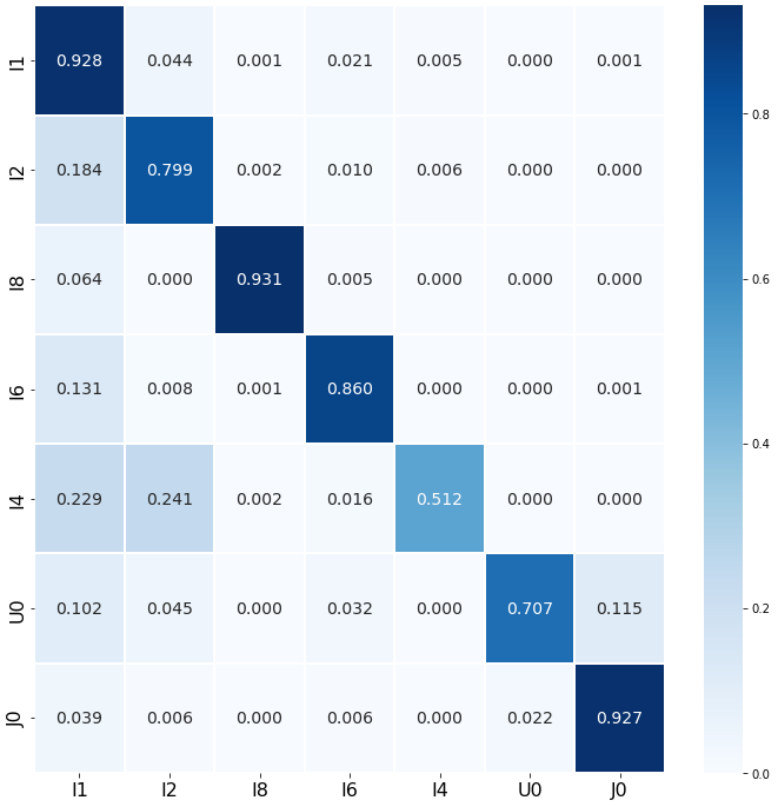

Рисунок 3.6 – Матрица ошибок для Random Forest

Наименее точно (около 51,2%) модель RF определяет диагноз «I4 Другие болезни сердца» и ожидаемо переносит данные протоколы жалоб пациентов к другим болезням, сходным с ССЗ.

2) Полиномиальный наивный байесовский классификатор (MultinomialNB) подходит для классификации с дискретными признаками. Для построения эффективной модели проведено исследование параметров с измерением accuracy и balanced_accuracy. Лучшая модель с параметрами fit_prior = “True” и alpha = 0.01 показала точность 82,95 % и сбалансированную точность 82,59%.

На рисунке 3.7 представлена матрица ошибок, которая демонстрирует то, что классификация заболеваний моделью MultinomialNB получена более равномерно, повышено качество идентификации класса «I4 Другие болезни сердца» в сравнении с моделью RF на 8%. При этом большинство заболеваний модель MultinomialNB определяет более эффективно в сравнении с другими (более 80,3% точности).

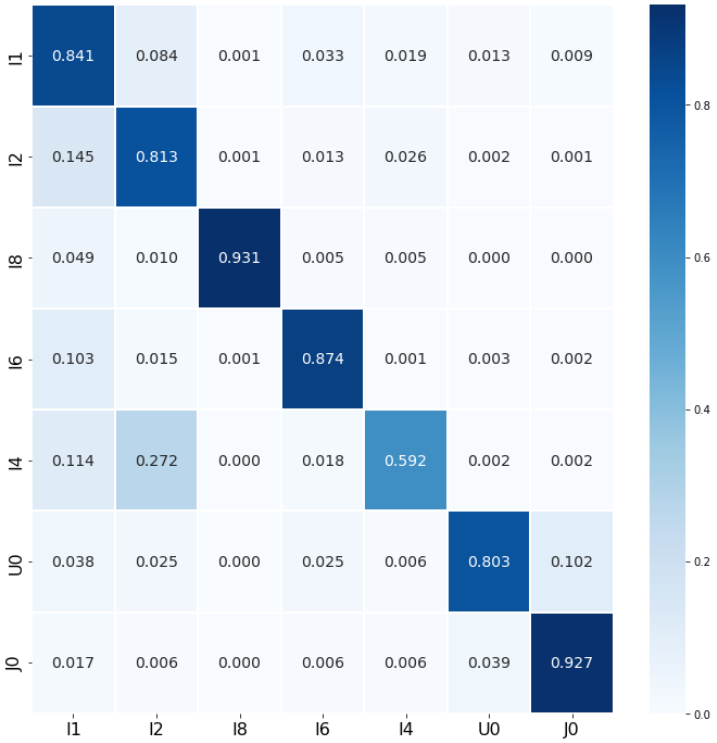

Рисунок 3.7 – Матрица ошибок для Multinomial Naive Bayes

3) Линейный метод опорных векторов.

Метод опорных векторов с линейным ядром (LinearSVC) обладает большей гибкостью при выборе штрафной функции и функции потерь, а также лучше масштабируется на больших наборах данных. Для построения эффективной модели LinearSVC проведено исследование параметров с измерением accuracy и balanced_accuracy. Лучшая модель с параметрами penalty='l2', loss='squared_hinge' и C=1 продемонстрировала точность 87,08 % и сбалансированную точность 85,01%.

На рисунке 3.8 представлена матрица ошибок модели LinearSVC, где классы «I1 Болезни, характеризующиеся повышенным кровяным давлением», «I6 Цереброваскулярные болезни», «I8 Болезни вен, лимфатических сосудов и лимфатических узлов» и «J0 Острые респираторные инфекции верхних дыхательных путей» определяются моделью с точностью более 90,7%. При этом, модель LinearSVC имеет более высокие показатели точности, чем MultinomialNB и Random Forest. Эффективность определения класса «I4 Другие болезни сердца» моделью LinearSVC выше в сравнении с моделью MultinomialNB на 9,2%.

Рисунок 3.8 – Матрица ошибок для LinearSVC

4) Логистическая регрессия

Модель Logistic Regression показала точность 87,17% и сбалансированную точность 85,20%. Данная модель машинного обучения продемонстрировала наибольшую сбалансированную точность среди рассмотренных подходов, в связи с этим модель логистической регрессии закреплена в качестве базовой модели для определения наиболее вероятной укрупненной группы заболеваний по МКБ-10 на основе жалоб пациентов.

5) Языковая модель BERT.

В рамках диссертационного иследования количество эпох для дообучения языковой модели BERT подбирается экспериментально (epoch = 2). В качестве предобученных моделей рассмотрены EnRuDR-BERT (отзывы потребителей о приеме лекарств) и RuBioBERT (корпус свободно доступных текстов в области биомедицины). В результате обучения моделей на жалобах и данных объективного осмотра пациентов наименьшая ошибка на обучающем и проверочном датасете получена на основе модели RuBioBERT и имеет следующие значения – train_loss: 0.5425, val_loss: 0.5644.

Функция softmax библиотеки torch используется для получения предсказанной вероятности принадлежности выборки к одной из семи групп заболеваний по МКБ. Классификация BERT на основе предобученной модели RuBioBERT показала точность 85,6 % и сбалансированную точность 81,79%. В целом модель показала сравнимые по качеству результаты с другими методами машинного обучения.

Рисунок 3.9 – Матрица ошибок для Logistic Regression

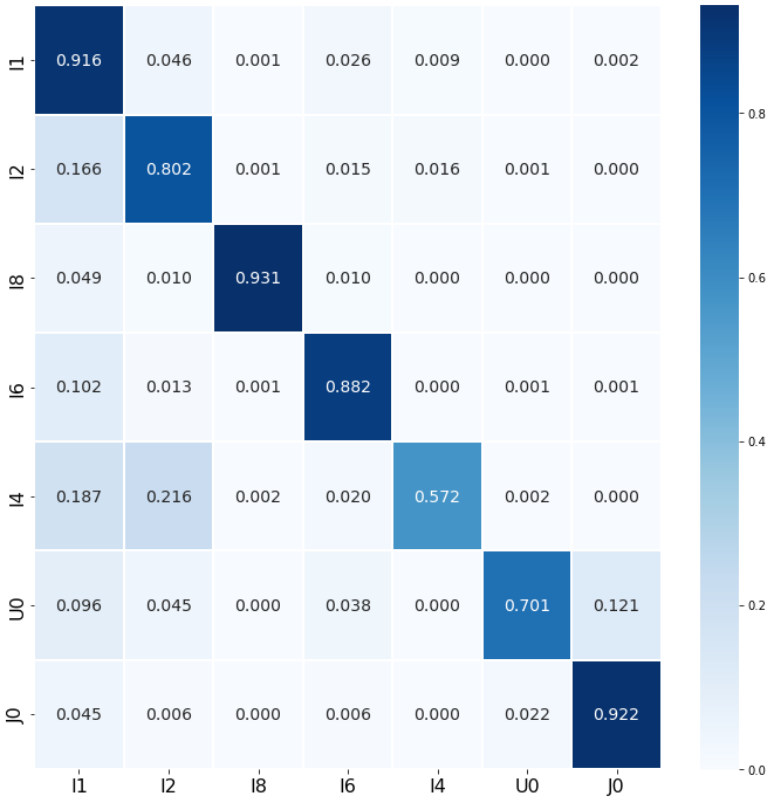

Рисунок 3.10 – Матрица ошибок для RuBioBERT

Сравнительный анализ методов при 5-крестной валидации показал, что по метрике accuracy наиболее эффективна модель логистической регрессии (Mean Accuracy = 85,96%, Mean Balanced Accuracy = 85,20%). При этом данная модель

имеет наименьшее стандартное отклонение при перекрестной проверке, что свидетельствует об устойчивости результата (таблица 3.4).

Таблица 3.4 – Сравнительный анализ моделей машинного обучения

Подход	**Mean Balanced Accuracy**	**Standard Deviation**
RandomForest	0.809161	0.005745
LinearSVC	0.850011	0.015853
MultinomialNB	0.825981	0.006703
LogisticRegression	**0.852052**	0.010730
EnRuDR-BERT	0.809521	0.005682
RuBioBERT	0.817935	0.004921

Примеры распределения вероятностей классов различных заболеваний моделью логистической регрессии нескольких жалоб пациентов представлены на рисунке 3.11. Жалобы выбраны из исходного набора данных случайным образом и содержат описание признаков ОРВИ, Covid-19 и различных сердечно-сосудистых заболеваний. Согласно комментариям терапевта, различия в вероятностях принадлежности к классам может обуславливаться тем, что описанные жалобы пациентов могут относиться к нескольким группам заболеваний. На практике врач принимает окончательное решение, опираясь на личный опыт и результаты дополнительных обследований.

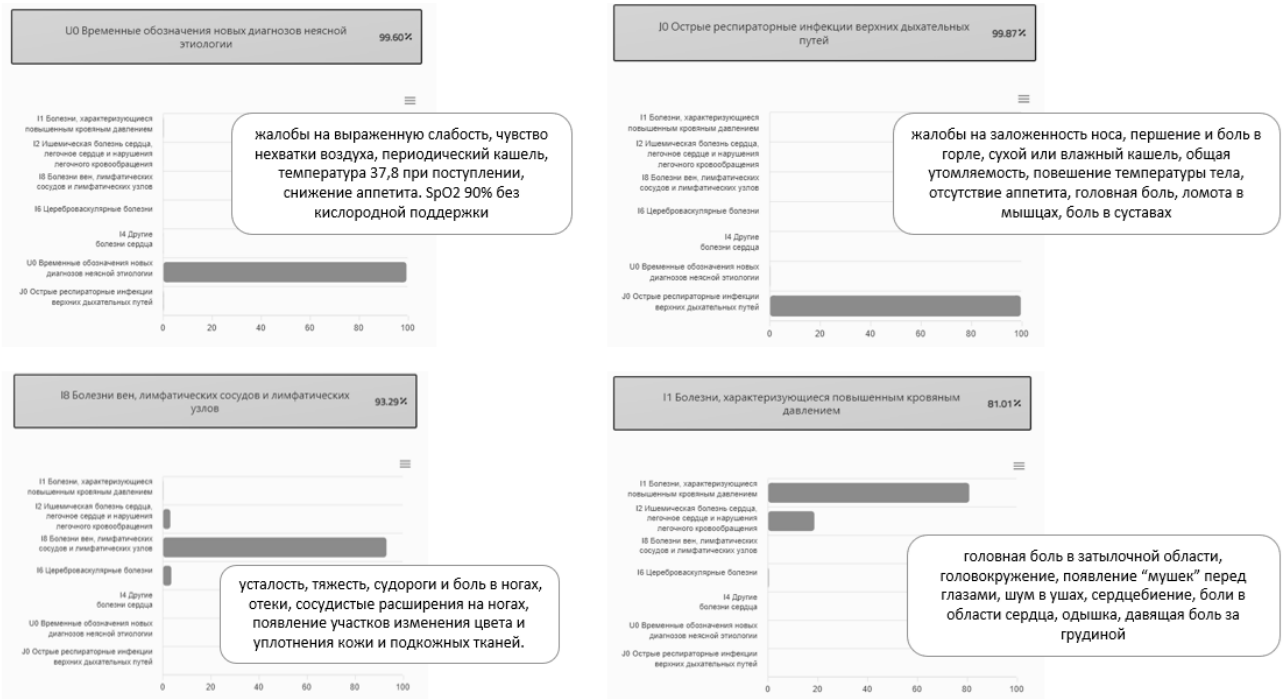

Рисунок 3.11 – Примеры классификации различных заболеваний и вероятности распределения классов для модели логистической регрессии

Таким образом, сравнительный анализ прогнозных моделей Random Forest, Multinomial Naive Bayes, Support Vector Machine, Logistic Regression и

BERT показал, что наиболее высокую точность классификации укрупненных групп заболеваний (средняя точность выше на 0,2%) продемонстрировал метод Logistic Regression – 85,20 %. При этом данная модель имеет наименьшее стандартное отклонение при перекрестной проверке (±1.07%), что свидетельствует об устойчивости результата прогнозирования. С другой стороны, модель Random Forest показала самую низкую точность классификации (80,91%) со стандартным отклонением ±0.57%. Применение предобученных языковых моделей на текстовом корпусе биомедицинской информации показало высокую точность классификации (Balanced Accuracy = 81,79%), однако не самую эффективную среди рассматриваемых методов.

Выводы третьей главы

1. Разработан метод и алгоритм прогнозирования группы заболеваний пациентов с ССЗ с использованием методов NLP для извлечения признаков из слабоструктурированной текстовой информации, а также группа моделей машинного обучения, которые используют уникальный узкоспециализированный неразмеченный корпус текстов и укрупненные группы заболеваний по МКБ-10;
2. Наиболее высокую точность классификации укрупненных групп заболеваний (на 1,7%) продемонстрировал метод Logistic Regression – Balanced Accuracy = 85,20 %. При этом данная модель имеет наименьшее стандартное отклонение при перекрестной проверке (±1.07%), что свидетельствует об устойчивости результата прогнозирования.
3. Применение предобученных моделей EnRuDR-BERT и RuBioBERT на текстовом корпусе отзывов потребителей на русском языке о фармацевтических продукта и на корпусе свободно доступных текстов в области биомедицины показало высокую точность классификации (Balanced Accuracy = 81,79%), однако не самое эффективное среди рассматриваемых методов. Для повышения качества решения задачи описания жалоб пациентов и постановке диагноза необходимо расширять исходный набор данных и дообучать модель на специализированных медицинских данных.

Глава 4. Разработка алгоритма автоматической генерации индивидуальных листов назначений и рекомендаций к лечению

В четвертой главе рассмотрен подход к автоматической генерации индивидуальных листов назначений и рекомендаций к лечению. Представлена формальная постановка задачи языкового моделирования для автоматической генерации текста, алгоритмы токенизации и языковые модели на базе архитектуры трансформер. Приведены результаты обучения языковых моделей генерации медицинского текста для заполнения индивидуальных листов назначений и рекомендаций к лечению.

4.1. Формализация задачи языкового моделирования для задачи автоматической генерации текста

Для осуществления поддержки принятия решений в процессе выставления диагноза заболеваний и генерации индивидуальных листов назначений и рекомендаций при работе со слабоструктурированной текстовыми данными ЭМК необходимо рассматривать подходы к построению языковых моделей.

Языковое моделирование – это процесс создания моделей, которые позволяют понимать и генерировать естественный язык. Разработанные языковые модели используются для предсказания вероятности последовательности слов или символов в тексте. Таким образом, для генерации индивидуальных листов назначений и рекомендаций к лечению необходимо построить языковую модель, способную генерировать медицинские тексты, соответствующие специфике данной области.

Постановка задачи языкового моделирования в данном контексте включает в себя определение целей генерации текста (например, составление плана лечения, назначений медикаментов и рекомендаций по процедурам), определение набора входных данных (медицинские истории, результаты обследований, симптомы и диагнозы пациента) и выбор подходящей архитектуры модели (например, рекуррентные нейронные сети или трансформеры).

Для формализации задачи языкового моделирования автоматической генерации индивидуальных листов назначений и рекомендаций к лечению введем следующие обозначения:

$d \in D' = \{D_{comp} \cup D_{obj}\}$ - множество текстовых документов ***жалобы пациентов на приеме и данные объективного осмотра пациента*** ($|D'| = 32\,231$);

$h \in H = D_{rec}$ - множество текстовых документов: ***назначения и рекомендации пациентам*** ($|H| = |D'|$);

W – словарь коллекции текстовых документов D' и H, причем

- $\forall d \in D'$: токен $d_j \in W,\ j \in \{1, \dots, |d|\}$;
- $\forall\ h \in H$: токен $h_j \in W,\ j \in \{1, \dots, |h|\}$;

$w_1^n = (w_1, \dots, w_n)$ – заданная последовательность слов/токенов текста $d' \epsilon D'$, $w_k \epsilon W$, $k \in \{1, \dots, |d'|$;

$\widetilde{w}_{n+1}^p = (\widetilde{w}_{n+1}, \dots, \widetilde{w}_p)$ – заданная последовательность слов/токенов текста h$' \epsilon H$, $\widetilde{w}_l \epsilon W$, $l \in \{1, \dots, |h'|$.

Тогда **постановку задачи языкового моделирования** можно сформулировать следующим образом:

Необходимо построить ***языковую модель*** $\rho: D' \to$ H – алгоритм, генерирующий ***последовательность слов*** $\widetilde{w}_{n+1}^p$ текста $h' \epsilon H$ для заданной последовательности слов w_1^n текста $d' \epsilon D'$ на основе оценки условной вероятности:

$$p\left(\widetilde{w}_{n+1}^p \middle| w_1^n\right) \text{ для } \forall d' \epsilon D' \text{ и } \forall h' \epsilon H, \tag{4.1}$$

с некоторой точностью $\widetilde{\text{eps}}$.

На основе марковского правила можно утверждать, что $p(w_n|w_1^{n-1}) \approx p(w_n|w_{n-k}^{n-1})$, при $k \ll n$. Тогда вероятность появления произвольной последовательности слов в тексте вычисляется следующим образом $p(w_1^n) = \prod_{i=1}^n p(w_i|w_{i-k}^{i-1})$.

Обучающая выборка в рамках диссертационного исследования исследования - неразмеченный корпус D жалоб пациентов и данных объективного осмотра на приеме у врача, а также корпус текстов H заключений (рекомендаций и назначений) врачей. Пример входных и выходных данных языковой модели генерации индивидуальных листов назначений и рекомендаций представлен на рисунке 4.1.

Набор данных для обучения языковых моделей генерации текста сформирован на основе алгоритмов, представленных в главе 2, позволяющих взаимодействовать с региональной МИС для выгрузки данных протоколов посещений и выделять информацию из разношаблонных XML-документов посредством рекурсивного обхода. Важно отметить, что рассмотренные протоколы приема пациентов включают не только ССЗ, но и острые респираторные инфекции, новые диагнозы неясной этиологии (эпидемия COVID-19). Назначение лечения пациентам с сопутствующими ССЗ должно учитывать совместимость препаратов и другие особенности процесса восстановления. Поэтому, все доступные протоколы лечения из истории болезней пациентов включены в исследование для построения модели генерации индивидуальных листов назначений и рекомендаций к лечению.

Современные языковые модели, основанные на нейронных сетях, позволяют достичь высокой точности и качества работы при решении задач анализа и генерации текста. Они способны адаптироваться к различным стилям и типам текстов (учитывать специфику медицинского текста), обучаться на больших объемах данных и эффективно работать с различными языками.

В виду того, что контекст необходимо предсказывать исключительно слева направо, то в качестве языковых моделей необходимо рассматривать класс

однонаправленных моделей. Кроме того, при использовании языковых моделей в области медицины необходимо обеспечить сохранность и неразглашение персональных данных пациентов.

Представление данных

'<|endoftext|>'

Жалобы пациента и данные объективного осмотра: $d' \epsilon D'$

Объективно: Общее состояние удовлетворительное. Кожные покровы чистые, умеренной влажности. Склеры обычные.
Т тела- 36,6 АД- 120/75 PS- 77 ЧД-18 Сатурация - 98-99%.
Грудная клетка конической формы. Границы легких в пределах нормы. В легких дыхание везикулярное, хрипов нет. Внешне грудная клетка не деформирована. Границы относительной сердечной тупости: Верхняя-3 ребро, левая на 1,5 см кнаружи от левой среднеключичной линии, правая на 0,5 см от правого края грудины. Тоны сердца ритмичные, приглушены. Живот обычной формы, мягкий, безболезненный.
Печень у правого края реберной дуги, безболезненная. Селезенка не пальпируется. С-м Пастернацкого отрицательный с обеих сторон. Стул регулярный, оформлен. Мочеиспускание свободное, безболезненное. Отеков нет. В позе Ромберга устойчива.

Жалобы: На интенсивную головную боль, шум в голове, головокружение, «мелькание мушек перед глазами», тошнота, при повышении АД, боли в области сердца давяще-сжимающего характера, возникающие после физической нагрузки -ходьба около 250 м, купируемые приемом нитроглицерина, общая слабость, одышку при умеренной физической нагрузке, перебои в работе сердца, периодическое учащенное сердцебиение, снижение памяти, внимания.

'<|body|>'

Рекомендации: $h' \epsilon H$

Режим труда и отдыха. Диета с ограничением животных жиров, жидкости, солёного. «Д» наблюдение участкового терапевта , кардиолога, эндокринолога . Контроль АД,ЧСС ежедневно. Продолжить прием препаратов: лозартан 100 мг 1 р в день, верапамил 40 мг 3 р в день, клопидогрель 75 мг 1 т 1 р в день (рекомендован в дальнейшем смена препарата на пероральные антикоагулянты), индапамид 1.5 мг 1 р в /день, спиронолактон 25 мг 1 т утр, моксонидин 0.4 мг н/н, Аторвастатин 20 мг 1 т веч. под контролем ХС, трансаминаз. Левотироксин натрия по назначению эндокринолога с контролем гормонов щитовидной железы. УЗИ печени в динамике. Ингаляционная терапия в соответствии с назначениями пульмонолога. Нитраты ситуационно.

'<|EOS|>'

Рисунок 4.1 – Пример входных и выходных данных языковой модели

Таким образом, для успешной реализации языковой модели автоматической генерации индивидуальных листов назначений и рекомендаций к лечению необходимо учитывать специфику медицинских данных, особенности процесса лечения и требования по безопасности и конфиденциальности информации.

Отметим, что предлагаемый подход к генерации рекомендаций и назначений на основе массива архивных выписок имеет несколько преимуществ и может быть полезным в медицинской практике, при соблюдении определенных условий:

1. Исторические данные: Архивные выписки и рекомендации должны содержать информацию о предыдущих случаях лечения пациента, его реакции на определенные методы лечения, применяемые препараты и т.д. Анализ этих данных может помочь принимать более обоснованные решения при назначении лечения.

2. Персонализация лечения: Использование архивных данных позволяет создавать персонализированные рекомендации и назначения на основе индивидуальных характеристик пациента, его истории болезни и реакции на предыдущее лечение. Это может улучшить эффективность лечения и снизить риск возникновения нежелательных эффектов.

3. Оптимизация процессов: Генерация рекомендаций позволяет оптимизировать процессы принятия решений в медицинской практике, сократить время, затрачиваемое на подготовку плана лечения, и повысить качество медицинского обслуживания.

Для согласования с клиническими стандартами при использовании подхода автоматической генерации индивидуальных листов назначений и рекомендаций к лечению на основе архивных данных были учтены следующие аспекты:

1. Актуальность данных: Выбранные архивные данные актуальны и соответствовали текущему состоянию пациента.

2. Соблюдение стандартов лечения: Генерация рекомендаций основана на клинических стандартах и рекомендациях, утвержденных медицинским сообществом (все заключения, используемые для обучения, утверждены лечащими врачами). Важно отметить, что врач должен учитывать эти стандарты также при принятии окончательного решения о назначении лечения новому пациенту.

3. Индивидуальный подход при корректировке заключения: несмотря на использование модели генерации рекомендаций, врач должен всегда учитывать индивидуальные особенности каждого пациента и его уникальные потребности при разработке плана лечения.

Таким образом, генерация рекомендаций на основе архивных выписок и рекомендаций может быть полезным инструментом в медицинской практике, при соблюдении актуальности данных для обучения, согласованности решения с клиническими стандартами и учете индивидуальных особенностей каждого пациента. Разработка и обучение соответствующих языковых моделей требует большого объема данных, алгоритмов обучения и валидации, а также постоянный мониторинг и оптимизацию модели для достижения высоких показателей качества текстовой генерации в медицинской области. В рамках диссертационного исследования рассмотрим современные архитектуры моделей генерации текста, а также алгоритмы эффективной токенизации текстовых данных.

4.2. Алгоритмы токенизации листа назначений и рекомендаций текста

Для применения языковых моделей на практике при решении задач обработки естественного языка необходимо представлять текст в векторном виде наиболее эффективно, чтобы учитывать возможные взаимосвязи внутри (семантический и синтаксический смысл текста). Реализацию данного этапа можно представить в 2 шагах: выделение токенов и преобразование их в эмбеддинги.

Токен (англ. token) – это минимальная единица текста, которую можно обработать в рамках задачи обработки естественного языка. Токен может быть словом, числом, знаком препинания и т.д. Токенизация - процесс разбиения текста на токены.

Эмбеддинг (англ. embedding) – это векторное представление слова или фразы в многомерном пространстве. Эмбеддинг позволяет представить слова в виде чисел, которые могут использоваться компьютерными алгоритмами для решения задач обработки естественного языка.

Таким образом, токен – это минимальная единица текста, а эмбеддинг – векторное представление этой единицы (слова или фразы) в многомерном пространстве.

Один из наиболее распространённых подходов к токенизации – ***токенизация BPE (Byte Pair Encoding)*** – метод разбиения текста на токены путем последовательного объединения наиболее часто встречающихся пар байтов.

Процесс токенизации BPE начинается с разбиения каждого слова на отдельные символы. Затем происходит итеративное объединение наиболее часто встречающихся пар символов, пока не будет достигнуто заданное количество токенов или пока не будут объединены все возможные пары символов. Данный подход наиболее эффективен при работе с большими словарями.

Алгоритм BPE (обучение):

1. *Вход*: Исходный словарь V– множество уникальных символов корпуса D, исходный набор правил $P = \emptyset$ – пустое множество, целевой размер словаря k.

2. Цикл (пока $|V| < k$):

а) вычисляем t_a, t_b – наиболее часто совстречаемая в корпусе D пара двух элементов словаря V;

б) формируем новый токен $t_{new} = t_a + t_b$

в) добавляем токен в словарь $V = V \cup \{t_{new}\}$ и запоминаем правило $P = P \cup \{t_a\ t_b \rightarrow t_{new}\}$

Алгоритм BPE (применение): последовательно применяем каждое из полученных правил P к коллекции документов D.

Для каждого токена на следующем этапе необходимо получить векторное представление, ввиду чего формируется эмбеддинг-матрица для языковых моделей.

Существует несколько способов вычисления эмбеддингов для токенов. Один из наиболее распространенных методов – это обучение нейронной сети, которая на вход получает one-hot представление токена и выдает его эмбеддинг. Другой метод – использование готовых предобученных эмбеддингов, таких как Word2Vec, GloVe и FastText.

После вычисления эмбеддингов для всех токенов в словаре, они сохраняются в эмбеддинг-матрицу W_e, которая используется в дальнейшем для вычисления эмбеддингов любых входных текстовых данных, которые поступают на вход модели.

Таким образом, современные алгоритмы векторизации и токенизации обладают высокой точностью и производительностью, что позволяет обрабатывать большие объемы текстовых данных быстро и эффективно. Алгоритмы токенизации разбивают текст на отдельные токены или единицы смысла, учитывая специфические особенности языка и контекста. Данный подход позволит более точно моделировать текстовую информацию и создавать качественные языковые модели для генерации индивидуальных листов назначений и рекомендаций к лечению в медицинской практике.

4.3. Языковые модели на базе архитектуры трансформер для автоматической генерации медицинского текста

Для эффективного анализа и коррекции медицинских документов, включая генерацию листов назначений и рекомендаций к лечению, врачам и медицинскому персоналу необходимо иметь доступ к удобным и быстрым инструментам автоматизированной обработки текстов. Рассмотрим основные теоретические аспекты разработки языковых моделей на базе архитектуры трансформер для автоматической генерации медицинского текста.

Архитектура трансформер является одной из наиболее эффективных и инновационных моделей в области обработки естественного языка и машинного обучения, предоставляющей возможность эффективно моделировать долгосрочные зависимости в тексте. Рссмотрим принципы работы ИНС на базе архитектуры трансформер, выделим основные этапы разработки языковых моделей, а также применение данных моделей для генерации медицинского текста.

Использование искусственных нейронных сетей для решения языкового моделирования можно рассматривать с точки зрения двух подходов:

1) Реализация моделирования $p(w_{n+1} = w|w_1^n)$ при помощи нейросетей для обработки последовательности (RNN, CNN, Transformers).

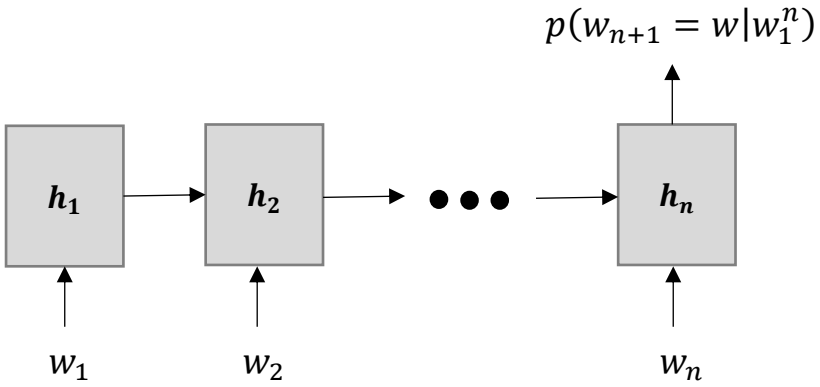

Рисунок 4.2 – Реализация языковой модели прогнозирования одного выхода

2) Реализация «однонаправленных» моделей и получение предсказания на каждом выходе, причем i-ый выход соответствует $p(w_{i+1} = w|w_1^i)$.

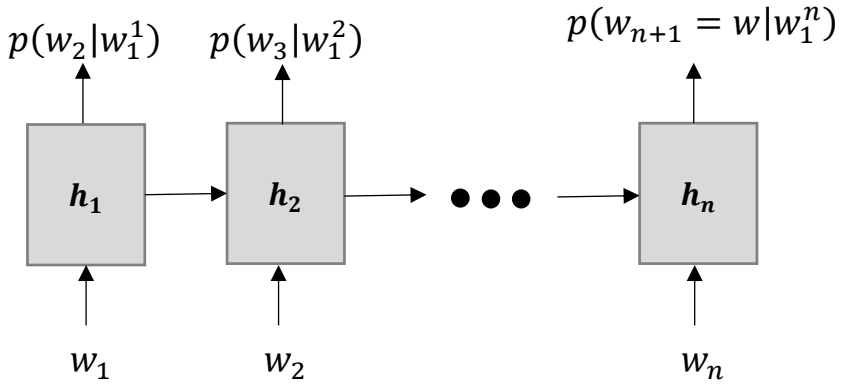

Рисунок 4.3 – Реализация языковой модели прогнозирования на всех выходах

Отметим, что слои для обработки последовательности в каждой позиции должны просматривать только предыдущие токены, что соответствует архитектуре RNN, декодеровщика трансформера и CNN при свертке по левому контексту ячейки (рисунок 4.4.).

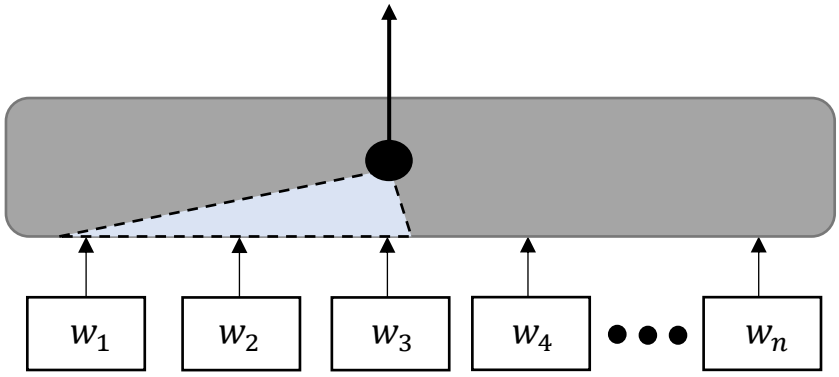

Рисунок 4.4 – Механизм обработки контекста токена для языковой модели выходах

Алгоритм автоматической генерации индивидуальных листов назначений и рекомендаций к лечению на основе работы нейронной сети в задачах языкового моделирования имеет следующий вид:

1. На вход подается последовательность токенов, представляющих слабоструктурированный текст жалоб пациентов и данных объективного осмотра:

$$w_1^n = \{w_1, \dots, w_n\}, \qquad w_i \in W; \tag{4.2}$$

2. Каждый токен преобразуется в эмбеддинг (для входного текста анамнеза заболевания формируется соответствующее векторное представление):

$$v_1^n = Embedding(w_1^n) = \{v_1, \dots, v_n\}; \tag{4.3}$$

3. Эмбеддинги подаются в слои для обработки последовательности:

$$h_1^n = model(v_1^n) = \{h_1, \dots, h_n\}; \tag{4.4}$$

4. К выходам на заданных позициях применяется линейный слой:

$$o_i = Uh_i + b; \tag{4.5}$$

5. Рассчитывается значение функционала для обучения (оценка ошибки генерации следующего токена для заданной входной последовательности, представляющего текст назначений и рекомендаций к лечению):

$$-\sum_{i=1}^{n} \log p\left(w = w_{i+1} \middle| w_1^i\right) = -\sum_{i=1}^{n} \log \operatorname*{softmax}_{w \in W} o_{tw} | w = w_{i+1}. \tag{4.6}$$

6. Обновление параметров языковой модели в соответствии с алгоритмом обратного распространения ошибки и возвращение к шагам 3-5, до тех пор, пока не выполнится заданное количество эпох обучения или установленная погрешность вычислений.

7. Генерация последовательности токенов текста назначений и рекомендаций к лечению на основе обученной на шагах 5-6 языковой модели.

Среди современных архитектур искусственных нейронных сетей для работы с текстовыми данными и генерации текста особенно выделяются рекуррентные нейронные сети (RNN), сверточные нейронные сети (CNN) и трансформеры. Рекуррентные нейронные сети эффективно справляются с последовательными данных, но могут столкнуться с проблемой затухающего градиента при обработке длинных последовательностей. Сверточные нейронные

сети наибелее успешно работают с локальными зависимостями в тексте, но могут иметь ограничения в обработке долгосрочных зависимостей.

Архитектура трансформер, представленная моделью BERT, представляет собой современный метод, показавший отличные результаты в задачах обработки естественного языка. Трансформеры работают параллельно, что позволяет моделировать долгосрочные зависимости в тексте и генерировать качественный текст с высокой точностью.

Модель BertGeneration – это модель BERT, которую можно использовать для задач последовательного преобразования с использованием EncoderDecoderModel [10]. Данная модель совместима с общедоступными предварительно обученными контрольными точками BERT, GPT-2 и RoBERTa. Архитектура блока энкодера для модели BERT представлена на рисунок 4.5.

Отметим, что ячейки энкодера содержат механизмы внимания (multi-headed attention) и нейронную сеть с прямой связью, которые также являются параметрами и определяют различные типы моделей BERT.

Механизмы внимания – это методы, используемые в глубоком обучении для обработки входных данных, чтобы выделить наиболее значимые части информации и сосредоточиться на них. Данные механизмы позволяют моделировать не только локальные зависимости между входными данными, но и глобальные зависимости, что улучшает качество предсказаний.

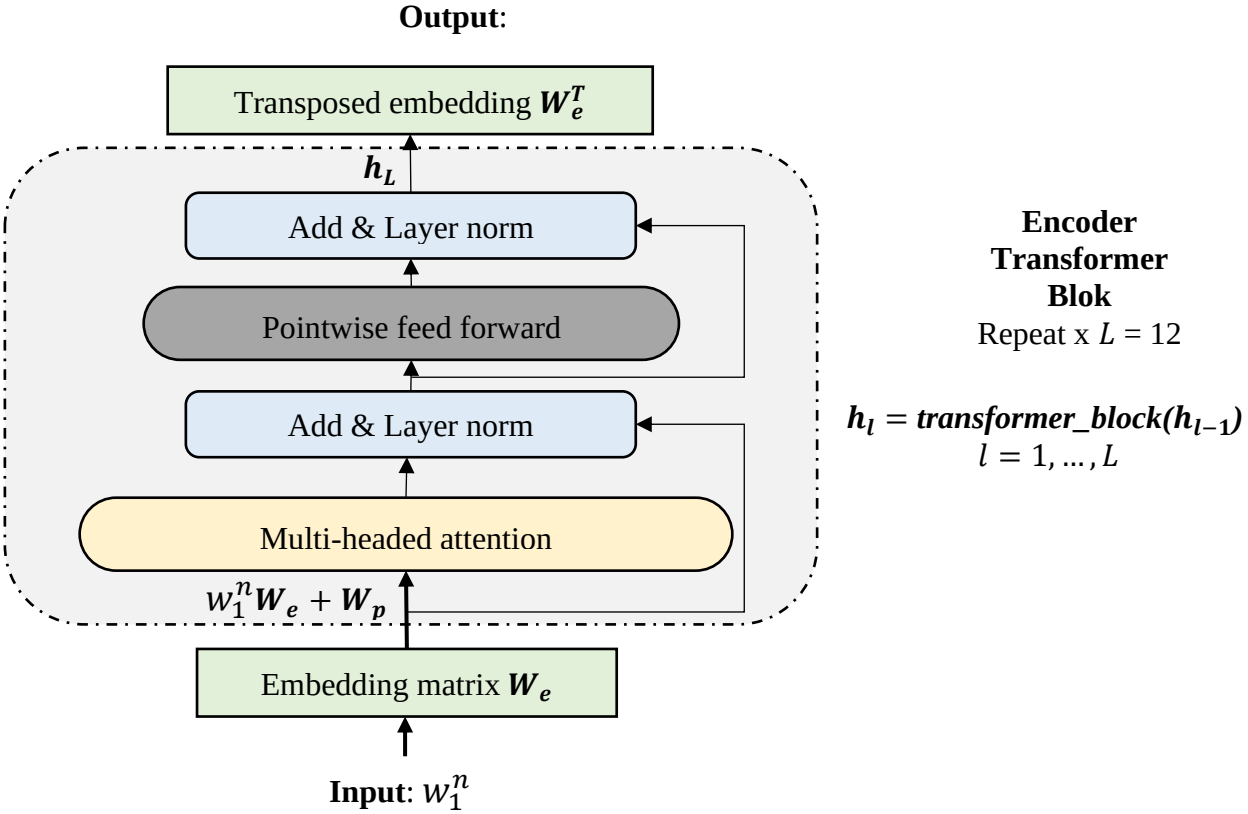

Рисунок 4.5 – Архитектура блока энкодера для модели BERT

Основная идея реализации механизмов внимания основана на присвоении весов различным частям входных данных, чтобы модель могла сфокусироваться на наиболее значимых аспектах. Например, в задаче машинного перевода

механизм внимания может помочь модели сфокусироваться на наиболее важных словах в предложении, которые нужно перевести.

Для формального описания модели внимания введем следующие обозначения:

- q – вектор-запрос, для которого вычисляется контекст;
- $K = (k_1, \dots, k_n)$ – векторы-ключи, для сравнения запроса;
- $V = (v_i, \dots, v_n)$ – векторы-значения, формирующие контекст;
- $a(k_i, q)$ – функция оценки сходства ключа k_i и запроса q;
- c – искомый вектор контекста, релевантный запросу.

Тогда, ***модель внимания*** – 3х-слойная сеть, вычисляющая выпуклую комбинацию значений v_i, релевантных запросу q:

$$c = Attn(q, K, V) = \sum_i v_i SoftMax_i\ a(k_i, q). \tag{4.7}$$

BERT используется двунаправленный подход к обработке текста и анализирует контекст как слева направо, так и справа налево, что позволяет лучше понимать связь между словами в предложении. Кроме того, в BERT используется механизм внимания, который позволяет модели обрабатывать входные данные параллельно и находить зависимости между словами в предложении:

$$c_t = Attn(q_i, K, V) = Attn\big(W_q h'_{t-1}, W_k H, W_v H\big), \tag{4.8}$$

где W_q, W_k, W_v – матрицы весов линейных нейронов (возможно упрощение $W_k, \equiv W_v$), а $H = (h_1, \dots, h_n)$ – входные векторы, h'_{t-1} - выходной вектор.

С технической точки зрения, предсказание выходной последовательности слов требует:

1. Добавление слоя классификации поверх вывода кодировщика.
2. Умножение выходных векторов на матрицу эмбеддингов, преобразование их в размерность словаря.
3. Расчет вероятности каждого слова в словаре с помощью softmax.

Вследствие реализации данного подхода BERT имеет высокую точность в задачах обработки естественного языка, таких как классификация текстов, ответы на вопросы и генерация текста. Однако, модель имеет высокую вычислительную сложность, требующую больших вычислительных ресурсов для обучения и использования модели.

Альтернативная архитектура языковой модели для генерации текста – **модель GPT** (Generative Pre-trained Transformer), которая является авторегрессионной моделью и генерирует новое слово на каждой итерации [61]. В отличие от модели BERT она состоит из блоков декодера, имеет другой механизм внимания и использует генеративное предварительное обучение. Модель GPT использует структуру декодера трансформера (рисунок 4.6).

Внутреннее внимание (Self-Attention) отличается от обычного механизма внимания тем, что анализирует зависимости только внутри входных данных:

$$c_i = Attn(q_i, K, V) = Attn\big(W_q h_i, W_k H, W_v H\big), \qquad h_i \in H, \tag{4.9}$$

где W_q, W_k, W_v – матрицы весов линейных нейронов (возможно упрощение $W_k, \equiv W_v$), а $H = (h_1, \dots, h_n)$ – входные векторы.

В настоящий момент уже представлена модель GPT-4, у которой увеличилось количество используемых параметров, что позволило обрабатывать до 32 000 токенов.

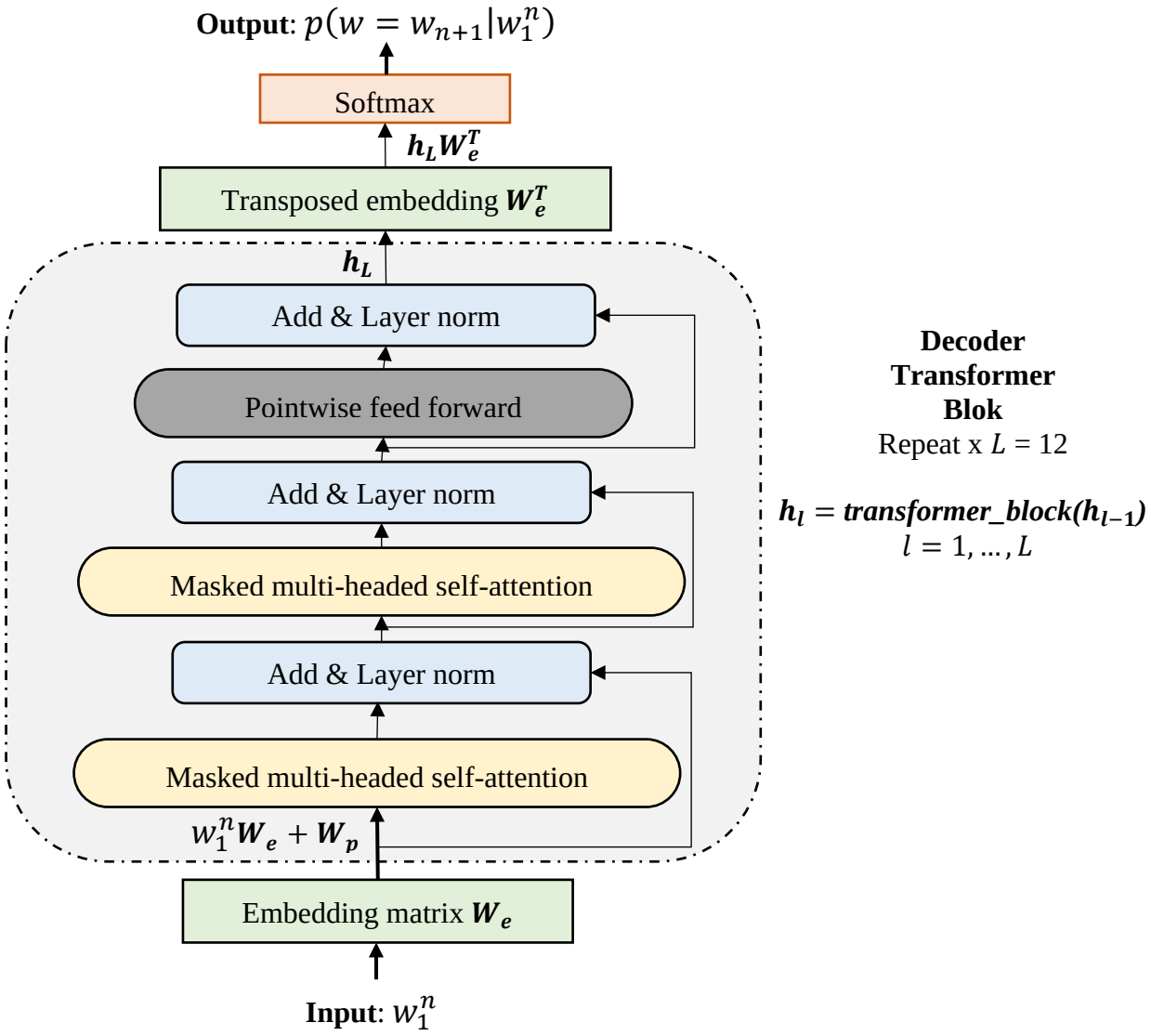

Рисунок 4.6 – Архитектура блока декодера для модели GPT

Схематичное представление взаимодействия слоев модели GPT с последовательностью блоков декодера продемонстрировано на рисунок 4.7.

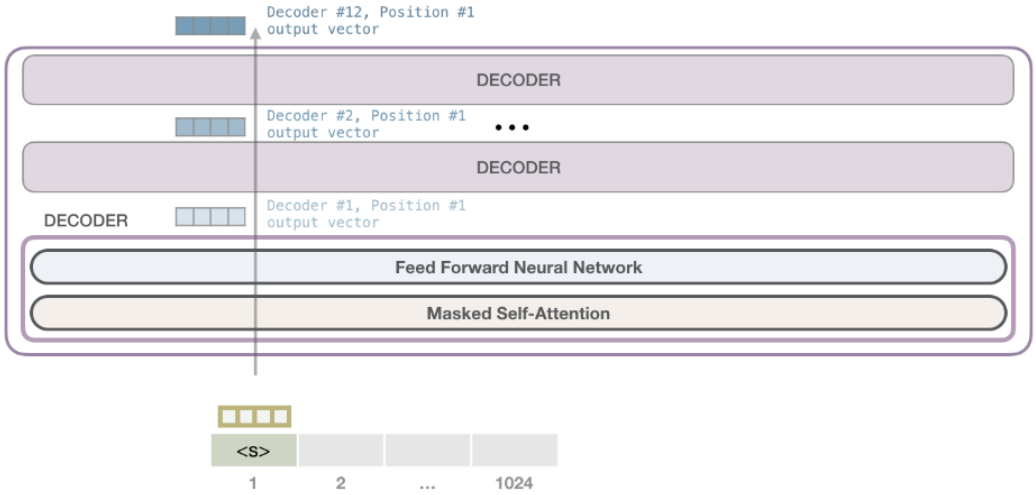

Рисунок 4.7 – Схематичное представление взаимодействия слоев модели GPT

В виду того, что модель GPT-2 и GPT-3 продемонстрировали высокую эффективность при решении биомедицинских задач на других языках [13], воспользуется данной архитектурой для анализа слабоструктурированных русскоязычных медицинских текстов в рамках решения задачи генерации клинических рекомендаций для пациента.

Отметим, что оценка качества рассмотренных языковых моделей играет важную роль в понимании их эффективности и применимости для конкретных задач в области обработки естественного языка и машинного обучения. Рассмотрим различные метрики оценки качества языковых моделей, которые позволят нам объективно сравнивать и анализировать результаты работы моделей.

Методы оценки качества языковых моделей.

Для оценки эффективности языковых моделей генерации текста возможно использование недифференцируемых критериев, которые рассчитываются по выборке пар предложений/текстов «генерация S, эталон S_0».

1) BiLingual Evaluation Understudy (BLEU):

$$BLEU = \min\left(1, \frac{\sum len(S)}{\sum len(S_0)}\right) \underset{(S_0,S)}{\text{mean}}\left(\sum_{n=1}^{4} \frac{\#\ n-\text{грамм}\ \epsilon\ \{S \cap\ S_0\}}{\#\ n-\text{грамм}\ \epsilon\ S}\right); \qquad (4.10)$$

2) Metric for Evaluation of Translation with Explicit Ordering (METEOR):

$$METEOR = F_{mean}(1-p), F_{mean} = \frac{10P \cdot R}{R+P}, p = 0.5\left(\frac{c}{u_m}\right), \qquad (4.11)$$

где $P = \frac{\#\ n-\text{грамм}\ \epsilon\ \{S \cap S_0\}}{\#\ n-\text{грамм}\ \epsilon\ S}$, $R = \frac{\#\ n-\text{грамм}\ \epsilon\ \{S \cap S_0\}}{\#\ n-\text{грамм}\ \epsilon\ S_0}$, c – число групп n-грамм, u_m – количество n-грамм, которые объединили в группы.

Разнообразие существующих метрик позволяет учитывать различные аспекты работы языковых моделей, такие как точность генерации текста, разнообразие и качество содержания, а также семантическую близость и соответствие заданному контексту.

Важно отметить, что не существует универсальной метрики, которая бы однозначно определяла качество работы языковой модели. Как правило, комбинирование или перебор различных методов позволяет получить более полное представление о ее производительности. Кроме того, необходимо учитывать особенности конкретной задачи и целевой аудитории при выборе методов оценки, чтобы убедиться в пригодности модели для решения поставленных задач.

4.4. Исследование эффективности алгоритма автоматической генерации индивидуальных листов назначений и рекомендаций к лечению

Для решения задачи генерации индивидуальных листов назначений и рекомендаций к лечению в рамках диссертационнного исследования выбрана модель GPT (Generative Pre-trained Transformer) по ряду причин.

Во-первых, модель GPT, благодаря своей архитектуре, способна генерировать текст с нуля, что является важным при создании индивидуальных рекомендаций, требующих более сложной логической структуры или контекста.

Во-вторых, GPT часто используется для задач генерации текста и является более обучаемой для специфических задач, что имеет большое значение при работе с медицинскими данными и создании индивидуальных листов назначений, требующих точности и контекстуальной уникальности.

Также отметим, что выбор между моделями GPT и BERT зависит от конкретных требований проекта, доступных ресурсов для обучения и имеющихся данных для применения моделей. Каждая из них имеет свои преимущества и ограничения, и их выбор зависит от конкретной задачи.

Рассмотрим предобученную на общих немедицинских данных нейросетевую модель GPT 3 Large в конфигурации «sberbank-ai/rugpt3large_based_on_gpt2» и проведем обучение корпуса языковых моделей в течении 100 эпох на узкоспециализированном корпусе из клинических текстов жалоб пациентов для каждой укрупненной группы заболеваний отдельно.

Изменение функции потерь в процессе обучения языковой модели для группы «**I1** Болезни, характеризующиеся повышенным кровяным давлением» представлено на рисунке 4.8, а использование соответствующих ресурсов GPU системы на рисунке 4.9.

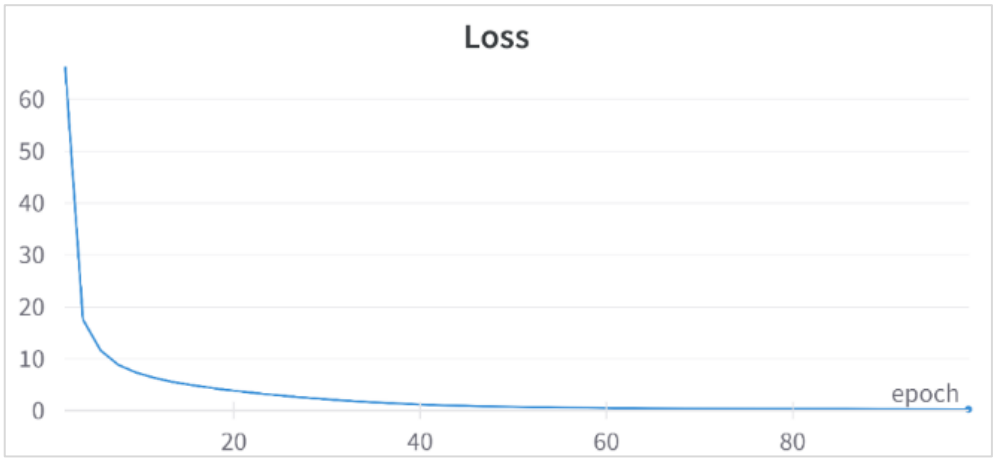

Рисунок 4.8 – Зависимость функции потерь loss от эпохи обучения для группы заболеваний I1

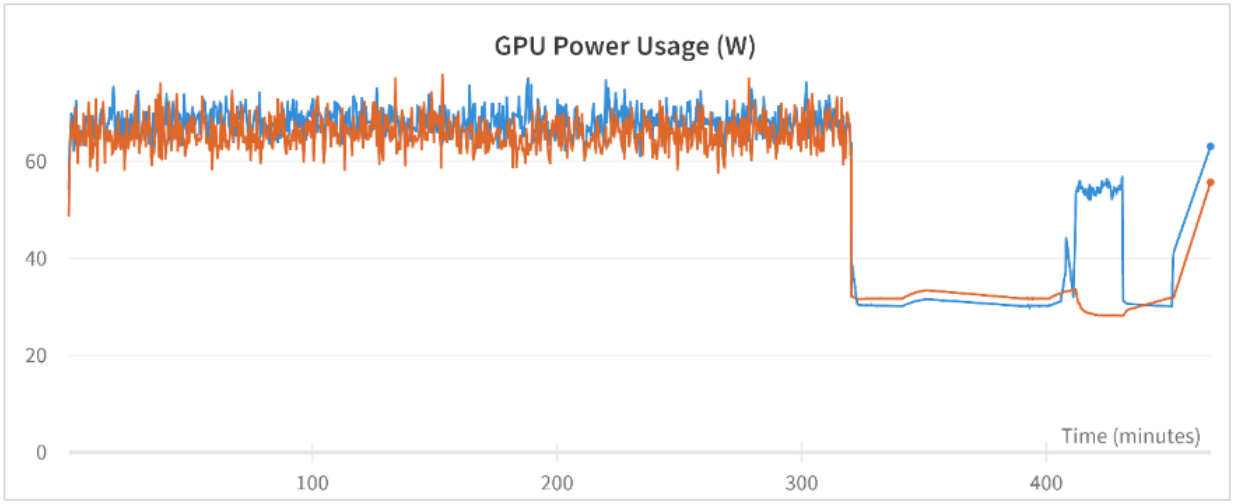

Рисунок 4.9 – Зависимость использования ресурсов GPU от времени обучения для группы заболеваний I1

Значение функции потерь (ошибка loss) на валидационном наборе данных после обучения модели в течении 100 эпох составила 0,178. Однако однозначно оценить эффективность результатов генерации по значению близости функции потерь к нулю невозможно, так как это зависит от контекста задачи и требований к модели. В связи с этим необходимо провести анализ результатов генерации модели и сравнить их с реальными назначениями врачей, чтобы сделать вывод о качестве работы алгоритма. Пример результатов генерации назначений и рекомендаций к лечению пациенту на основе жалоб и данных объективного осмотра представлен в таблице 4.1.

Для оценки качества сгенерированного текста воспользуемся метрикой BLEU (Bi - Lingual Evaluation Understudy), которое измеряет сходство текста, переведенного с помощью генеративной модели, с набором высококачественных справочных переводов. Данная метрика может использовать n-граммы различной длинны, вычислим значение для построенной модели генерации рекомендаций на тестовом наборе из 2000 протоколов жалоб и назначений (рисунок 4.10).

Таблица 4.1 – Пример генерации рекомендаций модели GPT 3 Large для группы заболеваний I1

Жалобы пациента	**Рекомендации врача**	**Результат генерации GTP-3 Large**
головокружение, шум в ушах, слабость, боль в голеностопных суставах	оак, бак, экг, консультация невролога, консультация ревматолога, кальцемин адванс 1 таб 2 раза день, терафлекс 1 таб 1 д ноопепт 1 т 2 раза день	оак бак экг консультация невролога
жалобы на нестабильность ад, дискомфорт в груди, одышка	контроль ад, пульса, стол 10, бисопролол 2 5 мг, лозартан 25 50 мг, верошпирон 25 мг, триметазидин 35 мг 2р	контроль срб бисопрол 2 5 мг утром лозартан 25 1р верошпирон 50мг
отечность ног, повышение ад, сухость во рту	стол р амбул, консультация эндокринолога, индапамид 1 5 мг, бисопролол 5 мг утро, амлодипин 10 мг вечер	стол р амбул консультация эндокринолога бисоприлом
головокружение, головная боль, неустойчивость приходьбе, слабость	метуцин холинальфастерат диклак калмирекс	магнезия холинальфастерат нош па

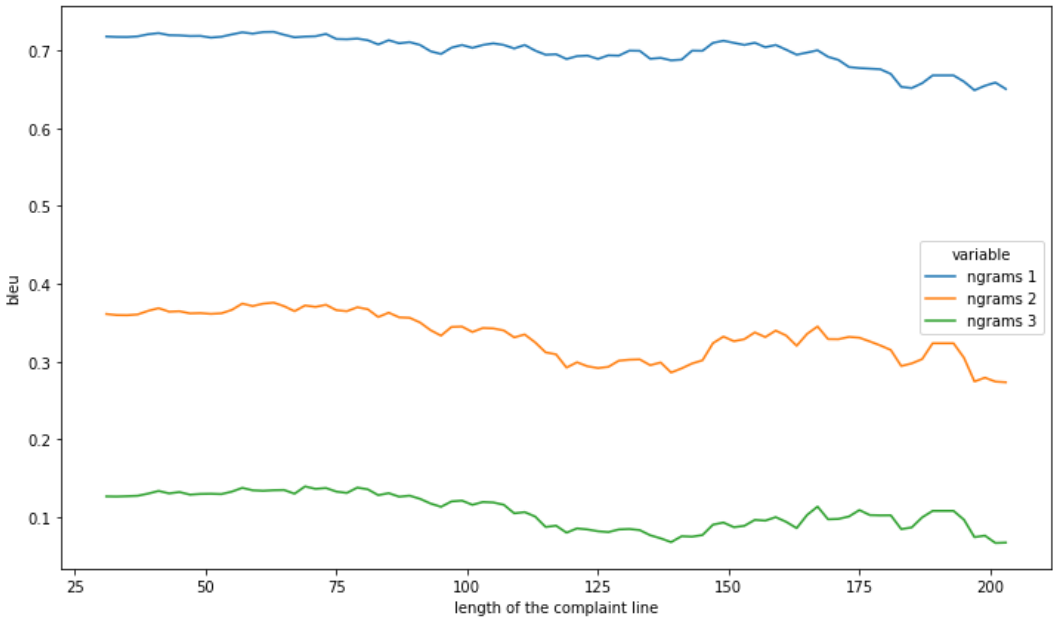

Рисунок 4.10 – Значение метрики BLEU при различных минимальных длинах входного текста

Заранее отметим, что набор данных представляет собой реальные данные медицинской информационной системы, поэтому сравнение с ответом врача – единственно доступный эталон. Однако, данные значения позволят будущим исследованиям ориентироваться в качестве языковых моделей. Среднее

значение BLEU2 составляет 0.33329, а среднее значение BLEU3 составляет 0.10725.

Рассчитанные показатели функции потерь на валидационном наборе (Loss), а также средние значения метрики BLEU для языковых моделей генерации рекомендаций для рассмотренных укрупненных групп заболеваний представлены в таблице 4.2.

Таблица 4.2 – Оценка качества языковых моделей для всех укрупненных групп заболеваний

Группа заболеваний	**Loss**	**BLEU1**	**BLEU2**	**BLEU3**
I1 Болезни, характеризующиеся повышенным кровяным давлением	0,1784626	0,68931	0,33329	0,10725
I2 Ишемическая болезнь сердца, Легочное сердце и нарушения легочного кровообращения	1,3144681	0,68558	0,34029	0,15476
I4 Другие болезни сердца	1,1687536	0,58589	0,30924	0,07675
I6 Цереброваскулярные болезни	0,1893345	0,67025	0,36373	0,09925
I8 Болезни вен, лимфатических сосудов и лимфатических узлов	0,2173053	0,66281	0,38055	0,10987
J0 Острые респираторные инфекции верхних дыхательных путей	0,3238005	0,67075	0,38373	0,13403
U0 Временные обозначения новых диагнозов неясной этиологии	1,2739228	0,70828	0,39079	0,16743

Таким образом, в рамках диссертационной работы исследованы методы и подходы к анализу слабоструктурированных русскоязычных медицинских текстов для решения задачи генерации клинических рекомендаций для пациента по семи укрупненным группам заболеваний.

В результате исследования обучен корпус языковых моделей GTP-3 Large для генерации индивидуальныых листов назначений и рекомендаций к лечению и получены результаты, демонстрирующие потенциально высокие возможности применения методов NLP для построения СППВР. Оценка сходства сгенерированных рекомендаций с реальными рекомендациями врачей на основе метрики BLEU на основе униграмм, биграмм и триграмм в среднем для всего корпуса языковых моделей составила примерно 0.66755, 0.35737 и 0.12133 соответственно. Так как значения BLEU языковых моделей генерации для униграмм и биграмм довольно большие, это указывает на высокую степень сходства с реальными рекомендациями врачей.

Выводы четвертой главы

1. В рамках данного исследования рассмотрен подход к анализу слабоструктурированных русскоязычных медицинских текстов в рамках решения задачи генерации индивидуальных рекомендаций для пациентов, основанный на модели глубокого обучения GPT 3 Large, предварительно обученной на русскоязычном тексте от sberbank-ai.

2. Разработаны метод и алгоритм генерации индивидуальных листов назначений и рекомендаций к лечению в рамках диагностированных заболеваний на основе предобученных языковых моделей трансформеров. Обучен корпус языковых моделей для рассмотренных укрупненных групп заболеваний для поддержки принятия врачебных решений, который в отличие от существующих шаблонов в МИС позволяет автоматизировать процесс заполнения документов и допускает коррекцию в соответствии с экспертным мнением врача.

3. Значение функции потерь языковых моделей на валидационном наборе данных составило не более 1.31 по всем укрупненным группам заболеваний. Оценка сходства сгенерированных рекомендаций с реальными рекомендациями врачей на основе метрики BLEU на основе униграмм, биграмм и триграмм в среднем для всего корпуса языковых моделей составила примерно 0.66755, 0.35737 и 0.12133 соответственно. Полученные результаты демонстрируют потенциально высокие возможности применения методов NLP для построения систем поддержки принятия врачебных решений.

Глава 5 Разработка автоматизированного программного комплекса интеллектуальной поддержки принятия врачебных решений при диагностике и лечении заболеваний

В пятой главе представлен прототип автоматизированного программного комплекса интеллектуальной поддержки принятия решений в медицинской практике с использованием веб-фреймворка для доступа к построенным алгоритмам и моделям ИИ. Кроме того, проведена оценка эффективности программного комплекса на практике в медицинских организациях с учетом затрат рабочего времени.

5.1 Внутренняя структура компонентов сервиса

Внедрение моделей искусственного интеллекта в медицинские информационные системы может значительно улучшить качество медицинского обслуживания и оптимизировать процессы работы медицинских учреждений.

Однако, необходимо учитывать, что внедрение ИИ в МИС требует высокой степени ответственности и внимательности, так как некорректное использование моделей ИИ может привести к серьезным последствиям для здоровья пациентов. Поэтому перед внедрением интеллектуальных моделей поддержки принятия решений в медицинску практику необходимо проводить тщательное тестирование и обучение персонала работе с новыми технологиями.

С технической точки зрения существует два основных подхода внедрения моделей ИИ в медицинские информационные системы:

- Разработка программного обеспечения, развернутого в закрытом контуре подсистемы МИС на ее сервере.
- Разработка внешнего веб-сервиса, развернутого за пределами контура МИС, который взаимодействует с ней с помощью API.

У каждого из представленных подходов имеются свои достоинства и недостатки, а выбор пути внедрения интеллектуальных моделей в МИС во многом зависит от доступных ресурсов проекта. Преимущества каждого из подходов представлены в таблице 5.1.

Таблица 5.1 Достоинства вариантов развертывания подсистемы моделей ИИ для внедрения в МИС

Закрытое развертывание	Взаимодействие с веб-сервисом
• повышенная безопасность при работе с данными; • за счет внутреннего развертывания отсутствует риск ошибки передачи данных при сбоях сети; • простота развертывания внутри контура.	• удобное обслуживание и обновление моделей ИИ; • простота интеграции в МИС с помощью API и уникального токена для каждого медицинского учреждения.

Ввиду особенностей развертывания моделей ИИ у каждого из описанных выше подходов имеются недостатки, представленные в таблице 5.2.

Таблица 5.2 Недостатки различных вариантов развертывания подсистемы моделей ИИ для внедрения в МИС

Закрытое развертывание	Взаимодействие с веб-сервисом
• сложность обновления моделей, и поддержания подсистемы моделей обучения ИИ; • сложность интеграции в медицинские учреждения, так как необходимы актуальные копии в каждой МИС	• необходимость устойчивости сервера к нагрузкам от нескольким медицинских учреждений; • необходим повышенный контроль безопасности, так как сервер фактически доступен в открытой сети, существует риск взлома.

Анализируя вышеперечисленные особенности реализации каждого подхода, а также беря во внимание тот факт, что развертывание моделей ИИ требует специального виртуального окружения на выполнения скриптов, которое сложно настраивать на стороне МИС, выбран вариант разработки внешнего веб-сервиса.

Определим основные компоненты веб-сервиса для предоставления результатов моделей ИИ:

- Модуль аутентификации. Для доступа медицинских учреждений к результатам моделей по предоставленному заранее логину и паролю.
- Хранилище моделей. Предназначено для управления моделями, обновления и добавления новых.
- Модуль обработки входных запросов. Является входной точкой, получает запрос, обрабатывает, выделяет входные данные, также отвечает за токенизацию.
- Модуль прогнозирования. Загружает модели и передает им на вход обработанные данные.

Так как построенные модели ИИ разработаны на языке программирования Python, для веб-разработки использован фреймворк Django.

Django – это высокоуровневый фреймворк для веб-разработки на языке Python, который имеет следующие достоинства:

1) Быстрое создание приложений: Django предоставляет множество готовых компонентов и функций, которые позволяют быстро создавать веб-приложения.

2) Масштабируемость: Django обладает высокой масштабируемостью, что позволяет создавать приложения любой сложности.

3) Безопасность: Django имеет встроенные механизмы безопасности, такие как защита от CSRF-атак и SQL-инъекций.

4) Административный интерфейс: Django предоставляет готовый административный интерфейс, который позволяет управлять данными в приложении без написания дополнительного кода.

5) ORM: Django имеет объектно-реляционную модель (ORM), которая позволяет работать с базами данных на более высоком уровне абстракции и упрощает работу с данными.

6) Гибкость: Django позволяет использовать различные шаблоны и расширения, что делает его гибким инструментом для создания различных приложений.

7) Сообщество: Django имеет большое сообщество разработчиков, которые создают и поддерживают множество дополнительных расширений и пакетов.

В целом, Django является мощным инструментом для создания веб-приложений, который обладает множеством достоинств и позволяет быстро и эффективно создавать приложения любой сложности.

Структура Django-приложения представляет из себя каталог, в котором находится пакет конфигурации, который содержит настройки проекта, файлы маршрутизации и другие компоненты, необходимые для запуска приложения. Помимо пакета конфигурации в него входят:

- Каталог приложения: каталог, который содержит код и ресурсы, связанные с конкретным приложением внутри проекта.
- Файлы моделей: файлы, которые определяют структуру базы данных и связи между таблицами.
- Файлы представлений: файлы, которые определяют, как данные будут отображаться в браузере.
- Файлы маршрутизации: файлы, которые определяют URL-адреса и связывают их с соответствующими представлениями.
- Файлы шаблонов: файлы, которые содержат HTML-разметку и определяют, как данные будут отображаться в браузере.
- Файлы статических ресурсов: файлы, такие как изображения, CSS-стили и JavaScript-скрипты, которые используются в приложении.
- Файлы форм: файлы, которые определяют формы для ввода данных пользователем.
- Файлы административного интерфейса: файлы, которые определяют пользовательский интерфейс для административной панели Django.
- Файлы миграции: файлы, которые содержат инструкции для обновления базы данных при изменении моделей.

Таким образом, структура приложения Django представляет собой нескольких компонентов, которые работают вместе для создания полноценного веб-приложения.

Для развертывания веб-приложения необходимо подготовить сервер. Во-первых, требуется установить необходимые зависимости, такие как Python, Django, базу данных и т.д. Во-вторых, создать виртуальное окружение для проекта. В-третьих, загрузить код проекта на сервер. В-четвертых, настроить

базу данных и другие параметры проекта в файле settings.py. В-пятых, установить и настроить HTTP-сервер Gunicorn. Наконец, настроить Nginx или Apache для проксирования запросов к серверу Django.

Схема веб-сервиса и внутренняя конфигурация на фреймворке Django представлена на рисунке 5.1.

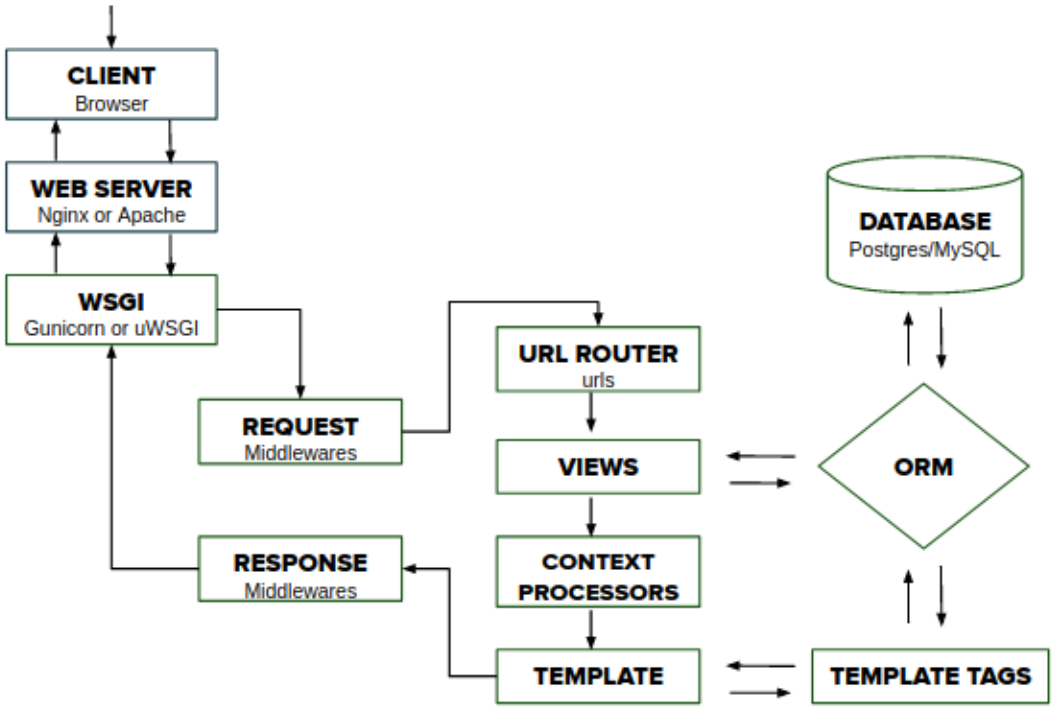

Рисунок 5.1 – Схема веб-сервиса на фреймворке Django.

В качестве сервера для развертывания моделей использован виртуальный частный сервер VPS с операционной системой Ubuntu 22.04. Сервер Ubuntu с моделями ИИ имеет следующие характеристики:

- Процессор Intel Core i7;
- Оперативная память 6 ГБ;
- Жесткий диск (SSD) емкостью 30 ГБ;
- Поддержка GPU (графических процессоров) для ускорения вычислений в моделях ИИ;
- Установленная и настроенная библиотека CUDA Toolkit для работы с GPU;
- Наличие необходимых программных компонентов, таких как Python, TensorFlow, PyTorch, Keras и т.д. в соответствующих версиях и настройках.

Для демонстрации работы прототипа автоматизированного программного комплекса интеллектуальной поддержки принятия решений в медицинской практике развернут тестовый веб-сервис с демо-приложением, представленный на рисунке 5.2.

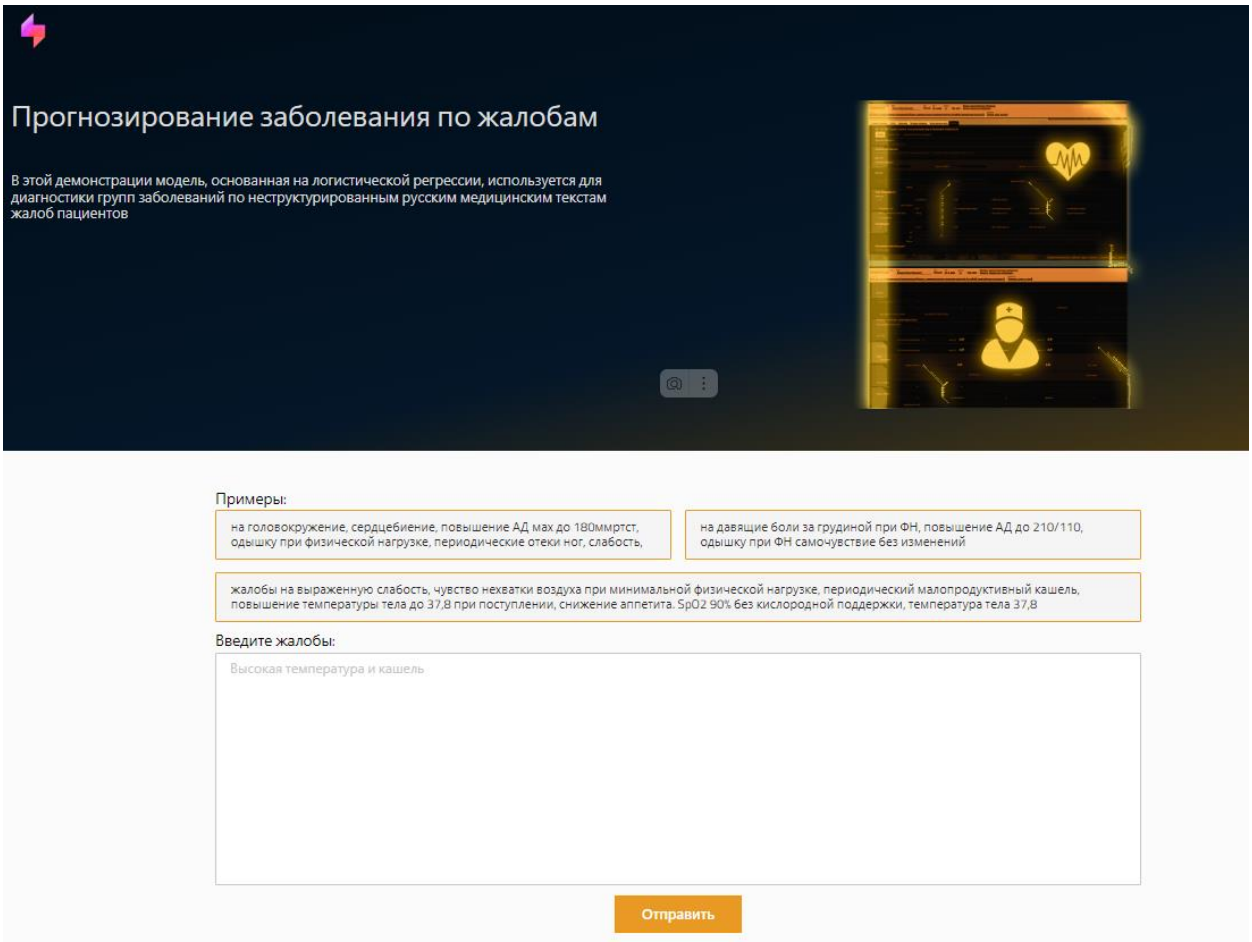

Рисунок 5.2 – Демо-страница с демонстрацией функционала разработанных моделей ИИ.

Для получения результатов моделей прогнозирования укрупненной группы заболеваний по МКБ-10 и сгенерации назначений и рекомендаций к лечению необходимо реализовать следующую последовательность действий:

1. В поле «Жалобы» заполняется соответствующая информация (либо можно выбрать один из представленных примеров).
2. Нажать на кнопку «Отправить» для передачи информации на сервер и получения результата обработки запроса от размещенных моделей ИИ.

После отправки жалоб на сервер с прогнозными моделями от него приходит ответ в формате JSON, который отображается на странице. Результат работы веб-сервиса представлен на рисунке 5.4. Вывод результатов моделей ИИ отображает следующую информацию:

- код укрупненной группы заболеваний по МКБ-10 и вероятность уверенности модели ИИ в прогнозе;
- сгенерированные назначения и рекомендации к лечению с предупреждением, что они являются искусственно полученными и носят исключительно ознакомительный характер для демонстрации работы веб-сервиса;
- график распределения вероятностей классификации по всем кодам МКБ-10, на которых обучена модель прогнозирования диагноза заболевания.

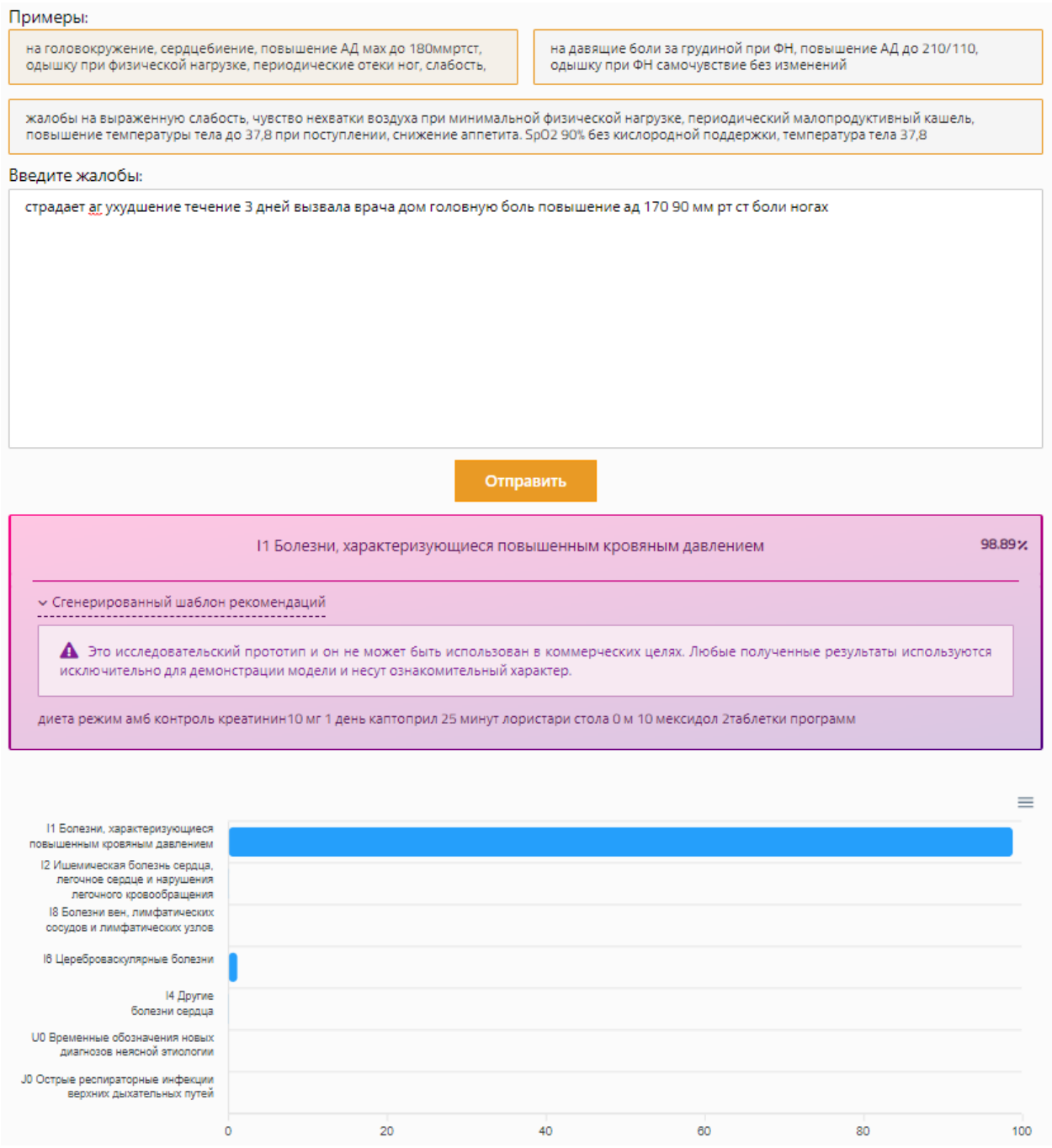

Рисунок 5.3 – Результаты работы демо-приложения

Таким образом, в рамках диссертационной работы построен прототип автоматизированного программного комплекса интеллектуальной поддержки принятия решений в медицинской практике. Использование веб-фреймворка позволяет получать доступ к построенным алгоритмам и моделям искусственного интеллекта, что значительно упрощает работу врачей и повышает эффективность и точность диагностики и лечения заболеваний.

5.2 Модуль взаимодействия с внешними МИС

Обеспечение взаимодействия сервера с моделями искусственного интеллекта и внешними медицинскими информационными системами представляет собой важный этап в разработке систем поддержки принятия решений в медицине. Модуль взаимодействия веб-сервиса с моделями ИИ и внешними МИС основывается на интеграции и синхронизации данных между ними. Для этого модуль может использовать различные протоколы и стандарты обмена информацией, такие как REST API, SOAP, HL7 и другие. Данный модуль демонстрирует важную роль взаимодействия и совместной работы различных систем в медицинской сфере, что позволяет оптимизировать обработку информации, улучшить качество принимаемых решений и повысить эффективность в области здравоохранения.

Для реализации модуля взаимодействия сервера с моделями ИИ и внешними медицинскими информационными системами можно использовать различные технологии и подходы. В рамках диссертационного исследования реализован подход на основе REST API (Representational State Transfer Application Programming Interface) для обмена данными между модулем и внешними системами. REST API – это архитектурный стиль взаимодействия компонентов распределенного приложения в сети, который позволяет обмениваться данными по протоколу HTTP и обеспечивает высокую скорость передачи данных и удобство в использовании.

REST API имеет четыре основных принципа: использование стандартных методов HTTP (GET, POST, PUT, DELETE), идентификация ресурсов по URL, использование многоуровневой архитектуры и поддержка кеширования данных для улучшения производительности.

Для обеспечения безопасности и конфиденциальности медицинских данных можно использовать различные методы шифрования и аутентификации, такие как SSL-шифрование и OAuth аутентификация. Данные методы позволяют защитить данные от несанкционированного доступа и обеспечить их конфиденциальность.

Таким образом, разработанный модуль взаимодействия сервера с моделями ИИ с внешними медицинскими информационными системами является важным элементом в системе поддержки принятия решений в медицине. Он обеспечивает связь между моделями искусственного интеллекта и внешними медицинскими информационными системами, что позволяет эффективно и безопасно анализировать медицинские данные для принятия обоснованныхе решений в области здравоохранения. В результате реализации технологии REST API, разработанный программный комплекс диагностики и лечения заболеваний пациентов с использованием веб-фреймворка может интегрироваться в действующие МИС и на основе слабоструктурированной текстовой информации протоколов автоматизировать процедуру принятия врачебных решений.

5.3 Оценка эффективности программного комплекса интеллектуальной поддержки принятия врачебных решений при диагностике и лечении заболеваний

Для оценки эффективности разработанного программного комплекса интеллектуальной поддержки принятия врачебных решений при диагностике и лечении заболеваний для пациентов с сердечно-сосудистыми заболеваниями проведена апробация результатов исследования в ГАУЗ «Бузулукской больнице скорой медицинской помощи» Оренбургской области.

Важно отметить, что смертность населения Оренбургской области по данным Росстата за 2020 и 2021 года значительно увеличилась (на 23,5% и 15,6% соответсвенно). Во многом данные показатели обусловлены влиянием пандемии COVID-19, однако среди основных причин смертности населения 1 место в течении 2019-2021 гг. занимают «Болезни системы кровообращения (БСК)» (не менее 38,6% от общего количества, рисунок 5.4).

Кроме того, в структуре заболеваемости БСК основной группой пациентов являются больные артериальной гипертонией (I10-I13) и ишемической болезнью сердца (I20- I25), для которых ежегодно впервые установленных диагнозов среди врослого населения 18 лет и более около 67 тыс. человек (таблица 5.3).

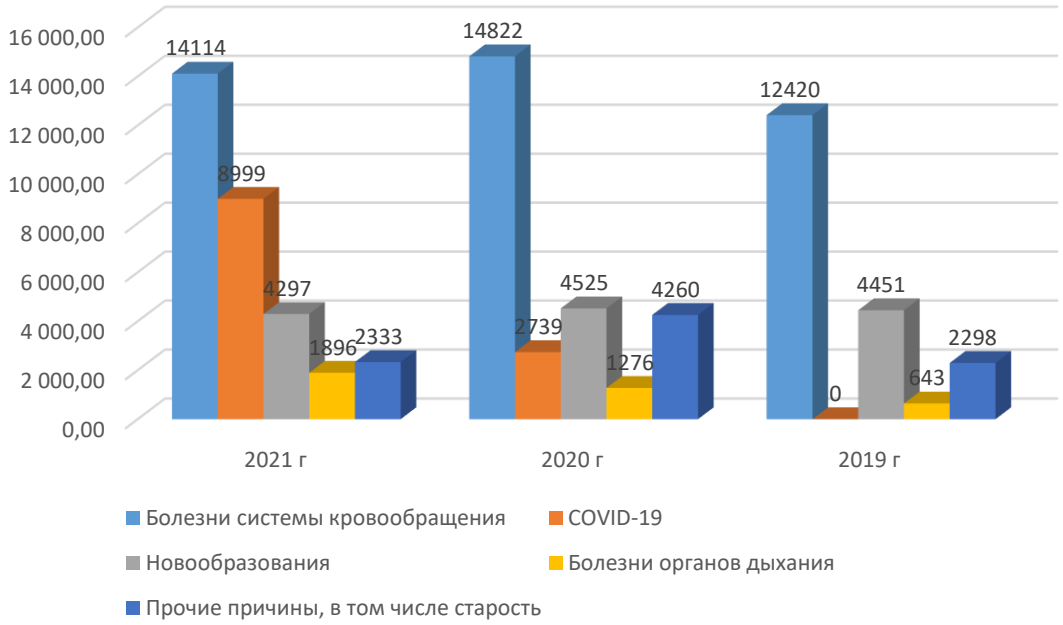

Рисунок 5.4 – Структура основных причин смертности населения в Оренбургской областей

Среднее число посещений по поводу заболеваний в одном случае по кардиологии составляет 3,1. При этом, врач должен оказывать медицинские услуги в соответсвии с расчётными нормами обслуживания для врачей амбулаторно-поликлинических учреждений (подразделений). Структура приема пациентов врачом-терапевтом включает в себя не только его основную деятельность (сбор анамнеза, опрос, оценка общего статуса и т.д.), но и

дополнительные виды деятельности, такие как подготовка инструментов и работа с документацией. Эффективное оказание медицинской услуги во многом зависит от грамотного распределения затрат рабочего времени.

Таблица 5.3 Количество зарегистрированных заболеваний

Заболевание	Код по МКБ-10	Зарегистрировано заболеваний		Состоит под диспансерным наблюдением
		Всего	Из них с впервые установленным диагнозом	
Артериальная гипертония	I10-I13	280184	51605	165959
Ишемическая болезнь сердца	I20- I25	114755	15431	65810

Связи с этим, для автоматизации рутинных и повторяющихся процессов (работы с документацией, заполнения электронных медицинских карт, формирования заключения) требуется внедрение систем поддержки принятия решений, которые позволяют сократить и грамотно перераспределить затраты рабочего времени.

Расчет затрат рабочего времени (Т) на посещение проводился по формуле:

$$\mathrm{T}_{\text{проц}} = \sum t_i \text{ x } n_i, \tag{5.1}$$

где:

t_i – затраты на трудовую операцию;

n_i – число трудовых операций в трудовом процессе.

Расчет средней длительности приёма проводится по формуле:

$$\mathrm{T}_{\text{ср}} = \sum \mathrm{T}_i \text{ x } \Pi_i, \tag{5.2}$$

где:

T_i – длительность того или иного приёма (лечебно-диагностического, профилактического, консультативного и т.д.);

Π_i – удельный вес того или иного приёма в структуре посещений (в долях от 1,0).

Затраты рабочего времени врача-терапевта на каждого пациента могут варьироваться в зависимости от различных факторов, таких как сложность состояния пациента, цель визита, наличие дополнительных обследований и процедур, а также индивидуальная политика медицинского учреждения. Однако, в общих чертах, типичная консультация у врача-терапевта может длиться около 15 минут.

В течение этого времени врач будет задавать вопросы о медицинской истории пациента, симптомах и проблемах, проводить физическое обследование, советовать по диагностике и лечению, а также отвечать на вопросы пациента. Если требуется дополнительное время на проведение дополнительных обследований, консультацию специалиста или предоставление дополнительных услуг, время приема может увеличиться.

В таблице 5.4. приведены данные средней длительности и структуры приёма больных врачем-терапевтом с учетом основных трудовых операций, необходимых к проведению на первичном осмотре.

Таблица 5.4 Средняя длительность и структура приёма врачом-терапевтом

Трудовые операции	**первичный**	
	мин	**%**
Всего	15,0	100,0
ОСНОВНАЯ ДЕЯТЕЛЬНОСТЬ	**10,03**	**65,53**
Сбор анамнеза, опрос	2,54	16,93
Осмотр	3,04	20,27
Процедуры, инструментальные исследования, диагностические пробы	2,43	16,20
Разъяснение диагностических мероприятий, получение согласия на лечение, разъяснение лечения	1,82	12,13
ДОПОЛНИТЕЛЬНЫЕ ВИДЫ ДЕЯТЕЛЬНОСТИ	**5,17**	**34,47**
Вспомогательная деятельность (мытьё рук, одевание перчаток, подготовка инструментов)	0,39	2,60
Работа с документацией (заполнение электронной мещдицинской карты; просмотр результатов анализов и исследований; выписка направления, листка нетрудоспособности и др.)	4,78	31,87

В рамках апробации в ГАУЗ «Бузулукской больнице скорой медицинской помощи» и ГАУЗ «Оренбургской областной клинической больнице имени В.И. Войнова» 4 участковых терапевта в течении недели использовали на приемах для автоматизации заполнения электронных медицинских карт, разработанный программный комплекс интеллектуальной поддержки принятия врачебных решений при диагностике и лечении заболеваний с фиксацией затрат рабочего времени. Общее количество приемов пациентов с ССЗ при использовании ИСППР – 56 чел. Усредненные результаты исследования представлены в таблице 5.5.

Относительно перечисленных типов выполняемой работы с документацией, автоматизация процесса происходит в части «Формирования листа назначений и рекомендаций». Среднее время генерации шаблона моделями ИИ составляет ≈ 7 сек. (или 0,1167 мин.), а время коррекции результатов генерации врачом-терапевтом для учета экспертного мнения ≈ 41 сек. (или 0,68 мин.).

Таблица 5.5 Результаты оценки затрат рабочего времени на работу с документацией

Тип выполняемой работы с документацией	Затраты рабочего времени до внедрения ИСППР		Затраты рабочего времени после внедрения ИСППР		
	мин	%	Время генерации мин	Время коррекции, мин	Общее время работы, мин
Всего	4,780	100	-	-	4,452
Просмотр амбулаторной карты	0,528	11,0	-	-	0,528
Запись анамнеза	1,458	30,5	-	-	1,458
Запись результатов осмотра пациента	1,291	27,0	-	-	1,291
Формирование листа назначений и рекомендаций	1,120	23,5	0,112	0,68	0,792
Другие виды работы с документацией	0,383	8,0	-	-	0,383

Таким образом, при внедрении программного комплекса интеллектуальной поддержки принятия решений в медицинской практике для прогнозирования группы заболеваний и генерации соответствующих индивидуальных листов назначений и рекомендаций к лечению при работе с документацией затраты рабочего времени снизились в среднем на 6,9%, а относительно общего времени на прием затраты снизились 2,18%.

Для расчета t-теста для сравнения процентного изменения затрат времени с нулевой гипотезой о том, что нет статистически значимого изменения, использованы следующие данные: уровень значимости $\alpha = 0,05$; количество степеней свободы $df = 55$. Соответствующее критическое значение $t = \pm 2,009$, а рассчитанное значение $t \approx 5,43$. Так как значение t попадает в критическую область, нулевая гипотеза отвергается. Таким образом, после внедрения программного комплекса имеется статистически значимое изменение в затратах времени на прием пациентов с ССЗ.

В связи с тем, что норма оказания медицинской услуги первичного приема составляет 15 мин на одного пациента, а рабочий день врача-терапевта составляет 7 часов, то максимальное количество пациентов в день – 28 чел., в неделю – 140 чел. При внедрении ИСППР в рабочий процесс возможна экономия общего времени приема 9,156 мин в день (прием в неделю в среднем 143 чел.).

Выводы пятой главы

1. Построен прототип программного комплекса интеллектуальной поддержки принятия решений в процессе диагностики и лечения заболеваний пациентов с использованием веб-фреймворка для доступа к построенным алгоритмам и моделям через REST API и отличающийся от существующих тем, что может интегрироваться в действующие МИС и на основе слабоструктурированной текстовой информации протоколов автоматизировать процедуру принятия врачебных решений.

2. Основные компоненты веб-сервиса для предоставления результатов моделей ИИ содержат модуль аутентификации, хранилище моделей, модуль обработки входных запросов и модуль для прогнозирования. В связи с тем, что построенные модели ИИ написаны на языке программирования Python выбран веб-фреймворк Django. Для обеспечения безопасности и конфиденциальности медицинских данных использованы методы SSL-шифрования и OAuth аутентификации.

3. Для оценки эффективности разработанного программного комплекса интеллектуальной поддержки принятия врачебных решений при диагностике и лечении заболеваний для пациентов с сердечно-сосудистыми заболеваниями проведена апробация результатов исследования в медицинских организациях Оренбургской области, которая показала, что при работе с документацией затраты рабочего времени снизились в среднем на 6,9%, а относительно общего времени на прием затраты снизились 2,18%.

Заключение

1) Осуществлен анализ научных проблем интеллектуальной поддержки принятия решений в медицинской практике. Предложена концептуальная модель анализа клинических данных и поддержки принятия решений для автоматизации процессов заполнения ЭМК, позволяющая структурировать текстовые данные и внедрять интеллектуальные модели для формирования рекомендаций к лечению диагностированных заболеваний.

2) Разработана иерархическая модель структурирования данных амбулаторных карт пациентов и обработки разношаблонных XML-документов МИС на основе рекурсивного подхода для обеспечения семантической интероперабельности.

3) Разработаны метод и алгоритм прогнозирования группы заболеваний с использованием методов NLP, а также моделей машинного обучения, которые используют уникальный узкоспециализированный неразмеченный корпус текстов и укрупненные группы заболеваний по МКБ-10 и имеют сбалансированную точность 85,20% (стандартное отклонение при перекрестной проверке ±1.07%, что свидетельствует об устойчивости результата прогнозирования).

4) Разработаны метод и алгоритм генерации индивидуальных листов назначений и рекомендаций к лечению в рамках диагностированных заболеваний для поддержки принятия врачебных решений на основе предобученных языковых моделей трансформеров, который в отличие от существующих шаблонов в МИС позволяет автоматизировать процесс заполнения документов и допускает коррекцию в соответствии с экспертным мнением врача с метрикой BLEU1 = 0,668 и BLEU2 = 0,357.

5) Построен прототип автоматизированного программного комплекса интеллектуальной поддержки принятия решений в медицинской практике, отличающийся применением модулей искусственного интеллекта для диагностирования заболеваний и формирования рекомендаций к лечению на основе методов обработки естественных языков. Апробация результатов исследования показала, что при работе с документацией затраты рабочего времени снизились в среднем на 6,9%, а относительно общего времени на прием – на 2,18%.

Направления будущих исследований. Для повышения качества решения задач постановки диагноза и формирования рекомендаций к лечению необходимо расширять исходный набор данных, исследовать языковые архитектуры большей размерности и дообучать модели ИИ на специализированных медицинских данных.

Список публикаций по теме исследования

В рецензируемых журналах из списка ВАК и отечественных изданиях, которые входят в международные базы данных и системы цитирования

1. Разработка модели генерации клинических рекомендаций для пациентов на основе неструктурированных текстовых данных / Л.С. Гришина, И.П. Болодурина, // Научно-технический вестник Поволжья, 2023. - № 8. - С. 53-56.
2. Разработка модели управления потоком пациентов с сердечно-сосудистыми заболеваниями методами интеллектуального анализа данных / И.П. Болодурина, А.М. Назаров, Д.И. Кича, Л.С. Забродина (Гришина), А.Ю. Жигалов // Вестник ЮУрГУ. Серия: Компьютерные технологии, управление, радиоэлектроника,2020. - Т. 20, № 2. - С. 105-115.
3. L. S. Grishina, A. Yu. Zhigalov, I. P. Bolodurina, E. L. Borshhuk, D. N. Begun, Yu. V. Varennikova, “Investigation of the efficiency of graph data representation for a cardiovascular disease predictive model by deep learning methods”, Dal'nevost. Mat. Zh., 22:2 (2022), 179–184.

В изданиях, индексируемых в Scopus и Web of Science

4. Bolodurina, I.; Shukhman, A.; Legashev, L.; Grishina, L.; Zhigalov, A. Extracting and Processing of Russian Unstructured Clinical Texts for a Medical Decision Support System. *Eng. Proc.* 2023, *33*, 41.
5. Development of a Model for Predicting Treatment of Cardiovascular Diseases Based on Machine Learning Methods / I. P. Bolodurina, D. I. Parfenov, A. Yu. Zhigalov, L. S. Zabrodina (Grishina) // Proceedings of the 2nd International Scientific and Practical Conference "Modern Management Trends and the Digital Economy: from Regional Development to Global Economic Growth", 16-17 April, 2020, Yekaterinburg, Russia - P. 984-989. - 6 с.

В прочих изданиях

6. Обработка русскоязычных неструктурированных медицинских текстов и вероятностное прогнозирование групп заболеваний / Л. В. Легашев, А. Е. Шухман, И. П. Болодурина, Л. С. Гришина, А. Ю. Жигалов // Врач и информационные технологии,2022. - № 4. - С. 52-63.
7. Разработка графовой модели структурных и семантических отношений между сущностями документов для интеллектуальной обработки больших данных / А. Ю. Жигалов, И. П. Болодурина, Д. И. Парфенов, Л. С. Гришина // Перспективные информационные технологии: сб. науч. трудов междунар. науч.-техн. конф., 18-21 апр., 2022, г. Самара. - С. 157-161.
8. Исследование современных архитектур генерации русскоязычного текста на основе неструктурированных медицинских данных/ И.П. Болодурина, Е.Л. Борщук, Л.С. Гришина, А.Ю. Жигалов // Всероссийская научно-методическая конференция ОГУ, 26-27 янв., 2023 г. - С. 3760-3764.

Свидетельство о государственной регистрации программ для ЭВМ

9. Модуль исследования эффективности графового представления данных для модели прогнозирования ССЗ на основе неструктурированных клинических текстов: свидетельство о гос. регистрации программы для ЭВМ 2023610238 / Л. С. Гришина, Ю. В. Варенникова, И. П. Болодурина, А. Е. Шухман, А. Ю. Жигалов, Л. В. Легашев.- опубл. 09.01.2023. - 1 с.

Список литературы

1. Kilgour, C. Experiences of women, hospital clinicians and general practitioners with gestational diabetes mellitus postnatal follow-up: A mixed methods approach. / C. Kilgour, F. Bogossian, L. Callaway, C. Gallois // Diabetes Res Clin Pract. – 2019. – vol. 148. – Pp. 32-42.

2. O'Connor, R. An audit of discharge summaries from secondary to primary care / R.O'Connor, C. O'Callaghan, R. McNamara, U. Salim // Ir J Med Sci. – 2019. – vol. 188. – Pp. 537-540.

3. Sun, W. Data Processing and Text Mining Technologies on Electronic Medical Records: A Review / W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, G. Wang // Journal of healthcare engineering. – 2018. – vol. 4302425. – Pp. 1-10.

4. Hanauer, D.A. Supporting information retrieval from electronic health records: A report of University of Michigan's nine-year experience in developing and using the Electronic Medical Record Search Engine (EMERSE) / D.A. Hanauer, Q. Mei, J. Law, R. Khanna, K. Zheng // Journal of biomedical informatics. – 2015. – vol. 55. – Pp. 290-300.

5. Rumshisky, A. Predicting early psychiatric readmission with natural language processing of narrative discharge summaries / A. Rumshisky, M. Ghassemi, T.Naumann, P.Szolovits, V. M. Castro, T. H. McCoy, R. H. Perlis // Translational psychiatry. – 2016. – Issue 10, vol. 6. – Pp. 1-5.

6. Tsopra, R. Level of accuracy of diagnoses recorded in discharge summaries: A cohort study in three respiratory wards. / R. Tsopra, J.C. Wyatt, P. Beirne, K. Rodger, M. Callister, D. Ghosh, I.J. Clifton, P. Whitaker, D. Peckham // Journal of Evaluation in Clinical Practice. – 2019. – Issue 1, vol. 25. – Pp. 36-43.

7. Graham, A. J. Evaluation of an electronic health record structured discharge summary to provide real time adverse event reporting in thoracic surgery / A. J. Graham, W. Ocampo, D. A. Southern, A. Falvi, D. Sotiropoulos, B. Wang, K. Lonergan, B. Vito, W. A. Ghali, S. D. P. McFadden // BMJ quality & safety. – 2019. – Issue 4, vol. 28. – Pp. 310-316.

8. Лебедев, Г. С. Классификация медицинских информационных систем / Г. С. Лебедев, Ю. Ю. Мухин // Транспортное дело России. – 2012. – № 6 (2). – С. 98-105.

9. Кравченко, Т. К. Экспертная система поддержки принятия решений / Т. К. Кравченко // Открытое образование. – 2010. – № 6. – С. 147-156.

10. Мишкин, И. А. Использование экспертных систем в ранней диагностике соматических заболеваний / И. А. Мишкин // Вестник современных исследований. – 2018. – № 12.4(27). – С. 118-124.

11. Госман, Д. А. Экспертная система прогнозирования риска заболевания туберкулёзом / Д. А. Госман // Донецкие чтения 2021: образование, наука, инновации, культура и вызовы современности: материалы VI Международной научной конференции, Донецк, 26–27 октября 2021 года. Том 3. – Донецк: Донецкий национальный университет, 2021. – С. 9-12.

12. Поворознюк, А. И. Нечеткая экспертная система прогноза риска развития профессионально обусловленных заболеваний / А. И. Поворознюк, Н. А. Чикина, И. В. Антонова // Вестник Национального технического университета Харьковский политехнический институт. Серия: Информатика и моделирование. – 2010. – № 13. – С. 127-132.

13. Choi, E. RETAIN: An Interpretable Predictive Model for Healthcare Using Reverse Time Attention Mechanism / E. Choi, M.T. Bahadori, J. Sun, J. Kulas, A. Schuetz, W. Stewart // Adv. Neural Inf. Process. Syst. – 2016. – vol. 29. – Pp. 3512-3520.

14. Krittanawong, C. Machine learning prediction in cardiovascular diseases: a meta-analysis / C. Krittanawong, H. Virk, S. Bangalore, Z. Wang, K. W. Johnson, R. Pinotti, H. Zhang, S. Kaplin, B. Narasimhan, T. Kitai, U. Baber, J. L. Halperin, W. Tang // Scientific reports. – 2020. – Issue 1, vol. 10. – Pp. 1-11.

15. Pasha, S. Cardiovascular disease prediction using deep learning techniques / S. Pasha, D. Ramesh, M. Sallauddin, A. Harshavardhan // IOP Conf. Series: Materials Science and Engineering. – 2020. – Vol. 981. – Pp. 1-6.

16. Bharti, R. Prediction of Heart Disease Using a Combination of Machine Learning and Deep Learning / R. Bharti, A. Khamparia, M.Shabaz, G. Dhiman, S. Pande, P. Singh // Computational Intelligence and Neuroscience. – 2021. – Vol. 21. – Pp. 1-11.

17. Subramanian, M. Prediction of cardiovascular disease using deep learning algorithms to prevent COVID 19 / M. Subramanian, Y. Arockia, A. Kumar, V.D. Kumar, D. Elangovan, B. Chitra // Journal of Experimental & Theoretical Artificial Intelligence. – 2021. – Vol. 1. – Pp. 1-15.

18. Lemmon, G. A Poisson binomial-based statistical testing framework for comorbidity discovery across electronic health record datasets / G. Lemmon, S. Wesolowski, A. Henrie, M. Tristani-Firouzi, M. Yandell // Nature Computational Science. – 2021. – Vol. 1. – Pp. 694–702.

19. Chase, H.S. Early recognition of multiple sclerosis using natural language processing of the electronic health record / H.S. Chase, L.R. Mitrani, G.G. Lu, D.J. Fulgieri // BMC medical informatics and decision making. – 2017. – Issue 1, vol. 17. – Pp. 1-8.

20. Zhao, S.S. Incorporating natural language processing to improve classification of axial spondyloarthritis using electronic health records / S.S. Zhao, C. Hong, T. Cai, C. Xu, J. Huang, J. Ermann, N.J. Goodson, D.H. Solomon, K.P. Liao // Rheumatology (Oxford). – 2020. – Issue 5, vol. 59. – Pp. 1059-1065.

21. Sada, Y. Validation of case finding algorithms for hepatocellular cancer from administrative data and electronic health records using natural language processing / Y. Sada, J. Hou, P. Richardson, H. El-Serag, J. Davila // Medical care. – 2016. – Issue 2, vol. 54. – Pp. 1-15.

22. Zheng, L. Web-based real-time case finding for the population health Management of Patients with Diabetes Mellitus: a prospective validation of the natural language processing–based algorithm with statewide electronic medical records / L. Zheng, Y. Wang, S. Hao, A.Y. Shin, B. Jin, A.D. Ngo, M.S. Jackson-Browne,

D.J. Feller, T. Fu, K. Zhang, X. Zhou, C. Zhu, D. Dai, Y. Yu, G. Zheng, Y.M. Li, D.B. McElhinney, D.S. Culver, S.T. Alfreds, F. Stearns, K.G. Sylvester, E. Widen, X.B. Ling // JMIR medical informatics. – 2016. – Issue 4, vol. 4. – Pp. 1-13.

23. Castro, V.M. Validation of electronic health record phenotyping of bipolar disorder cases and controls / V. M. Castro, J. Minnier, S.N. Murphy, I. Kohane, S. E. Churchill, V. Gainer, T. Cai, A. G. Hoffnagle, Y. Dai, S. Block, S. R. Weill, M. Nadal-Vicens, A. R. Pollastri, J. N. Rosenquist, S.Goryachev, D. Ongur, P. Sklar, R. H. Perlis, J. W. Smoller // American Journal of Psychiatry. – 2015. – Issue 4, vol. 172. – Pp. 363-372.

24. Hazlehurst, B. Using natural language processing of clinical text to enhance identification of opioid-related overdoses in electronic health records data / B. Hazlehurst, C.A. Green, N.A. Perrin, J. Brandes, D.S. Carrell, A. Baer, A. DeVeaugh-Geiss, P. M. Coplan // Pharmacoepidemiology and drug safety. – 2019. – Issue 8, vol. 28. – Pp. 1143-1151.

25. Wang, M. Deep learning model for multi-classification of infectious diseases from unstructured electronic medical records / M. Wang, Z. Wei, M. Jia, L. Chen, H. Ji // BMC medical informatics and decision making. – 2022. – Issue 1, vol. 22. – Pp. 1-13.

26. Ling, A.Y. Using natural language processing to construct a metastatic breast cancer cohort from linked cancer registry and electronic medical records data / A.Y. Ling, A.W. Kurian, J.L. Caswell-Jin, G.W. Sledge, N.H. Shah, S.R. Tamang // JAMIA open. – 2019. – Issue 4, vol. 2. – Pp. 528-537.

27. Weissman, G.E. Natural Language Processing to Assess Documentation of Features of Critical Illness in Discharge Documents of Acute Respiratory Distress Syndrome Survivors / G.E. Weissman, M.O. Harhay, R.M. Lugo, B.D. Fuchs, S.D. Halpern, M.E. Mikkelsen // Annals of the American Thoracic Society. – 2016. – Vol. 13. – Pp. 1538-1545.

28. Rumshisky, A.A. Predicting early psychiatric readmission with natural language processing of narrative discharge summaries / A.A. Rumshisky, M. Ghassemi, T. Naumann, P. Szolovits, V.M. Castro, T.McCoy, R.H. Perlis // Translational psychiatry. – 2016. – Issue 10, vol. 6. – Pp. 1-5.

29. Berman, A.N. Natural language processing for the assessment of cardiovascular disease comorbidities: The cardio-Canary comorbidity project / A.N. Berman, D.W. Biery, C. Ginder, O.L. Hulme, D. Marcusa, O. Leiva, W.Y. Wu, N. Cardin, J. Hainer, D.L. Bhatt, M.F. Carli, A. Turchin, R. Blankstein // Clinical Cardiology. – 2021. – Issue 9, vol. 44. – Pp. 1296-1304.

30. Müller, M. Covid-twitter-bert: A natural language processing model to analyse COVID-19 content on twitter / M. Müller, M. Salathé, P.E. Kummervold // Frontiers in artificial intelligence. – 2023. – Vol. 6. – Pp. 1-6.

31. Padmanaban, K.R. Applying machine learning techniques for predicting the risk of chronic kidney disease / K.R. Anantha Padmanaban, G. Parthiban // Indian Journal of Science and Technology. – 2016. – Issue 9, vol. 29. – Pp. 1-6.

32. Radha, P. Machine learning approaches for disease prediction from radiology and pathology reports / P. Radha, B. MeenaPreethi // Journal of Green Engineering. – 2019. – Issue 2, vol. 9. – Pp. 149-166.

33. Mathur, R. Parkinson disease prediction using machine learning algorithm / R. Mathur, P. Vibhakar, B. Devesh // In Emerging Trends in Expert Applications and Security: Proceedings of ICETEAS. – 2019. – Pp. 357–363.

34. Демченко, М. В., Каширина, И. Л., Фирюлина, М. А. Кластеризация состояний пациентов для модели назначения схем лечения атеросклероза / М.В. Демченко, И.Л. Каширина, М. А Фирюлина // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2021. – № (2). – С. 126-137.

35. Harrison, C.J. Machine learning in medicine: a practical introduction to natural language processing / C.J. Harrison, C.J. Sidey-Gibbons // In BMC Med Res Methodol. – 2021. – Issue 21, vol. 158. – Pp. 1-11.

36. Zhou, L. Adapting State-of-the-Art Deep Language Models to Clinical Information Extraction Systems: Potentials, Challenges, and Solutions / L. Zhou, H. Suominen, T. Gedeon // In JMIR Med Inform. – 2019. – Issue 7, vol. 25. – Pp. 1-10.

37. Демченко Е.В. Интеллектуальная система поддержки принятия решений для формирования схем лечения на основе методов машинного обучения с подкреплением: дис. ... канд. техн. наук: 05.13.01. - Нижегородского государственного технического университета им. Р.Е. Алексеева, Нижний Новгород, 2022 - 154 с.

38. Rasmy, L. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction / L. Rasmy, Y. Xiang, Z. Xie, C. Tao, D. Zhi // NPJ digital medicine. – 2021. – Issue 1, vol. 4. – Pp. 1-13.

39. Li, I. Neural natural language processing for unstructured data in electronic health records: A review / I. Li, J. Pan, J. Goldwasser, N. Verma, W. Wong, M. Nuzumlalı, B. Rosand, Y. Li, M. Zhang, D. Chang, R. Taylor, H. Krumholz, D. Radev // Computer Science Review. – 2022. – Vol. 46. – Pp. 1-29.

40. Syed, S. The h-ANN Model: Comprehensive Colonoscopy Concept Compilation Using Combined Contextual Embeddings / S. Syed, A. J. Angel, H. B. Syeda, C. F. Jennings, J. VanScoy, M. Syed, M. Greer, S. Bhattacharyya, M. Zozus, B. Tharian, F. Prior // NIH Public Access. – 2022. – Vol. 5. – Pp. 1-24.

41. Gudkov, V. Automatically ranked Russian paraphrase corpus for text generation / V. Gudkov, O. Mitrofanova, E. Filippskikh // In arXiv preprint arXiv:2006.09719. – 2020. – Pp. 1-6.

42. Gu, Y. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing / Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon // In ACM Trans. Comput. Healthcare. – 2022. – Issue 1, vol. 3. – Pp. 1-23.

43. Lee, J. BioBERT: a pre-trained biomedical language representation model for biomedical text mining / J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang // Bioinformatics. – 2020. – Issue 4, vol. 36. – Pp. 1234-1240.

44. Ive, J. Generation and evaluation of artificial mental health records for Natural Language Processing / J. Ive, N. Viani, J. Kam, J. Kam, L. Yin, S. Verma, S. Puntis, R. Cardinal, A. Roberts, R. Stewart, S. Velupillai // NPJ Digit Med. – 2020. – Issue 3, vol. 69. – Pp. 1-10.

45. Bejan, C.A. Improving ascertainment of suicidal ideation and suicide attempt with natural language processing / C.A. Bejan, M. Ripperger, D. Wilimitis, R. Ahmed, J. Kang, K. Robinson, T. Morley, D. Ruderfer, C. Walsh // Scientific Reports. – 2022. – Vol. 12. – Pp. 1-11.

46. Silver, J. Assessment of Women Physicians Among Authors of Perspective / J. Silver, J. Poorman, J. Reilly, N. Spector, R. Goldstein, R. Zafonte // Pediatric Journals. JAMA Netw Open. – 2022. – Issue 3, vol. 1. – Pp. 1-13.

47. Yalunin, A. RuBioRoBERTa: a pre-trained biomedical language model for Russian language biomedical text mining / A. Yalunin, A. Nesterov, D. Umerenkov // In arXiv preprint arXiv: 2204.03951. – 2022. – Pp. 1-5.

48. Blinov, P. RuMedBench: A Russian Medical Language Understanding Benchmark / P. Blinov, A. Reshetnikova, A. Nesterov, G. Zubkova, V. Kokh // In arXiv preprint arXiv: 2201.06499. – 2022. – Pp. 383-392.

49. Funkner, A.A. Negation Detection for Clinical Text Mining in Russian / A.A. Funkner, K. Balabaeva, S.V. Kovalchuk // MIE. – 2020. – Pp. 342-346.

50. Balabaeva, K. Automated Spelling Correction for Clinical Text Mining in Russian / K. Balabaeva, A.A. Funkner, S.V. Kovalchuk // MIE. – 2020. – Pp. 43-47.

51. Тутубалина Е.В. Модели и методы автоматической обработки неструктурированных данных в биомедицинской области: дис. ... доктор компьютерных наук: 05.13.01. – Казанский (Приволжский) федеральный университет, Казань, 2023 - 225 с.

52. Пищухина Т.А. Проектирование системы поддержки принятия решений на основе онтологии перекомпонуемого производства // Онтология проектирования. 2022. Т. 12, №2(44). С.231-244.

53. Nguyen, A. Classification of pathology reports for cancer registry notifications / A. Nguyen // Stud. Health Technol. Inform. – 2012. – vol. 178. – Pp. 150-156.

54. Coden, A. Automatically extracting cancer disease characteristics from pathology reports into a Disease Knowledge Representation Model / A. Coden // J. Biomed. Inform. – 2009. – vol. 42. – Pp. 937-949.

55. Tanenblatt, M. The ConceptMapper approach to named entity recognition / M. Tanenblatt // Proc. Seventh Conf. Int. Lang. Resour. Eval.Lr. –2010. – Pp. 546-551.

56. Sorace, J. Integrating pathology and radiology disciplines: an emerging opportunity? / J. Sorace, D. Aberle, D. Elimam, S. Lawvere, O. Tawfik, D. Wallace // BMC Medicine. – 2012. – vol. 10. – Pp. 1-6.

57. Kocbek, S. Evaluating classification power of linked admission data sources with text mining / S. Kocbek, L. Cavedon, D. Martinez, C. Bain, C. MacManus, G. Haffari, I. Zukerman // CEUR Workshop Proceedings. – 2015. – Pp. 1-7.

58. Bain C. Advancing data management and usage in a major Australian health service / C. Bain, C. Manus // 2014 International Conference on Data Science & Engineering (ICDSE). – 2014. – Pp. 1-6.

59. Spasic´, I. Keane Text mining of cancer-related information: review of current status and future directions / I. Spasic´ , J. Livsey, J.A. Keane // Int. J. Med. Inform. – 2014. – vol.83. – Pp.605–623.

60. Sokolov, A. Combining heterogeneous data sources for accurate functional annotation of proteins / A. Sokolov, C. Funk, K. Graim // BMC Bioinformatics. – 2013.

61. Hripcsak, G. Use of natural language processing to translate clinical information from a database of 889,921 chest radiographic reports / G. Hripcsak, J.H.M. Austin, P.O. Alderson // Radiology. – 2002. – Pp.157-163.

62. Goldgrab, D. Updates in heart failure 30-day readmission prevention / D. Goldgrab, K. Balakumaran, M.J. Kim, S.R. Tabtabai // Heart Fail. Rev. – 2019. – vol. 24. – Pp. 177-187.

63. Gilbert, A.V. An audit of medicines information quality in electronically generated discharge summaries–evidence to meet the Australian National Safety and Quality Health Service Standards / A.V. Gilbert, B.K. Patel, M.S. Roberts, D.B. Williams, J.H. Crofton, N.M. Morris, J. Wallace, A.L. Gilbert // J. Pharm. Wiley Online Libr. – 2017. – vol. 47. – Pp. 355-364.

64. Schwarz, C.M. A systematic literature review and narrative synthesis on the risks of medical discharge letters for patients' safety / C.M. Schwarz, M. Hoffmann, P. Schwarz, L.P. Kamolz, G. Brunner, G.J. Sendlhofer // BMC Health Servres. – 2019. – vol. 19. – Pp. 1-10.

65. Liang, H. Evaluation and accurate diagnoses of pediatric diseases using artificial intelligence / H. Liang, B.Y. Tsui, H. Ni, C.C. Valentim, S.L. Baxter, G. Liu, W. Cai, D.S. Kermany, X. Sun, J.J. Chen // Nat. Med. – 2019. – vol. 25. – Pp. 433-438.

66. Reátegui, R. Comparison of MetaMap and cTAKES for entity extraction in clinical notes / R. Reátegui, S.J. Ratté, D. Making // BMC Med. Inform. Decis. Mak. – 2018. – vol. 18. – Pp. 13-19.

67. Servid, S.A. Clinical intentions of antibiotics prescribed upon discharge to hospice care / S.A. Servid, B.N. Noble, E.K. Fromme, J.P. Furuno // J. Am. Heart Assoc. Wiley Online Libr. – 2018. – vol. 66. – Pp. 565-569.

68. Xu, J. Unsupervised medical entity recognition and linking in Chinese online medical text / J. Xu, L. Gan, M. Cheng, Q.J. Wu // J. Healthc. Eng. – 2018. – Pp. 1-13.

69. Jiménez-Zafra, S.M. How do we talk about doctors and drugs? Sentiment analysis in forums expressing opinions for medical domain / S.M. Jiménez-Zafra, M.T. Martín-Valdivia, M.D. Molina-González, L.A. Ureña-López // Artif. Intell. Med. – 2019. – vol. 93. – Pp. 50-57.

70. Abualigah, L. Sentiment Analysis in Healthcare: A Brief Review / L. Abualigah, H.E. Alfar, M. Shehab, A.M.A. Hussein // In Recent Advances in NLP: The Case of Arabic Language; Springer – 2020. – Pp. 129-141.

71. Melo, P.F. 10SENT: A stable sentiment analysis method based on the combination of off-the-shelf approaches. / P.F. Melo, D.H. Dalip, M.M. Junior, M.A. Gonçalves, F.J. Benevenuto // J. Assoc. Inf. Sci. Technol. – 2019. – vol. 70. – Pp. 242-255.

72. Al-Smadi, M. Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews / M. Al-Smadi, B. Talafha, M. Al-Ayyoub, Y.J. Jararweh // Int. J. Mach. Learn. Cybern. – 2019. – vol. 10. – Pp. 2163-2175.

73. Nguyen, A. T. A deep neural network language model with contexts for source code / A. T. Nguyen, T. D. Nguyen, H. D. Phan, T. N. Nguyen // 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), Campobasso, Italy. – 2018. – Pp. 323-334.

74. Sivarethinamohan R. Envisioning the potential of Natural Language Processing (NLP) in Health Care Management / R. Sivarethinamohan, S. Sujatha, P. Biswas // 2021 7th International Engineering Conference “Research & Innovation amid Global Pandemic" (IEC), Erbil, Iraq. – 2021. – Pp. 189-193.

75. Sankar, H. Intelligent sentiment analysis approach using edge computing-based deep learning technique / H. Sankar, V. Subramaniyaswamy, V. Vijayakumar, S. Arun Kumar, R. Logesh, A.J. Umamakeswari // Softw. Pract. Exp. Wiley Online Libr. – 2019.

76. Wang, Y. Feature Weighting Based on Inter-Category and Intra-Category Strength for Twitter Sentiment Analysis / Y. Wang, H.J. Youn // Appl. Sci. – 2019. – vol. 9. – Pp. 1-18.

77. Dehkharghani, R. SentiTurkNet: A Turkish polarity lexicon for sentiment analysis. / R. Dehkharghani, Y. Saygin, B. Yanikoglu, K.J. Oflazer // Lang. Resour. Eval. – 2016, – vol. 50. – Pp. 667-685.

78. Wang, Y. A review of sentiment semantic analysis technology and progress / Y. Wang, Y. Rao, L. Wu // 13th International Conference on Computational Intelligence and Security (CIS) – 2017. – Pp. 452-455.

79. Mohammad, S.M. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets / S.M. Mohammad, S. Kiritchenko, X.J. Zhu // Conference on Empirical Methods in Natural Language Processing. – 2019. – vol 1 – Pp. 1-8.

80. Gilmore-Bykovskyi, A.L. Hospital discharge documentation of a designated clinician for follow-up care and 30-day outcomes in hip fracture and stroke patients discharged to sub-acute care / A.L. Gilmore-Bykovskyi, K.A. Kennelty, E. DuGoff, A.J. Kind // BMC Health Servres. – 2018. – vol. 18 – Pp. 1-7.

81. Mehta, R.L. Assessing the impact of the introduction of an electronic hospital discharge system on the completeness and timeliness of discharge communication: A before and after study / R.L. Mehta, B. Baxendale, K. Roth, V. Caswell, I. Le Jeune, J. Hawkins, H. Zedan, A.J. Avery // BMC Health Servres. – 2017. – vol. 17. – Pp. 1-10.

82. Ooi, C.E. Improving communication of medication changes using a pharmacist-prepared discharge medication management summary / C.E. Ooi, O. Rofe, M. Vienet, R.A. Elliott // Int. J. Clin. Pharm. – 2017. – vol. 39 – Pp. 394-402.

83. Pereira-Kohatsu, J.C. Detecting and Monitoring Hate Speech in Twitter / J.C. Pereira-Kohatsu, L. Quijano-Sánchez, F. Liberatore, M.J.S. Camacho-Collados // Sensors. – 2019. – vol. 19. – Pp 1-37.

84. Flores, A.C. An Evaluation of SVM and Naive Bayes with SMOTE on Sentiment Analysis Data Set. / A.C. Flores, R.I. Icoy, C.F. Peña, K.D. Gorro // In Proceedings of the 2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST). – 2018. – Pp. 1-4.

85. Ahmad, M. SVM optimization for sentiment analysis / M. Ahmad, S. Aftab, M.S. Bashir, N. Hameed, I. Ali, Z.J. Nawaz // Int. J. Adv. Comput. Sci. Appl. – 2018. – vol. 9. – Pp. 393-938.

86. Gupta, S. Opinion Mining for Hotel Rating through Reviews Using Decision Tree Classification Method / S. Gupta, S. Jain, S. Gupta, A.J. Chauhan // Int. J. Adv. Res. Comput. Sci. – 2018. – vol. 9. – Pp. 180-184.

87. Ma, Y. Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis / Y. Ma, H. Peng, T. Khan, E. Cambria, A.J. Hussain // Cogn. Comput. – 2018. – vol. 10. – Pp. 639-650.

88. Spinczyk, D. Computer aided sentiment analysis of anorexia nervosa patients' vocabulary / D. Spinczyk, K. Nabrdalik, K.J. Rojewska // Biomed. Eng. Online. – 2018. – vol. 17. Pp. 1-12.

89. Jiang, K. Identifying tweets of personal health experience through word embedding and LSTM neural network / K. Jiang, S. Feng, Q. Song, R.A. Calix, M. Gupta, G.R. Bernard // BMC Bioinform. – 2018. – vol. 19. – Pp. 67-74.

90. LeCun, Y. Deep learning / Y. LeCun, Y. Bengio, G.J.N Hinton // Nature 2015. –vol. 521 – Pp. 436-444.

91. Sun, K. Generalized extreme learning machine autoencoder and a new deep neural network / K. Sun, J. Zhang, C. Zhang, J.J. Hu // Neurocomputing. –2017 – vol. 230. – Pp. 374-381.

92. Waheeb, S.A. Multi-Document Arabic Summarization Using Text Clustering to Reduce Redundancy / S.A. Waheeb, H.J. Husni // Int. J. Adv. Sci. Technol. – 2014. – vol. 2. – Pp. 194-199.

93. Waheeb, S.A. Multidocument Arabic Text Summarization Based on Clustering and Word2Vec to Reduce Redundancy / S.A. Waheeb, B. Chen, X. Shang // Information. – 2020. – vol. 11. – Pp. 1-13.

94. Huang, G.B. Extreme learning machine: Theory and applications / G.B. Huang, Q.Y. Zhu, C.K. Siew // Neurocomputing – 2006. – vol. 70. – Pp. 489-501.

95. Reese, R.M. Natural Language Processing with Java / R.M. Reese // Packt Publishing Ltd.: Birmingham, UK. – 2015.

96. Kho, A. Electronic Medical Records for Genetic Research: Results of the eMERGE Consortium / A. Kho, J. Pacheco, P. Peissig, L. Rasmussen, K. Newton, N. Weston, P. Crane, J. Pathak, C. Chute, S. Bielinski, I. Kullo, R. Li, T. Manolio, R. Chisholm, J. Denny // Science translational medicine. – 2011. – Issue 79, vol. 3. – Pp. 1-7.

97. Bao, J. Text Generation From Tables / J. Bao, D. Tang, N. Duan, Z. Yan, M. Zhou, T. Zhao // IEEE/ACM Transactions on Audio, Speech, and Language Processing. – 2019. – Issue 2, vol. 27. – Pp. 311-320.

98. Dien, T. T. Article Classification using Natural Language Processing and Machine Learning / T. T. Dien, B. H. Loc, N. Thai-Nghe // 2019 International Conference on Advanced Computing and Applications (ACOMP), Nha Trang, Vietnam. – 2019. – Pp. 78-84.

99. Yao, L. Deep learning of semisupervised process data with hierarchical extreme learning machine and soft sensor application / L. Yao, Z.J. Ge // IEEE Trans. Ind. Electron. – 2017. – vol. 65. – Pp. 1490-1498.

100. Huang, G. Trends in extreme learning machines: A review / G. Huang, G.B. Huang, S. Song, K.J.N.N. You // Neural Netw. – 2015. – vol 61. – Pp. 32–48.

101. Mikolov, T. Distributed representations of words and phrases and their compositionality / T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean // NIPS. – 2013. – Pp. 3111-3119.

102. Lipton, Z.C. Learning to Diagnose with LSTM Recurrent Neural Networks / Z.C. Lipton, D.C. Kale, C. Elkan, R. Wetzel // International Conference on Learning Representations (ICLR) – 2016.

103. Geraci, J. Applying deep neural networks to unstructured text notes in electronic medical records for phenotyping youth depression / J. Geraci, P. Wilansky, V. de Luca, A. Roy, J. L. Kennedy, J. Strauss // Evid-Based Ment Health. – 2017. – Issue 3, vol. 20. – Pp 83-90.

104. Nath, N. The quest for better clinical word vectors: Ontology based and lexical vector augmentation versus clinical contextual embeddings / N. Nath, S.H. Lee, M.D. McDonnell, I. Lee // Computers in Biology and Medicine. – 2021. – Vol. 134. – Pp. 1-11.

105. Yang, X. A large language model for electronic health records / X. Yang, A. Chen, N. PourNejatian, C. Harle, W. Hogan, E. Shenkman, J. Bian, Y. Wu // NPJ Digit Med. – 2022. – Issue 5, vol. 194. – Pp. 1-11.

106. Kocbek, S. Text mining electronic hospital records to automatically classify admissions against disease: Measuring the impact of linking data sources / S. Kocbek, L. C. D. Martinez // Journal of Biomedical Informatics. – 2016. – vol. 64. – Pp. 158-167.

107. Nguyen, A.N. Symbolic rule-based classificationof lung cancer stages from free-text pathology reports / A.N. Nguyen // J. Am. Med. Inform.Assoc. – 2010. – vol. 17. – Pp. 440-445.

Приложение А
(обязательное)

Листинг программного модуля обработки XML-протоколов

```
import XML.etree.ElementTree as ET
from lXML import etree
import re
import pprint
import os

def content_data_recursion(node, content_node_dict):

  data_name_bool = node.xpath('data')
  items_name_bool = node.xpath('items')
  events_name_bool = node.xpath('events')

  if(len(data_name_bool) != 0):
    data_name_arr =  data_name_bool[0].xpath('name/value/text()')
    data_name = data_name_arr[0]
    content_node_dict['name'] = data_name
    content_node_dict['data'] = {}
    content_node_dict['data'] = content_data_recursion(data_name_bool[0],
content_node_dict['data'])

  elif (len(events_name_bool)!=0):

    content_node_dict = content_data_recursion(events_name_bool[0], content_node_dict)


  elif (len(items_name_bool)!=0):
    items_values = {}
    '''
    content_items = {}
    for ind, item in enumerate(items_name_bool):
      if (len(item.xpath('value')) == 0):
        content_item_dict = {
          'name': item.xpath('name/value/text()')[0],
          'data': {}
        }
        content_items[ind] = content_data_recursion(item, content_item_dict['data'])

    content_node_dict = {
      'name': items_name_bool[0].xpath('name/value/text()')[0],
      'items': content_items
    }

    for ind, item in enumerate(items_name_bool):
      if (len(item.xpath('value')) > 0):
        if (len(item.xpath('value/value')) > 0):
```

```
        item_dict_one = {
            'name': item.xpath('name/value/text()')[0],
            'value': item.xpath('value/value/text()')[0],
        }
        items_values[ind]=item_dict_one

    return items_values
    '''
    for ind, item in enumerate(items_name_bool):
      if (len(item.xpath('value')) > 0):
        if (len(item.xpath('value/value')) > 0):
          item_name_arr = item.xpath('name/value/text()')
          item_value_arr = item.xpath('value/value/text()')

          if (len(item_name_arr) == 0) or (len(item_value_arr)==0):
            continue

          item_dict_one = {
            'name': item.xpath('name/value/text()')[0],
            'value': item.xpath('value/value/text()')[0],
          }
          items_values[ind]=item_dict_one
        else:
          content_item_dict = {
            'name': item.xpath('name/value/text()')[0],
            'data': {}
          }
          items_values[ind] = content_data_recursion(item, content_item_dict['data'])
    return {
        'name': items_name_bool[0].xpath('name/value/text()')[0],
        'items': items_values,
    }
  return content_node_dict

def parse_XML_file(filepath):
    namespaces = {'xsi':'http://www.w3.org/2001/XMLSchema-instance'}
    fp = open(filepath, mode ='r', encoding = "utf-8")
    parser = etree.XMLParser(recover=True)
    XML_text = fp.read()
    fp.close()
    root = None
    try:
        root = etree.fromstring(XML_text, parser=parser)
        pattern = r'<[^">]*<' # < =
        searching = re.findall(pattern, XML_text) # меняем
        for search_rep in searching:
          replace_substring = search_rep[:-1] # все кроме последнего тега
          replace_string = re.sub('<','&lt;', replace_substring)
          XML_text = re.sub(replace_substring, replace_string, XML_text)
        root = etree.fromstring(XML_text, parser=parser)
    except:
        # исправление XML
```

```
    try:
        # шаблон для правки

        pattern = r'<[^">]*<'
        XML_text_new = re.sub(pattern, '<', XML_text) # меняем
        root = etree.fromstring(XML_text_new, parser=parser)
    except:
        print(f'Error: {filepath}')
# после того как спарсили XML в root пробуем парсить
# Шаг 1: цепляемся к корневому тегу data
XML_data_arr = root.xpath('/version/data', namespaces=namespaces)
if len(XML_data_arr) == 0:
    return {}
XML_data = XML_data_arr[0] # data 1 элемент
data_name_arr = XML_data.xpath('name/value/text()')
data_name = 'Пусто'
if len(data_name_arr)>0:
    data_name = data_name_arr[0]
# Шаг 2: достаем все content
contents = XML_data.xpath('content')
values = XML_data.xpath('//value/value/text()')
#print(values)
#print(len(values))
contents_dicts = []
if len(contents) == 0:
    return {}
for ind, content in enumerate(contents):
    # каждый content обрабатываем отдельно - и вытаскиваем название секции content
    content_dict = {}
    content_data = {}
    content_name = ''
    is_have_name = False
    content_name_arr = content.xpath('name/value/text()')

    if len(content_name_arr)==0:
        content_name = f'content_{ind}'
    else:
        content_name = content_name_arr[0]
        is_have_name = True

    content_data_recursion(content, content_data)
    content_dict = {
        'name': content_name,
        'data': content_data
    }
    cut_dict(content_dict)

    if len(content_dict.keys()) > 0:
      contents_dicts.append(content_dict)

return {
    'name': data_name,
```

```
        'contents': contents_dicts
    }

d = parse_XML_file(filepath)
t = d['contents'][5]['data']
t

def cut_dict(r):
    node_remove = True
    keys_remove = []
    if isinstance(r, dict):
        if 'value' in r.keys():
            return False
        else:
            for key in r.keys():
                is_cut_node = cut_dict(r[key])
                if is_cut_node == True:
                    keys_remove.append(key)
                node_remove = is_cut_node and node_remove
    for key in keys_remove:
        if key=='name' and node_remove==False: # оставляем name при заполненных values
            continue
        r.pop(key)
    return node_remove

cut_dict(t)
t

from google.colab import drive
drive.mount('/content/gdrive')

name_XML = []

for filename in os.listdir("gdrive/My Drive/data_XML"):
    if filename.endswith("XML"):
        name_XML.append(filename)

len(name_XML)

contens = []
for ind, fp in enumerate(name_XML):
  print(fp)
  content_one = parse_XML_file("gdrive/My Drive/data_XML/"+fp)
  if ind % 20 == 0:
    pprint.pprint(content_one)
  contens.append(content_one)

pprint.pprint(contens)
```

Приложение Б
(обязательное)

Листинг программного модуля классификации заболеваний

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
from matplotlib import pyplot as plt
import os
import re
from math import floor
import scipy.stats as stats
import glob
import nltk

data_folder = "/kaggle/input/disgnosis/Data_diagnosis_complaints_with_anamnesis.xlsx"

df = pd.read_excel(data_folder)

## Обработка данных

def mkb_get_code(x):
    if str(x)=='nan':
        return np.nan
    else:
        if re.match("[A-Z]\d{1,2}\.{0,1}\d*",str(x)) is not None:
            return re.match("[A-Z]\d{1}",str(x)).group()
        else:
            return np.nan

df['mkb'] = df['mkb_code'].apply(lambda x: mkb_get_code(x))

"""Удаляем записи с пропущенными значениями по полям 'mkb' и 'patient complaints'"""

df = df[df['mkb'].isna() == False]

df['patient_condition'] = df['anamnesis_disease'] + " " +  df['patient_complaints']

df = df[df['patient_complaints'].isna() == False]

df = df[df['anamnesis_disease'].isna() == False]

"""Оставляет только поля 'mkb' и 'patient complaints'"""

df = df[["mkb","patient_condition"]]
```

```
"""## Базовые модели"""

# Commented out IPython magic to ensure Python compatibility.
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split

from io import StringIO
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import classification_report,confusion_matrix

df_copy = df.copy()

# encode для label
encode_dict = {}

def encode_cat(x):
    if x not in encode_dict.keys():
        encode_dict[x]=len(encode_dict)
    return encode_dict[x]

df_copy['label'] = df_copy['mkb'].apply(lambda x: encode_cat(x))

df_copy = df_copy[["label", "patient_condition"]]

nltk.download('stopwords')

from nltk.corpus import stopwords
", ".join(stopwords.words('russian'))
STOPWORDS = set(stopwords.words('russian'))

# Lower casing
def lower(text):
    low_text= text.lower()
    return low_text
df_copy['text_new']=df_copy['patient_condition'].apply(lambda x:lower(x))

#Remove stopwords & Punctuations
```

```
def punct_remove(text):
    punct = re.sub(r"[^\w\s\d]"," ", text)
    return punct
df_copy['text_new']=df_copy['text_new'].apply(lambda x:punct_remove(x))

#Remove extra white space left while removing stuff
def remove_space(text):
    space_remove = re.sub(r"\s+"," ",text).strip()
    return space_remove
df_copy['text_new']=df_copy['text_new'].apply(lambda x:remove_space(x))

def remove_stopwords(text):
    """custom function to remove the stopwords"""
    return " ".join([word for word in str(text).split() if word not in STOPWORDS])

df_copy['patient complaints']=df_copy['text_new'].apply(lambda x:remove_stopwords(x))

df_copy = df_copy.drop(columns=['text_new'])

df_copy = df_copy[["label", "patient complaints"]]

"""Преобразуем текст в вектора на основе метода tf-idf"""

tfidf = TfidfVectorizer(sublinear_tf=True, min_df=10, ngram_range=(1,4),
                        stop_words=list(STOPWORDS))
# We transform each text into a vector
features = tfidf.fit_transform(df_copy["patient complaints"]).toarray()

labels = df_copy["label"]

print("Each of the %d patient complaints is represented by %d features (TF-IDF score of unigrams
and bigrams)" %(features.shape))

import pickle
pickle.dump(tfidf, open("vectorizer.pickle", "wb"))

vectorizer = pickle.load(open("vectorizer.pickle", "rb"))

"""Делим данные на Train и Test"""

X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.30, stratify = labels,
                                                    random_state=1)

y_train.value_counts()

y_test.value_counts()
```

```
count = 0
for C_i in [0.01, 0.1,  1, 10, 100]:
    for penalty_i in ['l2']:
        count +=1
        LR = LogisticRegression(C=C_i,penalty=penalty_i)
        LR.fit(X_train, y_train)
        LR_predictions = LR.predict(X_test)
        print(C_i, penalty_i)
        print("accuracy",metrics.accuracy_score(y_test,LR_predictions))
        print("balanced_accuracy",metrics.balanced_accuracy_score(y_test,LR_predictions))
        print(confusion_matrix(y_test,LR_predictions))
        print(classification_report(y_test,LR_predictions))

LR = LogisticRegression(C=100,penalty='l2')
LR.fit(X_train, y_train)

import pickle
filename = 'finalized_model.sav'
pickle.dump(LR, open(filename, 'wb'))


"""## RandomForest c GridSearchCV """

count = 0
for n_estimators_i in [50, 100]:
    for max_depth_i in [70, 100, 150]:
        for criterion_i in ['gini', 'entropy']:
            count +=1
            rfc = RandomForestClassifier(random_state=42, n_estimators = n_estimators_i, criterion
=criterion_i, max_depth = max_depth_i)
            rfc.fit(X_train, y_train)
            rfc_predictions = rfc.predict(X_test)
            print(n_estimators_i, max_depth_i, criterion_i)
            print("accuracy",metrics.accuracy_score(y_test,rfc_predictions))
            #print("f1-score",metrics.f1_score(y_test,rfc_predictions))
            print("balanced_accuracy",metrics.balanced_accuracy_score(y_test,rfc_predictions))
            print(confusion_matrix(y_test,rfc_predictions))
            print(classification_report(y_test,rfc_predictions))

"""## SVC c GridSearchCV"""

param_grid = {'C': [0.1,1, 10],
              'gamma': [1,0.1,0.01,0.001],
              'kernel': ['linear','rbf', 'poly', 'sigmoid']}

count = 0
```

```
for C_i in [0.01, 0.1,  1, 10, 100]:
    for dual_i in [True,False]:
        count +=1
        svc = LinearSVC(random_state=42, penalty='l2', loss='squared_hinge', C = C_i, dual =dual_i)
        svc.fit(X_train, y_train)
        svc_predictions = svc.predict(X_test)
        print(C_i, dual_i)
        print("accuracy",metrics.accuracy_score(y_test,svc_predictions))
        #print("f1-score",metrics.f1_score(y_test,svc_predictions))
        print("balanced_accuracy",metrics.balanced_accuracy_score(y_test,svc_predictions))
        print(confusion_matrix(y_test,svc_predictions))
        print(classification_report(y_test,svc_predictions))

"""## MultinomialNB c GridSearchCV"""

cv_NB = MultinomialNB()

count = 0
for fit_prior_i in [False, True]:
    for alpha_i in [2, 1, 0.1, 0.01, 0.001, 0.0001, 0.00001]:
        count +=1
        MultiNB = MultinomialNB(fit_prior = fit_prior_i, alpha = alpha_i)
        MultiNB.fit(X_train, y_train)
        MultiNB_predictions = MultiNB.predict(X_test)
        print(fit_prior_i, alpha_i)
        print("accuracy",metrics.accuracy_score(y_test,MultiNB_predictions))
        print("balanced_accuracy",metrics.balanced_accuracy_score(y_test,MultiNB_predictions))
        print(confusion_matrix(y_test,MultiNB_predictions))
        print(classification_report(y_test,MultiNB_predictions))

"""## LogisticRegression c GridSearchCV"""

clf_LR = LogisticRegressionCV(cv=5, random_state=0)
clf_LR.fit(X_train, y_train)
clf_LR_predictions = clf_LR.predict(X_test)
print("accuracy",metrics.accuracy_score(y_test,clf_LR_predictions))
print("balanced_accuracy",metrics.balanced_accuracy_score(y_test,clf_LR_predictions))
print(confusion_matrix(y_test,clf_LR_predictions))
print(classification_report(y_test,clf_LR_predictions))

import pickle
filename = 'finalized_model.sav'
pickle.dump(clf_LR, open(filename, 'wb'))
# load the model from disk
loaded_model = pickle.load(open(filename, 'rb'))
result = loaded_model.score(X_test, Y_test)
print(result)
```

Приложение В
(обязательное)

Листинг программного модуля генерации рекомендаций

```
import numpy as np
import pandas as pd
import re
from math import floor
import scipy.stats as stats

# Загрузка данных "Data_diagnosis_complaints.xlsx

df = pd.read_excel('disgnosis/Data_diagnosis_complaints_with_anamnesis.xlsx')

"""## Обработка данных Замена полного кода МКВ на короткий шифр (пример "I11.9") """

def mkb_get_code(x):
    if str(x)=='nan':
        return np.nan
    else:
        if re.match("[A-Z]\d{1,2}\.{0,1}\d*",str(x)) is not None:
            return re.match("[A-Z]\d{1}",str(x)).group()
        else:
            return np.nan

df['mkb'] = df['mkb_code'].apply(lambda x: mkb_get_code(x))

df = df[df['mkb'].isna() == False]

df['patient_condition'] = df['anamnesis_disease'] + " " +  df['patient_complaints']

"""Удаляем записи с пропущенными значениями по полям 'recommendations' и 'patient
complaints"""

df = df[df['patient_complaints'].isna() == False]
df = df[df['recommendations'].isna() == False]
df = df[df['anamnesis_disease'].isna() == False]

"""Оставляет только поля "mkb","patient complaints",'recommendations','ext_diagnosis"""

df = df[["mkb","patient_condition",'recommendations','ext_diagnosis']]
df.shape

"""## Базовые модели"""

# Commented out IPython magic to ensure Python compatibility.
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

# Standard Data Science Libraries
import pickle
import math
import pandas as pd
import numpy as np
from numpy import array

# Neural Net Preprocessing
from sklearn.feature_extraction.text import CountVectorizer
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.preprocessing.sequence import pad_sequences
# Neural Net Layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Embedding

# Neural Net Training
from tensorflow.keras.models import load_model
from tensorflow.keras.callbacks import ModelCheckpoint
from keras.callbacks import EarlyStopping

from pickle import load

"""Перекодирование текста"""

# encode для label
encode_dict = {}

def encode_cat(x):
    if x not in encode_dict.keys():
        encode_dict[x]=len(encode_dict)
    return encode_dict[x]

df_copy['label'] = df_copy['mkb'].apply(lambda x: encode_cat(x))

# Lower casing
def lower(text):
    low_text= text.lower()
    return low_text

df_copy['text_new']=df_copy['patient_condition'].apply(lambda x:lower(x))
df_copy['text_new_rec']=df_copy['recommendations'].apply(lambda x:lower(x))
```

```
#Remove extra white space left while removing stuff
def remove_space(text):
    space_remove = re.sub(r"\s+"," ",text).strip()
    return space_remove

df_copy['text_new']=df_copy['text_new'].apply(lambda x:remove_space(x))
df_copy['text_new_rec']=df_copy['text_new_rec'].apply(lambda x:remove_space(x))

df_copy['patient complaints']=df_copy['text_new']
df_copy['recommendations']=df_copy['text_new_rec']

df_copy = df_copy.drop(columns=['text_new','text_new_rec'])

"""## Переключаемся на cuda"""

import torch, os, re, pandas as pd, json
from sklearn.model_selection import train_test_split
from transformers import DataCollatorForLanguageModeling, GPT2Tokenizer,
GPT2LMHeadModel, Trainer, TrainingArguments, AutoConfig
from datasets import Dataset

if torch.cuda.is_available():
    dev = "cuda:0"
else:
    dev = "cpu"
device = torch.device(dev)

"""## Рус токенайзер и GPT 2"""

from transformers import GPT2LMHeadModel, GPT2Tokenizer

def load_tokenizer_and_model(model_name_or_path):
    return GPT2Tokenizer.from_pretrained(model_name_or_path),
GPT2LMHeadModel.from_pretrained(model_name_or_path).cuda()

def generate(
    model, tok, text,
    do_sample=True, max_length=50, repetition_penalty=5.0,
    top_k=5, top_p=0.95, temperature=1,
    num_beams=None,
    no_repeat_ngram_size=3
    ):
    input_ids = tok.encode(text, return_tensors="pt").cuda()

    out = model.generate(
      input_ids.cuda(),
      max_length=max_length,
      repetition_penalty=repetition_penalty,
```

```
      do_sample=do_sample,
      top_k=top_k, top_p=top_p, temperature=temperature,
      num_beams=num_beams, no_repeat_ngram_size=no_repeat_ngram_size
      )
    return list(map(tok.decode, out))

"""## RuGPTLarge"""

tok, model = load_tokenizer_and_model("ai-forever/rugpt3large_based_on_gpt2") #("sberbank-ai/rugpt2large")

base_model = model
base_tokenizer = tok

base_model.num_parameters

model_name_or_path = "ai-forever/rugpt3large_based_on_gpt2" #"sberbank-ai/rugpt2large"

# special tokens are defined
bos = '[endoftext]'
eos = '[EOS]'
body = '[body]'
additional_special_tokens = [body]

special_tokens_dict = {'eos_token': eos, 'bos_token': bos, 'pad_token': '<pad>',
                       'sep_token': body}

#  'additional_special_tokens':additional_special_tokens}

# the new token is added to the tokenizer
num_added_toks = base_tokenizer.add_special_tokens(special_tokens_dict)

num_added_toks

# model configuration to which we add the special tokens
config = AutoConfig.from_pretrained('gpt2',
                                    bos_token_id=base_tokenizer.bos_token_id,
                                    eos_token_id=base_tokenizer.eos_token_id,
                                    pad_token_id=base_tokenizer.pad_token_id,
                                    sep_token_id=base_tokenizer.sep_token_id,
                                    output_hidden_states=False)

# we load the pre-trained model with custom settings
base_model_new =  GPT2LMHeadModel.from_pretrained(model_name_or_path, config=config, ignore_mismatched_sizes=True).cuda()

# model embeding resizing
base_model_new.resize_token_embeddings(len(base_tokenizer))
```

```
def process_dataset(df, input_text, output_text):
    # Remove rows with empty or null title or content
    titulo_vacio = (df[input_text].str.len() == 0) | df[input_text].isna()
    contenido_vacio = (df[output_text].str.len() == 0) | df[output_text].isna()
    df = df[~titulo_vacio & ~contenido_vacio]

      # Keep the first 100 words from the content
    df[output_text] = df[output_text].str.split(' ').apply(lambda x: ' '.join(x[:100]))

    return df

# Data cleansing
news_df = process_dataset(df_copy, input_text='patient complaints',
output_text='recommendations')

# We add the tokens
prepare_text = lambda x: ' '.join([bos, x['patient complaints'], body, x['recommendations'], eos])
news_df['text'] = news_df.apply(prepare_text, axis=1)

# Split in train and test
df_train_news, df_val_news = train_test_split(news_df, train_size = 0.9, random_state = 77)

# we load the datasets from pandas df
train_dataset = Dataset.from_pandas(df_train_news[['text']])
val_dataset = Dataset.from_pandas(df_val_news[['text']])

# tokenization

def tokenize_function(examples):
       return base_tokenizer(examples['text'], padding=True)

tokenized_train_dataset = train_dataset.map(
    tokenize_function,
    batched=True,
    num_proc=1
)

tokenized_val_dataset = val_dataset.map(
    tokenize_function,
    batched=True,
    num_proc=1
)

model_articles_path = './model_J0 '

training_args = TrainingArguments(
    output_dir=model_articles_path,          # output directory
```

```
    num_train_epochs=100,              # total # of training epochs
    per_device_train_batch_size=3,  # batch size per device during training
    per_device_eval_batch_size=1,   # batch size for evaluation
    warmup_steps=200,                # number of warmup steps for learning rate scheduler
    weight_decay=0.01,               # strength of weight decay
    logging_dir=model_articles_path,           # directory for storing logs
    prediction_loss_only=True,
    save_steps=10000
)

data_collator = DataCollatorForLanguageModeling(
        tokenizer=base_tokenizer,
        mlm=False
    )

trainer = Trainer(
    model=base_model_new,
    args=training_args,                 # training arguments, defined above
    data_collator=data_collator,
    train_dataset=tokenized_train_dataset,         # training dataset
    eval_dataset=tokenized_val_dataset,           # evaluation dataset

)

trainer.train()

trainer.save_model("model_GPT3 ru 14_06 J0 ver1")
base_tokenizer.save_pretrained("model_GPT3 ru 14_06 J0 ver1")

def generate_n_text_samples(model, tokenizer, input_text, device, n_samples = 5):
    text_ids = tokenizer.encode(input_text, return_tensors = 'pt')
    text_ids = text_ids.to(device)
    model = model.to(device)

    generated_text_samples = model.generate(
        text_ids,
        max_length= 100,
        num_return_sequences= n_samples,
        no_repeat_ngram_size= 2,
        repetition_penalty= 1.5,
        top_p= 0.92,
        temperature= .85,
        do_sample= True,
        top_k= 125,
        early_stopping= True
    )
    gen_text = []
    for t in generated_text_samples:
```

```
        text = tokenizer.decode(t, skip_special_tokens=True)
        gen_text.append(text)

        return gen_text


prompt = ' '.join([bos, df_copy['patient complaints'].iloc[200] ,  body])

texts = generate_n_text_samples(model,tok, prompt, device, n_samples =5)
```

Приложение Г
(обязательное)

Акты о внедрении результатов диссертации

УТВЕРЖДАЮ
Главный врач ГАУЗ «ООКБ им. В.И. Войнова», к.м.н.
_____ Редюков А.В.
«___» _____ 2023 г.

АКТ
о внедрении результатов диссертации
Гришиной Любови Сергеевны

Настоящий акт составлен о том, что научно-технические результаты диссертационной работы Л.С. Гришиной «Методы и алгоритмы интеллектуальной поддержки принятия решений в медицинской практике на основе обработки естественных языков» внедрены и используются в деятельности организационно-методического отдела ГАУЗ «Оренбургской областной клинической больницы имени В.И. Войнова».

Использование разработанной системы поддержки принятия решений на основе методов машинного обучения обеспечивает автоматизацию процессов анализа документации МИС, прогнозирования укрупненных групп заболеваний пациентов с ССЗ по МКБ-10 и динамического формирования персонифицированных шаблонов листа назначений и рекомендаций к лечению.

Главный внештатный кардиолог
Министерства здравоохранения Оренбургской области Ю.В. Золотова

УТВЕРЖДАЮ

Главный врач ГАУЗ «ББСМП
им. академика Н.А. Семашко»

_____ С.Ю. Кадочкин

«____» _____ 2023 г.

АКТ ВНЕДРЕНИЯ

Комплекса программных средств интеллектуальной поддержки принятия решений при диагностике и лечении заболеваний

Комплекс программных средств для интеллектуальной поддержки принятия врачебных решений при диагностике и лечении заболеваний, разработанный Л.С. Гришиной, внедрен в 2023 году в государственном автономном учреждении здравоохранения «Бузулукская больница скорой медицинской помощи имени академика Н.А. Семашко» в режиме опытной эксплуатации с целью автоматизации процессов анализа и заполнения документов и сокращения времени оказания медицинских услуг.

Данный комплекс оценивается как эффективный инструмент для прогнозирования укрупненной группы заболевания и выставления соответствующего кода по международной классификации болезней МКБ-10. Применение модуля генерации персонифицированных шаблонов листа назначений и рекомендаций обеспечивает автоматизацию процессов формирования заключений и снижает временные затраты в среднем на 7%.

Согласовано:

Заместитель главного врача
по лечебной работе О.С. Суркова

Заведующий отделением
анестезиологии и реанимации Д.А. Краснояров

Начальник отдела ИТ Е.А. Бондаренко

УТВЕРЖДАЮ

Ректор ФГБОУ ВО ОрГМУ Минздрава России, профессор, д.м.н.

Мирошниченко И.В.

«» 2023 г.

АКТ ВНЕДРЕНИЯ
результатов диссертационной работы Гришиной Л.С.

Результаты диссертационной работы аспиранта Гришиной Любови Сергеевны по теме «Методы и алгоритмы интеллектуальной поддержки принятия решений в медицинской практике на основе обработки естественных языков», представленные:

- концепцией автоматического извлечения информации из разношаблонных документов медицинской информационной системы;

- методом прогнозирования укрупненных групп заболеваний пациентов по МКБ-10 на основе слабоструктурированных текстовых данных электронных медицинских карт пациентов;

- методом автоматической генерации персонифицированных шаблонов листа назначений и рекомендаций к лечению для поддержки принятия врачебных решений;

используются в учебном процессе ФГБОУ ВО «Оренбургского государственного медицинского университета» на кафедре «Общественного здоровья и здравоохранения №1».

Представленные научные исследования можно охарактеризовать как научно обоснованные разработки, имеющие преимущества в сравнении с другими методами автоматизации процесса заполнения протокола оказания медицинских услуг и обеспечивающие решение важных прикладных задач диагностики и профилактики заболеваний, которые можно эффективно использовать в учебном процессе.

Заведующий кафедрой общественного здоровья и здравоохранения №1, профессор, д. м. н. Е.Л. Борщук

Приложение Д
(обязательное)

Свидетельство о регистрации программы

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2023610238

Модуль исследования эффективности графового представления данных для модели прогнозирования ССЗ на основе неструктурированных клинических текстов

Правообладатель: ***федеральное государственное бюджетное образовательное учреждение высшего образования «Оренбургский государственный университет» (RU)***

Авторы: ***Варенникова Юлия Викторовна (RU), Болодурина Ирина Павловна (RU), Шухман Александр Евгеньевич (RU), Жигалов Артур Юрьевич (RU), Гришина Любовь Сергеевна (RU), Легашёв Леонид Вячеславович (RU)***

Заявка № **2022685667**

Дата поступления **22 декабря 2022 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***09 января 2023 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат 68be80073[illegible]40f0a94eddbd24145d5c7

Владелец Зубов Юрий Сергеевич

Действителен с 26.05.2022 по 26.05.2025

Ю.С. Зубов