TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

---

Инженерная школа информационных технологий и робототехники
Направление подготовки 09.04.04 Программная инженерия
Отделение школы (НОЦ) Информационных технологий

# МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

| Тема работы |
|---|
| **Алгоритмы и программное обеспечение извлечения значимых предикторов из электронных медицинских записей** |

УДК 004.421:004.415.2:61

Студент

| Группа | ФИО | Подпись | Дата |
|---|---|---|---|
| 8ПМ9И | Котюбеев Роман Радиевич | | |

Руководитель ВКР

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ОИТ ИШИТР | Аксёнов С.В. | к.т.н. | | |

## КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ОСГН ШБИП | Гончарова Н.А. | к.э.н. | | |

По разделу «Социальная ответственность»

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ООД ШБИП | Антоневич О.А. | к.б.н. | | |

## ДОПУСТИТЬ К ЗАЩИТЕ:

| Руководитель ООП | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ОИТ ИШИТР | Савельев А.О. | к.т.н. | | |

Томск – 2021 г.

**ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП**
по направлению 09.04.04 «Программная инженерия»

| Код компетенции | Наименование компетенции |
|---|---|
| **Универсальные компетенции** | |
| УК(У)-1 | Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий |
| УК(У)-2 | Способен управлять проектом на всех этапах его жизненного цикла |
| УК(У)-3 | Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели |
| УК(У)-4 | Способен применять современные коммуникативные технологии, в том числе на иностранном (-ых) языке (-ах), для академического и профессионального взаимодействия |
| УК(У)-5 | Способен анализировать и учитывать разнообразие культур в процессе межкультурного взаимодействия |
| УК(У)-6 | Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки |
| **Общепрофессиональные компетенции** | |
| ОПК(У)-1 | Способен самостоятельно приобретать, развивать и применять математические, естественно-научные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте |
| ОПК(У)-2 | Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач |
| ОПК(У)-3 | Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями |
| ОПК(У)-4 | Способен применять на практике новые научные принципы и методы исследований |
| ОПК(У)-5 | Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем |
| ОПК(У)-6 | Способен самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности |
| ОПК(У)-7 | Способен применять при решении профессиональных задач методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе, в глобальных компьютерных сетях |
| ОПК(У)-8 | Способен осуществлять эффективное управление разработкой программных средств и проектов |
| **Профессиональные компетенции** | |
| ПК(У)-1 | Способен к созданию вариантов архитектуры программного средства |

| | |
|---|---|
| ПК(У)-2 | Способен разрабатывать и администрировать системы управления базам данных |
| ПК(У)-3 | Способен управлять процессами и проектами по созданию (модификации) информационных ресурсов |
| ПК(У)-4 | Способен проектировать и организовывать учебный процесс по образовательным программам с использованием современных образовательных технологий |
| ПК(У)-5 | Способен осуществлять руководство разработкой комплексных проектов на всех стадиях и этапах выполнения работ |

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Инженерная школа информационных технологий и робототехники
Направление подготовки (специальность) 09.04.04 Программная инженерия
Отделение школы (НОЦ) Информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП
_____ _____ Савельев А.О.
(подпись) (дата) (Ф.И.О.)

**ЗАДАНИЕ**
**на выполнение выпускной квалификационной работы**

В форме:

| Магистерской диссертации |
|---|

Студенту:

| **Группа** | **ФИО** |
|---|---|
| 8ПМ9И | Котюбееву Роману Радиевичу |

Тема работы:

| Алгоритмы и программное обеспечение извлечения значимых предикторов из электронных медицинских записей | |
|---|---|
| Утверждена приказом директора (дата, номер) | № 40-5/с от 09.02.2021 |

| Срок сдачи студентом выполненной работы: | 15.06.2021 |
|---|---|

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

| | |
|---|---|
| **Исходные данные к работе** | Объектом проектирования и разработки является алгоритм извлечения значимых предикторов из электронных записей первичного осмотра врачом в стационаре |
| **Перечень подлежащих исследованию, проектированию и разработке вопросов** | – Анализ предметной области;<br>– обзор методов извлечения информации;<br>– подготовка исходных данных;<br>– проектирование и программная реализация алгоритма извлечения предикторов;<br>– тестирование алгоритма;<br>– разработка веб-сервиса;<br>– выполнение раздела финансовый менеджмент;<br>– выполнение раздела социальная ответственность;<br>– выполнение раздела на английском языке. |
| **Перечень графического материала** | – UML-диаграмма;<br>– диаграмма потока данных;<br>– скриншоты веб-сервиса; |

| | |
|---|---|
| | – диаграмма Ганта. |

| **Консультанты по разделам выпускной квалификационной работы** | |
|---|---|
| **Раздел** | **Консультант** |
| Финансовый менеджмент, ресурсоэффективность и ресурсосбережение | Доцент ОСГН ШБИП, к.э.н., Гончарова Н.А. |
| Социальная ответственность | Доцент ООД ШБИП, к.б.н., Антоневич О. А. |
| Английский язык | Доцент ОИЯ, к.пед.н., Сидоренко Т.В. |
| **Названия разделов, которые должны быть написаны на русском и иностранном языках:** | |
| Раздел 1 Natural language processing | |

| | |
|---|---|
| **Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику** | 01.03.2021 |

**Задание выдал руководитель:**

| **Должность** | **ФИО** | **Ученая степень, звание** | **Подпись** | **Дата** |
|---|---|---|---|---|
| Доцент ОИТ ИШИТР | Аксёнов С.В. | к.т.н. | | |

**Задание принял к исполнению студент:**

| **Группа** | **ФИО** | **Подпись** | **Дата** |
|---|---|---|---|
| 8ПМ9И | Котюбеев Роман Радиевич | | |

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Инженерная школа информационных технологий и робототехники
Направление подготовки (специальность) 09.04.04 Программная инженерия
Уровень образования магистратура
Отделение школы (НОЦ) Информационных технологий
Период выполнения весенний семестр 2020 /2021 учебного года

Форма представления работы:

| Магистерская диссертация |
|---|

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН**
**выполнения выпускной квалификационной работы**

| Срок сдачи студентом выполненной работы: | 15.06.2021 |
|---|---|

| Дата контроля | Название раздела (модуля) / вид работы (исследования) | Максимальный балл раздела (модуля) |
|---|---|---|
| 10.03.2021 | Анализ предметной области | 5 |
| 15.03.2021 | Обзор методов извлечения информации | 5 |
| 20.03.2021 | Подготовка исходных данных | 10 |
| 20.04.2021 | Проектирование и программная реализация алгоритма извлечения предикторов | 30 |
| 10.05.2021 | Тестирование алгоритма | 10 |
| 22.05.2021 | Разработка веб-сервиса | 10 |
| 01.06.2021 | Финансовый менеджмент, ресурсоэффективность и ресурсосбережение | 10 |
| 01.06.2021 | Социальная ответственность | 10 |
| 01.06.2021 | Английский язык | 10 |

**СОСТАВИЛ:**
**Руководитель ВКР**

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ОИТ ИШИТР | Аксёнов С.В. | к.т.н. | | |

**СОГЛАСОВАНО:**
**Руководитель ООП**

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ОИТ ИШИТР | Савельев А. О. | к.т.н. | | |

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА**
**«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

| Группа | ФИО |
|---|---|
| 8ПМ9И | Котюбееву Роману Радиевичу |

| Школа | ИШИТР | Отделение школы (НОЦ) | ОИТ |
|---|---|---|---|
| **Уровень образования** | Магистратура | **Направление/специальность** | 09.04.04 Программная инженерия |

| **Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:** | |
|---|---|
| 1. *Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих* | 1. Оклад инженера – 9489;<br>2. Оклад научного руководителя – 35120 |
| 2. *Нормы и нормативы расходования ресурсов* | Месячная норма амортизации – 2,8% |
| 3. *Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования* | 1. Ставки налоговых отчислений во внебюджетные фонды (ст. 426 НК РФ) – 30%.<br>2. Районный коэффициент по г. Томску (ст. 426 НК РФ, Постановление Правительства РФ от 13.05.92. No309) – 1,3 |
| **Перечень вопросов, подлежащих исследованию, проектированию и разработке:** | |
| 1. *Оценка коммерческого и инновационного потенциала НТИ* | 1. Анализ потенциальных потребителей<br>2. Анализ конкурентоспособности;<br>3. SWOT-анализ. |
| 2. *Разработка устава научно-технического проекта* | Формирование цели, задач и ожидаемых результатов проекта |
| 3. *Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок* | 1. Планирование структуры работ проекта;<br>2. Определение трудоемкости выполнения работ;<br>3. Формирование бюджета;<br>4. Анализ рисков проекта |
| 4. *Определение ресурсной, финансовой, экономической эффективности* | Расчет показателя финансовой эффективности. |
| **Перечень графического материала** *(с точным указанием обязательных чертежей):* | |
| 1. *«Портрет» потребителя результатов НТИ*<br>2. *Сегментирование рынка*<br>3. *Оценка конкурентоспособности технических решений*<br>4. *Матрица SWOT*<br>5. *График проведения и бюджет НТИ*<br>6. *Оценка ресурсной, финансовой и экономической эффективности НТИ*<br>7. *Потенциальные риски* | |

| **Дата выдачи задания для раздела по линейному графику** | 22.02.2021 |
|---|---|

**Задание выдал консультант:**

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ОСГН ШБИП | Гончарова Н.А. | к.э.н | | |

**Задание принял к исполнению студент:**

| Группа | ФИО | Подпись | Дата |
|---|---|---|---|
| 8ПМ9И | Котюбеев Роман Радиевич | | |

# ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

| Группа | ФИО |
|---|---|
| 8ПМ9И | Котюбееву Роману Радиевичу |

| Школа | ИШИТР | Отделение (НОЦ) | |
|---|---|---|---|
| Уровень образования | Магистратура | Направление/специальность | 09.04.04 Программная инженерия |

Тема ВКР:

| Алгоритмы и программное обеспечение для извлечения значимых предикторов из электронных медицинских записей | |
|---|---|
| **Исходные данные к разделу «Социальная ответственность»:** | |
| 1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения | Объект исследования: алгоритмы извлечения значимых предикторов из электронных медицинских записей<br>Рабочая зона разработчика включает в себя: персональный компьютер (ПК), комнату с окнами, рабочий стол и стул |
| Перечень вопросов, подлежащих исследованию, проектированию и разработке: | |
| **1. Правовые и организационные вопросы обеспечения безопасности:**<br>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;<br>– организационные мероприятия при компоновке рабочей зоны. | – ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования.<br>– ГОСТ 21889–76* «Кресло человека-оператора. Общие эргономические требования».<br>– Трудовой кодекс РФ |
| **2. Производственная безопасность:**<br>2.1. Анализ выявленных вредных и опасных факторов<br>2.2. Обоснование мероприятий по снижению воздействия | Вредные факторы:<br>– недостаточная освещенность рабочей зоны;<br>– отклонение показателей микроклимата;<br>– повышенный уровень шума;<br>– перенапряжение зрительных анализаторов;<br>– Статические перегрузки, связанные с рабочей позой.<br>Опасные факторы:<br>– Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека |
| **3. Экологическая безопасность:** | – Атмосфера и гидросфера: отсутствует;<br>– Литосфера: при утилизации используемого ПК и утилизации люминесцентных ламп |
| **4. Безопасность в чрезвычайных ситуациях:** | Возможная ЧС:<br>– пожар; |

| Дата выдачи задания для раздела по линейному графику | 01.03.2021 |
|---|---|

**Задание выдал консультант:**

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ООД ШБИП | Антоневич О.А. | к.б.н. | | |

**Задание принял к исполнению студент:**

| Группа | ФИО | Подпись | Дата |
|---|---|---|---|
| 8ПМ9И | Котюбеев Роман Радиевич | | |

## Реферат

Выпускная квалификационная работа выполнена на 109 с., содержит 12 рис., 20 табл., 63 источников, 3 прил.

Ключевые слова: МЕДИЦИНСКИЕ ТЕКСТЫ, ИЗВЛЕЧЕНИЕ ИНФОРМАЦИИ, ПЕРВИЧНЫЙ ОСМОТР, ПАРСЕР YARGY, КОНТЕКСТНО-СВОБОДНАЯ ГРАММАТИКА, ВЕБ-СЕРВИС.

Целью работы является повышение эффективности анализа электронных медицинских записей с помощью разработки инструментов поиска и идентификации значимых предикторов, влияющих на оценку состояния пациента. Объектами исследования являются электронные записи первичного осмотра врачом в стационаре.

Описана проблематика обработки медицинских текстов в рамках задачи извлечения информации, проведен обзор методов обработки текстовых данных, дано сравнение основных библиотек, которые реализуют контекстно-свободную грамматику для русского языка. В качестве основного инструмента при проектировании алгоритма извлечения предикторов была выбрана библиотека Yargy. Приведена структура записи первичного осмотра, описан алгоритм извлечения предикторов с учетом этой структуры с помощью парсера Yargy. Результаты тестирования показали, что алгоритм эффективен, поскольку 44 из 49 предикторов были определены с высокой точностью. Также был разработан ориентированный на врачей веб-сервис, который отображает структурированный список извлеченных предикторов.

Полученные инструменты позволят выделять предикторы состояния пациента из текстов осмотров и дневников, написанных на естественном языке. Также результаты работы могут быть востребованы научными центрами, занимающимися применением машинного обучения в области медицины, так как спроектированный алгоритм позволяют собрать необходимые для обучения данные. Практическая значимость заключается в возможности использования веб-сервиса внутри одного или нескольких медицинских учреждений. Научной новизной является предложенный подход к извлечению фактов из медицинских текстов на русском языке.

## Обозначения и сокращения

В настоящей работе применены следующие термины и сокращения с соответствующими обозначениями:

Предиктор — медицинский факт, характеризующий состояние пациента;

NLP — natural language processing;

CL — computational linguistics;

СибГМУ — Сибирский государственный медицинский университет;

UMLS — unified medical language system;

cTakes — clinical text analysis and knowledge extraction system;

СИМП — система извлечения медицинских предикторов.

# Оглавление

# Введение

Врачи в наше время сталкиваются с напряженной умственной нагрузкой. Помимо приемов и осмотров пациентов им приходится иметь дело с электронными медицинскими записями, из которых нужно находить необходимые факты о том или ином пациенте. Кроме того, с 1 февраля 2021 года вступил в силу приказ Минздрава здравоохранения России №947н о полном переходе медицинской документации в электронные формы [1], что еще больше увеличит нагрузку на врачей при таком формате.

Электронная запись первичного осмотр хоть и и разбита на семантические блоки, но тем не менее набирается сплошным текстом. А поиск конкретного факта о пациенте требует беглого просмотра всего документа. Эти факты играют ключевую роль при назначении лечения, а также могут послужить обучающими данными при создании различных моделей машинного обучения для оценки эффективности лечения и прогноза выздоровления.

Данная работа посвящена разработке алгоритма по извлечению значимых предикторов (фактов) из электронных медицинских записей, а также интеграции данного алгоритма в веб-сервис. В отличие до других систем, предложенный алгоритм работает с текстами на русском языке и имеет более удобное обращение к нему, поскольку требует от сотрудника медицинского учреждения только наличие доступа к сети Интернет.

В разделе 1 рассмотрено применение области обработки текстов, написанных на естественном языке (Natural language processing), задачи и методы этой области, специфика обработки медицинских текстов, а также приведен обзор двух библиотек для извлечения фактов из текстов.

В разделе 2 описано проектирование алгоритма извлечения предикторов из электронных медицинских записей на основе контекстно-свободной грамматики и приведены результаты работы алгоритма.

В разделе 3 представлена разработка веб-сервиса, в который интегрирован спроектированный алгоритм извлечения предикторов.

В разделе 4 изложены вопросы, связанные с финансовым менеджментом и ресурсоэффективностью разработки.

В разделе 5 рассмотрены экологическая безопасность и техника безопасности работников на рабочем месте.

Практическая ценность алгоритма и веб-сервиса заключается в возможности его использования внутри одного или нескольких медицинских учреждений. Научную новизну имеют предложенный подход к извлечению значимых фактов из медицинских текстов.

Результаты работы опубликованы в ряде сборников, доложены на четырех международных конференциях, а также были отмечены на международном хакатоне от Microsoft (приложение В).

# 1 Обработка естественного языка

Обработка текстов, написанных на естественном языке, (от англ. natural language processing, NLP), а также вычислительная лингвистика (от англ. computational linguistics, CL) — две сферы вычислительных методов изучения и анализа естественных языка, о том, каким образом создаются языки, как они осваиваются и как различные языки связаны друг с другом [2]?

Сфера обработки текстов, написанных на естественном языке, с другой стороны, стремится решить практические задачи, связанные с языками, в том числе и извлечение необходимой информации. Различные дисциплины, такие как математика, лингвистика, компьютерные науки, машинное обучение, используются при решения задач обработки естественного языка [3].

К решаемым задачам NLP также относятся: машинный перевод, распознавание речи, реферирование, классификация, кластеризация, генерация, анализ тональности текстов. В этой работе будет рассматриваться задача **извлечения информации из текстов**. Основным приложением этой задачи является автоматическое выделение значимых для человека данных, как правило из большого набора текстов, и преобразование их в структурированную форму, что облегчает их последующую обработку и анализ. Системы с подобным приложением могут быть использованы во многие сферах деятельности, в частности, в медицине.

## 1.1 Обработка медицинских текстов

### 1.1.1 Специфика задачи извлечения информации

Задача извлечения информации схожа с традиционной задачей поиска релевантной информации. Различие заключается в том, что извлечение информации подразумевает вывод ее в структурированном виде (например, в формате JSON или XML), что делает информацию более подходящей для обработки [4]. Сложность данной задачи состоит в том, что целевые тексты никак не размечены, поэтому исследователю приходится прибегать к эври-

стическим и лингвистическим методам. Последующая извлеченная и структурированная информация может быть передана аналитикам данных или специалистам, которым она требуется.

Следует уточнить основные термины. Под **неструктурированным текстом** подразумевается любой текст, написанный человеком, и который не может быть интерпретирован компьютером. Как правило, компьютер только хранит и выдает информацию, но он не понимает, что именно в ней представлено. С другой стороны, **структурированный текст** — это текст, который определен, как с семантической, так и с вычислительной точки зрения [5]. В этом случае компьютер может интерпретировать единицы структурированного текста, а они в свою очередь могут быть применены к текстам другой направленности (принцип обобщения). Использование термина «извлечение» подразумевает, что необходимая семантическая информация явно присутствует в лингвистической организации текста, т.е. она легко представляется в виде лексических элементах (словах, словосочетаниях), грамматических конструкциях (фразах, предложениях, временных выражениях и т.д.) и структурной рубрикации (предложениях, абзацах, главах и т.д.) исходного текста.

Извлечение информации отличается от *реферирования*, при котором обычно из текстов извлекаются целые предложения, являющихся его резюме (кратким содержанием). Однако извлечение информации может быть первым шагом в резюмирования текста, в котором итоговое предложение далее сокращается до строки релевантных фраз, подобных заголовку газеты [6].

Во всех операциях извлечение информации часто является обязательным этапом предварительной обработки данных. Например, данные, извлеченные из полицейских отчетов, могут использоваться в интеллектуальном анализе для выявления общих тенденций преступности или в системе принятия решений на основе конкретного случая, которая будет пытаться предсказывать местоположение следующего преступления. Данные из медицинских отчетов могут применяться для выявления диагноза, сбора данных о пациенте, системах принятия решений.

### 1.1.2 Подзадачи извлечения информации

Извлечение информации само по себе является большой задачей, поэтому ее разбивают на подзадачи. К типовым подзадачам извлечения информация относятся:

- извлечение именованных сущностей (Named entity recognition, NER),
- разрешение кореференции (Coreference resolution),
- извлечение отношений (Relationship extraction),
- извлечение атрибутов, фактов и событий (Fact Extraction).

Подзадача *извлечения именованных сущностей* направлена на поиск и классификацию именованных сущностей, упомянутых в неструктурированном тексте, по заранее определенным категориям, таким как имена людей, организации, местоположения, медицинские коды, выражения времени, количества, денежные значения, проценты и т.д. Термин «именованная сущность» ограничивает задачу теми сущностями, для которых одно или множество слов последовательно обозначают некоторый референт. Например, в предложении «автомобильная компания, созданная Генри Фордом в 1903 году», именованной сущностью может быть имя «Форд» или «компания Форд (Ford Motor Company)»; хотя это слово может иметь и другие смыслы, исследователи сами определяют, какой он будет иметь референт [7]. Результатом извлечения именованных сущностей являются ответы на вопросы: что произошло, кто это сделал, когда, где, как и почему.

Подзадача *разрешения кореференций* связанна с нахождением всех цепочек упоминаний, которые ссылаются на одну и ту же сущность в тексте. Допустим, в предложении упоминается какая-то сущность, например, существительное «книга», а в следующих предложениях она упоминается в виде местоимений «она», «её» и «ей», тогда целью разрешения кореференций является установление всех упоминаний этой книги. С ее помощью можно ре-

шить множество NLP-задач, таких как автореферирование, вопрос-ответные системы, чат-боты и другие подзадачи извлечения информации [8].

Подзадача *извлечения отношений* заключается в поиске именованных сущностей, входящие в одну семантическую группу. Например, люди, организации, места, могут быть разбиты на ряд заранее определенных семантических категорий: «женат», «живет в», «работает в» и др. В биомедицине приложением данной подзадачи является объединение связанных болезней в группы из неструктурированного текста [9].

Подзадача *извлечения атрибутов, фактов и событий* направлена на поиск и классификацию различных фактов: события, мнения, отзывы, контактные данные, новости, объявления и т.д. Задача по извлечению фактов представляет наибольший интерес при анализе текстовых данных и является конечным по отношению к вышеупомянутым [5]. На сегодняшний день не существует универсального подхода к ее решению. Вместо этого рассматриваются частные задачи в зависимости от предметной области.

Также можно выделить следующие специфичные подзадачи извлечения информации: извлечение таблиц [10], извлечение содержимое таблиц [11], извлечение терминологии [12].

В этой работе будет рассматриваться подзадача извлечения атрибутов, фактов и событий, поскольку объектом исследования являются электронные медицинские записи, содержащие различные предикторы, которые находятся в тексте в неструктурированном виде.

### 1.1.3 Медицинские тексты и их специфика

Достаточный объем медицинских данных хранится в виде сплошного текста. В электронных медицинских картах медицинских учреждений накапливаются отчеты, написанные на естественном языке, включая анамнезы, выписки, истории болезней, различные результаты диагностических исследований и многие другие записи. Медицинские данные — большой источник информации, который по-прежнему трудно использовать из-за его неструктурированности [13]. Преобразование его в вычислимую форму может принести пользу биомедицинским исследованиям, эффективному ведению истории бо-

лезни пациентов и, в конечном итоге, поможет улучшить здравоохранение в целом.

Медицинские данные, такие как осмотры врачей, создают множество проблем для любого инструмента извлечения информации. Ниже перечислены некоторые из основных проблем, с которыми приходится сталкиваться.

**Нет строго стандарта по структуре записей.** Медицинские документы не имеют фиксированной структуры. Они могут быть разделены на разделы, однако нет чёткой стандартизации в отношении типов разделов, их заголовков или содержания. Они меняются от больницы к больнице, от врача к врачу.

**Медицинский жаргон.** Медицинские записи содержат большое количество медицинских терминов и жаргона. При том, что инструменты NLP, обученные работе с данными немедицинской области, очень плохо работают с медицинскими данными [14].

**Свойственный врачам язык.** Часто используются неполные фразы или неестественно длинные предложения. Кроме того, стиль написания зависит от источника документа [15].

**Аббревиатуры.** В области медицины широко используются аббревиатуры. Одно и то же сокращение может расширяться до различных терминов в зависимости от контекста и намерений автора. Аббревиатуры трудно нормализовать, классифицировать.

**Многозначность и синонимия.** Один медицинский термин может представлять две разные идеи в зависимости от контекста. Например, «воспаление» может относиться к проблеме кожи, проблеме на клеточном уровне и т.д. С другой стороны, «озноб» и «познабливание» имеют тот же смысл.

**Опечатки.** Врачи не являются профессиональными редакторами, у них загруженный график, поэтому ошибки в правописании слов неизбежны.

## 1.2 Методы извлечения фактов из медицинских текстов

Извлечение основных атрибутов из медицинских текстов предполагает извлечение медицинских терминов и фраз из электронных записей. Медицинские термины могут включать названия болезней, процедур, медицинских устройств, названий лекарств, жалобы, осложнения и т.д. Клинические объекты могут состоять из одного или нескольких слов, которые встречаются либо последовательно, либо «через слова» в одном и том же предложении.

К методам извлечения фактов из медицинских текстов относят **статистические моделирование** и **применение различных эвристик и правил** [5].

### 1.2.1 Методы статистического моделирования

Статистические моделирование — надежный и универсальный инструмент для многих задач NLP. Однако оно основывается на машинном обучении, поэтому сильно зависит от маркированных обучающих данных. Тем не менее, если данных достаточно, то такой метод может показать высокие результаты. Скрытые марковские модели, условные случайные поля, метод опорных векторов — это распространенные модели, используемые для извлечения медицинских фактов [14].

Более ранние работы [16] использовали генеративную модель маркировки последовательностей, а именно скрытые марковские модели для обнаружения клинических сущностей из текста.

В работах [17], [18] применяется дискриминационная структура на основе *марковской модели с максимальной энтропией* (от англ. Maximum-entropy Markov model). Этот метод позволяет использовать более широкий набор признаков. Лексические особенности, такие как униграммы, суффиксы, леммы, оказываются важными; лингвистические особенности, такие как часть речи, также играют роль. Кроме того, для создания дополнительных функций используются подходы, основанные на лексике.

Признаки, используемые в марковских моделях, также подходят для *условных случайных полей* (от англ. Conditional Random Fields, CRF). Авторы [19], [20] теоретически и эмпирически доказали, что некоторые ортографических признаки, таких как регистр букв, наличие знаков препинания и они более надежны и точны в задачах маркировки последовательностей. Модель CRF преодолевают проблему смещения меток, с которой сталкиваются марковские модели.

В работах [21], [22] применялась классификация на маркированных последовательностях на основе *метода опорных векторов*. Более того, в работе [21] использовались и совместное использование вышеупомянутых методов.

### 1.2.2 Методы, основанные на эвристических правилах

Методы, основанные на эвристических правилах, можно разделить на два подхода. Первый подход применяет лингвистические принципы для идентификации значимых атрибутов. Второй подход использует семантические признаки, лексемы, словари, а также подходы, основанные на поиске клинических атрибутов по регулярным выражениям.

Подход, применяющий лингвистические принципы, основан на синтаксическом анализе (парсинге) текста. Сначала выполняется синтаксический анализ, а выходные данные обрабатываются с использованием ряда созданных вручную правил для идентификации атрибутов. В частности, атрибуты, как правило, представляют собой существительные, которые встречаются в виде подлежащего. Авторы [23] выполняет ряд шагов по фильтрации текста с использованием заданных правил для выявления клинических атрибутов. Авторы [24] дополнительно выполняет сегментацию предложений. Они также используют двухэтапный подход — когда система, основанная на правилах, сопровождается классификатором, который определяет тип атрибута. Аналогичным образом в работе [25] используется несколько этапов фильтрации с использованием как эвристических правил, так и статистических моделей. В работе [26] авторы внедряют статистическую модель в систему, основанную

на эвристических правилах, используя частоту появления слов и количество совпадений.

С другой стороны, второй подход применяет семантические сети и лексические признаки, свойственные медицинским текстам. Например, система UMLS располагает словарем ключевых слов, ориентирована на различные стандарты и применяет множество внешних ресурсов с целью создания более эффективных и функционально-совместимых биомедицинских информационных систем и услуг, включая электронные медицинские записи [27]. UMLS содержит 60 групп биомедицинских словарей, атрибуты которых имеют около 20 миллионов связей.

Работа [28] посвящена системе DNorm, которая применяет автоматизированный метод определения болезней, упомянутых в медицинском тексте, а также решает задачу нормализации названия болезней. DNorm является целым фреймворком изучения сходства между упоминаниями и названиями понятий непосредственно из тренировочных данных.

Авторы работы [29] разработали систему MedEx, предназначенную для извлечения медицинских атрибутов. Она включает синтаксический анализ, поиск по словарю и регулярные выражения для извлечения атрибутов из медицинского текста.

Авторы [30] разработали систему анализа медицинского текста и извлечения знаний с открытым исходным кодом (лицензия Apache) под названием cTakes. Она предназначена для извлечение информации из электронной медицинской карты в неструктурированном виде. Система решает следующие задачи: определение сущности, например, относится ли словосочетание к симптому, анатомическому термину, диагнозу и другой медицинской категории.

В работе [31] система извлекает сущности из отчетов по вакцинации. Она определяет и классифицирует категорию медицинского термина, например, рвота — к желудочно-кишечному классу, лицо — к анатомии и т.д.

В работе [32] авторы выявляют скрытые зависимости в клинических данных путем извлечения информации из медицинских текстов на русском

языке. После извлечения происходит классификация по тяжести заболевания, течению заболевания, установление связи между заболеваниями и частями тела. Для этой задачи они применяли методы машинного обучения.

Вышеперечисленные системы нельзя с полной уверенностью отнести к подходу, который применяет *только* эвристические правила, поскольку включают себя различные инструменты по работе с медицинскими текстами, в том числе и с применением машинного обучения. Такая комбинация «утяжеляет» систему в целом, делает обработку данных более медленной.

### 1.2.3 Принципы работы методов, основанных на эвристических правилах

Эвристические правила могут быть определены через регулярные выражения, газеттиров, а также на основе комбинированного подхода лингвистических правил, газеттиров и регулярных выражений.

В первом подходе шаблоны для извлечения определяются с использованием такого формального языка, как *регулярные выражения*. Эти шаблоны сопоставляются с исходным входным текстом, а соответствующие данному шаблоны строки извлекаются. Например, если требуется извлечь даты, то шаблоны могут быть заданы в виде `dd.mm.yy` или `dd.mm.yyyy`, где `d` отвечает за день, `m` — за месяц, `y` — за год. В этом случае шаблоны отличаются годом: в первом перечисляются только две последние цифры, а во втором — четыре. Хотя регулярные выражения обеспечивает быстрый и простой поиск, у этого подхода есть ограничения: невозможно перечислить все шаблоны. В примере с датами между числами могут стоять не точки, а тире или даты набираться сначала с месяца и т.д. Более того, числа еще можно как-то «отлавливать», а делать это словами еще труднее. Но несмотря на очевидные ограничения, этот подход широко используется на практике.

Другой подход — хранить возможные значения значимого атрибута в заранее определенном списке, называемый *газеттиром* или *тезаурусом*. Применение газеттира возможно только для тех атрибутов, у которых есть конечное число возможных значений [33]. Хотя этот подход является быстрым и точным, но его ограничения заключаются в подготовке полного и точ-

ного набора всех возможных значений, т.е. ограничения схожи с регулярными выражениями.

Последний подход заключается в применении лингвистических правил исследуемого языка вместе с двумя вышеупомянутыми. Одним из таких лингвистических правил является *контекстно-свободная грамматика* [34]. Для большинства языков не просто любые последовательности символов образуют предложение, но существуют определенные структурные правила, которые определяют допустимые предложения. Эти структурные правила, вместе взятые, образуют грамматику языка. Здесь важно понимать, что, хотя конкретная грамматика однозначно определяет язык, в целом обратное неверно. У данного языка может быть много разных грамматик, описывающих его. Контекстно-свободная грамматика говорит, что слова в предложении могут быть заменены на эквивалентные так, что структура не изменится, а останется ли смысл или нет — не важно. В этой работе применяются правила контекстно-свободной грамматики.

## 1.3 Контекстно-свободная грамматика

Зная структуру предложений, мы можем обобщить её. Так, например, в предложениях «*Пациент №1 жалуется на озноб, кашель*» и «*У пациента №2 следующие жалобы: температура, жар*» заметна некая структура: после слова «пациент» и его номера могут идти глагол «жалуется» или существительное «жалобы», а далее идут перечисления жалоб через запятую. Обобщив данную структуру в виде правила, мы можем найти необходимые атрибуты, например, номер пациента и жалобы. В качестве пояснений строятся деревья разбора. Рисунок 1.1 показывает дерево разбора для первого предложения. Из предложения находятся номера пациента, жалобы и их количество.

В правилах контекстно-свободной грамматики можно указать любой объект. Например, мы можем ориентироваться на идущие подряд прилагательные или другой любой части речи, существительные определённого падежа, числа, знаки препинания и т.д. Более того, мы можем использовать газеттир и вычленять определенные слова, в том числе аббревиатуры. Также

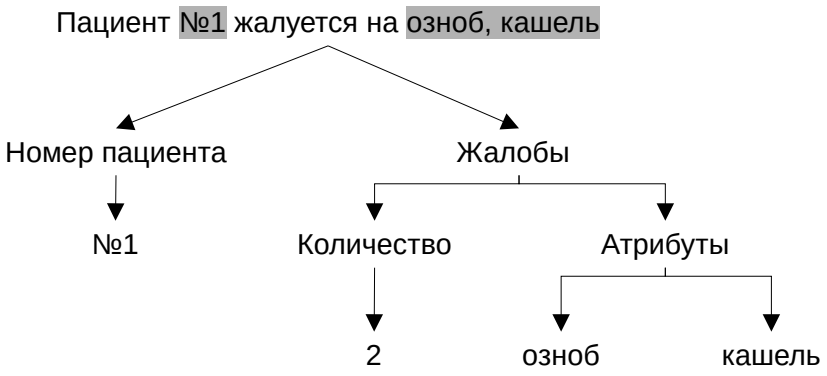


Рисунок 1.1 — Дерево разбора

для более удобного поиска слова нормализуются, чтобы не учитывать все его формы.

Стоит заметить, что если попадется предложение с другой структурой, которую не была учтена в соответствующих правилах, то из него может не получиться извлечь атрибуты. Например, в предложении *«У пациента №3 жар»* нет слов «жалуется» и «жалобы», поэтому жалобы не будут найдены, но будет найден номер пациента, поскольку он стоит после слова «пациент». Такие особенности роднит данный подход с регулярными выражениями, только вместо шаблонов применяются правила, которые учитывают структуру текста, а не отдельные шаблоны каждого входной строки.

### 1.3.1 Парсеры на основе контекстно-свободных грамматик

Правила передаются **парсеру**. Один из самых популярных англоязычных парсеров является парсер библиотеки NLTK [35]. Парсер в нем, скорее, выполняет вспомогательную роль, поскольку данная библиотека снабжена большим количеством функций для обработки текстов. Для русского языка существует несколько реализаций парсеров контекстно-свободных грамматик: Yargy [36] и Томита [37]. Прежде чем преступим к обзору парсеров дадим основные термины в рамках обработки естественного языка.

*Лемма* (иногда *лексема*) — набор всех форм одного слова. Например, «нож», «ножи» и «ножам» входят в одну лексему.

*Нормальная форма слова* — каноническая форма слова. Например, для существительных: форма единственного числа, именительного падежа.

*Граммема* — значение какой-либо грамматической характеристики слова. Например, «единственное число» или «наречие». Множество всех граммем, характеризующих данное слово, образует тег.

*Тег* — набор граммем, которые характеризуют данное слово. Например, для слова «ножам» тегом может быть `'СУЩ,неодуш,муж,мн,дат'`.

### 1.3.2 Томита-парсер

Томита-парсер разработан в компании Яндекс и написан на языке C++. Он доступен в виде бинарного файла с консольным интерфейсом. В открытом доступе отсутствует банк готовых грамматик.

Томита-парсер принимает на вход текст, написанный на естественном языке. На выходе исходный текст преобразуется в набор структурированных данных с помощью составленных исследователем газеттира и грамматик

*Грамматики* — набор неких правил, описывающих граммемы. Грамматика пишется на языке собственном языке Томита. Правило состоит из левой и правой части, которые разделены символом «`->`». Левая часть составляется из *нетерминала*, правая составляется из *терминалов и нетерминалов*.

Терминал — это лексический объект, который имеет конкретное неизменяемое значение. Например, терминалами могут быть леммы, граммемы, тэги, знаки пунктуации, специальные символы. Множество терминалов образует алфавит Томиты. Нетерминалы строятся из терминалов, т.е. это могут быть слова и словосочетания. Примером может служить следующая конструкция:

```
NounPhrase -> Adj Noun;
```

Здесь нетерминал `NounPhrase` состоит из двух терминалов `Adj` (прилагательное) и `Noun` (существительное). На основании этой грамматики можно

извлечь из текста все словосочетания, состоящие из прилагательного и существительного в любых формах и обязательно в этом порядке.

В грамматиках должно присутствовать *корневое правило*, т.е. откуда начинается разбиение. Например, в грамматике:

```
PP -> Prep Noun;
S -> Verb PP;
```

корневым правилом является `S`, состоящее из `Verb` (глагола) и `PP` (существительное с предлогом). Поэтому извлечение фактов из исходный текста будет отталкиваться от корневого правила, которое разбивается на составные.

### 1.3.3 Yargy-парсер

Парсер Yargy — проект с полностью открытым исходным кодом, написанный на языке программирования Python. В основе работы библиотеки Yargy лежит алгоритм Эрли — алгоритм синтаксического анализа предложения по контекстно-свободной грамматике с использованием динамического программирования [38].

Yargy-парсер использует морфологический анализатор Pymorphy2 [39] для определения формы слова. Данный анализатор на каждую словоформу на входе выдает несколько возможных результатов морфологической информации и лемм, тогда как анализатор Томита-парсера выбирает единственный вариант на основе контекста.

Несмотря на то, что он полностью основан на Tomita, все правила пишутся для парсера пишутся на языке Python, а не через консольный интерфейс. Например, пример выше с грамматикой в Томита переписывается следующим образом:

```
pp = rule(gram('PREP'),
          gram('NOUN'))
S = and_(gram('NOUN'),
         pp)
```

В репозитории Natasha можно найти набор готовых правил для Yargy для извлечения таких атрибутов, как адреса, имена, деньги, даты и т.д. [40].

# 2 Проектирование алгоритма

В этом разделе рассматривается проектирование алгоритма извлечения предикторов из электронных медицинских записей первичного осмотра врачом. В работе использовались 37 медицинских записей первичного осмотра врачом. Записи были предоставлены сотрудниками СибГМУ, они представлены в формате «txt». Значимые предикторы (атрибуты), соответствующие этим записям, также определены сотрудниками СибГМУ в виде Excel-таблиц. Количество предикторов равно 49.

## 2.1 Структура первичного осмотра

Запись первичного осмотра регламентируется единой государственной информационной системой в сфере здравоохранения [41, кн. 2, раздел 4]. Согласно данному регламенту первичный осмотр должен иметь структуру, представленную на рисунке 2.1.

Предоставленные записи с первичным осмотром в стационаре разбита на 10 блоков, каждый из которых имеет свой перечень предикторов. К этим блокам относятся:

– общая информация о пациенте;

– дата и время осмотра;

– жалобы;

– анамнез болезни;

– анамнез жизни;

– эпидемиологический анамнез;

– анамнез ВТЭ (венозная тромбоэмболия);

– объективный статус;

– локальный статус;

– диагноз.

В данной работе также была написана программа для разделения на соответствующие 10 блоков с учетом того, что врачи используют разную терминологию. Например, один врач может написать «Локальный статус», а другой «St. localis» или сократить «Эпидемиологический анализ» до «Эпиданамнеза».

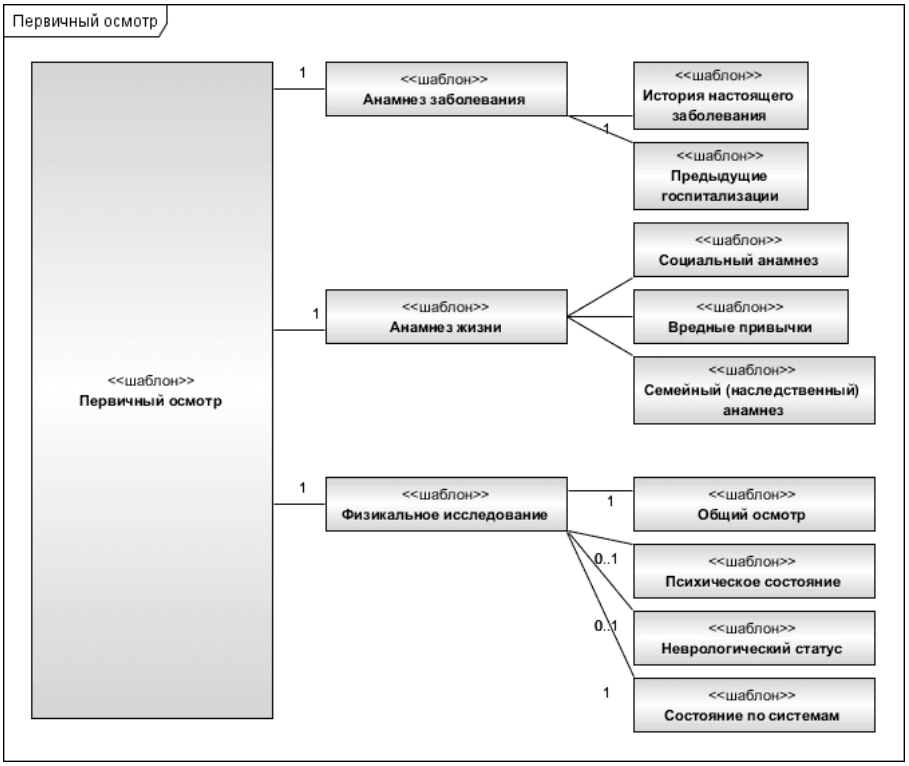


Рисунок 2.1 — Структура первичного осмотра

После анализа исходных данных значимые атрибуты были на разбиты на следующие категории:

– **числовые:** вес, рост, возраст, давление, температура и т.д.;

– **временные:** дата осмотра, дата заболевания и т.д.;

– **температурные:** температура осмотра, максимальная и минимальная;

– **половые:** пол;

– **симптоматические:** озноб, кашель, вялость и т.д.;

– **анамнестические:** аллергии, бытовые условия, речная рыба и т.д.;

– **объективно-диагностические:** гиперемия, эритема, размер отека, ППТ (правая половина тела), ЛПТ (левая половина тела) и т.д.

Каждая категория требует написания свойственных ей правил извлечения и хранения. Так, например, для численных предикторов в базу данных заносится число, для временных – даты, для симптоматического и гендерного предиктора — логический флаг (есть тот или иной симптом или нет), для всех остальных заносится вся найденная строка.

## 2.2 Подход к извлечению предикторов из первичного осмотра

### 2.2.1 Выбор парсера контекстно-свободных грамматик

Из двух рассматриваемых парсеров, рассматриваемых в предыдущей главе, был выбран Yargy, поскольку правила в нем пишутся на языке Python, у него есть готовые грамматики с поддержкой добавления новых, синтаксический анализатор, подробная документация, многочисленные примеры использования.

Yargy-парсер имеет меньшую производительность по сравнению с Томита. Это объясняется тем, что Томита написан на языке C++ и имеет различные алгоритмы оптимизации. Однако Томита уступает Yargy в удобстве пользования.

Правила в Yargy состоят из **предикатов**. Предикат — это функция, которая принимает на вход словоформу и возвращает `True`, если соответствующий факт был найден, и `False`, если не был найден. Правила и предикаты

могут логически комбинироваться при помощи операторов `and_`, `or_`, и `not_`. Таблица 2.1 демонстрирует полный список доступных предикатов в Yargy, где под `a` подразумевается входящая строка.

Таблица 2.1 — Предикаты Yargy

| **Предикат** | **Определение** |
|---|---|
| `eq(value)` | `a == value` |
| `caseless(value)` | `a.lower() == value.lower()` |
| `in_(value)` | `a in value` |
| `in_caseless` | `a.lower() in value` |
| `gte` | `a >= value` |
| `lte` | `a <= value` |
| `length_eq` | `len(a) == value` |
| `normalized` | Нормальная форма слова (равно) |
| `dictionary` | Нормальная форма слова `in value` |
| `gram` | есть ли value среди граммем слова |
| `type` | Тип токена равен `value` |
| `tag` | Тег токена равен `value` |
| `custom` | Пользовательская функция в качества предиката |
| `is_lower` | Все буквы прописные |
| `is_upper` | Все буквы строчные |
| `is_title` | Токен начинается со строчной буквы |

## 2.3 Составление грамматик

Поскольку в каждом блоке первичного осмотра требуется сделать одно и то же: извлечь соответствующие предикторы, — то было принято решение создать 10 классов, которые наследуются от одного общего. Каждый класс будет отвечать за свой блок. Конструктор общего класса принимает в качестве аргумента текст, который у наследуемых классов будет свой. На рисунке 2.2 показана UML-диаграмма классов. В ней обозначено только 3 наследуемых класса.

В общем классе имеются метод, который записывает в словарь предикторы в формате: значение и «сырая» строка. Значением может быть целочисленное или число с плавающей точкой, нормализованная строка и дата. «Сырая» строка хранит ту строку, которая была найдена как есть, т.е. в

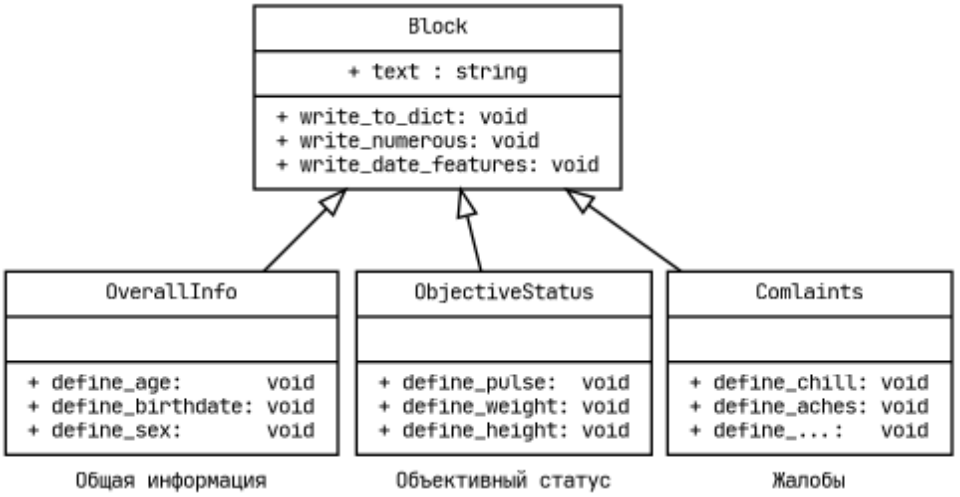


Рисунок 2.2 — UML-диаграмма классов

ненормализованном виде. Такой формат хранения предполагает, что значения можно хранить в базе данных, а с помощью сырой строки отыскивать их в исходном тексте первичного осмотра.

Также в общем классе присутствует вспомогательные методы, который вызывают метод записи в словарь. Таких методов имеется 5: запись целочисленных предикторов (вес, рост, пульс), чисел с плавающей точкой (температура), дат (осмотра, заболевания), бинарные значения, список слов. Данные методы необходимы для предварительной нормализации.

В наследуемых классов есть методы, которые производят извлечение предикторов. Все они начинаются со слова `define`. Для каждого предиктора вызываются свои вспомогательные методы. Так, например, для класса «Общая информация» — это методы для извлечения вораста, даты рождения и пола. В нем сначала определяется дата рождения, вызывается метод записи дат; из нее определяется возраст, вызывается метод записи целочисленных значений; определяется пол, вызывается главный метод записи. В классе «Жалобы» находятся бинарные предикторы (есть та или иная жалоба или нет) и после извлечения вызывается метод общего класса для записи бинарных предикторов. Также среди жалоб могут находиться температурные предикторы — температура, которая, вероятно, является максимальной.

В классе «Объективный статус» определяются рост, вес, пульс (целочисленные значения), оценка лимфатических узлов (обычная строка).

### 2.3.1 Правила для извлечения числовых предикторов

Числовые предикторы могут быть в блоках: «Общая информация», «Объективный статус». Мы предполагаем, что взрослый человек не может иметь рост ниже 50 см и выше 250 см, иметь массу меньше 10 кг и больше 200 кг или возраст более 120 лет. Поэтому для извлечения числовых предикторов применяются предикаты `gte` и `lte`, которые определяют пороговые значения.

Перед самим предиктором идет слово, например, «вес», «ЧСС», «давление», а после него может стоять знак пунктуации. Исходя из этого поведения для определения числовых предиктор нам требуются задать пороговые значения, ключевое слово и возможные знаки пунктуации. Например, правило для предиктор «Вес» имеет следующий вид:

```
WEIGHT = and_(gte(10), lte(150))
WEIGHT_RULE = or_(rule(normalized('вес'), '-', WEIGHT),
                  rule(normalized('вес'), '-', WEIGHT),
                  rule(normalized('вес'), ':', WEIGHT),
                  rule(normalized('вес'), WEIGHT))
```

### 2.3.2 Правила для извлечения температурных предикторов

Температурные предикторы могут находиться в блоках «Жалобы», «Анамнез болезни». Предполагается, что температура здорового человека не может быть меньше 35° и больше 41°. Поэтому для извлечения числовых предикторов применяются предикаты `gte` и `lte`, которые определяют пороговые значения.

Запись в словарь данной категории осуществляется как для чисел с плавающей точкой. В тексте встречаться температура, где десятичная часть отделяется от целого либо точкой, либо запятой:

```
DEGREES = and_(gte(35), lte(41))
SUBDEGREES = and_(gte(0), lte(9))
```

```
TEMP_RULE = or_(rule(DEGREES, ',', SUBDEGREES),
                rule(DEGREES, '.', SUBDEGREES),
                rule(DEGREES))
```

Среди значимых предикторов есть еще «Анамнестическое повышение температуры тела в начале заболевания» с префиксами «1» и «3». Первое имеет значение 0, если температура ниже 38°, или значение 1, если больше 38°. Соответствующий код для записи этой температуры:

```
max_temp = self.features_dict['temp_max'][0]
t_an_01 = None
if max_temp:
    t_an_01 = 0 if max_temp < 38.0 else 1
```

Температура с третьим префиксом имеет значение 0, если температура в пределах 37 – 38°; значение 1, если в пределах 38 – 39°; значение 2, если в пределах 39.9 – 40°; значение 3, если в выше 40°. Соответствующий код для записи этой температуры:

```
max_temp = self.features_dict['temp_max'][0]
if max_temp < 38:
    t_an_3 = 0
elif max_temp < 39:
    t_an_3 = 1
elif max_temp < 39.9:
    t_an_3 = 2
elif max_temp <= 40:
    t_an_3 = 3
```

### 2.3.3 Правила для извлечения временных предикторов

К временным предикторам относятся различные даты, которые встречаются в блоках: «Общая информация», «Дата и время осмотра», «Жалобы», «Анамнез болезни», «Анамнез жизни», «Эпидемиологический анамнез». «Анамнез ВТЭ». Предикаты `lte` и `gte` необходимы для задания пороговых значений: для дня — от 1 до 31, для месяца — от 1 до 12, для года — от 1910

до текущего года. В датах могут опускаться начальные цифры года, например, 20 вместо 2020. Также предусмотрен вид написания даты, при котором опускается год, т.е. указан только день и месяц.

Поскольку даты встречаются в нескольких блоках, то было принято решение написать функцию, которая находит их и возвращает в виде строки. Правила для извлечения временных предикторов выглядит следующим образом:

```
DAY = and_(gte(1), lte(31))
MONTH = and_(gte(1), lte(12))
YEAR = and_(gte(1), lte(19))

cur_year = datetime.now().year
YEAR_FULL = and_(gte(1900), lte(cur_year))
DATE = or_(
    rule(DAY, '.', MONTH, '.', YEAR_FULL),
    rule(DAY, '.', MONTH, '.', YEAR),
    rule(DAY, '.', MONTH)
)
parser = Parser(DATE)

date_str = None
for match in parser.findall(text):
    date_str = ''.join([_.value for _ in match.tokens])
```

Для хранение дат в базе данных следует их нормализовать. Для этого используется стандартная библиотека Python — `datetime`. В результате была написана функция, которая преобразует дату в виде строки в дату в виде объекта `datetime`:

```
def parse_date(date: str) -> datetime:
    pt = datetime.strptime
    if re.match('\d\d.\d\d.\d\d\d\d', date):
        return pt(date, '%d.%m.%Y')
    elif re.match('\d\d.\d\d.\d\d', date):
        date = date[:6] + '19' + date[6:]
        return pt(date, '%d.%m.%Y')
```

```
elif re.match('\d\d.\d\d', date):
    return pt(date, '%d.%m')
raise ValueError
```

### 2.3.4 Правила для извлечения полового предиктора

Половой предиктор находится в блоке «Общая информация о пациенте». В этом блоке он находится либо в скобках, либо без. После определения предиктора мужской пол определяется, как 1, а женский — 2.

Для определения этого предиктора мы не стали прибегать к использованию грамматик. Вместо этого применялись регулярные выражения. В тексте блока данный предиктор является одним единственной граммемой, состоящей из букв кириллического алфавита. Поэтому извлечение осуществляется достаточно просто:

```
sex = self.find_feature('[А-я]+', text)
```

### 2.3.5 Правила для извлечения симптоматических предикторов

Симптоматические предикторы находятся в блоках: «Жалобы», «Анамнез болезни», «Эпидемиологический анамнез» и «Анамнез жизни», «Объективный статус» и «Локальный статус». Для определения этих предикторов был составлен список из возможных симптомов, указанных сотрудниками СибГМУ. Поэтому здесь определяется есть ли у пациента тот или иной симптом, и если он был найден парсером, то он обозначается как 1, в противном случае как 0.

У симптома может быть синоним, например, «озноб» и «познабливание», он может находиться в разных формах. Поэтому чтобы не перечислять их через обычные предикаты, используется газеттир `morph_pipeline`, который, к тому же в Yargy работает быстрые, чем предикаты. Газеттир принимает на вход список возможных вариантов, включая словосочетания. Правила для извлечения некоторых симптомов формулируются следующим образом:

```
LETHARGY_RULE = morph_pipeline(['вялость', 'разбитость'])
CHILL_RULE = morph_pipeline(['озноб', 'познабливание'])
```

```
HEADACHE_RULE = morph_pipeline(['головная боль'])
WEAK_RULE = morph_pipeline(['слабость'])
ACHE_RULE = morph_pipeline(['ломота'])
```

### 2.3.6 Правила для определения анамнестических предикторов

Анамнестические предикторы находятся блоках с анамнезами: «Анамнез жизни», «Анамнез болезни», «Эпидемиологический анамнез». Данный вид предикторов определяют совокупность сведений, полученных от больного или его родственников и окружающих лиц в ходе медицинского обследования. Каждый анамнестический предиктор имеет разный формат хранения, следовательно, требуется определять грамматики для каждого из них. Например, предиктор «Бытовые условия» имеет значения: 1 (удовлетворительные) или 0 (неудовлетворительные), а предиктор «Тип места жительства» принимает значения: 0 (бездомный), 1 (частный дом, не благоустроенный), 2 (частный дом, благоустроенный), 3 (квартира, не благоустроенная), 4 (квартира благоустроенная). Анамнестические предикторы определяются по ключевым словам. Пример нахождения предиктора «Курение» выглядит следующим образом:

```
SMOKE_RULE = or_(
    rule('не', normalized('курит')),
    rule('не', normalized('употребляет'))
)
```

Однако не для всех предикторов этого типа удается получить значения категориального вида. Признак «Аллергическая реакция» не подается разбиению на категории, поскольку аллергических реакций имеется огромное множество, что их перечисление стало бы невыполнимым.

Терминалами в Yargy-парсере могут быть части речи. В некоторых предложениях встречаются лекарственные препараты, на которые у пациента имеется аллергическая реакция. Помимо препаратов могут встречаться предметы быта, например, шампунь, лосьон и т.д. Поэтому грамматика для

извлечения предиктора «Аллергическая реакция» следующая: сначала ищется само это словосочетание, затем извлекаются все терминалы в виде набора существительных и прилагательных, разделяющий однородные члены, которые являются аллергенами. Реализация данной грамматики приведена ниже.

```
ALLERG_RULE = or_(
    rule(
        normalized('не'),
        normalized('переносит')),
    rule(
        normalized('аллергическая'),
        normalized('реакция'),
        normalized('на'))
)
```

К такого же рода предикторам относится «Другие заболевания». Для определения этого предиктора был предварительно составлен список болезней, сопоставление с которым осуществляет Yargy-парсер.

### 2.3.7 Правила для определения объективно-диагностических предикторов

Объективно-диагностические предикторы находятся в блоках «Объективный статус» и «Диагноз». Они схожи с анамнестическими предикторами в том плане, что могут иметь категориальный вид или находиться в виде словосочетаний. Например, предиктор «Предрасполагающие факторы» может принимать значения 0 (нет), 1 (фоновые заболевания), 2 (наличие очагов хронической инфекции), 3 (профессиональные вредности), 4 (хронические соматические заболевания). Для их нахождения также подбираются ключевые слова:

```
factor_type_0 = morph_pipeline['микоз', 'диабет', 'ожирение',
                               'варикоз', 'недостаточность']
factor_type_1 = morph_pipeline['тонзилит','отит', 'синусит',
                               'кариес', 'пародонтоз']
factor_type_2 = morph_pipeline['резиновая обувь', 'загрязнения кожных']
factor_type_3 = morph_pipeline['соматические заболевания']
```

Если какой-либо из предрасполагающих факторов был обнаружен, то предиктору ставится соответствующее значение (0, 1, 2 или 3).

А вот предиктор «Диагноз» имеет полный строковый вид и находится в последнем блоке. Поскольку сам этот блок весь состоит из поставленного диагноза, то берется вся строка целиком.

### 2.3.8 Принцип работы алгоритма извлечения предикторов

Прежде всего исходный текст с первичным осмотром разбивается на 10 блоков. Это осуществляется с помощью регулярных выражений. В тексте содержится соответствующие названия каждого из блоков, поэтому разбивается путем извлечения подстроки текущего блока до следующего. В регулярных выражениях были предусмотрены регистр символа, сокращения и синонимы, например, «Локальный статус» и «St. localis» — это одно и то же.

Текст каждого блока передается соответствующему классу. Затем вызываются все их методы, начинающиеся со слова `define`. Классы содержат предикторы собственные словари, которые хранят сами предикторы. После выполнения методов словари объединяются в один общий словарь, который затем преобразуется в формат JSON.

Методы `define` записывают в словарь предикторы. В прошлых параграфах мы рассказали, как были составлены правила для извлечения, но не как они записываются. В родительском классе есть метод `write_to_dict`, который записывает в словарь предикторы (см. рис. 2.2). У него также есть 5 вспомогательных методов, которые отвечают за нормализацию различных видов предикторов и вызывают главный метод для записи:

– `write_num` — нормализация и запись целочисленных значений;

– `write_date` — нормализация и запись дат;

– `write_float` — нормализация и запись чисел с плавающей точкой;

– `write_binary` — нормализация и запись бинарных предикторов;

– `write_words` — нормализация и запись списка слов.

## 2.4 Результаты работы алгоритма

При тестировании использовалось 37 электронных записей с первичным осмотром. В каждом из них определены 49 предикторов. Их них 43 были определены с точностью 98 %, 5 факта с точностью 85 % и 1 предиктор с точностью 75 % (см. таб. 2.2).

При первом тестировании было обнаружено, что предиктор «Аллергическая реакция» распознается Yargy-парсером с низкой точностью. Это связано с тем, что аллергическая реакции пациента в осмотре, как на препараты, так и на другие факторы перечисляются в одном предложении без какой-либо последовательность. Например, в предложении «Аллергические реакции на пенициллин, димедрол, витамины группы В, на употребление рыбы», чтобы определить аллергическую реакцию на препараты, парсер должен знать, что из перечисленного является медицинским препаратом. Так как парсер не учитывает семантику, для этого необходимо создать словарь, где содержится список всех медицинских препаратов. В связи с тем, что количество существующих препаратов огромно и постоянно изменятся, так как некоторые препараты снимают с производства и появляется новые, создать такой словарь и постоянно его изменять является трудновыполнимой задачей.

При первичном тестировании точность определения предиктора «Употребление рыбы» составила всего 50 %. Это связано с тем, что в медицинском осмотре указывают не только положительные случаи употребления рыбы, но и отрицательные. Также было обнаружено, что данный предиктор указывается в осмотре несколькими способами. Например, «употребление речной рыбы карповых пород да», «Употребление речной рыбы карповых пород постоянно», «речную рыбу сем. карповых употребляет в разных видах» и «употребление речной рыбы карповых пород отрицает». Для каждого из способов было исправлено правило Yargy парсера. После этого удалось достичь 100 % точности нахождения данного предиктора.

Для предикторов «ЛПТ» (левая половина тела) и «ППТ» (правая половина тела), в которых указывается в какой области находится заболевание, при первичном тестировании точность составила 75 %. Сотрудники применя-

Таблица 2.2 — Точность определения предикторов

| Количество предикторов | Точность, % |
|---|---|
| 43 | 98 |
| 5 | 85 |
| 1 | 75 |

ют следующую кодировку для обозначения областей тела: 0=нет, 1=волосистая часть головы, 2=лицо, 2.1=ушная раковина, 3=носо-губной треугольник, 4=верхняя часть туловища, 5=нижняя часть туловища, 6=пах, половые органы, 7=верхняя часть спины, 8=нижняя часть спины, 9=плечо, 10=предплечье, 11= кисть, 12=бедро, 13=голень, 14=стопа. Снижение точности было связано с тем, что в некоторых осмотрах в данной графе указывается область без указания половины тела. Поэтому парсер был перенастроен таким образом, что если в диагнозе отсутствует информация о половине тела, то парсер ищет ее во блоке «Объективный статус». Также возникла проблема, что заболевания в области носа, указываются в осмотре без указания половины тела, поэтому после перенастройки парсер относит данные заболевания одновременно и в левую, и в правую половину тела. После внесения корректировок в работу парсера, точность определения данного предиктора составила 98 %.

При определении предикторов, связанных с лимфатическими узлами, возникли трудности с тем, что в осмотре может быть информация сразу о нескольких лимфоузлах в разных областях тела. Поэтому на данный момент парсер определяет предикторы только для лимфоузлов, которые первыми встречаются в осмотре.

Также в результате тестирования было обнаружено, что алгоритм с ошибками определяет кем направлен пациент в больницу (самостоятельно, поликлиникой или бригадой скорой помощи). Это связано с тем, что перед вызовом скорой помощи возможно обращение в поликлинику или пациент может после вызова скорой помощи не дождаться ее прибытия и самостоятельно обратиться в клинику. Таким образом, в первом случае сервис определяет, что пациент был направлен поликлиникой, а во втором — скорой помощью, так

как сервис фиксирует только первый факт обращения пациента в поликлинику или скорую помощь и игнорирует все последующие обращения. Точность извлечения данного предиктора составила 75 %.

При первичном тестировании предиктора «Другие заболевания в анамнезе» было обнаружено, что парсер определяет болезни, которых нет в анамнезе. Это было связано с тем, что иногда в осмотре в графе с семейным анамнезом указаны заболевания, которые имеются у родственников пациента. Таким образом, в зону поиска парсера попадала данная графа, откуда и возникали лишние заболевания. После изменения зоны поиска данного предиктора точность стала 98 %. Стоит отметить, что парсер может определять заболевания, находящиеся в специальном словаре. Список заболеваний в нем постоянно пополняется и расширяется.

Таким образом, точность определения таких предикторов, как «Аллергическая реакция», «Сопутствующие заболевания», «Болезненность лимфоузлов», «Увеличенность лимфоузлов» и «Размер лимфоузлов» равна 85 %. Точность определения предиктора «Кем направлен» равна 75 %.

# 3 Разработка веб-сервиса

Алгоритм извлечения предикторов работает только через код, написанный на Python, что затрудняет его использование для неспециалистов. Поэтому взаимодействие алгоритма и врача осуществляется через разработанный веб-сервис. Он выполнен так, чтобы в его интерфейсе можно было разобраться сразу. Для этого веб-сервис лишен многочисленных элементов, которые мешают понять его главную цель. Разработка веб-сервиса осуществлялась при помощи следующих инструментов:

- *фреймворк Flask* для написания программно-аппаратной части сервиса (backend);
- *HTML & CSS* для написания клиентской стороны пользовательского интерфейса (frontend);
- *PostgreSQL* в качестве системы управления базами данных (СУБД);

В этом разделе будет описана процедура разработки веб-сервиса с интеграцией алгоритма извлечения значимых предикторов, описанного в предыдущем разделе, а также с учетом вышеизложенных инструментов.

## 3.1 Принцип работы

Принцип работы веб-сервиса под названием *Система Извлечения Медицинских Предикторов* (сокр. СИМП) заключается в следующем: пользователь (врач) загружает файл с первичным осмотром в формате `txt`, этот файл прочитывается и при необходимости вручную редактируется, текст файла сохраняется в базе данных, далее текст разбивается на соответствующие 10 блоков, для каждого блока извлекаются значимые предикторы, которые сохраняются в базе данных и затем отображаются на веб-странице.


Рисунок 3.1 — Диаграмма потоков данных при процедуре извлечения предикторов

На рисунке 3.1 показана диаграмма потоков данных, которая описывает процедуру извлечения предикторов веб-сервиса. Отметим основные детали. Файл должен быть обязательно в формате `txt`, поскольку его проще прочитать. Прочитанный текст доступен для предварительного редактирования, например, для исправления опечаток. Более того, хранить текст в виде сессии затруднительно, поскольку текст большой, поэтому после редактирования файла загружается в базу данных.

## 3.2 Программно-аппаратная часть

В основе программно-аппаратной части веб-сервиса лежит фреймворк Flask, написанный на языке Python. Он реализует паттерн проектирования *Model View Controller* [42]. К основным преимуществам фреймворка Flask относятся:

– более «легковесный», чем Django;

– имеет встроенный сервер разработки и быстрый отладчик;

– прост в освоении;

– не требует тонкой настройки конфигурационных файлов.

Flask имеет встроенные средства защиты, которые предотвращают распространенные атаки в виде SQL-инъекций (XSS) и подделки межсайтовых запросов (CSRF) [43]. Для взаимодействия с базами данных Flask использует библиотеку объектно-реляционного отображения *SQLAlchemy*, что упрощает разработку, так как не нужно ориентироваться на конкретную СУБД.

Приложение состоит из двух основных представлений (views): `index` и `edit`. Представление `index` получает входной файл, и если файл обнаружен, то обрабатывает его, а затем происходит перенаправление на вид `edit`. Ниже представлено представление `index`.

```
@app.route('/', methods=['POST', 'GET'])
def index():
    if request.method == 'POST':
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']
        if (file == ''):
            flash('No selected file')
            return redirect(request.url)
        if is_txt(file):
            filename = secure_filename(file.filename)
            content = get_content(file)
            return redirect(url_for('edit', filename=filename))
    return render_template('index.html')
```

Представление `edit` отвечает за вызов бизнес-логики, т.е. вызов алгоритма извлечения предикторов. Данный алгоритм упакован в виде Python-модуля, который принимает на вход текст с первичным осмотром пациента. На выходе имеется извлеченные предикторы, которые записываются в базу данных.

## 3.3 Описание базы данных

База данных представлена в виде двух таблицы: `Patient` (Пациент), которая хранит поля с идентификатором и текстом первичного осмотра, и `Predictors` (Предикторы), которая хранит поля с 49 предикторами. На рисунке 3.2 изображена ER-диаграмма, которая отображает отношение *один-к-одному*.

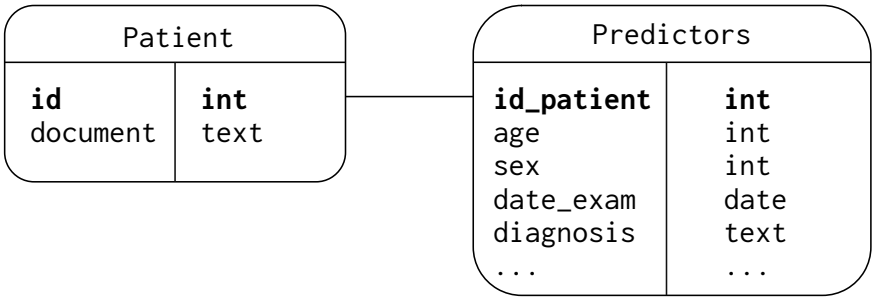


Рисунок 3.2 — Диаграмма потоков данных при процедуре извлечения предикторов

Взаимодействие фреймворка Flask с базой данных осуществляется через библиотеку SQLAlchemy. Объектно-реляционное отображение описанного отношения задается следующим образом:

```
class Patient(db.Model):
    __tablename__ = 'patient'
    id: Column(db.Integer, primary_key=True)
    document: Column(db.Text, nullable=False)


class Predictors(db.Model):
    __tablename__ = 'predictors'
    patient_id = db.Column(db.Integer, db.ForeignKey('patient.id'))
    age: Column(db.Integer)
    sex: Column(db.Integer)
    date_exam: Column(db.Date)
    # Остальные предикторы
```

Таким образом, данные вышеперечисленных классов могут быть преобразованы в таблицы СУБД. В качестве СУБД используется PostgreSQL.

## 3.4 Клиентская сторона пользовательского интерфейса

Клиентская сторона пользовательского интерфейса выполнена с помощью HTML & CSS. Для быстродействия веб-сервиса возможности JavaScript не были задействованы. Как уже было сказано в параграфе 3.1, веб-сервис состоит их двух представлений, которые выводят соответственно шаблоны веб-страниц `index.html` и `edit.html`.

Шаблон, соответствующий представлению `index`, состоит из формы, которая имеет два поля для ввода информации: поле для ввода файлов (`file`) и поле отправки файла (`submit`). Ниже представлен код соответствующего шаблона. Тэг `label` необходим для установки стилей над этими полями, иначе изменить названия кнопок без использования языка программирования JavaScript не получится.

```
<form method="post" enctype="multipart/form-data" >
  <label for="browse-file">Обзор...</label>
  <input type="file" name="file" id="browse-file">
  <label for="upload-file">Загрузить файл</label>
  <input type="submit" id="upload-file">
</form>
```

После передачи файла серверу отображается страница редактирования. Страница поделена на две части: слева находится сам текст, а справа список будет находиться список извлеченных предикторов, который появится после подтверждения об отправке данных. Если предиктор выражается в виде цепочки слов, например, перечисление болезней, то все они и отображаются. Если какие-то из предикторов отсутствуют в тексте, то им присваивается значение `None`.

Главная страница и страница редактирования имеют примечания по использованию веб-сервиса. На странице редактирования добавлена проверка орфографии, поэтому в форме будут подчеркнуты слова, которые отсутствуют в тезаурусе веб-браузера.

# Заключение

В данной работе была рассмотрена задача извлечения предикторов, подготовленные сотрудниками СибГМУ, из заключений первичного осмотра врачом в стационаре. Для решения этой задачи предложен подход, в основе которого лежит применение контекстно-свободной грамматики, реализованной через парсер Yargy.

В ходе решения данной задачи был представлен алгоритм извлечения предикторов, который применяется для каждого блока из первичного осмотра. Предложенный алгоритм показал достаточно высокую эффективность (98 %) на 43 предикторов из 49 заданных при общем количестве записей первичного осмотра равным 37. Низкая точность извлечения остальных предикторов объясняется ограничениями контекстно-свободной грамматики.

Эффективность может быть улучшена путем рассмотрения большого количества данных, поскольку правила извлечения могут стать более обобщенными. Также улучшение эффективности можно добиться, применив другие алгоритмы извлечения предикторов (атрибутов). Благодаря увеличению набора данных могут быть применены методы машинного обучения для извлечения 6 предикторов, которые были определены текущим алгоритмом с низкой точностью.

Для осуществления взаимодействия алгоритма извлечения предикторов и самих врачей был разработан веб-сервис. Он позволяет загружать и обрабатывать текстовые файлы первичных осмотров. А также веб-сервис переносит в базу данных извлеченные предикторы, а затем отображает их в виде структурированного списка. Веб-сервис позволит врачам быстрее получать значимые факты из медицинских текстов и совершать меньше ошибок при их определении.

Веб-сервис может быть улучшен путем добавления административной приборной панели, которая будет давать возможность врачам анализировать полученные результаты извлечения.

# 4 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

В рамках настоящей работы был предложен алгоритм по извлечению значимых предикторов из электронных медицинских записей с использованием контекстно-свободных грамматик, а также данный алгоритм был интегрирован в программный веб-интерфейс. В алгоритме используются специальные правила на основе контекстно-свободной грамматики. Программный веб-интерфейс дает сотруднику медицинского учреждения возможность получить характерные для данной электронной записи признаки и вывести их в структурированном виде.

Цель данного раздела — провести детальный анализ проекта по критериям конкурентоспособности и ресурсоэффективности, оценить перспективность проекта, определить трудоемкость и график работ, а также рассчитать интегральный показатель ресурсоэффективности. Для достижения цели рассмотрены аналитические инструменты и выполнены следующие задачи:

- Анализ потенциальных потребителей.
- Анализ конкурентных технических решений.
- SWOT-анализ.
- Определение целей и ожидаемого результата проекта.
- Планирование организационной структуры проекта.
- Формулирование ограничений и допущений проекта.
- Планирование структуры работ проекта.
- Построение плана проекта.
- Формирование бюджета.

– Анализ рисков проекта.

– Расчет показателя финансовой эффективности.

## 4.1 Предпроектный анализ

### 4.1.1 Потенциальные потребители результатов исследования

Для определения потенциальных потребителей требуется выявить целевой рынок. А для того, чтобы определить группу потребителей, которым необходима данная разработка, проводится *сегментирование целевого рынка.*

Рынок, на котором разработка потенциально может быть интересна, — рынок медицинских услуг. Разрабатываемый веб-сервис предназначен для вывода структурированной информации с электронных записей первичного осмотра. Продукт предлагается, в первую очередь, для врачей. Также он может быть использован для сбора и анализа медицинских данных, научно-исследовательских центрах.

Сегментация проводилась по следующим критериям: наличие поддержки обработки текстов на русском языке, наличие веб-сервиса, наличие поддержки обработки первичных осмотров лечащим врачом. Потенциальным потребителем могут быть медицинские учреждения и научные исследовательские центры.

Таблица 4.1 — Карта сегментирования рынка по обработке медицинских текстов

| | Поддержка текстов на русском | Наличие веб-сервиса | Обработка первичных осмотров |
|---|---|---|---|
| Медицинские учреждения | - | + | + |
| Научные центры | - | - | + |

Карта сегментирования представлена в таблице 4.1, где знаком «плюс» помечены занятые ниши, а знаком «минус» — незанятые. На основе построенной карты сегментирования приходим к следующему выводу: веб-сервис по работе с с первичными осмотрами на русском языке может занять свою нишу.

### 4.1.2 Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения

Анализ конкурентоспособности технического решения был проведен с помощью оценочной карты (таблица 4.2). Позиция собственной разработки и разработки конкурентов оценивается по каждому показателю экспертным путем по пятибалльной шкале, где 1 — наиболее слабая позиция, а 5 — наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять в сумме 1.

Таблица 4.2 — Оценочная карта конкурентных технических решений

| **Критерий оценки** | **Вес критерия** | **Балы** | | | **Конкурентоспособность** | | |
|---|---|---|---|---|---|---|---|
| | | $Б_ф$ | $Б_{К2}$ | $Б_{К2}$ | $К_ф$ | $К_{К1}$ | $К_{К2}$ |
| **Технические критерии оценки ресурсоэффективности** | | | | | | | |
| 1. Удобство | 0.1 | 5 | 3 | 3 | 0.5 | 0.3 | 0.3 |
| 2. Скорость обработки данных | 0.08 | 5 | 4 | 3 | 0.4 | 0.32 | 0.24 |
| 3. Потенциальные возможности расширения системы | 0.12 | 4 | 5 | 4 | 0.48 | 0.6 | 0.48 |
| 4. Доступность | 0.15 | 5 | 3 | 3 | 0.75 | 0.45 | 0.45 |
| 5. Независимость от мощностей технического оборудования | 0.15 | 5 | 4 | 4 | 0.75 | 0.6 | 0.6 |
| 6. Потребность в ресурсах памяти | 0.05 | 3 | 5 | 4 | 0.15 | 0.25 | 0.2 |
| 7. Удобство графического интерфейса | 0.05 | 5 | 5 | 3 | 0.25 | 0.25 | 0.15 |
| **Экономические критерии оценки ресурсоэффективности** | | | | | | | |
| 8. Актуальность на рынке | 0.1 | 5 | 5 | 5 | 0.5 | 0.5 | 0.5 |
| 9. Цена | 0.15 | 4 | 4 | 4 | 0.6 | 0.6 | 0.6 |
| 10. Послепродажное обслуживание | 0.05 | 4 | 5 | 3 | 0.2 | 0.25 | 0.15 |
| **Итого** | **1** | | | | **4.58** | **4.12** | **3.67** |

В качестве конкурентных систем рассматривались разработки Apache cTakes, США ($K1$) и система поддержки принятия клинических решений CDS, Россия ($K2$). Результаты анализа представлены в таблице 4.2.

Система Apache cTakes разработана в США и представляет собой обширную систему обработки естественного языка для извлечения информации из медицинских текстов. Однако из-за больших размеров система становится сложной в эксплуатации. Врачу при взаимодействии с ней понадобится достаточно много времени для того, чтобы разобраться, как все работает. Более того, система предназначена только для текстов на английском языке.

Система CDS разработана в России и представляет обширный комплекс решения различных задач. Система имеет средство навигации по истории болезни, отображение в виде блок-схем части информации. Однако, так же как и Apache cTakes, усложнена из-за расширенной функциональности. К преимуществам можно отнести возможность работы с текстами на русском языке.

С другой стороны, разработанное решение лишено перечисленных недостатков. Благодаря упрощенному графическому интерфейсу врачу не требуется тратить время на изучение системы, поскольку направлена на решение узкой задачи — обработку первичных осмотров на русском языке. Кроме того, веб-сервис, в отличие от десктопных приложений, не нужно никуда устанавливать. Разработанный веб-сервис выполняет все вычисления на стороне сервера, поэтому он не нуждается в дополнительных мощностях клиентского оборудования.

### 4.1.3 SWOT-анализ

SWOT — Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы) — представляет собой комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта. В таблице 4.3 представлены результаты матрицы SWOT. исследования внешней и внутренней среды проекта.

Основной возможность является переход на электронные записи в медицинских учреждениях. Действительно, врачи все больше нуждаются в компьютерных системах, которые снизят умственную нагрузку и повысят их продуктивность.

Таблица 4.3 — Матрица SWOT

| | **Сильные стороны:**<br>S1. Поддержка русского языка<br>S2. Обработка производится онлайн<br>S3. Простой интерфейс<br>S4. Не используется мощности клиентского оборудования | **Слабые стороны:**<br>W1. Решается узкий круг задач<br>W2. Требуется подключение к Интернету<br>W3. Зависимость от имеющихся данных |
|---|---|---|
| **Возможности:**<br>O1. Переход на электронные записи<br>O2. Занятость врачей | Веб-сервис поможет врачам снизить рабочую нагрузку с учетом постоянной занятости за заполнением текстовых записей.<br>При этом врачу достаточно иметь доступ в Интернет с любого компьютера в независимости от его вычислительных мощностей, т.е. на старых компьютерах веб-сервис будет исправно работать. | Возможность внедрения новых функциональных возможностей.<br>Получение дополнительных данных для улучшения работы алгоритма |
| **Угрозы:**<br>T1. Отключение Интернета<br>T2. Хакерские атаки на сервер<br>T3. Появление новых разработок с оптимизированными решениями | Внедрение дополнительного слоя защиты.<br>Улучшение текущего решения за счет получения новых данных | Создание прототипа в виде десктопного приложения |

Еще особенностью разрабатываемой системы является поддержка русского языка, доступность за счет подключения к сети Интернет и специализация на узкой задачи — обработки первичных осмотров. Также благодаря независимости от вычислительных мощностей клиентского технического оборудования веб-сервис может запускаться на старых компьютерах.

Недостатком разрабатываемого решения является зависимость от Интернета. Поэтому в те медицинские учреждения, которые его не имеют или временно его лишились, не смогут получить доступ к веб-сервису. Более того, сервер, на котором происходят вычисления, может быть подвергнут ха-

керской атаке. Если первую угрозу нельзя избежать, то со второй можно бороться с помощью усиления защиты исходного кода.

## 4.2 Инициация проекта

### 4.2.1 Цели и результат проекта

В данном разделе приводится информация о заинтересованных сторонах проекта, иерархии целей проекта и критериях достижения целей.

Под заинтересованными сторонами проекта понимаются лица или организации, которые активно участвуют в проекте или интересы которых могут быть затронуты как положительно, так и отрицательно в ходе исполнения или в результате завершения проекта. Так, заинтересованными сторонами являются **медицинские учреждения и научные центры.** Первые ожидают на выходе веб-сервис, а вторые возможность получать с веб-сервиса данные.

В таблице 4.4 представлена информацию о иерархии целей проекта и критериях достижения целей.

Таблица 4.4 — Цели и результаты проекта

| | |
|---|---|
| **Цели проекта** | Повышение эффективности анализа электронных медицинских записей с помощью разработки инструментов поиска и идентификации значимых признаков, влияющих на оценку состояния пациента. |
| **Ожидаемые результаты проекта** | Алгоритм нахождения и структурирования значимых признаков, интегрированный в веб-сервис. |
| **Критерии приемки результатов проекта** | Прохождение оценки точности алгоритма на исходных данных и прохождение тестирования веб-сервиса |
| **Требования к результату проекта** | Выполняются оба условия из ожидаемых результатов. Значения метрик оценки модели выше установленных пороговых величин. |

### 4.2.2 Организационная структура проекта

В таблице 4.5 представлена информация о рабочей группе проекта. Основными лицами являются инженер (разработчик) и его научный руководитель.

Таблица 4.5 — Рабочая группа проекта

| № | **ФИО, место работы, должность** | **Роль в проекте** | **Функции** |
|---|---|---|---|
| 1 | Котюбеев Роман Радиевич, магистрант ТПУ | Инженер | Проектирование, реализация, внедрение |
| 2 | Аксенов Сергей Владимирович, доцент ОИТ ТПУ, к.т.н. | Научный руководитель | Составление научных целей и задач, проверка документации |

### 4.2.3 Ограничения и допущения проекта

Ограничения проекта — это все факторы, которые могут послужить ограничением степени свободы участников команды проекта, а также «границы проекта» — параметры проекта или его продукта, которые не будут реализованных в рамках данного проекта. Ограничения проекта представлены в таблице 4.6.

Таблица 4.6 — Ограничения проекта

| **Фактор** | **Ограничения** |
|---|---|
| Бюджет проекта | 100 000 рублей |
| Источник финансирования | ТПУ |
| Сроки проекта | 01.02.2021–22.05.2021 |
| Дата утверждения плана управления проектом | 01.02.2021 |
| Дата завершения проекта | 22.05.2021 |

Таким образом, максимальный бюджет настоящего проекта установлен в сумме 100 000 рублей, а сроки выполнения составляют с 1 февраля по 22 мая.

## 4.3 Планирование управления проектом

Планирование проекта предполагает определение условий выполнения всех этапов и задач для установления порядка и последовательности. Группа процессов планирования состоит из процессов, осуществляемых для определения общего содержания работ, уточнения целей и разработки последовательности действий, требуемых для достижения данных целей.

Основные этапы планирования управления проектом в ходе работы определены следующим образом:

– определение структуры работ в рамках проекта;

– определение участников каждой работы;

– установление продолжительности работ;

– построение графика выполнения проекта.

**Определение структуры работ в рамках проекта и его участников.** Для составления структуры работ определяются ключевые события проекта, затем детальный перечень этапов и работ. На каждый вид работ определяется исполнитель. Распределение исполнителей по данным видам работ приведено в таблице 4.7

Таблица 4.7 — Распределение исполнителей по работам

| **Основные этапы** | **№ этапа** | **Содержание работ** | **Исполнители** |
|---|---|---|---|
| Разработка задания | 1 | Постановка задачи | Котюбеев Р.Р.<br>Аксёнов С.В. |
| Выбор направления исследования | 2 | Обзор научно-технической базы | Котюбеев Р.Р. |
| | 3 | Разработка и утверждение ТЗ | Котюбеев Р.Р.<br>Аксёнов С.В. |
| | 4 | Составление календарного плана | Котюбеев Р.Р. |
| | 5 | Разработка вариантов исполнения проекта | Котюбеев Р.Р.<br>Аксёнов С.В. |
| Разработка продукта | 6 | Выявление значимых признаков | Котюбеев Р.Р. |
| | 7 | Разработка алгоритма поиска признаков | Котюбеев Р.Р. |
| | 8 | Тестирование алгоритма | Котюбеев Р.Р. |
| | 9 | Разработка веб-сервиса | Котюбеев Р.Р. |
| | 10 | Интеграция алгоритма в веб-сервис | Котюбеев Р.Р. |
| | 11 | Тестирование веб-сервиса | Котюбеев Р.Р.<br>Аксёнов С.В. |
| Оформление отчета | 12 | Составление пояснительной записки | Котюбеев Р.Р. |

**Определение трудоемкости выполнения работ.** Для определения ожидаемых сроков выполнения проекта необходимо оценить его трудоемкость. Для этого воспользуемся формулой:

$$t_{\text{ож}\,i} = \frac{3t_{min\,i} + 2t_{max\,i}}{5},$$

где $t_{\text{ож}\,i}$ —ожидаемая трудоемкость выполнения $i$-ой работы, чел.-дн.;
$t_{min\,i}$ — минимально возможная трудоемкость выполнения заданной $i$-ой работы (оптимистическая оценка: в предположении наиболее благоприятного стечения обстоятельств), чел.-дн.;
$t_{max\,i}$ — максимально возможная трудоемкость выполнения заданной $i$-ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.;

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях $T_{p\,i}$, учитывающая параллельность выполнения работ несколькими исполнителями:

$$T_{p\,i} = \frac{t_{\text{ож}\,i}}{\text{Ч}_i},$$

где $\text{Ч}_i$ — численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

**Разработка графика проведения научного исследования.** Для удобства составления календарного плана и графика работ необходимо перевести длительность каждого из этапов из рабочих дней в календарные дни. Для этого воспользуемся следующей формулой:

$$T_{\text{к}\,i} = T_{p\,i} + k_{\text{кал}},$$

где $T_{\text{к}\,i}$ —продолжительность выполнения $i$-й работы в календарных днях;
$T_{p\,i}$ — продолжительность выполнения $i$-й работы в рабочих днях;
$k_{\text{кал}}$ — коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}} = \frac{365}{365 - 118} = 1{,}48,$$

где $T_{\text{кал}}$ — количество календарных дней в году; $T_{\text{кал}}$ — количество выходных дней в году; $T_{\text{вых}}$ — количество праздничных дней в году. выходных дней в году; $T_{\text{пр}}$ – количество праздничных дней в году. В соответствии с производственным календарем (для 6-дневной рабочей недели) в 2021 году: 365 календарных дней, 247 рабочих дней, 118 выходных/праздничных дней. В таблице 4.8 представлены подробные временные расчеты этапов отдельных видов работ. А на рисунке 4.1 представлена диаграмма Ганта.

Таблица 4.8 — Временные показатели проведения проекта

| **Наименование работы** | **Исполнители** | **Трудоемкость работ, чел.-дни** | | | **Длительность работ, дни** | |
|---|---|---|---|---|---|---|
| | | $t_{max\,i}$ | $t_{max\,i}$ | $t_{\text{ож}\,i}$ | $T_{\text{р}\,i}$ | $T_{\text{кал}}$ |
| Постановка задачи | Инженер | 2 | 4 | 2.8 | 3 | 4 |
| | Руководитель | 1 | 3 | 1.8 | 2 | 2 |
| Обзор научно-технической базы | Инженер | 7 | 8 | 7.4 | 7 | 9 |
| Утверждение ТЗ | Инженер | 8 | 10 | 8.8 | 9 | 11 |
| | Руководитель | 1 | 3 | 1.8 | 2 | 2 |
| Составление календарного плана | Инженер | 1 | 3 | 1.8 | 2 | 2 |
| Разработка вариантов исполнения проекта | Инженер | 5 | 7 | 5.8 | 6 | 7 |
| | Руководитель | 3 | 5 | 3.8 | 4 | 5 |
| Выявление значимых признаков | Инженер | 2 | 4 | 2.8 | 3 | 4 |
| Разработка алгоритма поиска признаков | Инженер | 11 | 13 | 11.8 | 12 | 15 |
| Тестирование алгоритма | Инженер | 11 | 13 | 11.8 | 12 | 15 |
| Разработка веб-сервиса | Инженер | 5 | 8 | 6.2 | 6 | 7 |
| Интеграция алгоритма в веб-сервис | Инженер | 6 | 8 | 6.8 | 7 | 9 |
| Тестирование веб-сервиса | Инженер | 10 | 16 | 12.4 | 12 | 15 |
| | Руководитель | 1 | 3 | 1.8 | 2 | 2 |
| Составление пояснительной записки | Инженер | 6 | 8 | 6.8 | 7 | 9 |

| № | Вид работ | Исполнитель | Т, кал.дн | Февраль | | | | Март | | | | | | Апрель | | | | Май | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| | | | | 1.2 | 8.2 | 15.2 | 22.2 | 1.3 | 8.3 | 15.3 | 22.3 | 29.3 | 5.4 | 12.4 | 19.4 | 26.4 | 3.5 | 10.5 | 17.5 | 22.5 |
| 1 | Постановка задачи | Инженер | 4 | | | | | | | | | | | | | | | | | |
| | | Руководитель | 2 | | | | | | | | | | | | | | | | | |
| 2 | Обзор научно-технической базы | Инженер | 9 | | | | | | | | | | | | | | | | | |
| 3 | Разработка и утверждение ТЗ | Инженер | 11 | | | | | | | | | | | | | | | | | |
| | | Руководитель | 2 | | | | | | | | | | | | | | | | | |
| 4 | Составление календарного плана | Инженер | 2 | | | | | | | | | | | | | | | | | |
| 5 | Разработка вариантов исполнения проекта | Инженер | 7 | | | | | | | | | | | | | | | | | |
| | | Руководитель | 5 | | | | | | | | | | | | | | | | | |
| 6 | Выявление значимых признаков | Инженер | 4 | | | | | | | | | | | | | | | | | |
| 7 | Разработка алгоритма поиска признаков | Инженер | 15 | | | | | | | | | | | | | | | | | |
| 8 | Тестирование алгоритма | Инженер | 15 | | | | | | | | | | | | | | | | | |
| 9 | Разработка веб-сервиса | Инженер | 7 | | | | | | | | | | | | | | | | | |
| 10 | Интеграция алгоритма в веб-сервис | Инженер | 9 | | | | | | | | | | | | | | | | | |
| 11 | Тестирование веб-сервиса | Инженер | 15 | | | | | | | | | | | | | | | | | |
| | | Руководитель | 2 | | | | | | | | | | | | | | | | | |
| 12 | Составление пояснительной записки | Инженер | 9 | | | | | | | | | | | | | | | | | |

Рисунок 4.1 — Календарный план-график проекта

### 4.3.1 Бюджет проекта

При планировании бюджета научного исследования должно быть обеспечено полное и достоверное отражение всех видов планируемых расходов, необходимых для его выполнения. Для полноты и достоверности учета всех расходов сгруппируем все затраты по следующим статьям:

– затраты на материалы;

– затраты на амортизацию;

– основная заработная плата исполнителей;

– дополнительная заработная плата исполнителей темы;

– отчисления во внебюджетные фонды (страховые отчисления);

– накладные расходы.

**Материальные расходы.** В расчет взяты только затраты на канцелярские товары в размере 1000 рублей.

Таблица 4.9 — Баланс рабочего времени

| **Показатели рабочего времени** | **Руководитель** | **Инженер** |
|---|---|---|
| Календарное число дней | 365 | 365 |
| Количество нерабочих дней<br>- выходные дни<br>- праздничные дни | 118 | 118 |
| Потери рабочего времени - отпуск - невыходы по болезни | 48 | 48 |
| Действительный годовой фонд рабочего времени | 199 | 199 |

**Основная заработная плата.** Заработная плата рассчитывается из суммы заработной платы исполнителя и научного руководителя исходя из трудоемкости каждого этапа и занятости каждого из них на данном этапе. Расходы по статье заработной плате рассчитываются по следующей формуле:

$$\text{С}_{\text{зп}} = \text{З}_{\text{осн}} + \text{З}_{\text{доп}},$$

где $\text{З}_{\text{осн}}$ — основная заработная плата; $\text{З}_{\text{доп}}$ — дополнительная заработная плата. Основная заработная плата *одного работника* рассчитывается по формуле:

$$\text{З}_{\text{осн}} = \text{З}_{\text{дн}} \cdot \text{Т}_{\text{р}},$$

где $\text{З}_{\text{дн}}$ — среднедневная заработная плата; $\text{Т}_{\text{р}}$ — продолжительность работ, выполняемых работником, раб.дн.

Среднедневная заработная плата рассчитывается по следующей формуле:

$$\text{З}_{\text{дн}} = \frac{\text{З}_{\text{м}} \cdot \text{М}}{\Phi_{\text{д}}},$$

где $\text{З}_{\text{дн}}$ — месячный должностной оклад работника, рублей;
М — количество месяцев работы без отпуска в течение года, при отпуске в 48 рабочих дней М = 10,4 месяца, 6-дневная неделя;
$\Phi_{\text{д}}$ — действительный годовой фонд рабочего времени научно-технического персонала, раб. дн. (см. таб. 4.9).

Месячный должностной оклад работника высчитывается по следующей формуле:

$$З_м = З_б \cdot (k_{пр} + k_д) \cdot k_р,$$

где $З_б$ —базовый оклад, руб.;
$k_{пр}$ — премиальный коэфф оплате труда);
$k_д$ — коэффициент доплат и надбавок (в НИИ — за расширение сфер обслуживания, за профессиональное мастерство, за вредные условия: 15–20 %);
$k_р$ — районный коэффициент, равный 1,3 (для Томска).

Для расчета основной заработной платы инженера берем оклад, равный окладу 9 489 рублей. Для расчета основной заработной платы руководителя возьмем оклад в 35 120 рублей. Тогда в таблице 4.10 приведены все результаты расчетов на основе вышеперечисленных формул.

Таблица 4.10 — Расчет основной заработной платы

| **Исполнители** | $З_б$, руб | $k_{пр}$ | $k_д$ | $k_р$ | $З_м$, руб | $З_{дн}$, руб | $Т_р$, раб.дн | $З_{осн}$, руб |
|---|---|---|---|---|---|---|---|---|
| Инженер | 9489 | 0.3 | 0.2 | 1.3 | 6168 | 322 | 86 | 27692 |
| Руководитель | 35120 | 0.3 | 0.2 | 1.3 | 22828 | 1193 | 10 | 11930 |

**Дополнительная заработная плата.** В данную статью включается сумма выплат, предусмотренных законодательством о труде, например, оплата очередных и дополнительных отпусков; оплата времени, связанного с выполнением государственных и общественных обязанностей; выплата вознаграждения за выслугу лет и т.п. (в среднем — 12 % от суммы основной заработной платы). Дополнительная заработная плата рассчитывается исходя из 10 –15 % от основной заработной платы, работников, непосредственно участвующих в выполнение темы:

$$З_{доп} = k_{доп} \cdot З_{осн},$$

где $З_{осн}$ — дополнительная заработная плата, руб.; $k_д$ — коэффициент дополнительной зарплаты (на стадии проектирования принимается равным 0,15); $З_{осн}$ — основная заработная плата, руб.

В таблице 4.11 приведена форма расчета основной и дополнительной заработной платы исполнителей. Дополнительная заработная плата высчитывалась по вышеприведенной формуле.

Таблица 4.11 — Заработная плата исполнителей

| **Заработная плата** | **Руководитель** | **Инженер** |
|---|---|---|
| Основная зарплата | 11 930 | 27 692 |
| Дополнительная зарплата | 1 790 | 4 154 |
| Зарплата исполнителя | 13 720 | 31 845 |
| Итого по статье $С_{зп}$ | 45 565 | |

Таким образом, зарплата научного руководителя за период исполнения проекта составляет 13 720 рублей, инженера — 31 845 рублей. Всего расходов по статье заработной платы — 45 565 рублей.

**Отчисления во внебюджетные фонды.** В данной статье расходов отражаются обязательные отчисления во внебюджетные фонды по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется по следующей формуле:

$$С_{внеб} = k_{внеб} \cdot (З_{осн} + З_{доп}),$$

где $k_{внеб}$ — коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Общие тарифы страховых взносов в 2021 году в ИФНС: 22 % — страхование по временной; 5,1 % — медицинское страхование; 2,9 % — страхование по временной нетрудоспособности.

Таким образом, отчисления во внебюджетные фонды, исходя из всех перечисленных страховых взносов, составляют:

$$С_{внеб} = 0{,}3 \cdot 45\,565 = 13\,670 \text{ руб.}$$

**Накладные расходы.** При выполнении проекта могут возникнуть косвенные издержки — накладные расходы, возникающие дополнительно к основным затратам, например, на консультационные услуги, оплату коммунальных услуг, расход на услуги связи (телефон, Интернет) и т.д.

Расчет накладных расходов ведется по следующей формуле:

$$\text{С}_{\text{накл}} = k_{\text{накл}} \cdot (\text{З}_{\text{осн}} + \text{З}_{\text{доп}}),$$

где $k_{\text{накл}}$ — коэффициент накладных расходов. Величину коэффициента накладных расходов можно взять в размере 16 %.

В результате сумма накладных расходов составляет:

$$\text{С}_{\text{накл}} = 0.16 \cdot 45\,565 = 7\,290 \text{ руб.}$$

**Формирование бюджета.** После выполнения всех расчетов по статьям можно определить плановую общую себестоимость проекта. В таблице 4.12 представлены статьи расходов проекта.

Таблица 4.12 — Бюджет затрат на разработку

| **Наименование** | **Сумма, руб** | **Удельный вес, %** |
|---|---|---|
| Затраты на материалы | 1000 | 1.36 |
| Затраты на основную заработную плату | 45 565 | 62.02 |
| Затраты на дополнительную заработную плату | 5 944 | 8.09 |
| Страховые взносы | 13 670 | 18.61 |
| Накладные расходы | 7 290 | 9.92 |
| Общий бюджет | 73 469 | 100 |

Исходя из расчета бюджета затрат следует, что наибольшая его часть приходится на основную и дополнительную заработную плату исполнителей (62,02 %). Также необходимо отметить, что расходы на страховые взносы (18,61 %) составляют значительную часть расходов. Затраты на материалы и накладные расходы составляют небольшую долю (суммарно 12 %). Это связано с отсутствием необходимости использования дорогостоящего оборудования и материалов.

### 4.3.2 Реестр рисков проекта

Существуют идентифицированные риски проекта, которые включают в себя возможные неопределенные события. Они могут возникнуть в проекте и вызвать последствия, которые влекут за собой нежелательные эффекты. Оценка рисков проекта представлена в таблице 4.13. Для каждого из них даны рекомендации по смягчению их воздействия.

В результате данного этапа были рассмотрены возможные риски при реализации настоящей работы. Основная часть рисков может привести к неконкурентоспособности разработанного решения. Однако их воздействие можно минимизировать благодаря проведению прототипирования, итеративности разработки, проведению технического анализа стоимости и проведению сравнительного тестирования.

Таблица 4.13 — Реестр рисков

| № | Риск | Потенциальное воздействие | Вероятность наступления | Влияние риска | Уровень риска | Способы смягчения риска | Условия наступления |
|---|---|---|---|---|---|---|---|
| 1 | Несоответствие разработанной и требуемой функциональности | Недостаточная функциональность может привести к неконкурентоспособности решения | 2 | 3 | средний | Создание прототипов, разработка сценариев использования, участие потенциальных пользователей | Неправильно поставлены задачи, неполный анализ качества разработки и ее перспективности на рынке |
| 2 | Постоянный поток изменений требований | Задержки выполнения работ | 2 | 2 | низкий | Установка ограничений для внесения изменений, итеративность разработки (внесения изменений в следующих итерациях) | Ошибки при постановке задачи |
| 3 | Технологическое отставание | Неконкурентоспособность устройства | 2 | 2 | низкий | Технический анализ, анализ стоимости, прототипирование | Недостаточная оценка существующих аналогов |
| 4 | Недостаточная производительность | Неконкурентоспособность устройства | 1 | 3 | средний | Проведение сравнительного тестирования, прототипирование | Ошибки при постановке задачи, недостаточный анализ качества разработки и ее перспективности на рынке |

## 4.4 Определение экономической эффективности

Показатели экономической эффективности проекта учитывают финансовые последствия его осуществления для предприятия, реализующего данный проект. В этом случае показатели эффективности проекта в целом характеризуют с экономической точки зрения технические, технологические и организационные проектные решения.

Определение эффективности происходит на основе расчета интегрального финансового показателя, который рассчитывается следующим образом:

$$I = \frac{\Phi_{\text{р}}}{\Phi_{max}},$$

где $\Phi_{\text{р}}$ — стоимость исполнения работ; $\Phi_{\text{max}}$ — максимально допустимая стоимость исполнения проекта;

Общий бюджет проекта составил 73 469 рублей. Исходя из ограничений, накладываемых на проект, максимальный бюджет не должен превышать 100 000 рублей. Таким образом, значения финансового показателя составляет:

$$I = \frac{73\,469}{100\,000} = 0{,}73.$$

Значения финансового показателя составляет 0,73, что свидетельствует об эффективном использовании финансовых ресурсов.

## Заключение по разделу

Результаты оценки востребованности разработки можно считать положительными, поскольку, во-первых, были выявлены потенциальные потребители разрабатываемого решения. Во-вторых, в результате анализа конкурентоспособности выяснилось, что разработанное решение является более предпочтительным для широкого класса задач и обладает достаточными конкурентными преимуществами благодаря новизне метода и особенностям получаемых результатов. В-третьих, проведенный SWOT-анализ показал перспективность разработки. Расширение функционала, оказание услуг по настрой-

ки или консультированию под каждую конкретную предметную область совместно с поддержанием стабильной ценовой политики позволит сохранять свою конкурентоспособность.

В данном разделе разработан план и сформирован бюджет технического решения. Продолжительность и сформирован проекта бюджет составила 107 календарных дней, а общий бюджет затрат составил 73 469 рублей. Таким образом, план-график и бюджет проекта успешно укладываются в ограничения. Разработанный реестр рисков отражает потенциальные пути преодоления внешних и внутренних рисков и способствует успешной реализации проекта, а также его дальнейшее существование, а рассчитанный интегральный финансовый показатель эффективности

# 5 Социальная Ответственность

В рамках настоящей работы был предложен алгоритм по извлечению значимых предикторов из электронных медицинских записей с использованием контекстно-свободных грамматик, а также на основе данного подхода был разработан программный веб-интерфейс. В алгоритме используются специальные правила в соответствии с контекстно-свободной грамматикой. Программный веб-интерфейс предоставляет работнику медицинского учреждения использовать алгоритм для извлечения значимых предикторов и выводить их в структурированном виде.

Разработанная система может использоваться в медицинских учреждениях, в которых применяются система электронной документации, с целью уменьшения умственной нагрузки с работников и облегчения работы с медицинскими электронными записями.

Разработка программного обеспечения и алгоритма осуществлялось на персональном компьютере с монитором, стоящими на рабочем столе в комнате с окнами и шторами.

## 5.1 Правовые и организационные вопросы обеспечения безопасности

В данном подразделе рассматриваются специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства. Указываются особенности трудового законодательства применительно к конкретным условиям проекта.

### 5.1.1 Специальные правовые нормы трудового законодательства

С 2021 г. вопрос установления перерывов во время работы за компьютером нормативно не урегулирован [44]. Основные правовые гарантии в

части обеспечения производственной безопасности регламентирует Трудовой кодекс РФ (ред. от 30.04.2021) [45].

Так как работа с данной библиотекой на предприятии подразумевает сбор и анализ персональных данных. Чтобы ограничить доступ к медицинским данным и обеспечить их безопасность, обработка данных должна осуществляться в соответствии с федеральным законом о защите персональных данных на основании Федеральный закон «О персональных данных» [46]:

- Обработка персональных данных должна осуществляться на законной и справедливой основе.
- Обработка персональных данных должна ограничиваться достижением конкретных, заранее определенных и законных целей. Не допускается обработка персональных данных, несовместимая с целями сбора персональных данных.
- Не допускается объединение баз данных, содержащих персональные данные, обработка которых осуществляется в целях, несовместимых между собой.
- Обработке подлежат только персональные данные, которые отвечают целям их обработки.
- Содержание и объем обрабатываемых персональных данных должны соответствовать заявленным целям обработки. Обрабатываемые персональные данные не должны быть избыточными по отношению к заявленным целям их обработки.

### 5.1.2 Организационные мероприятия при компоновке рабочей зоны

На основании ГОСТ 12.2.032-78 «Рабочее место при выполнении работ сидя» рабочее место для выполнения работ сидя организуют при легкой работе, не требующей свободного передвижения работающего, а также при работе средней тяжести в случаях, обусловленных особенностями технологического процесса [47].

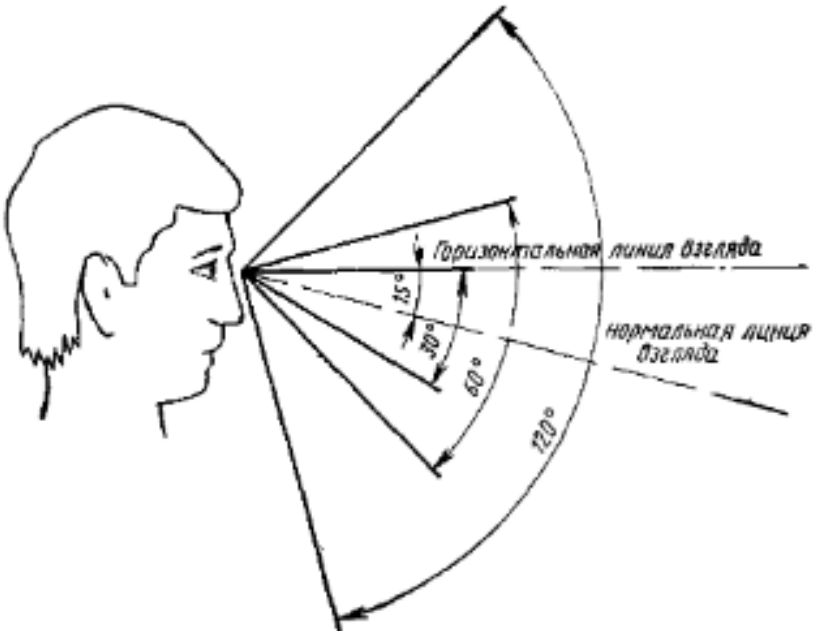


Рисунок 5.1 — Зона зрительного наблюдения в вертикальной плоскости

Средства отображения информации следует располагать в вертикальной плоскости под углом $\pm 15^\circ$ от нормальной линии взгляда и в горизонтальной плоскости под углом $\pm 15^\circ$ от сагитальной плоскости (Рисунок 5.1)

Взаимное расположение элементов рабочего места должно обеспечивать возможность осуществления всех необходимых движений и перемещений для эксплуатации и технического обслуживания оборудования [48].

Приборы на столе должны размещаться так, чтобы руки не скрещивались. Аварийные органы управления (кнопка выключения) должны располагаются в зоне досягаемости моторного поля.

## 5.2 Производственная безопасность

В подразделе проанализированы вредные и опасные факторы, которые могут возникать при проведении исследований в лаборатории, при разработке или эксплуатации проектируемого решения.

При рассмотрении безопасности в рабочей зоне были выявлены вредные и опасные факторы, которые могут возникнуть на рабочем месте. Фак-

Таблица 5.1 — Возможные вредные факторы

| Факторы (ГОСТ 12.0.003-2015) | Этапы работ | | | Нормативные документы |
|---|---|---|---|---|
| | Разработка | Изготовление | Эксплуатация | |
| Недостаточная освещенность рабочей зоны | + | + | + | СНиП 23-05-95*. Естественное и искусственное освещение |
| Отклонение показателей микроклимата | + | + | + | СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату |
| Превышение уровня шума на рабочем месте | + | + | + | СН 2.2.4/ 2.1.8.562-96 Шум на рабочих местах |
| Перенапряжение анализаторов | + | + | + | ГОСТ Р 50923-96 Дисплеи. Рабочее место оператора. |
| Статические перегрузки, связанные с рабочей позой | + | + | + | ГОСТ 12.2.032-78. Рабочее место при выполнении работ сидя |

Таблица 5.2 — Возможные опасные факторы

| Факторы (ГОСТ 12.0.003-2015) | Этапы работ | | | Нормативные документы |
|---|---|---|---|---|
| | Раз-тка | Изг-ние | Эксп-ция | |
| Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека | + | + | + | ГОСТ 12.1.038-82 ССБТ. Предельно допустимые значения напряжений прикосновения и токов |

торы считаются вредными, если его воздействие на человека может привести к заболеванию. Опасный фактор может привести к травме.

Были описаны мероприятия по защите разработчика и пользователей конечного продукта от действия данных факторов. В Таблица 5.1 приведе-

ны вредные факторы, влияющие на работу с компьютером; в таблице 5.2 приведены опасные факторы, которые могут присутствовать при работе за компьютером.

### 5.2.1 Недостаточная освещенность рабочей зоны

Вредное воздействие параметров освещения проявляется в отсутствии или недостатке естественного света, а также недостаточной освещенности рабочей зоны. Правильно спроектированное и рационально выполненное освещение производственных помещений оказывает положительное воздействие на работающих, способствует повышению эффективности и безопасности труда, снижает утомление и травматизм, а также сохраняет высокую работоспособность [49].

Таблица 5.3 — Нормируемые показатели освещенности

| Характеристика зрительной работы | Наименьший объект различения, мм | Разряд зрительной работы | Контраст объекта с фоном | Фон | Освещенность, лк |
|---|---|---|---|---|---|
| Очень высокой точности | 0,15-0,30 | IIб | Малый | Средний | 300 |

Искусственное освещение в помещениях для эксплуатации ПК должно осуществляться системой общего равномерного освещения. В случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов). Согласно СНиП 23-05-95 работу с ПК уровень освещенности стоит принять равной 300 лк при очень высокой точности зрительной работы (см. таб. 5.3).

Рассчитаем световой поток светильников типа ОД, а также их количество, используя методические указания [49].

Рабочее место представляет собой квадратное помещение с длиной $A = 6$ м, шириной $B = 3{,}5$ м, высотой $H = 2{,}5$ м.

Определим расчетную высоту подвеса светильников над рабочей поверхностью по следующей формуле:

$$h = H - h_p - h_c = 2{,}5 - 0{,}55 - 0{,}45 = 1{,}5 \text{ м},$$

где $h_p$ — расстояние от пола до рабочей поверхности стола, м; $h_c$ — расстояние от потолка до светильника, м.

Наивыгоднейшее расположение светильников $\lambda$ типа ОД равно 1,4. Расстояние между светильниками $L$ определяется как:

$$L = \lambda \cdot h = 1{,}4 \cdot 1{,}5 = 2{,}1 \text{ м}.$$

Определяем количество рядов светильников $n_{\text{ряд}}$ и количество светильников в ряду $n_{\text{св}}$:

$$n_{\text{ряд}} = \frac{B - \frac{2}{3}L}{L} + 1 = \frac{2{,}5 - \frac{2}{3} \cdot 2{,}1}{2{,}1} + 1 = 2,$$

$$n_{\text{св}} = \frac{A - \frac{2}{3}L}{l_{\text{св}} + l_{\text{мд}}} + 1 = \frac{6 - \frac{2}{3} \cdot 2{,}1}{1{,}23 + 0{,}5} + 1 = 2,$$

где $l_{\text{св}}$ — длина светильника, м; $l_{\text{мд}}$ — расстояние между концами светильников, м. Светильники ОД имеют длину 1,23 м, а расстояние между концами светильников примем равным 0,5 м.

Таким образом, размещаем светильники в два ряда, каждый их которых содержит два светильника типа ОД мощностью 40 Вт (с длиной 1,23 м); изображаем в масштабе план помещения и размещения на нем светильников (рис. 5.2). Учитывая, что в каждом светильнике установлено две лампы, общее число ламп в помещении $N = 8$.

Индекс помещения определяется по следующей формуле:

$$i = \frac{S}{h(A + B)} = \frac{6 \cdot 2{,}5}{1{,}9 \cdot (6 + 3{,}5)} = 1{,}47,$$

где $S$ площадь помещения, м$^2$ ; $A$ — длина комнаты, м; $B$ —— ширина комнаты, м; $h$ — высота подвеса светильников, м.

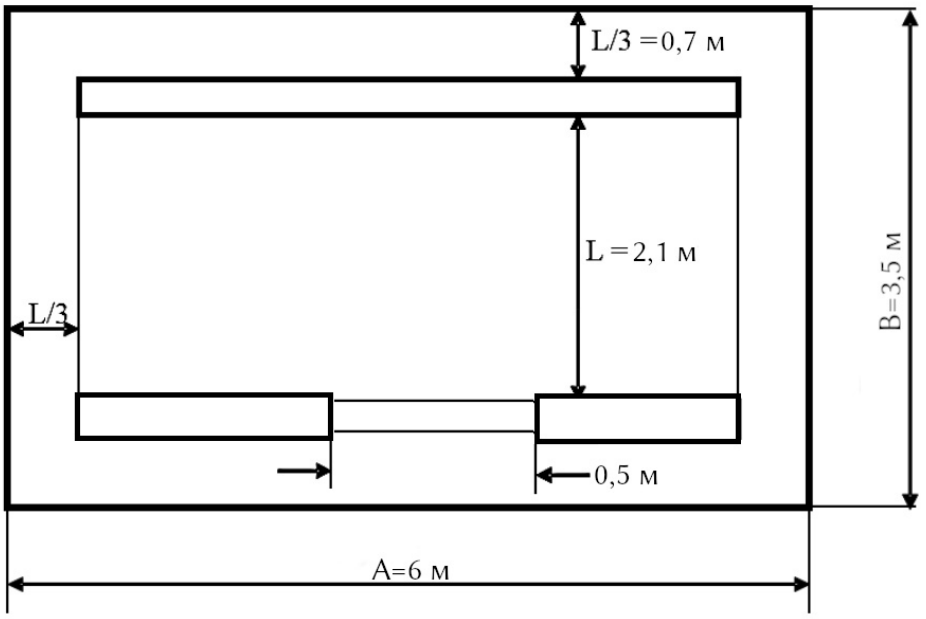


Рисунок 5.2 — План размещения светильников с люминесцентными лампами

В помещении свежепобеленный потолок, а также свежепобеленные стены с окнами без штор, поэтому примем коэффициенты отражения от стен $\rho_c = 50\%$ и коэффициенты отражения от потолка $\rho_n = 70\%$. По таблице коэффициентов использования светового потока для соответствующих значений $i, \rho_c, \rho_n$ коэффициент использования светового потока равен 0,56. Таким образом, количество светильников рассчитывается по формуле:

$$\Phi = \frac{ESZk_з}{N\eta},$$

где $\Phi$ — световой поток светильника, лм;

$E$ — нормируемая освещенность, лк;

$S$ — площадь помещения, м$^2$;

$k_з$ — коэффициент запаса;

$N$ — число ламп;

$\eta$ — коэффициент использования светового потока,

$Z$ — коэффициент неравномерности освещения.

Коэффициент запаса $k_з$ учитывает запыленность светильников и их износ. Для помещений с малым выделением пыли $k_з = 1{,}5$. Для люминес-

центных ламп $z = 1{,}1$ . Таким образом, световой поток определяется:

$$\Phi = \frac{300 \cdot 21 \cdot 1{,}1 \cdot 1{,}5}{8 \cdot 0{,}56} = 2320\,\text{лм}.$$

Выбираем ближайшую стандартную лампу ЛД 40 Вт. Она имеет нормированный поток равный 2320 лм [49, таб. 4.1]. Делаем проверку выполнения условия:

$$-10\% \leq \frac{\Phi_{\text{стан}} - \Phi_{\text{расч}}}{\Phi_{\text{стан}}} \cdot 100\% \leq 20\%,$$

$$-10\% \leq \frac{2300 - 2320}{2300} \cdot 100\% \leq 20\%,$$

$$-10\% \leq -0{,}008 \cdot 100\% \leq 20\%.$$

Как видим условия выполняются. Далее, определяем электрическую мощность осветительной установки:

$$P = 8 \cdot 40 = 320\,\text{Вт}.$$

Таким образом, для разрабатываемого помещения необходимо 4 светильника типа ОД мощностью 40 Вт. Учитывая, что в каждом светильнике установлено две лампы, в помещении требуется установить 8 ламп ЛД 40 Вт со световым потоком 2300 лм.

### 5.2.2 Отклонение показателей микроклимата

Источником отклонения показателей микроклимата являются интенсивная работа персонального компьютера.

СанПиН 2.2.4.548-96 [50] является нормативным документов, отвечающим за гигиенические требования к микроклимату производственных помещений. В документе указаны все нормативные требования к микроклимату на рабочих местах для всех видов производственных помещений.

Одним из необходимых условий труда является обеспечение нормального микроклимата в рабочей зоне, оказывающие значительное влияние на самочувствие человека. Показатели микроклимата должны обеспечивать сохранение теплового баланса человека с окружающей средой и поддержание

Таблица 5.4 — Оптимальные величины показателей микроклимата на рабочих местах

| Период года | Категория работ по уровню энергозатрат | Температура воздуха, °С | Температура поверхностей, °С | Относительная влажность воздуха, % | Скорость движения воздуха, м/с |
|---|---|---|---|---|---|
| Холодный | 1а | 22–24 | 20–24 | 40–60 | 0,1 |
| Теплый | (до 139 Вт) | 23–25 | 21–25 | 40–60 | 0,1 |

оптимального или допустимого теплового состояния организма. Оптимальный микроклимат является необходимым на рабочих местах, так как создает комфортное нахождение человека в рабочей зоне, а также обеспечивает его высокий уровень работоспособности. Такие микроклиматические условия обеспечивают благоприятное состоянию организму человека и не вызывают отклонений в состоянии его здоровья.

Согласно ГОСТ 12.0.002-2014 работа программиста подразумевает умственный труд [47] и имеет безопасные оптимальные условия труда (1-й класс) согласно руководству Р.2.2.2006-05 [51], поскольку обеспечивает максимальную производительность труда и минимальную напряжённость организма. Категорию работ следует отнести к категории 1а — энергозатраты не превышают 139 Вт.

При работе должны быть обеспечены оптимальные параметры микроклимата для категории работ 1а в соответствии с действующими санитарно-эпидемиологическими нормативами микроклимата производственных помещений. На других рабочих местах следует поддерживать параметры микроклимата на допустимом уровне, соответствующем требованиям СанПиН 2.2.4.548-96.

В таблице 5.4 представлены оптимальные и допустимые показатели микроклимата рабочей зоны.

### 5.2.3 Превышение уровня шума

Вентиляционные установки, кондиционеры, ЭВМ и его периферийные устройства, а также серверные комнаты являются источниками возникновения шумов.

Таблица 5.5 — Допустимые значения уровней звукового давления

| Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц | | | | | | | | | Эквивалентные уровни звука, дБА |
|---|---|---|---|---|---|---|---|---|---|
| 31,5 | 63 | 125 | 250 | 500 | 1000 | 2000 | 4000 | 8000 | |
| 86 | 71 | 61 | 54 | 49 | 45 | 42 | 40 | 38 | 50 |

СН 2.2.4/ 2.1.8.562-96 является нормативным документов, отвечающим за санитарные нормы шума на рабочих местах, в помещениях жилых, на территории жилой застройки, производственных помещений [52].

Повышенный шум на рабочем месте оказывает вредное влияние на организм работника в целом, вызывая неблагоприятные изменения в его органах и системах. При этом специфическим клиническим проявлением вредного действия шума является стойкое нарушение слуха (тугоухость), рассматриваемое как профессиональное заболевание [53].

Шум от ПЭВМ классифицируется как широкополосный, непостоянный, колеблющийся, поскольку имеет ширину более 1 октавы, изменяется непрерывно в течение 8-часового рабочего дня. Также среди источников шума выделяются осветительные приборы дневного света и шумы, проникающие извне.

Согласно СН 2.2.4/2.1.8.562-96 предельно допустимые уровни звукового давления в октавных полосах частот, уровни звука и эквивалентные уровни звука в квартире жилого помещения должны лежать в пределах, указанных в таблице 5.5

Предлагаемые средства защиты — снизить уровень шума в помещениях с помощью звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в области частот 63 – 8000 Гц.

### 5.2.4 Перенапряжение анализаторов

Работа за персональным компьютером сопряжена с воздействием вредных психофизиологических факторов, в частности, нервно-психических перегрузок таких, как перенапряжение анализаторов [54]. Основная характеристика анализаторов — высокая чувствительность; хотя не всякий раздра-

житель, действующий на анализатор, вызывает ощущение. Основным источником перенапряжения анализаторов является поступающая информация с монитора компьютера.

Согласно ГОСТ Р 50923-96 отношение яркостей в зоне наблюдения (экран, документ, поверхность стола) должно быть не более 10:1 [55]. В поле зрения оператора должны отсутствовать прямая и отраженная блескость. Для снижения блескости необходимо:

- оборудовать светопроемы солнцезащитными устройствами (шторами, регулируемыми жалюзи, внешними козырьками и т.д.);
- использовать для общего освещения светильники с рассеивателями и экранирующими решетками, яркость которых в зоне углов излучения более 50° от вертикали не должна превышать 200 кд/м;
- использовать для местного освещения светильники с непросвечивающим отражателем и защитным углом не менее 40°;
- размещать рабочий стол так, чтобы оконный проем находился сбоку (справа или слева), при этом дисплей должен располагаться на поверхности стола справа или слева от оператора;
- размещать рабочий стол между рядами светильников общего освещения;
- использовать дисплей, имеющий антибликовое покрытие экрана или антибликовый фильтр.

С 2021 года вопрос установления перерывов во время работы за компьютерами нормативно не урегулирован [44]. Работодатель может самостоятельно установить порядок предоставления перерывов в работе за компьютером для отдыха в правилах внутреннего трудового распорядка. Указанные перерывы включаются в рабочее время. Во время этих перерывов работник не должен выполнять другую работу [45].

### 5.2.5 Статические перегрузки, связанные с рабочей позой

Работа с ПЭВМ характеризуется монотонностью труда в сидячем положении. В положении сидя могут возникать застойные явления в органах таза, затруднение работы органов кровообращения и дыхания [49].

ГОСТ 12.2.032-78 устанавливает общие эргономические требования к рабочим местам при выполнении работ в положении сидя [56]. Допустимые нормы антропометрических показателей согласно ГОСТ Р ИСО 9241-5-2009 следующие [57]:

- высота сиденья должна равняться длине голеней пользователя до подколенной области или быть немного меньше;
- ступня составляет угол в 90° по отношению к подколенной части ноги;
- линия зрения заключена между горизонталью и 60° ниже горизонтали.
- бедра расположены приблизительно в горизонтальной позиции;
- ноги от колена до ступни — в вертикальной позиции;
- плечо расположено вертикально, предплечье— горизонтально;
- работа не требует сгибаний или разгибаний запястий;
- скручивание верхней части туловища отсутствует;
- позвоночник расположен вертикально;

### 5.2.6 Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека

В помещениях, оборудованных ПЭВМ, токи статического электричества чаще всего возникают при прикосновениях персонала к элементам

ПЭВМ. Подобные разряды опасности для человека не представляют, однако способны вызывать неприятные ощущения и вывести оборудование из строя.

Поражение электрическим током может привести к ожогам, судорогам, повреждению нервной системы, а также смерти. Возникновение пожара может привести к последствиям, описанным в ГОСТ 12.1.033-81 [58].

Также к опасностям использования электрического тока относятся возможность поражения электрическим током, а также воспламенения электронных устройств из-за воздействия различных условий— попадания влаги или повреждения изоляции.

Во избежание негативных последствий необходимо соблюдать правила пожарной и электрической безопасности. Подготовка к возникновению данных ситуаций должна производится до начала работы.

Требования безопасности при эксплуатации электрооборудования регламентируются следующими актами, изложенными ниже.

Правилами устройства электроустановок (издание шестое с отдельными разделами и главами в издании седьмом), утвержденными Главтехуправлением, Госэнергонадзором Минэнерго СССР 05.10.1979 г.

Правилами технической эксплуатации электроустановок потребителей, утверждёнными Приказом Минэнерго России от 13.01.2003 г. №6.

Межотраслевыми правилами охраны труда (правилами безопасности) при эксплуатации электроустановок (ПОТ РМ 016-2001), утвержденными Постановлением Минтруда России от 05.01.2001 г. №3.

**Исходя из вышеперечисленных нормативов должны на рабочем месте быть выполнены все рекомендации, изложенные ниже.**

Электрооборудование, имеющее контакты для подключения заземления, должно быть заземлено, а помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации оборудования.

Все крышки и защитные панели должны находиться на своих местах (при отсутствии крышки или защитной панели эксплуатация электрооборудования не допускается).

При работе с электрооборудованием не допускать попадания влаги на поверхность электрооборудования, а также запрещается работать с электрооборудованием влажными руками.

Вентиляционные отверстия электрооборудования не должны быть перекрыты находящимися вплотную стенами, мебелью, посторонними предметами.

Выдергивание штепсельной вилки электроприбора необходимо осуществлять за корпус штепсельной вилки, при необходимости придерживая другой рукой корпус штепсельной розетки.

Подключение и отключение разъемов компьютеров и оргтехники должно производиться при отключенном питании (за исключением подключения и отключения USB-устройств).

Удаление пыли с электрооборудования должно производиться в отключенном от электрической цепи состоянии.

Перед использованием электроприборов необходимо проверить надёжность крепления электророзетки, свериться с номиналом используемого напряжения.

Корпуса штепсельных розеток и выключателей не должны содержать трещин, оплавлений и других дефектов, способных снизить защитные свойства или нарушить надёжность контакта.

Кабели (шнуры) электропитания не должны содержать повреждений изоляции, сильных изгибов и скручиваний.

## 5.3 Экологическая безопасность

При проектирования и разработки указанной системы необходим ПЭВМ. В случае неисправной поломки или нестабильности ПЭВМ утилизируется. Составляющие его электронные компоненты оказывают серьезное воздействие на литосферу при их утилизации.

Федеральный закон №89 от 1998 г. «Об отходах производства и потребления» запрещает юридическим лицам самовольно избавляться от опасных отходов [59]. Этим видом деятельности, согласно постановлению Прави-

тельства РФ №340 от 2002 г., могут заниматься только специализированные структуры. В их число входят и фирмы, которые занимаются утилизацией электронных отходов. Обращение с отходами регламентируется ГОСТ Р 53692-2009 «Ресурсосбережение. Обращение с отходами» [60].

Вышедшее из строя ПЭВМ и сопутствующая оргтехника относится к IV классу опасности (малоопасные отходы) и подлежит специальной утилизации. Для оказания наименьшего влияния на окружающую среду, необходимо проводить специальную процедуру утилизации ПЭВМ и оргтехники, при которой более 90% отправится на вторичную переработку и менее 10% будут отправлены на свалки. При этом она должна соответствовать процедуре утилизации, как это указано в этапах технологического цикла отходов [60].

Люминесцентные лампы относят к ртутьсодержащим отходам, и для их утилизации действует Постановление Правительства РФ от 03.09.2010 №681 (ред. от 01.10.2013) «Об утверждении Правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде» [61]. Согласно постановлению, устанавливается порядок обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде.

## 5.4 Безопасность в чрезвычайных ситуациях

Наиболее вероятной чрезвычайной ситуацией при разработке системы мониторинга является пожар на рабочем месте. Причинами возгорания при работе с компьютером могут быть: токи короткого замыкания, неисправность устройства компьютера или электросетей; небрежность оператора при работе с компьютером; воспламенение ПЭВМ из-за перегрузки.

На основании ГОСТ 12.1.004-91 необходимо соблюдать следующие нормы пожарной безопасности [62]:

- в помещении должны находится средства тушения пожара, средства связи;
- электрическая проводка электрооборудования и осветительных приборов должна быть исправна;
- все сотрудники должны знать место нахождения средств пожаротушения, средств связи и номера экстренных служб;
- все сотрудники должны иметь компетенции по использованию указанных выше средств пожаротушения и связи.

В связи с возможностью возникновения пожара разработан следующий план действий [63].

- в случае возникновения пожара сообщить о нем руководителю,
- постараться устранить очаг возгорания имеющимися силами при помощи первичных средств пожаротушения (огнетушитель порошковый, углекислотный О-1П0 (з)-АВСЕ);
- привести в действие ручной пожарный извещатель, если очаг возгорания потушить не удается;
- сообщить о возгорании в службу пожарной охраны по телефону 01, 101 или 112; сообщить адрес, место и причину возникновения пожара;
- принять меры по эвакуации людей;
- встретить пожарную охрану, при необходимости сообщить всю необходимую информацию, оказать помощь.

Рабочее помещение, использованное при разработке системы, оборудовано в соответствии с требованиями пожарной безопасности: имеются порошковый огнетушитель, пожарная сигнализация и соответствующие средства связи.

## Заключение по разделу

В ходе выполнения работы над разделом «Социальная ответственность» были выявлены опасные и вредные факторы, воздействию которых может подвергнуться сотрудник, разрабатывающий систему автоматизированного мониторинга изменений земного покрова с использованием данных дистанционного зондирования Земли. Был проведен анализ нормативной документации.

Рабочее место, использованное при разработке системы, удовлетворяет требованиям безопасности. Выполняемая работа не сопряжена с высоким риском травматизма.

Освещение на рабочем месте соответствует нормам — используется несколько энергосберегающих ламп.

Уровни шума находятся в допустимых пределах — источником шума при эксплуатации ПК могут являться системы охлаждения и хранения постоянной памяти, однако уровень создаваемого ими шума находится в пределах нормы.

Микроклиматические условия соблюдаются за счет использования систем отопления и кондиционирования.

Защита от повреждений статическим электричеством обеспечивается путем защитного заземления и соблюдения правил безопасности на рабочем месте.

Во время работы делаются перерывы для снижения нагрузки и предотвращения нервно-психических перегрузок.

Помещение оборудовано согласно требованиям электробезопасности. В случае выхода из строев используемой электроники или ламп, отходы передаются в соответствующие компании.

# Список публикаций

[1] Котюбеев Р. Р. Применение правил для извлечения основных фактов из электронных медицинских записей на основе контекстно-свободной грамматики // Сборник избранных статей научной сессии ТУСУР. — 2021. — Май. — Принято к опубликованию.

[2] Котюбеев Р. Р. Извлечения значимых признаков из электронных медицинских записей на основе контекстно-свободной грамматики // Молодежь и современные информационные технологии: сборник трудов XVII Международной научно-практической конференции студентов, аспирантов и молодых учёных. — 2021. — Март. — Принято к опубликованию.

[3] Журман Д.А., **Котюбеев Р.Р.** Использование парсера yargy для извлечения значимых признаков из заключений осмотров лечащим врачом // Сборник избранных статей научной сессии ТУСУР. — 2020. — Май. — № 1-2. — С. 64–67.

[4] **Kotyubeev R.R.**, Zhurman D.A. Defining the states of the patients with erysipelas disease using different dimensionality reduction techniques // Молодежь и современные информационные технологии: сборник трудов XVII Международной научно-практической конференции студентов, аспирантов и молодых учёных, 17-20 февраля 2020 г., г. Томск / Томский политехнический университет. — 2020. — С. 93–94.

[5] Zhurman D.A., **Kotyubeev, R.R.** Machine learning model for evaluating the effectiveness of treatment for erysipelas // Молодежь и современные информационные технологии: сборник трудов XVII Международной научно-практической конференции студентов, аспирантов и молодых учёных, 17-20 февраля 2020 г., г. Томск / Томский политехнический университет. — 2020. — С. 152–153.

# Список литературы


[1] Приказ Минздрава РФ от 07.09.2020 № 947н «Об утверждении порядка организации системы документооборота в сфере охраны здоровья в части ведения медицинской документации в форме электронных документов». — М. : Минздрав России, 2021.

[2] Rao D., McMahan B. Natural language processing with PyTorch: build intelligent language applications using deep learning. — "O'Reilly Media, Inc.", 2019.

[3] Автоматическая обработка текстов на естественном языке и анализ данных : учебное пособие / Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Лукашевич Н.В. и Сапин А.С. — М. : Изд-во НИУ ВШЭ, 2017. — С. 269. — ISBN: 978–5–9909752-1-7.

[4] Grishman R. Information extraction // The Oxford handbook of computational linguistics / ed. by Mitkov R. — Oxford University Press, 2003.

[5] Moens M.-F. Information Extraction: Algorithms and Prospects in a Retrieval Context. — 2006. — Jan. — Vol. 21. — ISBN: 978-1-4020-4987-3.

[6] Appelt D., Israel D. J. Introduction to Information Extraction Technology // IJCAI 1999. — 1999.

[7] Nadeau D., Sekine S. A Survey of Named Entity Recognition and Classification // Lingvisticae Investigationes. — 2007. — Aug. — Vol. 30.

[8] Jurafsky D., Martin J. H. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition // Prentice Hall series in artificial intelligence. — 2000.

[9] Singhal A., Simmons M., Lu Z. Text mining for precision medicine: automating disease-mutation relationship extraction from biomedical literature //

Journal of the American Medical Informatics Association : JAMIA. — 2016. — Vol. 23 4. — P. 766–72.

[10] A framework for information extraction from tables in biomedical literature / Milosevic N., Gregson C., Hernandez R., and Nenadic G. // International Journal on Document Analysis and Recognition (IJDAR). — 2019. — Feb. — Vol. 22, no. 1. — P. 55–78. — Access mode: http://dx.doi.org/10.1007/s10032-019-00317-0.

[11] Miloševic N. A multi-layered approach to information extraction from tables in biomedical documents : Ph. D. thesis ; University of Manchester. — Manchester, 2018.

[12] Glossary extraction and utilization in the information search and delivery system for IBM Technical Support / Kozakov L., Park Y., Fin T., Drissi Y., Doganata Y., and Cofino T. // IBM Systems Journal. — 2004. — Feb. — Vol. 43. — P. 546 – 563.

[13] Shelmanov A., Smirnov I., Vishneva E. Information extraction from clinical texts in Russian // Computational Linguistics and Intellectual Technologies. — 2015. — P. 560–572.

[14] Ghoulam A., Barigou F., Belalem G. Information Extraction in the Medical Domain // Journal of Information Technology Research. — 2015. — Apr. — Vol. Aug. — P. 1–15.

[15] Маджаева С. И. Лингвистическая характеристика медицинского документа «История болезни» // Известия Волгоградского государственного педагогического университета. — 2011. — № 2. — С. 24–27.

[16] Collier N., Nobata C., Tsujii J. Extracting the names of genes and gene products with a hidden Markov model // COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics. — 2000.

[17] Exploring the boundaries: gene and protein identification in biomedical text / Finkel J., Dingare S., Manning C. D., Nissim M., Alex B., and Grover C. // BMC bioinformatics. — 2005. — Vol. 6, no. 1. — P. 1–9.

[18] Saha S. K., Sarkar S., Mitra P. Feature selection techniques for maximum entropy based biomedical named entity recognition // Journal of biomedical informatics. — 2009. — Vol. 42, no. 5. — P. 905–911.

[19] Grouin C. Biomedical entity extraction using machine-learning based approaches // substance. — 2014. — Vol. 6. — P. 1–611.

[20] Evaluating word representation features in biomedical named entity recognition tasks / Tang B., Cao H., Wang X., Chen Q., and Xu H. // BioMed research international. — 2014. — Vol. 2014.

[21] Recognition of medication information from discharge summaries using ensembles of classifiers / Doan S., Collier N., Xu H., Duy P. H., and Phuong T. M. // BMC medical informatics and decision making. — 2012. — Vol. 12, no. 1. — P. 1–10.

[22] Protein name tagging for biomedical annotation in text / Yamamoto K., Kudo T., Konagaya A., and Matsumoto Y. // Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine. — 2003. — P. 65–72.

[23] Detecting Gene Symbols and Names in Biological Texts A First Step toward Pertinent Information Extraction / Proux D., Rechenmann F., Julliard L., Pillet V., and Jacq B. // Genome Informatics. — 1998. — Vol. 9. — P. 72–80.

[24] Analysis of biomedical text for chemical names: a comparison of three methods. / Wilbur W. J., Hazard Jr G. F., Divita G., Mork J. G., Aronson A. R., and Browne A. C. // Proceedings of the AMIA Symposium / American Medical Informatics Association. — 1999. — P. 176.

[25] Annotation and disambiguation of semantic types in biomedical text: a cascaded approach to named entity recognition / Rebholz-Schuhmann D., Kirsch H., Gaudan S., Arregui M., and Nenadic G. // Proceedings of the 5th Workshop on NLP and XML: Multi-Dimensional Markup in Natural Language Processing. — 2006. — P. 11–18.

[26] Assessment of disease named entity recognition on a corpus of annotated sentences / Jimeno A., Jimenez-Ruiz E., Lee V., Gaudan S., Berlanga R., and Rebholz-Schuhmann D. // BMC bioinformatics / BioMed Central. — 2008. — Vol. 9. — P. 1–10.

[27] Bodenreider O. The unified medical language system (UMLS): integrating biomedical terminology // Nucleic acids research. — 2004. — Vol. 32, no. suppl_1. — P. D267–D270.

[28] Leaman R., Islamaj Doğan R., Lu Z. DNorm: disease name normalization with pairwise learning to rank // Bioinformatics. — 2013. — Vol. 29, no. 22. — P. 2909–2917.

[29] MedEx: a medication information extraction system for clinical narratives / Xu H., Stenner S. P., Doan S., Johnson K. B., Waitman L. R., and Denny J. C. // Journal of the American Medical Informatics Association. — 2010. — Vol. 17, no. 1. — P. 19–24.

[30] Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications / Savova G. K., Masanz J. J., Ogren P. V., Zheng J., Sohn S., Kipper-Schuler K. C., and Chute C. G. // Journal of the American Medical Informatics Association. — 2010. — Vol. 17, no. 5. — P. 507–513.

[31] Text mining for the Vaccine Adverse Event Reporting System: medical text classification using informative feature selection / Botsis T., Nguyen M. D., Woo E. J., Markatou M., and Ball R. // Journal of the American Medical Informatics Association. — 2011. — Vol. 18, no. 5. — P. 631–638.

[32] Технологии комплексного интеллектуального анализа клинических данных / Баранов А.А., Намазова-Баранова Л.С., Смирнов И.В., Девяткин Д.А., Шелманов А.О., Вишнёва Е.А., Антонова Е.В. и Смирнов В.И. // Вестник Российской академии медицинских наук. — 2016. — Т. 71, № 2.

[33] Singh S. Natural language processing for information extraction // arXiv preprint arXiv:1807.02383. — 2018.

[34] Chomsky N. Three models for the description of language // IRE Transactions on Information Theory. — 1956. — Vol. 2, no. 3. — P. 113–124.

[35] Loper E., Bird S. Nltk: The natural language toolkit // arXiv preprint cs/0205028. — 2002.

[36] Yargy-парсер [Электронный ресурс]. — 2021. — Режим доступа: https://github.com/natasha/yargy (дата обращения: 2021.05.26).

[37] Томита-парсер [Электронный ресурс]. — 2021. — Режим доступа: https://yandex.ru/dev/tomita/ (дата обращения: 2021.05.26).

[38] Earley J. An efficient context-free parsing algorithm // Communications of the ACM. — 1970. — Vol. 13, no. 2. — P. 94–102.

[39] Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts / ed. by Khachay M. Y., Konstantinova N., Panchenko A. et al. — Springer International Publishing, 2015. — Vol. 542 of Communications in Computer and Information Science. — P. 320–332.

[40] Репозиторий Natasha [Электронный ресурс]. — 2021. — Режим доступа: https://github.com/natasha (дата обращения: 2021.05.26).

[41] Решения по развитию прикладных информационных систем, входящих в ЕГИСЗ : Основные системотехнические решения по развитию «Системы ведения ИЭМК» // Единая государственная информационная система в сфере здравоохранения. — Минздрав России. — 2019. — книга 2 № 4. — С. 91.

[42] Patterns for Flask [Electronic resource]. — Flask documentation. — 2021. — Access mode: https://flask.palletsprojects.com/en/2.0.x/patterns/ (online; accessed: 2021.05.26).

[43] Security Considerations [Electronic resource]. — Flask documentation. — 2021. — Access mode: https://flask.palletsprojects.com/en/2.0.x/security (online; accessed: 2021.05.26).

[44] СП 2.4.3648-20 Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи. — М. : Главный санитарный врач РФ, 2020. — С. 54.

[45] Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 30.04.2021) (с изм. и доп., вступ. в силу с 01.05.2021). — М. : Собрание законодательства РФ, 2021. — С. 424.

[46] Федеральный закон «О персональных данных». Книга 2, ст. № 5. — 2006. — С. 28.

[47] ГОСТ 12.0.002-2014 Система стандартов безопасности труда — Взамен ГОСТ 12.0.002—80 ; введ. 30.09.2014. — М. : Изд-во стандартов, 2019. — С. 32.

[48] ГОСТ 22269-76 Рабочее место оператора. Взаимное расположение элементов рабочего места. Общие эргономические требования. — введ. 22.01.1978. — М. : Изд-во стандартов, 1976. — С. 32.

[49] Назаренко О.Б., Амелькович Ю.А. Безопасность жизнедеятельности: учебное пособие. — Томск : Издательство Томского политехнического университета, 2019. — С. 101.

[50] СанПиН 2.2.4.548-96 Гигиенические требованияк микроклимату производственных помещений — Взамен Санитарных норм микроклимата производственных помещений, утвержденных Минздравом СССР от 31 марта 1986 г. №4088-86 ; утвержден: 1.10.1996 г. — М. : Госкомсанэпиднадзор РФ, 1996. — С. 15.

[51] Р.2.2.2006-05 Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификация условий труда

— Взамен Р 2.2.755-99 ; введ. 01.11.2005. — М. : Главный санитарный врач РФ, 2005. — С. 142.

[52] СН 2.2.4/ 2.1.8.562-96 Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки. Санитарные нормы — введ. 31.10.1996. — М. : Госкомсанэпиднадзор России, 1996. — С. 13.

[53] ГОСТ 12.1.003-2014 Система стандартов безопасности труда. Шум. Общие требования безопасности — Взамен ГОСТ 12.1.003-83, ГОСТ 12.1.023-80 ; введ. 01.11.2015. — М. : Изд-во стандартов, 2014. — С. 34.

[54] ГОСТ 12.0.003–2015 ССБТ. Опасные и вредные факторы. Классификация — Взамен ГОСТ 12.0.003—7 ; введ. 09.06.2016. — М. : Изд-во стандартов, 2015. — С. 16.

[55] ГОСТ Р 50923-96 Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения — введ. впервые 10.06.1996. — М. : Изд-во стандартов, 2008. — С. 9.

[56] ГОСТ 12.2.032-78 Рабочее место при выполнении работ сидя. Общие эргономические требования — введ. 01.01.1979. — М. : Стандартинформ, 1986. — С. 9.

[57] ГОСТ Р ИСО 9241-5-2009 Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Требования к расположению рабочей станции и осанке оператора — введ. впервые 01.12.2010. — М. : Изд-во стандартов, 2010. — С. 28.

[58] ГОСТ 12.1.033-81 Система стандартов безопасности труда. Пожарная безопасность. Термины и определения — введ. 01.07.1982. — М. : Стандартинформ, 1988. — С. 9.

[59] Федеральный закон №89 от 1998 г. «Об отходах производства и потребления». Глава III, ст. № 9. — 1988. — С. 39.

[60] ГОСТ Р 53692-2009 Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов — введ. впервые 15.12.2009. — Москва : Стандартинформ, 2011. — С. 20.

[61] Постановление Правительства РФ от 03.09.2010 No 681 (ред. от 01.10.2013) «Об утверждении Правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде». — 2013.

[62] ГОСТ 12.1.004-91 Пожарная безопасность. Общие требования — Взамен ГОСТ 12.1.004-85 ; введ. 01.07.1992. — Москва : Стандартинформ, 2006. — С. 68.

[63] Федеральный закон «О пожарной безопасности» от 21 декабря 1994 №69-ФЗ. — 1994.

## Приложение А
## (обязательное)

## Natural language processing

Студент:

| **Группа** | **ФИО** | **Подпись** | **Дата** |
|---|---|---|---|
| 8ПМ9И | Котюбеев Роман Радиевич | | |

Руководитель ВКР:

| **Должность** | **ФИО** | **Ученая степень, звание** | **Подпись** | **Дата** |
|---|---|---|---|---|
| Доцент ОИТ ИШИТР | Аксёнов Сергей Владимирович | к.т.н. | | |

Консультант-лингвист отделения иностранных языков ШБИП:

| **Должность** | **ФИО** | **Ученая степень, звание** | **Подпись** | **Дата** |
|---|---|---|---|---|
| Доцент ОИТ ШБИП | Сидоренко Татьяна Валерьевна | к.пед.н. | | |

# A Natural language processing

*Natural language processing* (NLP) and *computational linguistics* (CL) are two areas of computational study of human language. CL employs computational methods to understand properties of human language. How do we understand language? How do we produce language? How do we learn languages? What relationships do languages have with one another?

Natural language processing, on the other hand, aims to develop methods to solve practical problems including information extraction. NLP researchers should have knowledge in math, linguistics, computer science, machine learning and deep neural networks.

Some of the NLP problems are machine translation, information extraction, question answering, summarization, sentiment analysis, automatic speech important for a particular person or organization, i.e. a disease for a doctor, from unstructured texts and converting it to the structured format. The structured data is easy to store and analyze. The applications that solve the information extraction problem can be integrated into many fields, including healthcare.

## A.1 Processing clinical documents

### A.1.1 Challenges of information extraction

Information extraction is similar to the traditional problem of finding relevant information. The key difference is that the result of information extraction is stored in the structured format, for example, JSON or XML. It makes the data more convenient to process in the future. A challenge of information extraction is that input texts are not annotated so the NLP researchers have to employ different heuristic and linguistic approaches. The following extracted and structured information is ready to use in data analysis.

Key terms should be clarified. An unstructured text is any kind of document that is written by a human and cannot be interpreted by a computer. In

fact, a computer only stores and displays information, but it does not understand what exactly is represented in it as a human does. On the other hand, a structured text is defined semantically and computationally. A computer interprets the structured information, and it can be applied to documents from other fields (generalization principle). The term "extraction" means that the text contains relevant semantic information presented in lexical forms such as word and phrase, in grammatical constructions such as sentence and tense, in rhetorical forms such as paragraph, section, chapter, etc.

Information extraction differs from summarization that outputs a summary of the text. However, information extraction can be the first step in summarization in which the summary sentence is further reduced to a string of relevant phrases similar to a newspaper headline.

Information extraction is the essential step to process the data. For example, the data extracted from police reports can be used in intellectual analysis to determine common tendencies of criminals, or it can be used in the decision-making system to define the next crime location. The data from electronic medical reports can be used to identify the diagnosis, collect the necessary information and also make the decision-making system.

### A.1.2 Information extraction sub-tasks

Information extraction includes several sub-tasks. Typical information extraction sub-tasks are:

- named entity recognition (NER),
- coreference resolution,
- relationship extraction,
- fact extraction.

The named entity recognition sub-task aims to find and classify "named entities" such as names, organizations, places, medical codes, tenses, numbers, currency, percentages, etc. in the unstructured document. The term "named

entity" implies that only some phrases have a reference. For instance, the sentence "the automotive company created by Henry Ford in 1903" is referred to as "Ford" or "Ford Motor Company." Name entity recognition helps to answer such questions as what happened, who did it, when, where, how and why.

Coreference resolution aims to find all references which refer to the same entity in the text. For example, if we have a sentence that contains the noun "Mr. Smith" and the following sentences mention it as pronouns (she, her), then the goal of coreference resolution is to find all references to this man. Coreference resolution helps to solve many NLP problems2 such as summarization, question-answer systems, chatbots.

Relationship extraction ai1ms to find named entities that are in the same semantic group. For instance, people, organizations, places can be split into categories: "married", "lives is in", "work in", etc.

Fact extraction aims to find and classify attributes, facts and events, for example, reviews, contacts, news, ads, diseases, numbers, etc. Fact extraction is the last step in processing the document, and the result is the structured data. Today there is no one-size-fits-all approach to solve this sub-task. Instead, NLP researchers try to solve particular problems of a subject area.

There are also specific information extraction sub-tasks such as table extraction, table information extraction, comments and terminology extraction.

In this work we solve fact extraction problem since the object of study is electronic medical records.

### A.1.3 Challenges of clinical documents

A sufficient amount of medical data is stored as unstructured texts. Electronic medical records consist of reports written in natural language, e.g., anamnesis, case reports, various results of diagnostics and many other types of records. Medical data is a great source of knowledge that is difficult to use because of lack of structure. Transforming it into an easy to calculate form helps to effectively keep medical history.

Clinical documents have many challenges for any information extraction approach. There are common challenges listed below.

**Non Standard Document Structure.** Medical documents have no fixed structure. They may be divided into sections however there is no standardization on the type of sections or their headings or contents. This depends on hospital to hospital, doctor to doctor.

**Medical Jargon.** Medical documents contain a large number of medical terms and jargon. NLP tools trained on non-medical domain data perform very poorly on medical data.

**Abbreviations.** The medical domain experiences an abundant use of abbreviations. Often the same abbreviation can be non-medical or medical or can expand to different terms based on the context and intention of the writer. Abbreviations are hard to normalize, classify or resolve.

**Polysemy and Synonymy.** A single medical term can represent two different ideas based on context. This is known as polysemy, e.g. “inflammation” may refer to a skin problem, a cellular level problem, a non medical activity etc. Further, a single concept canbe expressed through many different words. This is known as synonymy, e.g. “foetus”and “baby” mean the same in many medical contexts.

**Transcription Errors.** Most reports are dictated by doctors and typed by third-party. This introduces a wide array of transcription errors. Inaudible words are left as blanks. Apart from this the process of transcription also introduces a wide array of grammatical and casual spelling errors.

## A.2 Approaches of information extraction from clinical documents

Information extraction from clinical documents involves the extraction of medical terms and phrases from electronic records. Medical terms can include the names of diseases, procedures, medical devices, names of drug, complaints, complications, etc. Clinical objects can consist of one or more words that occur either sequentially or "through words"in the same sentence.

Statistical models and rule-based approaches are the main methods of information extraction from documents.

### A.2.1 Statistical models

Statistical models are a reliable tool that used in different NLP tasks. It is based on machine learning and therefore relies heavily on labeled training data. However, if there is enough data, then this method can show good results. Hidden Markov models, conditional random fields, and support vector machines are common models used to extract clinical facts.

Earlier work used a generative sequence labelling model such as hidden Markov models for clinical entity detection from text [16]. Transition probabilities between entity types and non-entities are used to make predictions along with the output probability of aunigram given its type.

There is also a maximum entropy Markov model (MEMM). This method allows the use of a wider variety of features [17]. Lexical features such as unigrams, suffixes, lemma are found to be influential. Further, linguistic features such as part of speech also play a role. Further, vocabulary-based approaches are used to create additional features.

Conditional Random Fields are another popular method. Features used with MEMM are also found suitable with CRF [19]. A number of orthographic features such as case information, presence of punctuation, etc, is also found to provide additional cues. CRFs overcome the label-bias problem faced by MEMMs and have been theoretically and empirically proven to be more robust and accurate in sequence of labeled documents.

Other works classify labeled sequences using Support Vector Machines. The authors of [21] applied combined listed above methods for information extraction.

### A.2.2 Rule-based approaches

Rule-based approaches can be split into two categories. The first employs linguistic methods to identify relevant facts. The second applies semantic features, gazetteers, lexemes, and methods based on regular expressions.

Rule-based approaches usually rely on parsing. Syntactical parsing is perfor-med and its output is post-processed using a number of hand-crafted rules to identify named entities. In particular, named entities tend to be noun phrases occurring at the subject (or sometimes object) position of sentences. Some works

perform a number of rule-based filtering steps to identify clinical entities. Another work performs sentence segmentation using rule based methods. They also perform a 2 stage approach where a rule-based system is followed by a classifier that identifies entity type. The author of [23] employs a number of stages of filtering using both rule-based and statistical models. He includes a statistical model in the rule-based system by making use of word frequency.

On the other hand, the second approach applies semantic networks and lexical features that are contained in clinical texts. For instance, the UMLS (Unified Medical Language System) integrates over 2 million names for some 900 000 concepts from more than 60 families of biomedical vocabularies, as well as 12 million relations among these concepts. Vocabularies integrated in the Metathesaurus.

Apache cTakes is a natural language processing system for extracting information from medical texts. It carries out tokenization, normalization, and definition of parts of speech within itself. The system solves problems: determining the essence, for example, whether the phrase refers to a symptom, anatomical term, diagnosis, etc., resolving coreferences, extracting relationships. cTAKES supports adding words to a dictionary. Various text documents of the patient can be collected to represent a single structure in the database.

Text mining for the Vaccine Adverse Event Reporting System: medical text classification using informative feature selection [31]. In this paper, the data set was a vaccination text report compiled by the Vaccine Adverse Event Reporting System. Tokenization, normalization, and definition of parts of speech were performed on the texts. The system proposed in the work defines the category of a medical term, for example, vomiting to the gastrointestinal class, face to anatomy. Each report was marked as positive or negative for machine learning. A total of 6034, of which 5797 are negative, 237 are positive. In the end, the text turned out with a positive or negative label.

The authors tried to solve the following problems: automatic diagnosis of chronic diseases in children, identifying the most significant features for diagnosing the disease, identifying hidden dependencies in clinical data by extracting

information from clinical texts. The data set was the results of examinations, medical history, examination results (ultrasound, ECG, etc.). Linguistic rules are used to extract medical terms (diseases, drugs, procedures) are recognized. After extraction, there is a classification according to the severity of the disease, the course of the disease, the establishment of a connection between diseases and parts of the body.

MedEx is a more recent clinical fact extraction tool which combines parsing, lexicon lookup and regular expressions to extract clinical facts from a text.

Thus, our work unlike the listed above is able to extract significant features from the texts of examinations written in Russian. Our system is much simpler since it does not include a lot of functionalities.

### A.2.3 How rule-based approaches work

Rule-based approaches can be defined using regular expressions, gazetteers or using a combined approach of linguistic rules, gazetteers and regular expressions.

The first method of the rule-based approach employs patterns of formal language so-called regular expressions. The strings that match these patterns are retrieved. For example, to find dates, we can determine patterns as `dd.mm.yy` or `dd.mm.yyyy` where `dd` is a day number (1 – 31), `mm` is a month number (1 – 12), `yy` is a year number (00 – 21), `yyyy` is a full year number (0000 – 2021). A year number makes difference between these patterns. Although regular expressions are a simple and fast tool, it has drawbacks — it is not impossible to enumerate all patterns. We can separate date numbers by slash or hyphen. Moreover, numbers are easy to extract but there are challenges when we try to apply regular expressions to words. Nevertheless, this approach has a wide range of practical uses.

Another method is to store possible lexical combinations of named entities in a list called the gazetteer. It can be applied to entities that have a finite state number. Although this method is accurate, it relies on collecting all lexical combinations so it is difficult to maintain.

The last method is to use linguistic rules with the two above methods together. One of the applications of linguistic rules is context-free grammar. For most human languages there are not only sequences of symbols but there are certain structures that define the sentences. These structures form grammar. A specific grammar uniquely defines a language, in general, the opposite is not true. A language can have many rules that describe it. A context-free grammar says that if words in a sentence can be replaced by the equivalents, the structure does not change. In this work, we apply context-free grammar to achieve our goals.

## A.3 Context-free grammar

If we know the sentence structure, we can generalize it. For instance, the sentence "Patient №1 complains of fever, cough" and the sentence "Patient №2 has complaints about fever, cough" has a similar structure — there is an enumeration of complaints follows the words "Patient", "complains", "complaints". The rules that catche this structure can be applied to find the complaints.

Any object can be specified in rules within context-free grammar. For example, we can focus on a sequence of adjectives, verbs or nouns, numbers, punctuation, etc. Moreover, we can use gazetteer and find certain words, including abbreviations. It is convenient if the words are normalized beforehand to not take into account all its forms.

Notice, if we have a sentence that has another structure that was not defined in the previous rules, we cannot extract given entities. The sentence "Patient №3 has a fever" does not contain the words "complains" or "complaints" so complaints will not be found. Such feature has similarity with regular expressions, but instead of, patterns we use rules that count the structure of the document.

### A.3.1 Context-free parsers

The rules form through a parser. One of the most popular English parsers is NLTK (natural language toolkit). Yargy and Tomita are parsers for the Russian language. Before we make a review of these parsers, let us give definitions of some NLP terms.

*A lexeme* is a unit of lexical meaning that underlies a set of words that are related through inflection. For example, “school”, “schools”, “schooling” has the same lexeme.

*A normal form* is a canonical or dictionary form of a word, i.e., singular nouns, infinitive verbs.

*A grammeme* in linguistics is a unit of grammar. For example, “singular” or “adverb”.

*A tag* is a set of grammemes that characterize a given word forms a tag. For example, the word “cats” has a tag defined as ‘noun, animated, plural’.

### A.3.2 Tomita parser

The Tomita parser was developed by Yandex and implemented in C ++. It is available as a binary file, but there are no ready-to-use grammars in the public domain. The Tomita parser uses its own language to describe grammars. The Tomita parser receives text in natural language as input, and then using gazetteer and the grammars, it transforms text into a set of structured data.

The grammars are a set of rules describing grammemes. The rule has left and right values, separated by the “->” symbol. There is a nonterminal on the left side and terminals and nonterminals on the right side .

A terminal is an object that has a specific, immutable value. Terminals can be lemmas, grammes, tags, punctuation symbols, special characters. Multiple terminals make up Tomita’s alphabet. Nonterminals are built from terminals, i.e. they can be words and phrases.

Tomita’s grammar does not interact directly with the parser, but through the root dictionary which is an entity that collects information about all grammars, gazetteers, additional files, etc. Compilation of grammars and extraction takes place through the console interface.

### A.3.3 Yargy parser

The Yargy parser is a completely open source project written in the Python programming language. The Yargy library is based on the Earley algorithm for parsing sentences in context-free grammar using dynamic programming.

The Yargy parser employs Pymorphy2 morphological analyzer to define the forms of the word. Pymorphy2 outputs several possible results of morphological information and lemmas for a input word, while the Tomita parser analyzer selects a single option based on the context.

Despite being completely based on Tomita, the rules are described in Python. Moreover, the Yargy repository contains a set of predefined rules to extract attributes such as addresses, names, money, dates, etc.

## Приложение Б
## (справочное)

## Дипломы конференций и конкурсов


Рисунок Б.1 — Диплом второй степени, «Научная сессия ТУСУР — 2021»

Рисунок Б.2 — Сертификат победителя хакатона «AI For Health, Microsoft — 2020»

## Приложение В
## (справочное)

## Скриншоты веб-севриса

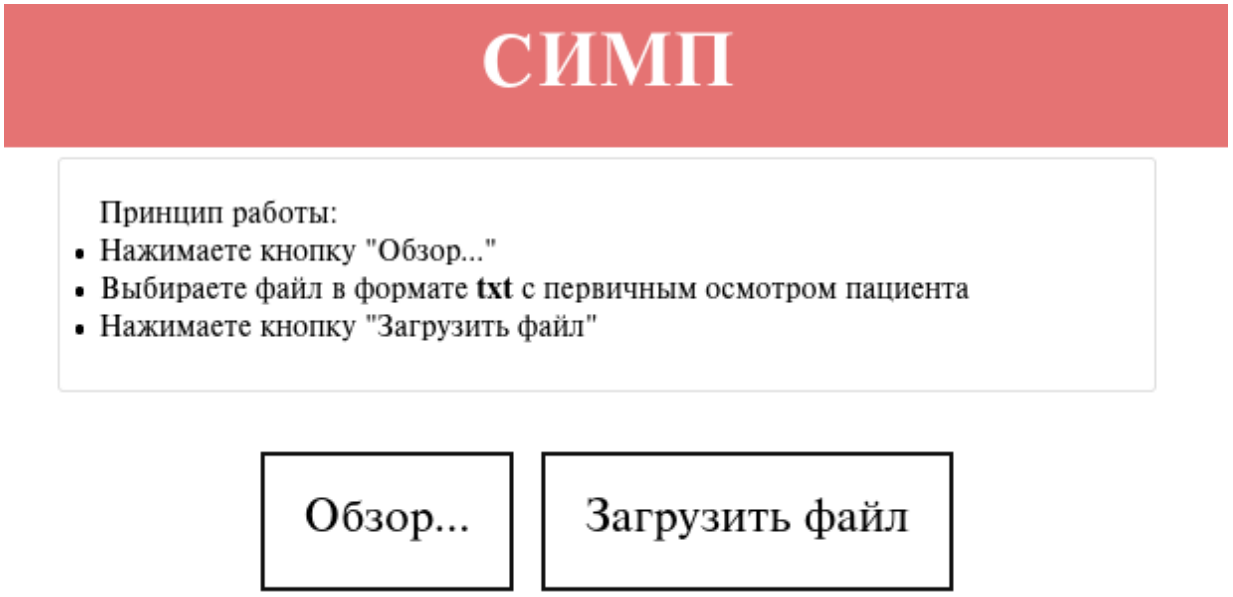


Рисунок В.1 — Скриншот главной страницы веб-сервиса


Рисунок В.2 — Скриншот страницы редактирования