

## **Интеллектуальная обработка клинических рекомендаций бронхолегочных заболеваний: программное решение для извлечения знаний**

Е. А. Коровин, С. А. Чиглинцева, Е. Ю. Сазонова, О. Н. Сметанина  
ФГБОУ ВО «Уфимский университет науки и технологий», Уфа, Российская Федерация

**Резюме:** Данное исследование было мотивировано необходимостью снижения времени поиска рекомендаций для лечения пациентов с заболеваниями органов дыхания за счет использования рекомендательной системы. Проблема состояла в ограниченности времени приема для врачей и сложности поиска справочной информации о бронхолегочных заболеваниях или клинических рекомендациях. Цель работы заключалась в разработке и реализации рекомендательной системы на основе инженерии знаний, способной упростить и ускорить поиск информации о лечении и профилактике данных заболеваний.

Для достижения поставленной цели был проведен аналитический обзор в области бронхолегочных заболеваний, включающий классификацию, методы диагностики и лечения, анализ клинических рекомендаций. Также были рассмотрены различные подходы к рекомендательным системам в медицине. Для поиска и отбора медицинских рекомендаций была выбрана модель дистрибутивной семантики word2vec как наиболее рациональная. Был разработан алгоритм, который сочетал поиск совпадающих предложений по регулярным выражениям и модель поиска дистрибутивно-семантических связей word2vec.

В результате исследования были выявлены ограничения в выборе решений для предобработки и генерации медицинских клинических рекомендаций, а также зависимость от иностранных программных продуктов. Был создан алгоритм, способный генерировать индивидуальные рекомендации для пациентов на основе анализа совпадающих предложений по регулярным выражениям. Также были разработаны архитектура и структура базы данных для хранения информации о пациентах, записях приема и диагнозах.

Разработанная рекомендательная система с интегрированным алгоритмом дистрибутивной семантики word2vec значительно сокращает время, затрачиваемое врачами на поиск справочной информации и может быть полезной для специалистов медицинской отрасли. Работа имеет практическую ценность и может служить основой для дальнейших исследований и развития автоматизированных систем в медицине.

**Ключевые слова:** рекомендательные системы, машинное обучение, медицина, здравоохранение, информационные системы, web-приложение, обработка естественного языка, извлечение знаний, методы анализа текстовых данных, бронхолегочные заболевания, дистрибутивная семантика

**Для цитирования:**

### **Введение**

Бронхолегочные заболевания значительно влияют на здоровье и качество жизни, вызывая высокую смертность и инвалидность. Исследование в этой области выявило

сложности диагностики и лечения, обусловленные различными факторами. Недостаток информации усложняет принятие правильных решений врачами, а медицинские ошибки при таких заболеваниях имеют серьезные последствия. Разработка рекомендательной системы может помочь врачам принимать обоснованные решения и улучшить результаты лечения.

В настоящее время по оценкам Всемирной организации здравоохранения, более 300 миллионов человек страдают от бронхиальной астмы, а от ХОБЛ – более 200 миллионов (Источник: <https://www.who.int/news-room/fact-sheets/detail/asthma>). В 2019 году было зарегистрировано 461 000 смертей от астмы. Проблема данного заболевания актуальна для всех стран, не зависимо от уровня развития [1] .

Разработанное программное обеспечение для диагностики и выбора тактики лечения заболеваний легких имеет широкий спектр применения. Оно может быть использовано в медицинских учреждениях для улучшения диагностических мероприятий, а также при проведении научных исследований и в учебном процессе. Реализация результатов работы в задачах лечения и диагностики бронхолегочных заболеваний значительно повышает эффективность диагностических процедур и работы врачей.

### **Сравнительный анализ методов машинного обучения**

Сложности эффективного поиска информации в медицинских документах представляют значительную проблему для врачей и медицинского персонала, особенно в условиях ограниченного времени и ресурсов. Кроме того, каждый источник информации может иметь свою индивидуальную структуру и формат, что затрудняет единое понимание и использование.

Поскольку у нас ограниченная доступность данных (около 100 документов), в создании системы были задействованы подходы обработки естественного языка и поиск по регулярным выражениям. Ручная предобработка данных в исправлении ошибок и проблем некорректного форматирования документов из формата PDF значительно повысила точность и надежность результатов исследований, несмотря на затраты времени и усилий.

Проведен сравнительный анализ методов для получения рекомендаций, где были рассмотрены три подхода: поиск по регулярным выражениям, генеративный подход и интеллектуальный поиск.

#### **Метод поиска по регулярным выражениям:**

Метод основан на формальных правилах, заданных в виде регулярных выражений. Регулярные выражения представляют собой последовательности символов, которые используются для определения шаблонов в тексте. Это позволяет осуществлять поиск и извлечение информации, исходя из сопоставления шаблона с текстом. Этот метод полезен для выполнения простых операций, таких как извлечение шаблонных данных или поиск слов или фраз, соответствующих определенным критериям.

#### **Генеративный подход:**

Генеративный подход базируется на моделировании вероятностных распределений для генерации новых данных, которые могут быть похожи на обучающий набор. В контексте обработки текста и естественного языка, генеративные модели могут быть обучены на большом объеме текстовых данных и использоваться для генерации новых текстовых примеров. Рекуррентные нейронные сети, такие как LSTM (Long Short-Term Memory), часто используются для моделирования зависимостей между словами и генерации последовательностей текста. Генеративные модели могут быть применены для создания новых текстовых данных, например, для

генерации стихов, создания диалоговых систем или автоматического формирования ответов.

#### **Интеллектуальный поиск:**

Интеллектуальный поиск включает в себя применение методов машинного обучения и искусственного интеллекта для обработки поисковых запросов и предоставления релевантных результатов. Этот подход обычно используется в системах поиска информации, где задача состоит в нахождении наиболее подходящих документов или ресурсов на основе запроса пользователя.

Интеллектуальный поиск представляет собой широкий спектр методов, основанных на применении различных моделей машинного обучения. Среди этих методов можно выделить модели ранжирования, такие как BM25 и TF-IDF, а также нейронные сети, включая сверточные нейронные сети (CNN) и трансформеры. Целью интеллектуального поиска является предоставление пользователю наиболее релевантных результатов в ответ на его поисковый запрос. Это достигается за счет анализа текста, классификации и ранжирования документов с использованием методов машинного обучения.

Каждый из этих методов обладает своими уникальными преимуществами и ограничениями в контексте задач извлечения информации из текста и обработки естественного языка. В таблице представлено более подробное сравнение данных методов.

Таблица – Сравнение методов получения рекомендаций

Методы	Интерпретируемость	Исходный размер датасета	Время поиска решения	Точность решения
Поиск по регулярным выражениям	+	Размеченный корпус от 1000 рекомендаций	Менее секунды	Не измеряется (возьмем за нормальную)
Интеллектуальный поиск	+	Размеченный корпус от 1000 рекомендаций	Менее секунды	Превышает нормальную
Генеративный подход	-	Минимальный размер корпуса превышает 1 000 000 рекомендаций	2-3 секунды	Зависит от размера датасета (не достигает минимальной)

В данном исследовании рассматривается задача определения смысловой и контекстной близости слов в медицинских клинических рекомендациях. У нас есть набор предложений в качестве входных данных, однако объем данных ограничен, что исключает возможность использования глубоких нейронных сетей. Поэтому наш

подход заключается в применении методов, способных дать хорошие результаты при ограниченных ресурсах.

Одной из ключевых задач является классификация задачи дистрибутивной семантики, фокусирующейся на смысловой близости слов [4]. Хорошо подготовленные данные позволяют рассмотреть модель, основанную на архитектуре Skip-gram, которая входит в семейство моделей word2vec.

Первоначальные методы извлечения таксономических отношений из текстов, основанные на шаблонах, обладали низкой полнотой, так как требовали определенных конструкций в ограниченном числе предложений. Однако появление векторных представлений слов, эмбедингов, предоставило новые возможности для извлечения знаний из текстов. Векторные представления формируются из контекстов, в которых слова упоминаются, и сходство контекстов приводит к сходству векторных представлений слов. Это позволяет автоматически определять семантическую близость слов на основе текстовых коллекций и значительно повышает точность извлечения таксономических отношений [5].

Метод word2vec позволяет представлять слова в виде числовых векторов на основе их контекста. Этот метод использует две матрицы: матрицу  $W$  для центральных слов и матрицу  $D$  для контекстных слов (1). Размерности этих матриц зависят от размера словаря и размерности векторов. Обучение модели word2vec основано на методе максимального правдоподобия (2).

$$W, D \in \mathbb{R}^{Vocab \times EmbSize} \quad (1)$$

$$\sum_{CenterW_i} P(CtxW_{-2}, CtxW_{-1}, CtxW_{+1}, CtxW_{+2} | CenterCtxW_i; W, D) \rightarrow \max_{W, D} \quad (2)$$

Метод Word2Vec стремится предсказать распределение соседних слов в заданном окне, учитывая параметры модели. Для этого слова в окне представляются через произведение более простых распределений. Эти распределения определяют вероятность встретить контекстное слово рядом с центральным (3). Для моделирования категориального распределения, которое является дискретным и принимает значения из фиксированного набора, используется функция softmax (4).

$$P(CtxW_{-2}, CtxW_{-1}, CtxW_{+1}, CtxW_{+2} | CenterCtxW_i; W, D) = \prod_j P(CtxW_j | CenterW_i; W, D) \quad (3)$$

$$P(CtxW_i | CenterW_i; W, D) = \frac{e^{w_i \cdot d_j}}{\sum_{j=1}^{|V|} e^{w_i \cdot d_j}} = softmax \simeq \frac{e^{w_i \cdot d_j^+}}{\sum_{j=1}^k e^{w_i \cdot d_j}}, k \ll |V| \quad (4)$$

Сходство слов в модели Word2Vec вычисляется через скалярное произведение векторов центрального и контекстного слов. Однако вычисление полного softmax по всем словам словаря может быть вычислительно неэффективным. Вместо этого используется метод отрицательного сэмплирования (negative sampling), который позволяет сократить количество вычислений softmax путем выбора случайных слов. Таким образом, обучение Word2Vec сводится к обучению классификатора, который предсказывает, могут ли два слова встретиться вместе.

Идея Skip-gram в Word2Vec заключается в предсказании контекстных слов на основе центрального слова. Этот подход эффективно моделирует взаимосвязи между словами и позволяет использовать полученные векторы для различных задач обработки естественного языка, таких как кластеризация слов, поиск похожих слов и машинный

перевод. Обучение Word2Vec модели оптимизируется путем минимизации функции потерь с помощью стохастического градиентного спуска.

Исследования и анализы показывают, что модель Word2Vec успешно захватывает семантические и синтаксические аспекты слов [6]. Она способна обучаться на больших объемах текстовых данных и порождать семантически богатые векторные представления слов. В связи с этим Word2Vec становится одним из самых популярных методов для работы с естественным языком и извлечения смысловой информации из текстов.

### Описание алгоритма для поиска рекомендаций

Программное обеспечение выпускной квалификационной работы состоит из скрипта Preprocessing.ipynb. Его можно разбить на три основных этапа: предварительная обработка, обучение модели word2vec и поиск рекомендаций.

Во время предварительной обработки требуется нормализовать текст рекомендаций и сохранить их в фрейм данных. В качестве входных данных поступает file.txt - документ, содержащий клинические рекомендации.

Алгоритм предобработки текста:

Шаг 1. Загрузка модулей и текста из файла file.txt.

Шаг 2. Разделение текста на предложения.

Шаг 3. Нормализация слов в каждом предложении.

Шаг 4. Перевод представления предложения в список нормализованных слов

Шаг 4. Создание двух фреймов данных df:

- с двумя столбцами: «Предложения» и «Нормализованный текст», который представлен на рисунке
- со столбцом «Предложения» на рисунке.

	Предложения	Нормализованный текст
0	Хроническая обструктивная болезнь легких (ХОБЛ...	[хронический, обструктивный, болезнь, лёгкий, ...
1	Обострения и коморбидные состояния являются не...	[обострение, коморбидный, состояние, являться,...
2	Курение остается основной причиной ХОБЛ.	[курение, оставаться, основной, причина, хобл]
3	По некоторым оценкам, в индустриальных странах...	[по, некоторый, оценка, индустриальный, страна...
4	В развивающихся странах существенное повреждаю...	[в, развивающийся, страна, существенный, повре...
5	Этиологическую роль также могут играть професс...	[этиологический, роль, также, мочь, играть, пр...

Рисунок - Фрейм данных fileFull.csv

	Предложения
0	Хроническая обструктивная болезнь легких (ХОБЛ...
1	Обострения и коморбидные состояния являются не...
2	Курение остается основной причиной ХОБЛ.
3	По некоторым оценкам, в индустриальных странах...
4	В развивающихся странах существенное повреждаю...

Рисунок – Фрейм данных filePart.csv

Шаг 5. Сохранение фреймов данных под именами «fileFull.csv» и «filePart.csv».

В данном исследовании для поиска рекомендаций используется методика "обогащения" знаний, которая начинается с поиска прямых совпадений по исходному

пользовательскому запросу. Затем этот список рекомендаций расширяется за счет дополнительных вариантов, определенных на основе смысловой близости слов. Такой подход позволяет эффективно увеличить число рекомендаций, особенно в случаях, когда исходный запрос возвращает небольшое количество результатов.

Для генерации дополнительных рекомендаций используется обученная модель word2vec архитектуры Skip-gram. Эта модель обучается на больших текстовых наборах данных, выявляя контекстуальные связи между словами. В результате модель строит векторные представления, где семантически близкие слова имеют близкие векторы. Таким образом, при запросе пользователя модель может искать семантически близкие слова для расширения списка рекомендаций. Эти методы позволяют эффективно расширять перечень рекомендаций на основе смысловой близости слов и контекстуальных связей в текстовых данных.

На этапе поиска входными данными будут являться диагноз, степень его тяжести и дополнительные сведения, а в качестве выходных данных будут клинические рекомендации. Алгоритм поиска рекомендаций:

Шаг 1. Приведение запроса пользователя к виду списка нормализованных слов.

Шаг 2. Поиск прямых совпадений по имеющемуся корпусу рекомендаций.

*Если обнаружены совпадения, то совпадения сохраняются в виде множества потенциальных рекомендаций.*

Шаг 3. Поиск «близких» по смыслу и контексту слов по диагнозу, с использованием обученной модели

Шаг 4. Отбор рекомендаций содержащих «близкие» к исходному диагнозу слова.

Шаг 5. Объединение множества потенциальных рекомендаций и множества рекомендаций по контексту

### **Аналитический обзор систем принятия решений и инструментальные средства разработки веб-приложений**

При обзоре существующих средств было выявлено, что UpToDate, Dynamed и KlinikalKey предоставляют справочную информацию и быструю оценку симптомов для консультирования, как и симптом-чекеры такие как Symptomate, Ubie AI Symptom Checker, Ada, Babylon Health, WebMD, Glass AI. Перечисленные приложения предназначены для быстрой оценки симптомов и предоставления рекомендаций, но не позволяют вести учет пациентов и их медицинских историй. Для этой цели используются специализированные информационные системы для врачей, такие как электронные медицинские записи (EMR) и электронные медицинские архивы (EHR), которые обеспечивают безопасное хранение и управление медицинскими данными пациентов в соответствии с требованиями конфиденциальности и защиты персональных данных. Разработка программного решения обусловлена отсутствием функциональности существующих систем ведения учета пациентов и их медицинской истории.

Для выполнения поставленных задач было решено использовать библиотеки NLTK и PyMorphy2 для высокой точности и надежности различных видов анализа текста, поскольку в медицинских рекомендациях и пособиях редко требуется извлекать и анализировать именованные сущности, какие возможности предоставляет библиотека Natasha [2, 3].

Для создания рекомендательной системы была выбрана среда разработки VS Code ввиду ее удобного интерфейса и широкой поддержки расширений,

обеспечивающих разнообразные инструменты для работы с Python и его библиотеками, включая отладку, автодополнение кода и управление зависимостями.

Для реализации проекта был выбран фреймворк Django, который предоставляет готовые компоненты, такие как аутентификация, маршрутизация и работа с базами данных, упрощая процесс разработки. Для базы данных проекта было решено выбрать MySQL, поскольку она легко интегрируется с интерфейсом API, обладает простотой использования, гибкостью и высокой производительностью.

### **Описание функциональных возможностей, структуры и интерфейса программного обеспечения**

Данное программное обеспечение обладает широким спектром функциональных возможностей, включая авторизацию пользователей, возможность осуществления поиска пациентов и добавления новых записей в базу данных, а также изменение имеющейся информации о пациентах. Также система позволяет осуществлять поиск и просмотр диагнозов и записей приема, добавление новых записей приема и генерацию текстовых рекомендаций. Редактирование формы с клиническими рекомендациями и сохранение рекомендаций после внесения изменений в буфер обмена также доступны в данной системе.

Программа для анализа и генерации текста состоит из следующих функций:

`normalize_word(word)` нормализует слово при помощи модуля `rumorphy2` и извлекает нормальную форму слова для дальнейшей обработки;

`process_sentence(sentence)` обрабатывает предложение, которое токенизируется при помощи `nltk.word_tokenize`, затем каждое слово нормализуется при помощи `normalize_word`. Нормализованные слова, не являющиеся стоп-словами, сохраняются в списке и возвращаются;

`search_sentences(query)` ищет предложения в фрейме данных, содержащие все слова из заданного запроса (`query`). Для каждой строки в фрейме данных проверяется, содержатся ли все слова из запроса в нормализованном тексте. Если совпадения найдены, предложение добавляется в список `found_sentences`. Если совпадения не найдены, возвращается сообщение об ошибке.

Схема базы данных приложения представлена на Рисунке.

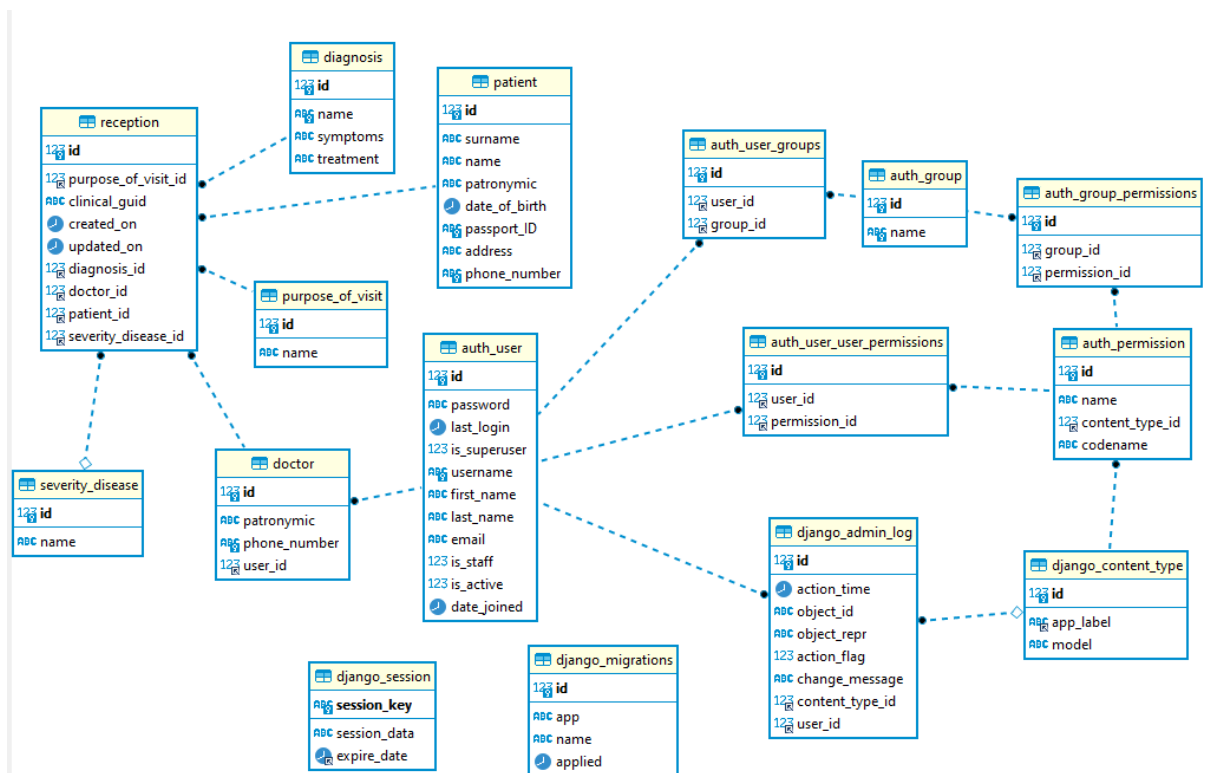


Рисунок – Схема базы данных

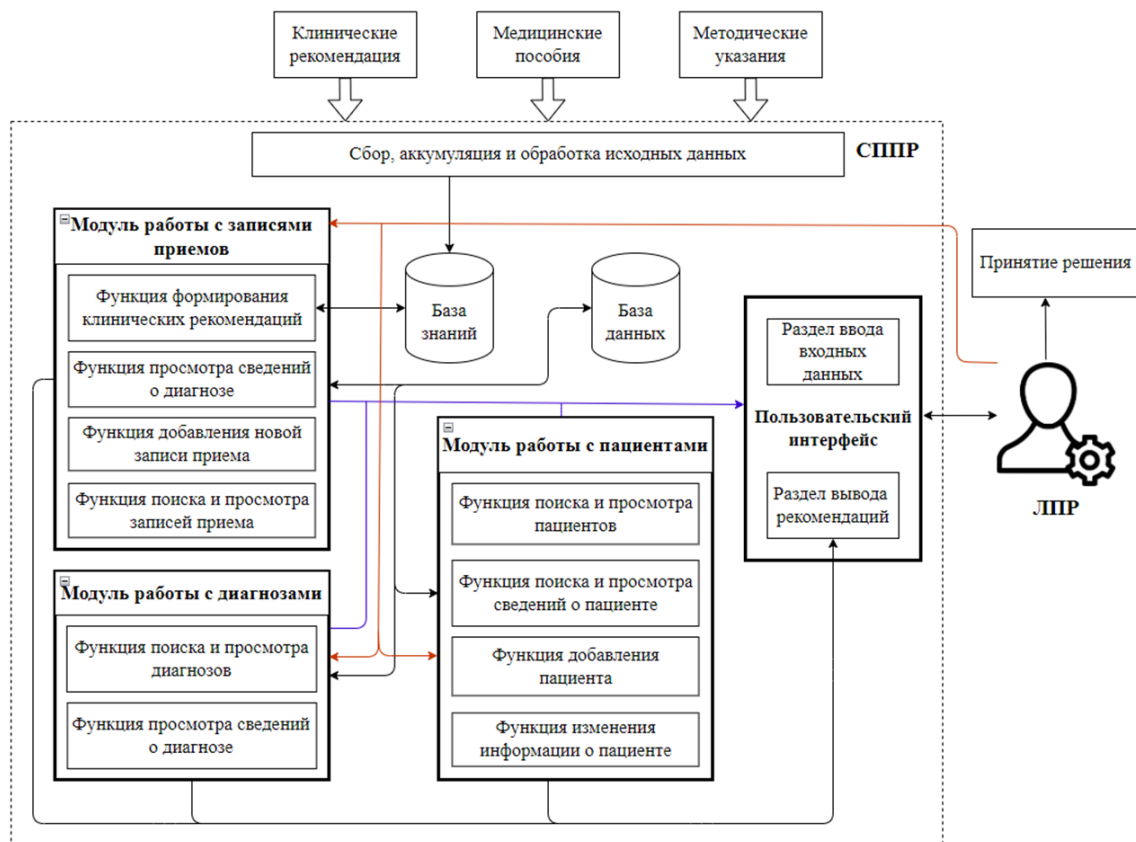
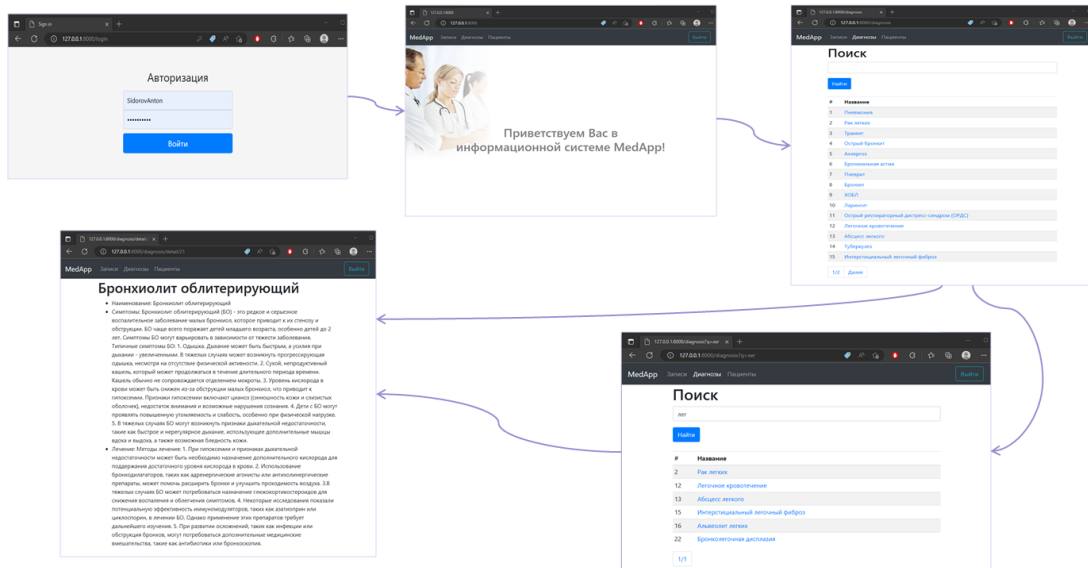


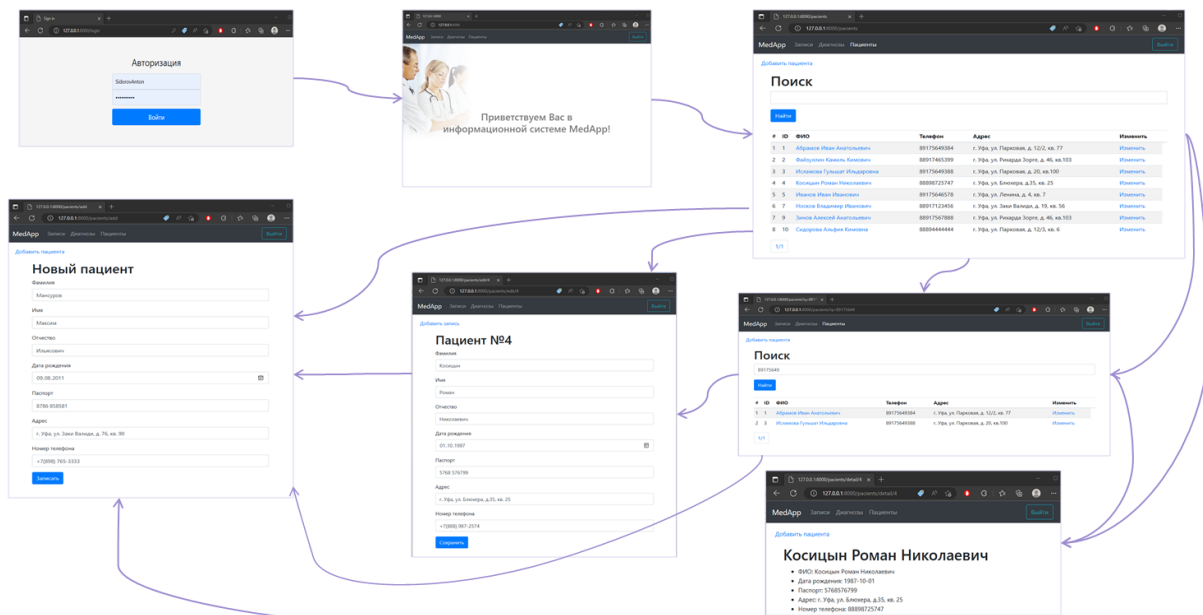
Рисунок - Структурная схема системы поддержки принятия решений при работе приложения



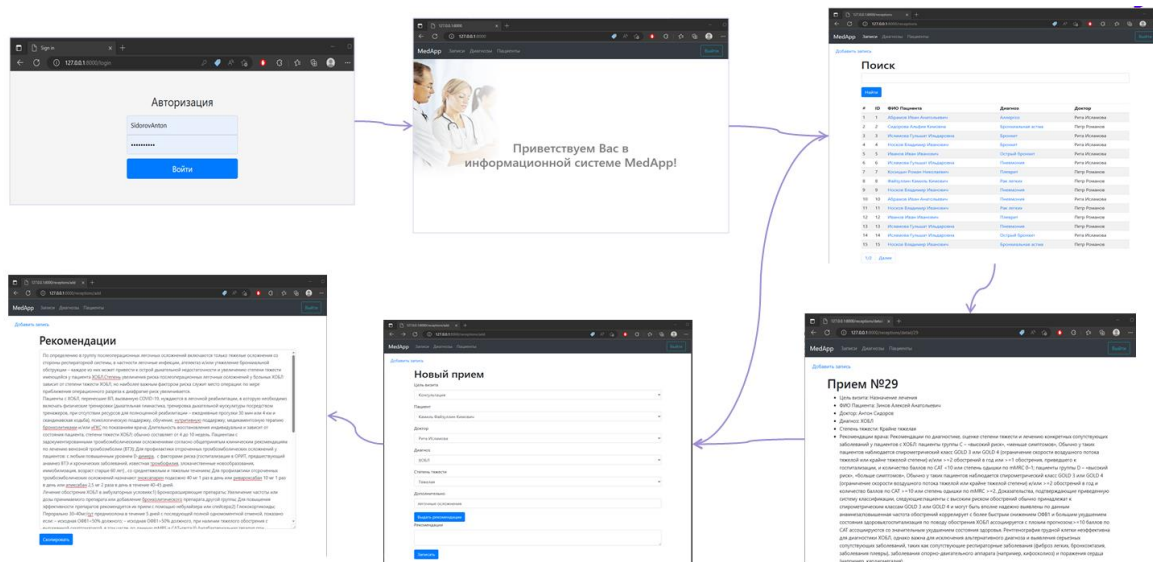
Логин и пароль создает администратор системы в администрировании Django, вручную внося данные о пользователях системы и генерируя уникальные логин и пароль для них. Неавторизованный пользователь может просматривать только страницу авторизации, а авторизованный пользователь может видеть все страницы. Для работы с записями приема, или диагнозами, или пациентами авторизованный пользователь с приветственной страницы выбирает в верхнем меню раздел, в котором необходимо совершить действия.



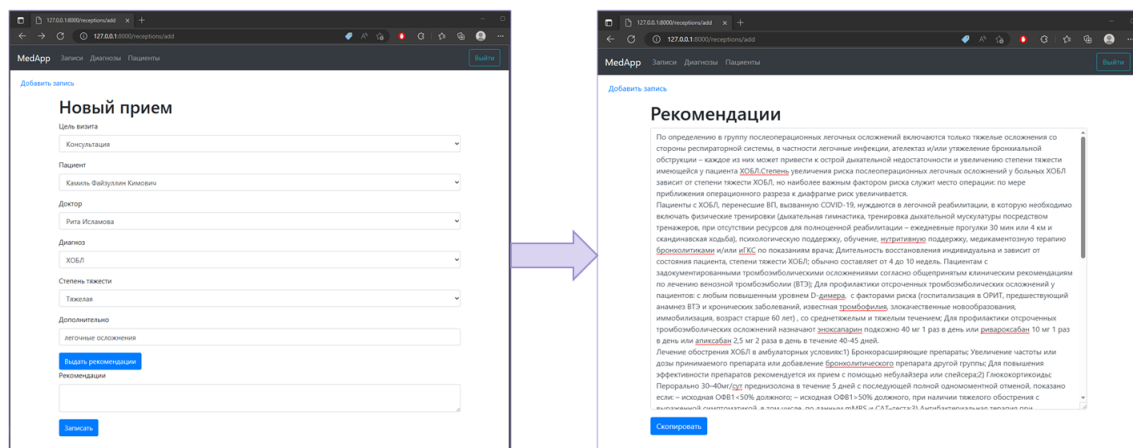
### Сценарий работы приложения в разделе «Диагнозы»



### Сценарий работы приложения в разделе «Пациенты»



## Сценарий работы приложения в разделе «Записи»



## Пример генерации рекомендаций Результаты вычислительных экспериментов

В ходе исследования были реализованы три метода: поиск по регулярным выражениям, генеративный подход и интеллектуальный поиск, и проанализированы временные затраты и релевантность выдаваемых рекомендаций. Результаты эксперимента приведены в таблице.

Таблица – Результаты вычислительного эксперимента

Метод	Временные затраты	Релевантность
Поиск по регулярным выражениям	Ожидание выдачи ответа и генерации текста занимает до 3-х секунд	В качестве результата выводится текст по нужной тематике, содержащий введенные слова в различных формах и последовательностях.

Генеративный подход	Ожидание выдачи ответа и генерации текста занимает до 2 минут	В качестве результата выводится несвязный текст с некорректной грамматикой.
Интеллектуальный поиск	Ожидание выдачи ответа и генерации текста занимает до 5-ти секунд	В качестве результата выводится текст, содержащий введенное слово и близкие по контексту к нему термины.

Для генеративного подхода требуется большой объем данных для обучения, поэтому в данном эксперименте этот метод показал неэффективный результат, пример полученного результата представлен на рисунке.

```
prompt = 'температура'
length = 500
recommendation = generate_text(model, tokenizer, prompt, length)
print(recommendation)
```

The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Pl Setting `pad\_token\_id` to `eos\_token\_id`:50256 for open-end generation.  
температуру Эдытра плодовео заещступии: утлия просмер в дазаботчееск задь инік. Наеде бребы роспуй вбрениета.

#### Рисунок – Генеративный подход

Интеллектуальный поиск требует обработки больших текстовых наборов данных и выделяет контекстуальных связей между словами, однако для существующего корпуса предложений он дал релевантные ответы. Для более точного поиска по тексту рекомендуется использовать сочетание методов интеллектуального поиска и поиска по регулярным выражениями.

В русском языке существуют различные группы родственных слов, имеющие разные формы и контекстные значения. Тезаурусы не лишены ошибок, наиболее распространенные из них включают пропущенные или неправильные связи между словами, устаревшие значения, отсутствие новых или разговорных выражений. Векторные представления слов позволяют извлекать новые знания из текстовых коллекций, открывая перспективы для улучшения автоматической обработки текстов на естественном языке [7].

Также было проведено сравнение временных затрат на поиск клинических рекомендаций в документе вручную, с помощью навигации и с помощью разработанного в ходе выпускной квалификационной работы алгоритма. Результаты эксперимента приведены в Таблице.

Таблица – Сравнение временных затрат на поиск рекомендаций вручную, с помощью навигации и алгоритма

Способ	Временные затраты
Вручную	В среднем человек может просмотреть и оценить содержимое каждой страницы примерно за 10-15 секунд. Для документа с 100 страницами потребуется примерно до 16-25 минут.

Навигация в программах по просмотру документа	Для опытного пользователя встроенной навигации время поиска информации в документе с 100 страницами может составлять до 5-10 минут. Однако поиск осуществляется по определенной последовательности ключевых слов или символов, а совместно встречающиеся слова в одном предложении могут не учитываться, если они расположены не друг за другом и имеют иные формы.
Разработанный алгоритм	Время поиска информации клинических рекомендаций в документе с 100 страницами может составлять до 1-2 минут.

По результатам эксперимента можно сделать вывод, что с применением автоматизированного поиска с учетом индивидуальных особенностей, существенно сократилось время, затрачиваемое врачами на поиск справочной информации о бронхолегочных заболеваниях или клинических рекомендаций, связанных с лечением пациента. Эти факторы способствуют повышению эффективности работы врачей.

### Заключение

Для достижения улучшения процесса поиска рекомендаций для лечения пациентов с заболеваниями органов дыхания была разработана рекомендательная система, основанная на инженерии знаний.

В ходе исследования были рассмотрены различные подходы к рекомендательным системам в медицине, а также изучены особенности бронхолегочных заболеваний, их классификация, методы диагностики и лечения, анализировались клинические рекомендации в этой области. В области бронхолегочных заболеваний был проведен аналитический обзор, включающий классификацию, методы диагностики и лечения, анализ клинических рекомендаций. Полученные знания послужили основой для создания механизма генерации рекомендаций и разработки рекомендательной системы.

В результате анализа были выявлены ограничения в выборе решений для предобработки и генерации медицинских клинических рекомендаций, а также зависимость от иностранных программных продуктов.

Для поиска и отбора медицинских рекомендаций была выбрана модель дистрибутивной семантики word2vec как наиболее рациональная. Для этой модели был разработан алгоритм, который сочетал поиск совпадающих предложений по регулярным выражениям и модель поиска дистрибутивно-семантических связей word2vec. Такой подход позволил упростить и ускорить поиск информации о лечении и профилактике бронхолегочных заболеваний для специалистов медицинской отрасли. Это, в свою очередь, положительно сказалось на решении проблемы ограниченного времени приема для врачей.

Авторами работы был создан алгоритм, который генерирует индивидуальные рекомендации для пациентов. Для этого использовался алгоритм поиска совпадающих предложений по регулярным выражениям.

Также были разработаны архитектура и структура базы данных для хранения информации о пациентах, записях приема и диагнозах. Для удобства использования системы врачами и медицинским персоналом было разработано веб-приложение с интуитивным интерфейсом. Реализация проекта осуществлялась с использованием следующих технологий: IDE VS Code, DBeaver для работы с MySQL, язык программирования Python 3.9, фреймворк Django для создания веб-приложения,

инструментарий Bootstrap для создания GUI, а также различные библиотеки, такие как datetime, pyperclip, nltk, os, pandas, pymorphy2, для обработки данных и анализа текста.

Для оценки качества разработанного ПО было проведено тестирование на основе экспериментальных данных и анализ результатов по ГОСТ 28195-89. Анализ эффективности показал, что применение автоматизированного поиска с учетом индивидуальных особенностей значительно сокращает время, затрачиваемое врачами на поиск справочной информации о бронхолегочных заболеваниях или клинических рекомендациях по лечению пациента.

Работа выполнена при финансовой поддержке РНФ (проект No РНФ 22-19-00471).

### СПИСОК ИСТОЧНИКОВ

1. Вафина А.Р. Оценка ведения больных бронхиальной астмой в условиях реальной клинической практики: проспективное наблюдательное исследование Москва, 2022
2. Жаббарова Р.У., Бурнашев Р.Ф. Инструментарий обработки лингвистической информации. Science and Education. 2023. №4. [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/instrumentariy-obrabotki-lingvisticheskoy-informatsii>.
3. Шульман В.Д., Максименко О.Е., Волхонцева П.Д.. Анализ программных средств морфологического анализа Международный журнал гуманитарных и естественных наук. №3-2. 2022. [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/analiz-programmnyh-sredstv-morfologicheskogo-analiza>
4. Шахмаметова Г.Р., Зулкарнеев Р.Х., Евграфов А.А. Методы обработки текстовых данных в системе принятия клинических решений при диагностике болезней органов дыхания. Информационные технологии интеллектуальной поддержки принятия решений (ITIDS'2019). Труды VII Всероссийской научной конференции (с приглашением зарубежных ученых): в 3 томах. Том 2. Уфа : УГАТУ, 2019. - С. 245-248.
5. Тихомиров Михаил Михайлович Методы автоматизированного пополнения графов знаний на основе векторных представлений Москва — 2022
6. Baroni M., Lenci A. (2011), How we BLESSed distributional semantic evaluation, Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics, Association for Computational Linguistics, Edinburgh, pp. 1–10.
7. Loukachevitch N. V. (2019), Corpus-based Check-up for Thesaurus, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5773–5779.

### ИНФОРМАЦИЯ ОБ АВТОРАХ

**Коровин Евгений Алексеевич**, бакалавр, кафедра Вычислительной математики и кибернетики, Уфимский университет науки и технологий, Уфа, Российская Федерация  
*e-mail:* [arkvinst@gmail.com](mailto:arkvinst@gmail.com)

**Чиглинцева Светлана Андреевна**, бакалавр, кафедра Вычислительной математики и кибернетики, Уфимский университет науки и технологий, Уфа, Российская Федерация  
*e-mail:* [s\\_chiglintseva@inbox.ru](mailto:s_chiglintseva@inbox.ru)

**Сазонова Екатерина Юрьевна**, кандидат технических наук, доцент, кафедра Вычислительной математики и кибернетики, Уфимский университет науки и технологий, Уфа, Российская Федерация  
*e-mail:* [ekaterina\\_rassadnikova@mail.ru](mailto:ekaterina_rassadnikova@mail.ru)

**Сметанина Ольга Николаевна**, доктор технических наук, доцент, кафедра Вычислительной математики и кибернетики, Уфимский университет науки и технологий, Уфа, Российская Федерация  
*e-mail:* [smoljushka@mail.ru](mailto:smoljushka@mail.ru)