МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ШКОЛА КОМПЬЮТЕРНЫХ НАУК

Кафедра программного обеспечения

**ОТЧЁТ О РАЗРАБОТКЕ MVP**

версия проекта №1

СЕРВИС АГРЕГАЦИИ ТЕКСТОВ НАУЧНЫХ СТАТЕЙ НА РУССКОМ ЯЗЫКЕ ПО УГЛЕРОДНОЙ ТЕМАТИКЕ С ИСПОЛЬЗОВАНИЕМ ТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

02.03.03 Математическое обеспечение и администрирование информационных систем

Профиль «Технологии программирования и анализа больших данных»

|  |  |  |
| --- | --- | --- |
| Выполнили работу  (групповой проект)  студентки 4 курса  очной формы обучения |  | Кобылкина Анна Андреевна  Марочкина Виктория Витальевна |
| Руководитель  Старший преподаватель |  | Глазкова Анна Валерьевна |

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc185546886)

[ГЛАВА 1. ИЗУЧЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ 5](#_Toc185546887)

[1.1. АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ И АНАЛОГОВ 5](#_Toc185546888)

[1.2. ОСНОВНЫЕ МЕТОДЫ ТЕМАТИЧЕСКОГО АНАЛИЗА ТЕКСТОВ 6](#_Toc185546889)

[1.3. МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ 9](#_Toc185546890)

[1.4. МЕТРИКИ ОЦЕНКИ КАЧЕСТВА ТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ 11](#_Toc185546891)

[ГЛАВА 2. СОЗДАНИЕ МЕХАНИЗМА СБОРА И ХРАНЕНИЯ ДАННЫХ НАУЧНЫХ СТАТЕЙ, ПОСТРОЕНИЕ МОДЕЛИ ТЕМАТИЧЕСКОГО АНАЛИЗА ТЕКСТОВ 15](#_Toc185546892)

[2.1. СТРУКТУРА БАЗЫ ДАННЫХ ДЛЯ ХРАНЕНИЯ ИНФОРМАЦИИ О НАУЧНЫХ СТАТЬЯХ 15](#_Toc185546893)

[2.2. СБОР И ПЕРВИЧНАЯ ОБРАБОТКА ДАННЫХ 16](#_Toc185546894)

[2.3. ПОСТРОЕНИЕ И ОЦЕНКА ТЕМАТИЧЕСКИХ МОДЕЛЕЙ 21](#_Toc185546895)

[2.4. ВИЗУАЛИЗАЦИЯ РЕЗУЛЬТАТОВ ТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ 28](#_Toc185546896)

[ГЛАВА 3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ И ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА 30](#_Toc185546897)

[3.1. АРХИТЕКТУРА ВЕБ-ПРИЛОЖЕНИЯ И ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ 30](#_Toc185546898)

[3.2. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ 32](#_Toc185546899)

[3.3. РЕАЛИЗАЦИЯ ПОИСКА С ИСПОЛЬЗОВАНИЕМ ELASTICSEARCH 32](#_Toc185546900)

[3.4. СОЗДАНИЕ СЕРВЕРНОГО API 33](#_Toc185546901)

[3.5. ОПИСАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА 34](#_Toc185546902)

[3.6. ТЕСТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ 41](#_Toc185546903)

[ЗАКЛЮЧЕНИЕ 45](#_Toc185546904)

[СПИСОК ЛИТЕРАТУРЫ 46](#_Toc185546905)

[ПРИЛОЖЕНИЯ 1-7 48](#_Toc185546906)

# ВВЕДЕНИЕ

В различных научных областях накопление большого объёма информации приводит к трудностям в её поиске, систематизации и анализе. Учёным становится сложно быстро находить релевантные данные и выявлять ключевые направления исследований. Эта проблема характерна для многих научных дисциплин. В данном проекте она рассматривается на примере углеродной тематики.

В условиях роста выбросов парниковых газов и глобальных климатических изменений [13], учёные ищут новые подходы к снижению уровня CO₂ в атмосфере. Карбоновые полигоны, такие как Тюменский карбоновый полигон, становятся площадкой для исследования углеродного баланса и разработки решений для минимизации климатических рисков. Однако специалисты, такие как экологи, биологи и химики, сталкиваются с рядом трудностей:

* Разрозненность данных: научные статьи по углеродной тематике находятся в разных источниках, что затрудняет их поиск.
* Трудоёмкость систематизации: для поиска публикаций необходимо вручную использовать ключевые слова, такие как "углеродный баланс" или "поглощение CO₂", что требует значительных временных ресурсов.
* Сложности анализа: после сбора информации её обработка и выделение ключевых тем требуют дополнительных усилий и инструментов.

Цель: разработка сервиса для систематизации и анализа текстов научных статей на русском языке по углеродной тематике с использованием методов тематического моделирования.

Основные функции системы:

* Поиск научных публикаций по базе данных с использованием ключевых слов и разделов, интересующих исследователей.
* Анализ собранных данных с помощью тематического моделирования для определения ключевых исследовательских направлений.
* Представление результатов в виде интерактивных графиков.

Для достижения цели были поставлены следующие задачи:

1. Изучение алгоритмов тематического моделирования для анализа текстов научных статей.
2. Разработка архитектуры приложения для автоматизированного сбора, анализа и визуализации данных по углеродной тематике.
3. Создание структуры базы данных для хранения информации о научных статьях, их метаданных и результатов анализа тематического моделирования.
4. Реализация механизма сбора и агрегирования научных статей на русском языке, включая фильтрацию данных по ключевым словам и тематическим разделам.
5. Сравнение и выбор методов тематического моделирования для анализа текстов по углеродной тематике.
6. Построение тематической модели для анализа собранных текстов с целью выявления ключевых исследовательских направлений.
7. Разработка API для предоставления доступа к собранным данным и результатам анализа.
8. Создание веб-интерфейса для визуализации результатов тематического моделирования, включая интерактивные графики и облака слов.
9. Тестирование всех компонентов системы, включая сбор данных, работу тематической модели, API и пользовательский интерфейс, для обеспечения корректности работы и удобства использования.

Назначение разработки: использование сервиса сотрудниками Тюменского карбонового полигона. Наше приложение поможет учёным сосредоточиться на интерпретации данных, минимизируя временные затраты на их сбор и предварительный анализ.

# ГЛАВА 1. ИЗУЧЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ

## 1.1. Анализ существующих решений и аналогов

Для оценки существующих решений в области поиска и агрегации научных материалов, а также для выявления подходящих аналогов, были рассмотрены несколько популярных платформ, предоставляющих доступ к научным публикациям.

eLibrary — одна из крупнейших российских научных электронных библиотек, предоставляющая пользователям доступ к обширному количеству научных публикаций, журналов и материалов конференций. Платформа поддерживает поиск по различным параметрам, таким как ключевые слова, авторы и журналы, и предлагает функции для отслеживания цитируемости и индексации научных изданий. Также eLibrary предоставляет доступ как к открытым материалам, так и к статьям по подписке.

КиберЛенинка — российская платформа для бесплатного доступа к научным и учебным материалам. Платформа предоставляет пользователям возможность поиска по ключевым словам, авторам и названиям статей. КиберЛенинка фокусируется на обеспечении открытого доступа к знаниям и является важным источником для академического сообщества, особенно для тех, кто ищет свободный доступ к научным исследованиям и образовательным ресурсам.

Проведём сравнение сервисов по критериям (Таблица 1). Существующие решения, такие как eLibrary и КиберЛенинка, предоставляют удобные инструменты для поиска и получения научных материалов, однако они не предлагают специализированных возможностей для тематического моделирования и анализа научных трендов в узких областях, таких как углеродные технологии. Эти платформы могут быть полезными для начального поиска информации, однако для анализа и извлечения актуальных исследовательских тем необходимо внедрение специализированных методов обработки текстовых данных и тематического моделирования.

Таблица

Сравнение сервисов

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | eLibrary | КиберЛенинка | Наше приложение |
| Поиск статей | да | да | да |
| Индексация научных журналов | да | нет | нет |
| Методы анализа тем (тематическое моделирование) | нет | нет | да |
| Визуализация тем и трендов | нет | нет | да |
| Доступ к полным текстам статей | да | да | нет (только ссылки на исходные материалы) |

## 1.2. Основные методы тематического анализа текстов

Тематическое моделирование представляет собой набор методов, направленных на выявление скрытых тематических структур в текстовых данных. Известные способы тематического моделирования включают в себя группу алгебраических моделей (LSA, NMF) и вероятностных моделей (pLSA , LDA). В последние годы появился новый класс тематических моделей, которые комбинируют вероятностные процессы и модели распределенных векторов (Top2Vec, BERTopic)[6].

Вероятностные модели тематического анализа основаны на предположениях о вероятностных распределениях слов в темах и тем в документах. Эти модели позволяют выявлять скрытые темы, учитывая статистические закономерности в корпусе текстов. Темы всех слов в документе предполагаются независимыми [4].

Латентное распределение Дирихле (LDA).

Принципы работы [3,4,5]:

* 1. Представление документов с помощью bag-of-words.
  2. Каждый документ представляет собой смесь тем.
  3. Каждая тема характеризуется распределением вероятностей слов.
  4. Генерация слов происходит путем выбора темы из распределения тем документа и затем слова из распределения слов темы.

1. Преимущества [1]:
   1. LDA позволяет выявить скрытые темы в текстовых данных, которые могут быть не очевидны при поверхностном чтении.
   2. Модель обобщает документы на уровне тем, что полезно для анализа и категоризации больших объемов текстов.
   3. Результаты LDA часто легко интерпретировать, так как каждая тема связана с набором характерных слов.
2. Ограничения [1,4]:
   1. Требует предварительного задания числа тем .
   2. Предположение о независимости слов внутри тем.
   3. Результаты могут сильно зависеть от исходных параметров и инициализации.

В комбинированных моделях к векторным представлениям применяются процедуры кластеризации со снижением размерности и ранжированием слов для формирования тем. Эти методы лучше учитывают контекстуальные и семантические особенности текста [6].

BERTopic.

1. Принципы работы [6, 8, 12]:
   1. Представление документов: Использование моделей, таких как BERT или Sentence Transformers, для преобразования документов в векторные представления​.
   2. Снижение размерности: Применение алгоритма UMAP для уменьшения размерности эмбеддингов.
   3. Кластеризация: Использование алгоритма HDBSCAN для группировки векторов эмбеддингов в кластеры.
   4. Извлечение ключевых слов: Определение наиболее характерных слов для каждой темы на основе частотности и важности слов в кластере c помощью с-TF-IDF, MMR.
2. Преимущества [8]:
   1. Учет семантических связей между словами и документами.
   2. Гибкость в определении числа тем (не требует предварительного задания ).
3. Ограничения [8]:
   1. Зависимость от качества используемых моделей эмбеддингов.
   2. Необходимость настройки параметров UMAP и HDBSCAN для требуемых в задаче результатов.

Проведём сравнение вероятностных моделей и комбинированных моделей (Таблица 2) и определим, какие методы будем рассматривать для решения нашей задачи.

Таблица

Сравнительный анализ вероятностных моделей и комбинированных моделей

|  |  |  |
| --- | --- | --- |
| Критерий | Вероятностные модели (LDA) | Комбинированные модели (BERTopic) |
| Представление текста | мешок слов (bow) | векторные эмбеддинги (bert, sentence transformers) |
| Необходимость задания числа тем | да (требуют ) | нет(автоматическое определение) |
| Учет порядка слов | нет | да (через эмбеддинги) |
| Семантическая глубина | низкая(основывается на частотах слов) | высокая(захватывает контекст и семантику) |
| Вычислительная сложность | высокая при больших данных | высокая для эмбеддингов, но оптимизирована для кластеризации |

Для решения задачи тематического моделирования в рамках данного проекта мы рассмотрим оба подхода: вероятностные модели и комбинированные модели. LDA выбран как представительный пример вероятностных моделей, подходит для работы с небольшими и средними корпусами. BERTopic выбран как пример комбинированных моделей, позволяет учитывать порядок слов и более точно захватывать семантическую глубину. Мы проведем эксперименты с обоими методами и выберем наиболее подходящий для решения задачи, учитывая характеристики корпуса данных и требуемую точность результатов.

## 1.3. Математическая постановка задачи

Входные данные.

Корпус текстов на русском языке, который представляет собой набор документов , где — количество документов, а — текст документа. Каждый документ состоит из последовательности слов , где ​ — количество слов в документе ​, а — слово из словаря , где — размер словаря.

Документы могут быть представлены в двух формах:

1. Мешок слов (Bag-of-Words, BoW): каждый документ представляется как вектор частот слов , где — количество вхождений слова в документ ​.
2. Векторное представление (Embeddings): для методов, основанных на векторных представлениях текста, каждый документ может быть представлен как вектор ​ в пространстве признаков, полученном с помощью эмбеддингов.

Задача.

Необходимо выделить набор тематических групп , где — количество тем, а каждая тема ​ характеризуется набором слов , представляющих центральные слова для данной темы.

1. В LDA каждая тема описывается вероятностным распределением по словам, где — это вероятность того, что слово ​ принадлежит теме​.
2. В BERTopic каждая тема также характеризуется набором слов ​, но вместо вероятностей используется вес слов, который отражает их значимость для данной темы в процессе кластеризации. Эти веса могут быть интерпретированы как степень важности слова для темы .

Для каждого документа ​ необходимо определить его принадлежность к одной или нескольким темам. В зависимости от используемого метода, эта принадлежность может быть представлена различными способами:

1. В LDA определяются вероятности принадлежности документа к каждой теме , что позволяет моделировать распределение тем в документе через комбинацию тематических распределений.
2. В BERTopic документу присваивается один или несколько кластеров (тем), которые наилучшим образом отражают его содержание. В этом случае каждая тема может быть представлена как кластер документов, объединённых по схожести.

Выходные данные.

Результатом работы модели тематического моделирования является набор тематических групп , каждая из которых представляет собой кластер документов, относящихся к определённому аспекту углеродной тематики. Каждая тема определяется группой слов ​, характерных для соответствующей области. Дополнительно для каждого документа определяется его принадлежность к темам, которая может быть представлена либо распределением (для вероятностных моделей), либо назначением к кластерам (для кластеризующих методов).

## 1.4. Метрики оценки качества тематического моделирования

Для оценки качества тематического моделирования используются метрики, каждая из которых позволяет выявить различные аспекты работы модели. В нашем случае будут рассмотрены следующие метрики:

1. Когерентность c\_v [9].

Когерентность темы c\_v​ измеряет, насколько логично и согласованно расположены слова в теме. Она основана на усреднении нормализованных PMI (Pointwise Mutual Information) значений между словами, описывающими тему. Она сочетает в себе четыре шага: сегментацию, оценку вероятностей, вычисление меры подтверждения (NPMI) и агрегирование.

* Диапазон значений: от 0 до 1, где 1 указывает на очень высокую когерентность темы (т.е. тема логична и состоит из слов, которые часто встречаются вместе в документах), а 0 — на низкую когерентность.
* Высокое значение когерентности означает, что слова в теме тесно связаны между собой, что делает тему более интерпретируемой и понятной.

1. Перплексия [2, 5].

Перплексия является важной метрикой для вероятностных моделей, таких как LDA. Она измеряет, насколько хорошо модель предсказывает новые данные (документы). Перплексия рассчитывается как экспонента средней отрицательной логарифмической вероятности для всех слов в тестовом наборе, что позволяет измерить степень неопределенности модели в отношении данных.

* Диапазон значений: перплексия всегда положительна, и чем ниже ее значение, тем лучше модель (меньше неопределенности и ошибок в предсказаниях).
* Низкое значение перплексии указывает на то, что модель хорошо предсказывает вероятности появления слов в новых данных.

Формула для расчёта перплексии выглядит следующим образом:

где:

— количество документов в корпусе;

​ — количество слов в документе ;

— вероятность слова в документе , рассчитанная моделью.

1. Разделимость[11].

Разделимость (separability) измеряет, насколько хорошо темы могут быть разделены друг от друга, т.е. насколько различны топики, выделенные моделью. Разделимость оценивает степень, в которой темы не пересекаются, и насколько они хорошо отделены друг от друга. Это важно для того, чтобы темы не были слишком похожи и не содержали схожих слов.

* Диапазон значений: измеряется в диапазоне от 0 до 1. Значения, близкие к 1, означают высокую разделимость (темы хорошо отделены друг от друга), а значения, близкие к 0, указывают на низкую разделимость (темы схожи и не могут быть легко различимы).
* Разделимость полезна при оценке качества кластеризации и в тех случаях, когда важно, чтобы каждая тема была достаточно уникальной.

1. Перекрытие[11].

Перекрытие измеряет, насколько сильно перекрываются различные темы. Это может быть важно для оценки качества кластеризации, особенно если модели необходимо выделять четкие, но взаимосвязанные темы. Перекрытие помогает выявить, насколько сильно темы пересекаются.

* Диапазон значений: от 0 до 1. Значение 0 указывает на отсутствие перекрытия (в документе нет тем, вероятность которых превышает заданный порог), а значение 1 означает, что все темы в документе имеют вероятность выше порога, что говорит о полном перекрытии.
* Перекрытие полезно для понимания, насколько пересекаются темы и где есть возможность для улучшения разделения.

1. Экспертная оценка [10].

Экспертная оценка измеряет интерпретируемость тем, выделенных моделью. Она основывается на мнении экспертов, которым предоставляются топ-10 слов и словосочетаний для каждой темы, упорядоченные по убыванию степени принадлежности. Экспертам предлагается определить, можно ли дать теме обобщённое название, что свидетельствует о её осмысленности и последовательности. Оценка рассчитывается как отношение числа интерпретированных тем к общему числу тем.

* Диапазон значений: от 0 до 1. Значение 0 означает, что ни одна из тем не была признана интерпретируемой, тогда как значение 1 указывает на то, что все темы могут быть интерпретированы и названы.
* Экспертная оценка используется для понимания практической применимости и качества выделенных тем. Она дополняет автоматические метрики, оценивая интерпретируемость и осмысленность тем с точки зрения человека. Это важно для оценки, насколько темы подходят для реальных задач, таких как анализ научных данных, и для повышения доверия пользователей к результатам модели.

Для вероятностной модели LDA будут применяться все метрики, для BERTopic, основанной на эмбеддингах и кластеризации, будут использоваться: когерентность c\_v​, чтобы оценить связность ключевых слов в темах, перекрытие, чтобы понять степень пересечения между темами и экспертная оценка. Метрика перплексии не используется, так как она применяется только к вероятностным моделям, а разделимость не является релевантной из-за природы кластеризации в BERTopic, где темы часто пересекаются, и слова могут принадлежать сразу нескольким темам, что снижает значение разделимости как оценки качества.

# ГЛАВА 2. СОЗДАНИЕ МЕХАНИЗМА СБОРА И ХРАНЕНИЯ ДАННЫХ НАУЧНЫХ СТАТЕЙ, ПОСТРОЕНИЕ МОДЕЛИ ТЕМАТИЧЕСКОГО АНАЛИЗА ТЕКСТОВ

## 2.1. Структура базы данных для хранения информации о научных статьях

Для хранения данных о научных статьях, а также результатов их анализа, была разработана структура базы данных, построенная на основе документоориентированной модели хранения данных. Для этого используется Elasticsearch — нереляционное хранилище документов в формате JSON, в котором был создан индекс article (Приложение 1).

Основной объект хранения — это документ, представляющий статью с метаданными. Описание структуры данных представлена в таблице 3.

Таблица 3

Описание структуры данных о научных статьях

|  |  |  |
| --- | --- | --- |
| Название поля | Тип данных | Описание |
| title | строка | название статьи |
| authors | массив строк | список авторов статьи |
| journal | строка | журнал, в котором опубликована статья |
| year | целое число | год публикации |
| field\_of\_science | строка | раздел, к которому относится статья |
| keywords | объект, содержащий 2 поля (тип: строка) | ключевые слова на русском и английском языках |
| annotation | строка | аннотация статьи |
| citations | объект, содержащий 2 поля (тип: строка) | ссылки на цитирование в форматах гост и apa |
| link | ключевое слово | ссылка на источник статьи |
| topic\_number | целое число | идентификатор темы, определённый в результате тематического моделирования |

## 2.2. Сбор и первичная обработка данных

На первом этапе был проведён сбор научных статей с сайта КиберЛенинка за период 2014–2024 годов.

Список тематических разделов, связанных с углеродной тематикой, был предоставлен экспертами. В него вошли такие направления, как биологические науки, химические технологии, экологические биотехнологии и другие. Полный список тематических разделов представлен в приложении 2. На основе каждого раздела были извлечены ссылки на статьи и годы их публикации. Эти данные использовались для последующего извлечения метаданных статей, что позволило сформировать набор данных из более чем 211 тысяч статей (Рисунок 1).

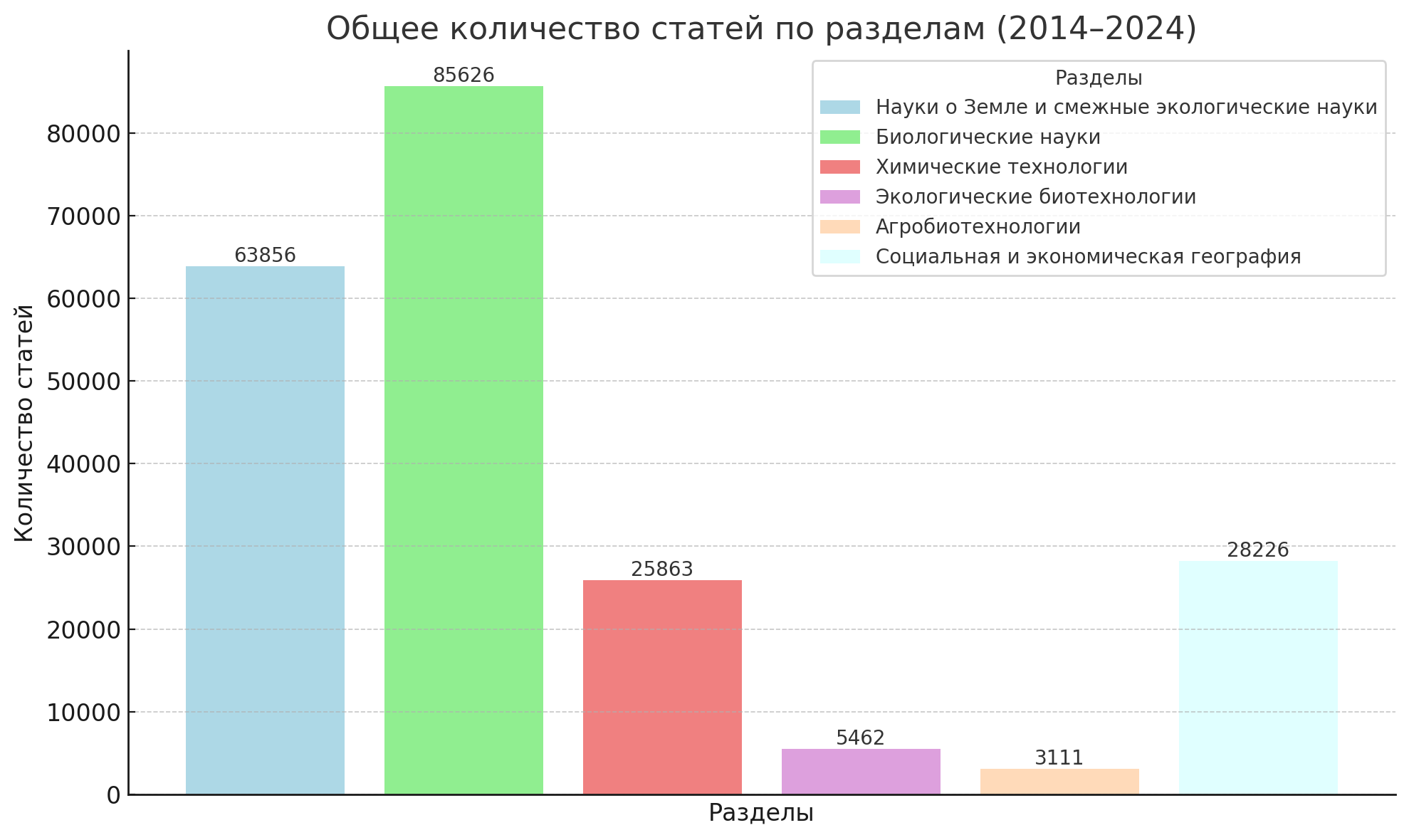


Рис. 1. Количество статей для каждого раздела.

Второй этап состоял из извлечения ключевых метаданных статей из собранных ссылок:

* Название статьи.
* Авторы.
* Журнал, в котором опубликована статья.
* Год публикации.
* Раздел, к которому относится статья.
* Ключевые слова на русском и английском языках.
* Аннотация статьи.
* Ссылки на цитирование в форматах ГОСТ и APA.

Процесс включал:

* Автоматизированный парсинг страниц для извлечения данных с помощью библиотеки BeautifulSoup, которая использовалась для работы с HTML-контентом.
* Фильтрацию статей с некорректными символами (например, украинскими) или без русских букв в названии.
* Проверку данных на полноту и корректность.

В результате обработано более 109 тысяч статей с полными метаданными (Рисунок 2).

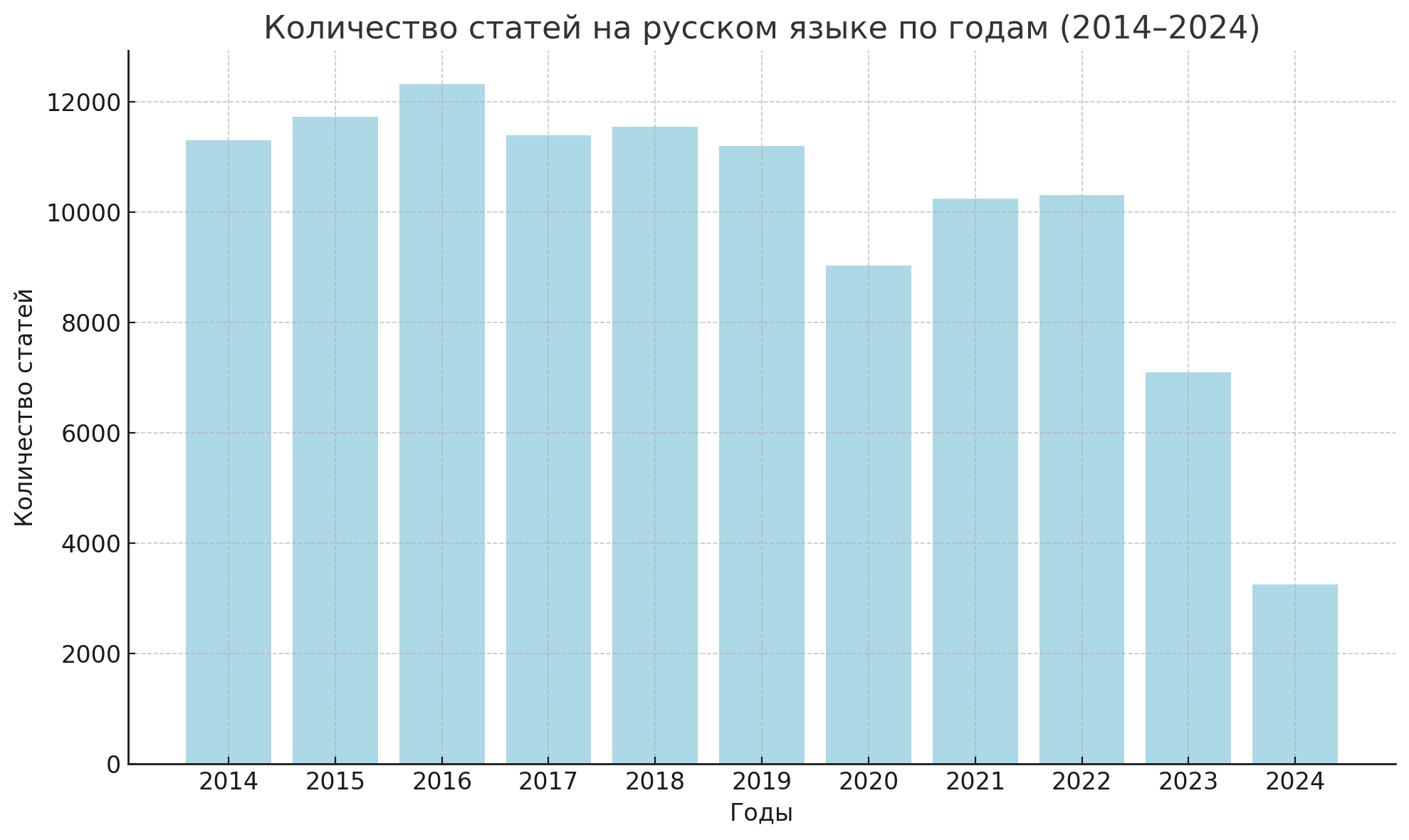


Рис. 2. Количество статей для каждого года до фильтрации.

Для отбора релевантных статей на третьем этапе применялась фильтрация по заданным ключевым словосочетаниям, предоставленным экспертами (например, "углеродный баланс", "парниковые газы", "низкоуглеродное развитие"). Далее данные были подготовлены для последующего заполнения базы данных. Полный список ключевых слов представлен в приложении 3.

Процесс включал:

* Лемматизацию и очистку текста, включая приведение данных к единому формату:
  + spaCy:
    - Для токенизации текста и получения информации о частях речи.
    - Использовалась русскоязычная модель ru\_core\_news\_sm для анализа структуры текста.
  + pymorphy3:
    - Для лемматизации слов на русском языке.
    - Обеспечивала преобразование слов в их базовую форму с учетом морфологических характеристик (часть речи, падеж, число и т.д.).
  + nltk:
    - Для работы со списками стоп-слов (стандартный набор для русского языка).
    - Использовалась для удаления часто встречающихся, но неинформативных слов, таких как "и", "в", "на".
  + re:
    - Для очистки текста с использованием регулярных выражений.
    - Удаление цифр, знаков препинания, английских слов и прочих нерелевантных символов.
  + functools (lru\_cache):
    - Для оптимизации функций, связанных с лемматизацией, с помощью кэширования результатов.
    - Позволила ускорить обработку повторяющихся слов.
* Поиск ключевых словосочетаний в названиях, аннотациях и ключевых словах.
* Исключение нерелевантных статей и сохранение результатов в отдельные файлы.
* Отфильтрованные данные были преобразованы в формат JSON, соответствующий структуре индекса.

После фильтрации было получено 905 статей (Рисунок 3).



Рис. 3. Количество статей для каждого года после фильтрации.

Предобработка текстов:

1. Объединение данных: для создания единого текста, содержащего всю необходимую информацию, данные из столбцов Title, Russian Keywords и Annotation были объединены.
2. Выделение предложений на русском языке:для каждой строки текста был выполнен разбор предложений. С помощью библиотеки langdetect определялся язык каждого предложения, и в итоговый текст включались только те предложения, которые были определены как написанные на русском языке.
3. Фильтрация текстов по языку: после обработки текстов на уровне предложений дополнительно проверялось, что весь текст относится к русскому языку. Для этого снова применялась библиотека langdetect. Тексты, не соответствующие данному критерию, исключались.
4. Токенизация и нормализация:
   1. Приведение текста к нижнему регистру.
   2. Удаление всех символов, кроме букв и пробелов.
   3. Разделение текста на токены с использованием библиотеки razdel.
   4. Исключение стоп-слов, используемых в русском языке (например, предлогов и союзов), с использованием библиотеки stop\_words.
   5. Приведение каждого токена к его начальной (лемматизированной) форме с использованием библиотеки pymorphy3.
5. Удаление пустых текстов: на заключительном этапе из набора данных исключались строки, где текст после обработки остался пустым.

Результат предобработки данных: набор данных, состоящий из лемматизированных слов, очищенных от лишних символов, стоп-слов и текстов на нерусском языке. Пример кода предобработки данных представлен в приложении 4. Количество обработанных текстов: 905.

## 2.3. Построение и оценка тематических моделей

Процесс выбора модели состоит из двух этапов. На первом этапе проводится построение моделей с различными параметрами, оценка их качества с использованием метрик, таких как когерентность (c\_v), перплексия, разделимость и перекрытие, и отбор моделей по метрикам. На втором этапе отобранные моделей оцениваются экспертами, которые анализируют интерпретируемость тем на основе списков ключевых слов и их осмысленность.

Этап 1. Построение моделей и отбор по значениям метрик.

LDA.  
Подготовка данных:

1. Создание словаря и корпуса: для представления текстов в формате, пригодном для тематического моделирования, был использован Dictionary из библиотеки Gensim. Были выполнены следующие шаги:

* Фильтрация редких слов, встречающихся в более чем 50% документов.
* Преобразование текстов в мешок слов (bag-of-words).

1. Диапазон количества тем: был задан диапазон количества тем от 5 до 25 с шагом 1, чтобы изучить влияние количества тем на качество модели.

Описание параметров модели:

* num\_topics: количество тем, задаваемое для каждой модели в диапазоне от 5 до 25. Этот параметр определяет число тематических групп, которые модель выделяет.
* alpha и eta: гиперпараметры, управляющие распределением вероятностей. Установлены в 'auto', что позволяет модели автоматически подстраивать их для оптимального обучения.
  + alpha отвечает за распределение тем в документах: низкое значение приводит к более узкоспециализированным темам, а высокое — к равномерному распределению.
  + eta регулирует распределение слов в темах: низкое значение фокусирует темы на небольшом наборе слов, а высокое делает темы более размытыми.
* passes: количество проходов по всему корпусу. Установлено значение 50 для стабилизации распределения тем и слов.
* iterations: количество итераций обновления распределения на каждом проходе. Установлено значение 5.
* random\_state: фиксированный параметр случайности (42), чтобы обеспечить воспроизводимость результатов.

Для каждого количества тем вычислялись метрики: когерентность (Рисунок 4), перплексия (Рисунок 5), разделимость (Рисунок 6), перекрытие (Рисунок 7).



Рис. 4. Значение когерентности c\_v для моделей LDA.



Рис. 5. Значение перплексии для моделей LDA.

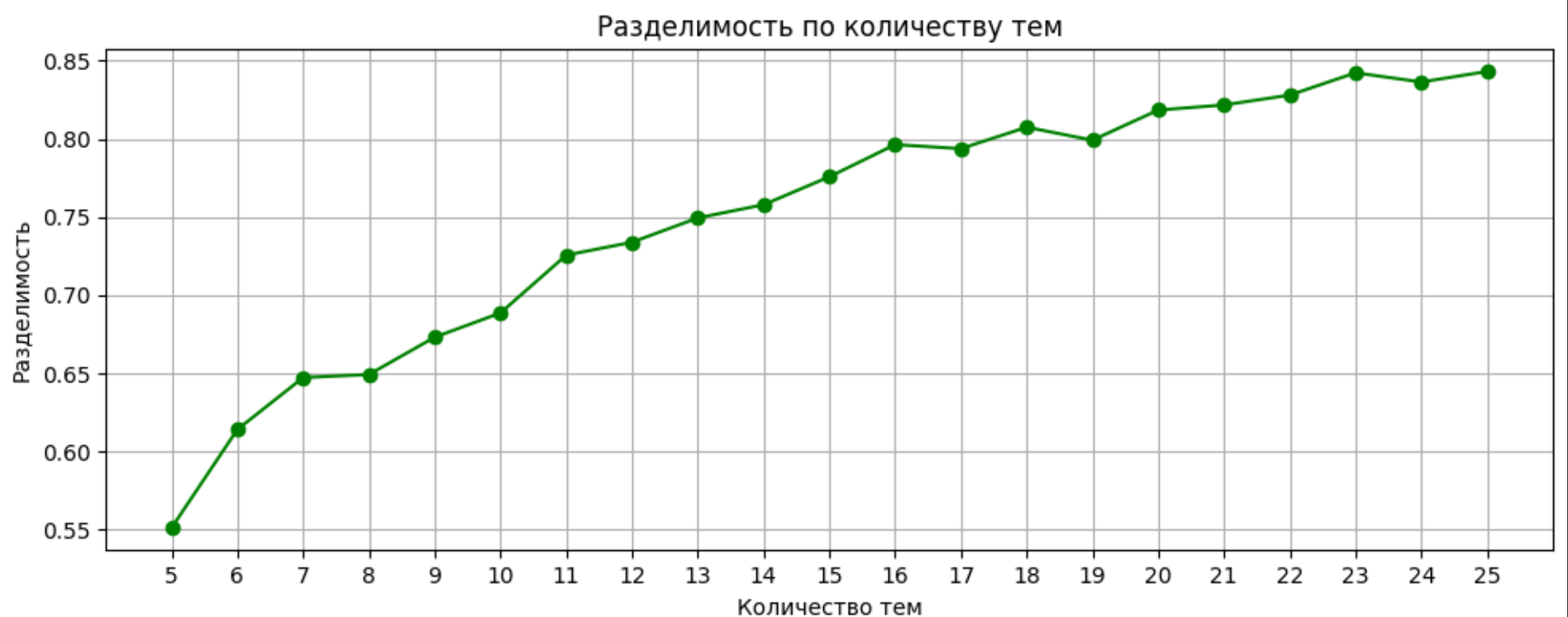


Рис. 6. Значение разделимости для моделей LDA.

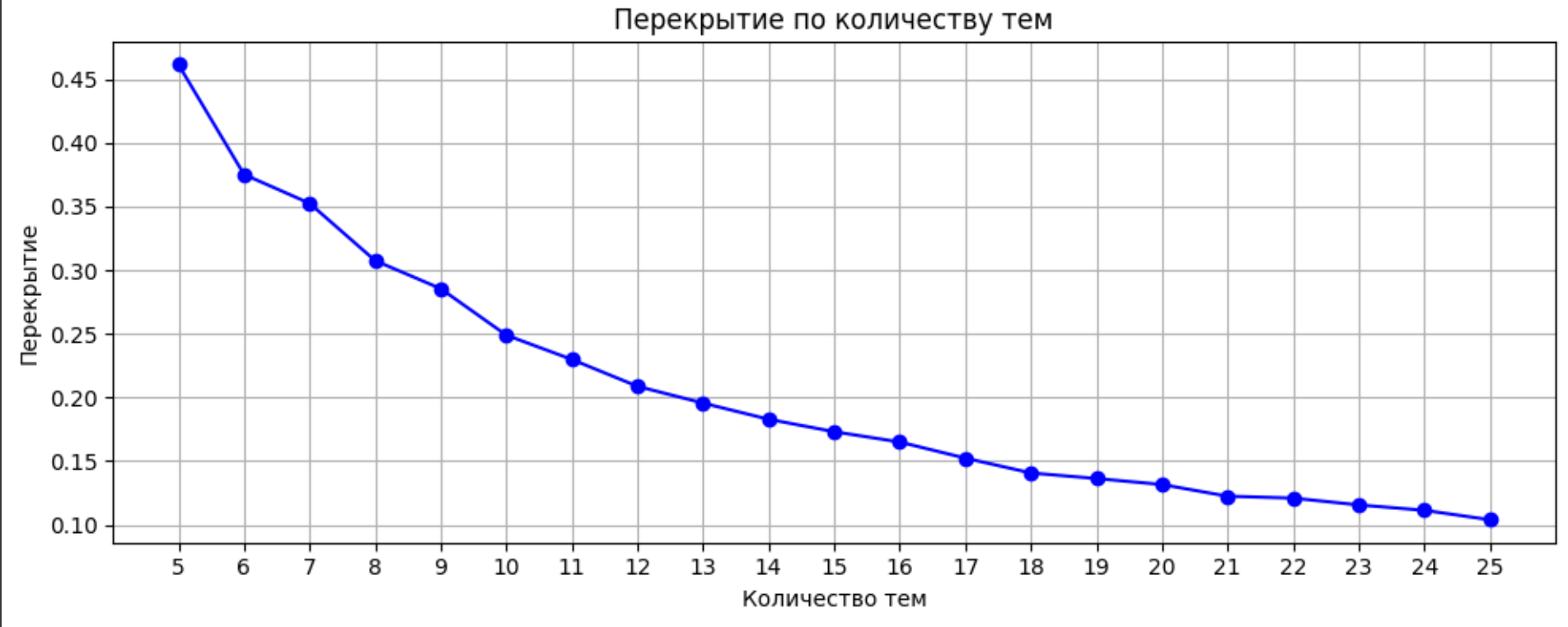


Рис. 7. Значение перекрытия для моделей LDA.

Количество тем выбиралось как компромисс между минимальной перплексией, максимальной когерентностью (c\_v). Для рассмотрения оставлены модели с 6 и 14 темами (Таблица 4).

Таблица 4

Модели LDA и значения метрик

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LDA | Кол-во тем | Когерентность (c\_v) | Перплексия | Разделимость | Перекрытие |
| lda\_model\_1 | 6 | 0.4772 | 940.3951 | 0.61 | 0.37 |
| lda\_model\_2 | 14 | 0.4721 | 923.3507 | 0.75 | 0.18 |

BERTopic.

1. Эмбеддинги SentenceTransformer: для получения векторных представлений текстов использовалась модель "multilingual-e5-large-instruct". Эта модель позволяет обрабатывать тексты на нескольких языках, что важно для работы с текстами, содержащими названия или термины на иностранных языках.
2. Снижение размерности с UMAP: метод UMAP применялся для уменьшения размерности эмбеддингов и улучшения качества кластеризации. Основные параметры:

* n\_neighbors — количество соседей для определения локальной структуры данных;
* n\_components=3 — снижение размерности для визуализации;
* min\_dist — минимальное расстояние между точками в проекции;
* metric='cosine' — использование косинусного расстояния для измерения схожести.

1. Кластеризация с HDBSCAN: для выделения кластеров применялся алгоритм HDBSCAN, автоматически определяющий число кластеров и устойчивый к шуму. Основные параметры:

* min\_cluster\_size — минимальный размер кластера;
* min\_samples — минимальное количество точек для определения ядра кластера;
* cluster\_selection\_method='eom' — выбор кластеров на основе плотности.

1. Векторизация текста: для преобразования текстов в числовые представления использовался CountVectorizer с анализом биграмм. Это позволяет учитывать как отдельные слова, так и их сочетания, что способствует более точному выделению признаков. Параметр: ngram\_range=(1, 2).
2. TF-IDF с ClassTfidfTransformer: для уменьшения влияния частотных, но незначимых слов применялся модифицированный TF-IDF, который усиливает значимость редких слов. Параметр: reduce\_frequent\_words=True.
3. Представление тем: для улучшения интерпретируемости тем использовались:

* KeyBERTInspired — для выделения ключевых слов;
* Maximal Marginal Relevance (MMR) — для выбора ключевых слов с балансом между релевантностью и разнообразием. Параметр: diversity=0.4, 40% разнообразия и 60% релевантности, для уменьшения дублирования схожих слов, но с сохранением связи с темой.

1. Автоматический выбор количества тем: позволяет модели автоматически определять количество тем на основе структуры данных. Параметр: nr\_topics="auto".

Выбиралась модель с более высокой когерентностью, значения когерентности при разных параметрах представлены на рисунке 8. На количество тем влияли параметры n\_neighbors, min\_cluster\_size, min\_samples. При увеличении получались более общие и большие кластеры, при уменьшении - маленькие и детализированные.

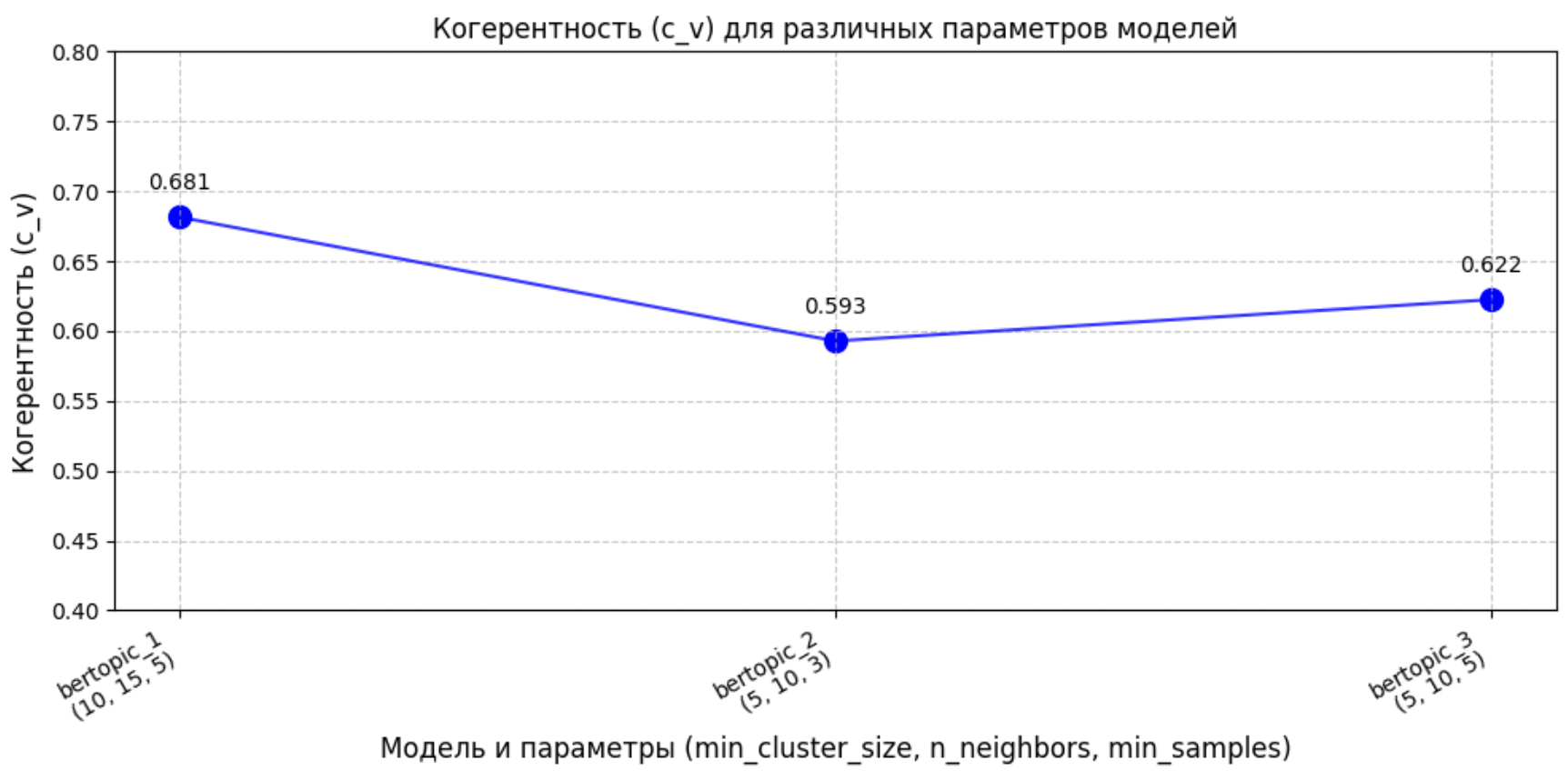


Рис. 8. Значение когерентности c\_v для моделей BERTopic.

Все три модели оставлены для рассмотрения (Таблица 5).

Таблица 5

Модели BERTopic и значения метрик

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| BERTopic | min\_cluster\_size | n\_neighbors | min\_samples | Когерентность (c\_v) | Перекрытие | Кол-во тем |
| bertopic\_1 | 10 | 15 | 5 | 0.6814 | 0.37 | 13 |
| bertopic\_2 | 5 | 10 | 3 | 0.5928 | 0.21 | 25 |
| bertopic\_3 | 5 | 10 | 5 | 0.6224 | 0.36 | 20 |

Модели BERTopic демонстрируют более высокие значения когерентности по сравнению с LDA. Это связано с тем, что BERTopic использует эмбеддинги текста, которые захватывают семантический контекст и взаимосвязи между словами, тогда как LDA опирается на частотный анализ слов, что ограничивает её способность учитывать смысловую глубину текста.

Результат этапа 1: выбрано 5 моделей, из них 2 модели LDA и 3 модели BERTopic, для проведения экспертной оценки.

Этап 2. Экспертная оценка и выбор модели.

Отобранные на основе показателей метрик модели были отправлены на оценку экспертам. Им были предоставлены полученные темы в виде списков слов и словосочетаний (топ 10 для каждой темы), упорядоченных по убыванию степени принадлежности, было предложено решить, является ли каждая из предоставленных тем в какой-то степени последовательной, осмысленной и интерпретируемой. Индикатором такой темы служит возможность дать ей некоторое обобщенное название [10]. Для подсчёта результатов использовалась формула:

Оценка = (кол-во интерпретированных тем) / (кол-во всех тем).

Составлена общая таблица моделей с указанием основной метрики (когерентность c\_v) и оценки экспертов (Таблица 5).

Таблица

Общая таблица моделей

|  |  |  |  |
| --- | --- | --- | --- |
| Модель | Кол-во тем | Когерентность | Оценка экспертов |
| lda\_model\_1 | 6 | 0.4772 | 1.0 |
| lda\_model\_2 | 14 | 0.4721 | 0.85 |
| bertopic\_1 | 13 | 0.6814 | 1.0 |
| bertopic\_2 | 25 | 0.5928 | 0.96 |
| bertopic\_3 | 20 | 0.6224 | 0.95 |

Эксперты отмечают, что модели с меньшим количеством тем описывают более общие и широкие темы, без достаточной детализации. А модели BERTopic сочли более интерпретируемыми, поскольку для описания тем используются n-граммы. Это позволяет человеку лучше понять контекст.

Результат этапа 2: выбрана модель bertopic\_3, которая наилучшим образом сочетает рекомендации экспертов по количеству тем и высокие значения метрик. Код для инициализации и обучения bertopic\_3 представлен в приложении 5.

## 2.4. Визуализация результатов тематического моделирования

Для более глубокого понимания результатов тематического моделирования и выявления взаимосвязей между темами важна визуализация, которая позволяет наглядно представить распределение тем в пространстве, их иерархию и структуру. В рамках проекта была выбрана модель BERTopic. Она предоставляет встроенные методы для создания интерактивных карт тем и их визуализации.

Основные методы визуализации, использованные в проекте:

1. Карта тем (Topic Map):  
   Метод позволяет отобразить темы на плоскости, используя UMAP для снижения размерности. Темы визуализируются как кластеры, а их близость друг к другу указывает на степень их семантической связи. Это даёт возможность изучать расположение тем и их группы в пространстве.
2. Иерархия тем (Hierarchical Topics):  
   Построение иерархической структуры тем, где более крупные темы делятся на подкатегории. Это помогает понять, какие темы связаны между собой и как они соотносятся с более общими категориями.
3. Частотная диаграмма тем (Bar Chart of Topic Frequencies):  
   Позволяет отобразить частоту появления каждой темы в корпусе, что помогает выявить наиболее распространённые и важные темы.
4. Ключевые слова темы (Word Scores):  
   Построение списков ключевых слов для каждой темы с визуализацией их значимости. Это помогает лучше понять содержание темы через её основные слова.
5. Динамика изменения тем (Topic Evolution):  
   Если корпус содержит временную метку, данный метод позволяет отследить, как частота тем изменяется со временем, что помогает выявить тренды в исследовательских направлениях.

В рамках проекта первые два метода визуализации, "Карта тем" и "Иерархия тем", используются в их оригинальном виде, предоставляемом моделью BERTopic. Третий и четвёртый методы, "Частотная диаграмма тем" и "Ключевые слова темы", имеют схожую цель — отображение частотности и ключевых слов для каждой темы. Но при большом количестве тем визуализации становятся плохо интерпретируемыми из-за слияния линий и точек. В связи с этим данные метрики будут представлены в виде таблиц или списков, что позволит обеспечить их читаемость и удобство анализа. Кроме того, в корпусе присутствуют временные метки, что делает возможным использование метода "Динамика изменения тем". Однако при большом количестве тем результаты визуализации оказываются перегруженными и теряют информативность. Поэтому для анализа отдельных тем будут построены графики, отражающие их частотность по годам и другим параметрам. Такой подход позволяет сфокусироваться на интересующих темах и выявить их динамику в различных контекстах.

# ГЛАВА 3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ И ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

## 3.1. Архитектура веб-приложения и используемые технологии

Архитектура веб-приложения представляет собой клиент-серверную модель, состоящую из трех ключевых компонентов: клиентской части (Frontend), серверной части (Backend) и базы данных (Рисунок 9). Компоненты взаимодействуют через запросы.

Пользователь взаимодействует с клиентской частью через веб-страницу. На стороне клиента отображаются результаты поиска, графики и интерактивные карты, предоставляя удобный интерфейс для работы с системой. Клиентская часть отправляет HTTP-запросы серверной части для получения данных.

Серверная часть отвечает за обработку запросов пользователя. Она принимает запросы от клиента, взаимодействует с базой данных для поиска или анализа данных и возвращает результаты клиенту. Серверная часть построена таким образом, чтобы обрабатывать два основных типа запросов: поиск научных статей и построение графиков/интерактивных карт.

База данных предназначена для хранения научных статей, их метаданных и аналитических данных. Она принимает запросы от серверной части в виде DSL-запросов и возвращает результаты в формате JSON.

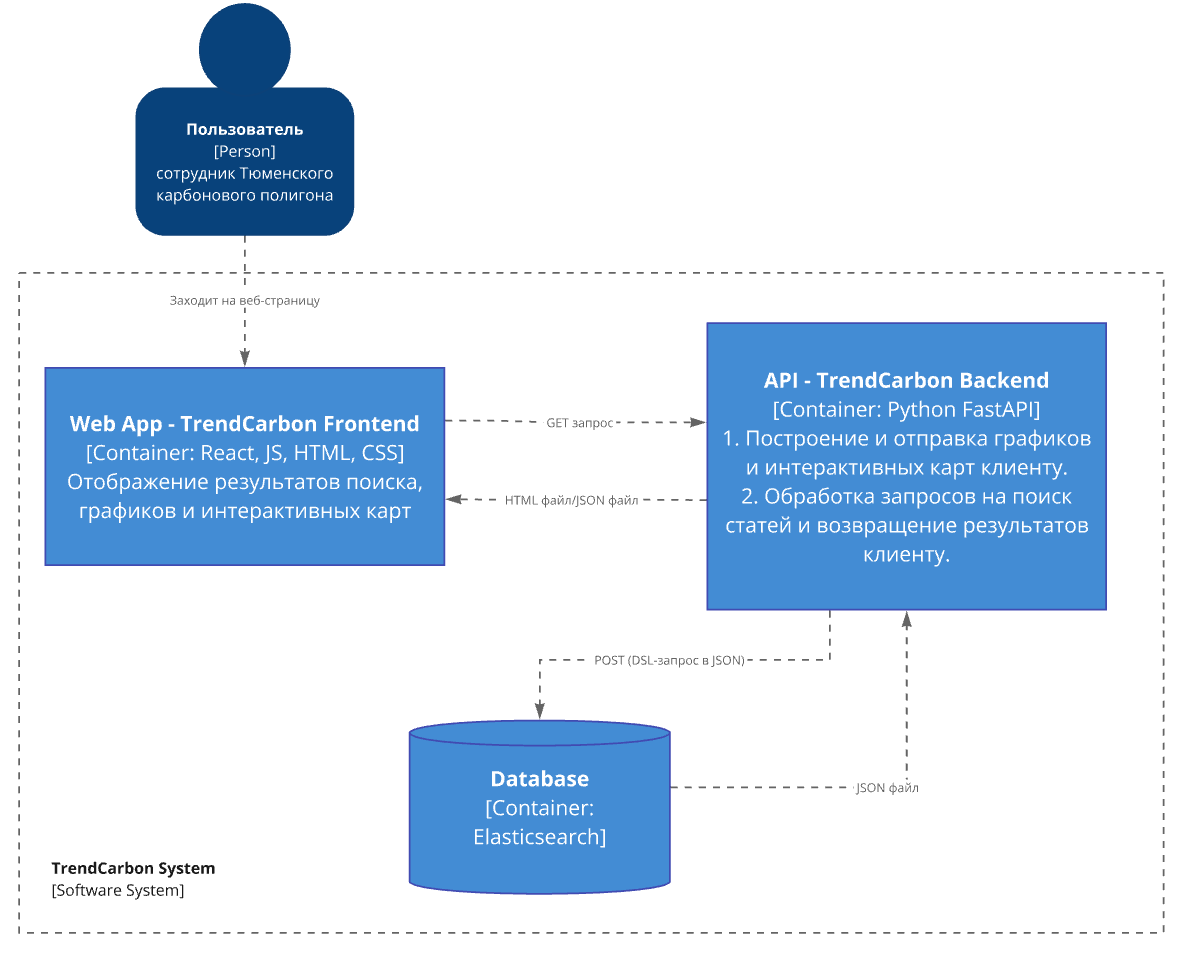


Рис. 9. Архитектура приложения в C4-нотации.

Используемые технологии.

1. Клиентская часть:
   1. React: библиотека для создания интерактивного пользовательского интерфейса.
   2. HTML/CSS/JavaScript: для верстки и стилизации компонентов интерфейса.
2. Серверная часть:
   1. FastAPI: фреймворк для создания REST API на языке Python.
3. Хранилище данных:
   1. Elasticsearch: поисковая система, используемая как база данных для хранения информации о статьях и результатах анализа.

Клиентская часть позволяет пользователям работать с результатами тематического моделирования и поиска статей, а серверная часть обрабатывает запросы и предоставляет доступ к данным, хранящимся в Elasticsearch.

## 3.2. Функциональные возможности

Разработанная система предоставляет пользователям возможность поиска, анализа и визуализации научных статей по углеродной тематике. Основные функциональные возможности включают:

1. Поиск статей:
   1. Полнотекстовый поиск по ключевым словам, авторам, названию, журналу и другим атрибутам.
2. Тематическое моделирование и анализ:
   1. Кластеризация статей по темам на основе алгоритмов тематического моделирования.
   2. Визуализация результатов тематического моделирования — построение интерактивной карты тем и иерархии тем.
3. Визуализация данных:
   1. Построение гистограмм для анализа количества публикаций по годам.
   2. Создание круговых диаграмм для распределения статей по разделам.
4. Интерактивный интерфейс:
   1. Удобный доступ к результатам поиска и визуализации.

## 3.3. Реализация поиска с использованием ElasticSearch

Для реализации поиска данных использовался поисковый движок Elasticsearch.

Основные этапы реализации поиска:

1. Интеграция с Elasticsearch:
   1. Подключение приложения к индексу Elasticsearch, содержащему данные о статьях.
   2. Настройка параметров поиска с учётом специфики текстов на русском языке, включая обработку морфологии.
2. Механизм поиска:
   1. Полнотекстовый поиск по нескольким полям (название, авторы, ключевые слова, аннотация, журнал) с использованием запроса multi\_match.
   2. Поддержка поиска по всем документам с помощью match\_all, если строка запроса отсутствует.
   3. Пагинация результатов для удобства отображения данных.
3. Результаты поиска:
   1. Получение данных о статьях с указанием ключевых метаданных.
   2. Постраничный вывод результатов для удобства пользователей.

Код реализации приведен в приложении 6.

## 3.4. Создание серверного API

В приложении используется FastAPI для создания API, которое обеспечивает обработку пользовательских запросов, взаимодействие с Elasticsearch и визуализацию данных. Серверная часть позволяет обрабатывать запросы к индексу статей.

Основной функционал API:

* Проверка доступности Elasticsearch: При старте приложения выполняется проверка соединения с Elasticsearch, а возможные ошибки подключения обрабатываются для обеспечения стабильной работы.
* Поиск данных: Реализованы запросы типа match\_all, multi\_match и term для поиска по полям (например, title, authors, keywords.russian, annotation).
* Визуализация данных: Методы для создания гистограммы распределения статей по годам и круговой диаграммы по научным направлениям используют библиотеку Plotly, а результаты возвращаются в формате HTML-страниц.
* Работа со статическими файлами: Методы для предоставления заранее подготовленных визуализаций и аналитических отчетов из папки visualizations.

Разработанное API объединяет возможности поиска, анализа и визуализации данных.

## 3.5. Описание пользовательского интерфейса

Пользовательский интерфейс веб-приложения разработан для удобного взаимодействия исследователей с результатами тематического моделирования и поиска научных статей. Он предоставляет доступ к двум основным страницам: «О статьях» и «Тематики», каждая из которых ориентирована на решение конкретных задач анализа данных. Пример кода реализации приведён в приложении 7.

Раздел «О статьях» позволяет пользователю находить статьи по ключевым словам (Рисунок 10,11). Основные элементы страницы:

* Строка поиска: позволяет вводить поисковые запросы, используя ключевые слова, фразы или имена авторов.
* Карточки статей: результаты поиска отображаются в виде карточек, содержащих следующую информацию:
  + Название статьи.
  + Год публикации.
* Каждая карточка является интерактивной: при клике на карточку открывается отдельное окно или страница с детализированной информацией о статье (Рисунок 12). В этом разделе пользователю становятся доступна следующая информация (при наличии):
  + Полное название статьи.
  + Список авторов.
  + Год публикации.
  + Раздел, в котором опубликована статья.
  + Ключевые слова на русском и английском языках.
  + Полный текст аннотации статьи.
  + Цитирования ГОСТ и АРА.
  + Ссылка на источник.
  + Номер темы (или -1, если тема отсутствует).



Рис. 10. Страница «О статьях».

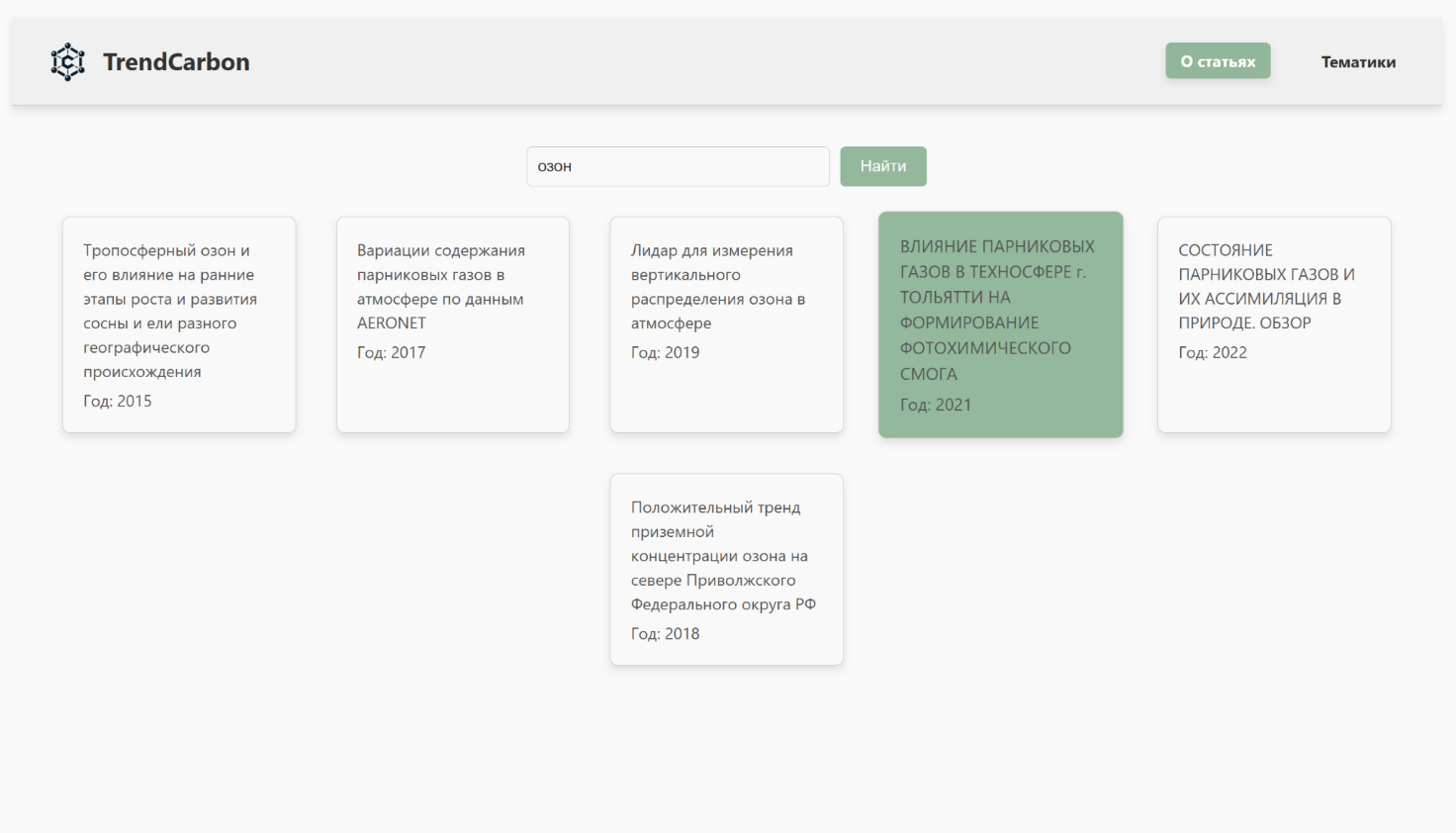


Рис. 11. Работа функции поиска.

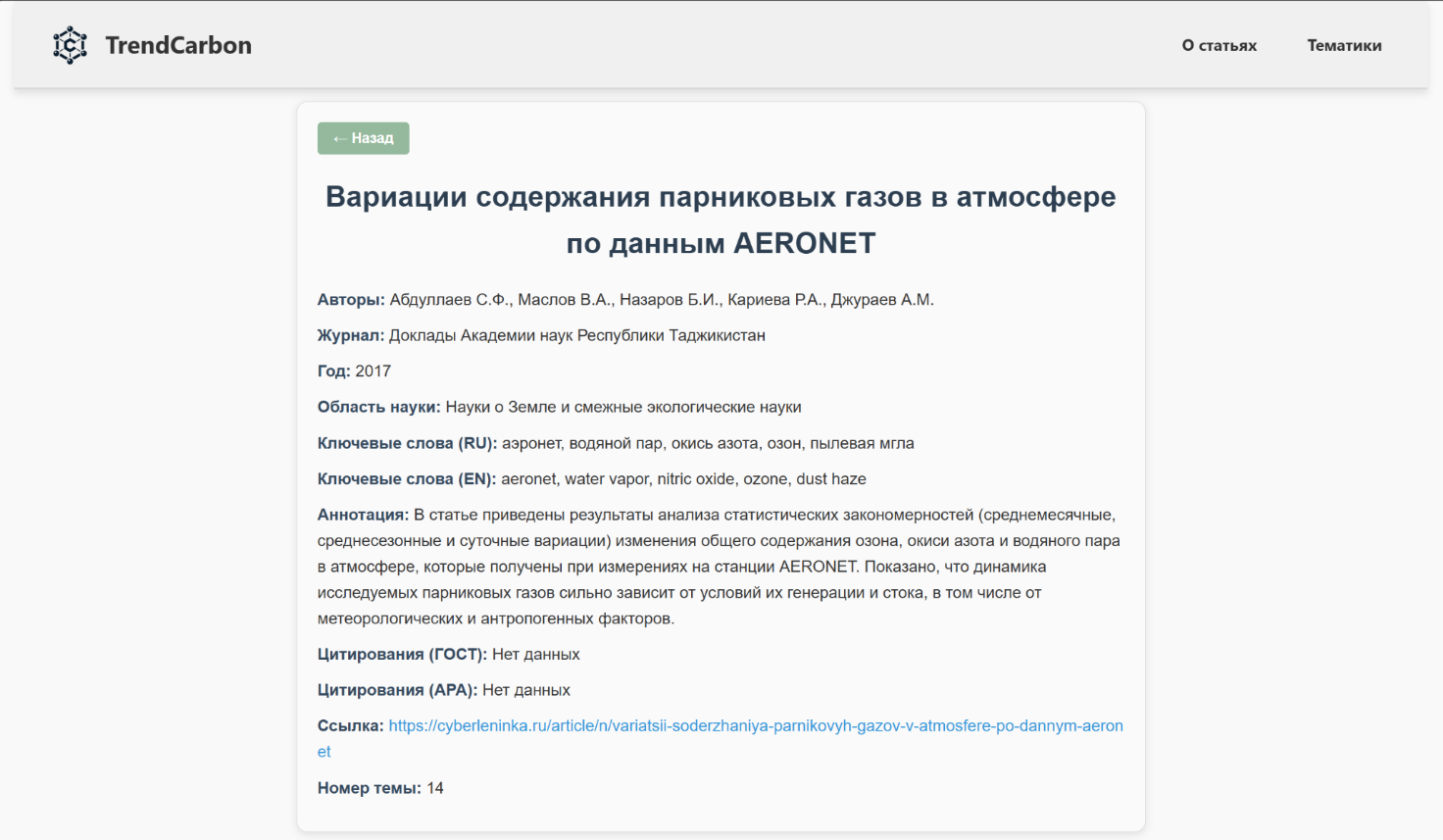


Рис. 2. Страница просмотра информации о статье.

Раздел «Тематики» предоставляет результаты тематического моделирования с использованием визуализаций и текстовых данных:

* Интерактивная карта тем: позволяет визуально представить расположение тем в пространстве, используя метод UMAP. Темы отображаются в виде кластеров, а их близость указывает на семантическую связь (Рисунок 13). Также на странице есть инструкция по использованию карты (Рисунок 14).

Интерпретация карты:

Эта визуализация помогает понять распределение и взаимосвязь между темами, выделенными в текстах. Каждый пузырёк на карте представляет одну тему, а его размер указывает на относительную важность темы в анализируемых данных.

Как читать карту:

* Оси D1 и D2: Это двумерное пространство, в котором темы расположены на основе их сходства. Чем ближе два пузырька, тем более схожи соответствующие темы.
* Размер пузырьков: Указывает на долю текста, которую покрывает тема. Крупные пузырьки представляют более важные темы.
* Ползунок Тема: Можно использовать ползунок внизу, чтобы выбрать интересующую тему. После выбора на карте подсвечиваются ключевые слова, связанные с этой темой.

Как использовать:

* Навести курсор на пузырёк, чтобы увидеть основные ключевые слова и их вероятности для данной темы.
* Использовать ползунок внизу для выбора темы, чтобы сосредоточиться на её ключевых характеристиках.
* При необходимости увеличить или уменьшить масштаб карты с помощью кнопок в правом верхнем углу (иконки с + и -).

Пример использования:

Крупный пузырь в правом верхнем углу означает, что эта тема охватывает большую часть текстов. Близко расположенные пузырьки могут указывать на схожие или пересекающиеся темы. Это полезно для выявления взаимосвязей между темами.

* Иерархия тем: на странице представлена иерархическая структура тем в виде графа, демонстрирующая взаимосвязи между крупными и подчинёнными темами (Рисунок 15).

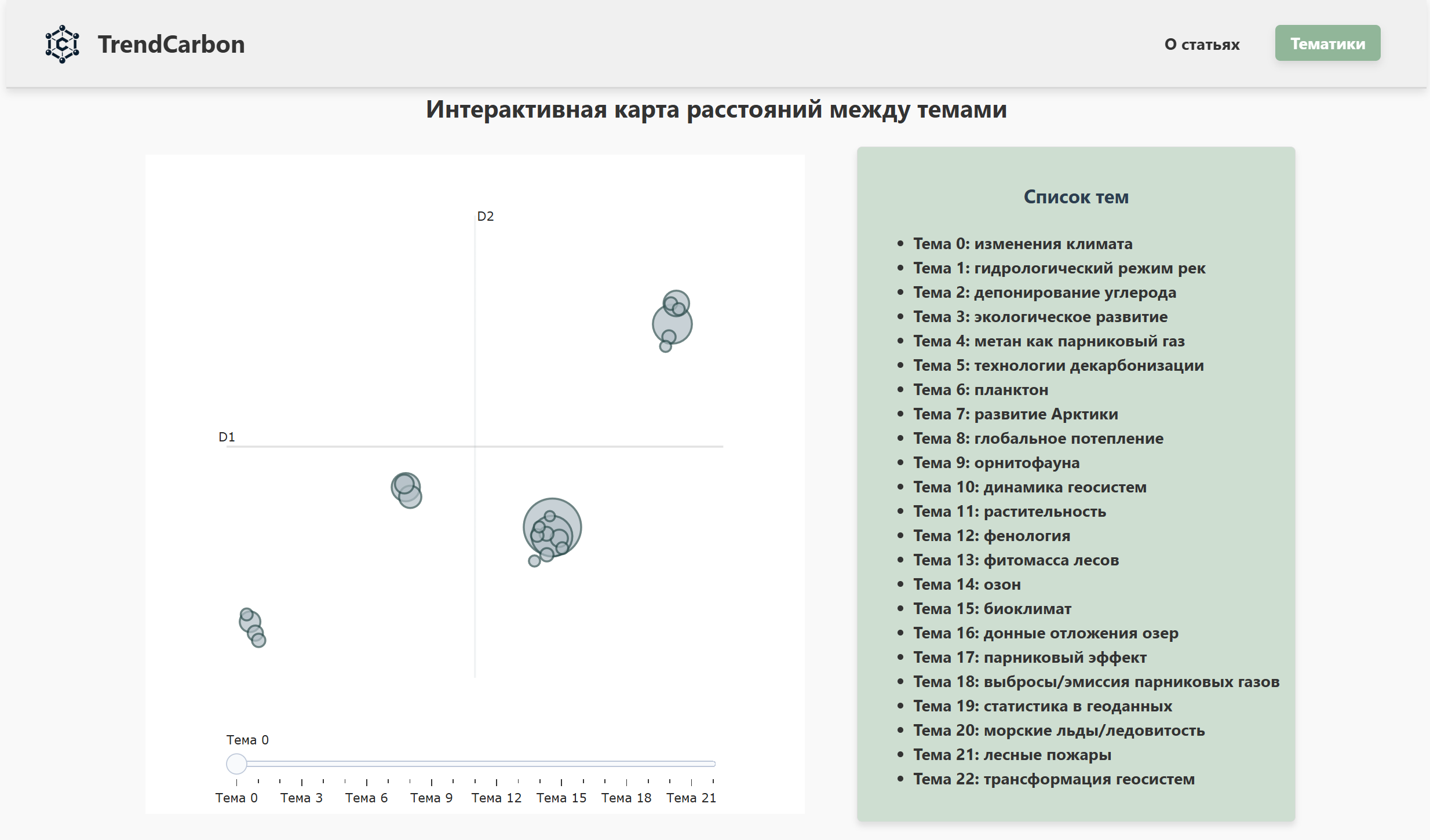


Рис. 13. Страница «Тематики»: интерактивная карта.

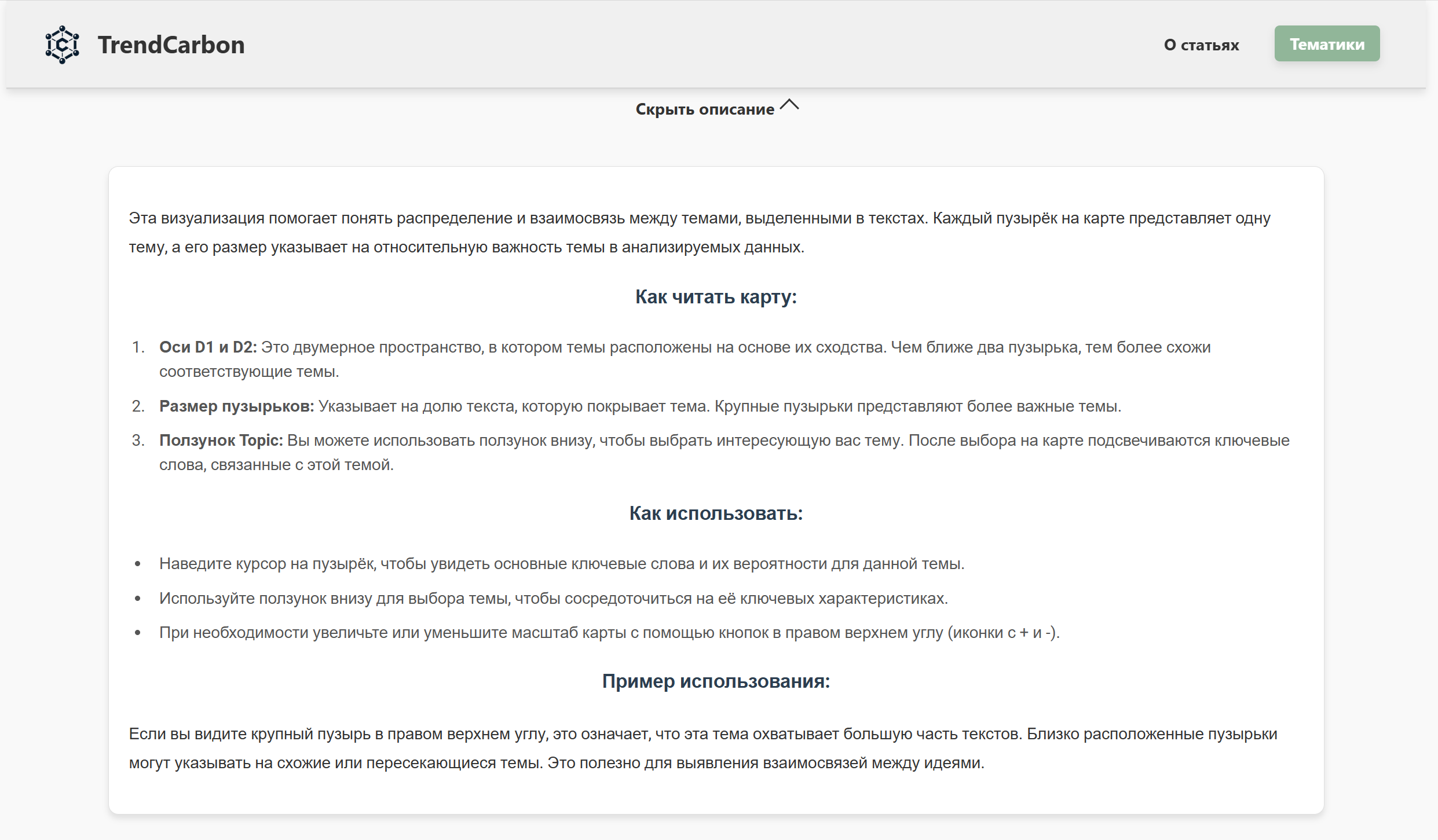


Рис. 14. Инструкция по использованию интерактивной карты.

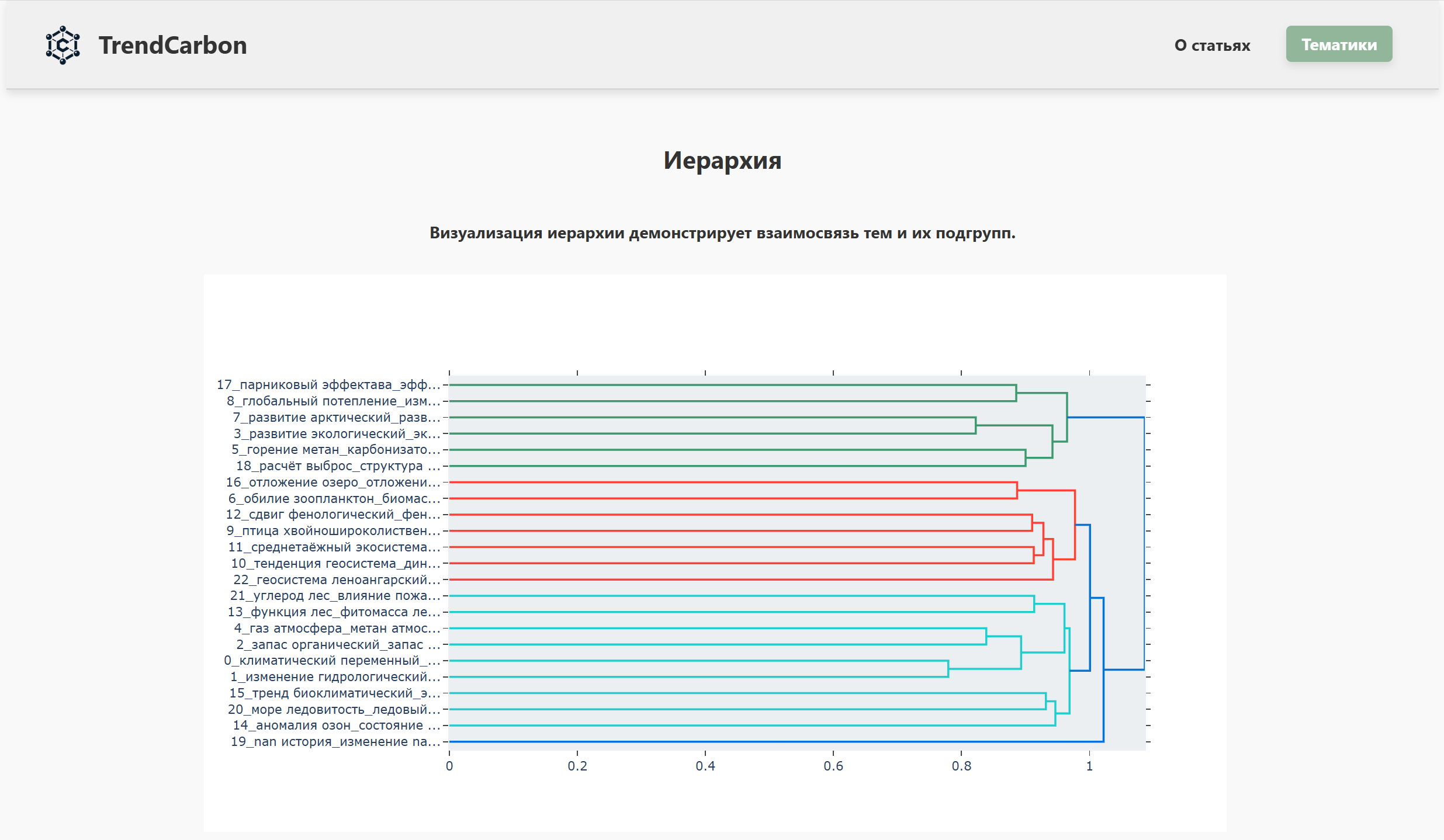


Рис. 15. Продолжение страницы «Тематики»: иерархия тем.

Каждая тема из списка тем интерактивна: пользователь может кликнуть на интересующую его тему, чтобы перейти на отдельную страницу с подробной информацией о ней.

На странице просмотра темы представлена следующая информация:

1. Топ-10 слов, описывающих тему (Рисунок 16):
   * Список ключевых слов, наиболее характерных для данной темы, позволяет быстро понять её основное содержание.
2. Гистограмма распределения статей по годам (Рисунок 17):
   * График показывает, как менялось количество публикаций по выбранной теме в разные годы, что помогает анализировать динамику интереса к теме.
3. Круговая диаграмма по разделам (Рисунок 18):
   * Диаграмма отражает, в каких научных разделах чаще всего публикуются статьи, относящиеся к данной теме, что позволяет оценить её междисциплинарный характер.

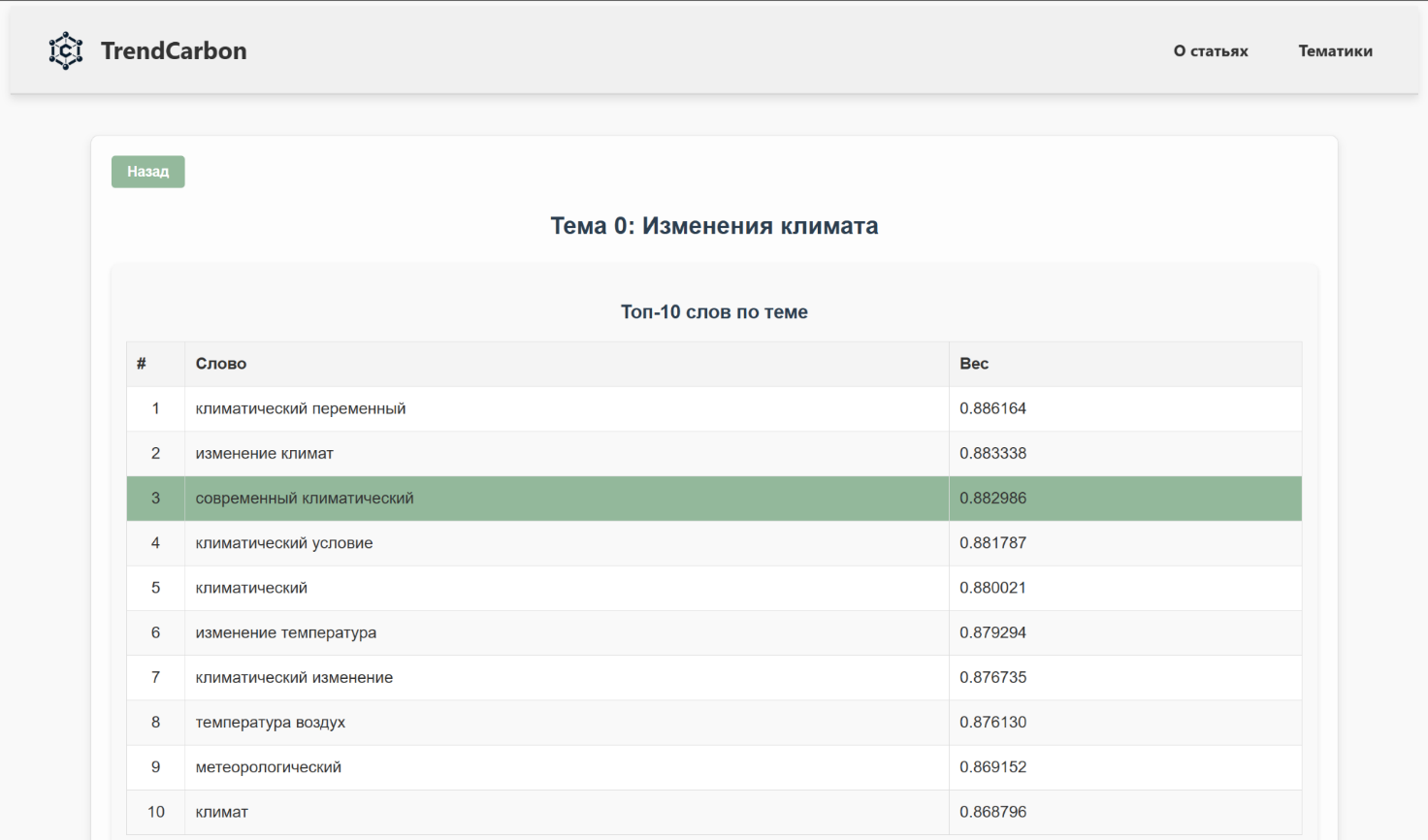


Рис. 16. Страница просмотра информации о теме: топ слов.

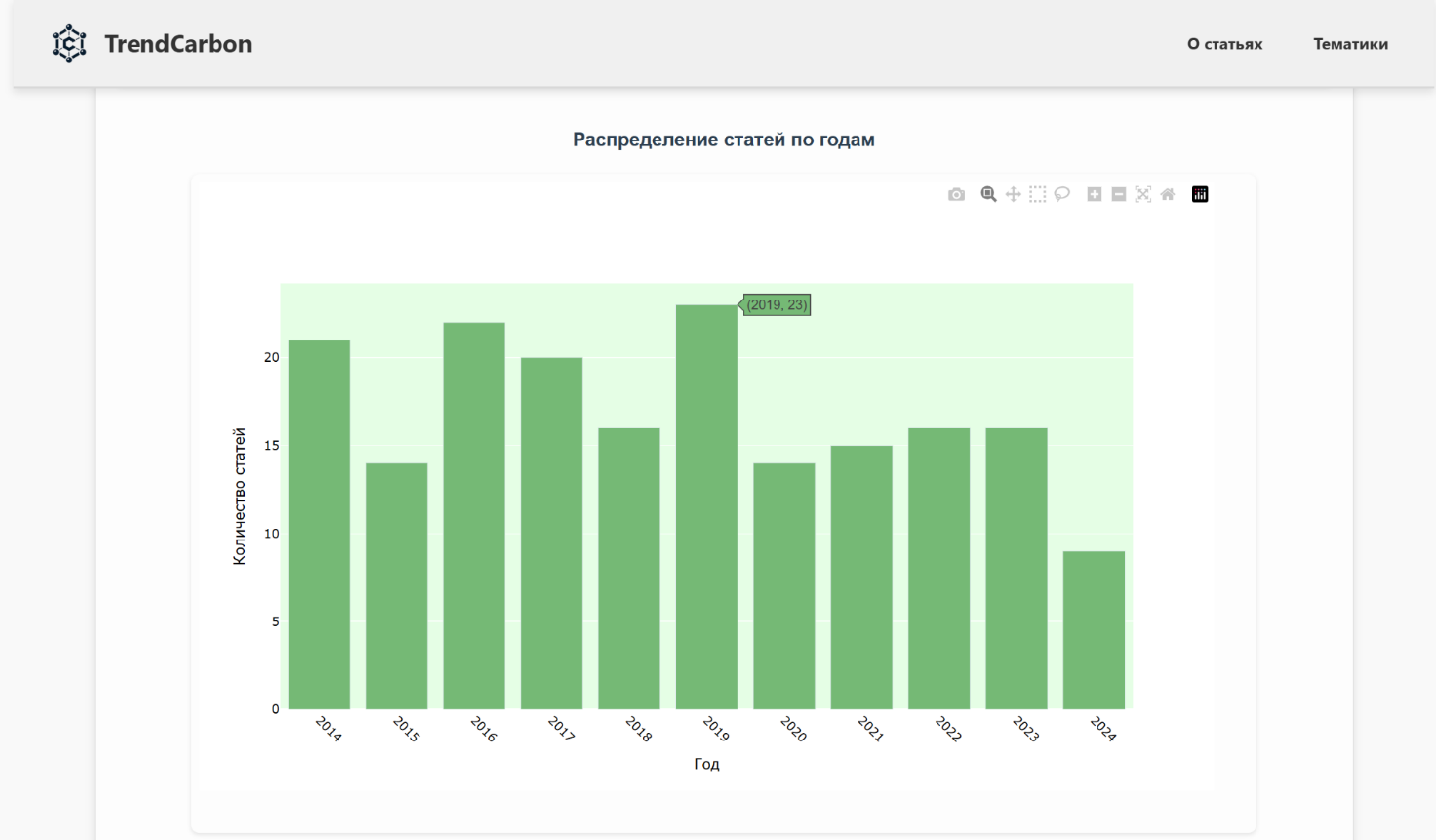


Рис. 17. Гистограмма распределения статей по годам.

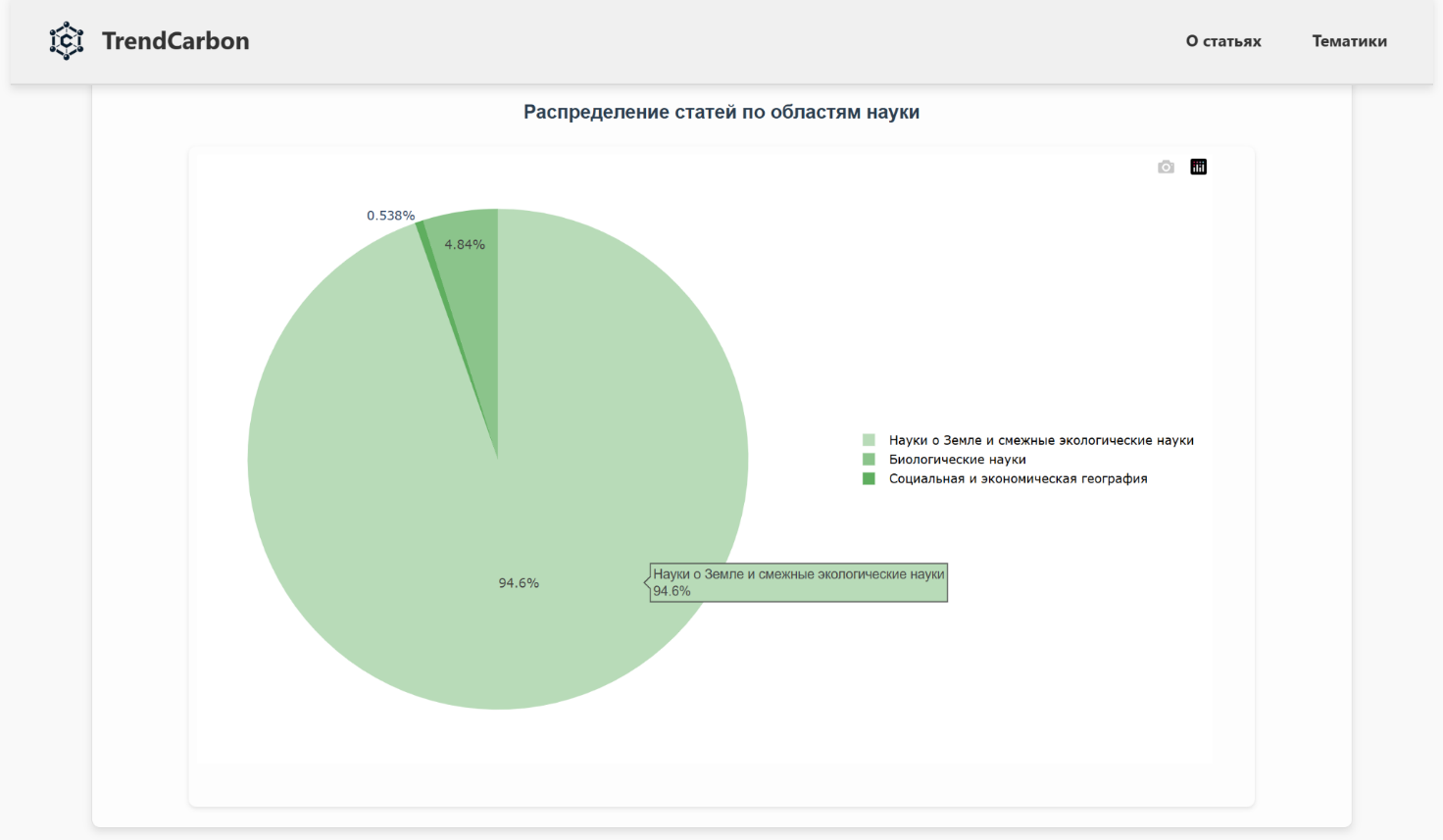


Рис. 18. Круговая диаграмма по разделам.

## 3.6. Тестирование веб-приложения

Тестирование серверной части:

На этапе тестирования серверной части приложения была проведена проверка работы всех эндпоинтов API и их взаимодействия с базой данных Elasticsearch. Тестирование проводилось вручную с использованием Postman:

1. Эндпоинт проверки состояния Elasticsearch:

* GET /health
* Цель: Убедиться в доступности базы данных Elasticsearch.
* Результат: Сервер возвращает статус 200 OK и сообщение «Elasticsearch is reachable».

1. Эндпоинт поиска статей:

* GET/search?query=ключевое\_слово&page=номер\_страницы&size=количество\_записей
* Цель: Проверить полнотекстовый поиск статей по заданному ключевому слову.
* Результат: Возвращается массив найденных статей с ключевыми атрибутами (название, авторы, журнал, год, ключевые слова, аннотация и ссылка).

1. Эндпоинт генерации гистограммы публикаций по годам:

* GET /generate\_histogram\_year?topic\_number=номер\_темы
* Цель: Проверить генерацию графика, отображающего количество публикаций по годам для выбранной темы.
* Результат: Возвращается HTML-страница с интерактивным графиком.

1. Эндпоинт генерации круговой диаграммы по разделам:

* GET /generate\_pie\_chart?topic\_number=номер\_темы
* Цель: Проверить генерацию диаграммы, отображающей распределение статей по тематическим разделам.
* Результат: Возвращается HTML-страница с интерактивной диаграммой.

1. Эндпоинты взаимодействия с Elasticsearch:

* GET /article/\_doc/1: Проверка получения документа из базы данных по идентификатору.
* GET /article/\_search: Проверка полнотекстового поиска в базе данных.
* Результат: Успешный доступ к данным в ElasticSearch и корректное выполнение запросов.

Итоги тестирования:

* Все эндпоинты корректно обрабатывают запросы и возвращают данные в ожидаемом формате.
* Elasticsearch стабильно взаимодействует с серверной частью приложения, обеспечивая доступ к данным.
* При некорректных параметрах запросов сервер возвращает стандартные ошибки (например, 400 Bad Request или 404 Not Found).

Тестирование клиентской части:

Для клиентской части приложения было проведено ручное тестирование, которое включало проверку функциональности основных элементов интерфейса и взаимодействия пользователя с приложением. В процессе тестирования была проверена корректность работы всех страниц и полей, а также функциональность визуальных элементов.

1. Страница «О статьях»:

* Тестировалась поисковая строка, обеспечивающая возможность ввода и отправки поискового запроса для поиска статей по базе данных.
* Проверялась работа кнопки для отправки запроса, а также корректность отображения результатов поиска, включая наличие нужных фильтров и правильность вывода информации о статьях.
* Также оценивалась обработка ошибок в случае отсутствия данных или неправильного запроса (отображение сообщения «Ничего не найдено» при пустом ответе).

1. Страница «Тематики»:

* Оценивалась работа визуализаций результатов тематического моделирования, включая диаграммы и графики, отображающие тематические распределения.

Итоги тестирования: все основные функции, такие как поиск статей по базе данных, отображение результатов, а также взаимодействие с графиками и диаграммами на странице "Тематики", продемонстрировали корректную работу.

# ЗАКЛЮЧЕНИЕ

В результате реализации проекта мы создали работоспособный прототип (MVP) сервиса, который способен агрегировать тексты научных статей на русском языке по углеродной тематике и анализировать их с помощью методов тематического моделирования. Были успешно выполнены все поставленные задачи. Полученный инструмент позволяет исследователям и специалистам по углеродным технологиям отслеживать и определять актуальные исследовательские ниши.

Достигнув цели проекта и создав MVP, мы заложили основу для дальнейшего совершенствования сервиса. В перспективе планируется:

* Автоматизация сбора, обработки и обновления научных статей по углеродной тематике для поступления новых данных в базу данных.
* Автоматическое обновление тематической модели при добавлении новых данных для обеспечения актуальности результатов анализа.
* Разработка функционала для персонализированного доступа, сохранения запросов и результатов анализа статей.

Таким образом, реализованный проект стал отправной точкой для формирования полноценного инструмента поддержки исследований и принятия обоснованных решений в сфере углеродных технологий.

# СПИСОК ЛИТЕРАТУРЫ

1. Анализ текстовых данных с использованием тематического моделирования // habr.com : сайт. – URL: https://habr.com/ru/companies/otus/articles/757010/ (дата обращения: 15.12.2024)
2. Madison, Pickett Exploring Coherence Metrics for Optimizing Topic Models of Humpback Song / Pickett Madison // Massachusetts Institute of Technology : электронный журнал. – URL: https://www.mbari.org/wp-content/uploads/Pickett.pdf. – Дата публикации: 2020.
3. Новиков, А.А. МЕТОДЫ ТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ПРИ РАЗРАБОТКЕ ПРОЕКТНЫХ РЕШЕНИЙ : специальность 1-45 80 01 «Системы и сети инфокоммуникаций» : автореферат диссертации на соискание степени магистра / Новиков Алексей Алексеевич. – Минск, 2024. – 13 с.
4. Милкова, М.А. ТЕМАТИЧЕСКИЕ МОДЕЛИ КАК ИНСТРУМЕНТ «ДАЛЬНЕГО ЧТЕНИЯ» / М.А. Милкова // Цифровая экономика : электронный журнал. – URL: http://digital-economy.ru/arkhiv-zhurnala/ds. – Дата публикации: 07.05.2019.
5. Коршунов, Антон Тематическое моделирование текстов на естественном языке / Антон Коршунов, Андрей Гомзин // Труды ИСП РАН : электронный журнал. – URL: https://www.ispras.ru/proceedings/docs/2012/23/isp\_23\_2012\_215.pdf. – Дата публикации: 2012. – ISSN 2220-6426
6. Митрофанова О.А., Атугодаге М.М. Динамическое тематическое моделирование русскоязычного корпуса юридических документов // Terra Linguistica. 2023. Т. 14. № 1. С. 70–87. DOI: 10.18721/JHSS.14107
7. Краснов, Ф.В, ОЦЕНКА ПРИКЛАДНОГО КАЧЕСТВА ТЕМАТИЧЕСКИХ МОДЕЛЕЙ ДЛЯ ЗАДАЧ КЛАСТЕРИЗАЦИИ / Ф.В, Краснов, Е.Н. Баскакова, И.С. Смазневич // Вестник Томского государственного университета. Управление, вычислительная техника и информатика : электронный журнал. – URL: https://cyberleninka.ru/article/n/otsenka-prikladnogo-kachestva-tematicheskih-modeley-dlya-zadach-klasterizatsii. – Дата публикации: 2021. – ISSN 2311-2085
8. Grootendorst M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure //arXiv preprint arXiv:2203.05794. – 2022
9. Röder M., Both A., Hinneburg A. Exploring the space of topic coherence measures // Proceedings of the Eighth ACM Interna-tional Conference on Web Search and Data Mining. 2015. С. 399–408.
10. Нокель, М. А. ТЕМАТИЧЕСКИЕ МОДЕЛИ: ДОБАВЛЕНИЕ БИГРАММ И УЧЕТ СХОДСТВА МЕЖДУ УНИГРАММАМИ И БИГРАММАМИ / М. А. Нокель, Н. В. Лукашевич // Вычислительные методы и программирование. – 12.03.2015. – Т. 16, № 2. – С. 215–234.
11. Optimizing Semantic Coherence in Topic Models / Mimno David, M. Wallach Hanna, McCallum Andrew [и др.] // Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing . – Edinburgh, Scotland, UK : Association for Computational Linguistics, 2011. – С. 262–272.
12. BERTopic : сайт. – URL: https://maartengr.github.io/BERTopic (дата обращения: 15.12.2024)
13. Федеральная служба государственной статистики : сайт. – URL: https://rosstat.gov.ru/folder/11194 (дата обращения: 15.12.2024)

# ПРИЛОЖЕНИЯ

## Приложение 1

Структура данных для индекса article в Elasticsearch

{

  "title": {

    "type": "text"

  },

  "authors": {

    "type": "text"

  },

  "journal": {

    "type": "text"

  },

  "year": {

    "type": "integer"

  },

  "field\_of\_science": {

    "type": "text"

  },

  "keywords": {

    "type": "object",

    "properties": {

      "russian": {

        "type": "text"

      },

      "english": {

        "type": "text"

      }

    }

  },

  "annotation": {

    "type": "text"

  },

  "citations": {

    "type": "object",

    "properties": {

      "gost": {

        "type": "text"

      },

      "apa": {

        "type": "text"

      }

    }

  },

  "link": {

    "type": "keyword"

  },

  "topic\_number": {

    "type": "integer"

  }

}

## Приложение 2

Тематические разделы КиберЛенинки

Список разделов, предоставленных экспертами:

1. Науки о Земле и смежные экологические науки
2. Биологические науки
3. Химические технологии
4. Экологические биотехнологии
5. Агробиотехнологии
6. Социальная и экономическая география

## Приложение 3

Ключевые слова для фильтрации

Список ключевых слов, предоставленных экспертами:

1. Климатические изменения / Изменения климата
2. Парниковый газ / Парниковые газы
3. Эмиссия парниковых газов / Эмиссия углерода
4. Углеродные единицы
5. Потенциал секвестрации / Секвестрационный потенциал
6. Адаптация к изменениям климата
7. Климатические проекты
8. Низкоуглеродное развитие
9. Углеродная нейтральность
10. Климатическая политика
11. Климатически активные газы
12. Углеродный баланс
13. Технологии улавливания углерода / Парниковых газов
14. Природно-климатические решения
15. Карбоновый полигон
16. Углеродный след
17. Карбоновая тематика

## ПРИЛОЖЕНИЕ 4

Предобработка данных

import pandas as pd

import re

from pymorphy3 import MorphAnalyzer

from stop\_words import get\_stop\_words

from razdel import tokenize

from langdetect import detect

from langdetect.lang\_detect\_exception import LangDetectException

stop\_words = set(get\_stop\_words("ru"))

morph = MorphAnalyzer()

df = pd.read\_csv('data.csv', delimiter=',')

df['text'] = df['Title'].astype(str) + " " + df['Russian Keywords'].astype(str) + " " + df['Annotation'].astype(str)

df.dropna(subset=['text'], inplace=True)

def extract\_russian\_sentences(text):

    sentences = re.split(r'[.!?]', text)

    russian\_sentences = []

    for sentence in sentences:

        try:

            if detect(sentence.strip()) == 'ru':

                russian\_sentences.append(sentence.strip())

        except LangDetectException:

            continue

    return " ".join(russian\_sentences)

df['text'] = df['text'].apply(extract\_russian\_sentences)

def is\_russian\_text(text):

    try:

        return detect(text) == 'ru'

    except LangDetectException:

        return False

df = df[df['text'].apply(is\_russian\_text)]

def preprocess\_text\_for\_bertopic(text):

    text = re.sub(r'[^\w\s]', '', text.lower())

    tokens = [

        morph.parse(token.text)[0].normal\_form

        for token in tokenize(text)

        if token.text.isalpha() and token.text not in stop\_words

    ]

    return " ".join(tokens)

df['processed\_text'] = df['text'].apply(preprocess\_text\_for\_bertopic)

texts = df['processed\_text'].tolist()

texts = [text for text in texts if text.strip()]

print(f"Обработано текстов для BERTopic: {len(texts)} из {len(df)}")

## ПРИЛОЖЕНИЕ 5

Инициализация и обучение модели

#bertopic\_3

from umap import UMAP

from hdbscan import HDBSCAN

from bertopic import BERTopic

from sentence\_transformers import SentenceTransformer

from sklearn.feature\_extraction.text import CountVectorizer

from bertopic.vectorizers import ClassTfidfTransformer

from bertopic.representation import KeyBERTInspired

from bertopic.representation import MaximalMarginalRelevance

embedding\_model = SentenceTransformer("intfloat/multilingual-e5-large-instruct", device=device)

umap\_model = UMAP(n\_neighbors=10, n\_components=3, min\_dist=0.1, metric='cosine', random\_state=86)

hdbscan\_model = HDBSCAN(min\_cluster\_size=5, min\_samples=5, metric='euclidean', algorithm='best', cluster\_selection\_method='eom', prediction\_data=True)

vectorizer\_model = CountVectorizer(ngram\_range=(1, 2))

ctfidf\_model = ClassTfidfTransformer(reduce\_frequent\_words=True)

mmr = MaximalMarginalRelevance(diversity=0.4)

kb = KeyBERTInspired()

representation\_models = [mmr, kb]

topic\_model = BERTopic(embedding\_model=embedding\_model, umap\_model=umap\_model, hdbscan\_model=hdbscan\_model, vectorizer\_model=vectorizer\_model, ctfidf\_model=ctfidf\_model, representation\_model=representation\_models, nr\_topics="auto")

topics, probabilities = topic\_model.fit\_transform(texts)

## Приложение 6

Функция формирования и отправки запроса в Elasticsearch

@app.get("/search")

def search\_articles(

        query: str = Query(None, description="Search query"),

        page: int = Query(1, ge=1, description="Page number (starting from 1)"),

        size: int = Query(100, ge=1, le=1000, description="Number of results per page (max 1000)")

):

    try:

        # Формируем запрос к Elasticsearch

        search\_query = {

            "query": {

                "match\_all": {}

            }

        } if not query else {

            "query": {

                "multi\_match": {

                    "query": query,

                    "fields": [

                        "title",

                        "authors",

                        "journal",

                        "keywords.russian",

                        "annotation"

                    ],

                    "operator": "and"

                }

            }

        }

        # Параметры пагинации

        from\_ = (page - 1) \* size

        # Выполнение запроса

        response = es.search(index=ELASTICSEARCH\_INDEX, body=search\_query, from\_=from\_, size=size)

        # Форматирование результатов

        results = [

            {

                "id": hit["\_id"],

                "title": hit["\_source"].get("title", ""),

                "authors": hit["\_source"].get("authors", ""),

                "journal": hit["\_source"].get("journal", ""),

                "year": hit["\_source"].get("year", ""),

                "field\_of\_science": hit["\_source"].get("field\_of\_science", ""),

                "keywords": hit["\_source"].get("keywords", {}),

                "annotation": hit["\_source"].get("annotation", ""),

                "citations": hit["\_source"].get("citations", {}),

                "link": hit["\_source"].get("link", "")

            }

            for hit in response["hits"]["hits"]

        ]

        return results

    except Exception as e:

        raise HTTPException(status\_code=500, detail=f"Error occurred: {str(e)}")

## ПРИЛОЖЕНИе 7

Код обработчика отправки формы поиска на странице «О статьях»

const handleSearchSubmit = (e) => {

        e.preventDefault();

        if (searchQuery.trim() === '') return;

        setPage(1);

        setArticlesData([]);

        const fetchArticlesForSearch = async () => {

            setIsLoading(true);

            try {

                const response = await fetch(`${API\_URL}/search?query=${searchQuery}&page=1&size=50`);

                if (!response.ok) {

                    throw new Error('Ошибка при загрузке данных с сервера');

                }

                const data = await response.json();

                setArticlesData(data);

                setHasMore(data.length === 50);

            } catch (err) {

                setError('Не удалось загрузить статьи. Пожалуйста, попробуйте позже.');

                console.error('Ошибка запроса:', err);

            } finally {

                setIsLoading(false);

            }

        };

        fetchArticlesForSearch(); };