

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»

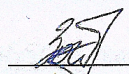
Магистерская программа: «Вычислительные машины, комплексы, системы и сети»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему

СИСТЕМА ЛИНГВИСТИЧЕСКОГО
ОПРЕДЕЛЕНИЯ ПОДОБИЯ ОПИСАНИЯ
ПУБЛИКАЦИЙ

Студент


(подпись, дата)

Забородина Ксения Андреевна
(ФИО полностью)

Руководитель


(подпись, дата)

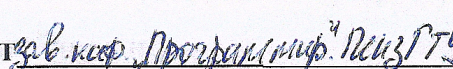
А.Н. Токарев
(фамилия, инициалы)

Нормоконтролёр


(подпись, дата)

А.С. Бычков
(фамилия, инициалы)

Рецензент


(должность, место работы) (подпись, дата)

А.И. Мартышкин
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 02.06.2022 № 13)

Заведующий кафедрой

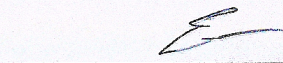

(подпись)

М.А. Митрохин
(фамилия, инициалы)

Работа защищена с отметкой Отлично (протокол заседания ГЭК от

14 июля 2022 № 1)

Секретарь ГЭК


(подпись)

Е.И. Гурин
(фамилия, инициалы)

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
« 21 » сентября 20 20 г.

**ЗАДАНИЕ
НА МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ**

1. Студент Забродина Ксения Андреевна гр. 20ВВм1 факультета ВТ
2. Тема работы «Система лингвистического определения подобия описания публикаций»
Тема утверждена приказом ПГУ №656/о от "21" сентября 2020 г.
3. Руководитель магистерской диссертации к.т.н., доцент Токарев А.Н.
4. Задание на работу (назначение разработки, исходные данные и т.п.)

Назначение разработки: разработка системы лингвистического определения подобия описания публикаций.

Разрабатываемая система должна включать в себя алгоритм составления рекомендательных подборок с применением корректирующих коэффициентов и методов определения уровня подобия.

5. Перечень подлежащих разработке вопросов
 1. Анализ предметной области и критериев оценки подобия публикаций.
 2. Выявление методов, используемых для формирования релевантного списка подобных публикаций.
 3. Разработка системы лингвистического определения подобия описания публикаций.
 4. Анализ полученных результатов.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области и критериев оценки подобия публикаций	14%	21.09.2020 22.12.2020	
2.	Анализ опыта построения алгоритмов прогнозирования	17%	23.12.2020 04.03.2021	
3.	Разработка системы лингвистического определения подобия описания публикаций	34%	05.03.2021 29.11.2021	
4.	Отладка модуля определения подобия	13%	30.11.2021 21.01.2022	
5.	Анализ полученных результатов исследования	12%	22.01.2022 06.04.2022	
6.	Оформление выпускной работы	10%	07.04.2022 31.05.2022	

Дата выдачи задания "21" сентября 2020 г.

Руководитель магистерской диссертации  21.09.2020 (подпись, дата) Токарев А.Н.

Задание к исполнению принял студент  21.09.2020 (подпись, дата) Забродина К.А.

Работу к защите допустить
Декан факультета  (подпись, дата) Фионова Л.Р.

10.06.2022

РЕФЕРАТ

Пояснительная записка содержит 83 листа, 12 рисунков, 7 таблиц, 11 источников, 2 приложения.

РАССТОЯНИЕ ЛЕВЕНШТЕЙНА, РЕЛЕВАНТНЫЕ ПОДБОРКИ,
ХОЛОДНЫЙ СТАРТ, КОНТЕНТНАЯ ФИЛЬТРАЦИЯ,
КОРРЕКТИРУЮЩИЕ КОЭФФИЦИЕНТЫ

Объектом исследования является описание публикации в базе знаний.

Предметом исследования является система лингвистического определения подобия описания публикаций.

Целью исследования данной работы является разработка рекомендательного модуля, содержащего в себе функции вычисления уровня подобия слов, с целью предоставления пользователю релевантной подборки по интересующей его теме.

Научная новизна состоит в модификация существующих алгоритмов подбора рекомендаций посредством применения корректирующих коэффициентов и методов определения уровня подобия.

Результатом данной работы является система лингвистического определения подобия описания публикаций, позволяющая ускорить и повысить эффективность поиска релевантных источников в базах знаний.

					ПГУ 1.090401.02.001 ПЗ			
Изм.	Лист	№ докум	Подпись	Дата				
Разраб.		Забродина К.А.		31.05.22	Система лингвистического определения подобия описания публикаций	Лит.	Лист	Листов
Провер.		Токарев А.Н.		31.05.22			2	83
Н. Контр.		Бычков А.С.		31.05.22	Пояснительная записка	ФВТ, каф. ВТ гр. 20ВВм1		
Утв.		Митрохин М.А.		31.05.22				

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	5
ВВЕДЕНИЕ	6
1 Анализ предметной области	8
1.1 Обзор существующих научных электронных библиотек	8
1.1.1 eLIBRARY.RU	8
1.1.2 Scopus	10
1.1.3 Web of Science	11
1.2 Обзор основных видов рекомендательных систем	13
1.2.1 Контентная фильтрация	15
1.2.2 Коллаборативная фильтрация	18
1.2.3 Фильтрация, основанная на знаниях	20
1.2.4 Опыт построения алгоритмов прогнозирования	21
1.3 Обзор алгоритмов лингвистического определения подобия текстовых данных	22
1.4 Выводы по разделу	25
2 Разработка системы лингвистического определения подобия описания публикаций	26
2.1 Исходные данные для построения подсистемы определения лингвистического подобия. Описание реляционной модели	26
2.2 Выходные данные для построения подсистемы определения лингвистического подобия	31
2.3 Описание алгоритма определения подобия	31
2.4 Выводы по разделу	35
3 Результаты работы системы лингвистического определения подобия описания публикаций	36
3.1 Описание API получения исходных данных	36
3.2 Описание API подсистемы лингвистического определения подобия для осуществления межсистемного взаимодействия	47

3.3 Тестирование системы лингвистического определения подобия описания публикаций	49
3.4 Выводы по разделу	61
ЗАКЛЮЧЕНИЕ	62
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	63
ПРИЛОЖЕНИЕ А Листинги программы	65
ПРИЛОЖЕНИЕ В Презентация	73

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

БД – база данных

РИНЦ – российский индекс научного цитирования

СУБД – система управления базами данных

API – (от англ. Application programming interface) – программный интерфейс приложения

CMS – (от англ. Content Management System) – это система контроля данных

HTTP – (от англ. HyperText Transfer Protocol – «протокол передачи гипертекста») – протокол прикладного уровня передачи данных, изначально – в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных

JSON – (от англ. JavaScript Object Notation) – формат обмена данными, который основан на языке программирования JavaScript

REST – (от англ. Representational State Transfer – «передача состояния представления») – архитектурный стиль взаимодействия компонентов распределённого приложения в сети

SQL – (от англ. Structured Query Language) – язык структурированных запросов

TF-IDF – (от англ. TF — term frequency, IDF – inverse document frequency) – статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса

UML – (от англ. Unified Modeling Language) – язык графического описания для объектного моделирования в области разработки программного обеспечения

WoS – (от англ. Web of Science) – поисковая интернет-платформа, объединяющая реферативные базы данных публикаций в научных журналах и патентов, в том числе базы, учитывающие взаимное цитирование публикаций

ВВЕДЕНИЕ

Начальным этапом научно-фундаментальных или прикладных исследований является выявление существующих решений исследуемого вопроса. При этом важным источником информации являются научные труды, материалы которых можно найти в специализированных базах знаний. Поиск необходимых публикаций в данных системах, как правило, осуществляется на основании только одного или нескольких критериев, предоставляемых пользователем, вследствие чего возникает проблема медленного и мало эффективного подбора релевантных источников. Решением данного вопроса является создание системы лингвистического определения подобия описания публикаций, которая позволит исследователям получить интересующую их информацию, не осуществляя фактического поиска конкретного научного труда, и, благодаря этому, повысить востребованность целевой базы знаний.

Подбор релевантной информации планируется достигнуть путем реализации рекомендательного модуля, позволяющего осуществлять многокритериальный поиск публикаций, основанный не только на параметрах, передаваемых пользователем, но и на ранее собранных данных о его взаимодействии с системой.

При создании подобного продукта необходимо учитывать возможность отсутствия информации о пользователе, а также дату публикации научного труда с целью применения наиболее актуальной научно-исследовательской базы.

Объект исследования: описания публикации в базе знаний.

Предмет исследования: система лингвистического определения подобия описания публикаций.

Цель исследования: разработка рекомендательного модуля, содержащего в себе функции вычисления уровня подобия слов, с целью предоставления пользователю релевантной подборки по интересующей его теме.

Задачи исследования:

- анализ критериев оценки подобия публикаций,
- выявление методов, используемых для формирования релевантного списка подобных публикаций,
- выявление единой шкалы подобия, учитывающей все критерии оценки схожести публикаций,
- разработка системы лингвистического определения подобия описания публикаций.

Актуальность (новизна) исследования: создание системы лингвистического определения подобия описания публикаций, позволяющей классифицировать объекты интереса на основе предпочтений пользователя, а также учитывать дату публикации научного труда для добавления в подборку наиболее актуальной информации.

Практическая и научная значимость: модификация существующих алгоритмов подбора рекомендаций посредством применения корректирующих коэффициентов и методов определения уровня подобия.

Границы исследования: основной акцент ставится на изучении подходов формирования релевантных подборок, а также создании алгоритма лингвистического определения подобия описания публикаций. В границы исследования не входят такие вопросы, как способы хранения и обмена данных вне рекомендательного модуля.

1 Анализ предметной области

1.1 Обзор существующих научных электронных библиотек

Начальным этапом научно-фундаментальных или прикладных исследований является выявление существующих решений исследуемого вопроса. Помощниками в данной задаче являются научные электронные библиотеки, позволяющие не только узнать детальную информацию о публикации, журнале, конференции и авторах, но и просмотреть полный текст статьи, тем самым предоставляя возможность более углубленно изучить интересующую тематику.

Среди наиболее популярных баз знаний выделяют следующие:

- eLIBRARY.RU,
- Scopus,
- Web of Science.

1.1.1 eLIBRARY.RU

Платформа eLIBRARY.RU является одной из самых крупных баз знаний научных публикаций России. Данная система предоставляет возможность поиска научной информации, опубликованной не только под российским индексом научного цитирования, но и входящих в ядро РИНЦ.

Преимуществами данной платформы являются:

- бесплатный открытый доступ к большинству научных трудов с возможностью просмотра полного текста публикации,
- поиск цитируемых и цитирующих публикаций,
- просмотр библиометрических показателей авторов,
- просмотр статистики публикационной активности авторов и научных журналов.

Основными недостатками данной системы являются:

- узконаправленный и малоэффективный поиск,
- отсутствие сортировки результатов поиска,
- недоступность части контента ввиду наличия платных подписок,

– высокая вероятность некорректной работы алгоритма привязки публикации к конкретному автору,

– отсутствие возможности подбора рекомендаций на основе предпочтений пользователя.

Интерфейс научной электронной библиотеки eLIBRARY.RU представлен на рисунке 1.

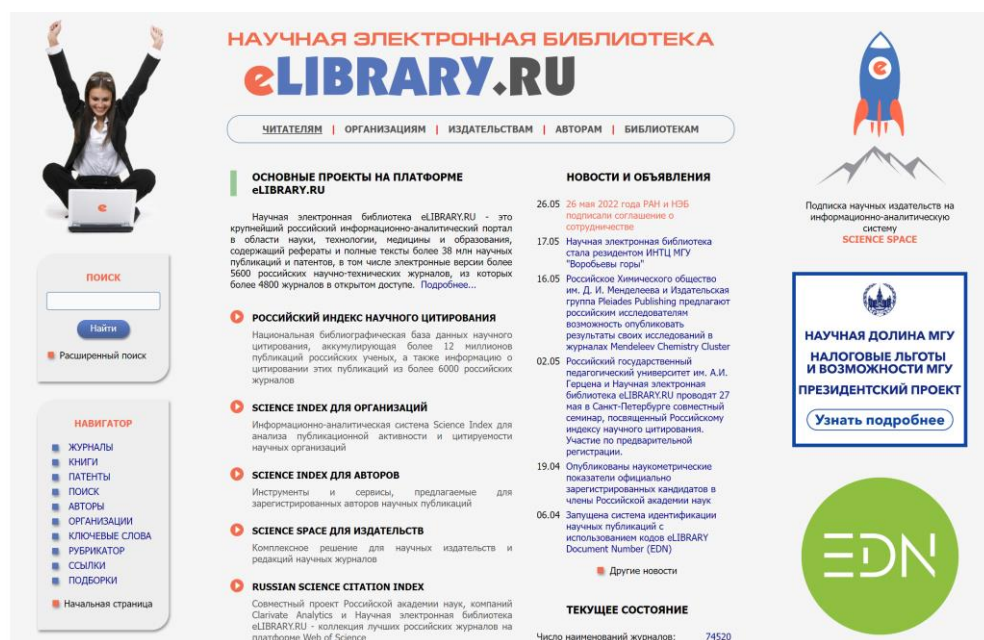


Рисунок 1 – Интерфейс электронной библиотеки eLIBRARY.RU

Интерфейс публикационной активности автора представлен на рисунке 2.

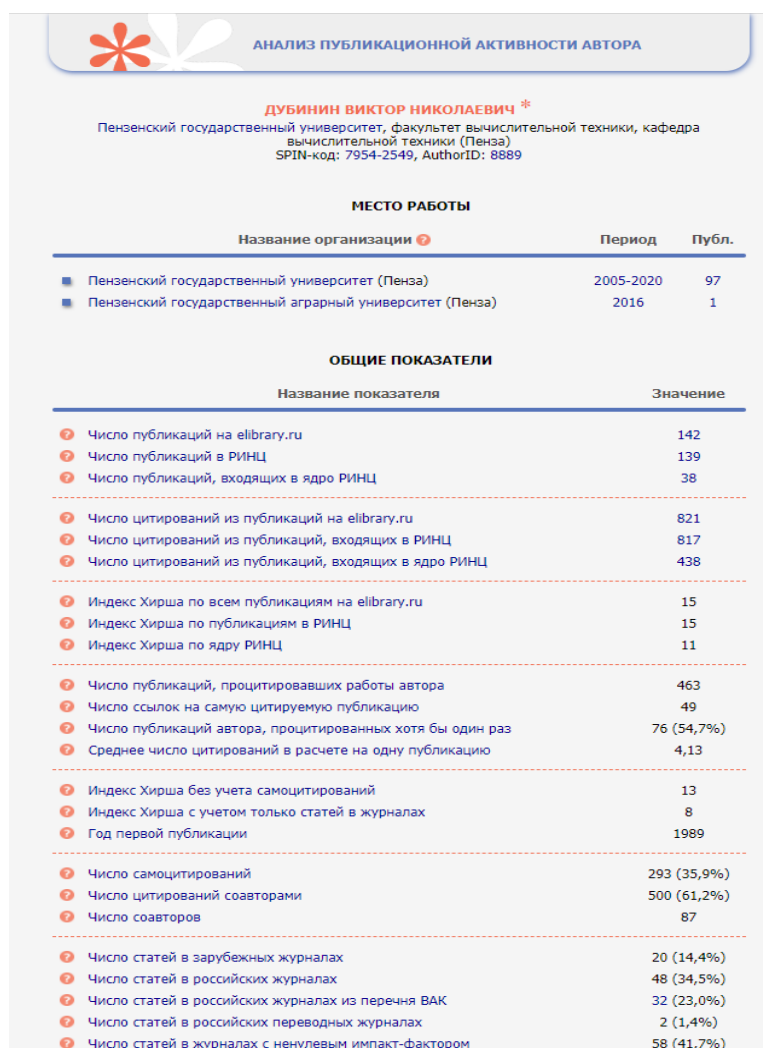


Рисунок 2 – Интерфейс публикационной активности автора в eLIBRARY.RU

1.1.2 Scopus

Международная реферативная база знаний Scopus предоставляет возможность поиска научной литературы по различным направлениям. Данная система содержит аннотацию, краткое описание и ключевые слова, а также информацию о количестве цитирований публикаций. Кроме того, платформа предоставляет анализ данных о научной активности авторов или изданий и их визуализацию.

Преимуществами данной системы является: возможность ознакомления с данными зарубежных авторов, публикации которых представлены в данной базе знаний, эффективный поиск по необходимым параметрам, наличие графического представления публикационной активности авторов.

К отрицательным чертам можно отнести:

- необходимость платной подписки для доступа к большей части контента,
- отсутствие возможности подбора рекомендаций на основе предпочтений пользователя.

Интерфейс реферативной базы знаний Scopus представлен на рисунке 3.

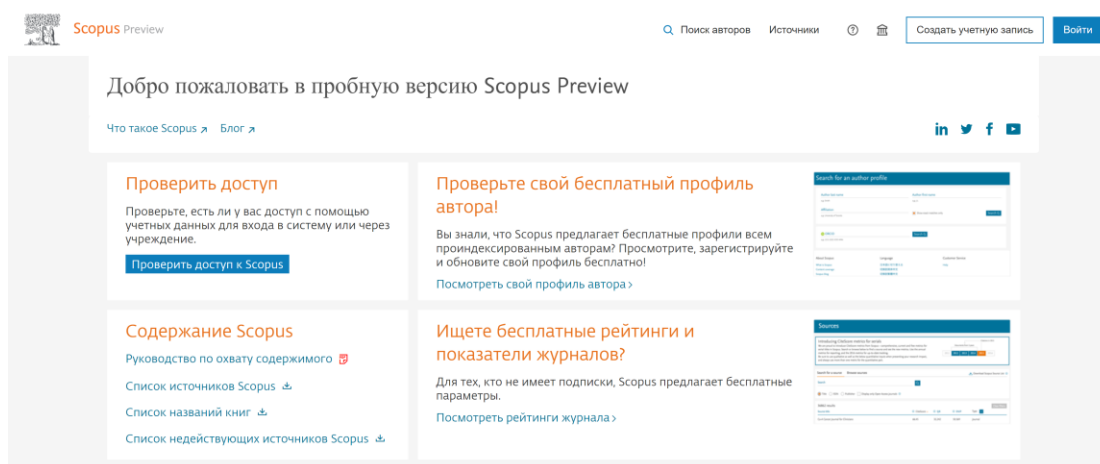


Рисунок 3 – Интерфейс электронной библиотеки Scopus

Интерфейс публикационной активности автора представлен на рисунке 4.



Рисунок 4 – Интерфейс публикационной активности автора Scopus

1.1.3 Web of Science

Web of Science (WoS) – ведущая наукометрическая база данных, предоставляющая возможность поиска и анализа, а также управления библиографической информацией.

Главным преимуществом базы знаний является большой объем научных трудов, накопленной за долгий период существования платформ, и

интерактивное и детальное представление публикационной активности авторов и изданий.

К недостаткам системы можно отнести: отсутствие полных текстов статей, необходимость платной подписки для доступа к данным, а также отсутствие рекомендательного модуля на основе интересов пользователя.

Интерфейс реферативной базы знаний WoS представлен на рисунке 5.

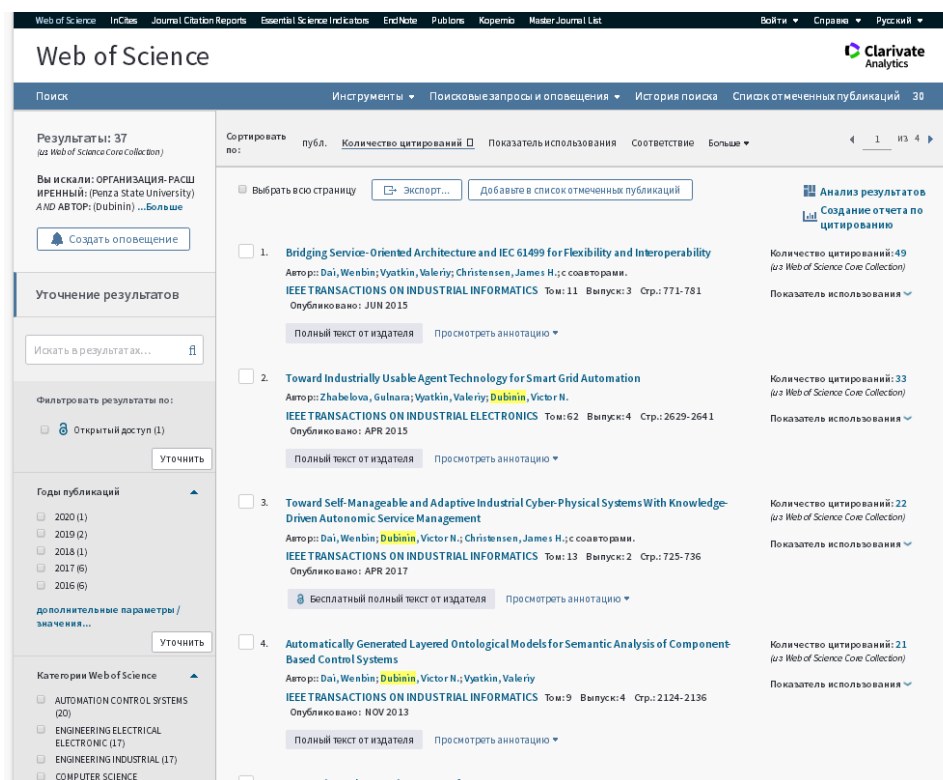


Рисунок 5 – Интерфейс электронной библиотеки WoS

Интерфейс публикационной активности автора представлен на рисунке 6.

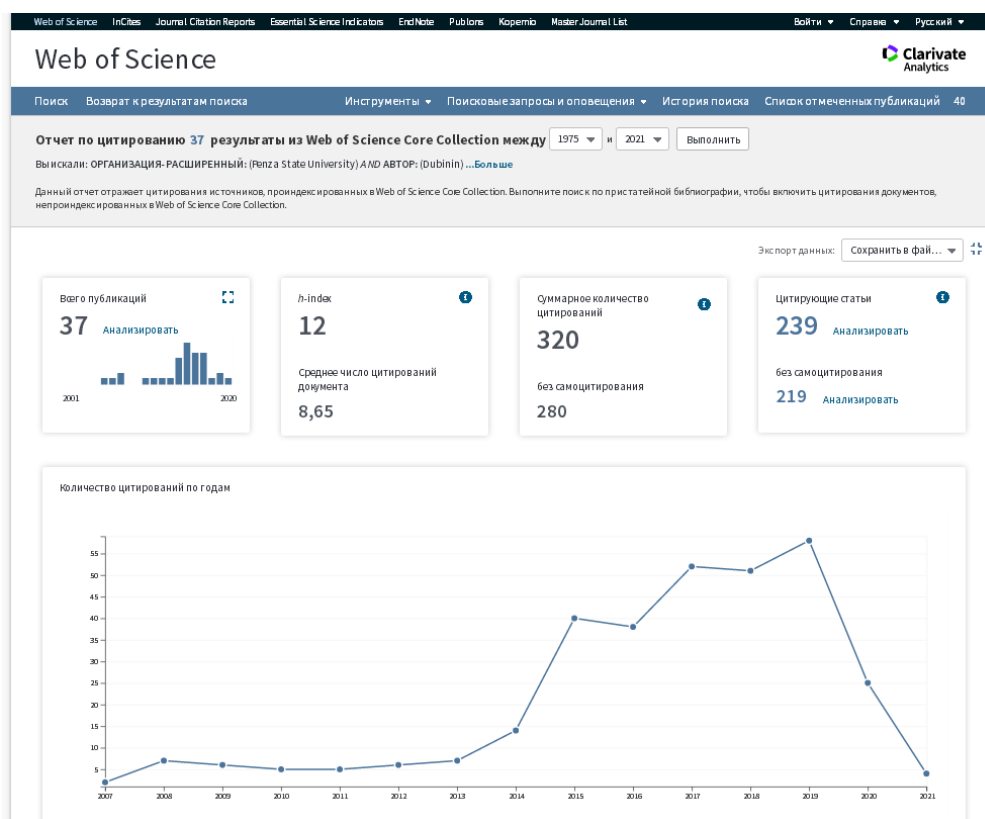


Рисунок 6 – Интерфейс публикационной активности автора WoS

На основании приведенного анализа существующих баз знаний, можно сделать вывод о том, что каждая из этих систем нацелена на поиск информации по основным параметрам, исключая возможность предоставления системой релевантных подборок публикаций на основе имеющейся информации о взаимодействии пользователя с объектами интереса. Однако практика применения рекомендательных алгоритмах в информационных системах таких, как Яндекс.Музыка, Netflix и Storytel, доказало свою эффективность и позволило значительно увеличить востребованность данных платформ.

1.2 Обзор основных видов рекомендательных систем

Системы определения подобиya формируют рейтинговые прогнозы на основе собранной информации, получение которой может происходить явно или неявно [1]. В первом случае данные добываются посредством сбора оценок пользователей, во втором – на основе его взаимодействия с системой. Рейтинговые механизмы, оперируя полученными данными, производят

оценку уровня релевантности того или иного продукта или услуги, с последующим формированием массива наиболее подходящих конкретному пользователю объектов интереса.

В настоящий момент огромное количество коммерческих систем, а также социальных сетей внедрили рекомендательные системы в свои продукты. В следствии чего, можно говорить о разнообразии области применения рекомендательных систем и подходов к их реализации [2].

Существующие в настоящее время подходы к формированию релевантных подборок подразделяются на два вида:

- 1) основанные на моделях, где сначала создается модель предпочтений пользователя, объектов интереса и взаимосвязей между ними, после чего на ее основе вычисляются оценки,
- 2) основанные на данных, где прогнозирование рейтингов происходит в результате вычисления некоторой меры схожести по всем накопленным данным.

Механизмы реализации прогнозов зависят от исходных данных и методов их обработки. Выделяют следующие типы исходных данных:

- социальные, объединяющие информацию о всех пользователях системы, а также формирующие коллективные паттерны поведения,
- индивидуальные, включающие в себя данные о целевом пользователе,
- знания о рекомендуемых объектах интереса и об их использовании.

Подходы, базирующиеся на моделях, в сравнении с алгоритмами на основе данных являются более точными при вычислении оценок подобия, однако применение данных механизмов может привести к необходимости постоянного переобучения модели, занимающего значительный период времени, при условии наличия большой, постоянно обновляющейся базы объектов [3].

Главный принцип методов построения рекомендательных систем на основе данных заключается в прогнозировании рейтинга на базе накопленных

пользовательских предпочтений, что позволяет достигнуть высокого уровня точности прогнозирования, а также адаптироваться под изменения объектов интереса.

Исходя из выше описанного, целесообразно рассматривать в рамках данной работы и поставленной задачи именно методы, основанные на данных.

В данных алгоритмах применяются метрики подобия, позволяющие рассчитать уровень «похожести» между пользователями или объектами интереса на основе их взаимодействия. Выделяют четыре основных вида алгоритмов, позволяющих реализовать рекомендательную систему на основе данных:

- базирующийся на контенте,
- базирующийся на коллаборативной фильтрации,
- базирующийся на знании.

Наглядное представление классификации рекомендательных механизмов, основанных на данных, отражено на рисунке 7.

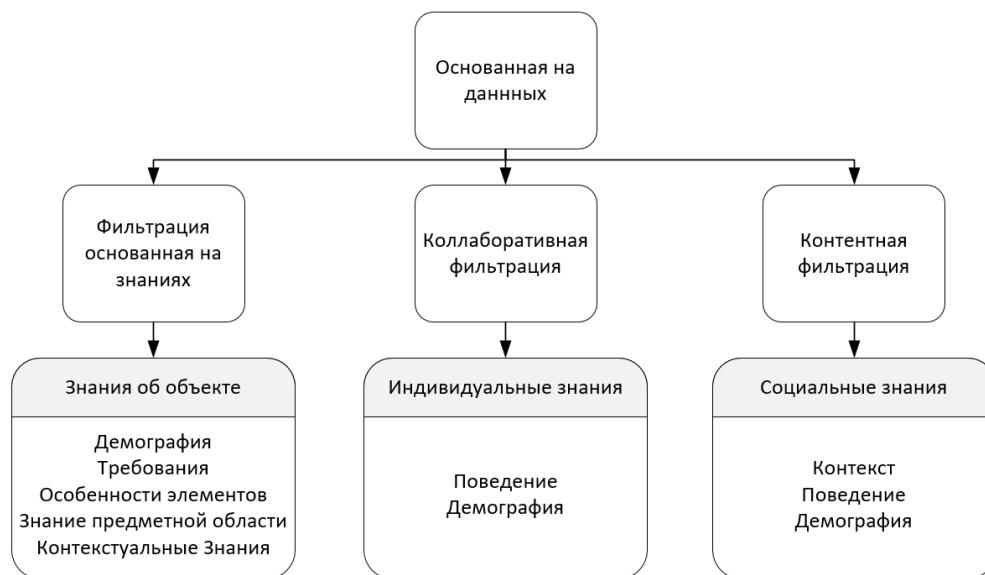


Рисунок 7 – Классификация механизмов прогнозирования основанных на данных

1.2.1 Контентная фильтрация

Алгоритм фильтрации, базирующийся на контенте, заключается в поиске объектов интересов, подходящих под заданные пользователем

характеристики, при этом принято считать, что характеристики не изменяемы с течением времени. В основном данный метод применяется для рекомендации элементов, содержащих текстовую информацию, таких как статьи, новостные сообщения и т.п.

В качестве входных параметров в данном алгоритме передается массив характеристик пользователей и массив категорий, каждый из элементов которого описывает конкретный вид объектов интереса. Основной целью данного алгоритма является поиск наиболее подходящих, к заданной пользователем категории, объектов интереса.

В основе данного метода лежит анализ набора данных об уже существующих результатах взаимодействия пользователя с системой. Механизм ранжирования в данном случае базируется на поиске заданных ключевых характеристик в содержимом объекта интереса и сопутствующем ему описании. Преимущество фильтрации на основе контента по сравнению с традиционными подходами к поиску информации заключается в обработке объектов интереса по не персонализированным данным о предпочтениях пользователя.

Если описать данный механизм более формально, то получается, что у объекта интереса s имеется набор атрибутов $Content(s)$, из которого извлекается набор необходимых признаков и определяется уровень релевантности s для конкретного пользователя. Поскольку подобные рекомендательные системы ориентированы на обработку текстовой информации, то, зачастую, в качестве атрибутов поиска выступают ключевые слова. При этом главным показателем при поиске по ключевым словам является мера частоты употребления слова в документе (TF-IDF) [4]. Данная мера определяется следующим образом: допустим, что N – общее число объектов интереса (документов), k_j – заданное ключевое слово, n_i – количество элементов из массива объектов интереса релевантных k_j , $f_{j,i}$ – количество

упоминаний k_j в исследуемом документе d_i . В таком случае, частота $TF_{j,i}$ употребления k_j в d_i определяется как:

$$TF_{j,i} = \frac{f_{j,i}}{\max_z f_{z,i}}, \quad (1)$$

где $\max_z f_{z,i}$ – количество упоминаний всех ключевых слов k_z в d_i .

При этом использование только частоты употребления ключевого слова, зачастую, мало эффективно, поскольку это не помогает отличить объект интереса, соответствующий заданному параметру, от не соответствующего в случае, если ключевое слово встречается в большинстве документов. Тогда, помимо простой частоты используется ещё и обратная частота IDF_j , рассчитываемая для k_j как:

$$IDF_j = \log \frac{N}{n_i} . \quad (2)$$

В таком случае, общая частота $(w_{j,i})$ k_j в d_i вычисляется по формуле:

$$w_{j,i} = TF_{j,i} \times IDF_j. \quad (3)$$

Набор атрибутов d_i определяется как:

$$Content(d_i) = (w_{1,i}, \dots, w_{k,i}) . \quad (4)$$

Механизм составления релевантных рекомендаций, исходя из значения общей частоты употребления ключевого слова в документе, позволяет достигнуть высокого уровня подобия между объектами интереса, что является как преимуществом, так и недостатком рекомендательных систем, основанных на контенте. Достоинство заключается в возможности

рекомендаций новых публикаций, даже если они не были оценены другими пользователями.

Недостатком является то, что система рекомендует только объекты интереса наиболее похожие на заданную категорию, из-за чего у пользователя отсутствует возможность ознакомления с наименее подходящими элементами, но потенциально ему интересными. Также необходимо чтобы слишком похожие объекты не попадали в рекомендательный прогноз, поскольку они могут являться, своего рода, «клонами» исходных объектов заданного подмножества.

На основании выше описанного, можно сделать вывод о необходимости контроля и регулирования пороговых значений прогноза, в случае неправильного подбора которых снижается точность прогнозирования системы.

1.2.2 Коллаборативная фильтрация

Метод, позволяющий создавать рекомендательные прогнозы на основании предшествующих взаимодействий пользователя с системой, называется коллаборативной фильтрацией. Персонализированные данные о предпочтениях пользователя могут быть описаны, как самим пользователем – отзыв, «лайк», покупка и т.п., так и выявлены в результате взаимодействия пользователя с системой – просмотр, клик и т.д.

При прогнозировании данный алгоритм обрабатывает не только действия текущего пользователя, но и других со схожими предпочтениями.

Входными данными для текущего способа являются: массив пользователей и массив объектов интереса. Соотношение между массивами описывается в виде метрики востребованности какого-либо продукта у конкретного потребителя, для чего применяется система оценок или баллов, выставяемых самим пользователем. На основании выставленных значений система осуществляет поиск схожих элементов одного из массивов. От того

какой именно набор данных будет выбран, будет зависеть реализация коллаборативной фильтрации.

В случае если полученный рейтинг объектов интереса используется для нахождения пользователей со схожими предпочтениями, такой метод называется пользовательским [5]. Суть данного механизма заключается, в том, что для текущего пользователя с помощью функции подобия составляется список похожих потребителей, при этом, для оценки схожести рейтинговых векторов применяются специальные показатели, одним из которых является коэффициент корреляции Пирсона. Допустим, что $\bar{A} = (a_1, \dots, a_n)$ и $\bar{B} = (b_1, \dots, b_n)$ – рейтинговые вектора двух пользователей, со средними значениями $\hat{a} = \sum_{i=1}^n a_i/n$ и $\hat{b} = \sum_{i=1}^n b_i/n$ соответственно. В таком случае, коэффициент корреляции Пирсона двух пользователей рассчитывается как:

$$Pearson(\bar{A}, \bar{B}) = \frac{\sum_{i=1}^n (x_i - \hat{x}) \times (y_i - \hat{y})}{\sqrt{\sum_{i=1}^n (x_i - \hat{x})^2} \times \sqrt{\sum_{i=1}^n (y_i - \hat{y})^2}} . \quad (5)$$

После вычисления данного коэффициента для текущего пользователя и остальных клиентов системы формируется группа субъектов с наивысшими показателями. Затем, средние баллы, выставляемые членами данной группы, возвращаются в качестве прогноза о востребованности конкретного объекта интереса у целевого пользователя.

Для повышения надёжности данного метода рекомендуется производить нормализацию всех рейтингов обрабатываемых пользователей путём вычисления среднего значения данного показателя. Данные меры позволяют избегать ситуаций, когда потребители при оценке одного и того же объекта интереса применяют различные диапазоны значений.

Помимо пользовательского метода, нередко находит применение способ, применяющий рейтинговую матрицу для нахождения схожих элементов, называемый контентным. Так же, как и в предыдущем методе,

перед расчётом показателей подобия, производится нормализация оценок, выставяемых пользователями. Затем производится вычисление самого коэффициента схожести элементов.

Допустим, что $\bar{Z} = (z_1, \dots, z_n)$ и $\bar{Y} = (y_1, \dots, y_n)$ являются нормализованными оценками двух объектов интереса, тогда их сходство определяется как:

$$\text{Cosine}(\bar{Z}, \bar{Y}) = \frac{\sum_{i=1}^n z_i \times y_i}{\sqrt{\sum_{i=1}^n z_i^2} \times \sqrt{\sum_{i=1}^n y_i^2}}. \quad (6)$$

Данное сходство называется скорректированным косинусным, поскольку перед его расчётом осуществляется нормализация показателей.

После вычисления данного коэффициента для всех объектов интереса, производится подбор наиболее схожих с текущим элементом. Полученная выборка проверяется на наличие оценок от целевого пользователя, затем вычисляется среднее значение их рейтингов, которое потом представляется как рекомендательный прогноз для всех элементов данной группы от текущего потребителя.

Оба описанных метода позволяют достигнуть высокой точности прогнозирования, но при этом каждый из них требует накопления большого объема целевой информации, без которой прогнозирование будет осуществляться некорректно.

1.2.3 Фильтрация, основанная на знаниях

Рекомендательные системы, основанные на знаниях, в отличие от описанных ранее методов используют более сложную структуру характеристик (правил), описывающих предпочтительные объекты интереса. Несмотря на это, данный способ имеет много общего с методом, основанным на контенте, из-за чего в некоторых классификациях его рассматривается как частный случай.

Принцип работы методов рекомендации, основанных на знаниях, заключается в прогнозировании востребованности того или иного объекта интереса у целевого пользователя на базе знаний в текущей предметной области, а именно на определении степени соответствия назначения элемента потребностям и предпочтениям пользователей. В качестве входных значений данный способ использует множества объектов интереса и правил, описывающих свойства предпочтительных элементов. При этом правила могут быть как прописаны в самой системе, так и заданы пользователем. В последнем случае в рекомендательной системе должны быть предусмотрены ограничения на ввод критериев.

Основной задачей систем на базе знаний является составление рекомендательного прогноза из объектов интереса, удовлетворяющих заданному пользователем набору правил.

Следует отметить, что областью применения данного метода составления рекомендаций являются сервисы, в которых пользователь имеет поверхностное представление об объекте интереса, но несмотря на это требуется достичь максимальной точности прогнозирования [6].

Преимущества данного метода заключаются в возможности обоснования рекомендованного прогноза, т.е. в предоставлении, при необходимости, списка правил, которые удовлетворяют конкретному объекту интереса, входящему в рекомендацию, а также в высоком уровне точности результатов, достигаемых за счёт возможности изменения и комбинирования настроек рекомендации. Но при этом следует учитывать, что данный подход требует больших затрат на реализацию, поскольку возникает необходимость в детальной проработке предметной области и потребность в сборе обширной базы знаний о предметной области, в которой применяется система.

1.2.4 Опыт построения алгоритмов прогнозирования

Несмотря на множество подходов к реализации рекомендательных систем, предоставляющих достаточно точное прогнозирование предпочтений

пользователей, существует ряд серьезных проблем, которые оказывают негативное влияние на качество моделей, а также затрудняют их разработку. Ключевой из них является проблема холодного старта, которая проявляется тогда, когда невозможно дать надежные прогнозы ввиду недостаточности или полного отсутствия входных данных о пользователе или объекте интереса.

Частными случаями являются:

- регистрация нового пользователя в систему, о котором на данный момент ничего не известно и который еще ничего просмотрел,
- добавление новых, еще никем не оцененных товаров или услуг.

Только что зарегистрированные пользователи не могут получить персонализированные рекомендации на основе их предпочтений. В результате чего информационные системы могут иметь убытки, поскольку клиенты не могут найти релевантную для них информацию и покидают систему. Исходя из этого, сервисы должны предоставлять потребителям возможность выбора предпочтений на момент регистрации, после чего предлагать ему подготовленный набор возможных объектов интереса.

К проблеме холодного запуска рекомендательной системы можно отнести недостаточное количество откликов пользователей на объекты интереса.

Для решения проблемы холодного запуска необходимо корректно подобрать алгоритм прогнозирования, а также применить дополнительные параметры, позволяющие игнорировать разреженность данных.

1.3 Обзор алгоритмов лингвистического определения подобию текстовых данных

Лингвистические алгоритмы определения схожести в своей структуре основаны на таких базовых методах, как:

- профилирование и синтаксический анализ текстовой информации с целью определения характерных черт стилистики автора и дальнейшей его идентификации,

- стандартизация с целью приведения текстов к единому формату,
- сопоставление, необходимое для выявления уровня подобия информации.

В основном механизмы сравнения текстовых записей подразделяют на две основные группы:

- детерминированные,
- вероятностные.

Первая категория базируется на строгом сопоставлении всех атрибутов исходных объектов, при этом соответствие одного объекта с другим может варьироваться в зависимости от требований, поставленных системой, т.е. процентное соотношение совпадающих параметров может быть дифференцировано [7]. К недостаткам данного механизма можно отнести: необходимость перенастройки существующих связей между сопоставляемыми объектами, в случае пополнения или изменения набора данных, что может привести к увеличению нагрузки на целевую систему.

Вторая категория основывается на вычислении уровня подобия атрибутов текстовых объектов с последующим определением весового коэффициента для каждого из параметров. При этом достижение связности между ними зависит от порогового значения: чем выше вес относительно заданного, тем больше вероятность корреляции, в противном случае параметры считаются независимыми. Для определения уровня подобия, зачастую, используют такие методы, как: сходство Джаро-Винклера или расстояние Левенштейна.

Алгоритм Джаро-Винклера основан на вычислении расстояния Джаро, т.е. меры сходства между двумя исходными строками. При этом шкала оценки варьируется от 0 до 1, означающая либо отсутствие сходства, либо точное совпадение соответственно.

Расстояние Джаро между двумя строками s_1 и s_2 рассчитывается следующим образом:

$$d_j = \begin{cases} 0 & \text{если } m = 0 \\ \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) & \text{в противном случае} \end{cases}, \quad (7)$$

где d_j – расстояние Джаро между j символами строк s_1 и s_2 ;

m – число совпадающих символов;

t – половина от общего числа транспозиций.

При этом символ из первой строки совпадает с соответствующим символом из второй строки только в случае, когда они являются идентичными и расположены не дальше, чем:

$$\frac{\max(|s_1|, |s_2|)}{2} - 1. \quad (8)$$

Число транспозиций в данном случае определяется как половина от общего количества тождественных символов с разными порядковыми номерами.

К отрицательной особенности данного алгоритма можно отнести низкий уровень эффективности при непосредственной работе с исходными строками ввиду игнорирования их семантики, из чего вытекает необходимость предварительной нормализации. Применение данного метода целесообразно только в тех случаях, когда обрабатываются строки малой длины, в противном случае для получения результатов необходимо продолжительные временные затраты, что негативно влияет на производительность целевой системы в целом.

Второй метод базируется на расстоянии Левенштейна, которое определяет разницу между двумя строками путем вычисления минимального количества операций, требуемых для модификации одной последовательности символов в другую [8].

$$d(i, j) = \begin{cases} \max(i, j) & \text{если } \min(i, j) = 0 \\ \min \begin{pmatrix} d(i-1, j)+1, \\ d(i, j-1)+1, \\ d(i-1, j-1)+m(s_1[i], s_2[j]) \end{pmatrix} & \text{в противном случае} \end{cases}, \quad (9)$$

где $d(i, j)$ – расстояние Левенштейна между строками s_1 и s_2 ;

$m(s_1[i], s_2[j])$ – функция, осуществляющая сравнение двух символов разных строк. В случае их равенства возвращается 0, в противном случае – 1.

В результате полученное значение обратно пропорционально соответствию исходных данных между собой.

Целесообразно использовать данный алгоритм для определения уровня подобия, поскольку входную информацию необходимо проверить не только на количество вхождений конкретного слова в строку, но и его семантических вариаций. Это позволит значительно увеличить точность алгоритма определения подобия.

1.4 Выводы по разделу

Проведя анализ существующих онлайн-библиотек, а также подходов к оставлению рекомендаций, можно сделать вывод о целесообразности реализации рекомендательного модуля, базирующегося не только на собранных ранее данных о взаимодействии пользователя с системой, но и учитывающего разреженность данных ввиду новизны научно-фундаментальных или прикладных исследований.

2 Разработка системы лингвистического определения подобия описания публикаций

Разрабатываемая система должна выдавать релевантные объекты интереса на основе данных о взаимодействии пользователя с системой, а также учитывать лингвистические особенности описания публикаций. В основе разработанного алгоритма лежат методы интеллектуального анализа данных, поскольку они позволяют сформировать релевантные подборки публикаций по интересующей пользователя теме, что и является основной целью выполнения данной работы.

Для достижения поставленных целей в работе используются метод контентной фильтрации, поскольку он позволяет достигнуть высокого уровня подобия между объектами интереса, и вероятностный механизм сравнения текстовых записей – расстояние Левенштейна ввиду возможности эффективной обработки большого объема текстовой информации. Кроме того, для решения проблемы холодного старта в данной системе применяется корректирующий коэффициент новизны.

2.1 Исходные данные для построения подсистемы определения лингвистического подобия. Описание реляционной модели

Исходные данные для построения системы лингвистического определения подобия описания публикаций можно представить в виде реляционной модели данных.

Объекты интереса, рассматриваемые в рамках реализации данной работы, имеют набор универсальных характеристик, позволяющих получить полную информацию о конкретной публикации, к ним относятся:

- наименование,
- список авторов,
- тип публикации,
- год издания,
- количество страниц,

- источник,
- данные конференции,
- ключевые слова,
- аннотация,
- список используемой литературы,
- библиометрические показатели,
- ссылка на полный текст статьи.

Реляционная модель входных данных состоит из следующего набора связанных между собой отношений (таблиц):

- публикаций (Publications),
- авторы (Authors),
- организации (Organizations),
- журналы (Magazines),
- конференции (Conferences),
- выпуск (Issue),
- издательство (Publishing house).

Подробное описание атрибутов таблицы «Publications» представлено в таблице 1.

Таблица 1 – Структура таблицы «Publications»

Атрибут	Описание атрибута
id	идентификатор публикации
title	наименование публикации
authors_ids	авторы публикации
type	тип публикации
annotation	аннотация публикации
keywords	ключевые слова
year	год издания
magazines	журнал
conference	данные конференции
pages	количество страниц

Продолжение таблицы 1

Атрибут	Описание атрибута
publication_ids	список используемой литературы
bibliometric_indicators	библиометрические показатели
link_publication	ссылка на полный текст статьи

Подробное описание атрибутов таблицы «Authors» представлено в таблице 2.

Таблица 2 – Структура таблицы «Authors»

Атрибут	Описание атрибута
id	идентификатор автора
author_id	персональный идентификатор автора
username	имя пользователя
email	почта пользователя
organization_id	идентификатор организации
publications_ids	идентификаторы публикаций

Подробное описание атрибутов таблицы «Organizations» представлено в таблице 3.

Таблица 3 – Структура таблицы «Organizations»

Атрибут	Описание атрибута
id	идентификатор организации
title	наименование организации
authors_ids	идентификаторы авторов

Подробное описание атрибутов таблицы «Magazines» представлено в таблице 4.

Таблица 4 – Структура таблицы «Magazines»

Атрибут	Описание атрибута
id	идентификатор журнала
ISSN	международный стандартный серийный номер
title	наименование журнала
publisher_id	идентификатор издательства

Продолжение таблицы 4

Атрибут	Описание атрибута
year	год основания
topic	тематика
issues_ids	идентификаторы выпусков
impact_factor	импакт фактор журнала

Подробное описание атрибутов таблицы «Conferences» представлено в таблице 5.

Таблица 5 – Структура таблицы «Conferences»

Атрибут	Описание атрибута
id	идентификатор конференции
ISSN	международный стандартный серийный номер
title	наименование конференции
address	место проведения конференции
publisher_id	идентификатор издательства
year	год издания
topic	тематика
issue_ids	идентификаторы выпусков
impact_factor	импакт фактор журнала

Подробное описание атрибутов таблицы «Issue» представлено в таблице 6.

Таблица 6 – Структура таблицы «Issue»

Атрибут	Описание атрибута
id	идентификатор выпуска
year	год издания
val	выпуск
pages	количество страниц
source_id	идентификатор источника
publications_ids	идентификаторы публикаций

Подробное описание атрибутов таблицы «Publishing_house» представлено в таблице 7.

Таблица 7 – Структура таблицы «Publishing_house»

Атрибут	Описание атрибута
id	идентификатор издательства
title	наименование издательства
address	адрес издательства
representatives_ids	представители издательства

Взаимосвязи между объектами и субъектом системы лингвистического определения подобия описания публикаций можно представить в виде реляционной модели, представленной на рисунке 8.

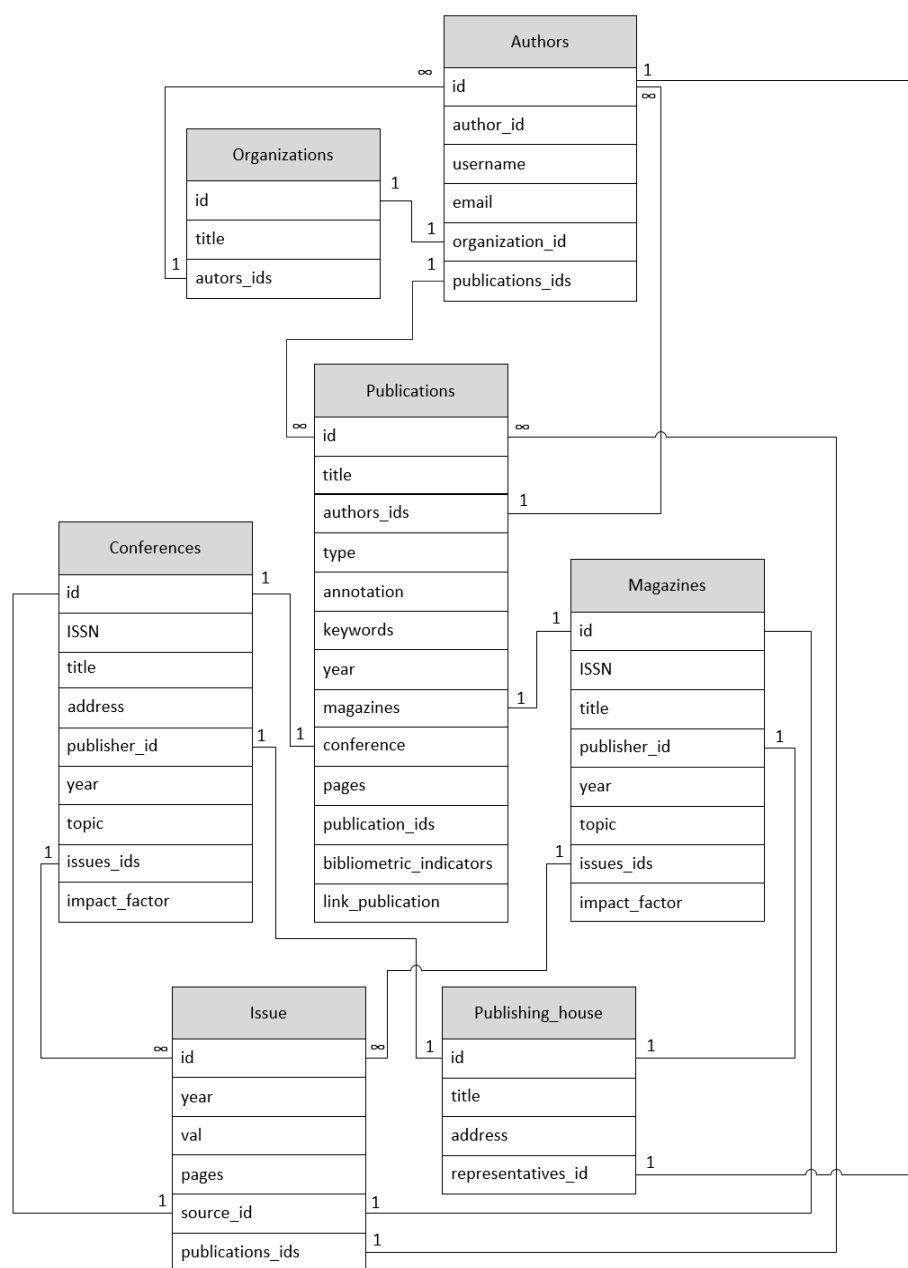


Рисунок 8 – Реляционная модель исходных данных

2.2 Выходные данные для построения подсистемы определения лингвистического подобия

Результатом формирования рекомендательных прогнозов должен являться список идентификаторов публикаций, отсортированных в порядке релевантности, а также дублирование номера страницы, заданного числа элементов и их общее количество.

2.3 Описание алгоритма определения подобия

Начальным этапом формирования подборки публикаций являются использование метода контентной фильтрации, включающего сравнение текстовой информации посредством вычисления расстояния Левенштейна. Завершающим этапом является применение коэффициента новизны к сформированному прогнозу для конечного ранжирования объектов интереса по дате публикации с целью внедрения наиболее актуальных публикаций.

Для сравнения объектов интереса необходимо сформировать массив отношений, который включает в себя комбинации элементов. В дальнейшем на основе данного списка осуществляется вычисление подобия элементов по коэффициенту TF-IDF, которое происходит на основании таких характеристик, как название, описание и ключевых слов. При этом коэффициент TF-IDF рассчитывается как произведение прямой и обратной частоты употребления ключевых слов в параметрах объектов интереса [9].

После чего осуществляется дополнительный синтаксический анализ описания публикаций посредством вычисления расстояния Левенштейна. Это обусловлено дополнительным детальным определением подобия, учитывающим семантическое варьирование, то есть отличие близких по написанию слов. В результате работы механизма прогнозирования строится модель, описывающая отношение между объектами интереса по индексу «схожести».

На заключительной стадии составления подборки рекомендаций (q) применяется коэффициент новизны, представленная в формуле (10), для

новых объектов интереса, имеющих низкую оценку релевантности ввиду отсутствия информации о них.

$$q(k) = y + l_k, \quad (10)$$

где k – текущая информация об объекте интереса относительно пользователя;

y – список атрибутов вместе с условными вероятностями;

l_k – текущее значение коэффициента новизны данного объекта интереса.

Значение коэффициента новизны варьируется в диапазоне от 0 до 1, где 0 соответствует статье, имеющей наиболее давнюю дату опубликования, а 1 – совершенно новой. Рассчитывается данный параметр на предварительной обработке данных путем нормализации даты издания по следующей формуле:

$$l_k = \frac{t_k - t_{min}}{t_{max} - t_{min}}, \quad (11)$$

где t_k – дата публикации текущей статьи;

t_{min} – дата публикации статьи, имеющей наиболее давнюю дату опубликования;

t_{max} – дата публикации новой статьи, имеющей наиболее раннюю дату опубликования.

Реализация механизма классификации была вынесена в отдельный модуль со следующим функционалом:

- вывод информации о начале построения контентной модели,
- фиксация времени начала процесса,
- создание экземпляра класса `sqlalchemy.engine.Engine`, предоставляющего подключение к базе данных,
- подключение к базе данных,

- чтение данных (идентификатор объекта интереса, его описание и версия) из таблицы объекта интереса,
- проверка полученных данных на пустоту. В случае, если данные отсутствуют, вывод соответствующего сообщения и завершение вычислений,
- формирование массива отношений между элементами,
- вычисление коэффициента TF-IDF для всех элементов,
- вычисление расстояния Левенштейна,
- вычисление прогноза для каждого объекта интереса,
- вычисление оценок релевантности (прогноза) на основе коэффициента новизны,
- ранжирование рекомендательных прогнозов,
- вывод информации о записи результатов в базу данных,
- запись результатов в базу данных,
- вывод информации о завершении процесса выявления схожих объектов и времени обучения.

Алгоритм работы контентной фильтрации представлен на рисунке 9.

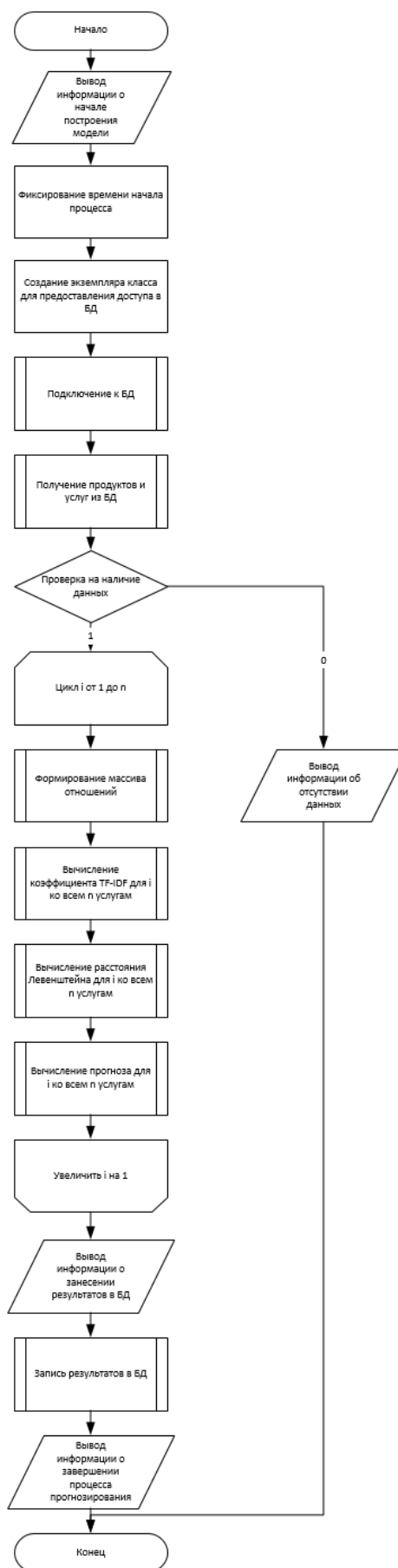


Рисунок 9 – Алгоритм определения подоби́я

2.4 Выводы по разделу

При проектировании системы лингвистического определения подобия описания публикаций был разработан алгоритм определения схожести объектов интереса, сочетающий в себе метод контентной фильтрации, дополнительный синтаксический анализ описания посредством вычисления расстояния Левенштейна и корректирующий коэффициент новизны. Данный рекомендательный модуль позволит решить задачу прогнозирования научных статей на основе предпочтений пользователя, при этом учитывая возможную разреженность данных, а также дату публикации источника.

3 Результаты работы системы лингвистического определения подобия описания публикаций

3.1 Описание API получения исходных данных

Рекомендательный модуль был разработан с применением таких технологий, как Python 3.8, в частности фреймворка FastAPI, поскольку именно данные средства позволяют создать быстрые сервера со встроенными валидацией, сериализацией и асинхронностью. Кроме того, в качестве системы управления базами данных (СУБД) была выбрана объектно-реляционный PostgreSQL ввиду возможности гибкого доступа к БД, хранения и организации данных [10].

Исходная информация, необходимая для построения системы лингвистического определения подобия описания публикаций, запрашивается у сторонней системы управления содержанием CMS посредством REST API в формате JSON [11].

Схема взаимодействия модулей целевой системы представлена на рисунке 10.

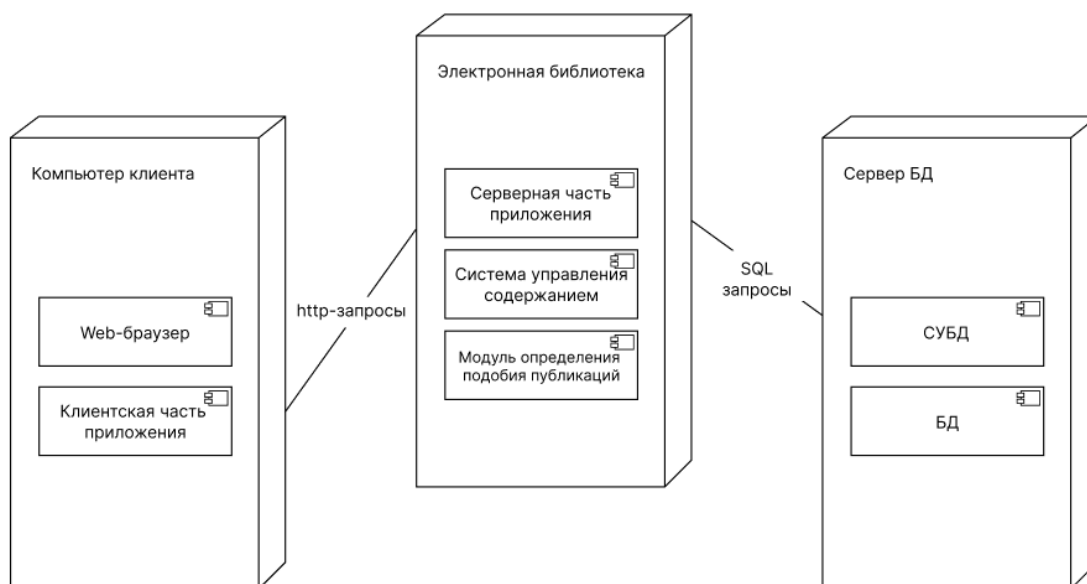


Рисунок 10 – UML диаграмма развертывания целевой системы

Список публикаций. Запрашивается у системы управления содержанием целевой платформы при помощи соответствующего запроса, входными данными для которого являются номер страницы и общее количество публикаций на странице. В случае успешного ответа, рекомендательный модуль получает следующие данные:

- идентификатор публикации,
- наименование публикации,
- авторы публикации,
- тип публикации,
- аннотация публикации,
- ключевые слова,
- год издания,
- журнал,
- данные конференции,
- количество страниц,
- список используемой литературы,
- библиометрические показатели,
- ссылка на полный текст статьи.

Пример описания запроса на получение списка публикаций представлен ниже.

Метод: GET

URL: api/v1/publications

Заголовок запроса: x-app-token: 9f3a3401-e5f2-4a5b-bc8f-ea6a9566a3d7

Тело запроса:

```
{  
  "page": number,  
  "pageSize": number,  
}
```

Ответ:

200 OK

```
{
  "page": number,
  "pageSize": number,
  "total": number,
  "items": [
    {
      "id": "string",
      "title": "string",
      "authors_ids": [{
        "id": "string",
        "name": "string"
      }],
      "type": number,
      "annotation": "string",
      "key_words": ["string"],
      "year": "string"
      "magazines": {
        "id": "string",
        "ISSN": "string",
        "title": "string",
      },
      "conference": {
        "id": "string",
        "title": "string",
        "address": "string",
        "event_end": "string",
        "event_start": "string"
      },
    },
  ],
}
```

```

        "pages": "string",
        "publications": [{
            "id": "string",
            "title": " string",
            "context": string"
        }],
        "bibliometric_indicators": "string",
        "link_publication": "string"
    }
]
}
Bad Request
{ "errorMessage": "string" }

```

Список авторов. Запрашивается у системы управления содержанием целевой платформы при помощи соответствующего запроса. В случае успешного ответа, рекомендательный модуль получает следующие данные:

- идентификатор автора,
- персональный идентификатор автора,
- имя пользователя,
- почта пользователя,
- идентификатор организации,
- идентификаторы публикаций.

Пример описания запроса на получение списка авторов представлен ниже.

Метод: GET

URL: api/v1/authors

Заголовок запроса: x-app-token: 9f3a3401-e5f2-4a5b-bc8f-ea6a9566a3d7

Ответ:

200 OK

```
{
  "items": [
    {
      "id": "string",
      "author_id": "string",
      "username": "string",
      "email": "string",
      "organization_id": "string",
      "publications_ids": ["string"]
    }
  ]
}
```

Bad Request

```
{ "errorMessage": "string" }
```

Список организаций. Запрашивается у системы управления содержанием целевой платформы при помощи соответствующего запроса. В случае успешного ответа, рекомендательный модуль получает следующие данные:

- идентификатор организации,
- наименование организации,
- идентификаторы авторов.

Пример описания запроса на получение списка авторов представлен ниже.

Метод: GET

URL: api/v1/organizations

Заголовок запроса: x-app-token: 9f3a3401-e5f2-4a5b-bc8f-ea6a9566a3d7

Ответ:

200 OK

```

{
  "items": [
    {
      "id": "string",
      "title": "string",
      "authors_ids": ["string"]
    }
  ]
}

Bad Request

{ "errorMessage": "string" }

```

Список журналов. Запрашивается у системы управления содержанием целевой платформы при помощи соответствующего запроса. В случае успешного ответа, рекомендательный модуль получает следующие данные:

- идентификатор журнала,
- международный стандартный серийный номер,
- наименование журнала,
- идентификатор издательства,
- год основания,
- тематика,
- идентификаторы выпусков,
- импакт фактор журнала.

Пример описания запроса на получение списка авторов представлен ниже.

Метод: GET

URL: api/v1/magazines

Заголовок запроса: x-app-token: 9f3a3401-e5f2-4a5b-bc8f-ea6a9566a3d7

Ответ:

200 OK

```

{
  "items": [
    {
      "id": "string",
      "ISSN": "string",
      "title": "string",
      "publisher_id": "string",
      "year": "string",
      "topic": "string",
      "issues_ids": ["string"],
      "impact_factor": "string"
    }
  ]
}

```

Bad Request

```
{ "errorMessage": "string" }
```

Список конференций. Запрашивается у системы управления содержанием целевой платформы при помощи соответствующего запроса. В случае успешного ответа, рекомендательный модуль получает следующие данные:

- идентификатор конференции,
- международный стандартный серийный номер,
- наименование конференции,
- место проведения конференции,
- идентификатор издательства,
- год издания,
- тематика,
- идентификаторы выпусков,
- импакт фактор журнала.

Пример описания запроса на получение списка авторов представлен ниже.

Метод: GET

URL: api/v1/conferences

Заголовок запроса: x-app-token: 9f3a3401-e5f2-4a5b-bc8f-ea6a9566a3d7

Ответ:

200 OK

```
{
  "items": [
    {
      "id": "string",
      "ISSN": "string",
      "title": "string",
      "publisher_id": "string",
      "address": "string",
      "year": "string",
      "topic": "string",
      "issues_ids": ["string"],
      "impact_factor": "string"
    }
  ]
}
```

Bad Request

```
{ "errorMessage": "string" }
```

Выпуск. Запрашивается у системы управления содержанием целевой платформы при помощи соответствующего запроса. В случае успешного ответа, рекомендательный модуль получает следующие данные:

- идентификатор выпуска,
- год издания,

- выпуск,
- количество страниц,
- идентификатор источника,
- идентификаторы публикаций.

Пример описания запроса на получение списка авторов представлен ниже.

Метод: GET

URL: api/v1/issue

Заголовок запроса: x-app-token: 9f3a3401-e5f2-4a5b-bc8f-ea6a9566a3d7

Тело запроса:

```
{
  "id": "string",
  "issue_id": "string",
}
```

Ответ:

200 OK

```
{
  "id": "string",
  "year": "string",
  "val": "string",
  "pages": "string",
  "source_id": "string",
  "publications_ids": ["string"]
}
```

Bad Request

```
{ "errorMessage": "string" }
```

Издательство. Запрашивается у системы управления содержанием целевой платформы при помощи соответствующего запроса. В случае успешного ответа, рекомендательный модуль получает следующие данные:

- идентификатор издательства,
- наименование издательства,
- адрес издательства,
- представители издательства.

Пример описания запроса на получение списка авторов представлен ниже.

Метод: GET

URL: api/v1/publishing_hous

Заголовок запроса: x-app-token: 9f3a3401-e5f2-4a5b-bc8f-ea6a9566a3d7

Тело запроса:

```
{
  "publishing_hous_id": "string",
}
```

Ответ:

200 OK

```
{
  "id": "string",
  "title": "string",
  "address": "string",
  "representatives_ids": ["string"]
}
```

Bad Request

```
{ "errorMessage": "string" }
```

Объекты интереса. В случае успешного ответа, получаются следующие атрибуты:

- дублирование номера страницы,
- дублирование количества элементов страницы,
- общее количество элементов в списке объектов интереса,
- список конференций,

- список журналов,
- список авторов,
- список публикаций.

Пример описания запроса на получение списка объектов интереса представлен ниже.

Метод: GET

URL: : api/activities

Заголовок запроса: x-app-token: 9f3a3401-e5f2-4a5b-bc8f-ea6a9566a3d7

Тело запроса:

```
{  
  "page": number,  
  "pageSize": number  
}
```

Ответ:

200 OK

```
{  
  "page": number,  
  "pageSize": number,  
  "total": number,  
  "conferences": [  
    {  
      "id": "string",  
    }  
  ],  
  "magazines": [  
    {  
      "id": "string",  
    }  
  ],  
}
```

```

"authors": [
  {
    "id": "string",
  },
],
"publications": [
  {
    "id": "string",
  },
],
}
Bad Request
{ "errorMessage": "string" }

```

3.2 Описание API подсистемы лингвистического определения подобия для осуществления межсистемного взаимодействия

В качестве данных, необходимых для построения модели, выступают взаимосвязи субъектов с объектами интереса. Общая схема цикла обратной связи представлен на рисунке 11.

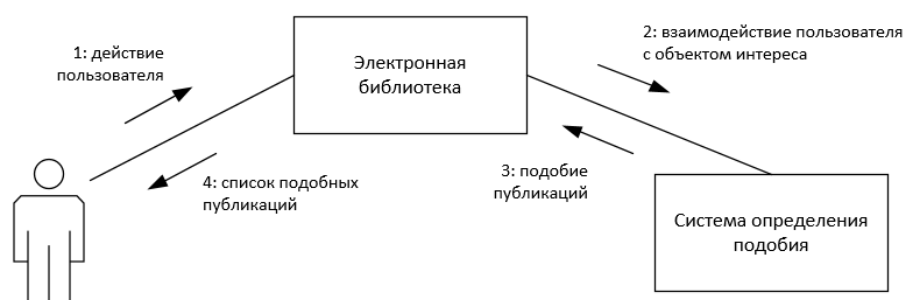


Рисунок 11 – UML диаграмма кооперации

Типы неявных взаимосвязей между пользователем и объектом интереса могут быть следующими:

- SEARCH (1): пользователь произвел поиск публикации,
- CLICK (2): пользователь перешел на страницу публикации,

- VIEWING (3): пользователь просмотрел полный текст публикации,
- COLLECTION (4): пользователь добавил публикацию в подборку.

Для заполнения данных связей между пользователем и публикацией реализован соответствующий POST запрос.

Метод: POST

URL: api/rating

Тело запроса:

```
[{
  "user_id": "308",
  "publication_id": "62139e4dff049b4d4536d21e",
  "rating": "CLICK"
},
{
  "user_id": "308",
  "publication_id": "62139e4dff049b4d4536d21e",
  "rating": "COLLECTION"
}]
```

Ответ:

200 OK

422 идентификатор пользователя совпадает с идентификатором статьи

Bad Request

```
{ "errorMessage": "string" }
```

Для получения списка рекомендаций публикаций на основе интересов пользователя реализован следующий запрос:

Метод: GET

URL: /api/search

Заголовок запроса: x-app-token: 9f3a3401-e5f2-4a5b-bc8f-ea6a9566a3d7

Тело запроса:

```
{  
  "user_id": "string",  
  "page": number,  
  "pageSize": number  
}
```

Ответ:

200 OK

```
{  
  "page": number,  
  "pageSize": number,  
  "total": number,  
  "items": [  
    {  
      "id": "string"  
    }  
  ]  
}
```

Bad Request

```
{ "errorMessage": "string" }
```

При успешной обработке данного запроса возвращается код 200 и список идентификаторов публикаций, соответствующих поиску, а в случае ошибки выдается код 400 и сообщение «Bad Request».

Листинг системы лингвистического определения подобия описания публикаций представлен в приложении А.

3.3 Тестирование системы лингвистического определения подобия описания публикаций

При создании рекомендательный модуля исходные тестовые данные базировались на существующей информации о публикации в системе eLIBRARY.RU. Получение обучающей выборки происходило посредством

запроса у сторонней системы управления содержанием. На основе этих данных изучались параметры объектов интереса, и соответственно, отлаживался алгоритм прогнозирования. Общее количество публикаций, использованных на данном этапе равно ста пятидесяти.

Далее использовалась контрольная выборка, состоящая из пятидесяти ранее не использованных в процессе разработки статей, для конечного получения результата работы алгоритма прогнозирования.

Для определения метрики точности работы разработанного алгоритма определения подобия были проведены вычисления по формуле 12.

$$precision_n = \frac{\sum_{i=1}^n (Rel_i * Dist_i)}{\sum_{i=1}^n Dist_i} * 100, \quad (12)$$

где Rel_i – ожидаемый коэффициент релевантности объекта интереса;

$Dist_i$ – спрогнозированный коэффициент подобия публикаций.

На основании проведенных вычислений, было выявлено, что метрика точности разработанного механизма определения подобия примерно равна 87,53 процентам. Исходя из этого, можно сделать вывод, что данный метод удовлетворяет поставленным в рамках данной работы задачам.

Пример списка публикаций контрольной выборки представлен ниже:

```
{
  "page": 1
  "pageSize": 2
  "total": 220,
  "items": [
    {
      "id": "46260813",
      "title": "FORMALIZED DESCRIPTION OF CYBER-PHYSICAL
      SYSTEMS MODELS USING TEMPORARY NON-
      DETERMINISTIC AUTOMATA",
```

```

    "authors_ids": [{
      "id": " 6282028f6fb500db9bed6bf5",
      "name": "TROKOZ D A",
    }, {
      "id": "618sa04944880caf5c64k711",
      "name": "ZABRODINA K A",
    }, {
      "id": "618b7604944880caf5c04a31",
      "name": "SAFRONOVA V S",
    }, {
      "id": "628600e2f4b073bcac9e2a22",
      "name": "VESELOVA M D",
    }, {
      "id": "6285f746f4b073187c9e2768",
      "name": "SINEV M P",
    }, {
      "id": "62822cf36fb500216eed742d",
      "name": "MITROHIN M A",
    }, {
      "id": "625e95825d6a64390099c7c2",
      "name": "GURIN E I",
    }, {
      "id": "6273bff9bc59a48c5ddf8f15",
      "name": "PASHCHENKO T Y",
    }
  ],
  "type": 4,

```

"annotation": " The principles of building finite state automata taking into account temporal logic were considered. In this case, time delays of the input

symbol are added to the elements of the finite state automaton, which allow the automaton to change its state without producing an output symbol, as well as delays of the output symbol that determine the necessary time interval for the corresponding transition. The main approaches to the definition of the concept of a process, such as “process is a state” and “process is a transition”, which are characterized by the principles of signal processing, taking into account the possibility of saving the state and forming a queue for signal processing, are distinguished. The scientific novelty of this article lies in the approach to the formation of a hierarchical cyber system and the use of temporary SNDA. This method allows to synchronize processes in multithreaded data processing in complex organized, multi-level systems and excludes the possibility of unstable operation, and also improves the performance of the entire complex. As a visual representation of the model of the created principle of temporal non-deterministic automata, a system of grinding and transferring grain in a mill was used.",

```
"keywords": [],
"year": "2021"
"magazines": null,
"conference": {
  "id": "61f8ee1a087fa3f630934d32",
  "title": "II INTERNATIONAL SCIENTIFIC CONFERENCE ON
METROLOGICAL SUPPORT OF INNOVATIVE TECHNOLOGIES
(ICMSIT II-2021)",
  "address": " St.Petersburg",
  "event_end": "2021-05-06T21:00:00.000Z", "event_start": "2021-05-
03T21:00:00.000Z",
},
"pages": "22068",
"publications": [{
  "id": null,
```

"title": "Minzilia Sagadeeva 2018 On Mathematical Model of Optimal Dynamical Measurement in Presence of Multiplicative and Additive Effects International Russian Automation Conference (RusAutoCon) 1-4.",

"context": "...Introduction When designing complex, multilevel systems, as a rule, there is a need to compile a mathematical model describing the operation of the whole complex, based on finite automata [1]...",

},{

"id": "3shc84tt682",

"title": "Liguo D, Yunpeng C, Chen W and Yan J 2009 Fuzzy Temporal Logic on Fuzzy Temporal Constraint Networks Sixth International Conference on Fuzzy Systems and Knowledge Discovery.",

"context": "...This approach allows to consider in more detail the mechanism of the system operation, namely the state of all objects and their interaction at a particular point in time, as a result of which you can get a formal description of the product being developed [2]..."

},{

"id": "6fskht603081",

"title": "Sotnikov A P, Shabaldina N V and Gromov M L 2017 Experiments on Parallel Composition of Timed Finite State Machines the Institute for System Programming of the RAS 29(3) 233-46.",

"context": "...However, the resulting mathematical model does not always provide for the time component, which is necessary for synchronizing processes in multithreaded data processing [3]..."

},{

"id": "4fs4tt608072",

```

    "title": "Ben M 2007 Using Temporal Logic to Analyse Temporal
    Logic: A Hierarchical Approach Based on Intervals Journal of Logic
    and Computation 17(2) 333-409.",
    "context": "...The solution to this problem is the use of temporal finite
    state automata based on temporal logic, which, in turn, combines the
    concepts of classical logic and temporal aspects [4]..."
  }, {
    "id": "42fdtt608574",
    "title": "Toshimitsu U and Shigemasa T 2015 Nonblocking Supervisory
    Control of Discrete Event Systems Modeled by Mealy Automata with
    Nondeterministic Output Functions IEEE Transactions on Automatic
    Control 61(3) 799-804.",
    "context": "...Within the framework of the presented article, two types
    of finite automata will be considered: deterministic and non-
    deterministic [5]..."
  }],
  "bibliometric_indicators": "",
  "link_publication": "/item.asp?id=46260813"
}, {
  "id": "41653709",
  "title": "МЕТОДЫ АВТОМАТИЗИРОВАННОГО
  ТЕСТИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ КОНЕЧНЫХ
  АВТОМАТОВ",
  "authors_ids": [{
    "id": "6282028f6fb500db9bed6bf5",
    "name": "ТРОКОЗ ДМИТРИЙ АНАТОЛЬЕВИЧ",
  }], {
    "id": "618sa04944880caf5c64k711",
    "name": "ЗАБРОДИНА КСЕНИЯ АНДРЕЕВНА",
  }

```

```

    },{
      "id": "618b7604944880caf5c04a31",
      "name": "САФРОНОВА ВАЛЕРИЯ СЕРГЕЕВНА",
    },{
      "id": "628600e2f4b073bcac9e2a22",
      "name": "ВЕСЕЛОВА МАРИНА ДМИТРИЕВНА",
    },{
      "id": "6285f746f4b07318awe42762",
      "name": "ХОЛУДЕНЕВА АЛИНА ОЛЕГОВНА",
    },
  ],
  "type": 4,

```

"annotation": "Данная статья посвящена описанию методологии автоматизированного тестирования, способной охватить все уровни автоматизированного тестирования распределенных информационных систем со сложной архитектурной организацией, при помощи теории автоматов, которые необходимы для наглядного представления данного механизма. Интерпретирование последовательности выполнения тестового сценария при помощи конечного автомата позволяет облегчить понимание процесса выполнения автоматизированного тестирования. Такой вид тестирования, в отличие от ручного, включает в себя различные уровни, которые обеспечивают контроль качества продукта на всём его жизненном цикле. Существуют различные множество методологий автоматизированного тестирования, которые разработаны для реализации данных уровней. Из этого множества можно выделить две основные группы: методика на основе ключевых слов и методика, основанная на использовании моделей в явном виде. Однако, каждая из них имеет свои недостатки, сужающие круг области применения. Предложенная в данной статье новая методология является универсальной, сочетающая в себе высокоуровневую абстракцию, которая облегчает процесс создания

тестового набора, и углубленную проверку спецификаций системы при помощи механизма тестирования, позволяющего повысить качество разрабатываемого продукта путем полного покрытия тестовых сценариев. Спроектированная в данной статье методология актуальна для систем со сложной архитектурной организацией.",

"key_words": ["ТЕСТИРОВАНИЕ", "АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ", "МЕТОДОЛОГИИ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ", "КОНЕЧНЫЙ АВТОМАТ"],

"year": "2019"

"magazines": {

"id": "758ee1a0s87fa3",

"ISSN": "2221-951X",

"title": "XXI ВЕК: ИТОГИ ПРОШЛОГО И ПРОБЛЕМЫ НАСТОЯЩЕГО ПЛЮС",

},

"conference": null,

"pages": "14-19",

"publication_ids": [{

"id": null,

"title": "Гребенюк В.М. Оценка целесообразности внедрения автоматизированного тестирования //Интернет журнал Науковедение. - 2013. - №1 (14).",

"context": "...кластеризацией программных модулей на структурные взаимосвязанные компоненты, количество критических ошибок возрастает с геометрической прогрессией, что приводит к ухудшению качества разрабатываемой системы. [1][2]..."

},{

"id": "273a98cfa4682",

"title": "Кузнецов А. А., Ковалёв С. П. Тестирование и мониторинг в распределенных автоматизированных системах технологического управления //ЖВТ. - 2009. - №4. EDN: LACTDJ",

"context": "...кластеризацией программных модулей на структурные взаимосвязанные компоненты, количество критических ошибок возрастает с геометрической прогрессией, что приводит к ухудшению качества разрабатываемой системы. [1][2]..."

},{

"id": "627396400f4eb",

"title": "Галимова Е.Ю., Коваленко А.Н. Выбор способа тестирования как решение многокритериальной задачи //ИВД. - 2016. - №3 (42). EDN: YGSSXP",

"context": "...Виды тестирования Тестирование можно классифицировать на две основные группы: ручное и автоматизированное [3]..."

},{

"id": "2355b8b2abr6",

"title": "Raluca Lefticaru, Florentin Ipate. (2007).\"State based Testing is Functional Testing\". Academic and Industrial Conference Practice and Research Techniques.",

"context": "...Они позволяют покрыть большой объем тест-кейсов, что значительно повышает производительность процесса отладки, а также облегчают работу и экономят время специалистов по тестированию, так как берут на себя проверку статического функционала [4]..."

},{

"id": "62738140967a3",

"title": "Butler K.M. (1996). "Integrating automated diagnosis into the testing and failure analysis operations". Proceedings of Proceedings International Test Conference 1996. Test and Design Validity Conference, Washington,DC,USA.",

"context": "...Автоматизация тестирования повышает надежность процесса отладки в целом, поскольку осуществляется проверка всех деталей, описанных в тест-кейсах [5]..."

}},

"bibliometric_indicators": "",

"link_publication": "/item.asp? id=41653709"

}

]

}

Пример списка взаимосвязей пользователя с объектами интереса представлен ниже:

{

"page": 1,

"pageSize": 30,

"total": 220,

"conferences": [{

"id": "61f8ee1a087fa3f630934d32"

},{

"id": "6r671a0dada3fyu59ex48b75"

}},

"magazines": [{

"id": "758ee1a0s87fa373185300e3"

}},

"authors": [{

"id": "6282028f6fb500db9bed6bf5",

```

    },{
      "id": "6285f746f4b07318awe42762",
    },
    "publications": [{
      "id": "we424b07318a4626kk0gh03",
    },{
      "id": "hwe4e416537094b07k318a5",
    },
  ],
}

```

Контентная модель строится на основе данных о взаимодействии пользователя с системой, полученных из CMS. После чего для каждого объекта интереса строится TF-IDF дистанция и расстояние Левенштейна относительно другого для оценки схожести по названию, описанию и ключевым словам. Получение конечных релевантных подборок происходит с учетом применения коэффициента новизны. Пример результата работы лингвистической системы определения подобия описания публикаций представлено на рисунке 12.

Data Output

	id [PK] character varying (255)	_id [PK] character varying (255)	dist double precision
1	5e55a346b333bd0019d18d2c	5e5615e9826e00001944cfd6	0.751318030803585
2	5e55a346b333bd0019d18d2c	5e5615f8826e00001944cfdc	0.63815829147706
3	5e55a346b333bd0019d18d2c	5e561606826e00001944cfe0	0.6246623304795325
4	5e55a346b333bd0019d18d2c	5e561615826e00001944cfe2	0.7208253837752725
5	5e55a346b333bd0019d18d2c	5e56161f826e00001944cfe4	0.7333326419705033
6	5e55a346b333bd0019d18d2c	5e561627826e00001944cfe6	0.5523124653867582
7	5e55a346b333bd0019d18d2c	5e56162f826e00001944cfe8	0.8297828501393059

Рисунок 12 – Результат работы системы

В итоговую подборку рекомендаций, пример которой представлен ниже, для n объектов интереса (id), будут выбраны все взаимосвязи относительно другого объекта ($_id$) с минимальной дистанцией ($dist$).

Тело запроса:

```

{

```

```
"user_id": "1632226467299",  
"page": 1,  
"pageSize": 10  
}
```

Ответ:

200 OK

```
{  
  "page": 1,  
  "pageSize": 10,  
  "total": 220,  
  "items": [  
    {  
      "id": "629325740900fd00017c4d6e"  
    },  
    {  
      "id": "6257d12a45cc11f9e82697fe"  
    },  
    {  
      "id": "62932d000900fd00017c4d74"  
    },  
    {  
      "id": "61bb646814277dec6448c187"  
    },  
    {  
      "id": "62932d170900fd00017c4d75"  
    },  
    {  
      "id": "61f955492d49e6f7fa67f3fe"  
    },  
  ],  
}
```

```

    {
      "id": "617945728a667bcb6e2b9e4d"
    },
    {
      "id": "621f15d48d4cc56aca3b66be"
    },
    {
      "id": "61f93f4f2d49e6f7fa670477"
    },
    {
      "id": "611a6c6e57f8c10019f8c8a4"
    }
  ]
}

```

В результате выполнения алгоритма определения подобия была предложена подборка релевантных объектов, состоящая из десяти публикаций, включающая в себя совершенно новые не просмотренные ранее текущим пользователем статьи.

На основании полученных результатов и метрики точности можно сказать о работоспособности данного алгоритма, а также его соответствии поставленным ранее задачам.

3.4 Выводы по разделу

В результате проектирования системы лингвистического определения подобия описания публикаций были описаны запросы получения входных данных, необходимых для работы механизма прогнозирования, а также запрос на формирование релевантных подборок. Также были проверены результаты работы программы, и на основании данной проверки сделаны выводы о работоспособности разработанного алгоритма рекомендаций.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы было проведено исследование современных подходов к реализации систем лингвистического определения подобия публикаций, в результате которого были выявлены особенности каждого из механизмов, а также проанализирована проблема холодного старта. В результате чего был разработан алгоритм, позволяющий осуществить подбор релевантных научных трудов на основании их описания, с учетом дополнительной лингвистической проверки и даты публикации.

Исследование подходов к формированию рекомендаций, а также основные выводы по ним были сформированы в рамках хоздоговорной научно-исследовательской работы «Исследование способов построения рекомендательной системы туристических маршрутов».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Jannach D. et al. Recommender Systems: An Introduction. – Cambridge University Press, 2010, p. 360.
2. Николенко С. Рекомендательные системы //СПб: Изд-во Центр Речевых Технологий. – 2012, с. 53.
3. Chen Y., Zhao X., de Rijke M. Top-N recommendation with high-dimensional side information via locality preserving projection //Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. – 2017. – С. 985-988.
4. Adomavicius G., Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions //IEEE transactions on knowledge and data engineering. – 2005. – Т. 17. – №. 6. – С. 734-749.
5. Королева Д. Е., Филиппов М. В. Анализ алгоритмов обучения коллаборативных рекомендательных систем //Инженерный журнал: наука и инновации. – 2013. – №. 6 (18). – С. 23.
6. Bobadilla J. et al. Recommender systems survey //Knowledge-based systems. – 2013. – Т. 46. – С. 109-132.
7. Fedorov A. V. Introduction into General Theory of Translation (linguistic problems) //St Petersburg, Moscow: Filologia Tri. – 2002.
8. Левенштейн В. И. Двоичные коды с исправлением выпадений и вставок символа 1 //Проблемы передачи информации. – 1965. – Т. 1. – №. 1. – С. 12-25.
9. Ricci F., Rokach L., Shapira B. Introduction to recommender systems handbook //Recommender systems handbook. – Springer, Boston, MA, 2011. – С. 1-35.
10. Моргунов Е. П., Рогова Е. В., Лузанова П. В. PostgreSQL. Основы языка SQL //учеб. пособие/ЕП Моргунов. – 2018.

11. Горнаков С. Осваиваем популярные системы управления сайтом (CMS). – Litres, 2022, с. 336.

ПРИЛОЖЕНИЕ А

Листинги программы

Файл `content_reco.py`

```
import time
from concurrent.futures.thread import ThreadPoolExecutor

import numpy as np
import pandas as pd

import requests
from fuzzywuzzy import fuzz
from loguru import logger
from sqlalchemy import create_engine

from app.core.config import DATABASE_URL, TF_IDF_RECEIVING_URL
from app.utils.sql_utils import pandas_df_to_sql_table

REQUEST_THREAD_NUMBER = 20

def _build_relation_df(df_items: pd.DataFrame) -> pd.DataFrame:
    offset = 1
    frames = []
    n = df_items.shape[0]
    for idx, row in df_items.iterrows():
        if offset >= n:
            break
        sub_df = df_items[offset: n][['id']]
        _n = sub_df.shape[0]
        sub_df = sub_df.assign(_id=pd.Series(np.full(_n, row['id'])).values)
        frames.append(sub_df)
        offset += 1
    return pd.concat(frames)
```

```

def build_content_recommendation():
    logger.info("Start build content model...")
    start_time = time.time()
    engine = create_engine(DATABASE_URL.url)
    with engine.connect() as con:
        df_items = pd.read_sql_table("cms_entity", con, columns=['id', 'text_ru', '_version'])
    if df_items.empty:
        logger.info("Empty table cms_entity. Return")
        return
    df_relation = _build_relation_df(df_items)

    def _get_text(_id: str) -> str:
        r = df_items[df_items['id'] == _id].iloc[0]
        return r['text_ru']

    def _get_v(_id: str) -> int:
        r = df_items[df_items['id'] == _id].iloc[0]
        return r['_version']

    def _calculate_dist(row: pd.Series) -> dict:
        idx, data = row
        logger.debug("Calculate distance for {0} text", idx)
        str1 = _get_text(data['id'])
        str2 = _get_text(data['_id'])
        r = requests.post(TF_IDF_RECEIVING_URL, json={'FirstText': str1, 'SecondText': str2})
        lev = fuzz.ratio(str1, str2)
        json = r.json()
        return {
            'id': data['id'],
            '_id': data['_id'],
            'dist': json['Distance'],
            'lev_dist': lev,
            'root_dist': json['RootDistance'],

```

```

        '_version': _get_v(data['id'])
    }

```

```

with ThreadPoolExecutor(REQUEST_THREAD_NUMBER) as executor:
    items_dist = []
    for item in executor.map(_calculate_dist, df_relation.iterrows()):
        items_dist.append(item)
    logger.info("Write content model to db...")
    pandas_df_to_sql_table(pd.DataFrame(items_dist), DATABASE_URL.url,
'df_distance_items')
    logger.info("Finish build content model in {0} seconds".format(time.time() - start_time))

```

Файл **relation.py**

```

from enum import Enum

```

```

class RelationType(str, Enum):
    search = 'SEARCH'
    click = 'CLICK'
    viewing = 'VIEWING'
    collection = 'COLLECTION'

    @property
    def val_as_int(self):
        if self.value == self.search:
            return 1
        elif self.value == self.click:
            return 2
        elif self.value == self.viewing:
            return 3
        elif self.value == self.collection:
            return 4
        else:
            raise TypeError

```

Файл users.py

```
from typing import Optional
```

```
from app.models.common import DateTimeModelMixin, IDModelMixin
```

```
from app.models.domain.rwmodel import RWModel
```

```
from app.services import security
```

```
class User(RWModel):
```

```
    username: str
```

```
    email: str
```

```
    bio: str = ""
```

```
    image: Optional[str] = None
```

```
class UserInDB(IDModelMixin, DateTimeModelMixin, User):
```

```
    salt: str = ""
```

```
    hashed_password: str = ""
```

```
    def check_password(self, password: str) -> bool:
```

```
        return security.verify_password(self.salt + password, self.hashed_password)
```

```
    def change_password(self, password: str) -> None:
```

```
        self.salt = security.generate_salt()
```

```
        self.hashed_password = security.get_password_hash(self.salt + password)
```

Файл entity_type.py

```
from enum import Enum
```

```
class EntityType(str, Enum):
```

```
    publication = 'publication'
```

```

@property
def val_as_int(self):
    if self.value == self. publication:
        return 0
    else:
        raise TypeError

```

Файл cms.py

```
import datetime
```

```
from pydantic import BaseConfig, BaseModel
```

```
def convert_datetime_to_realworld(dt: datetime.datetime) -> str:
    return dt.replace(tzinfo=datetime.timezone.utc).isoformat().replace("+00:00", "Z")

```

```
def convert_field_to_camel_case(string: str) -> str:
    return "".join(
        word if index == 0 else word.capitalize()
        for index, word in enumerate(string.split("_"))
    )

```

```
class RWModel(BaseModel):
    class Config(BaseConfig):
        allow_population_by_field_name = True
        json_encoders = {datetime.datetime: convert_datetime_to_realworld}
        alias_generator = convert_field_to_camel_case

```

Файл security.py

```
import bcrypt
from passlib.context import CryptContext
```

```
pwd_context = CryptContext(schemes=["bcrypt"], deprecated="auto")
```

```
def generate_salt() -> str:  
    return bcrypt.gensalt().decode()
```

```
def verify_password(plain_password: str, hashed_password: str) -> bool:  
    return pwd_context.verify(plain_password, hashed_password)
```

```
def get_password_hash(password: str) -> str:  
    return pwd_context.hash(password)
```

Файл authentication.py

```
from app.db.errors import EntityDoesNotExist  
from app.db.repositories.users import UsersRepository
```

```
async def check_username_is_taken(repo: UsersRepository, username: str) -> bool:  
    try:  
        await repo.get_user_by_username(username=username)  
    except EntityDoesNotExist:  
        return False  
  
    return True
```

```
async def check_email_is_taken(repo: UsersRepository, email: str) -> bool:  
    try:  
        await repo.get_user_by_email(email=email)  
    except EntityDoesNotExist:  
        return False
```

```
return True
```

Файл similarity_publications.py

```
from collections import ChainMap
```

```
from typing import List
```

```
from app.core.config import CMS_HEADERS, CMS_DETAILS_URL
```

```
from app.models.common import Pageable
```

```
async def get_cms_entities(ids: List[str], language: str, paging: Pageable):
```

```
    resp = {'items': []}
```

```
    isOk = True
```

```
    if len(ids) > 0:
```

```
        async with AsyncClient() as client:
```

```
            r = await client.get(CMS_DETAILS_URL, params={'language': language, 'ids[]': ids},  
headers=CMS_HEADERS)
```

```
            resp = r.json()
```

```
            isOk = r.status_code == 200
```

```
    if isOk:
```

```
        resp.update({
```

```
            'page': paging.page,
```

```
            'pageSize': paging.size,
```

```
            'total': len(ids)
```

```
        })
```

```
    return resp
```

Файл sql_utils.py

```
from typing import List
```

```
import pandas as pd
```

```
from sqlalchemy import create_engine
```

```

def pandas_df_to_sql_table(df: pd.DataFrame, db_url: str, table: str) -> None:
    engine = create_engine(db_url)
    with engine.connect() as con:
        q = "TRUNCATE {0}".format(table)
        con.execute(q)
    df.to_sql(table, engine, index=False, if_exists='append', method='multi')

```

```

def get_cms_languages(db_url: str) -> List[str]:
    engine = create_engine(db_url)
    with engine.connect() as con:
        q = "SELECT code FROM cms_lang"
        res = con.execute(q)
        return [r[0] for r in res.fetchall()]

```

ПРИЛОЖЕНИЕ В

Презентация

СИСТЕМА ЛИНГВИСТИЧЕСКОГО ОПРЕДЕЛЕНИЯ ПОДОБИЯ ОПИСАНИЯ ПУБЛИКАЦИЙ



Выполнил студент группы 20BВм1

Забродина Ксения Андреевна

Научный руководитель

Токарев Андрей Николаевич

Пенза, 2022 г.

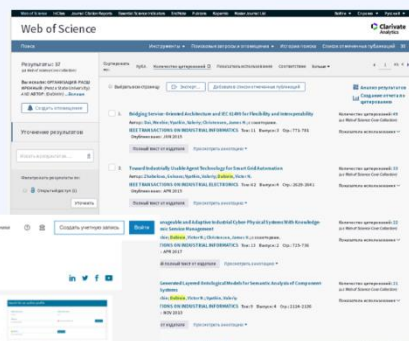
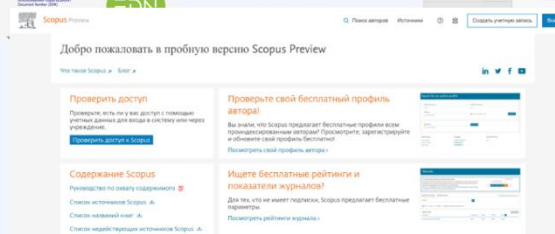
АНАЛОГИ ТУРИСТИЧЕСКИХ ПЛАТФОРМ

eLIBRARY.RU

Web of Science



Scopus



КЛАССИФИКАЦИЯ МЕХАНИЗМОВ ПРОГНОЗИРОВАНИЯ



КОНТЕНТНАЯ ФИЛЬТРАЦИЯ

Мера частоты употребления слова в документе (TF-IDF):

$$TF_{j,i} = \frac{f_{j,i}}{\max_z f_{z,i}},$$

где $\max_z f_{z,i}$ – количество упоминаний всех ключевых слов k_z в d_i .

Обратная частота IDF:

$$IDF_j = \log \log \frac{N}{n_i}.$$

Общая частота $w_{j,i}$:

$$w_{j,i} = TF_{j,i} \times IDF_j.$$

КОЛЛАБОРАТИВНАЯ ФИЛЬТРАЦИЯ

Коэффициент корреляции Пирсона двух пользователей рассчитывается как:

$$Pearson(\bar{A}, \bar{B}) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \times \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}.$$

Сходство нормализованных оценок двух объектов интереса определяется как:

$$Cosine(\bar{Z}, \bar{Y}) = \frac{\sum_{i=1}^n z_i \times y_i}{\sqrt{\sum_{i=1}^n z_i^2} \times \sqrt{\sum_{i=1}^n y_i^2}}.$$

ФИЛЬТРАЦИЯ, ОСНОВАННАЯ НА ЗНАНИЯХ

Принцип работы методов рекомендации, основанных на знаниях, заключается в прогнозировании востребованности того или иного объекта интереса у целевого пользователя на базе знаний в текущей предметной области, а именно на определении степени соответствия назначения элемента потребностям и предпочтениям пользователей.

В качестве входных значений данный способ использует множества объектов интереса и правил, описывающих свойства предпочтительных элементов. При этом правила могут быть как прописаны в самой системе, так и заданы пользователем. В последнем случае в рекомендательной системе должны быть предусмотрены ограничения на ввод критериев.

ПРОБЛЕМА ХОЛОДНОГО СТАРТА

Проблема холодного старта проявляется тогда, когда невозможно дать надежные прогнозы ввиду недостаточности или полного отсутствия входных данных о пользователе или объекте интереса.

Частными случаями являются:

- регистрация нового пользователя в систему, о котором на данный момент ничего не известно и который еще ничего просмотрел,
- добавление новых, еще никем не оцененных товаров или услуг.

Для решения проблемы холодного запуска необходимо корректно подобрать алгоритм прогнозирования, а также применить дополнительные параметры, позволяющие игнорировать разреженность данных.

АЛГОРИТМЫ ЛИНГВИСТИЧЕСКОГО ОПРЕДЕЛЕНИЯ ПОДОБИЯ ТЕКСТОВЫХ ДАННЫХ

В основном механизмы сравнения текстовых записей подразделяют на две основные группы:

- детерминированные,
- вероятностные.

Первая категория базируется на строгом сопоставлении всех атрибутов исходных объектов, при этом соответствие одного объекта с другим может варьироваться в зависимости от требований, поставленных системой, т.е. процентное соотношение совпадающих параметров может быть дифференцировано.

Вторая категория основывается на вычислении уровня подобию атрибутов текстовых объектов с последующим определением весового коэффициента для каждого из параметров. При этом достижение связности между ними зависит от порогового значения: чем выше вес относительно заданного, тем больше вероятность корреляции, в противном случае параметры считаются независимыми.

АЛГОРИТМЫ ЛИНГВИСТИЧЕСКОГО ОПРЕДЕЛЕНИЯ ПОДОБИЯ ТЕКСТОВЫХ ДАННЫХ

Алгоритм Джаро-Винклера основан на вычислении расстояния Джаро, т.е. меры сходства между двумя исходными строками.

Расстояние Джаро между двумя строками s_1 и s_2 рассчитывается следующим образом:

$$d_j = \begin{cases} 0 & \text{если } m = 0 \\ \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) & \text{в противном случае} \end{cases},$$

При этом символ из первой строки совпадает с соответствующим символом из второй строки только в случае, когда они являются идентичными и расположены не дальше, чем:

$$\frac{\max(|s_1|, |s_2|)}{2} - 1.$$

АЛГОРИТМЫ ЛИНГВИСТИЧЕСКОГО ОПРЕДЕЛЕНИЯ ПОДОБИЯ ТЕКСТОВЫХ ДАННЫХ

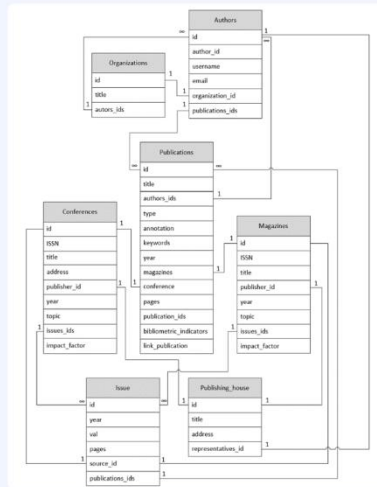
Второй метод базируется на расстоянии Левенштейна, которое определяет разницу между двумя строками путем вычисления минимального количества операций, требуемых для модификации одной последовательности символов в другую.

$$d(i, j) = \begin{cases} \max(i, j) & \text{если } \min(i, j) = 0 \\ \min \begin{pmatrix} d(i-1, j)+1, \\ d(i, j-1)+1, \\ d(i-1, j-1)+m(s_1[i], s_2[j]) \end{pmatrix} & \text{в противном случае} \end{cases},$$

где $d(i, j)$ – расстояние Левенштейна между строками s_1 и s_2 ;

$m(s_1[i], s_2[j])$ – функция, осуществляющая сравнение двух символов разных строк. В случае их равенства возвращается 0, в противном случае – 1.

ИСХОДНЫЕ ДАННЫЕ



КОРРЕКТИРУЮЩИЕ КОЭФФИЦИЕНТЫ

На заключительной стадии составления подборки рекомендаций (q) применяется коэффициент новизны для новых объектов интереса, имеющих низкую оценку релевантности ввиду отсутствия информации о них.

$$q(k) = y + l_k,$$

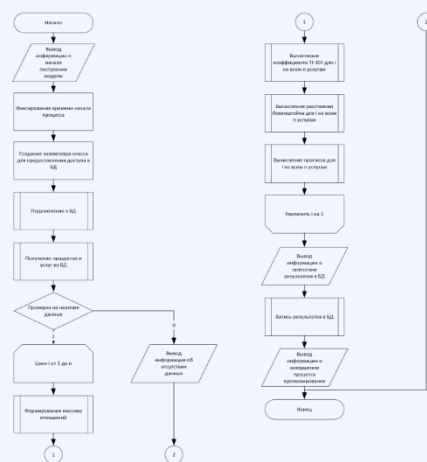
где k – текущая информация об объекте интереса относительно пользователя;
 y – список атрибутов вместе с условными вероятностями;
 l_k – текущее значение коэффициента новизны данного объекта интереса.

Нормализации даты публикации:

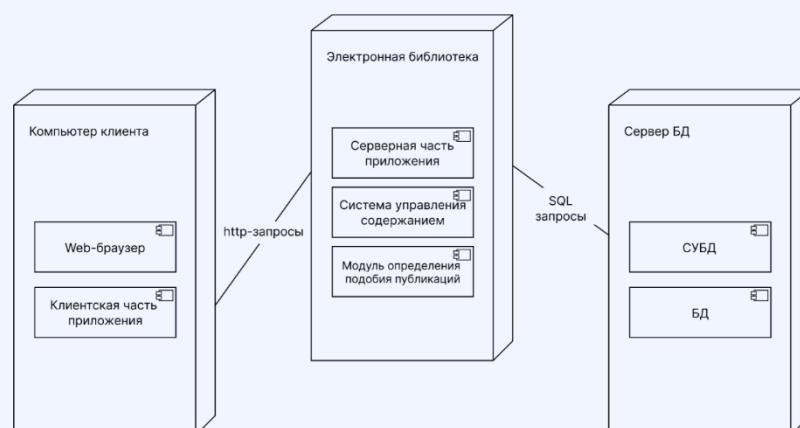
$$l_k = \frac{t_k - t_{min}}{t_{max} - t_{min}},$$

где t_k – дата публикации текущей статьи;
 t_{min} – дата публикации статьи, имеющей наиболее давнюю дату опубликования;
 t_{max} – дата публикации новой статьи, имеющей наиболее раннюю дату опубликования.

АЛГОРИТМ СОСТАВЛЕНИЯ РЕКОМЕНДАЦИЙ



UML ДИАГРАММА РАЗВЕРТЫВАНИЯ ЦЕЛЕВОЙ СИСТЕМЫ



API ПОЛУЧЕНИЯ ИСХОДНЫХ ДАННЫХ

Пример описания запроса на получение списка авторов:

Метод: GET

URL: api/v1/authors

Заголовок запроса: x-app-token: 9f3a3401-e5f2-4a5b-bc8f-ea6a9566a3d7

Ответ:

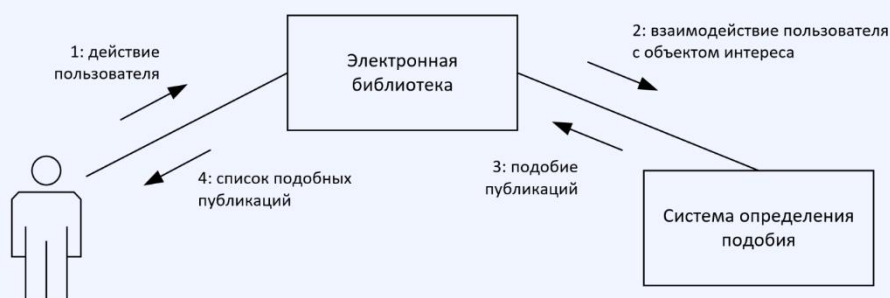
200 OK

```
{
  "items": [
    {
      "id": "string",
      "author_id": "string",
      "username": "string",
      "email": "string",
      "organization_id": "string",
      "publications_ids": ["string"]
    }
  ]
}
```

Bad Request

```
{ "errorMessage": "string" }
```

UML ДИАГРАММА КООПЕРАЦИИ



НЕЯВНЫЕ ВЗАИМОСВЯЗИ МЕЖДУ ПОЛЬЗОВАТЕЛЕМ И ОБЪЕКТАМИ ИНТЕРЕСА

Типы неявных взаимосвязей между пользователем и объектом интереса могут быть следующими:

- SEARCH (1): пользователь произвел поиск публикации,
- CLICK (2): пользователь перешел на страницу публикации,
- VIEWING (3): пользователь просмотрел полный текст публикации,
- COLLECTION (4): пользователь добавил публикацию в подборку.

Запрос для заполнения данных связей между пользователем и публикацией:

Метод: POST
URL: api/rating
Тело запроса:

```
[[{"user_id": "308", "publication_id": "62139e4dff049b4d4536d21e", "rating": "CLICK"}, {"user_id": "308", "publication_id": "62139e4dff049b4d4536d21e", "rating": "COLLECTION"}]]
```

Ответ:
200 OK

ОПИСАНИЕ API ПОДСИСТЕМЫ ЛИНГВИСТИЧЕСКОГО ОПРЕДЕЛЕНИЯ ПОДОБИЯ

Для получения списка рекомендаций публикаций на основе интересов пользователя реализован следующий запрос:

Метод: GET
URL: /api/search
Заголовок запроса: x-app-token: 9f3a3401-e5f2-4a5b-bc8f-ea6a9566a3d7
Тело запроса:

```
{ "user_id": "string", "page": number, "pageSize": number }
```

Ответ:
200 OK

```
{ "page": number, "pageSize": number, "total": number, "items": [ { "id": "string" } ] }
```

Bad Request
{ "errorMessage": "string" }

ОПИСАНИЕ АРІ ПОДСИСТЕМЫ ЛИНГВИСТИЧЕСКОГО ОПРЕДЕЛЕНИЯ ПОДОБИЯ

Метрика точности работы разработанного алгоритма :

$$precision_n = \frac{\sum_{i=1}^n (Rel_i * Dist_i)}{\sum_{i=1}^n Dist_i} * 100 ,$$

где Rel_i – ожидаемый коэффициент релевантности объекта интереса;
 $Dist_i$ – спрогнозированный коэффициент подобия публикаций.

Метрика точности разработанного механизма определения подобия примерно равна 87,53 процентам.

Результат работы системы:

Data Output			
	id [PK] character varying (255)	id [PK] character varying (255)	dist double precision
1	5e55a346b333bd0019d18d2c	5e5615e9826e00001944cfd6	0.751318030803585
2	5e55a346b333bd0019d18d2c	5e5615f8826e00001944cfd6	0.63815829147706
3	5e55a346b333bd0019d18d2c	5e561606826e00001944cfe0	0.6246623304795325
4	5e55a346b333bd0019d18d2c	5e561615826e00001944cfe2	0.7208253837752725
5	5e55a346b333bd0019d18d2c	5e56161f826e00001944cfe4	0.7333326419705033
6	5e55a346b333bd0019d18d2c	5e561627826e00001944cfe6	0.5523124653867582
7	5e55a346b333bd0019d18d2c	5e56162f826e00001944cfe8	0.8297828501393059

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы было проведено исследование современных подходов к реализации систем лингвистического определения подобия публикаций, в результате которого были выявлены особенности каждого из механизмов, а также проанализирована проблема холодного старта. В результате чего был разработан алгоритм, позволяющий осуществить подбор релевантных научных трудов на основании их описания, с учетом дополнительной лингвистической проверки и даты публикации.

Исследование подходов к формированию рекомендаций, а также основные выводы по ним были сформированы в рамках хозяйственной научно-исследовательской работы «Исследование способов построения рекомендательной системы туристических маршрутов».

СПАСИБО ЗА ВНИМАНИЕ!

