МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование кафедры)

09.03.03 Прикладная информатика
(код и наименование направления подготовки)

Бизнес-информатика
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка информационной системы для анализа поисковых запросов в социальных сетях»

Обучающийся А.Е. Баранов _____
(Инициалы Фамилия) (личная подпись)

Руководитель к.т.н, доцент, О.В. Аникина
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

Аннотация

Тема бакалаврской работы – «Разработка информационной системы для анализа поисковых запросов в социальных сетях».

Исследование поисковых запросов помогает выявить основные темы, которые интересуют конкретного пользователя. Методы анализа запросов применяются для рекомендаций товаров и контента в соцсетях. Благодаря этим методам можно лучше понять интересы и предпочтения пользователя.

Актуальность работы заключается в разработке системы для анализа поисковых запросов и визуализации полученных результатов.

Объектом исследования бакалаврской работы является технологии анализа текстовых данных (text mining).

Предметом исследования бакалаврской работы является система для анализа и визуализации текстовых запросов.

Цель данной выпускной квалификационной работы - создание информационной системы для анализа поисковых запросов в социальных сетях с целью выявления интересов пользователя.

Методы исследования – технологии текстовых данных (text mining), технологии проектирования информационных систем, технологии программирования.

Данная работа состоит из введения, трех глав, заключения и списка используемой литературы.

В первой главе исследования рассматриваются характеристики анализа текстовых данных, в следующей главе представлены подходы к анализу поисковых запросов и визуализации полученных результатов, а в третьей главе подробно описана практическая реализация системы для анализа поисковых запросов в социальных сетях.

Бакалаврская работа состоит из 41 страницы текста, 27 рисунков и 20 источников.

Оглавление

Введение

Технологии text mining растут из-за необходимости в анализе естественных языков. Они используются для чат-ботов, обработки текстов, определения темы и эмоций [1], [5]. В бакалаврской работе представлена система, анализирующая запросы пользователя в соцсетях с применением text mining [8].

Выбор темы обусловлен ростом популярности соцмедиа. По статистике Вконтакте, ее ежедневно посещает 54% россиян [4]. Чтобы удерживать интерес, важно понимать предпочтения пользователей.

Для определения интересов пользователя можно использовать как информацию, указанную им в своем профиле (например, место проживания, хобби и т.д.), так и статистические данные о страницах, которые он просматривал в социальных сетях (вроде тематических сообществ и групп).

В этом исследовании предлагается дополнить перечень источников данных о предпочтениях пользователя, включив в него информацию, извлеченную из анализа его поисковых запросов.

Одним из трудностей использования поисковых запросов для выявления интересов пользователя является их частое представление в текстовой форме без определенной структуры. Для получения нужной информации из такого текста требуется применение методов text mining [7].

Цель работы - создать систему анализа запросов для определения интересов пользователя.

Чтобы достичь поставленной цели, следует решить ряд задач:

- разработать модель процесса выявления интересов пользователя;
- спроектировать систему анализа поисковых запросов в социальных сетях;
- осуществить реализацию и провести тестирование созданной системы;

Используемые методы исследования включают в себя технологии анализа текстов (text mining), методы проектирования информационных систем и программные технологии.

Практическое значение этой бакалаврской работы связано с созданием программного решения для определения интересов пользователя, исходя из анализа его текстовых поисковых запросов. Работа включает в себя введение, три главы, заключение, список литературы и приложения.

В первой главе работы представлен обзор актуальных исследований в области text mining, рассматриваются вопросы удержания внимания пользователей социальных медиа, также проводится моделирование процесса выявления интересов пользователей с использованием нотации IDEF0.

Во второй главе акцент делается на дизайне системы анализа поисковых запросов в социальных сетях. Здесь представлена схема работы предлагаемого программного решения и детализирован алгоритм обработки текстовой информации.

Третья глава охватывает этапы создания программного продукта. В ней также демонстрируются исходы тестирования программного модуля на основе реальных поисковых запросов автора.

В заключении изложены итоги проведенной выпускной квалификационной работы.

В рамках бакалаврской работы будет разработано приложение на языке Python, обеспечивающее анализ текстовых поисковых запросов. Оно включает в себя функции: импорт и просмотр данных запросов, их первичную обработку, удаление стоп-слов, приведение слов к их основной форме, частотный анализ и визуализацию наиболее употребляемых слов в форме "облака тегов". Проведено тестирование программы на актуальных текстовых запросах.

Бакалаврская работа состоит из 40 страниц текста, 26 рисунков, и 20 источников.

Глава 1 Технологии анализа текстовых данных

1.1 Сферы использования технологий анализа текстовых данных

В последние годы области использования искусственного интеллекта стали гораздо шире. Научные статьи демонстрируют, что искусственный интеллект находит применение в робототехнике, анализе изображений, автоматическом переводе текстов, бизнес-аналитике, распознавании образов, текстовых задачах, извлечении информации, экспертных и безопасностных системах, а также при обработке речи и анализе текстов на естественном языке [9]. Все эти области применения представлены на рисунке 1.

Развитие областей применения искусственного интеллекта тесно связано с его способностью совершенствовать решение разнообразных практических задач, что, в свою очередь, приводит к повышению эффективности и производительности в различных областях.

Одно из активно развивающихся направлений сейчас - анализ поведения пользователей в социальных сетях. Анализируя активность пользователей, можно решить целый ряд задач:

- выявление и удаление фальшивых профилей;
- противодействие незаконным действиям в социальных сетях;
- продление активности пользователя на платформе социальной сети;
- реклама продуктов, соответствующих интересам пользователя;
- создание рекомендательных систем для контента на основе интересов пользователей;
- косвенный анализ реакции пользователей на разнообразные события;

Таким образом, искусственный интеллект может служить мощным инструментом для анализа и оптимизации поведения пользователей в социальных сетях.

Рисунок 1 – Области использования искусственного интеллекта

Компании, которые управляют социальными сетями, стремятся прежде всего максимизировать свою прибыль. Основной задачей является увлечение

пользователя так, чтобы он проводил на платформе максимальное количество времени [11]. Чтобы достичь этого, важно знать и учитывать интересы каждого пользователя, предлагая ему контент, который будет для него наиболее привлекательным.

Для определения интересов пользователя могут использоваться как личные данные из его профиля (например, город проживания или хобби) так и информация о его активности в социальных сетях, включая посещенные тематические группы и паблики.

В ходе данного исследования предлагается дополнить перечень источников данных о предпочтениях пользователя, включив в него информацию, полученную на основе его поисковых запросов.

Основная сложность использования поисковых запросов для определения интересов заключается в их текстовой форме, часто представленной в произвольном стиле. Для выделения ключевой информации из таких текстовых данных требуется применять методы text mining.

Анализ научных источников выявил, что процесс text mining осуществляется по алгоритму, изображенному на рисунке 2. Начинается все с сбора информации, которой могут служить текстовые файлы, веб-страницы и отзывы пользователей. Далее происходит парсинг собранного материала, выделяя текстовую информацию [16], [17]. На следующем этапе производится очистка текста от избыточных или неактуальных данных, используя определенные критерии, такие как стартовые и стоп-слова, а также списки для фильтрации. При обработке больших текстовых массивов данные преобразуются в векторные представления. Завершающим этапом является анализ полученных признаков для достижения цели исследования. В процессе анализа может потребоваться настройка методов извлечения данных [18], [19], [20].

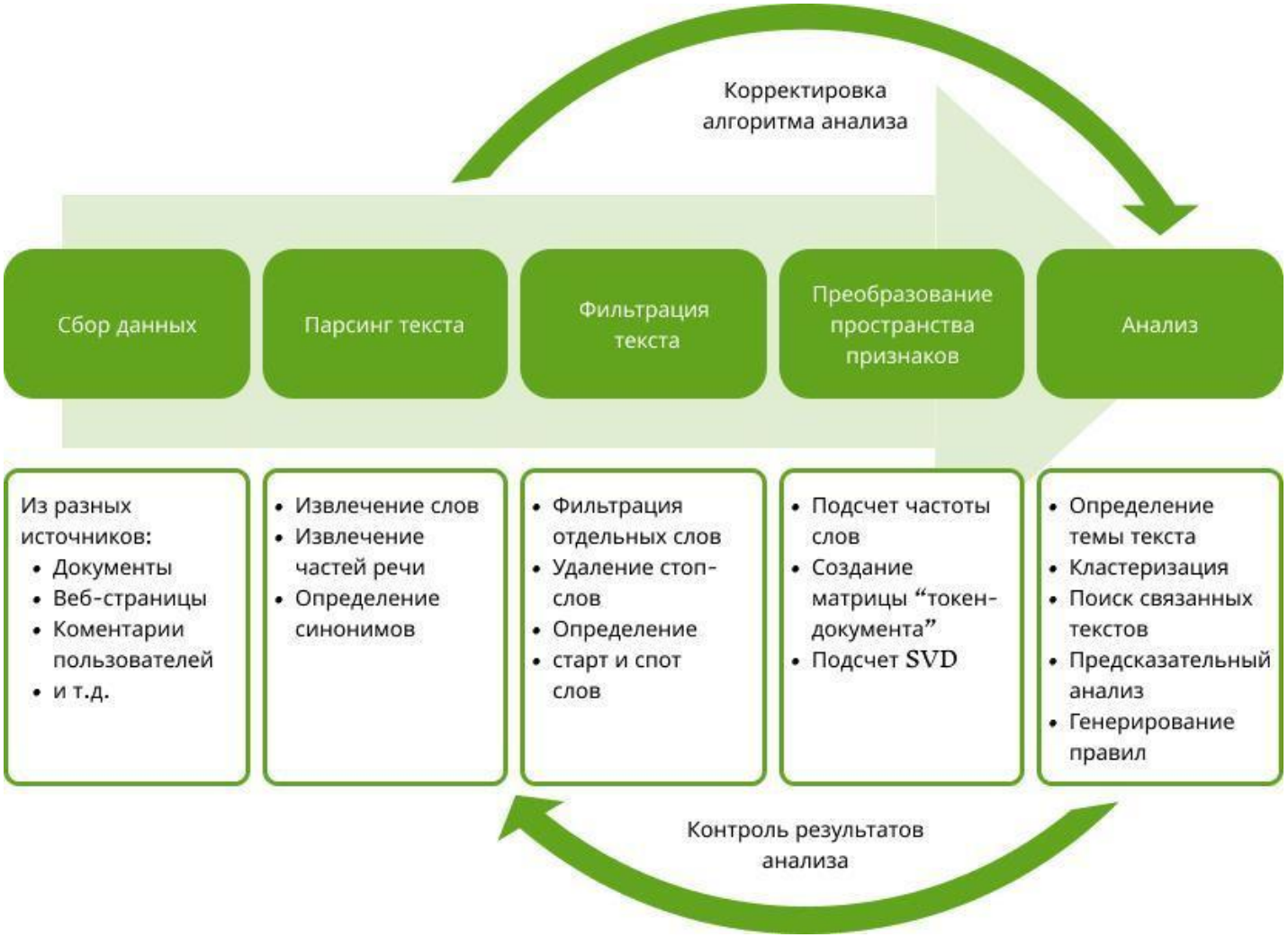

Рисунок 2 – Структура изучения текстовых материалов (text mining)

В научной литературе часто утверждается, что задачи, решаемые с помощью text mining, можно группировать на четыре основные категории: классификационные задачи, задачи кластерного анализа, регрессионные задачи и задачи аффинитивного анализа [10], [12], [13]. Примеры для каждой из этих категорий представлены на рисунке 3.

Рисунок 3 – Различные категории задач при обработке текстовой информации

В данном контексте мы будем применять text mining для выявления тем, которые интересуют пользователя, анализируя его поисковые запросы в социальных сетях. Закончив теоретическое обсуждение text mining, давайте переходить к моделированию процедуры определения интересов пользователя.

1.2 Моделирование метода выявления интересов пользователя

Для моделирования метода выявления интересов пользователя мы будем использовать программное решение ERwin Process Modeler [6], [14].

Входные данные для процесса определения интересов включают в себя:

- активность пользователя в социальной сети;
- персональные данные пользователя;

К активности пользователя можно отнести его действия в социальной сети, такие как: момент входа в профиль, используемое устройство для доступа, оставленные комментарии, взаимодействие с контентом (например, нажатие кнопки "Like") и поисковые запросы с указанием времени их выполнения. Под персональными данными понимается информация из профиля пользователя и его уникальный идентификатор.

Процесс выявления интересов пользователя находится под регулированием законодательства РФ и внутренними нормами социальной сети, указанными в пользовательском соглашении [15].

Для реализации этого процесса привлекаются квалифицированные сотрудники, разнообразные технические ресурсы, а также информационная система, разработанная в рамках данной бакалаврской работы.

По завершении процесса мы получаем перечень тем, которые интересуют пользователя, и графическое представление его интересов. Модель TO-BE данного процесса изображена на рисунке 4.

Рисунок 4 – Модель TO-BE, уровень A-0

Разберем подробнее каждый этап процесса "Определение интересов пользователя". Этот комплексный процесс включает в себя ряд важных шагов, начиная с сбора поисковых запросов. Далее следует этап подготовки данных к анализу, на котором осуществляется организация и структурирование данных для последующего исследования. Завершающим этапом является анализ данных, на котором проводится детальное изучение и выделение основных трендов и паттернов, позволяющих определить интересы конечного пользователя.

На первом этапе мы изучаем активность пользователя и составляем список его поисковых запросов. Затем этот список запросов подготавливается к анализу. В рамках подготовки, согласно методике text mining, текст преобразуется в ряд токенов (слова в их базовой форме). Данный этап реализуется с помощью разрабатываемой нами информационной системы. После этого список токенов анализируется, что позволяет:

– выделить список тем, которые интересуют пользователя;
– создать визуализации, отражающие интересы пользователя;

С учетом вышеизложенного, модель TO-BE процесса "Определение интересов пользователя" представлена на рисунке 5.

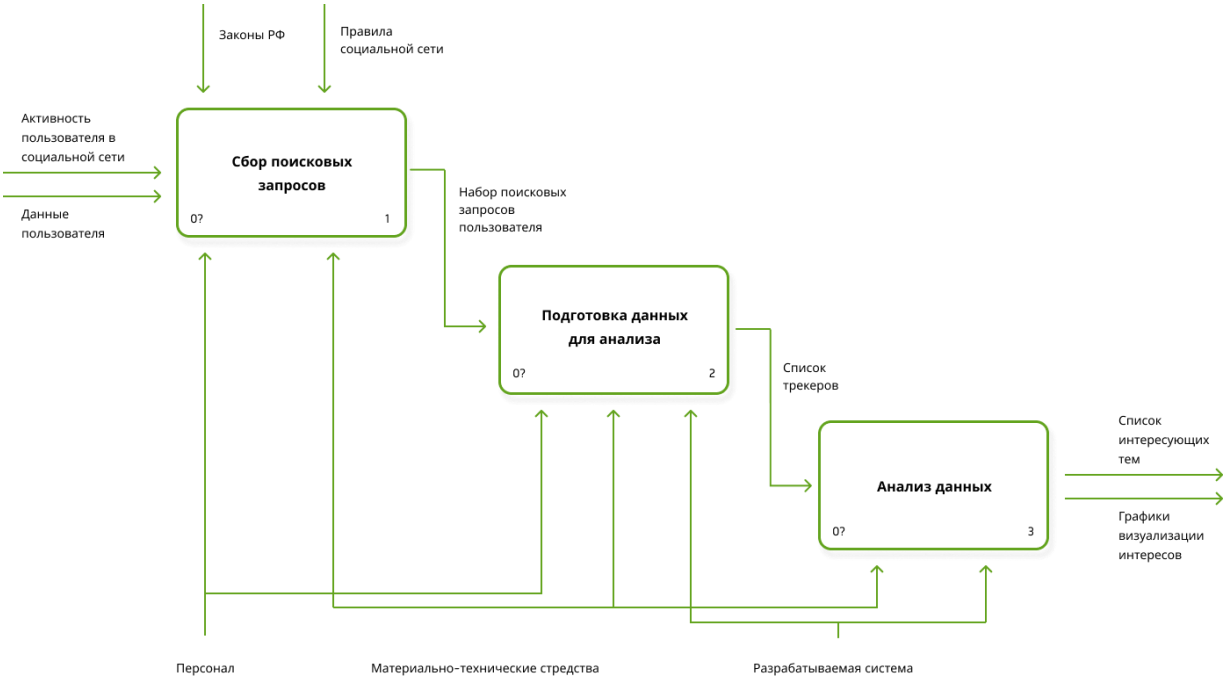

Рисунок 5 – Модель TO-BE, декомпозиция

Таким образом, проведено моделирование процесса «Определение интересов пользователя» и определена роль разрабатываемой системы в данном процессе.

Выводы по главе 1

Выводы по главе одной бакалаврской работы можно описать следующим образом:

- исследование литературных источников показало, что ключевой проблемой для компаний, управляющих социальными сетями, является задача удержания клиентов на своем ресурсе;
- чтобы заинтересовать пользователей, сервисам важно понимать их предпочтения и предоставлять соответствующий контент;
- в данной работе предлагается выявлять интересы клиентов на основе их текстовых запросов с использованием технологий text mining;
- основные этапы text mining включают: сбор информации, разбор текста, очистка текста, изменение структуры данных и их последующий анализ;
- с применением методологии IDEF0 было создано функциональное представление процесса "Определение интересов пользователя", где были выделены ключевые элементы процесса и роль планируемой информационной системы;

Глава 2 Разработка системы анализа поисковой активности в социальных сетях

2.1 Методика исследования поисковых запросов

При выполнении бакалаврской работы была разработана схема работы программного модуля (рисунок 6). Пользователь взаимодействует с социальной сетью, ища контент по актуальной для него теме. Все запросы пользователя сохраняются либо в текстовом файле, либо в базе данных. Для более детального изучения интересов пользователя поступившие запросы передаются в специально разработанный программный модуль.

Для выявления тем, которые привлекали внимание пользователя в определенный временной интервал, такой как, например, за последний месяц, в программный модуль отправляются только запросы, сделанные за этот временной интервал, а не все накопленные данные.

Программный модуль позволяет анализировать интересы не только индивидуального пользователя, но и целых групп, объединенных определенными характеристиками, такими как место проживания. Для этого на обработку направляются совокупные поисковые запросы данной группы пользователей.

Если у нас есть доступ к запросам пользователя из различных социальных сетей, их можно совместить в один список для более глубокого анализа предпочтений пользователя.

На основе этого агрегированного списка программный модуль, применяя частотный анализ, определяет ключевые интересы пользователя.

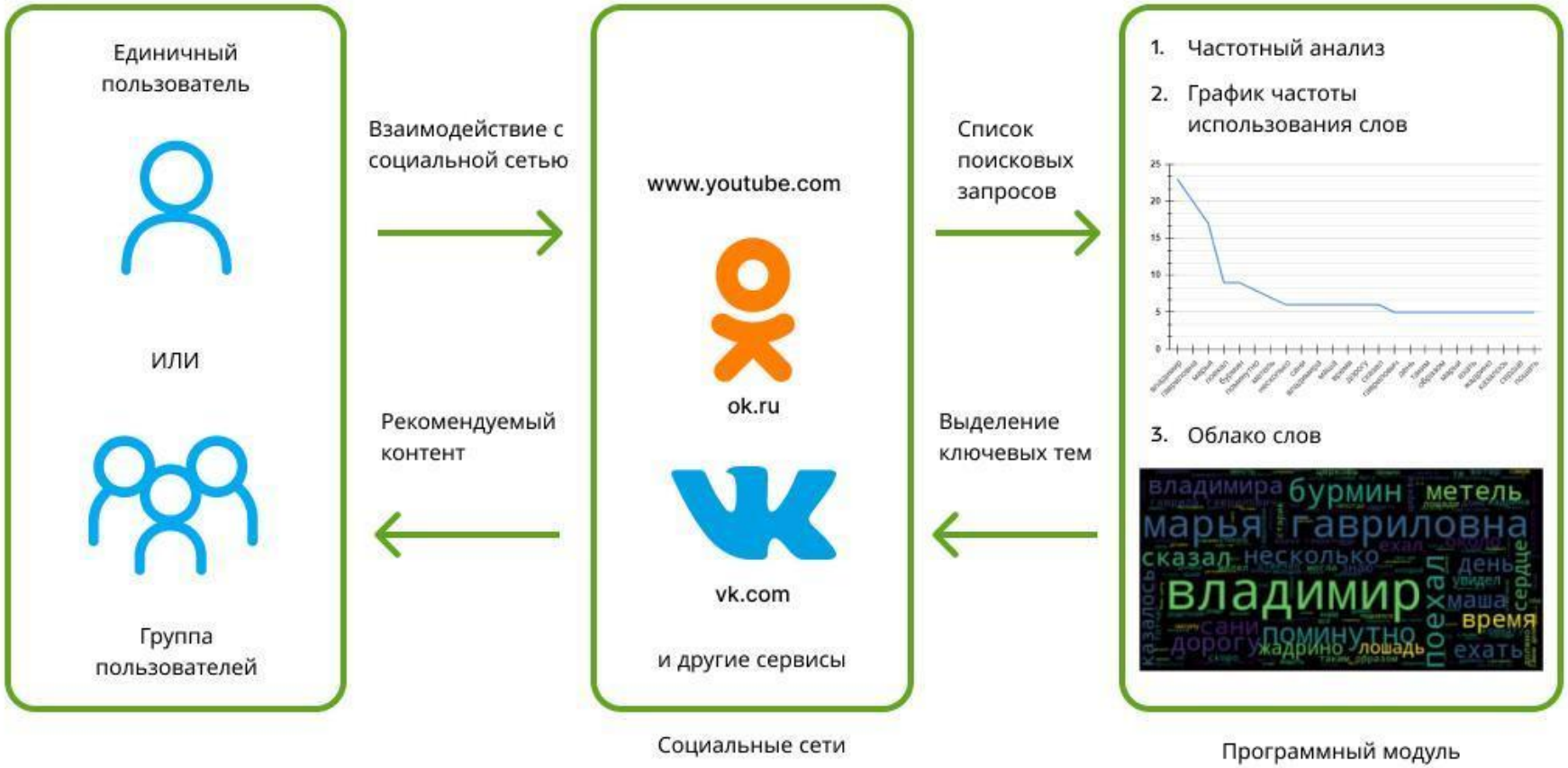

Рисунок 6 – функционирования ПО для изучения поисковой активности пользователей социальных сетей

Основным применением выявленных ключевых тем служит передача этих данных в систему социальной сети. Исходя из этих данных, социальная сеть адаптирует рекомендации для пользователя. Например, если пользователь увлекается программированием, в его ленте будут предложены группы, видео и другие материалы на эту тему.

Для легкости восприятия результатов, программный модуль на основе частотного анализа формирует диаграмму популярности слов и создает облако тегов.

Для последующего машинного анализа программный модуль предоставляет набор данных в виде вектора слов, включая пары "слово - частота его употребления" (рисунок 7). Эту информацию можно далее проанализировать с использованием алгоритмов машинного обучения для выявления скрытых зависимостей. Однако данная проблематика выходит за рамки рассматриваемой бакалаврской работы.

Для воплощения описанных функций в программном модуле применяются следующие готовые компоненты:

– библиотека NLTK для обработки текстовых данных;
– библиотека Matplotlib для визуализации данных в виде графиков;
– библиотека WordCloud для создания облака тегов;

Применение готовых компонентов значительно упрощает процесс разработки ПО, так как многие необходимые функции уже включены в их методы.

Для работы программного модуля выбрана платформа облачных вычислений Google Colab. Тем не менее, при необходимости, его можно запустить локально на определенном сервере с помощью платформы Anaconda.

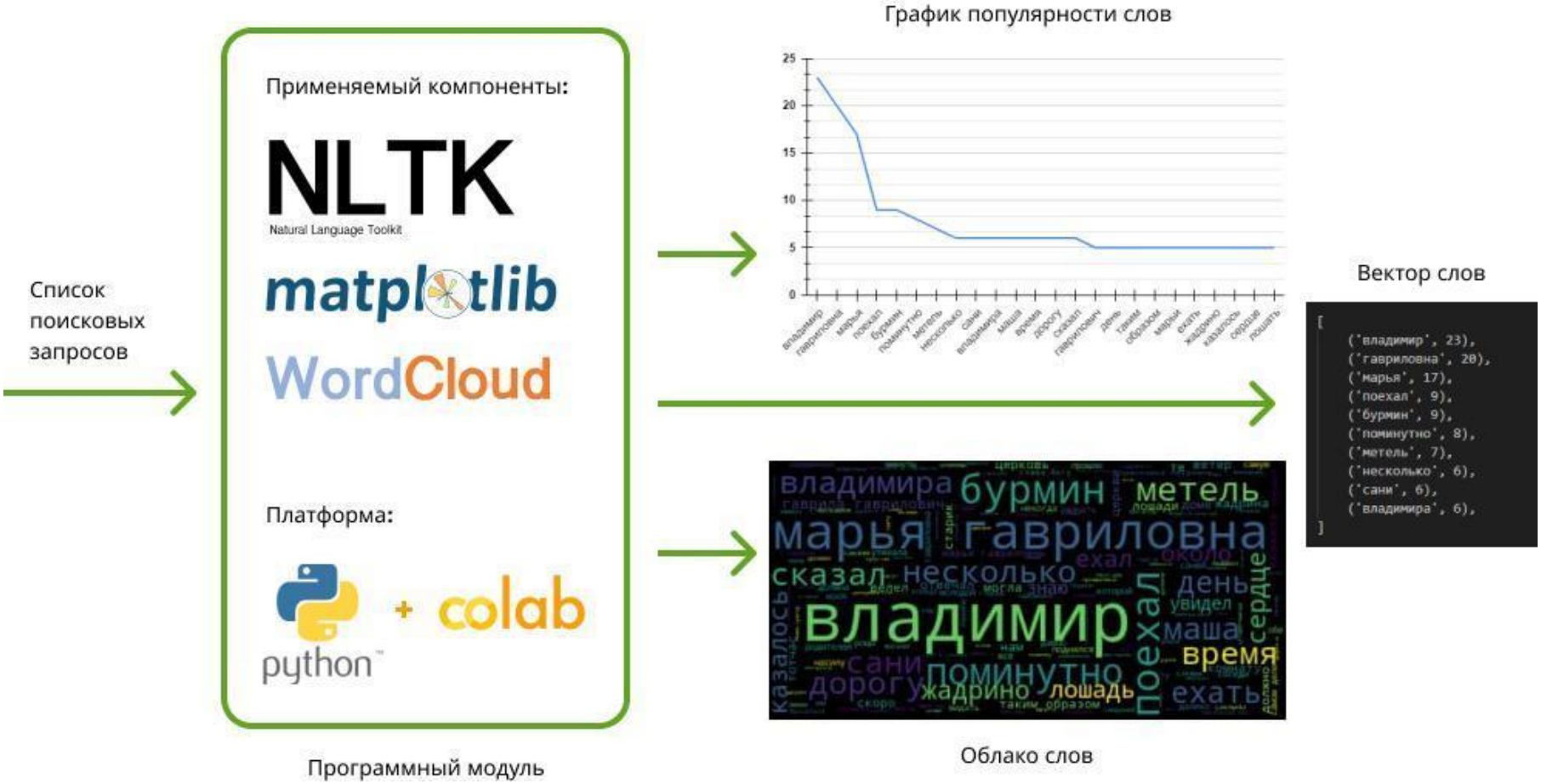

Рисунок 7 – Модель действия ПО для исследования поисковой активности пользователей в соцсетях

Теперь, после определения функциональных возможностей программного модуля, подробно рассмотрим алгоритм обработки текстовых данных.

2.2 Подробное описание алгоритма обработки текстовых данных из поисковых запросов

Давайте подробно рассмотрим процесс анализа текстовых данных из поисковых запросов, который состоит из нескольких ключевых этапов (см. рисунок 8):

Загрузка данных:

- начало процесса закладывается загрузкой списка запросов, который хранится в текстовом файле. Это может быть, например, CSV-файл или простой текстовый документ;
- после загрузки содержимого файла, все данные собираются и помещаются в строковую переменную для дальнейшего анализа;

Токенизация:

- этот этап представляет собой процесс разделения исходного текста на базовые единицы или "токены". В большинстве случаев токены представляют собой отдельные слова или фразы;
- во время этой стадии удаляются различные служебные символы, в том числе знаки препинания и символы переноса строки, что позволяет очистить список от лишних элементов;

Удаление стоп-слов:

- стоп-слова - это общеупотребительные слова, которые встречаются почти в каждом тексте, но при этом не несут конкретного значения для анализа (например, "и", "или", "на" и т. д.);
- удаление этих слов помогает сфокусироваться на ключевых словах и фразах, которые действительно важны для понимания смысла текста;

Частотный анализ:

- на этом этапе для каждого уникального слова из списка вычисляется, сколько раз оно встречается в исходных данных;
- результатом является массив или список, в котором каждому слову соответствует число его упоминаний;

Визуализация:

- с использованием полученного массива данных строятся графические представления: графики, демонстрирующие популярность слов, и облака тегов;
- эти визуализации помогают легко и наглядно увидеть, какие слова и фразы являются наиболее релевантными или часто встречающимися в запросах;

Этот метод анализа позволяет глубоко понять интересы и предпочтения пользователей на основе их поисковых запросов в социальных сетях, выявляя ключевые темы и направления интереса.

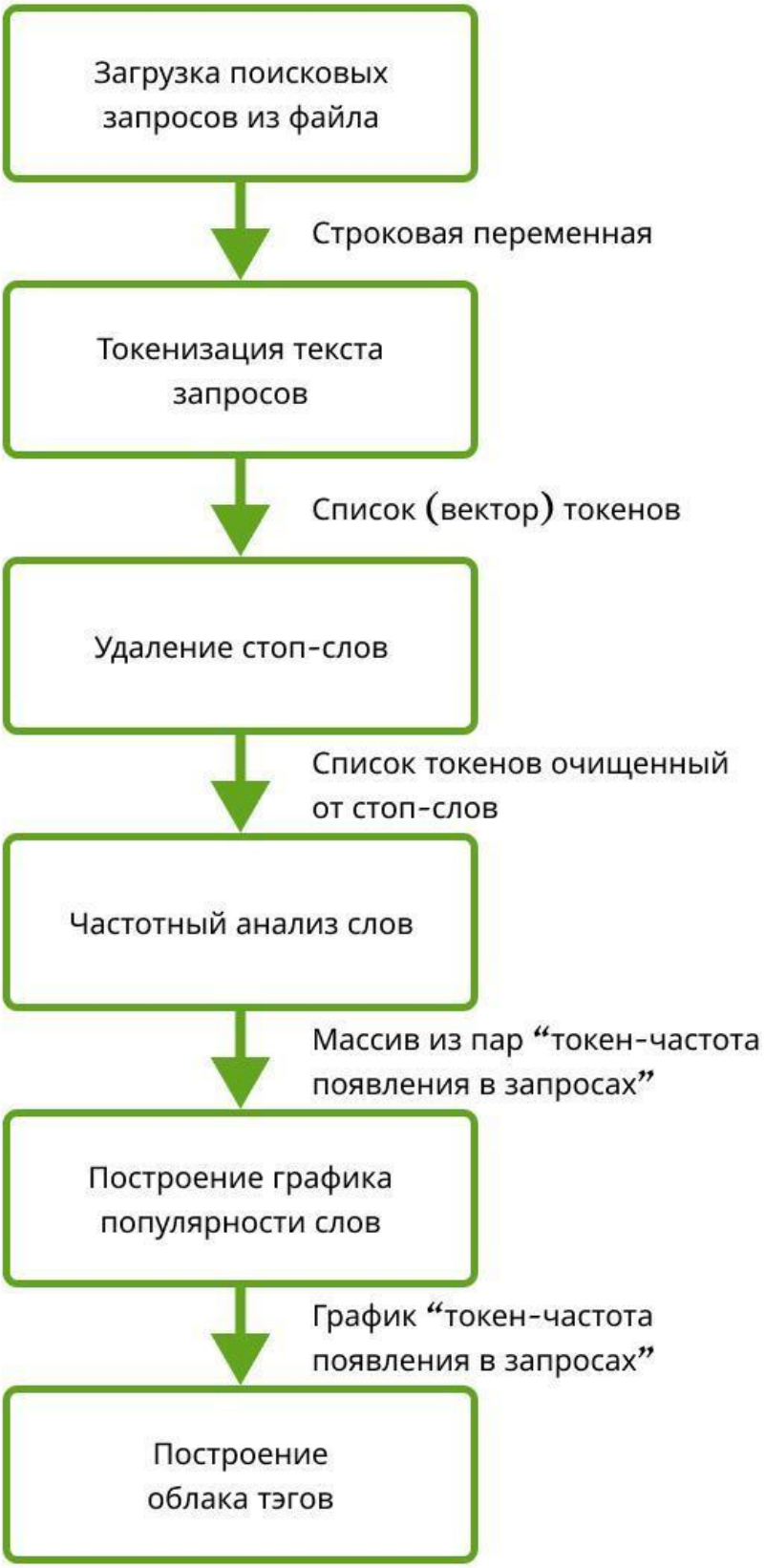

Рисунок 8 – Алгоритм анализа данных

График формируется следующим способом: на оси X размещаются токены, упорядоченные по убыванию частоты их появления, а на оси Y отображается число их упоминаний.

Для создания облака слов используется прямоугольное полотно, где каждый токен из запросов пользователя представлен в виде слова. При этом размер шрифта слова прямо пропорционален его частоте встречаемости: чем чаще слово появляется, тем оно крупнее.

Теперь, после подробного разбора методики анализа текстовых данных, переходим к этапу разработки программного компонента нашей системы.

Выводы по главе 2

Выводы по второй главе бакалаврской работы:

- была представлена модель работы программного обеспечения, которая учитывает взаимодействие пользователя в социальной сети. В рамках этой модели формируется список поисковых запросов пользователя, и определение тем интереса осуществляется путем проведения частотного анализа текста запросов;
- описан метод анализа текстовых данных из поисковых запросов с целью выявления ключевых интересов пользователя. Этот метод включает следующие этапы: извлечение поисковых запросов из документа начинается с токенизации текста, которая представляет собой процесс разделения на отдельные элементы, или токены. Затем применяется фильтрация стоп-слов, исключающая общие слова. После этого проводится частотный анализ слов для определения часто встречающихся терминов. Полученные результаты визуализируются в виде диаграммы популярности терминов и облака слов, представляющего собой графическое изображение ключевых терминов с различным размером, основанном на их частоте в тексте.

Глава 3 Создание приложения для исследования поисковых запросов

3.1 Особенности выполнения программного блока

При создании программного обеспечения был выбран язык программирования Python из-за следующих преимуществ:

- наличие открытой библиотеки NLTK, которая предоставляет методы для анализа текстовых данных [2];
- возможность создавать многоплатформенные приложения без глубокой переработки кода [3];
- поддержка веб-инструментов для интерактивной разработки скриптов, например, Google Colab;
- доступность бесплатной библиотеки wordcloud для представления данных в графическом виде;

Разработанный программный модуль предоставляет такой функционал:

- импорт данных поисковых запросов из файла и их просмотр;
- первичный анализ текстовой информации;
- исключение стоп-слов из текстовых запросов;
- преобразование слов к их базовой форме;
- частотное исследование слов;
- отображение результатов частотного исследования в форме словесного облака;

Рассмотрим детали реализации данного ПО.

Чтобы загрузить данные пользовательских запросов, применяется базовый метод open(). В параметрах указываются: режим открытия файла (“r” – режим чтения), кодировка и наименование файла. Данные после чтения файла сохраняются в переменной под названием text (рисунок 9).

```
f = open('text.data', "r", encoding="utf-8")
text = f.read
```

Рисунок 9 – Код для импорта пользовательских поисковых запросов из файла

Далее проводится проверка типа данных, импортированных из файла. Этот этап критичен, поскольку при преобразовании тип данных может быть интерпретирован неверно.

Чтобы определить тип данных, применяется функция type(). Для определения объема загруженных данных применяется функция len(). Если речь идет о строковых переменных, то данная функция предоставляет информацию о количестве символов в тексте (рисунок 10).

```
typeof(text)
str

len(text)
22968
```

Рисунок 10 – Код для верификации правильности импортированных данных

Чтобы обеспечить пользователю контроль над загружаемой информацией, на экран демонстрируются первые 300 символов импортированных данных. Это достигается при помощи операции взятия подстроки, как демонстрируется на рисунке 11. Такой подход позволяет пользователю быстро определить, были ли данные загружены корректно, и проверить наличие спецсимволов в тексте.

```
text[:300]
'Метель \n\n\nКони мчатся по буграм, \n\nТопчут снег глубокий
клоками; \n\nЧерный вран, свистя крылом, \n\nВьется над санями
гривы\n\n\xa0\xa0\xa0Ж'
```

Рисунок 11 – Код для отображения первых 300 символов из текста

Далее производится первичный анализ текстовой информации. В этом процессе текст очищается от знаков препинания, специфических символов (как, например, символы перевода строки) и ненужных пробелов.

Перед началом очистки, все символы текста приводятся к нижнему регистру. Такой шаг обусловлен необходимостью обеспечения корректности результатов при частотном анализе. Ведь слова, записанные с использованием букв разного регистра, интерпретируются как разные строковые последовательности. Приведение к нижнему регистру достигается через стандартный метод lower(), встроенный в язык программирования. Код, который выполняет это действие, представлен на рисунке 12.

```
text = text.lower()
```

Рисунок 12 – Код для конвертации всех символов текста в нижний регистр

Следующий этап — удаление из текста знаков пунктуации. Мы можем использовать готовый набор символов из строки punctuation. Чтобы обратиться к этой строке, нужно импортировать библиотеку string. Для информирования пользователя на экране отображается содержимое этой строки (смотрите рисунок 13).

```
import string
string.punctuation

'!#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

Рисунок 13 – Загруженный набор символов для очищения текстовой информации

Как показано на рисунке 14, строка в переменной punctuation не включает специальные символы и некоторые распространенные знаки пунктуации. Именно поэтому строка punctuation была расширена с помощью программирования.

```
spec_chars = string.punctuation + '\n\xa0«»\t-...'

text = "".join([ch for ch in text if ch not in spec_chars])
```

Рисунок 14 – дополнительные символы для очистки текста

Для проведения очистки исходных данных был создан код, который затем был включен в функцию с наименованием remove_chars_from_text. В эту функцию передаются два аргумента: исходный текст (text) и символы (chars), которые требуется удалить. Для оценки эффективности выполнения кода вставлена команда %%time, которая измеряет время выполнения и отображает результат в миллисекундах.

Пример использования данной функции с переменными text и spec_char представлен на рисунке 15.

```
def remove_chars_from_text(text, chars):
    return "".join([ch for ch in text if ch not in chars])

text = remove_chars_from_text(text, spec_chars)
```

```
Wall time: 4.02 ms
```

Рисунок 15 – Вызов функции для очистки текста

При необходимости произвести удаление числовых значений из текста, можно воспользоваться функцией remove_chars_from_text. Для этого необходимо передать в аргументы этой функции значение из встроенной переменной string.digits. На рисунке 16 представлен пример соответствующего кода.

```
text = remove_chars_from_text(text, string.digits)
```

```
Wall time: 2.99 ms
```

Рисунок 16 – Очистка текста от числовых значений

Представленный код для очистки текста можно назвать эффективным, так как тестирование указывает на то, что обработка текста объемом в 23000 символов занимает лишь несколько миллисекунд.

В последующем происходит токенизация текста. Этот процесс включает в себя разбиение текста на отдельные элементы, такие как слова. Для этой задачи в приложении применяется функция word_tokenize() из библиотеки nltk.

Результатом токенизации является список, сохраненный в переменной text_tokens. Содержимое этой переменной можно просмотреть с помощью операции взятия подстроки, пример которой представлен на рисунке 17.

```python
from nltk import word_tokenize
text_tokens = word_tokenize(text)

print(type(text_token), len(text_tokens))
text_tokens[:10]
```

```
<class 'list'> 3402
[
    'метель',
    'кони',
    'мчаться',
    'по',
    'буграм',
    'топчут',
    'снег',
    'глубокой',
    'вот',
    'в',
]
```

Рисунок 17 – Программный код для разделения текста на токены

Далее выполняется частотный анализ появления токенов в первоначальных поисковых запросах. Этот анализ заключается в подсчете частоты упоминания каждого токена. Для этого используется функция FreqDist() из библиотеки nltk. Результаты частотного анализа сохраняются в форме словаря в переменной fdist.

С использованием метода most_common можно получить список наиболее часто встречающихся токенов в поисковых запросах. Этому методу передается число, указывающее, сколько токенов нужно отобразить. Код, демонстрирующий выполнение частотного анализа, изображен на рисунке 18.

```
from nltk.probability import FreqDist
fdist = FreqDist(text)
fdist
```

```
Wall time: 6.98 ms
FreqDist({'и': 146, 'в': 101, 'не': 69, 'что': 54, 'с': 44, 'он': 42,
```

```
fdist.most_common(5)
```

```
[('и', 146), ('в', 101), ('не', 69), ('что', 54), ('с', 44)]
```

Рисунок 18 – Программный код для частотного анализа

Для отображения данных частотного анализа используется функция plot. Она позволяет графически представить наиболее частые теги в поисковых запросах. У функции plot два аргумента:

– количество наиболее популярных токенов, которые будут отображаться на графике;
– метод подсчёта частоты: если значение cumulative=false, то частотность каждого токена рассматривается независимо от других;

Построение графика происходит так: токены по оси X распределены в зависимости от убывания их частоты упоминания. Ось Y, называемая "Counts", показывает количество упоминаний каждого токена. Значения соединены линиями.

Код и образец графического отображения частотного анализа можно увидеть на рисунке 19.

Согласно рисунку, союзы и предлоги встречаются чаще всего, но они не несут особой информативности для анализа пользовательских предпочтений. По этой причине такие слова стоит исключить. Процесс исключения нерелевантных союзов и предлогов называется удалением стоп-слов.

```
fdist.plot(30, cumultive=False)
```

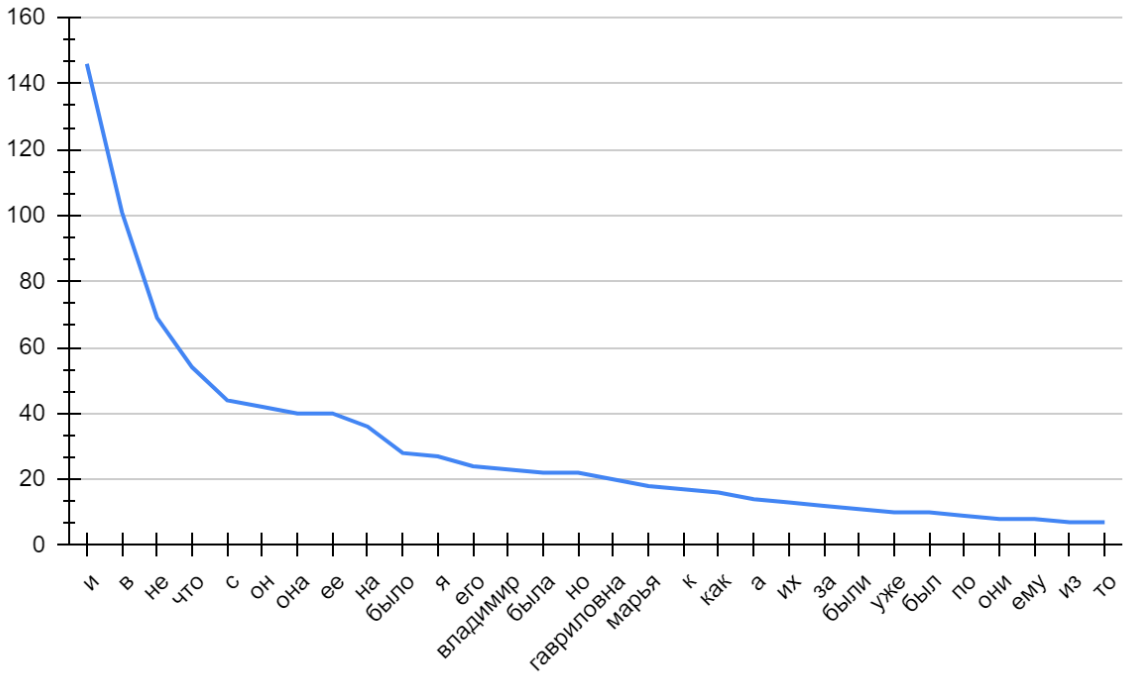

Рисунок 19 – Код отображения данных частотного анализа

Для исключения стоп-слов необходимо начать с формирования словаря, содержащего эти слова. В библиотеке nltk.corpus уже доступен готовый словарь стоп-слов на русском языке. Путем применения функции words() можно получить список этих слов и сохранить его, например, в переменной russian_stopwords. Это также дает возможность расширять этот список с помощью метода extend().

Для определения количества слов в словаре используется функция len, выводящая длину списка в виде числа.

По умолчанию в библиотеке nltk.corpus список русских стоп-слов содержит 153 слова. Пример кода для создания списка стоп-слов представлен на рисунке 20.

```
from nltk.corpus import stopwords
russian_stopwords = stopwords.words("russian")
russian_stopwords.extend(['это', 'нею'])
```

Рисунок 20 – Программный код создания списка стоп-слов

Для устранения стоп-слов из списка токенов используется программа, представленная на рисунке 21. Обычно время выполнения этого этапа не превышает 10 миллисекунд. Для проверки результатов удаления стоп-слов осуществляется подсчет оставшихся токенов с использованием функции len, которая определяет длину списка text_tokens.

```
text_tokens = [token.strip() for token in text_tokens if token not in russian_stopwords]
```

```
Wall time: 6.98 ms
```

Рисунок 21 – Программный код для исключения стоп-слов из списка

После удаления стоп-слов из списка токенов проводится частотный анализ, основанный на ранее описанном алгоритме. В этом контексте применяется метод FreqDist() из библиотеки nltk. Частота появления каждого токена сохраняется в словаре под именем fdist_sw. Чтобы отобразить самые распространенные токены в поисковых запросах, используется метод most_common. Код и результаты анализа после удаления стоп-слов можно увидеть на рисунке 22.

```
fdist_sw = FreqDist(text)
fdist_sw.most_common(10)
```

```
[
    ('владимир', 23),
    ('гавриловна', 20),
    ('марья', 17),
    ('поехал', 9),
    ('бурмин', 9),
    ('поминутно', 8),
    ('метель', 7),
    ('несколько', 6),
    ('сани', 6),
    ('владимира', 6),
]
```

Рисунок 22 – Программный код частотного анализа

Далее проводится визуализация данных частотного анализа с помощью облака слов. В этом облаке размер слова коррелирует с частотой его появления в поисковых запросах: чем чаще слово упоминается, тем оно крупнее. Этот способ представления позволяет быстро понять ключевые интересы пользователя на основе его поисковой активности.

Для создания облака слов используется библиотека worldcloud, а для демонстрации готового облака - библиотека matplotlib.

Перед созданием облака текст подвергается обработке с помощью метода join(). А для создания самого облака применяется метод generate(). Пример кода для формирования облака слов можно увидеть на рисунке 23.

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

text_raw = " ".join(text)

wordcloud = WordCloud.generate(text_raw)
```

Рисунок 23 – Код отображения данных частотного анализа

Код и образец созданного облака слов представлены на рисунке 24.

```
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Рисунок 24 – Создание облака слов на основании данных частотного анализа

Теперь протестируем функционал приложения на основе реальных поисковых запросов из социальных сетей.

3.2. Результаты тестирования приложения

Приложение тестировалось на основе поисковых запросов, совершенных за день в различных социальных и поисковых платформах. К ним относятся такие запросы, как «новости», «дзен», «вконтакте».

На этапе разделения данных на токены было выявлено 60 уникальных единиц. Перед исключением стоп-слов проведен анализ частотности этих токенов. Результаты этого анализа можно увидеть на рисунке 20.

После исключения стоп-слов число уникальных токенов уменьшилось до 50. Данные частотного анализа после этого этапа изображены на рисунке 26.

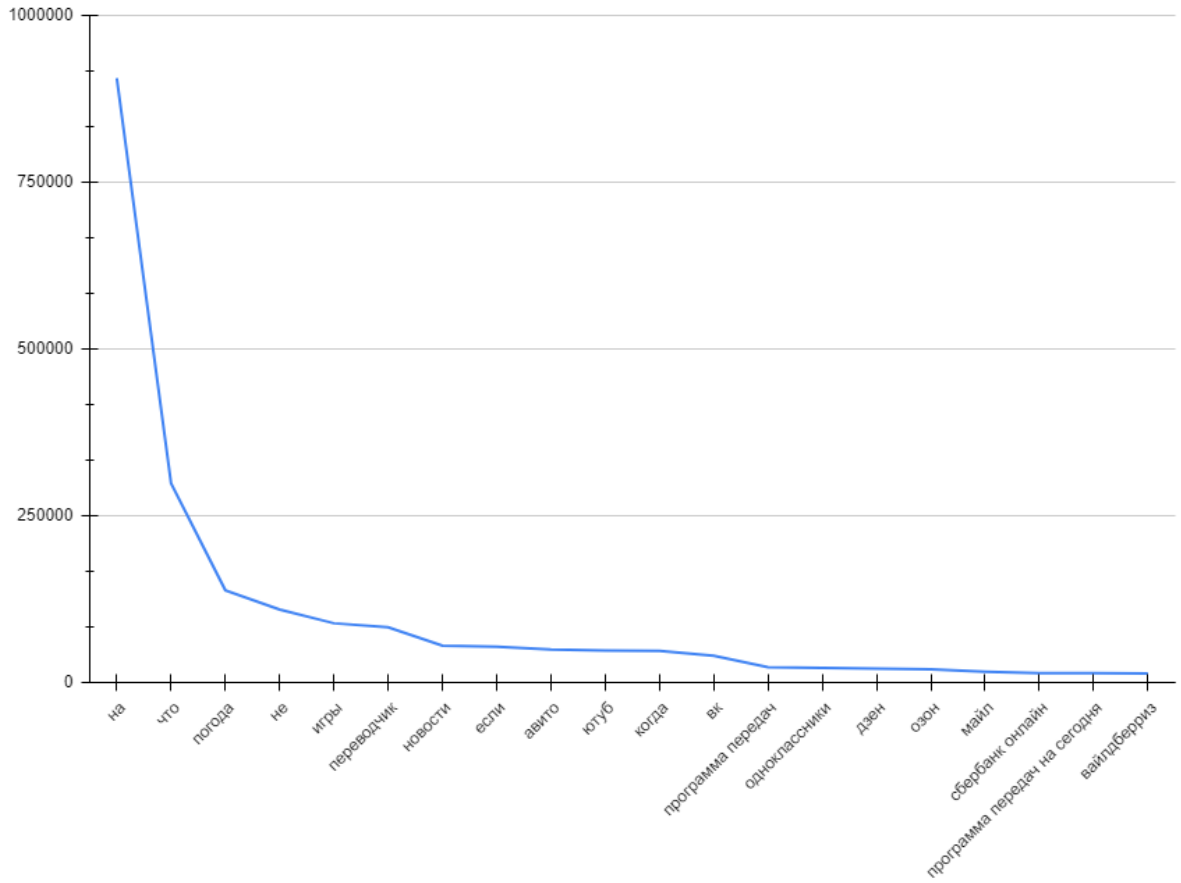

Рисунок 25 – Частотный анализ до удаления стоп-слов

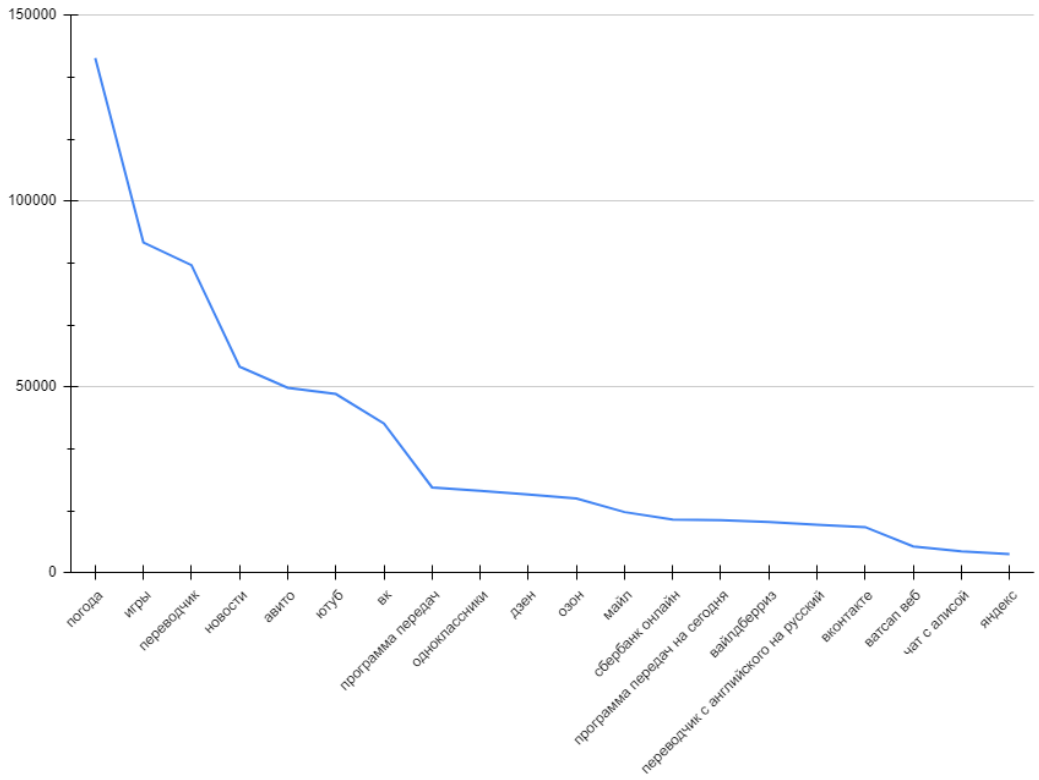

Рисунок 26 – Частотный анализ после удаления стоп-слов

На основе данных частотного анализа приложение было построено облако слов, представленное на рисунке 27.

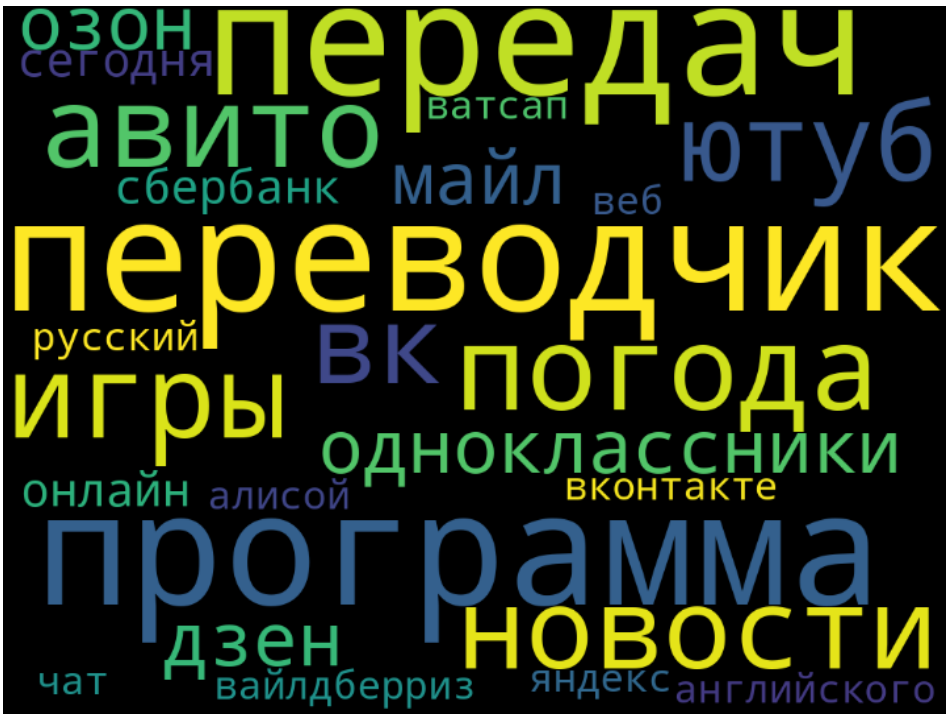

Рисунок 27 – Визуализация результатов частотного анализа

Из рисунка 26 видно, что ключевыми темами для пользователя, осуществляющего поисковые запросы, стали погода, игры и переводчик.

На основе данных тестирования приложения можно заключить, что оно функционирует правильно и предоставляет точные результаты при анализе поисковых запросов.

3.3. Расчет экономической эффективности

Рассчитаем экономическую эффективность для компании АО "НИИ АО", для которой разрабатываем информационную систему для анализа поисковых запросов в социальных сетях.

Затраты на разработку:

- Затраты на персонал, оборудование, программное обеспечение: 550000 рублей;
- Затраты на техническую инфраструктуру: 250000 рублей;
- Общие затраты: 800000 рублей;

Ожидаемые доходы от использования системы для анализа поисковых запросов: 900000 рублей в год. Улучшение анализа на 20% может значительно повысить эффективность использования системы и привести к улучшению бизнес-показателей. Необходимо учесть потенциальные риски, такие как технические проблемы, изменения в требованиях клиентов и конкурентные факторы. Также необходимо сравнить этот проект с альтернативными решениями, чтобы убедиться в его конкурентоспособности.

Проект по разработке информационной системы для анализа поисковых запросов в социальных сетях предоставляет обширные возможности для улучшения эффективности и достижения положительного ROI. Рекомендуется регулярное обновление анализа с учетом изменений во внешней среде и фактических данных.

Выводы по главе 3

Выводы к третьей главе бакалаврской работе:

- было создано приложение на Python для анализа текстовых поисковых запросов. Его функционал включает в себя загрузку и просмотр данных, предобработку текста, удаление стоп-слов, приведение слов к их базовой форме, частотный анализ и визуализацию наиболее популярных слов с использованием облака слов;
- проведено тестирование программы на реальных текстовых запросах, и результаты показали надежность и эффективность разработанного приложения;

Заключение

В заключении рассмотрим итоги бакалаврской работы:

- изучение литературы показало, что важной целью для компаний, управляющих социальными сетями, является удержание аудитории на своих ресурсах;
- чтобы привлекать внимание пользователей, платформам нужно выявлять их интересы, чтобы предлагать актуальный контент;
- в данной работе предложен способ определения интересов посетителей на основе их текстовых запросов, анализируемых методами text mining;
- литературный обзор выделил ключевые этапы text mining: сбор информации, обработка текста, его фильтрация, преобразование признаков и анализ данных;
- используя методологию IDEF0, было проведено моделирование процесса "Определение интересов пользователя", что выделило ключевые компоненты процесса и определило функциональную роль планируемой информационной системы;
- раскрыта последовательность программы, которая включает сбор информации о пользователях, создание списка запросов и выявление интересов путем анализа частотности встречаемости текстовых элементов;
- предложен алгоритм анализа запросов интересующих тем. Этот алгоритм обрабатывает и анализирует текстовые запросы, создание графиков популярности слов и облаков слов;
- на Python создано приложение для анализа запросов, включая этапы обработки, очистки от лишних слов, приведение слов к их основной форме и визуализацию результатов;
- Программа была протестирована на реальных текстовых запросах, что подтвердило ее эффективность и корректную работу.

Список используемой литературы и используемых источников

1. Агеев М. С. Автоматическая рубрикация текстов: методы и проблемы / М.С. Агеев, Б.В. Доброе, Н.В. Лукашевич // Ученые записки казанского государственного университета, 2011. – №5. – с. 26-40

2. Григорьев Е.А. Разведочный анализ данных с помощью Python / Григорьев Е.А., Климов Н.С. // E-Scio. 2021. №3 (42). URL: https://cyberleninka.ru/article/n/razvedochnyy-analiz-dannyh-s-pomoschyu-python (дата обращения: 22.09.2023).

3. Гришков, Д.Ю. Язык высокого уровня программирования Python / Гришков Данила Юрьевич, Аусилова Назерке Мырзабековна // НИР/S&R. 2023. №1 (9). URL: https://cyberleninka.ru/article/n/yazyk-vysokogo-urovnya-programmirovaniya-python (дата обращения: 22.09.2023).

4. а Ершов, В.Е. Тенденции развития рекламной деятельности в социальных сетях / Ершов Вадим Евгеньевич // Вестник евразийской науки. 2017. №4 (31). URL: https://cyberleninka.ru/article/n/tendentsii-razvitiya-reklamnoy-deyatelnosti-v-sotsialnyh-setyah (дата обращения: 22.09.2023).

5. Корелов, С.В. Предобработка текстов электронных писем в задаче обнаружения спама / С.В. Корелов, А.М. Петров, Л.Ю. Ротков, А.А. Горбунов // Труды учебных заведений связи, 2021. – №5. – с. 81-92

6. Леоненков А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose [Электронный ресурс] : учебное пособие. М. : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. 218 с. [Электронный ресурс]. URL: https://www.iprbookshop.ru/97554.html (дата обращения: 06.09.2023).

7. Маннинг, К.Д. Введение в информационный поиск / Г Кристофер Д. Маннинг, Прабхакар Рагхаван, Хайнрих Шютце. : Пер. с англ. - М. : ООО “И.Д. Вильямс”, 2016 - 438с.

8. Мкртычев С.В., Гущина О.М., Очеповский А.В. Прикладная информатика. Бакалаврская работа [Электронный ресурс] : электрон. учеб-метод. пособие. Тольятти. ТГУ: Изд-во ТГУ, 2020. 2 оптический диск.

9. Тарасова А.Н. Сентиментальный анализ постов в социальных сетях посредством Python / Тарасова А.Н., Иванов К.О. // Символ науки. 2023. №4-1. URL: https://cyberleninka.ru/article/n/sentimentalnyy-analiz-postov-v-sotsialnyh-setyah-posredstvom-python (дата обращения: 22.09.2023).

10. Чибирова, М.Э. Анализ данных и регрессионное моделирование с применением языков программирования Python и R / Чибирова Марина Эльбрусовна // Научные записки молодых исследователей. 2020. №3. URL: https://cyberleninka.ru/article/n/analiz-dannyh-i-regressionnoe-modelirovanie-s-primeneniem-yazykov-programmirovaniya-python-i-r (дата обращения: 22.09.2023).

11. Amasaki, S. The Effects of Vectorization Methods on Non-Functional Requirements Classification / Sousuke Amasaki, Pattara Leelaprute // 2019 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2019. – IEEE, Prague, Czech Republic, 2019. – pp.56-76.

12. Bird, S. Natural Language Processing with Python / Steven Bird, Ewan Klein, Edward Loper. – Published by O'Reilly Media, Inc., 2015. – 452p.

13. Bugueno, M. Learning to combine classifiers outputs with the transformer for text classification / Margarita Bugueno, Marcelo Mendoza // Intelligent Data Analysis, 2020 – № 24. – pp. 15-41

14. Business Process Model and Notation [Электронный ресурс]. URL: https://www.omg.org/spec/BPMN/3.0/About-BPMN/ (дата обращения: 22.09.2023).

15. Gao, G. Research on Routing Selection Algorithm Based on Genetic Algorithm / Guohong Gao, Baojian Zhang, Xueyong Li, Jinna Lv // International Conference on Intelligent Computing and Information Science – International Conference, ICICIS 2011, Chongqing, China, January 19-20, 2014. Proceedings,

Part II: Intelligent Computing and Information Science. – Springer-Verlag Berlin Heidelberg 2014. – pp. 253-258

16. Higuchi, T. Special Section on Nonparametric Approach to Time Series Analysis / Tomoyuki Higuchi, Genshiro Kitagawa // Annals of the Institute of Statistical Mathematics, 2008. - №55 (135). - Springer Nature Switzerland AG 2006. - pp.103-114

17. Jurafsky, D. Speech and Languge Processing: An Introduction to Natural Language Proccerring, Computation Linguistics and Speech Recognition / Jurafsky, Daniel; H. James, Martin. – Stanford University, 2022. – 414 p.

18. Kowsari, K. Text Classification Algorithms: A Survey / Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, Donald Brown // Machine Learning on Scientific Data and Information. – Cornell University, 2020. – pp. 101-165.

19. Srividhya, V. Evaluating Preprocessing Techniques in Text Categorization / V. Srividhya, R. Anitha // International Journal of Computer Science and Application Issue 2010. – pp. 52-53.

20. Sun, C. How to Fine-Tune BERT for Text Classification? / Chi Sun, Xipeng Qiu, Yige Xu, Xuanjing Huang // Computation and Language, 2021. – Cornell University, 2021. – pp. 25-35.