

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Г.И. Радченко
« ___ » _____ 2020 г.

Сервис для поиска целевой аудитории в социальных сетях с анализом
тональности комментариев при помощи сверхточных нейронных сетей

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,
к.п.н., доцент каф. ЭВМ
_____ Ю.Г. Плаксина
« ___ » _____ 2020 г.

Автор работы,
студент группы КЭ-405
_____ М.А. Мещанкин
« ___ » _____ 2020 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ
_____ С.В. Сяськов
« ___ » _____ 2020 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

_____ Г.И. Радченко

«___» _____ 2020 г.

ЗАДАНИЕ

на выпускную квалификационную работу бакалавра

студенту группы КЭ-405

Мещанкин Михаил Александрович

обучающемуся по направлению

09.03.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Сервис для поиска целевой аудитории в социальных сетях с анализом тональности комментариев при помощи сверточных нейронных сетей» утверждена приказом по университету от 24 апреля 2020 г. №627

2. **Срок сдачи студентом законченной работы:** 1 июня 2020 г.

3. **Исходные данные к работе:**

1. Пьюривал, С. Основы разработки веб-приложений. — СПб.: Питер, 2015. — 272 с;
2. Дронов, В.А. Django 2.1. Практика создания веб-сайтов на Python. — СПб.: БХВ-Петербург, 2019. — 672 с;

3. Стружкин, Н. П. Базы данных: проектирование. Практикум : учебное пособие для академического бакалавриата / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2018. — 291 с;

4. Analysis of Deep Convolutional Neural Network Architectures – <https://pdfs.semanticscholar.org/6831/bb247c853b433d7b2b9d47780dc8d84e4762.pdf>.

4. Перечень подлежащих разработке вопросов:

- провести анализ предметной области;
- спроектировать веб-приложение;
- обучить нейронную сеть на основе размеченного набора данных;
- реализовать спроектированное веб-приложение;
- протестировать созданное веб-приложение.

5. Дата выдачи задания: 1 декабря 2019 г.

Руководитель работы _____ /Ю.Г. Плаксина/

Студент _____ /М.А. Мещанкин /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2020	
Разработка модели, проектирование	01.04.2020	
Реализация системы	01.05.2020	
Тестирование, отладка, эксперименты	15.05.2020	
Компоновка текста работы и сдача на нормоконтроль	24.05.2020	
Подготовка презентации и доклада	30.05.2020	

Руководитель работы _____ /Ю.Г. Плаксина/

Студент _____ /М.А. Мещанкин/

Аннотация

Мещанкин М.А. Сервис для поиска целевой аудитории в социальных сетях с анализом тональности комментариев при помощи сверточных нейронных сетей. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2020, 88 с., библиогр. список – 24 наим.

В рамках выпускной квалификационной работы было разработано веб-приложение для поиска целевой аудитории социальной сети ВКонтакте. Главной целью проекта является разработка сервиса для систематизации поиска целевой аудитории для малого бизнеса ВКонтакте и предоставления возможностей для взаимодействия с ней. Систематизация поиска реализована через автоматизацию поиска сообществ и людей, а также использование нейронных сетей с помощью, которых выделяются сильные и слабые стороны конкурентов.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	9
1.1. ОБЗОР АНАЛОГОВ.....	10
1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ.....	13
1.2.1 ВЫБОР ЯЗЫКА ПРОГРАММИРОВАНИЯ ДЛЯ СЕРВЕРНОЙ ЧАСТИ.....	13
1.2.2 ВЫБОР ФРЕЙМВОРКА.....	15
1.2.3 ВЫБОР АРХИТЕКТУРЫ НЕЙРОННОЙ СЕТИ.....	16
1.2.4 ВЫБОР СИСТЕМЫ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ.....	20
1.2.5 ВЫБОР СРЕДСТВ РАЗРАБОТКИ КЛИЕНТСКОЙ ЧАСТИ ВЕБ- ПРИЛОЖЕНИЯ.....	21
1.3. ВЫВОД.....	22
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ.....	23
2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	23
2.1.1 ОСНОВНЫЕ ТРЕБОВАНИЯ К ФУНКЦИОНАЛУ СИСТЕМЫ ...	23
2.1.2 ТРЕБОВАНИЯ К ФУНКЦИОНАЛУ СИСТЕМЫ ЗАДАЧ.....	24
2.1.3 ТРЕБОВАНИЯ К ФУНКЦИОНАЛУ СИСТЕМЫ АДМИНИСТРИРОВАНИЯ.....	25
2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	25
2.2.1 ТРЕБОВАНИЯ К ПОЛЬЗОВАТЕЛЯМ.....	25
2.2.2 ТРЕБОВАНИЯ К СИСТЕМЕ БЕЗОПАСНОСТИ.....	26
2.2.3 ТРЕБОВАНИЯ К СИСТЕМЕ НАВИГАЦИИ.....	26
2.2.4 ТРЕБОВАНИЯ К СИСТЕМЕ УВЕДОМЛЕНИЙ.....	27
2.2.5 ТРЕБОВАНИЯ К ЛИНГВИСТИЧЕСКОМУ ОБЕСПЕЧЕНИЮ ...	27

3. ПРОЕКТИРОВАНИЕ	27
3.1. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ	27
3.2. АРХИТЕКТУРА НЕЙРОНОЙ СЕТИ.....	28
3.3. ОПИСАНИЕ ДАННЫХ	30
4. РЕАЛИЗАЦИЯ	33
4.1. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ	33
4.2. ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ	55
4.3. РЕАЛИЗАЦИЯ НЕЙРОННОЙ СЕТИ	56
5. ТЕСТИРОВАНИЕ.....	58
5.1. ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ.....	58
6. ЗАКЛЮЧЕНИЕ	65
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	66
ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ФУНКЦИЙ СБОРА ПОСТОВ	69
ПРИЛОЖЕНИЕ Б ИСХОДНЫЙ КОД ФУНКЦИЙ ПОИСКА ГРУПП	73
ПРИЛОЖЕНИЕ Г ИСХОДНЫЙ КОД ФУНКЦИЙ АНАЛИЗА КОММЕНТАРИЕВ	76
ПРИЛОЖЕНИЕ Д ИСХОДНЫЙ КОД ФУНКЦИЙ АНАЛИЗА ГРУПП .	79
ПРИЛОЖЕНИЕ Е ИСХОДНЫЙ КОД ФУНКЦИЙ ПОИСКА ЛИДОВ В ЛИЧНЫХ СООБЩЕНИЯХ.....	80
ПРИЛОЖЕНИЕ Ж ИСХОДНЫЙ КОД ФУНКЦИЙ СБОРА УЧАСТНИКОВ	83
ПРИЛОЖЕНИЕ З ИСХОДНЫЙ КОД ФУНКЦИЙ СБОРА ОБСУЖДЕНИЮ.....	86

ВВЕДЕНИЕ

Сбор и обработка данных для продвижения бизнеса в условиях социальных сетей особенно актуальна в наше время, так как социальные сети с каждым годом увеличивают количество пользователей. Благодаря сбору и анализу данных мы можем понять, кто является целевой аудиторией нашего продукта или услуг, а при помощи искусственного интеллекта оценить их удовлетворенность той или иной услугой.

Целью представленной выпускной квалификационной работы является разработка веб-приложения, обеспечивающего сбор и обработку данных пользователей социальной сети Вконтакте, а также предоставления функционала для взаимодействия с ними.

Для достижения поставленной цели, необходимо решить следующие поставленные задачи:

1. Провести детальный анализ имеющихся аналогов.
2. Провести анализ потребностей пользователей исходя из них составить функционал разрабатываемой системы.
3. Проанализировать и обосновать выбор средств разработки проекта.
4. Разработать серверную часть веб-приложения.
5. Разработать frontend часть веб-приложения.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

В настоящее время значимость интернета в жизни человека сильно выросла. Появилось множество интернет-магазинов, социальных сетей, форумов, блогов. По данным проведённых исследований только в социальной сети ВКонтакте в месяц оставляют хотя бы 1 публичное сообщение 36 456 000 человек [1]. Эти данные содержат информацию, которая может быть полезна для проведения исследований и анализа. Существует целое направление компьютерной лингвистики. Науки о языке и о способах работы с ним при помощи компьютерных методов [2]. Одно из актуальных направлений компьютерной лингвистики – анализ тональности текстов. Он позволяет оценить эмоции автора и дать им оценку.

В настоящее время на выбор товаров и услуг сильно влияет мнение других людей. По результатам исследований 88% людей учитывают и доверяют отзывам из интернета, а 72% заявили, что они увеличивают их лояльность к компании [3]. А значит, эта информация вызывает интерес у профессионалов, занимающихся анализом поведения пользователей в глобальной сети, а пользователи оставляющие комментарии и отзывы интересуют специалистов по таргетированной рекламе для предложения им похожих товаров и услуг.

Рассмотрим таргетированную рекламу в ВКонтакте. Для нацеливания на аудиторию групп в рамках самой социальной сети существует инструмент “Выбор сообществ”. Но данного инструмента может быть недостаточно для детального выбора аудитории, так как этот инструмент ограничен 100 сообществами и помимо этого, в сообществах могут быть накрученные люди, люди, попавшие туда случайно, не заинтересованные люди. При помощи разрабатываемого сервиса мы сможем устранить данные проблемы. Например, таким способом как пересечение аудитории нескольких групп, что даст нам

аудиторию, которая заинтересована в услуги или товаре. Рекламный кабинет ВКонтakte таких опций не предоставляет. Также не очень удобно анализировать сообщества конкурентов для выявления болей и потребностей их клиентов. Для этого необходимо вручную перебирать комментарии, выписывать их и проводить соответствующую аналитику. В одной группе может быть множество комментариев и кроме полезных, может быть спам и т.д.

Соответственно потребность заказчика в автоматизации сбора информации из социальной сети и последующей работы с ней, а также аналитике сообществ конкурентов в социальной сети ВКонтakte.

1.1. ОБЗОР АНАЛОГОВ

В настоящее время уже существуют сервисы для поиска целевой аудитории: `segment-target`, `vk.barkov`, `smm.sexy`. Но не один из них не использует в своей работе нейронные сети, а также в большинстве нет инструментов для взаимодействия с целевой аудиторией.

Перейдем к рассмотрению аналогов.

`segmento-target.ru` – сервис с такими возможностями, как: поиск по ВКонтakte, сбор данных, выделение активной аудитории и т.д. Автор сервиса: Ричард Попов [4]. Особенностью `segmento` является поиск по собственной базе данных, которую они обновляют раз в 3 дня, что может влиять на актуальность данных, а она играет большую роль в таргетированной рекламе. Подписчики, пришедшие только, что имеют большую ценность для таргетолога, а при обновлении базы данных раз в три дня их можно упустить. Преимущества и недостатки указаны в таблице 1.1.

Таблиц 1.1 – Преимущества и недостатки Segmento-target

Преимущества	Недостатки
Быстрота работы, возможность выделить активную аудиторию, отслеживание людей, которые только вступили, удобное представление выполненных задач.	Segmento работает со своей базой данных, данные на какое-то время могут устаревать. Сервис платный.

vk.barkov.net – предоставляет возможности поиска и сбора людей, сообществ, позволяет находить активных пользователей и т.д. Автор сервиса: Илья Барков [5]. В качестве минусов можно обозначить: нет сохранения задач, после обновления страницы полученный результат пропадает. Нельзя посмотреть выполненную пользователем задачу в понятном для него виде, возможно только сохранение в файл. Преимущества и недостатки указаны в таблице 1.2.

Таблица 1.2 – Преимущества и недостатки vk.barkov.net

Преимущества	Недостатки
Множественность критериев по которым можно искать и собирать людей и сообщества.	Выполненные задачи не сохраняются. Нет удобного отображения выполненной задачи. Сервис платный.

smm.sexu – предоставляет возможности поиска и сбора сообществ и пользователей, поиск активной аудитории, поиск промо постов. В качестве минусом можно отметить: не удобный вывод информации о выполненной задаче, информация автоматически подгружается при прокрутке страницы, что делает её очень длинной при большом количестве информации и создает неудобства для пользователя. Преимущества и недостатки указаны в таблице 1.3.

Таблица 1.3 – Преимущества и недостатки smm.sexy

Преимущества	Недостатки
Есть удобное представление выполненной задачи.	Неудобный интерфейс пользователя. Сервис платный.

Церебро.рф – сервис является одним из лидеров на рынке [6]. Имеет большое комьюнити, которое делится друг с другом полезными советами, имеет онлайн тех. поддержку, есть база кейсов, накопленная за несколько лет, видео уроки по работе с сервисом. Преимущества и недостатки указаны в таблице 1.4.

Таблица 1.4 – Преимущества и недостатки церебро

Преимущества	Недостатки
База кейсов. Удобное отображение выполненных задач. Задачи сохраняются.	Нет функций для взаимодействия с целевой аудиторией. Сервис платный.

1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ

1.2.1 ВЫБОР ЯЗЫКА ПРОГРАММИРОВАНИЯ ДЛЯ СЕРВЕРНОЙ ЧАСТИ

Для разработки серверной части приложения могут быть использованы следующие языки программирования: PHP, JAVA, C#, C, C++, Python. Выбор языков осуществлялся, опираясь на данные индекса ТЮВЕ [7]. Далее рассмотрим достоинства и недостатки перечисленных нами средств для разработки серверной части приложения.

1) PHP

Преимущества:

- кроссплатформенность;
- низкий порог входа;
- бесплатное распространение.

Недостатки:

- неоднозначность кода;
- узкопрофильность.

2) JAVA

Преимущества:

- объектно-ориентированный язык программирования;
- кроссплатформенность;
- автоматическая работа с памятью.

Недостатки:

- низкая производительность.

3)C#

Преимущества:

- объектно-ориентированный язык программирования;
- присутствует синтаксический сахар;
- большое количество библиотек и шаблонов.

Недостатки:

- низкая производительность.

4)C++ и C

Преимущества:

- возможность работать на низком уровне;
- одни из самых быстрых языков программирования [8];
- универсальность, используется в микроконтроллерах, приложениях на персональных ПК, мобильных приложениях и так далее.

Недостатки:

- сложный для освоения новичками;
- нет автоматической работы с памятью.

5)Python

Преимущества:

- низкий порог вхождения;
- большое количество библиотек для всевозможных ситуаций;
- универсальность.

Недостатки:

- медленная скорость исполнения кода.

Для разработки веб-приложения был выбран язык Python. Фактором выбора данного языка было то, что он позволяет разрабатывать сложные сервисы без использования сложных конструкций и тратя на это оптимальное количество времени. Так же Python обладает такими преимуществами, как

простой синтаксис, большое комьюнити, наличие лицензии на свободное распространения программного обеспечения и большое количество фреймворков. Для упрощения разработки в данной работе мы будем использовать фреймворк. Он позволяет сделать систему безопасной, легко масштабируемой и модернизируемой, способной выдержать значительные нагрузки.

1.2.2 ВЫБОР ФРЕЙМВОРКА

Для Python существует множество фреймворков рассмотрим одни из самых популярных и рекомендуемых к изучению: Flask, Bottle, Django [9].

1) Flask

Преимущества:

- встроенный сервер разработки и быстрый отладчик;
- возможность подключения любого ORM;
- поддержка безопасных cookies.

Недостатки:

- легко совершить неверное решение, что приведет к серьезной проблеме в будущем;
- могут возникнуть проблемы с безопасностью.

2) Bottle

Преимущества:

- поддерживает запросы на вывод функциональных выводов;
- удобный доступ к данным форм, загрузке файлов, файлам cookie, заголовкам;
- встроенный сервер разработки HTTP.

Недостатки аналогичны фреймворку Flask.

3) Django

Преимущества:

- хорошая система защиты;
- легко масштабируется;
- возможность писать серверную и клиентскую части сайта на одном языке.

Недостатки:

- медленное развитие;
- использование шаблонов маршрутизации с указанием URL.

Для разработки выберем фреймворк Django. Так как он содержит в себе большинство компонентов, которые понадобятся нам для написания веб-приложения. Дополнительные преимущества: ORM (Object-Relational Mapping), миграция баз данных, авторизация пользователей, панель администрирования, формы, защита от SQL-инъекций и подделки межсайтовых запросов.

1.2.3 ВЫБОР АРХИТЕКТУРЫ НЕЙРОННОЙ СЕТИ

Нейронные сети представляют собой попытку имитировать биологическую нервную систему в отношении как архитектуры, так и стратегий обработки информации [10]. Простейший пример нейронной сети показан на рисунке 1.1.

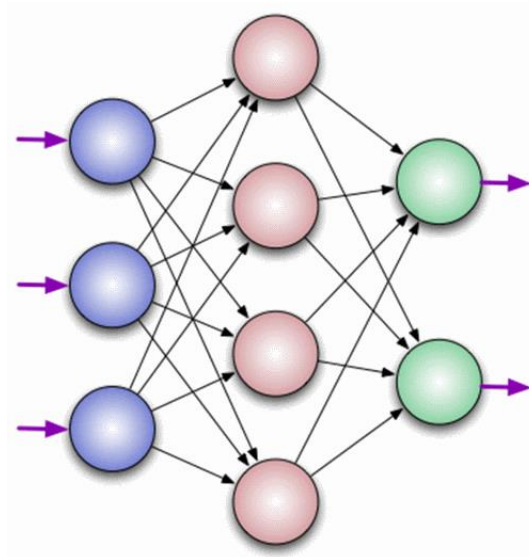


Рисунок 1.1 – Простейшая нейронная сеть

Сеть состоит из простых элементов обработки, которые связаны между собой с помощью весов. Веса представляют силу связи между нейронами, чем больше вес между двумя нейронами, тем больше первый нейрон оказывает влияние на второй, пример на рисунке 1.2 [11].

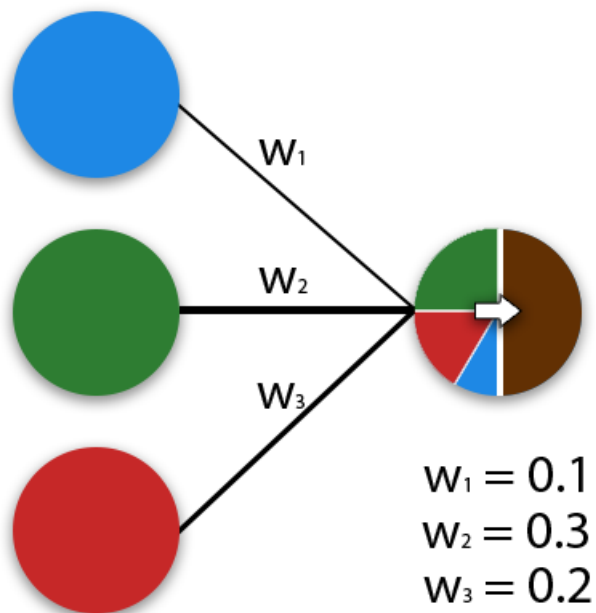


Рисунок 1.2 – Влияния веса

Важной частью нейронной сети является функция активации – это функции, которые принимают входной сигнал и преобразуют его в выходной сигнал [12]. Помимо однослойной нейронной сети, представленной на рисунке 1.1, существуют многослойные. Их отличие в том, что их нейроны находятся в группах слоев, что важно, каждый нейрон прошлого слоя связан с нейронами текущего слоя, пример находится на рисунке 1.3 [13].

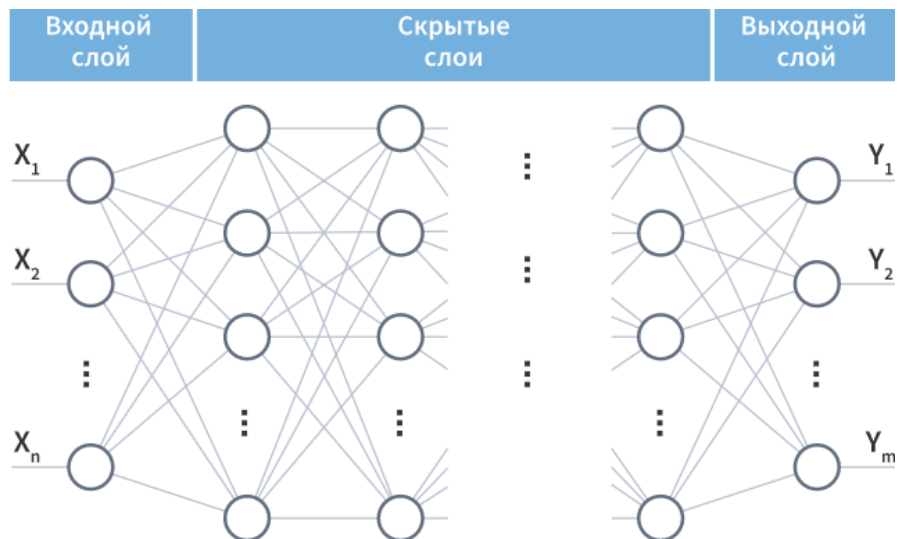


Рисунок 1.3 – Многослойная нейронная сеть

В рамках работы рассмотрим: нейронные сети прямого распространения, рекуррентные нейронные сети, сверточные нейронные сети. Чтобы выбрать из них самую подходящую для решения поставленной задачи рассмотрим их особенности.

Нейронные сети прямого распространения. Неплохо подходят для классификаций. Но, наличие большое количество параметров требует большой выборки для обучения, если у вас специфическая задача, то составить её самостоятельно трудоемкий процесс. В многослойных сетях встречается проблема затухающего градиента. Заключается она в том, что из-за большого количества слоев производная ошибки сильно уменьшается в конце сети, это влияет на то, что веса на входе слабо изменяются [14].

Рекуррентные нейронные сети. Их особенность заключается в том, что они запоминают информацию, извлечённую из предыдущих входных данных, и учитывают ее [15]. В данных сетях существует проблема аналогичная предыдущим. Есть так же проблемы затухания градиента при большой глубине. Медленное обучение из-за повторяющихся вычислений. Для 10 эпох (эпохи – это количество воздействий на модель обучающей выборки. Одна эпоха равняется одному проходу вперед и назад всех обучающих примеров) время обучения в 1.77 раз больше, чем для сверточных нейронных сетей [16]. При использовании функций активации `relu` и `tanh` могут возникать проблемы при обработке длинных последовательностей [17].

Сверточные нейронные сети. Хорошо подходят для классификации, детекции, сегментации. Особенности данных сетей – это слои свертки, которые ищут определённые признаки объекта и слои субдискретизации, которые уменьшают размер свернутого элемента, находят доминирующие признаки и уменьшают вычислительные мощности. Так же могут возникнуть проблемы с затуханием градиента, но её можно избежать, выбрав подходящую функцию активации.

Так как задача нашей сети – определение тональности текстов (эмоциональной окраски положительный текст или отрицательный). Для разработки нейронной сети выберем сверточную архитектуру, так как она показывает хорошие результаты в определении тональности текстов, требует меньшей выборки для обучения, чем сети прямого распространения и по проведённым исследованиям при работе с текстами показывают себя несколько лучше, чем рекуррентные сети [18]. Так же, как было сказано раньше, сверточные нейронные сети быстрее обучаются.

1.2.4 ВЫБОР СИСТЕМЫ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ

Для хранения информации о пользователях необходима база данных. Базы данных бывают двух типов реляционные и нереляционные. Проведем их сравнение и выберем концепцию, которая позволит реализовать поставленные задачи.

Ключевая особенность реляционных баз данных — это хранение информации в строгой структуре, в свою очередь эти структуры зачастую представляют собой объекты из реальной жизни. Преимущества и недостатки реляционных баз данных представлены в таблице 1.5.

Таблица 1.5 – Преимущества и недостатки реляционных баз данных

Преимущества	Недостатки
Данные отображаются в понятном для пользователя виде.	Медленный доступ к данным.
Правила проектирования базируются на математическом аппарате.	Большой расход памяти.
Соответствие требованиям ACID (atomicity, consistency, isolation, durability).	Не всегда данные могут быть представлены в таблице.

Нереляционные базы данных в свою очередь предлагают хранение данных не в строго определенной структуре. Основные модели хранения данных: ключ-значение, документно-ориентированная, колоночно-ориентированная, графовая. Преимущества и недостатки нереляционных баз данных представлены в таблице 1.6.

Таблица 1.6 – Преимущества и недостатки нереляционных баз данных

Преимущества	Недостатки
Требуется меньше затрат для масштабирования чем реляционные базы данных.	Сильная привязка к конкретной базе данных.
Так как не требуется глубоких знаний SQL-запросов, это снижает порог входа для начала работы с NoSQL.	Ограниченность встроенных языков запросов.
Шаринг – распределение информации по разным узлам сети.	Нереляционные базы данных имеют разные средства работы с ними, что усложняет переход с одной базы данных на другую.

Так как база данных будет содержать не больше девяти таблиц (так как сервис находится в стадии разработки) данные можно будет легко представить в виде таблиц. Масштабируемость не требуется. Выберем реляционную базу данных. Из существующих реляционных баз выберем SQLite, так как она по умолчанию встроена в Django, имеет высокую скорость работы и при необходимости не составит труда перенести на более мощную SQL СУБД.

1.2.5 ВЫБОР СРЕДСТВ РАЗРАБОТКИ КЛИЕНТСКОЙ ЧАСТИ ВЕБ-ПРИЛОЖЕНИЯ

Для разработки клиентской часть веб-приложения были выбраны следующие инструменты:

- фреймворк Bootstrap 4;
- язык программирование JavaScript с использованием Ajax и jQuery;
- язык разметки HTML;
- таблица каскадных стилей CSS.

1)Фреймворк Bootstrap 4

Bootstrap – это среда, разработанная для создания динамических веб-сайтов и веб-приложений в основе которой лежит адаптивная сетка. Она

поддерживает все основные браузеры и быструю загрузку адаптивных веб-страниц. Содержит большую библиотеку готовых компонентов, стилей и тем, самым упрощает разработку.

2)jQuery и Ajax

jQuery – это быстрая, небольшая и многофункциональная библиотека JavaScript. Она имеет следующий функционал, как манипулирование HTML страницами, обработка событий, анимация. Так же упрощает взаимодействия с Ajax. Поддержка осуществляется множеством браузеров.

Ajax используется для: создания удобного веб-интерфейса, частичной перезагрузки страницы, что экономит трафик, а так же удобного взаимодействия с пользователями.

1.3. ВЫВОД

Анализ существующих веб-приложений для поиска целевой аудитории показал, что на данный момент на рынке нет сервисов, совмещающих в себе современные технологии, удобный пользовательский интерфейс и возможности взаимодействия с целевой аудиторией, автоматизации рутинной работы с рекламным кабинетом. Из этого можно сделать вывод, что разработка приложения без перечисленных недостатков является актуальной задачей. Требуется разработать приложение, которое не содержит функциональной избыточности, а имеет только необходимые пользователям инструменты. Для решения данной задачи были выбраны следующие решения: язык программирования Python, фреймворк Django, СУБД SQLite, сверточные нейронные сети, язык программирования JavaScript, фреймворк Bootstrap 4, технология AJAX, библиотека jQuery, HTML разметка, каскадные стили CSS.

2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

2.1.1 ОСНОВНЫЕ ТРЕБОВАНИЯ К ФУНКЦИОНАЛУ СИСТЕМЫ

1. Регистрация и авторизация через соц. сеть ВКонтакте.

2. Поддержка системы задач.

3. Решение следующих задач:

а) поиск сообществ в социальной сети ВКонтакте с фильтром по:

- ключевым словам в названии группы;
- количеству участников;
- типу сообщества;
- геоположению.

б) поиск постов:

- с фильтром по дате поста.

в) поиск комментариев с фильтром по:

- ключевым словам;
- дате постов, с которых нужно собирать комментарии;
- количеству лайков;
- максимальной длине комментария.

г) анализ тональности комментариев. Загрузка комментариев осуществляется из ранее созданной задачи сбора комментариев. При выводе результатов задачи добавить диаграмму со статистикой положительных, отрицательных и нейтральных отзывов. Сделать возможность скрывания из результата отрицательных, нейтральных и положительных комментариев;

д) анализ групп нахождение в ней целевой аудитории. Загрузка групп осуществляется из ранее созданной задачи поиска групп, файла или через ввод группы в текстовое окно;

е) поиск сообщений пользователей в обсуждениях с фильтром по:

- ключевым словам;
- дате сообщений, которые нужно собирать;
- количеству лайков.

ё) сбор участников сообществ с фильтром по:

- пересечению участников.

ж) возможность рассылки сообщений от имени сообщества по тем пользователям, которые разрешили сообществу отправлять сообщения.

з) возможность сбора людей, у которых открыты личные сообщений для выбранного сообщества.

и) возможность сбора людей за определённый период из личных сообщений сообщества, которые оставили там свой номер.

4. Для всех задач если это возможно реализовать три варианта загрузки изначальных данных:

- текстовое поле с ссылками;
- загрузка из файла;
- загрузка из ранее созданных задач.

5. Обеспечить возможность получения сервисных ключей пользователей для работы с API VK.

6. Обеспечить возможность получения сервисных ключей сообществ для работы с API VK.

7. Наличие системы администрирования сервиса.

8. Наличие обучающих материалов.

2.1.2 ТРЕБОВАНИЯ К ФУНКЦИОНАЛУ СИСТЕМЫ ЗАДАЧ

1. На каждой странице с задачей обеспечить показ выполненных задач.

2. По нажатию на кнопку рядом с задачей обеспечить вывод результата задачи в форме таблицы.

3. Для каждой выполненной задачи помимо названия задачи должна быть выведена следующая информация о задаче:

- дата создания задачи;
- количество элементов, находящихся в задачи (пример: количество групп).

4. Обеспечить возможность сохранение выполненной задачи в файл в разных вариациях, например id или ссылки.

2.1.3 ТРЕБОВАНИЯ К ФУНКЦИОНАЛУ СИСТЕМЫ АДМИНИСТРИРОВАНИЯ

1. Администрирование пользователей:

- редактирование пользователей;
- удаление пользователей;
- получение информации о пользователи.

2. Редактирование пользователем своего профиля:

- смена email;
- смена пароля;
- добавление или удаления доступа к группам для сервиса.

3. Автоматическое удаление устаревших задач, если они не помечены пользователем как важные.

2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

2.2.1 ТРЕБОВАНИЯ К ПОЛЬЗОВАТЕЛЯМ

Каждый пользователь, работающий в системе, обязан:

- зарегистрироваться в системе и заполнить следующие обязательные поля в регистрационной форме:
 - имя пользователя (необязательно настоящее);

- e-mail пользователя;
- пароль для входа в систему;
- повторный пароль для входа в систему;
- авторизоваться через социальные сети и получить у них следующие обязательные поля:
 - id пользователя;
 - имя и фамилия пользователя;
 - ключ доступа пользователя;

Каждый пользователь обязан получить регистрационные данные через разрабатываемую систему или социальную сеть, можно использовать любой из вариантов. Пользователь должен иметь возможность воспользоваться любым функционалом системы.

2.2.2 ТРЕБОВАНИЯ К СИСТЕМЕ БЕЗОПАСНОСТИ

Система должна обладать следующими средствами обеспечения безопасности:

- пользователь при входе в систему должен указывать логин и пароль, если это авторизация не через социальные сети;
- пользователь при регистрации в системе должен вести пароль два раза (во избежание ошибки);
- пароль должен храниться в зашифрованном виде;

2.2.3 ТРЕБОВАНИЯ К СИСТЕМЕ НАВИГАЦИИ

Система должна обладать следующими навигационными ссылками на каждой странице:

- главную страницу;
- поиск групп;
- сбор постов;
- сбор комментариев;
- сбор участников;

- сбор обсуждений;
- анализ комментариев;
- анализ групп;
- сообщества сообщения;
- сообщества открыты личные;
- сообщества сбор лидов из личных сообщений;
- подключить группу;
- сменить e-mail;
- сменить пароль.

2.2.4 ТРЕБОВАНИЯ К СИСТЕМЕ УВЕДОМЛЕНИЙ

Сервис должен уведомлять пользователя сообщением о выполненной задаче.

2.2.5 ТРЕБОВАНИЯ К ЛИНГВИСТИЧЕСКОМУ ОБЕСПЕЧЕНИЮ

Сервис должен быть доступен на русском языке.

3. ПРОЕКТИРОВАНИЕ

3.1. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ

Для разработки веб-приложения использовался фреймворк Django, который использует архитектуру MVC (Model-View-Controller). Особенность данной архитектуры в том, что она выделяет фрагменты, которые занимаются решением определённых задач. Первый фрагмент работает с данными приложения, второй работает с внешним видом, третий контролирует работу приложения. Часто специалисты Django используют другой термин, заменяя controller на template Model-View-Template.

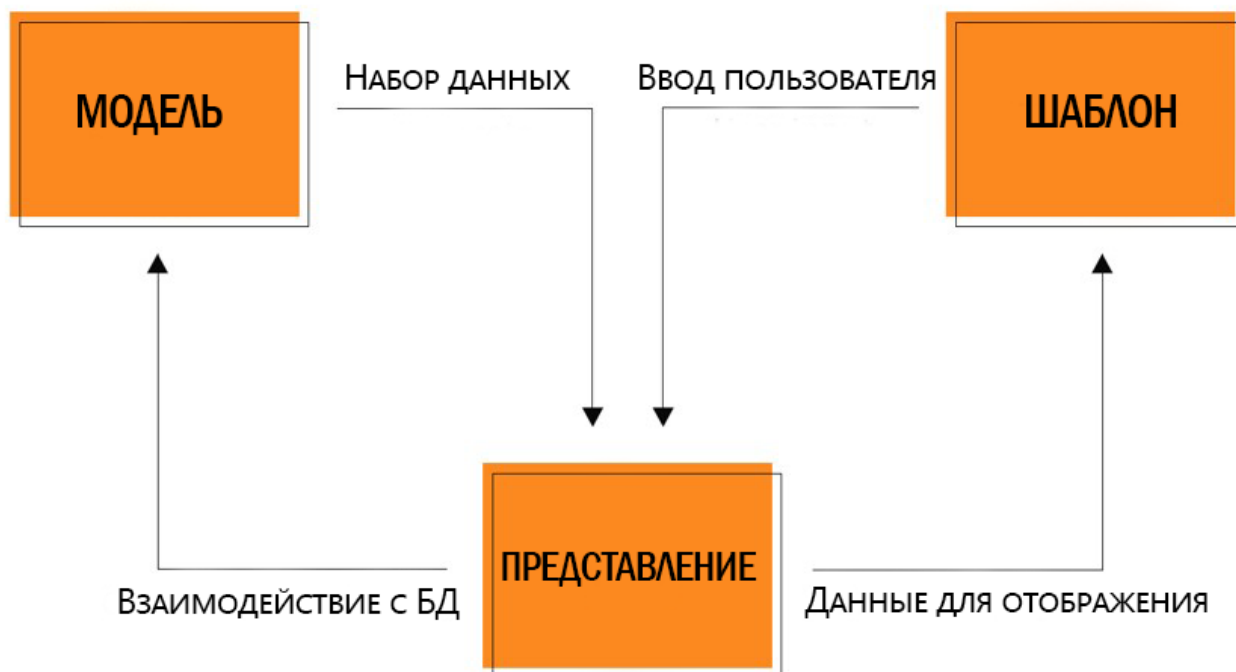


Рисунок 3.1 – Схема архитектуры MVC

В документации по Django указано, что model это «источник информации о данных, в которых содержатся ключевые поля и поведение данных» [19]. Модель занимается только данными и не знает ничего о других уровнях. Взаимодействие между ними происходит при помощи API. View занимается тем, что получает HTTP запросы и отправляет ответы на данные запросы. То есть View принимает данные от Model и передаёт их в нужном виде в template. Template состоит из файлов HTML, с их помощью отображаются данные. Шаблоны могут быть как статистические так и динамические.

3.2. АРХИТЕКТУРА НЕЙРОНОЙ СЕТИ

В качестве архитектуры была выбрана сверточная нейронная сети. Она работает с матрицами, на которые накладывается n количество фильтров, которые перемещаются по матрице. В сверточных нейронных сетях текст организуется в матрицу, где каждая строка представляется вектором, слова или

символа. Сверточные слои “сканируют” текст, как изображение, разбивая его на элементы, и определяя, соответствует ли каждый элемент метки или нет.

На рисунке 3.2 показано, как сверточный фильтр перемещается по предложению. Он поэлементно перемножает содержимое окна на содержимое фильтра, результат суммируется и записывается в матрицу результата. Это этап “объединения”, уменьшающий размерность слов и сохраняющий лишь показатель вероятности принадлежности их к конкретной метке.

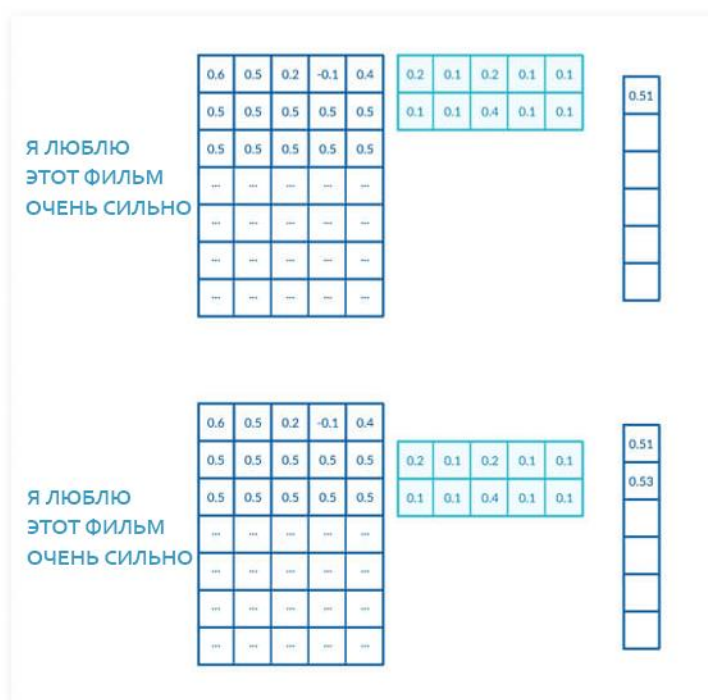


Рисунок 3.2 – Пример работы фильтра

На заключительном этапе эти оценки являются входными данными для полносвязного нейронного слоя. Нейроны каждого предыдущего слоя связаны с одним нейроном полносвязного слоя. Каждый нейрон получает веса, которые определяют приоритет наиболее подходящей метки – например, «позитивное настроение» или «негативное настроение». Последний выходной слой связан со всеми нейронами полносвязного слоя. От количества нейронов

выходного слоя зависит количество классификаций. У нас их 2 положительная и отрицательная классификация. Но мы можем упростить вычисления и сократить количество нейронов до одного используя в качестве активации сигмоидальную функцию изображённую на рисунке 3.3, где 0 - отрицательный комментарий, а 1 положительный.

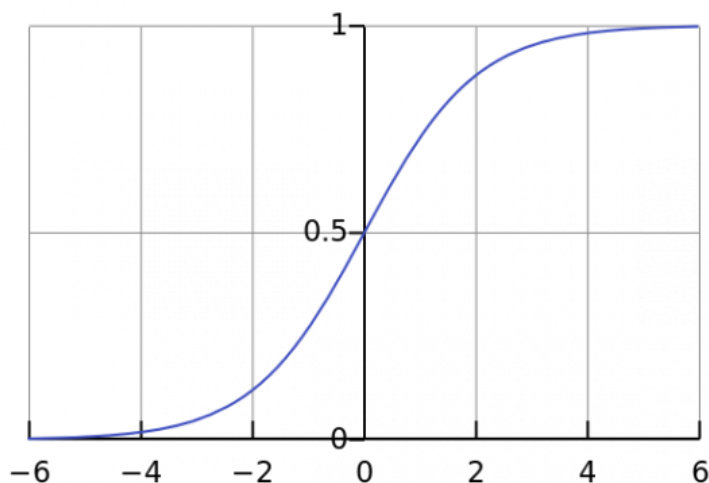


Рисунок 3.3 – Сигмоидальная функция

3.3. ОПИСАНИЕ ДАННЫХ

В разрабатываемой базе данных хранятся данные, которые пользователь вводит при регистрации или авторизации через социальные сети. Для аналитики самых популярных поисковых тематик хранятся данные о созданных задачах по сбору групп и постов. В таблице 3.1 описаны поля таблицы “user”.

Таблица 3.1 – описание полей таблицы “user”

Название поля	Назначение
username	Имя пользователя (никнейм)
first_name	Имя (при авторизации через соц. сети)
last_name	Фамилия (при авторизации через соц. сети)

Продолжение таблицы 3.1

Название поля	Назначение
email	Емаил

В таблице 3.2 описаны поля таблицы “viewsallmodel”.

Таблица 3.2 – описание полей таблицы “viewsallmodel”

Название поля	Назначение
user_name	Имя пользователя (никнейм). Связь с таблицей “user”
name_task	Название задачи
type_task	Тип задачи
num_elemnts	Количество элементов в задачи
task_date	Дата создания задачи

В таблице 3.3 описаны поля таблицы “parsingpost”.

Таблица 3.3 – описание полей таблицы “parsingpost”

Название поля	Назначение
user_name	Имя пользователя (никнейм). Связь с таблицей “user”
group	Название группы
start_date	От какой даты начинать сбор постов
end_date	До какой даты собирать посты
task_name	Название задачи
task_date	Дата создания задачи

В таблице 3.4 описаны поля таблицы “parsinggroup”.

Таблица 3.4 – описание полей таблицы “parsinggroup”

Название поля	Назначение
user_name	Имя пользователя (никнейм). Связь с таблицей “user”
group	Название группы
number_start	От какого количества участников собирать
number_end	До какого количества участников собирать
type_group	Тип сообщества
sort_group	Вариант сортировки сообществ
country_group	Страна указанная в сообществе
city_group	Город указанный в сообществе
name	Название задачи
task_date	Дата создания задачи

В таблице 3.5 описаны поля таблицы “groupinfo”.

Таблица 3.5 – описание полей таблицы “groupinfo”

Название поля	Назначение
user_name	Имя пользователя (никнейм). Связь с таблицей “user”
group_name	Название группы
img_name	Ссылка на аватар группы
id_group	ID группы
token_group	Токен группы

На рисунке 3.4 показана схема базы данных веб-приложения.

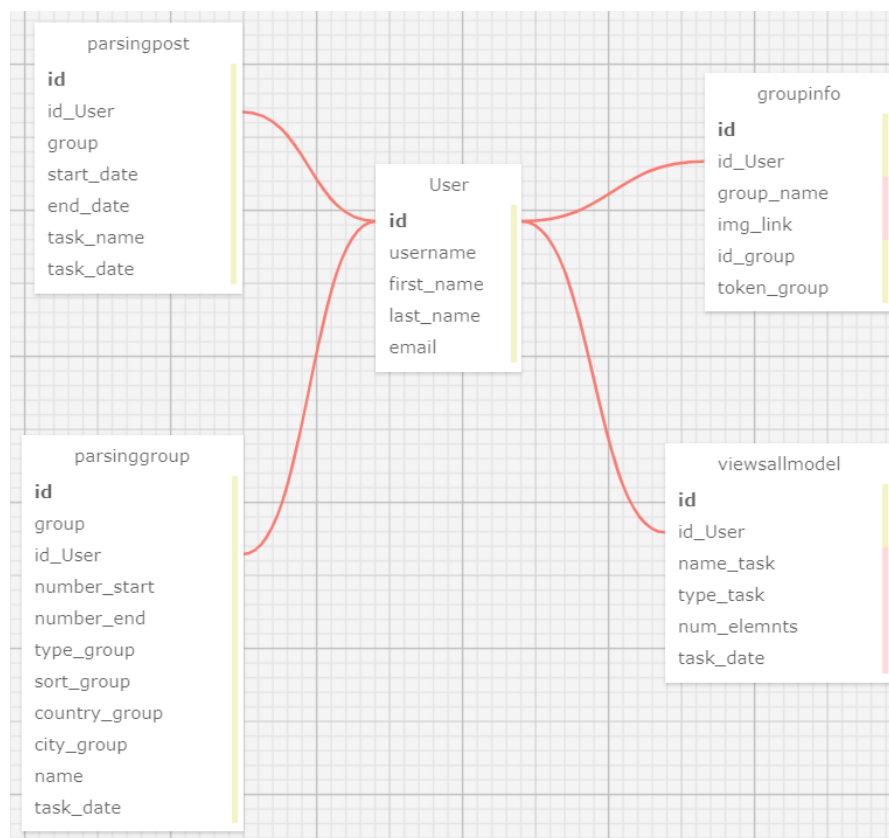


Рисунок 3.4 – Схема базы данных веб-приложения

4. РЕАЛИЗАЦИЯ

4.1. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ

Для начала работы на сайте пользователю необходимо пройти процедуру регистрации, а после авторизации на сайте. Так же у него есть возможность помимо регистрации на сайте авторизоваться через социальную сеть ВКонтакте.

На рисунках 4.1 представлена регистрация в веб-приложение.

Регистрация аккаунта

Имя пользователя

E-mail адрес

Пароль

Пароль (еще раз)

Регистрация

The image shows a registration form on a dark purple background. It consists of four white input fields stacked vertically, labeled 'Имя пользователя', 'E-mail адрес', 'Пароль', and 'Пароль (еще раз)'. Below these fields is a dark grey button with the text 'Регистрация'.

Рисунок 4.1 – Регистрация аккаунта

На рисунках 4.2, 4.3 представлена авторизация в веб-приложение стандартным способом и через социальные сети. При авторизации через ВКонтакте, если человек уже авторизован в браузере он просто зайдет на сайт, если нет ему потребуется ввести логин и пароль.

Авторизация через Email/Пароль [Забыли пароль?](#)

Ваш Email

Ваш пароль

Войти

Создать аккаунт

Вы проходите процедуру авторизации в сервисе "SUSU".
Для авторизации/регистрации могут быть использованы аккаунты в социальных сетях.

Авторизация через ВКонтакте

The image shows a login/registration page on a dark purple background. On the left, there is a section titled 'Авторизация через Email/Пароль' with a link 'Забыли пароль?'. Below this are two white input fields for 'Ваш Email' and 'Ваш пароль'. At the bottom of this section are two buttons: 'Войти' (dark grey) and 'Создать аккаунт' (light grey). On the right, there is a text block explaining the service and a button 'Авторизация через ВКонтакте'.

Рисунок 4.2 – Стандартная авторизация

Рисунок 4.3 – Авторизация через ВКонтакте

Рисунок 4.3 – Авторизация через ВКонтакте

Система удаления, добавления, подтверждения почту представлена на рисунке 4.4. У системы есть возможность привязки нескольких почт, выбора основной из них. Также, при авторизации через социальные сети, если у человека указан email и он совпадает с email'ом из системы, тогда аккаунты соединяются.

Рисунок 4.4 – Система работы с почтой

Рисунок 4.4 – Система работы с почтой

На рисунке 4.5 представлена форма смены пароля.

Рисунок 4.5 – Форма смены пароля

На рисунке 4.6 представлена система работы с группами. Для добавления группы нужно указать ID группы. Все группы пользователя, к которым в данный момент есть доступ у сервиса, отображены в блоке “Авторизованные группы”. У ВКонтакте отсутствует возможность удаления ключа доступа через API, поэтому данной функции в сервисе нет, но пользователь всегда может удалить его через сообщество ВКонтакте.

Авторизованные группы

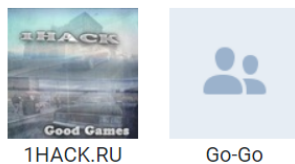


Рисунок 4.6 – Система добавления групп

На рисунках 4.7 и 4.8 представлена система получения Токена пользователя для работы с методами VK API. Для безопасности система запрашивает ключ на 24 часа, по истечению этого времени пользователю

нужно получить новый ключ. Чтобы получить новый токен достаточно нажать кнопку “Обновить подключение” и если он авторизован ВКонтакте, то сервис автоматически получит его. Данное условие работает, если пользователи авторизуется на сайте не с помощью социальных сетей, иначе токен получается автоматически при авторизации.



Рисунок 4.7 – У токена истек срок действия

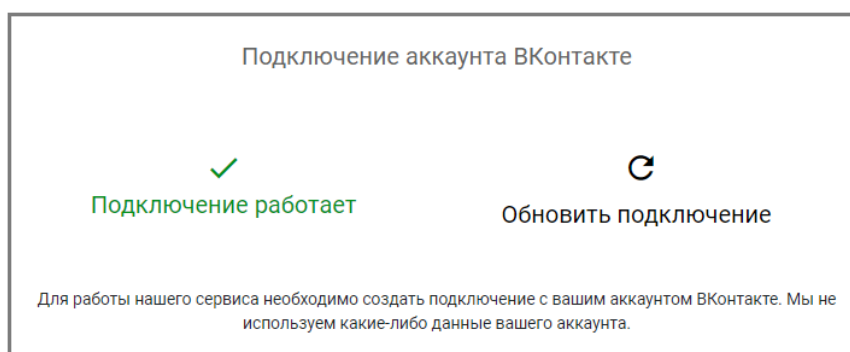


Рисунок 4.8 – Токен находится в рабочем состоянии

Теперь рассмотрим то, как выглядит полная страница сайта. На ней присутствует навигационное меню по задачам, которые возможно создать, навигация по действиям с профилем, сама форма создания задачи, а также список уже выполненных задач. Полная страница представлена на рисунке 4.9.

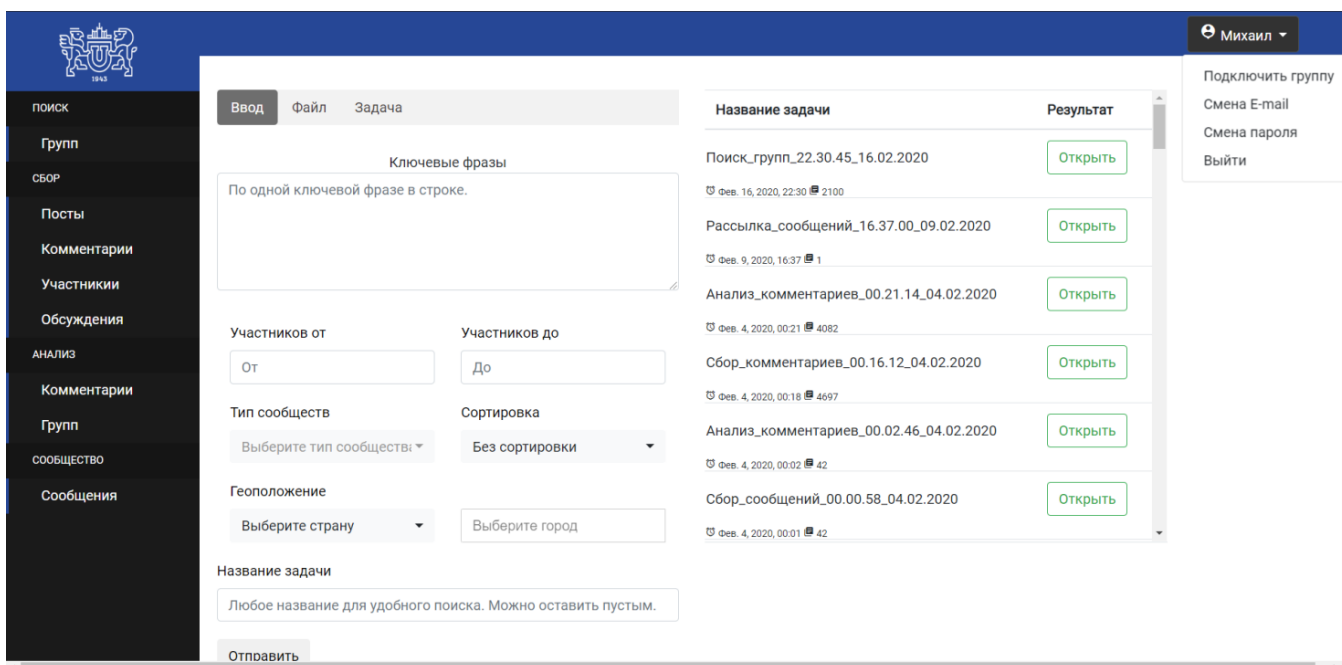


Рисунок 4.9 – Общий вид страницы сайта

Рассмотрим формы задач, которые можно создать.

1. Поиск групп

На рисунке 4.9 представлена форма создания задачи поиска групп. В ней есть возможность ввести ключевые слова, которые должны стоять в названии группы, минимальное и максимальное количество участников, тип сообщества (возможно несколько), вариант сортировки результата, геоположение. Так как сервис создаётся для российского сегмента, то количество стран ограничено ближайшими странами, название города предлагается автоматически по первым буквам для этого используется база городов Вконтакте. Пример работы приведен на рисунке 4.10. Исходный код функций поиска групп приведен в листинге Б.1 приложения Б.

Геоположение
Россия

Название задачи
Любое название для удобного поиска

Отправить

Че

- Челябинск, Челябинская область
- Чебоксары, Чувашская
- Набережные Челны, Татарстан
- Череповец, Вологодская область
- Черкесск, Карачаево-Черкесская
- Чехов, Московская область
- Кирово-Чепецк, Кировская область

Рисунок 4.10 – Пример работы автозаполнения

2. Сбор постов

В форме сбора постов указываются ссылки на сообщества, от какого до какого числа должна быть дата создания поста и название задачи. Если название будет не указано, то система присвоит его в соответствие с названием задачи. Помимо обычного ввода у пользователя есть возможность загрузить файл с ссылками на группы, а также взять данные из ранее созданной задачи по поиску групп. Реализация представлена на рисунках 4.11, 4.12, 4.13, 4.14. Исходный код функций сбора постов приведен в листинге А.1 приложения А.

Ввод **Файл** Задача

Сообщества Вконтакте

По одной ссылке в строке. Сообщества должны быть указаны в виде ссылки или текстового id(club423)

От

От

До

До

Название задачи

Название задачи

Отправить

Рисунок 4.11 – Форма создания задачи

Ввод **Файл** Задача

Выберите файл | Файл не выбран

Период, за который собирать посты

Любое название для удобного поиска. Можно оставить пустым.

Отправить

Рисунок 4.12 – Форма создания задачи с загрузкой информации из файла

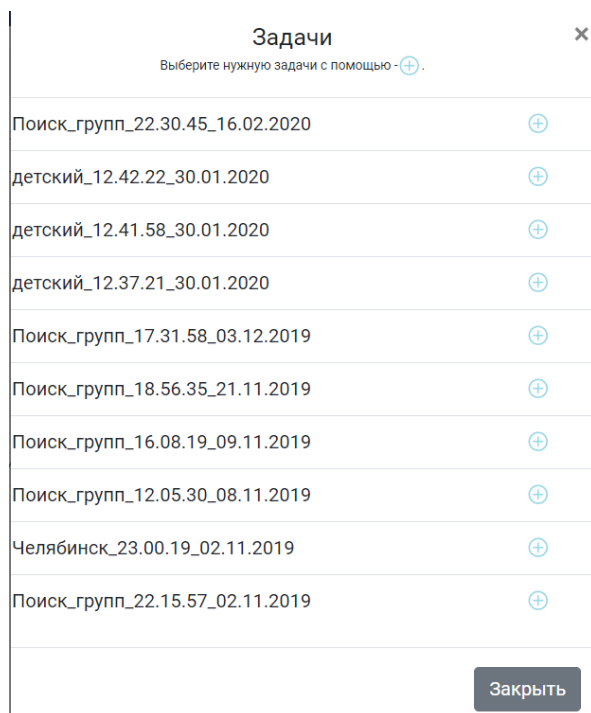


Рисунок 4.13 – Выбор задачи, из которой требуется взять данные

На рисунке 4.13 представлены все ранее выполненные задачи у которых данные подходят для выполнения задачи поиска постов.

Рисунок 4.14 – Форма создания задачи с загрузкой информации из другой задачи

3. Сбор комментариев

В форме сбора комментариев ввод данных так же можно осуществить, введя ссылки на группы в textarea, загрузив из файла или другой задачи. Помимо этого, есть возможность оставить комментарии только с определённой ключевой фразой. Необходимо указать с какой по какую дату собирать посты, с которых будут собраны комментарии. Фильтр по количеству лайков на комментарии и ограниченнее на длину комментария. Форма представлена на рисунке 4.15.

Ввод Файл Задача

Сообщества Вконтакте

По одной ссылке в строке. Сообщества должны быть указаны в виде ссылки или текстового id(club423)

Ключевые фразы

По одной ключевой фразе в строке

Период, за который собирать ком.

Количество лайков

От До

Максимальная длина комментария

До

Название задачи

Любое название для удобного поиска. Можно оставить пустым.

Рисунок 4.15 – Форма сбора комментариев

4. Форма сбора участников

Форма сбора участников включает в себя так же 3 варианта загрузки информации, а также возможность выбора в каком количестве сообществ должен состоять человек, чтобы попасть в финальный результат задачи.

Реализация представлена на рисунке 4.16. Исходный код функций сбора участников приведен в листинге Ж.1 приложения Ж.

The screenshot shows a web form with a top navigation bar containing 'Ввод', 'Файл', and 'Задача'. The main section is titled 'Сообщества Вконтакте' and contains a text input field with the instruction: 'По одной ссылке в строке. Сообщества должны быть указаны в виде ссылки или текстового id(club423)'. Below this is a section 'Состоят в группах' with two radio buttons: 'От(Включительно)' and 'До(Включительно)'. The next section is 'Название задачи' with a text input field containing the placeholder: 'Любое название для удобного поиска. Можно оставить пустым.'. At the bottom is an 'Отправить' button.

Рисунок 4.16 – Форма сбора участников

5. Форма сбора обсуждений

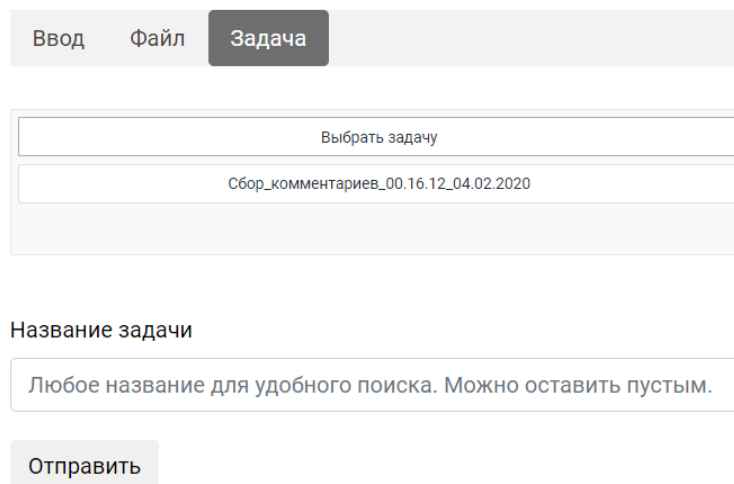
Форма сбора сообщений с обсуждений включает в себя 3 варианта загрузки информации. Фильтр, по ключевым словам, в сообщениях. Период, за который собирать сообщения. Фильтр по количеству лайков на сообщение. Реализация показана на рисунке 4.17. Исходный код функций сбора обсуждений приведен в листинге 3.1 приложения 3.

The screenshot shows a web form with a top navigation bar containing 'Ввод', 'Файл', and 'Задача'. The main section is titled 'Сообщества Вконтакте' and contains a text input field with the instruction: 'По одной ссылке в строке. Сообщества должны быть указаны в виде ссылки или текстового id(club423)'. Below this is a section 'Ключевые фразы в сообщениях' with a text input field containing the placeholder: 'По одной ключевой фразе в строке'. The next section is 'Период за который собирать сооб.' with two date pickers. Below that is a section 'Количество лайков' with two radio buttons: 'От' and 'До'. The next section is 'Название задачи' with a text input field containing the placeholder: 'Любое название для удобного поиска. Можно оставить пустым.'. At the bottom is an 'Отправить' button.

Рисунок 4.17 – Форма сбора обсуждений

6. Анализ тональности комментариев

Для анализа комментариев задачу можно загрузить только из ранее созданных задач по сбору комментариев и обсуждений. Реализация показана на рисунке 4.18. Исходный код функций анализа тональности комментариев приведен в листинге Г.1 приложения Г.



Ввод Файл **Задача**

Выбрать задачу

Сбор_комментариев_00.16.12_04.02.2020

Название задачи

Любое название для удобного поиска. Можно оставить пустым.

Отправить

Рисунок 4.18 – Форма анализа комментариев и обсуждений

7. Анализ целевой аудитории групп

Анализ групп представляет собой проверку групп на то, сколько аудитории подходящей по параметрам в ней состоит. Параметры, которые можно задать: возраст, пол, семейное положение, геоположение. Реализация формы показана на рисунке 4.19. Исходный код функций анализа целевой аудитории групп приведен в листинге Д.1 приложения Д.

Выберите группу

1HACK.RU

Введите текст сообщения

Введите текст сообщения

Добавьте нужные кнопки(Нажмите на кнопку, чтобы выбрать цвет)

Да

Нет

Не знаю

Тест

Выберите файл с ID пользователей, которым отправится сообщение

Выберите файл | Файл не выбран

Название задачи

Любое название для удобного поиска. Можно оставить пустым.

Отправить

Рисунок 4.20 – Форма рассылки сообщений

9. Сообщество открытые личные сообщения

Данная задача предназначена для сбора людей, которым сообщества может отправлять личные сообщения. Данный инструмент будет удобен для задачи рассылок сообщений. Для сбора людей достаточно выбрать нужные сообщества из списка авторизованных и дать название задачи. На рисунке 4.21 приведена реализация формы.

Выберите сообщество для сбора пользователей

1HACK.RU

Название задачи

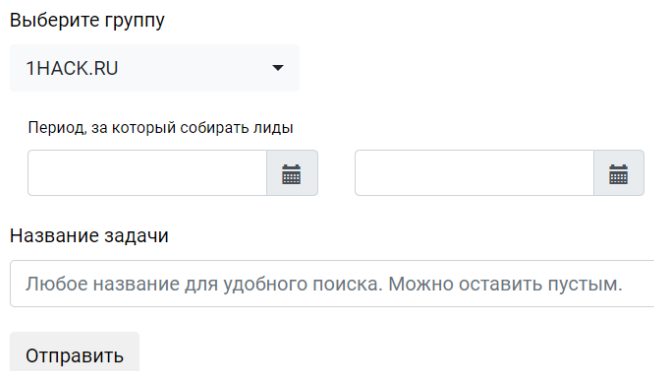
Любое название для удобного поиска. Можно оставить пустым.

Отправить

Рисунок 4.21 – Форма сбора людей, открывших личные сообщения

10. Сбор лидов из личных сообщений

Данная задача предназначена для сбора людей, которые оставили свой номер телефона в личных сообщениях в группе. Она будет полезная для ретаргетинга на данных людей или обзвона после массовой рассылки, когда трудно найти номера среди большого количества сообщений. На рисунке 4.22 приведена реализация формы.



Выберите группу

1НАСК.RU

Период, за который собирать лиды

Название задачи

Любое название для удобного поиска. Можно оставить пустым.

Отправить

Рисунок 4.22 – Сбор лидов из личных сообщений

Перейдем к системе задач. Задачи пользователя находятся справа от формы задач на каждой странице кроме главной и страниц настройки профиля. Они загружаются автоматически из базы данных, список содержит все задачи, когда-либо созданные пользователем. В дальнейшем планируется сделать автоматическое удаление задач через 24 часа, возможность выбора задачи, как важной, и удаления задачи самим пользователем. Из всех задач есть возможность скачать результаты в нескольких исполнениях. На рисунке 4.23 приведена реализация блока задач. Исходный код функций сбора лидов из личных сообщений приведен в листинге E.1 приложения E.

Название задачи	Результат
Поиск_групп_22.30.45_16.02.2020 🕒 Фев. 16, 2020, 22:30 📄 2100	Открыть
Рассылка_сообщений_16.37.00_09.02.2020 🕒 Фев. 9, 2020, 16:37 📄 1	Открыть
Анализ_комментариев_00.21.14_04.02.2020 🕒 Фев. 4, 2020, 00:21 📄 4082	Открыть
Сбор_комментариев_00.16.12_04.02.2020 🕒 Фев. 4, 2020, 00:18 📄 4697	Открыть
Анализ_комментариев_00.02.46_04.02.2020 🕒 Фев. 4, 2020, 00:02 📄 42	Открыть
Сбор_сообщений_00.00.58_04.02.2020 🕒 Фев. 4, 2020, 00:01 📄 42	Открыть











Рисунок 4.23 – Блок задач

Для каждой задачи существует свой шаблон вывода результатов. Рассмотрим все существующие варианты.

1. Поиск групп.

Результаты задачи ×

Скачать
ID LINK CSV

	Большой Челябинск Публичная страница	👤33800
	Интересный Челябинск Публичная страница	👤37498
	Спортивный Челябинск Публичная страница	👤39322
	Челябинский урбанист Публичная страница	👤47545
	Инцидент Челябинск Публичная страница	👤47898
	Автошкола ДОСААФ Челябинск Публичная страница	👤8608
	Полиция Челябинской области Публичная страница	👤13224
	TELEFAKT.RU Новости Челябинска Публичная страница	👤26597
	ЧелГУ. Челябинский государственный университет Публичная страница	👤31354
	Челябинская область Публичная страница	👤22259

Prey 1 2 ... 209 210 Next

Заккрыть

Рисунок 4.24 – Шаблон вывода результата поиска групп

Как видно из рисунка 4.24 результат поиска групп содержит название группы, тип группы, фотографию группы и количество её участников. Так же есть возможность скачать результаты в виде id групп, ссылок групп и csv файл со всеми данными.

2. Сбор постов.



Рисунок 4.25 – Шаблон вывода результатов сбора постов

Как видно из рисунка 4.25 вывод представляет собой посты, адаптированные под формат постов из Вконтакте. Форма содержит название группы, её фото, возможность перейти в паблик и в пост, дату поста, текст, изображение, количество лайков, комментариев, репостов и просмотров. Если

изображения нет, оно заменяется изображением о том, что его не было. Есть возможность скачать ссылки постов, сообществ или всю информацию.

3. Сбор комментариев.

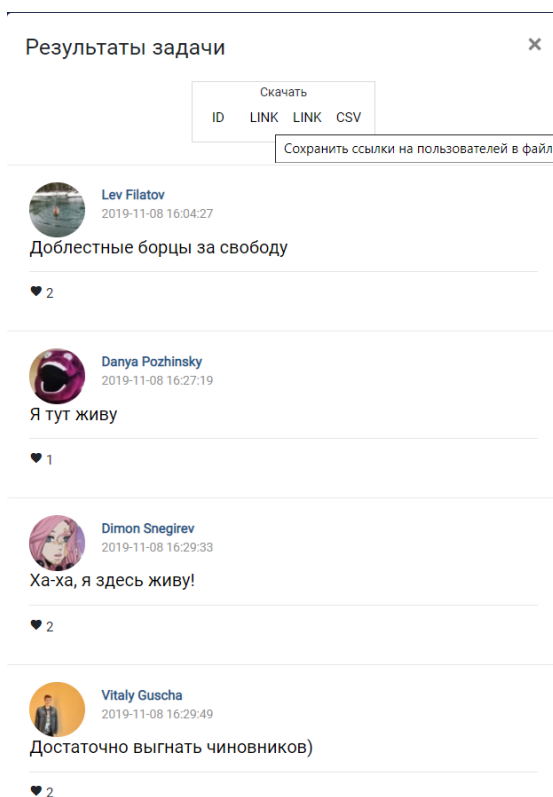












Рисунок 4.26 – Шаблон вывода сбора комментариев

Как видно из рисунка 4.26 вывод комментариев происходит по одному, подкомментарии не учитываются. В результате шаблон содержит информацию о дате комментария, самом комментарии, фото человека, возможность перейти к посту, под которым был оставлен комментарий, перейти на страницу человека. Можно скачать ссылки на людей и посты.

4. Сбор участников.

Как видно из рисунка 4.27 вывод шаблона содержит информацию об имени и фамилии, половой принадлежности, дате рождения, геопозиции пользователя и фотографии. Данные можно скачать, как ID пользователей, так и

ссылки. Часть или вся информация, которая неизвестна о городе, стране, дате рождения закрашена. Исходный код функций сбора участников приведен в приложение Ж.

Скачать				
ID	LINK	CSV		
 Анна Лозинская 2847531		Жен. 28.6.1987 33	Россия, Ульяновск	
 Венера Алексеева 1893388		Жен. 23.7.1984 36	Россия, Ульяновск	
 Татьяна Доркаева 1795893		Жен. 31.1	Россия, Ульяновск	
 Елена Ли 1747114		Жен. 6.6.2000 20	Россия, Ульяновск	
 Юлия Филатова 1672190		Жен. 8.7.1981 39	Россия, Ульяновск	
 Дина Камалова 1362444		Жен.	Россия, Ульяновск	
 Елена Усачёва 1281310		Жен. 11.1	Россия, Ульяновск	
 Эльмира Замалетдинова 1082225		Жен.	Россия, Ульяновск	
 Галина Щедрина 620461		Жен.	Россия, Ульяновск	
 Мариша Никитина 385640		Жен. 19.7	Россия, Ульяновск	

Prev 1 2 ... 261 262 Next

Закреть

Рисунок 4.27 – Шаблон вывода сбора участников

5. Сбор сообщений представлен в таком же виде, как и комментарии.

6. Анализ комментариев.

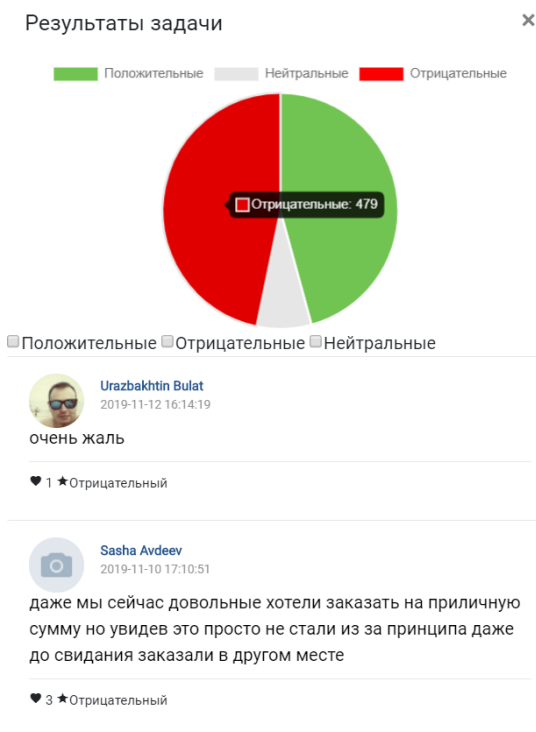


Рисунок 4.28 – Шаблон вывода анализа комментариев

Вывод анализа комментариев содержит диаграмму с информацией по соотношению нейтральных, отрицательных и положительных отзывов. Она представлена на рисунке 4.28. Так же повторяется шаблон по сбору обычных комментариев и добавляется информация об оценке комментария. Есть возможность отфильтровать комментарии по определённой тональности. Исходный код функций анализа комментариев приведен в приложение Г.

7. Анализ групп.

Результаты задачи











		Скачать
ID	LINK	CSV
Сохранить id сообщений в файл		
	KING SIZE roll суши Челябинск	
17249	ЦА:11119	64.46%
	СушиПицца74.рф Доставка Суши и Пиццы Челябинск	
22719	ЦА:15811	69.59%
	Суши ЦУКИЮМИ. Челябинск	
13998	ЦА:9363	66.89%
	Камикадзе - суши, пицца. Доставка Челябинск	
21681	ЦА:0	0.00%
	Суши Фривей Челябинск	
25254	ЦА:16664	65.99%
	Эбису. Суши и выпечка. Теплотех.	
5166	ЦА:3444	66.67%
	Ресторан доставки "BUSHIDO"Челябинск суши, роллы	
5824	ЦА:4005	68.77%
	Доставка суши и пиццы "Жуй - Смакуй", Челябинск	
6777	ЦА:4531	66.86%

Рисунок 4.29 – Шаблон вывода анализа групп

Анализ групп предоставляет список групп с информацией о количестве человек в числовом и процентом соотношении, которые подходят по критериям заданным при создании задачи. Исходный код функций анализа групп приведен в приложение Д.

8. Результатом от рассылки сообщения является текстовое сообщение о количестве отправленных сообщений.

9. Результат задачи сбора людей с открытыми личными сообщениями показан на рисунке 4.30.

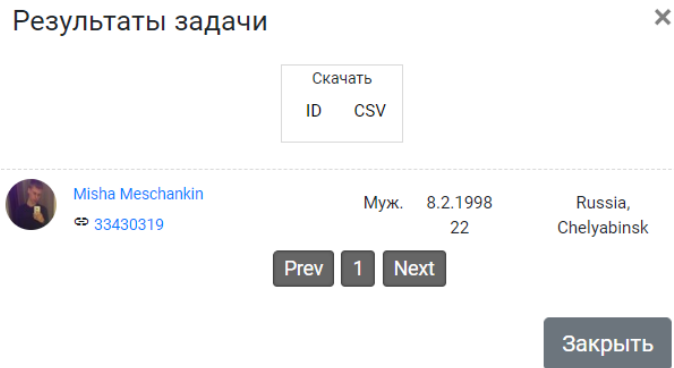


Рисунок 4.30 – Шаблон вывода людей с открытыми личными сообщениями

В результате выполнения задачи выводится фамилия и имя, пол, дата рождения, возраст и геолокация человека, а также ссылка на профиль.

10. Результат задачи сбора лидов из личных сообщений показан на рисунке 4.31.

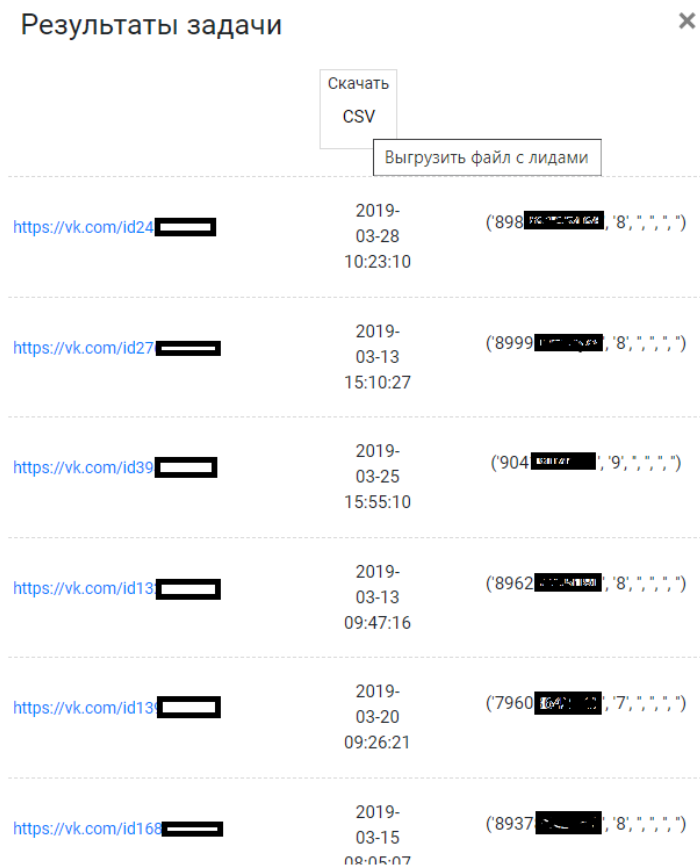


Рисунок 4.31 – Шаблон вывода людей, отправивших свой номер

В результате выполнения задачи выводится ссылка на человека, дата сообщения с номером и сам номер телефона. Исходный код функций сбора лидов из личных сообщений приведен в приложение Е.

4.2. ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ

Обучение будет производиться на базе текстов, собранных Юлией Рубцовой [20]. Тексты заранее размечены на положительные и отрицательные. Тексты собраны из русского язычного Twitter'а. Тексты из twitter были взяты, так как ВКонтакте это не типичная площадка для отзывов, такая как Яндекс маркет [21]. Часто отзывы имеют не типичное представление пример на рисунках 4.33, 4.34, 4.35, 4.36. Так же в социальной сети собрано огромное количество тематик, поэтому узкие обучающие выборки нам не подходят. Для того, чтобы исключить некоторые виды ошибок в текстах и привести их к одному виду перед началом обучения были убраны знаки пунктуации, слова были приведены к нижнему регистру. Так как входным слоем является матрица, для её заполнения все слова были переведены в векторное пространство при помощи инструмента Word2Vec [22]. Он сопоставляет каждому слову вектор, и на выходе мы получаем его координаты. Главная особенность в том, что он анализирует, какие слова встречаются рядом и, если слово встречается рядом n -ое количество раз, то скорее всего они близки по смыслу. Пример работы представлен на рисунке 4.32. На рисунке видно, что близкие по смыслу слова стоят рядом. Для создания нейронной сети используется библиотека Keras [23]. Настройки нейронной сети выбирались исходя из рекомендаций в работе [24].

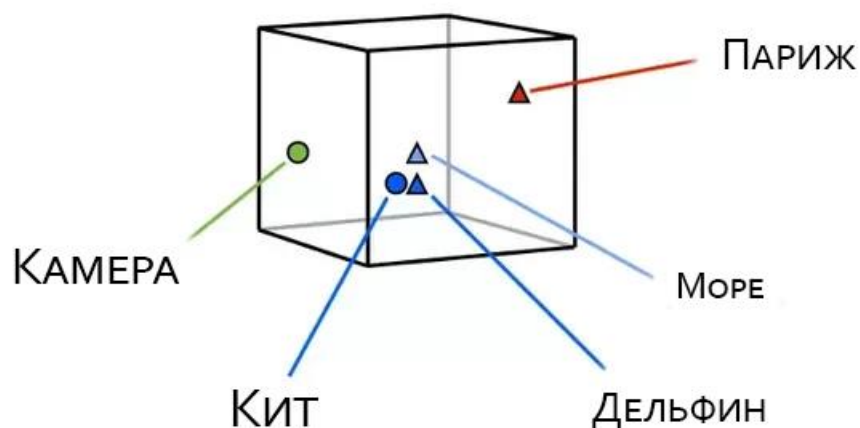


Рисунок 4.32 – Word2Vec пример

4.3. РЕАЛИЗАЦИЯ НЕЙРОННОЙ СЕТИ

После обучение сверточной нейронной сети и проверки её точности на тестовой выборке, её точность составила 78%. Что является вполне неплохим результатом.

Проверим её на практике и убедимся, что это решение является лучшим, чем просто проверять тексты на нахождение в нем слов, которые заранее заготовлены в словаре. Недостатки данного метода в том, что мы можем указать не все слова, которые определяют эмоции человека. На примерах ниже приведены результаты работы сверточной нейронной сети, где 1 – это позитивный комментарий, 0 – это негативный комментарий (Результат работы в самом конце скриншота в первом примере на рисунке 4.33 это $[[0.203066]]$, что означает, что нейронная сеть считает, что он отрицательный). В этих примерах нет ключевых слов, по которым можно было бы явно определить тональность комментария, но нейронная сеть справляется с этой задачей.

1 комментарий: “зачем вы покупаете эти платья это далеко не первый обзор на них мало того что это дешевый не дышащий полиэстр так еще и кошмарные

лекала и нелепый крой который даже самой стройной девочке накидывает килограммы”.

```
In [31]: text = ['зачем вы покупаете эти платья это далеко не первый обзор на них мало того что это дешевый не дышащий полиэстр та  
test_my = get_tokenizer(tokenizer, text)  
< >  
In [32]: predicted = loaded_model.predict(test_my.reshape(1,26))  
print(predicted)  
[[0.39649668]]
```

Рисунок 4.33 – Пример 1

2 комментарий: “заказывала такой дозатор в моем случае попался неудачный масло текло по бутылке и оно не распыляло, а плевалось в итоге выкинула”.

```
In [41]: text = ['заказывала такой дозатор в моем случае попался неудачный масло текло по бутылке и оно не распыляло\  
а плевалось в итоге выкинула']  
test_my = get_tokenizer(tokenizer, text)  
In [42]: predicted = loaded_model.predict(test_my.reshape(1,26))  
print(predicted)  
[[0.4171243]]
```

Рисунок 4.34 – Пример 2

3 комментарий: “Больше пакет напоминает” негативный комментарий про футболку в магазине.

```
In [43]: text = ['Больше пакет напоминает']  
test_my = get_tokenizer(tokenizer, text)  
In [44]: predicted = loaded_model.predict(test_my.reshape(1,26))  
print(predicted)  
[[0.19041498]]
```

Рисунок 4.35 – Пример 3

3 комментарий: “ у меня таких две белая и голубая но через пару стирок вид потеряли не самый лучший материал”.

```
In [95]: text = ['у меня таких две белая и голубая но через пару стирок вид потеряли не самый лучший материал']  
test_my = get_tokenizer(tokenizer, text)  
In [96]: predicted = loaded_model.predict(test_my.reshape(1,26))  
print(predicted)  
[[0.06708711]]
```

Рисунок 4.36 – Пример 4

Как видно из рисунков 4.33, 4.34, 4.35, 4.36 нейронная сеть справляется весьма неплохо, но бывают случаи ошибок. В случае перебора слов ошибок было бы куда больше по сравнению с нашей (выбранной, или обученной?)

нейронной сетью. Проблема заключается в том, что даже человеку иногда трудно оценить тональность комментария.

5. ТЕСТИРОВАНИЕ

5.1. ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ

На данный момент было проведено только альфатестирование. Проведем несколько модульных тестов и покажем работу веб-приложения при выполнении этих тестов.

Тестирование возможности регистрации пользователя.

Проверка введённых пользователем данных при регистрации представлена на рисунке 5.1 и 5.2, 5.3.

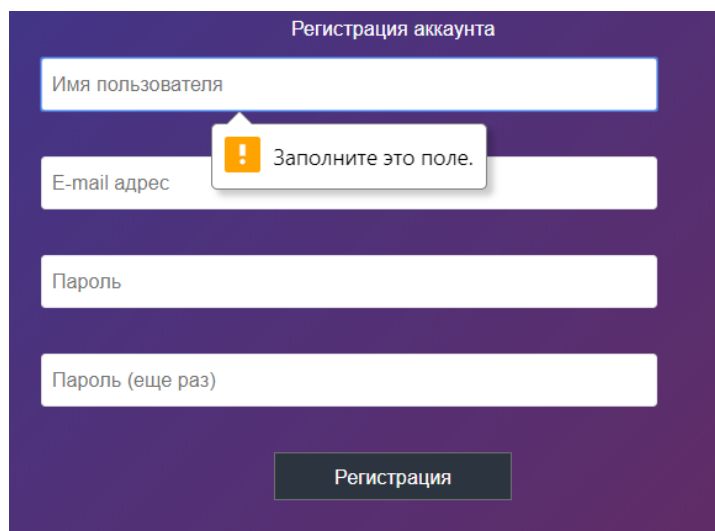


Рисунок 5.1 – Проверка обязательных полей

Регистрация аккаунта

Имя пользователя

E-mail адрес

Пароль

Пароль (еще раз)

- Вы должны ввести одинаковый пароль дважды.

Регистрация

Рисунок 5.2 – Проверка паролей на совпадение

Регистрация аккаунта

Имя пользователя

- Пользователь с таким именем уже существует.

E-mail адрес

Пароль

- Введённый пароль состоит только из цифр.

Пароль (еще раз)

Регистрация

Рисунок 5.3 – Проверка легкости пароля и дублирования данных

Проверим возможность авторизоваться в сервисе с невалидными данным рисунок 5.4.

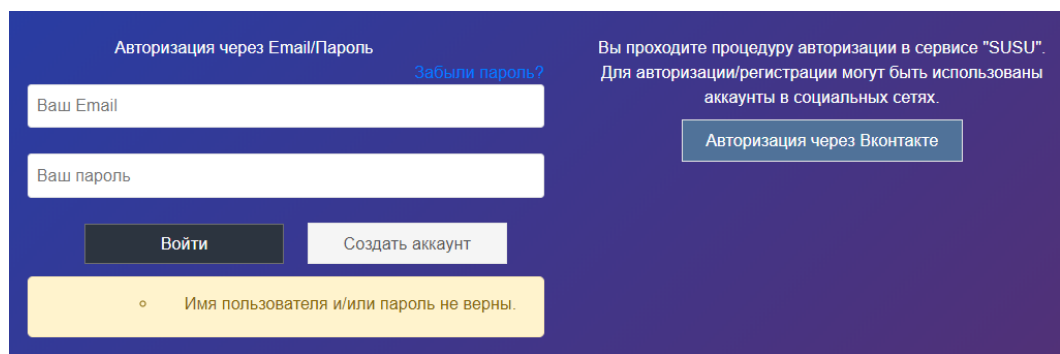
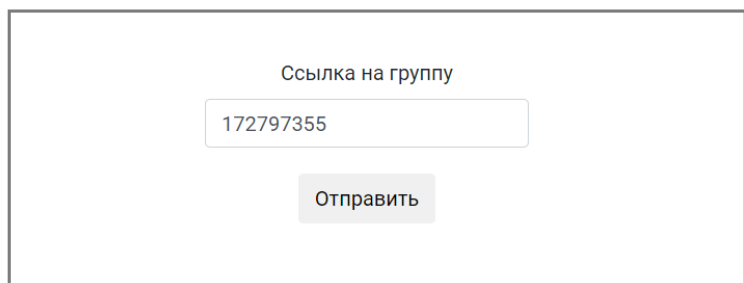


Рисунок 5.4 – Проверка авторизации по не валидным данным

Проверка возможности подключения группы ВКонтакте. Как видно из рисунка 5.5 в данный момент авторизована одна группа, добавим ещё одну.



Авторизованные группы

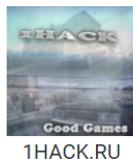


Рисунок 5.5 – Добавление группы

При добавлении группы открывается окно предоставления доступа рисунок 5.6.

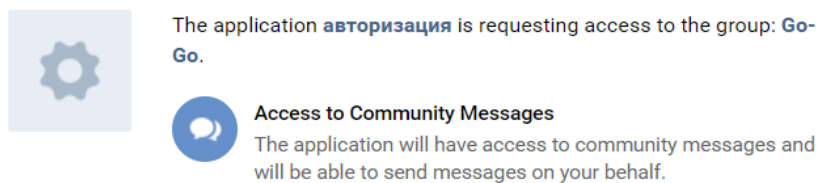


Рисунок 5.6 – Окно подтверждения доступа к группе

Ссылка на группу

Введите ID группы

Отправить

Авторизованные группы

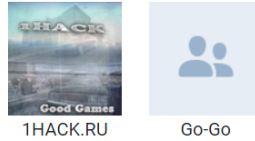


Рисунок 5.7 – Группа авторизована

5008 ***** 3f39 · Show

Access settings: community messages

Created by M. Meschankin, today at 4:47 pm for авторизация application

Delete all application tokens

Рисунок 5.8 – Токен созданный в группе

Как видно из рисунка 5.7 и 5.8 группа успешно добавлена.

Поиск групп Челябинска с фильтром по участникам и геоположению.

Ввод Файл Задача

Ключевые фразы

По одной ключевой фразе в строке.

Участников от Заполните это поле. в до

От До

Тип сообществ Сортировка

Выберите тип сообществ: ▾ Без сортировки ▾

Геоположение

Выберите страну ▾ Выберите город

Название задачи

Любое название для удобного поиска. Можно оставить пустым.

Отправить

Рисунок 5.9 – Проверка обязательных полей

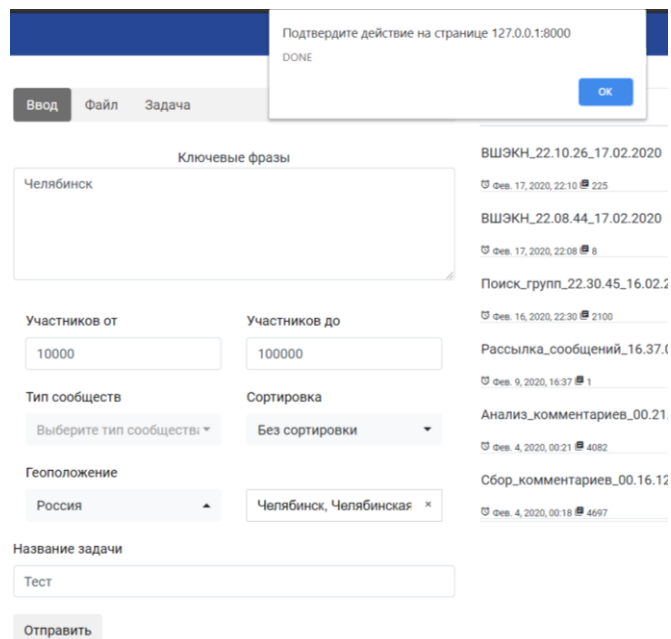


Рисунок 5.10 – Завершение выполнения задачи.

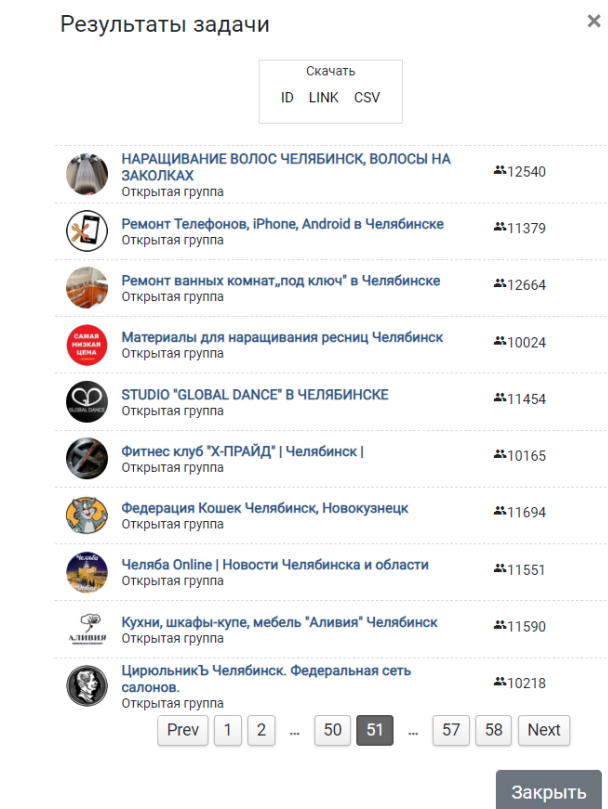


Рисунок 5.11 – Результаты задачи

Как видно по рисунку 5.9 форма содержит обязательные для заполнения поля. Из рисунков 5.10 и 5.11 видно, что задача выполнена корректно и соответствует требуемым параметрам.

Проверка рассылки сообщений. Для этого запишем в текстовый файл два id, первый – человека который разрешил отправку сообщений и второй – человека который не разрешал отправку сообщений.

Выберите группу

1HACK.RU

Введите текст сообщения

Введите текст сообщения

Добавьте нужные кнопки (Заполните это поле, чтобы выбрать цвет)

Текст кнопки

Текст кнопки

Текст кнопки

Текст кнопки

Выберите файл с ID пользователей, которым отправится сообщение

Выберите файл | Файл не выбран

Название задачи

Любое название для удобного поиска. Можно оставить пустым.

Отправить

Рисунок 5.12 – Проверка обязательных полей

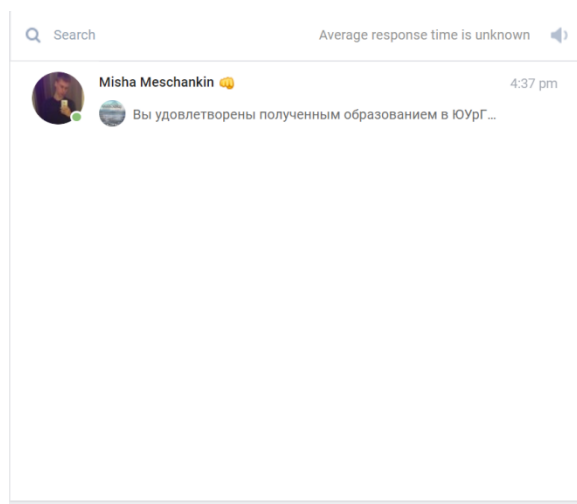


Рисунок 5.13 – Отправленные сообщения

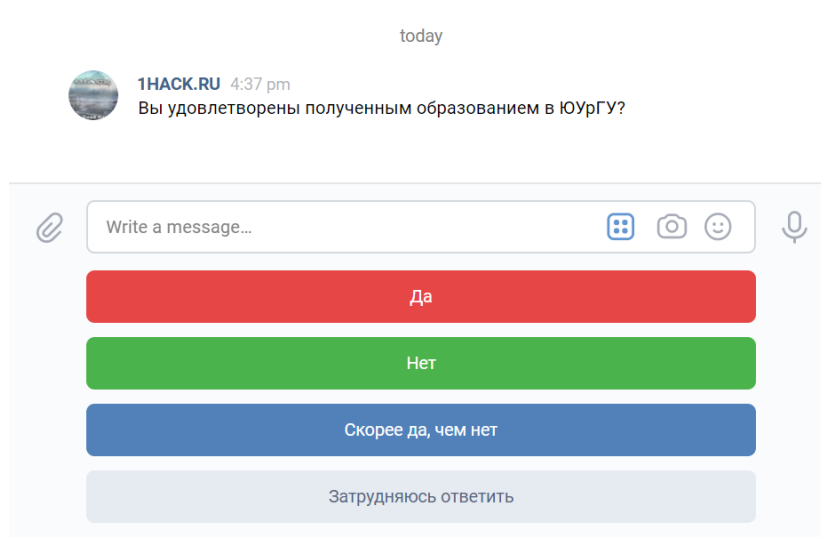


Рисунок 5.14 – Полученное сообщение

Как видно из рисунка 5.12 форма содержит обязательные поля во избежание ошибок. Из рисунка 5.13 видно, что сообщение успешно отправилось 1 человеку, который разрешил сообществу отправлять ему сообщения. Из рисунка 5.14 видно, что сообщение пришло с заданными нами кнопками.

6. ЗАКЛЮЧЕНИЕ

В рамках дипломной работы было спроектировано и реализовано веб-приложение для анализа и работы с целевой аудиторией в социальной сети ВКонтакте.

В веб-приложение были реализованы следующие функции:

1. Система задач.
2. Система администрирования.
3. Поиск сообществ.
4. Сбор постов из сообществ.
5. Сбор комментариев из постов и обсуждений.
6. Сбор участников сообществ.
7. Анализ тональность комментариев с помощью сверточных нейронных сетей.
8. Анализ групп на наличие в ней целевой аудитории по критериям.
9. Рассылка сообщений по пользователям ВКонтакте.
10. Сбор людей, открывших личные сообщения сообществу.
11. Сбор людей, отправивших свой номер телефона сообществу.

В дальнейшем планируется:

- сделать возможность отправлять серию сообщений в разные периоды времени;
- предоставить возможность пользователям отправлять сообщения в определённое время;
- улучшить систему анализа комментариев, проведя тестирование других архитектур нейронных сетей, а также собрав более релевантную базу текстов для обучения нейронной сети;
- добавить возможность автоматического регулирования ставок в рекламном кабинете ВКонтакте;
- добавить систему анализа рекламной компании за определённый период.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Социальные сети в России. – <https://br-analytics.ru/blog/socseti-v-rossii-osen-2018/>. Дата обращения: 24.05.2020.
2. Компьютерная лингвистика. – <https://habr.com/ru/company/abbyu/blog/202306>. Дата обращения: 24.05.2020.
3. User Generated Reviews: Do They Affect Conversions?. – <https://cxl.com/blog/user-generated-reviews/>. Дата обращения: 24.05.2020.
4. Автор сервиса Сегменто Трагет. – <https://vk.com/mrbowie>. Дата обращения: 24.05.2020.
5. Сервис Барков.нет. – <https://vk.barkov.net/about.aspx>. Дата обращения: 24.05.2020.
6. Сервисы для парсинга аудиторий. – <https://blog.ingate.ru/detail/5-servisov-dlya-parsinga-auditorii-v-sotssetyakh/>. Дата обращения: 24.05.2020.
7. Индекс ТЮБЕ. – <https://www.tiobe.com/tiobe-index/>. Дата обращения: 24.05.2020.
8. ТОП-10 быстрых языков программирования. – <https://zen.yandex.ru/media/skillbox/top10-bystryh-iazykov-programmirovaniia-5c613edb1174ac00ad9fda86>. Дата обращения: 24.05.2020.
9. Top 10 Python Web Frameworks. – <https://hackernoon.com/top-10-python-web-frameworks-to-learn-in-2018-b2ebab969d1a>. Дата обращения: 24.05.2020.
10. Neural networks. – <https://www.sciencedirect.com/topics/neuroscience/neural-networks>. Дата обращения: 24.05.2020.

11. Нейронные сети для начинающих. – <https://habr.com/ru/post/312450/>. Дата обращения: 24.05.2020.
12. Secret Sauce behind the beauty of Deep Learning: Beginners guide to Activation Functions. – <https://towardsdatascience.com/secret-sauce-behind-the-beauty-of-deep-learning-beginners-guide-to-activation-functions-a8e23a57d046>. Дата обращения: 24.05.2020.
13. Multi-Layer Neural Networks with Sigmoid Function Deep Learning for Rookies. – <https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f>. Дата обращения: 24.05.2020.
14. Нейронные сети. Обработка текстов. – <http://tpc.at.ispras.ru/wp-content/uploads/2011/10/lecture61.pdf>. Дата обращения: 24.05.2020.
15. Recurrent Neural Networks. – <https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce>. Дата обращения: 24.05.2020.
16. Dropout effects on Convolutional vs. Recurrent Neural Networks. – <http://www.doganaskan.com/blog/posts/cnnvsrnn.html>. Дата обращения: 24.05.2020.
17. Introduction to Recurrent Neural Network. – <https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/>. Дата обращения: 24.05.2020.
18. An Empirical Evaluation of Generic Convolutional and Recurrent Networks. – <https://arxiv.org/abs/1803.01271>. Дата обращения: 24.05.2020.
19. Документация Django. – <https://docs.djangoproject.com/en/3.0/>. Дата обращения: 24.05.2020.
20. Сайт Юлии Рубцовой. – <http://study.mokoron.com/>. Дата обращения: 24.05.2020.

21. Яндекс маркет. – <https://market.yandex.ru/my/reviews>. Дата обращения: 24.05.2020.
22. Word2Vec. – <https://habr.com/ru/post/446530/>. Дата обращения: 24.05.2020.
23. Keras. – <https://keras.io/>. Дата обращения: 24.05.2020.
24. Analysis of Deep Convolutional Neural Network Architectures. – <https://pdfs.semanticscholar.org/6831/bb247c853b433d7b2b9d47780dc8d84e4762.pdf>. Дата обращения: 24.05.2020.