

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт ИнЭО.

Направление подготовки 09.03.01 Информатика и вычислительная техника.

Кафедра Автоматики и компьютерных систем.

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Веб-приложение для поиска данных в социальных сетях по заданной предметной области с использованием латентно–семантического анализа

УДК 004.774:004.455.1:004.4'414

Студент

Группа	ФИО	Подпись	Дата
3–8В2В2	Абдурахимов Ильдар Ринатович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. АиКС	Лунева Е.Е	К.Т.Н		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры Менеджмента	Рахимов Т.Р	К.Э.Н		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Инженер	Маланова Н.В	К.Т.Н		

ДОПУСТИТЬ К ЗАЩИТЕ:

И.о.Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
АиКС	Суходоев М.С.	К.Т.Н		

Код результата	Результат обучения (выпускник должен быть готов)
<i>Профессиональные компетенции</i>	
ПК1	Разрабатывать бизнес–планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием.
ПК2	Осваивать методики использования программных средств для решения практических задач
ПК3	Разрабатывать интерфейсы "человек – электронно–вычислительная машина".
ПК4	Разрабатывать модели компонентов информационных систем, включая модели баз данных.
ПК5	Разрабатывать компоненты программных комплексов и баз данных, использовать современные инструментальные средства и технологии программирования.
ПК6	Обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности.
ПК7	Готовить презентации, научно–технические отчеты по результатам выполненной работы, оформлять результаты исследований в виде статей и докладов на научно–технических конференциях.
ПК8	Готовить конспекты и проводить занятия по обучению сотрудников применению программно–методических комплексов, используемых на предприятии.
ПК9	Участвовать в настройке и наладке программно–аппаратных комплексов.
ПК10	Сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем.
ПК11	Инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем.

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт ИнЭО.

Направление подготовки 09.03.01 Информатика и вычислительная техника.

Кафедра АИКС.

УТВЕРЖДАЮ:

Зав. кафедрой

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
3-8В2В2	Абдурахимов Ильдар Ринатович

Тема работы:

Утверждена приказом проректора–директора (директора) (дата, номер)	18.04.2017, №2773/с

Срок сдачи студентом выполненной работы:	30.05.2017
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе

(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).

Объектом проектирования является веб-приложение, позволяющее выполнять поиск данных из социальных сети по одному хэштегу.

Описание работы веб-приложения:

Пользователь передает веб-приложению одно ключевое слово, после, веб-приложение делает выборку данных по указанному ключевому слову (хэштегу). Далее производится анализ выбранных данных на наличие других хэштегов и формирование документов для выполнения ЛСА. После анализа полученных документов формируется вывод данных пользователю.

Разрабатываемое веб-приложение должно содержать следующий функционал:

- Возможность задания количества ключевых слов, по которым будет проводится дополнительный анализ.
- Латентно–семантический анализ с целью определить насколько связаны слова в документе. Т.е. решается задача оценки семантической близости (сходство): слово–слово. По результатам решения данной задачи выбрать те слова, которые максимально связаны с исходным ключевым словом.
- Латентно–семантический анализ с целью определить связь ключевых слов и документов (выбранных сообщений). В качестве результирующих данных, формируется таблица с данными о результате анализа.

	<p>Основные требования:</p> <p>1. Среда разработки VS 2015–2017 ASP.NET MVC.</p> <p>1. Пользователь выполняет аутентификацию с использованием своих регистрационных данных в соц. сети твиттер. Использовать для этого возможности VS 2015–2017</p> <p>2. Используя полученный токен от пользователя, следует выполнить поисковые запросы к соц. сети twitter с одним ключевым словом. Далее выполняется обработка с поиском похожих ключевых слов – выявление наиболее похожих слов.. Провести латентно–семантический анализ с целью определить связь выделенных ключевых слов и документов (выбранных сообщений).</p>
<p>Перечень подлежащих исследованию, проектированию и разработке вопросов <i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<p>– Анализ предметной области.</p> <p>– Проектирование методики осуществляющей поиск данных из социальных сетей по одному хэштегу.</p> <p>– Проектирование веб-приложения.</p> <p>– Разработка веб-приложения.</p> <p>– Тестирование.</p> <p>– Реализация.</p> <p>– Разработка программной документации.</p>
<p>Перечень графического материала <i>(с точным указанием обязательных чертежей)</i></p>	<p>Презентация в формате *.pptx на X слайдах, содержащих в себе:</p> <ul style="list-style-type: none"> – титульный слайд; – актуальность; – цель; – задачи; – обзор аналогов; – теоретический анализ; – предлагаемая методика поиска (X слайда); – архитектура разработанного компонента – технологии разработки; – реализация вывода исходных данных; – визуализация результатов данных (2 5 слайда); – результаты.
<p>Консультанты по разделам выпускной квалификационной работы <i>(с указанием разделов)</i></p>	
<p>Раздел</p>	<p>Консультант</p>

Названия разделов, которые должны быть написаны на иностранном языке.	
- заключение.	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	06.02.2017
---	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. АиКС	Лулева Е.Е.	К.Т.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3-8B2B2	Абдурахимов Ильдар Ринатович		

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт – ИнЭО.
Направление подготовки (специальность) – 09.03.01 Информатика и вычислительная техника.

Уровень – образования Бакалавриат.

Кафедра – Автоматики и компьютерных систем.

Период выполнения – весенний семестр 2016/2017 учебного года

Форма представления работы:

Бакалаврская работа
(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ–ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	30.05.2017
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	<i>Основная часть</i>	75
	<i>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</i>	15
	<i>Социальная ответственность</i>	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. АиКС	Лунева Е.Е.	К.Т.Н.		

СОГЛАСОВАНО:

И.о.Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
АиКС	Суходоев М.С.	К.Т.Н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
3–8В2В2	Абдурахимов Ильдар Ринатович

Институт	ИнЭО	Кафедра	АиКС
Уровень образования	Бакалавр	Направление/специальность	Информатика и вычислительная техника
		ь	

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. <i>Стоимость ресурсов научного исследования (НИ): материально–технических, энергетических, финансовых, информационных и человеческих</i>	
2. <i>Нормы и нормативы расходования ресурсов</i>	
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	
2. <i>Планирование и формирование бюджета научных исследований</i>	
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	

Перечень графического материала:

—

Дата выдачи задания для раздела по линейному графику	06.02.2017
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Рахимов Т.Р.	К.Т.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3–8В2В2	Абдурахимов Ильдар Ринатович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
3–8В2В2	Абдурахимов Ильдар Ринатович

Институт	Кибернетики	Кафедра	АиКС
Уровень образования	Бакалавриат	Направление/специальность	Информатика и вычислительная техника

Исходные данные к разделу «Социальная ответственность»:

веб-приложение для поиска данных в социальных сетях по заданной предметной области с использованием латентно–семантического анализа

Объектом исследования является социальная сеть Twitter, предметом исследования является процесс поиска данных из социальной сети, позволяющий пользователю передавать одно ключевое слово и получать выборку данных от веб-приложения по указанному ключевому слову (хэштегу) используя латентно семантический анализ.

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Производственная безопасность:

1.2 Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения

Проводится анализ выявленных вредных факторов производственной среды, таких как:

- повышенная напряженность электромагнитного поля;
- недостаток естественного освещения;
- недостаточная освещенность рабочей зоны;
- повышенный уровень шума;
- отклонение показателей микроклимата.

2. Экологическая безопасность:

В работе проведен анализ негативного воздействия на литосферу, а также проведен анализ потребления электроэнергии.

3. Безопасность в чрезвычайных ситуациях:

Защита в чрезвычайных ситуациях:

- выбор наиболее типичной ЧС;
- принятие превентивных мер по предупреждению ЧС.

4. Правовые и организационные вопросы обеспечения безопасности:

4.1 Специальные правовые нормы трудового законодательства
4.2 Организация рабочего места

Правовые и организационные вопросы обеспечения безопасности:

- Специальные правовые нормы трудового законодательства
- Организационные мероприятия при компоновке рабочей зоны

Дата выдачи задания для раздела по линейному графику

06.02.2017

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Инженер кафедры ЭБЖ	Маланова Наталья Викторовна	К.Т.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3–8В2В2	Абдурахимов Ильдар Ринатович		

Реферат

Выпускная квалификационная работа содержит: 104 страницы, 8 рисунков, 26 таблиц, 16 источников, 3 приложения.

Ключевые слова: Веб-приложение для поиска данных в социальных сетях по заданной предметной области с использованием латентно-семантического анализа.

Объектом исследования является веб-приложение, позволяющее выполнять поиск данных из социальных сетей по одному хэштегу.

Цели и задачи исследования (работы): разработка веб-приложения, позволяющий пользователю выполнять поиск данных из социальной сети по одному хэштегу, и представлять результат в виде таблицы на веб странице.

Работа представлена ведением, 4 разделами и заключением, приведен список публикаций, список использованных источников.

В разделе «Теоретический анализ задачи поиска данных из социальной сети по хэштегу» изложен обзор теоретических и практических возможностей решения поставленной задачи. Произведена оценка актуальности задачи. Приведено описание метода поиска слова, основанного на латентно-семантическом анализе. Определены основные задачи для достижения цели.

В разделе «Проектирование и реализация веб-приложения» приведены все важные этапы создания программного приложения – от его проектирования до реализации.

Раздел «Финансовый менеджмент» описывает коммерческую оценку разрабатываемого программного приложения. Оценка дает возможность определить коммерческую привлекательность программного приложения.

Раздел «Социальная ответственность» представляет собой описание экологической безопасности и производственной безопасности при эксплуатации разработанного программного компонента.

В заключении приведены выводы по выполненной работе, этапы и трудности в процессе выполнения поставленной задачи, а так же пути

преодоления этих трудностей. Описаны полученные опыт и навыки после выполнения работы, достигнутый результат.

Оглавление

Определения, обозначения, сокращения, нормативные ссылки	15
Объект и методы исследования	16
1. Теоретический анализ задачи поиска данных из социальной сети по одному хештегу	17
1.1 Актуальность задачи расширенного поиска	17
1.2 Обзор существующих способов и подходов к поиску данных по одному ключевому слову	17
1.3 Обзор существующих способов реализации веб-приложения, которое позволяет выполнять поиск из социальной сети по одному хештегу	18
1.3.1 Что такое веб приложение.....	18
1.3.2 API – ключи.....	19
1.3.3 Получение API-авторизации/аутентификации.....	21
1.4 Латентно–семантический анализ, основные теоретические сведения	21
1.4.1 Область применения ЛСА	24
1.4.2 Актуальность ЛСА	25
1.5 Цель работы и задачи	32
1.6 Техническое задание на разработку программного сервиса	34
1.6.1 Описание работы веб-приложения.....	34
2. Проектирование и реализация программного сервиса	34
2.1 Авторизация в социальной сети Twitter	40
2.2 Получение данных из социальной сети.....	41
2.3 Подготовка полученных данных к ЛСА	42
2.4 Латентно семантический анализ документов.....	43
2.5 Интерфейс пользователя.....	47
2.6 Результат разработки	50
3. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	51
3.1 Актуальность проекта.....	52
3.2 Цели и задачи проекта.....	52
3.4 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	53
3.2 Определение возможных альтернатив проведения научных исследований	59
3.3 Планирование научно–исследовательских работ.....	60
3.4 Бюджет научно–технического исследования (НТИ)	63
4. Социальная ответственность.....	73
4.1 Производственная безопасность.....	74
4.2 Экологическая безопасность	79
4.3 Безопасность в чрезвычайных ситуациях	80
4.4 Правовые и организационные вопросы обеспечения безопасности.....	81

4.4.2 Организация рабочего места.....	82
ЗАКЛЮЧЕНИЕ	84
CONCLUSION	85
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	86
Приложение А.....	88
ПРИЛОЖЕНИЕ Б.....	100
ПРИЛОЖЕНИЕ В.....	102

Введение

В настоящее время социальные сети часто используются для выражения пользователями своего мнения по интересующим их темам, событиям, товарам или же услугам. Социальная сеть так же направлена на построение сообществ в интернете из людей со схожими интересами или деятельностью. Связь осуществляется посредством сервиса внутренней почты или мгновенного обмена сообщениями.

В современно мире огромное количество людей пользуются социальными сетями. Одна из крупнейших в мире социальных сетей Twitter позволяет своим пользователям делиться новостями, своими событиями из жизни, общаться на различные темы. Все это формирует большой поток данных и позволяет использовать его для анализа социальных настроений общества. Что в данный момент больше всего волнует людей или узнать мнение касательно определенной темы, будь то политика, медицина, искусство, спорт. Для анализа данных такого рода необходимы специальные алгоритмы, позволяющие оценить смысл, выявить тематику каждого сообщения социальной сети и на их основе сделать вывод. Одним из алгоритмов является латентно семантический анализ текста.

Целью работы является разработка веб-приложения для поиска данных в социальных сетях по заданной предметной области с использованием латентно–семантического анализа.

Разрабатываемое веб приложение проектируется и реализуется на основе технологии ASP.NET MVC 5.

Разрабатываемое приложение осуществляет упорядочивание информации во время поиска по хештегу с использованием разложения диагональной матрицы по сингулярным значениям. Задачей является реализация латентно семантического анализа записей из социальной сети Twitter на основе платформы для научных вычислений Accord Framework.Net.

Определения, обозначения, сокращения, нормативные ссылки

API – набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах.

HMAC – код аутентификации (проверки подлинности) сообщений.

SAML – язык разметки, основанный на языке XML. Открытый стандарт обмена данными аутентификации и авторизации между участниками, в частности, между поставщиком учётных записей (англ. identity provider) и поставщиком сервиса (англ. service provider).

SSO – технология единого входа (англ. Single Sign - On) – технология, при использовании которой пользователь переходит из одного раздела портала в другой без повторной аутентификации.

SWOT – анализ (Strengths, Weaknesses, Opportunities, и Threats – метод стратегического планирования, заключающийся в выявлении факторов внутренней и внешней среды организации.

LSA – Латентный семантический анализ.

MVC – Model-View-Controller – схема использования нескольких шаблонов проектирования.

ГОСТ – Межгосударственный стандарт.

Объект и методы исследования

Объектом исследования является социальная сеть Twitter, предметом исследования является процесс поиска данных из социальной сети, позволяющий пользователю передавать одно ключевое слово и получать выборку данных от веб-приложения по указанному ключевому слову (хэштегу) используя латентно семантический анализ. А так же архитектурный стиль веб-приложения, его подход, в котором единое приложение строится из набора небольших сервисов, работа для каждого из которых заключается в собственном процессе и коммуникаций с остальными, используя легковесные механизмы, обычно как правило протокол HTTP [4]. Данные сервисы как правило строятся вокруг бизнес-потребностей с использованием автоматизированной среды и с независимым развертыванием. Сами по себе сервисы могут использовать разные технологии хранения данных и писаться на разных языках.

Методы исследования и использования этих сервисов заключается в изучении, сборе и систематизации данных для последующего анализа и реализации поставленной задачи.

1. Теоретический анализ задачи поиска данных из социальной сети по одному хештегу

Приступая к проектированию и разработке программного приложения, необходимо провести анализ предметной области и поставленной задачи. Данный раздел содержит в себе обзор существующих теоретических и практических наработок, позволяющее выполнять поиск данных из социальной сети по одному хэштегу. Изложен обзор теоретических и практических возможностей решения поставленной задачи. Произведена оценка актуальности задачи. Приведено описание метода поиска слова, основанного на латентно–семантическом анализе. Определены основные задачи для достижения цели.

1.1 Актуальность задачи расширенного поиска

На сегодняшний день хэштег – это без преувеличения необходимый инструмент. С его помощью пользователи могут упорядочивать информационный хаос социальных сетей и фильтровать из фонового шума нужную информацию. Интернет-маркетологи пользуются хэштегами чтобы доносить информацию до целевой аудитории.

Отсюда следует что, основная функция хэштега – это группировка постов по темам для удобного поиска пользователями [1].

1.2 Обзор существующих способов и подходов к поиску данных по одному ключевому слову

Не смотря на то что во всех социальных сетях реализуется данная идея, существует топ 5 онлайн инструментов, которые реализуют этот поиск в виде отдельного сервиса [2].

1. Ritetag.
2. Tagdef.
3. Trendsmap.
4. Hashtags.
5. Tagboard.

Вышеперечисленные сервисы имеют разные способы сбора, анализа и представления тэгов. Но смысл у всех остается единственным – это сделать выборку из самых популярных хэштегов.

Отличие разрабатываемого программного приложения не только в поиске популярных хэштегов и представления их, а вывод данных по смысловому значению тэгов.

Таким образом, разработка программного приложения с применением архитектуры веб-приложений, позволит решать такие задачи, как:

1. Выполнять ввод одного хэштега.
2. Проводить анализ данных из социальной сети твиттер на предмет поиска схожих по смыслу хэштегов.
3. Формировать на основе множества хэштегов расширенный поиск данных из социальной сети твиттер.
4. Представлять полученные данные в графическом и табличном виде.

1.3 Обзор существующих способов реализации веб-приложения, которое позволяет выполнять поиск из социальной сети по одному хэштегу

1.3.1 Что такое веб приложение

Веб-приложение представляет из себя программу для пользователя или пользовательское приложение, в котором интерфейс пользователя (UI) представляется в браузере в виде веб-страниц, а основная часть содержится на удаленном веб сервере [3].

Суть запуска веб-приложения не отличается от загрузки обычной веб-страницы: осуществляется ввод ссылки в браузер – после чего пользователю представлен ответ на запрос, которой является интерфейсом пользователя. Основной плюс – это то, что приложение абсолютно не зависит от операционной системы установленной на пользовательском компьютере, оно является кроссплатформенным. Но существует так же один момент, который создает некоторые трудности в работе веб-приложений – это разные

реализации спецификаций в браузерах и возможность настраивать параметры отображения шрифтов, в последствии чего способные привести к некорректной работе некоторых приложений. Стоит так же отметить, что данный минус хотя и существует, но существенного влияния на работу большинства веб-приложений не оказывает.

1.3.2 API – ключи

API-ключи используются всеми открытыми интерфейсами API таких сервисов, как Twitter, Flickr, Google и AWS. API-ключи идентифицируют в сервисе того, кто осуществляет вызов, накладывая ограничения на действия. Часто ограничения выходят за пределы простого предоставления доступа к определенным ресурсам и могут распространяться на действия в ограничении скорости определенных абонентов для представления качественного сервиса другим людям.

Конкретный рабочий механизм зависит от используемой технологии используемой API-ключи в собственных подходах к обмену данными между веб-приложениями. Некоторые системы применяют один общий API-ключ и используя подход, похожий HMAC (проверки подлинности). Более распространенный подход заключается в применении пары из открытого и закрытого ключей. Обычно управление ключами осуществляется централизованно, так же, как и люди централизованно управляли бы определением идентичности людей.

Частично, популярность API-ключей обусловлена тем фактом, что их применение для программ совсем не сложно. В сравнении с обработкой SAML-квотирования аутентификация на основе API-ключа намного проще и понятнее.

Конкретные возможности систем сильно отличаются друг от друга. Есть несколько вариантов как в коммерческой области, так и в области программ с открытым кодом. Некоторые средства просто обрабатывают обмен API-ключами и выполняют некоторые основные функции управления ключами.

Другие средства предлагают все, вплоть до включения ограничения скорости трафика, монетизации, API-каталогов и систем обнаружения.

Некоторые API-системы позволяют создавать мост от API-ключей к существующим сервисам каталогов. Это дает возможность выпускать API-ключи для принципалов (представленных людьми или системами) в организации и следить за жизненным циклом этих ключей примерно так же, как управляли бы их обычными учетными данными. Это дает возможность разрешать доступ к сервисам различными способами, но при сохранении единого источника достоверности, например с использованием SAML при аутентификации людей для SSO (технология единого входа) и API-ключей при обмене данными между сервисами (рис. 1) [4].

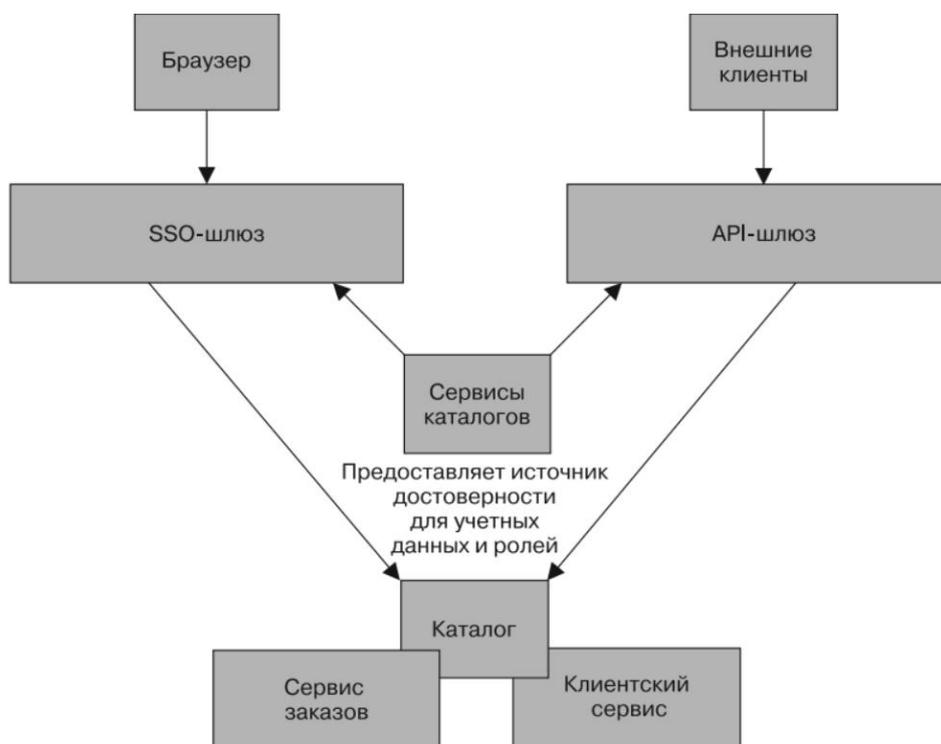


Рисунок 1 – Взаимная аутентификация и авторизация сервисов

Существует множество библиотек под конкретные решения и приложения которые можно использовать для своих задач.

Twitterinvi является лучшим сервисом библиотек с# для архитектурного стиля взаимодействия компонентов распределённого приложения в сети и

Stream API. Tweetinvi поддерживает портативные библиотеки, которые могут быть полезными для поставленной задачи [5].

1.3.3 Получение API-авторизации/аутентификации

Аутентификация позволяет пользовательскому приложению, использовать oauth(авторизацию) на request_token, являющимся запросом авторизации пользователя [6].

Этот метод является заменой из протокола oauth 1.0 проверки подлинности для приложений, использующих обратный вызов проверки подлинности. Этот метод будет использоваться пользователем, вошедшим в систему в качестве учетной записи для авторизации и получения доступа, если параметр force_login имеет значение true.

Для этого метода используется https, и для всех других API-токен переговоров тоже.

Этот метод отличается от получения протокола oauth / авторизовать в том, что если пользователь уже удовлетворил запрос, перенаправление будет происходить без необходимости повторного одобрения заявки [7].

Пользователь не может использовать запрос Токена, пока не будет санкционированных пользователей. Получение авторизации пользователей включает следующие шаги:

Пользователь направляет пользователя к поставщику услуг.

Для того, чтобы иметь возможность обменять запрос доступа, пользователь должен получить одобрение путем перенаправления запроса к поставщику услуг.

Пользователь создает http – запрос Get на URL провайдера и авторизует пользователей с помощью следующих параметров:

oauth_token:

Запрос Токена, полученного на предыдущем шаге. Поставщик услуг может объявить такой параметр, который требуется, или принимать запросы на URL-адрес авторизации пользователей без нее, в этом случае он предложит пользователю ввести его вручную.

Примечание: если поставщик услуг знает, что пользователь должен быть запущен на мобильные устройства и телеприставки, поставщик услуг должен обеспечить пользователю авторизацию на URL и запрос маркера подходящего для ручного ввода [7].

Поставщик услуг проверяет идентичность пользователя и просит согласия. Протокол `oauth` не указывает, как поставщик услуг выполняет проверку подлинности пользователя. Однако в ней определяется набор необходимых шагов:

Поставщик услуг должен сначала проверить личность пользователя прежде чем авторизовать. Он может запрашивать у пользователя данные для входа в случае, если пользователь не внес их.

Поставщик услуг представляет пользователю информацию о запрашиваемом доступе (как зарегистрированному пользователю от разработчика). Информация включает в себя время доступа на защищенном ресурсе. Информация может включать в себя другие детали, характерные для поставщика услуг.

Пользователю должен быть предоставлено или отказано в разрешениях от поставщика услуг, чтобы дать потребителю доступ к защищенным ресурсам по имени. Если пользователю запрещен доступ к ресурсу, то поставщик услуг не должен предоставлять доступ к защищенным ресурсам.

При отображении какой-либо не верной идентифицирующей информации пользователя, поставщик услуг должен информировать пользователя, если он не может гарантировать идентичности этого пользователя. Метод, по которому поставщик услуг информирует пользователя и качество гарантии идентичности выходит за рамки эта Спецификация [7].

После аутентификации пользователя поставщиком услуг и получения гаранта разрешения доступа пользователю, пользователь должен быть уведомлен, что запрос маркера был уполномочен и готов обменяться на маркер доступа. Если пользователю запрещен доступ, пользователь может быть уведомлен о том, что запрос маркера был отозван.

Если пользователь предоставил URL-адрес обратного вызова в `oauth_callback`, поставщик услуг создает `http`-запрос `Get` на URL-адрес, и перенаправляет веб-браузер пользователя на URL-адрес со следующими параметрами:

`oauth_token`: Запрос Токена пользователю разрешено или запрещено.

URL-адрес обратного запроса может включать в себя пользовательские параметры запроса. Поставщик услуг должен сохранить их без изменений и добавить `oauth_token` параметр в существующий запрос [7].

1.4 Латентно–семантический анализ, основные теоретические сведения

Латентный семантический анализ (LSA) является теорией и методом для извлечения и представляет контекстное использование значения слов с помощью статистических вычислений применительно к большому своду текста или, латентный семантический анализ представляет собой математико–статистический метод для извлечения и представления подобия смысла слов и отрывков путем анализа больших тел текста [8]. Основная идея заключается в том, что совокупность информации обо всех текстовых контекстах, в которых данное слово делает и не появляется, предоставляет набор взаимных ограничений, которые в значительной степени определяют сходство значения слов и множества слов друг к другу. Адекватность отражения LSA в человеческих знаниях было установлено в различных формах.

LSA может быть истолковано двумя способами:

- просто в качестве практического целесообразного получения приближенных оценок контекста, использования взаимозаменяемости слов в больших текстовых сегментах, а также

из видов, имеющих смысловые сходства между словами и текстовыми сегментами и что такие отношения могут отражать, или

- в качестве модели вычислительных процессов и представлений, лежащих в основе приобретения и использования знаний.

Сходство в результате тесно имитируют человеческие суждения означающее сходство и человеческой деятельности на основе такого сходства в различных формах.

Как указывалось выше, понимать LSA как об этом думал Ландауэр Дюмей (1996; 1997) который предложил представлять LSA как фундаментальную теорию расчетов приобретения и представления знаний. Они поддерживают что ее базовый механизм можно объяснить давней и главной загадкой. Индуктивная собственность на обучение по которым люди получают гораздо больше знаний, чем кажется, доступность знаний в опыте [8].

1.4.1 Область применения ЛСА

ЛСА был разработан и запатентован в 1988 году группой американских инженеров–исследователей. В области информационного поиска данный подход называют латентно–семантическим индексированием (ЛСИ) [8].

Впервые ЛСА был применен для автоматического индексирования текстов, выявления семантической структуры текста и получения псевдодокументов. Затем этот метод был довольно успешно использован для представления баз знаний и построения когнитивных моделей. В США этот метод был запатентован для проверки знаний школьников и студентов, а так же проверки качества обучающих методик [8].

Недавно Google объявил, что он переходит от поиска по ключевым словам к полностью семантическому поиску. Использование IT гигантов ЛСА подходов, говорит не только об актуальности ЛСА как инструмента для эффективного поиска среди массива данных, но и указывает на область в которых данный подход является очень актуальным и удобным [8].

Основную поддержку этого утверждения приходят от использования ЛСА для получения меры схожести по смыслу слова из текста. Результаты исследований ЛСА показали, что:

- смысл сходства получается таким, что полученные результаты тесно совпадают с человеческим мышлением.
- уровень ЛСА по приобретению таких знаний из текста приближается к человеческим.

В этом и других способах, ЛСА выполняет мощный сравнительно с человеческим сравнением индукции знаний анализ.

1.4.2 Актуальность ЛСА

Область современных информационных поисков чрезвычайно разнообразна и многогранна. Данная область включает такие задачи, как собственно, поиск информации и ее фильтрация, кластеризация и рубрикация документов, поиск на поставленные вопросы ответы, поиск похожих документов и дубликатов, автоматическое аннотирование документа и группы документов и многое другое. Если эти операции выполняет человек, ему необходимо определить основную тему и под темы, основное содержание документа что требует обычно большой объем знаний о языке, мире, организации связного текста и прочих знаний.

Латентно–семантический анализ (ЛСА) является методом обработки информации на естественном языке, благодаря которому происходит анализ взаимосвязей между коллекцией документов и терминами в них встречающимися, сопоставляя некоторые факторы (тематики) всем документам и термам. Основа метода латентно–семантического анализа заключается принцип факторного анализа, а так же, выявление латентных связей изучаемых объектов или явлений. Факторного анализ имеет возможность осуществить выявление скрытых переменных факторов, отвечающих за наличие линейных статистических связей корреляций между наблюдаемыми переменными.

Актуальность его использования вызвана экспоненциальным ростом объема информации и низким качеством точности поиска. Главные цели использования ЛСА – выявление семантических связей между терминами и латентных зависимостей внутри множества текстовых документов, распределения (классификации) документов на группы, расширения поисковых запросов, и некоторых других задач.

Одной из проблем широкого использования ЛСА в информационных системах является его высокая сложность и значительное снижение скорости вычисления при увеличении объема входных данных.

Латентно–семантический анализ отображает отдельные слова в так называемое «семантическое пространство», в котором и производятся все дальнейшие сравнения. При этом делаются следующие предположения:

- Документы это просто набор слов. Порядок слов в документах игнорируется. Важно только то, сколько раз то или иное слово встречается в документе.
- Семантическое значение документа определяется набором слов, которые как правило идут вместе. Например, в биржевых сводках, часто встречаются слова: «фонд», «акция», «доллар»
- Каждое слово имеет единственное значение. Это, безусловно, сильное упрощение, но именно оно делает проблему разрешимой [8].

Пример работы ЛСА заключается в том, чтобы первостепенно из заголовков исключить стоп-символы, слова, которые встречаются в каждом тексте и не несут в себе смысловой нагрузки, это, все предлоги, союзы, частицы и множество других слов.

После, ЛСА производит операцию стемминга, при условии если набор текстов достаточно большой, иначе эту операцию можно опустить так же как и текст на английском языке, в силу того, что количество вариаций любой словоформы в английском языке существенно меньше чем в русском.

После исключаются слова, встречающиеся в единственном экземпляре что, сильно упрощает математические вычисления. После остаются индексируемые слова, которые обозначены курсивом:

1. Британская *полиция* знает о местонахождении *основателя WikiLeaks*
2. В *суде США* начинается процесс *против* россиянина, рассылавшего спам
3. *Церемонию вручения Нобелевской премии* мира бойкотируют 19 *стран*
4. В *Великобритании* арестован *основатель* сайта *Wikileaks* Джулиан Ассандж
5. Украина игнорирует *церемонию вручения Нобелевской премии*
6. Шведский *суд* отказался рассматривать апелляцию *основателя Wikileaks*
7. НАТО и *США* разработали планы обороны *стран* Балтии *против* России
8. *Полиция Великобритании* нашла *основателя WikiLeaks*, но, не *арестовала*
9. В Стокгольме и Осло сегодня состоится *вручение Нобелевских премий* [9].

Следующим шагом составляется частотная матрица индексируемых слов, которая соответствуют индексированным словам, а столбцы – документам. В каждой ячейке матрицы указано какое количество раз слово встречается в соответствующем документе. Пример представлен в таблице 1 [9].

Таблица 1 – Частотная матрицу индексируемых слов

	T1	T2	T3	T4	T5	T6	T7	T8	T9
wekileaks	0	0	0	1	0	0	0	1	0
арестова	0	0	0	1	0	0	0	1	0
великобритан	0	0	0	1	0	0	0	1	0
вручен	0	0	1	0	1	0	0	0	1
нобелевск	0	0	1	0	1	0	0	0	1
основател	1	0	0	1	0	1	0	1	0
полиц	1	0	0	0	0	0	0	1	0
прем	0	0	1	0	1	0	0	0	1
прот	0	1	0	0	0	0	1	0	0
стран	0	0	1	0	0	0	1	0	0
суд	0	1	0	0	0	1	0	0	0
сша	0	1	0	0	0	0	1	0	0
церемон	0	0	1	0	1	0	0	0	0

Далее проводится сингулярное разложение полученной матрицы. Сингулярное разложение это математическая операция раскладывающая матрицу на три составляющих. Т.е. исходную матрицу M мы представляем в виде [9]:

$$M = U * W * V^t$$

где U и V^t – ортогональные матрицы, а W – диагональная матрица. Причем диагональные элементы матрицы W упорядочены в порядке убывания. Диагональные элементы матрицы W называются сингулярными числами. Пример представлен в таблицах 2,3,4 [9].

Таблица 2 – Сингулярное разложение полученной матрицы

wekileaks	0.57	– 0.01	0.01	–0.2	0.13	0.16	– 0.16	– 0.25	– 0.64
арестова	0.34	–0	0.07	0.41	– 0.42	– 0.02	0.1	0.17	– 0.01
великобритан	0.34	–0	0.07	0.41	– 0.42	– 0.02	0.1	0.17	– 0.01
вручен	0	0.52	0.07	– 0.06	– 0.08	– 0.15	– 0.17	0.02	– 0.07
нобелевск	0	0.52	0.07	– 0.06	– 0.08	– 0.15	– 0.17	0.02	– 0.07
основател	0.57	– 0.01	0.01	–0.2	0.13	0.16	– 0.16	– 0.25	0.64
полиц	0.31	0	0.05	0.07	0.57	–0.6	0.29	0.37	–0
прем	0	0.52	0.07	– 0.06	– 0.08	– 0.15	– 0.17	0.02	– 0.25
прот	0.02	0.03	– 0.61	0.13	– 0.05	– 0.22	0	– 0.25	0
стран	0.01	0.22	– 0.31	0.39	0.41	0.56	– 0.22	0.4	0
суд	0.02	0.03	– 0.61	0.13	– 0.05	– 0.22	0	– 0.25	0
сша	0.02	0.03	– 0.61	0.13	– 0.05	– 0.22	0	– 0.25	0
церемон	0	0.38	0.03	0.02	0.08	0.31	0.82	– 0.29	0

Таблица 3 – Сингулярное разложение полученной матрицы

3.41	0	0	0	0	0	0	0	0
0	3.3	0	0	0	0	0	0	0
0	0	2.27	0	0	0	0	0	0
0	0	0	1.49	0	0	0	0	0
0	0	0	0	1.19	0	0	0	0
0	0	0	0	0	0.98	0	0	0
0	0	0	0	0	0	0.71	0	0
0	0	0	0	0	0	0	0.43	0
0	0	0	0	0	0	0	0	0

Таблица 4 – Сингулярное разложение полученной матрицы

T1	T2	T3	T4	T5	T6	T7	T8	T9
0.43	0.05	0.01	0.54	0	0.37	0.01	0.63	0
0	0.02	0.65	-0.01	0.59	0	0.09	-0.01	0.47
0.03	-0.7	-0.04	0.06	0.1	-0.16	0.67	0.09	0.09
-0.22	-0.24	0.15	0.28	-0.11	-0.68	0.44	0.33	-0.13
0.69	-0.32	0.22	-0.49	-0.12	-0.03	-0.27	0.02	-0.19
-0.03	0.3	0.14	-0.17	0.44	-0.15	-0.3	0.24	-0.71
-0.3	0.12	0.4	-0.39	-0.53	0.12	-0.23	0.46	0.13
0.35	0.35	0.35	0.35	-0.35	-0.35	-0.35	-0.35	0

Достоинством сингулярного разложения является выделение ключевых составляющих матриц, позволяя игнорировать шумы. Согласно простым правилам произведения матриц, видно, что столбцы и строки соответствующие меньшим сингулярным значениям дают наименьший вклад в итоговое произведение. Например, мы можем отбросить последние столбцы матрицы U и последние строки матрицы V^t , оставив только первые 2. Важно, что при этом гарантируется, оптимальность полученного произведения.

Разложение такого вида называют двумерным сингулярным разложением представленном в таблицах 5,6,7.

Таблица 5 – Двумерное сингулярное разложение

wekileaks	0.57	-0.01
арестова	0.34	-0
великобритан	0.34	-0
вручен	0	0.52
нобелевск	0	0.52
основател	0.57	-0.01
полиц	0.31	0
прем	0	0.52
прот	0.02	0.03
стран	0.01	0.22
суд	0.02	0.03
сша	0.02	0.03
церемон	0	0.38

Таблица 6 – Двумерное сингулярное разложение

3.41	0
0	3.33

Таблица 7 – Двумерное сингулярное разложение

T1	T2	T3	T4	T5	T6	T7	T8	T9
0.43	0.05	0.01	0.54	0	0.37	0.01	0.63	0
0	0.02	0.65	-0.01	0.59	0	0.09	-0.01	0.47

Следует отметить на графике точки, соответствующие отдельным текстам и словам представленным на рисунке 2:

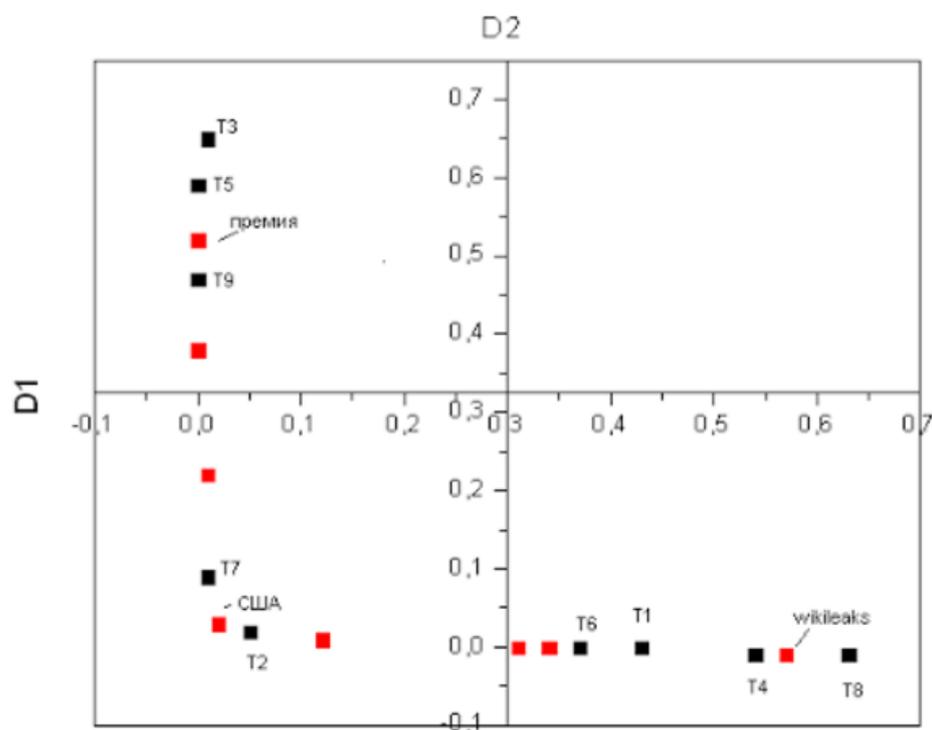


Рисунок 2 – График соответствия

Из данного графика видно, что статьи образуют три независимые группы, первая группа статей располагается рядом со словом «wikileaks», и действительно, если мы посмотрим названия этих статей становится понятно, что они имеют отношение к wikileaks. Другая группа статей образуется вокруг слова «премия», и действительно в них идет обсуждение нобелевской премии.

На практике, конечно, количество групп будет намного больше, пространство будет не двумерным, а многомерным, но сама идея остается той же. Определяются местоположения слов и статей в пространстве и используются эту информацию для определения тематики статьи [9].

1.5 Цель работы и задачи

Целью работы является разработка веб-приложения для поиска данных в социальных сетях по заданной предметной области с использованием латентно–семантического анализа.

Для достижения поставленной цели должны быть решены следующие задачи:

- Изучение технологии ASP.NET MVC 5.

- Основной задачей является реализация латентно семантического анализа данных из социальной сети Twitter на основе платформы для научных вычислений Accord Framework.Net. Поскольку данная платформа содержит необходимые функции для вычисления необходимых метрик.
- Второстепенной задачей является реализация дружественного веб-интерфейса для использования разработанной методики.
- Последним этапом должна быть успешно выполненное тестирование и отладка программного приложения.
- Веб-приложение должно отвечать требованиям технического задания.

1.6 Техническое задание на разработку программного сервиса

Объектом проектирования является веб-приложение, позволяющее выполнять поиск данных из социальной сети по одному хэштегу используя латентно семантический анализ.

1.6.1 Описание работы веб-приложения

Пользователь передает веб-приложению одно ключевое слово, после, веб-приложение делает выборку данных по указанному ключевому слову (хэштегу). Далее производится анализ выбранных данных на наличие других хэштегов и формирование документов для выполнения ЛСА. После анализа полученных документов формируется вывод данных пользователю.

Разрабатываемое веб-приложение должно содержать следующий функционал:

- Возможность задания количества ключевых слов, по которым будет проводится дополнительный анализ.
- Латентно–семантический анализ с целью определить насколько связаны слова в документе. Т.е. решается задача оценки семантической близости (сходство): слово–слово. По результатам решения данной задачи выбрать те слова, которые максимально связаны с исходным ключевым словом.
- Латентно–семантический анализ с целью определить связь ключевых слов и документов (выбранных сообщений). В качестве результирующих данных, формируется таблица с данными о результате анализа.

Основные требования:

1. Среда разработки VS 2015–2017 ASP.NET MVC.

1) Пользователь выполняет аутентификацию с использованием своих регистрационных данных в социальной сети Twitter. Использовать для этого возможности VS 2015-2017

2) Используя полученный токен от пользователя, следует выполнить поисковые запросы к соц. сети Twitter с одним ключевым словом. Далее выполняется обработка с поиском похожих ключевых слов – выявление наиболее похожих слов. Провести латентно–семантический анализ с целью определить связь выделенных ключевых слов и документов (выбранных сообщений).

3) Результаты анализа выводятся ниже в табличном виде. Таблица содержит следующие данные:

- место в списке, номер
- релевантность
- текст твита
- хештеги
- опубликовал пользователь
- дата публикации
- ссылка на твит в социальной сети

2. Проектирование и реализация программного сервиса

Разрабатываемое веб приложение проектируется и реализуется на основе технологии ASP.NET MVC 5, и имеет микросервисную архитектуру. Поскольку приложение нацелено на предоставление сервиса, то его можно разбить на несколько модулей:

- модуль авторизации;
- модуль получения данных;
- модуль подготовки данных;
- модуль анализа полученных данных;
- интерфейс пользователя.

Модуль авторизации реализует логику авторизации пользователя в социальной сети Twitter и контролирует доступ к сервису только авторизованным пользователям.

Модуль получения данных реализует логику получения твитов из социальной сети по ключевому слову.

Модуль подготовки данных реализует логику предварительного приведения твитов к виду, пригодному для проведения латентно семантического анализа их содержимого и связанных хештегов.

Модуль анализа полученных данных реализует непосредственно латентно семантический анализ полученных документов (твитов и хештегов).

Интерфейс пользователя позволяет в дружеской интуитивной форме воспользоваться сервисом, авторизоваться, выполнять запросы и анализировать полученные результаты.

Для авторизации и отправки/получения запросов к социальной сети Twitter используется библиотека Tweetinvi.

Диаграмма вариантов использования приведена на рисунке 3. Диаграмма компонентов приведена на рисунке 4. Диаграмма классов приведена на рисунке 5. Диаграмма пакетов приведена на рисунке .

Рисунок 3 – Диаграмма вариантов использования

Рисунок 4 – Диаграмма компонентов

Рисунок 5 – Диаграмма классов

Модуль авторизации реализуется классами: `TwitterAuthController`, `Helper` и `AuthorizationOnTwitterAttribute` с использованием классов из библиотеки `Tweetinvi`. Подробнее в разделе 2.1.

Модуль получения данных реализуется классом `HomeController`. Подробнее в разделе 2.2.

Модуль подготовки данных реализуется классами: `TweetAnalyzer` и `LsaTweetResult`. Подробнее в разделе 2.3.

Модуль анализа полученных данных реализуется классом `LSA` с использованием классов из библиотеки `Accord.Math`. Подробнее в разделе 2.4.

Рисунок 6 – Диаграмма пакетов

2.1 Авторизация в социальной сети Twitter

В приложении модуль авторизации реализуется контроллером **TwitterAuthController**, атрибутом **AuthorizeOnTwitterAttribute** и глобально статичным классом **Helper**.

Контроллер **TwitterAuthController** содержит в приватных полях информацию о токене и ключе приложения, необходимых для обращения к Twitter. Эти ключи генерируются в специальном приложении, которое разработчик создает внутри социальной сети, и дальнейшие запросы ведутся от имени этого приложения, с учетом учетных данных пользователя сервиса. Контроллер реализует 2 взаимосвязанных метода:

- *public ActionResult Index()* – который проверяет, авторизован ли пользователь, и если нет, то перенаправляет его на страницу авторизации в социальной сети Twitter.
- *public ActionResult ValidateCredentials(string oauth_verifier, string authorization_id)* – метод в обратном запросе от социальной сети получает 2 параметра **oauth_verifier** и **authorization_id**, на основе которых формируются учетные данные, необходимые для осуществления запросов к социальной сети.

Атрибут **AuthorizeOnTwitterAttribute** наследует **AuthorizeAttribute** и переопределяет его метод *void OnAuthorization(AuthorizationContext filterContext)*, который проверяет наличие учетных данных и в случае их отсутствия перенаправляет запрос на метод *public ActionResult Index()* контроллера **TwitterAuthController** для дальнейшей авторизации. Атрибутом помечаются все методы действий контроллеров, которые требуют авторизации.

В библиотеке Tweetinvi используется потоковая логика управления статичными классами. Тем самым при обращении одного и того же пользователя его учетные данные могут храниться в полях статичных классов в контексте потока. ASP NET MVC приложение является многопоточным и запросы одного пользователя могут обрабатывать различные потоки, это

означает что его учетные данные могут хранится независимо друг от друга и приложение не узнает, что пользователь уже авторизовался, или наоборот вышел. Решением этой проблемы является создание статичного класса **Helper** в контексте домена приложения – тем самым храня учетные данные для сессии в одном месте – поле **Cred**. Хранить данные в контексте HTTP запроса осложняется использованием сложных объектов для учетных данных библиотеки Tweetinvi.

2.2 Получение данных из социальной сети

Загрузкой данных из социальной сети управляет контроллер

HomeController. Включает два метода:

- *public ActionResult Index()* – метод реализует формирование главной страницы, помечен атрибутом **AllowAnonymous**, что разрешает его выполнение для неавторизованных пользователей.
- *public ActionResult GetTweets(string hashtag, int variantAnalyze, int? countSimilarWord)* – метод реализует логику загрузки твитов из социальной сети по хештегу и ЛСА полученных твитов. Параметр **hashtag** передает ключевое слово, по которому требуется произвести поиск, параметр **variantAnalyze** передает тип латентно семантического анализа (по одному ключевому слову, по группе слов), необязательный параметр **countSimilarWord** указывает количество похожих слов, которые будут включены в группу для ключа латентно семантического анализа.

2.3 Подготовка полученных данных к ЛСА

За подготовку данных отвечает класс **TweetAnalyzer**. Содержит три метода:

- *private Dictionary<long, string> getDocumets(List<ITweet> tweets)* – реализует логику формирования документов из твитов: добавляет к тексту сообщения связанные с ним хештеги и возвращает набор пар **ID документа – Текст документа**. Параметр **tweets** передает набор твитов из социальной сети.
- *public List<LsaTweetResult>AnalyzeKeywordToTweet(List<ITweet> tweets, string keyWord, int maxCountBestTweets = 5)* – возвращает отсортированный по убыванию релевантности список твитов и степень их релевантности по отношению к заданному ключевому слову. Параметр **tweets** – передает список твитов, параметр **keyword** передает ключевое слово, параметр **maxCountBestTweets** передает максимальное количество наиболее похожих по смыслу к ключевому слову твитов, которые вернет метод (по умолчанию равен 5). Если параметр **maxCountBestTweets** меньше 1, то возвращаются все обработанные твиты.
- *public List<LsaTweetResult>AnalyzeKeywordsToTweet(List<ITweet> tweets, string keyWord, int maxCountBestTweets = 5, int maxCountSimilarWord = 5)* – возвращает отсортированный по убыванию релевантности список твитов и степень их релевантности по отношению к найденной группе слов, близких по смыслу к ключевому слову. Параметр **tweets** – передает список твитов, параметр **keyword** передает ключевое слово, параметр **maxCountBestTweets** передает максимальное количество наиболее похожих по смыслу к ключевому слову твитов, которые вернет метод (по умолчанию равен 5), параметр **maxCountSimilarWord** передает максимально количество похожих слов, из которых будет сформирована ключевая группа для дальнейшего анализа методом ЛСА. Если параметр **maxCountBestTweets** меньше 1, то возвращаются все обработанные твиты. Если параметр **maxCountSimilarWord** меньше 1, то выбрасывается исключение о некорректности операции.

2.4 Латентно семантический анализ документов

Для проведения латентно семантического анализа необходимо выполнить несколько шагов:

- разбиение документов на слова
- удаление стоп-слов
- стемминг оставшихся слов
- построение матрицы «слова – документы»
- разложение полученной матрицы методом SVD (сингулярного разложения матриц)
- выявление интересующих метрик таких как слово – слово, слово – документ, группа слов – документ из полученных матриц.

Стоп слова подразумевают под собой слова, символы, знаки, которые не несут в себе никакой полезной информации и служат для связки слов и предложений, к ним относят:

- союзы и союзные слова
- местоимения
- предлоги
- частицы
- междометия
- указательные слова
- цифры
- знаки препинания
- вводные слова
- ряд некоторых существительных, глаголов, наречий (например, сайт, давать, всегда, однако и др.)

Список используемых стоп слов зафиксирован в коде приложения и приведен в приложении А.

Стемминг – это процесс нахождения основы слова для заданного исходного слова [10]. Основа слова может не совпадать с морфологическим корнем слова.

Из получившегося набора слов убираются все слова, которые встречаются один раз и не несут полезной нагрузки для анализа.

Матрица «слова – документы» строится по следующему принципу: в строках матрицы содержатся слова (оставшиеся после фильтрации по стоп-словам и стемминга), в столбцах содержатся документы (номер документа – ID твита). В каждую ячейку матрицы записывается число, показывающее, сколько раз конкретное слово встречается в конкретном документе. Это наиболее точный способ построения качественной матрицы «слова – документы», поскольку разнообразие получившихся в ячейках чисел позволяет в дальнейшем определить больше зависимостей между словами, документами и темами документов.

Метод SVD разбивает матрицу «слова – документы» на три матрицы: «слова – темы» (прямоугольная матрица), «темы – темы» (квадратная матрица), «темы – документы» (прямоугольная матрица). Особенность сингулярного разложения заключается в том, что оно игнорирует шумы и выделяет ключевые составляющие матрицы, основываясь на контексте документов.

По построенным матрицам можно определить искомые метрики. В данной работе это семантическая близость:

- слово – слово
- слово – документ
- группа слов – документ.

По матрице «слова – темы» можно определить метрику «слово–слово». По матрицам «слова–темы» и «темы–документы» можно определить метрики «слово–документ» и «группа слов–документ».

Главная проблема состоит в том, что матрицы имеют большое количество строк и столбцов, тем самым создавая необходимость вычислять расстояния между словами, темами и документами в многомерном пространстве, где каждое слово, тема и документ представляется своим вектором.

Для решения этой задачи используется мера косинусного сходства. Поскольку она наиболее ориентирована на работу с текстовыми данными. Косинусное сходство вычисляется по формуле [9]:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

где A и B – вектора слов/тем/документов.

Чем больше получившееся значение, тем ближе сходство (от 0 до 1 включительно).

Для того чтобы вычислить метрику «слово–слово» необходимо вычислить расстояние (по формуле косинусного сходства) между двумя векторами: вектором **ключевого слова** и вектором **анализируемого слова**. Вектора слов – это строки в матрице «**слова–темы**», полученной после SVD разложения.

Для того чтобы вычислить метрику «слово–документ» необходимо вычислить расстояние (по формуле косинусного сходства) между двумя векторами: вектором **ключевого слова** и вектором **анализируемого документа**. Вектор ключевого слова равен строке этого слова из матрицы «**слова – темы**». Вектор анализируемого документа равен столбцу этого документа из матрицы «**темы–документы**».

Для того чтобы вычислить метрику «группа слов–документ» необходимо вычислить метрику «слово–слово» между ключевым словом и остальными словами в матрице «**слова – темы**», затем сформировать группу слов с наибольшей метрикой. Вычислить метрику «слово–документ» между

каждым словом из сформированной группы и документом, из матрицы «темы–документы». Сумма метрик «слово–документ» из группы слов, будет являться искомой метрикой «группа слов–документ».

Вся описанная выше логика реализуется в классе **LSA**. Класс имеет один конструктор: *public LSA(ISimilarity<double[]> metric)*, который инициализирует необходимые поля и использует в качестве параметра объект **metric**, реализующий метод вычисления расстояний между векторами. Содержит ряд методов:

- *private List<string> getAllWords(bool returnUniqueWords, params KeyValuePair<long, string>[] documents)* – возвращает список всех слов из полученных документов. Слова отфильтрованы по списку стоп–слов и над ними выполнен стемминг. Параметр **returnUniqueWords** определяет, будут ли возвращены только те уникальные слова, которые встречаются в документах более одного раза (значение true) или все слова, вне зависимости от уникальности и частоты появления их документах (значение false). Параметр **documents** передает список документов. Обработка выполняется параллельно над всеми документами.

- *public Dictionary<long, double> AnalyzeWordToDocument(Dictionary<long, string> documents, string keyWord)* – возвращает список идентификаторов документов и их релевантность ключевому слову, вычисляемую методом ЛСА. Параметр **documents** передает список документов для анализа, параметр **keyword** передает ключевое слово, по которому выполняется поиск близких по тематике документов.

- *public Dictionary<long, double> AnalyzeWordsToDocument(Dictionary<long, string> documents, string keyWord, int countSimilarWord)* – возвращает список идентификаторов документов и их релевантность группе похожих по смыслу слов, вычисляемую методом ЛСА. Параметр **documents** передает список документов для анализа, параметр **keyword** передает ключевое слово, по которому выполняется поиск близких

по смыслу слов, параметр **countSimilarWord** передает максимальный размер группы наиболее похожих по смыслу слов.

Операции SVD разложения, вычисление метрики (косинусного расстояния) и вспомогательных методов и расширений реализуются в библиотеке Accord.Math. Исходный код приведен в приложении А.

2.5 Интерфейс пользователя

Интерфейс пользователя в ASP NET MVC приложении представляется в виде представлений.

Поскольку приложение одностраничное и не предполагает навигации по другим страницам, то реализуется одна страница, состоящая из двух представлений: базовое и пользовательское.

Базовое представление **Layout** описывает разметку html-страницы, включая верхнюю панель навигации, загрузку всех необходимых скриптов и таблиц стилей.

Пользовательское представление **Index** контроллера HomeWorkController описывает разметку блока html-страницы, который помещается внутрь базового представления **Layout**. Представление описывает разметку элементов управления, таких как:

- label – заголовок поля ввода хештега
- input – поле ввода хештега
- label – заголовок поля выбора типа ЛСА
- radiobutton – поле выбора типа ЛСА (2 варианта)
- label – заголовок поля ввода количества похожих слов
- input – поле ввода количества похожих слов
- button – кнопка отправки запроса

Интерфейс пользователя содержит следующие элементы:

- 1) Поле ввода хештега. Здесь пользователь вводит желаемый хештег, по которому он хочет выполнить анализ.

2) Список одиночного выбора варианта анализа. Пользователю может выбрать, каким способом будет выполнен анализ:

- Анализ «Хештег – твит» (слово – документ) представленный на рисунке 7, позволяет найти близкие по смыслу к ключевому слову твиты.
- Анализ «Группа хештегов – твит» (группа слов – документ) представленный на рисунке 8, формирует группу похожих по смыслу на хештег слов и далее оценивает схожесть этой группы с каждым документом.

3) Поле ввода группы похожих слов. Позволяет задать максимальный размер группы похожих слов, которая будет формироваться при анализе «Группа хештегов – твит»

4) Таблица с результатами анализа.

- Место – место в списке по результатам поиска похожих твитов. Чем меньше – тем более похожий твит.
- Релевантность – степень соответствия, рассчитанная алгоритмом ЛСА на основе полученных данных. Чем больше – тем лучше.
- Текст твита – содержание твита.
- Хештеги – прикрепленные к твиту хештеги автором твита
- Опубликовал пользователь – ник автора твита

Результаты анализа представлены ниже на рисунках 7,8.

Латентно семантический анализатор твитов по хеш тегу

Введите хэш-тег

Выберите вариант анализа
 Анализ типа Хештег - Твит
 Анализ типа Группа похожих на хештег слов - Твит

Введите размер группы слов

Результаты запроса

Место	Релевантность	Текст твита	Хештеги	Опубликовал пользователь	Дата публикации	Ссылка на твит
1	0,843466017587534	#Нефть дорожает на данных о снижении добычи в США https://t.co/N3P441ooLx	#Нефть	Сыктывкар	09.06.2017 13:20:29	Перейти в Twitter
2	0,107381572294019	#СБУ почала розслідування обставин вибуху на території посольства США за ч. 1 ст. 258 (терористичний акт) КК України https://t.co/LIJzAqP1cg	#СБУ	СБ України	08.06.2017 16:00:14	Перейти в Twitter
3	0,101724271545877	Сирийские войска под бомбами: коалиция во главе с США нанесла удар #новости #сегодня #followback	#новости #сегодня #followback	Алёна Антонова	09.06.2017 13:06:12	Перейти в Twitter
4	0,0892597874689177	США хотят перехватить стратегическую инициативу в Сирии — Игорь Морозов Сенатор ...		News Front	09.06.2017 13:16:00	Перейти в Twitter

Рисунок 7 – Интерфейс пользователя.

Латентно семантический анализатор твитов по хеш тегу

Введите хэш-тег

Выберите вариант анализа
 Анализ типа Хештег - Твит
 Анализ типа Группа похожих на хештег слов - Твит

Введите размер группы слов

Результаты запроса

Место	Релевантность	Текст твита	Хештеги	Опубликовал пользователь	Дата публикации	Ссылка на твит
1	0,909806550452453	Квалификация #ЧМ2018 в зоне КОНКАКАФ: Мексикам оторвалась на 5 очков, США вошли в тройку: https://t.co/qY4tdSwMR https://t.co/XoUxkAJ0Bx	#ЧМ2018	Чемпионат мира 2018	09.06.2017 13:06:16	Перейти в Twitter
2	0,806585590885187	Госдеп поблагодарил полицию за реакцию на взрыв в посольстве США в Киеве https://t.co/rFYep3F7Vc #Общество https://t.co/w65o2xRV0B	#Общество	РИА Новости Украина	09.06.2017 12:45:38	Перейти в Twitter
3	0,805113052675408	#США считает, что удары по войскам Сирии не являются эскапацией https://t.co/3Nhg887xie	#США	Вадим	09.06.2017 12:40:20	Перейти в Twitter

Рисунок 8 – Интерфейс пользователя.

2.6 Результат разработки

В результате разработки был проведен масштабный обзор существующих решений по реализации алгоритмов ЛСА. В результате чего из-за отсутствия реализаций под существующие ограничения (язык разработки C#, используемая технология ASP NET MVC) было принято решение разработать собственную библиотеку, которая удовлетворяла бы всем заявленным требованиям. В качестве основы математического аппарата была использована библиотека Accord.Math из платформы машинного обучения AccordFramework для .Net приложений. Были реализованы методы латентно семантического анализа для вычисления метрик «слово–слово», «слово–документ», «группа слов – документ».

Проведен обзор существующих библиотек для интеграции с социальной сетью Twitter, по итогам которого, была выбрана кроссплатформенная библиотека Tweetinvi. Так же выполнена непосредственно интеграция с социальной сетью Twitter, которая выступает в качестве источника данных для анализа – твитов, что позволяет пользователям оценить настроения общества касательно интересующей темы.

Разработано веб-приложение, которое позволяет интуитивно сформировать запрос и получить подробную и легко воспринимаемую информацию о проведенном анализе.

3. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

В процессе разработки программного приложения стоит учитывать коммерческую перспективность и потенциал, которые способны определить необходимый бюджет для продвижения создаваемого продукта.

Раздел финансовый менеджмент, ресурсоэффективность и ресурсосбережение представляет коммерческую оценку разрабатываемого программного продукта. Эта оценка позволяет увидеть у программного продукта его коммерческую конкурентоспособность, его привлекательность, а также учесть соответствие продукта современным требованиям в области ресурсосбережения и ресурсоэффективности.

При проведении анализа и оценки программного продукта, был определен сегмент пользователей, создан план научно–исследовательской работы, а также произведена оценка эффективности разработанного компонента.

3.1 Актуальность проекта.

Приложение для поиска данных из социальной сети, позволяющий пользователю передавать одно ключевое слово и получать выборку данных от веб-приложения по указанному ключевому слову (хэштегу) используя латентно семантический анализ для дальнейшего анализа для извлечения контекстно–зависимых значений лексических единиц при помощи статистической обработки больших корпусов текстов.

Разработка нового способа анализа является сегодня актуальным. Способ который облегчит поиск в бурно растущем интернет пространстве, так как способы нынешние не всегда эффективны при больших объемах и разнообразии информации. Актуальность проекта заключается в его эксклюзивности и малое количество конкурентов. Перспективность дает возможность развития латентного поиска в разных социальных сетях аналогично глобальным поисковым системам.

3.2 Цели и задачи проекта.

Целью работы является разработка веб-приложения для поиска данных в социальных сетях по заданной предметной области с использованием латентно–семантического анализа.

Для реализации программного приложения должны быть решены следующие задачи:

- Основной задачей является реализация методики латентно семантического анализа записей из социальной сети Twitter на основе платформы для научных вычислений Accord Framework.Net. Поскольку данный фреймворк содержит необходимые функции для вычисления нужных метрик.
- Второстепенной задачей является реализация дружественного веб интерфейса для использования разработанной методики.

Веб-приложение должно отвечать требованиям технического задания.

3.4 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

3.4.1 Потенциальные потребители результатов исследования

Программный компонент может быть использован как физическим лицом так и коммерческой организацией. Приоритет в большей заинтересованности отдается коммерческим организациям, отсюда можно сделать вывод что способ поиска по хэштегу с использованием латентно–семантического анализа является удачным в реализации семантического поиска потенциальными потребителями созданного программного продукта.

Спецификация продукта обуславливает важные критерии сегментирования рынка. Пример тому сфера деятельности организаций, так как способ поиска по хэштегу с использованием латентно–семантического анализа производится в рамках узкой специализации или области, которые, более или менее совпадают со сферами деятельности организаций, использующих этот программный продукт.

3.4.2 Анализ конкурентных технических решений

Анализ аналоговых решений позволяет выявить недостатки своего продукта и повысить тем самым конкурентоспособность. Так же анализ учитывает достоинства своего продукта и недостатки продуктов конкурентных.

6. Основными конкурентами программного продукта являются сервисы, как «Ritetag» (K1), «Tagdef» (K2) и «Trendsmar» (K3). Все сервисы предоставляются в бесплатном режимах и возможности латентного поиска отсутствуют, вследствие чего конкурентоспособность существенно снижается.

Отличительным преимуществом разрабатываемого программного компонента сравнительно с конкурентными решениями вышеперечисленными, является уникальный функционал, а также удобство

эксплуатации программного приложения. Данный фактор влияет на высокую конкурентоспособность продукта.

Для подробного анализа конкурентных решений, составляется оценочная карта для сравнения конкурентных технических решений, представленная в таблице 8.

Таблица 8 – Оценочная карта для сравнения конкурентных технических решений (разработок)

Критерии оценки	Вес критерия	Баллы				Конкурентоспособность			
		Б _ф	Б _{к1}	Б _{к2}	Б _{к3}	К _ф	К _{к1}	К _{к2}	К _{к3}
1	2	3	4	5	6	7	8	9	10
Технические критерии оценки ресурсоэффективности									
1. Повышение производительности труда пользователя	0,2	5	3	3	2	0,6	0,3	0,2	0,2
2. Удобство в эксплуатации (соответствует требованиям потребителей)	0,2	4	4	2	4	1	0,8	0,6	0,8
3. Потребность в ресурсах памяти	0,1	5	3	2	2	0,4	0,8	0,4	0,5
4. Функциональная мощность (предоставляемые возможности)	0,5	3	2	2	2	2,6	1	1	1
Итого	1					4,6	2,9	2,2	2,5
Экономические критерии оценки эффективности									
1. Конкурентоспособность продукта	0,5	4	3	4	4	2,5	1	2	2
2. Уровень проникновения на рынок	0,3	1	2	4	2	0,3	0,6	0,4	0,5
3. Цена	0,2	5	4	1	2	1	1,2	0,8	0,5
Итого	1					3,8	2,8	3,2	3

Исходя из показателей критериев конкурентоспособности, очевидно, что конкурирующие приложения имеют уровень выше среднего проникновения на рынок. Основными недостатками альтернативных решений – это часть функциональных возможностей, которая довольно ограничена в сравнительно с описываемым в данной работе программным приложением. Уникальность функционала приложения играет важную роль в привлекательности продукта для конечных пользователей подобных систем.

3.4.3 Технология QuaD

Технология QuaD дает возможность оценить рентабельность вложения денежных средств в разрабатываемое приложение, дает возможность получить оценку перспективности и качеству выполненной работы. Результат показателей оценки представлен в таблице 9.

Таблица 9 – Оценочная карта для сравнения конкурентных технических решений (разработок)

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение (3/4)	Средневзвешенное значение (5x2)
1	2	3	4	5	6
Показатели оценки качества разработки					
1. Повышение производительности труда пользователя	0,05	90	100	0,9	0,045
2. Удобство в эксплуатации (соответствует требованиям потребителей)	0,20	100	100	1	0,200
3. Потребность в ресурсах памяти	0,20	80	100	0,8	0,16
4. Функциональная мощность (предоставляемые возможности)	0,20	100	100	1	0,2
Показатели оценки коммерческого потенциала разработки					
5. Конкурентоспособность продукта	0,30	100	100	1	0,3
6. Уровень проникновения на рынок	0,02	90	100	0,15	0,003
7. Цена	0,03	900	100	0,5	0,015
Итого	1				0,923

После получения среднего значение показателя перспективности и качества равно 0,923 это говорит о том, что разрабатываемое приложение является перспективным.

3.4.4 SWOT–анализ

На первом этапе SWOT– анализа определяются сильные и слабые стороны разработанного программного решения, а также определяются угрозы и возможности, способные возникнуть во внешней среде продукта.

Во втором этапе анализа выявляются соответствия между сильными и слабыми сторонами разработки внешним условиям ее окружающей среды. Это соответствие или несоответствие способствует тому чтобы выявить необходимость в проведения стратегических изменений.

Построенные в рамках второго этапа анализа, интерактивные матрицы проекта представлены в виде таблиц 3–6.

Таблица 10 – Интерактивная матрица сильных возможностей и сторон

Сильные стороны проекта						
Направления развития		C1	C2	C3	C4	C5
	B1	–	–	0	–	–
	B2	0	+	+	–	+
	B3	–	–	–	–	+
	B4	+	–	–	–	–

Таблица 11 – Интерактивная матрица слабых сторон и возможностей

Слабые стороны проекта				
Сдерживающие факторы		Сл1	Сл2	Сл3
	B1	–	–	–
	B2	+	–	–
	B3	+	0	–
	B4	–	–	–

Таблица 12 – Интерактивная матрица сильных сторон и угроз

Сильные стороны проекта						
Угрозы развития		C1	C2	C3	C4	C5
	У1	–	+	–	+	–
	У2	–	–	–	–	–
	У3	–	–	+	–	+
	У4	–	–	+	–	+

Таблица 13 – Интерактивная матрица слабых сторон и угроз

Слабые стороны проекта				
Уязвимости		Сл1	Сл2	Сл3
	У1	+	–	0
	У2	–	–	+
	У3	–	–	–
	У4	–	–	–

Итоговый результат SWOT–анализа представлен в таблице 14.

Таблица 14 – Матрица SWOT

	Сильные стороны:	Слабые стороны:
	<p>С1. Современные технологии разработки.</p> <p>С2. Уникальность функционала приложения.</p> <p>С3. Аудитория пользователей широкая и целевая.</p> <p>С4. Понятный интерфейс.</p> <p>С5. Интерес к разработанному приложению.</p>	<p>Сл1. Уровень проникновения на рынок программного обеспечения низкий.</p> <p>Сл2. Необходимость в финансировании.</p> <p>Сл3. Задержка выдачи результатов</p>

Возможности:	Направления развития:	Сдерживающие факторы:
<p>V1. Подорожание конкурентных разработок.</p> <p>V2. Высокий спрос на продукт.</p> <p>V3. Возникновение финансирования разработки</p> <p>V4. Привлечение и объединение конкурентных креативных решений.</p>	<p>1. B2B3C2C3C5 –</p> <p>При спросе на товар необходимо использовать финансирование расширяя функционал продукта и привлекать новую аудиторию .</p> <p>2. B4C1 –</p> <p>Необходимо использовать и привлекать перспективных конкурентов к совместной взаимовыгодной деятельности используя современные технологии разработки.</p>	<p>1. Сл2B2 – необходимо привлечь спонсоров. Аргументом может служить высокий спрос.</p> <p>2. B3Cл1 – при получении финансирования, модернизировать продукт, чтобы он стал привлекательным на рынке.</p>

Угрозы:	Угрозы развития:	Уязвимости:
У1. Отсутствие понимания или интереса пользователя к продукту.	1. У1С2С4 – разрекламировать уникальный функционал и понятный интерфейс.	1. У1Сл1 – Занять рынок своим продуктом пробуждая интерес, знакомя клиента с продуктом.
У2. Ошибки в различных операционных систем.	2. У3У4С3С5 – попробовать перенять функционал у конкурента либо разработать ноу хау для клиента в своем приложении и отвлечь от конкурентных аналогов .	2. У2Сл3 – привлечение в персонал узких специалистов для выявления и тестирования «багов» и задержек в работе приложения. Сопровождение клиентов для выявления и модернизации продукта
У3. Конкурентные приложения с лучшим функционалом.		
У4. Потеря целевой аудитории пользователей от конкурентных аналогов.		

3.2 Определение возможных альтернатив проведения научных исследований

С помощью морфологического подхода можно определить возможные альтернативы в проведении научных исследований. Данный подход и его результат использования представлен в таблице 15.

Таблица 15 – Морфологическая матрица

	1	2
А. Среда разработки	Microsoft Visual Studio	SharpDeveloper
Б. Представление в виде SaaS–приложения	–	+
В. Использование дополнительных ресурсов для хранения данных и их обработки	–	+

Исходя из полученных данных морфологической матрицы, направления научных исследований при реализации программного приложения возможно в 3 вариантах:

- Исполнение A1B1B1;
- Исполнение A2B2B2;
- Исполнение A1B2B3.

При дальнейших расчетах данные варианты будут использованы в качестве различных исполнений реализации разработки.

3.3 Планирование научно–исследовательских работ

3.3.1 Структура работ в рамках научного исследования

В научно–исследовательской работе важной частью планирования является виды запланированных работ и определение рабочей группы, которые выполняются участниками этой группы.

Рабочая группа в рамках описываемой работы состоит из двух участников: научный руководитель (руководитель) и студент.

Перечень этапов работ и распределения исполнителей представлен в таблице 16.

Таблица 16 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение задания	Руководитель
Выбор направления исследований	2	Поиск и изучение материалов по теме	Студент, руководитель
	3	Анализ источников и представленной в них информации	Студент
	4	Выбор направления исследований	Студент, руководитель
	5	Календарное планирование работ по теме	Студент
Проектирование приложения	6	Проектирование архитектуры	Студент
	7	Выбор технологий разработки	Студент
	8	Выбор подходов к решению поставленной задачи	Студент, руководитель
	9	Выбор средства обеспечения хранения данных	Студент
Реализация приложения	10	Реализация компонента получения данных из социальной сети	Студент
	11	Реализация компонента обработки и анализа данных	Студент
	12	Реализация компонента визуализации данных	Студент
	13	Реализация обеспечения хранения данных в базе	Студент
Тестирование	14	Тестирование и исправление ошибок работы приложения	Студент
Анализ результатов работы и оформление пояснительной записки	15	Оценка соответствия программного приложения заявленным требованиям к нему	Студент, руководитель
	16	Оформление пояснительной записки	Студент, руководитель

3.3.2 Определение трудоемкости выполнения работ

На данном этапе работы описывается разработанное программное приложение. Трудовые затраты на разработку программных приложений занимают большую часть ее стоимости. Для каждого из участников научного исследования важно определить трудоемкость работы.

Исходя из ожидаемой трудоемкости работ, можно определить продолжительность одной работы, которая также будет зависеть от числа исполнителей, выполняющих эту работу.

3.3.3 Разработка графика проведения научного исследования

Для разработки графика проведения научного исследования, необходимо определить временные рамки его выполнения, определить трудоемкость выполнения для каждой задачи по дням.

Значение продолжительности выполнения i -ой работы в календарных днях определяется по формуле: $T_{ki} = T_{ri} \cdot k_{\text{кал}}$, где T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{ri} – продолжительность выполнения i -й работы в рабочих днях;

$k_{\text{кал}}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}},$$

где $T_{\text{кал}}$ – количество календарных дней в году;

$T_{\text{вых}}$ – количество выходных дней в году;

$T_{\text{пр}}$ – количество праздничных дней в году.

В 2017 году общее количество календарных дней составляет 365 ($T_{\text{кал}}$), так же праздничные и выходные дни это ($T_{\text{вых}} + T_{\text{пр}}$) в общей сложности 119.

Таким образом, $K_{\text{кал}} = 366 / (366 - 119) = 1,48$.

Результирующие значения продолжительности выполнения работ в календарных днях остальные временные показатели проведения научного исследования представлены в приложении А.

На основе представленной в приложении Б таблице, строится календарный в виде диаграммы Ганта план–график, который является наиболее удобным и наглядным. Приложение В отображает диаграмму Ганта, где С – студент и Р – руководитель. Исходя из того, что график требуется строить для максимального по длительности исполнения работ в рамках научно исследовательского проекта на основе приложения А, длительность работ для исполнения выбираем №2 (158 календарных дней).

3.4 Бюджет научно–технического исследования (НТИ)

Для планирования бюджета НТИ, необходимо отразить все расходы, связанные с его выполнением. В последующих пунктах будет подробно рассмотрена каждая из статей расходов.

3.4.1 Расчет материальных затрат НТИ

В расчетах материальных затрат учитывается стоимость оборудования и материалов, используемых при разработке проекта.

Для разработки программного обеспечения используется персональный компьютер, клавиатура, мышь. Расход на покупку персонального компьютера и комплектующих будет основной статьей при расчете материальных затрат, результат данного расчета представлен в таблице 17.

Расчет материальных затрат осуществляется по следующей формуле:

$$Z_m = (I + K_T) \sum_{i=1}^m C_i \cdot N_{\text{расх}i}$$

где m – количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{расхi}$ – количество материальных ресурсов i -го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м² и т.д.);

$Ц_i$ – цена приобретения единицы i -го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м² и т.д.); k_T – коэффициент, учитывающий транспортно–заготовительные расходы.

Затраты по доставке от цен на приобретение материальных ресурсов, учитываются как транспортные расходы в пределах 15–25% от стоимости материалов.

В данном случае показатели m и $N = 1$, коэффициент k_T примем равным $k_T = 0.15$. В таком случае формула расчета затрат будет иметь вид $Z_m = 1.15 * Ц_i$.

Таблица 17 – Материальные затраты

Наименование	Единица измерения	Количество			Цена за ед., руб.			Затраты на материалы, (Z _м), руб.		
		Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Персональный компьютер с периферийным и устройствами	шт	1	1	1	52200	45000	52200	59880	57350	59880
Итого								59880	57350	59880

3.4.2 Основная заработная плата исполнителей темы

Данный раздел осуществляет расчет заработной платы работников, участвующих разработке программного продукта. В заработную плату включает в себя дополнительную и основную заработную плату:

$$Z_{зн} + Z_{осн} + Z_{доп} ,$$

где $Z_{осн}$ – основная заработная плата;

$Z_{доп}$ – дополнительная заработная плата (10–20 % от $Z_{осн}$).

Оклад руководителя, согласно приказу ТПУ (должность – доцент, степень – кандидат технических наук) без учета районного коэффициента составляет 23264 рублей 86 копеек. Отсюда месячный оклад руководителя составит $Z_m = 1,3 * 23264,18 = 30244,318$ рублей. Месячный оклад студента рассчитывается из минимального размера заработной платы, без учета районного коэффициента который составляет 6204 рубля. Месячный оклад студента составит $Z_m = 1,3 \cdot 6204 = 8065,2$ руб.

Среднедневную заработную плату руководителя можно получить имея значение месячного оклада и вычисляется по следующей формуле:

$$Z_{дн} = \frac{Z_m \cdot M}{F_d}$$

где Z_m – месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года:

($M = 11,2$ месяца для руководителя, $M = 10,4$ месяца для студента);

F_d – действительный годовой фонд рабочего времени, раб. дн. (для студента – 275 дней, для руководителя – 247).

Среднедневная заработная плата руководителя составит: $Z_{дн} = (30244,318 * 11,2) / 247 = 1371,4$ руб.

Среднедневная заработная плата студента составит: $Z_{дн} = (8065,2 * 10,4) / 275 = 305$ руб.

Расчет основной заработной платы для каждого из вариантов исполнений работы представлен в таблице 18.

Таблица 18 – Расчёт основной заработной платы

Исполнитель	Оклад, руб./мес.	Средне – дневна я ставка	Затраты времени, раб. дни			Кoeffи циент	Фонд з/платы, руб.		
			Исп. 1	Исп. 2	Исп. 3		Исп. 1	Исп. 2	Исп. 3
Руководител ь	23264,18	1371.4	12.5	13.5	13.5	1.3	17142.5	18513.9	18513.9
Студент	6204	305	76.5	92.5	84.5		23332.5	28212.5	25772.5
Итого							40475	46726.4	44286.4

После расчета основной заработной платы можно вычислить дополнительную оплату участников разработки и вычислить заработную плату общую.

3.4.3 Дополнительная заработная плата исполнителей темы

Для получения расчета дополнительной заработной платы используется следующая формула:

$$Z_{доп} = k_{доп} + Z_{осн} ,$$

где $k_{доп}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12 – 0,15).

Примем коэффициент дополнительной заработной платы равным 0,14.

С использованием данных таблицы 15, а также вычисленных значений дополнительной заработной платы, можно вычислить полное значение заработной платы. Результаты вычислений представлены в таблице 19.

Таблица 19 – Расчёт дополнительной и итоговой заработной платы

Исполнитель	Основная заработная плата, руб.			Коэффициент дополнительной заработной платы	Дополнительная заработная плата, руб.			Итоговая заработная плата, руб.		
	Исп. 1	Исп. 2	Исп. 3		Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Научный руководитель	17142,5	18513,9	18513,9	0,14	2399,95	2591,94	2591,94	19542,45	21105,84	21105,84
Студент	23332,5	28212,5	25772,5		3266,55	3949,75	3608,15	26599,05	32162,25	29380,65
Итого								46141,5	53268,09	50486,49

3.4.4 Отчисления во внебюджетные фонды (страховые отчисления)

Отчисления во внебюджетные фонды являются отчислениями в органы пенсионного фонда (ПФ), и медицинского страхования (ФФОМС) государственного социального страхования (ФСС), от затрат на оплату труда работников. Согласно пункту 1 ст.58 закона №212–ФЗ для учреждений осуществляющих образовательную и научную деятельность от 2014 года водится пониженная ставка – 27,1%. Данная ставка будет использоваться при расчете отчислений во внебюджетные фонды. В таблице 20 представлены результаты расчетов.

Таблица 20 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.			Дополнительная заработная плата, руб.		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Руководитель проекта	17142.5	18513.9	18513.9	2399,95	2591,94	2591,94
Студент–дипломник	23332.5	28212.5	25772.5	3266,55	3949,75	3608,15
Коэффициент отчислений во внебюджетные фонды	0,271					
Итого						
Размер отчислений (Исполнение 1)	12504,34					
Размер отчислений (Исполнение 2)	14435,65					
Размер отчислений (Исполнение 3)	13681,83					

3.4.5 Накладные расходы

Не вошедшие в описанные ранее статьи расходов, накладные расходы содержат в себе прочие затраты. За пример могут браться расходы на оплату аренды дополнительных веб-ресурсов для поддержки функционала некоторых из исполнений решения или электричества. По следующей формуле определяется величина накладных расходов:

$$Z_{\text{накл}} = (\text{сумма статей } 1 \div 4) \cdot k_{\text{нр}}$$

где $k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы.

Величина коэффициента накладных расходов может быть взята в размере 16%.

Сумма накладных расходов для каждого из исполнений будет составлять:

$$\text{Исполнение 1: } Z_{\text{накл}} = (159000,84 / 4) \cdot 0,60 = 23850,126 \text{ руб.}$$

$$\text{Исполнение 2: } Z_{\text{накл}} = (171780,14 / 4) \cdot 0,60 = 25767,021 \text{ руб.}$$

$$\text{Исполнение 3: } Z_{\text{накл}} = (168334,72 / 4) \cdot 0,60 = 25250,208 \text{ руб.}$$

3.4.6 Формирование бюджета затрат научно–исследовательского проекта

После того, как была рассмотрена каждая из статей расходов, можно приступить к формированию бюджета затрат проекта. Результаты вычисления итогового бюджета по каждому из исполнений проекта представлены в таблице 21.

Таблица 21 – Расчет бюджета затрат НИИ

Наименование статьи	Сумма, руб.		
	Исп.1	Исп.2	Исп.3
1. Материальные затраты НИИ	59880	57350	59880
2. Затраты по основной заработной плате исполнителей темы	40475	46726.4	44286.4
3. Затраты по дополнительной заработной плате исполнителей темы	46141,5	53268,09	50486 , 49
4. Отчисления во внебюджетные фонды	12504,34	14435,65	13681,83
5. Накладные расходы	23850,126	25767,021	25250,208
6. Бюджет затрат НИИ		171780,14	168334,72

Размер бюджета которой представлен в таблице 21, для каждого из исполнений, менее затратным является исполнение 1.

3.4.7 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности

Для определения эффективности исследования, первоначально необходимо вычислить ресурсоэффективность и интегральный показатель финансовой эффективности.

Интегральный финансовый показатель вычисляется по следующей формуле:

$$I_{\text{финр}}^{\text{исп}i} = \frac{\Phi_{ri}}{\Phi_{\text{max}}}$$

где $I_{\text{финр}}^{\text{исп}i}$ – интегральный финансовый показатель разработки;

Φ_{ri} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно–исследовательского проекта.

На основе значений, представленных в таблице 16 указываются значения показателей стоимости. Отсюда следует что для каждого из исполнений интегральный финансовый показатель имеет следующие значения:

Исполнение 1: $I_{\text{финр}} = 159000,84 / 171780,14 = 0,92$;

Исполнение 2: $I_{\text{финр}} = 171780,14 / 171780,14 = 1$;

Исполнение 3: $I_{\text{финр}} = 168334,72 / 171780,14 = 0,97$.

Интегральный показатель ресурсоэффективности определяется с использованием следующей формулы:

$$I_{pi} = \sum a_i \cdot b_i$$

I_{pi} – интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки;

a^i – весовой коэффициент i -го варианта исполнения разработки;

b_i^a, b_i^p – бальная оценка i -го варианта исполнения разработки,

устанавливается экспертным путем по выбранной шкале оценивания;

n – число параметров сравнения.

Бальные оценки каждого из исполнений и весовые коэффициенты оценочных критериев представлены в таблице 22.

Таблица 22 – Сравнительная оценка характеристик вариантов исполнения

Оценочные критерии	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
1. Функциональная мощность (предоставляемые возможности)	0,2	4	5	4
2. Скорость работы	0,15	4	4	4
3. Потребность в ресурсах памяти	0,2	4	5	3
4. Удобство в эксплуатации	0,15	4	4	5
5. Повышение производительности труда пользователя	0,1	4	5	5
6. Цена	0,2	5	3	4
Итого	1			

На основании таблицы 22 можно произвести расчет интегральных показателей ресурсоэффективности для каждого из исполнений:

$$I_{p-исп1} = 4*0,2 + 4*0,15 + 4*0,2 + 4*0,15 + 4*0,1 + 5*0,2 = 4,2;$$

$$I_{p-исп2} = 5*0,2 + 4*0,15 + 5*0,2 + 4*0,15 + 5*0,1 + 3*0,2 = 4,3;$$

$$I_{p-исп3} = 4*0,2 + 4*0,15 + 3*0,2 + 5*0,15 + 5*0,1 + 4*0,2 = 4,05.$$

Интегральный показатель эффективности вариантов исполнения разработки определяется на основе значений интегрального показателя ресурсоэффективности и интегрального финансового показателя по следующей формуле:

$$I_{\text{исп1}} = \frac{I_p - \text{исп1}}{I_{\text{финр}}}$$

$$I_{\text{исп1}} = 4,2/0,92 = 4,56$$

$$I_{\text{исп2}} = 4,3/1 = 4,3$$

$$I_{\text{исп3}} = 4,05/0,97 = 4,17$$

Вычисления интегрального показателя эффективности вариантов исполнения разработки, дает возможность произвести определение сравнительной эффективности проекта по следующей формуле:

$$Э_{\text{ср}} = \frac{I_{\text{исп1}}}{I_{\text{исп2}}}$$

$$Э_{\text{ср1}} = 4,56/4,3 = 1,06$$

$$Э_{\text{ср1}} = 4,3/4,3 = 1$$

$$Э_{\text{ср1}} = 4,17/4,3 = 0,96$$

Результаты вычисления показателей эффективности представлены в таблице 23.

Таблица 23 – Сравнительная эффективность разработки

№ п/п	Показатели	Исп.1	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	0,92	1	0,97
2	Интегральный показатель ресурсоэффективности разработки	4,2	4,3	4,05
3	Интегральный показатель эффективности	4,56	4,4	4,17
4	Сравнительная эффективность вариантов исполнения	1,06	1	0,96

Согласно значениям показателей эффективности, с позиции финансовой и ресурсной эффективности наиболее эффективным является вариант исполнения 1. Данный вариант имеет наименьшее время исполнения – 89 рабочих дней а бюджет затрат, составляющий 159000,84 рублей.

4. Социальная ответственность

Целью описываемой работы является разработка программного приложения или компонента, с которым будет впоследствии взаимодействовать пользователь. В данной разработке значение имеет обеспечение производственной и экологической безопасности на рабочем месте пользователя.

В разделе представлены: описание производственной, экологической безопасности на рабочем месте. Рассматриваются возможные чрезвычайные ситуации и способы предупреждения их возникновения, а так же различного рода опасные и вредные факторы в процессе разработки и непосредственной эксплуатации программного приложения.

Первым этапом обеспечения безопасности как разработчика, так и пользователя является выявление вышеуказанных факторов и чрезвычайных ситуаций, что позволяет определить необходимые меры предосторожности. Опасными или вредными факторами могут являться отклонение температуры воздуха от нормы, недостаточная освещенность, высокий уровень статического электричества, повышенный уровень шума в рабочем помещении.

Немаловажным этапом является обеспечения безопасности является составление перечня организационных и правовых мероприятий и их последующее внедрение. Обеспечение безопасного рабочего места с благоприятными условиями и соблюдение перечня мер предосторожности позволит минимизировать воздействие опасных и вредных факторов и избежать возникновения чрезвычайных ситуаций.

4.1 Производственная безопасность

Пользователь подвергается воздействию работая с персональным компьютером рядом вредных и опасных производственных факторов. Далее представлены факторы и рекомендации по их предотвращению и устранению, документы, содержащие такие рекомендации и требования.

4.1.1 Анализ выявленных вредных факторов при разработке эксплуатации проектируемого решения

Таблица 24 – Опасные и вредные факторы при выполнении работ с автоматизированной беспроводной системой измерений параметров окружающей среды

Источник фактора, наименование видов работ	Факторы (по ГОСТ 12.0.003–2015)		Нормативные документы
	Вредные	Опасные	
Веб-приложение, позволяющее выполнять поиск данных из социальной сети по одному хэштегу используя латентно семантический анализ	<ol style="list-style-type: none"> 1. Повышенная напряженность электромагнитного поля; 2. Недостаток естественного освещения; 3. Недостаточная освещенность рабочей зоны; 4. Повышенный уровень и другие неблагоприятные характеристики шума; 5. Неоптимальный микроклимат помещения. 	<ol style="list-style-type: none"> 1. Электрический ток. 	<p>СанПиН 2.2.4.548–96. Гигиенические требования к микроклимату производственных помещений [11].</p> <p>СанПиН 2.2.4.1191–03. Электромагнитные поля в производственных условиях [12].</p> <p>СанПиН 2.2.2/2.4.1340–03. Гигиенические требования к персональным электронно–вычислительным машинам и организации работы [13].</p> <p>ГОСТ 12.1.045–84 ССБТ. Электростатические поля. Допустимые уровни на рабочих местах и требования к проведению контроля [14].</p> <p>ГОСТ 12.2.032–78 ССБТ. «Рабочее место при выполнении работ сидя. Общие эргономические требования» [15].</p> <p>Требования по пожарной безопасности устанавливаются ГОСТ 12.1.004–91 [16].</p>

4.1.1.1 Отклонения показателей микроклимата рабочей зоны

Достаточно значимое влияние на производительность труда работника, и на его здоровье оказывает температура воздуха на рабочем помещении. Вредными факторами могут быть как пониженная, так и повышенная температура. Повышенная температура способна привести сотрудника к: утомляемости, вялости, к головокружениям, обморокам или к обострению существующих заболеваний. Пониженная температура воздуха способна повлечь возникновение различные простудные заболевания.

Согласно СанПиН 2.2.4.548–96 «Гигиенические требования к микроклимату производственных помещений», показатели микроклимата на рабочих местах должны соответствовать значениям, представленным в таблице 15 (допустимые значения), однако более комфортны для работы условия, соответствующие оптимальным значениям, представленным в таблице 25 [11].

Таблица 25 – Оптимальные значения показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работы	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	1а	22 – 24	60–40	0,1
Теплый	1а	23 – 25	60–40	0,1

Таблица 26 – Допустимые значения показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работы	Температура воздуха, °С		Относительная влажность воздуха, %	Скорость движения воздуха, м/с	
		диапазон ниже оптимальных величин	диапазон выше оптимальных величин		для диапазона температур воздуха ниже оптимальных величин, не более	для диапазона температур воздуха выше оптимальных величин, не более
Холодный	1а	20,0–21,9	24,1–25,0	15–75	0,1	0,1
Теплый	1а	21,0–22,9	25,1–28,0	15–75	0,1	0,2

Для получения допустимой температуры на рабочем месте способствует регулирование системы отопления или кондиционирования, также могут быть использованы такие дополнительные устройства.

4.1.1.2 Повышенный уровень электромагнитных излучений

Согласно СанПиН 2.2.4.1191–03 «Электромагнитные поля в производственных условиях», 8 часовой рабочий день для сотрудника на своем рабочем месте, с предельно допустимым уровнем напряженности должен составлять не более 8 кА/м, а уровень магнитной индукции – 10 мТл. Соблюдение данных норм дает возможность избежать негативного воздействия электромагнитных излучений [12].

Исследования по влиянию электромагнитных излучений на человеческий организм показали, что у трудящихся или живущих вблизи источников сильного излучения ухудшается самочувствие, повышается вероятность возникновения онкологических заболеваний

4.1.1.3 Недостаточная освещенность рабочей зоны

Естественное и искусственное освещение рабочего места оказывает влияние на физическое состояние и на работу сотрудника. Не надлежащего качество освещения ведет к ухудшению зрения работника.

Согласно СанПиН 2.2.2/2.4.1340–03, на поверхности рабочего стола освещенность пользователя ПЭВМ должна быть 300 – 500 лк. При освещении блики должны отсутствовать на поверхности экрана. Поверхность экрана должна быть до 300 лк. Так же, существуют общие требования и рекомендации к организации освещения на рабочем месте, например:

- Система общего равномерного должно регулировать искусственное освещение в помещениях для эксплуатации ПЭВМ.
- Рабочие места следует размещать таким образом, чтобы естественный свет падал преимущественно слева, а дисплеи монитора были ориентированы боковой стороной к световым проемам; [13].

Соблюдение данных мер позволит сохранить зрение работника или избежать пагубного воздействия на глаза.

4.1.1.4 Повышенный уровень шума на рабочем месте

Ненормированный показатель шума на рабочих местах оказывает влияние на психологическое состояние работника. У сотрудника на поставленной ему задаче снижаются концентрация и сосредоточенность, а повышается уровень утомляемости и стресса. Так же повышенный уровень шума может привести к нарушению слуха или являться помехой для коммуникаций между сотрудниками.

Для избегания вышеуказанных последствий воздействия описываемого фактора, необходимо соблюдать следующие требования, обозначенные в СанПиН 2.2.2/2.4.1340–03 «Гигиенические требования к персональным электронно–вычислительным машинам и организации работы», согласно которым на рабочих местах в помещениях для размещения ЭВМ уровень шума не должен превышать 50 дБА. [12]

4.1.2 Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения

4.1.2.1 Высокий уровень статического электричества в рабочем помещении

Головные боли, нарушения сна, чрезмерная раздражительность и эмоциональность у работника может быть вызвана повышенным уровнем статического электричества. Основной опасностью статического электричества является возникновения быстрого искрового разряда между частями электрооборудования. Искровой разряд способен привести к выходу оборудования из строя и к возникновению пожара, получение электрических травм у работника.

ГОСТ 12.1.045–84 ССБТ «Электростатические поля. Допустимые уровни на рабочих местах и требования к проведению контроля» регламентирует значения показателей электростатических полей и уровня напряженности. Согласно этому регламенту, предельно допустимый уровень напряженности электростатических полей устанавливается равным 60 кВ/м в течение 1 часа [14].

Способом снижения статического электричества является заземление коммуникаций и оборудования, на которых могут появиться заряды статического электричества. Увеличение уровня влажности воздуха помещения является одним из действенных способов снижения статического напряжения.

4.1.2.2 Высокий уровень напряжения в электросети

Опасным фактором в электросети является высокий уровень напряжения, так как существует вероятность короткого замыкания, что, а это в свою очередь, может повлечь за собой возникновение пожара. Так же опасность представляет поражение током, в результате которого человек может получить ряд травм таких как ожоги, механические повреждения и многие другие.

Сетевой фильтр или же стабилизатор напряжения, которые позволят защитить от скачков напряжения может помочь избежать подобных ситуаций. Непосредственные источники электропитания, которые требуется обозначить специальными знаками так же требуют осторожности.

4.2 Экологическая безопасность

Существует много факторов, которые негативно влияют на экологию при взаимодействии пользователя с разработанным программным продуктом. Причина тому эксплуатация персонального компьютера так как он является потребителем электроэнергии, а также его утилизация.

Однако персональный компьютер, подключенный к сети электроснабжения, непосредственно эксплуатируется пользователем не все время подключения. В данном случае имеет место неоправданное потребление электроэнергии.

Неправильно произведенная утилизация ПЭВМ так же представляет собой немалую опасность.

При производстве ПЭВМ используются такие материалы как пластик и металл, а их утилизацией должны заниматься специализированные службы. Первоначально организация списывает ПЭВМ.

Процесс разборки делится ПЭВМ на составляющие части; передача этих частей на переработку и аффинаж (процесс получения металлов путем отделения их от примесей).

4.3 Безопасность в чрезвычайных ситуациях

Чрезвычайными ситуациями в рассматриваемых помещениях могут быть пожары. Требования по пожарной безопасности устанавливаются ГОСТ 12.1.004–91 [16].

Пожар – это неконтролируемый процесс горения за пределами специализированного очага, наносящий вред жизни и здоровью людей, их имуществу, интересам общества и государства.

При работе с любыми электроприборами, в том числе персональными компьютерами, чрезвычайной ситуацией является возникновение пожара.

Для минимизации или предотвращения возникновения чрезвычайной ситуации или же возможного ущерба в случае ее возникновения, необходимо соблюдать правила противопожарной безопасности.

Основные причины возникновения пожара:

- включение электроприбора в неисправную розетку;
- нарушение требований пожарной безопасности при эксплуатации электроприборов;
- нарушение правил проведения электрогазосварочных и огневых работ;
- случайный или умышленный поджог;
- пренебрежение опасностью, незнание и недооценка возможных последствий пожара;
- курение в помещении.

Чтобы предотвратить или снизить риск возникновения пожара, требуется соблюдать правила пожарной безопасности и правила содержания

здания, где располагается рабочие места пользователей. Помещения должны содержаться в чистоте. Лестничные клетки, коридоры, двери эвакуационных выходов и подходы к средствам тушения должны быть свободны и не загромождены. Мебель не должна препятствовать быстрой эвакуации людей. Электрические кабели должны быть в состоянии, исключающие поражение электрическим током

В регламенте должны содержаться запреты на выполнение следующих действий:

- эксплуатация провода электроприборов с поврежденной изоляцией;
- использование поврежденных розеток, рубильников, вилок и прочего электрооборудования;
- обертывание светильников, бытовых приборов бумагой, тканью и другими горючими материалами;
- применение открытого огня;
- курение в помещении;
- использование неисправной или незаземленной аппаратуры.

4.4 Правовые и организационные вопросы обеспечения безопасности

Специальные правовые нормы трудового законодательства Государственный надзор и контроль в организациях независимо от организационно–правовых форм и форм собственности осуществляют специально уполномоченные на то государственные органы и инспекции в соответствии с федеральными законами. Согласно трудовому кодексу РФ:

– продолжительность рабочего дня не должна превышать 40 часов в неделю;

– во время регламентированных перерывов целесообразно выполнять комплексы упражнений и осуществлять проветривание помещения [16].

Существуют также специализированные органы, осуществляющие государственный контроль и надзор в организациях на предмет соблюдения существующих правил и норм.

К таким органам относятся:

- Федеральная инспекция труда;
- Государственная экспертиза условий труда Федеральной службы по труду и занятости населения;
- Федеральная служба по надзору в сфере защиты прав потребителей и благополучия человека и др.

4.4.2 Организация рабочего места

Разработанное программное обеспечение которое эксплуатируется на рабочем месте пользователя, организовывается в соответствии с требованиями документов: ГОСТ 12.2.032–78 ССБТ «Рабочее место при выполнении работ сидя. Общие эргономические требования» и СанПиН 2.2.2/2.4.1340–03 «Гигиенические требования к персональным электронно–вычислительным машинам и организации работы». В этих документах содержатся требования к организации непосредственной рабочей зоны и помещения в котором эта зона располагается.

Выполнение в вышеуказанных документах требований обеспечит необходимые условия работы за персональным компьютером, минимизирует влияние опасных и вредных факторов, что дает возможность минимизировать или исключить получение сотрудником производственных травм. Основные общие требования к организации рабочих мест пользователей электронно–вычислительных машин, согласно СанПиН 2.2.2/2.4.1340–03 представлены далее.

При размещении рабочих мест, расстояние между рабочими столами с видеомониторами, должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов – не менее 1,2 м.

- Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600 – 700 мм, но не ближе 500 мм с учетом размеров алфавитноцифровых знаков и символов.

- Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования.

Поверхность рабочего стола должна иметь коэффициент отражения 0,5 – 0,7.

- Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе с компьютером. Рабочий стул (кресло) должен быть подъемно–поворотным, регулируемым по высоте и углам наклона сиденья и спинки.

- Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений [15].

ЗАКЛЮЧЕНИЕ

Социальная сеть сегодня является мощным инструментом для поиска и распространения информации в сети интернет. Поиск информации с веб-приложением в социальных сетях по одному хештегу с использованием латентно–семантического анализа может быть полезным при решении таких задач как (маркетинговые исследования, формирование политических прогнозов, поиск пользователей, компетентных в заданной предметной области и т.д.). Несмотря на непрозрачность и трудоемкость ЛСА, он может весьма успешно применяться для задач, где важно понять семантику сообщения, расширить или обобщить «смыслы» поискового запроса.

Результатом выпускной квалификационной работы стало веб-приложение для поиска данных в социальных сетях по заданной предметной области с использованием латентно–семантического анализа.

В ходе выполнения работы была спроектирована архитектура приложения, разработано веб–приложение, построенное на основе шаблона проектирования MVC, осуществляющее получение записей из сети Twitter по одному хештегу. Были реализованы следующие виды латентно семантического анализа для вычисления метрик «слово–слово», «слово–документ», «группа слов – документ».

Разработан веб интерфейс приложения, который позволяет интуитивно сформировать запрос и получить подробную и легко воспринимаемую информацию о проведенном анализе.

Результаты работы компонента могут быть использованы как обычными пользователями, так и организациями, так как приложение позволяет отказаться от жесткой привязки лексической единицы к какому–либо из кластеров, а с другой представить целостную систему связей между словами.

CONCLUSION

Social network today is a powerful tool for searching and dissemination of information on the Internet. Search information from the web application in social media, one hashtag using latent semantic analysis are able to be useful for solving the heterogeneous task. Despite the opacity and complexity of LSA, it can be very successfully use for applications where it is important to capture the semantics of the message, to extend or generalize the "meanings" of the search query.

The result of the final qualifying work was a component that searches on one hashtag in the social network Twitter, using latent semantic analysis.

In the course of work was designed the application architecture, developed the web application based on the MVC design pattern, those receiving entries from Twitter one hashtag. Was implemented using latent semantic analysis to compute metrics "word–word", "word document", "a group of words – document".

Developed a web interface application that allows you to intuitively query and receive detailed and easy–to–understand information on the conducted analysis.

Both ordinary users and organizations can use the results of the work of the component, as the app allows you to refuse a rigid binding of lexical units to any of the clusters, and on the other to provide a coherent system of relations between words.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Использование #хештегов в социальных сетях [Электронный ресурс] – URL: <https://spark.ru>, свободный. Яз. Рус. Дата обращения 19.12.2016.
2. ТОП 5 инструментов для подбора и анализа хэш-тегов [Электронный ресурс] – URL: <https://habrahabr.ru/post/292974>, свободный. Яз. Рус. Дата обращения 19.12.2016.
3. Три «мощных» хештега для использования в соцсетях <https://msdn.microsoft.com/ru-ru/ff380144.aspx>, свободный. Яз. Рус. Дата обращения 19.12.2016.
4. Создание микросервисов. – СПб.: Питер, 2016. – 304 с.: ил. – (Серия «Бест-селлеры O'Reilly»).
5. Tweetinvi, the best Twitter C# library for the REST and Stream API [Электронный ресурс] – URL: <https://github.com/linvi/tweetinvi>, свободный. Яз. ENG. Дата обращения 21.12.2016.
6. Авторизация по протоколу OAuth на примере Desktop Twitter-клиента [Электронный ресурс] – URL: <https://habrahabr.ru/post/86846/>, свободный. Яз. РУС. Дата обращения 21.12.2016.
7. Send secure authorized requests API [Электронный ресурс] – URL: <https://dev.twitter.com/oauth>, свободный. Яз. ENG. Дата обращения 21.12.2016.
8. An Introduction to Latent Semantic Analysis [Электронный ресурс] – URL: <http://lsa.colorado.edu/papers/dp1.LSAintro.pdf>, свободный. Яз. Англ. Дата обращения 27.04.2017.
9. Использование #хештегов в социальных сетях [Электронный ресурс] – URL: <https://habrahabr.ru/post/110078>, свободный. Яз. ENG. Дата обращения 21.12.2016.
10. Latent Semantic Analysis@ CU Boulder [Электронный ресурс] – URL: <http://lsa.colorado.edu>, свободный. Яз. ENG. Дата обращения 27.04.2017.
11. СанПиН 2.2.4.548–96. Гигиенические требования к микроклимату производственных помещений.

12. СанПиН 2.2.4.1191–03. Электромагнитные поля в производственных условиях.
13. СанПиН 2.2.2/2.4.1340–03. Гигиенические требования к персональным электронно–вычислительным машинам и организации работы.
14. ГОСТ 12.1.045–84 ССБТ. Электростатические поля. Допустимые уровни на рабочих местах и требования к проведению контроля.
15. ГОСТ 12.2.032–78 ССБТ. «Рабочее место при выполнении работ сидя. Общие эргономические требования»
16. ГОСТ 12.1.004–91 ССБТ. Пожарная безопасность. Общие требования. – М.: Изд–во стандартов, 2006. – 67 с.

ПРИЛОЖЕНИЕ А

LSA.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using Accord.Math;
using Accord;
using Accord.Math.Decompositions;
using Accord.Math.Distances;
using System.Threading.Tasks;

namespace TwitterMicroservice.LSA
{
    /// <summary>
    /// Реализует методы ЛСА
    /// </summary>
    public class LSA
    {
        private List<string> _stopWords;
        private ISimilarity<double[]> _metric;

        /// <summary>
        /// Возвращает список слов из документов
        /// </summary>
        /// <param name="returnUniqueWords">Возвращать только уникальные слова, которые
        встречаются минимум 2 раза в документе/ах</param>
        /// <param name="documents">Список документов</param>
        /// <returns></returns>
        private List<string> getAllWords(bool returnUniqueWords, params KeyValuePair<long,
string>[] documents)
        {
            List<string> result = new List<string>();
            List<string> rawWords = new List<string>();
            char[] separators = new char[] { ' ', ',', '.', '!', '?', ':', ';', '-', '|', '#', '"', "\"' };

            Parallel.ForEach(documents, document =>
            {
                var words = document.Value
                    .Split(separators, StringSplitOptions.RemoveEmptyEntries)
                    .Where(p => !_stopWords.Contains(p.ToLower())) //исключили стоп слова
                    .ToList()
                    .ConvertAll(p =>
                    {
                        if (p.Length > 2)
                        {
                            p = p.Substring(0, p.Length - 2); //удаляем окончания – приводим слова к
основной форме
                        }
                        return p.ToLower();
                    });
            });
        }
    }
}
```

```

    });

    lock (rawWords)
    {
        rawWords.AddRange(words);
    }
});

if (returnUniqueWords)
{
    //оставляем только уникальные слова, которых минимум 2 штуки в коллекции
    foreach (var group in rawWords.GroupBy(p => p))
    {
        if (group.Count() > 1)
        {
            result.Add(group.Key);
        }
    }
}
else
{
    result.AddRange(rawWords);
}

return result;
}

public LSA(ISimilarity<double[]> metric)
{
    if (metric is null)
    {
        throw new ArgumentNullException("metric");
    }
    _metric = metric;
    _stopWords = new List<string>()
    {
        "a",
        "about",
        "all",
        "am",
        "an",
        "and",
        "any",
        "are",
        "as",
        "at",
        "be",
        "been",
        "but",
        "by",
        "can",
        "could",

```

"do",
"for",
"from",
"has",
"have",
"I",
"if",
"in",
"is",
"it",
"me",
"my",
"no",
"not",
"of",
"on",
"one",
"or",
"so",
"that",
"the",
"them",
"there",
"they",
"this",
"to",
"was",
"we",
"what",
"which",
"will",
"with",
"would",
"you",
"a",
"будем",
"будет",
"будете",
"будешь",
"буду",
"будут",
"будучи",
"будь",
"будьте",
"бы",
"был",
"была",
"были",
"было",
"быть",
"в",
"вам",

"вами",
"вас",
"весь",
"во",
"вот",
"все",
"всё",
"всего",
"всей",
"всем",
"всём",
"всеми",
"всему",
"всех",
"всею",
"всез",
"всю",
"вся",
"вы",
"да",
"для",
"до",
"его",
"едим",
"едят",
"ее",
"её",
"ей",
"ел",
"ела",
"ем",
"ему",
"емь",
"если",
"ест",
"есть",
"ешь",
"еще",
"ещё",
"ею",
"же",
"за",
"и",
"из",
"или",
"им",
"ими",
"имь",
"их",
"к",
"как",
"кем",

"ко",
"когда",
"кого",
"ком",
"кому",
"комья",
"которая",
"которого",
"которое",
"которой",
"котором",
"которому",
"которою",
"которую",
"которые",
"который",
"которым",
"которыми",
"которых",
"кто",
"меня",
"мне",
"мною",
"мною",
"мог",
"моги",
"можете",
"могла",
"могли",
"могло",
"могу",
"могут",
"мое",
"моё",
"моего",
"моей",
"мое",
"моём",
"моему",
"моею",
"можем",
"может",
"можете",
"можешь",
"мои",
"мой",
"моим",
"моими",
"моих",
"мочь",
"мою",
"моя",

"мы",
"на",
"нам",
"нами",
"нас",
"наш",
"наша",
"наше",
"нашего",
"нашей",
"нашем",
"нашему",
"нашею",
"наши",
"нашим",
"нашими",
"наших",
"нашу",
"не",
"него",
"нее",
"неё",
"ней",
"нем",
"нём",
"нему",
"нет",
"нею",
"ним",
"ними",
"них",
"но",
"о",
"об",
"один",
"одна",
"одни",
"одним",
"одними",
"одних",
"одно",
"одного",
"одной",
"одном",
"одному",
"одною",
"одну",
"он",
"она",
"оне",
"они",
"оно",

"от",
"по",
"при",
"с",
"сам",
"сама",
"сами",
"самим",
"самими",
"самих",
"само",
"самого",
"самом",
"самому",
"саму",
"свое",
"своё",
"своего",
"своей",
"своем",
"своём",
"своему",
"своею",
"свои",
"свой",
"своим",
"своими",
"своих",
"свою",
"своя",
"себе",
"себя",
"собой",
"собою",
"та",
"так",
"такая",
"такие",
"таким",
"такими",
"таких",
"такого",
"такое",
"такой",
"таком",
"такому",
"такою",
"такую",
"те",
"тебе",
"тебя",
"тем",

```

    "теми",
    "тех",
    "то",
    "тобой",
    "тобою",
    "того",
    "той",
    "только",
    "том",
    "томах",
    "тому",
    "тот",
    "тою",
    "ту",
    "ты",
    "у",
    "уже",
    "чего",
    "чем",
    "чём",
    "чему",
    "что",
    "чтобы",
    "эта",
    "эти",
    "этим",
    "этими",
    "этих",
    "это",
    "этого",
    "этой",
    "этом",
    "этому",
    "этот",
    "этою",
    "эту",
    "я"
};
}

/// <summary>
/// Возвращает словарь ID документа – Близость по теме к ключевому слову (от 0 до
1, чем больше – тем ближе)
/// </summary>
/// <param name="documents">Словарь документов ID документа – Текст
документа</param>
/// <param name="keyWord">Ключевое слово, по которому выполняется поиск
близких по тематике документов</param>
/// <returns></returns>
public Dictionary<long, double> AnalyzeWordToDocument(Dictionary<long, string>
documents, string keyWord)
{

```

```

if (documents is null)
{
    throw new ArgumentNullException("documents");
}
if (string.IsNullOrEmpty(keyWord) || string.IsNullOrWhiteSpace(keyWord))
{
    throw new ArgumentException("keyWord");
}
var words = getAllWords(true, documents.ToArray());
int m = words.Count;
int n = documents.Count;
double[,] matrixWordDocument = new double[m, n]; //исходная матрица слова –
документы
for (int i = 0; i < m; i++)
{
    for (int j = 0; j < n; j++)
    {
        var wordsDoc = getAllWords(false, documents.ElementAt(j));
        int countWordInDocument = 0;
        foreach (var word in wordsDoc)
        {
            if (words.Contains(word))
            {
                countWordInDocument++;
            }
        }
        matrixWordDocument[i, j] = countWordInDocument;
    }
}

//выполнили SVD разложение
SingularValueDecomposition svd = new
SingularValueDecomposition(matrixWordDocument);
var u = svd.LeftSingularVectors; //слова – темы
var v = svd.RightSingularVectors; //темы – документы

#region нашли ключевое слово в выборке (индекс вектора слова в матрице u)

    keyWord = keyWord.Length > 2 ? keyWord.Substring(0, keyWord.Length -
2).ToLower() : keyWord.ToLower();
    int indexKeyWord = words.FindIndex(p => p == keyWord);
    if (indexKeyWord == -1)
    {
        throw new InvalidOperationException("Невозможно вычислить расстояние между
документами, ключевое слово не найдено в массиве слов из документов");
    }
}
#endregion

#region построили вектор ключевого слова

double[] vectorKeyWord = u.GetRow(indexKeyWord);
#endregion

```

```

Dictionary<long, double> result = new Dictionary<long, double>();

for (int i = 0; i < v.GetLength(1); i++)
{
    double[] vectorDocument = v.GetColumn(i);
    double similarity = _metric.Similarity(vectorKeyWord, vectorDocument);
    long idDocument = documents.ElementAt(i).Key;

    result.Add(idDocument, similarity);
}

return result;
}
/// <summary>
/// Возвращает словарь ID документа – Близость по теме к группе слов, близких по
смыслу к ключевому слову (от 0 до 1, чем больше – тем ближе)
/// </summary>
/// <param name="documents">Словарь документов ID документа – Текст
документа</param>
/// <param name="keyWord">Ключевое слово, по которому выполняется поиск
близких по тематике слов, а по найденной группе – документов</param>
/// <param name="countSimilarWord">Количество искомых похожих слов</param>
/// <returns></returns>
public Dictionary<long, double> AnalyzeWordsToDocument(Dictionary<long, string>
documents, string keyWord, int countSimilarWord)
{
    if (documents is null)
    {
        throw new ArgumentNullException("documents");
    }
    if (string.IsNullOrEmpty(keyWord) || string.IsNullOrWhiteSpace(keyWord))
    {
        throw new ArgumentException("keyWord");
    }
    if (countSimilarWord < 1)
    {
        throw new ArgumentException("countSimilarWord", "Минимально допустимое
значение: 1");
    }
    var words = getAllWords(true, documents.ToArray());
    int m = words.Count;
    int n = documents.Count;
    double[,] matrixWordDocument = new double[m, n]; //исходная матрица слова –
документы
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            var wordsDoc = getAllWords(false, documents.ElementAt(j));
            int countWordInDocument = 0;
            foreach (var word in wordsDoc)

```

```

        {
            if (words.Contains(word))
            {
                countWordInDocument++;
            }
        }
        matrixWordDocument[i, j] = countWordInDocument;
    }
}

//выполнили SVD разложение
SingularValueDecomposition svd = new
SingularValueDecomposition(matrixWordDocument);
var u = svd.LeftSingularVectors; //слова – темы
var v = svd.RightSingularVectors; //темы – документы

#region нашли ключевое слово в выборке (индекс вектора слова в матрице u)

    keyWord = keyWord.Length > 2 ? keyWord.Substring(0, keyWord.Length -
2).ToLower() : keyWord.ToLower();
    int indexKeyWord = words.FindIndex(p => p == keyWord);
    if (indexKeyWord == -1)
    {
        throw new InvalidOperationException("Невозможно вычислить расстояние между
документами, ключевое слово не найдено в массиве слов из документов");
    }
}
#endregion

#region построили вектор ключевого слова

    double[] vectorKeyWord = u.GetRow(indexKeyWord);
}
#endregion

#region Получили слова, похожие по смыслу на ключевое слово в контексте
документов

    List<(int indexWord, double similar)> similarWords = new List<(int indexWord, double
similar)>();
    for (int i = 0; i < u.GetLength(0); i++)
    {
        double[] vectorWord = u.GetRow(i);
        double similarity = _metric.Similarity(vectorKeyWord, vectorWord);

        similarWords.Add((i, similarity));
    }

    similarWords = similarWords
        .OrderByDescending(p => p.similar)
        .Take(countSimilarWord)
        .ToList();
}
#endregion

```

```

Dictionary<long, double> result = new Dictionary<long, double>();

for (int i = 0; i < v.GetLength(1); i++)
{
    double[] vectorDocument = v.GetColumn(i);
    double similarityDocumentToWords = 0;

    for (int j = 0; j < similarWords.Count; j++)
    {
        double[] vectorSimilarWord = u.GetRow(similarWords[j].indexWord);
        double similarity = _metric.Similarity(vectorSimilarWord, vectorDocument);
        similarityDocumentToWords += similarity;
    }
    result.Add(documents.ElementAt(i).Key, similarityDocumentToWords);
}

return result;
}
}}
}

```

ПРИЛОЖЕНИЕ Б

Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ									Исполнители			Длительность работ в рабочих днях <i>T_{pi}</i>			Длительность работ в календарных днях <i>T_{ki}</i>		
	<i>t_{min}</i> , чел–дни			<i>t_{max}</i> , чел–дни			<i>t_{ожі}</i> , ел–чдни											
Составление и утверждение задания	2	3	3	3	4	4	2,4	3,4	3,4	1	1	1	2,4	3,4	3,4	4	5	5
Поиск и изучение материалов по теме	14	14	14	20	20	20	16,4	16,4	16,4	2	2	2	8,2	8,2	8,2	12	12	12
Анализ источников и представленной в них информации	7	7	7	10	10	10	8,2	8,2	8,2	1	1	1	8,2	8,2	8,2	12	12	12
Выбор направления исследований	4	4	4	6	6	6	4,8	4,8	4,8	2	2	2	2,4	2,4	2,4	4	4	4
Календарное планирование работ по теме	1	1	1	2	2	2	1,4	1,4	1,4	1	1	1	1,4	1,4	1,4	2	2	2
Проектирование архитектуры	5	5	5	8	8	8	6,2	6,2	6,2	1	1	1	6,2	6,2	6,2	9	9	9
Выбор технологий разработки	1	1	1	2	2	2	1,4	1,4	1,4	1	1	1	1,4	1,4	1,4	2	2	2
Выбор подходов к решению поставленной задачи	4	4	4	6	6	6	4,8	4,8	4,8	2	2	2	2,4	2,4	2,4	4	4	4
Выбор средства обеспечения хранения данных	1	1	1	2	2	2	1,4	1,4	1,4	1	1	1	1,4	1,4	1,4	2	2	2
Реализация компонента получения данных из социальной сети	6	7	6	8	9	8	6,8	7,8	6,8	1	1	1	6,8	7,8	6,8	10	12	10

Реализация компонента обработки и анализа данных	16	22	19	22	28	25	18,4	24,4	21,4	1	1	1	18,4	24,4	21,4	27	36	32
Реализация компонента визуализации данных	7	10	9	10	12	11	8,2	10,8	9,8	1	1	1	8,2	10,8	9,8	12	16	15
Реализация обеспечения хранения данных в базе	6	8	7	7	10	9	6,4	8,8	7,8	1	1	1	6,4	8,8	7,8	10	13	12
Тестирование и исправление ошибок работы приложения	7	11	9	10	14	12	8,2	12,2	10,2	1	1	1	8,2	12,2	10,2	12	18	15
Оценка соответствия программного приложения заявленным требованиям к нему	2	2	2	4	4	4	2,8	2,8	2,8	2	2	2	1,4	1,4	1,4	2	2	2
Оформление пояснительной записки	10	10	10	14	14	14	11,6	11,6	11,6	2	2	2	5,8	5,8	5,8	9	9	9
Итого							109,4	126,4	118,4				89	106	98	133	158	147

ПРИЛОЖЕНИЕ В

Календарный план–график проведения научно–исследовательских работ

№ работ	Вид работ	Исполнитель	Т. к. / кал. / дн.	Продолжительность выполнения работ													
				февр.			март			апрель			май			июнь	
				1	2	3	1	2	3	1	2	3	1	2	3	1	2
1	Составление и утверждение задания	Р	5	1													
2	Поиск и изучение материалов по теме	С, Р	12														
3	Анализ источников и представленной в них информации	С	12														
4	Выбор направления исследований	С, Р	4														
5	Календарное планирование работ по теме	С	2														
6		С	9														

	Проектирование архитектуры																	
7	Выбор технологий разработки	С	2					■										
8	Выбор подходов к решению поставленной задачи	С, Р	4					■ ✍										
9	Выбор средства обеспечения хранения данных	С	2					■										

