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

Школа _____ИШИТР_____
Направление подготовки ___09.04.01 Информатика и вычислительная техника_____
Отделение школы (НОЦ) ___Отделение информационных технологий_____

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Определение отношений между пользователями социальной сети Twitter на основе анализа текста сообщений

УДК __004.773.6:316.472:004.5.-047.44_____

Студент

Группа	**ФИО**	**Подпись**	**Дата**
8ВМ6Г	Шаяхметов Бекзат Мейрамбайулы		

Руководитель

Должность	**ФИО**	**Ученая степень, звание**	**Подпись**	**Дата**
Доцент	Цапко Ирина Валериевна	Кандидат технических наук		

КОНСУЛЬТАНТЫ:
По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	**ФИО**	**Ученая степень, звание**	**Подпись**	**Дата**
Доцент	Рыжакина Татьяна Гавриловна	Кандидат экономических наук		

По разделу «Социальная ответственность»

Должность	**ФИО**	**Ученая степень, звание**	**Подпись**	**Дата**
Ассистент	Авдеева Ирина Ивановна	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	**ФИО**	**Ученая степень, звание**	**Подпись**	**Дата**
Доцент	Кочегурова Елена Алексеевна	Кандидат технических наук		

Томск – 2018 г.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Код результатов	Результат обучения (выпускник должен быть готов)
	Общепрофессиональные компетенции
Р1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
Р2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
Р3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
Р4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
	Профессиональные компетенции
Р5	Выполнять инновационные инженерные проекты по разработке аппаратных и программных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
Р6	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы.
Р7	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации аппаратных и программных средств автоматизированных систем различного назначения.
	Общекультурные компетенции
Р8	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.
Р9	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.
Р10	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
Р11	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.

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

Школа _____ИШИТР_____
Направление подготовки ___09.04.01 Информатика и вычислительная техника_____
Отделение школы (НОЦ) ___Отделение информационных технологий_____

УТВЕРЖДАЮ:
Руководитель ООП

_____ _____ _____
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

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

Студенту:

Группа	**ФИО**
8ВМ6Г	Шаяхметову Бекзату Мейрамбайулы

Тема работы:

Определение отношений между пользователями социальной сети Twitter на основе анализа текста сообщений	
Утверждена приказом директора (дата, номер)	Приказ №2621/с от 16.04.2018 г.

Срок сдачи студентом выполненной работы:	25.05.2018

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	1. Существующие готовые решения задач анализа естественного языка. 2. Алгоритмы машинного обучения. 3. Методы кодирования слов в векторной форме. 4. Требование организовать сбор тренировочного набора данных. 5. Требование реализовать программный сервис классификации отношений между пользователями социальной сети.

Перечень подлежащих исследованию, проектированию и разработке вопросов	1. Рассмотреть существующие алгоритмы анализа естественного языка. 2. Определить функционал программного сервиса. 3. Спроектировать архитектуру программного сервиса. 4. Реализовать загрузчик сообщений социальной сети. 5. Реализовать алгоритм конкатенации собранных пар сообщений. 6. Реализовать классификатор на основе выбранной архитектуры машинного обучения.
Перечень графического материала	Презентация в формате .pptx на 18 слайдов.

Консультанты по разделам выпускной квалификационной работы

Раздел	Консультант
Финансовый менеджмент, русурсоэффективность и ресурсосбережение	Рыжакина Татьяна Гавриловна
Социальная ответственность	Авдеева Ирина Ивановна

Названия разделов, которые должны быть написаны на русском и иностранном языках:

Аналитический обзор
Проектирование и реализация программного сервиса
Тестирование классификатора
Интерфейс и функциональные возможности

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Цапко Ирина Валериевна	Кандидат технических наук		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ6Г	Шаяхметов Бекзат Мейрамбайулы		

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

Школа_____ИШИТР_____
Направление подготовки ____09.04.01 Информатика и вычислительная техника_____
Уровень образования _____магистратура_____
Отделение школы (НОЦ)___ Отделение информационных технологий _____
Период выполнения _____осенний/весенний семестр 2017/2018 учебного года____

Форма представления работы:

Магистерская диссертация

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

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	

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

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Цапко Ирина Валериевна	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Кочегурова Е.А.	к.т.н.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8ВМ6Г	Шаяхметову Бекзату Мейрамбайулы

Школа	ИШИТР	**Отделение**	Отделение информационных технологий
Уровень образования	Магистратура	**Направление/специальность**	09.04.01 Информатика и вычислительная техника

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих 2. Нормы и нормативы расходования ресурсов 3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Работа с информацией, представленной в российских и иностранных научных публикациях, аналитических материалах, статических бюллетенях и изданиях, нормативно-правовых документах; анкетирование; опрос.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Проведение предпроектного анализа. Определение целевого рынка и проведение его сегментирования. Выполнение SWOT-анализа проекта
2. Определение возможных альтернатив проведения научных исследований	Определение целей и ожиданий, требований проекта. Определение заинтересованных сторон и их ожиданий.
3. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	Составление календарного плана проекта. Определение бюджета НТИ
4. Определение ресурсной, финансовой, экономической эффективности	Проведение оценки экономической эффективности определения отношений между пользователями социальной сети «Twitter» на основе анализа текста сообщений.
Перечень графического материала (с точным указанием обязательных чертежей):	
1. Оценка конкурентоспособности технических решений 2. Матрица SWOT 3. График проведения и бюджет НТИ 4. Расчёт денежного потока 5. Оценка ресурсной, финансовой и экономической эффективности НТИ	

Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Рыжакина Татьяна Гавриловна	Кандидат экономических наук		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ6Г	Шаяхметов Бекзат Мейрамбайулы		

Тема: «Определение отношений между пользователями социальной сети Twitter на основе анализа текста сообщений»
ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	**ФИО**
8ВМ6Г	Шаяхметову Бекзату Мейрамбайулы

Школа	ИШИТР	**Отделение**	Информационных технологий
Уровень образования	Магистратура	**Направление/специальность**	09.04.01 Информатика и вычислительная техника

Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования является разрабатываемый программный комплекс, который позволяет производить автоматизированный анализ сообщений социальной сети Twitter на наличие оттенков согласия или несогласия. Разработка системы происходит в помещениях и требует работы с компьютерами и другими электронными устройствами, которые являются источниками вредных излучений и могут оказывать негативное влияние на здоровье и жизнедеятельность человека.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Производственная безопасность	Возможные вредные факторы в офисном помещении: • Повышенный уровень электромагнитного излучения • Пониженная или повышенная температура воздуха. • Недостаточное или неправильное освещение. • Шум • Психофизические факторы Возможные опасные факторы в офисном помещении: • Короткое замыкание. • Электрический ток. • Статическое электричество.
2. Экологическая безопасность	В процессе разработки и эксплуатации искусственной нейронной сети возможно образование следующих видов отходов: • образование твердых отходов, относящихся к IV классу

	опасности (системный блок компьютера, принтеры, сканеры, клавиатура, манипулятор "мышь") и жидких отходов. • Жидкие отходы: сточные воды. • Люминесцентные лампы.
3. Безопасность в чрезвычайных ситуациях	Наиболее типичная чрезвычайная ситуация при работе в офисе – пожар. Превентивные меры включают инструктаж по пожарной безопасности, контроль состояния проводки и электрических приборов, своевременное профилактическое обслуживание.
4. Правовые и организационные вопросы обеспечения безопасности	Параметры рабочего места офисного работника регулируются ГОСТ 12.2.032–78 ССБТ, СанПиН 2.2.2/2.4.1340–03, «Трудовой кодекс Российской Федерации» от 30.12.2001 №197-ФЗ.

Дата выдачи задания для раздела по линейному графику	**14.03.2018**

Задание выдал консультант:

Должность	**ФИО**	**Ученая степень, звание**	**Подпись**	**Дата**
Ассистент	Авдеева Ирина Ивановна	-		**14.03.2018**

Задание принял к исполнению студент:

Группа	**ФИО**	**Подпись**	**Дата**
8ВМ6Г	Шаяхметов Бекзат Мейрамбайулы		**14.03.2018**

РЕФЕРАТ

Выпускная квалификационная работа содержит 138 страниц, 50 рисунков, 30 таблиц, 2 приложения, и 40 использованных литературных источников.

Ключевые слова: нейронная сеть, классификатор, векторное представление слов, сети долгосрочной памяти, рекуррентные нейронные сети, согласие/несогласие, социальная сеть Twitter.

Объектом исследования являются различные подходы и архитектуры машинного обучения для решения задач анализа естественного языка.

Цель работы: реализация классификатора отношений между пользователями социальной сети на основе анализа текста сообщений с последующей визуализации в виде графа.

Работа представлена введением, 6 разделами (главами) и заключением, приведен список использованных литературных источников.

В результате проделанной работы был реализован алгоритм классификации, программный сервис для автоматизации процесса классификации, а также был собран тренировочный набор данных в виде пар сообщений из социальной сети «Twitter».

В ходе работы была решена проблема подачи двух сообщений на вход нейронной сети путем специального алгоритма конкатенации. Реализованный в ходе работы классификатор обладает точностью классификации равной 82-85% – тестирование проводилось на тестовых данных, не входящих в тренировочный набор.

В будущем планируется провести ряд изменений в реализованном классификаторе: улучшить точность классификатора путем увеличения тренировочного набора данных, добавить третий класс классификации – нейтральный.

СОКРАЩЕНИЯ

ИНС – искусственные нейронные сети

РНС – рекуррентные нейронные сети

LSTM – long-short term memory

RNN – recurrent neural network

ANN – artificial neural networks

AI – artificial intelligence

СУБД – система управления базами данных

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Каждый день во всемирной паутине появляется огромное количество контента: проводятся опросы, высказываются мнения, разгораются споры. Рост различных дискуссионных форумов и социальных сетей открыло людям новые методы для выражения своих мнений. В дискуссиях зачастую участники соглашаются или не соглашаются с мнениями других участников. Порой в таких дискуссиях стоит задача классифицировать отношение между пользователями социальной сети на основе анализа сообщений для дальнейшего проведения каких-либо социологических опросов, маркетинговых исследовании, выявить наличие споров, идеологических позиции участников, но проводить анализ огромных данных вручную часто очень сложно и затратно. Для автоматизации этой задачи, связанной с анализом текста, используются различные методы анализа естественного языка.

Одной из самых распространённых задач является анализ сентимента предложений, то есть оценка эмоционального тона текста. Тем не менее, задачи анализа сообщений на наличие оттенков согласия/несогласия недостаточно распространены. Данная проблема освящены в работах [1, 2] англоговорящих авторов. В данных работах [1, 2] проделан только краткий обзор методов анализа сообщений на наличие оттенков (не)согласия, однако реализация этих методов не продемонстрирована.

По причине того, что сказывается не проработанность способов программной реализаций задач классификаций отношений между пользователями социальных сетей на основе анализа их сообщений, появляется сложность в создании тренировочных данных для глубокого машинного обучения. Классификация сообщений на наличие оттенков согласия или несогласия в целом не является тривиальной задачей. Стандартные методы классификаций – метод Байеса и метод опорных векторов – не в состоянии «понять» смысл цитаты и его комментария, чтобы

определить является ли комментарий (не)согласием на предыдущую цитату. Эти методы не учитывают порядок слов в предложениях.

Эту проблему решают методы машинного обучения, а точнее нейронные сети. Они позволяют алгоритмически понимать структуру предложений и как слова взаимосвязаны между собой. Задача нейронных сетей состоит не в понимании каждого слова, а скорее в понимании последовательности этих слов.

Целью настоящей магистерской работы является реализация классификатора отношений между пользователями социальной сети Twitter на основе анализа их сообщений.

Для достижения текущей цели нынешней работы необходимо решение следующих задач:

1. выбор архитектуры нейронной сети для решения задач анализа естественного языка;
2. адаптация выбранной архитектуры для решения задач выявления оттенков согласия и несогласия в сообщениях;
3. создание обучающей выборки данных;
4. проверка точности классификатора;
5. разработка веб-приложения для работы с классификатором.

1. АНАЛИТИЧЕСКИЙ ОБЗОР

С повышением аппаратных мощностей возрастают требования к полученным результатам классификаторов на основе нейронных сетей. Ценность программных систем с более точным конечным результатом имеют большую ценность нежели системы, основной особенностью которых является скорость выполнения.

Существуют различные методы анализа/классификации естественного языка, к основным можно отнести следующие:

1. Наивный классификатор Байеса.
2. Метод опорных векторов (support vector machine).
3. Нейронные сети глубокого машинного обучения.
    - Свёрточные нейронные сети.
    - Рекуррентные нейронные сети.
4. Дерево принятия решений.

Согласно работе [3], методы наивного классификатора Байеса и опорных векторов не в состоянии учитывать порядок слов в тексте. Данные два подхода классификации естественного языка используют метод «корзины слов» для представления текстовой информации в числовой форме, а это значит, что в таких методах структура предложений и порядок слов никак не учитываются.

В работах [3, 4, 5] утверждается, что для анализа сообщений могут быть использованы классификаторы на основе искусственных нейронных сетей. Популярными архитектурами для анализа последовательности слов являются сверточные нейронные сети (CNN) и нейронные сети долгосрочной памяти (LSTM). Также в работе [4] говорится, что использование методов глубинного обучения дает точность в диапазоне 75%-95% в зависимости от использованных наборов тренировочных данных, что является основным приоритетом в наше время.

Отличительной особенностью рекуррентных нейронных сетей архитектур LSTM от двух других является то, что этот метод способен работать с входными данными нефиксированной длины, а также тот факт, что в отличии от методов наивного классификатора Байеса и опорных векторов, где слова в предложениях анализируются по отдельности, в нейронных сетях архитектур LSTM происходит анализ последовательности слов.

В работе авторов [6] продемонстрирован процесс поэтапного анализа предложений нейронной сетью архитектуры LSTM, что в свою очередь говорит о том, что сети данной архитектуры учитывают природу входных данных: их анализ производится последовательно, точно так же, как человек читает и анализирует информацию. На рисунке 1 продемонстрирован процесс последовательного анализа элементов естественного языка. С каждым последующим словом коэффициент эмоционального тона изменяется в пределах от 0 до 1.

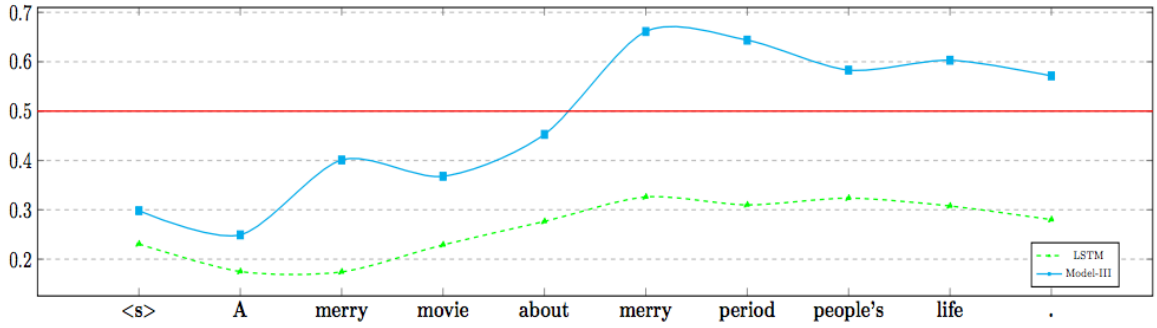


Рисунок №1 –процесс последовательной обработки предложения архитектурой LSTM [6]

По словам авторов работы [6], искусственная нейронная сеть архитектуры LSTM является самой популярной архитектурой, используемой в анализе естественного языка, так как эти нейронные сети работают с входными данными нефиксированной длины, а также в состоянии алгоритмически понимать структуру предложений в силу своего рекуррентного строения.

1.1. Рекуррентные нейронные сети (РНС)

Известно, что, читая какую-либо литературу, человек понимает каждое слово опираясь на понимание предыдущего. Нейронные сети классической архитектуры (однослойная нейронная сеть прямого распространения) не в состоянии воспроизвести эту архитектуру и это, возможно, их самый главный недостаток [7]. Однако, рекуррентные нейронные сети способны решить эту проблему.

Нейронные сети рекуррентной архитектуры — это сети со встроенным циклом внутри, что позволяет сохранять информацию, полученную на предыдущей итерации, не выбрасывая ее (рисунок 2).

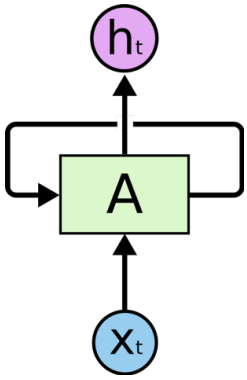


Рисунок №2 – строение РНС [7]

На рисунке №2 продемонстрирована часть нейронной сети «А», принимающая на вход входные данные «x_t» и подающие на выход значение «h_t». Цикл позволяет проходить информации от одного шага нейронной сети к другому. РНС можно считать копиями одной и той же сети, каждая передающая сообщение последующей (рисунок 3).

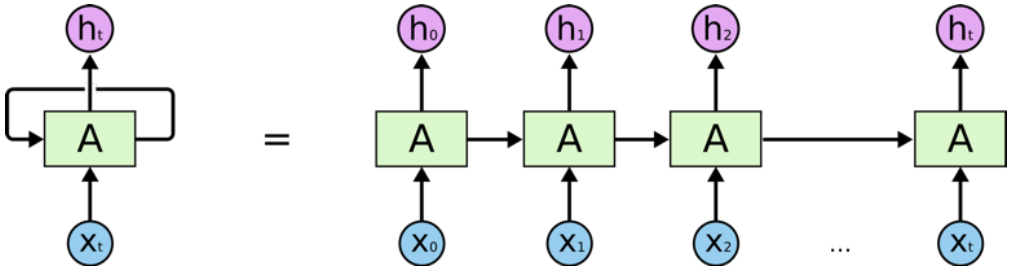


Рисунок №3 – РНС изнутри [7]

Эта цепь-образная структура говорит о том, что рекуррентные нейронные сети тесно связаны с последовательностями и списками.

1.1.1. Сети долгосрочной памяти (LSTM)

Сети долгосрочной памяти (LSTM сети) – особый вид РНС, способные обучаться долгосрочным зависимостям. Они были представлены Сеппом Хохрайтером и Юргеном Шмидтхубером в 1997 году и были усовершенствованы и популяризированы многими людьми в последующих работах. LSTM сети широко применяются в решении многих современных проблем. [7]

Сети долгосрочной памяти изначально были сконструированы для решения проблем долгосрочной зависимости. Их основная задача состоит в запоминании информации на протяжении длительного времени.

Все рекуррентные нейронные сети имеют форму цепи, состоящей из повторяющихся модулей нейронной сети. В стандартной РНС этот повторяющийся модуль имеет простейшую структуру в виде слоя с функцией активации «tanh» (гиперболический тангенс) (рисунок №4). [7]

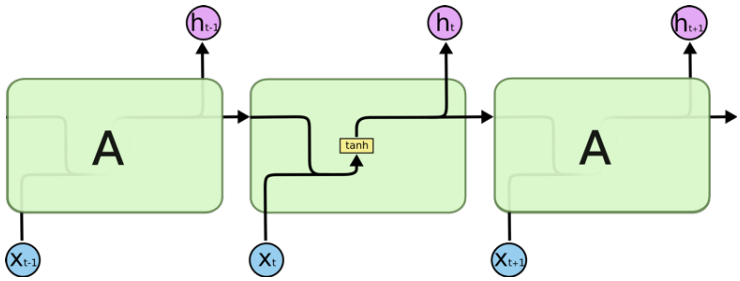


Рисунок №4 – повторяющийся модуль в стандартной РНС [7]

LSTM сети также имеют форму цепи с повторяющимися модулями, но в отличии от стандартной рекуррентной нейронной сети, LSTM сети имеют немного другую структуру – вместо 1 слоя с функцией активации «tanh», сети долгосрочной памяти имеют целых 4 слоя, взаимодействующие между собой особым образом (рисунок №5) [7].

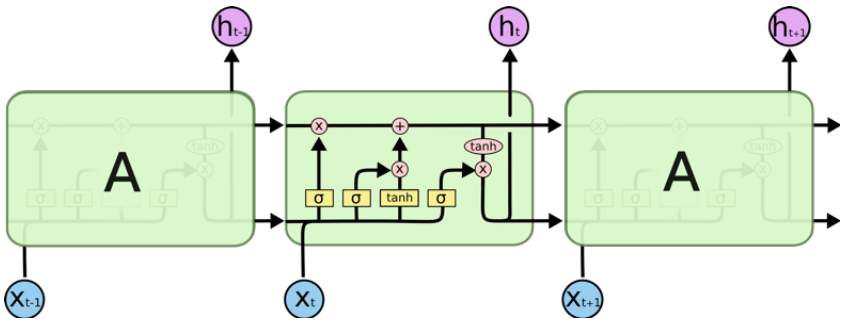


Рисунок №5 – повторяющийся модуль в LSTM сетях [7]

1.2. Методы кодирования текстовых данных

Искусственные нейронные сети не работают с текстовым представлением данных – они работают только с числовыми данными. Существует различное количество методов представления текстовых данных в числовой (векторной) форме:

- корзина слов (bag of words);
- one-hot vectors;
- векторное представление (word embeddings).

Суть корзины слов состоит в представлении всего текстового документа в векторной форме. Создаётся словарь из корпуса интересуемых нас текстов:

$$A = [A_1, A_2, \ldots, A_n]$$

Где A_n – уникальное слово из корпуса текстов.

В итоге результатом представления конечного предложения в числовой форме является вектор B:

$$B = [B_1, B_2, \ldots, B_m]$$

Где m – длина словаря A,

B_m – количество повторении элемента A_n в конечном предложений.

К примеру, имеется корпус данных «It was the best of times. It was the worst of times. It was the age of wisdom. It was the age of foolishness». Нужно сделать из данного набора предложений словарь уникальных слов без учета знаков пунктуации. Словарь будет выглядеть следующим образом:

D = [«It», «was», «the», «best», «of», «times», «worst», «age», «wisdom», «foolishness»]

В конечном итоге, используя данный словарь можно представить другие предложения в векторной форме. Допустим предложение «*It was the best game ever*». Сопоставляя словарь D с конечным предложением получается вектор V следующего вида:

V = [1, 1, 1, 1, 0, 0, 0, 0, 0, 0]

Метода «one-hot» схож с методом корзины слов. Однако данный метод представляет не документ в виде вектора, а каждый элемент документа в векторной форме. К примеру, имеется словарь:

D = [«laptop», «car», «building», «money», «fame», «keyword», «using»]

Следовательно, слово «money» будет выглядеть следующим образом:

M = [0, 0, 0, 1, 0, 0, 0]

Каждый элемент предложения состоит из векторов с нулевыми элементами, построенные по принципу вектора «M». Длина вектора равна длине словаря.

В итоге, предложение рода «I like using my laptop in the car» используя метод «one-hot» выглядит следующим образом:

sentence = [
[0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 1], [0, 0, 0, 0, 0, 0, 0], [1, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0, 0]
]

Векторное представление слов (word embeddings) – это обобщенное название для различных подходов к моделированию языка и обучению представлений в обработке естественного языка, где слова и фразы из словаря сопоставляются с вектором действительных чисел [22]. Задача векторного представления слов заключается в нахождении вектора для слов и их контекста в словаре для удовлетворения некоторого заранее обозначенного критерия (например, для предугадывания рядом стоящих слов).

Одной из отличительных особенностей векторного представления (word embeddings) заключается в его относительной малоразмерности (размерность варьируется от 50 до 600), в то время как корзина слов и one-hot являются высокоразмерными (размерность может достигать до 100 000), так как при построении векторов слов размерностью вектора является длина словаря. Также векторное представление слов имеет способность обобщать смысловой оттенок слова используя семантическую близость схожих векторов, что является невозможным в подходе one-hot vector или bag-of-words.

1.3. Задача классификации отношений между пользователями

Задача классификации отношений является бинарной классификацией. На вход бинарному классификатору подается последовательность a = {a_1, a_2, …, a_i} данных произвольной длины, состоящая из сообщений пользователя А и пользователя Б, соответственно выходными значениями является бинарное множество $B = \{0;\ 1\}$.

Задача классификации может быть решена методами машинного обучения с учителем, которые используют предварительно классифицированные наборы тренировочных данных для обучения модели нейронной сети.

Для создания тренировочного набора данных требуется участие эксперта и вспомогательного программного обеспечения для сбора и анализа данных. Обучающая пара представлена в виде последовательных текстовых данных (пар сообщений пользователя А и Б) и соответственно их классов (согласие или несогласие).

2. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО СЕРВИСА

Создание программного сервиса включает в себя уточнение и детализация требований (определение групп пользователей и соответствующих им функциональных возможностей), проектирование архитектуры (создание компонентной архитектуры и детализация микроархитектуры отдельных микрокомпонентов) и интерфейса пользователя (создание макета интерфейса пользователя) с последующей их реализацией.

Предполагается использование программного сервиса двумя категориями пользователей: посетитель и администратор. Администратор имеет те же права что и пользователь, но дополнительно имеет возможность управлять состоянием нейронной сети: запускать процесс повторной тренировки, и изменять параметры.

2.1. Функциональные требования

Пользователь программной сервиса должен иметь доступ к сервису с любого устройства, имеющего браузер и выход в интернет. Пользователь сервиса имеет возможность:

- Производить поиск пар сообщений и комментарий социальной сети Twitter при помощи ключевые слов с последующей классификацией и визуализацией отношений между авторами данных сообщений в виде графа.
- Выполнять анализ собственных данных, введённых вручную.

Пользователь в праве запросить набор любых сообщений используя ключевые слова. После подтверждения пользователем ключевых слов, осуществляется поиск сообщений социальной сети. По окончанию процесса поиска, найденные сообщения должны подаваться на вход нейронной сети для дальнейшей их классификации.

После процесса классификации этих сообщений, программный сервис должен вывести пользователей и их отношения друг к другу в виде графа. Отношение пользователей друг к другу определяется нейронной сетью.

Также пользователь может удостовериться в точности результатов нейронной сети: сервис помимо графа и класса отношений пользователей должна вывести их сообщения.

Дополнительным функционалом является возможность анализа собственно введенных пар сообщений (сообщения и комментария к нему). То есть каждый пользователь может вручную ввести два сообщения, после чего сервис произведет классификацию данных сообщений используя компонент нейронной сети. Ответ сервиса должен быть на подобии «*the message has (dis)agreement sentiment*» в зависимости от результатов классификатора.

Также программный сервис нуждается в двухуровневом доступе для обеспечения безопасности: пользователь и администратор. На любой странице веб-приложения гость имеет возможность авторизоваться (войти под своей учетной записью), а также иметь обратную возможность –закончить сессию. Гость, авторизовавшийся на сайте, имеет возможность изменять параметры нейронной сети и обучать нейронную сеть по новой.

2.2. Архитектура программного сервиса

Одной из основных задач нынешней работы является реализация программного сервиса для классификации отношений между пользователями социальной сети Twitter на основе анализа пар сообщений. Как видно на рисунке №6, создаваемый программный сервис состоит из четырех компонентов:

- Компонент веб-приложение состоит из пользовательского веб-интерфейса, для взаимодействия пользователя с сервисом в браузере.
- Компонент СУБД – это программное обеспечение на сервере, хранящее и предоставляющее данные по запросу пользователя.

- Компонент классификатор – основной компонент, классифицирующий отношение между пользователями.
- Компонент загрузчик сообщений – компонент, предназначенный для загрузки пар сообщений из социальной сети Twitter.

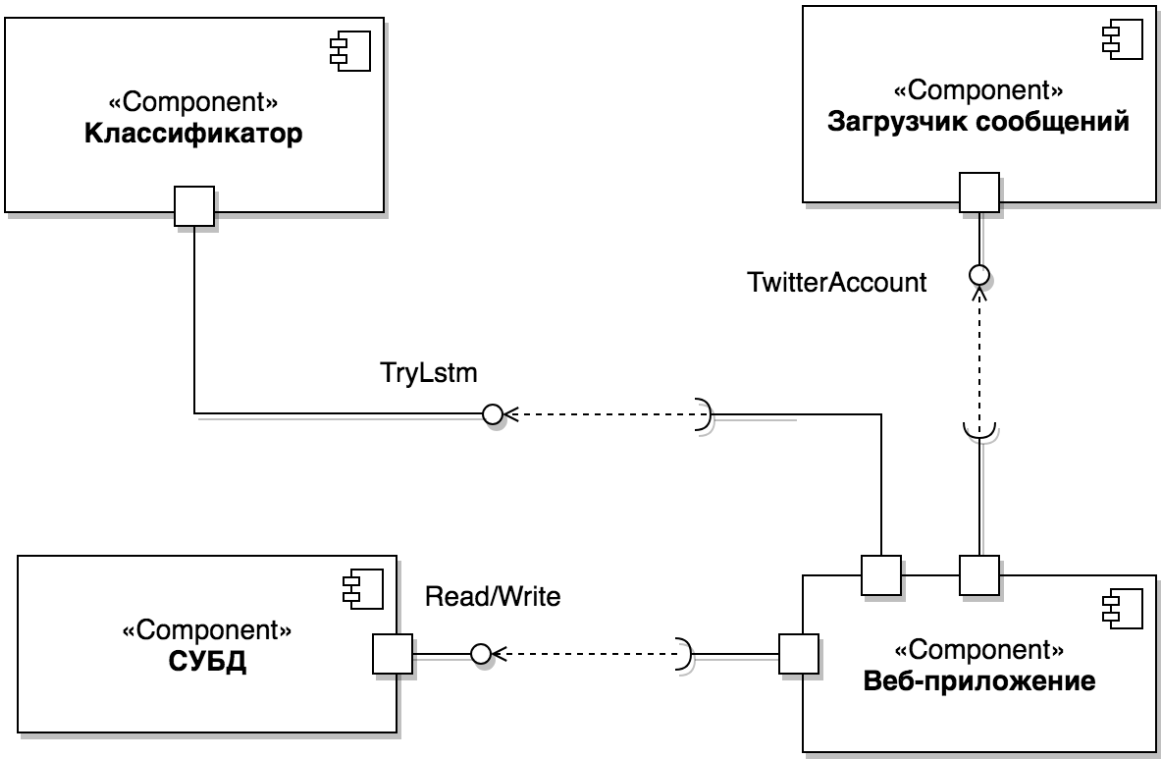


Рисунок №6 – компонентная диаграмма программного сервиса

2.2.1. Компонент веб-приложение

По причине того, что основная библиотека машинного обучения TensorFlow написана на высокоуровневом языке программирования Python, для клиент серверного взаимодействия целесообразнее всего выбрать веб-фреймворк «Django», поддерживающий языка Python.

Согласно источнику [8], одними из основных преимуществ «Django» являются:

- **Скорость разработки**: фреймворк был разработан таким образом, чтобы помочь разработчикам сделать приложение как можно быстрее.

Начиная от идеи, производства, заканчивая релизом продукта, «Django» помогает сделать этот процесс эффективным и экономичным.

- **Полностью укомплектованный**: «Django» включает в стандартный набор решении, которые помогают с аутентификацией пользователя, картами сайтов, администрированием контента, RSS-каналами и так далее.
- **Безопасность**: Фреймворк самостоятельно контролирует процесс обеспечения безопасности.
- **Расширяемость: «**Django» может удовлетворить самые нагруженные запросы на сторону сервера.
- **Многосторонность:** Управление контентом, научные вычислительные платформы и даже поддержка крупных организации – все эти аспекты очень эффективно управляются с помощью веб-фреймворка «Django».

Нынешнее веб-приложение реализовано при помощи веб-фреймворка Django по шаблону проектирования MTV (model-template-view), являющийся прямым аналогом шаблона MVC. MVC (model-view-controller) – шаблон проектирования, который предполагает разделение логики на три отдельных компонента: модель, представление, контроллер – таким образом, что модификация каждого из компонентов проходит независимо от двух других.

2.2.1.1. Клиент-серверная часть

Клиентская и серверная части, отвечающая за взаимодействия клиент-сервера, состоит из следующих классов (рисунок №7):

1) Базовый класс «View» предназначен для обеспечения связей между пользователем и сервисом: контролирует ввод данных пользователем используя модель и представление для реализации необходимой реакции;

2) Класс «URL» предназначен для обработки запросов пользователя, для перенаправления к соответствующим контроллерам;

3) Класс «Model» является базовым классом в системе «Django», обеспечивающий построение сущности взаимодействующих с системами управления базами данных;

4) Папка с файлами «template» предназначена для формирования визуализируемых объектов;

5) Класс «TryLSTMView» основной задачей которого является обработка запросов от клиентской части, и их перенаправление на компонент нейронной сети для последующего анализа.

6) Класс «TrainPageView» нужен для работы со страницей настройки нейронной сети. Авторизованный пользователь может перетренировать модели нейронной сети.

7) Основной задачей класса «TweetSearch» является обработка запросов пользователя и их перенаправление компоненту поиска сообщений социальной сети Twitter.

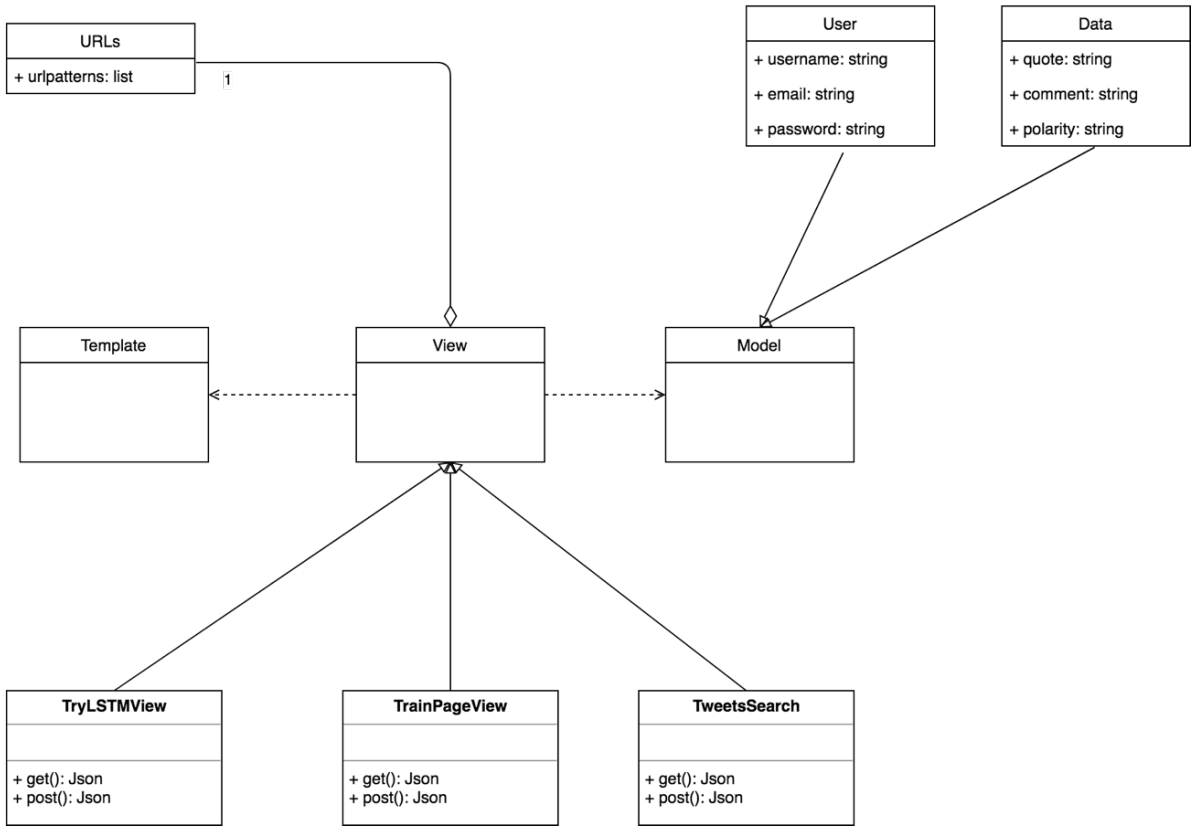


Рисунок №7 – диаграмма классов компонента клиент-сервер

2.2.2. Компонент загрузчик данных

Прежде чем проектировать и реализовывать алгоритм нейронной сети глубокого обучения требуется подготовить набор тренировочных данных. Для того, чтобы собрать данные из социальной сети Twitter в формате сообщение-комментарий необходимо реализовать программный компонент для взаимодействия с программным интерфейсом социальной сети посредствам API.

Существуют строгие ограничения в количестве выгружаемых данных за раз и количестве запросов в пятнадцатиминутном окне, поэтому для корректной работы с базой данных социальной сети Twitter нужно изучить ограничения API [9].

На официальном веб-сайте социальной сети Twitter имеется документация по работе с API [9]. Данный ресурс утверждает, что перед сбором данных требуется авторизовать свое приложение в системе. Для этого следует перейти в раздел «Application management» (управление приложением) и зарегистрировать свое приложение (рисунок №8). После регистрации приложения, разработчику становятся доступны индивидуальные ключи и токены доступа к программному интерфейсу социальной сети.

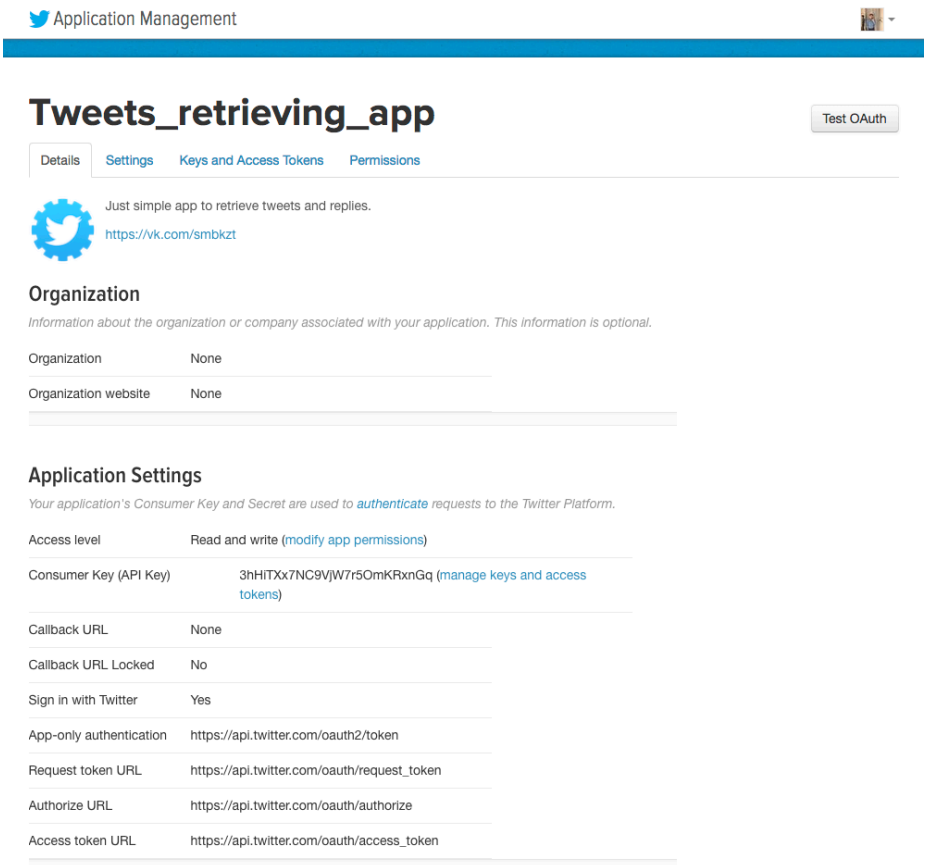


Рисунок №8 – Процесс регистрации приложения в системе

Одним из основных требований к сервису является разработка компонента поиска сообщений социальной сети Twitter. Для реализации данного требования в языке программирования Python имеется расширяющая функционал дополнительная библиотека «Tweepy» [10]. Для того, чтобы установить данную библиотеку в окружение языка Python можно использовать стандартный сборщик проектов «pip»:

```
pip install tweepy
```

Как сказано в официальной документации [10], Tweepy – это Python библиотека, позволяющая с легкостью получать данные из социальной сети Twitter. «Tweepy» имеет в своем функционале готовые методы поиска, сортировки сообщений используя API социальной сети Twitter, а также «стриминг сообщений» (Streaming API).

Streaming API – это инструмент для получения публичных данных из социальной сети. Streaming API нужен всем, кто занимается изучением данных из соцмедиа. Это могут быть научные статистические исследования, анализ

восприятия бренда, проверка эффективности маркетинговой стратегии и многое другое. Главное, что лежит в основе — выборка публичного контента с определенными словами (например, название торговой марки). Для реализации данного программного сервиса Streaming API необходим для непрерывного сбора постоянно появляющейся новейшей информации (сообщений) из социальной сети Twitter.

Компонент поиска сообщений состоит из двух классов (рисунок №9). Первый класс, класс «TwitterAccount», является классом инициализации доступа к социальной сети. В данном классе описан метод доступа к API используя ранее полученные ключи и токены доступа.

```
def get_api(self):
        try:
            self.consumer_key = config.consumer_key
            self.consumer_secret = config.consumer_secret
            self.access_token = config.access_token
            self.access_secret = config.access_secret

            self.auth = OAuthHandler(self.consumer_key,
self.consumer_secret)
            self.auth.set_access_token(self.access_token,
self.access_secret)
            self.api = tweepy.API(self.auth,
wait_on_rate_limit_notify=True)
        except tweepy.TweepError as exception:
            logging.exception(exception)
        return self.api
```

При помощи переменной «api» можно приступать к поиску сообщений. Для этого нужно использовать функцию библиотеки Tweepy – «user_timeline»:

```
user_tweets = self.api.user_timeline(id=self.user,
                                     count=self.num_of_tweets,
                                     pages=10)
```

Функция «user_timeline» выгружает сообщения конкретного пользователя. В аргументы функция получает идентификационный номер желаемого пользователя, количество выгружаемых сообщений за раз, и количество страниц, требуемых для выгрузки.

Второй класс компонента является реализацией Streaming API. Streaming API это инструмент для выгрузки больших объемов данных, который не использует постоянных запросов, что очень удобно для работы с API в условиях жестких ограничениях. Данный класс наследуется напрямую от класса «tweepy.StreamListener», перегружая его функции «on_status» и «on_error».

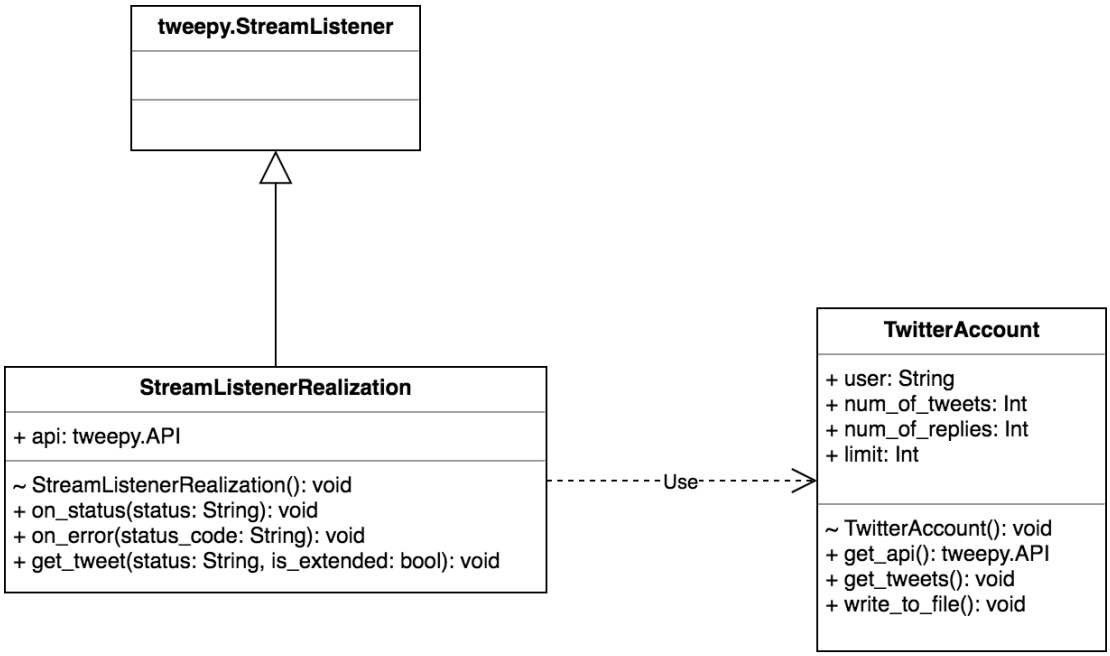


Рисунок №9 – диаграмма классов компонентной части загрузки данных

Создание тренировочного набора данных с применением компонентной части загрузчика данных

Помимо загрузки данных для анализа, компонентная часть загрузчика сообщений была использована для создания тренировочного набора данных. Для выполнения данной задачи предварительно был создан словарь признаков согласия/несогласия для автоматизированного поиска пар сообщений социальной сети Twitter, с последующим присвоением соответствующих классов им классов (согласие/несогласие).

Перед началом процесса создания тренировочного набора, стояла проблема подачи нейронной сети двух сообщений разом. Для решения проблемы одновременной обработки двух сообщений был использован специфичный разделитель вида «< – >». Данный разделитель помещается между двумя сообщениями пользователей (сообщение и комментарии) для корректной их конкатенации, который предположительно должен увеличить точность вычислении нейронной сети.

К примеру, требуется обработать два сообщения: «I consider football as the best sport ever» и комментарий к нему «For me football is not the best game…». Для корректной обработки этих сообщений нейронной сетью, следует произвести их конкатенацию с помощью разделителя: «I consider football as the best sport ever < – > For me football is not the best game…». В итоге конкатенированные сообщения подаются на вход нейронной сети в виде одной последовательности.

В результате работы компонента по сбору данных было получено около 20 000 пар сообщений в формате сообщение-комментарий (рисунок №10). Данные сообщения записаны и отсортированы в соответствующий им текстовый файл: «agreed.polarity», «disagreed.polarity» (рисунок №11).

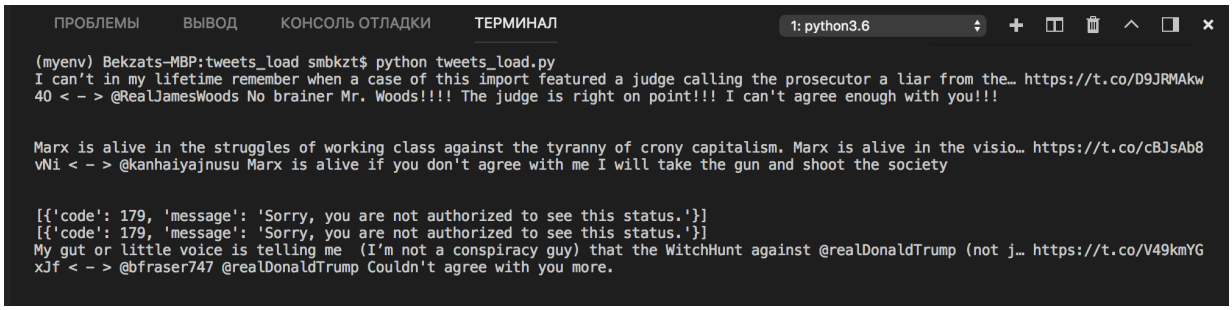


Рисунок 10 – Процесс сбора данных

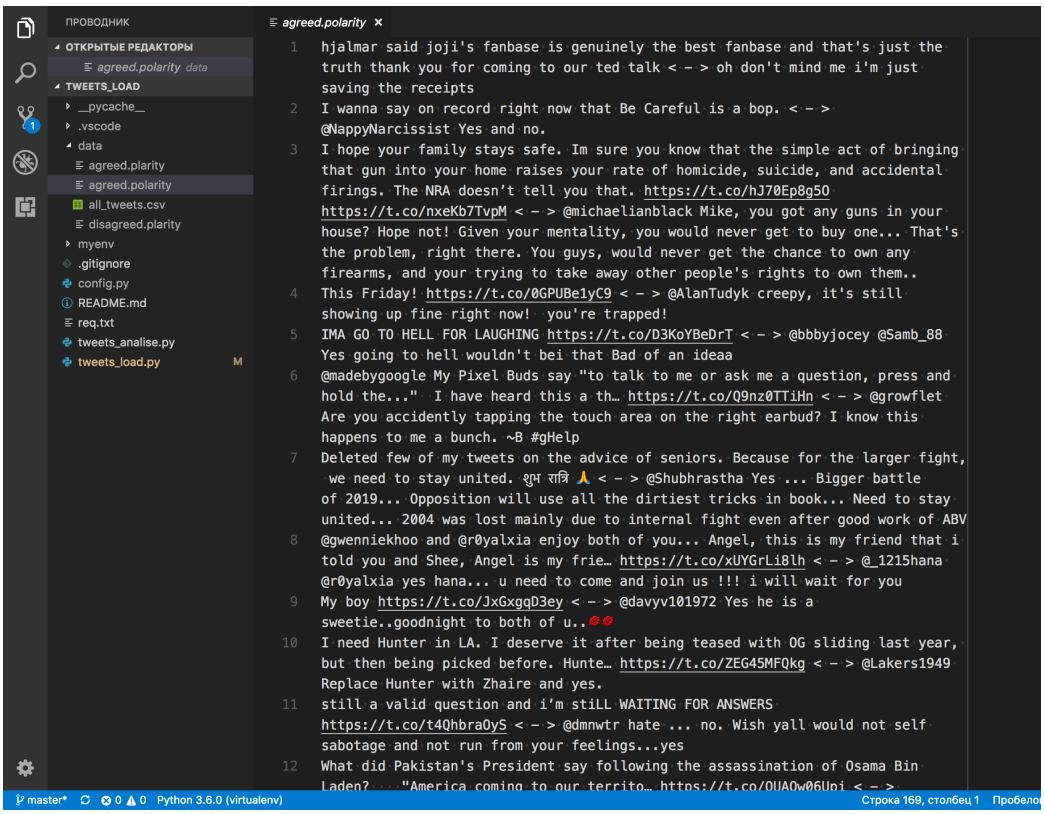


Рисунок №11 – полученные пары сообщений

2.2.3. Компонент классификатор

Для реализации алгоритма нейронной сети необходимо выбрать соответствующую библиотеку для удобной и быстрой разработки. В настоящее время на рынке искусственного интеллекта существует большое количество библиотек машинного обучения, каждая из которых имеет недостатки и преимущества перед своими конкурентными: библиотеки специализируются на конкретных задачах, будь то реализация специфичной архитектуры или поддержка различных технологий.

Сравнение библиотек глубокого обучения

Существует два вида библиотек: символьные и императивные. Символьные фреймворки имеют преимущество перед императивными в возможностях многократного использования памяти, а также в автоматической оптимизации на основе графов зависимостей [11].

Среди популярных библиотек машинного обучения, представленных в таблице №1, значительным преимуществом обладает библиотека от компании Google «TensorFlow», ввиду ее лучшей поддержки нейронных сетей архитектуры RNN (рекуррентных нейронных сетей), простоты использования и поддержка высокоуровневой библиотеки «Keras». Также популярность библиотеки TensorFlow подтверждена статистическими данными использования библиотек машинного обучения на ресурсе «GitHub», продемонстрированная на рисунке №12. Следовательно, целесообразнее всего выбрать библиотеку машинного обучения от компании Google.

Таблица 1 – сравнение библиотек машинного обучения [11].

	Языки	Тренировочные материалы	Поддержка CNN	Поддержка RNN	Лёгкость в исп.	Скорость	Поддержка GPU	Поддержка Keras
Theano	Python, C++	++	++	++	+	++	+	+
TensorFlow	**Python, C++**	**+++**	**+++**	**++**	**+++**	**++**	**++**	**+**
Torch	Lua, Python	+	+++	++	++	+++	+	
Caffe	C++	+	++		+	+	+	
MXNet	R, Python, Julia, Scala	++	++	+	++	++	+++	
CNTK	C++	+	+	+++	+	++	+	

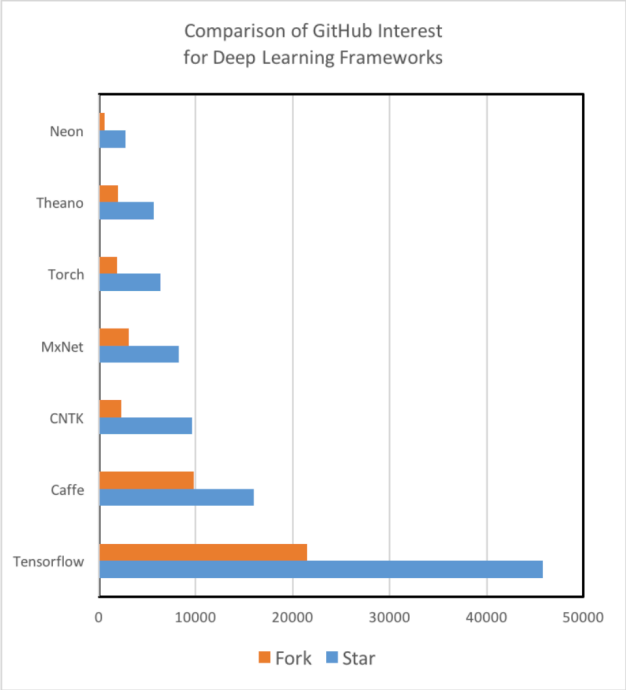


Рисунок №12 – Сравнительный интерес к библиотекам машинного обучения [11]

Как было упомянуто в первой главе (ГЛАВА 1. Аналитический обзор), для реализации сервиса анализа сообщений социальной сети Twitter была выбрана архитектура нейронной сети долгосрочной памяти (LSTM), поскольку LSTM сети зарекомендовали себя в решении проблем анализа естественного языка.

В ходе реализации алгоритма, стояла проблема подачи на вход нейронной сети двух различных сообщений – сообщение пользователя А и ответ от пользователя Б. Для решения данной проблемы было решено использовать специальный разделитель, с помощью которого были разделены сообщения в тренировочном наборе данных (раздел «создание тренировочного набора данных с применением компонентной части загрузчика данных»).

Общий алгоритм процесса классификации отношений между двумя пользователями социальной сети Twitter на основе анализа их сообщений приведен на диаграмме деятельности (рисунок №13)

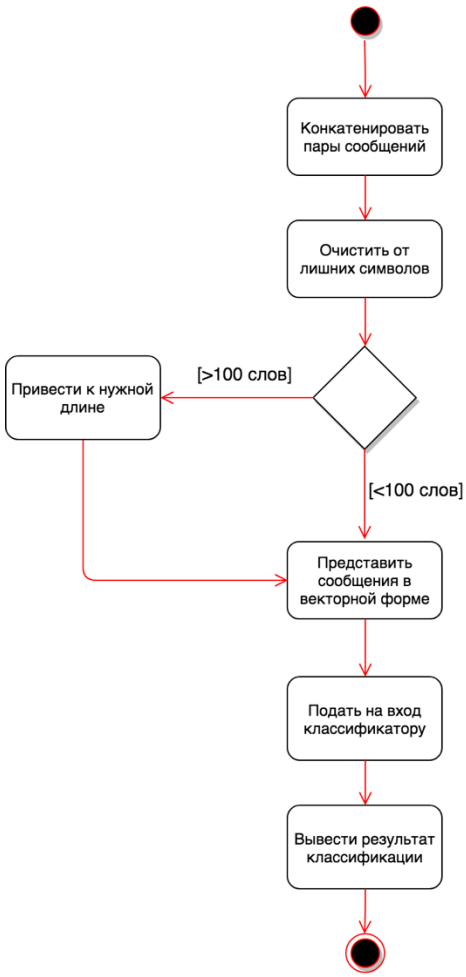


Рисунок №13 – общий алгоритм процесса классификации

В качестве реализации нейронной сети архитектуры LSTM был использован исходный код, полученный из источника [12] с внесением некоторых изменений:

1. Исходный код преобразован в стиль ООП: функционал разделен на классы, произведены наследования классов и инкапсуляция. Данное изменение было выполнено для удобного дальнейшего использования исходного кода как независимого компонента.
2. Добавлен метод для загрузки векторного представления слов «GloVe».
3. Однослойная нейронная LSTM-сеть изменена на двухслойную. Согласно автору работы [13] это повысит устойчивость нейронной сети к обучению.

4. Модифицирован метод очистки данных: метод очистки данных дополнен функцией очистки сообщений от ссылок, лишних пробелов, символов пунктуации, повторений букв (более 2).
5. Добавлен разделитель « < – >» для корректной конкатенации двух сообщений.

Компонент нейронной сети состоит из 3 классов (рисунок №14):

1. Класс «PrepareData»
2. Класс «RnnModel»
3. Класс «TryLSTM»

Класс «PrepareData» реализует основной функционал предварительной подготовки тренировочных данных для последующего обучения нейронной сети на их основе. В подготовительный этап входит процесс создания векторов слов, удаление ненужных символов (знаки препинания и т.д.), создания тренировочных и тестовых наборов данных.

Очистка тренировочных наборов данных от лишних символов дает значительный прирост в аккуратности получаемых результатов, так как при удалении слов от лишних символьных знаков, шансы найти слово в словаре вырастают в разы. К примеру, предыдущее пример «For me football is not the best game…» в процессе анализа будет разделено на составляющие в виде слов. Последнее слово, имеющее троеточие, запишется как «game…». При векторном представлении слов, слово «game…» не будет распознано, по причине несоответствии со словарем. Значимая часть кода, реализующая очистку сообщений от лишних символов, продемонстрирована ниже:

```
@staticmethod
    def clean_string(string: str) -> str:
        """Cleans messages from punctuation and mentions"""
        seperator = " < - > "
        cleaned_string = ''
        cut_sentence_until = int(config.maxSeqLength/2) -
int(len(seperator)/2)

        # Delete the urls
        string = re.sub('https?://[A-Za-z0-9./]+', '', string.lower())
```

```
    # Delete all punctuation marks
    string = string.split(seperator)
    for num, part in enumerate(string, 1):
        for char in part:
            if char not in punctuation:
                cleaned_string += char
        if num == 1:
            cleaned_string += seperator
    # delete repeated whitespaces (more than 2)
    if re.search(r'\s{2,}', cleaned_string):
        cleaned_string = re.sub(r'\s{2,}', " ", cleaned_string)
                                ...
                                ...
                                ...
    return cleaned_string
```

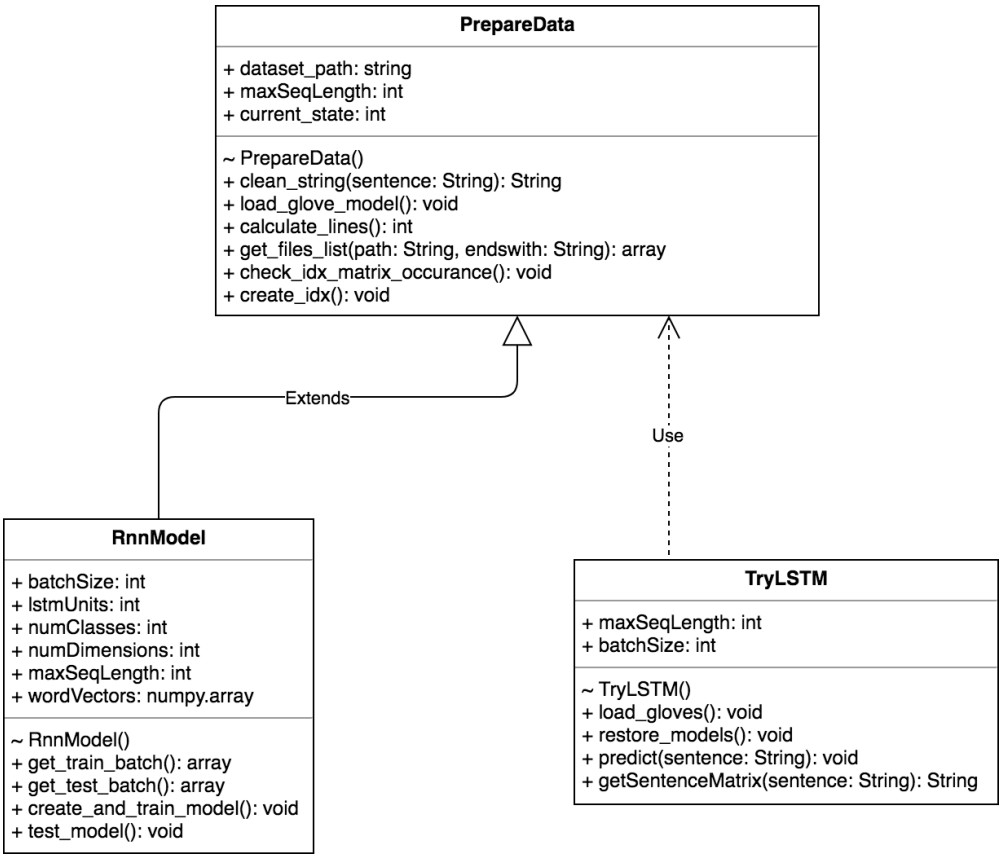


Рисунок №14 – диаграмма классов компонента нейронной сети

Класс «RnnModel» отвечает за реализацию архитектуры LSTM рекуррентных нейронных сетей. Данный класс является ключевым в

компоненте нейронной сети, так как реализует основную архитектуру алгоритма классификации сообщений.

Библиотека машинного обучения TensorFlow представляет из себя понятие «черного ящика»: разработчик использует готовый функционал для реализации различных архитектур машинного обучения, не имея углубленных знаний алгоритмов этих методик. Граф работы алгоритма продемонстрирован на рисунке №15.

Часть алгоритма обучения продемонстрирован в коде ниже:

```
for i in range(config.training_steps+1):
        nextBatch, nextBatchLabels = self.__get_train_batch()
        sess.run(optimizer, {input_data: nextBatch,
                             labels: nextBatchLabels}
                 )
        # Write summary to Tensorboard
        if i % 100 == 0:
            print(f"Iterations: {i}/{config.training_steps}")
            summary = sess.run(merged,
                               {input_data: nextBatch,
                                labels: nextBatchLabels}
                               )
            writer.add_summary(summary, i)
    save_path = f"{log_dir}pretrained_lstm.ckpt"
    saver.save(sess, save_path, global_step=config.training_steps)
    print(f"Model saved to: {save_path}")
    writer.close()
    sess.close()
```

После реализации алгоритма машинного обучения нужно выбрать параметры нейронной сети. Выбор правильных параметров нейронной сети является важным аспектом эффективной тренировки нейронных сетей глубоко обучения. Позже в разделе тестирования можно заметить, что точность и потери, а также скорость обучения во время тренировки нейронной сети могут варьироваться в зависимости от выбора тех или иных параметров: метод оптимизации (Adam, Adadelta, SGD и другие), шаг обучения, количество LSTM блоков, и размерность векторного представления.

РНС известны своей сложностью в тренировке, по причине того, что они имеют большое количество временных шагов (каждое слово в предложениях является временным шагом) [14].

В ходе проделанного исследования в главе 3 (ГЛАВА 3. Тестирование разработки), были выбраны следующие оптимальные параметры нейронной сети.

Таблица 2 – выбранные параметры нейронной сети

	Алгоритм оптимизации	Скорость обучения	Слои	Количество блоков LSTM	Размер выборки	Количество итерации
1.	RMSProp	0.001	2	16	50	2000

Скорость обучения. Шаг обучения становится чрезвычайно важным аспектом, по причине того, что, если выбрать большой шаг, весовые значения будут сильно колебаться, а при медленном значении шага процесс обучения может затянуться. Значение шага в «0.001» является золотой серединой для начала. Если в процессе обучения значение ошибки минимизируется очень медленно, то следует увеличить шаг обучения, и соответственно уменьшить, если ошибка нестабильна [21].

Алгоритм оптимизации. Среди исследователей нейронных сетей нет единого мнения об идеальном алгоритме оптимизации, однако оптимизация по Адаму широко популярна из-за наличия адаптивного уровня обучения [21]. В разделе тестирования приведен подробный разбор каждого алгоритма оптимизации.

Блоки LSTM. Это значение во многом зависит от средней длины входных текстовых данных. Большее количество блоков LSTM обеспечивает модель возможностью хранить больше информации, однако данный подход будет занимать больше времени для обучения и обойдется дорого в вычислительной мощности машины. Так как средняя длина

конкатенированных сообщений пользователей А и Б не превышает 100 слов, то значением размерности блоков LSTM можно выбрать 64 блока.

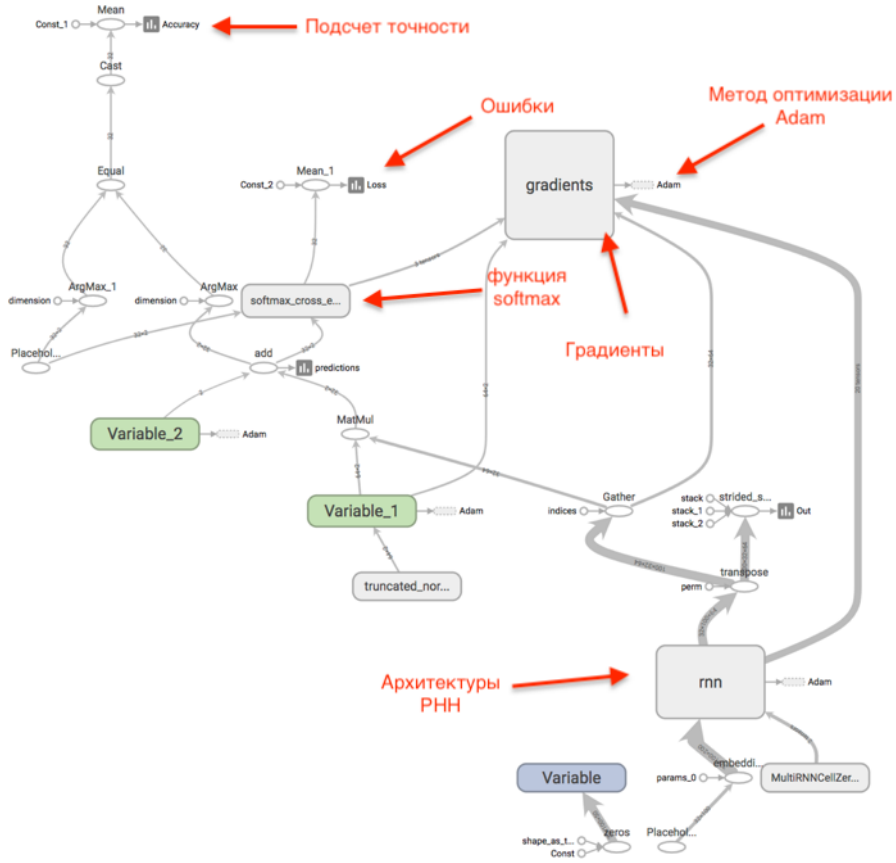


Рисунок №15 – вычислительный граф LSTM

Размерность векторного представления. Размеры для векторного представления слов обычно варьируются от 50 до 300. Больший размер означает, что вектор способен хранить больше информации о слове, в тоже время размер модели на диске будет соответственно выше. Для получения приемлемых характеристик быстродействия была выбрана размерность векторного представления равной 50.

Класс «TryLSTM» организует процесс работы с обученной моделью нейронной сети. Для работы с обученной моделью нейронной сети нужно использовать встроенный функционал библиотеки TensorFlow для восстановления сессий (загрузка обученной модели). Данный процесс осуществляется путем использования функции «restore», после чего сессия восстанавливается на последнем сохраненном моменте:

```
# Restoring the meta and latest model
path = ".".join([tf.train.latest_checkpoint(self.__path),"meta"])
saver = tf.train.import_meta_graph(path)
saver.restore(self.sess, tf.train.latest_checkpoint(self.__path))
```

2.2.4. Компонент СУБД

В качестве системы управления базами данных выбрана СУБД SQLite. По словам разработчика [15], SQLite обладает рядом преимуществ по сравнению с его аналогами:

- Не требуются конфигурации (не нужна предварительная «инсталляция»).
- Не требует сервера для работы с БД.
- Состоит из 1 файла.
- Кросс платформенная база данных (написанная БД на одной машине может быть перенесена на любую другую машину с любой архитектурой).
- Компактность (размер файла может достигать менее 500 Кбайт).
- Поддержка языка SQL.

2.3. Требования к пользовательскому интерфейсу

Интерфейс программного сервиса по классификации отношений между пользователями социальной сети является веб-интерфейсом. Веб-интерфейс должен иметь навигационное меню по сайту для удобного переключения между страницами.

Интерфейс пользователя разделен на 4 основные страницы (рисунок №16). Находясь на любой из страниц можно получить доступ к любой другой странице. Страница настройки нейронной сети доступна только авторизованным пользователям, соответственно доступ к данной странице проходит через страницу авторизации.

Рисунок №16 – структура веб-интерфейса

На первой странице (странице поиска сообщений) интерфейс должен состоять из логотипа, поля для ввода сообщений и кнопки (рисунок №17).

После ввода ключевых слов и подтверждения данных, гость должен увидеть анимацию загрузки. После успешной обработки запроса, анимация загрузки должна исчезнуть и блок с полем для ввода информации и картинкой должна плавно смениться на граф и блок сообщений (рисунок №19).

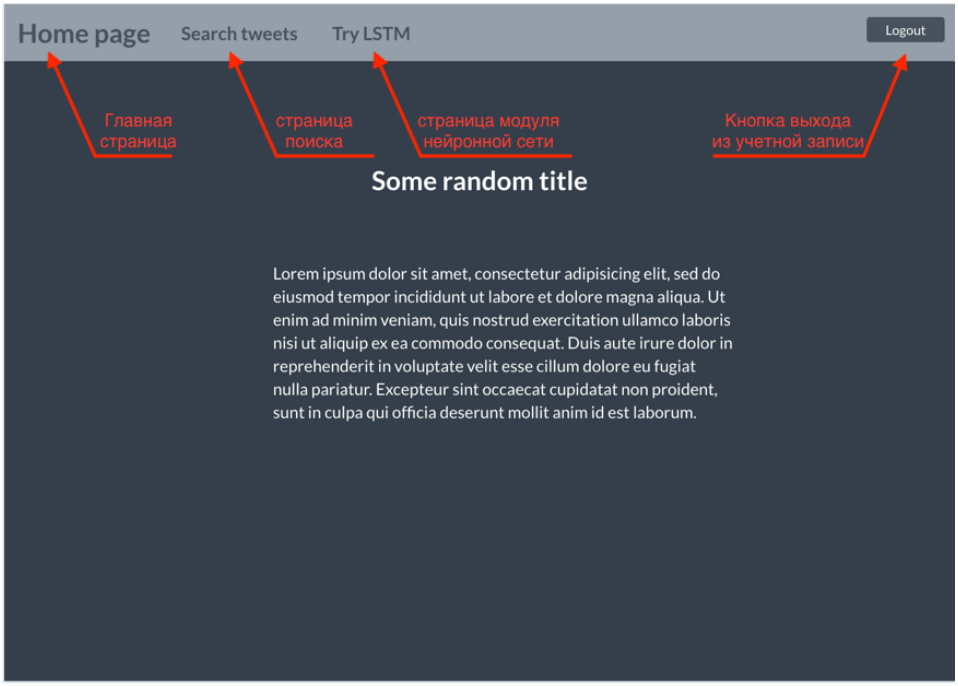


Рисунок №17 – прототип пользовательского интерфейса главной страницы

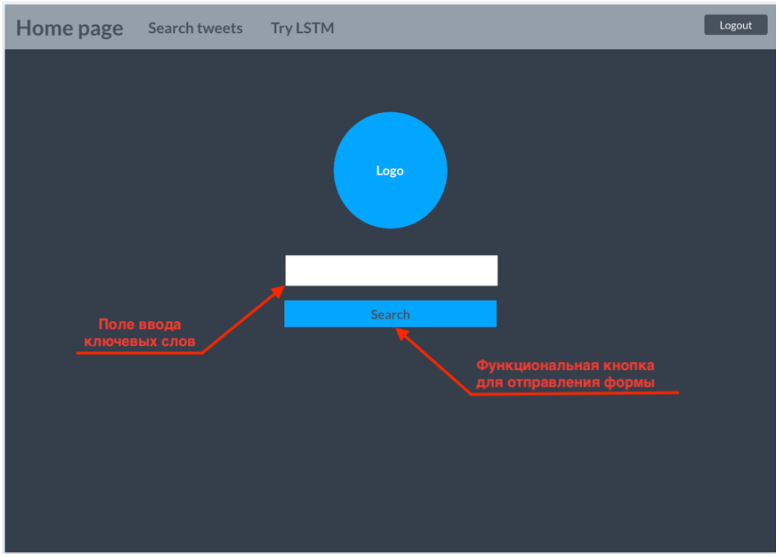


Рисунок №18 – прототип первой страницы интерфейса

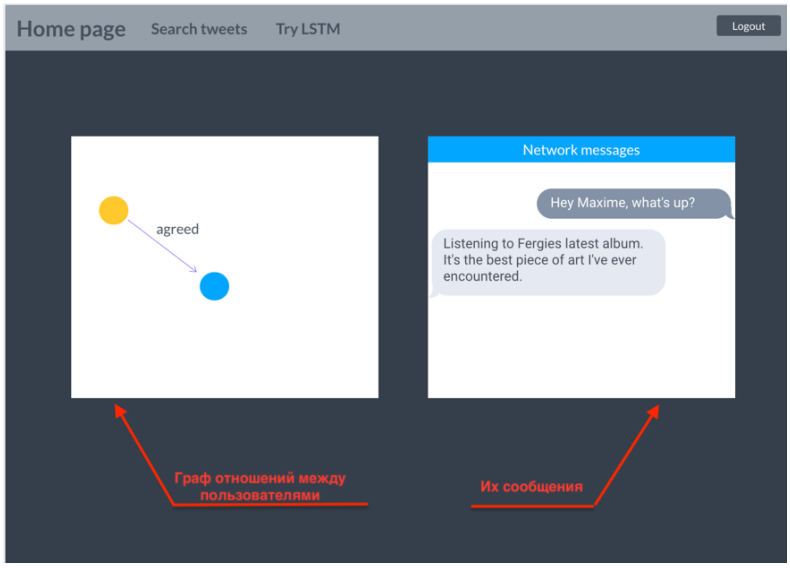


Рисунок №19 – веб-интерфейс после завершения поиска

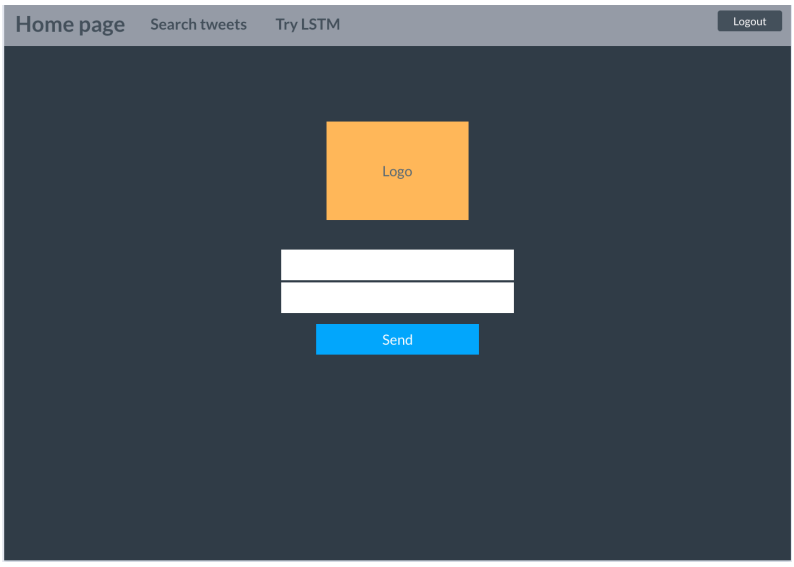


Рисунок №20 – прототип второй страницы

На второй странице гость должен увидеть логотип, два поля ввода и кнопку (рисунок №20). Заполнив данные два поля соответствующей информацией (сообщение и комментарии к нему) и подтвердив данные нажатием кнопки отправить, гость получит ответ от сервиса в виде красного/зеленого блока сообщении. Цвет блока будет напрямую зависеть от вывода нейронной сети. При условии, что нейронная сеть вернет отрицательный результат (несогласие), то блок соответственно окрасится в красный цвет, если положительный (согласие), то в зеленый.

2.4. Развертывания сервиса

В результате нынешней проделанной разработки была развернута программный сервис на хостинге «DigitalOcean» [16], с последующим присвоением доменного имени «bekzat-shayakhmetov.me» [17] при помощи ресурса «NameCheap» [18].

Диаграмма развертывания продемонстрирована на рисунке №21. Веб-приложение запущено на Unix-подобном сервере Ubuntu 16.06 с использованием веб-серверов nginx [19] и gunicorn [20].

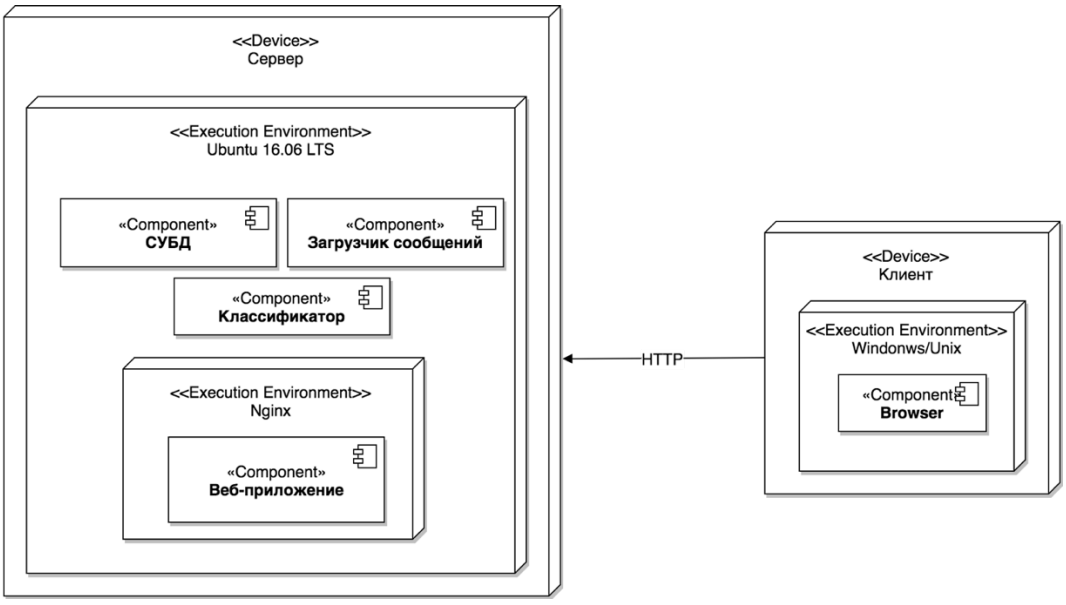


Рисунок №21 – диаграмма развертывания программного сервиса

3. ТЕСТИРОВАНИЕ КЛАССИФИКАТОРА

Тестирование реализованной архитектуры LSTM-сети производилось на собранном наборе тренировочных данных различной предметной области: бытовое общение, спорт, политика. В результате работы компонента загрузчика данных было собрано более 5 тысяч примеров согласия/несогласия, тестовый набор данных составил около 10% от всего тренировочного набора.

Для вычисления точности классификатора использовался встроенный функционал библиотеки «TensorFlow», который рассчитывает точность классификатора путем вычисления процентного соотношения правильно полученных результатов с общим количеством примеров:

$$\text{Точность} = \frac{\text{Число правильных результатов}}{\text{Общее число примеров}} * 100\%$$

3.1. Тестирование параметров нейронной сети

В данной главе приведен подробный анализ различных параметров нейронной сети. Сравнительная таблица различных параметров приведена в таблица №3. Каждое значение различных параметров проверялось при равных условиях относительно друг друга.

Таблица 3 – Сравнительная таблиц различных значений параметров нейронной сети

Размер выборки	Блоки LSTM	Оптимизация	Итоговая точность, %	Потери, %	Время тренировки, мин:сек	Количество итерации, шаг
Тестирование размера выборки						
50	64	Adam	~84	0.023	07:44	2000
20	64	Adam	~86	0.011	14:16	5000
Тестирование количество блоков LSTM						
50	**4**	Adam	84.1	0.16	02:25	2000
50	**8**	Adam	85.5	0.17	02:42	2000

Продолжение таблицы №3.

50	**16**	Adam	84	0.10	03:03	2000
50	**32**	Adam	85	0.05	04:55	2000
50	**64**	Adam	86	0.15	05:47	2000
Тестирование методов оптимизации						
50	64	**SGD**	48.6	6.74	07:07	2000
50	64	**Adadelta**	47.8	7.12	06:55	2000
50	64	**Adam**	83.6	0.001	06:48	2000
50	64	**RMSProp**	84.8	0.001	07:02	2000

Максимальное количество слов. Для корректной обработки входных данных нейронной сетью необходимо задать ограничения в размере слов в парах сообщений. Если этот размер слишком велик, матрица векторного представления слов будет заполнена нулями, что повлияет на точность, а если данный параметр слишком маленький, то сообщения могу быть обрезаны так, что смысл сообщения потеряет свою актуальность.

Необходимо определить среднее количество слов в двух конкатенированных предложениях. Для этого нужно подсчитать среднее количество слов поделив общее количество слов во всех примерах тренировочного набора на количество этих примеров. Для автоматизации данного процесса лучшим решением является написание скрипта на языке программирования Python (рисунок №22).

Рисунок №22 – скрипт подсчета среднего количества слов в тренировочном наборе данных

В результате работы написанного скрипта для подсчета слов получено среднее количество слов в тренировочном наборе данных: 49 слов.

Методы оптимизации. Как было упомянуто в главе №2 (ГЛАВА 2. Проектирование и реализация программного сервиса), одним из предпочитаемых методов оптимизации является метод оптимизации Adam (Adam Optimizer). На рисунке №27 продемонстрировано сравнение процессов обучения при одинаковых условиях (параметры указаны в таблице №3), но с различными методами оптимизации. Шагом обучения выбрано значение в «0.001». Для сравнения выбраны методы оптимизации Adam, SGD, RMSProp и Adadelta. Оранжевым обозначен график обучения с использованием метода оптимизации Adam. На графике видно, как метод оптимизации Adam во много раз эффективнее двух других методов (SGD и Adadelta). Однако два метода –

метод оптимизации Adam и RMSPropr – имеют почти идентичные результаты в аккуратности и в минимизировании потерь.

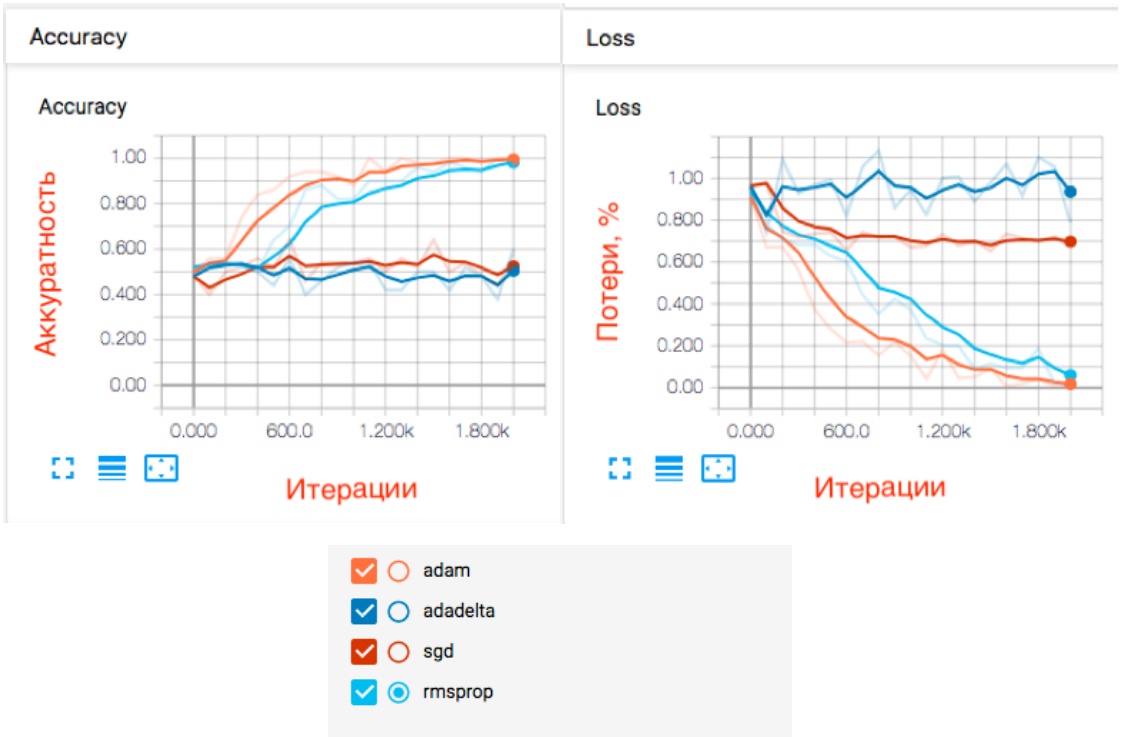


Рисунок №27 – сравнение метода оптимизации Adam, SGD, Adadelta, и RMSProp

Размер выборки. Выборка – это количество тренировочных данных, подаваемых нейронной сети на каждую итерацию. Выбирать размер выборки стоит с расчетом на ресурсоэффективность сервиса. Чем больше выборка, тем больше тренировочных данных оно будет в себе содержать, следовательно, тем больше памяти будет требовать. Тестирование выборки следует проводить с расчетом:

$$batch\_size_1 * iteration_1 = batch\_size_2 * iteration_2$$

где 1 – модель с меньшим количеством выборки, но с большим количеством итерации (Выборки=20, итерации=5000);

где 2 – модель с меньшим количеством итерации, но с большим количеством выборки (Выборки=50, итерации=2000);

Тестирование производилось при одинаковых параметрах нейронной сети (таблица №3). Два подхода показывают приблизительно одинаковый результат аккуратности и потерь при обучении (рисунок №23), однако модель нейронной сети с большим размером выборки и меньшим количеством итераций выигрывает по времени обучения почти в два раза (рисунок №24). Модели с большим количеством итерации понадобилось вдвое меньше времени – 14 минут, против 7 минут для модели с меньшим количеством итераций.

Рисунок №23 – сравнение результатов моделей с разным количеством выборок

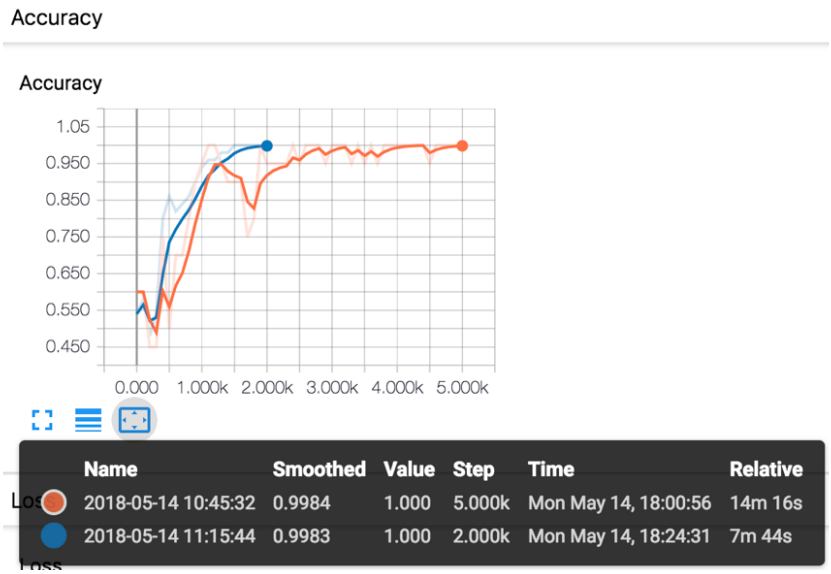


Рисунок №24 – сравнение временных показателей

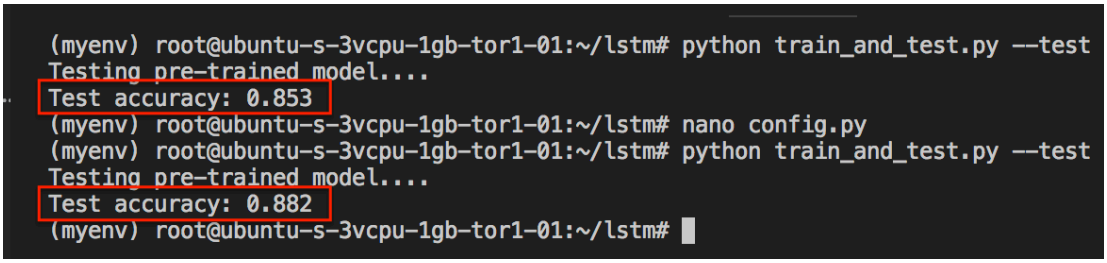


Рисунок №25 – точность разных моделей нейронной сети

Проведенный тест на тренировочном наборе данных не выявил значительных изменений в работе модели. Оба варианта показали почти идентичный результат с отклонением в пару процентов (рисунок №25).

Количество LSTM блоков. В ходе проведенного тестирование были использованы одинаковые условия для каждого значения количества блоков LSTM (таблица №3).

Как показано на рисунке №26 разница в процессе обучения между графами с 8, 16, 32 и 64 количествами блоков минимальна. В ходе тестирования выявлено, что для текущего количества тренировочных наборов данных самым оптимальным значением блоков LSTM является значение 32, так как точность при данном значении схоже относительно других вариантов, однако время и потери обучения значительно ниже по сравнению с другими параметрами (таблица №3).

Но стоит учесть тот факт, что увеличение блоков LSTM замедляет процесс обучения: в среднем требуется вдвое больше времени. К примеру, нейронная сеть с параметрами количества блоков LSTM равной 16 провела процесс обучения за 3 минуты 3 секунды, в то время как нейронная сеть с 64 количеством блоков обучилась за 5 минут 47 секунд (таблица №3).

Исходя из данного примера, можно сделать вывод, что для маленького набора тренировочных данных лучше всего использовать меньшее количество блоков LSTM. Следовательно, чем больше тренировочный набор, тем больше блоков желательно использовать.

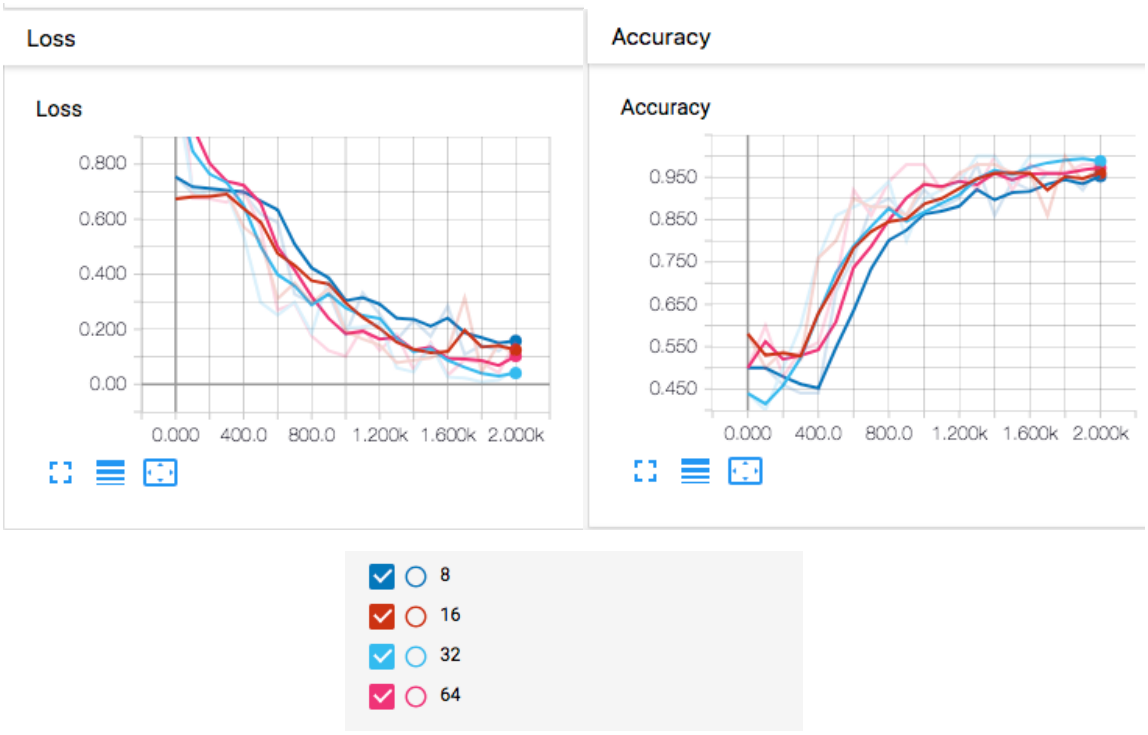


Рисунок №26 – сравнение разных количеств блоков

Шаг обучения. Чтобы выбрать оптимальный шаг обучения текущей архитектуры нейронной сети, следует провести опыт с тремя различными шагами при использовании метода оптимизации Adam: 0.01 (бордовый график), 0.001 (оранжевый график), 0.0001 (синий график). Как видно на рисунке №28, при стандартном шаге в 0.001, процесс обучения проходит более эффективно – потери при обучении минимальны, а аккуратность максимальна.

Рисунок №28 – сравнение значений шага обучения

4.2. Проверка влияния разделителя сообщений на точность классификации

Как было упомянуто выше, для решения проблем конкатенации двух различных сообщений (сообщение и комментарий к нему) было решено объединить два сообщения при помощи разделителя. На сравнительных рисунках №29, №30, №31 видно, что результаты тренировки нейронной сети отличаются незначительно при использовании и исключении разделителя – аккуратность при тренировке практически не отличаются друг от друга, а вот точность модели при тестировании отличаются примерно на 2.3%: точность модели с разделителем **89.2%**, без разделителя **86.9%,** что в свою очередь не несет никаких серьезных изменений.

Однако, если провести тест на собственно введенных пар сообщений, которых нет ни в тренировочном ни в тестовом корпусе данных, то результаты значительно отличаются. На рисунках №32 и №33 видно, что при одинаковых условиях и входных данных нейронная сеть с разными обученными моделями реагирует на сообщения согласия или несогласия по-разному: нейронная сеть, обученной на данных с разделителем, показывает более точный результат по сравнению с моделью без разделителя. Так как комментарий рода «yeah, it is brilliant» (Да, он шикарен) несет смысловой оттенок согласия на сообщение «I think the new Tarantino’s movie is masterpiece» (Я думаю, что новый фильм Тарантино – шедевр), а комментарий рода «Nope, it was terrible» (Не, он ужасен) указывает на обратное. Модель нейронной сети без разделителя утверждает со 100% вероятностью, что первое и второе сообщения являются несогласием, что в свою очередь указывает на неточность модели.

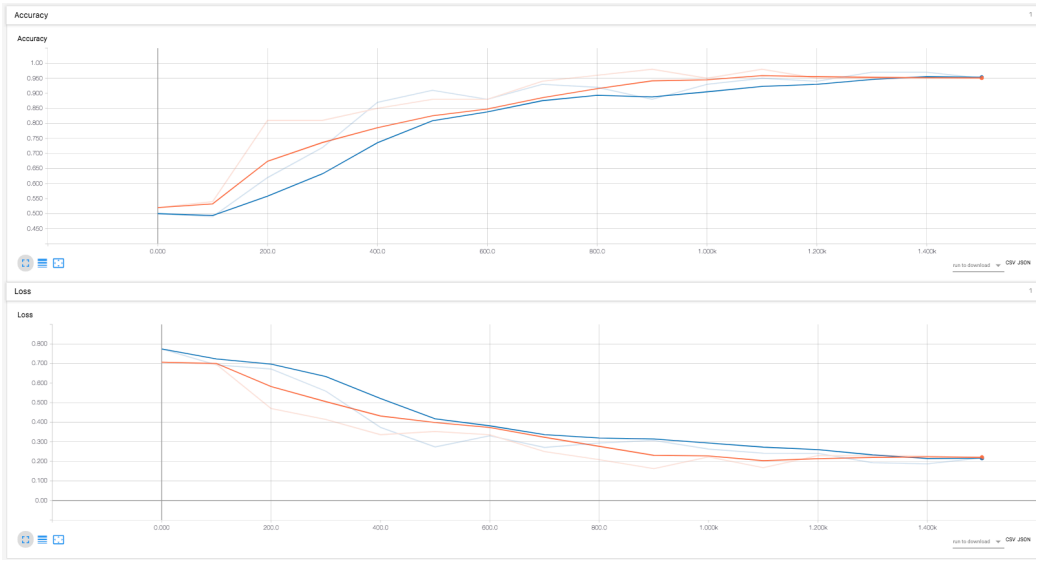


Рисунок №29 – сравнительный график процесса обучения с разделителем и без него

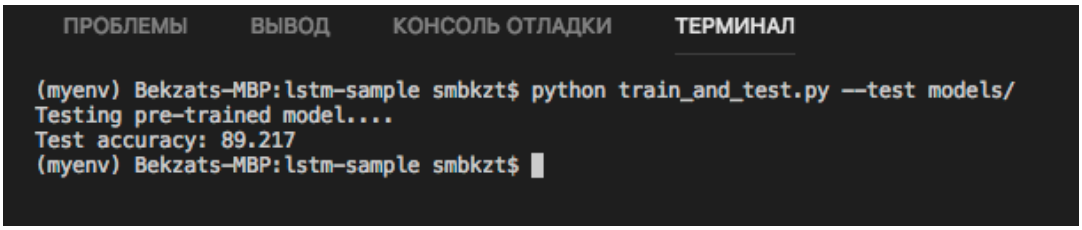


Рисунок №30 – Тестирование на тренировочном наборе данных с разделителем

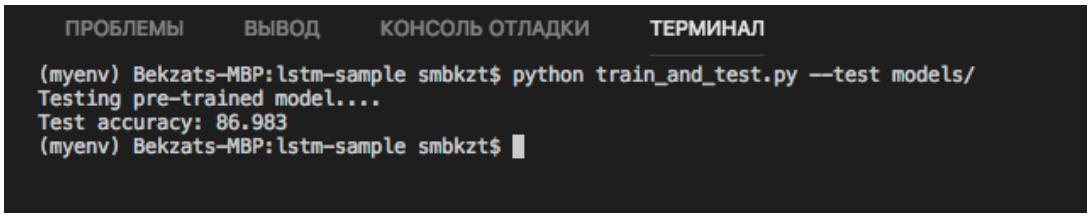


Рисунок №31 – тестирование на тренировочном наборе данных без разделителя

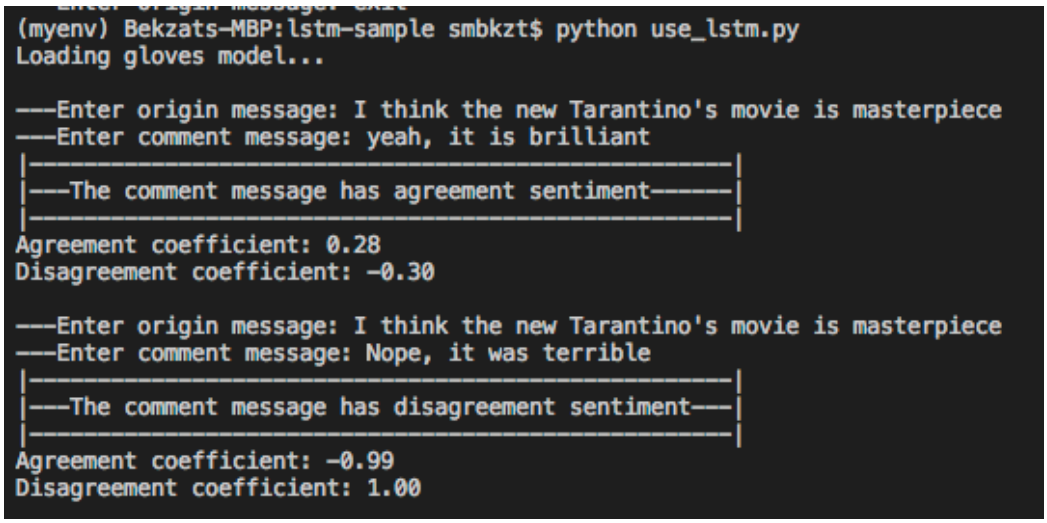


Рисунок №32 – Тест с использованием разделителя на новых данных

Рисунок №33 – Тест без использования разделителя на новых данных

4. ИНТЕРФЕЙС И ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

Как показано на рисунке №34, реализованное веб-приложение содержит навигационное меню в верхней части страницы для удобного перехода по страницам. На навигационном меню расположены ссылки для доступа к главной странице, странице поиска и анализу сообщений, страница анализа собственных данных, и страница настройки модели нейронной сети.

На рисунке №35 изображена первая страница – страница тестирования работы нейронной сети в режиме реального времени. На странице расположены два поля для ввода текстовой информации и кнопка отправки сообщений компоненту нейронной сети.

Как видно на рисунке №36, ответ от сервиса пользователь получает в виде сообщения типа «the comment message has (dis)agreement sentiment» (комментарий имеет оттенок (не)согласия). Цвет блока с ответом варьируется между красным и зеленым, в зависимости от ответа сервиса.

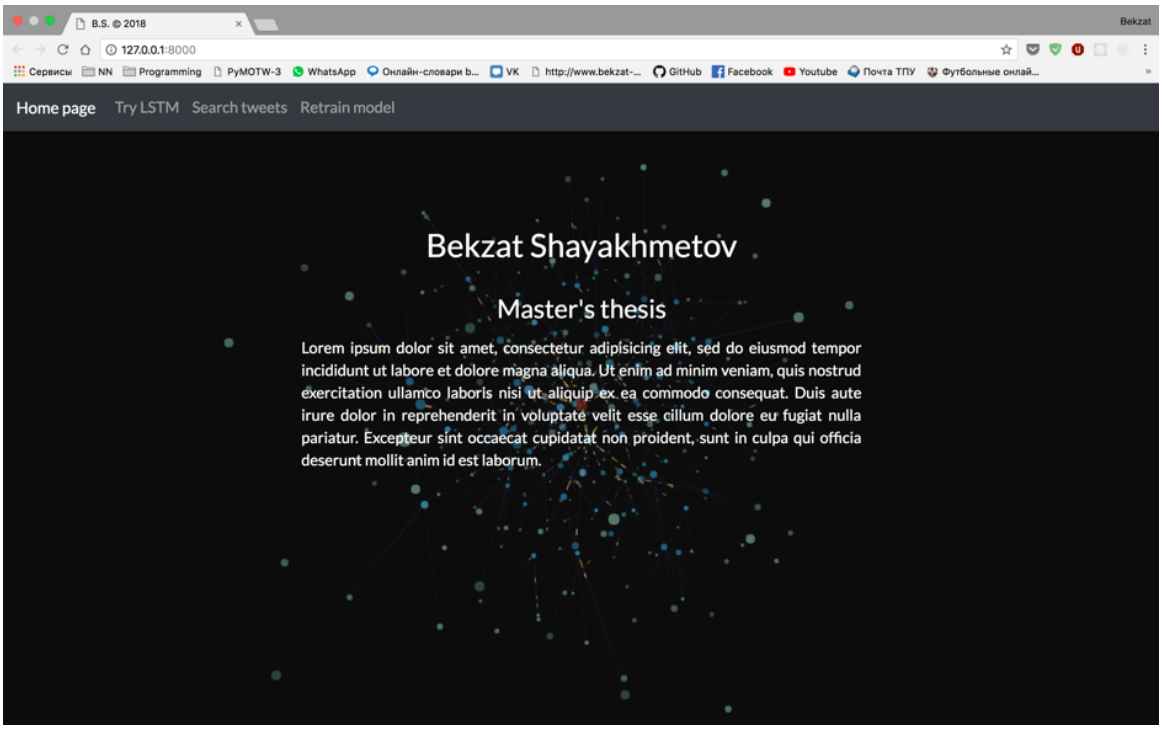


Рисунок №34 – Главная страница веб-приложения

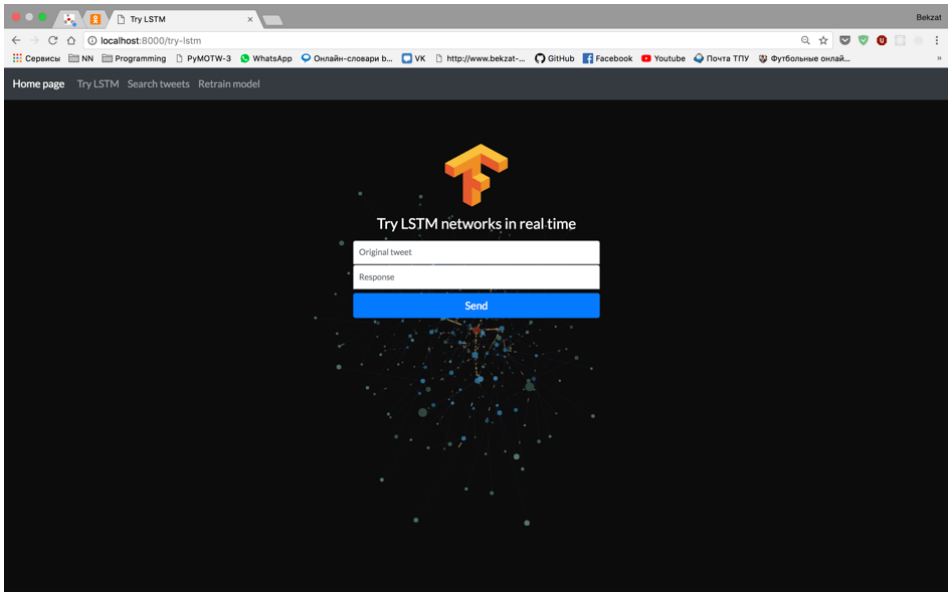


Рисунок №35 – Первая страница

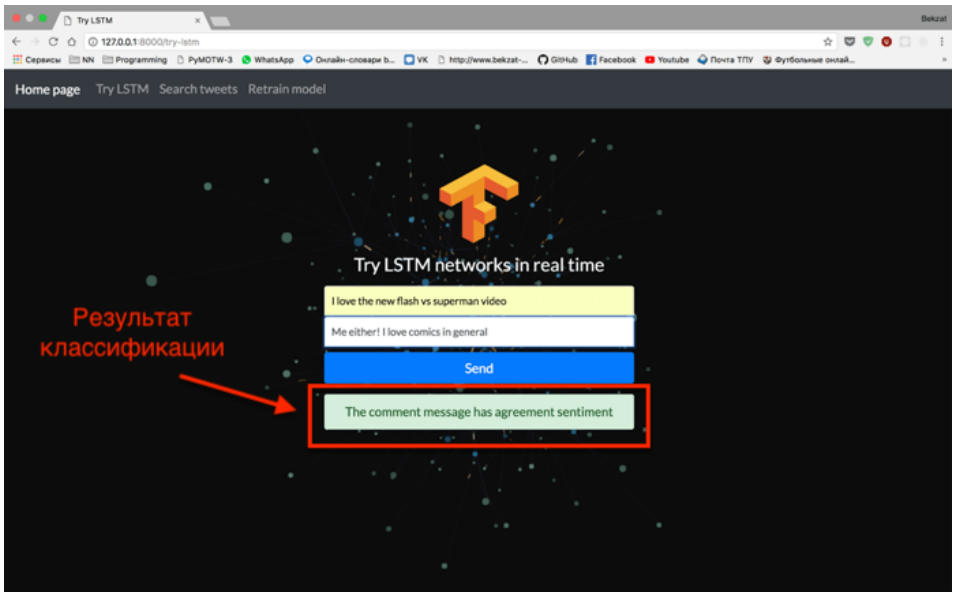


Рисунок №36 – Положительный ответ сервиса

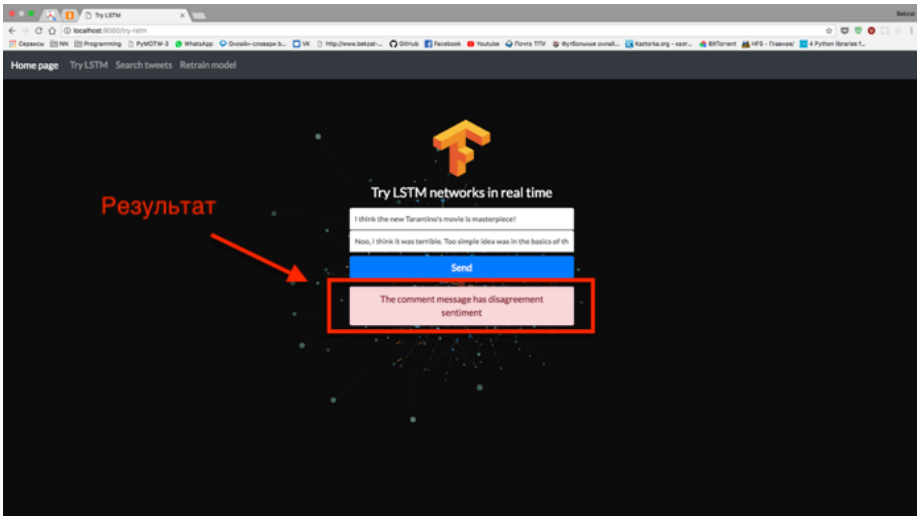


Рисунок №37 – отрицательный ответ сервиса

Вторая страница веб-приложение призвана автоматизировать поиск и анализ сообщений социальной сети Twitter (рисунок №38). На странице можно увидеть лишь одно поле для ввода ключевых слов, по которым сервис производит поиск соответствующих данных (сообщений и комментарий). При совершении поиска путем нажатия кнопки, сервис аналогичным с первой страницей способом отправляет асинхронным методом введённые ключевые слова. На серверной стороне два компонента начинают выполнять работу, обмениваясь между собой данными: происходит поиск и анализ сообщений.

После того, как два вышеуказанных компонента завершают свою работу, конечным этапом является визуализация отношений между пользователями социальной сети в виде графа (рисунок №40). На графе отображены все авторы найденных ранее сообщений. Вершины графа отображены в виде профильных изображений пользователей, а ребрами этих вершин являются результаты проведенного анализа на основе нейронной сети архитектуры LSTM. Вершины имеют направление и статус – пользователь Б согласен/не согласен с пользователем А.

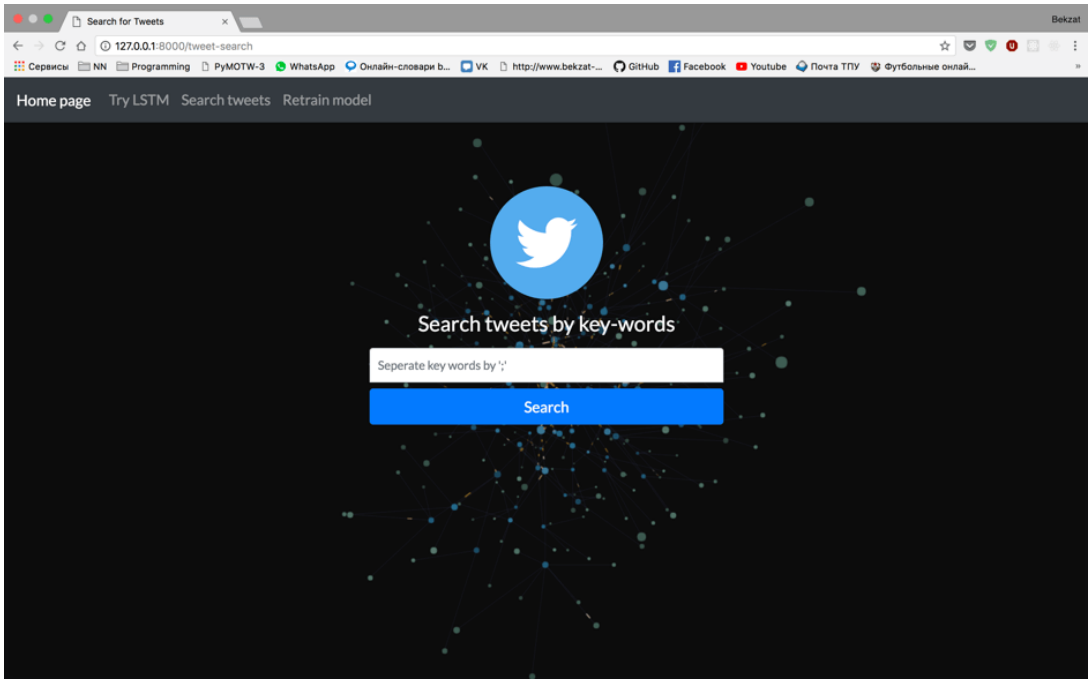


Рисунок №38 – Вторая страница веб приложения (страница поиска)

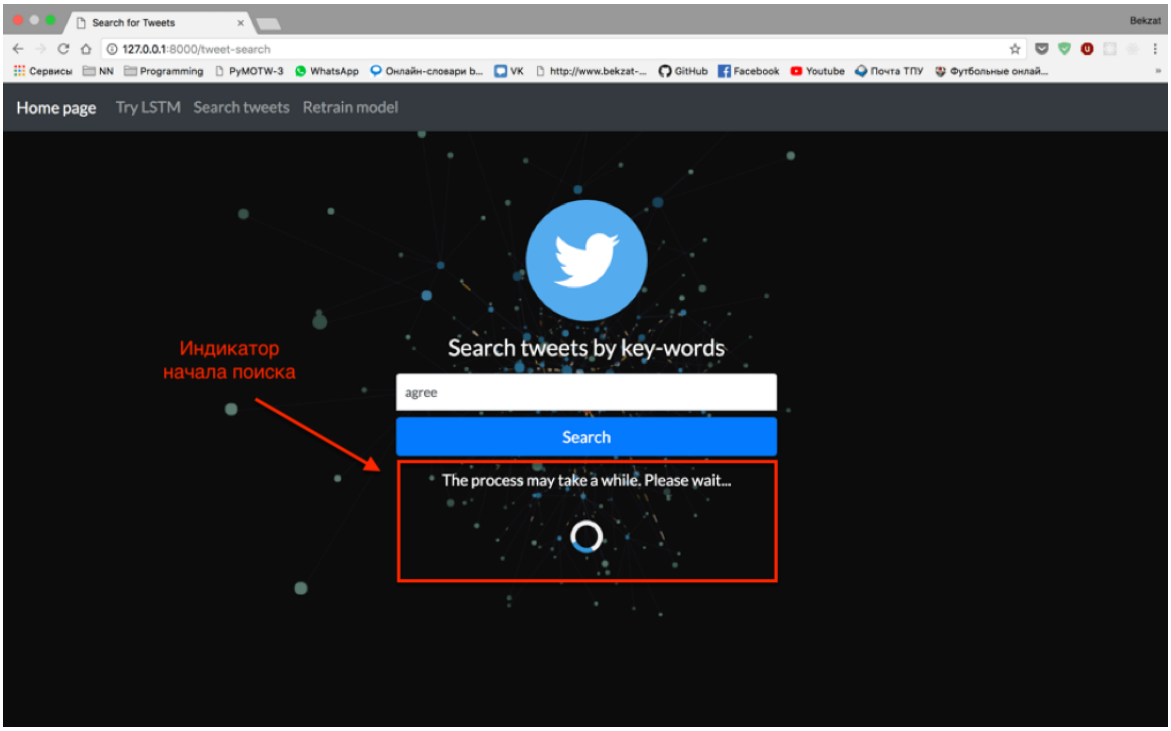


Рисунок №39 – Процесс поиска сообщений

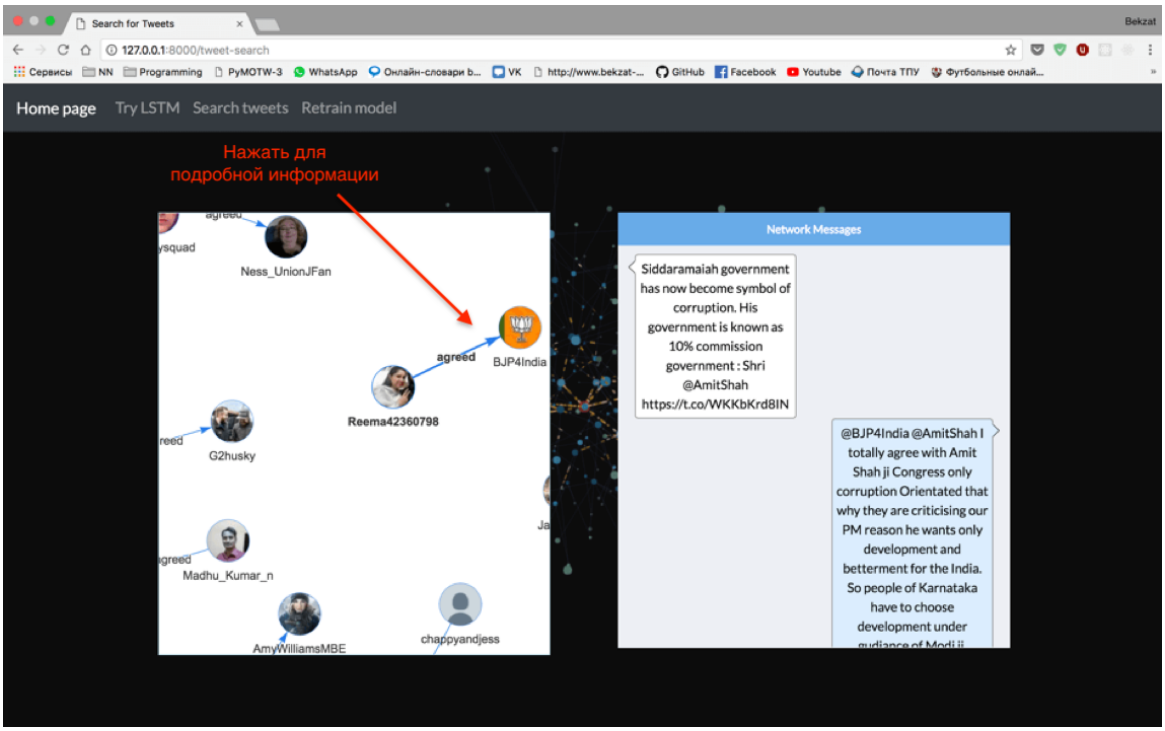


Рисунок №40 – Результат классификации отношений между пользователями социальной сети «Twitter»

5. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

5.1. Предпроектный анализ

5.1.1. Потенциальные потребители результатов исследования

Разработка, которой посвящена данная работа, представляет собой многопользовательское веб-приложение для анализа сообщений социальной сети «Twitter» на основе нейронной сети архитектуры LSTM.

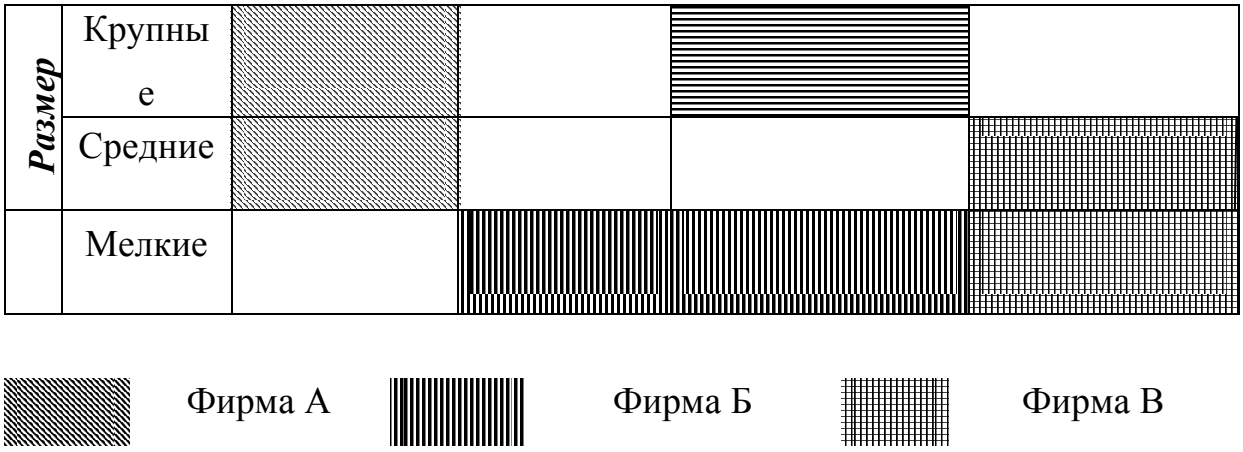
Исходя из особенностей веб-приложения, можно судить о круге лиц, которые потенциально будут заинтересованы в разработке. Целевым рынком нынешней разработки являются маркетинговые компании, основной деятельностью которых является проведение исследовании рынка на различные тематики, таких как выявление мнений, споров, проведение голосовании. Все компании, нуждающиеся в автоматизации процесса анализа больших данных социальной сети, будут заинтересованы в такого рода программном продукте. Однако, в силу наличия в приложении искусственного интеллекта, работа может быть интересна также для лиц, занимающихся научно-исследовательской деятельностью, связанной с методами машинного обучения.

Сегментировать рынок услуг можно по степени потребности использования данных расчетов. Результат сегментирования представлены на таблице 5.1.

Таблица 5.1 – Карта сегментирования рынка услуг по разработке интернет-ресурсов.

	Вид интернет-ресурса			
	Нейронные сети	Поиск сообщений социальных сетей	Веб-сервисы по анализу сообщений	Автоматизация процессов

Продолжение таблицы №5.1.



5.1.2. Анализ конкурентных решений

Данная разработанное решение по классификации отношении между пользователей социальной сети является уникальной в своем роде, так как содержит в себе несколько взаимосвязанных компонентов, аналогов такой системы не обнаружено. Поэтому в качестве конкурентов были рассмотрены похожие решения по отдельным компонентам. К примеру, веб-сервис по поиску сообщении из социальной сети, или любой ресурс, где реализован искусственный интеллект. В конечном итоге, в качестве конкурирующих решении были выбраны следующие продукты:

1. Онлайн ресурс «sentistrength» [1]
2. Веб-сервис «tone-analyzer» [2]
3. Анализатор сообщений от разработчика «sentdex» [3]

Экспертная оценка основных технических характеристик данных продуктов представлена в таблице 5.2.

Таблица 5.2 – оценочная карта сравнения конкурентных технических решений

№	Критерии оценки	Вес критерия	Баллы				Конкурентоспособность			
			Бф	Бк1	Бк2	Бк3	Кф	К1	К2	К3
	1	2	3	4	5	6	7	8	9	10
Технические критерии оценки ресурсоэффективности										

Продолжение таблица №5.2.

1	Повышение производительности труда пользователя	0,2	5	2	2	3	1	0,4	0,4	0,6
2	Удобство в эксплуатации (соответствует требованиям потребителей)	0,15	5	3	2	4	0,75	0,45	0,3	0,6
3	Помехоустойчивость	0,03	4	3	4	5	0,12	0,09	0,12	0,15
4	Энергоэкономичность	0,01	4	5	5	4	0,04	0,05	0,05	0,04
5	Надежность	0,05	4	3	3	5	0,2	0,15	0,15	0,25
6	Потребность в ресурсах памяти	0,05	3	4	4	3	0,15	0,2	0,2	0,15
7	Функциональная мощность (предоставляемые возможности)	0,1	5	1	1	2	0,5	0,1	0,1	0,2
8	Простота эксплуатации	0,1	5	2	2	4	0,5	0,2	0,2	0,4
9	Качество пользовательского интерфейса	0,07	4	1	1	5	0,28	0,07	0,07	0,35
Экономические критерии оценки эффективности										
1	Конкурентоспособность продукта	0,01	5	2	2	4	0,05	0,02	0,02	0,04
2	Уровень проникновения на рынок	0,01	1	4	3	1	0,01	0,04	0,03	0,01
3	Цена	0,09	4	2	2	2	0,36	0,18	0,18	0,18
4	Послепродажное обслуживание	0,08	5	2	2	4	0,4	0,16	0,16	0,32
5	Финансирование научной разработки	0,04	5	5	4	3	0,2	0,2	0,16	0,12
6	Срок выхода на рынок	0,01	4	5	5	5	0,04	0,05	0,05	0,05
	Итого	1	63	44	42	54	**4,6**	**2,36**	**2,19**	**3,46**

Исходя из проведенного анализа можно заключить, что уязвимость конкурентных технологических решений связана, прежде всего с отсутствием повышения производительности труда, то есть предоставляемые системами возможности не достаточны, для реализации успешного процесса автоматизации. Реализации первых двух конкурентов очень схожи, онлайн ресурс tone-analyzer и SentiStrength, поэтому коэффициенты их конкурентоспособности едва различимы. Данные компании обладают практически единственным боле-менее значительным достоинством – это уровень их проникновения на рынок, они располагают широкой базой тайных покупателей из разных регионов страны, а многие предприятия пользуются их услугами.

Наиболее сильным конкурентом можно считать стандартный функционал системы от разработчика «sendtex». Его основными достоинствами являются надежность и помехоустойчивость, удобность в использовании, однако, как и два предыдущих конкурента, данный онлайн ресурс не удовлетворяет требованиям, предъявляемым к системе, необходимой для автоматизации процессов.

Преимуществом собственной разработки помимо того, что, она в десятки раз сокращает время выполнения процесса, можно считать то, что данный продукт на рынке является уникальным. Аналогов разработанной системы не существует. Также сильной стороной является то, что данная система проста в использовании так как разрабатывалась с тем учетом, что большинство ее пользователей не будут иметь большого опыта работы с компьютерами – ведь доступ к системе можно получить из любого удобного устройства включая как телефоны, планшеты, так и умные телевизоры.

5.1.3. SWOT-анализ

SWOT-анализ применяют для исследования внешней и внутренней среды проекта. Матрица составляется на основе анализа рынка и

конкурентных технических решений, и показывает сильные и слабые стороны проекта, возможности и угрозы для разработки.

Первый этап заключается в описании сильных и слабых сторон проекта, в выявлении возможностей и угроз для реализации проекта, которые проявились или могут появиться в его внешней среде. Матрица SWOT представлена в таблице 5.3.

Таблица 5.3 – SWOT-анализ

		Сильные стороны	**Слабые стороны**
		С1. Хорошо спроектированная архитектура нейронной сети, удобный web-интерфейс и их налаженное взаимодействие. С2. Широкий спектр дополнительного функционала, облегчающий работу пользователя (к примеру, возможность настраивать нейронную сеть). С3. Дружелюбный и интуитивно понятный интерфейс и полная документация. С4. Постоянная поддержка разработчика. С5. Доступ к платным ресурсам посредствам студенческих поддержек. С6. Использование облачных сервисов для проведения вычислений.	СЛ1. Использование IP-адреса при обращении к web-приложению, что затрудняет доступ к ресурсу. СЛ2. Потребность в больших объемах вычислительных ресурсов (мощностей компьютера). СЛ3. Высокие денежные затраты на разработку СЛ4. Неточность полученных результатов. СЛ5. Временами происходящие отказы системы.
	Возможности		
В 1	Покупка доменного имени	В1С5. Использование бесплатных сервисов для студентов позволит получить доменное имя.	В1СЛ3 Поддержка доменного имени может увеличить денежные затраты.
В 2	Работа алгоритма онлайн	В2С6. Облачные вычисления позволят сократить нагрузку на локальную машину.	В2СЛ5. Выходу системы на рынок может воспрепятствовать использование IP-адреса для обращения к web-приложению, а также системные ошибки, вызывающие крах системы.
В 3	Получить финансирование	В2В5С4С5. Хорошо спроектированная архитектура нейронной сети и постоянная поддержка разработчика позволит доработать систему и выйти в плюс по прибылям.	
В 4	Выход системы на рынок	В4С2С3С4. Широкий спектр функционала, дружественный интерфейс, документация, а также поддержка разработчика способствуют распространению системы на рынок.	В3В2В5СЛ5. Высокая стоимость разработки и недоработка старых ошибок могут стать помехой в расширении функционала.
В 5	Доработка в связи с пожеланиями		

Продолжение таблицы №5.3.

Угрозы				
У	1	Неприятие автоматизации пользователями	У1У2С2С3С4. Дружелюбный интерфейс, понятная документация позволят избежать неправильного выполнения инструкций, а также неприятия автоматизации. Также избеганию неприятия способствует автоматическое логгирование ошибок и другие скрытые возможности системы.	У4СЛ1 из-за несвоевременного финансирования невозможно купить доменное имя.
У	2	Неверное выполнение инструкций пользователем		У1У2У3СЛ1СЛ3СЛ4 Неудобная работа на портативных устройствах, медленная работа системы, несвоевременная поддержка системных администраторов, медленная работа системы, а также нежелание работать в определенных браузерах может привести к отказу пользователей работать с системой.
У	3	Медленная работа системы	У3С4. Грамотная поддержка разработчика снизит вероятность медленной работы системы	
У	4	Несвоевременное финансирование	У5С4С5. Постоянная поддержка разработчика и финансирование предприятия способствуют системе всегда оставаться актуальной.	У4У5СЛ2СЛ3СЛ4 Несвоевременное финансирование высокие мощностные и денежные затраты могут привести к остановке развития системы искусственного интеллекта, что в будущем полностью может потерять актуальность.
У	5	Потеря актуальности		

Второй этап состоит в выявлении соответствия сильных и слабых сторон научно-исследовательского проекта внешним условиям окружающей среды. Это соответствие или несоответствие должны помочь выявить степень необходимости проведения стратегических изменений.

Соотношения параметров представлены в таблице 5.4.

Таблица 5.4 – Интерактивная матрица проекта

Сильные стороны проекта							
Возможности проекта		С1	С2	С3	С4	С5	С6
	В1	-	-	-	0	+	0
	В2	-	-	-	+	+	+
	В3	-	+	+	-	0	0
	В4	-	+	+	+	-	0
	В5						

Продолжение таблицы №5.4.

Слабые стороны проекта						
Возможности проекта		СЛ1	СЛ2	СЛ3	СЛ4	СЛ5
	В1	-	-	+	0	-
	В2	-	-	-	-	+
	В3	-	-	-	-	+
	В4	-	+	+	+	-
	В5	0	-	0	-	+

Сильные стороны проекта							
Угрозы проекта		С1	С2	С3	С4	С5	С6
	У1	0	+	+	+	-	-
	У2	-	+	+	+	-	-
	У3	-	0	-	+	0	-
	У4	-	-	-	-	-	-
	У5	-	-	0	+	+	-

Слабые стороны проекта						
Угрозы проекта		СЛ1	СЛ2	СЛ3	СЛ4	СЛ5
	У1	+	-	+	+	-
	У2	+	-	+	+	0
	У3	+	-	+	+	-
	У4	+	-	-	-	-
	У5	0	+	+	+	-

5.1.4. Оценка готовности проекта к коммерциализации

На какой бы стадии жизненного цикла не находилась научная разработка полезно оценить степень ее готовности к коммерциализации и выяснить уровень собственных знаний для ее проведения (или завершения). Для этого необходимо заполнить специальную форму, содержащую показатели о степени проработанности проекта с позиции коммерциализации и компетенциям разработчика научного проекта. Перечень вопросов приведен в табл. 5.5.

Таблица 5.5 – Бланк оценки степени готовности научного проекта к коммерциализации

№ п/п	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
1.	Определен имеющийся научно-технический задел	4	4
2.	Определены перспективные направления коммерциализации научно-технического задела	3	5
3.	Определены отрасли и технологии (товары, услуги) для предложения на рынке	3	4
4.	Определена товарная форма научно-технического задела для представления на рынок	4	3
5.	Определены авторы и осуществлена охрана их прав	4	2
6.	Проведена оценка стоимости интеллектуальной собственности	4	2
7.	Проведены маркетинговые исследования рынков сбыта	3	2
8.	Разработан бизнес-план коммерциализации научной разработки	2	3
9.	Определены пути продвижения научной разработки на рынок	3	4
10.	Разработана стратегия (форма) реализации научной разработки	5	4
11.	Проработаны вопросы международного сотрудничества и выхода на зарубежный рынок	3	4
12.	Проработаны вопросы использования услуг инфраструктуры поддержки, получения льгот	2	3
13.	Проработаны вопросы финансирования коммерциализации научной разработки	3	3
14.	Имеется команда для коммерциализации научной разработки	2	2
15.	Проработан механизм реализации научного проекта	5	5
	ИТОГО БАЛЛОВ	50	40

Итоговые значения проработанности научного проекта и знания у разработчика лежат в диапазоне от 40 до 50, что говорит о средней перспективности проекта. Многие аспекты вывода продукта на рынок не были учтены, а также проявляется недостаток знаний. Следовательно, требуется дополнительные затраты на наём или консультации у соответствующих специалистов.

5.1.5. Методы коммерциализации результатов научно-технического исследования

Перспективность данного научного исследования выше среднего, поэтому не все аспекты рассмотрены и изучены. Таким образом, для организации предприятия этого недостаточно (пункт 4 – 8 не подходят). Но так как основной научно-технический задел определен, этого достаточно для коммерциализации для следующих методов (пункты 1 - 3): Торговля патентной лицензией; передача ноу-хау и инжиниринг. Степени проработанности научного проекта и уровень знаний разработчика достаточно для реализации пунктов, которые были выбраны.

5.2. Инициация проекта

В рамках процессов инициации определяются изначальные цели и содержание и фиксируются изначальные финансовые ресурсы. Определяются внутренние и внешние заинтересованные стороны проекта, которые будут взаимодействовать и влиять на общий результат научного проекта.

5.2.1. Цели и результаты проекта

Перед определением целей необходимо перечислить заинтересованные стороны проекта. Информация по заинтересованным сторонам представлена в таблице 5.6:

Таблица 5.6 – Заинтересованные стороны проекта

Заинтересованные стороны проекта	**Ожидание заинтересованных сторон**
Пользователь	Простота в использовании программного продукта
Разработчик	Получение прибыли со своего продукта
Научный руководитель, студент	Выполненная выпускная квалификационная работа

Цели и результат проекта представлены в таблице 5.7:

Таблица 5.7 – Цели и результат проекта

Цели проекта:	• Собрать набор тренировочных данных из социальных сетей • Спроектировать функционал в соответствии с требованиями. • Произвести расчет стоимости разработки • Создать техническое задание и проектные решения • Реализовать алгоритм нейронной сети. • Разработать веб-решение для проблем анализы естественного языка. • Произвести тестирование • Внедрить разработку
Ожидаемые результаты проекта:	Успешное внедрение разработки в соответствующие компании.
Критерии приемки результата проекта:	Успешное тестирование функционала в соответствии с функциональным требованием.

Продолжение таблицы 5.8.

Требования к результату проекта:	**Требование:**
	• Выполненные все пункты функционального требования и требования к пользовательскому интерфейсу. • Разработанный функционал полностью соответствует проектным решениям.

5.2.2. Ограничения и допущения проекта

Ограничения проекта – это все факторы, которые могут послужить ограничением степени свободы участников команды проекта, а также «границы проекта» - параметры проекта или его продукта, которые не будут реализованных в рамках данного проекта. Эту информацию представить в табличной форме (табл. 5.8).

Таблица 5.8 – Ограничения проекта

Фактор	**Ограничения**
1.2.3.1 Бюджет проекта	80 000 рублей
1.2.3.1.1 Источник финансирования	НИТПУ
1.2.3.2 Сроки проекта	01.01.2017 – 31.05.2018
1.2.3.2.1 Фактическая дата утверждения плана управления проектом	12.12.2017
1.2.3.2.2 Плановая дата завершения проекта	31.05.2018

5.3. Планирование управления научно-техническим проектом

5.3.1. Иерархическая структура работ проекта

Группа процессов планирования состоит и процессов, осуществляемых для определения общего содержания работ, уточнения целей и разработки последовательности действий, требуемых для достижения данных целей.

План управления научным проектом должен включать в себя следующие элементы:

- иерархическая структура работ проекта;
- контрольные события проекта;
- план проекта;
- бюджет научного исследования.

Иерархическая структура работ (ИСР) – детализация укрупненной структуры работ. В процессе создания ИСР структурируется и определяется содержание всего проекта. На рисунке №5.2 представлен шаблон иерархической структуры.

Рисунок 5.1 – Иерархическая структура по ВКР

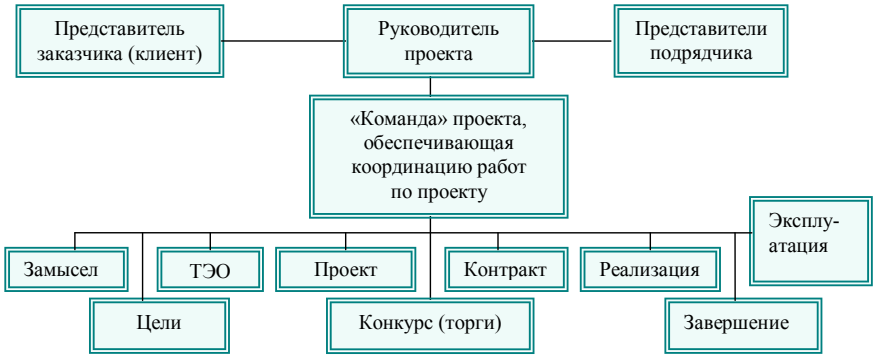


Рисунок 5.2 – Проектная структура проекта

В данном проекте будет использована проектная структура проекта, так как она подходит больше, потому что технология является новой и не исследуемой ранее, сложность проекта высока. Пример проектной структуры изображен на рисунке 3.

5.3.2. План проекта

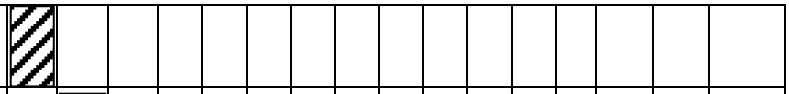
Диаграмма Ганта – это тип столбчатых диаграмм (гистограмм), который используется для иллюстрации календарного плана проекта, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

График строится в виде табл. 5.9. с разбивкой по месяцам и декадам (10 дней) за период времени выполнения научного проекта. При этом работы на графике следует выделить различной штриховкой в зависимости от исполнителей, ответственных за ту или иную работу.

Таблица 5.9. – Календарный план-график проведения НИОКР по теме

Код работы (из ИСР)	Вид работ	Исполнители	$Т_к$, ч.	Продолжительность выполнения работ															
				Янв.			Февр.			Март			Апр.			Май.			Июнь
				1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1
1	Выбор направления исследования	Р, С	5																

Продолжение таблицы №5.9.

2	Описание требований	Р	10	
3	Патентный поиск	С	10	
4	Составление технического задания	Р	10	
5	Изучение литературы	С	40	
6	Проектирование модуля по сбору данных	С	20	
7	Проектирование модуля нейронной сети	С	20	
8	Проектирование веб-интерфейса для нейронной сети	С	20	
9	Сбор данных для обучения искусственного интеллекта	С	40	
10	Разработка модуля нейронной сети	С	80	
11	Разработка веб-интерфейса	С	40	
12	Тестирование модуля нейронной сети	С	20	
13	Тестирование веб-модуля	С	20	
14	Написание документации	С	50	
15	Проверка работы	Р	20	

5.3.3. Бюджет научного исследования

При планировании бюджета научного исследования должно быть обеспечено полное и достоверное отражение всех видов планируемых расходов, необходимых для его выполнения. В процессе формирования бюджета, планируемые затраты группируются по статьям, представленным в таблице.

Затраты на электроэнергию

Этот пункт включает в себя стоимость всех материалов, необходимых для выполнения НИР.

К категории материалов относят:

1. Хостинг, доменное имя;
2. Электроэнергия.

Для данной разработки требуется специальное оборудование в виде персонального компьютера, но так как в наличии имелся личный ноутбук он не будет заносится в статью материальных расходов.

Разработка проводилась в течении 4 месяца (в среднем 20 дней в месяц) по 6 часов (480 часов), официально заявленная мощность оборудования 0,06 кВт/час.

Затраты на электроэнергию рассчитываются по формуле:

$$С_{эл} = Ц_{эл} \times Р \times F_{об}$$

где Цэл– тариф на электроэнергию (3,5 руб за 1 кВт-ч);

Р – мощность оборудования, кВт;

Fоб – время использования оборудования, ч.

$$С_{эл} = 3{,}5 \times 0{,}06 \times 480 = 100{,}8 \text{ руб.}$$

Так же в статью материальных расходов можно занести покупку хостинга и доменного имени:

$$С_{м} = С_{эл} + С_{х} + С_{ди}$$

$С_{х}$ – затраты на хостинг (*619,11 руб. в месяц*);

$С_{ди}$ – затраты на доменное имя (*595 руб. в год*);

Необходимо рассчитать затраты на хостинг за 5 месяцев пользования:

$$С_х = 5 \times 2000 = 10000 \text{ руб.}$$

$$С_м = 100{,}8 + 10000 = 10100 \text{ руб.}$$

Среда и средство разработки, программный софт и другие комплектующие, нужные для разработки, распространяются бесплатно и не требуют дополнительных затрат.

Таблица 5.10 – Расчет затрат по статье «Спецоборудование для научных работ»

№ п/п	Наименование оборудования	Кол-во единиц оборудования	Цена единицы оборудования, тыс.руб.	Общая стоимость оборудования, тыс.руб.
1.	Персональный компьютеры	1	-	-
2.	Linux сервер	5	2 000	10 000
3	Электроэнергия	-	-	100,8
4.	Среда разработки Visual Studio Code	1	-	-

Основная заработная плата

В настоящую статью включается основная заработная плата научных и инженерно-технических работников, рабочих макетных мастерских и опытных производств, непосредственно участвующих в выполнении работ по данной теме. Величина расходов по заработной плате определяется исходя из трудоемкости выполняемых работ и действующей системы оплаты труда. В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы (размер определяется Положением об оплате труда). Расчет основной заработной платы сводится в табл. 5.11.

Таблица 5.11 – Расчет основной заработной платы

№ п/п	Наименование этапов	Исполнители по категориям	Трудоемкость, чел.-дн.	Заработная плата, приходящаяся на один чел.-дн., руб	Всего заработная плата по тарифу (окладам), руб.
1		Руководитель		17 000	17 000
2		Магистр		2 650	2 650
			Итого:		19 650

$$C_{зп} = З_{осн} + З_{доп}, \quad (5.5)$$

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

$З_{доп}$ – дополнительная заработная плата.

Основная заработная плата $З_{осн}$ руководителя рассчитывается по следующей формуле:

$$З_{осн} = З_{дн} \bullet Т_{раб} \quad (5.6)$$

где $Т_{раб}$– продолжительность работ, выполняемых научно-техническим работником, раб.дн. (таблица 14);

$З_{дн}$– среднедневная заработная плата работника, руб.

Значит, для руководителя:

$$З_{осн} = 17000 * 1{,}3 = 22100 \text{ рублей}$$

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн}=(З_м \bullet М)/F_д \quad (5.7)$$

где $З_м$ – месячный должностной оклад работника, руб (в качестве месячного оклада магистра выступает стипендия, которая составляет 2650 руб);

М – количество месяцев работы без отпуска в течение года:

при отпуске в 45раб. дней М=10,4 месяца, 6 - дневная неделя;

$F_д$ – действительный годовой фонд рабочего времени научно-технического персонала (в рабочих днях) (табл.14). Тогда,

Для руководителя:

$$З_{дн} = \frac{22100 * 10,4}{254} = 904,8 \text{ рублей}$$

Для дипломника:

$$З_{дн} = \frac{2650 * 10,4}{217} = 127 \text{ рублей}$$

Баланс рабочего времени представлен в таблице 5.12.

Таблица 5.12 – Баланс рабочего времени

Показатели рабочего времени	**Руководитель**	**Магистр**
Календарное число дней	365	365
Количество нерабочих дней - выходные дни - праздничные дни	 52 14	 82 14
Потери рабочего времени - отпуск - невыходы по болезни	 45 –	 52 –
Действительный годовой фонд рабочего времени	254	217

Таблица 5.13 – Результаты расчета основной заработанной платы

Исполнители	**$З_б$, руб.**	**$k_р$**	**$З_м$, руб**	**$З_{дн}$, руб.**	**$Т_р$, раб. дн.**	**$З_{осн}$, руб.**
Руководитель	17000	1.3	22100	904,8	48	22100
Магистр	2650		2650	127	76	2650
					Итого по статье $З_{осн}$:	24750

Дополнительная заработная плата научно-производственного персонала

Дополнительная заработная плата включает оплату за непроработанное время (очередной и учебный отпуск, выполнение государственных обязанностей, выплата вознаграждений за выслугу лег и т.п.) и рассчитывается исходя из 10-15% от основной заработной платы, работников, непосредственно участвующих в выполнение темы:

$$З_{доп} = k_{доп} * З_{осн} \quad (4.9)$$

где $З_{доп}$ – дополнительная заработная плата, руб.;

$k_{доп}$ – коэффициент дополнительной зарплаты ($k_{доп}$ =0,1);

$З_{осн}$ – основная заработная плата, руб.

Для руководителя:

$$З_{доп} = 22100 * 0{,}1 = 2210 \text{ рублей}$$

В таблице 5.14 приведен расчёт основной и дополнительной заработной платы.

Таблица 5.14 – Заработная плата исполнителей ВКР, руб

Заработная плата	**Руководитель**	**Магистр**
Основная зарплата	22100	2650
Дополнительная зарплата	2210	–
Зарплата исполнителя	24310	2650
Итого		26960

Отчисления на социальные нужды

Статья включает в себя отчисления во внебюджетные фонды.

$$С_{внеб} = k_{внеб} \cdot (З_{осн} + З_{доп}) = 0{,}3 * (22100 + 2210) = 7293 \text{ руб.} \quad (4.10)$$

где $k_{внеб}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Накладные расходы

В эту статью относятся расходы по содержанию, эксплуатации и ремонту оборудования, производственного инструмента и инвентаря, зданий, сооружений и др. В расчетах эти расходы принимаются в размере 70 - 90 % от суммы основной заработной платы научно-производственного персонала данной научно-технической организации.

Накладные расходы составляют 80-100 % от суммы основной и дополнительной заработной платы, работников, непосредственно участвующих в выполнение темы.

Расчет накладных расходов ведется по следующей формуле:

$$С_{накл}= к_{накл} * (З_{осн} + З_{доп}) \quad (4.11)$$

где $k_{накл}$ – коэффициент накладных расходов.

$$С_{накл}= 0,3*(22100 + 2210) = 7293 \text{ руб.}$$

Формирование бюджета затрат научно-исследовательского проекта.

Рассчитанная величина затрат научно-исследовательской работы является основой для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку научно-технической продукции.

Таблица 5.15 – Бюджет затрат НТИ

Затраты по статьям						
№	Специальное оборудование для научных (экспериментальных) работ	Основная заработная плата	Дополнительная заработная плата	Накладные расходы	Отчисления на социальные нужды	Итого плановая себестоимость
1	10 100	22100	26960	7293	7293	**76 396**
2	100 000	50 000	5 000	16 500	16 500	**188 000**

В результате было получено, что бюджет затрат НТИ составит 76 396 руб. При этом затраты у конкурентов составляют 188 000 рублей, из чего можно сделать вывод что полученный продукт будет экономичней, чем у конкурентов.

5.3.4. Организационная структура проекта

В практике используется несколько базовых вариантов организационных структур: функциональная, проектная, матричная.

Для выбора наиболее подходящей организационной структуры можно использовать табл. 5.16.

Таблица 5.16 – Выбор организационной структуры научного проекта

Критерии выбора	Функциональная	Матричная	Проектная
Степень неопределенности условий реализации проекта	Низкая	Высокая	Высокая
Технология проекта	Стандартная	Сложная	Новая
Сложность проекта	Низкая	Средняя	Высокая
Взаимозависимость между отдельными частями проекта	Низкая	Средняя	Высокая
Критичность фактора времени (обязательства по срокам завершения работ)	Низкая	Средняя	Высокая
Взаимосвязь и взаимозависимость проекта от организаций более высокого уровня	Высокая	Средняя	Низкая

В данном случае выбор лежит к проектной структуре проекта из-за особенностей разработки. Составляющая проекта является модульные системы, работающие в постоянном взаимодействии с другими модулями.

Также основной причиной выбора проектной структуры является то, что технология проекта является новой, и имеются ограниченные сроки реализации.

5.3.5. План управления коммуникациями проекта

План управления коммуникациями отражает требования к коммуникациям со стороны участников проекта. Пример плана управления коммуникациями приведен в табл. 5.17.

Таблица 5.17 – Пример плана управления коммуникациями

№ п/п	**Какая** информация передается	**Кто** передает информацию	**Кому** передается информация	**Когда** передает информацию
1.	Статус проекта	Руководитель проекта	Представителю заказчика	Ежеквартально (первая декада квартала)
2.	Обмен информацией о текущем состоянии проекта	Исполнитель проекта	Участникам проекта	Еженедельно (пятница)
3.	Документы и информация по проекту	Ответственное лицо по направлению	Руководителю проекта	Не позже сроков графиков и к. точек
4.	О выполнении контрольной точки	Исполнитель проекта	Руководителю проекта	Не позже дня контрольного события по плану управления

5.3.6. Реестр рисков проекта

Идентифицированные риски проекта включают в себя возможные неопределенные события, которые могут возникнуть в проекте и вызвать

последствия, которые повлекут за собой нежелательные эффекты. Информацию по данному разделу необходимо свести в таблицу (табл. 5.18).

Таблица 5.18 – Реестр рисков

№	Риск	Потенциальное воздействие	Вероятность наступления (1-5)	Влияние риска (1-5)	Уровень риска*	Способы смягчения риска	Условия наступления
1	Потеря актуальности		2	5	средний	Внедрение нового функционала в процессе жизненного цикла ПО	Слишком хаотичное изменение рынка нейронных сетей, появление новых инструментов
2	Неточность ПО		4	5	высокий	Модификация алгоритма ПО	С течением времени нейронная сеть теряет свою точность

5.4. Определение ресурсной, финансовой, бюджетной, социальной и экономической эффективности исследования

5.4.1. Оценка абсолютной эффективности исследования

Динамические методы оценки инвестиций базируются на применении показателей:

- чистая текущая стоимость (**NPV**);
- срок окупаемости (**DPP**);
- внутренняя ставка доходности (**IRR**);
- индекс доходности (**PI**).

Все перечисленные показатели основываются на сопоставлении чистых денежных поступлений от операционной и инвестиционной деятельности, и их приведении к определенному моменту времени. Теоретически чистые денежные поступления можно приводить к любому моменту времени (к будущему либо текущему периоду). Но для практических целей оценку инвестиции удобнее осуществлять на момент принятия решений об инвестировании средств.

5.4.1.1. Чистая текущая стоимость (NPV)

Данный метод основан на сопоставлении дисконтированных чистых денежных поступлений от операционной и инвестиционной деятельности.

Если инвестиции носят разовый характер, то **NPV** определяется по формуле

$$\mathbf{NPV} = \sum_{t=1}^{n} \frac{\mathbf{ЧДП_{оп_t}}}{(1+i)^t} - \mathbf{I_0},$$

где $\mathbf{ЧДП_{оп_t}}$ – чистые денежные поступления от операционной деятельности;

$\mathbf{I_0}$ – разовые инвестиции, осуществляемые в нулевом году;

$\mathbf{t}$ – номер шага расчета ($\mathbf{t}$ =0, 1, 2… $\mathbf{n}$);

$\mathbf{n}$ – горизонт расчета;

i – ставка дисконтирования (желаемый уровень доходности инвестируемых средств).

Чистая текущая стоимость является абсолютным показателем. Условием экономичности инвестиционного проекта по данному показателю является выполнение следующего неравенства: **NPV**>0.

Чем больше **NPV**, тем больше влияние инвестиционного проекта на экономический потенциал предприятия, реализующего данный проект, и на экономическую ценность этого предприятия.

Таким образом, инвестиционный проект считается выгодным, если **NPV** является положительной.

Таблица 5.19 - Расчет чистой текущей стоимости по проекту в целом

№	Наименование показателей	Шаг расчета				
		0	1	2	3	4
1.	Выручка от реализации, тыс. руб.	0	99,184	99,184	99,184	99,184
2.	Итого приток, тыс. руб.	0	99,184	99,184	99,184	99,184
3.	Инвестиционные издержки, тыс. руб.	-76,396	0	0	0	0
4.	Операционные затраты, тыс. руб. С+Ам+ФОТ	0	33,824	33,824	33,824	33,824
5.	Налогооблагаемая прибыль		65,360	65,360	65,360	65,360
6.	Налоги, тыс. руб Выр-опер=донал. приб*20%	0	13,072	13,072	13,072	13,072
7.	Итого отток, тыс. руб. Опер.затр.+ налоги	-76,396	46,896	46,896	46,896	46,896
8.	Чистый денежный поток, тыс. руб. ЧДП=Пчист+Ам Пчист=Пдонал.-налог	-76,396	52,290	52,290	52,290	52,290
9.	Коэффициент дисконтирования (приведения при i=20%)	1,0	0,833	0,694	0,578	0,482
10.	Дисконтированный чистый денежный поток, тыс. руб. (с8*с9)	-76,396	43,557	36,289	30,223	25,203
11.	То же нарастающим итогом, тыс. руб. (**NPV** = 58,976 тыс. руб.)	-76,396	-32,739	3,550	33,773	58,976

Таким образом, чистая текущая стоимость по проекту в целом составляет 58,976 тысяч рублей, что позволяет его эффективности.

Дисконтированный срок окупаемости

Как отмечалось ранее, одним из недостатков показателя простого срока окупаемости является игнорирование в процессе его расчета разной ценности денег во времени.

Этот недостаток устраняется путем определения дисконтированного срока окупаемости.

Рассчитывается данный показатель примерно по той же методике, что и простой срок окупаемости, с той лишь разницей, что последний не учитывает фактор времени.

Наиболее приемлемым методом установления дисконтированного срока окупаемости является расчет кумулятивного (нарастающим итогом) денежного потока (см. табл. 5.20).

Таблица 5.20 – Дисконтированный срок окупаемости

№	Наименование показателя	Шаг расчета				
		0	1	2	3	4
1.	Дисконтированный чистый денежный поток (i=0,20)	-76,296	43,557	36,289	30,223	25,203
2.	То же нарастающим итогом	-76,296	-32,739	3,550	33,773	58,976
3.	Дисконтированный срок окупаемости	$PP_{дск}$ = 1+32,739/36,289 =0,92года				

Внутренняя ставка доходности (IRR)

Для установления показателя чистой текущей стоимости (NPV) необходимо располагать информацией о ставке дисконтирования, определение которой является проблемой, поскольку зависит от оценки экспертов. Поэтому, чтобы уменьшить субъективизм в оценке эффективности инвестиций на практике широкое распространение получил метод, основанный на расчете внутренней ставки доходности (IRR).

Между чистой текущей стоимостью (NPV) и ставкой дисконтирования

(**i**) существует обратная зависимость. Эта зависимость следует из таблицы 5.21 и графика, представленного на рисунке 5.1.

Таблица 5.21 - Зависимость **NPV** от ставки дисконтирования

No	Наименование показателя	0	1	2	3	4	
1	Чистые денежные потоки	-76,296	99,184	99,184	99,184	99,184	
2	коэффициент дисконтирования						
	i=0,1	1	0,909	0,826	0,751	0,683	
	i=0,2	1	0,833	0,694	0,578	0,482	
	i=0,3	1	0,769	0,592	0,455	0,35	
	i=0,4	1	0,714	0,51	0,364	0,26	
	i=0,5	1	0,667	0,444	0,295	0,198	
	i=0,6	1	0,625	0,39	0,244	0,095	
	i=0,7	1	0,588	0,335	0,203	0,07	
	i=0,8	1	0,556	0,309	0,171	0,095	
	i=0,9	1	0,526	0,277	0,146	0,077	
	i=1	1	0,5	0,25	3:00	0,006	
3	Дисконтированный денежный поток, тыс. руб						
	i=0,1	-76,296	72,887	66,232	60,218	54,766	177,807
	i=0,2	-76,296	66,793	55,648	46,346	38,649	131,14
	i=0,3	-76,296	61,661	47,469	36,484	28,064	97,382
	i=0,4	-76,296	57,251	40,894	29,187	20,848	71,884
	i=0,5	-76,296	53,483	35,602	23,654	15,876	52,319
	i=0,6	-76,296	50,115	31,272	19,565	7,617	32,273
	i=0,7	-76,296	47,148	26,862	16,277	5,613	19,604
	i=0,8	-76,296	44,582	24,777	13,711	7,617	14,391
	i=0,9	-76,296	42,177	22,211	11,707	6,174	5,973
	i=1	-76,296	40,092	20,046	10,023	0,481	-5,654

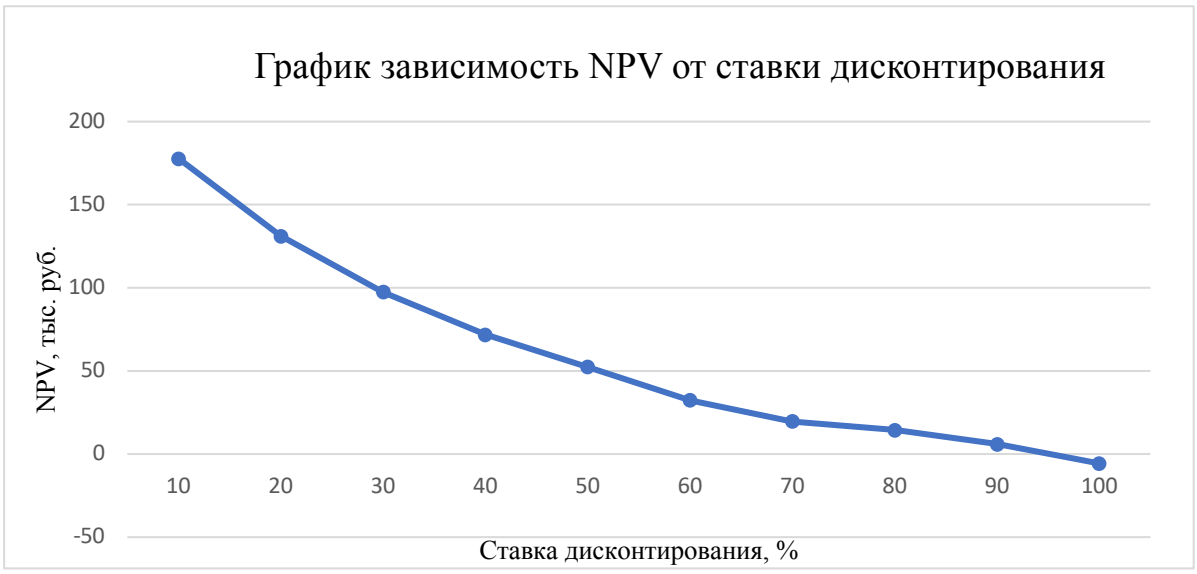


Рисунок 5.3 – Зависимость NPV от ставки дисконтирования.

Из таблицы и графика следует, что по мере роста ставки дисконтирования чистая текущая стоимость уменьшается, становясь отрицательной. Значение ставки, при которой **NPV** обращается в нуль, носит название «внутренней ставки доходности» или «внутренней нормы прибыли». Из графика получаем, что IRR составляет 0.93.

Индекс доходности (рентабельности) инвестиций (PI)

Индекс доходности показывает, сколько приходится дисконтированных денежных поступлений на рубль инвестиций.

Расчет этого показателя осуществляется по формуле

$$PI = \sum_{t=1}^{n} \frac{ЧПД_t}{(1+i)^t} / I_0,$$

где I_0 – первоначальные инвестиции.

$$\text{PI} = \frac{43{,}557 + 36{,}289 + 30{,}233 + 25{,}203}{126{,}253} = 1{,}07$$

PI =1,07>1, следовательно, проект эффективен при i=0,2;

NPV=131,14 тыс. руб.

Социальная эффективность научного проекта учитывает социально-экономические последствия осуществления научного проекта для общества в целом или отдельных категорий населений или групп лиц, в том числе как

непосредственные результаты проекта, так и «внешние» результаты в смежных секторах экономики: социальные, экологические и иные внеэкономические эффекты.

Таблица 5.22 – Критерии социальной эффективности

ДО	ПОСЛЕ
Поиск и анализ данных вручную	Нейронная сеть самостоятельно производит анализ сообщений в социальных сетях, тем самым процесс получения, анализа данных становится автоматизированным.
Нехватка удобных сервисов по анализу текста	Повышение конкурентоспособности рынка искусственного интеллекта.
Трата времени на скучные, однотипные задачи	С ростом популярности искусственного интеллекта возрастает их качество, тем самым в скором будущем вся рутинная и скучная работа перейдет в руки искусственных нейронных сетей.

5.4.2. Оценка сравнительной эффективности исследования

Интегральный показатель финансовой эффективности научного исследования получают в ходе оценки бюджета затрат трех (или более) вариантов исполнения научного исследования. Для этого наибольший интегральный показатель реализации технической задачи принимается за базу расчета (как знаменатель), с которым соотносится финансовые значения по всем вариантам исполнения.

Интегральный финансовый показатель разработки определяется как:

$$I_{\text{финр}}^{\text{исп}i} = \frac{\Phi_{pi}}{\Phi_{max}}, \quad (5.13)$$

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

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

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги).

Полученная величина интегрального финансового показателя разработки отражает соответствующее численное увеличение бюджета затрат

разработки в разах (значение больше единицы), либо соответствующее численное удешевление стоимости разработки в разах (значение меньше единицы, но больше нуля).

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum a_i \cdot b_i, \quad (5.14)$$

где 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.

Расчет интегрального показателя ресурсоэффективности приведен в форме таблицы (табл. 5.23).

Таблица 5.23 – Сравнительная оценка характеристик вариантов исполнения проекта

Критерии / **ПО**	**Весовой коэффициент параметра**	**Текущий проект**	**Аналог 1**	**Аналог 2**
1. Способствует росту производительности труда пользователя	0,1	5	3	4
2. Удобство в эксплуатации (соответствует требованиям потребителей)	0,15	5	2	3
3. Помехоустойчивость	0,15	3	3	3
4. Энергосбережение	0,20	4	3	3
5. Надежность	0,25	4	4	4
6. Точность анализа	0,15	4	4	4
ИТОГО	1	25	19	22

$$I_{ф}^{p} = \frac{Ф_{i}^{p}}{Ф_{max}} = \frac{5}{25} = 0.2$$

$$I_{ф}^{a} = \frac{Ф_{i}^{a}}{Ф_{max}} = \frac{4}{19} = 0.21$$

$$I_{т}^{p} = 5 * 0{,}1 + 5 * 0{,}15 + 3 * 0{,}15 + 4 * 0{,}2 + 4 * 0{,}25 + 5 * 0{,}15 = 3{,}94$$

$$I_{т1}^{a} = 3 * 0{,}1 + 2 * 0{,}15 + 3 * 0{,}15 + 3 * 0{,}2 + 4 * 0{,}25 + 2 * 0{,}15 = 3{,}15$$

$I_{т2}^{a}$ =4*0,1+3*0,15+3*0,15+3*0,2+4*0,25+4*0,05 = 3,5

$$I_{финр}^{p} = \frac{I_{т}^{p}}{I_{ф}^{p}} = \frac{3.94}{0.2} = 19.7$$

$$I_{финр}^{a} = \frac{I_{т}^{a}}{I_{ф}^{a}} = \frac{3.15}{0.21} = 21$$

$$Э_{ср} = \frac{I_{ф}^{p}}{I_{ф}^{a}} = \frac{0.2}{0.21} = 0.95$$

Таблица 5.24 – сравнительная эффективность разработки

№ п/п	Показатели	Аналог	Разработка
1	Интегральный финансовый показатель разработки	0.21	0.2
2	Интегральный показатель ресурсоэффективности разработки	3.15	3.94
3	Интегральный показатель эффективности	15	19.7
4	Сравнительная эффективность вариантов исполнения	0.95	1,05

Сравнение значений интегральных показателей эффективности позволяет судить о приемлемости существующего варианта решения поставленной в магистерской диссертации технической задачи с позиции финансовой и ресурсной эффективности. В ходе проведения анализа показателей эффективности инвестиций была получена чистая текущая стоимость (NPV) – 58,976тыс. руб. Таким образом, данный инвестиционный проект можно считать выгодным, NPV является положительной величиной. Дисконтированный срок окупаемости проекта ($\mathbf{РР_{дск}}$) составляет 0.92 года. Внутренняя ставка доходности (IRR) – 0.93, что позволяет признать инвестиционный проект экономически оправданным, так как выполняется

условие неравенства IRR> i. Индекс доходности (PI) – 1.07, и, основываясь на том, что данная величина превышает единицу, можно утверждать, что данная инвестиция приемлема.

Сравнение значений интегральных показателей эффективности показало, что более эффективным вариантом решения поставленной в бакалаврской работе технической задачи с позиции финансовой и ресурсной эффективности является исполнение 3 - модифицирование плазмой атмосферного давления.

6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

В настоящем разделе рассматриваются вопросы производственной и экологической безопасности, безопасность в чрезвычайных ситуациях, и также правовые и организационные вопросы обеспечения безопасности, связанные с выполнением работ по разработке и эксплуатации решения.

В рамках выпускной работы магистра выполняется проектирование и разработка архитектуры искусственных нейронных сетей. Данное решение призвано классифицировать отношения между пользователями социальной сети «Twitter».

Потребность в подобном решении обусловлена наличием спроса в анализе больших объемов данных в интернете. Данная система призвана полностью автоматизировать процесс анализа сообщений между пользователями социальных сетей.

6.1. Производственная безопасность

Поскольку все работы при разработке и эксплуатации решения выполняются в маленьком жилом помещении с постоянно работающими электронными устройствами, то основными источниками вредных и опасных факторов являются электронно-вычислительные устройства и элементы электрической сети этого помещения. Перечень вредных и опасных факторов, характерных для текущего решения представлены в таблице 5.1.

Таблица 5.1 – Опасные и вредные факторы при выполнении работ по проектированию и разработке алгоритма нейронной сетей

Источник фактора, наименование видов работ	Факторы (по ГОСТ 12.0.003-74)		Нормативные документы
	Вредные	Опасные	
Рабочим местом является жилое помещение. В помещении рабочим местом является место за персональным компьютером. Технологический процесс представляет собой работы пользователя с программным продуктом.	1. Повышенный уровень электромагнитных излучений. 2. Отклонение показателей микроклимата в помещении. 3. Недостаточная освещенность. 4. Повышенный уровень шумов. 5. Психофизические факторы.	1. Электрический ток. 2. Короткое замыкание. 3. Статическое электричество.	1.Уровень электромагнитного излучения: СанПиН 2.2.2/2.4.1340-03 и СанПиН 2.2.4.1191-03. 2. Требования к микроклимату: СанПиН 2.2.4.548–96. 3. Освещение – СП 52.13330.2011. 4. Шумы – СН 2.2.4/2.1.8.562-96.

6.1.1. Повышенный уровень электромагнитных излучений

Основные работы, связанные с разработкой и эксплуатацией решения, выполняются в жилом помещении, где находятся различные технические приборы: персональные компьютеры, факсы, принтеры, сканеры, мобильные устройства, электрическая проводка и прочее. Все эти устройства являются источниками электромагнитных излучений, которые вызывают у человека функциональные нарушения нервной системы, слабость, раздражительность, быструю утомляемость, ослабление памяти, нарушение сна и многое другое.

Гигиенические требования к персональным электронно-вычислительным машинам и организации работы СанПиН 2.2.2/2.4.1340-03 определяют величину напряженности электромагнитного поля и плотность магнитного потока: в диапазоне 5 Гц ÷ 2 кГц эти величины не должны превышать 25 В/м и 250 нТл соответственно; в диапазоне 2 кГц ÷ 400кГц - 2,5 В/м и 25 нТл.

Ниже перечислены способы, позволяющие уменьшить действие электромагнитного излучения на организм человека:

- следует установить монитор компьютера на расстоянии 60-80 см от глаз, но не менее 50 см. А системный блок компьютера на максимально возможное расстояние;
- необходимо делать 15-ти минутные перерывы во время работы за компьютером каждые 2 часа;
- по окончанию рабочего дня следует отключить от сети все возможные электрические устройства;
- мобильные телефоны стоит отложить на максимально возможное расстояние;
- оргтехнику нужно разместить на расстоянии не менее 1.5 от рабочего места:
- в случае мощных электромагнитных излучений, следует использовать средства защиты, ограничивающие поступление электромагнитного излучения на рабочее место: специальных экранов, поглотителей излучений и других средств индивидуальной защиты.

Считается, что одним из самых эффективных способов, позволяющих избавится от действий электромагнитных излучений является прогулка по свежему воздуху. Работникам следует регулярно бывать в лесу, гулять в парке, ходить в походы и т.д.

6.1.2. Отклонение показателей микроклимата в помещении

Микроклимат помещения – это комплекс метеорологических условий в данном помещении.

К показателям, характеризующим микроклимат в производственных помещениях, относятся:

- температура воздуха;
- влажность воздуха;
- скорость движения воздуха;
- температура поверхностей;
- тепловое облучение (при наличии источников лучистого тепла).

Постоянное отклонение от нормальных параметров микроклимата приводит к перегреву или переохлаждению человеческого организма и связанным с ними негативным последствиям: при перегреве – к обильному потоотделению, учащению пульса и дыхания, резкой слабости, головокружению, появлению судорог, а в тяжелых случаях – возникновению теплового удара. При переохлаждении возникают простудные заболевания, хронические воспаления суставов, мышц и др.

По степени физической тяжести работа оператора ЭВМ относится к категории лёгких работ. В соответствии с временем года и категорией тяжести работ определены параметры микроклимата согласно требованиям [6] и приведены в Таблице 5.2.

Таблица 5.2. – Оптимальные величины показателей микроклимата на рабочих местах

Период года	Категория работ	Температура воздуха, С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Категория 1а	23-25	40-60	0.1
Теплый	Категория 1а	20-22	40-60	0.1

Таблица 5.3. – допустимые величины показателей микроклимата на рабочих местах

Период года	Температура воздуха, С	Относительная	Скорость движения воздуха, м/с

Категория работ	Ниже оптимальных, не более	Выше оптимальных, не более	влажность воздуха, %	Ниже оптимальных, не более		Выше оптимальных, не более
Холодный	Категория 1а	20-21.9	24.1-25	15-75	0.1	0.1
Теплый	Категория 1а	21-22.9	25.1-28	15-75	0.1	0.2

Рабочее место соответствует нормам микроклимата «СанПиН 2.2.4.548-96» о допустимой величине показателей микроклимата, так как в зимнее время в помещении предусмотрена система отопления. Она обеспечивает достаточное, постоянное и равномерное нагревание воздуха. Соответственно в летнее время в помещении предусмотрена система вентиляции, обеспечивающий приток прохладного воздуха, а также предусмотрены мероприятия по проветриванию помещения.

6.1.3. Недостаточная освещенность рабочей зоны

Производственное освещение — неотъемлемый элемент условий трудовой деятельности человека. При правильно организованном освещении рабочего места обеспечивается сохранность зрения человека и нормальное состояние его нервной системы, а также безопасность в процессе производства. Производительность труда и качество выпускаемой продукции находятся в прямой зависимости от освещения.

Рабочая зона или рабочее место оператора ЭВМ освещается таким образом, чтобы можно было отчетливо видеть процесс работы, не напрягая зрения, а также исключается прямое попадание лучей источника света в глаза. Кроме того, уровень необходимого освещения определяется степенью

точности зрительных работ. Наименьший размер объекта различения составляет 0.5 - 1 мм. В помещении присутствует естественное освещение. По нормам освещенности [8] и отраслевым нормам, работа за ПК относится к зрительным работам высокой точности для любого типа помещений.

Требования к освещению на рабочих местах, оборудованных ПК, представлены в таблице 5.4.

Таблица 5.4. – требования к освещению на рабочих местах

Освещенность на рабочем столе	300-500 лк
Освещенность экрана ПК	Не выше 300 лк
Блики на экране	Не выше 40 кд/м2
Прямая блескость источника света	200 кд/м2
Показатель ослеплённости	Не более 20
Показатель дискомфорта	Не более 15
Отношение яркости	
– между рабочими поверхностями	3:1 – 5:1
– между поверхностями стен и оборудования	10:1
Коэффициент пульсации	Не более 15%

6.1.4. Повышенный уровень шума

Одним из важных факторов, влияющих на качество выполняемой работы, является шум. Шум ухудшает условия труда, оказывая вредное действие на организм человека. Работающие в условиях длительного шумового воздействия испытывают раздражительность, головные боли, головокружение, снижение памяти, повышенную утомляемость, понижение аппетита, боли в ушах и т. д. Такие нарушения в работе ряда органов и систем

организма человека могут вызвать негативные изменения в эмоциональном состоянии человека вплоть до стрессовых.

Под воздействием шума снижается концентрация внимания, нарушаются физиологические функции, появляется усталость в связи с повышенными энергетическими затратами и нервно- психическим напряжением, ухудшается речевая коммутация. Все это снижает работоспособность человека и его производительность, качество и безопасность труда.

Нынешняя работа относится к первой категории трудовой деятельности СН 2.2.4/2.1.8.562-96 [9]: «Творческая деятельность, научная деятельность, конструирование и проектирование, программирование». Предельно допустимым уровнем звукового давления для данной трудовой деятельности принято считать 50 дБА. Допустимые уровни звукового давления в помещениях жилых и общественных зданий не должно превышать 55 дБА в период с 7 до 23ч, и 45 дБА в период с 23 до 7ч.

На рабочем месте уровень шума соответствует санитарным нормам СН 2.2.4/2.1.8.562-96 [9], так как работа данной магистерской диссертации подразумевает работу только за персональным компьютером. Проектирование и разработка программного обеспечения проводится в жилом помещении, где расположен только 1 персональный компьютер, уровень шума от электронных приборов минимален. Другие электронные приборы, производящие шумы (принтер), расположены на безопасном расстоянии и приводятся в использование очень редко.

6.1.5. Психофизиологические факторы

Как известно, любой вид деятельности человека порождает возникновение различных видов опасностей. Наибольшее количество опасностей возникает, в первую очередь, в процессе трудовой деятельности.

Это обусловлено двумя причинами: в течение суток человек занимается трудовой деятельностью (работа, учеба, спорт, активный отдых и т д), то есть повышается вероятность проявления опасностей; производственные процессы, в которых осуществляется преобразование веществ, энергии и информации и возникают основные техногенные опасности.

В любой трудовой деятельности человека можно выделить два компонента: физиологический и психический.

Физиологический компонент связан с физиологическими возможностями каждого индивидуума и определяется работой его мышц, системы кровообращения, дыхания, сердечно-сосудистой системы, опорно-двигательного аппарата. Действие этих систем координируется центральной нервной системой. В этом процессе используется большое количество энергии, кислорода для активизации обменных процессов. Отрасль физиологии, которая изучает изменения функционального состояния человека в зависимости от характера и типа трудовой деятельности и разрабатывает оптимальные режимы (условия) труда и отдыха, называется физиологией труда.

Психический компонент определяется психическими процессами и психическими свойствами личности. Психологи выделяют познавательные процессы, с помощью которых человек познает мир (ощущения, восприятия, внимание, память, воображение, мышление и речь), и психические свойства (или состояние личности), которые регулируют общение людей друг с другом, непосредственно руководят поступками и действиями. Психологические состояния отличаются разнообразием и характером. Они обусловливают особенности психической деятельности в конкретный период времени и могут положительно или отрицательно влиять на протекание всех психических процессов.

Меры по устранению психофизиологических факторов:

- соблюдать чистоту и порядок на рабочем месте;
- не работать в условиях повышенного шума;
- не нарушать инструкции по техники безопасности;
- делать перерыв каждые 2-3 часа;
- соблюдать рекомендации при работе с электронными устройствами (рекомендуемое расстояние до монитора, правильная осанка, правильное расположение устройств ввода и т.д.).

6.1.6. Электрический ток

В процессе использования электроприборов и электрооборудования может возникнуть опасность поражения электрическим током. По опасности поражения током рабочая зона относится к помещениям без повышенной опасности. Чтобы исключить опасность поражения необходимо соблюдать следующие правила электробезопасности:

- перед включением прибора в сеть должна быть визуально проверена его электропроводка на отсутствие возможных видимых нарушений изоляции, а также на отсутствие замыкания токопроводящих частей на корпус;
- при появлении признаков замыкания необходимо немедленно, отключить от электрической сети устройство и устранить неисправность;
- запрещается при включенном устройстве одновременно прикасаться к приборам, имеющим естественное заземление (например, радиаторы отопления, водопроводные краны и др.)
- запрещается эксплуатация оборудования в помещениях с повышенной опасностью;

- запрещается включать и выключать устройство при помощи штепсельной вилки. Штепсельную вилку включать и выключать из розетки можно только при выключенном устройстве.

Существуют следующие способы защиты от поражения током в электроустановках:

- предохранительные устройства;
- защитное заземление;
- применение устройств защитного отключения (УЗО);
- зануление.

Самый распространенный способ защиты от поражения током при эксплуатации измерительных приборов и устройств - защитное заземление, которое предназначено для превращения "замыкания электричества на корпус" в "замыкание тока на землю" для уменьшения напряжения прикосновения и напряжения шага до безопасных величин (выравнивание потенциала).

6.2. Экологическая безопасность

Разработка и эксплуатация решения подразумевает в основном использование электронно-вычислительных и сетевых устройств. Пользуясь устройствами, соответствующими санитарным нормам и стандартам экологической безопасности, позволит исключить влияние на окружающую среду.

Экологическая безопасность и охрана окружающей среды являются одними из важнейших факторов при выполнении работ любого характера. При работе в офисном помещении за персональным ПК отсутствуют выбросы в окружающую среду и нет влияния на жилищную зону.

Поскольку при разработке данной магистерской диссертации использовался компьютер, необходимо помнить о правильной утилизации компьютерного лома после выхода из строя данного ПК. В соответствии с постановлением правительства №340 [4] юридическим лицам запрещено самостоятельно утилизировать компьютерную технику. Необходимо найти организацию, которая занимается утилизацией в частном порядке. Это относится к следующим видам отходов:

- образование твердых отходов, относящихся к IV классу опасности (системный блок компьютера, принтеры, сканеры, клавиатура, манипулятор "мышь") и жидких отходов; образование твердых отходов, относящихся к IV классу опасности (системный блок компьютера, принтеры, сканеры, клавиатура, манипулятор "мышь") и жидких отходов;
- Жидкие отходы: сточные воды;
- Люминесцентные лампы.

6.3 Безопасность в чрезвычайных случаях

6.3.1. Анализ вероятных ЧС, которые могут возникнуть на рабочем месте

Наиболее вероятной чрезвычайной ситуацией в жилом помещении является возникновение пожара или взрыва. Данная ситуация может возникнуть по ряду причин: короткое замыкание в электрической проводке, являющееся следствием нарушения изоляции, электросоединений и электрораспределительных щитов; возгорание электрических устройств, по причине внутренней неисправности; возгорание мебели и устройств искусственного освещения.

В основном возникновение пожара является следствием нарушения правил пожарной безопасности и правил эксплуатации электрических устройств.

Технический регламент о требованиях пожарной безопасности помещения по пожарной и взрывной опасности определяет жилое помещение, в котором выполнялись работы по разработке решения, как в категории В – умеренная пожароопасность.

6.3.2. Мероприятия по предотвращению ЧС

Существует ряд мероприятий, позволяющие уменьшить вероятность возникновения пожаров. К первым относятся эксплуатационные мероприятия, в основе которых лежит выбор и использование современных автоматических средств сигнализации, автоматических стационарных систем и первичных средств пожаротушения, разработка методов и применение устройств ограничения распространения огня и т.д.

Ко вторым относятся организационные мероприятия, направленные на обучение сотрудников правилам пожарной безопасности, разработку и реализацию норм и правил пожарной безопасности, инструкций эксплуатации рабочего оборудования, планов эвакуации и прочих (рисунок 6.1).

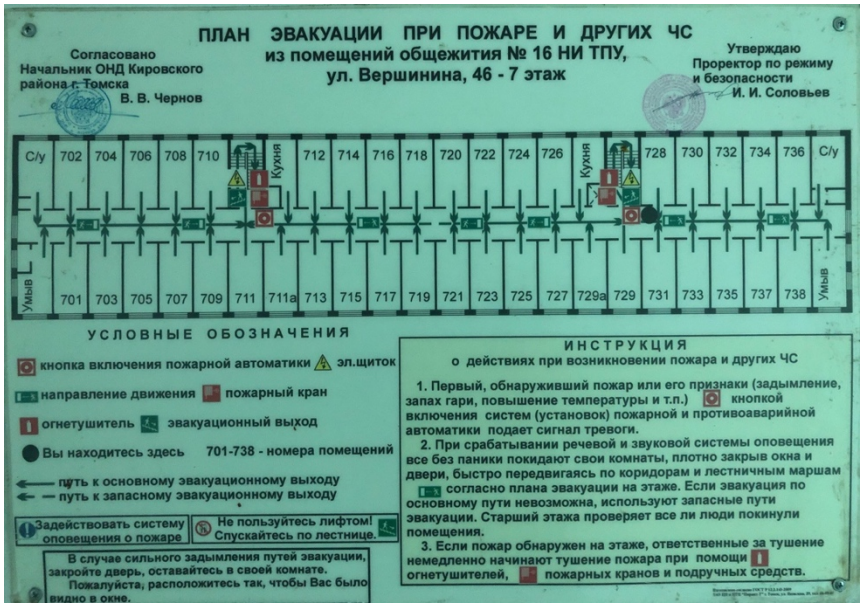


Рисунок 6.1 – план эвакуации при ЧС

6.4. Правовые и организационные вопросы обеспечения безопасности

Организация рабочего места программиста предполагает соблюдение ряда условий: оптимальное размещение оборудования, входящего в состав рабочего места и достаточное рабочее пространство, позволяющее осуществлять необходимые движения и перемещения. Взаимное расположение всех элементов рабочего места должно соответствовать физическим и психологическим требованиям, мониторы персональных компьютеров должны быть расположение по отношению к источникам естественного света сбоку, преимущественно слева.

Основными элементами рабочего места является стол и кресло, рациональный подбор этих элементов позволяет создать комфортную рабочую обстановку.

Рабочий стол должен соответствовать следующим требованиям:

- высота стола должна быть выбрана с учетом возможности сидеть свободно, в удобной позе, при необходимости опираясь на подлокотники;
- нижний части стола должно быть предусмотрено пространство для ног, высотой не менее 60 см, шириной – не менее 50 см, глубиной на уровне колен – не менее 45 см и на уровень вытянутых ног – не менее 65 см;
- поверхность стола не должна создавать бликов;
- конструкция стола должна предусматривать наличие выдвижных ящиков, для хранения документаций и канцелярских принадлежностей.

Рекомендуемая высота сиденья рабочего кресла над уровнем пола должна находится в пределах 42-50 см. поверхность сиденья мягкая, ширина и глубина поверхности сиденья не менее 40 см, передний край закругленный, а угол наклона спинки – регулируемый.

Стоит также отметить, что для комфортной и качественной работы на компьютере существенное значение имеют размеры знаков, плотность их размещения, контраст и соотношение яркостей символов и фона экрана. Для

рекомендуемого расстояние от глаз работника до монитора, лежащего в диапазоне 60-80 см, высота знака должна быть не менее 3 мм, оптимальное соотношение ширины и высоты знака составляет 3:4, расстояние между знаками – 15-20% их высоты и соотношение яркости фона экрана и символов должно находится в пределах от 1:2 до 1:15.

Вывод по главе

В данной главе диссертационной работы были рассмотрены вопросы обеспечения безопасных, безвредных условий труда, необходимых для комфортного написания дипломной работы. Также в главе были выделены факторы, оказывающие вредное и опасное влияние на студента в ходе написания работы.

В результате проведенной работы было выявлено, что помещение, где производилась разработка текущей ВКР, является помещением без повышенной опасности по степени вероятности поражения электрическим током. Рабочее место студента оснащено достаточным для комфортной работы освещением, уровень шума соответствует нормам.

С точки зрения комфортности микроклимата рассматриваемого помещения в летнее время есть смысл применить искусственную (механическую) вентиляцию (кондиционеры).

Также с точки зрения пожарной безопасности, рабочее место соответствует необходимым нормам.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы был разработан программный сервис для автоматизации процесса классификации отношений между пользователями социальной сети Twitter на основе анализа текста сообщений, создан тренировочный набор данных, разработан классификатор на основе нейронной сети архитектуры LSTM. Для повышения точности работы классификатора и решения проблемы подачи на вход нейронной сети двух сообщений, был предложен способ конкатенации пар сообщений.

В целях обеспечение безопасности были введены две группы доступа к программному сервису: гость и администратор. Помимо всех тех возможностей, что обладает гость, администратор обладает возможностью изменять настройки и тренировочные данные нейронной сети, а также совершать повторный процесс тренировки.

В результате проведенного тестирования различных настроек, были выбраны оптимальные параметры нейронной сети – количество слоев, количество блоков LSTM, шаг обучения, метод оптимизации. Реализованный в ходе работы классификатор обладает точностью классификации равной 82-85%. Значения точности получены в ходе использования тестовых данных, не входящих в тренировочный набор.

В ходе дальнейшего развития программного сервиса планируется улучшение точности классификатора на данных, не входящие в тренировочный и тестовый набор. Для реализации этой цели требуется собрать больший набор тренировочных данных различных областей. А также для значительного улучшения алгоритма классификации можно расширить функционал классификатора для распознавания третьего класса: «нейтральный» – когда сообщение не является ни согласием, ни несогласием.

СПИСОК ЛИТЕРАТУРЫ

1. Stefano Menini, Sara Tonelli. Agreement and Disagreement: Comparison of Points of View in the political domain // Association for Computational Linguistic. – 2014.
2. Amita Misra & Marilyn A. Walker. Topic Independent Identification of Agreement and Disagreement in Social Media Dialogue // Proceedings of the SIGDIAL 2013 Conference – 2013г. – с. 41-50.
3. Банокин П.И., Лунева Е.Е., Ефремов А.А., Кочегурова Е.А. Исследование применимости рекуррентных сетей LSTM в задаче поиска экспертов пользователей социальных сетей – 2015г.
4. Kim Y. Convolutional Neural Networks for Sentence Classification // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. – Stroudsburg, USA: Association for Computational Linguistics – 2014г. – с. 1746-1752.
5. Dos Santos C. N., Gatti M. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts // COLING. – 2014. – с. 69-78.
6. Pengfei Liu, Xipeng Qiu, Xuanjing Huang // Recurrent Neural Network for Text Classification with Multi-Task Learning. – Twenty-Fifth International Joint Conference on Artificial Intelligence – 2017г. – с. 2873-2879.
7. Understanding LSTM Networks // colah.github.com URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs (дата обращения: 01.05.2018).
8. Advantages and Disadvantages of Django // hackernoon URL: https://hackernoon.com/advantages-and-disadvantages-of-django-499b1e20a2c5 (дата обращения: 01.05.2018).
9. Docs – Twitter Developers // Twitter URL: https://dev.twitter.com/docs (дата обращения: 01.05.2018).
10. Tweepy Documentation // Tweepy URL: http://docs.tweepy.org/en/v3.5.0/ (дата обращения: 01.05.2018).

11. Comparison of Deep Learning Libraries // Kdnuggets URL: https://www.kdnuggets.com/2017/03/getting-started-deep-learning.html (дата обращения: 01.05.2018).
12. Perform sentiment analysis with LSTMs, using TensorFlow // O'Reilly Media URL: https://www.oreilly.com/learning/perform-sentiment-analysis-with-lstms-using-tensorflow (дата обращения: 01.05.2018).
13. Sak H., Senior A., Beaufays F. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling // INTERSPEECH-2014. - Singapore: ISC, 2014. - –°. 338-342.
14. Razvan Pascanu, Tomas Mikolov , Yoshua Bengio. On the difficulty of training Recurrent Neural Networks.
15. Distinctive features of SQLite // SQLite URL: https://www.sqlite.org/different.html (дата обращения: 01.05.2018).
16. Digital Ocean: Cloud Computing // DigitalOcean URL: www.digitalocean.com (дата обращения: 01.05.2018).
17. Bekzat-Shayakhmetov's Master Thesis // B.S. URL: www.bekzat-shayakhmetov.me (дата обращения: 01.05.2018).
18. Domain Name Registration // NameCheap URL: https://www.namecheap.com (дата обращения: 01.05.2018).
19. NGINX | High performance web server // nginx URL: https://www.nginx.com (дата обращения: 01.05.2018).
20. Gunicorn – Python WSGI HTTP server for Unix // Gunicorn URL: http://gunicorn.org (дата обращения: 01.05.2018).
21. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning // MachinelearningMastery URL: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning (дата обращения: 01.05.2018).
22. Word Embeddings // Wikipedia URL: https://en.wikipedia.org/wiki/Word_embedding (дата обращения: 01.05.2018).

23. ГОСТ 12.0.003-74. Опасные и вредные производственные факторы. Классификация.
24. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.
25. Безопасность жизнедеятельности. Электробезопасность на предприятиях ИО. [Электронный ресурс]: 2010. – Режим доступа: http://www.bezzhd.ru, свободный.
26. Постановление правительства РФ №340. О порядке установления требований к программам в области энергосбережения и повышения энергетической эффективности организаций, осуществляющих регулируемые виды деятельности. [Электронный ресурс]: 2010. – Режим доступа: http://pravo.gov.ru/proxy/ips/?docbody=&nd=102138354, свободный.
27. ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования. – М.: Госстандарт России, 1987.
28. СанПиН 2.2.4.548 – 96. Гигиенические требования к микроклимату производственных помещений. М.: Минздрав России, 1997.
29. СанПиН 2.2.1/2.1.1.1278 – 03. Гигиенические требования к естественному, искусственному и совмещённому освещению жилых и общественных зданий. М.: Минздрав России, 2003.
30. СП 52.13330.2011. Свод правил. Естественное и искусственное освещение.
31. СН 2.2.4/2.1.8.562 – 96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории застройки.
32. СанПиН 2.2.2/2.4.1340 – 03. Санитарно-эпидемиологические правила и нормативы «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы». – М.: Госкомсанэпиднадзор, 2003.

Приложение А

(Обязательное)

Identification of relationships between users of the Twitter social network based on messages analysis

Студент:

Группа	ФИО	Подпись	Дата
8ВМ6Г	Шаяхметов Бекзат Мейрамбайулы		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Цапко Ирина Валериевна	к.т.н.		

Консультант-лингвист отделения иностранных языков ШБИП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст. преподаватель	Кудряшова Александра Владимировна			

Introduction

Every day a huge amount of content appears on the World Wide Web: the polls are conducted, opinions are expressed, and disputes are appeared. The growth of various discussion forums and social networks has provided people with new ways for to express their opinions. In discussions, participants often agree or disagree with the views of each other. Sometimes in such discussions there is a task to classify the relationships between users of social networks based on the analysis of messages. This task is being solved in order to conduct any sociological surveys, marketing researches, identifying the presence of disputes and the ideological positions of users. But manual analysis of the huge data is often very difficult and expensive. In order to automate this problem various methods of natural language analysis are used.

One of the most common tasks of machine learning is the sentiment analysis, i.e. evaluation of the emotional tone of a text. Nevertheless, the task of message analysis on presence of agreement/disagreement is not widespread. This problem has been sanctified in the articles [1, 2] of English-speaking authors. In these works [1, 2] only a brief overview of the message analysis methods has been made, but the implementation of these methods is not sanctified.

Due to the fact that there is a lack of software implementations of the agreement/disagreement classifiers, it becomes difficult to create training data for the deep machine learning.

Classifying messages for the presence of agreement/disagreement sentiments is not a trivial task. The standard methods of classifications – the Bayes method and the method of support vector machines – are unable to "understand" the meaning of the messages in order to determine whether the comment is about agreement or disagreement. These methods do not take into account the order of words in sentences.

This problem is solved by the methods of machine learning. They allow us to algorithmically understand the structure of sentences and how words are related with

each other. The task of neural network methods is not to understand each word, but rather to understand the sequence of these words.

The purpose of this master's work is the implementation of the classifier of relations between users of the Twitter social network using the analysis of their messages.

To achieve the current goal of the current work, it is necessary to solve the following tasks:

1. choose a neural network architecture for solving problems of natural language analysis;

2. adapt the chosen architecture for solving problems of identifying shades of agreement or disagreement in messages;

3. create a training data sets;

4. check the accuracy of the classifier;

5. develop a web application for working with the classifier.

1. Analytics review

With the growth of the popularity of neural networks and the messages analysis in general, user requirements to a software product are increasing. Nowadays, the software systems with more accurate results are more valuable than the systems, whose main feature is the speed of execution.

There are various methods for analyzing or classifying a natural language, several main methods are:

1. The naive Bayes classifier.

2. Support vector machine method

3. Methods of deep machine learning;

According to the author of the article [2], the simplest methods of natural language analysis are not able to take into account the order of words in text. These methods are the Bayes method of naive classifier and the support vector machine method.

In the articles of the authors [2, 3, 4] it is stated that for the message analysis it is better to use classifiers based on artificial neural networks. Convolutional neural networks (CNN) and neural networks of long-short term memory (LSTM) are considered as popular architectures for analysis of sequences of words.

A distinctive feature of the deep learning method from the other two is that this method is able to work with input data of non-fixed length, and also the fact that in comparison with methods of naive Bayes classifier and support vector machine where the words in the sentences are analyzed separately, in neural networks of CNN and LSTM architectures the main target of analysis is the sequence of words.

According to the author [3], the use of deep learning methods gives 75% - 95% accuracy depending on what training data sets are being used. Having the maximum of accuracy is the main priority nowadays.

1.1. Artificial neural networks

Artificial neural networks or artificial intelligence (AI) are perhaps one of the most exciting technologies of the decade. AI has already succeeded in various areas of the scientific field: from the speech recognition systems to the classification of various types of cancer and genetic engineering. A neural network is a sequence of neurons connected by synapses. The concept of a neuron came into programming straight out of biology. Due to its structure, machines are able to analyze and even remember information. AI is able not only to analyze incoming information, but also to reproduce it from its memory.

A neuron is a computing unit that receives information at the input, makes some calculations with this information, and passes it to the output. Neurons are divided into 3 types: input, hidden, output.

A synapse is the connection between two neurons. Synapses have only one parameter – weight, which modifies the input information during the transmission from one neuron to another.

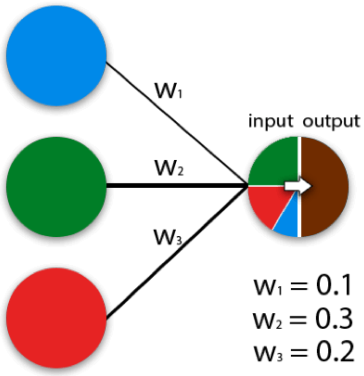


Figure 1 Weights of neuron

Activation functions

Activation function is the normalization methods of input data. The most popular activation functions are:

1. Sigmoidal activation function

$$f(x) = \frac{1}{1 + e^{-x}}$$

This function varies from [0; 1] and has the shape of "S" letter. The sigmoid function is very simple in understanding and use.

2. Hyperbolic tangent

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

The hyperbolic tangent function is centralized relative to 0, since it varies from [-1; 1]. Consequently, optimization is much easier, and this function is more preferable in comparison with sigmoidal.

3. Activation function ReLu (Rectified Linear units)

$$f(x) = \max(0, x)$$

In recent years, the activation function "rectifier" has gained huge popularity. The activation function "ReLu" does not require large computational resources – its

implementation is possible with a simple threshold transformation of the activation matrix at zero.

1.2. Recurrent neural networks (RNN)

People do not start thinking from scratch every time. While reading any literature, any person understands every word based on the understanding of the previous ones. Human's thoughts have consistency.

Traditional neural networks are not able to reproduce this architecture, and this is perhaps their biggest drawback. For example, imagine that you want to classify an event that occurs at a certain point of the movie. It is unclear how classical neural networks can use their understanding about previous events of the film to understand the subsequent ones.

Recurrent neural networks are designed to solve this problem. RNN is a network with a built-in loop inside, which allows you to save information without throwing it away (Fig. 2).

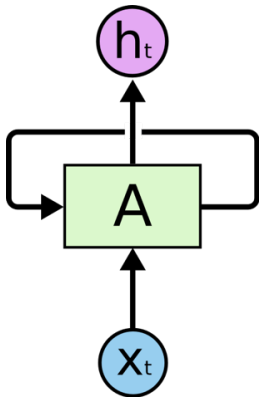


Figure 2 Structure of RNN

Figure 2 shows a part of the neural network "A", which takes input "x_t " and outputs "h_t". The loop allows you to pass information from one step to another. At first glance, it seems that the recurrent neural networks are a bit confusing and "magical", but if you look at it deeper, you will realize there is nothing complicated. Recurrent neural networks can be considered as the copies of the same network, each transmitting a subsequent message (Fig. 3).

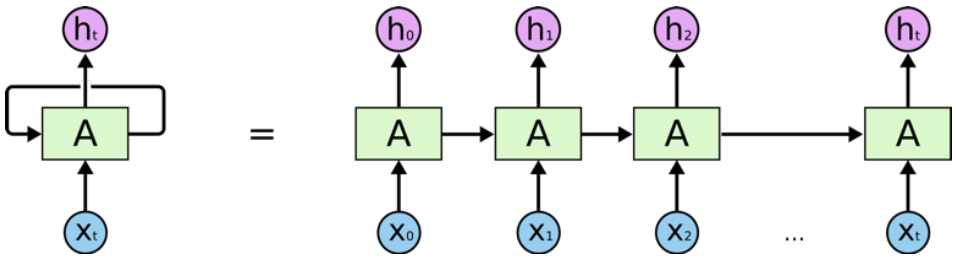


Figure 3 Insides of RNN

This chain-shaped structure suggests that recurrent neural networks are closely related to sequences and lists.

Problem of long-term dependencies

One of the greatest strengths of the RNN is its ability to link the previous information with the current tasks. For instance, understanding previous video frames can help to understand the current frame.

Sometimes we do not need a lot of information to perform specific tasks. For example, consider a language model whose task is to anticipate the last word in the sentence. Let's suppose that there is a sentence such as "a rainbow appeared in the *sky*." In this example, we do not need a further context for determining desired word in the sentence, since it is clear without context that the next word should be "sky". In such cases, the distance between the desired information and the point where it is needed is not large, and the RNN is able to learn to use nearby information (Fig. 4).

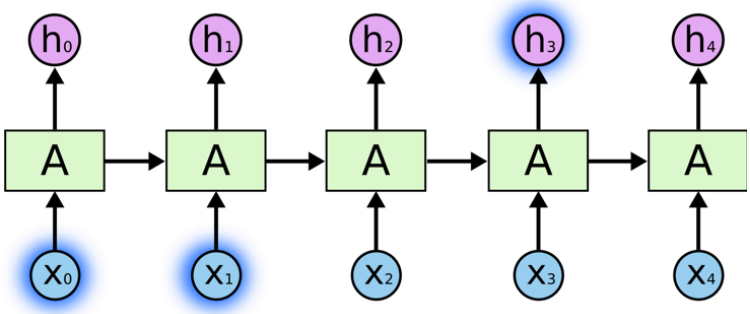


Figure 4 Gap between the relevant information and the place it needs

However, there are tasks where a larger context is required, for example, in a sentence like "I grew up in France... I speak French fluently." The nearby information says that the next word should be the name of the language, but if we want to close in, then we need information about France from the previous sentence.

It is likely that the gap between the relevant information and the point where this information is needed will be large (Fig. 5). However, with the growth of this gap, a neural network loses the ability to connect and understand this information.

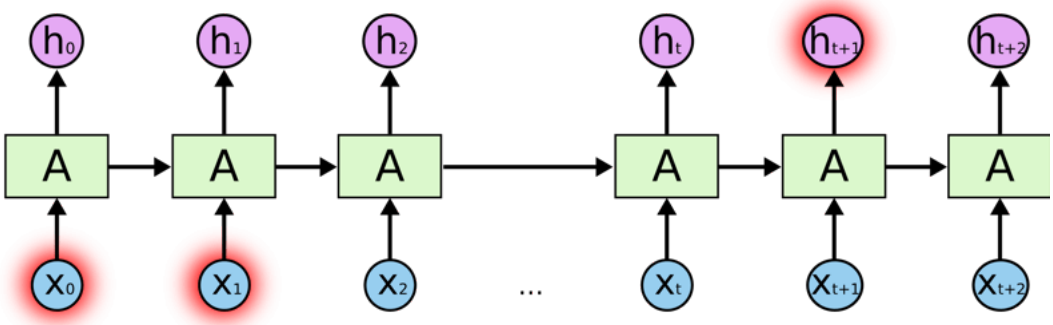


Figure 5 Increase of the gap

Fortunately, the LSTM architecture of recurrent neural networks does not have such problems with long-short term dependencies

1.3. Long-short term memory networks (LSTM)

Long-short term memory networks (LSTM networks) are a specific kind of recurrent neural networks able to learn long-term dependencies. They were presented by Sepp Hohrater and Jürgen Schmidtchuber in 1997 and have been refined and popularized by many people in subsequent works. LSTM networks are widely used in solving many modern problems.

Networks of long-term memory were originally designed to solve the problems of long-term dependence. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

All recurrent neural networks have the form of a chain consisting of repeating modules of the neural network. In the classic RNN this repeating module has the simplest structure in the form of a layer with the activation function "tanh" (hyperbolic tangent) (Fig. 6).

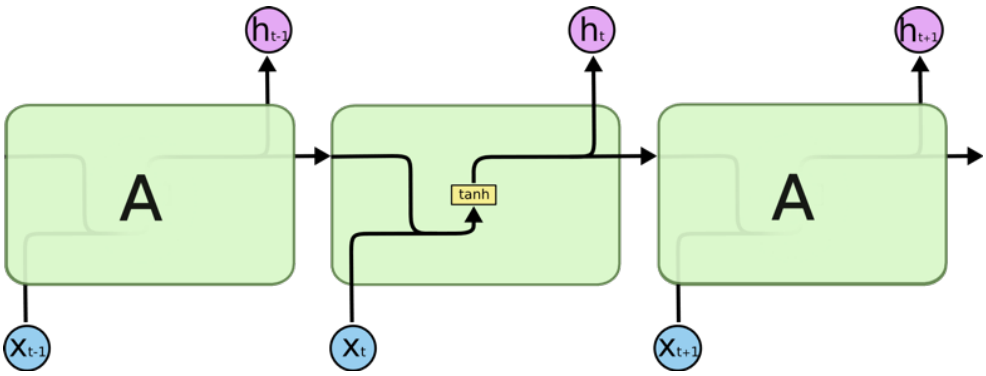


Figure 6 Repeating unit in the standard RNN

LSTM networks also have a form of a chain with repeating modules, but unlike a standard RNN, LSTM networks have a slightly different structure – instead of one layer with the "tanh" activation function, long-memory networks have four layers interacting in a special way (Fig. 7).

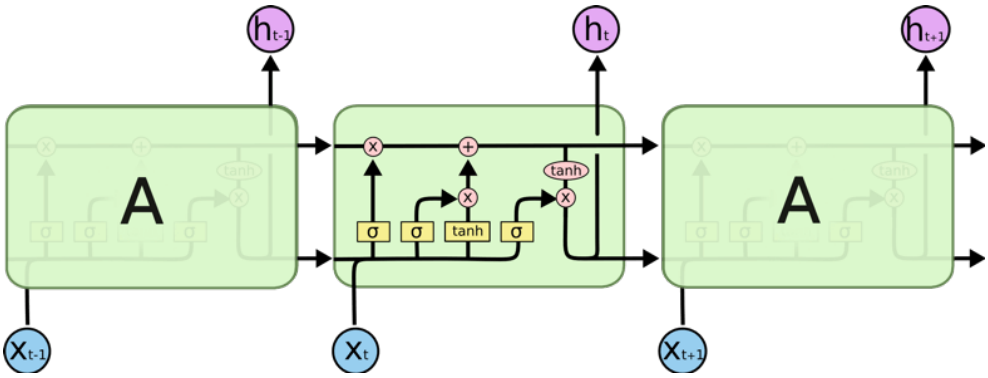


Figure 7 Repeating module in LSTM networks

The main idea of LSTM

The main aspect of LSTM networks is a cell state, which is a horizontal line that runs at the very top of the diagram (Fig. 8). The cell state is similar to a conveyor belt. It runs straight through the entire chain with only small linear interactions.

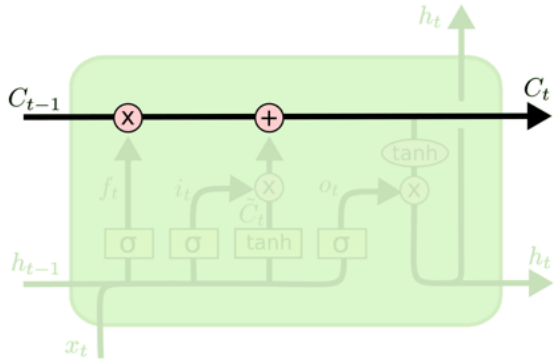


Figure 8 LSTM networks cell's state

Long-memory networks have the ability to delete or add information to a state cell using structures called filters.

Filters "decide" whether to pass information further or throw it away. They consist of a sigmoidal grid layer and a point multiplication operation.

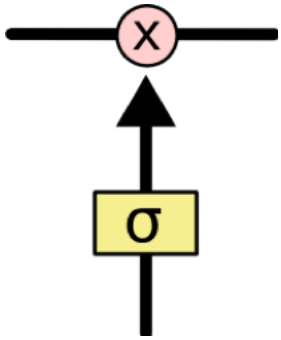


Figure 9 Structure of filters

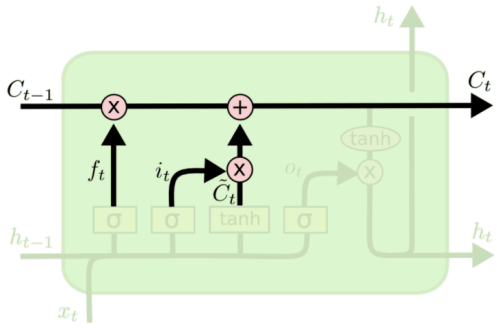
The sigmoidal layer outputs values in the interval between 0 and 1, describing how much this element is important to be memorized. A value of 0 means "do not let anything further", the value 1 says "pass it further". Long-short memory networks have 3 filters of this kind, to protect and control a cell's state.

The first step in our LSTM is to decide what information we're going to throw away from the cell state. This solution takes a sigmoidal layer called the forget gate layer. At the input, it receives the values h_{t-1} and x_t and returns values in the interval between 0 and 1 for each element in the cell of the state C_{t-1}. As mentioned above, 1 denotes that the information is 100% important and is required in the future, 0 signals that network should discard / forget this information.

The next step is to decide which new information will be stored in the cell state. This stage consists of 2 parts. First of all, the sigmoidal layer called the "input filter layer" decides which values need updating. Next, the hyperbolic tangent layer constructs a vector of new candidate values C_t, which can be added to the state cell in the future.

Now it is time to update the old cell's state C_{t-1} to the new one C_t. We have already decided what to do on the previous stages, the last task to carry out is to fulfill it.

We must multiply the old state by f_t, forgetting the things we decided to forget earlier. Then add $i_t * C_t$. These are new candidate values, scaled by how much we decided to update each state value.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 10 Deeper view of the cell

Finally, we need to decide what we're going to output. The output data will be based on our cell state, some of filters will be applied to them. Firstly, a sigmoidal layer is in use, which decides what kind of information to be returned from the cell state. Then, the cell state values are being passed through the tanh-layer in order to receive output values ranging from -1 to 1 and then they get multiplied by the output values of sigmoidal layer. This helps to display only the desired information.

1.4. AI training algorithms

There are many different algorithms for artificial neural networks to be trained; each of them has its own features. The main purpose of the training process is to constantly adjust the NN weights using some learning algorithms.

1. Training with a teacher

The neural network training with a teacher assumes that for each input vector from the training set, there is a required value of an output vector, called a target vector. These vectors form a training pair. The network weights are changed until they get a deviation acceptable level of the output vector from the target vector.

2. Training without a teacher

From the perspective of the biological roots of the ANNs, training a neural network without a teacher is a much more natural model of training. The training set

consists of only input vectors. The neural network learning algorithm adjusts the network weights in the way that it will learn by itself how to get to the desired results.

3. Backpropagation method

The main idea of the method is in specifying the desired output for each input example. If the actual output of the network does not match the desired one, the network weights will be corrected respectively. To calculate the correction value, the difference between the actual and the desired output is used. It is important to note that the correction of the weights will happen only in the cases when the error is occurred [2].

1.5. Comparison of machine learning libraries

There is a large number of libraries for implementing machine learning architectures. Frameworks are divided into 2 categories: symbolic and imperative. Symbolic frameworks have an advantage over imperatives in the possibilities of reusing memory, as well as in automatic optimization based on dependencies graph. The most popular symbolic frameworks are TensorFlow and Theano.

One of the advantages of TensorFlow is that it is oriented not only to neural network trainings, so you can use collections of graphs and queues as parts for high-level components. Also, Tensorflow has a transparent modular architecture with a lot of visual representations. Visualization of graphs in Tensorflow is implemented much better than in Theano.

Below is a comparative table of some of the most popular libraries of machine learning [5].

Comparative Table №1 and Figure №11 show that the most popular library for machine learning is a TensorFlow library. It has advantages over its analogues in all listed features.

Table 1 Comparison of machine learning libraries

	Languages	Training data	CNN Support	RNN Support	Ease in use	Speed	GPU Support	Keras Support
Theano	Python, C++	++	++	++	+	++	+	+
TensorFlow	**Python, C++**	+++	+++	++	+++	++	++	+
Torch	Lua, Python	+	+++	++	++	+++	++	
Caffe	C++	+	++		+	+	+	
MXNet	R, Python, Julia, Scala	++	++	+	++	++	+++	
CNTK	C++	+	+	+++	+	++	+	

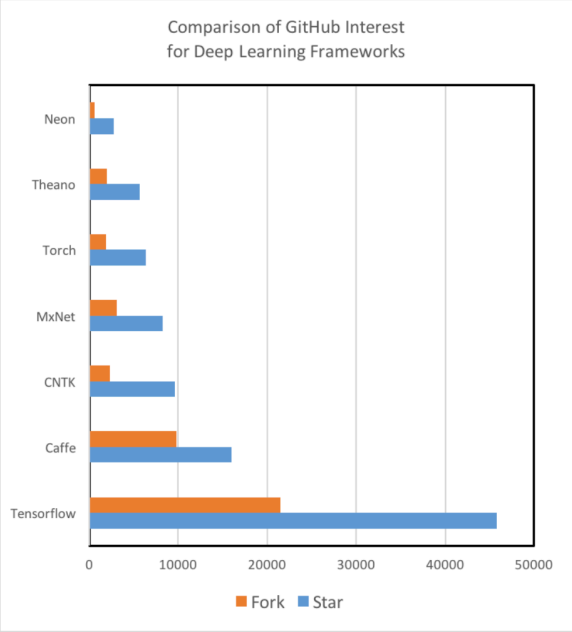


Figure 11 Comparison of libraries popularity

CONCLUSION

As a result of the work, a software service for automation of process of relationships between the users of social networks classification was developed, and a minimum set of training data sets were collected using the designed and implemented software component. The development is based on the text message analysis.

In order to provide safety, two access groups were developed: a guest and an administrator. In addition to all the features that the guest has, the administrator has the ability to change the settings and training data sets of the neural network, as well as perform a training process.

During the research various methods and architectures of natural language analysis were explored and analyzed, and in order to solve the goal, the most suitable method was chosen. The articles [5, 6, 7] state that LSTM neural networks are the most preferred architecture for solving problems of natural language analysis due to their recurrent architecture.

During the implementation of the architecture of LSTM-networks, there was a problem of inability of neural networks to analyze two messages at the same time. In order to solve this problem, the algorithm for message concatenating was implemented – the special separator was used.

During the research, the optimal parameters of the neural network were chosen – the number of layers, the number of LSTM units, the training steps, the optimization method.

The further purpose of this work is to improve the accuracy of the classifier on data sets, which are not included in the training or test sets. To achieve this goal, it is required to collect a larger set of training data from different fields. And also, for a significant improvement in the classification algorithm, it is required to extend the classifier functionality for recognition the third class: "neutral". Neutral class means that the message is neither agreement nor disagreement.

Приложение Б. Листинг программной системы

Листинг компонента классификатор

Файл «train_and_test.py»

```
import os
import re
import datetime
import argparse
# import requests
# from lxml.html import fromstring

from os import listdir
from random import randint
from string import punctuation
from os.path import isfile, join

import numpy as np
import tensorflow as tf

import config

# requests.packages.urllib3.disable_warnings()


class PrepareData():
    """Preparing dataset to be inputed in TF"""

    def __init__(self, path: str):
        self.__dataset_path = path if path.endswith("/") else path
+ "/"
        self.__maxSeqLength = config.maxSeqLength
        self.__current_state = 0
        self.__overall_line_number = 0
        self.__check_idx_matrix_occurance()

    @staticmethod
    def clean_string(string: str) -> str:
        """Cleans messages from punctuation and mentions"""
        seperator = " < - > "
        cleaned_string = ''
        cut_sentence_until = int(config.maxSeqLength/2) -
int(len(seperator)/2)

        #  Delete tweet mentions
        string = re.sub(r"@[A-Za-z0-9]+", "", string)
```

```
        # Replace urls with website titles
        # tweets = string.split(" < - > ")
        # for tweet in tweets:
        #     if len(tweet) < 50:
        #         url = re.search('https?://[A-Za-z0-9./]+', tweet)
        #         if url:
        #             try:
        #                 reponse = requests.get(url.group(0),
verify=False)
        #                 tree = fromstring(reponse.content)
        #                 title = tree.findtext('.//title')
        #                 print(string)
        #                 print(title + "\n")
        #                 string = re.sub('https?://[A-Za-z0-
9./]+',
        #                                 f' {title} ',
        #                                 string)
        #             except Exception as error:
        #                 print(error)
        #     else:
        #         string = re.sub('https?://[A-Za-z0-9./]+', '',
string)

        # Delete the urls
        string = re.sub('https?://[A-Za-z0-9./]+', '',
string.lower())

        # Delete all punctuation marks
        string = string.split(seperator)
        for num, part in enumerate(string, 1):
            for char in part:
                if char not in punctuation:
                    cleaned_string += char
            if num == 1:
                cleaned_string += seperator

        # delete repeated whitespaces (more than 2)
        if re.search(r'\s{2,}', cleaned_string):
            cleaned_string = re.sub(r'\s{2,}', " ", cleaned_string)

        # Check whether the length of the sentences are more than
max+50
        # If it is max, cut 2 sentences from the center (seperator)
        if len(cleaned_string.split()) > config.maxSeqLength + 50:
            new_line = " "
            cleaned_string = cleaned_string.split(seperator)
```

```
            for number, line in enumerate(cleaned_string):
                line_ = line.split(" ")[:cut_sentence_until]
                for word in line_:
                    new_line += word
                    new_line += " "
                if number == 0:
                    new_line += seperator
            cleaned_string = new_line
        return cleaned_string

    def __get_words_list(self) -> list:
        """Loads the glove model"""

        wordsList = np.load('data/wordsList.npy')
        wordsList = wordsList.tolist()
        return wordsList

    def __get_files_list(self, path: str,  endswith: str) -> list:
        """Finds files with .polarity extension in the desired
path"""

        list_of_files = [path + f for f
                         in listdir(path)
                         if isfile(join(path, f)) and
                         f.endswith(endswith)]
        return list_of_files

    def __calculate_lines(self) -> int:
        # Get the list of all files in folder
        self.filesList = self.__get_files_list(
            self.__dataset_path, ".polarity")

        for file in self.filesList:
            with open(file, 'r', encoding="utf-8", errors="ignore")
as f:
                lines = f.readlines()
            if "data/agreed.polarity" == file:
                agr_lines = len(lines)
            else:
                dis_lines = len(lines)
        self.__overall_line_number = agr_lines + dis_lines
        return agr_lines, dis_lines

    def __check_idx_matrix_occurance(self):
        """Checks if any idx matrix exists"""
        rnn = RNNModel()
```

```
        rnn.set_agr_lines, rnn.set_dis_lines =
self.__calculate_lines()
        idsMatrix = self.__get_files_list(self.__dataset_path,
"idsMatrix.npy")
        if len(idsMatrix) >= 1:
            ans = input(
                "Found 'idsMatrix'. Would you like to recreate it?
(y/n) ")
            if ans in ["y", "", "Yes", "Y"]:
                self.__create_idx()
            else:
                print("Continue...")
        else:
            print("Haven't found the idx matrix models.")
            self.__create_idx()
        rnn.create_and_train_model()

    def __create_idx(self):
        """Function of idx creation"""
        wordsList = self.__get_words_list()
        ids = np.zeros((self.__overall_line_number + 1,
self.__maxSeqLength),
                       dtype='int32')
        for file in sorted(self.filesList):
            f = open(f"{file}", "r", encoding="utf-8",
errors="ignore")
            print(f"\nStarted reading file - {file}....")
            lines = f.readlines()
            for num, line in enumerate(lines, 1):
                if num % 100 == 0:
                    current_line = num + self.__current_state
                    print(
                        f"Reading line number: \

{current_line}/{self.__overall_line_number}")
                cleaned_line = self.clean_string(line)
                splitted_line = cleaned_line.split()
                for w_num, word in enumerate(splitted_line):
                    try:
                        get_word_index = wordsList.index(word)
                        ids[self.__current_state + num][w_num] = \
                            get_word_index
                    except ValueError:
                        # repeated_found = re.match(r'(.)\1{2,}',
word)
                        # if repeated_found:
                        #     print(word)
```

```
                        ids[self.__current_state + num][w_num] =
399999
                    if w_num >= self.__maxSeqLength - 1:
                        break
                f.close()
            # To continue from "checkpoint"
            self.__current_state += len(lines)
        np.save('data/idsMatrix', ids)
        print("Saved ids matrix to the 'model/idsMatrix';")


class RNNModel():
    """Class of TF models creation"""
    os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'  # Avoid the tf
warnings

    def __init__(self):
        self.__batchSize = config.batchSize
        self.__lstmUnits = config.lstmUnits
        self.__numClasses = config.numClasses
        self.__numDimensions = config.numDimensions
        self.__maxSeqLength = config.maxSeqLength
        self.__wordVectors = np.load('data/wordVectors.npy')
        self.__agr_lines = int
        self.__dis_lines = int
        self.learning_rate = config.learning_rate

    @property
    def get_agr_lines(self):
        return self.__agr_lines

    @property
    def get_dis_lines(self):
        return self.__dis_lines

    @get_agr_lines.setter
    def set_agr_lines(self, value):
        self.__agr_lines = value

    @get_dis_lines.setter
    def set_dis_lines(self, value):
        self.__dis_lines = value

    def __get_train_batch(self):
        """Returning training batch function"""
        labels = []
        arr = np.zeros([self.__batchSize, self.__maxSeqLength])
```

```
        for i in range(self.__batchSize):
            if i % 2 == 0:
                num = randint(
                    1, int(self.__agr_lines - (self.__agr_lines *
0.1)))
                labels.append([1, 0])  # Agreed
            else:
                from_line = int(self.__agr_lines +
                                (self.__dis_lines * 0.1)) + 1
                to_line = int(self.__agr_lines + self.__dis_lines)
                num = randint(from_line, to_line)
                labels.append([0, 1])  # Disagreed
            arr[i] = self.ids[num]
        return arr, labels

    def __get_test_batch(self):
        """Returning training batch function"""
        labels = []
        f = open("data/agreed.polarity", errors="ignore",
encoding="utf-8")
        agr_lines = len(f.readlines())
        f = open("data/disagreed.polarity", errors="ignore",
encoding="utf-8")
        dis_lines = len(f.readlines())
        f.close()

        arr = np.zeros([self.__batchSize, self.__maxSeqLength])
        agr_from_line = int(agr_lines - (agr_lines * 0.1)) + 1
        agr_to_line = agr_lines
        dis_from_line = agr_lines + 1
        dis_to_line = int(agr_lines + (dis_lines * 0.1)) + 1

        for i in range(self.__batchSize):
            if i % 2 == 0:
                num = randint(agr_from_line, agr_to_line)
                labels.append([1, 0])  # Agreed
            else:
                num = randint(dis_from_line, dis_to_line)
                labels.append([0, 1])  # Disagreed
            arr[i] = self.ids[num]
        return arr, labels

    def create_and_train_model(self):
        """Creates the TF model"""
        self.ids = np.load('data/idsMatrix.npy')
        print("Creating training model...")
        tf.reset_default_graph()
```

```
        sess = tf.InteractiveSession()
        labels = tf.placeholder(tf.float32,
                                [self.__batchSize,
self.__numClasses])
        tf.add_to_collection("labels", labels)

        input_data = tf.placeholder(tf.int32,
                                    [self.__batchSize,
self.__maxSeqLength])
        # We are saving to the collections, in order to resore it
later
        tf.add_to_collection("input_data", input_data)

        data = tf.Variable(tf.zeros([self.__batchSize,
                                     self.__maxSeqLength,
                                     self.__numDimensions]),
dtype=tf.float32)

        data = tf.nn.embedding_lookup(self.__wordVectors,
input_data)
        cells = []
        for _ in range(config.cells):
            lstm_cell =
tf.contrib.rnn.BasicLSTMCell(self.__lstmUnits)
            lstm_cell = tf.contrib.rnn.DropoutWrapper(
                cell=lstm_cell,
                output_keep_prob=0.75
            )
            cells.append(lstm_cell)
        cell = tf.contrib.rnn.MultiRNNCell(cells)
        initial_state = cell.zero_state(self.__batchSize,
tf.float32)
        value, final_state = tf.nn.dynamic_rnn(cell, data,

initial_state=initial_state,
                                               dtype=tf.float32)

        weight = tf.Variable(tf.truncated_normal(
            [self.__lstmUnits,
             self.__numClasses])
        )
        bias = tf.Variable(tf.constant(0.1,
shape=[self.__numClasses]))
        value = tf.transpose(value, [1, 0, 2])
        last = tf.gather(value, int(value.get_shape()[0]) - 1)
        prediction = (tf.matmul(last, weight) + bias)
```

```
        # Adding prediction to histogram
        tf.summary.histogram('predictions', prediction)

        # Here we are doing the same
        tf.add_to_collection("prediction", prediction)
        tf.add_to_collection("max_seq_length", self.__maxSeqLength)
        tf.add_to_collection("batch_size", self.__batchSize)

        correct_pred = tf.equal(tf.argmax(prediction, 1),
                                tf.argmax(labels, 1))
        accuracy = tf.reduce_mean(tf.cast(correct_pred,
tf.float32))
        tf.add_to_collection("accuracy", accuracy)

        loss =
tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits_v2(
            logits=prediction, labels=labels)
        )
        optimizer = tf.train.AdamOptimizer(
            learning_rate=self.learning_rate).minimize(loss)
        tf.add_to_collection("optimizer", optimizer)
        tf.summary.scalar('Loss', loss)
        tf.summary.scalar('Accuracy', accuracy)
        tf.summary.histogram("Out", value[:, -1])
        merged = tf.summary.merge_all()

        # ------ Below is training process ---------
        folder_name = datetime.datetime.now().strftime("%Y-%m-
%d_%H-%M-%S")
        log_dir = "models/" + str(folder_name) + "/"
        writer = tf.summary.FileWriter(log_dir, sess.graph)
        with open(f"{log_dir}configs.txt", 'w') as f:
            f.write("Number of dimensions:
{}\n".format(config.numDimensions))
            f.write("Sequence length:
{}\n".format(config.maxSeqLength))
            f.write("Batch sizes: {}\n".format(config.batchSize))
            f.write("LSTM units: {}\n".format(config.lstmUnits))
            f.write("Number of classes:
{}\n".format(config.numClasses))
            f.write("Cells: {}\n".format(config.cells))
            f.write("Training steps:
{}\n".format(config.training_steps))

        saver = tf.train.Saver()
        sess.run(tf.global_variables_initializer())
```

```
        for i in range(config.training_steps+1):
            # Next Batch of reviews
            nextBatch, nextBatchLabels = self.__get_train_batch()
            sess.run(optimizer, {input_data: nextBatch,
                                 labels: nextBatchLabels}
                     )
            # Write summary to Tensorboard
            if i % 100 == 0:
                print(f"Iterations: {i}/{config.training_steps}")
                summary = sess.run(merged,
                                   {input_data: nextBatch,
                                    labels: nextBatchLabels}
                                   )
                writer.add_summary(summary, i)
            if i % 200 == 0 and i != 0:
                val_acc = []
                val_state = sess.run(cell.zero_state(
                    self.__batchSize, tf.float32))
                nextBatch, nextBatchLabels =
self.__get_test_batch()
                feed = {input_data: nextBatch,
                        labels: nextBatchLabels,
                        initial_state: val_state}
                summary, batch_acc, val_state = sess.run(
                    [merged, accuracy, final_state],
feed_dict=feed)
                val_acc.append(batch_acc)
                avg_acc = np.mean(val_acc)
                print("\nVal acc: {:.3f}\n".format(avg_acc))
            # Save the network every 10,000 training iterations
            # if (i % 1000 == 0 and i != 0):
            #     save_path = saver.save(sess,
            #
"models/pretrained_lstm.ckpt",
            #                            global_step=i)
            #     print(f"Saved to {save_path}")
        save_path = f"{log_dir}pretrained_lstm.ckpt"
        saver.save(sess, save_path,
global_step=config.training_steps)
        print(f"Model saved to: {save_path}")
        writer.close()
        sess.close()

    def test_model(self, dir_):
        # Starting the session
        self.ids = np.load('data/idsMatrix.npy')
        with tf.Session() as sess:
```

```
            path = ".".join([tf.train.latest_checkpoint(dir_),
"meta"])
            # Get collections
            saver = tf.train.import_meta_graph(path)
            accuracy = tf.get_collection("accuracy")[0]
            input_data = tf.get_collection("input_data")[0]
            labels = tf.get_collection("labels")[0]

            saver.restore(sess, tf.train.latest_checkpoint(dir_))
            print("Testing pre-trained model....")
            test_acc = []
            for i in range(20):
                nextBatch, nextBatchLabels =
self.__get_test_batch()
                cur_acc = sess.run(accuracy,
                                   {input_data: nextBatch,
                                    labels: nextBatchLabels}
                                   )
                test_acc.append(cur_acc)
            print("Test accuracy:
{:.3f}".format(np.mean(test_acc)))


if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument("--train", help="Train the model")
    parser.add_argument("--test", help="Test trained model")
    args = parser.parse_args()
    if args.train:
        train = PrepareData(args.train)
    elif args.test:
        test = RNNModel()
        test.test_model(args.test)
```

Листинг компонента загрузчик данных

```
#!/Library/Frameworks/Python.framework/Versions/3.6/bin/python3.6
# # -*- coding: UTF-8 -*-

import re
import time
import logging
import string

import tweepy
from tweepy import OAuthHandler

import config


class StreamListener(tweepy.StreamListener):
    def __init__(self):
        self.tw = TwitterAccount()
        self.api = self.tw.get_api()

    def on_status(self, status):
        if status.in_reply_to_status_id:
            try:
                self.get_tweet(status, True)
            except AttributeError:
                self.get_tweet(status, False)

    def get_tweet(self, status, is_extended):
        try:
            if is_extended:
                full_text_repl_tw =
status.extended_tweet.get('full_text')
                tweet =
self.api.get_status(id=status.in_reply_to_status_id,
                                                tweet_mode='extended')
                tweet_text = tweet.full_text
            else:
                full_text_repl_tw = status.text
                tweet =
self.api.get_status(id=status.in_reply_to_status_id)
                tweet_text = tweet.text

            if not tweet.in_reply_to_status_id:
                origin_tweet = re.sub("\n", " ", tweet_text)
                reply_tweet = re.sub("\n", " ", full_text_repl_tw)
```

```
                # Write tweets to the tweets file
                text_to_write = f'{origin_tweet} < - >
{reply_tweet}\n'
                self.tw.write_to_file(text=text_to_write,
                                      file='data/all_tweets.csv')

                exclude = set(string.punctuation)
                text_without_puncts = ''.join(ch for ch in
reply_tweet
                                              if ch not in
exclude).lower()
                agree = ["cant agree with you more", "cant agree
more",
                         "couldnt agree more", "couldnt agree with
you more"]
                disagree = ["dont agree", "dont agree",
                            "cant agree with", "cant agree"]
                if any(i in text_without_puncts for i in agree):
                    self.tw.write_to_file(text=text_to_write,

file='data/agreed.plarity')
                    print(f"{text_to_write}\n")
                elif any(i in text_without_puncts for i in
disagree):
                    self.tw.write_to_file(text=text_to_write,

file='data/disagreed.plarity')
                    print(f"{text_to_write}\n")
                else:
                    self.tw.write_to_file(text=text_to_write,

file='data/agreed.plarity')
        except tweepy.TweepError as error:
            print(error)

    def on_error(self, status_code):
        print("Got an error: ", status_code)
        if status_code == 420:
            return False


class TwitterAccount():
    def __init__(self, user="wylsacom"):
        self.user = user
        self.num_of_tweets = 10
        self.num_of_repl = 20
```

```
        self.limit = 0

    def get_api(self):
        try:
            self.consumer_key = config.consumer_key
            self.consumer_secret = config.consumer_secret
            self.access_token = config.access_token
            self.access_secret = config.access_secret

            self.auth = OAuthHandler(self.consumer_key,
self.consumer_secret)
            self.auth.set_access_token(self.access_token,
self.access_secret)
            self.api = tweepy.API(self.auth,
wait_on_rate_limit_notify=True)
        except tweepy.TweepError as exception:
            logging.exception(exception)
        return self.api

    # TODO: search some users tweets and get their ids
    # Using that ids search for tweets using 'api.get_status(id=)
method'
    def get_tweets(self):
        user_tweets = self.api.user_timeline(id=self.user,

count=self.num_of_tweets,
                                             pages=10)
        self.limit += len(user_tweets)

        for num, tweet in enumerate(user_tweets):
            tweet.text = re.sub("\n", " ", tweet.text)
            print(num, tweet.text)
            self.write_to_file(f"Tweet number {num}: \n")
            if self.limit >= 320:
                print("No more tweets")
                print("Sleep for 15min....")
                self.limit = 0
                time.sleep(900)
                continue
            else:
                try:
                    # replies = self.api.search(q=f"@{self.user}",
                    #                           since_id=tweet.id)
                    status = tweepy.Cursor(self.api.search,
                                           q=f"@{self.user}",
                                           since_id=tweet.id,
                                           rpp=self.num_of_repl,
```

```
                                            pages=1
                                            ).items()
                    self.limit += self.num_of_repl
                    for reply in status:
                        if reply.in_reply_to_status_id == tweet.id:
                            reply_user =
str(reply.user.screen_name).strip()
                            self.write_to_file(
                                f'"{self.user} tweeted:
{tweet.text}";\n')
                            self.write_to_file(
                                f'"{reply_user} replied:
{reply.text}";\n')
                            self.write_to_file("-------------------
----\n")
                            print("----------Нашел ответ!----------
\n")
                            print("Ищу ответ на следующий твит\n")
                            break

                except tweepy.TweepError as error:
                    logging.exception(error)
                    time.sleep(900)
                    continue

    def write_to_file(self, text, file="data.csv"):
        with open(file, 'a') as file:
            file.write(text)


if __name__ == '__main__':
    # parser = argparse.ArgumentParser()
    # parser.add_argument("--username",
    # help="Choose the user whose tweets to download")
    # args = parser.parse_args()
    # if args.username:
    #     print(f"Searching {args.username} tweets")
        # tweeter = TwitterAccount(user=args.username)
        # twitter.get_api()
        # twitter.get_tweets()
    # else:
    #     TwitterAccount()

    twitter = TwitterAccount()
    api = twitter.get_api()
    streamListener = StreamListener()
    stream = tweepy.Stream(auth=api.auth,
```

```
                        listener=streamListener,
                        tweet_mode='extended')

stream.filter(track=["disagree", "#disagree", "don't agree",
                     "can't agree with",
                     "cant agree with", "couldn't agree with",
                     "couldnt agree with"],
              languages=["en"],
              async=True)
```