

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Южно-Уральский государственный университет»  
(национальный исследовательский университет)  
Высшая школа экономики и управления  
Кафедра «Информационные технологии в экономике»

РАБОТА ПРОВЕРЕНА

Рецензент, директор ООО «Технопарк ИТ»

\_\_\_\_\_ / Д.В. Козленков /

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой, д.т.н., с.н.с.

\_\_\_\_\_ / Б.М. Суховилов /

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Технологии искусственного интеллекта в анализе  
эмоционального компонента речи

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ-09.04.03.2021.406.ВКР

Консультант, исполнительный  
директор ООО «Сол АйТи»

\_\_\_\_\_ / О.А. Агеев /

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Руководитель,  
Заведующий кафедрой, д.т.н., с.н.с.

\_\_\_\_\_ / Б.М. Суховилов /

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Автор  
студент группы ЭУ-221

\_\_\_\_\_ / Ю.А. Лapidус /

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Нормоконтролёр  
ст. преподаватель

\_\_\_\_\_ / Е.Н. Горных /

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Челябинск 2021

## АННОТАЦИЯ

Лapidус Ю.А. Технологии искусственного интеллекта в анализе эмоционального компонента речи. – Челябинск: ЮУрГУ, ЭиУ; 2021, 91 с. 29 ил., библиогр. список – 67 наименований

Автоматическая обработка естественного языка, sentiment-анализ, анализ тональности текста, понимание естественного языка, формальные модели эмоций, рекуррентные нейронные сети.

Natural Language Processing, Emotional Artificial Intelligence, Sentiment Analysis, General Language Understanding, Formal Emotion Classification, Recurrent Neural Networks.

Объектом исследования являются коллекции машиночитаемых текстовых документов на естественном языке. Предметом исследования являются методы sentiment-анализа.

Целью научно-исследовательской работы является изучение теоретических основ и разработка практических методов sentiment-анализа. В процессе работы были изучены классические и современные методы sentiment-анализа, приведено сравнение эффективности методов и разработан метод sentiment-анализа близкий по эффективности к лучшим практикам индустрии.

Основные результаты научно-исследовательской работы были представлены на 7 Выставке научно-технических и творческих работ студентов ЮУрГУ - 2020, где работа была удостоена диплома I степени.

По теме работы опубликована статья в сборнике "Молодой исследователь".

# ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
1 АФФЕКТИВНЫЕ ВЫЧИСЛЕНИЯ.....	11
1.1 Эмоциональная речь.....	12
1.2 Алгоритмы обнаружения эмоций в речи.....	13
1.3 Базы данных.....	13
1.4 Речевые дескрипторы.....	15
1.5 Формальное моделирование эмоций.....	15
1.6 Существующие формальные модели эмоций.....	19
1.7 Перспективы использования формальных моделей эмоций для анализа эмоционального компонента речи.....	32
2 СЕНТИМЕНТ-АНАЛИЗ. ОБЗОР КЛАССИЧЕСКИХ МЕТОДИК СЕНТИМЕНТ-АНАЛИЗА.....	34
2.1 Метод поиска по ключевым словам.....	35
2.2 Метод с использованием графов знаний.....	36
2.3 Метод, использующий полный лингвистический анализ на разных языковых уровнях....	37
2.4 Метод, основанный на свёрточных нейронных сетях.....	38
2.5 Метод аспектно-эмоционального анализа текста.....	39
2.6 Метод sentiment-анализа с использованием BERT.....	40
2.7 Сравнение подходов sentiment-анализа.....	41
3 РАБОТА С НЕЙРОННЫМИ СЕТЯМИ.....	43
3.1 Глубинные нейронные сети.....	43
3.2 Обучение глубинных нейронных сетей.....	44
3.3 Оценка качества обучения нейронных сетей.....	45

3.4 Библиотеки для глубоких нейронных сетей. Выбор языка программирования и библиотек.....	47
<b>4 СЕНТИМЕНТ-АНАЛИЗ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ.....</b>	<b>49</b>
4.1 Обработка естественного языка с помощью нейронных сетей.....	49
4.2 Представление текста для анализа нейросетью.....	49
4.3 Подготовка среды разработки.....	52
4.4 Данные для анализа.....	53
4.5 Разработка нейронной сети на базе библиотек Keras и Tensorflow, представление текста вектором One Hot Encoding.....	56
4.6 Разработка нейронной сети для sentiment-анализа с использованием плотного вектора.....	60
4.7 Недостатки использования полносвязных нейронных сетей для sentiment-анализа.....	64
4.8 Разработка рекуррентной нейронной сети для sentiment-анализа.....	65
<b>ЗАКЛЮЧЕНИЕ.....</b>	<b>83</b>
<b>БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....</b>	<b>85</b>

## ВВЕДЕНИЕ

Эмоции являются важным компонентом как письменного, так и устного дискурса, всегда сопровождая человеческую речь. Для автоматического выявления эмоций в речи используются методики sentiment-анализа [1].

Существующая на протяжении уже нескольких десятилетий, технология sentiment-анализа становится всё более актуальной с увеличением количества текстового контента (в том числе, генерируемого пользователями), доступного в сети Интернет.

Хотя работы по этому направлению ведутся достаточно давно, относительно приемлемые результаты эффективности sentiment-анализа были достигнуты лишь в последние годы. Прорыв был связан с использованием технологий глубокого обучения и комбинированных подходов, а также с увеличением количества доступных для анализа и обучения таких систем данных.

Объект исследования - базы текстовых документов на естественном языке.

Предмет исследования - методы sentiment-анализа.

Цель выпускной квалификационной работы - изучение теоретических основ sentiment-анализа и создание системы для практического применения sentiment-анализа.

Задачами работы являются:

- знакомство с классическими и современными исследованиями в области sentiment-анализа;
- изучение теории аффективных вычислений;
- сопоставление и сравнение эффективности методов sentiment-анализа;

- изучение архитектур современных нейронных сетей используемых для обработки текста;
- выбор инструментария для анализа эмоционального компонента речи;
- разработка и оценка метода анализа эмоционального компонента речи;
- выводы и перспективы использования метода анализа эмоционального компонента речи.

Данное исследование находится на стыке информатики, лингвистики и психологии.

Научная новизна работы заключается в построении и настройке рекуррентной нейронной сети, показавшей результаты на уровне лучших практик индустрии. Также была предложена собственная формальная модель эмоций, рассмотрены перспективы применения существующих формальных моделей эмоций в сентимент-анализе. Кроме того, архитектуры формальных моделей эмоций впервые были описаны на русском языке.

Взаимосвязи с другими научными направлениями показаны на рисунке 1.



Рисунок 1 – Место сентимент-анализа в научной парадигме

Практическая значимость работы состоит в том, созданная нами нейронная сеть может использоваться для анализа других датасетов, оценке контента сайтов и любых других текстовых материалов; автоматический анализ эмоционального компонента речи может применяться при маркетинговых исследованиях восприятия компании или бренда, классификации и ранжировании отзывов, комментариев, контента социальных сетей, мониторинге общественного мнения, разработке голосовых помощников и др.

## 1 АФФЕКТИВНЫЕ ВЫЧИСЛЕНИЯ

Под аффективными вычислениями (англ. "Affective Computing") понимается научная область, включающая теорию анализа, выделения, интерпретации, моделирования эмоций и создание практических методов для их обработки. Аффективные вычисления находятся на стыке информатики и психологии [63] и являются довольно молодой научной областью, основные идеи которой возникли из статьи Розалинд Пикард 1995 года [49] и ее книги «Affective Computing» [50]. Однако, существуют и более ранние исследования эмоций, которые велись в рамках философии и психологии [21].

Основная задача, которая стоит перед исследователями в области аффективных вычислений — сделать человеко-машинную коммуникацию более естественной и комфортной для человека. Для этого нам необходимо понимать, что наши чувства воспринимаются верно, ощущать эмпатию.

Существует два метода классификации эмоций: непрерывный или категориальный.

В рамках категориального подхода используются отдельные классы состояний, такие как счастливое, грустное, сердитое, испуганное, удивлённое, раздражённое эмоциональное состояние. Непрерывный подход работает с понятиями негативное, позитивное, спокойное, возбужденное эмоциональное состояние. Для того, чтобы машины распознавали эмоции в рамках данных подходов, используются различные алгоритмы и нейронные сети. Возможно и сочетание методов для классификации [14].



## **1.1 Эмоциональная речь**

Чувства, которые испытывает человек, влияют на его речь, и с помощью анализа речи возможно распознать эмоции. Например, если человек находится в возбужденном эмоциональном состоянии (сердится, радуется, боится), он говорит более быстро и громко, высота тона при этом может значительно изменяться. В то же время, речь усталого или скучающего человека становится медленнее, однообразнее по тону, ниже и менее разборчивой [18]. Одни эмоции проще обнаружить в речи чем другие, так как они проявляются наиболее ярко, это гнев [24] и одобрение [54].

Параметры речи и просодические характеристики (интонация, высота тона, скорость речи), возможно визуализировать и обработать с помощью методов распознавания образов [24], [39].

Средняя точность сентимент-анализа составляет 70-80% [46].

Возможно, эта точность не покажется очень высокой, однако стоит вспомнить, что и сами люди также не могут в совершенстве распознавать чужие эмоции - средняя человеческая точность примерно 60% [24]. Точность анализа эмоций можно повысить, если брать во внимание не только речь, но и выражение лица, позу или различные физиологические характеристики показывают более высокую точность [32].

## **1.2 Алгоритмы обнаружения эмоций в речи**

Процесс обнаружения эмоций в устной речи или тексте требует создания надежной и достаточно обширной базы данных, базы знаний или модели векторного пространства [48], кроме того необходимо выбрать как именно классифицировать эмоции [33].

В настоящее время наиболее часто используемыми классификаторами являются линейные дискриминантные классификаторы (LDC), принцип k-ближайший сосед (k-NN), ассоциативная гауссова модель смещения (GMM), метод опорных векторов (SVM), искусственные нейронные сети (ANN), алгоритмы дерева решений и скрытые марковские модели (HMM). Выбор подходящего классификатора — очень важная часть исследовательской работы, поскольку от того, насколько верно выбрана модель, зависит общая производительность системы [32].

### **1.3 Базы данных**

Большая часть существующих систем обнаружения эмоций используют для обучения базы данных. Создание базы данных с нуля является очень трудоёмким процессом [28] и исследователи, как правило, используют готовые наборы данных для обучения классификатора. Для обучения систем обнаружения эмоций в устной речи часто используются наборы данных, полученные с помощью парсинга художественных фильмов. В таких наборах данных, как правило, эмоции классифицируются категориально, согласно теории основных эмоций Пола Экмана. Обычно выделяются гнев, страх, отвращение, удивление, радость, печаль, а остальные эмоции определяются как смесь этих основных эмоций [25]. Тем не менее, упомянутые базы данных речевых фрагментов, полученные из художественных фильмов, широко используются, так как они предлагают исследователям высокое качество звука и достаточное количество примеров, что способствует высокой точности в распознавании эмоций [57].

Однако для реальных приложений предпочтительны фрагменты естественной речи, записанной в обычных условиях, однако создание обширной

базы из таких фрагментов представляется сложной и трудоёмкой задачей. Кроме того, у образцов естественной речи есть и свои недостатки: меньшая эмоциональная окрашенность, посторонние шумы, плохое качество записи.

Существует корпус для анализа устной речи созданный компанией Sony на основе записей робота Sony Aibo, взаимодействовавшего с детьми 10-13 лет [60].

Создание единой базы данных для всех работ в области аффективных вычислений могло бы предоставить исследователям большие возможности для работы и сопоставления методов [42].

#### **1.4 Речевые дескрипторы**

Использование большого количества классов в анализе эмоций представляется перспективным направлением для повышения точности результатов, однако сложность распознавания и производительность систем также будет возрастать. Поэтому, как правило исследователи выбирают лишь самые важные для них классы/категории эмоций, удаляя избыточные или нежелательные. Часто используется лишь два класса (позитивные/негативные эмоции), но в некоторых исследованиях упоминается использование более 200 различных функций [57, стр. 241].

#### **1.5 Формальное моделирование эмоций**

В последние годы наблюдается рост сотрудничества между психологией и информатикой в области компьютерного моделирования эмоций. Однако, чтобы реализовать свой потенциал, обмен между двумя дисциплинами, а также междисциплинарная координация требует дальнейшего улучшения [49].

Формальные модели эмоций (классификация эмоций) - способы, с помощью которых можно отличить одну эмоцию от другой. В настоящий

момент это один из наиболее обсуждаемых вопросов в программировании эмоций.

Компьютерное моделирование эмоций - междисциплинарная задача между, в частности, психологией и информатикой [50]. Цели компьютерного моделирования эмоций в значительной степени соответствуют общим целям искусственного интеллекта [11]. Первая цель - достичь лучшего теоретического понимания эмоций в естественных (то есть людях) и искусственных агентах с помощью создания их вычислительных моделей. Вторая цель - обогатить архитектуру искусственных агентов с помощью механизмов эмоций, подобных человеческим, и, таким образом, наделить их способностью «иметь» эмоции.

Проекты, преследующие первую цель, можно назвать теоретическими и они чаще встречаются в исследованиях по психологии. Ученые в области прикладного компьютерного моделирования эмоций чаще преследуют вторую цель, хотя некоторые исследователи в обеих дисциплинах преследуют обе цели. На самом деле, эти две цели тесно связаны: искусственные агенты с по-настоящему человеческими механизмами эмоций предполагают достаточно точные вычислительные модели этих механизмов [6]. В то же время, один из лучших способов достичь глубокого теоретического понимания психических процессов, в том числе эмоций — это попытаться синтезировать их в искусственных агентах.

Основываясь на функциональном подходе к ментальным состояниям [19], мы предлагаем понимать эмоциональное состояние как способность искусственных агентов иметь внутренние состояния, которые функционально эквивалентны или, по крайней мере, похожи на эмоции у людей, т. е. которые

играют причинные роли в архитектуре агента, имитирующие те, которые играют эмоции у людей. Свойства эмоций, которые могут быть смоделированы в искусственных агентах включают самосознание эмоций, когда они наступают, однако не утверждается, что искусственные агенты, по крайней мере, те, которые существуют в настоящее время, осознают свои эмоции как чувство качественного феноменального опыта [12].

Дополнительная цель аффективных вычислений - создание искусственных агентов с модулем теории разума, который позволяет им распознавать, объяснять, предсказывать и придумывать способы влияния на эмоции других агентов. Вычислительные модели эмоций могут быть полезны и с этой целью: декларативные версии этих моделей могут использоваться в качестве компонентов модуля теории разума искусственных агентов; альтернативно или дополнительно агенты могут быть запрограммированы использовать свои механизмы эмоций в «режиме симуляции», чтобы предсказать эмоции других [51].

Тесная связь теоретического и прикладного компьютерного моделирования эмоций - одна из причин, почему в последние годы наблюдается рост сотрудничества между психологией и информатикой в исследованиях эмоций [45].

Эмоции считаются необходимыми в первую очередь для автономных агентов с ограниченными ресурсами, действующих в динамических и лишь частично известных средах. Вычислительное моделирование эмоций требует возвращения к первоначальной цели искусственного интеллекта для создания «полных» искусственных агентов; а также переориентация на первоначальную

цель когнитивной науки, чтобы создать единую теорию разума. Сотрудничество психологии и компьютерной науки в области эмоций, таким образом, может способствовать объединению не только психологии эмоций, но и психологии в целом.

Для использования в прикладных исследованиях в области программировании эмоций, желательно чтобы психология могла предоставить нам некую окончательную правильную теорию эмоций (UCTE) или окончательную унифицированную теорию разума (UUTM). Более того, эта теория в идеале могла бы быть сформулирована как вычислительная модель или, по крайней мере, была бы доступна в формате, который легко поддается реализации в виде компьютерной программы [53]. К сожалению, психология еще не дошла до этой стадии. Во-первых, не существует единой теории эмоций; существует множество теорий: Стронгмен [62] перечисляет не менее 150 психологических и философских теорий эмоций, которые предлагались на протяжении истории, и это не исчерпывающий список. Безусловно, некоторые из этих теорий представляют собой не более чем расплывчатые гипотезы, в них много избыточности, и многие теории представляют только исторический интерес. Однако даже если ограничить внимание обсуждаемыми в настоящее время теориями эмоций, останется всё же немало. В настоящее время выпускается международный журнал *International Journal of Synthetic Emotions*, в каждом выпуске которого обсуждаются подобные теории [17],[52].

Следует заметить, что точные отношения между различными теориями эмоций иногда трудно оценить даже экспертам.

Теории эмоций в психологии не формулируются как вычислительные модели и большинство из них не описаны достаточно точно и полно, чтобы быть непосредственно реализованы в виде компьютерных программ. Однако в последнее время взаимодействие между психологией и информатикой в области эмоций создало потребность дать краткое и точное изложение психологической теории для неспециалистов. Эта потребность могла побудить психологов попытаться структурировать и уточнить теоретические основы при поддержке специалистов в области информатики и логики[22].

Одна из возможностей систематизировать и унифицировать существующие теории эмоций — использование формального языка. Однако этот подход, хоть и уместен для избранных теорий, имеющих особое значение, вероятно, слишком громоздок как общая стратегия. Альтернативной, более быстрой и жизнеспособной стратегией может быть: 1) разбить существующие теории эмоций на их компонентные предположения и 2) переформулировать эти предположения в общих концептуальных рамках [56]. Эта общая структура могла бы быть формальным языком, возможно, обогащенным набором специальных понятий (например, вера и желание ) или универсальной когнитивной архитектурой; тем не менее, даже не до конца формальные или неформальные реконструкции теории могут существенно способствовать уточнению и унификации, если концепции используются четко и последовательно [41]. Конечной целью этой второй стратегии было бы сокращение многих существующих теорий эмоций к меньшему набору основных предположений. Кроме того, аналогичные предположения разных теорий эмоций легче сравнивать, расплывчатые предположения легче прояснить, а противоречивые

предположения легче идентифицировать, когда они имеют общую концептуальную основу.

## **1.6 Существующие формальные модели эмоций**

Существуют два различных подхода к классификации эмоций:

1. Эмоции — это дискретные и принципиально разные конструкции. Данный подход обычно используется в сентимент-анализе, где эмоции классифицируются как положительные и отрицательные либо положительные, отрицательные и нейтральные.

2. Эмоции имеют непрерывный характер, можно классифицировать их по группам и осям. Данный подход используется при моделировании более сложных систем искусственного интеллекта.

Основными подходами в настоящее время являются OCC, KARO, EMA, CogAff, Affective Computing и модель Фоминых-Леонтьева.

### **1.6.1 Модель OCC**

В своей книге «Когнитивная структура эмоций» [47] Ортони, Клор и Коллинз предложили новаторскую и оригинальную модель эмоций, дающую четкую и убедительную структуру условий, вызывающих эмоции, и переменные, влияющие на их интенсивность. Эта психологическая модель популярна в информатике для систем распознавания эмоций и программирования эмоций искусственных персонажей. Модель OCC описывает иерархию, которая классифицирует 22 типа эмоций. Структурная модель OCC представлена на рисунке 2.



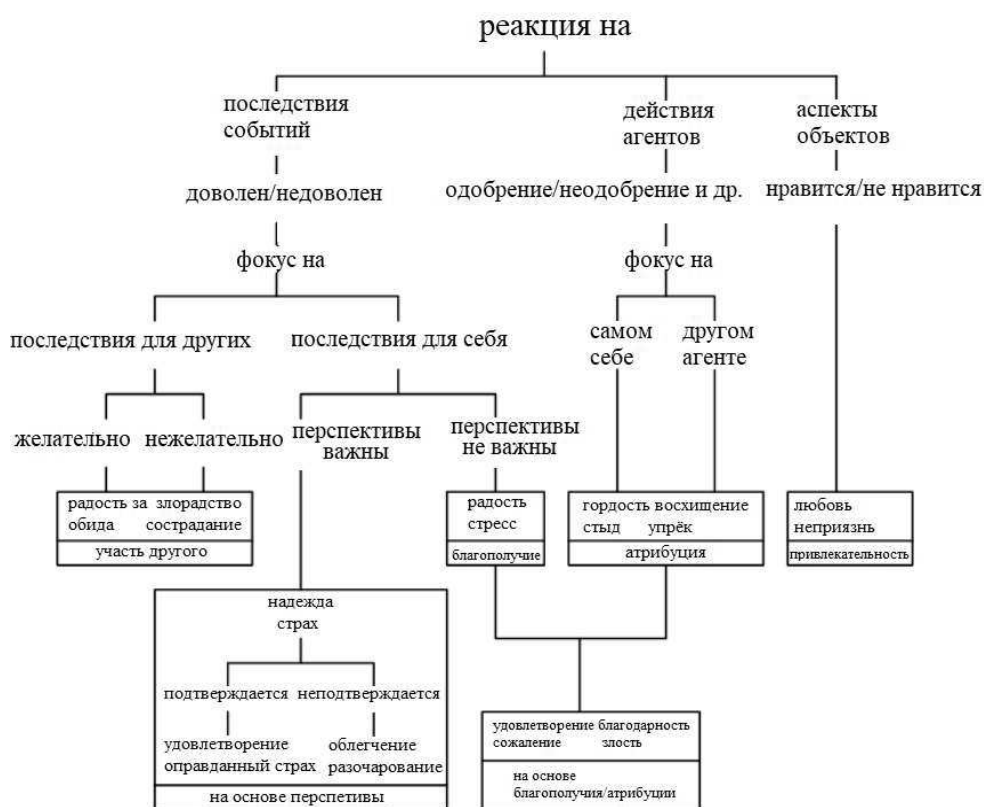


Рисунок 2 – Структурная модель классификации эмоций OCC, перевод наш

Структура состоит из трех ветвей, а именно эмоций, касающихся последствий события (например, радость и жалость), действия агентов (например, гордость и упрек) и аспекты объектов (например, любовь и ненависть). Кроме того, некоторые ветви объединяются, образуя группу сложные эмоции, а именно эмоции, связанные с последствиями событий, вызванных действиями агентов (например, благодарность и злость). Поскольку эти понятия (т.е. события, действия, и объекты) также широко используются в моделях агентов, модель OCC считается подходящей для использования в искусственных агентах [61].

### 1.6.2 Модель KARO

Модель KARO появилась как расширение модели OCC с помощью формальной логики, динамической логики и дополнительных операторов

мотивации [53]. KARO моделирует эмоции на логическом языке с использованием операторов знания, убеждений, действий, способностей и желаний. Система использует четыре свободных (не связанных с внешними объектами) эмоции, ассоциированных с определенными установками:

- счастье, вызванное достижением целей,
- печаль, вызванная невыполнением целей,
- гнев, вызванный разочарованием в активном плане, и
- страх, вызванный конфликтом целей или опасностью.

В системе KARO используется когнитивное понятие эмоции, которое может быть использовано в практической реализации. Однако это только теоретическое описание эмоциональных агентов с акцентом на динамику психических состояний агента и последствия его действий.

Реализация KARO оценивалась путем запуска его на платформе iCAT. ICAT представляет собой робота — кота, способного выражать эмоции с помощью выражения лица.

Роботот iCAT от исследовательской группы компании Philips представлен на рисунке 3.

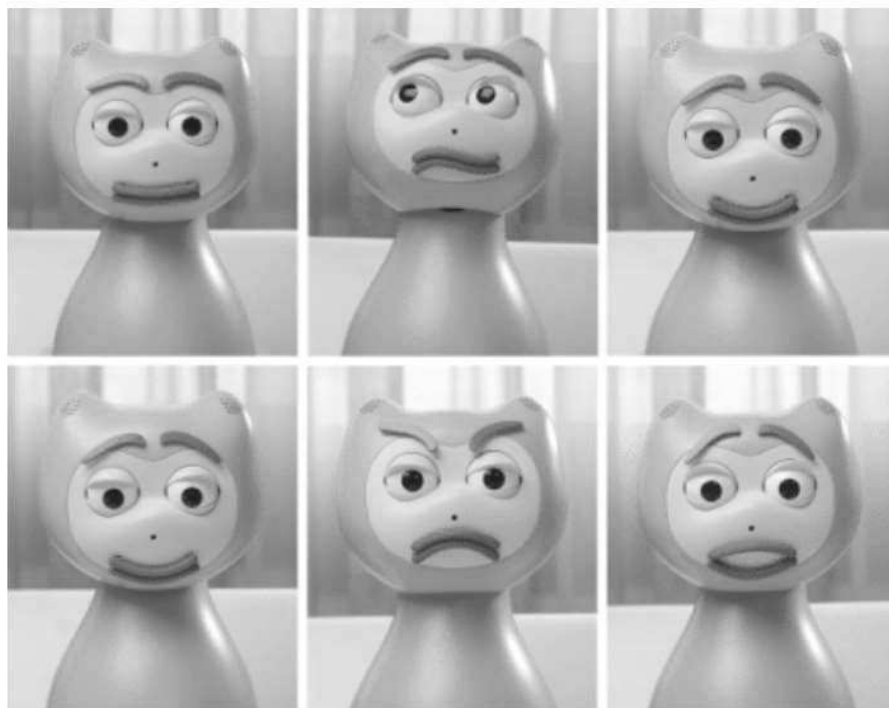


Рисунок 3 – Робокот от исследовательской группы компании Philips

Взаимодействие пользователя с роботом iCAT представлено на рисунке 4.



Рисунок 4 – Взаимодействие пользователя с роботом iCAT

KARO использует определенные когнитивные и мотивационные компоненты в задаче создания эмоций. Система KARO оценивает эмоции (как ярлыки) и изменяет приоритеты агента. Безусловно, система построена без учёта эволюционных, соматических или когнитивных теориях оценки; тем не менее, она реализует некоторые элементы теории OCC.

### 1.6.3 Модель ЕМА

Формальная модель эмоций ЕМА [26] также создана на основе модели ОСС. В рамках данной модели для описания эмоций используются такие переменные как желательность ( $D$ ), вероятность события ( $p$ ) и другие. Эмоции могут определяться с помощью этих переменных, например:

Радость  $D > 0, p = 1$

Надежда  $D > 0, p < 1$

Страх  $D < 0, p < 1$

Дистресс (горе)  $D < 0, p = 1$

Модель ЕМА является динамической моделью, так как эмоциональная реакция рассматривается в совокупности с вызвавшим её стимулом, а также принимается во внимание тот факт, что одни эмоции каузируют другие. В вычислительной модели динамических эмоциональных процессов ЕМА принимаются основные принципы, разделяемые оценочными теориями эмоций: оценка - это процесс интерпретации отношения человека с окружающей средой. Эту интерпретацию можно охарактеризовать с помощью набора оценочных критериев (англ appraisal dimensions) и определенные эмоции связаны с определенными конфигурациями этих критериев [44].

Модель ЕМА представлена на рисунке 5.



Рисунок 5 – Модель ЕМА, приведенная в работе [44] (перевод наш)

#### 1.6.4 Модель CogAff

Модель CogAff, (аббревиатура от Cognition and Affect, «познание и чувства») была разработана в Школе компьютерных наук Бирмингемского университета. Авторы модели пытаются определить понятие разум и понять, как он работает.

В работе приводится схематическое представление того, как устроен человеческий разум — от самых древних базовых структур до сложных новейших [59]. Реактивные механизмы в системах искусственного интеллекта могут строиться похожим образом.

На рисунке 6 представлена модель CogAff, в которой разделены три уровня когнитивных механизмов.

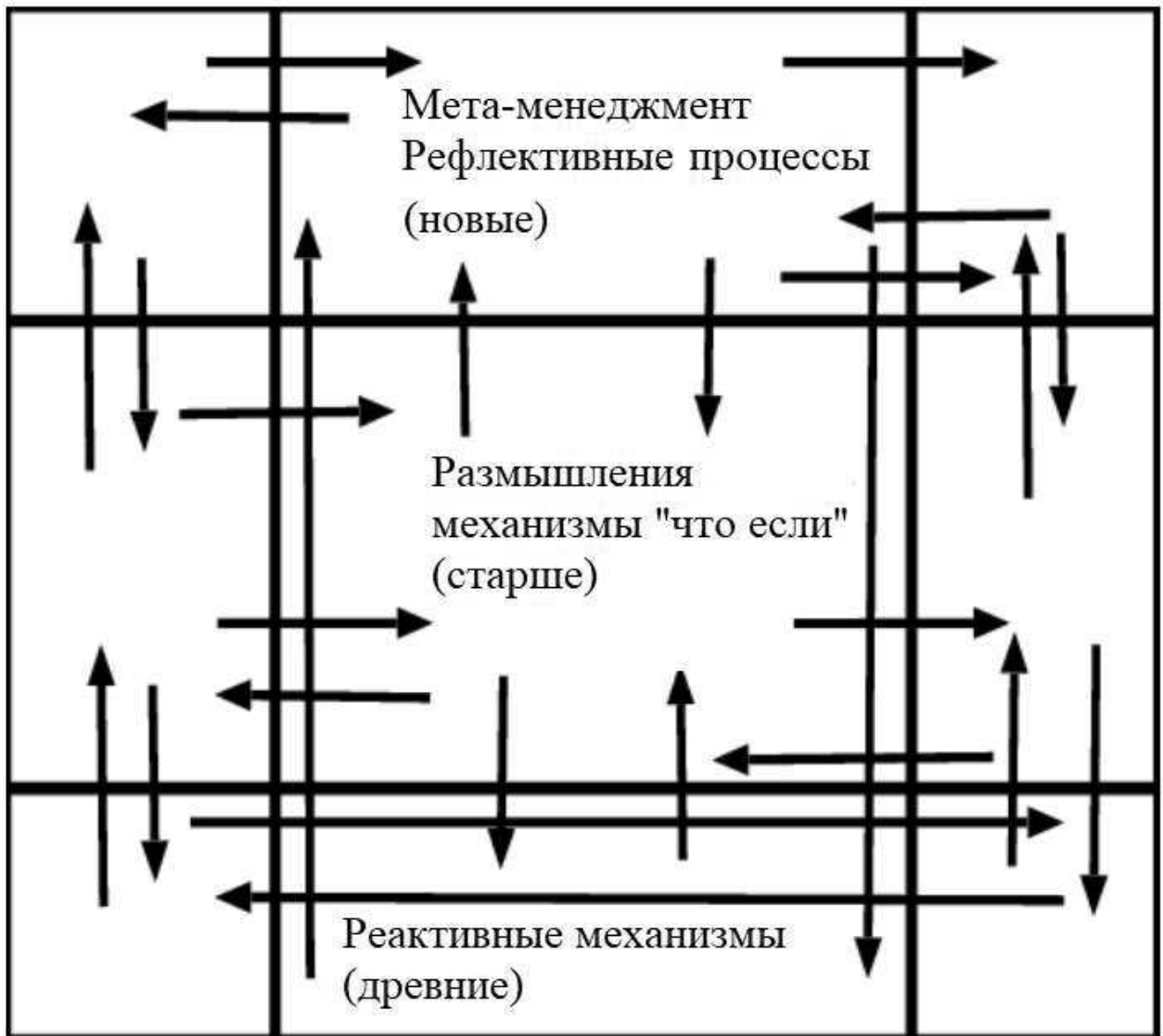


Рисунок 6 – модель CogAff

### 1.6.5 Модель Фоминых-Леонтьева

Модель Фоминых-Леонтьева также довольно близка к ОСС, однако используется другой набор параметров и аргументов для описания эмоций. В рамках данной модели эмоции описываются с помощью функций, заданных набором параметров, определяющих контекст [8]. Набор параметров различается для каждого вида эмоций. Базовые эмоции определяются через изменение

ресурсов  $R$  (приобретение или потерю) и уровень достижений. Если событие уже произошло, величина  $R$  известна и можно определить следующее:

$R > 0$ , то есть ресурс увеличился, возникают положительные эмоции (радость)

$R < 0$ , то есть ресурс уменьшился, возникают отрицательные эмоции (огорчение)

Саму функцию для радости и огорчения можно определить так:

$$E = F(R).$$

Если событие ещё не произошло, его исход неизвестен и неизвестно изменение величины  $R$ . Однако агент может испытывать эмоции по поводу предстоящего события и оценивать вероятность  $P$  наступления его исходов.

$PR > 0$ , то есть ожидание, что исход увеличит ресурсы, возникает эмоция надежды,

$PR < 0$ , то есть ожидание, что исход уменьшит ресурсы, возникает эмоция надестраха.

Саму функцию для надежды и страха можно определить так:

$$E = F(p, PR).$$

### **1.6.6 Модель эмоций FCM**

Салмерон [55] предлагает использовать нечеткие когнитивные карты (FCM) в качестве инструмента для прогнозирования человеческих эмоций, смоделированных на основе концепции Тейера [64], которая учитывает как возбуждение (уровень возбуждения/спокойствия агентов), так и валентность (положительная или отрицательная в целом). Состояния/значения возбуждения и валентности можно интерпретировать как вертикальные и горизонтальные

координаты текущей точки / состояния на круге эмоций, показанном на рисунке 7.

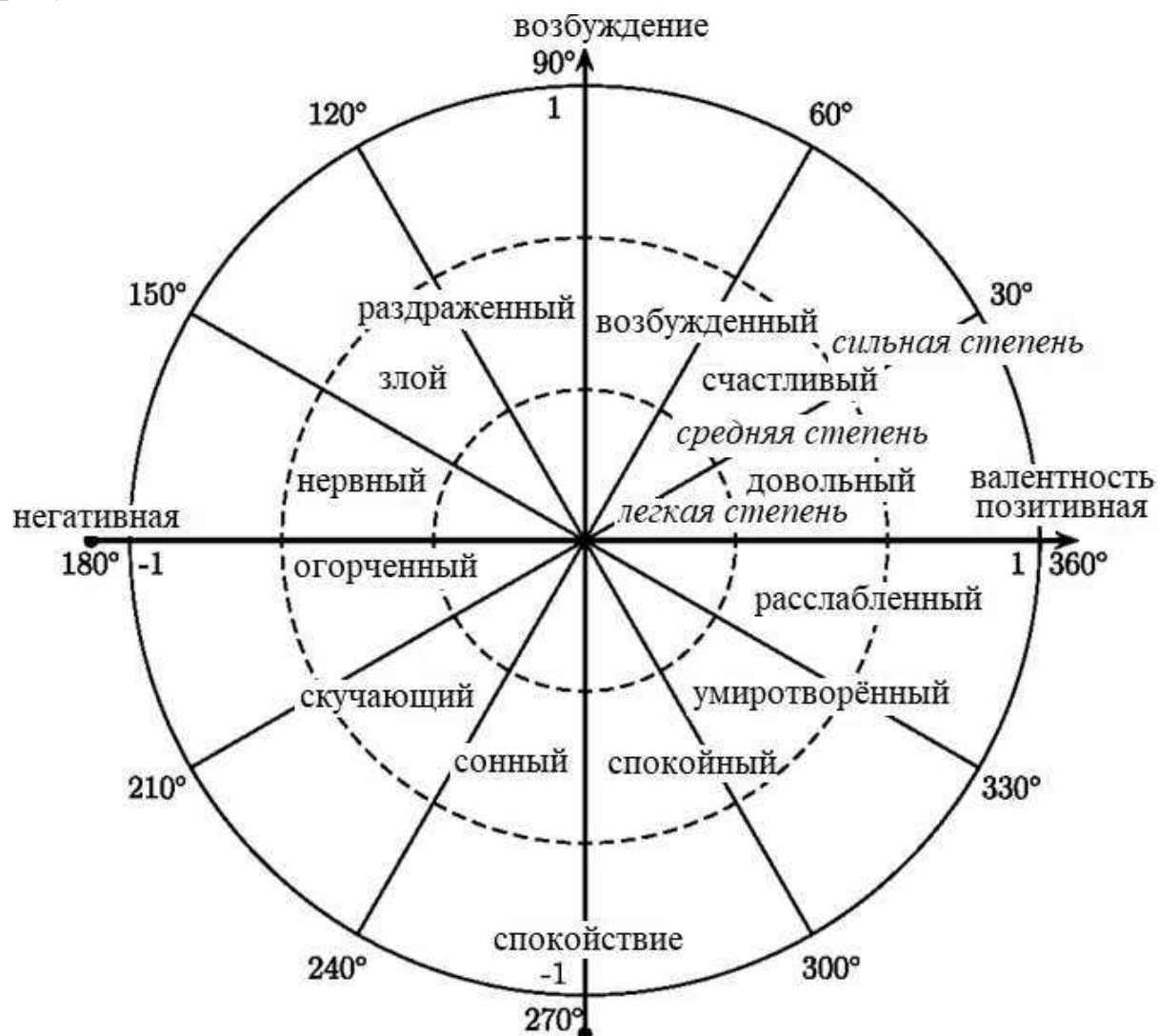


Рисунок 7 – Круг эмоций, возбуждения и валентности (Thayer, 1989; Salmeron, 2012)

Модель построена с использованием информации из анализа настроения и понятий биопсихологии. Эмоции в этой модели могут принимать одно из двенадцати значений: возбужденное, счастливое, довольное, расслабленное, умиротворенное, спокойное, сонное, скучающее, грустное, нервное, злое и раздраженное состояния. Кроме того, Салмерон (2012) оценивает каждую



эмоцию по трехвалентной шкале: легкая, средняя и сильная. В итоге у модели 36 возможных эмоциональных состояний.

Нечеткие когнитивные карты собирают и представляют знания, необходимые для конкретной нечеткой нейронной системы вывода (FNIS) эмоций. Можно описать FNIS как нейронную сеть следующим образом: стимулы входного слоя, скрытый слой с такими элементами как «ожидание» и «тревога», которые, в свою очередь, генерируют системные эмоциональные выходы: валентность и возбуждение. Переходы между слоями сети имеют веса, и система всегда способна вызвать определенную эмоцию.

Описанная система не полностью использует нечеткую модель эмоций, поскольку эмоции представлены только ярлыками, присвоенными конкретным 36 диапазонам эмоциональных переменных (производных от возбуждения и валентности). Эту модель можно отнести к оценочной теории эмоций. Компоненты, участвующие в порождении эмоций, можно разделить на когнитивные (входящие в FCM / FNIS) и мотивационные (рассмотрение ожидания и беспокойства в скрытом слое FNIS как смеси настроения и убеждений).

### **1.6.7 Модель эмоций xEmotion**

Концепция ISD, интеллектуальной системы принятия решений, позволяет агенту конструировать индивидуальные эмоции в подсистеме, называемой xEmotion [35], в ответ на наблюдения и взаимодействия.

Подсистема xEmotion основана на нескольких различных теориях эмоций, включая временное разделение. Прежде всего, она реализует соматическую теорию эмоций [67] как концепцию предэмоций. Предэмоции - это наиболее

грубая форма эмоций, которые связаны с впечатлениями (основными характеристиками, воспринимаемыми агентом), такими как боль и быстрое движение. Также они могут появиться в результате подсознательной реакции. Второй компонент  $xEmotion$  - это субэмоции - эмоции, связанные с определенными объектами или ситуациями, которые имеют важный информативный аспект для агента. Очевидно, они связаны с оценочной теорией эмоций. Однако оценка субэмоций происходит в случае сильного эмоционального состояния (в отличие от классического оценочного подхода). Это состояние запоминается / ассоциируется с определенными воспринимаемыми объектами (в текущей сцене и во время сильных эмоций), и следующее появление этого объекта (в будущем) вызовет определенное субэмоциональное состояние. Субэмоции будут угасать с течением времени (согласно кривой забывания).

Эти два элемента могут изменять фактическое эмоциональное состояние агента, называемое классической эмоцией, на которое также влияют промежуток времени и общая скорость выполнения потребностей агента (помимо эмоциональных компонентов). Очевидно, что эмоции агента часто находятся в нейтральном или квазинейтральном состоянии, из которого они развиваются с течением времени. С другой стороны, на систему эмоций также влияет основная мотивационная система агента, основанного на ISD (Интеллектуальная система принятия решений), которая представляет собой систему потребностей агента. Система потребностей ISD аналогична системе убеждений, желаний, намерений (BDI); однако ориентирована на долгосрочные цели. Настроение (долговременная эмоция) автоматически создается на основе классической эмоции с использованием квазипроизводной механизации.

В подсистеме xEmotion ISD есть 24 возможных эмоции. Однако только субэмоции и классические эмоции могут преобладать над одной из 24 обозначенных / лингвистических эмоций, смоделированных как нечеткие двумерные переменные. Применяемые предэмоции, ввиду их простоты, могут предполагать только одну из восьми первичных эмоций, также смоделированных с использованием нечеткого описания.

Таким образом, эмоции в системе ISD имеют свои определенные цели. Их основная роль - сузить возможности выбора реакций агента (но не решить, какая именно реакция должна быть реализована). Это означает использование эмоций и обобщенной запланированной переменной для управления агентом в соответствии со схемой SVC. Таким образом, текущая рабочая точка всей системы ISD может быть соответствующим образом адаптирована. Это не единственная функция эмоций. Настроение, которое можно считать производным параметром, регулирует значение нечетких коэффициентов подсистемы потребностей, позволяя потребностям агента достигать состояния удовлетворения быстрее или медленнее.

Концепция ISD изначально применялась как «мыслящий» механизм, который управляет простым мобильным роботом с использованием ограниченной иерархии потребностей Маслоу, а также для управления интерактивным аватаром DictoBot и автоматическим автономным драйвером xDriver [36].

Таким образом, xEmotion, как часть концепции принятия решений ISD, представляет собой эмоциональную систему, основанную на оценочном и соматическом подходах к эмоциям. В нем есть некоторые второстепенные элементы, взятые из эволюционной теории эмоций, особенно применение эмоций

в качестве механизма самозащиты. Когнитивный (субэмоции), физиологический (предэмоции) и мотивационный (потребности) компоненты участвуют в создании эмоций. Эта система меняет свою рабочую точку и, следовательно, свое поведение / реакцию, принимая во внимание (внутренние) и классические эмоции, также интерпретируемые с использованием нечеткой логики.

### **1.7 Перспективы использования формальных моделей эмоций для анализа эмоционального компонента речи**

В рамках первой главы были рассмотрены основные существующие формальные модели эмоций. В первую очередь стоит заметить, что психология не готова на данный момент дать единое и чёткое понимание того, что такое эмоция, как можно классифицировать эмоции и это создаёт трудности для формального моделирования эмоциональных процессов.

Специалисты в области искусственного интеллекта используют различные формальные модели эмоций, которые отличаются между собой подходом. Так, в рамках некоторых моделей эмоции понимаются как дискретные и принципиально разные конструкции, классы, в рамках других — как непрерывный континуум.

Примечательно, что в сентимент-анализе на данный момент используется единственный подход — категориальный и обнаруженные в тексте эмоции принято делить на две (положительные, отрицательные) или на три категории (положительные, отрицательные, нейтральные).

Нам представляется перспективным использование в сентимент-анализе второго подхода: классификация эмоциональных состояний, обнаруженных в тексте, как сущностей, имеющих непрерывный характер. Данный подход

позволит точнее проанализировать и обработать всё многообразие человеческих эмоций, не вырождая их в две или три примитивные категории.

Для представления эмоций можно использовать многомерную систему координат, центр которой представляет нейтральное эмоциональное состояние. Осями могут служить спокойствие-возбуждение, позитивная-негативная валентность, как в модели FCM. Полноту картины добавляют и аспекты, упомянутые в модели ОСС и Фоминых-Леонтьева (эмоции связанные с собой или другими) и в модели ЕМА (вероятность события).

Таким образом, мы предлагаем использование следующих осей в системе координат формальной модели эмоций:

спокойствие-возбуждение

позитивная-негативная общая валентность

внутренняя направленность - внешняя направленность (эмоции связанные с собой или другими)

положительная-отрицательная результативность

вероятность события (наступившее, вероятное, невероятное)

непосредственность эмоциональной реакции — осознанность эмоциональной реакции.

Данная формальная модель эмоций способна довольно точно описать большое количество эмоциональных состояний.

Однако необходимо изучить технические возможности современных нам технологий для того, чтобы сделать выводы о возможности практической имплементации данной формальной модели в анализе эмоционального компонента речи.

## 2 СЕНТИМЕНТ-АНАЛИЗ. ОБЗОР КЛАССИЧЕСКИХ МЕТОДИК СЕНТИМЕНТ-АНАЛИЗА

Сентимент-анализ— выявление тональности языковой единицы при помощи методов обработки естественного языка, статистики, машинного обучения [4]. Также в литературе встречается синонимичный термин – анализ тональности текста. В англоязычной литературе используется термин *sentiment analysis* и близкий к нему *opinion mining* [1]. Сентимент-анализ является областью на стыке информатики, прикладной лингвистики и психологии. Задачи сентимент-анализа решаются методами автоматической обработки естественного языка, однако стоит понимать, что наиболее важна в этой области для нас эмоциональная составляющая текста, его тональность, а не его структура и формальные признаки. Задача представляется сложной, поскольку мы, в некотором роде, пытаемся научить компьютер читать "между строк" [34].

Объём данных, пригодных и интересных для сентимент-анализа, очень велик. На тематических сайтах (например, [booking.com](http://booking.com), [imdb.com](http://imdb.com), [amazon.com](http://amazon.com) и т.п.) пользователи оставляют отзывы об отелях, фильмах, ресторанах, различных товарах и услугах, делятся своим опытом и впечатлениями. В социальных сетях незамедлительно появляются комментарии пользователей на любое событие, любую новость. Эмоции являются важной составляющей как отзывов, так и комментариев. При этом тенденция такова, что уровень доверия к такому контенту постепенно растёт, также, как растёт и объём этого контента [7].

Технология сентимент-анализа может быстро и эффективно решить проблему мониторинга общественного мнения по таким важным для владельцев брендов вопросам как: «что думают клиенты о нашем продукте», «каково мнение

об услугах конкурента», «был ли опыт взаимодействия с нашим продуктом положительным или негативным» [8]. При этом благодаря простоте использования, технологию сентимент-анализа можно использовать в фоновом режиме, проводя постоянный мониторинг данных, например, чтобы не упустить момента распространения какого-либо негатива [5].

Автоматическая оценка тональности текста также важна в информационном поиске, ранжировании информации, разработке чат-ботов и голосовых помощников.

## **2.1 Метод поиска по ключевым словам**

Метод поиска по ключевым словам - самый простой способ сентимент-анализа.

В основе этого подхода лежит предположение, что текст, в котором встречается слово, имеющее некоторую эмоциональную окраску, также является эмоционально-окрашенным [21]. Извлечение эмоций является сложной проблемой из-за субъективной природы отношений между словами и эмоциями. Однако в работах по психологии и лингвистике [29], [5] используются термины, к которым будет уместно прибегнуть в данной работе. Итак, существуют слова, которые непосредственно относятся к эмоциональному состоянию (например, "радость", "улыбаться", "страх"), так называемые непосредственно эмоциональные слова (англ. "direct emotion words"), а другие слова - косвенно эмоциональные слова (англ. "indirect emotion words"), например "убийца", "кошмар" косвенно связаны с эмоциональным состоянием, вызывая его через ассоциативные связи [31].

Минусом классического метода поиска по ключевым словам является неверная интерпретация отрицательных предложений: фразы "я счастлив" и "я не счастлив" будут отнесены к одной модальности [37].

Существует также метод поиска по ключевым словам, учитывающий отрицания [16]. В рамках этого метода выделяются слова, употреблённые с частицей "не" и их тональность меняется на противоположную.

Однако игнорирование синтаксической структуры предложения также приводит к нежелательным последствиям: фразы "едва ли он был счастлив" и "он был счастлив" будут классифицированы одинаково.

## **2.2 Метод с использованием графов знаний**

Более совершенный подход был представлен в работе [16]. В рамках представленной работы был использован подход, при котором для классификации эмоций в тексте были использованы базовые знания. Базовые знания были представлены в виде графа, узлами которого являлись объекты реального мира, а гранями - взаимоотношения между ними.

Но поскольку не учитывались синтаксические связи между словами, этот подход также допускал ошибки при анализе сложных синтаксических структур. Так, предложения "он приготовил невкусное блюдо следуя этому рецепту" и "невозможно приготовить невкусное блюдо следуя этому рецепту" классифицировались одинаково. Другая проблема метода состоит в том, что размеченных данных, подходящих для анализа, к сожалению, очень мало.



### **2.3 Метод, использующий полный лингвистический анализ на разных языковых уровнях**

В рамках данного подхода происходит автоматическая обработка текста на морфологическом, синтаксическом и лексическом уровнях.

Например, в системе, разработанной Masum et al. [40] обработка текста происходит в 4 этапа:

1) На этом этапе слова классифицируются по частям речи, а также указывается их валентность (положительная/отрицательная). Для классификации используются WordNet (для выделения глаголов, наречий и прилагательных), ConceptNet (для выделения существительных), OpenMind (для выделения именованных сущностей)

2) Далее происходит синтаксический анализ текста. С помощью синтаксического парсера Machine Syntax выполняется извлечение триплетов (Субъект-Предикат-Объект).

3) Далее, с учётом полученных данных производится sentiment-анализ. На этом этапе вычисляется полная модальность (положительная/отрицательная) предложения. Учитываются веса отдельных слов и триплетов.

4) Выделение эмоций по модели OCC (предложенной Ortony, Core and Collins)

Архитектура системы анализа эмоций, предложенная в работе Masum представлена на рисунке 8.

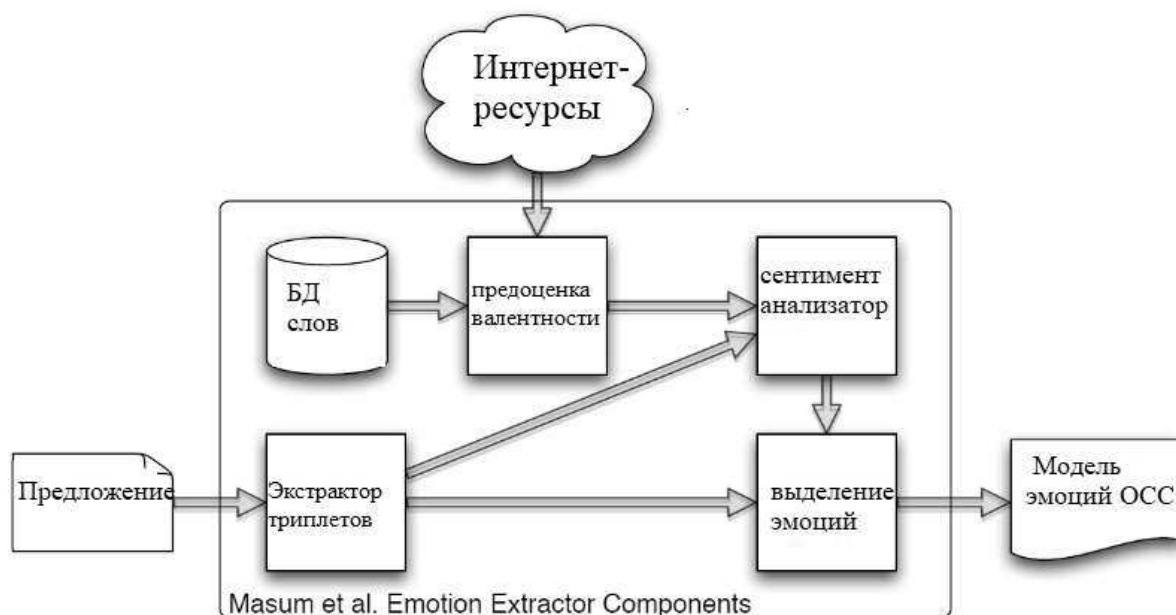


Рисунок 8 – Архитектура системы анализа эмоций разработанной Masum et al (перевод наш)

## 2.4 Метод, основанный на свёрточных нейронных сетях

В исследованиях последних лет для сентимент-анализа часто используются нейронные сети, в частности сети глубокого обучения. Поскольку задача сентимент-анализа технически похожа на задачу классификации (найти и выделить в тексте эмоцию, а затем классифицировать фрагмент текста на основании найденных эмоций), наиболее часто используется свёрточная нейронная сеть.

Рассмотрим в качестве примера работу Генри Мишель Нассифа из Массачусетского Технологического Университета, 2016 г. [38].

Нассиф разработал систему сентимент-анализа в контенте генерируемом пользователем, и взял за основу свёрточную нейронную сеть.

Перед работой с нейросетью производится предварительная обработка текста: каждое слово представляется в виде вектора, который вычисляется

с помощью системы open source Word2Vec [41], используя массив векторов GoogleNews. Стоит заметить, что векторы семантически близких слов также оказываются "вычислительно" рядом.

После вычисления векторов всех слов, предложение свёртывается с использованием свёрточных фильтров различных размеров (униграммы, биграммы, триграммы) и генерируется карта распределения характеристик. Карты распределения характеристик затем подаются на вход свёрточной нейронной сети, как это представлено на рисунке 9.

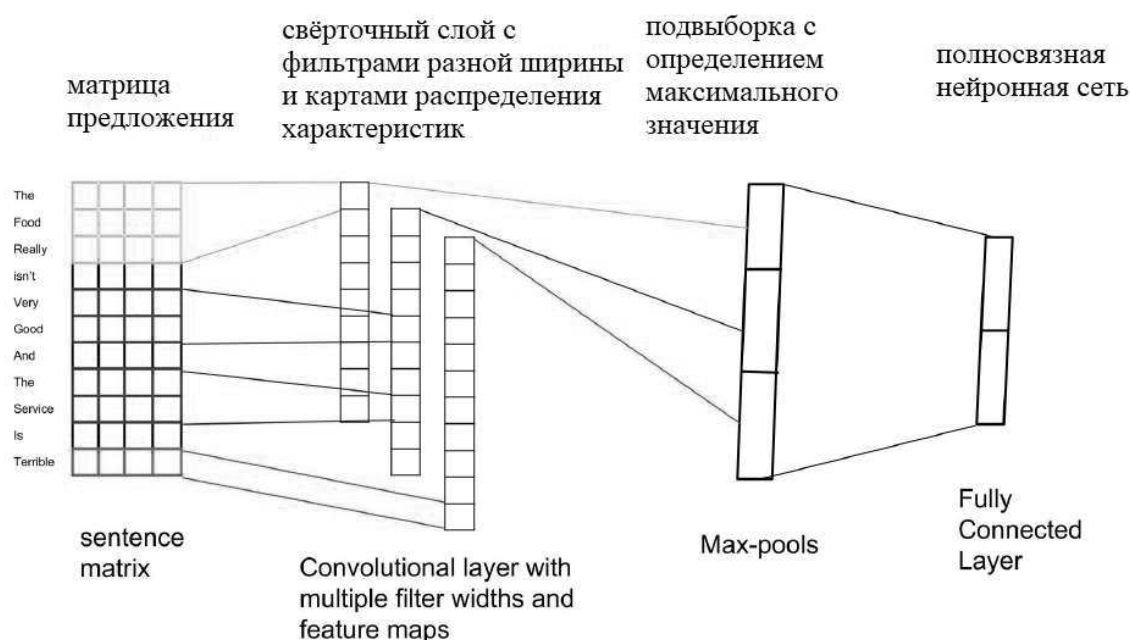


Рисунок 9 – Общая архитектура системы sentiment-анализа с помощью свёрточной нейронной сети (перевод наш)

## 2.5 Метод аспектно-эмоционального анализа текста

Впервые метод аспектно-эмоционального анализа текста был озвучен на мероприятии SemEval в 2014 году [43]. Принципиальная особенность предложенного нового подхода заключалась в том, что вместо общей оценки тональности текста более детальное внимание уделяется конкретным аспектам.

Например, мы хотим выяснить, какие эмоции вызывает у людей какое-либо медицинское учреждение. Возможно, sentiment-анализ показывает преобладание положительной или отрицательной тональности в отзывах. Однако, это малоинформативно. Для руководства учреждения необходимо знать что именно (персонал, услуги, интерьер, расположение, часы работы и др.) вызывает положительные или отрицательные эмоции у пациентов. Для решения этой задачи и был разработан аспектно-эмоциональный анализ текста (aspect-based sentiment analysis). В исследовании [2] описывается разработанная автором система аспектно-эмоционального анализа текстов. В системе используется алгоритм извлечения аспектных терминов. После того, как аспектные термины извлечены, определяется их тональность.

## **2.6 Метод sentiment-анализа с использованием BERT**

В совсем новой работе 2019 года исследователями из Университета Гётеборга, Швеция, предлагается использовать BERT для sentiment-анализа. BERT - предобученная на больших объёмах текстов нейронная сеть от Google, работающая сразу в двух направлениях текста (учитывает и левый и правый контекст) [27].

Хорошие результаты были получены при пост-обучении BERT [10].

В рамках исследования учитывалось 4 модальности (положительная, нейтральная, отрицательная и конфликтная). Конфликтная модальность означает, что часть фрагмента является положительной, часть отрицательной и поэтому однозначное определение модальности невозможно.

Одной из стандартных задач BERT является категоризация. При этом сопоставляется некоторое предложение и аспект. Система вычисляет вероятность,

относится ли аспект к предложению или нет. Результат выдаётся в виде метки аспекта (положительный, отрицательный, нейтральный, конфликтный).

Другой стандартной задачей BERT является сопоставление двух предложений. Система вычисляет вероятность, по которой два предложения могут встречаться вместе в одном контексте. Для целей sentiment-анализа исследователи использовали в качестве первого предложения - реальное предложение, которое необходимо проанализировать, в качестве второго - аспект (положительный, отрицательный, нейтральный).

Наиболее высоких результатов исследователям удалось достичь, комбинируя методы сопоставления и категоризации [30].

## 2.7 Сравнение подходов sentiment-анализа

Проанализируем рассмотренные нами методы sentiment-анализа. В начале таблицы представлены классические методы, в конце современные. Самый современный из рассматриваемых методов создан в 2019 году и имеет эффективность 87%.

Таблица 1 — Сравнение эффективности методов sentiment-анализа

Метод	Особенности метода	Точность метода
Метод поиска по ключевым словам	Текст анализируется на наличие ключевых слов с выраженной эмоциональной окраской	56-68%
Метод с использованием графов знаний	Из текста извлекаются графы – структуры, вершинами которых являются концепты, а гранями – отношения между ними. Требуется предварительная ручная разметка, в автоматическом режиме графы извлекаются плохо	41-71%, в зависимости от размеченного материала

Окончание таблицы 1

Метод	Особенности метода	Точность метода
Метод, использующий полный лингвистический анализ на разных языковых уровнях	Текст проходит несколько уровней анализа: морфологический, синтаксический, семантический. Сентимент-анализ учитывает лингвистическую информацию	50-72%
Метод аспектно-эмоционального анализа текста	В тексте выделяются аспекты – то, относительно чего будет производиться сентимент-анализ	78.7%
Метод, основанный на свёрточной нейросети	Текст, с предварительно размеченными векторами, подаётся на вход свёрточной нейронной сети, которая выполняет сентимент-анализ и присваивает одно из трёх значений (положительный, нейтральный, негативный)	Положительные - 83% Нейтральные - 73% Негативные - 67%
Подход с использованием двунаправленной нейросети BERT	Рекуррентная нейронная сеть	Категоризационная модель - 87.5 %

## 3 РАБОТА С НЕЙРОННЫМИ СЕТЯМИ

### 3.1 Глубинные нейронные сети

Глубинная (глубокая) нейронная сеть — искусственная нейронная сеть, имеющая входной, выходной и несколько промежуточных слоёв [58]. Независимо от типа корреляции (линейной или нелинейной), глубинная нейронная сеть способна превратить входные данные в выходные с помощью корректного метода математических преобразований.

Глубокое обучение — совокупность методов машинного обучения, основанных на обучении представлениям, а не специализированным алгоритмам под конкретные задачи. Глубокие нейронные сети являются одним из самых популярных методов машинного обучения которые используются на практике. Нейронная сеть строится из простых вычислительных элементов — искусственных нейронов. Искусственные нейроны работают примерно также, как нейроны головного мозга, хотя, конечно, в гораздо более упрощенном виде. Этот метод сейчас используется все чаще по сравнению с другими методы машинного обучения потому что в традиционном машинном обучении нам необходимо выбрать из большого объема имеющихся данных те данные, которые позволяют нам решить нашу задачу. Такие данные называются признаками [23]. Выбором важных признаков для различных задач занимается человек и если кому-то удалось найти подходящие признаки, то задача решается, а если признаки найти не удастся то, метод машинного обучения работает плохо. Выигрышная особенность нейронных сетей заключается в том, что они автоматически отделяют нужные данные от ненужных и умеют выбирать правильные признаки. С другой стороны, так как в нейронной сети приходится обрабатывать гораздо

больше данных по строению с другими методами машинного обучения, использование нейронных сетей требует большой вычислительной мощности.

Существует множество применений для нейронных сетей, от азвлекательных (генерировать изображение в стиле различных известных художников, сочинять музыку в стиле известных исполнителей или даже генерировать музыку по изображениям) до практических (машинный устный и письменный перевод, беспилотное управление автомобилями, интерактивные диалоговые системы, робототехника, медицинское диагностирование и др)[66].

Несмотря на то, что большая часть теории нейронных сетей придумана еще в прошлом веке, именно сейчас происходит значительный рост их популярности. Это можно объяснить двумя причинами: во-первых, существенный рост производительности компьютеров, во-вторых, произошел резкий рост накопленного объема данных, и этих объемов данных стало достаточно для того, чтобы обучить нейронную сеть решать сложные задачи.

### **3.2 Обучение глубинных нейронных сетей**

Обучение нейронных сетей - это подбор весов, которые соответствуют входам таким образом, чтобы сеть решала поставленную нами задачу. В оригинальной модели искусственных нейронных сетей [9] Маккаллоха и Питтса обучение не производилось. Весы должны были быть заранее заданы таким образом, чтобы сеть решала необходимую задачу. Стандартные задачи, которые могут решать нейронные сети, это задача классификации и регрессии. Задача классификации состоит в том, что сеть должна дать ответ, принадлежит объект какому-то заданному классу или не принадлежит. Задачу регрессии можно описать так: на вход сети поступают вещественные числа и сеть должна выдать



тоже вещественное число. Например, при анализе временных рядов изменения курса валюты, сеть должна предсказать значение курса доллара в следующий момент времени. Основные подходы, которые используются в обучении в нейронных сетях: обучение без учителя, обучение с учителем, обучение с подкреплением. В случае обучения с учителем у нас должен быть набор данных (сигналов), которые подаются на вход нейронной сети и для которых заранее известен правильный ответ. При обучении без учителя используются данные, для которых правильный ответ заранее неизвестен. Обучение без учителя основывается на выявлении структурных различий в данных. Этот метод часто применяется в задаче классификации изображений [66].

При обучении с подкреплением отсутствуют заранее подготовленные наборы данных. Нейронная сеть действует как агент во внешней среде и получает сигналы от внешней среды о правильности или неправильности выполнении действий. Для обучения нейронных сетей чаще всего используется обучение с учителем и в рамках нашей работы также будет рассматриваться только этот подход.

### **3.3 Оценка качества обучения нейронных сетей**

В нейронных сетях и в машинном обучении существует такая проблема как переобучение [66]. Переобучение возникает когда нейронная сеть адаптируется к конкретным особенностям обучающей выборки, которую мы используем. Характеристика сети хорошо работать на тех данных, которые она видела в процессе обучения, называется обобщающей способностью или генерализацией. Переобучение приводит к тому, что обобщающая способность нейронной сети снижается. Сеть может хорошо распознавать те данные, которые

она видела в процессе обучения, но на новых данных работать очень плохо. Поэтому оценивать качество обучения нейронной сети нужно на тех данных, которые она в процессе обучения не видела [15]. Для этого используются два отдельных набора данных: один набор данных для обучения (training set) и другой набор данных для тестирования (test set). После того, как обучение нейронной сети завершено, мы проверяем качество обучения на отдельном наборе для тестирования. Для того, чтобы оценить качество обучения нейронной сети, в Keras используется метод Evaluate. В этот метод мы передаем набор данных для тестирования с правильными ответами, а метод возвращает качество работы. Однако качество работы сети зависит не только от тех параметров, которые мы определяем в процессе обучения (веса входов в нейроны на каждом слое сети), но и от гиперпараметров (количество слоев в нейронной сети и нейронов в каждом слое, количество эпох обучения, размер мини-выборки для обучения, величина параметра скорости обучения, который говорит, насколько сильно мы изменяем веса при каждом шаге и архитектура нейронной сети [64]).

Гиперпараметры не определяются в процессе обучения, а должны быть явно заданы разработчиком. Гиперпараметры подбираются таким образом, чтобы качество работы сети было максимальным [20]. При подборе гиперпараметров сети также может возникнуть переобучение. Нейронная сеть обучается на обучающем наборе данных, качество работы сети проверяется на наборе данных для тестирования. При изменении гиперпараметров обучение и оценка нейронной сети проводится повторно и возможна дальнейшая настройка гиперпараметров до тех пор, пока качество работы не станет приемлемым. Однако для полной оценки качества работы сети и ее обобщающей способности

не достаточно двух наборов данных и часто дополнительно используется проверочная выборка (validation set).

Ошибка на наборе данных для обучения и для проверки должна снижаться, а качество обучения увеличиваться. В случае, если ошибка на наборе данных для обучения снижается, а на наборе данных для проверки ошибка увеличивается, качество обучения на проверочном наборе данных снижается. Это является признаком переобучения и необходимо останавливать процесс обучения. Если не обнаружить процесс переобучения, обобщающая способность нейронной сети будет низкой, как и польза от практического использования [66].

### **3.4 Библиотеки для глубоких нейронных сетей. Выбор языка программирования и библиотек**

Обучение нейронных сетей это достаточно сложная задача, но, в то же время, это задача типовая. Сейчас существует большое количество библиотек, которые позволяют обучать нейронные сети [15]. В рамках нашей работы мы ознакомимся с существующими библиотеками и оценим возможности их применения.

Существуют библиотеки глубокого обучения для разных языков, таких как C++, Python, Java. Свои библиотеки выпустили крупные компании и университеты.

В рамках данной работы было решено использовать Python, по нескольким причинам. Этот язык достаточно простой, но он хорошо подходит для работы и экспериментов с глубокими нейронными сетями [20]. Программы, написанные на Python можно легко модифицировать, например, чтобы опробовать различные

гиперпараметры. В то же время программа при таких больших модификациях остается читаемой и её можно легко поддерживать.

Минус программ на Python в том, что они работают медленнее чем программы на C++, но существующие библиотеки глубокого обучения используют различные методы для ускорения программ. Одна из основных библиотек, которую мы будем использовать это Tensorflow. Tensorflow - это библиотека машинного обучения от компании Google, которая в настоящий момент являются одной из лидирующих компаний, на практике применяющих машинное обучение [20]. Tensorflow — это библиотека с открытыми исходными кодами. Два основных компонента Tensorflow, которые включены в название, это работа с многомерными матрицами (тензорами), которые широко применяются при обучении нейронных сетей, и вычисления в виде потоков данных через графы расстояний. Tensorflow позволяет писать программы на C++ и Python. Также для разработки мы будем использовать библиотеку Keras. Keras - это библиотека глубокого обучения, которую используют Tensorflow и Theano для выполнения эффективных вычислений [66]. Keras является надстройкой над Tensorflow, эта библиотека позволяет описывать нейронную сеть в понятном для разработчика виде, то есть задавать её слои, использование функции активации, метод оптимизации для уменьшения ошибки и другие параметры, важные для обучения нейронной сети. Используя эти параметры, библиотека Keras строит необходимую сеть и для проведения вычислений будет вызывать высокоэффективные методы из библиотеки Tensorflow. В качестве языка программирования в Keras используется Python.

## **4 СЕНТИМЕНТ-АНАЛИЗ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ**

Проведя сравнение существующих методов sentiment-анализа в предыдущей главе, мы пришли к выводу, что наиболее эффективными подходами в настоящий момент являются подходы, основанные на нейронных сетях.

### **4.1 Обработка естественного языка с помощью нейронных сетей**

Нейронные сети активно используются в обработке естественного языка [13]. Традиционные лингвистические задачи, которые можно решить с помощью нейронных сетей: классификация текстов, определение эмоциональной окраски текста, извлечение сущностей из текста, реферирование/аннотирование, чат-боты, машинный перевод, генерация текста [3].

Для обработки текстов используются различные архитектуры нейросетей: рекуррентные нейронные сети (простые, GRU, LSTM), одномерные свёрточные нейронные сети, сети с вниманием [15].

### **4.2 Представление текста для анализа нейросетью**

Поскольку входными данными в нейронную сеть являются числа, так как нейронная сеть умеет работать лишь с ними, перед началом работы нам необходимо преобразовать текст в набор чисел. В случае работы со структурированными данными, также необходимо представить их в виде, пригодном для обработки нейронной сетью. Категориальные признаки, например, оценка 0-5, можно представить в виде вектора One hot encoding с помощью специального значения Dummy variables в библиотеке Pandas.

Представление текста в набор чисел называется векторизацией и это довольно сложный процесс.

На первом этапе (токенизация) текст разбивается на отдельные части, каждая из которых будет представляться в цифровом виде отдельно. В соответствии с уровнями, текст можно представить в виде последовательности символов, слов или целых предложений. В отличие от классических способов обработки текстов, при работе с нейросетями нет необходимости прибегать к методам извлечения признаков из текста (N-граммы, метод bag of words и др).

Далее следует этап векторизации. После того, как текст разбивается на токены, каждый токен должен быть преобразован в число. Для этого может использоваться один из следующих методов: числовое кодирование, плотные векторные представления и One hot encoding.

При числовом кодировании каждому символу ставится числовой код в соответствии с кодировкой (как в кодировках ASCII, UTF-8 и др.).

При использовании метода One hot encoding каждому символу может соответствовать не одно число, а числовой вектор. Вектор содержит столько чисел, сколько токенов может быть использовано, при этом все его элементы, кроме одного, который и соответствует токену, равны нулю.

Метод плотных векторных представлений заключается в том, что каждому токену сопоставляется вектор, но, в отличие от метода One hot encoding, в векторе могут использоваться не только нули и единицы, но и любые числа [15].

В нейронных сетях плотное векторное представление слов определяется в процессе обучения. На первом этапе элементы векторов инициализируются

случайными числами. Изменение значений векторов происходит с помощью метода обратного распространения ошибки.

Двумерное плотное представление векторов, которые используются для представления слов, показано на рисунке 10. Как можно заметить, слова с противоположным значением занимают противоположные положения в системе координат.



Рисунок 10 – Двумерное плотное представление векторов, которые используются для представления слов

Плотные векторные представления также используются для визуализации семантических графов, как показано на рисунке 11.

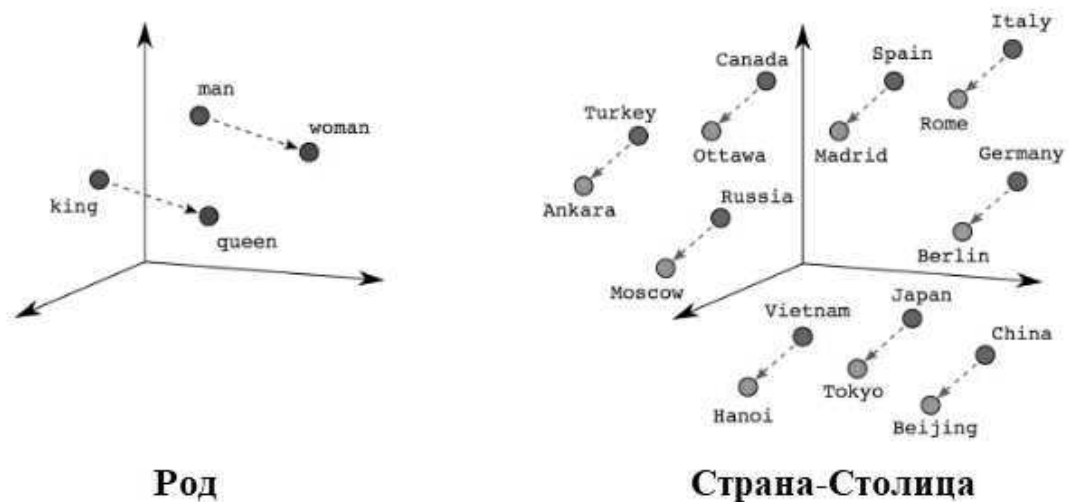


Рисунок 11 – Визуализация семантических графов с помощью плотных векторных представлений

Как можно заметить, благодаря обучению, слова сгруппированы по категориям: слова с положительной эмоциональной окраской находятся в правом верхнем углу, слова с отрицательной эмоциональной окраской находятся в нижнем левом углу. При этом близкие по значению слова находятся в векторном представлении близко друг к другу, а антонимы (good, bad) находятся в противоположном конце системы координат.

Таким образом, векторное представление представляется перспективным методом для задачи сентимент-анализа.

### 4.3 Подготовка среды разработки

Современные нейронные сети имеют высокие требования к аппаратной и программной среде [15]. Проанализировав теоретически и опробовав на практике несколько опций работы с нейросетями (установка на десктопную рабочую станцию под ОС Linux, установка на удаленный сервер VPS под ОС Linux и использование облачных технологий), за простоту настройки и скорость



работы нами была выбрана облачная технология Google Colaboratory. Сервис Google Colaboratory отличается удобством в работе: там уже установлен Python, распространенные библиотеки и датасеты для работы с нейронными сетями (TensorFlow, Keras, Numpy и др.), а вычисления происходят на мощных GPU Nvidia Tesla T4. Технические характеристики GPU – 2560 ядер CUDA, 8.1 Tflops, 16 ГБ памяти. Кроме того, существует возможность совместной работы и хранения результатов в облачном хранилище, как и у любых Google документов.

Однако стоит заметить, что платформа Google Colaboratory имеет интерфейс только на английском языке, что может создать некоторые неудобства в работе для отдельных разработчиков.

#### **4.4 Данные для анализа**

Для анализа использовался открытый набор данных Yelp Open Dataset, представляющий собой отзывы пользователей сайта <https://www.yelp.com/>, общий вид которого представлен на рисунке 12.

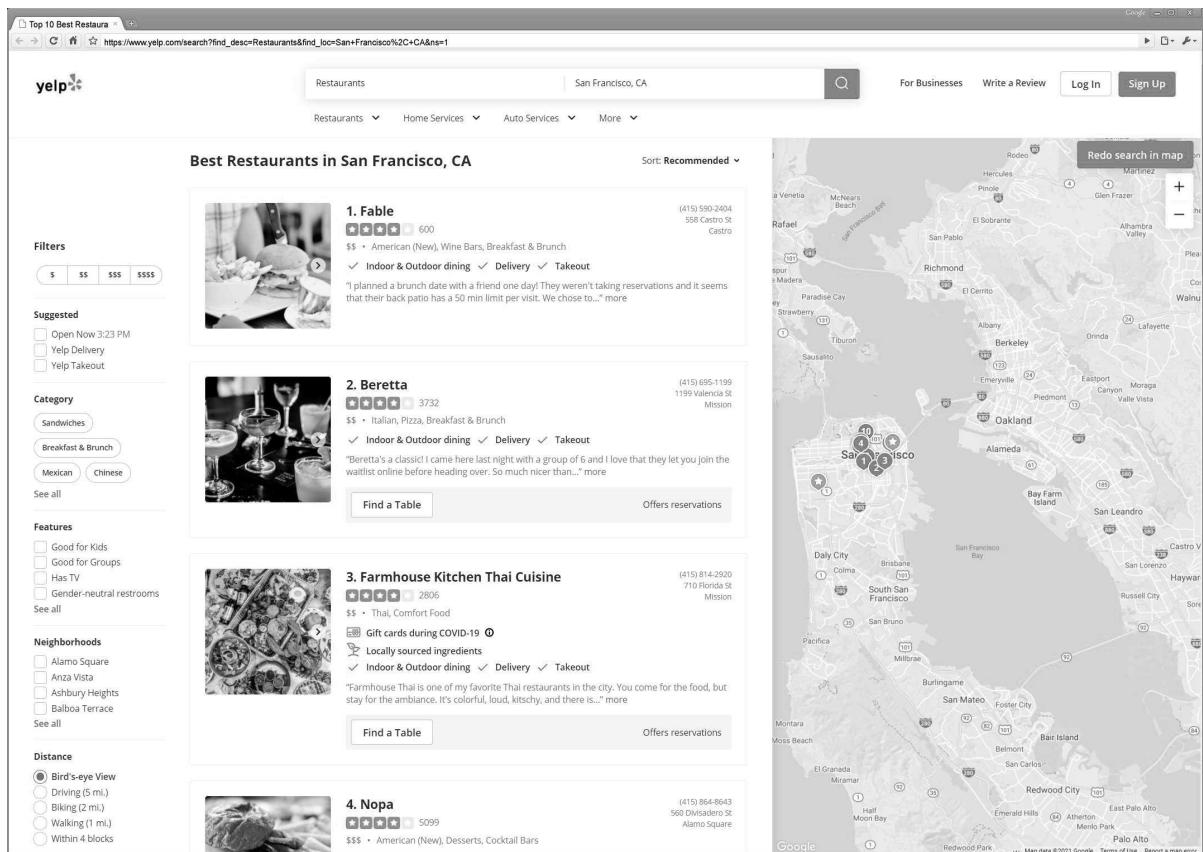


Рисунок 12 – Отзывы пользователей на сайте Yelp.com

На сайте можно найти отзывы пользователей по таким категориям как рестораны, службы доставки, услуги электрика, сантехника, уборка дома, ландшафтные работы, техническое обслуживание, ремонт автомобилей, прачечные, химчистки, салоны красоты и многим другим услугам. Помимо самих отзывов в текстовом виде, пользователи также дают оценку в звёздах, что можно удобно использовать для целей сентимент-анализа и классификации отзывов на положительные и отрицательные.

Используемый нами набор данных содержит 560 000 записей для обучения нейронной сети и 38 000 записей для тестирования: по 280 000 записей для обучения и по 19 000 записей для тестирования на каждую полярность. Отрицательная полярность относится к классу 1, а положительная - к классу 2.

Набор данных был выгружен с сайта <https://course.fast.ai/datasets> в виде архива. Файл `Readme.txt` содержит описание набора данных. Файлы `train.csv` и `test.csv` содержат все обучающие образцы в виде значений, разделенных запятыми. В них 2 столбца, соответствующие индексу класса (1 и 2) и тексту отзыва. Текст отзыва экранируется двойными кавычками (`"`), а любая внутренняя двойная кавычка экранируется двумя двойными кавычками (`" "`). Новые строки экранируются обратной косой чертой, за которой следует символ `"n"`, то есть `"\n"`.

Отзывы не ограничены по объёму, встречаются как очень короткие отзывы, так и достаточно длинные. Для целей сентимент-анализа необходимо будет унифицировать отзывы по длине текста.

Пример набора данных представлен на рисунке 13.

```
test.csv - Блокнот
Файл  Правка  Формат  Вид  Справка
the sticks - not an established D&B. \n\nI was just really turned off from that experience. \n\nThe good
is that D&B provides a different type of entertainment when you want to mix things up. I remember going
here with my grandparents when I was a kid and it was the best treat ever! We would eat at the
restaurant and then spend hours playing games. This place holds some really good memories for me. \n
\nIt's a shame that my experience last night has spoiled the high standards I held for it."
"1", "I used to love D&B when it first opened in the Waterfront, but it has gone down hill over the
years. The games are not as fun and do not give you as many tickets and the prizes have gotten cheaper
in quality. It takes a whole heck of a lot of tickets for you to even get a pencil! The atmosphere is
okay but it used to be so much better with the funnest games and diverse groups of people! Now, it is
run down and many of the games are app related games (Fruit Ninja) and 3D Experience rides. With such
\"games\", you can't even earn tickets and they take a lot of tokens! Last time I went, back in the
winter, many of the games were broken, which made for a negative player experience. I would go to D&B to
play some games again in the future, but it is no longer one of my favorite places to go due to the
decline of fun games where you can earn tickets."
"2", "Like any Barnes & Noble, it has a nice comfy cafe, and a large selection of books. The staff is
very friendly and helpful. They stock a decent selection, and the prices are pretty reasonable.
Obviously it's hard for them to compete with Amazon. However since all the small shop bookstores are
gone, it's nice to walk into one every once in a while."
"1", "Meh, I've experienced better is an understatement.\n\nFriday's is like the \"fine dining chain\"
for every yinzer in Pittsburgh...\n\n1. When we were seated, it was quick which was a surprise.
Somehow this restaurant gets packed and I never understand why. I was happy about that.\n2. I ordered
one of their \"skinny margaritas\", blackberry to be exact. It was a nice size, however, at
$6.00/pop, that was half of my food bill.\n3. The waitress started off attentive, but after our food
came out she was gone. I ordered a turkey burger with pickle and mustard. Loaded mashed potato on the
side because I wanted heartburn ha!\n4. Food came. My burger had lettuce on it only. Waitress was
supposed to go and get our next round of drinks. I had to wait to ask for my pickle.\n5. The loaded
potatoes were more like rubber than potatoes and not what I was expecting. Disappointment.\n6. The
waitress then went into oblivion and only returned with our check. \n\nNever again will I eat at
Friday's. I will, however, go should a happy hour arise..."
"2", "Unos has been around for ever, & I feel like this restaurant chain peak in popularity in the 80's.
Honestly the decor inside still kind of looks 80s to me even though its nice with sleek booth and
exposed brick.\n\nIf you haven't died and he recently I ordered you to come back and have a meal here
again because honestly the food is really quite good!\n\nThey have the best chicken salad wrap ever! I
love that rap so much I want agreed to walk the south side River Trail from where the Steelers practice
all the way to the damn waterfront just because I knew that I could convince my boyfriend to go to Unos
with me for lunch. Full disclosure: I made him call is a cab and we took a taxi back to the parking
lot after lunch.\n\nListen... The food and pizza and service are very good, surprisingly so! I don't
know why this place is not busier but next time you're down at the Waterfront please do consider dining
here!"
"1", "Stars are for the food only. 2.5 rating\n\nGot there at 8pm on Saturday night, they told us it
was an hour wait which was expected, but it only took 30minutes to be seated so that was a bonus.
Started off with the lettuce wraps and i used to be a fan of these but they just seem to bore me now
(FYI-love the lettuce wraps at the cheesecake factory). I had the Singapore rice noodles. This was so
so as i thought immediately that lu lu noodles and asian cafe are better. The fried rice was
disappointing and the vegetables tasted raw. Some good notes were the Dynamite shrimp, plump, juicy,
tangy with a small kick. Sweet and sour chicken was just the same minus the kick. \n\nThe service was
annoying. We were there less than 45 minutes and it seemed like our waiter was in a rush to get us out
of there. Constantly asking if he could remove my plate. I had to sternly say at one point that I
will let you know when i am finished. I found it rude especially since there wasn't even a line at
this point. \n\nI think we will travel elsewhere following a movie. The result never quite met the
anticipation."
"1", "Our last few visits have left something to desire, One of the things I always notice is the
```

Рисунок 13 – Примеры из набора данных для тестирования

## 4.5 Разработка нейронной сети на базе библиотек Keras и Tensorflow, представление текста вектором One Hot Encoding

Попробуем выполнить sentiment-анализ с помощью полносвязной нейронной сети и вектора One Hot Encoding

Загрузим необходимые сущности из библиотеки Keras

— Sequential model - будем использовать полносвязную модель нейронной сети;

— Dense layer - этот слой используется для классификации и он обрабатывает полученные ранее значения в 0 или 1;

— Numpy - пакет для работы с числами в Python;

— Pandas — библиотека для работы с данными

— Matplotlib - библиотека Python для визуализации данных двумерной графикой (для использования в НИР)

На листинге 1 представлена загрузка библиотек.

Листинг 1 – Загрузка библиотек

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.text import Tokenizer
%matplotlib inline
```

Далее происходит загрузка данных. Для их кодирования будем использовать словарь, состоящий из 10 000 наиболее употребимых слов, что описано на листинге 2.

Листинг 2 – Загрузка словаря

```
max_words=10000
(x_train, y_train), (x_test, y_test) =
imdb.load_data(num_words=max_words)
```

В массиве каждое число соответствует одному слову из частотного словаря в 10 000 слов. В качестве примера посмотрим как представляется один из отзывов набора данных. Код представлен на листинге 3.

Листинг 3 – Представление текстового отзыва в качестве числового массива

```
x_train[10]
[1, 194, 1153, 194, 8255, 78, 228, 5, 6, 1463, 4369, 5012,
134, 26, 4, 715, 8, 118, 1634, 14, 394, 20, 13, 119, 954, 189, 102,
5, 207, 110, 3103, 21, 14, 69, 188, 8, 30, 23, 7, 4, 249, 126, 93,
4, 114, 9, 2300, 1523, 5, 647 <...>]
```

Функция для кодирования one hot encoding. В функции задаётся вектор заданной размерности (в нашем случае размерность 10000). В вектор записываются нули (если слово под порядковым номером элемента не встретилось) и единицы (там где слово встретилось). Работа функции кодирования one hot encoding представлена на листинге 4.

```
Листинг 4 – Работа функции кодирования one hot encoding
def vectorize_sequences(sequences, dimension=10000):
    results = np.zeros((len(sequences), dimension))
    for i, sequence in enumerate(sequences):
        results[i, sequence] = 1.
    return results
```

```
x_train = vectorize_sequences(x_train, max_words)
x_test = vectorize_sequences(x_test, max_words)
```

Выведем какой-нибудь вектор. Предположим, вектор отзыва №5 с учётом 50 первых слов. Он состоит из 0 и 1. К сожалению, мы теряем информацию о позиции слова в тексте и частоте его встречаемости. Вывод массивы представлен на листинге 5.

```
Листинг 5 – Вывод вектора отзыва
x_train[5][:50]
array([0., 1., 1., 0., 1., 0., 0., 0., 1., 0., 1., 0., 1., 0.,
0., 1., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0.,
```

```
0., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 1., 0., 1., 0., 0.,  
0., 0.] )
```

```
len(x_train[0])  
10000  
y_train[0]  
1
```

На листинге 6 показано создание нейронной сети. Создаём нейронную сеть из 3 слоёв: 128 нейронов, 64 нейрона и 1 нейрон. Типы слоёв - полносвязные. Размерность соответствует максимально используемому количеству слов (10000)

Листинг 6 – Создание нейронной сети

```
model = Sequential()  
model.add(Dense(128, activation='relu',  
input_shape=(max_words,)))  
model.add(Dense(64, activation='relu'))  
model.add(Dense(1, activation='sigmoid'))
```

Скомпилируем нейронную сеть с помощью компилятора Adam как показано на листинге 7. В качестве функции потерь используется бинарная кросс-энтропия, т.к. наша задача является задачей классификации с двумя классами.

Перекрестная энтропия определяется следующим образом:

$$H(p,q) = E_p[-\log(q)] = H(p) + D_{KL}(p||q)$$

$H(p)$  – энтропия  $p$

$DKL$  – относительная энтропия, т. е. расстояние Кульбака-Лейблера от  $p$  до  $q$ .

В качестве метрики используется Accuracy, которая показывает долю правильных ответов. Далее подберём гиперпараметры нейронной сети, как показано на листинге 7.

Листинг 7 – Компиляция и подбор гиперпараметров нейронной сети

```
model.compile(optimizer='adam',  
loss='binary_crossentropy',  
metrics=['accuracy'])  
  
history = model.fit(x_train,
```

```
y_train,  
epochs=10,  
batch_size=128,  
validation_split=0.1)
```

Визуализируем результаты тренировочной выборки с помощью графика, как представлено на листинге 8.

Листинг 8 – Визуализация результатов тренировочной выборки

```
plt.plot(history.history['acc'],  
label='Доля верных ответов на обучающем наборе')  
plt.plot(history.history['val_acc'],  
label='Доля верных ответов на проверочном наборе')  
plt.xlabel('Эпоха обучения')  
plt.ylabel('Доля верных ответов')  
plt.legend()  
plt.show()
```

Результат визуализации представлен на рисунке 14.

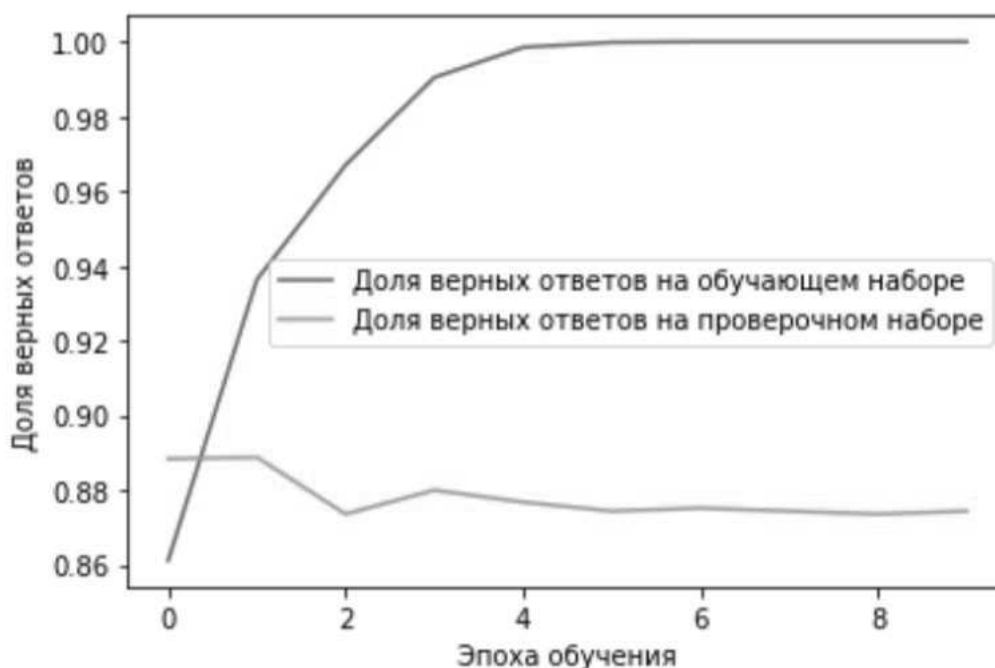


Рисунок 14 – Визуализация работы разработанной нейронной сети

Проверяем работу сети на тестовом наборе данных, как показано на листинге 9.

Листинг 9 – Результаты работы нейронной сети

```
scores = model.evaluate(x_test, y_test, verbose=1)
```

```
25000/25000 [=====] - 4s 145us/sample  
- loss: 0.9792 - acc: 0.8636
```

На тренировочной выборке нейронная сеть достигла точности 86%.

#### 4.6 Разработка нейронной сети для сентимент-анализа с использованием плотного вектора

Продолжим разработку нейронной сети, однако используем другой метод представления текста - плотные векторные представления.

Метод плотных векторных представлений заключается в том, что каждому токену текста (символу, слову, предложению) ставится в соответствие вектор относительно небольшой длины, в котором могут использоваться любые значения, а не только нули и единицы.

Подключим необходимые нам модули, в том числе слой `embedding`, который используется для создания плотных векторных представлений. Загрузка библиотек представлена на листинге 10.

Исходный формат данных: отзывы разделены на отдельные слова, каждое слово представлено числом, соответствует частоте, с которой это слово встречается в наборе данных. Правильные ответы (положительный отзыв или отрицательный) приведены в бинарном формате: 1 — отзыв положительный, 0 — отзыв отрицательный.

```
Листинг 10 – Загрузка библиотек  
# загрузим необходимую модель - полносвязная нейронная сеть  
from tensorflow.keras.models import Sequential  
  
# загрузим слои нейронной сети  
# слой embedding используется для создания плотных векторных п  
редставлений  
from tensorflow.keras.layers import Dense, Embedding, Flatten,  
Dropout  
from tensorflow.keras import utils  
from tensorflow.keras.preprocessing.sequence import pad_se-  
quences
```



```
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files
%matplotlib inline
```

Создаваемая нами нейронная сеть может обрабатывать данные фиксированной длины, поэтому необходимо ограничить длину отзыва — не более 250 слов. Унифицируем отзывы по длине с помощью функции `pad_sequences`. С помощью этой функции все отзывы, длина которых больше 250 слов, будут обрезаны, а отзывы меньшей длины будут дополнены специальными символами-заполнителями. Процесс унификации описан в листинге 11.

Листинг 11 – Унификация отзывов по длине

```
# Унифицируем отзывы по длине
maxlen = 250
x_train = pad_sequences(x_train, maxlen=maxlen, padding='post'
)
x_test = pad_sequences(x_test, maxlen=maxlen, padding='post')
```

Создадим нейронную сеть, в которой будем использовать плотные векторные представления. Нейронная сеть — последовательная, тип `Sequential`. Добавим слой `embedding` - слой плотных векторных представлений слов. Первый параметр это количество слов (в нашем случае — 10000), второй параметр - длина вектора, которым будут представляться слова. Для простоты и удобства визуализации в этом примере мы будем использовать длину векторов 2.

Укажем размер входных данных `maxlen` (в нашем случае - 250 слов).

Создадим второй слой - `dropout` для снижения переобучения. Слой `embedding` выдает массив, в котором 250 элементов размерности 2. Преобразуем этот массив в плоский вектор с помощью слоя `Flatten`. На выходе нейронной сети зададим один нейрон, который соответствует задаче бинарной классификации: если выход нейрона — 1, то отзыв положительный, если выход — 0, отзыв

отрицательный. Для классификации используется функция Sigmoid. Создание нейронной сети описано на листинге 12.

Листинг 12 – Создание нейронной сети, подключение слоев

```
model = Sequential()  
# слой embedding - слой плотных векторных представлений слов  
model.add(Embedding(max_words, 2, input_length=maxlen))  
# слой - dropout для снижения переобучения  
model.add(Dropout(0.25))  
# слой flatten - преобразование массива в плоский вектор  
model.add(Flatten())  
# бинарная классификация с помощью функции Sigmoid  
model.add(Dense(1, activation='sigmoid'))
```

Слой `embedding` определяет векторы для представления слов, которые определяются в процессе обучения нейронной сети. На первом этапе, при обучении других слоёв нейронной сети, векторы слоя `embedding` инициализируются случайными значениями, затем выполняется обычное обучение нейронной сети с помощью алгоритма обратного распространения ошибки.

Для компиляции модели будет использоваться оптимизатор `Adam`, функция ошибки — бинарная перекрестная энтропия которая подходит для задачи бинарной классификации.

В качестве метрики качества обучения вычисляется доля правильных ответов.

Приступим к обучению модели с помощью наборов `x_train` (данные для обучения) и `y_train` (набор правильных ответов). Количество эпох — 10, размер мини-выборки — 128, для проверочной выборки используются 10 процентов из набора данных `x_train` (данные для обучения) и `y_train`. Фитинг модели описан на листинге 13.

Листинг 13 – Подбор гиперпараметров нейронной сети

```
history = model.fit(x_train,  
y_train,  
epochs=10,  
batch_size=128,  
validation_split=0.1)
```

Сеть обучается успешно и на последней эпохе обучения доля правильных ответов на обучающем наборе данных составляет 92%, на проверочном наборе данных — 88 %. Можно заметить, что доля правильных ответов растёт в процессе обучения.

Процесс обучения нейронной сети представлен на рисунке 15.

```
history = model.fit(x_train,  
y_train,  
epochs=10,  
batch_size=128,  
validation_split=0.1)
```

Epoch	176/176	Time	loss	accuracy	val_loss	val_accuracy
Epoch 1/10	176/176	5ms/step	0.6927	0.5064	0.6827	0.6040
Epoch 2/10	176/176	4ms/step	0.6581	0.6739	0.5355	0.7868
Epoch 3/10	176/176	4ms/step	0.4824	0.8090	0.3969	0.8444
Epoch 4/10	176/176	4ms/step	0.3581	0.8650	0.3473	0.8604
Epoch 5/10	176/176	4ms/step	0.2989	0.8855	0.3207	0.8700
Epoch 6/10	176/176	4ms/step	0.2672	0.8971	0.3060	0.8784
Epoch 7/10	176/176	4ms/step	0.2440	0.9084	0.2967	0.8812
Epoch 8/10	176/176	4ms/step	0.2221	0.9206	0.2908	0.8848
Epoch 9/10	176/176	4ms/step	0.2093	0.9254	0.2883	0.8844
Epoch 10/10	176/176	4ms/step	0.1956	0.9293	0.2863	0.8832

Рисунок 15 – Обучение нейронной сети в 10 эпох

Повторим опыт с использованием большего количества эпох.

Процесс обучения нейронной сети в 17 эпох представлен на рисунке 16.

```
history = model.fit(x_train,
                    y_train,
                    epochs=17,
                    batch_size=128,
                    validation_split=0.1)

Epoch 1/17
176/176 [=====] - 1s 5ms/step - loss: 0.1892 - accuracy: 0.9296 - val_loss: 0.2840 - val_accuracy: 0.8868
Epoch 2/17
176/176 [=====] - 1s 4ms/step - loss: 0.1785 - accuracy: 0.9357 - val_loss: 0.2874 - val_accuracy: 0.8844
Epoch 3/17
176/176 [=====] - 1s 4ms/step - loss: 0.1696 - accuracy: 0.9397 - val_loss: 0.2871 - val_accuracy: 0.8852
Epoch 4/17
176/176 [=====] - 1s 4ms/step - loss: 0.1651 - accuracy: 0.9420 - val_loss: 0.2886 - val_accuracy: 0.8852
Epoch 5/17
176/176 [=====] - 1s 4ms/step - loss: 0.1580 - accuracy: 0.9444 - val_loss: 0.2895 - val_accuracy: 0.8828
Epoch 6/17
176/176 [=====] - 1s 4ms/step - loss: 0.1515 - accuracy: 0.9457 - val_loss: 0.2936 - val_accuracy: 0.8820
Epoch 7/17
176/176 [=====] - 1s 4ms/step - loss: 0.1458 - accuracy: 0.9500 - val_loss: 0.2943 - val_accuracy: 0.8848
Epoch 8/17
176/176 [=====] - 1s 4ms/step - loss: 0.1416 - accuracy: 0.9507 - val_loss: 0.2965 - val_accuracy: 0.8856
Epoch 9/17
176/176 [=====] - 1s 5ms/step - loss: 0.1384 - accuracy: 0.9508 - val_loss: 0.2992 - val_accuracy: 0.8840
Epoch 10/17
176/176 [=====] - 1s 4ms/step - loss: 0.1311 - accuracy: 0.9547 - val_loss: 0.3038 - val_accuracy: 0.8808
Epoch 11/17
176/176 [=====] - 1s 4ms/step - loss: 0.1261 - accuracy: 0.9568 - val_loss: 0.3081 - val_accuracy: 0.8832
Epoch 12/17
176/176 [=====] - 1s 4ms/step - loss: 0.1255 - accuracy: 0.9561 - val_loss: 0.3143 - val_accuracy: 0.8764
Epoch 13/17
176/176 [=====] - 1s 5ms/step - loss: 0.1198 - accuracy: 0.9586 - val_loss: 0.3131 - val_accuracy: 0.8784
Epoch 14/17
176/176 [=====] - 1s 5ms/step - loss: 0.1182 - accuracy: 0.9604 - val_loss: 0.3166 - val_accuracy: 0.8820
Epoch 15/17
176/176 [=====] - 1s 4ms/step - loss: 0.1146 - accuracy: 0.9609 - val_loss: 0.3199 - val_accuracy: 0.8788
Epoch 16/17
176/176 [=====] - 1s 4ms/step - loss: 0.1100 - accuracy: 0.9610 - val_loss: 0.3250 - val_accuracy: 0.8772
Epoch 17/17
176/176 [=====] - 1s 5ms/step - loss: 0.1098 - accuracy: 0.9624 - val_loss: 0.3294 - val_accuracy: 0.8780
```

Рисунок 16 – Обучение нейронной сети в 17 эпох

Однако, увеличение количества эпох привело к обратному результату — качество обучения на проверочном наборе немного снизилось, что говорит о переобучении нейронной сети и подчеркивает важность верного подбора параметров.

Можно сделать вывод, что плотные векторные представления слов могут использоваться для представления текста в цифровом виде для последующей обработки с помощью нейронной сети.

#### 4.7 Недостатки использования полносвязных нейронных сетей для сентимент-анализа

При анализе текста полносвязной нейронной сетью (слои dense в библиотеке Keras) на вход каждому нейрону попадают все токены (слова), которые анализируются независимо друг от друга. Информация о порядке слов теряется. Таким образом, хотя лексическая информация обрабатывается неплохо, синтаксические конструкции, также обладающие своей семантикой,

игнорируются. В целом, работа полносвязной нейронной сети похожа на работу более примитивных систем sentiment-анализа, работающих на основе простого поиска по словарю. Отрицательные частицы, некоторые наречия (hardly, never), сослагательное наклонение, фразеологические обороты часто меняют смысл предложения на прямо противоположный, однако полносвязная нейронная сеть не может это идентифицировать.

Порядок слов и связи между ними в тексте имеют большую смысловую нагрузку. Таким образом, более перспективный подход представляется в анализе последовательности токенов (слов), а не просто изолированных токенов. Необходимо использовать нейронную сеть с более сложной архитектурой, способной анализировать последовательность токенов.

#### **4.8 Разработка рекуррентной нейронной сети для sentiment-анализа**

Для задачи анализа последовательности могут использоваться рекуррентные нейронные сети. Рекуррентные нейронные сети отличаются от нейронных сетей с прямым распространением сигнала. В нейронных сетях с прямым распространением сигнала (feedforward networks) циклы запрещены, а сигнал может передаваться лишь в одном направлении - от входа к выходу. Так работают полносвязные нейронные сети и свёрточные нейронные сети. В рекуррентных нейронных сетях используются циклы, при этом выход нейрона может быть подключен к его входу или ко входу всех нейронов текущего слоя. Рекуррентные нейронные сети, благодаря наличию в них циклических соединений, лучше подходят для анализа последовательностей [20].

Архитектура рекуррентных нейронных сетей и сетей с прямым распространением сигнала представлена на рисунке 17.

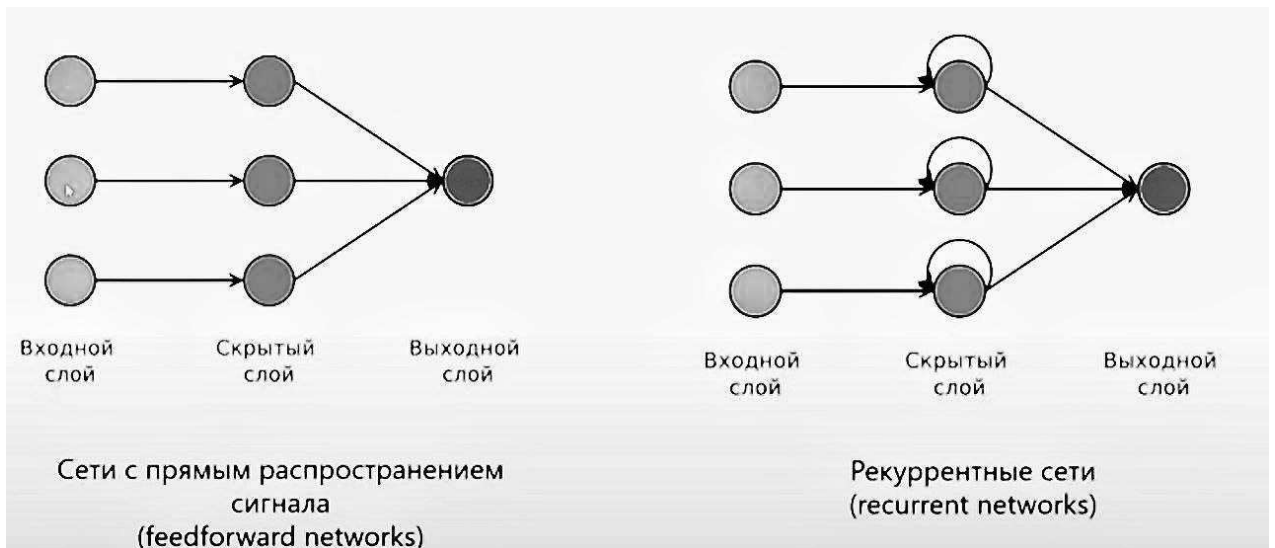
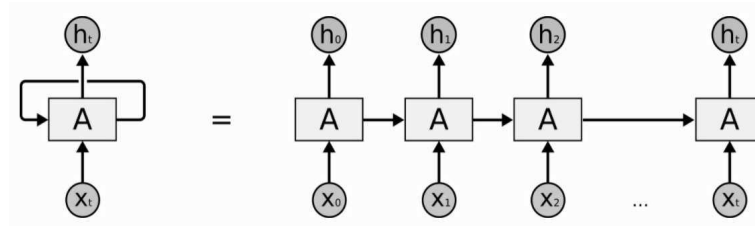


Рисунок 17 – Архитектура рекуррентных нейронных сетей и сетей с прямым распространением сигнала

Рекуррентную нейронную сеть также можно представить как сеть с прямым распространением сигнала, развёрнутую во времени.

Создадим несколько копий нейронной сети с прямым распространением сигнала. На вход копий нейронной сети поступают элементы последовательности (на вход первой копии  $x_1$  поступает первый элемент последовательности, на вход второй копии  $x_2$  поступает второй элемент последовательности и так далее). Каждая копия рекуррентной нейронной сети выдаёт выходное значение ( $a_1, a_2, a_3$  и так далее), а также значения  $h_1, h_2, h_3$  и так далее, которые поступают на входы следующих копий нейронной сети. Эти значения можно назвать скрытым состоянием, учитывающим предыдущие этапы анализа последовательности [15].

Архитектура рекуррентных нейронных сетей представлена на рисунке 18.



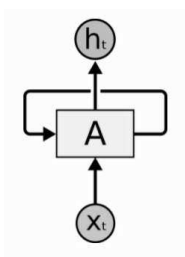


Рисунок 18 – Архитектура рекуррентных нейронных сетей

В отличие от полносвязных нейронных сетей, рекуррентная нейронная сеть может работать с входными данными любой длины.

Для создания рекуррентной нейронной сети в библиотеке Keras используется слой SimpleRNN.

Для обучения рекуррентных нейросетей используется разворачивание во времени и обратное распространение ошибки [64]. Следует заметить, что количество слоёв развёрнутой сети зависит от длины последовательности входных данных: например, для задачи анализа отзыва длиной 200 слов будет использоваться 200 слоёв нейронной сети. Сеть становится очень глубокой и её обучение требует существенных вычислительных ресурсов. Также возникает проблема исчезающего градиента: сигнал об изменении весов уменьшается при передаче от слоя к слою. Для преодоления этих проблем используются специальные архитектуры рекуррентных нейронных сетей - LSTM (Long-Short Term Memory) и GRU (Gated Recurrent Unit) [64].

Архитектура рекуррентных нейронных сетей - LSTM представлена на рисунке 19.

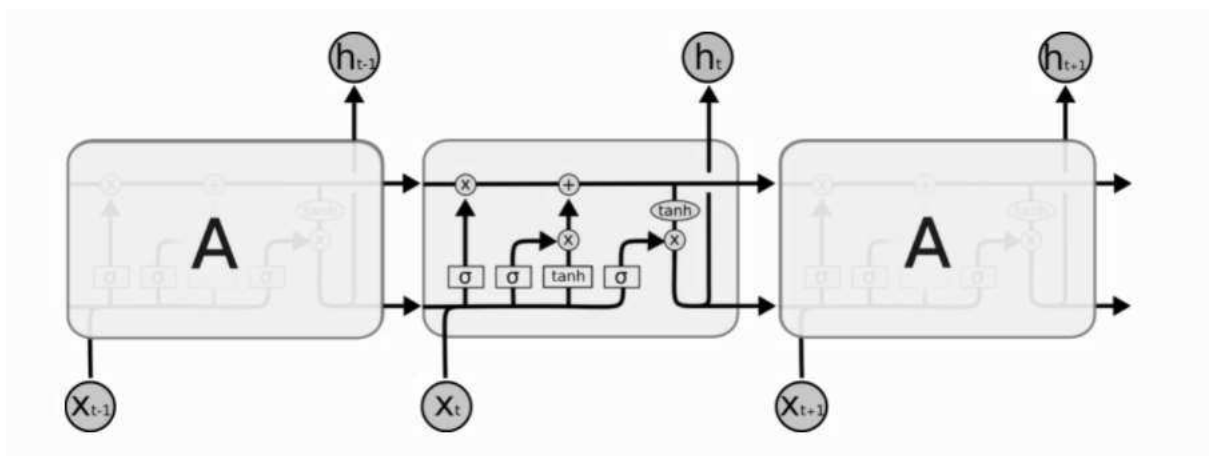


Рисунок 19 – Архитектура рекуррентных нейронных сетей - LSTM

Сети LSTM устроены достаточно сложно. В этих сетях элементом сети являются не один нейрон или слой нейронов, а целый набор слоев, взаимодействующих друг с другом по определенным правилам. Такие наборы в сетях LSTM называются ячейками (LSTM cell) [20]. На рисунке представлена сеть LSTM развернутая во времени. В середине — текущая копия сети, слева — предыдущая копия, а справа — будущая копия. В каждый момент времени на вход сети поступают некоторые элементы последовательности, и в каждый момент времени сеть выдает какое-то выходное значение. Стоит заметить, что сеть LSTM передает на вход своей копии в следующий момент времени не одно значение, а два.

Архитектура ячейки LSTM представлена на рисунке 20.



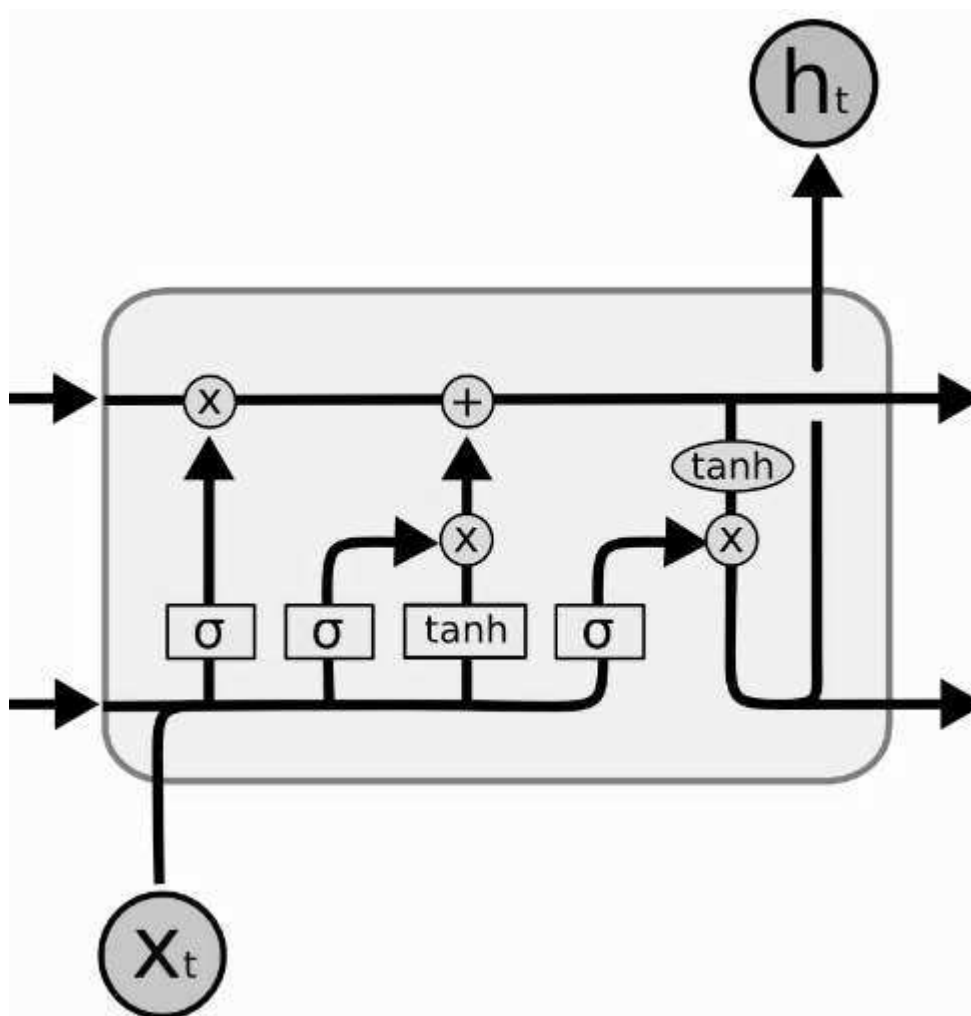


Рисунок 20 – Архитектура ячейки LSTM

Ячейка LSTM состоит из нескольких слоёв нейронов, которые соединяются между собой с помощью операции поэлементного умножения или сложения. Основой в ячейке LSTM является состояние, именно оно позволяет сохранить данные на длительный промежуток времени [66]. На рисунке состояние представлено как выделенное соединение в верхней части ячейки. Слева — сигнал, который поступает с предыдущего этапа работы сети и передается на следующий этап. И именно благодаря тому, что в сети LSTM есть данная выделенная линия для передачи состояния, ячейка LSTM может хранить данные неограниченно долго. Для того, чтобы управлять состоянием ячейки

LSTM, используются так называемые вентили (англ. “gate”), представляющие собой слой нейронов и имеющие на выходе функцию активации сигмоид, в котором значение меняется от нуля до единицы и последующая операция поэлементного умножения [20].

Архитектура вентиля, управляющего состоянием ячейки сети LSTM представлена на рисунке 21.

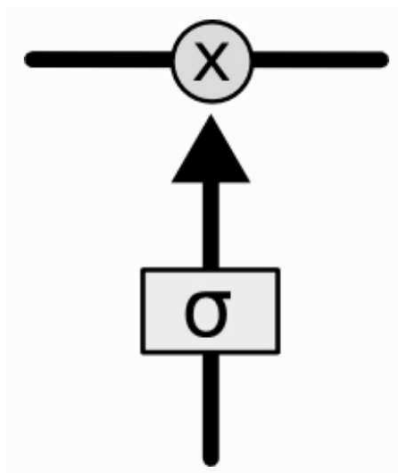


Рисунок 21 – Архитектура вентиля в сети LSTM

В случае, если нейронный слой сигмоид возвращает 1, элемент сигнала проходит без изменений, если на выходе из сигмоида 0, элемент сигнала не передается, а также сигнал может быть ослаблен, если на выходе из сигмоида значения в диапазоне от 0 до 1. В сетях LSTM используются три типа таких вентилях: это вентиль забвения (forget gate), входной вентиль (input gate) и выходной вентиль (output gate). Вентиль забвения управляет процессом стирания данных из ячейки памяти, входной вентиль отвечает за запись данных в ячейку памяти, выходной вентиль нужен для того, чтобы определить какой сигнал будет подаваться на выход нейронной сети [15].

Для обучения нейронных сетей LSTM используется обычный метод обучения с учителем и метод обратного распространения ошибки. Особенность обучения нейронной сети LSTM в том, что она одновременно учится выдавать правильные сигналы на выход и формировать сигналы вентилях (входных, выходных и вентилях забывания), для того чтобы получать верные выходные значения [20].

Сеть LSTM сейчас является очень популярной, однако её недостаток в том, что в ней достаточно много элементов и для обучения требуются значительные вычислительные ресурсы.

Создадим свою собственную нейронную сеть на основе архитектуры LSTM и используем её для сентимент-анализа отзывов на фильмы с сайта `uelp.com`.

Обратимся к облачной платформе Google Colab. Поскольку сеть имеет сложную архитектуру и требует значительных вычислительных ресурсов, мы можем подключить модуль GPU для того, чтобы сеть обучалась быстрее.

Подключение аппаратного ускорителя показано на рисунке 22.

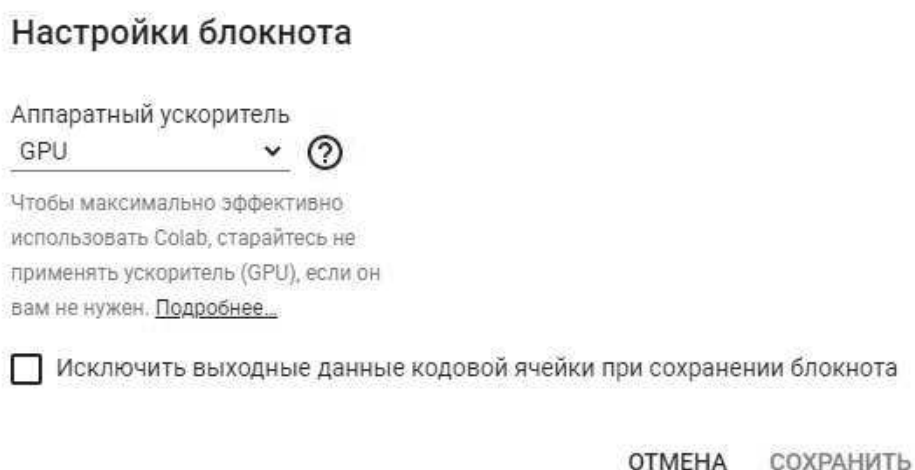


Рисунок 22 – Подключение аппаратного ускорителя GPU

Для начала работы следует подключить необходимые модули из библиотеки Tensorflow и другие библиотеки и утилиты Python, которые будут использоваться для работы с данными, как показано на листинге 14. Архитектура — последовательная нейронная сеть Sequential. Для создания слоя LSTM в библиотеке Tensorflow есть соответствующая команда.

Листинг 14 – Подключение библиотек для начала работы с сетью LSTM

```
%tensorflow_version 2.x
#Sequential model - будем использовать последовательную модель
нейронной сети
from tensorflow.keras.models import Sequential
# загрузим слои нейронной сети для работы
# Dense layer - этот слой используется для классификации и он
обрабатывает полученные ранее значения в 0 или 1
# Embedding Преобразовывает положительные целые числа (индексы
) в плотные (dense) векторы фиксированного размера
from tensorflow.keras.layers import Dense, Embedding, LSTM
# импортируем утилиты для Keras
from tensorflow.keras import utils
# импортируем утилиты для обработки последовательностей
from tensorflow.keras.preprocessing.sequence import pad_sequences
# импортируем токенизатор
from tensorflow.keras.preprocessing.text import Tokenizer
# импортируем обратный вызов - инструмент для настройки поведения
модели Keras во время обучения
from tensorflow.keras.callbacks import ModelCheckpoint
# импортируем pandas - библиотека для работы с данными
import pandas as pd
# импортируем numpy - пакет для работы с числами в Python
import numpy as np
# импортируем Matplotlib - библиотека Python для визуализации
данных двумерной графикой
import matplotlib.pyplot as plt
%matplotlib inline
```

Как описано в листинге 15, ограничим максимальное количество используемых слов как 10000 и максимальную длину отзыва как 100.

Листинг 15 – Настройка максимальной длины отзывов и объёма словаря

```
num_words = 10000
max_review_len = 100
```

Максимальная длина отзывов — 100 слов. Если отзыв больше 100 слов, берутся только первые 100. Если длина отзыва меньше 100 слов, отзыв дополняется нулями. На рисунке 23 показан отзыв, преобразованный в массив.

```
x_train[5002]
array([[ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  1, 103,  51,  13, 197,  16,  6, 4796,
        239,  7, 107, 555,  11, 132,  39, 8921, 1872, 138,  16,
       4093, 1768, 2959,  34, 1620,  2,  11,  4, 755,  12,  43,
        286,  4, 172, 209,  2,  5, 5938,  17, 1000,  34, 1768,
        279, 29, 3892,  89, 338,  4, 114,  16, 1240, 1747, 103,
       3844, 3819,  8, 169, 2875, 112, 7062,  34, 160, 132, 29,
        100, 24, 535, 101, 250,  8, 858,  15, 196,  19, 57,
       3154, 39, 90,  5, 24, 169, 3533, 18,  6, 910, 20,
         14, 1367, 3939, 1101, 935,  4, 545, 1744, 173,  54, 1240,
       2875,  2,  56, 209, 112, 1021,  55, 835, 179,  35, 2384,
         8, 529, 1663, 1793,  8, 1066,  7, 61, 223, 637, 1231,
        19,  4,  86, 31,  5, 353,  8, 859, 13, 126, 219,
         4, 755], dtype=int32)
```

Рисунок 23 – Преобразование отзыва в массив

Далее, как указано на листинге 16 происходит загрузка набора данных, а также можно распаковать архив и убедиться, что данные отображаются в верном формате

#### Листинг 16 – Просмотр загруженных данных

```
!head yelp_review_polarity_csv/test.csv
"2","Contrary to other reviews, I have zero complaints about
the service or the prices. I have been getting tire service here for
the past 5 years now, and compared to my experience with places like
Pep Boys, these guys are experienced and know what they're doing. \
nAlso, this is one place that I do not feel like I am being taken
advantage of, just because of my gender. Other auto mechanics have
been notorious for capitalizing on my ignorance of cars, and have
sucked my bank account dry. But here, my service and road coverage
has all been well explained - and let up to me to decide. \nAnd they
just renovated the waiting room. It looks a lot better than it did
in previous years."
```

"1","Last summer I had an appointment to get new tires and had to wait a super long time. I also went in this week for them to fix a minor problem with a tire they put on. They \"fixed\" it for free, and the very next morning I had the same issue. I called to complain, and the \"manager\" didn't even apologize!!! So frustrated. Never going back. They seem overpriced, too."

С помощью команды `!wc`, как можно наблюдать на листинге 17 можно узнать сколько записей содержится в тренировочном и тестовом наборах.

```
Листинг 17 – Просмотр количества данных в дата сетах
# сколько записей содержится в тренировочном наборе
!wc -l yelp_review_polarity_csv/train.csv
560000 yelp_review_polarity_csv/train.csv

# сколько записей содержится в тестовом наборе
!wc -l yelp_review_polarity_csv/test.csv
38000 yelp_review_polarity_csv/test.csv
```

Далее создаётся сущность `train` для работы с тренировочным набором, как показано на листинге 18.

```
Листинг 18 – Подготовка тренировочного набора к работе
train = pd.read_csv('yelp_review_polarity_csv/train.csv',
```

Для того, чтобы первая строка не считалась заголовком (по умолчанию), нужно прописать обратное. Набор данных состоит из двух колонок. В первой колонке указан класс отзыва (1 для отрицательных отзывов и 2 для положительных), во второй содержится сам текст отзыва пользователя.

```
Листинг 19 – Разметка колонок тестового набора
header=None,
# две колонки - класс (1 или 2) и сам отзыв
names=['Class', 'Review'])
```

Теперь необходимо преобразовать текст к виду, пригодному для обработки нейронной сетью. Для этого следует разбить текст на отдельные слова — токены, а также присвоить каждому токену номер из словаря. Для выполнения указанных задач вызовем токенизатор из библиотеки `Keras` и обучим его на нашем наборе данных, как показано на листинге 20

Листинг 20 – Вызов и обучение токенизатора

```
tokenizer = Tokenizer(num_words=num_words)
# Обучим токенизатор на отзывах Yelp
tokenizer.fit_on_texts(reviews)
# Просмотрим словарь токенизатора
tokenizer.word_index
```

Можно ознакомиться со словарём токенизатора. Самые частотные слова идут в начале списка, далее менее употребляемые.

Автоматически сгенерированный на основе дата сета частотный словарь токенизатора представлен на рисунке 24.

```
{'the': 1,
 'and': 2,
 'i': 3,
 'to': 4,
 'a': 5,
 'was': 6,
 'of': 7,
 'it': 8,
 'for': 9,
 'in': 10,
 'is': 11,
 'n': 12,
 'that': 13,
 'my': 14,
 'we': 15,
 'this': 16,
 'but': 17,
 'with': 18,
 'they': 19,
 'you': 20,
 'on': 21,
 'not': 22,
 'have': 23,
 'had': 24,
 'at': 25,
 'were': 26,
 'so': 27,
 'are': 28,
 'food': 29,
 'be': 30,
 'place': 31,
```

Рисунок 24 – Словарь слов из отзывов сайта Yelp.com, составленный с помощью токенизатора

Можно сравнить полученный словарь токенизатора со списком наиболее употребляемых слов английского языка Oxford English Corpus [65].

Как можно заметить, наиболее частотные слова по большей части совпадают, однако есть и отличия, обусловленные спецификой используемого набора данных. Так, личные местоимения I (я) и we (мы) в частотном словаре отзывов Yelp используются значительно чаще чем в общем корпусе английского языка, поскольку отзывы пишутся, как правило, от первого лица. Кроме того, в словаре токенизатора очень частотными словами оказались food и place (29 и 31 место). В общем корпусе английского языка эти слова не попали в первые 100 по употребимости. Таким образом, использовать токенизатор вместо стандартного частотного словаря является оправданным решением.

Со списком наиболее употребляемых слов английского языка Oxford English Corpus можно ознакомиться на рисунке 25.



Word ↕	Parts of speech ↕	OEC rank ↕
the	Article	1
be	Verb	2
to	Preposition	3
of	Preposition	4
and	Conjunction	5
a	Article	6
in	Preposition	7
that	Conjunction et al.	8
have	Verb	9
I	Pronoun	10
it	Pronoun	11
for	Preposition	12
not	Adverb et al.	13
on	Preposition	14
with	Preposition	15
he	Pronoun	16
as	Adverb, conjunction, et al.	17
you	Pronoun	18
do	Verb, noun	19
at	Preposition	20
this	Determiner, adverb, noun	21

Рисунок 25 – Список наиболее употребляемых слов английского языка

Далее мы можем преобразовать текст отзыва в числовой массив сопоставляя слова текста соответствующим им числам словаря токенизатора.

Листинг 21 – Вывод отзыва в виде текста и в виде числового массива

```
index = 3
print(reviews[index])
print(sequences[index])
```

На листинге 22 показано преобразование текста в числовой массив.

## Листинг 22 – Преобразование текста в числовой массив

I'm writing this review to give you a heads up before you see this Doctor. The office staff and administration are very unprofessional. I left a message with multiple people regarding my bill, and no one ever called me back. I had to hound them to get an answer about my bill. \n\nSecond, and most important, make sure your insurance is going to cover Dr. Goldberg's visits and blood work. He recommended to me that I get a physical, and he knew I was a student because I told him. I got the physical done. Later, I found out my health insurance doesn't pay for preventative visits. I received an \$800.00 bill for the blood work. I can't pay for my bill because I'm a student and don't have any cash flow at this current time. I can't believe the Doctor wouldn't give me a heads up to make sure my insurance would cover work that wasn't necessary and was strictly preventative. The office can't do anything to help me cover the bill. In addition, the office staff said the onus is on me to make sure my insurance covers visits. Frustrating situation! [93, 1655, 16, 288, 4, 163, 20, 5, 2816, 48, 136, 20, 155, 16, 1416, 1, 644, 129, 2, 28, 42, 1682, 3, 214, 5, 2128, 18, 1185, 100, 2928, 14, 486, 2, 57, 44, 151, 240, 32, 54, 3, 24, 4, 76, 4, 43, 59, 1266, 56, 14, 486, 12, 3795, 2, 180, 1486, 123, 169, 71, 1476, 11, 116, 4, 1208, 939, 1528, 2, 2201, 215, 62, 844, 4, 32, 13, 3, 43, 5, 5005, 2, 62, 635, 3, 6, 5, 2588, 72, 3, 141, 229, 3, 79, 1, 5005, 282, 330, 3, 266, 36, 14, 1654, 1476, 398, 278, 9, 1528, 3, 579, 59, 4613, 591, 486, 9, 1, 2201, 215, 3, 208, 278, 9, 14, 486, 72, 93, 5, 2588, 2, 78, 23, 134, 902, 4239, 25, 16, 2618, 46, 3, 208, 566, 1, 1416, 450, 163, 32, 5, 2816, 48, 4, 123, 169, 14, 1476, 53, 1208, 215, 13, 168, 2597, 2, 6, 5422, 1, 644, 208, 82, 250, 4, 426, 32, 1208, 1, 486, 10, 1315, 1, 644, 129, 106, 1, 11, 21, 32, 4, 123, 169, 14, 1476, 4121, 1528, 2945, 1143]

Затем следует ограничить длину отзыва, чтобы все отзывы были одинаковой длины. Длина отзывов указывалась в программе выше.

Подготовительные мероприятия завершены. Можно перейти к созданию нейронной сети, как показано на листинге 23. Нейронная сеть будет последовательной, состоящей из трёх слоёв. Первый слой Embedding, второй слой LSTM и выходной слой — сигмоид из одного нейрона.

## Листинг 23 – Создание нейронной сети LSTM

```
model = Sequential()  
model.add(Embedding(num_words, 128, input_length=max_review_  
n))  
model.add(LSTM(128))
```

```
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

Таким образом, создаётся последовательная модель, которая затем компилируется оптимизатором `rmsprop`. Функция ошибки - бинарная перекрестная энтропия.

Для того, чтобы снизить переобучение, вызывается функция `callback`, как представлено на листинге 24. Эта функция оценивает нейронную сеть на каждой более успешной эпохе обучения и сохраняет наиболее успешное состояние по указанному пути.

Листинг 24 – Вызов функции `callback`

```
model_save_path = 'Best_NN.h5'
checkpoint_callback = ModelCheckpoint(model_save_path,
                                     monitor='val_accuracy',
                                     save_best_only=True,
                                     verbose=1)
history = model.fit(x_train,
                  y_train,
                  epochs=8,
                  batch_size=64,
                  validation_split=0.1,
                  callbacks=[checkpoint_callback])
```

Запускаем обучение нейронной сети на 8 эпох. Так как сеть обладает сложной архитектурой, обучение происходит достаточно долго — около 15 минут.

К концу обучения точность работы на обучающем наборе составляет 92%, на тренировочном наборе — 96%.

Обучение нейронной сети архитектуры LSTM представлено на рисунке 26.

```

Epoch 1/8
7875/7875 [=====] - 113s 13ms/step - loss: 0.1971 - accuracy: 0.9221 - val_loss: 0.1241 - val_accuracy: 0.9530

Epoch 0001: val_accuracy improved from -inf to 0.95302, saving model to Best_NN.h5
Epoch 2/8
7875/7875 [=====] - 103s 13ms/step - loss: 0.1429 - accuracy: 0.9451 - val_loss: 0.1137 - val_accuracy: 0.9590

Epoch 0002: val_accuracy improved from 0.95302 to 0.95898, saving model to Best_NN.h5
Epoch 3/8
7875/7875 [=====] - 104s 13ms/step - loss: 0.1282 - accuracy: 0.9516 - val_loss: 0.1017 - val_accuracy: 0.9624

Epoch 0003: val_accuracy improved from 0.95898 to 0.96243, saving model to Best_NN.h5
Epoch 4/8
7875/7875 [=====] - 102s 13ms/step - loss: 0.1194 - accuracy: 0.9553 - val_loss: 0.1024 - val_accuracy: 0.9627

Epoch 0004: val_accuracy improved from 0.96243 to 0.96271, saving model to Best_NN.h5
Epoch 5/8
7875/7875 [=====] - 102s 13ms/step - loss: 0.1116 - accuracy: 0.9585 - val_loss: 0.1011 - val_accuracy: 0.9626

Epoch 0005: val_accuracy did not improve from 0.96271
Epoch 6/8
7875/7875 [=====] - 101s 13ms/step - loss: 0.1049 - accuracy: 0.9615 - val_loss: 0.1031 - val_accuracy: 0.9616

Epoch 0006: val_accuracy did not improve from 0.96271
Epoch 7/8
7875/7875 [=====] - 102s 13ms/step - loss: 0.0980 - accuracy: 0.9643 - val_loss: 0.1064 - val_accuracy: 0.9607

Epoch 0007: val_accuracy did not improve from 0.96271
Epoch 8/8
7875/7875 [=====] - 100s 13ms/step - loss: 0.0920 - accuracy: 0.9670 - val_loss: 0.1075 - val_accuracy: 0.9617

Epoch 0008: val_accuracy did not improve from 0.96271

```

Рисунок 26 – Обучение нейронной сети LSTM в 8 эпох

Как можно наблюдать, качество обучения на обучающем наборе, достигнув максимума 96.271% на четвертой эпохе, далее незначительно падает. Это говорит о том, что происходит переобучение нейронной сети. Однако благодаря функции callback была сохранена наиболее продуктивная версия нейронной сети.

Нейронная сеть обучена и можно оценить результаты её работы с помощью тренировочного набора.

```

Листинг 25 – Оценка качества работы нейронной сети LSTM
model.evaluate(x_test, y_test, verbose=1)
1188/1188 [=====] - 6s 5ms/step -
loss: 0.1311 - accuracy: 0.9508
[0.13109911978244781, 0.9507631659507751]

```

Созданная нами нейронная сеть показала на тренировочном наборе достойный результат 95.08%.

Для наилучшей иллюстрации работы нейронной сети проведём анализ на псевдоотзыве, написанном нами самостоятельно, как можно увидеть на листинге 26.

Листинг 26 – Тестирование нейронной сети с помощью отзыва, написанного самостоятельно.

```
text = '''The restaurant is very nice. The waiter was friendly
and other staff was helpful. The food is delicious. I will come again'''
```

Вызванный токенизатор преобразует отзыв в числовой массив, как представлено на листинге 27.

Листинг 27 – Преобразование введенного отзыва в числовой массив

```
sequence = tokenizer.texts_to_sequences([text])
sequence
```

```
[[1,
 104,
 11,
 42,
 92,
 1,
 382,
 6,
 161,
 2,
 80,
 129,
 6,
 549,
 1,
 29,
 11,
 187,
 3,
 66,
 124,
 108]]
```

Далее, на листинге 28 массив будет приведён к длине, заданной в начале программы — 100 элементов. Так как отзыв короче чем 100 элементов, он дополняется нулями.

Листинг 28 – Приведение введенного отзыва к заданной в программе длине

```
data = pad_sequences(sequence, maxlen=max_review_len)
```

data

На листинге 29 применяется команда `predict` для предсказания тональности отзыва

Листинг 29 – Предсказание тональности отзыва

```
result = model.predict(data)
result
array([[0.99932027]], dtype=float32)
if result[[0]] < 0.5:
    print('Отзыв отрицательный')
else:
    print('Отзыв положительный')
```

Отзыв положительный

В качестве ответа программа возвращает «отзыв положительный», это верный ответ.

С полным кодом созданной нейронной сети можно ознакомиться по ссылке <https://colab.research.google.com/drive/1ljTC627A7EOpUxKOlhF4BP-sHZAnKCp30?usp=sharing>

## ЗАКЛЮЧЕНИЕ

В ходе выпускной квалификационной работы были проанализированы классические и современные исследования и разработки в области аффективных вычислений и автоматического сентимент-анализа.

Были изучены подходы российских и зарубежных учёных: Блинова П.Д., Стаценко А., Леонтьева, Фоминых, Ли, Ортони, Клора, Коллинза, Пикард, Миколова, Хоанга и др.

Нами были рассмотрены и проанализированы формальные модели эмоций, метод поиска по ключевым словам, метод с использованием графов знаний, метод полного лингвистического анализа, метод аспектно-эмоционального анализа текста и методы, основанные на нейронных сетях.

Поскольку наиболее высокие результаты показали подходы, основанные на нейронных сетях, наша разработка также сфокусирована на них.

Для анализа был выбран набор данных отзывов пользователей сайта  `Yelp.com`  на английском языке. На указанном сайте пользователи размещают отзывы о различных компаниях в сфере услуг (клининг, ремонт, ландшафтные работы, медицина, общепит и др.), а также дают оценку в звёздах от 1 до 5.

В ходе исследования было создано три нейронные сети с различной архитектурой, показавшие результаты 86%, 88% и 95%.

Наилучшие результаты показала созданная нами рекуррентная нейронная сеть архитектуры LSTM, достигнув точности 95% на тренировочной выборке, что соответствует лучшим показателям индустрии на задачах сентимент-анализа.

Наша научно-исследовательская работа может быть продолжена и усовершенствована. Нам представляется перспективным использование для

сентимент-анализа предложенной нами формальной модели эмоций вместо простой бинарной классификации, что позволит проанализировать эмоции речи и передать информацию о них наиболее полно.



## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Анализ тональности текста. — <https://ru.wikipedia.org/wiki/> (дата обращения: 19.12.2019)
2. Блинов П.Д. Метод, алгоритмы и программная система аспектно-эмоционального анализа: Автореф. дис. канд. техн. наук. — М., 2016. — 19 с.
3. Большакова, Е.И. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика / Е.И. Большакова, Э.С. Клышинский, Д.Э. Ланде, А.А. Носков, О.В. Пескова. Е.В. Ягунова — М.: СССР-США СП Параграф, 1990. — 160 с.
4. Сарбасова А. Н. Исследование методов sentiment-анализа русскоязычных текстов // Молодой ученый. — 2015. — №8. — С. 143-146.
5. Стаценко А.С., Эмоционально-оценочная лексика как средство реализации речевой интенции: Монография / Стаценко А.С. — М. : Прометей, 2011. - 118 с.
6. Гратч Дж., Марселла С., Независимая от предметной области структура для моделирование эмоций // Исследование когнитивных систем, вып. 5. С. 269-306, 2004
7. Клековкина М. В., Котельников Е. В., Метод автоматической классификации текстов по тональности, основанный на словаре эмоциональной лексики. — <http://ceur-ws.org/Vol-934/paper15.pdf> [Электронный ресурс]
8. Леонтьев В. О. Формулы эмоций. Одиннадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2008.. — Труды конференции. Т. 1

9. Маккалох Дж., Питтс У. Логические исчисления идей, относящихся к нервной деятельности.// Автоматы. М.: ИЛ, 1956.
10. Прохоров А., Керимов А. Сентимент-анализ и продвижение в социальных медиа. — <http://compress.ru/article.aspx?id=23115> [Электронный ресурс]
11. Пфайфер Р. Символы, закономерности и поведение: к новому пониманию интеллекта, Конф. Искусственный интеллект, С. 1-15, 1996.
12. Райзензейн, Райнер; Бордген, Сандра; Холтбернд, Томас; Мац, Дениз. Доказательства сильной диссоциации между эмоциями и выражениями лица: случай удивления. Журнал личности и социальной психологии. 2006. — 295–315.
13. Уланов А., Обработка текстов на естественном языке. Лекция 10. Анализ мнений. — [https://compscicenter.ru/media/slides/nlp\\_2014\\_spring/2014\\_04\\_28\\_nlp\\_2014\\_spring.pdf](https://compscicenter.ru/media/slides/nlp_2014_spring/2014_04_28_nlp_2014_spring.pdf) [Электронный ресурс]
14. Aleix, and Shichuan Du, Martinez. A model of the perception of facial expressions of emotion by humans: Research overview and perspectives. The Journal of Machine Learning Research. 2012.— 13 (1): 1589–1608.
15. Anis Daneyal, Ultimate Step by Step Guide to Deep Learning Using Python: Artificial Intelligence and Neural Network Concepts Explained in Simple Terms (Ultimate Step by Step Guide to Machine Learning). — Kindle Edition. - 2020
16. Aspect-Based Sentiment Analysis Using BERT. — <https://www.aclweb.org/anthology/W19-6120.pdf> [Электронный ресурс]
17. Bibhuprada Sushree, Priyadarshini B. A Comprehensive Study on Architecture of Neural Networks and Its Prospects in Cognitive Computing, International Journal of Synthetic Emotions (IJSE), . —IGI Global, vol. 11(2), 37-55, July 2020.

18. Breazeal, C. and Aryananda, L. Recognition of affective communicative intent in robot-directed speech. — *Autonomous Robots* 12 1. — 2002. pp. 83–104.
19. Block Ned, *Readings in Philosophy of Psychology, Volume I, Introduction: What Is Functionalism?* — Harvard University Press.— 2013
20. Burns Samuel, *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn and Tensorflow (Step-by-Step Tutorial For Beginners Book 1)* Kindle Edition. — 2019.
21. Clark Davidson Elliott. *The affective reasoner: a process model of emotions in a multi-agent system.* PhD thesis. — Northwestern University.— 1992.
22. Conte, Rosaria, Castelfranchi, Cristiano. *Cognitive and Social Action.* — Univ.College London Press.— 1995.
23. Dechter Rina. *Learning while searching in constraint-satisfaction problems.* — University of California, Computer Science Department, Cognitive Systems Laboratory. — 1986
24. Dellaert, F., Polizin, t., and Waibel, A., *Recognizing Emotion in Speech,* In *Proc. Of ICSLP 1996.* — Philadelphia, PA, pp.1970-1973.— 1996
25. Ekman, P. & Friesen, W. V. *The repertoire of nonverbal behavior: Categories, origins, usage, and coding.* — *Semiotica*, 1969.— 49–98.
26. Gratch J., Marsella S., *A domain-independent framework for modeling emotions,* *J. of Cognitive Systems Research*, 5(4), 269—306, 2004.
27. Jacob Devlin, Ming-Wei Chang, Kenton Lee. *BERT: Pre-training of deep bidirectional transformers for language understanding.* In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers),* pages

4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics, — 2019.

28. Hudlicka E., Guidelines for Developing Computational Models of Emotions, *International Journal of Synthetic Emotions (IJSE)* 2 (1), 26-79, 2011

29. Gerald L. Clore, Andrew Ortony, and Mark A. Foss. The psychological foundations of the affective lexicon. *Journal of Personality and Social Psychology*, 53(4):751–766, 1987.

30. Hoang, Mickel et al. Aspect-Based Sentiment Analysis Using BERT. 2019. — <https://www.aclweb.org/anthology/W19-6120.pdf> [Электронный ресурс]

31. Hu and Liu, A list of positive and negative opinion words or sentiment words for English. — <http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon> [Электронный ресурс]

32. Hudlicka, Eva. To feel or not to feel: The role of affect in human-computer interaction. *International Journal of Human-Computer Studies*, 2003. — 59 (1–2): 1–32.p. 24

33. Hugo Liu, Ted Selker, and Henry Lieberman. Visualizing the affective structure of a text document. — *International Conference for Human-computer Interaction (CHI)*, 2003.— pages 740–741

34. Introduction to Sentiment Analysis, <http://www.lct-master.org/files/MullenSentimentCourseSlides.pdf> [Электронный ресурс]

35. Kowalczyk and Czubenko. Dedykowany Dla Inteligentnych Systemów Decyzyjnych, in Polish (Computational model of emotions dedicated to intelligent decision systems). *Pomiary Automatyka Robotyka* 17, 2013.— 60–65.

36. Kowalczyk Z, Czubenko M, Computational approaches to modeling artificial emotion—an overview of the proposed solutions *Frontiers in Robotics and AI* 3, 21 2016.
37. Lakshmi Kaushik, John H. L. Hansen. Automatic Audio Sentiment Extraction Using Keyword Spotting. — Conference Paper, September 2015.
38. Learning Sentiment and Semantic Relatedness in User Generated Content Using Neural Models by Henry Michel Nassif, . — MASSACHUSETTS INSTITUTE OF TECHNOLOGY, June 2016.
39. Lee, C.M.; Narayanan, S.; Pieraccini, R., Recognition of Negative Emotion in the Human Speech Signals, Workshop on Auto. Speech Recognition and Understanding, Dec 2001.
40. Mostafa Al Masum Shaikh, Helmut Prendinger, and Mitsuru Ishizuka. Sentiment assessment of text by analyzing linguistic features and contextual valence assignment. . — *Applied Artificial Intelligence*, 22(6):558–601, 2008.
41. Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. — 2013.
42. Moshfeghi, Yashar. Role of emotion in information retrieval, 2012 — PhD thesis // <http://theses.gla.ac.uk/3118/>. [Электронный ресурс]
43. Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Semeval-2014 task 4: Aspect based sentiment analysis. . — Proceedings of the 8th international workshop on semantic evaluation, SemEval 2014, pages 27–35.
44. Marsella, S., Gratch, J. Modeling coping behaviors in virtual humans: Don't worry, be happy. . — Paper presented at the second international joint conference on autonomous agents and multi-agent systems, Melbourne, Australia, 2003.

45. Mascarenhas, S., Dias, J., Prada, R., Paiva, A. A dimensional model for cultural behavior in virtual agents. . —Applied Artificial Intelligence, 24, 2010 — 552–574.
46. Neiberg, D; Elenius, K; Laskowski, K. Emotion recognition in spontaneous speech using GMMs. . — Proceedings of Interspeech, 2006.
47. Ortony, A.; Clore, G. L.; and Collins, A. The Cognitive Structure of Emotions. Cambridge. — UK: Cambridge University Press, 1988.
48. Osgood Charles, May William, Miron Murray. Cross-Cultural Universals of Affective Meaning. — Univ. of Illinois Press, 1975.
49. Picard, Rosalind. Affective Computing. — MIT Technical Report #321 (Abstract), 1995.
50. Picard, Rosalind. Affective Computing.. — Cambridge, MA: MIT Press. p.1., 1997
51. Pynadath David V., Marsella Stacy C., PsychSim: Modeling Theory of Mind with Decision-Theoretic Agents. — Proc. International Joint Conference Artificial Intelligence, pp. 1181-1186, 2005.
52. Rana Fathalla, Emotional Models: Types and Applications. — International Journal of Synthetic Emotions (IJSE), IGI Global, vol. 11(2), 2020.— pages 1-18, July.
53. Reisenzein, Rainer and Hudlicka, Eva and Dastani, Mehdi and Gratch, Jonathan and Hindriks, Koen and Lorini, Emiliano and Meyer, John-Jules Computational Modeling of Emotion: Towards Improving the Inter- and Intradisciplinary Exchange. . —IEEE Transactions on Affective Computing, vol. 4 (n° 3), 2013,— pp. 246-266.

54. Roy, D.; Pentland, A. Automatic spoken affect classification and analysis. —Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, 1996. — 10-01.— pp. 363–367.
55. Salmeron, J. L. Fuzzy cognitive maps for artificial emotions forecasting. — Appl. Soft Comput., 2012.— 12,— 3704–3710.
56. Scherer R. Klaus, Banziger Tanja, Roesch Etienne, A Blueprint for Affective Computing, a sourcebook and manual. — Oxford Univ. Press, 2010.
57. Scherer, K. R., Banziger, T. On the use of actor portrayals in research on emotional expression. In K. R. Scherer, T. Banziger, E. Roesch (Eds.), Blueprint for affective computing: A sourcebook. —Oxford, England: Oxford University Press, 2010.
58. Schmidhuber, J. Deep Learning in Neural Networks: An Overview. — Neural Networks, 2015 — pp. 85–117.
59. Sloman A., Beyond shallow models of emotion. — Cognitive Processing, 2001 — 2(1), 177—198.
60. Steidl, Stefan. FAU Aibo Emotion Corpus. —Pattern Recognition Lab, 2011.
61. Steunebrink, Bas, Dastani, Mehdi, Ch, John-Jules, Meyer. The OCC model revisited. —Department of Information and Computing Sciences, Utrecht University, The Netherlands, 2009.
62. Strongman K.T., The Psychology of Emotion: From Everyday Life to Theory, fifth ed. Wiley, 2003.
63. Tao, Jianhua; Tieniu Tan. Affective Computing: A Review. Affective Computing and Intelligent Interaction, 2005. — pp. 981–995.

64. Thayer, R. E. *The Biopsychology of Mood and Arousal*. —New York, NY: Oxford University Press, 1989.
65. *The Oxford English Corpus: Facts about the language*.— Oxford University Press., 2011.
66. Venkata Reddy Konasani, Shailendra Kadre, *Machine Learning and Deep Learning Using Python and TensorFlow 1st Edition*. — McGraw Hill, 2021.
67. Zajonc et al., Damasio, Zajonc, R. B., Murphy, S. T., and Inglehart, M.. *Feeling and facial efference: implications of the vascular theory of emotion*. *Psychol. Rev.* 96, 1994.— pp. 395–416.