Санкт-Петербургский государственный университет

МАКЕЕВ Кирилл Владимирович

Выпускная квалификационная работа

Анализ эмоциональной окраски отзывов о сфере обслуживания

Уровень образования: магистратура

Направление 45.04.02 «Лингвистика»

Основная образовательная программа ВМ.5805. «Компьютерная и прикладная лингвистика»

Научный руководитель:
доцент, Кафедра математической лингвистики,
Хохлова Мария Владимировна
Рецензент:
доцент, ФГАОУВО «Южно-Уральский государственный университет (национальный исследовательский университет»,
Бабина Ольга Ивановна

Санкт-Петербург
2024

Содержание

Введение

С развитием интернета и использованием социальных сетей объем текстовой информации, доступной для анализа, резко возрос. Применение методов автоматической обработки текстов (NLP, Natural Language Processing) стало важным инструментом в различных областях, таких как маркетинг, анализ отзывов, рекомендательные системы и другие.

Одной из наиболее важных задач в области NLP является определение эмоциональной окраски. Авторское отношение к обсуждаемым событиям, продуктам или услугам, выраженное через эмоциональную окраску текста, играет ключевую роль в успешном управлении бизнесом и принятии решений. Анализ эмоциональной окраски текстовых данных позволяет автоматически выявлять отзывы, комментарии или сообщения с положительной, негативной или нейтральной эмоциональной окраской, что представляет собой важный инструмент для оценки информации и принятия управленческих решений в различных сферах деятельности.

Актуальность работы связана с растущей потребностью компаний и организаций в понимании эмоциональной реакции потребителей на предлагаемые продукты и услуги.

Научная новизна данного исследования заключается в том, что на сегодняшний день наблюдается ограниченное количество научных работ, посвященных анализу эмоциональной окраски отзывов о сфере обслуживания. В данной работе описывается разработка и применение нейросетевого подхода к анализу тональности. В работе проводится сравнительный анализ различных методов анализа тональности, что позволяет выявить их преимущества и недостатки в контексте анализа тональности отзывов о сфере обслуживания.

Гипотеза исследования предполагает, что разработка и обучение нейронной сети, специализированной на классификации, позволит создать

эффективное средство для автоматического выявления эмоциональной окраски отзывов о сфере обслуживания.

Объектом исследования являются отзывы о сфере обслуживания, а **предметом исследования** - их эмоциональная окраска.

Целью работы является разработка и обучение нейросетевого классификатора для анализа эмоциональной окраски отзывов о сфере обслуживания на материале отзывов, размещенных на платформе «2ГИС».

Для достижения поставленной цели предполагается решение следующих **задач**:

1. Сбор и подготовка данных для обучения нейросетевого классификатора.
2. Разработка архитектуры нейронной сети для анализа эмоциональной окраски отзывов.
3. Обучение нейросетевого классификатора на полученных данных.
4. Оценка точности и эффективности созданного классификатора.

Материалом исследования являются отзывы о ресторанах, кафе и продовольственных магазинах, размещенные на платформе «2ГИС».

Практическая значимость работы заключается в возможности использования разработанного нейросетевого классификатора для автоматического анализа эмоциональной окраски отзывов о сфере обслуживания.

Теоретическая значимость работы заключается в расширении применимости методов глубокого обучения, в частности нейронных сетей, для анализа текстовой информации применительно к материалу, отличающемуся своей разнородностью (в контексте эмоциональной окраски отзывов о сфере обслуживания). Полученные результаты могут быть полезны для оптимизации качества обслуживания, что поможет улучшить удовлетворенность клиентов и принимать более обоснованные управленческие решения.

Выпускная квалификационная работа состоит из содержания, введения, двух глав, заключения и списка используемой литературы.

В первой главе рассматривается теоретический аспект анализа тональности отзывов о сфере обслуживания. Описывается суть анализа тональности, его важность в сфере обслуживания и используемые материалы для анализа. Также в главе описываются методы, используемые для решения задачи: лингвистические методы, методы, основанные на машинном обучении и гибридные методы. Каждый из них подробно рассматривается, дается описание принципов работы.

Во второй главе приводится практическая реализация анализа тональности отзывов о сфере обслуживания. Описываются процессы сбора и подготовки данных, которые были выполнены в ходе работы. Подробно характеризуется процесс создания и обучения классификатора на выбранных данных. На основе полученных результатов проводится анализ и оценка эффективности разработанного классификатора.

В заключении подводятся итоги исследования, обсуждаются его результаты и предлагаются возможные направления для дальнейших исследований.

1. Теоретические основы анализа тональности

1.1. Понятие «анализ тональности», сферы применения и материалы

«Анализ тональности текста, или сентимент-анализ (sentiment analysis), — область компьютерной лингвистики и интеллектуального анализа текста, ориентированная на извлечение из него субъективных мнений и эмоций человека» [Двойникова, Карпов 2020]. Анализ тональности текста является важной задачей обработки естественного языка (NLP – Natural Language Processing), а также одним из ключевых инструментов для понимания мнения, выраженного в текстовой форме.

Он используется в различных областях, таких как политическое прогнозирование, анализ социальных медиа, финансовый анализ и др. В общем, он может применяться везде, где есть необходимость анализировать текстовые данные для понимания эмоциональной составляющей или оценки отношения к определенным темам.

В работе [Liu 2012: 7] автор определяет анализ тональности, или извлечение мнений (opinion mining), как «область исследований, которая анализирует мнения, настроения, оценки, отношения и эмоции людей по отношению к таким объектам, как продукты, услуги, организации, люди, проблемы, события, темы, и их атрибуты».

Существует несколько разных названий, применимых к различным задачам анализа тональности, такие как анализ настроений, поиск мнений, анализ субъективности, анализ аффектов, извлечение отзывов, анализ эмоций и другие. В настоящее время все эти задачи объединены под термином «анализ тональности» и, по сути, представляют одну область исследований.

Научные исследования по анализу тональности начали появляться с середины 1990-х. Одной из первых научных исследований на тему тонального

анализа можно считать работу [Wiebe 1994]. Целью данной работы было отслеживание точки зрения персонажей художественных текстов.

Целью другой работы [Hatzivassiloglou, McKeown 1997] было определение положительной или отрицательной семантической направленности прилагательных. В результате работы был создан алгоритм кластеризации, который определяет тональность прилагательных, точность которого составляет более 90%.

В статье [Wiebe, Bruce, O'Hara 1999] демонстрируется процедура автоматического формулирования одной лучшей метки при наличии нескольких несогласных судей. Авторам удалось добиться высокой производительности классификатора, средняя точность которого составляет 81,5 %.

В работе [Hatzivassiloglou, Wiebe 2000] рассматривалась схожая со статьей [Hatzivassiloglou, McKeown 1997] проблема: субъективность прилагательных.

Впервые термин «анализ тональности» появился в исследовании [Nasukawa, Yi 2003]. В своей работе авторы применяли подход, основанный на семантическом анализе, синтаксическом парсере и тональном словаре. В результате, прототип системы достиг точности 75-95% (в зависимости от данных) при анализе тональности веб-страниц и новостных статей.

Термин «извлечение мнений» впервые появился в работе [Dave, Lawrence, Pennock 2003]. Результатом стал метод автоматического различения положительных и отрицательных отзывов. Авторы представили классификатор, который использует методы информационного поиска для извлечения признаков и оценки.

Есть несколько причин, по которым область анализа тональности стала активным направлением исследований начиная с 2000-х гг.:

1. Анализу тональности можно найти применение практически в любой сфере;

2. Анализ тональности влечет за собой множество сложных исследовательских задач, которые не были изучены ранее;
3. Быстрое развитие социальных сетей способствовало увеличению интереса к исследованиям в области анализа тональности, поскольку появилось значительное количество данных, содержащих мнения и отзывы. Без этих данных многие исследования были бы невозможными.

Классификации тональностей

В анализе тональности есть несколько различных методов классификации, которые используются для определения эмоциональной окраски текста. Например:

1. Бинарная классификация.

Это самый простой подход, где тексты классифицируются как позитивные или негативные. Он обычно используется для задач, где интересует только общая эмоциональная оценка. Например, в работах [Turney 2002] и [Pang, Lee, Vaithyanathan 2002].

2. Мультиклассовая классификация.

Здесь тексты могут быть классифицированы как позитивные, негативные или нейтральные. Иногда добавляются и другие классы, такие как "сильно позитивные" или "сильно негативные", чтобы улучшить точность оценки. Например, такой метод использовался в [Pang, Lee 2005] и [Snyder, Barzilay 2007].

3. Классификация на основе эмоций.

Этот метод классификации определяет эмоции, выраженные в тексте, такие как радость, грусть, страх и т.д. Текст может быть классифицирован в соответствии с преобладающей эмоцией или сочетанием нескольких эмоций. Такой метод использовался в работе [Ho et al. 2020].

4. Аспектно-ориентированная классификация.

В этом случае анализируется не только общая эмоциональная окраска текста, но и его эмоциональная окраска относительно определенных аспектов или сущностей. Например, отзыв о ресторане может быть положительным в целом, но негативным по отношению к качеству обслуживания. Аспектно-ориентированная классификация использовалась в работах [Hu, Liu 2004] и [Cataldi et al. 2013].

5. Классификация на основе субъективности и объективности.

Этот метод фокусируется на определении того, является ли текст субъективным или объективным. Текст считается субъективным, если он выражает мнение, чувства, эмоции или оценки автора. Такие тексты могут содержать отзывы, комментарии, рецензии, мнения и т.д. Текст считается объективным, если он представляет факты, нейтральную информацию или описание событий без выражения мнения или эмоций. Объективные тексты могут включать новости, научные статьи, технические отчеты, инструкции и т.д. Такой вид классификации использовался в работах [Su, Markert 2008] и [Pang, Lee 2004].

Уровни, на которых проводится анализ тональности

Уровни использования анализа тональности могут быть различными и зависят от конкретных потребностей и контекста. В работе [Liu 2012: 10-12] автор говорит о трех уровнях:

1. Уровень документа.

«На данном уровне требуется определить, выражает ли весь документ положительное или отрицательное настроение. Эта задача известна как классификация тональности на уровне документов. Предполагается, что каждый документ выражает мнение о конкретном объекте, например, об одном продукте. Поэтому этот уровень анализа не подходит для документов, описывающих или сравнивающих несколько объектов».

2. Уровень предложения.

«Для этого уровня необходимо провести анализ предложений с целью определения того, содержат ли они положительное, отрицательное или нейтральное высказывание. При этом нейтральное высказывание, как правило, означает отсутствие четкой оценки. Этот уровень анализа тесно связан с оценкой субъективности».

3. Уровень сущностей и аспектов.

«На этом уровне проводится аспектный анализ текста, в результате которого определяются основные характеристики объекта и выраженные оценки настроения для каждой из них». Например, если объектом является смартфон, то аспектами могут быть время автономной работы, дизайн, качество камеры и так далее. Таким образом, аспектный анализ настроения представляет собой более детальный подход к анализу отзывов, фокусирующийся на конкретных аспектах или характеристиках объекта.

Оценка качества

Для оценки качества работы модели используются такие метрики, как *Accuracy* (правильность), *Precision* (точность), *Recall* (полнота) и *F-мера*. Для описания этих метрик, необходимо составить матрицу ошибок (confusion matrix) [Михайличенко 2022].

Таблица 1. Матрица ошибок для бинарной классификации

	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Neagtive (FN)	True Negative (TN)

Матрица включает в себя четыре элемента:

1. True Positive.

Это количество объектов, которые модель правильно классифицировала как позитивные.

2. True Negative.

Это количество объектов, которые модель правильно классифицировала как негативные.

3. False Positive.

Это количество объектов, которые модель неправильно классифицировала как позитивные, хотя на самом деле они негативные. Эту ошибку также называют ошибкой 1 рода.

4. False Negative.

Это количество объектов, которые модель неправильно классифицирована как негативные, хотя на самом деле они позитивные. Эту ошибку также называют ошибкой 2 рода.

На основании этой информации, метрики, характеризующие качество модели, вычисляются по следующей схеме:

1. $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$.

Accuracy определяется как отношение числа правильно классифицированных объектов ко всему числу объектов. Это показатель точности прогнозирования по всем классам.

2. $Precision = \frac{TP}{TP+FP}$.

Precision определяет долю правильно предсказанных положительных объектов относительно всех объектов положительного класса. То есть, это мера точности предсказания положительного класса.

3. $Recall = \frac{TP}{TP+FN}$.

Recall оценивает долю правильно обнаруженных положительных объектов относительно всех объектов, принадлежащих положительному классу.

4. $F - \text{мера} = \frac{2 \bullet Precision \bullet Recall}{Precision+Recall}$.

F-мера представляет собой среднее гармоническое значение между *Precision* и *Recall*. *F-мера* учитывает как точность, так и полноту модели.

Области применения

Несколько примеров областей, где может быть использован анализ тональности:

1. Отслеживание общественного мнения

Анализ эмоциональной окраски позволяет компаниям отслеживать отзывы, комментарии и публикации в социальных сетях, блогах и других онлайн-площадках для определения отношения к своим продуктам, услугам или бренду.

2. Клиентский сервис

Компании могут использовать анализ тональности для отслеживания обратной связи клиентов из различных источников, таких как электронная почта, чаты в реальном времени, социальные медиа, обзоры продуктов и т.д. Это помогает быстро выявлять негативные отзывы и проблемы, требующие внимания, и предпринимать соответствующие действия.

Анализ тональности также позволяет оценить общее настроение и удовлетворенность клиентов по отношению к продуктам, услугам или определенным аспектам работы компании. Путем анализа тональности обратной связи клиентов компании могут определить сильные и слабые стороны своего бизнеса и принять меры для улучшения качества обслуживания. Также анализ тональности помогает компаниям следить за репутацией своего бренда.

3. Аналитика рынка

Анализ тональности может использоваться для прогнозирования рыночных тенденций на основе общественного мнения и настроений инвесторов, например, в работе [Ren, Wu, Liu 2018]. Анализ тональности может помочь выявить эмоциональные реакции рынка на новости и события, что помогает инвесторам принимать обоснованные решения.

4. Маркетинговые исследования

С помощью анализа тональности можно оценить реакцию аудитории на рекламные кампании. Анализируя тональность комментариев, можно выявить, как рекламная компания воспринимается ее аудиторией.

Анализ тональности позволяет изучить общественное мнение о конкурентах и их продуктах. Анализируя тональность отзывов и комментариев о продуктах конкурентов, маркетологи могут выявить их сильные и слабые стороны, а также понять, какие аспекты привлекают или раздражают потребителей.

Анализ тональности позволяет выявлять новые тренды и изменения в потребительском поведении. Маркетологи могут использовать эти данные для адаптации своих маркетинговых стратегий и разработки новых продуктов и услуг, отвечающих на изменяющиеся потребности рынка.

5. Политический анализ и прогнозирование

Тональный анализ позволяет отслеживать общественное мнение о политических событиях и фигурах путем анализа новостных статей, социальных медиа, блогов, форумов и других онлайн-источников. Это позволяет аналитикам и политологам оценить, как реагирует общественность на политические инициативы, решения и события.

Анализ тональности может использоваться для прогнозирования результатов выборов. Анализируя общественное мнение о кандидатах и партиях, а также настроения избирателей, политические аналитики могут делать предположения о том, какие кандидаты имеют больше шансов на победу.

Анализ тональности позволяет выявлять политические тренды и изменения в общественном мнении на основе анализа больших объемов текстовой информации. Это помогает аналитикам и политологам понять, какие темы и проблемы наиболее актуальны для избирателей и какие направления политики наиболее перспективны.

Анализ тональности также может быть использован и в других сферах деятельности, таких как здравоохранение, образование, туризм, экология и другие. Применение анализа тональности ограничивается лишь доступностью текстовых данных и потребностью в понимании общественного мнения.

Что может выступать в роли материала для анализа эмоциональной окраски:

- отзывы о товарах и услугах и различных заведениях;
- комментарии и публикации в социальных сетях;
- новостные тексты, в которых могут быть выражены взгляды на происходящие события;
- публикации в различных форумах и блогах;
- научные статьи и др. текстовые источники.

1.2. Методы для проведения анализа тональности

На рисунке 1 представлены существующие подходы к проведению анализа тональности. Рассмотрим рисунок подробнее.

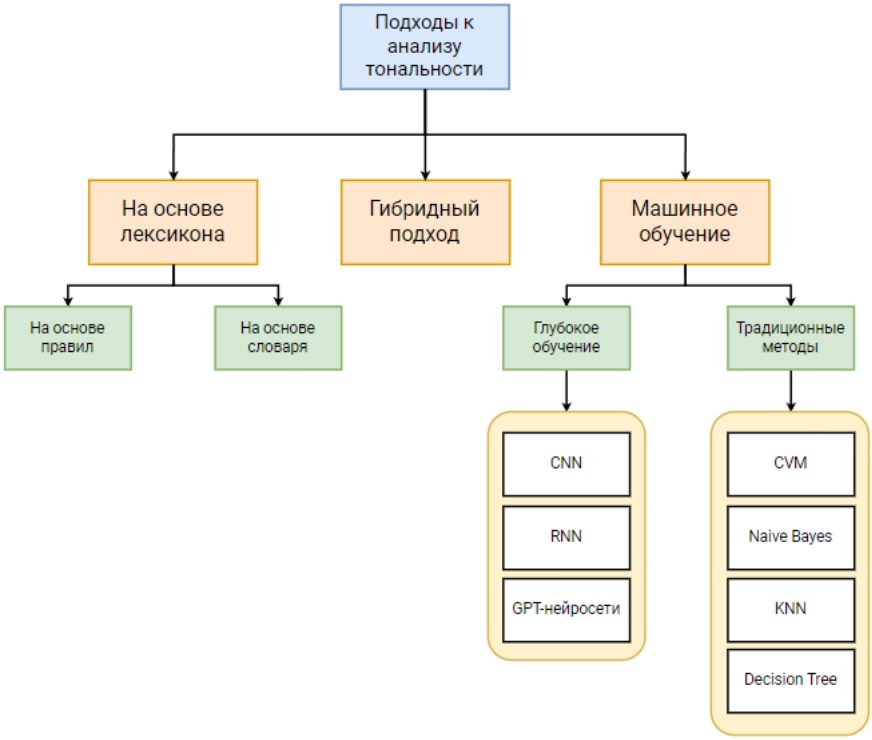


Рисунок 1. Подходы к анализу тональности

1.2.1. Лингвистические методы

1. Словарный подход.

Словарный подход к анализу тональности предполагает использование заранее определенных списков слов, известных как тональные словари. В тональных словарях слова относятся к положительному, отрицательному и нейтральному значениям настроения.

[Kochergina 2015] выделяет следующие методы создания тонального словаря:

1. Ручной подход (manual approach).

Словарь составляется вручную лингвистом-экспертом. Данный подход характеризуется высоким качеством подбора материала.

2. Подход на основе словаря (dictionary-based approach).

Для создания тональных словарей допускается использовать синонимы и антонимы из других словарей: вручную составляется небольшой список эмоционально окрашенных слов, далее алгоритм автоматически расширяет список с помощью синонимов и антонимов, используя для этого уже существующие словари (например, WordNet, или другие онлайн-словари). Найденные слова добавляются в список эмоционально окрашенных слов. Процесс продолжается до тех пор, пока алгоритм не сможет найти новых слов.

Такой подход был использован в нескольких работах. Например, в статье [Hu, Liu 2004], где после завершения алгоритма была использована ручная проверка для исправления ошибок. В работе [Valitutti, Strapparava, Stock 2004] также был использован данный метод.

В статье [Kim, Hovy 2004] использовался вероятностный метод, чтобы удалить ошибки и присвоить словам веса тональности.

В работе [Mohammad, Dunne, Dorr 2009] авторы дополнительно использовали в своем исследовании аффиксы, чтобы увеличить объем словаря (например, honest-dishonest).

Главное преимущество данного подхода заключается в простоте и высокой скорости нахождения эмоционально окрашенных слов. В итоговом тональном словаре могут быть ошибки, для их исправления можно провести ручную проверку.

Недостатком подхода является то, что ручная проверка – это очень длительный процесс. Также слова, полученные из словаря, не зависят от контекста и сферы деятельности, в которой используется слово.

3. Подход на основе корпуса (corpus-based approach).

Основан на корпусе текстов или текстовых коллекциях. Подход позволяет анализировать значение слова в контексте. Есть возможность конкретизировать область поиска.

Существует два сценария использования данного подхода [Liu 2012: 95]. Первый предполагает обнаружение эмоционально окрашенных слов и определение их тональности в корпусе, имеющем конкретную тематику, используя список эмоционально окрашенных слов общего назначения. Второй сценарий предполагает адаптацию списка эмоционально окрашенных слов общего назначения к конкретной области с использованием корпуса, соответствующего этой области.

Однако, создание тонального словаря для конкретной области является непростой задачей, поскольку в одной и той же области слово может иметь разную тональность в зависимости от контекста.

У каждого подхода есть свои достоинства и недостатки. Ручной подход является длительным и трудоемким процессом, а автоматизированные методы могут допускать ошибки. Поэтому подходы к созданию тонального словаря могут комбинировать и использовать совместно.

Хотя словарный подход к анализу тональности легко реализуем, он имеет ряд недостатков:

- Ограниченное понимание контекста. Словарный подход не учитывает контекст, в котором используются слова, что может привести к неправильной интерпретации тональности текста. Некоторые слова могут быть классифицированы как положительные, хотя на самом деле могут выражать отрицательное настроение.
- Некорректная классификация отрицаний и интенсификаторов. Отрицания («не», «нет») и интенсификаторы («очень», «чрезвычайно» и др.) могут существенно изменить настроение предложения, так как словарный подход не всегда может справиться с их корректной классификацией.
- Словарь может охватывать не все области и не учитывать специфические нюансы, например, профессиональная лексика

разных сфер деятельности, что может привести к неточностям и ошибкам в анализе тональности конкретных тем.

- В тексте может встречаться сарказм, который затруднительно обнаружить автоматически. Некорректная классификация саркастически окрашенных слов может исказить результаты работы алгоритма анализа тональности.

Для русского языка доступны следующие тональные словари [Двойникова, Карпов 2020: 23-24]:

- RuSentiLex. «Включает в себя слова и словосочетания, для которых указаны часть речи или синтаксический тип группы, лемматизированная форма, тональность и источник информации. Кроме того, авторы ввели отдельный класс тональности, который обозначает смешанную оценку слова, так как некоторые слова могут иметь различную тональность в зависимости от контекста».
- LinisCrowd. «Словарь строится на пользовательском интернет-контенте, сфокусированном на социально-политической тематике. Словарь был создан на основе размеченных текстов, полученных из социальной сети Facebook» (принадлежат корпорации Meta, которая признана в РФ экстремистской).
- WordNetAffect. «Лексический ресурс, в котором содержатся слова, отражающие эмоции. Словарь был разработан на основе онтологии WordNet. Для русского языка были вручную переведены синсеты с английского языка, используемого в WordNetAffect».

Также для русского языка существуют эмоционально окрашенные корпусы текстов: RuTweetCorp, РОМИП 2012, RuSentiment, SentiRuEval-2015, SentiRuEval-2016, Auto_reviews.

2. Подход на основе правил

«Подход к анализу тональности, основанный на правилах, опирается на заранее определенные правила для определения настроения текста. Кроме этого, в подходе, основанном на правилах, также используются тональные словари» [Двойникова, Карпов 2020: 23-24]. В таких алгоритмах система автоматически маркирует входные данные на основе набора правил, чтобы определить полярность настроения. Для выполнения правил используются следующие методы NLP:

- Стемминг. Стемминг — это процесс сокращения производных (или иногда производных) слов до их словесной основы, базовой или корневой формы.
- Частеречная разметка. Представляет собой этап автоматизированной обработки текста, который направлен на определение частей речи и грамматических особенностей слов в тексте.
- Синтаксический анализ. Это метод NLP, при котором последовательность слов или токенов на естественном или формальном языке сопоставляется с грамматикой этого языка.
- Токенизация. Это процесс разделения текста на отдельные элементы. Токены могут представлять собой слова, символы, фразы или другие текстовые компоненты.

Как обычно работает анализ тональности на основе правил:

1. Написание правил. Лингвистические правила формулируются на основе языковых шаблонов и конструкций, которые могут указывать на тональность. Эти правила могут быть как простыми (например, списки ключевых слов), так и более сложными, включающие в себя, например, синтаксический и семантический анализ.
2. Создание тонального словаря. Создается словарь слов/словосочетаний, в котором каждому слову присваивается значение полярности.

3. Анализ текста. Автоматический анализ текста производится в соответствии с заданными правилами. Настроение всего текста определяется на основе оценок настроений его составных частей. Объединение может быть как простым (подсчет количества положительных и отрицательных слов), так и сложным (учет семантических связей и синтаксических структур в тексте.

Достоинства подхода:

- Системы, основанные на правилах, обычно являются легко интерпретируемыми, так как правила определены в явном виде. По этой причине, можно легче понять, как проводится анализ тональности и почему была сделана та или иная классификация.
- Систему, на основе правил, можно адаптировать к разным областям, добавляя или изменяя правила и словарь.
- Подход, основанный на правилах, не нуждается в больших вычислительных ресурсах и наборах данных.

Однако, у этого подхода имеется ряд недостатков.

- Также как и в словарном подходе, составление правил производится экспертом вручную, поэтому является трудоемким и долгим процессом.
- Чтобы обеспечить точность и релевантность системы анализа тональности, необходимо постоянное обновление правил и словаря.
- При классификации текстов других тематик необходимо переписать правила и словарь в соответствии с данной темой, иначе в результате классификации могут возникнуть ошибки и неточности.
- Также к недостаткам можно отнести трудности с обработкой неоднозначностей, выявления сарказма, отрицаний и интенсификаторов.

В целом, подход к анализу тональности на основе правил требует тщательной разработки правил и словаря, и не всегда может достичь такого же уровня производительности, как подходы, которые основаны на машинном обучении.

1.2.2. Подход, основанный на машинном обучении

Основная идея этого метода: на заранее размеченных данных осуществляется обучение классификатора, который затем применяется для классификации новых текстов.

1. Традиционные методы машинного обучения

Зачастую задача анализа эмоциональной окраски отзывов состоит в отнесении отзывов к двум классам: положительному и отрицательному. Отзывы используются для обучения и тестирования модели, так как эти отзывы часто содержат оценки, например, от 1 до 5 звезд, где отзыв с 4 или 5 звездами является положительным, а отзыв с 1 или 2 звездами является отрицательным. Однако «возможно использование и нейтрального класса, например, можно присвоить всем отзывам с оценкой 3 звезды статус нейтрального отзыва» [Самигулин, Джурабаев 2021: 56].

Поскольку это задача классификации текста, можно использовать различные текстовые классификаторы, относящиеся к методам машинного обучения. Например, метод опорных векторов (SVM), наивный байесовский классификатор (Naïve Bayes), метод k-ближайших соседей (KNN) и деревья решений (Decision tree).

SVM

Метод опорных векторов заключается в построении гиперплоскости между кластерами точек, которые необходимо классифицировать. Основная задача метода – это найти такую гиперплоскость, которая максимизирует расстояние между классами и однозначно классифицирует точки. Опорные вектора лежат на границах разделяющей гиперплоскости. После обучения

модели, опорные вектора используются для классификации новых данных. На рисунке 2 визуализирован принцип работы SVM модели.

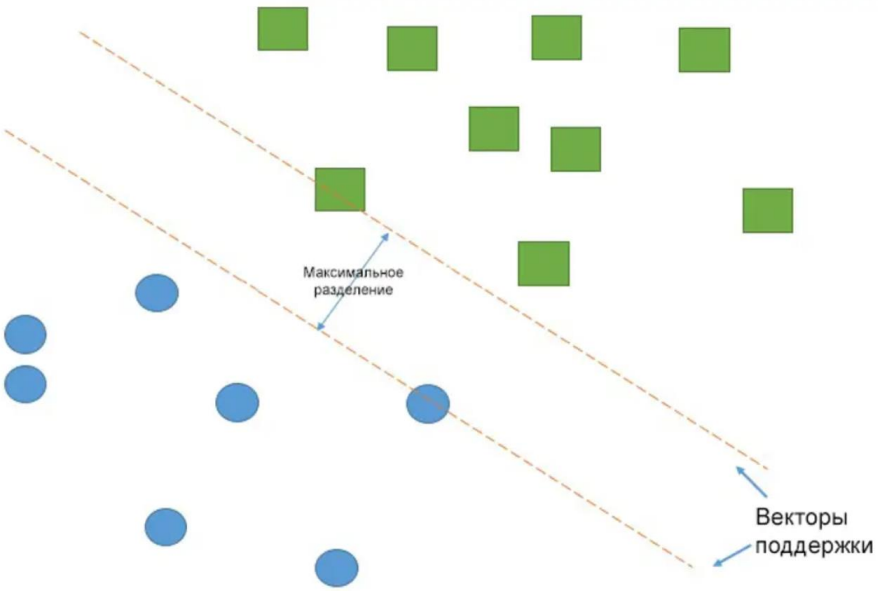


Рисунок 2. Принцип работы SVM

Naïve Bayes

Наивный байесовский классификатор является простым вероятностным классификатором. Он основан на предположении о независимости признаков внутри каждого класса. То есть считается, что наличие конкретного признака в классе не зависит от наличия других признаков в этом классе.

Во время обучения классификатор вычисляет вероятности появления каждого слова в каждом классе (например, положительной и отрицательной тональности). При классификации нового текста классификатор использует теорему Байеса для вычисления вероятности того, к какому классу принадлежит текст. После этого тексту присваивается класс с наибольшей вероятностью.

Теорема Байеса выглядит следующим образом:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Где:

- $P(A|B)$ – вероятность того, что документ B принадлежит классу A.
- $P(B|A)$ – вероятность появления документа B среди документов класса A.
- $P(A)$ – безусловная вероятность класса A.
- $P(B)$ – безусловная вероятность появления документа B.

Текст классифицируется как принадлежащий к классу с наибольшей условной вероятностью $P(A|B)$.

KNN

Метод k-ближайших соседей основан на поиске кратчайшей дистанции между объектом, который необходимо классифицировать, и ближайшим к нему классифицированным объектом из обучающего набора. В результате классификации, объект будет относится к тому классу, к которому принадлежит ближайший объект уже классифицированного набора. На рисунке 3 представлена визуализация принципа работы KNN модели.

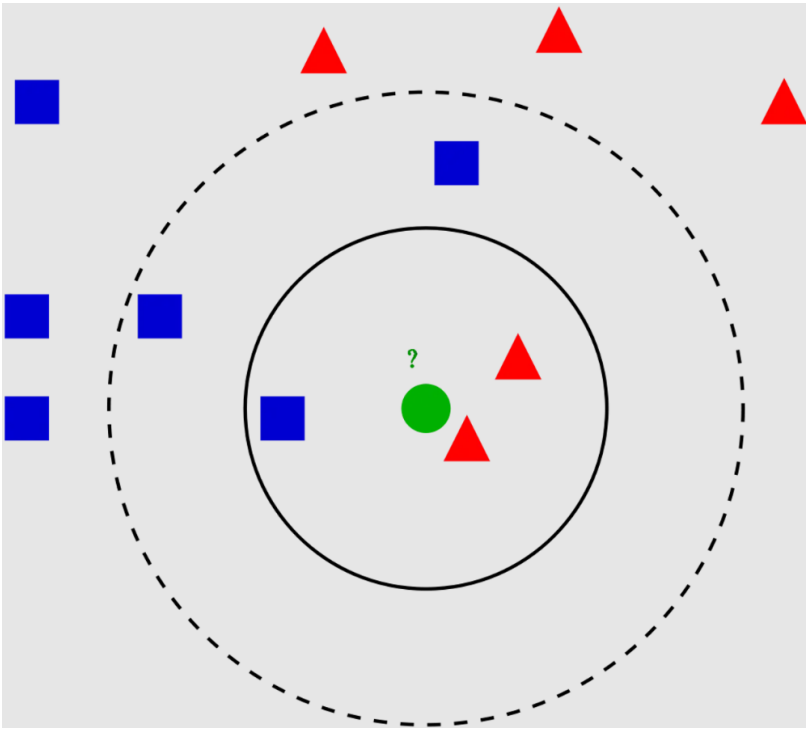


Рисунок 3. Принцип работы KNN

Decision Tree

Классификатор представляется в виде дерева, в котором узлы являются проверкой одного из признаков, каждая ветвь – результат этой проверки, а каждый лист – конечный результат. Дерево решений позволяет понять, принадлежит ли объект к конкретному классу.

Дерево решений создается на основе признаков, характеризующих тексты, таким образом, чтобы максимально разделять данные на классы. Затем дерево обучается на подготовленных данных. Во время обучения алгоритм подстраивается таким образом, чтобы минимизировать ошибку классификации. После завершения обучения алгоритм используется для классификации новых текстов. Для этого текст проходит через каждый узел дерева, начиная с корня, и в зависимости от значений признаков определяется путь, по которому следует движение по дереву, пока не будет достигнут лист, который представляет собой прогноз (например, имеет ли текст положительную или отрицательную тональность).

Если соединить несколько деревьев решений между собой, получится Random forest (случайный лес). Случайный лес – это такой ансамблевый метод (то есть, объединение нескольких алгоритмов машинного обучения), где каждое дерево создается отдельно, с использованием различных подмножеств данных и признаков. Окончательное решение принимается путем голосования или усреднения результатов всех деревьев.

2. Глубокое обучение

В последние годы в задачах анализа тональности все чаще стали применяться различные нейронные сети: CNN, RNN, LSTM, GRU, а также Трансформеры. Это связано со стремительным развитием нейросетей, которое повлекло за собой создание таких библиотек, как TensorFlow и Keras [Семина 2020: 56].

В традиционных методах машинного обучения признаки выявляются и извлекаются вручную или с использованием специальных методов. В моделях глубокого обучения признаки выявляются и извлекаются автоматически.

Автоматическое извлечение признаков позволяет достичь более высокой производительности.

Нейронные сети и глубокое обучение в настоящее время обеспечивают наилучшее решение многих задач в области обработки естественного языка.

Непосредственно перед использованием любых методов машинного обучения необходимо выполнить несколько шагов [Dang, Moreno-García, De la Prieta 2020]:

1. Подготовка данных.

Необходимо собрать набор данных для анализа тональности. Обычно, набор состоит из текстов, имеющих метки настроения.

2. Предварительная обработка данных.

После сбора данных необходимо их предобработать. Предобработка может состоять из таких процессов, как:

- Токенизация (разбить каждое предложение на токены),
- Лемматизация (приведение словоформы к ее лемме),
- Удаление стоп-слов,
- Приведение текста к нижнему регистру,
- Удаление пунктуации.

3. Векторизация текста.

Чтобы передать текст на вход модели, необходимо преобразовать текст в числовой вид. Для этого используются методы векторного представления слов, например, Word2Vec, GloVe, fastText или TF-IDF.

Word2Vec – неглубокая искусственная нейронная сеть, состоящая из двух слоев, которая используется для преобразования текста в векторы слов. Модель преобразует лингвистический контекст в числовые значения, и векторы размещаются в векторном пространстве таким образом, что слова с похожим контекстом будут расположены близко друг к другу. Word2Vec фиксирует синтаксическое и семантическое сходство между словами. Модель

Word2Vec была предложена исследователями из компании Google в статье [Mikolov et al. 2013].

Существуют две архитектуры Word2Vec:

a) CBOW. Эта модель предсказывает текущее слово на основе контекста, то есть окружающих слов.
b) Skip-gram. В этой модели используется текущее слово для предсказания слов, находящихся рядом в предложении.

GloVe

Это алгоритм, который формирует векторы слов, исходя из частоты встречаемости пар слов в больших корпусах текстов. То есть, алгоритм представляет смысловые связи между словами через разницу векторов. Например, если из вектора для слова "муж" вычесть вектор слова "мужчина" и прибавить вектор слова "женщина", результат должен быть близок к вектору для слова "жена".

GloVe пытается уменьшить расхождение между произведением векторов слов и логарифмом их совместной встречаемости. Это позволяет модели выявлять различные типы связей между словами.

GloVe показывает лучшие результаты в задачах распознавания именованных сущностей. Также алгоритм способен улавливать семантику редких слов, и эффективен при использовании небольших корпусов.

Модель была разработана исследователями из Стэнфордского университета и представлена в статье [Pennington, Socher, Manning 2014].

fastText – библиотека, которая включает в себя заранее обученные векторные представления слов и классификатор, способный разделять слова на классы.

Отличие fastText от других алгоритмов эмбеддинга в том, что fastText работает на уровне символов, а не слов. Вместо того, чтобы рассматривать слово как единое целое, fastText разбивает его на несколько n-грамм. Например, для слова «apple» с n = 3, n-граммами могут быть <ap, app, ppl, ple,

le>. Угловые скобки используются для обозначения начала и конца слова. Каждая n-грамма представляется отдельным вектором, который назначается случайно и изменяются в процессе обучения. Затем предсказывается контекст слова на основе его n-грамм. Входом в нейронную сеть являются суммы векторов всех n-грамм, составляющих слово. Выходом нейронной сети является векторное представление слова.

Главное преимущество fastText над другими алгоритмами эмбеддинга заключается в его способности эффективно обрабатывать слова, которые не встречались в модели ранее.

Модель fastText была разработана отделом по исследованию искусственного интеллекта компании Facebook (принадлежат корпорации Meta, которая признана в РФ экстремистской) и опубликована в статье [Bojanowski et al. 2017].

TF-IDF (Term frequency-inverse document frequency) – это числовая статистическая метрика, определяющая значимость слова в рамках документа. Она базируется на методе «мешка слов», однако, не учитывает только частотность появления слов. При использовании TF-IDF часто встречающиеся слова не будут подавлять менее частые, но более важные слова.

Алгоритм включает в себя два ключевых понятиях:

- Частота слова (Term Frequency) определяет, с какой регулярностью конкретное слово встречается в документе. Формула для частоты слова выглядит следующим образом: TF = (количество вхождений слова в документе) / (общее количество слов в документе).
- Обратная частота документа (Inverse Document Frequency) уменьшает значимость часто повторяющихся слов. Формула для обратной частоты документа выглядит следующим образом: IDF = log(общее количество документов / количество документов, содержащих это слово). Для сглаживания воздействия IDF в

окончательном расчёте применяется логарифмирование. IDF измеряет важность слова во всем корпусе.

Вектор получается в результате произведения TF на IDF для каждого слова в каждом документе. Получившийся вектор отражает важность слова относительно конкретного документа и относительно всего корпуса.

В алгоритме векторизации TF-IDF также применима концепция n-грамм. Мы можем объединять слова в группы по два, три, четыре и более, чтобы создать итоговый набор признаков.

Далее подробно рассмотрим каждый метод глубокого обучения.

DNN

Глубокие нейронные сети (DNN – deep neural network) – это искусственные нейронные сети, в которых данные проходят через несколько слоев обработки информации. Каждый слой принимает данные, преобразует их и передает результат следующему слою. Глубокие нейронные сети включают в себя входной слой, скрытые слои, состоящие из нейронов, в которых происходит обработка данных, и выходной слой. Выходной слой может включать в себя один или несколько нейронов.

CNN

Сверточная нейронная сеть (CNN – convolutional neural network) – это архитектура нейронной сети прямого распространения, изначально направленная на решение задач распознавания образов. Для задачи анализа тональности используются одномерные сверточные нейронные сети (1D CNNs), которые, в отличие от обычных CNN, получают на вход не двумерный, а одномерный массив данных.

Сверточная нейронная сеть включает в себя три вида слоев: сверточные слоев, пулинговые слои (или слои субдескретизации) и полносвязные слои. Сверточный слой обычно является первым в нейронной сети такого типа. После сверточного слоя могут идти дополнительные сверточные слои или пулинговые слои, а завершающим слоем часто является полносвязный слой.

Сверточные слои отвечают за извлечение характеристик из входных данных с помощью фильтров свертки (или ядер свертки), которые являются матрицами параметров. Ядра перемещаются по входным данным с определенным шагом, известным как шаг свертки.

Каждое ядро вычисляет скалярное произведение между значениями входных данных и значениями фильтра, а затем суммирует полученные результаты. Это создает новое значение в выходных данных. После этого фильтр смещается на один шаг и повторяет процесс до тех пор, пока фильтр не охватит все входные данные. Конечный результат – это серия произведений входных данных и значений фильтра, которая называется картой признаков (также активацией, либо свернутыми признаками).

Пулинговые слои (операция подвыборки), следующие за слоями свертки, уменьшают размерность данных и улучшает вычислительную эффективность. Среди наиболее часто встречающихся видов пулинга выделяются максимальная подвыборка (max-pooling) и усреднение (average-pooling).

Пулинг разбивает входные данные из сверточного слоя на небольшие регионы. В каждом регионе подвыборки вычисляется предварительное значение, которые объединяются в новую карту признаков. Полученная карта признаков имеет меньшую размерность по сравнению с исходной. Пулинг помогает уменьшить количество параметров в сети, сделать модель менее чувствительной к небольшим изменениям в данных и улучшить вычислительную эффективность.

Перед тем, как передать данные в полносвязный слой, выходные данные из пулинг слоя преобразуются в одномерный вектор. Этот процесс называется разглаживанием (flattening). Разглаживание делается для того, чтобы данные могли быть переданы в полносвязный слой, поскольку полносвязный слой может принимать на вход только одномерные векторы. Процесс

разглаживания устраивает все значения из многомерной структуры данных (матрицы или тензора) в одну линейную последовательность.

После извлечения признаков, данные передаются полносвязным слоям, которые занимаются классификацией.

CNN имеет ряд недостатков:

- - неспособность хорошо моделировать долгосрочные зависимости в последовательных данных;
- - не учитывают порядок слов внутри текста;
- - трудности с выбором оптимальных параметров фильтра свертки.

RNN

Рекуррентные нейронные сети используются для работы с последовательными данными, поэтому этот тип нейросетей часто используется в тех задачах NLP, в которых существенную роль играет зависимость от предыдущих элементов последовательности.

Рекуррентные нейронные сети обрабатывают каждый элемент в последовательности, запоминая при этом внутреннее состояние (скрытое состояние, hidden state). В контексте анализа настроений это означает, что RNN могут учитывать последовательную природу языка, где порядок слов может существенно влиять на общее выражаемое настроение.

Основная идея состоит в том, чтобы запоминать предыдущие состояния входных данных, чтобы учитывать контекст при обработке следующего элемента последовательности.

Существуют 4 типа RNN:

1. Один к одному. Направляет один вход на один выход. По сути, представляет из себя обычную нейронную сеть.
2. Один ко многим. Направляет один вход на несколько выходов. Используется, например, для генерации описания изображений.
3. Многие ко многим. Используются несколько входов для предсказания нескольких выходов. Используются для перевода.

4. Многие к одному. Несколько входов для одного выхода. Именно этот тип RNN используется для задачи анализа тональности.

Рекуррентная нейронная сеть состоит из входного слоя, выходного слоя и нескольких скрытых слоев. Входной слой получает данные для обработки, затем в скрытых слоях происходит анализ этих данных, и выходной слой возвращает результат.

Входные последовательности проходят через скрытые слои, которые работают циклически: скрытые слои способны сохранять предыдущие данные в краткосрочной памяти и использовать их для предсказания следующих последовательностей.

В скрытом состоянии хранится информация о последовательности, обработанной до этого. Скрытый слой обновляется на каждом временном шаге, используя текущий вход и предыдущее скрытое состояние. Эта связь позволяет сети улавливать зависимости во входной последовательности. Новое скрытое состояние становится входом для следующего временного шага. Алгоритм повторяется для каждого элемента в последовательности.

В процессе обучения веса рекуррентных связей настраиваются с использованием алгоритма обратного распространения ошибки через время (Backpropagation through time), чтобы минимизировать функцию потерь. Этот алгоритм представляет собой расширение классического метода обратного распространения ошибки (Backpropagation). Этот процесс позволяет нейросети обучиться осмысленному представлению выходной последовательности. BPTT учитывает временную структуру данных, позволяя градиентам распространяться по последовательности данных.

Хотя традиционные рекуррентные сети используются в задачах анализа настроения, их возможности ограничены, так как с помощью RNN становится трудно улавливать дальние зависимости из-за проблемы исчезающего градиента. Для решения этой задачи разработаны более продвинутые архитектуры рекуррентных нейронных сетей, такие как LSTM (Long Short-

Term Memory) и GRU (Gated Recurrent Unit). Эти архитектуры обеспечивают более рациональное управление долгосрочными зависимостями.

LSTM

Архитектура нейросети основана на алгоритме долгой краткосрочной памяти (long short-term memory). Впервые эта архитектура была опубликована в работе [Bengio, Simard, Frasconi 1994].

LSTM является вариацией RNN. Слой LSTM обеспечивает передачу информации через многие временные интервалы, что способствует решению проблемы улавливания дальних зависимостей, существующей в обычной рекуррентной нейронной сети.

Основная идея архитектуры LSTM заключается в том, что она «способна сохранять информацию для последующего использования, что предотвращает постепенное затухание предыдущих сигналов в процессе обработки» [Шолле 2018: 371].

Принцип работы LSTM:

1. На каждом временном этапе LSTM принимает входные данные и предыдущее скрытое состояния.
2. LSTM имеет три компонента: входной, «забывания» и выходной. Эти компоненты отслеживают, какая информация должна быть сохранена, а какая забыта.
3. LSTM содержит скрытое состояние, которое проходит через время и модифицируется в соответствии с входными данными и состоянием элементов.
4. На каждом временном шаге LSTM генерирует новое скрытое состояние, которое зависит от входных данных предыдущего скрытого состояния.

«Характерной чертой LSTM нейросетей являются ее блоки памяти. Каждый блок состоит из элементов, которые контролируют его состояние и поведение. Элемент «забывания» определяет информацию для удаления из

памяти; входной элемент задает исходные данные для обновления памяти; выходной элемент определяет данные, которое будет передано на выходе блока памяти» [Рапаков и др. 2023: 51].

Благодаря своей архитектуре, LSTM способен эффективно улавливать долгосрочные зависимости в последовательных данных и предотвращать проблему затухания градиента. Поэтому LSTM является распространенным методом в задачах NLP, в том числе и в задачах анализа эмоциональной окраски. Однако, у LSTM есть ряд недостатков:

1. LSTM требует больше вычислительных ресурсов, так как архитектура является более сложной, чем обычные рекуррентные нейронные сети.
2. Если набор данных недостаточно большой, или в наборе данных большое количество шумов, LSTM может столкнуться с проблемой переобучения.
3. Если веса не инициализированы должным образом, либо скорость обучения слишком высока, LSTM может столкнуться с проблемой градиентного взрыва.

GRU

GRU (Gated Recurrent Unit) – это еще один тип рекуррентных сетей. Он был представлен в статье [Cho et al. 2014]. Архитектура была создана с целью упрощения процесса обучения (по сравнению с LSTM), сохраняя при этом эффективность.

Особенностью GRU является то, что ее элементами не являются отдельные нейроны, а комплекс нейронов, называемый модулем. Этот модуль включает в себя фильтры (или вентили), которые определяют, как информация будет использоваться для вычисления выходных значений на текущем слое и значений скрытого слоя на следующем шаге [Новиков, Абдулнагимов, Агеев 2021: 44].

У GRU есть скрытое состояние в виде вектора, который включает в себя сведения о предыдущих элементах последовательности.

- Элемент обновления (Update Gate) проводит анализ входных данных вместе с предыдущим скрытым состоянием и принимает решение, какую информацию нужно сохранить для передачи на следующий шаг.
- Элемент сброса (Reset Gate) решает, какую информацию из предыдущего скрытого состояния следует забыть. Это помогает модели игнорировать несущественную информацию из предыдущего состояния.

Затем, используя результаты элемента обновления и элемента сброса, вычисляется текущее скрытое состояние. Оно учитывает как новую информацию, так и информацию из предыдущего состояния, в зависимости от результатов элементов.

На последнем шаге скрытое состояние передается на выходной слой, который делает итоговое предсказание.

Достоинства архитектуры GRU:

1. GRU демонстрирует сопоставимую с LSTM производительность.
2. GRU является более компактной архитектурой по сравнению с LSTM: параметры GRU включают в себя 6 матриц весов, в то время как LSTM включает в себя 8 матриц. Это делает GRU более эффективным с точки зрения использования ресурсов.
3. GRU демонстрирует преимущество в эффективном использовании памяти, в сравнении с LSTM. Это позволяет расширить окно обучения.
4. По сравнению с LSTM, GRU имеет преимущество в скорости обучения, что обусловлено его более простой структурой.

Недостатки:

1. В GRU нет явной ячейки памяти, как в LSTM, что делает его менее эффективным в сохранении долгосрочных зависимостей.
2. GRU может столкнуться с затуханием градиента, если набор данных слишком велик или веса не инициализированы должным образом.
3. Если набор данных, используемый для обучения, необъективен или нерепрезентативен, GRU может показывать недостаточную способность работы на новых данных.

Трансформеры

В настоящее время архитектура Transformer является ключевым элементом в задачах обработки естественного языка. Она была представлена исследователями компании Google в статье [Vaswani et al. 2017].

Основная суть статьи состоит в том, что с помощью механизма внимания, которое называется «нейронное внимание», можно создавать большие модели последовательностей, не содержащие ни сверточных, ни рекуррентных слоев [Шолле 2018: 371].

Механизм внимания является основной составляющей частью трансформера. Он позволяет модели обращать внимание на различные части входных данных в зависимости от их значимости для выполнения конкретной задачи. Механизм внимания позволяет модели обрабатывать длинные последовательности данных, сохраняя контекст.

Для обработки каждого токена последовательности используется многослойный перцептрон. Он обрабатывает токены независимо друг от друга. Он применяется для обработки входных данных после каждого блока механизма внимания.

Слой нормализации применяется для стабилизации обучения и ускорения сходимости модели. Может применяться перед каждым слоем многослойного перцептрона и между слоями внимания.

Затем, в обработанные данные добавляются параллельные соединения (residual connections) между слоями для облегчения обучения модели и предотвращения проблемы затухания градиента.

Поскольку трансформер не может понимать порядок элементов в последовательности, в модель добавляют позиционное кодирование (positional encoding). Оно добавляется к входным эмбеддингам, чтобы модель могла учитывать позицию в последовательности.

Существует несколько архитектур трансформеров, которые имеют уникальные особенности в подходах к архитектуре, предварительном обучении, механизмах внимания и других составляющих. Существуют такие архитектуры трансформеров, как:

1. BERT

Архитектура BERT была разработана отделом по исследованию искусственного интеллекта компании Google и представлена в статье [Devlin et al. 2018].

BERT – это архитектура трансформера, которая предварительно обучена на обширных текстовых наборах. Этот метод показал выдающиеся результаты в различных задачах обработки естественного языка, включая анализ тональности, например, в [Zhao et al. 2021], благодаря способности к предварительному обучению на масштабных корпусах текста.

Архитектура состоит из многоуровневого стека трансформерных кодировщиков, где каждый кодировщик включает в себя множество слоев внимания и полносвязных слоев. BERT предварительно обучается на двух задачах: прогнозирование случайно скрытых слов (Masked Language Model, MLM) и прогнозирование следующего предложения в тексте (Next Sentence Prediction, NSP). Эти задачи обеспечивают модели более широкое и качественное восприятие текста.

BERT также представляет собой подход к созданию многоязычных моделей, которые могут работать с различными языками, используя общую архитектуру и предварительное обучение.

Одна из ключевых особенностей BERT – это способность понимать контекст слов в обоих направлениях, что позволяет ему улавливать семантические зависимости в тексте.

Недостатки BERT включают в себя контекстуальные ограничения, так как BERT работает с ограниченным контекстом внутри предложения, что может быть недостаточно для полного понимания семантических зависимостей в длинных текстах или в текстах со сложными зависимостями между словами.

2. RoBERTa

Архитектура RoBERTa была опубликована в статье [Liu et al. 2019]. Архитектура разработана исследователями из Вашингтонского университета в Сиэтле, США, совместно с исследователями из отдела по исследованию искусственного интеллекта компании Facebook (принадлежат корпорации Meta, которая признана в РФ экстремистской).

RoBERTa представляет собой улучшенную версию архитектуры BERT, которая была обучена с использованием ряда оптимизаций и дополнительных данных. Основными изменениями стали увеличение размера обучающего набора, дополнительная предварительная обработка текста и обучение без маскирования входных токенов. Эти изменения позволили достичь более высокой производительности по сравнению с оригинальной архитектурой BERT.

RoBERTa стала широко используемой архитектурой в области анализа тональности. Например, в статье [Dai et al. 2021], где авторы сравнивают архитектуру RoBERTa с другими моделями. Эксперименты показали, что предобученная модель на основе RoBERTa превосходит показатели других моделей и может быть использована для аспектного анализа тональности.

Также активно исследуется сочетание модели RoBERTa с другими архитектурами. Например, в статье [Tan, Lee, Lim 2023] RoBERTa используется совместно с GRU архитектурой.

В статье [Tan et al. 2022] RoBERTa используется совместно с LSTM архитектурой.

Достоинствами RoBERTa можно считать высокую производительность и масштабируемость. Однако, модель также является требовательна к ресурсам, так как она имеет сложную архитектуру.

3. XLNet

Архитектура XLNet была разработана исследователями из Университета Карнеги Меллон совместно с исследователями из компании Google и представлена в статье [Yang et al. 2019].

XLNet основан на идее авторегрессивного предварительного обучения, при котором модель предсказывает каждый токен в последовательности на основе всех предшествующих токенов.

Основное отличие XLNet от других трансформеров заключается в использовании маскирования перестановок (permutation masking) вместо маскирования слов, что позволяет модели учитывать контекст в обоих направлениях при предсказании токенов. Это позволяет модели учитывать более широкий контекст и улучшает ее способность к адаптации к различным задачам.

Также модель вводит новый метод обучения под названием «контрастное обучение» (Contrastive Teaching). Концепция контрастного обучения заключается в том, что обучение происходит не только на основе сходства, но и на основе различий. Следовательно, для этого подхода нам нужны как примеры схожести, так и примеры различия.

Таким образом, к достоинствам модели можно отнести учет контекста в обоих направлениях, высокую обобщающую способность. К недостаткам

можно отнести сложность обучения, требовательность к данным и сложность интерпретации результатов.

В некоторых работах XLNet выступает в качестве модели для анализа тональности. Например, в статье [Danyal et al. 2024], где авторы используют XLNet для задачи анализа тональности отзывов на фильмы. Это может быть использовано для выявления общих характеристик фильмов на основе тональности их отзывов.

В исследовании [Habbat, Anoun, Hassouni 2022] авторы используют модель XLNet совместно с GRU и сверточными нейронными сетями для задачи анализа тональности отзывов. В работе используются три набора данных на французском языке: отзывы покупателей Amazon (американская технологическая компания, занимающаяся электронной коммерцией); набор данных AlloCiné (французская онлайн-платформа, посвященная киноиндустрии); а также набор данных Twitter. Точность составила 96,5%, 90,1% и 89,6% соответственно.

4. GPT

GPT (Generative Pre-trained Transformer) – это серия моделей, основанных на архитектуре трансформера и предварительно обученных на больших объемах данных.

GPT впервые была представлена в статье [Radford et al. 2018], опубликованной исследователями из компании OpenAI.

GPT обучается в режиме авторегрессии, предсказывая следующий токен последовательности на основе предыдущих токенов. Модель предварительно обучается на больших корпусах текстов. Это предоставляет модели возможность изучить общие языковые схемы и механизмы, поэтому модель может использоваться для решения разнообразный задач NLP. GPT может создавать тексты, сохраняя семантическую и синтаксическую связанность, и применяется в разных областях, включая анализ тональности, машинный

перевод, аннотирование, автоматический анализ текста для выделения ключевых фраз и т.д.

Преимуществами модели GPT является способность к генерации текста, обладающего высокой связанностью и натуральностью. Модель является универсальной и может быть применена к широкому спектру задач. Несмотря на свои преимущества, модель GPT имеет ограничение на длину контекста, что может привести к ограниченной способности к пониманию длинных и сложных текстов. Обучение GPT требует значительных вычислительных ресурсов.

Имеется несколько моделей GPT, доступных для онлайн использования:

1. ChatGPT – это языковая модель, разработанная OpenAI. Чат был запущен 30 ноября 2022 года. В настоящее время пользователям доступна модель ChatGPT-3.5, а также платная модель ChatGPT-4.
2. Чат Microsoft Bing – является частью поисковой системы Bing компании Microsoft. Система основана на модели GPT-3.5.
3. YaGPT – это генеративная языковая модель, разработанная компанией Яндекс. 28 матра 2024 года была представлена модель третьего поколения YaGPT 3 [«Яндекс» представил… 2024].
4. GigaChat – модель, разработанная командой SberDevices. Модель работает на базе нейросетевой архитектуры GigaNet, которая сочетает в себе принципы работы трансформеров и рекуррентных нейронных сетей [GigaChat API 2024].

Нейронные сети, основанные на GPT, могут использоваться для анализа тональности. Например, в статье [Chumakov, Kovantsev, Surikov 2023] исследуется извлечение триплетов аспектных настроений с помощью GPT моделей.

1.2.3. Гибридный подход

Гибридный подход сочетает в себе как элементы методов на основе словарей и правил, так и элементы методов машинного обучения. Гибридный подход стремится объединить преимущества двух подходов для улучшения качества анализа тональности.

В работе [König, Brill 2006] описано создание гибридного классификатора для задачи определения эмоциональной окраски. В работе применяется подход, который комбинирует тональные словари с методом опорных векторов. Результаты исследования показывают, что использование классификатора, основанного на гибридном подходе, повышает точность в сравнении с классификаторами, основанными только на машинном обучении.

Другой пример гибридного подхода представлен в работе [Lakshmi, Raj, Vikram 2017]. В исследовании авторы совместили метод глубокого обучения CNN с методом k-ближайших соседей. В результате, алгоритм повышает эффективность и точность на больших наборах данных, чем метод, основанный только на глубоком обучении.

Также несколько примеров исследований было приведено выше, при описании методов RoBERTa и XLNet.

1.3. Выводы к Главе 1

В первой главе рассмотрено понятие анализа тональности и его сферы применения, а также материалы, используемые при осуществлении анализа тональности. Рассмотрены различные методы, включая лингвистические подходы, подходы, основанные на машинном обучении и гибридные подходы.

В результате, можно сделать вывод, что существует множество подходов к анализу эмоциональной окраски, каждый из которых имеет свои достоинства и недостатки. Для успешного проведения анализа тональности крайне важно выбрать метод, наилучшим образом отвечающий целям и задачам исследования.

2. Практическая реализация

2.1. Подбор данных

Для проведения анализа эмоциональной окраски отзывов о сфере обслуживания в качестве основного источника данных были выбраны отзывы пользователей, размещенные на платформе «2ГИС». «2ГИС» — популярный онлайн-сервис, предоставляющий информацию об организациях, их местонахождении, контактах и деятельности. Кроме того, 2ГИС предлагает систему отзывов, где пользователи могут оставлять комментарии и оценки местам, которые они посетили, что делает эту платформу ценным ресурсом для анализа мнений клиентов.

Преимущества использования «2ГИС» как источника данных:

- «2ГИС» широко используется в разных регионах, что позволяет собирать отзывы из различных географических областей, охватывая разнообразные мнения и опыт пользователей. В нашем случае, использовались отзывы на организации, находящиеся на территории Санкт-Петербурга и Ленинградской области.
- «2ГИС» предлагает звездную рейтинговую систему, в которой пользователи могут оценивать свои впечатления по шкале от 1 до 5 звезд. Такая система позволяет количественно оценить общую удовлетворенность клиентов и провести сравнительный анализ между организациями. Отзывы с большим количеством звезд (4–5), как правило, указывают на положительный опыт, а отзывы с низким количеством звезд (1–2) указывают на отрицательный опыт.
- Платформа постоянно пополняется новыми отзывами, что дает возможность анализировать актуальные данные.

Для извлечения данных с веб-страниц «2ГИС» использовалась готовая программа «Parser2GIS» [Трофимов А. 2022]. Этот инструмент позволяет автоматизировать процесс сбора данных и получить большой объем информации для анализа. Для работы программы необходим установленный веб-браузер Google Chrome. Это обусловлено тем, что программа использует браузерный движок Chrome для загрузки данных и выполнения кода.

Программа «Parser2GIS» имеет собственный интерфейс, который отличается удобством для пользователя и простотой использования. Интерфейс обеспечивает интуитивно понятный доступ к основным функциям программы, позволяя легко настраивать параметры парсинга, запускать процесс сбора данных и наблюдать процесс проведения парсинга в реальном времени, предоставляя пользователю информацию о текущем состоянии выполнения задач.

На рисунке 4 представлен интерфейс программы.

Рисунок 4. Интерфейс программы «Parser2GIS»

Как видно на рисунке 4, интерфейс программы включает четыре основных окна:

1. URL: окно для ввода адресов веб-страниц, о которых необходимо собрать информацию. При нажатии на кнопку «…» справа от окна URL пользователю предоставляется возможность выбрать рубрики организаций, информацию о которых пользователь желает получить. Это позволяет фильтровать данные и получать информацию только о конкретных типах организаций. Данная функция отображена на рисунке 5. В нашем случае были использованы рубрики «Досуг / Развлечения / Общественное питание», а также «Продукты питания / Напитки», в которых были выбраны подрубрики «Кафе», «Рестораны» и «Продовольственные магазины».

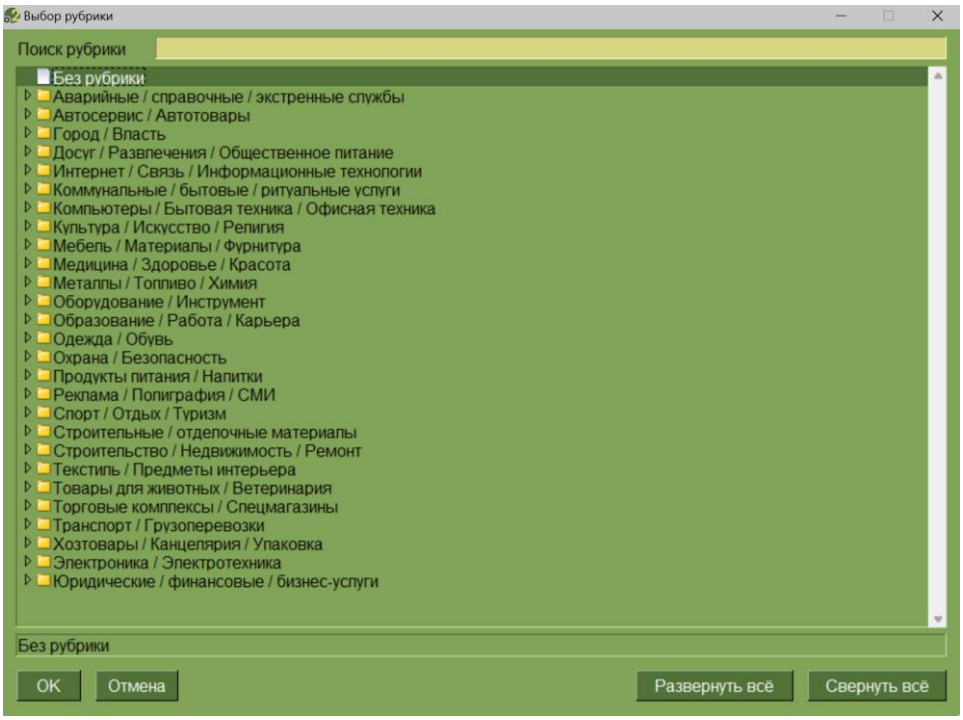


Рисунок 5. Выбор рубрики для парсинга

После выбора рубрики организации, пользователю предоставляется возможность выбрать страну и город, для которых необходимо выполнить парсинг данных. Список стран включает в себя Россию, Казахстан, Беларусь, Азербайджан, Киргизию, Узбекистан, Чехию, Египет, Италию, Саудовскую Аравию, Кипр, ОАЭ, Чили, Катар, Оман, Бахрейн и Кувейт. Выбор страны и города позволяет уточнить область сбора информации и сосредоточиться на

конкретном регионе. В нашем случае, была выбрана Россия, город Санкт-Петербург. Возможность выбора страны и города представлена на рисунке 6.

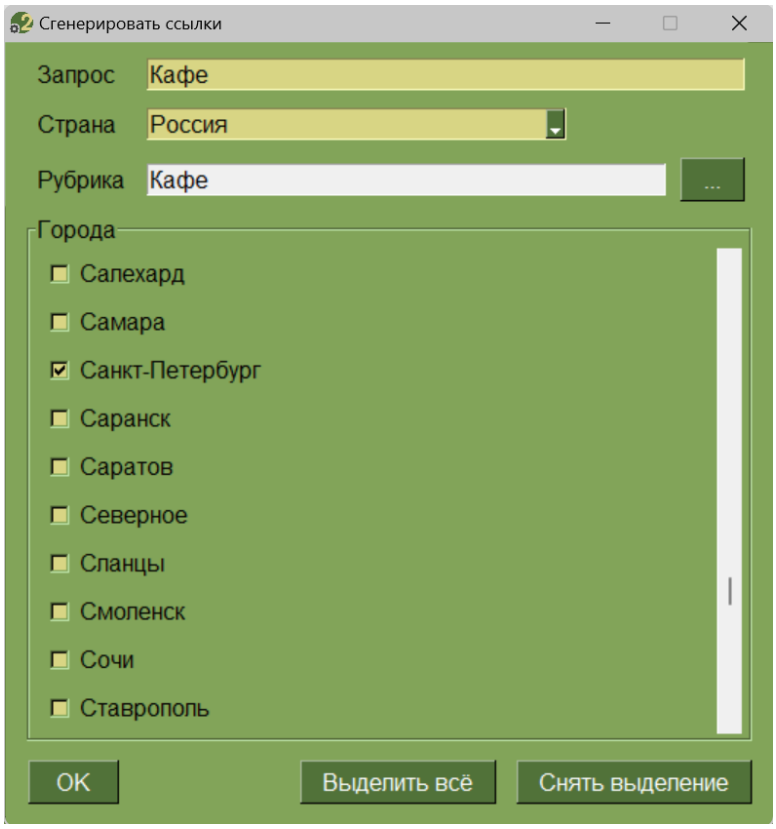


Рисунок 6. Выбор страны и города для парсинга

После выбора страны и города появляется сгенерированная ссылка, по которой будет происходить сбор данных. Также в программе есть возможность сгенерировать сразу несколько ссылок на разные рубрики, что позволяет собирать информацию сразу о нескольких рубриках.

2. Тип файла для сохранения результатов: окно для выбора формата файла, в котором будут сохраняться результаты парсинга. На выбор предлагаются три типа файла для сохранения результатов: CXV, XLSX и JSON. В нашем случае, результаты парсинга сохранялись в формате XLSX.

3. Путь для сохранения файла: окно для указания директории на компьютере, куда будет сохранен файл с результатами.
4. Лог: окно, в котором в реальном времени отображается процесс проведения парсинга, включая информацию о текущем состоянии выполнения задач, что позволяет наблюдать ход выполнения парсинга.

В программе имеется кнопка настройки, предоставляющая доступ к ряду параметров программы. Окно с настройками можно увидеть на рисунке 7.

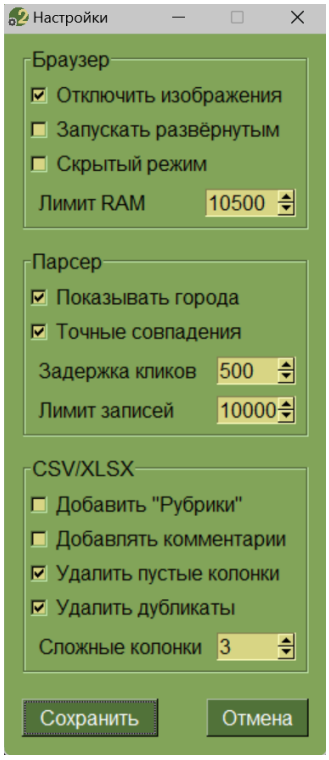


Рисунок 7. Настройки программы

Настройки браузера предоставляют возможность отключить изображения во время выполнения кода, а также выбрать режим запуска браузера: свернутый или скрытый. Кроме того, имеется возможность установить лимит оперативной памяти (RAM), при достижении которого процесс сбора данных будет прерван.

Настройки парсера включают в себя ряд параметров. Пользователь может выбрать опцию "показывать города", что позволит отображать города в списке выбора при настройке параметров. Дополнительно, активация опции

"точные совпадения" позволит программе пропускать ссылки, возвращающие сообщение "Точных совпадений не найдено". Также можно задать задержку кликов по записям в миллисекундах и установить лимит записей.

Настройки для форматов CSV/XLSX также предоставляют различные возможности. Пользователь может выбрать опцию "добавить рубрики" для включения информации о рубриках в файл с результатами. Дополнительно, есть возможность добавить комментарии к ячейкам (таким как адрес и др.), а также удалить пустые колонки и дубликаты. Пункт "сложные колонки" позволяет выбрать количество колонок для результата с несколькими возможными значениями, что особенно полезно в случае множественных значений, таких как номера телефонов.

После определения параметров начинается процесс сбора данных в окне Google Chrome, которое открывается автоматически. Программа открывает ссылку, сгенерированную ранее, и начинает проходить по выбранной рубрике по алфавитному порядку, собирая информацию о каждой организации в данной категории до тех пор, пока все организации не будут обработаны. Этот процесс представлен на рисунке 8.

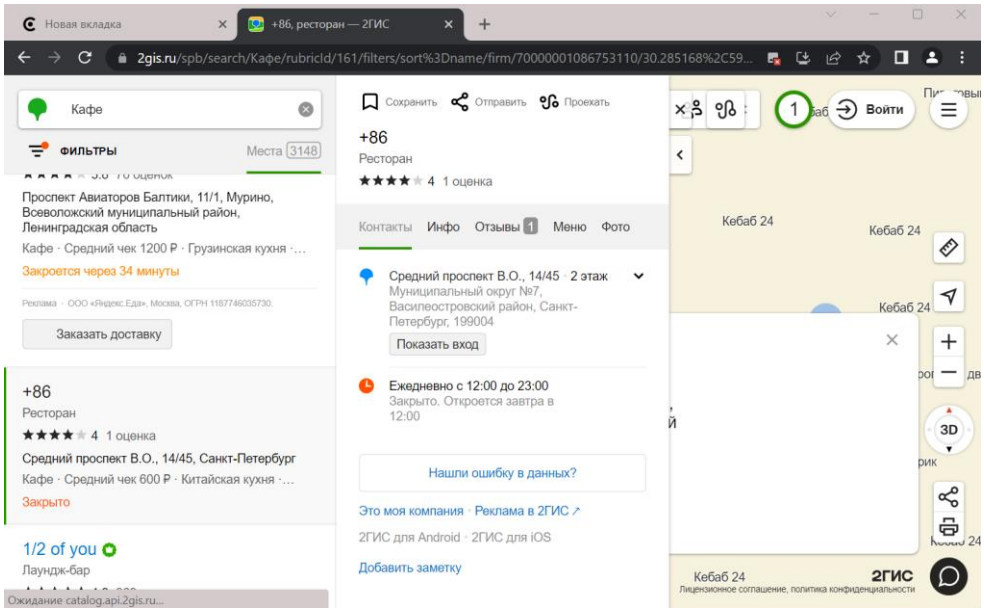


Рисунок 8. Выполнение парсинга в Google Chrome

В это время в окне программы «Лог» можно наблюдать информацию о собираемых данных, включая текущий статус выполнения парсинга. Процесс сбора данных может быть прерван в любое время с помощью нажатия на кнопку «Стоп». Сбор данных будет завершен, а полученная информация сохранится в файл в состоянии, на момент остановки парсинга. «Лог» программы во время выполнения парсинга можно увидеть на рисунке 9.

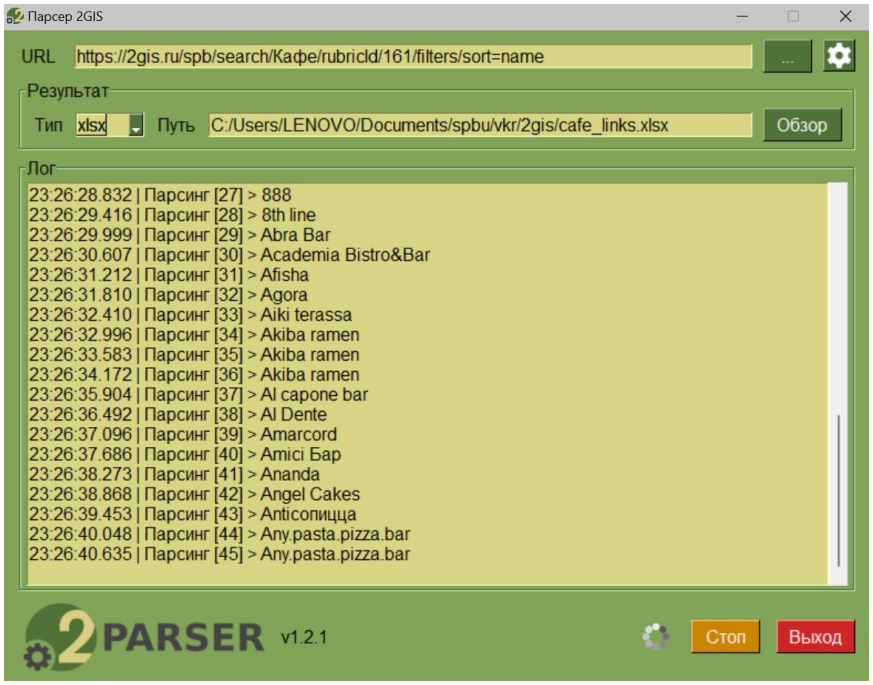


Рисунок 9. Окно «Лог» во время выполнения парсинга

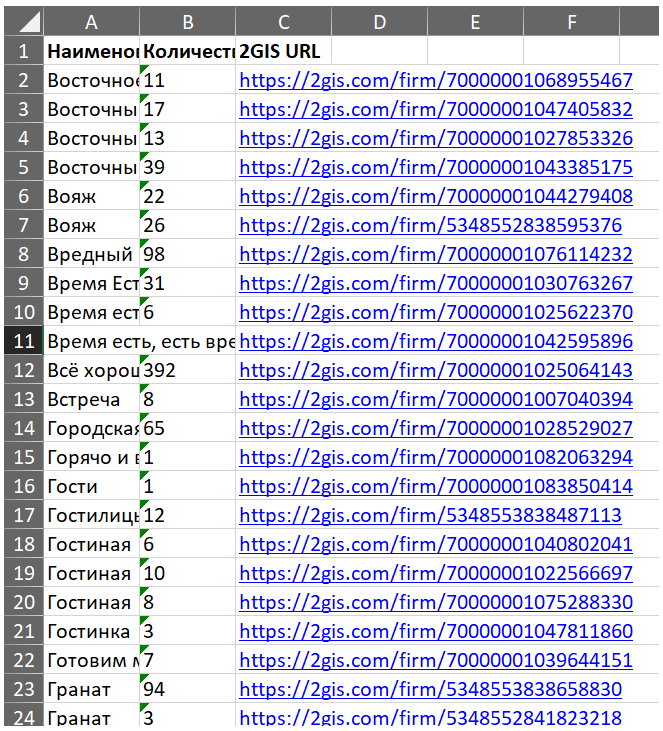
Файл, полученный в результате работы программы (в нашем случае – файл в формате XLSX), содержит несколько колонок с подробной информацией об организациях. Каждая строка файла представляет отдельную организацию, а колонки включают в себя следующие данные:

- наименование организации;
- описание;
- адрес;
- комментарий;
- почтовый индекс;
- микрорайон;
- район;

- город;
- округ;
- регион;
- страна;
- часы работы;
- часовой пояс;
- общий рейтинг отзывов;
- количество отзывов;
- номера телефонов;
- адреса электронной почты;
- веб-сайт организации;
- ссылки на социальные сети организации;
- широта;
- долгота;
- ссылка на карточку организации на сайте 2ГИС.

Эта структура данных представляет обширную информацию о каждой организации, что удобно для последующего анализа и использования данных.

Для дальнейшей обработки были сохранены только три колонки: название организации, количество отзывов и ссылка на карточку организации в 2ГИС. Остальные колонки были удалены вручную. Получившаяся структура данных после удаления ненужных колонок представлена на рисунке 10.



	A	B	C	D	E	F
1	Наименов	Количест	2GIS URL			
2	Восточно	11	https://2gis.com/firm/70000001068955467			
3	Восточны	17	https://2gis.com/firm/70000001047405832			
4	Восточны	13	https://2gis.com/firm/70000001027853326			
5	Восточны	39	https://2gis.com/firm/70000001043385175			
6	Вояж	22	https://2gis.com/firm/70000001044279408			
7	Вояж	26	https://2gis.com/firm/5348552838595376			
8	Вредный	98	https://2gis.com/firm/70000001076114232			
9	Время Ест	31	https://2gis.com/firm/70000001030763267			
10	Время ест	6	https://2gis.com/firm/70000001025622370			
11	Время есть, есть вре		https://2gis.com/firm/70000001042595896			
12	Всё хорош	392	https://2gis.com/firm/70000001025064143			
13	Встреча	8	https://2gis.com/firm/70000001007040394			
14	Городская	65	https://2gis.com/firm/70000001028529027			
15	Горячо и г	1	https://2gis.com/firm/70000001082063294			
16	Гости	1	https://2gis.com/firm/70000001083850414			
17	Гостилиц	12	https://2gis.com/firm/5348553838487113			
18	Гостиная	6	https://2gis.com/firm/70000001040802041			
19	Гостиная	10	https://2gis.com/firm/70000001022566697			
20	Гостиная	8	https://2gis.com/firm/70000001075288330			
21	Гостинка	3	https://2gis.com/firm/70000001047811860			
22	Готовим м	7	https://2gis.com/firm/70000001039644151			
23	Гранат	94	https://2gis.com/firm/5348553838658830			
24	Гранат	3	https://2gis.com/firm/5348552841823218			

Рисунок 40. Формат файла с результатами после удаления ненужных колонок

Для дальнейшей обработки был разработан код на языке Python, который извлекает ссылки из файла с данными об организациях и осуществляет фильтрацию организаций на основе колонки "количество отзывов". Организации, для которых количество отзывов равно нулю, исключаются из дальнейшей обработки. Полученные в результате обработки ссылки сохранялись в текстовый файл формата TXT. В результате обработки было получено 10,779 ссылок на организации.

Для парсинга отзывов с сайта 2ГИС был написан код на Python, использующий библиотеки *BeautifulSoup* и *Requests*. Этот код принимает на вход текстовый файл с ссылками, полученный на предыдущем этапе, и для каждой ссылки выполняет алгоритм по поиску HTML-тегов, содержащих отзывы и количество звезд для каждого отзыва.

Из-за структуры HTML-кода платформы «2ГИС», для каждой ссылки удалось собрать максимум 12 отзывов. Отзывы и соответствующее количество звезд, полученные в результате работы алгоритма, записывались в текстовый файл в следующем формате:

review: *текст отзыва*

stars: *количество звезд отзыва*

В результате проведенного анализа было установлено следующее количество словоупотреблений для различных рубрик:

- Рубрика "кафе": 592,952 слова;
- Рубрика "рестораны": 552,648 слов;
- Рубрика "магазины": 292,386 слов.

Эти данные свидетельствуют о большом объеме отзывов и активном участии пользователей в оценке организаций в данных категориях. Наибольшее количество словоупотреблений наблюдается в рубрике "кафе", что может указывать на более высокий уровень интереса и вовлеченности пользователей в оценивании кафе по сравнению с другими категориями. Это также может свидетельствовать о большем количестве посещений кафе или более детальных и объемных отзывах.

Рубрика "рестораны" занимает второе место по количеству словоупотреблений, что также демонстрирует высокий уровень активности пользователей, но несколько уступает кафе. Это может быть связано с меньшей детализацией отзывов или меньшим количеством посещений ресторанов по сравнению с кафе.

Наименьшее количество словоупотреблений зафиксировано в рубрике "магазины". Это может быть связано с различными факторами, такими как меньший интерес пользователей к оставлению отзывов или более лаконичный характер отзывов о магазинах.

В целом, собранные данные предоставляют важные сведения о предпочтениях и поведении пользователей в оценке организаций различных сфер обслуживания.

Для эффективной предобработки текста отзывов необходимо разработать стратегию фильтрации избыточной информации, которая не несет смысловой нагрузки и может исказить результаты анализа. Одним из

основных инструментов для этой цели является составление списка стоп-слов, которые игнорируются при анализе текста. Для создания такого списка была использована библиотека *Counter*, которая позволяет подсчитать частоту встречаемости слов в тексте. Пример частотного списка для категории «Кафе» представлен на рисунке 11.

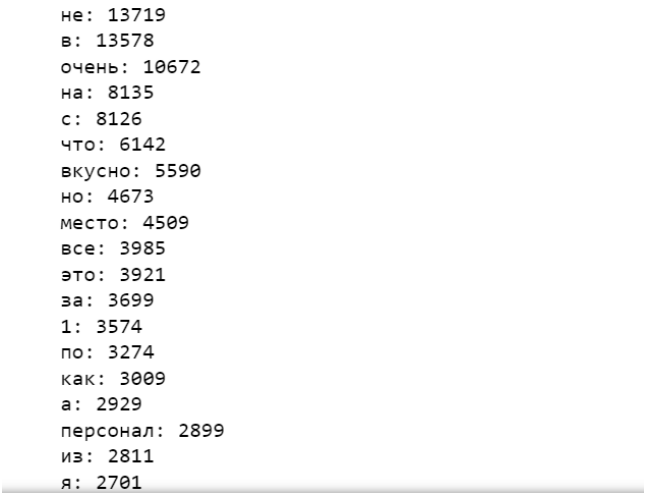


Рисунок 11. Частотный список слов для категории «Кафе»

Проанализировав полученный частотный список слов, были выявлены слова, которые встречаются слишком часто и не несут информационной ценности для последующего анализа. На основе этого анализа был составлен словарь стоп-слов. Этот словарь будет использоваться в дальнейшем для исключения стоп-слов из текста отзывов перед проведением более глубокого анализа.

Далее, с использованием библиотек *nltk* и *pymorphy2*, были осуществлены несколько этапов предобработки данных. В первую очередь было проведено удаление стоп-слов. Для этого созданный ранее стоп-словарь был объединен со стоп-словарем, включенным в библиотеку *nltk*. Также в процессе предобработки текста было решено удалить частицу "не" из списка стоп-слов, предоставляемом библиотекой *nltk*, поскольку это слово имеет значение для анализа эмоциональной окраски текста.

Затем из текстов была удалена пунктуация, чтобы исключить символы, не являющиеся буквами или цифрами, которые не имеют значения для последующего анализа текста.

Далее все тексты были приведены к нижнему регистру. Это позволяет сделать тексты единообразными для обработки, так как слова с разным регистром считаются разными словами, что может привести к искажению результатов анализа.

Наконец, была проведена лемматизация текстов. Лемматизация — это процесс приведения слова к его нормальной форме, или лемме. Она позволяет учитывать различные формы одного слова как одно и то же слово, что упрощает анализ текста и повышает точность результатов.

После проведения предобработки текста количество словоупотреблений значительно уменьшилось. Для категории "кафе" количество словоупотреблений составило 453,999, для ресторанов - 415,784, а для магазинов - 220,388.

Процентное уменьшение словоупотреблений составило:

Для кафе: ((592,952 - 453,999) / 592,952) * 100% ≈ 23.39%

Для ресторанов: ((552,648 - 415,784) / 552,648) * 100% ≈ 24.78%

Для магазинов: ((292,386 - 220,388) / 292,386) * 100% ≈ 24.59%

Таким образом, общее количество словоупотреблений в корпусе после предобработки составляет 1,090,171 слово, что сократилось по сравнению с общим количеством словоупотреблений до предобработки, которое составляло 1,437,986 слов.

В результате обработки данных были получены три отдельных файла с отзывами, каждый из которых соответствовал одной из исследуемых категорий: кафе, рестораны и магазины. Для упрощения дальнейшей обработки и анализа текстового корпуса отзывов возникла необходимость объединения этих файлов в единый набор данных.

С использованием кода на языке Python была создана единая таблица в формате XLSX. Таблица включает несколько ключевых колонок: текст отзыва, количество звезд, присвоенных данному отзыву, и новая колонка под названием "category", которая указывает категорию отзыва (ресторан, кафе или магазин). Этот объединенный формат данных способствует более эффективному и целостному анализу отзывов, позволяя учитывать категориальные различия и проводить сравнительный анализ внутри единого файла.

Далее был проведен анализ баланса отзывов. Этот анализ направлен на выявление неравномерности распределения отзывов по категориям и по количеству звезд, что является важным шагом для корректного построения и интерпретации классификатора.

Анализ баланса по колонке "category" необходим для выявления разницы в количестве отзывов между категориями. Это позволяет определить, есть ли существенные различия в количестве отзывов для каждой категории и учитывать это при построении моделей классификатора.

Анализ по колонке "stars" важен для определения баланса между различными оценками, которые могут быть даны отзывам. Неравномерное распределение оценок может повлиять на работу модели, особенно если одна или несколько категорий оценок представлены существенно чаще или реже других.

Распределение отзывов по категориям можно наблюдать на рисунке 12.

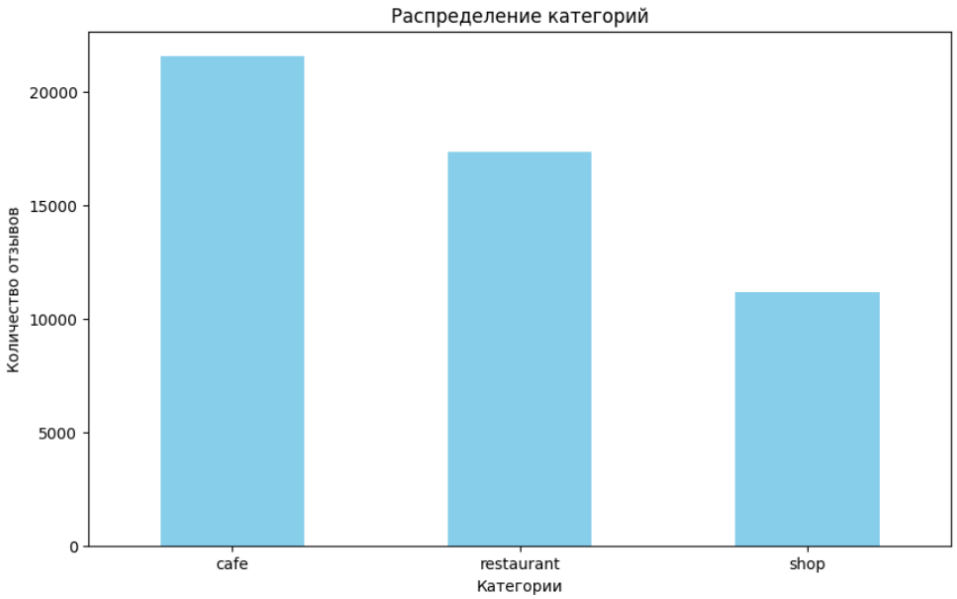


Рисунок 12. Распределение отзывов по категориям

Также более подробные результаты по распределению отзывов по категориям можно увидеть в таблице 2.

Таблица 2. Распределение отзывов по категориям

Категория	Количество отзывов
Кафе	21560
Рестораны	17343
Магазины	11154

Из результатов анализа распределения по категориям видно, что наблюдается неравномерность в количестве отзывов между различными категориями. Категория "кафе" имеет наибольшее количество отзывов, за ней следует категория "рестораны", а замыкает список категория "магазины". Это неравномерное распределение может повлиять на работу классификатора, поэтому необходимо учитывать этот фактор при разработке и интерпретации модели.

Распределение отзывов по количеству звезд можно наблюдать на рисунке 13.

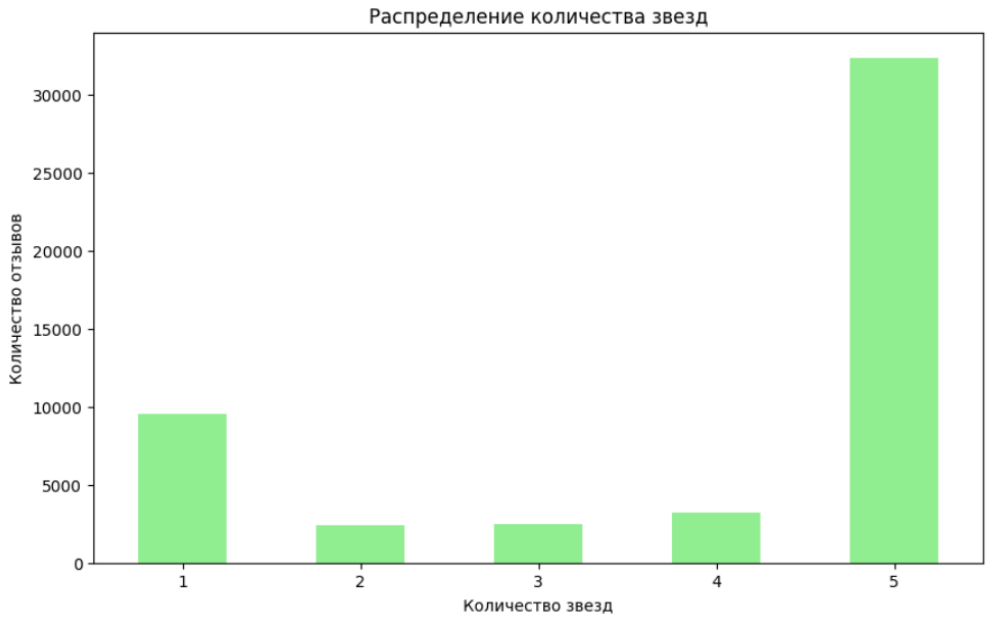


Рисунок 13. Распределение отзывов по количеству звезд

Также более подробное распределение можно увидеть в таблице 3.

Количество звезд	Количество отзывов
5	32380
4	3204
3	2509
2	2437
1	9527

На рисунке 14 можно увидеть общий график распределения по категориям и по количеству звезд.

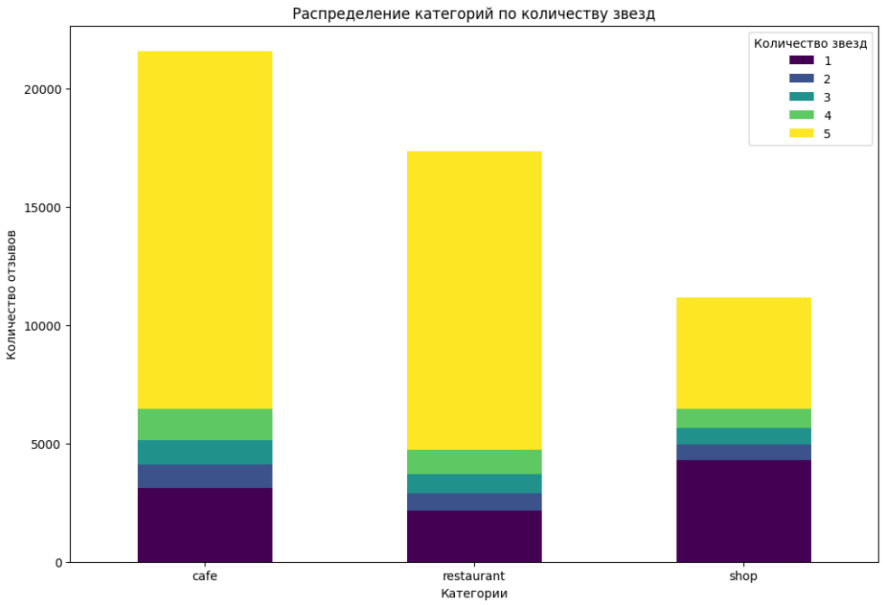


Рисунок 14. Распределение отзывов по категориям и количеству звезд

Из анализа распределения по количеству звезд видно, что большинство отзывов имеют оценку в 5 звезд. Это указывает на преобладание положительных отзывов в общем корпусе данных. На втором месте по числу отзывов оценка 1 звезда. Оценки в 4, 3 и 2 звезды находятся на третьем, четвертом и пятом местах соответственно.

Согласно полученному распределению, можно сделать предположение, что люди, оставляющие отзывы, зачастую склонны к крайностям в своих эмоциональных реакциях. Люди стремятся к упрощению, и это может влиять на их решение поставить 5 звезд или 1 звезду, чтобы выразить свои общие впечатления вместо того, чтобы разделить свою оценку на более мелкие категории.

2.2. Реализация классификатора

Для проведения анализа тональности полученных отзывов был создан классификатор на языке программирования Python, основанный на архитектуре GRU (Gated Recurrent Unit). Выбор GRU обоснован несколькими факторами:

1. GRU позволяет учитывать последовательную природу текстовых данных, что важно для анализа эмоциональной окраски отзывов. GRU способен эффективно обрабатывать последовательности переменной длины, сохраняя информацию о контексте.
2. В архитектуре есть встроенные механизмы, которые регулируют процесс забывания информации о предыдущих состояниях. Это позволяет модели сосредотачиваться на наиболее значимых элементах текста.
3. GRU также был выбран из-за его относительной простоты в реализации и обучении, по сравнению с более сложными моделями, такими как LSTM.

Работа с более сложными моделями, основанными на архитектуре Transformer, позволявшими получить еще более высокую точность, осложнялась следующими факторами:

1. Использование сложных моделей требует значительных вычислительных ресурсов и времени на обучение.
2. Использование сложных моделей затруднялось ограничениями в Google Colab. Google ввел ограничения на использование платной версии Google Colab на территории Российской Федерации. Бесплатная версия Google Colab имеет существенные ограничения, такие как количество оперативной памяти, ограничения по времени работы сессий и доступ к вычислительным ресурсам.

Таким образом, выбор GRU в качестве основы для классификатора эмоциональной окраски отзывов был обусловлен необходимостью эффективной и быстрой обработки данных в условиях ограниченных вычислительных ресурсов. GRU обеспечивает достаточную точность и быстроту обучения, что делает его оптимальным выбором для данной задачи.

Объяснение работы классификатора

В начале работы модель загружает файл с отзывами в формате XLSX и считывает его с помощью библиотеки *pandas*. Это позволяет работать с данными в формате *DataFrame*.

После загрузки данных создается функция *“sentiment_category”*, которая предназначена для классификации отзывов на основе их рейтинга (количества звезд). Функция принимает на вход значение количества звезд, присвоенное отзыву, и возвращает категорию тональности отзыва в соответствии с установленными критериями:

- если рейтинг отзыва составляет 1 или 2 звезды, отзыв классифицируется как негативный, и функция возвращает значение 0;
- если рейтинг отзыва составляет 3 звезды, отзыв классифицируется как нейтральный, и функция возвращает значение 1;
- если рейтинг отзыва составляет 4 или 5 звезд, отзыв классифицируется как позитивный, и функция возвращает значение 2.

Результат применения функции сохраняется в новый столбец *“sentiment”* в *DataFrame*. Таким образом, каждый отзыв получает соответствующую категорию тональности на основе его рейтинга. Это позволяет более эффективно анализировать и классифицировать отзывы в дальнейшем. Этот фрагмент кода можно увидеть на рисунке 15.

```
# Загрузка данных
df = pd.read_excel('reviews_data.xlsx')

# Преобразование звезд в категории тональности
def sentiment_category(stars):
    if stars in [1, 2]:
        return 0  # Negative
    elif stars == 3:
        return 1  # Neutral
    else:
        return 2  # Positive

df['sentiment'] = df['stars'].apply(sentiment_category)
```

Рисунок 15. Загрузка данных и преобразование звезд в категории тональности

Следующий фрагмент кода занимается созданием тонального словаря и извлечением тональных признаков из текста отзывов. Для этого сначала создается объект *"SentimentIntensityAnalyzer"* из библиотеки *VADER* (Valence Aware Dictionary and sEntiment Reasoner). *VADER* — это специализированный инструмент для анализа тональности текста, который особенно хорошо работает с короткими текстами, такими как отзывы, комментарии и сообщения в социальных сетях.

Этот объект *"SentimentIntensityAnalyzer"* будет использоваться для оценки тональности текста отзывов. VADER анализирует текст, оценивая эмоциональную окраску слов и фраз, и вычисляет численные значения тональности текста.

После создания объекта анализатора, создается функция, которая принимает текст на вход и использует метод *"polarity_scores"* объекта анализатора для вычисления тональных показателей текста. Метод *"polarity_scores"* анализирует входной текст и возвращает словарь с четырьмя ключами:

- *neg*: Оценка негативной тональности текста. Значение варьируется от 0 до 1 и показывает долю текста, содержащую негативные слова и выражения.

- *neu*: Оценка нейтральной тональности текста. Значение варьируется от 0 до 1 и показывает долю текста, содержащую нейтральные слова и выражения.
- *pos*: Оценка позитивной тональности текста. Значение варьируется от 0 до 1 и показывает долю текста, содержащую позитивные слова и выражения.
- *compound*: Комплексная оценка тональности текста. Значение варьируется от -1 (негативная) до 1 (позитивная). Это агрегированная метрика, которая суммирует все значения и нормализует их для определения общей эмоциональной окраски текста.

Далее в коде применяется функция *"extract_sentiment_features"* ко всем обработанным отзывам в столбце *"processed_review" DataFrame* с помощью метода *apply*. Метод *apply* вызывает эту функцию для каждого элемента столбца *"processed_review"*.

Поскольку функция *"extract_sentiment_features"* возвращает список из четырех элементов, представляющих оценки тональности, каждый из этих элементов преобразуется в отдельный столбец *DataFrame*. Результатом выполнения этой строки кода будет создание четырех новых столбцов: *neg, neu, pos* и *compound*, содержащих соответствующие оценки тональности для каждого отзыва.

Таким образом, каждый отзыв будет представлен четырьмя числовыми значениями, отражающими его эмоциональную окраску: долю негативных, нейтральных и позитивных выражений, а также общую комплексную оценку. Эту часть кода можно увидеть на рисунке 16.

```
[6]  # Создание тонального словаря
     analyzer = SentimentIntensityAnalyzer()
     def extract_sentiment_features(text):
         vs = analyzer.polarity_scores(text)
         return [vs['neg'], vs['neu'], vs['pos'], vs['compound']]

     # Извлечение тональных признаков
     df[['neg', 'neu', 'pos', 'compound']] = df['processed_review'].apply(lambda x: pd.Series(extract_sentiment_features(x)))
```

Рисунок 16. Создание тонального словаря и извлечение признаков

Так как ранее мы выяснили, что данные являются несбалансированными, следующая часть кода отвечает за их балансировку.

Здесь создаются отдельные *DataFrame* для положительных, негативных и нейтральных отзывов, фильтруя исходный *DataFrame* по значению столбца *“sentiment”*.

Далее происходит увеличение числа нейтральных отзывов путем повторного выбора (с заменой) из исходных данных. Функция *“resample”* из модуля *“sklearn.utils”* выполняет эту задачу. Параметр *“n_samples”* устанавливает количество отобранных образцов. Он устанавливается на максимальное количество отзывов в категориях положительных отзывов, чтобы сбалансировать данные. Далее, такой же алгоритм производит увеличение выборки и негативных отзывов.

Затем сбалансированный *DataFrame* создается путем объединения отдельных *DataFrame* для каждой категории. Это дает равное количество отзывов в каждой категории и улучшает работу модели, которая может быть склонна к переобучению из-за дисбаланса классов.

На рисунке 17 показан отрывок кода, отвечающий за сбалансирование данных.

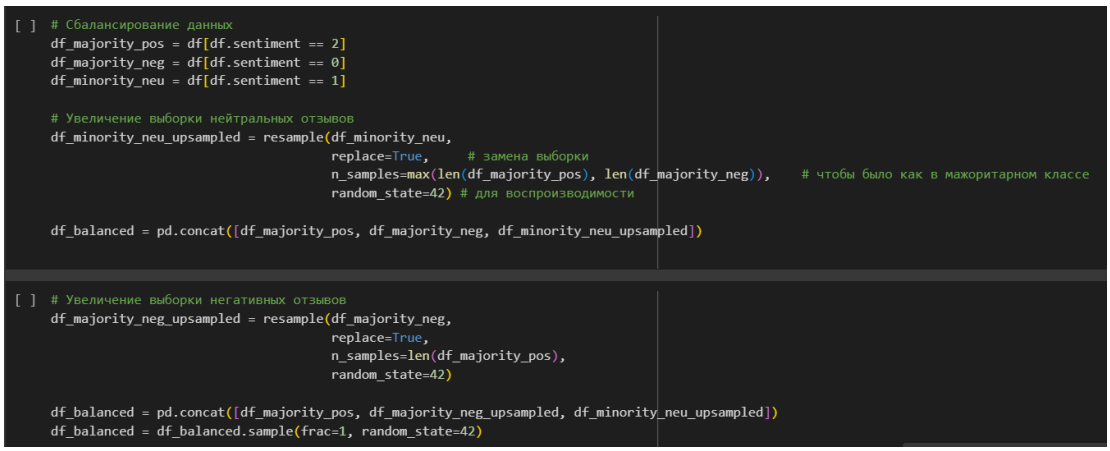


Рисунок 17. Сбалансирование данных

Следующая часть отвечает за перемешивание данных на признаки и целевую переменную.

- *X*: Признаки (или независимые переменные) — это столбцы *DataFrame*, которые будут использоваться для предсказания целевой переменной. Здесь *X* включает в себя текстовые данные (*"processed_review"*) и тональные признаки (*neg*, *neu*, *pos*, *compound*), которые были извлечены ранее.
- *y*: Целевая переменная (или зависимая переменная) — это столбец *"sentiment"*, который содержит категорию тональности отзыва (0, 1 или 2).

Затем данные разделяются на тренировочную и тестовую выборки:

"X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)"

Таким образом, после выполнения этого кода данные будут разделены на четыре набора: *"X_train"* и *"y_train"* для обучения модели, и *"X_test"* и *"y_test"* для оценки ее производительности.

Следующая часть отвечает за подготовку текстовых данных для подачи в модель. Она включает токенизацию текста, паддинг последовательностей и объединение текстовых данных с тональными признаками.

Создается экземпляр класса *Tokenizer* из библиотеки *Keras*. Токенизатор будет использоваться для преобразования текстов в последовательности

чисел. Токенизатор обучается на текстовых данных из тренировочного набора. Это создает словарь, в котором каждому уникальному слову соответствует уникальный индекс. Тексты из тренировочного набора преобразуются в последовательности чисел на основе словаря, созданного токенизатором. Тексты из тестового набора также преобразуются в последовательности чисел, используя тот же словарь.

Затем происходит паддинг последовательностей. Паддинг (padding) последовательностей — это метод, используемый в обработке текстовых данных и других последовательностей для приведения всех входных данных к одной длине. В GRU, как и в других RNN, последовательности различной длины могут вызвать проблемы при обучении, так как нейронная сеть ожидает входных данных фиксированной длины. Паддинг решает эту проблему, добавляя нули к последовательностям, чтобы они соответствовали длине самой длинной последовательности в наборе данных. Максимальная длина последовательности в нашем случае – 100. Функция *“pad_sequences”* из библиотеки *Keras* преобразует последовательности из тренировочного и тестового набора.

Далее тональные признаки извлекаются из тренировочного и тестового набора для подачи в модель. На рисунке 17 можно увидеть данную часть кода.

```
[ ]  # Токенизация текста
     tokenizer = Tokenizer()
     tokenizer.fit_on_texts(X_train['processed_review'])
     X_train_seq = tokenizer.texts_to_sequences(X_train['processed_review'])
     X_test_seq = tokenizer.texts_to_sequences(X_test['processed_review'])

     # Паддинг последовательностей
     max_len = 100
     X_train_pad = pad_sequences(X_train_seq, maxlen=max_len)
     X_test_pad = pad_sequences(X_test_seq, maxlen=max_len)

     # Объединение текстовых данных с тональными признаками
     X_train_features = np.array(X_train[['neg', 'neu', 'pos', 'compound']])
     X_test_features = np.array(X_test[['neg', 'neu', 'pos', 'compound']])
```

Рисунок 17. Токенизация, паддинг и извлечение тональных признаков

Следующая часть кода использует библиотеку *Keras* для определения и компиляции нейронной сети. Модель имеет два входа: текстовые данные и

дополнительные признаки тональности. Модель предсказывает одну из трех категорий тональности (негативная, нейтральная, позитивная).

Модель включает в себя следующие компоненты:

- Слой *Embedding*, преобразующий каждое слово в вектор, размерностью 128;
- GRU слои:
    - Один двунаправленный слой с 64 нейронами, возвращающий полные последовательности;
    - Дополнительный слой с 64 нейронами, возвращающий конечное состояние последовательности.
- Слой *“concatenate”*, который объединяет выход слоя GRU и дополнительные признаки тональности;
- Полносвязный слой с 64 нейронами, функция активации *ReLU* (вводит в модель нелинейность, что позволяет сети учиться сложным зависимостям в данных) и регуляризацией L2 (добавляет штрафы за большие веса в функцию потерь, что помогает избежать переобучения модели);
- Выходной полносвязный слой с 3 нейронами (по одному для каждой категории тональности) и функцией активации *softmax* для получения вероятностей классов.
- Модель компилируется с оптимизатором *Adam*, функцией потерь *“sparse_categorical_crossentropy”*, которая используется для многоклассовой классификации, и метрикой *“accuracy”* для оценки качества модели.

Часть кода с определением модели можно увидеть на рисунке 18.

```
# Определение модели
input_text = Input(shape=(max_len,))
input_features = Input(shape=(4,))
embedding_layer = Embedding(input_dim=len(tokenizer.word_index)+1, output_dim=128, input_length=max_len)(input_text)
gru = Bidirectional(GRU(64, return_sequences=True))(embedding_layer)
gru = GRU(64)(gru)
concat = Concatenate()([gru, input_features])
dense1 = Dense(64, activation='relu', kernel_regularizer=l2(0.01))(concat)
dropout = Dropout(0.5)(dense1)
output = Dense(3, activation='softmax')(dropout)

model = Model(inputs=[input_text, input_features], outputs=output)
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

Рисунок 18. Определение модели

На следующем этапе происходит обучение модели. Модель обучается, используя тренировочные данные, в течение 10 эпох. Размер батча, или количество примеров, обрабатываемых моделью одновременно, составляет 128. В процессе обучения модель использует 20% от общего объема тренировочных данных для валидации. Это позволяет отслеживать ее производительность на этих данных и предотвращать переобучение, обеспечивая лучшую обобщающую способность модели. Код для обучения можно увидеть на рисунке 19.

```
[ ]  # Обучение модели
     history = model.fit([X_train_pad, X_train_features], y_train, epochs=10, batch_size=128, validation_split=0.2)
```

Рисунок 19. Обучение модели

2.3. Оценка полученных результатов

Для сравнения с нашей моделью был создан классификатор Multinomial Naive Bayes, который используется в качестве базовой линии (бейслайна). Классификатор Multinomial Naive Bayes является простым вероятностным алгоритмом машинного обучения, основанным на принципе наивного байесовского классификатора.

Такое сравнение позволяет оценить эффективность исследуемой модели относительно простых и широко используемых алгоритмов, что важно для понимания её потенциала.

Мы получили следующие метрики и матрицу ошибок для наивного байесовского классификатора:

- *Accuracy* (точность) составляет 0.88, что означает, что наш классификатор правильно классифицировал 88% общего количества образцов.
- *Precision* (точность) равна 0.86, что говорит о том, что 86% объектов, которые наш классификатор отнес к положительному классу, действительно принадлежат к этому классу.
- *Recall* (полнота) также равен 0.88, что означает, что наш классификатор правильно классифицировал 88% истинно положительных образцов относительно всех истинно положительных и истинно отрицательных образцов.
- *F1 Score* (F-мера) составляет 0.87, что представляет собой гармоническое среднее между точностью и полнотой и дает более сбалансированную оценку производительности классификатора.

Матрица ошибок показывает, что классификатор хорошо справляется с классификацией объектов классов "negative" и "positive", но имеет трудности с классификацией объектов класса "neutral". Матрицу ошибок можно увидеть на рисунке 20.

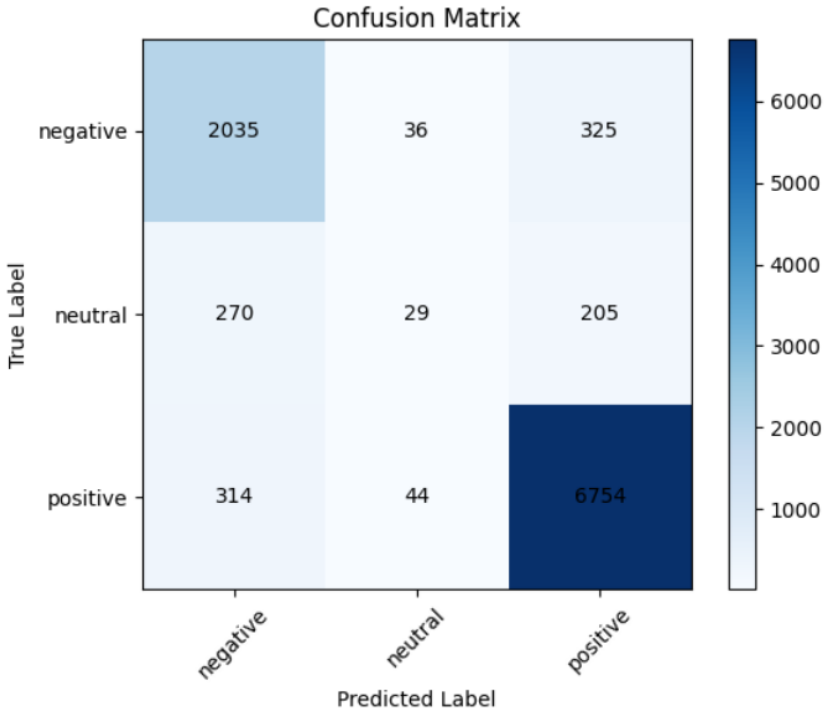


Рисунок 20. Матрица ошибок для бейслайн модели

Несмотря на то, что наивный байесовский классификатор показал хорошие результаты, GRU модель имеет потенциал для более точного и глубокого анализа данных.

На рисунке 21 можно наблюдать процесс обучения нашей GRU модели.

```
Epoch 1/10
534/534 [==============================] - 380s 696ms/step - loss: 0.6137 - accuracy: 0.7950 - val_loss: 0.3092 - val_accuracy: 0.9049
Epoch 2/10
534/534 [==============================] - 367s 688ms/step - loss: 0.2379 - accuracy: 0.9333 - val_loss: 0.2374 - val_accuracy: 0.9285
Epoch 3/10
534/534 [==============================] - 365s 684ms/step - loss: 0.1583 - accuracy: 0.9581 - val_loss: 0.1923 - val_accuracy: 0.9461
Epoch 4/10
534/534 [==============================] - 343s 643ms/step - loss: 0.1165 - accuracy: 0.9716 - val_loss: 0.1928 - val_accuracy: 0.9499
Epoch 5/10
534/534 [==============================] - 365s 684ms/step - loss: 0.0977 - accuracy: 0.9763 - val_loss: 0.1907 - val_accuracy: 0.9527
Epoch 6/10
534/534 [==============================] - 346s 649ms/step - loss: 0.0853 - accuracy: 0.9793 - val_loss: 0.1754 - val_accuracy: 0.9576
Epoch 7/10
534/534 [==============================] - 367s 687ms/step - loss: 0.0711 - accuracy: 0.9830 - val_loss: 0.2014 - val_accuracy: 0.9553
Epoch 8/10
534/534 [==============================] - 366s 685ms/step - loss: 0.0640 - accuracy: 0.9846 - val_loss: 0.1974 - val_accuracy: 0.9593
Epoch 9/10
534/534 [==============================] - 367s 688ms/step - loss: 0.0582 - accuracy: 0.9860 - val_loss: 0.1818 - val_accuracy: 0.9571
Epoch 10/10
534/534 [==============================] - 369s 691ms/step - loss: 0.0561 - accuracy: 0.9865 - val_loss: 0.1706 - val_accuracy: 0.9617
```

Рисунок 21. Обучение GRU

Какие выводы можно сделать на основе обучения модели:

1. На протяжении всех 10 эпох обучения наблюдается стабильное повышение точности модели на тренировочных данных, начиная с 79,50% на первой эпохе и достигая 98,65% на последней.

2. Валидационная точность модели также демонстрирует значительное улучшение, увеличиваясь с 90,65% на первой эпохе до 96,17% на последней.
3. Параллельно с ростом точности, потери на тренировочных данных последовательно снижаются с 0.6137 до 0.0561, что свидетельствует о том, что модель успешно минимизирует ошибку предсказаний.
4. Валидационные потери также уменьшаются, начиная с 0.3092 на первой эпохе и снижаясь до 0.1706 на последней, что говорит о хорошей обобщающей способности модели.
5. В некоторых эпохах (например, на четвертой и седьмой) наблюдается небольшое увеличение валидационных потерь, что может указывать на начальные признаки переобучения. Однако эти колебания незначительны, и общая тенденция показывает снижение валидационных потерь.
6. Валидационная точность продолжает увеличиваться даже на последних эпохах, что свидетельствует о том, что модель не испытывает значительных проблем с переобучением и сохраняет свою способность обобщать на новые данные.
7. Финальные метрики показывают высокие значения точности на тренировочных данных (98.65%) и валидационных данных (96.17%), что свидетельствует о высокой эффективности модели GRU в задаче классификации тональности отзывов.
8. Высокая точность на валидационных данных особенно важна, так как она демонстрирует, что модель способна хорошо работать не только на обучающей выборке, но и на новых данных.

Результаты обучения модели GRU значительно превосходят показатели наивного байесовского классификатора, который показал точность 88.07% и F1-меру 86.55%. Модель GRU, достигшая точности 96.17% на валидационных

данных, демонстрирует более высокую способность к точной классификации отзывов.

Улучшение метрик на модели GRU по сравнению с наивным байесовским классификатором указывает на её большую сложность и способность учитывать более глубокие зависимости в данных, что делает её более подходящей для анализа тональности текстов.

На рисунке 22 можно увидеть визуализацию обучения модели GRU.

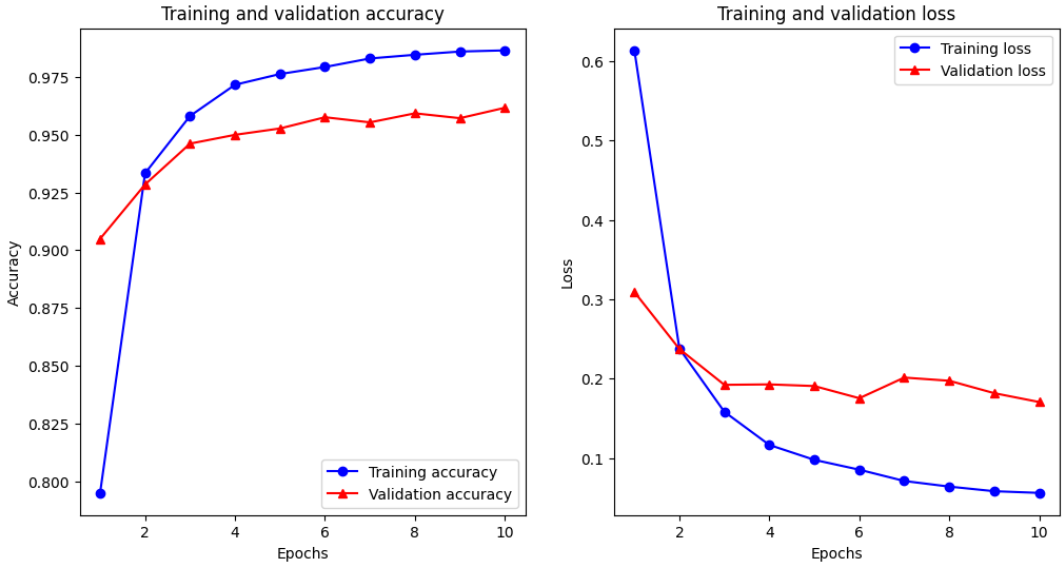


Рисунок 22. Визуализация обучения GRU

График Training and Validation Accuracy:

На графике видно, что точность обучения (синяя линия) постоянно увеличивается по мере увеличения количества эпох, начиная с приблизительно 80% на первой эпохе и достигая 98% к десятой эпохе. Это указывает на то, что модель хорошо обучается на тренировочных данных.

Точность валидации (красная линия) также увеличивается по мере увеличения количества эпох, однако после примерно пятой эпохи рост замедляется и стабилизируется около 95-96%. Этот феномен может свидетельствовать о том, что модель достигает своей максимальной производительности на валидационных данных.

График Training and Validation Loss:

На графике видно, что потери на тренировочных данных (синяя линия) значительно снижаются с каждой эпохой, начиная с 0.6 на первой эпохе и уменьшаясь до примерно 0.05 на десятой эпохе. Это указывает на то, что модель успешно минимизирует ошибку на тренировочных данных.

Потери на валидационных данных (красная линия) также уменьшаются в первые несколько эпох, достигая минимального значения около 0.17 на пятой-шестой эпохах, после чего стабилизируются с незначительными колебаниями. Этот график указывает на то, что модель достаточно хорошо работает на валидационных данных.

На основании графиков, можно сделать вывод, что оптимальное количество эпох для данной модели находится в диапазоне от 6 до 10. При увеличении числа эпох далее, улучшение точности на валидационных данных становится незначительным, в то время как риск переобучения возрастает.

На рисунке 23 можно наблюдать матрицу ошибок для модели GRU.

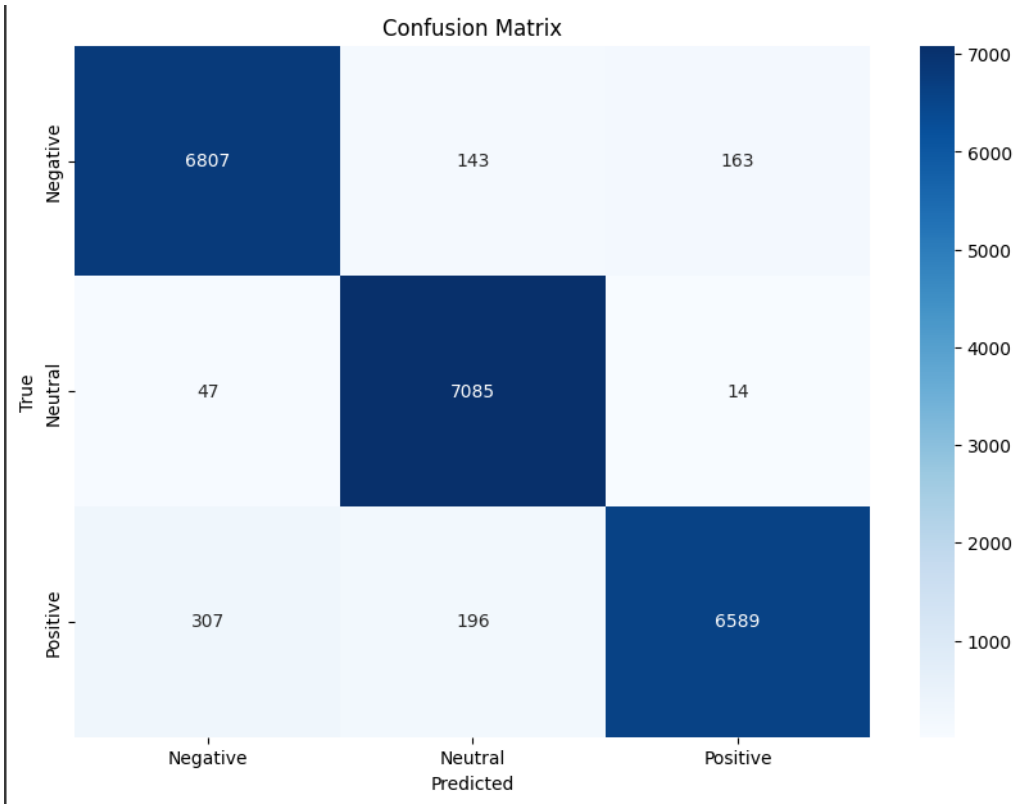


Рисунок 23. Матрица ошибок GRU

Модель GRU, по сравнению с наивным байесовским классификатором, продемонстрировала значительно более высокую точность в классификации отзывов. Особенно заметны улучшения в точности классификации негативных и нейтральных отзывов. Этот успех можно отнести к процессу балансировки классов, который был применен к данным перед обучением модели GRU. В процессе балансировки количество негативных и нейтральных отзывов было искусственно увеличено, что позволило модели лучше справляться с классификацией этих категорий.

Хотя GRU модель немного уступает наивному байесовскому классификатору в точности классификации позитивных отзывов, общее количество правильно классифицированных отзывов остается высоким.

Использование GRU модели с предварительной балансировкой классов показало, что такая стратегия позволяет значительно улучшить точность классификации негативных и нейтральных отзывов. Этот подход делает GRU модель более надежной и точной для анализа эмоциональной окраски отзывов, особенно в задачах, где важно улавливать тонкости негативных и нейтральных отзывов. В отличие от наивного байесовского классификатора, который показал хорошие результаты только для позитивных отзывов, GRU модель обеспечивает более сбалансированное и точное распределение классификации по всем категориям.

2.4. Выводы к Главе 2

В данной главе была проведена практическая реализация модели для анализа эмоциональной окраски. Процесс включал в себя подбор данных, реализацию классификатора и оценку полученных результатов.

Были реализованы две модели: модель наивного байесовского классификатора, который использовался как бейслайн, и модели на основе архитектуры GRU. Обе модели были обучены на одном и том же наборе данных.

Результаты показали, что модель GRU значительно превзошла результаты бейслайна, демонстрируя более высокую точность при проведении анализа тональности. Это подтверждает гипотезу о том, что использование нейросетевых моделей машинного обучения может значительно улучшить качество анализа эмоциональной окраски.

Заключение

В ходе данного исследования была успешно разработана и применена нейросетевая модель для анализа тональности текстовых отзывов о сфере обслуживания. Сравнительный анализ различных методов определения эмоциональной окраски, включая лингвистические подходы, методы машинного обучения и гибридные методы, позволил определить достоинства и недостатки каждого из методов, а также показал превосходство нейросетевого подхода в данном контексте.

В исследовании были успешно решены основные поставленные задачи исследования:

1. Был проведен сбор и предварительная обработка данных для использования в обучении модели.
2. Была разработана и реализована архитектура нейросетевого классификатора, способного определять эмоциональную окраску текстовых данных.
3. На основе подготовленных данных был проведен процесс обучения нейросетевой модели для определения тональности отзывов.
4. Была выполнена оценка точности и эффективности разработанного нейросетевого классификатора на основе полученных результатов анализа.

Разработанный нейросетевый классификатор продемонстрировал высокую точность и эффективность в определении эмоциональной окраски отзывов о сфере обслуживания на основе данных с платформы "2ГИС". Это подтверждает гипотезу о возможности создания эффективного инструмента для автоматизированного анализа эмоциональной окраски текстовых данных в данной сфере.

Для дальнейшего развития исследования рекомендуется увеличить объем и разнообразие данных, изучить влияние эмоциональной окраски на

поведение потребителей, улучшить архитектуру нейронной сети, расширить область применения модели на другие сферы.

Список использованной литературы

1. Bengio Y., Simard P., Frasconi P. Learning long-term dependencies with gradient descent is difficult [Electronic resource] // IEEE Transactions on Neural Networks. 1994. Vol. 5. № 2. p. 157-166. URL: https://www.comp.hkbu.edu.hk/~markus/teaching/comp7650/tnn-94-gradient.pdf (date of treatment: 23.03.2024).
2. Bojanowski P., Grave E., Joulin A., et al. Enriching word vectors with subword information [Electronic resource] // Transactions of the association for computational linguistics. 2017. Vol. 5. p. 135-146. URL: https://arxiv.org/pdf/1607.04606 (date of treatment: 18.03.2024).
3. Cataldi M., Ballatore A., Tiddi I., et al. Good Location, Terrible Food: Detecting Feature Sentiment in User-Generated Reviews [Electronic resource] // Social Network Analysis and Mining. 2013. Vol. 3. p. 1149-1163. URL: https://www.researchgate.net/publication/257801355_Good_Location_Terrible_Food_Detecting_Feature_Sentiment_in_User-Generated_Reviews (date of treatment: 02.03.2024).
4. Cho K., van Merrienboer B., Bahdanau D., et al. On the properties of neural machine translation: Encoder-decoder approaches [Electronic resource] // arXiv preprint arXiv:1409.1259. 2014. URL: https://arxiv.org/pdf/1409.1259 (date of treatment: 27.03.2024).
5. Chumakov S., Kovantsev A., Surikov A. Generative approach to Aspect Based Sentiment Analysis with GPT Language Models [Electronic resource] // Procedia Computer Science. 2023. Vol. 229. p. 284-293. URL: https://www.sciencedirect.com/science/article/pii/S1877050923020203 (date of treatment: 17.04.2024).
6. Dai J., Yan H., Sun T., et al. Does syntax matter? A strong baseline for Aspect-based Sentiment Analysis with RoBERTa [Electronic resource] //

arXiv preprint arXiv:2104.04986. 2021. URL: https://arxiv.org/pdf/2104.04986 (date of treatment: 05.04.2024).

7. Dang N. C., Moreno-García M. N., De la Prieta F. Sentiment Analysis Based on Deep Learning: A Comparative Study [Electronic resource] // Electronics. 2020. Vol. 9. № 3. p. 483. URL: https://doi.org/10.3390/electronics9030483 (date of treatment: 17.03.2024).
8. Danyal M. M., Khan S. S., Khan M., et al. Proposing sentiment analysis model based on BERT and XLNet for movie reviews [Electronic resource] // Multimedia Tools and Applications. 2024. p. 1-25. URL: https://link.springer.com/article/10.1007/s11042-024-18156-5 (date of treatment: 08.04.2024).
9. Dave K., Lawrence S., Pennock D. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews [Electronic resource] // Proceedings of the 12th international conference on World Wide Web. 2003. p. 519-528. URL: https://www.researchgate.net/publication/2904559_Mining_the_Peanut_Gallery_Opinion_Extraction_and_Semantic_Classification_of_Product_Reviews (date of treatment: 26.02.2024).
10. Devlin J., Chang M., Lee K., et al. BERT: Pre-training of deep bidirectional transformers for language understanding [Electronic resource] // arXiv preprint arXiv:1810.04805. 2018. URL: https://arxiv.org/pdf/1810.04805 (date of treatment: 03.04.2024).
11. GigaChat API [Электронный ресурс] // Решения для бизнеса: технологии и сервисы для компании от экосистема Сбербанка. 2024. URL: https://developers.sber.ru/docs/ru/gigachat/api/overview (дата обращения: 15.04.2024).
12. Habbat N., Anoun H., Hassouni L. Combination of GRU and CNN deep learning models for sentiment analysis on French customer reviews using

XLNet model [Electronic resource] // IEEE Engineering Management Review. 2022. Vol. 51. № 1. p. 41-51. URL: https://ieeexplore.ieee.org/abstract/document/9900423 (date of treatment: 08.04.2024).

13. Hatzivassiloglou V., McKeown K. R. Predicting the Semantic Orientation of Adjectives [Electronic resource] // 35th annual meeting of the association for computational linguistics and 8th conference of the european chapter of the association for computational linguistics. 1997. p. 174-181. URL: https://aclanthology.org/P97-1023.pdf (date of treatment: 25.02.2024).
14. Hatzivassiloglou V., Wiebe J. M. Effects of Adjective Orientation and Gradability on Sentence Subjectivity [Electronic resource] // COLING 2000 Volume 1: The 18th international conference on computational linguistics. 2000. URL: https://aclanthology.org/C00-1044.pdf (date of treatment: 25.02.2024).
15. Ho V., Nguyen D., Nguyen D. et al. Emotion Recognition for Vietnamese Social Media Text [Electronic resource] // Computational Linguistics: 16th International Conference of the Pacific Association for Computational Linguistics, PACLING 2019, Hanoi, Vietnam, October 11–13, 2019, Revised Selected Papers 16. Springer Singapore, 2020. p. 319-333. URL: https://www.researchgate.net/publication/342618463_Emotion_Recognition_for_Vietnamese_Social_Media_Text (date of treatment: 02.03.2024).
16. Hu M., Liu B. Mining and Summarizing Customer Reviews // Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. 2004. p. 168-177. URL: https://www.cs.uic.edu/~liub/publications/kdd04-revSummary.pdf (date of treatment: 02.03.2024).
17. Kim S., Hovy E. Determining the sentiment of opinions [Electronic resource] // COLING 2004: Proceedings of the 20th International

Conference on Computational Linguistics. 2004. p. 1367-1373. URL: https://www.researchgate.net/publication/228889549_Determining_the_sentiment_of_opinions (date of treatment: 12.03.2024).

18. Kochergina K. S. Approaches to Forming an Evaluative Lexicon (Juridical Linguistic Aspect) [Electronic resource] // Актуальные проблемы лингвистики и литературоведения: сборник материалов I (XVI) Международной конференции молодых ученых (9-11 апреля 2015 г.). 2015. № 16. с. 365-367. URL: http://vital.lib.tsu.ru/vital/access/manager/Repository/vtls:000534180 (date of treatment: 12.03.2024).
19. König A. C., Brill E. Reducing the human overhead in text categorization [Electronic resource] // Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006. p. 598-603. URL: https://www.sciencedirect.com/science/article/pii/S1877050923020203 (date of treatment: 19.04.2024).
20. Lakshmi B. S., Raj P. S., Vikram R. R. Sentiment analysis using deep learning technique CNN with KMeans [Electronic resource] // International journal of pure and applied mathematics. 2017. Vol. 114. № 11. p. 47-57. URL: https://acadpubl.eu/jsi/2017-114-7-ICPCIT-2017/articles/11/6.pdf (date of treatment: 18.04.2024).
21. Liu B. Sentiment Analysis and Opinion Mining [Electronic resource] // Springer Nature. 2022. URL: https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf (date of treatment: 24.02.2024).
22. Liu Y., Ott M., Goyal N., et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach [Electronic resource] // arXiv preprint arXiv:1907.11692. 2019. URL: https://arxiv.org/pdf/1907.11692 (date of treatment: 05.04.2024).

23. Mikolov T., Chen K., Corrado G., et al. Efficient estimation of word representations in vector space [Electronic resource] // arXiv preprint arXiv:1301.3781. 2013. URL: https://arxiv.org/pdf/1301.3781 (date of treatment: 17.03.2024).
24. Mohammad S. M., Dunne C., Dorr B. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus [Electronic resource] // Proceedings of the 2009 conference on empirical methods in natural language processing. 2009. p. 599-608. URL: https://www.researchgate.net/publication/216017177_Generating_high-coverage_semantic_orientation_lexicons_from_overtly_marked_words_and_a_thesaurus (date of treatment: 12.03.2024).
25. Nasukawa T., Yi J. Sentiment analysis: Capturing favorability using natural language processing [Electronic resource] // Proceedings of the 2nd international conference on Knowledge capture. 2003. p. 70-77. URL: https://www.researchgate.net/publication/220916772_Sentiment_analysis_Capturing_favorability_using_natural_language_processing (date of treatment: 25.02.2024).
26. Pang B., Lee L. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts [Electronic resource] // arXiv preprint cs/0409058. 2004. URL: https://www.cs.cornell.edu/home/llee/papers/cutsent.pdf (date of treatment: 03.03.2024).
27. Pang B., Lee L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales [Electronic resource] // arXiv preprint cs/0506075. 2005. URL: https://www.cs.cornell.edu/home/llee/papers/pang-lee-stars.pdf (date of treatment: 01.03.2024).
28. Pang B., Lee L., Vaithyanathan S. Thumbs up? Sentiment Classification using Machine Learning Techniques [Electronic resource] // arXiv preprint

cs/0205070. 2002. URL: https://www.cs.cornell.edu/home/llee/papers/sentiment.pdf (date of treatment: 28.02.2024).

29. Pennington J., Socher R., Manning C. D. Glove: Global vectors for word representation [Electronic resource] // Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014. p. 1532-1543. URL: https://nlp.stanford.edu/pubs/glove.pdf (date of treatment: 18.03.2024).
30. Radford A., Narasimhan K., Salimans T., et al. Improving language understanding by generative pre-training [Electronic resource] // 2018. URL: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (date of treatment: 08.04.2024).
31. Ren R., Wu D. D., Liu T. Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine [Electronic resource] // IEEE Systems Journal. Vol. 13. № 1. 2018. p. 760–770. URL: https://www.researchgate.net/publication/324048495_Forecasting_Stock_Market_Movement_Direction_Using_Sentiment_Analysis_and_Support_Vector_Machine (date of treatment: 06.03.2024).
32. Snyder B., Barzilay R. Multiple Aspect Ranking using the Good Grief Algorithm [Electronic resource] // Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference. 2007. p. 300-307. URL: https://people.csail.mit.edu/bsnyder/papers/multi_aspect-snyder.pdf (date of treatment: 01.03.2024).
33. Su F., Markert K. From Words to Senses: A Case Study of Subjectivity Recognition [Electronic resource] // Proceedings of the 22nd international conference on computational linguistics (Coling 2008). 2008. p. 825-832.

URL: https://aclanthology.org/C08-1104.pdf (date of treatment: 03.03.2024).

34. Tan K. L., Lee C. P., Lim K. M. Roberta-GRU: a hybrid deep learning model for enhanced sentiment analysis [Electronic resource] // Applied Sciences. 2023. Vol. 13. № 6. p. 3915. URL: https://www.mdpi.com/2076-3417/13/6/3915 (date of treatment: 05.04.2024).
35. Tan K. L., Lee Ch. P., Anbananthen K. S. M., et al. RoBERTa-LSTM: a hybrid model for sentiment analysis with transformer and recurrent neural network [Electronic resource] // IEEE Access. 2022. Vol. 10. p. 21517-21525. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9716923 (date of treatment: 05.04.2024).
36. Turney P. D. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews [Electronic resource] // arXiv preprint cs/0212032. 2002. URL: https://aclanthology.org/P02-1053.pdf (date of treatment: 28.02.2024).
37. Valitutti A., Strapparava C., Stock O. Developing Affective Lexical Resources [Electronic resource] // PsychNology J. 2004. Vol. 2. № 1. p. 61-83. URL: https://www.researchgate.net/publication/220168839_Developing_Affective_Lexical_Resources (date of treatment: 12.03.2024).
38. Vaswani A., Shazeer N., Parmar N., et al. Attention is all you need [Electronic resource] // Advances in neural information processing systems. 2017. Vol. 30. URL: https://arxiv.org/pdf/1706.03762 (date of treatment: 02.04.2024).
39. Wiebe J. M. Tracking Point of View in Narrative [Electronic resource] // arXiv preprint cmp-lg/9407019. 1994. URL: https://aclanthology.org/J94-2004.pdf (date of treatment: 24.02.2024).

40. Wiebe J. M., Bruce R. F., O'Hara T. P. Development and Use of a Gold-Standard Data Set for Subjectivity Classifications [Electronic resource] // Proceedings of the 37th annual meeting of the Association for Computational Linguistics. 1999. p. 246-253. URL: https://aclanthology.org/P99-1032.pdf (date of treatment: 25.02.2024).
41. Yang Z., Dai Z., Yang Y., et al. XLNet: Generalized Autoregressive Pretraining for Language Understanding [Electronic resource] // Advances in neural information processing systems. 2019. Vol. 32. URL: https://arxiv.org/abs/1906.08237 (date of treatment: 08.04.2024).
42. Zhao L., Li L., Zheng X. A BERT based Sentiment Analysis and Key Entity Detection Approach for Online Financial Texts [Electronic resource] // 2021 IEEE 24th International conference on computer supported cooperative work in design (CSCWD). 2021. p. 1233-1238. URL: https://arxiv.org/ftp/arxiv/papers/2001/2001.05326.pdf (date of treatment: 02.04.2024).
43. Двойникова А. А., Карпов А. А. Аналитический обзор подходов к распознаванию тональности русскоязычных текстовых данных [Электронный ресурс]// Информационно-управляющие системы. 2020. №. 4 (107). с. 20-30. URL: https://cyberleninka.ru/article/n/analiticheskiy-obzor-podhodov-k-raspoznavaniyu-tonalnosti-russkoyazychnyh-tekstovyh-dannyh (дата обращения: 24.02.2024).
44. Михайличенко А. А. Аналитический обзор методов оценки качества алгоритмов классификации в задачах машинного обучения [Электронный ресурс] // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. 2022. № 4 (311). с. 52-59. URL: https://cyberleninka.ru/article/n/analiticheskiy-obzor-metodov-otsenki-

kachestva-algoritmov-klassifikatsii-v-zadachah-mashinnogo-obucheniya (дата обращения: 06.03.2024).

45. Новиков С. В., Абдулнагимов А. И., Агеев Г. К. Нейросетевые технологии при полунатуральном моделировании в цифровом образовательном процессе университета 4.0 [Электронный ресурс] // Вестник Уфимского государственного авиационного технического университета. 2021. Т. 25. № 3 (93). с. 42-49. URL: https://cyberleninka.ru/article/n/neyrosetevye-tehnologii-pri-polunaturnom-modelirovanii-v-tsifrovom-obrazovatelnom-protsesse-universiteta-4-0 (дата обращения: 27.03.2024).
46. Рапаков Г. Г., Горбунов В. А., Дианов С. В., и др. Исследование LSTM-нейросетевого подхода при моделировании временных рядов [Электронный ресурс] // Вестник Череповецкого государственного университета. 2023. № 3 (114). с. 47-54. URL: https://cyberleninka.ru/article/n/issledovanie-lstm-neyrosetevogo-podhoda-pri-modelirovanii-vremennyh-ryadov (дата обращения: 27.03.2024).
47. Самигулин Т. Р., Джурабаев Анвар А. Э. У. Анализ тональности текста методами машинного обучения [Электронный ресурс] // Научный результат. Информационные технологии. 2021. Т. 6. № 1. с. 55-62. URL: https://cyberleninka.ru/article/n/analiz-tonalnosti-teksta-metodami-mashinnogo-obucheniya (дата обращения: 16.03.2024).
48. Семина Т. А. Анализ тональности текста: современные подходы и существующие проблемы [Электронный ресурс] // Социальные и гуманитарные науки. Отечественная и зарубежная литература. Сер. 6, Языкознание: Реферативный журнал. 2020. № 4. с. 47-64. URL: https://cyberleninka.ru/article/n/analiz-tonalnosti-teksta-sovremennye-podhody-i-suschestvuyuschie-problemy (дата обращения: 17.03.2024).

49. Трофимов А. Парсер сайта 2GIS для сбора адресов и контактов предприятий России и стран СНГ [Электронный ресурс] // GitHub.com. 2022. URL: https://github.com/interlark/parser-2gis?tab=readme-ov-file (дата обращения: 05.05.2024)
50. Шолле Ф. Глубокое обучение на Python. СПб.: Питер, 2018. 400 с.
51. «Яндекс» представил YandexGPT 3 [Электронный ресурс] // Информационное агентство ТАСС. 2024. 28 марта. URL: https://tass.ru/ekonomika/20380549 (дата обращения: 15.04.2024).