МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра математической кибернетики и компьютерных наук

СРАВНЕНИЕ РАБОТЫ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ РАСПОЗНАВАНИЯ ЭМОЦИЙ ПО ТЕКСТУ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Захаровой Юлии Алексеевны

Научный руководитель
зав.каф.техн.пр., к.ф.-м.н, доцент _____ И. А. Батраева

Заведующий кафедрой
к.ф.-м. н., доцент _____ С. В. Миронов

Саратов 2024

ВВЕДЕНИЕ

В современном мире, характеризующемся широким использованием цифровых технологий и интернета, объем текстовой информации постоянно растет. В социальных сетях, блогах, форумах и других онлайн-платформах ежедневно публикуются миллионы текстовых сообщений, в которых люди выражают свои мысли, чувства и эмоции. Распознавание эмоций по тексту стало важной задачей, так как позволяет анализировать и интерпретировать эмоциональное состояние пользователей.

Целью данной работы является сравнительный анализ различных алгоритмов машинного обучения для распознавания эмоций по тексту. В рамках исследования будут решены следующие задачи:

- изучение и выбор алгоритмов машинного обучения, наиболее подходящих для задачи распознавания эмоций;
- подготовка и предобработка текстовых данных для обучения моделей;
- обучение и тестирование выбранных моделей на подготовленных данных;
- сравнительный анализ точности и эффективности работы алгоритмов.

Структура и объем работы. Для решения поставленных задач выполнена выпускная квалификационная работа, включающая в себя введение, 3 основные главы, заключение, список использованных источников из 20 наименований и 2 приложения. Работа изложена на 67 страницах, содержит 36 рисунков.

Первая глава имеет название «Анализ предметной области» и содержит основную информацию о подходах к решению задачи распознавания эмоций по тексту: определения, методы машинного обучения, подходящие для ешения, метрики оценки.

Вторая глава имеет название «Программная реализация», данная глава содержит подробное описание процесса выполнения работы.

Третья глава под названием «Сравнительный анализ» содержит информацию о результатах выполнения работы, а также сводную диаграмму.

Выпускная квалификационная работа заканчивается заключением, списком использованных источников, а также приложениями А-Б с описанием исходных данных и кодом.

Краткое содержание работы

В первом разделе «Анализ предметной области» рассказано о проблемах распознавания эмоций по тексту, а также приведены теоретические данные, необходимые для работы.

Основные трудности при работе с эмоциями в тексте возникают по нескольким причинам. Это могут быть лингвистические, технологические или психологические проблемы.

Для начала работы с текстом необходимо провести его предобработку. Задачи предобработки решает область искусственного интеллекта, называемая обработкой естественного языка (NLP). В работе в рамках предобработки были выполнены следующие шаги:

1. Загрузка данных

2. Удаление непоказательных признаков

3. Токенизация

4. Лемматизация

5. Удаление стоп-слов

6. Балансировка

7. Эмбеддинг

8. Разделение данных.

Далее в работе подробно описаны алгоритмы.

Для балансировки использовался метод оверсэмплинга, для эмбеддинга - модель BERT.

В разделе **«Машинное обучение»** представлены наиболее подходящие для решения задачи распознавания эмоций алгоритмы. Для сравнения были выбраны следующие алгоритмы:

— Гауссовский наивный Байес:

Гауссовский наивный байесовский классификатор хорошо подходит для задач анализа текста, так как предполагает, что каждый признак (слово в нашем случае) влияет на класс (эмоцию) независимо от других признаков, что часто оправдано в текстовых данных. Кроме того, данный метод

хорошо работает с категориальными данными, что делает его подходящим выбором для анализа текста с учетом эмоциональной окраски.

— Логистическая регрессия

Простота в реализации и интерпретации делает ее привлекательным выбором, особенно когда важны как точность предсказаний, так и понимание влияния каждого признака на результат.

— Метод опорных векторов

SVM ищет оптимальную гиперплоскость, которая максимально разделяет классы, что делает его хорошим выбором для задачи распознавания эмоций по тексту.

— Метод k ближайших соседей

В контексте задачи распознавания эмоций по тексту, где важно учитывать контекст и смысл текста, kNN может быть эффективным, так как он использует схожесть текстов для классификации.

— Сверточные нейронные сети

CNN хорошо справляются с извлечением локальных иерархических признаков из текстовых данных, что позволяет им автоматически выявлять особенности и контекст эмоций в тексте.

— Рекуррентные нейронные сети

Рекуррентные нейронные сети (RNN) были выбраны для решения задачи распознавания эмоций по тексту из-за их способности учитывать последовательность слов в тексте. Это особенно важно для анализа эмоций, так как эмоциональный контекст часто зависит от последовательности слов и их взаимодействия.

Для оценки результатов работы описаны метрики:

— Точность (Accuracy)

Определяет долю правильно классифицированных наблюдений от общего числа наблюдений.

— Точносить (Precision)

Определяет долю правильно предсказанных положительных наблюдений из всех предсказанных положительных.

— Полнота (Recall)

Определяет долю правильно предсказанных положительных наблюдений

из всех реальных положительных наблюдений.

— F-мера (F-score)

Гармоническое среднее точности и полноты, которое позволяет найти баланс между этими метриками.

Далее описаны технологии, подходящие для решения поставленных задач. В их число входят такие языки программирования как Python, R, Java, Scala, Julia. Из представленных языков был выбран Python по нескольким причинам:

1. Простота и понятность кода.

2. Богатая экосистема библиотек.

3. Широкое применение.

4. Гибкость и масштабируемость.

В качестве среды разработки был выбран Visual Studio Code, в частность Jupiter Notebook.

Также для работы с алгоритмами машинного обучения, отображения графиков и обработки текста необходимо использовать библиотеки. Библиотеки, используемые в работе:

— Pandas

Pandas предоставляет структуры данных и функциональность для эффективной работы с данными, особенно для анализа данных. Основными структурами данных в Pandas являются Series (одномерный массив с метками) и DataFrame (двумерная таблица данных). Pandas позволяет выполнять операции с данными, такие как слияние, группировка, фильтрация и агрегация, а также работать с пропущенными данными.

— NumPy

NumPy предоставляет поддержку для многомерных массивов и матриц, а также математических функций для работы с ними. NumPy обеспечивает эффективные вычисления с массивами данных, что делает его основой для многих библиотек машинного обучения. Он также предоставляет функциональность для работы с линейной алгеброй, случайными числами и другими математическими операциями.

— Matplotlib

Matplotlib позволяет создавать различные типы графиков и визуализаций данных. Matplotlib обладает широким спектром возможностей для создания качественных графиков, включая линейные графики, гистограммы, диаграммы рассеяния и тепловые карты. Он также поддерживает кастомизацию графиков, что позволяет создавать информативные и привлекательные визуализации.

— Seaborn

Seaborn является надстройкой над Matplotlib и предоставляет более высокоуровневый интерфейс для создания статистических графиков. Seaborn упрощает создание сложных статистических графиков, таких как boxplot, violinplot, pairplot и т. д. Он также предоставляет удобные функции для работы с данными, такие как автоматическая группировка и агрегация данных.

— NLTK

NLTK предоставляет инструменты для обработки естественного языка, такие как токенизация, лемматизация, стемминг и морфологический анализ. NLTK содержит большое количество корпусов и грамматик для анализа текста на различных языках. Он также предоставляет функции для работы с частями речи, синтаксическим анализом и машинным обучением в контексте обработки текста.

— tqdm

tqdm используется для создания прогресс-баров в циклах, чтобы отслеживать прогресс выполнения задачи. tqdm позволяет легко добавить индикацию прогресса в ваш код, что делает его более информативным и удобным для пользователей. Он поддерживает различные стили и настраиваемый интерфейс для соответствия вашим потребностям.

— TensorFlow

TensorFlow является одной из самых популярных библиотек для создания и обучения моделей машинного обучения и глубокого обучения. Он поддерживает различные уровни абстракции, что позволяет разработчикам выбирать подходящий уровень сложности для своего проекта. TensorFlow также предоставляет возможности для распределенного обучения и экспорта моделей для использования в различных окружениях.

Следом идет раздел «Программная реализация». В нем рассказано о процессе разработки приложения и приведены конкретные этапы реализации.

Данные для исследования были взяты с сайта kaggle.com. Они представлены в виде таблицы в csv-файле. Он содержит 3 столбца:

1. Идентификатор записи

2. Текст

3. Номер соответствующей эмоции. В исследовании участвуют 6 эмоций, считающихся основными:

a) Грусть

b) Радость

c) Любовь

d) Гнев

e) Страх

f) Удивление.

Данные для исследования были взяты с сайта kaggle.com. Они представлены в виде таблицы в csv-файле.

Для качественной предобработки данных были выполнены следующие операции: токенизация, лемматизация и удаление стоп-слов. Эти этапы необходимы для приведения текста к более удобному для анализа виду, что улучшает качество моделей машинного обучения. Все необходимые функции для выполнения этих операций были реализованы с помощью библиотеки NLTK (Natural Language Toolkit).

Балансировка данных необходима для устранения дисбаланса классов в датасете, когда некоторые классы представлены значительно больше или меньше других. Это важно, потому что модели машинного обучения могут плохо справляться с предсказанием редких классов, если обучены на несбалансированных данных. Балансировка позволяет улучшить точность и надежность модели для всех классов, обеспечивая равное внимание каждому классу в процессе обучения.

Эмбеддинг (embedding) в контексте обработки естественного языка представляет собой способ представления слов или текста в виде векторов в многомерном пространстве. Эмбеддинги позволяют модели машинного

обучения понимать семантические отношения между словами и использовать эту информацию для более эффективного анализа и обработки текста.

На этапе выполнения эмбеддинга текста возникает сложность, связанная со временем выполнения программы. На больших объемах данных применение алгоритма может достигать нескольких часов. Чтобы сократить время выполнения и уменьшить нагрузку на устройство, применен алгоритм обработки данных по частям и сохранение промежуточных результатов, так называемая пакетная обработка.

Также данные необходимо разделить на выборки (тестовую и обучающую). Эта операция легко производится путем подключения библиотеки sklearn, модуль train_test_split, который содержит функцию разделения данных в отношении 80/20.

Далее для каждого из шести алгоритмов представлены примеры реализации, а также рисунки, отображающие вывод после выполнения. Для рекуррентных и сверточных нейронных сетей в работе представлены графики потерь и точности на обучении и валидации.

В третьем разделе приведен сравнительный анализ различных алгоритмов машинного обучения, применяемых для задачи распознавания эмоций по тексту. Этот анализ направлен на выявление сильных и слабых сторон каждого метода, а также на оценку их эффективности и применимости в различных контекстах. Сравнительный анализ позволяет определить наиболее подходящий алгоритм для данной задачи, учитывая такие параметры, как точность, временная сложность, использование ресурсов и другие метрики производительности. Для проведения экспериментального исследования эффективности алгоритмов машинного обучения в контексте задачи распознавания эмоций по тексту предполагается осуществление тестирования на варьирующемся объеме данных. Конкретно, алгоритмы запущены на различных размерах обучающих наборов, включая 3000, 10000 и 100000 текстовых экземпляров.

В данном разделе показаны значения метрик для каждого алгоритма для каждого из трех вариантов объемов данных. В заключение, в тексте работы содержится сводная сравнительная диаграмма, демонстрирующая различия при выполнении работы.

ЗАКЛЮЧЕНИЕ

В данной дипломной работе было проведено сравнительное исследование алгоритмов машинного обучения для задачи распознавания эмоций по тексту. В ходе выполнения работы были рассмотрены и протестированы следующие алгоритмы: гауссовский наивный байес, логистическая регрессия, метод опорных векторов (SVM), метод ближайших соседей (k-NN), свёрточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN).

Основные результаты и выводы работы позволяют сделать следующие утверждения:

- Лучшие алгоритмы: Логистическая регрессия, SVC и нейронные сети (CNN, RNN) показывают наивысшую производительность по всем метрикам.
- Низкая производительность: Наивный Байес и kNN демонстрируют низкие и средние результаты соответственно.
- Объем данных: Все алгоритмы, кроме Наивного Байеса, улучшаются с увеличением объема данных.
- Рекомендации: Для распознавания эмоций по тексту предпочтительны Логистическая регрессия, SVC и нейронные сети, особенно на больших наборах данных.

В процессе выполнения работы была создана система распознавания эмоций по тексту с использованием различных алгоритмов машинного обучения. Эта система может быть применена в различных областях, таких как анализ настроений в социальных сетях, автоматическое модераторство комментариев, поддержка клиентов и многое другое.

Таким образом, поставленные задачи были решены и цель достигнута.

Основные источники информации:

1. Дворников, С. В. Распознавание эмоций в текстовом сообщении [Электронный ресурс] / С. В. Дворников // Научно-образовательный журнал для студентов и преподавателей "StudNet". — 2021.

2. Ульянов, Н. В. Введение в анализ данных [Электронный ресурс] / Н. В.

Ульянов, Ш. А. Ахмедова // Актуальные проблемы авиации и космонавтики.

3. Михайличенко, А. А. Аналитический обзор методов оценки качества алгоритмов классификации в задачах машинного обучения [Электронный ресурс] / А. А. Михайличенко // Научный журнал "Вестник АГУ".

4. Давыдов, А. В. Сравнение различных языков программирования, применяемых в машинном обучении [Электронный ресурс] / А. В. Давыдов, А. К. Жусупова, О. С. Салыкова // Международный научный журнал "Вестник науки".

5. Зулунов, Р. М. Использование python для искусственного интеллекта и машинного обучения [Электронный ресурс] / Р. М. Зулунов, В. Н. Солиев // Электронный научный журнал "Потомки Аль-Фаргани".

6. Акжолов, Р. К. Предобработка текста для решения задач nlp [Электронный ресурс] / Р. К. Акжолов, А. В. Верига // Международный научный журнал "Вестник науки".