СОДЕРЖАНИЕ

Введение1

1. Теоретическая часть2
   1. Терминология по проекту и глоссарий2
   2. Стек технологий2
2. Практическая часть3
   1. Архитектура ПО3
   2. Разработка проекта3

Заключение4

Список литературы5

ВВЕДЕНИЕ

Современные технологии коммуникации играют ключевую роль в повседневной жизни, бизнесе и социальном взаимодействии. Однако просто отправки сообщений уже недостаточно - важно понимать эмоциональную окраску передаваемой информации. Сентимент-анализ (анализ тональности текста) позволяет автоматически определять эмоциональную нагрузку сообщений, что может быть полезно для улучшения качества общения, анализа обратной связи и принятия решений на основе данных.

**Актуальность** проекта обусловлена растущим спросом на инструменты, способные не только передавать текстовые сообщения, но и анализировать их эмоциональную составляющую. Такие решения могут применяться в службах поддержки, соцсетях, бизнес-коммуникациях и психологических исследованиях.

**Цель проекта** - разработка десктопного приложения для отправки сообщений с функцией сентимент-анализа, позволяющего пользователям оценивать эмоциональную окраску текста в реальном времени.

**Задачи проекта:**

* Изучить принципы работы сентимент-анализа и существующие решения.
* Разработать архитектуру десктопного приложения с поддержкой отправки сообщений.
* Реализовать алгоритм анализа тональности текста.
* Обеспечить удобный интерфейс для взаимодействия с пользователем.
* Протестировать приложение на точность определения эмоциональной окраски текста.

**Объект исследования** - процесс автоматического анализа тональности текста и его интеграция в системы обмена сообщениями.

**Предмет исследования** - методы и алгоритмы сентимент-анализа, применяемые для оценки эмоциональной окраски текстовых сообщений в реальном времени.

Данный проект представляет собой практическую реализацию современных методов обработки естественного языка и может быть полезен как для индивидуального использования, так и для интеграции в бизнес-процессы, требующие анализа эмоциональной реакции пользователей.

1. Теоретическая часть
   1. Терминология по проекту и глоссарий

Сентимент анализ – метод обработки текстовых данных, направленный на определение эмоциональной окраски высказываний. Он классифицирует сообщения как положительные, отрицательные или нейтральные с использованием алгоритмов обработки естественного языка и методов машинного обучения

* Машинное обучение

Машинное обучение представляет собой область искусственного интеллекта, позволяющую программам самостоятельно находить закономерности в больших объемах данных и делать предсказания. В контексте сентимент анализа алгоритмы обучаются на размеченных текстах, чтобы впоследствии автоматически определять эмоциональную тональность новых сообщений

* Обработка естественного языка (NLP)

Обработка естественного языка включает технологии и методы, позволяющие компьютеру понимать, анализировать и интерпретировать человеческий язык. Ключевые этапы NLP – предобработка текста, токенизация, стемминг или лемматизация и синтаксический анализ

* Токенизация

Токенизация – процесс разбиения текста на отдельные элементы (слова, знаки препинания, фразы). Этот этап является необходимым для последующего анализа структуры и семантики текста

* Классификация

Классификация – метод распознавания и группировки объектов (в данном случае текстовых сообщений) по предварительно заданным категориям, что в сентимент анализе позволяет определять позитивные, негативные или нейтральные высказывания

* Глубокое обучение

Глубокое обучение – подвид машинного обучения, основанный на использовании нейронных сетей с несколькими слоями, позволяющих выявлять сложные закономерности в данных. Применяется для улучшения точности анализа эмоциональной окраски текста

* API

API (интерфейс программирования приложений) – набор правил и протоколов, с помощью которых обеспечивается обмен данными между различными компонентами приложения. В рассматриваемом решении API обеспечивает интеграцию модулей отправки сообщений и анализа их содержания

* Фронтенд и Бекенд

Фронтенд – клиентская часть приложения, обеспечивающая визуальное отображение и взаимодействие с пользователем. Бекенд – серверная часть, реализующая бизнес-логику, обработку данных, работу с базой данных и интеграцию модулей, включая сентимент анализ

* UI и UX

UI (пользовательский интерфейс) относится к визуальному оформлению приложения, а UX (пользовательский опыт) – к удобству использования и удовлетворенности пользователя от взаимодействия с продуктом. Хорошее сочетание UI и UX является критически важным для успешного восприятия приложения конечными пользователями

* Предварительная обработка данных

Предварительная обработка данных (Data Preprocessing) включает очистку, нормализацию текста, удаление стоп-слов, пунктуации и лишних символов. Этот этап помогает подготовить данные к дальнейшему анализу и снижает шум, повышая точность классификации

* Выделение признаков

Выделение признаков – процесс преобразования текста в формат, пригодный для машинного обучения. Одними из популярных методов являются мешок слов (Bag of Words) и TF-IDF (Term Frequency-Inverse Document Frequency), которые помогают количественно представить текстовые данные

* Модель наивного Байеса

Модель наивного Байеса – простой и эффективный алгоритм классификации, часто используемый для решения задач сентимент анализа благодаря своей способности работать с текстовыми данными и учитывать частоту появления слов

* Нейронные сети

Нейронные сети – вычислительные модели, имитирующие работу биологических нейронов, применяемые для сложных задач классификации и распознавания образов. В сентимент анализе их используют для выявления скрытых закономерностей в языке

* Обратная связь и валидация

Обратная связь – механизм, с помощью которого приложение или система получает данные о корректности своих действий (например, правильности определения тональности). Валидация – этап проверки точности работы модели, включающий сравнение предсказаний алгоритма с размеченными тестовыми данными

* Интеграция микросервисов

Интеграция микросервисов – подход к разработке, когда функциональность приложения разбивается на независимые сервисы, способные взаимодействовать друг с другом посредством API. Это повышает масштабируемость и гибкость разработки, позволяя улучшать или обновлять отдельные компоненты без влияния на всю систему.

* 1. Стек технологий
* Язык программирования

Использование Python для реализации серверной логики, алгоритмов обработки естественного языка и машинного обучения. Дополнительно можно применять JavaScript (или TypeScript) при создании клиентской части

* Фреймворк для серверной разработки

FastAPI или Flask для разработки REST API, обеспечивающих обмен данными между клиентской частью и сервером. Эти фреймворки позволяют быстро разворачивать и масштабировать приложение

* Библиотеки для обработки естественного языка

NLTK, spaCy или библиотеки от HuggingFace (например, Transformers) – для обработки текстов, токенизации, лемматизации и анализа тональности сообщений. Они позволяют интегрировать готовые модели сентимент анализа или обучить свои

* Фреймворки и библиотеки машинного обучения

scikit-learn для классического машинного обучения (например, наивный Байес, логистическая регрессия) и TensorFlow или PyTorch для глубокого обучения. В зависимости от сложности и требований к точности можно выбирать подходящий инструмент

* База данных

PostgreSQL или MongoDB для хранения информации об отправленных сообщениях, пользователях и результатах анализа. Реляционные базы данных подходят при строгой схеме данных, а NoSQL – при гибких и динамичных структурах

* Асинхронная обработка задач

Celery с брокером сообщений, таким как RabbitMQ или Redis, позволит организовать фоновые задачи для анализа сообщений, отправки уведомлений и интеграции с внешними сервисами

* Фронтенд

React или Vue.js для создания современного и отзывчивого пользовательского интерфейса. Использование библиотек UI (например, Material-UI или Vuetify) улучшит визуальное представление и взаимодействие с пользователем

* Система контроля версий и CI/CD

Git для контроля версий кода, GitLab CI, Jenkins или GitHub Actions для организации процессов непрерывной интеграции и доставки, позволяющих быстро тестировать и деплоить обновления

* Мониторинг и логирование

Инструменты вроде Prometheus для мониторинга, Grafana для визуализации метрик и ELK-стека (Elasticsearch, Logstash, Kibana) для централизованного логирования, что поможет оперативно обнаруживать и устранять проблемы.

1. Практическая часть
   1. Архитектура ПО
   2. Разработка проекта

Заключение