НПОУ «ЯКУТСКИЙ КОЛЛЕДЖ ИННОВАЦИОННЫХ ТЕХНОЛОГИЙ» (НПОУ «ЯКИТ»)

Отделение информационных технологий и туризма

Специальность 09.02.07 «Информационные системы и программирования»

ПРОЕКТНАЯ РАБОТА

по дисциплине: Языки программирования

по теме:

Создание десктопного приложения для семантического анализа русского языка с использованием библиотеки Hugging Face Transformers на Python

Исполнитель студент гр. КИСП-23(1)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Л.П. Лебедева

ПАВ

подпись, дата

Руководитель,

преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.А.Фёдоров

подпись, дата

Якутск, 2025

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc194747538)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc194747539)

[1.1 Терминология 5](#_Toc194747540)

[1.2 Распределение ролей и работы 6](#_Toc194747541)

[1.3 Стек технологий 7](#_Toc194747542)

[2.ПРАКТИЧЕСКАЯ ЧАСТЬ 8](#_Toc194747543)

[2.1. Архитектура приложения 8](#_Toc194747544)

[2.2.Разработка проекта по ролям 9](#_Toc194747545)

[2.3.Контроль выполнения плана 10](#_Toc194747546)

[ЗАКЛЮЧЕНИЕ 11](#_Toc194747547)

# ВВЕДЕНИЕ

Актуальность создания десктопного приложения для семантического анализа текста обусловлена ростом объема текстовой информации, потребностью в автоматизации анализа и развитием технологий обработки естественного языка, обеспечивающих высокую точность и скорость обработки.

Цель проекта — разработать десктопное приложение для семантического анализа русского языка с использованием библиотеки Hugging Face Transformers.

Задачи проекта:

1. Изучение теории: Исследовать методы и подходы семантического анализа.
2. Выбор технологий: Определить подходящие технологии и инструменты.
3. Разработка интерфейса: Создать удобный графический интерфейс для ввода текста и вывода результатов.
4. Имплементация анализа: Реализовать семантический анализ с применением предобученных моделей.
5. Документация: Подготовить инструкцию и представить результаты проекта.

Объектом исследования является процесс семантического анализа текстов на русском языке, а также применяемые в этом процессе алгоритмы и технологии обработки естественного языка.

Предметом исследования выступает десктопное приложение, реализующее семантический анализ текста с использованием библиотеки Hugging Face Transformers.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ
   1. Терминология

1. Библиотека Hugging Face Transformers

* Transformers: Нейронные сети глубокого обучения, используемые для обработки естественного языка. Они применяются для различных задач, таких как классификация текста, перевод, генерация текста и многое другое.
* Tokenizer: Преобразовывает входной текст в числовые представления (токены), которые могут быть использованы моделью.
* Pipeline: Высокоуровневый API, который упрощает использование моделей Transformers для конкретных задач, таких как классификация текста, вопросно-ответная система и т.д.
* Model: Предварительно обученная модель, такая как BERT, RoBERTa, GPT и другие, которая используется для обработки текста.

2. Семантический анализ

* Семантический анализ: Процесс определения смысла текста путем анализа его содержания и контекста.
* Классификация текста: Задача, при которой текст классифицируется по определенным категориям, таким как позитивный/негативный, тема, жанр и т.д.
* Нейронная сеть: Алгоритм машинного обучения, основанный на биологической структуре мозга, состоящий из слоев нейронов, соединенных между собой.
* Embedding: Представление слова или предложения в виде вектора чисел, которое сохраняет семантические свойства текста.

3. Python и PyQt5

* PyQt5: Библиотека для создания графических интерфейсов на Python. Она основана на библиотеке Qt и предоставляет широкий набор виджетов и инструментов для разработки настольных приложений.
* QApplication: Класс, представляющий основное приложение Qt. Он управляет главным циклом событий и параметрами приложения.
* QWidget: Базовый класс для всех виджетов в Qt. Используется для создания окон и контейнеров.
* QPushButton: Класс, представляющий кнопку в графическом интерфейсе.
* QLabel: Класс, представляющий текстовую метку в графическом интерфейсе.
* QFileDialog: Класс, представляющий диалоговое окно для открытия и сохранения файлов.
* QTextEdit: Класс, представляющий многострочное поле ввода текста.
* QVBoxLayout: Класс, представляющий вертикальное расположение виджетов в контейнере.

4. Работа с файлами и текстовыми данными

* with open(): Контекстный менеджер для работы с файлами. Автоматически закрывает файл после окончания блока.
* read(): Метод чтения содержимого файла.
* encoding: Параметр, указывающий кодировку файла.

5. Другие полезные термины

* GitHub: Платформа для хостинга и совместного использования кода. Часто используется для управления версиями и сотрудничества в команде.
* Debug: Процесс нахождения и исправления ошибок в коде.
* Push: Действие отправки изменений в удалённый репозиторий (например, на GitHub).

Эти термины охватывают основные аспекты разработки десктопного приложения для семантического анализа текста с использованием библиотеки Hugging Face Transformers на Python.

* 1. Распределение ролей и работы

Лебедева Лира Петровна занималась написанием отчета, отвечала за исправление и написание отчета.

План по выполнению работы по части написания отчета:

1. Написание структуры отчета.
2. Написание отчета.
3. Просмотр отчета на наличие ошибок.
4. Исправление ошибок.

Динганорбоев Эрдэм Булатович занимался поиском информации, отвечал за теорию и сбор материала.

План по выполнению работы по части поиска информации:

1. Написание плана работы.
2. Поиск информации.
3. Выделение важных терминов и определений.

* Иннокентьев Влад Александрович занимался программированием, отвечал за репозиторий.

План по выполнению работы по части программирования:

1. Создание основы.
2. Создание UI.

* Сделать главное окно
* Элемент

1. Добавление Hugging Face библиотеки.
2. Написание кода.

Николаев Аслан Анатольевич занимался программированием, отвечал за корректность кода.

План по выполнению работы по части исправления кода:

1. Просмотр кода.
2. Debug кода.
3. Push кода в GitHub.
   1. Стек технологий

Для реализации десктопного приложения были выбраны следующие технологии:

1. Языки программирования:
   * Python: Основной язык для разработки приложения, обладающий богатой экосистемой библиотек для обработки текста.
2. Библиотеки:
   * Hugging Face Transformers: Библиотека, предоставляющая доступ к предобученным моделям для семантического анализа.
   * Tkinter: Библиотека для создания графического пользовательского интерфейса (GUI).
3. API:

В контексте разработки десктопного приложения для семантического анализа русского языка с использованием библиотеки Hugging Face Transformers на Python, API (интерфейсы прикладного программирования) играют важную роль в обеспечении взаимодействия между различными компонентами системы. Вот основная терминология, связанная с API в данном контексте:

1. Hugging Face API

* Transformer Pipeline: Высокоуровневый API, предоставляемый библиотекой Hugging Face, который упрощает использование предобученных моделей для различных задач обработки естественного языка, таких как классификация текста, генерация текста и т.д.
* Tokenizer: Интерфейс, используемый для преобразования текста в токены, которые могут быть интерпретированы моделью.
* Model: Предварительно обученная языковая модель, доступная через API Hugging Face. Примеры включают BERT, RoBERTa и другие.

2. PyQt5 API

* QApplication: API для создания основного объекта приложения Qt, управляющего жизненным циклом и событиями.
* QWidget: API для создания виджетов, являющихся основой графического интерфейса.
* QPushButton: API для создания кнопок, реагирующих на клики пользователя.
* QLabel: API для отображения текста в графическом интерфейсе.
* QFileDialog: API для открытия диалоговых окон для выбора файлов.
* QTextEdit: API для создания полей ввода текста.
* QVBoxLayout: API для организации виджетов вертикально в контейнере.

3. Flask/Django API (если требуется веб-интерфейс)

* Flask/Django View: API для создания представлений, обрабатывающих запросы и возвращающих ответы.
* Request Handling: API для обработки входящих запросов и извлечения данных.
* Response Generation: API для формирования ответов и возврата их клиенту.

4. API для работы с файлами

* open(): Функция для открытия файлов в Python.
* read(): Метод для чтения содержимого файла.
* write(): Метод для записи данных в файл.
* close(): Метод для закрытия открытого файла.

5. API для работы с базой данных (если применимо)

* SQLAlchemy: ORM-фреймворк для работы с реляционными базами данных.
* Session Management: API для управления сессиями базы данных.
* Querying: API для выполнения SQL-запросов.

Эти API предоставляют программистам мощные инструменты для создания надежных и эффективных приложений, обеспечивая взаимодействие между различными компонентами системы и упрощая разработку.

2.ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1. Архитектура ПО

UML Diagram

Sequense UML Diagram

User Flow

Tabel Database

2.2.Разработка проекта по ролям

2.3.Контроль выполнения плана

Проект был направлен на разработку программного обеспечения для семантического анализа текста с использованием предобученной модели из библиотеки *Hugging Face*. Команда состояла из четырех человек, каждый из которых имел свою зону ответственности. Были определены четкие роли и этапы выполнения работы.

**Этапы выполнения**

1. **Подготовка и планирование**
   * Участниками была проведена работа по определению целей и задач проекта, составлению планов работы и созданию структуры отчета.
   * Динганорбоев Эрдэм Булатович разработал план работы и начал поиск необходимой информации.
   * Лебедева Лира Петровна создала структуру отчета и начала его написание.
2. **Проектирование и архитектура**
   * Иннокентьев Влад Александрович приступил к разработке основы проекта и созданию пользовательского интерфейса.
   * Николаев Аслан Анатольевич провел проверку предлагаемой архитектуры и предложил улучшения.
3. **Реализация**
   * Иннокентьев Влад Александрович продолжил работу над реализацией основной логики программы и интеграцией библиотеки *Hugging Face*.
   * Николаев Аслан Анатольевич активно участвовал в процессе отладки и устранения ошибок.
4. **Тестирование и интеграция**
   * Николаев Аслан Анатольевич выполнил тестирование программы и подготовил её к финальному деплою.
   * Вся команда участвовала в интеграции результатов и проведении общего тестирования.
5. **Документация и завершение**
   * Лебедева Лира Петровна завершила оформление отчета, добавив в него документацию по использованию разработанного ПО.
   * Все участники защитили проект перед руководством и подготовили заключительную документацию.

**Анализ текущего состояния**

На данный момент проект находится на стадии завершения. Большинство запланированных задач выполнено, однако остаются некоторые нерешенные проблемы и задачи, которые требуют дополнительного внимания.

* **Проблемы и препятствия**
  + Задержки в процессе разработки возникли из-за сложностей с интеграцией библиотеки *Hugging Face* и необходимостью внесения изменений в архитектуру проекта.
  + Некоторое время было потрачено на устранение ошибок, связанных с обработкой больших объемов текста.
* **Успешность выполнения задач**
  + Динганорбоев Эрдэм Булатович успешно завершил сбор и обработку необходимой информации.
  + Лебедева Лира Петровна выполнила значительную часть работы по подготовке отчета, хотя остается необходимость внести последние правки.
  + Иннокентьев Влад Александрович справился с разработкой основы проекта и интерфейса, но требует дополнительной помощи в завершении некоторых модулей.
  + Николаев Аслан Анатольевич продемонстрировал высокий уровень профессионализма в отладке и тестировании кода, но предстоит еще ряд тестов для полной уверенности в стабильности программы.

**Рекомендации и дальнейшие шаги**

* Необходимо завершить интеграцию оставшихся модулей и провести дополнительные тесты.
* Следует уделить внимание документированию кода и инструкций по эксплуатации программы.
* Рекомендуется пересмотреть процессы коммуникации внутри команды для предотвращения задержек в будущем.

**Заключение**

Проект достиг значительных успехов благодаря совместным усилиям всей команды. Несмотря на возникающие трудности, большинство задач было выполнено в срок. Остаются незначительные доработки, которые планируется завершить в ближайшее время.

ЗАКЛЮЧЕНИЕ

Создание десктопного приложения для семантического анализа текста на русском языке с использованием библиотеки Hugging Face Transformers представляет собой эффективный способ автоматизации анализа текстовой информации. Использование современных методов обработки естественного языка позволяет значительно улучшить качество анализа и расширить возможности взаимодействия с пользователями. Разработка такого приложения может быть полезной в различных областях, включая маркетинг, социологические исследования и автоматизацию клиентского обслуживания.