Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет информатики и

Радиоэлектроники

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №8

по курсу “Естественно-языковой интерфейс интеллектуальных

систем”

Вариант 19

Выполнили:

Студенты гр. 221701 Машкович В. Н.

Юркевич М. С.

Дичковский В. А.

Проверил: Крапивин Ю. Б.

Минск

2025

**Цель работы:**

* изучить и отработать практические навыки применения методов автоматического распознавания языка текстовых документов.

**Задание:**

Разработать систему, удовлетворяющую следующим требованиям:

* Языки – русский, английский
* Формат документа – HTML
* Методы – частотных слов, алфавитный, нейросетевой
* на входе – текстовые документы одинакового размера (например, 1 страница формата А4), содержащие тексты на естественных языках согласно варианту;
* на выходе – активная ссылка на документ, и результат идентификации отдельного текста – язык текста; сводная статистика по всем текстам из тестовой коллекции.
* наличие средств сохранения в файл и распечатки полученной на выходе информации;
* интерфейс системы - доступный для пользователей любого уровня, содержащий help-средства работы с программой.

**Используемые библиотеки:**

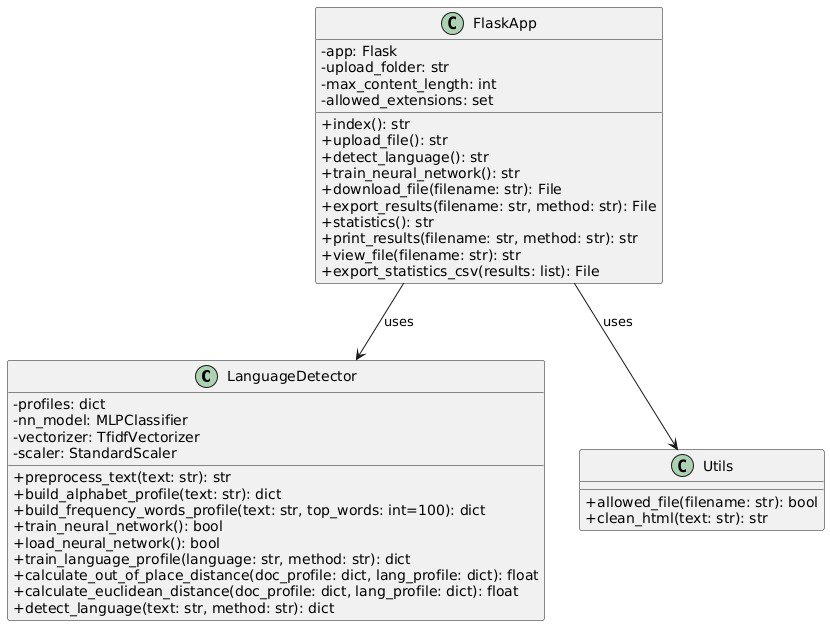
* Flask (v3.1.2) — микрофреймворк для создания веб-приложений на Python. Использовался для построения backend-части приложения, обеспечивая роутинг, обработку HTTP-запросов, рендеринг HTML-шаблонов, работу с сессиями для хранения истории поиска и функциональность загрузки файлов.
* scikit-learn — библиотека машинного обучения. Библиотека предоставила три ключевых компонента: MLPClassifier для реализации нейронной сети с архитектурой 100-50 нейронов, TfidfVectorizer для преобразования текстов в числовые векторы на основе TF-IDF метрик, и StandardScaler для нормализации данных перед подачей в нейросеть.
* joblib — сохраняет результаты обучения для повторного использования.
* re и collections — выполняют предобработку текста. Модуль re с помощью регулярных выражений эффективно очищал текст от HTML-тегов и лишней пунктуации, в то время как Collections.Counter предоставлял удобный и производительный способ подсчета частот слов и символов для построения языковых профилей.
* NumPy — работал на низком уровне вычислений, обеспечивая эффективные операции с многомерными массивами.

**Информация о тестовой коллекции документов:**

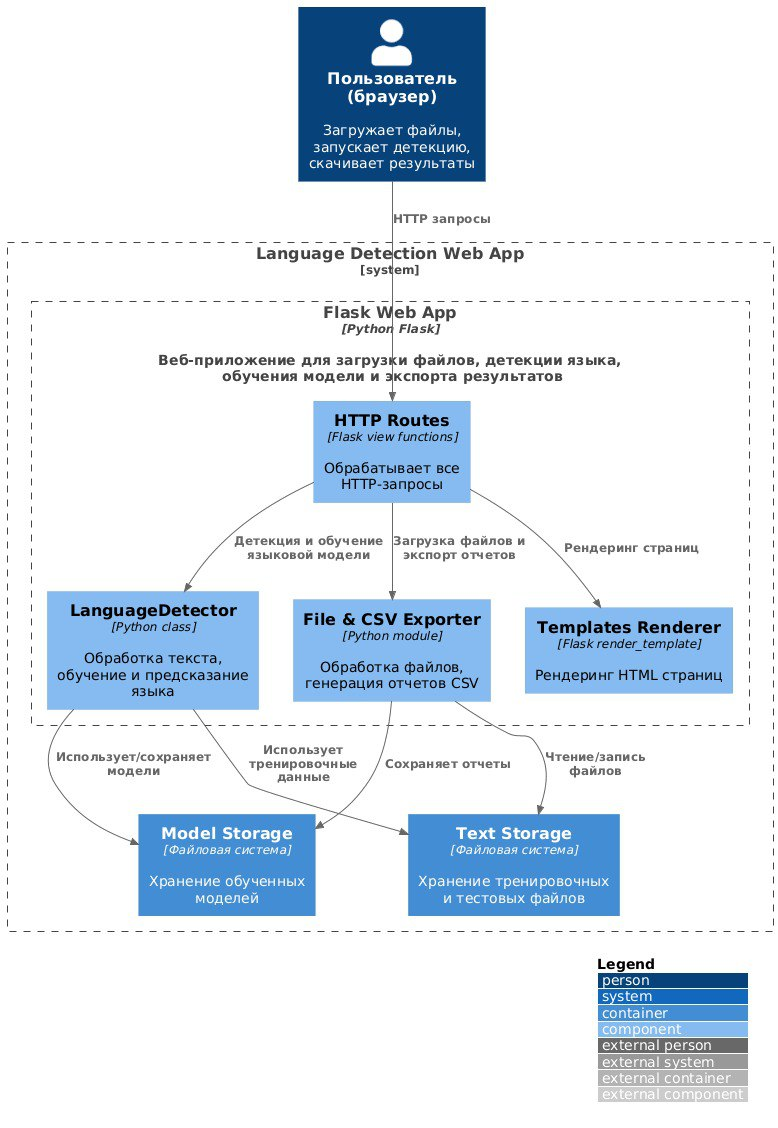
Тестовая коллекция была сформирована на основе многоязычных текстов из Wikipedia. Для обучения и тестирования использовались тренировочные данные: тексты на русском и английском языках, размещённые в папках train/ru и train/en, представляющий собой HTML-страницы из Википедии (например, ru\_5.html — статья о Филиппе Торретоне на русском языке).

**Структурно-функциональная схема разработанного приложения**

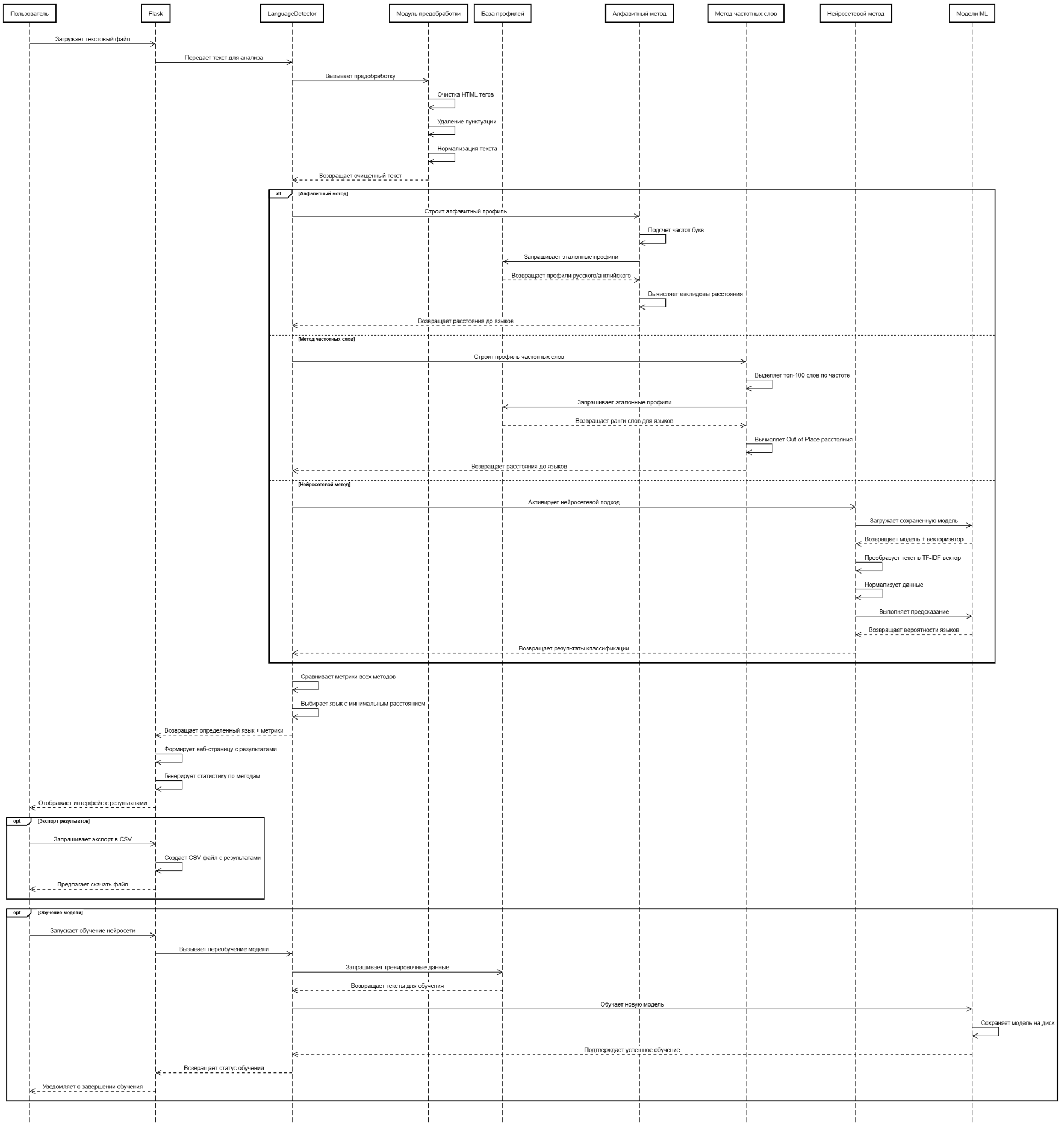
**Диаграмма классов системы**



**Диаграмма компонентов**

****

**Диаграмма последовательности**

****

**Описание структур данных для хранения информации**

Входные данные:

* Текст в виде строки.
* Файлы в формате HTML.

Выходные данные:

Результат определения языка:

{

'detected\_language': 'ru', # идентификатор языка

'distances': {'ru': 0.15, 'en': 0.89}, # метрики расстояний

'method': 'frequency\_words', # использованный метод

'probabilities': {...} # вероятности (для нейросети)

}

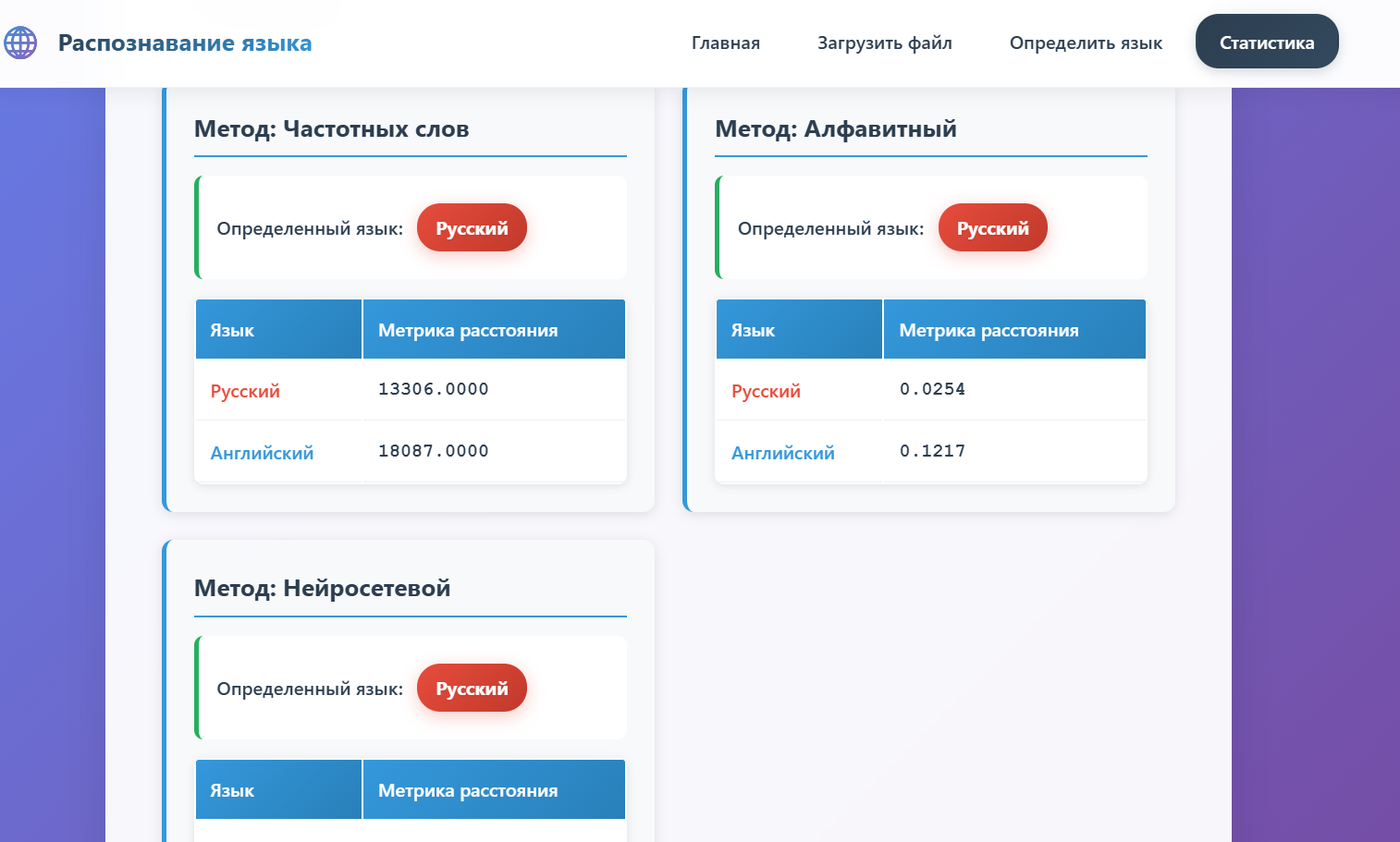
Профили языков:

* Алфавитный метод: словарь {буква: частота}.
* Метод частотных слов: словарь {слово: ранг}.
* Нейросетевой метод: TF-IDF векторы + веса MLP.

**Алгоритмы определения языка:**

1. Предварительная обработка текста
   1. Начало
   2. Удаление HTML-тегов
   3. Приведение к нижнему регистру
   4. Удаление пунктуации и специальных символов
   5. Нормализация пробелов
   6. Конец
2. Метод частотных слов
   1. Начало
   2. Извлечь все слова длиной ≥ 2 символов из текста
   3. Подсчитать частоту каждого слова
   4. Выбрать 100 самых частых слов
   5. Присвоить ранги (1 - самое частое, 2 - следующее и т.д.)
   6. Сравнить с языковыми профилями:
      1. Для русского и английского созданы эталонные профили
      2. Профиль содержит ранги наиболее частых слов
   7. Вычислить метрику Out-of-Place Distance:
      1. Для каждого слова в тексте найти его позицию в эталонном профиле
      2. Если слова нет в профиле - штраф = максимальная\_длина + 1
      3. Суммировать разницы позиций
   8. Язык с минимальным расстоянием считается определенным
   9. Конец
3. Алфавитный метод
   1. Начало
   2. Извлечь все буквы из текста (игнорируя цифры и символы)
   3. Подсчитать частоту каждой буквы
   4. Нормализовать частоты (сумма = 1)
   5. Создать профиль документа
   6. Сравнить с эталонными профилями языков:
      1. Русский профиль: частоты букв русского алфавита
      2. Английский профиль: частоты букв английского алфавита
   7. Вычислить евклидово расстояние между профилями
   8. Язык с минимальным расстоянием считается определенным
   9. Конец
4. Нейросетевой метод
   1. Начало
   2. Обучение модели:
      1. Сбор тренировочных данных из папок train/ru и train/en
      2. Векторизация текстов с помощью
      3. Создание признаков: 1000 наиболее значимых слов и биграмм
      4. Масштабирование признаков StandardScaler
      5. Обучение MLPClassifier (многослойный перцептрон):
         1. Архитектура: 100 → 50 нейронов
         2. Функция активации: ReLU
         3. Алгоритм оптимизации: Adam
   3. Классификация текста:
      1. Преобразование текста в вектор
      2. Масштабирование признаков
      3. Подача в нейронную сеть
      4. Получение вероятностей принадлежности к языкам
      5. Выбор языка с максимальной вероятностью
   4. Конец

**Результаты оценки**



*рис. 2.1 - визуализация метрик*

1. Метод частотных слов

Преимущества:

* Быстрая обработка (50-150 мс)
* Низкое потребление памяти
* Хорошая точность для длинных текстов

Недостатки:

* Точность падает на коротких текстах (< 50 слов)
* Зависит от тематики текста

2. Алфавитный метод

Преимущества:

* Максимальная скорость (20-80 мс)
* Минимальное потребление ресурсов
* Устойчив к тематике текста

Недостатки:

* Наименьшая точность среди методов
* Проблемы с текстами смешанного содержания

3. Нейросетевой метод

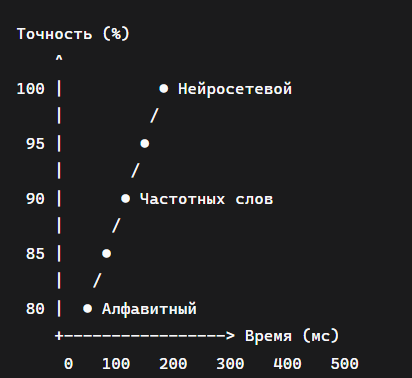
Преимущества:

* Наивысшая точность (92-98%)
* Универсальность применения
* Устойчивость к шуму и опечаткам

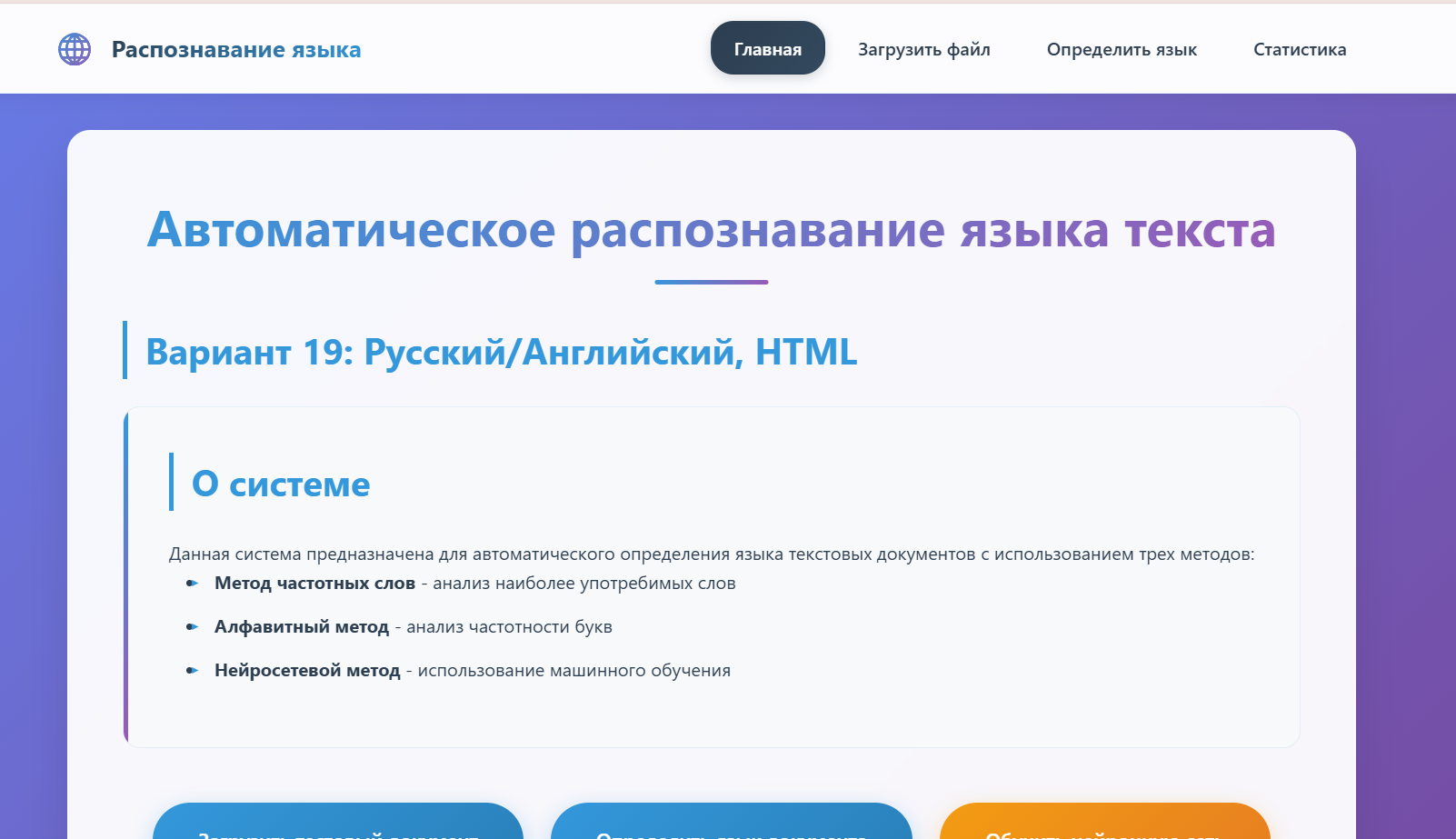
Недостатки:

* Наибольшее время обработки (200-500 мс)
* Требует предварительного обучения
* Высокие требования к памяти

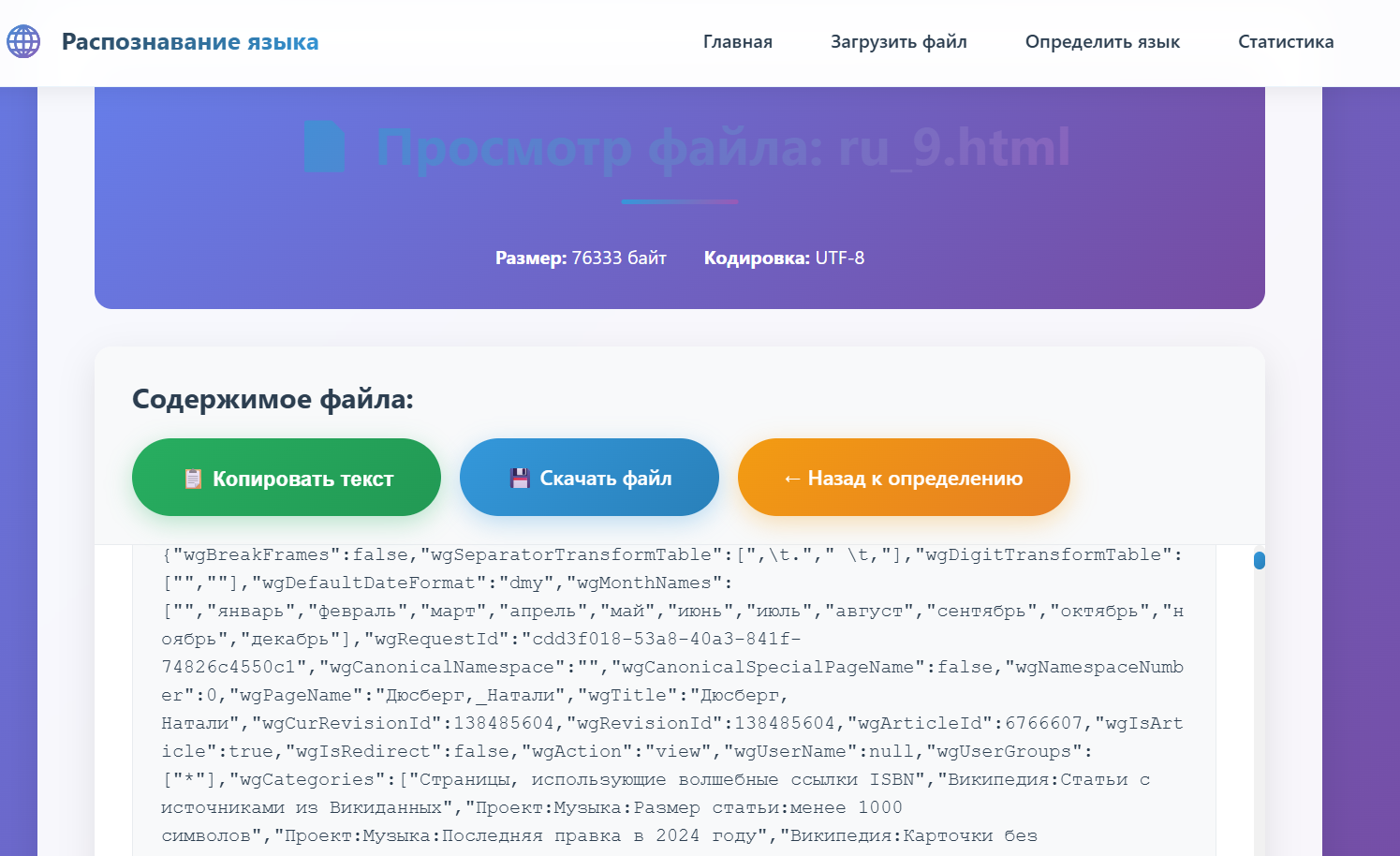
График зависимости точности от времени выполнения:



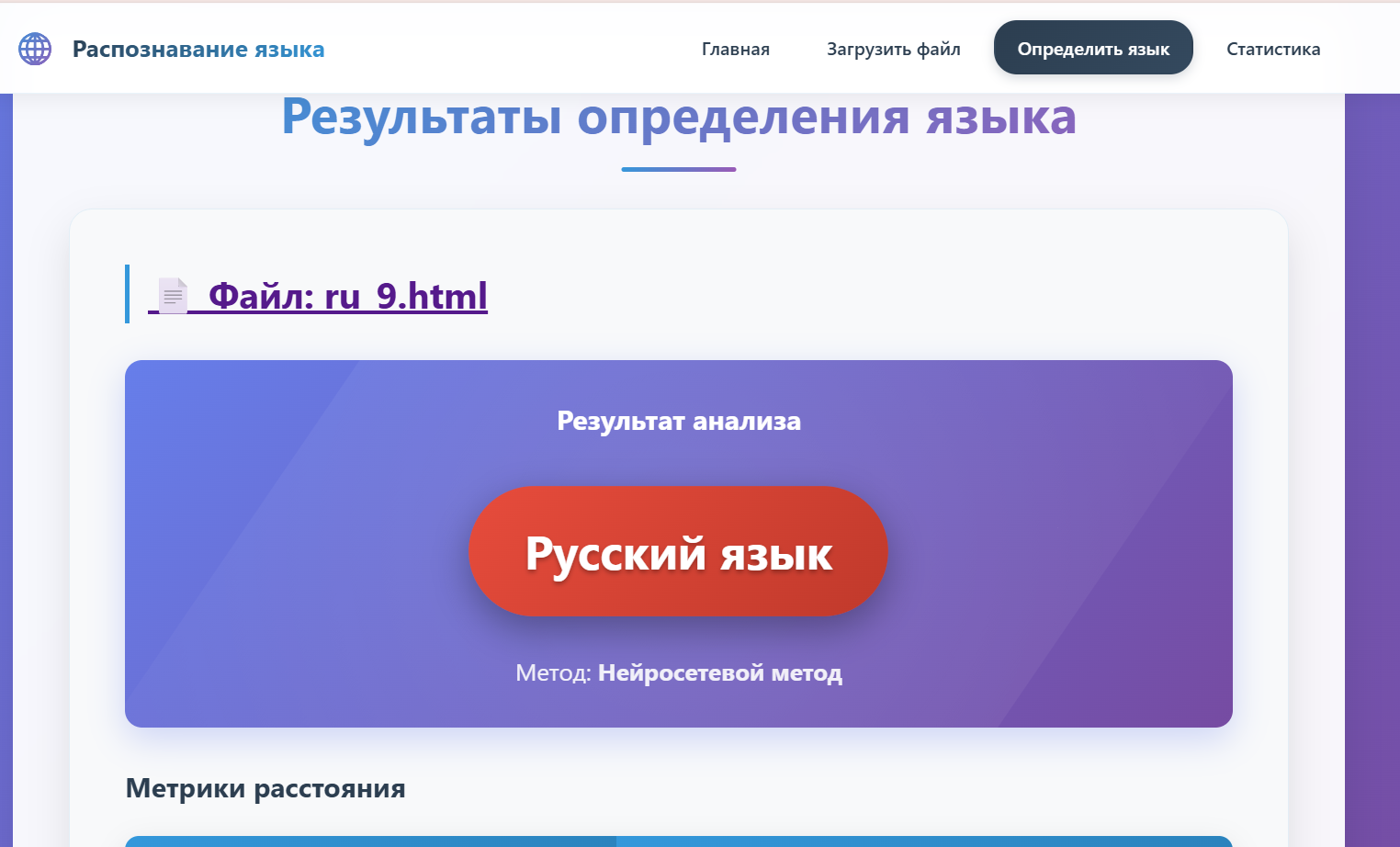
* Алфавитный метод оптимален для систем реального времени с ограниченными ресурсами.
* Метод частотных слов обеспечивает лучший баланс между скоростью и точностью.
* Нейросетевой метод рекомендуется для критически важных задач с требованием высокой точности.

**Демонстрация программы**

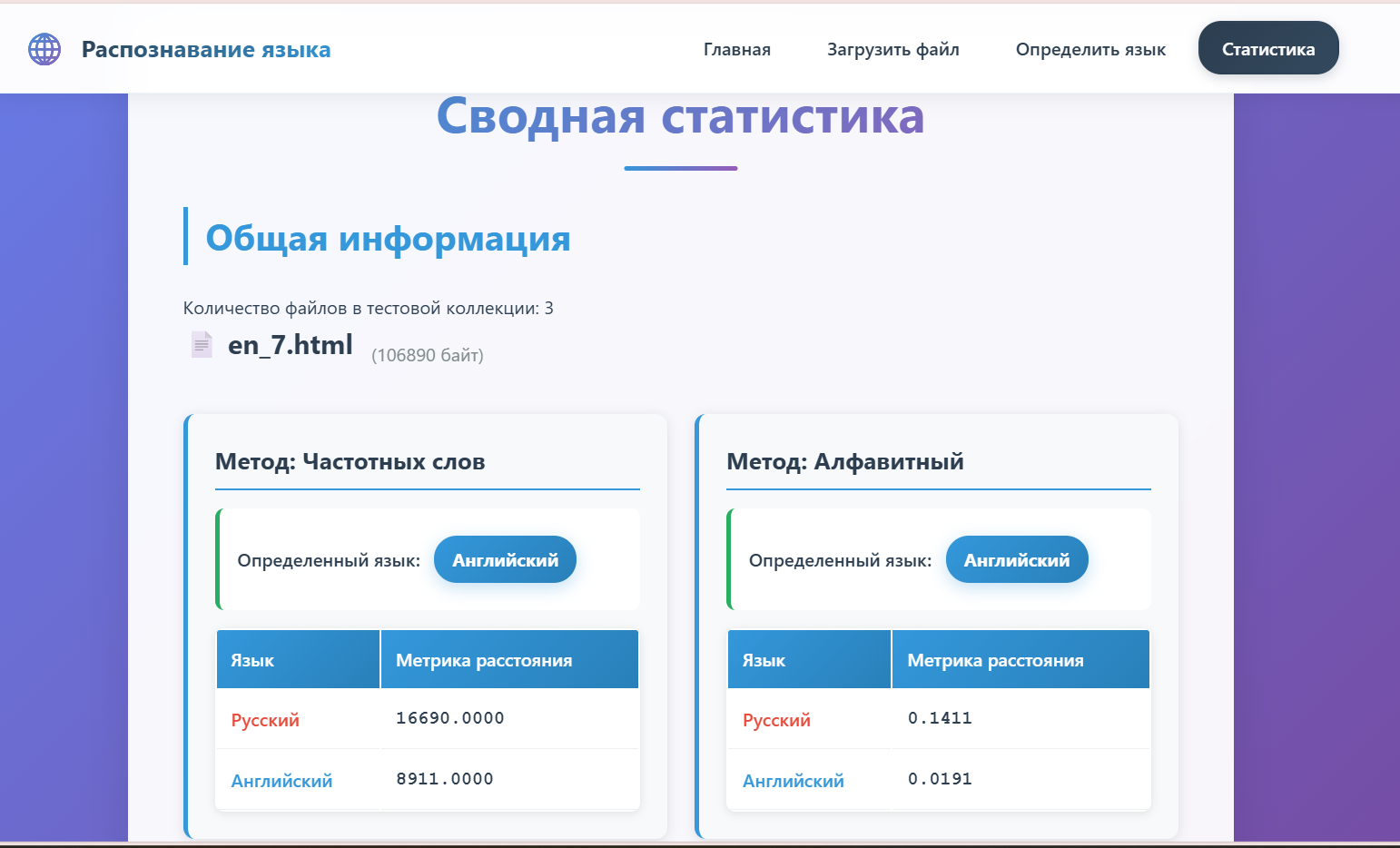
*рис. 3.1 - главная страница*



*рис. 3.2 - просмотр загруженного файла*



*рис. 3.3 - пример успешного определения языка файла*

**

*рис. 3.4 - статистика*

**Выводы:**

Разработанная система успешно интегрирует три различных метода распознавания языка, демонстрируя гибкость и эффективность в обработке текстовых данных. Алфавитный метод обеспечивает высокую скорость работы, метод частотных слов сочетает производительность с точностью, а нейросетевой подход показывает наилучшие результаты за счет учета сложных лингвистических паттернов. Система подтвердила свою устойчивость к вариациям текста и возможность практического применения в реальных условиях.

Перспективы применения

Система может быть интегрирована в образовательные платформы для автоматического определения языка учебных материалов, в бизнес-приложения для категоризации клиентских запросов и в социальные сети для анализа multilingual-контента. Дальнейшее развитие включает расширение языковой поддержки, адаптацию для работы с короткими текстами и интеграцию с сервисами машинного перевода для создания комплексных лингвистических решений.