Учреждение образования
“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”
Кафедра интеллектуальных информационных технологий

Отчёт по практической работе №2
по дисциплине “Естественно-языковой интерфейс интеллектуальных систем”

Выполнил: ст. группы 121701
Пашин Н. А.
Маевский В. Г.

Проверил: Крапивин Ю.Б.

Минск 2024

Цель работы: изучить и отработать практические навыки применения методов автоматического распознавания языка текстовых документов.

Задание: Вариант 4
Язык текста: русский, немецкий.
Формат документа: html.
Реализуемый метод: N-грамм, алфавитный, нейросетевой.

Требования к разрабатываемой системе:

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

Выполнение:

Тестовый набор документов

Тестовый набор документов представляет собой 2 html файла, содержащих полный алфавит 2 языков: русского и немецкого.

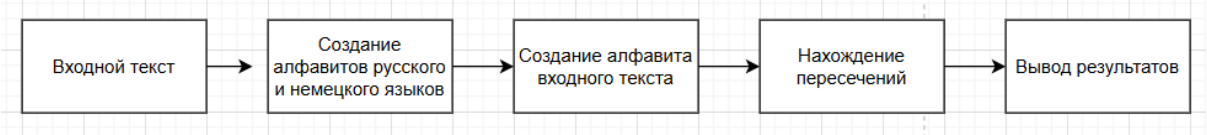
Структура разработанной системы

Система реализована на языке Python версия 3.10 . Графический интерфейс системы создан с помощью библиотеки tkinter. Система может принимать тексты путем ввода абсолютного пути файла. Определять языки текстов система может 3-мя способами: алфавитным, N-граммным и нейросетевым. Все результаты сохраняются в текстовый файл.

Алгоритм определения языка

1. Алфавитный

Алфавитный метод создает 2 алфавита: русский и немецкий. Далее создается алфавит входного файла. Потом находится 2 пересечения: алфавит входного файла с русским алфавитом и с немецким. Сравнив значение, определяется язык текста.



2. N-граммный

В начале алгоритма инициализируются 2 языковых профиля, состоящих из наиболее часто встречающихся 2-грамм для данного языка. Далее исходный текст разбивается на 2-граммы. Этот список анализируется на соответствие с языковыми профилями. В результате данного анализа получаем значения для немецкого и русского языков. По данным значениям определяется язык текста.

3. Нейросетевой

Сначала метод читает текст из указанного файла по пути file_path. Затем создается DataFrame, содержащий этот текст. После этого текст в DataFrame преобразуется в числовую матрицу признаков с использованием уже обученного CountVectorizer(sklearn.feature_extraction.text), который извлекает триграммы из текста.

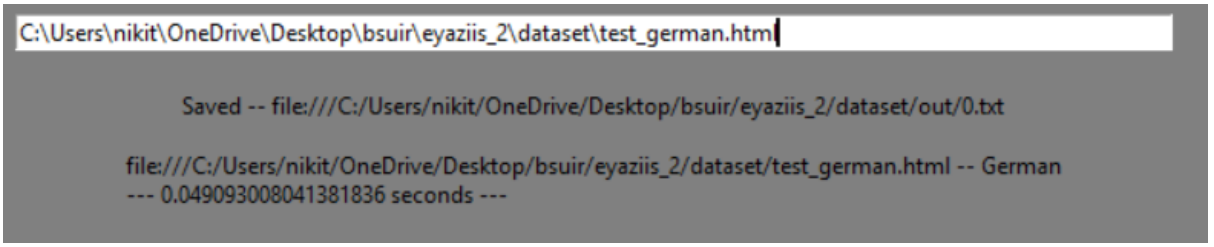
Затем полученная матрица признаков нормализуется, чтобы привести данные к единому диапазону, используя минимальные и максимальные значения, которые были определены во время обучения модели. Далее, обученная нейронная сеть используется для предсказания языка текста. Она возвращает вероятности для каждого языка, и метод извлекает индекс класса с наибольшей вероятностью.

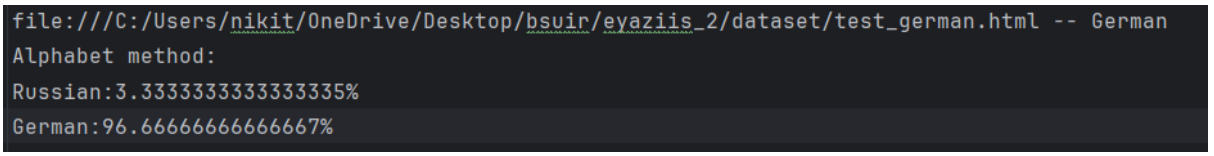
Индекс преобразуется обратно в метку языка с помощью LabelEncoder, и метод возвращает предсказанный язык текста из файла.

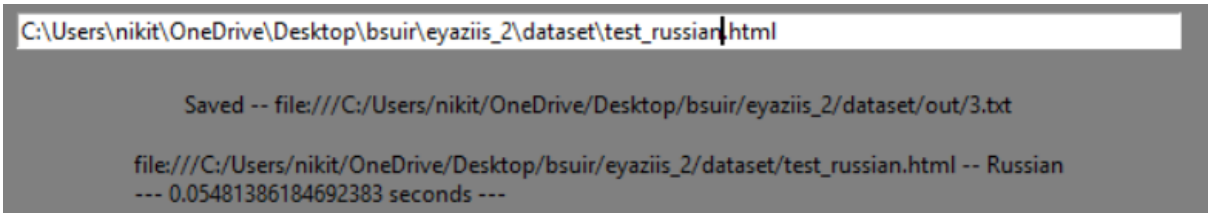
Тестирование

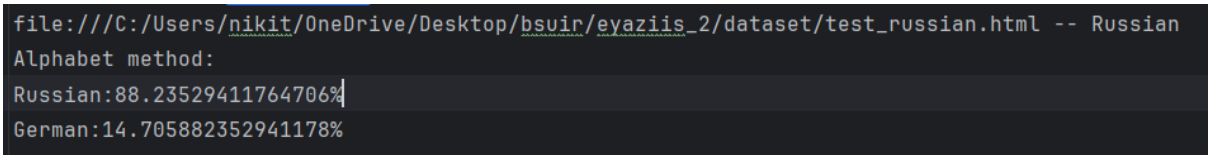
Результаты тестов:

1. Алфавитный

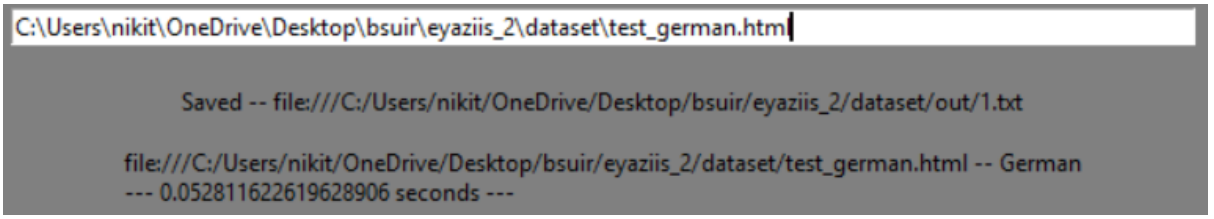


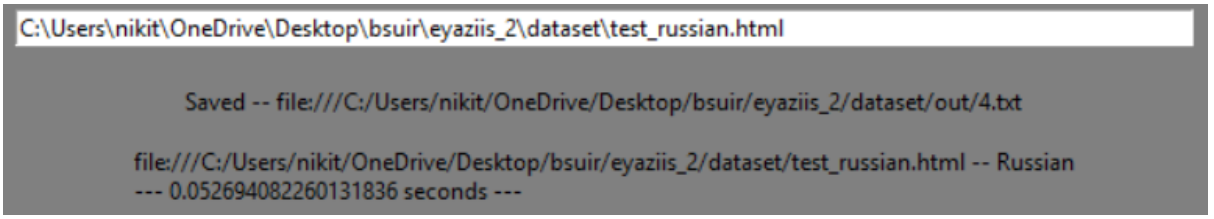


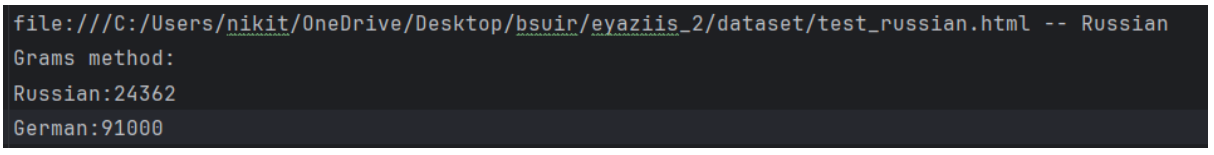




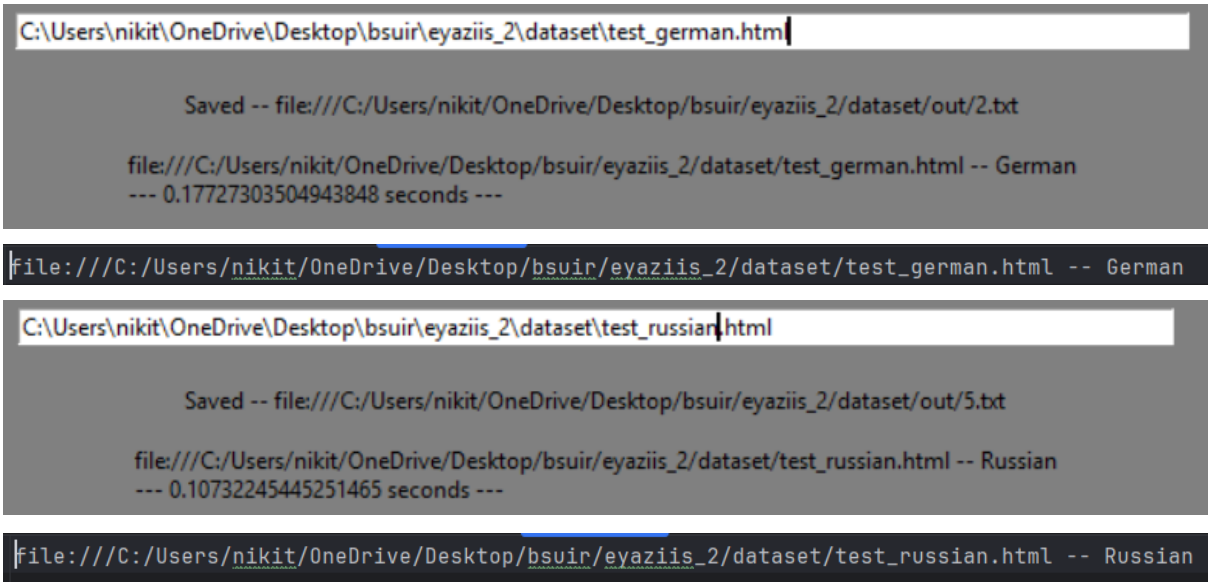
2. N-грамм







3. Нейросетевой



Оценка полученных результатов

Все методы показали 100% эффективность. Однако, N-граммный метод показал наивысшую скорость.

Описание и особенности применения готовых к использованию компонент

Tkinter– это набор библиотек и инструментов для разработки графических пользовательских интерфейсов (GUI) на языке программирования Python.
Scikit – библиотека для Python, предназначенная для выполнения научных и инженерных вычислений. Она строится поверх более базовой библиотеки NumPy и предоставляет множество дополнительных функций и инструментов, которые облегчают выполнение различных вычислительных задач. В данном случае для преобразования текстовых данных в числовые векторы.
Collections – библиотека, которая предоставляет дополнительные структуры данных и инструменты для работы с ними, которые расширяют стандартные возможности встроенных типов данных (списков, кортежей, словарей и множеств).
Keras — это высокоуровневый API для создания и обучения нейронных сетей, который работает поверх таких библиотек, как TensorFlow, Theano и Microsoft Cognitive Toolkit (CNTK).

Вывод

В ходе выполнения лабораторной работы были изучены и применены на практике различные методы определения языка текста. В итоге самым быстрым оказался n-граммный метод.