Лабораторная работа 14

Уровень 1

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

**1)Отличия способов токенизации**

**.split()**: Разделяет текст по пробелам. Простой метод, но не учитывает знаки препинания как отдельные токены и может некорректно обрабатывать сложные случаи (например, дефисы, сокращения).

**Регулярные выражения**: Позволяют гибко определять, что считать словом (например, \b\w+\b извлекает последовательности букв и цифр). Однако требует ручной настройки шаблона, что может быть неудобно для сложных текстов.

**NLTK**: Использует предобученные модели для токенизации, учитывающие особенности языка (например, пунктуацию, сокращения). Обычно даёт более точные результаты, но работает медленнее и требует загрузки ресурсов.(а также вызывает небольшие трудности в использовании)

**2)Отличия в результате**

**len**: Просто считает общее количество токенов в списке. Не предоставляет информацию о частотности отдельных слов.

**Counter**: Создаёт словарь, где ключами являются слова, а значениями — их частота. Общее количество слов совпадает с len, но Counter даёт дополнительную информацию о распределении слов.

**3)Отличия стемминга и лемматизации**

**Стемминг**: Урезает слова до их основы (стебля), игнорируя грамматические формы. Например, "бегущий" и "бег" станут "бег". Может быть менее точным, так как не учитывает контекст и может путать разные слова с одинаковой основой.

**Лемматизация**: Приводит слова к их словарной (нормальной) форме, учитывая морфологию. Например, "бегущий" станет "бежать". Более точна, но требует больше вычислительных ресурсов и морфологического анализатора (например, pymorphy2 для русского языка). (В коде реализовать не получилось из-за большой загрузки)