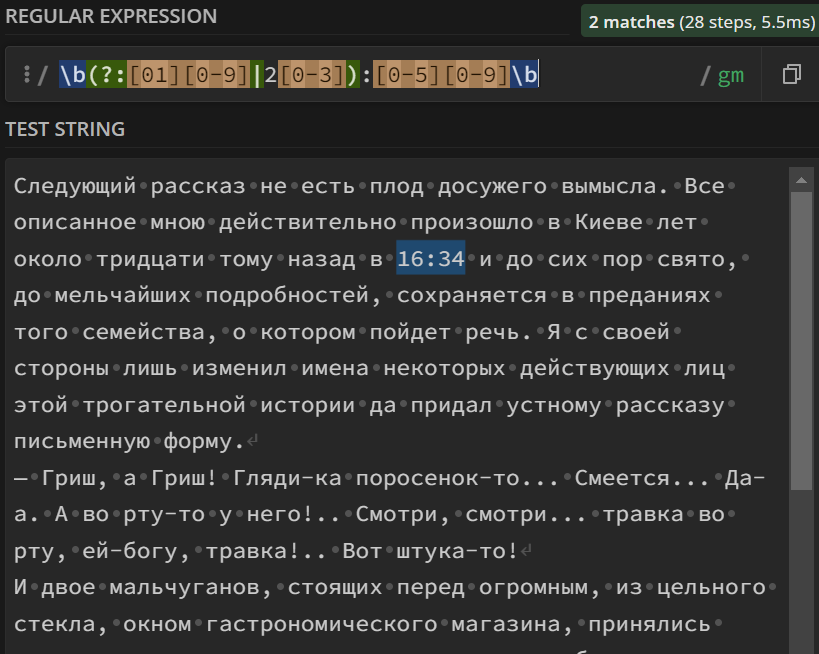
Лабораторная работа 10 (уровень 1)

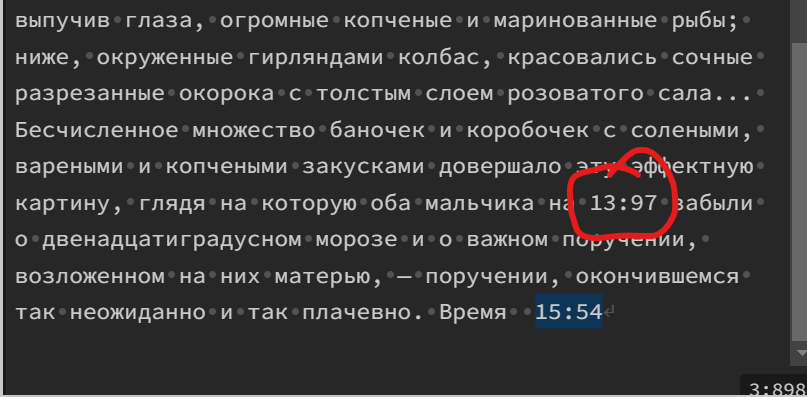
*Лабораторная работа 10.1*

*Задания уровня 1*

1.1

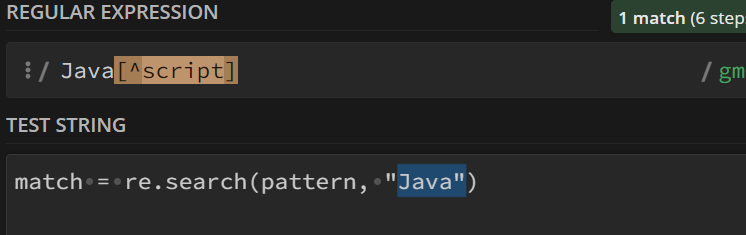
Я возьму любой текс и прост вставлю туда время

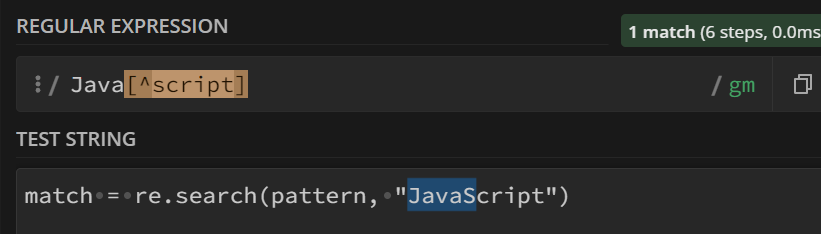




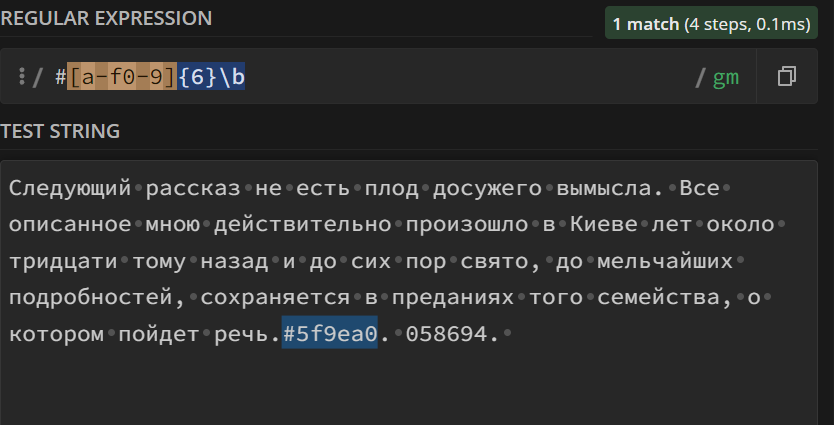
13:97 не учтено

1.2

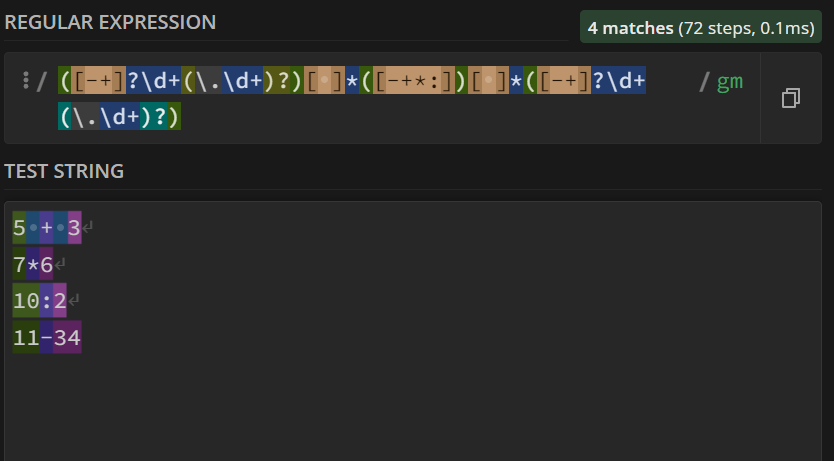




1.3



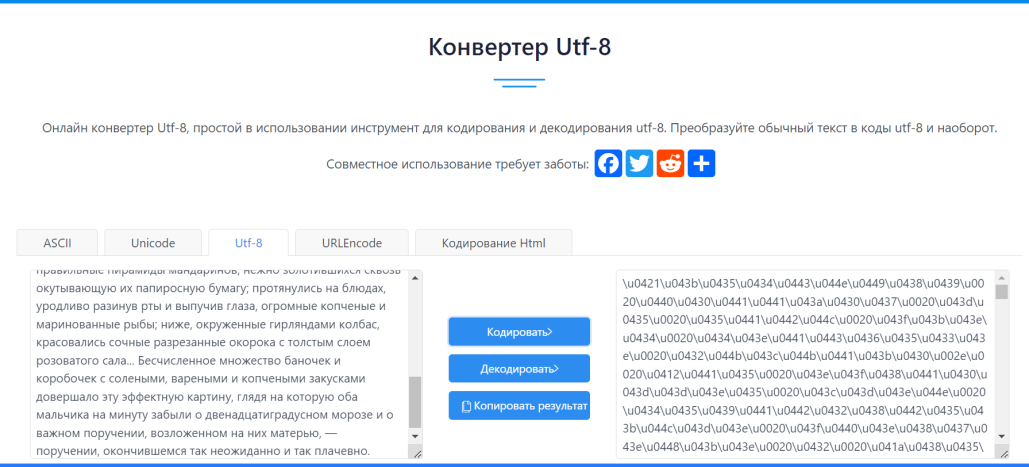
1.4



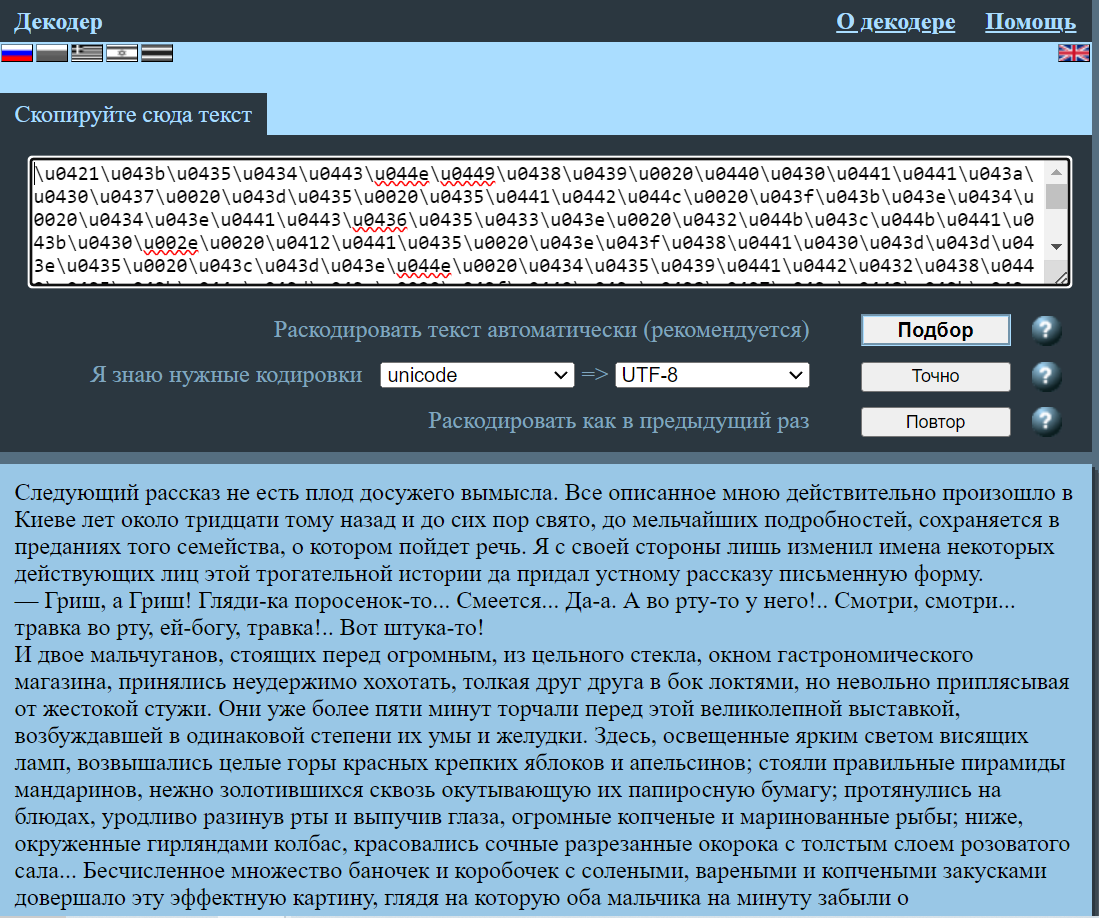
2.1

Берем текст из Куприна «Чудесный доктор».

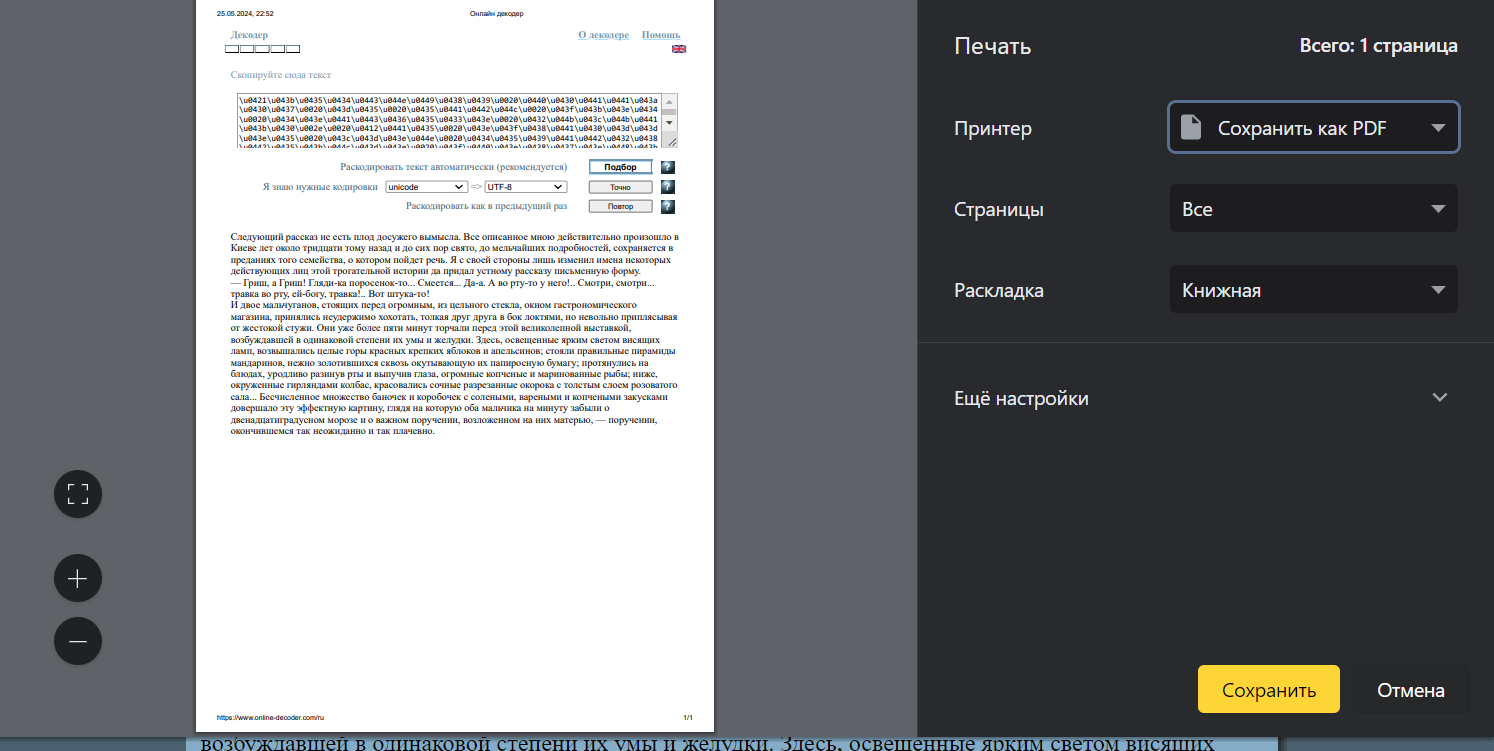
Кодируем его в UTF-8.



Вставляем полученный результат в декодер:



2.2



2.3

Информация на цифровых носителях хранится в закодированном виде и представляет собой последовательность 0 и 1. В основе запоминающего устройства может лежать любой физический эффект, который обеспечивает приведение системы к двум устойчивым состояниям: одно из них будет

обозначать 0, другое - 1 Так, в современных устройствах используются:

•наличие/отсутствие заряда в конденсаторе

•направление намагниченности

•отражение/рассеяние света от поверхности CD, DVDили Blu-ray диска

Минимальная единица хранения информации - 1 бит.

С помощью него можно закодировать два значения 0 и 1.

Упорядоченность информации обеспечивается файловой системой, которая определяет формат содержимого и способ его хранения.

Название FAT расшифровывается как File Allocation Table, или "таблица размещения файлов". Это значит, что "адрес" каждого файла на диске прописан в специальной таблице. Дисковое пространство разбито на кластеры, и каждый файл занимает не менее одного из них.

Обычно большие файлы записываются в кластеры, идущие подряд, но, когда свободного места на диске остается мало, нужного количества свободных ячеек подряд может не быть. Однако если суммарное количество свободных кластеров достаточно для записи файла, он записывается по кусочкам в разные места диска. Это называется фрагментацией. Правда, разбитые на кусочки файлы читаются гораздо медленнее, поэтому существует дефрагментация — перераспределение файлов на диске таким образом, чтобы каждый из них хранился в непрерывной последовательности кластеров.

Практически любая файловая система иерархична: файлы объединяются в каталоги (папки), которые образуют одно или несколько деревьев.

Кроме того, у файла есть расширение - несколько символов после точки, которые обозначают его тип (формат): txt, doc, jpg, mp3, html...

Для перевода последовательности нулей и единиц в текстовые символы используется специальная таблица, которая называется кодировкой; если текст был зашифрован с помощью одной кодировки, а расшифрован с помощью другой, то мы увидим кракозябры.

ASCII (1963 год) — 7-битная кодировка, включающая в себя 128 символов.

ISO/IEC 646 (1972 год) — группы кодировок, основанных на ASCII, где редко используемые символы из ASCII заменялись на необходимые.

KOI8 (1974 год, СССР) — Код Обмена Информацией.

ISO/IEC 8859 (середина 80-х). Ранние кодировки были ограничены 7 битами из-за особенностей некоторых протоколов передачи данных.

Windows-125x (начало 90-х) — группа кодировок, разработанная компанией Microsoft для ОС Windows.

UTF-8 (Unicode Transformation Format, 8-bit — «формат преобразования Юникода, 8-битный», 1993 год) — одна из общепринятых и стандартизированных кодировок текста, которая позволяет хранить символы Юникода, используя переменное количество байт (от 1 до 6).

Portable Document Format (PDF) — межплатформенный формат электронных документов, разработанный фирмой Adobe Systems.

*Лабораторная работа 10.2*

*Задания уровня 1*

1.1

1. spaCy: бесплатная библиотека с открытым исходным кодом для обработки естественного языка на Python.

2. NLTK (Natural Language Toolkit): популярная библиотека для обработки естественного языка на Python.

3. Hugging Face Transformers: коллекция предварительно обученных моделей машинного обучения для обработки естественного языка, включая модели для майнинга текста.

4. Apache OpenNLP: набор инструментов с открытым исходным кодом для обработки естественного языка на Java.

5. TextBlob: простая в использовании библиотека Python для обработки естественного языка и майнинга текста.

6. RapidMiner: платформа для науки о данных и машинного обучения, которая включает в себя функции для майнинга текста.

7. KNIME: еще одна платформа науки о данных и машинного обучения с возможностями для майнинга текста.

8. GATE (General Architecture for Text Engineering): фреймворк для обработки естественного языка с открытым исходным кодом и инструментами для майнинга текста.

9. MALLET (Machine Learning for Language Toolkit): набор инструментов Java для обработки естественного языка, включая моделирование тем и кластеризацию.

10. Voyant Tools: онлайн-инструмент для визуализации и анализа больших текстовых корпусов.

Для сравнения возьмем: spaCy и NLTK

spaCy лучше всего подходит для задач, требующих высокой производительности и точности, таких как распознавание именованных сущностей, синтаксический анализ и классификация текста.

NLTK лучше подходит для задач, требующих большей настройки и гибкости, таких как токенизация, тегирование частей речи и лемматизация.

Преимущества spaCy:

* Высокая производительность: spaCy - очень быстрая библиотека, что делает ее подходящей для обработки больших объемов текста.
* Точность: spaCy также очень точна, что делает ее идеальной для задач, требующих высокой надежности.
* Простота использования: spaCy имеет удобный и интуитивно понятный API, что облегчает его использование даже для начинающих.
* Предварительно обученные модели: spaCy поставляется с набором предварительно обученных моделей для различных языков, что позволяет сразу приступить к работе.
* Поддержка различных языков: spaCy поддерживает более 100 языков, что делает его ценным инструментом для многоязычной обработки текста.

Недостатки spaCy:

* Менее настраиваемый: spaCy менее настраиваемый, чем некоторые другие библиотеки обработки естественного языка, такие как NLTK. Это может быть недостатком для пользователей, которым требуется больший контроль над процессом обработки текста.
* Меньшее сообщество: Сообщество spaCy меньше, чем у более устоявшихся библиотек, таких как NLTK. Это может затруднить поиск ответов на вопросы или помощь в устранении неполадок.
* Меньшее количество ресурсов: Для spaCy доступно меньше ресурсов, таких как учебные пособия и документация, по сравнению с более зрелыми библиотеками.

Преимущества NLTK:

* Широкий спектр функций: NLTK предоставляет широкий спектр функций для обработки естественного языка, включая токенизацию, тегирование частей речи, синтаксический анализ и семантический анализ.
* Настраиваемость: NLTK очень настраиваемый, что позволяет пользователям создавать собственные алгоритмы и модели для конкретных задач.
* Большое сообщество: NLTK имеет большое и активное сообщество, что облегчает поиск ответов на вопросы и помощь в устранении неполадок.
* Огромное количество ресурсов: для NLTK доступно множество ресурсов, таких как учебные пособия, документация и примеры кода.

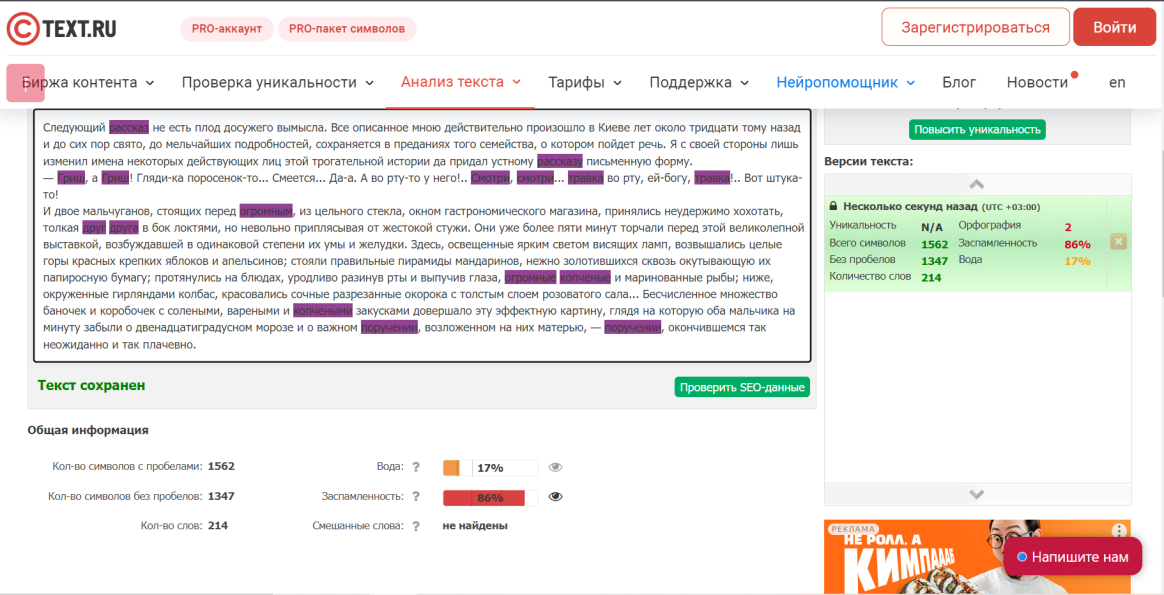
Поддержка многих языков: NLTK поддерживает широкий спектр языков, что делает его ценным инструментом для многоязычной обработки текста.

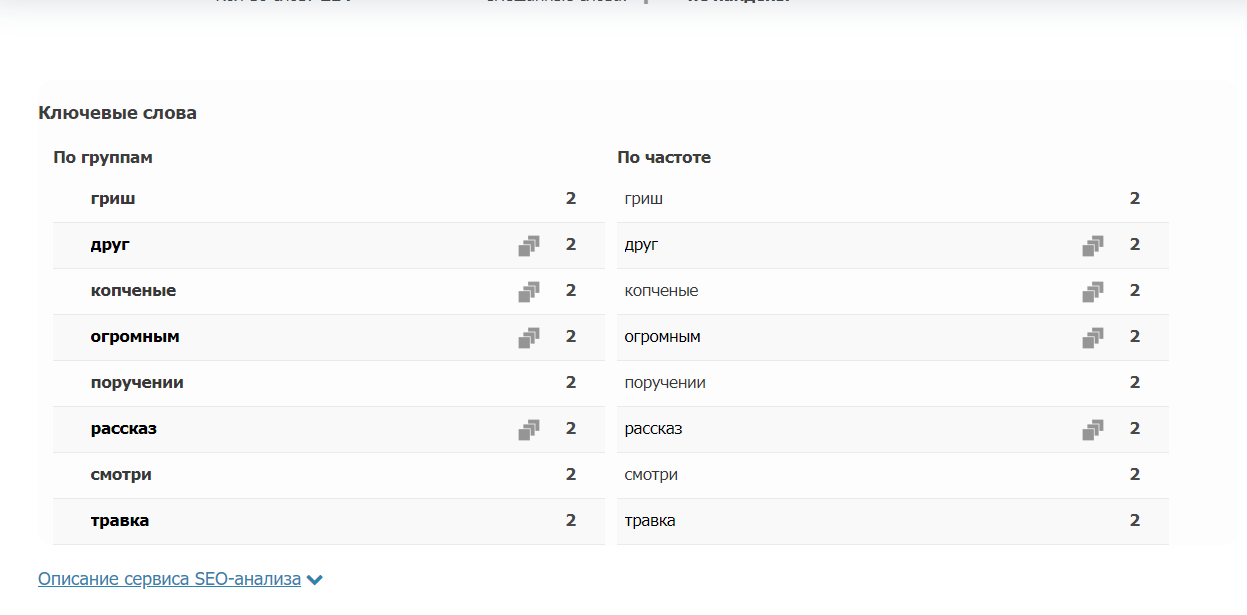
Недостатки NLTK:

* Более медленная: NLTK может быть медленнее, чем некоторые другие библиотеки обработки естественного языка, такие как spaCy.
* Менее удобный: NLTK может быть менее удобным в использовании, чем некоторые другие библиотеки, особенно для начинающих.
* Меньшая точность: NLTK может быть менее точным, чем некоторые другие библиотеки, особенно для задач, требующих высокой надежности.

1.2

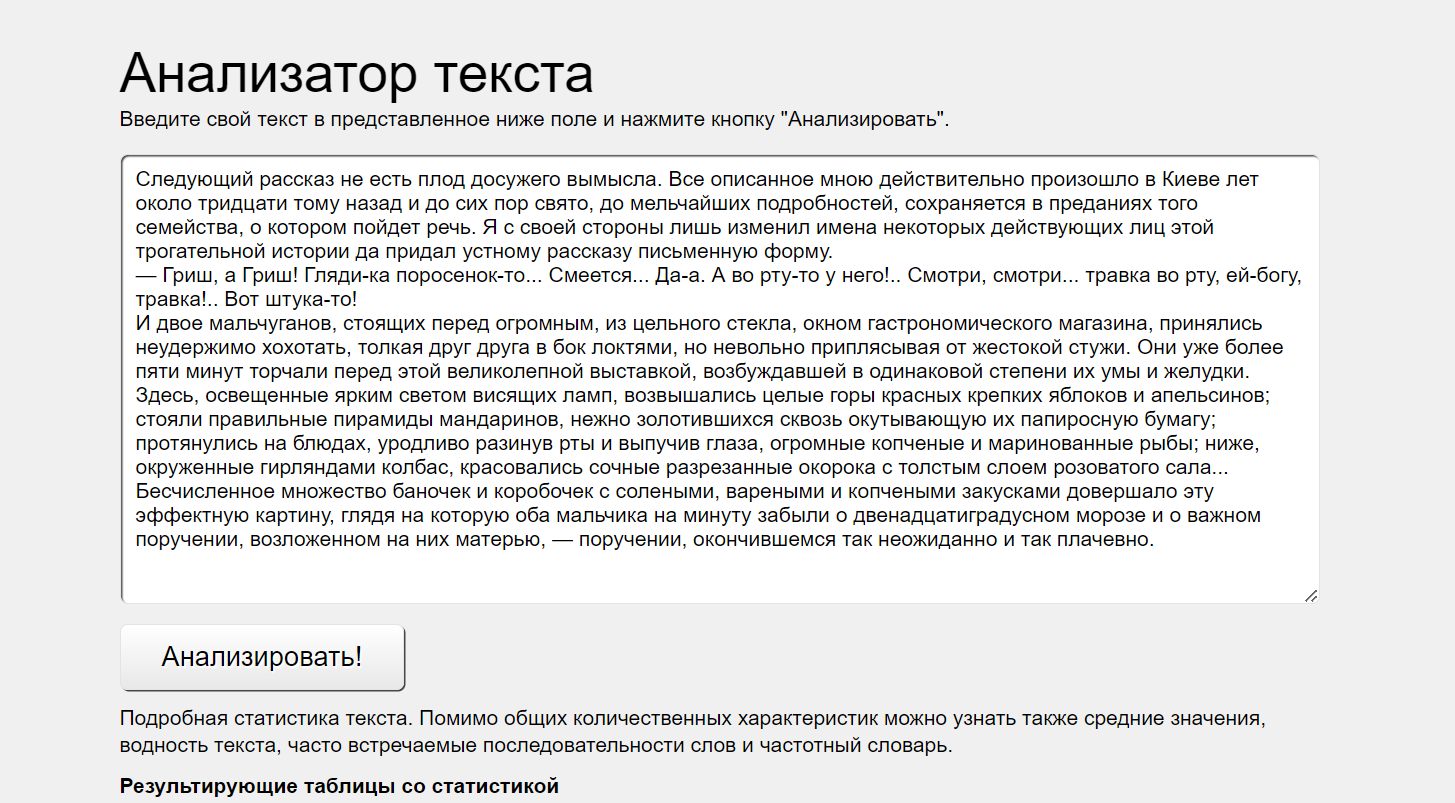
TEXT.RU - это онлайн-сервис для проверки уникальности и качества текстов на русском языке. Он также предоставляет различные инструменты для анализа и оптимизации текстов под поисковые системы.

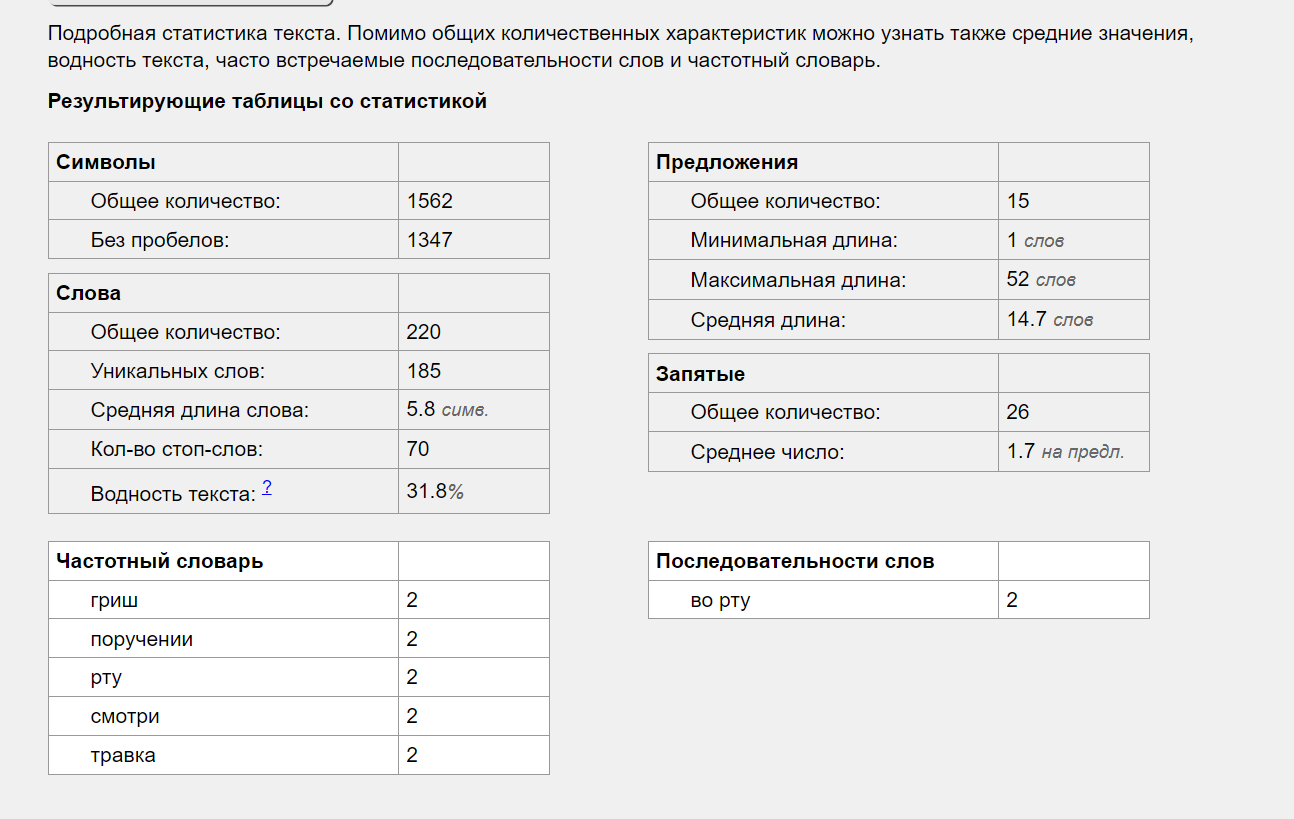




TextAnalyzer - онлайн-инструмент для анализа текста на русском языке. Он предлагает различные функции, в том числе:

* Проверка орфографии и грамматики
* Подсчет количества символов, слов, предложений и абзацев
* Определение водности текста
* Выделение ключевых слов и фраз
* Анализ тональности текста
* Определение читабельности текста





Адвего Лингвист — это профессиональный инструмент для проверки содержания текста, его стилистики, читабельности и осмысленности

