**Лабораторная работа 10**

**Уровень 1**

Перечислите 10 ресурсов для TextMining. Сравните 2 (1 уровень) / 3 (2 уровень) / 4 (3 уровень)  из них, проанализировав в них один и тот же текст. Обобщите выводы, указав возможности, преимущества, недостатки ресурсов. Составьте сопоставительную таблицу

**10** ресурсов для TextMining:

**1. Python Natural Language Toolkit (NLTK)**

**2. R programming language**

**3. TextBlob**

**4. Stanford NLP toolkit**

**5. Gensim**

**6. Apache OpenNLP**

**7. RapidMiner**

**8. IBM Watson NLP**

**9. Aylien Text Analysis API**

**10. Orange Data Mining software**

Сравню 2 из них:

1.**NLTK**

**2.StanfordNLP**

1. **NLTK (NaturalLanguageToolkit)** - библиотека для обработки естественного языка на языке программирования Python.

2. **Stanford NLP** - библиотека с открытым исходным кодом для анализа текста на английском языке.

Для анализа мы возьмем текст: "Технологии обработки естественного языка становятся все более популярными в последнее время."

**1. NLTK:**

- **Токенизация**: ["Технологии", "обработки", "естественного", "языка", "становятся", "все", "более", "популярными", "в", "последнее", "время"]

- **Лемматизация**: ["технология", "обработка", "естественный", "язык", "становиться", "весь", "более", "популярный", "в", "последний", "время"]

2. **StanfordNLP**:

- **Токенизация**: ["Технологии", "обработки", "естественного", "языка", "становятся", "все", "более", "популярными", "в", "последнее", "время"]

- **Лемматизация**: ["Технологии", "обработка", "естественный", "язык", "становиться", "все", "большой", "популярный", "в", "последний", "время"]

**Обобщение:**

- NLTK предоставляет **богатый функционал** для обработки текста, включая токенизацию и лемматизацию.

- Stanford NLP также предоставляет мощные инструменты для анализа текста, но **может иметь некоторые отличия** в результате обработки.

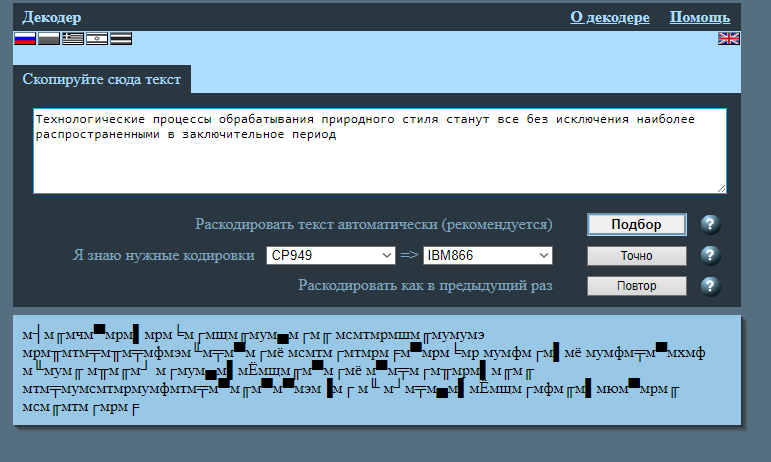
- Оба ресурса имеют свои преимущества и недостатки, и выбор зависит от конкретных потребностей и задач

|  |  |  |
| --- | --- | --- |
| Ресурс | **NLTK** | **StanfordNLP** |
| Токенизация | + | + |
| Лемматизация | + | + |
| Скорость | Средне | Выше |
| Простота | Выше | Средне |

Форматы данных и кодировки

<http://www.online-decoder.com/ru>

Использую текст: Технологические процессы обрабатывания природного стиля станут все без исключения наиболее распространенными в заключительное период



**законспектируйте приведённую выше информацию**

**Конспект**

Информация на цифровых носителях хранится в закодированном виде и представляет собой последовательность **0 и 1**. Как это устроено физически? 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, DVD или 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. Иными словами, файловая система - это раздел жесткого диска, выделенный для хранения файлов на компьютере и другом электронном оборудовании: мобильных телефонах, фотоаппаратах и т.д. Рассмотрим ее устройство на примере FAT - файловой системы, впервые созданной в 1976-1977 годах Биллом Гейтсом и Марком МакДональдом.

Название FAT расшифровывается как FileAllocationTable, 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. Файловая система FAT состоит из следующих частей:

1)          Загрузочная запись

а) BPB (BIOS parameterblock) – содержит указание на тип файловой системы и физические характеристики носителя

б) FSINFO – сообщает о том, сколько кластеров свободно

2)          FAT – та самая таблица соответствия файлов и кластеров, в которых они расположены

3)          Собственно файлы

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. Расположение файла в такой структуре описывается с помощью пути.

Кроме того, у файла есть расширение - несколько символов после точки, которые обозначают его тип (формат): txt, doc, jpg, mp3, html... Проще говоря, расширение указывает на то, какая информация (текстовая, графическая и т.д.) находится в файле и как правильно ее раскодировать.

Например, если мы попробуем вручную поменять расширение картинки с jpg на txt и затем открыть ее в текстовом редакторе, мы увидим "кракозябры", потому что байтовая последовательность, которая обозначает цвета пикселей, будет расшифрована неправильно – в данном случае как текстовые символы.

Дело в том, что 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.

Кодировки:

ASCII (1963 год) — 7-битная кодировка, включающая в себя 128 символов: 33 непечатных управляющих символа (влияющих на обработку текста и пробелов) и 95 печатных символов, включая цифры, буквы латинского алфавита в строчном и прописном вариантах и ряд пунктуационных символов.

ISO/IEC 646 (1972 год) — группы кодировок, основанных на ASCII, где редко используемые символы из ASCII заменялись на необходимые. Группа включала в себя варианты кодировки для Канады, Китая, Кубы, Германии, Дании, Финляндии, Франции, Великобритании, Греции, Венгрии, Ирландии, Японии, Южной Кореи, Мальты, Норвегии, Швеции и Югославии.

KOI8 (1974 год, СССР) — Код Обмена Информацией. Как следует из названия, это была 8-битная кодировка, что позволяло включить в нее в два раза больше символов. KOI8 включала в себя цифры, буквы латинского и русского алфавита, а также знаки пунктуации, спецсимволы и псевдографику.

ISO/IEC 8859 (середина 80-х). Ранние кодировки были ограничены 7 битами из-за особенностей некоторых протоколов передачи данных. Однако со временем эти ограничения свою актуальность потеряли, в то время как необходимость в дополнительных символах для языков, использующих латинский алфавит, только росла.

Windows-125x (начало 90-х) — группа кодировок, разработанная компанией Microsoft для ОС Windows. В Windows-1251 вошли все символы русского и близких к нему языков: украинского, белорусского, болгарского, сербского и македонского.

UTF-8 (UnicodeTransformationFormat, 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)

Unicode – это стандарт кодирования символов, включающий в себя знаки почти всех письменных языков мира. В настоящее время стандарт является доминирующим в Интернете

Если кодировка сбита, то по характеру кракозябр можно понять, в какой кодировке был текст и в какой кодировке он отображается.