3. Практическое занятие: Токенизация и нормализация текста

Цель занятия

Научиться преобразовывать неструктурированный текст в набор подготовленных токенов, пригодных для анализа, используя методы очистки, токенизации и приведения слов к начальной форме.

Задание 1. Очистка и предварительная обработка

Текст из интернета часто содержит «шум»: HTML-теги, ссылки и лишнюю пунктуацию. Напишите функцию, которая принимает строку и возвращает очищенный текст в нижнем регистре без лишних символов.

Python
```
import re

raw_text = "NLP — это интересно! Посети сайт https://example.com <br> для 100% погружения."

def preprocess_text(text):
    text = text.lower()
    # Удаление ссылок
    text = re.sub(r'https?://\S+|www\.\S+', '', text)
    # Удаление HTML-тегов
    text = re.sub(r'<.*?>', '', text)
    # Удаление спецсимволов и цифр (оставляем только буквы и пробелы)
    text = re.sub(r'[^a-za-яё\s]', '', text)
    # Удаление лишних пробелов
    text = re.sub(r'\s+', ' ', text).strip()
    return text

clean_text = preprocess_text(raw_text)
print("Результат очистки:", clean_text)
```

Задание 2. Токенизация на разных уровнях

Токенизация — это процесс разбиения текста на единицы. Разделите текст на слова (word tokenization) и на N-граммы (сочетания слов).

Python
```
# Токенизация по словам (простой метод)
tokens = clean_text.split()
print("Токены слов:", tokens)

# Создание биграмм (сочетаний по два слова)
def get_bigrams(words):
    return [f"{words[i]} {words[i+1]}" for i in range(len(words)-1)]
```

```
bigrams = get_bigrams(tokens)
print("Биграммы:", bigrams)
```

Задание 3. Нормализация: Стемминг и Лемматизация

Чтобы компьютер понимал, что «бежать», «бегу» и «бежал» — это одно действие, используется нормализация. Сравните два подхода: стемминг (грубое отсечение окончаний) и лемматизацию (приведение к словарной форме).

Примечание: для качественной лемматизации русского языка рекомендуется библиотека PyMorphy2 или Mystem.

Python
```
# Пример концепции на словах: "кошками", "бежала", "интересно"

# Стемминг (пример логики)
# "кошками" -> "кошк"
# "бежала" -> "беж"

# Лемматизация (словарная форма)
# "кошками" -> "кошка"
# "бежала" -> "бежать"

# Задание: объясните, в каком случае стемминг может испортить смысл слова
(например, "организация" и "орган").
```

Задание 4. Удаление стоп-слов

Стоп-слова (предлоги, союзы, частицы) часто не несут смысловой нагрузки для классификации текстов. Отфильтруйте список токенов, удалив из него часто встречающиеся союзы.

Python
```
stop_words = ["и", "это", "для", "на", "в", "с"]
filtered_tokens = [word for word in tokens if word not in stop_words]

print("Токены после удаления стоп-слов:", filtered_tokens)
```

Итоговое задание для студента

Возьмите произвольный абзац текста (например, из новостной статьи) и примените к нему полный цикл обработки:

1. Очистка от символов и приведение к нижнему регистру.
2. Токенизация по словам.
3. Удаление стоп-слов.
4. Вывод списка уникальных нормализованных токенов (лемм).

Контрольные вопросы

1. Чем токенизация по словам отличается от токенизации по предложениям?
2. Почему лемматизация считается более сложным процессом, чем стемминг?
3. В каких задачах NLP (например, поиск по ключевым словам или анализ тональности) нельзя удалять стоп-слово «не»?