Московский государственный технический университет им. Н.Э. Баумана
Факультет «Головной учебно-исследовательский и методический центр
профессиональной реабилитации лиц с ограниченными возможностями здоровья (инвалидов)»
Кафедра «Системы обработки информации и управления»

Лабораторная работа 8

по дисциплине «Методы машинного обучения в АСОИУ»

" Предобработка текстов "

СТУДЕНТ:
студент группы ИУ5Ц-21М
Москалик А.А.

ПРЕПОДАВАТЕЛЬ:
Гапанюк Ю.Е.

Москва, 2024

Цель лабораторной работы: изучение методов предобработки текстов.

Задание:

Для произвольного предложения или текста решите следующие задачи:

- Токенизация.
- Частеречная разметка.
- Лемматизация.
- Выделение (распознавание) именованных сущностей.
- Разбор предложения.

Ход работы

1. Импорт библиотек

Этот код выполняет подготовку к анализу текста, используя две популярные библиотеки для обработки естественного языка: nltk и spaCy.

- nltk.download('punkt'): Загружает ресурсы для токенизатора punkt. Этот токенизатор предварительно обучен работать с многими европейскими языками и используется для разбиения текста на предложения и слова.

- nltk.download('averaged_perceptron_tagger'): Загружает ресурсы для частеречного теггера, который использует алгоритм перцептрона. Это обеспечивает функциональность маркировки частей речи (POS tagging), позволяя определить, является ли слово существительным, глаголом и т. д.

```
import nltk
import spacy
from nltk.tokenize import word_tokenize
from spacy import displacy

nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
```

2. Текст для анализа

Текст содержит информацию о компании Apple Inc.:

```
text = "Apple Inc. is an American multinational technology company headquartered in Cupertino, California."
```

– "Apple Inc." указывает на название компании.

– "is an American multinational technology company" описывает компанию как американскую мультинациональную технологическую компанию.
– "headquartered in Cupertino, California" указывает на местоположение главного офиса компании в городе Купертино, штат Калифорния.

```
# Инициализация среды Gym
env = gym.make('CartPole-v1')

# Получение размеров состояния и действий
state_size = env.observation_space.shape[0]
action_size = env.action_space.n

# Инициализация агентов и оптимизаторов
actor = Actor(state_size, action_size)
critic = Critic(state_size)

actor_optimizer = optim.Adam(actor.parameters(), lr=0.01)
critic_optimizer = optim.Adam(critic.parameters(), lr=0.01)
```

3. Токенизация

Токенизация — это процесс разделения текста на более мелкие части, называемые токенами. Токены обычно представляют собой слова, числа или знаки пунктуации. Токенизация является одним из первых шагов в обработке и анализе текста, поскольку она помогает подготовить данные для дальнейшей обработки,такой как частеречная разметка, лемматизация и синтаксический разбор.

```
tokens = word_tokenize(text)
print("Tokens:", tokens)
```

- word_tokenize(text) — функция из библиотеки NLTK, используемая для токенизации строки text. Она разбивает текст на слова, учитывая знаки пунктуации как отдельные токены.

В переменную tokens сохраняется список токенов, полученных в результате токенизации текста.

- print("Tokens:", tokens) — выводит на экран список токенов. В данном случае вывод будет следующим:

```
Tokens: ['Apple', 'Inc.', 'is', 'an', 'American', 'multinational', 'technology', 'company', 'headquartered', 'in', 'Cupertino', ',', 'California', '.']
```

4. Частеречная разметка

Частеречная разметка (POS tagging, part-of-speech tagging) — это процесс присваивания каждому слову в тексте соответствующей части речи, такой как существительное, глагол, прилагательное и так далее. Этот процесс является важным шагом в анализе текста, поскольку понимание роли каждого слова в предложении помогает в дальнейшем синтаксическом и семантическом анализе.

```
pos_tags = nltk.pos_tag(tokens)
print("POS Tags:", pos_tags)
```

- nltk.pos_tag(tokens) — функция, которая принимает список токенов и возвращает список кортежей, где каждый кортеж содержит токен и соответствующий ему тег части речи. Для определения частей речи используется предварительно обученная модель.

В переменной pos_tags сохраняется список кортежей с токенами и их частеречными тегами.

- print("POS Tags:", pos_tags) — выводит на экран список частеречных тегов. В данном случае вывод будет следующим:

```
POS Tags: [('Apple', 'NNP'), ('Inc.', 'NNP'), ('is', 'VBZ'), ('an', 'DT'), ('American', 'JJ'), ('multina
tional', 'NN'), ('technology', 'NN'), ('company', 'NN'), ('headquartered', 'VBD'), ('in', 'IN'), ('Cuper
tino', 'NNP'), (',', ','), ('California', 'NNP'), ('.', '.')]
```

- NNP обозначает собственное имя (proper noun).
- VBZ обозначает глагол в настоящем времени, 3-е лицо единственного числа.
- DT обозначает артикль или указательное местоимение.
- JJ обозначает прилагательное.
- NN обозначает общее существительное (common noun) в единственном числе.
- VBD обозначает глагол в прошедшем времени.
- IN обозначает предлог или подчинительный союз.

5. Лемматизация

Лемматизация — это процесс приведения слова к его базовой форме, или лемме, который обычно включает сведение различных форм слова (например, множественное число, разные времена глаголов) к единой, канонической форме. Этот процесс помогает уменьшить морфологическую сложность текста и улучшить

производительность многих задач обработки естественного языка, таких как семантический анализ, машинный перевод и поиск информации.

```
nlp = spacy.load("en_core_web_sm")
doc = nlp(text)
lemmas = [(token.text, token.lemma_) for token in doc]
print("Lemmas:", lemmas)
```

```
Lemmas: [('Apple', 'Apple'), ('Inc.', 'Inc.'), ('is', 'be'), ('an', 'an'), ('American', 'american'), ('m
ultinational', 'multinational'), ('technology', 'technology'), ('company', 'company'), ('headquartered',
'headquarter'), ('in', 'in'), ('Cupertino', 'Cupertino'), (',', ','), ('California', 'California'),
('.', '.')]
```

6. Распознавание именованных сущностей

Распознавание именованных сущностей (Named Entity Recognition, NER) — это процесс идентификации и классификации ключевых информационных элементов в тексте в категории, такие как имена людей, организаций, местоположений, дат, продуктов и других специфических групп. NER помогает в структурировании неструктурированного текста и является важной частью многих приложений обработки естественного языка, таких как извлечение информации, автоматическое резюмирование, поиск по именам и машинный перевод.

```
entities = [(entity.text, entity.label_) for entity in doc.ents]
print("Named Entities:", entities)
```

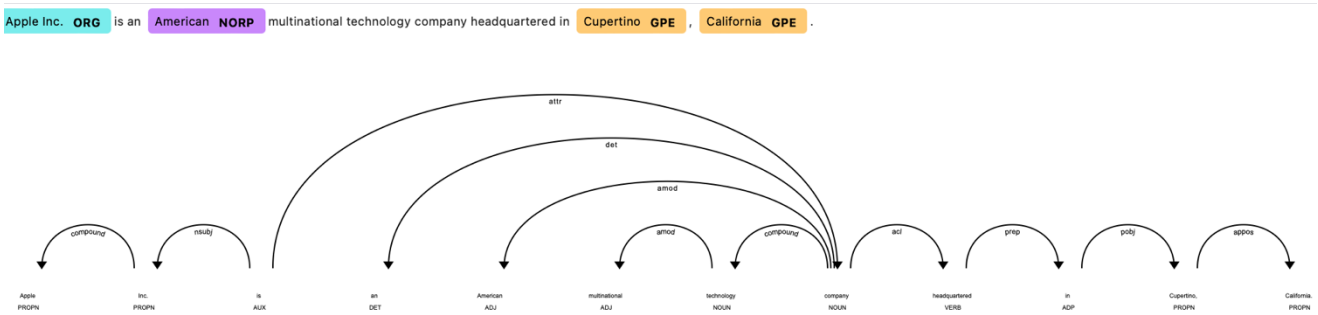
```
Named Entities: [('Apple Inc.', 'ORG'), ('American', 'NORP'), ('Cupertino', 'GPE'), ('California', 'GP
E')]
```

- 'Apple Inc.' классифицирована как 'ORG' (организация).
- 'American' классифицировано как 'NORP' (национальная или религиозная группа).
- 'Cupertino' и 'California' классифицированы как 'GPE' (геополитическая единица, такая как страна, город, штат).

7. Визуализация анализа

Визуализация анализа текста с использованием инструментов библиотеки spaCy, таких как displacy, позволяет наглядно представить результаты обработки естественного языка.

```
displacy.render(doc, style='ent')  # Для именованных сущностей
displacy.render(doc, style='dep')  # Для разбора предложения
```



Вывод:

На изображении показаны:

1. Распознавание именованных сущностей (NER):
   - Apple Inc. помечена как организация (`ORG`).
   - American помечено как национальная принадлежность (`NORP`).
   - Cupertino, California помечено как геополитическая единица (`GPE`).
2. Разбор предложения (Dependency Parsing):
   - Структура зависимостей между словами в предложении визуализирована с помощью стрелок, указывающих отношения между словами (например, `nsubj`, `attr`, `amod`).

Этот вывод демонстрирует, как spaCy анализирует текст, выделяя ключевые структурные элементы и семантические свойства. Такой визуальный анализ очень полезен для понимания взаимосвязей в тексте и для проверки правильности обработки текста алгоритмами.