

Тест начат	Пятница, 8 декабря 2023, 11:52
Состояние	Завершены
Завершен	Пятница, 8 декабря 2023, 12:58
Прошло времени	1 ч. 6 мин.
Оценка	Еще не оценено

Вопрос **1**

Верно

Баллов: 1,00 из 1,00

Расчет косинуса угла между словами

Напишите программу, которая рассчитывает косинус угла между словами cat и cow, используя библиотеку spacy и модель en_core_web_sm для векторизации слов.

Напишите код программы в самостоятельно созданном Python-ноутбуке. Вы можете использовать следующий код, чтобы начать писать программу:

```
import spacy
import numpy as np

nlp = spacy.load('en_core_web_sm')

word_1 = 'cat'
word_2 = 'cow'
```

Результат вычислений округлите до 3-х знаков после запятой и введите в поле ввода ответа в L

Дополнительные материалы

- 1. [SpaCy — NL Pub](#)
- 2. [Можно всё: решение NLP задач при помощи spacy / Хабр](#)

Ответ:

0.775

✓

Спасибо! Уверены, что вы хорошо постарались и поработали!

Предлагаем свериться с возможным вариантом решения.

```
import spacy
import numpy as np

nlp = spacy.load('en_core_web_sm')

word_1 = 'cat'
word_2 = 'cow'

token_1 = nlp(word_1)
token_2 = nlp(word_2)

def cosine(v1, v2):
    if np.linalg.norm(v1)*np.linalg.norm(v2) > 0:
        return np.dot(v1, v2) / (np. linalg.norm(v1)*np.linalg.norm(v2))
    else:
        return 0

cosine_ = cosine(token_1.vector, token_2.vector)

cosine_
```

Вопрос **2**
Выполнен
Балл: 1,00

Расчет косинусной меры между предложениями

Напишите программу, которая будет сравнивать два предложения и выводить значение косинус угла между их векторными представлениями.

Предложения могут быть любыми, например:

- «Сегодня очень холодно на улице» и «На улице сегодня очень холодно»;
- «Обезьяны любят бананы» и «Собаки не любят молоко».

Программа должна выводить значение косинусной меры угла между векторами, которые соотве каждому из предложений.

Для решения данной задачи вам понадобится библиотека spacy и модель для работы с русским. Используйте следующий код для их установки:

```
!pip install spacy
!python -m spacy download ru_core_news_lg
```

Далее выполните импорт библиотеки spacy и русскоязычной модели ru_core_news_lg:

```
import spacy
nlp = spacy.load("ru_core_news_lg")
```

Сохраните код вашей программы в отдельный файл и загрузите решение в LMS.

```
import spacy
import numpy as np
# Загрузка русскоязычной модели
nlp = spacy.load("ru_core_news_lg")

# Функция для вычисления косинусной меры угла между двумя предложениями
def calculate_cosine_similarity(sentence1, sentence2):
    vector1 = nlp(sentence1).vector
    vector2 = nlp(sentence2).vector


    # Вычисление косинуса угла между векторами
    cosine_similarity = vector1.dot(vector2) / (np.linalg.norm(vector1) * np.linalg.norm(vector2))

    return cosine_similarity

# Пример предложений для сравнения
sentence1 = "Сегодня очень холодно на улице"
sentence2 = "На улице сегодня очень холодно"

# Вычисление косинусной меры угла между предложениями
cosine_similarity = calculate_cosine_similarity(sentence1, sentence2)

# Вывод результата
print(f"Косинусная мера угла между предложениями: {cosine_similarity}")
```

 [2.Расчет косинусной меры между предложениями.ipynb](#)

Спасибо! Уверены, что вы хорошо постарались и поработали!

Предлагаем свериться с возможным вариантом решения.

```
import spacy
import numpy as np

nlp = spacy.load("ru_core_news_lg")

def cosine_similarity(s1, s2):
    doc1 = nlp(s1)
    doc2 = nlp(s2)
    vec1 = doc1.vector
    vec2 = doc2.vector
    if np.linalg.norm(vec1)*np.linalg.norm(vec2) > 0:
        return np.dot(vec1, vec2) / (np.linalg.norm(vec1)*np.linalg.norm(vec2))
    else:
        return 0

s1 = "Сегодня очень холодно на улице"
s2 = "На улице сегодня очень холодно"
similarity = cosine_similarity(s1, s2)
print(f"Косинусная мера угла между предложениями '{s1}' и '{s2}': {similarity}")

s1 = "Обезьяны любят бананы"
s2 = "Собаки не любят молоко"
similarity = cosine_similarity(s1, s2)
print(f"Косинусная мера угла между предложениями '{s1}' и '{s2}': {similarity}")
```

После выполнения программы мы получаем следующий результат:

- Косинусная мера угла между предложениями «Сегодня очень холодно на улице» и «Не сегодня очень холодно»: 0.9280714988708496.
- Косинусная мера угла между предложениями «Обезьяны любят бананы» и «Собаки не молоко»: 0.646365225315094.

Как видим, первые два предложения почти идентичны, поэтому значение косинусной меры угла Вторые два предложения различны, поэтому значение косинусной меры угла близко к 0,6.

Вопрос **3**
Выполнен
Балл: 1,00

Анализ похожих товаров по их описанию

Допустим, вы аналитик данных в компании, которая занимается продажей мебели. Ваша задача определить, какие товары наиболее похожи друг на друга по описанию. Для этого необходимо использовать косинусную меру угла с помощью библиотеки `spacy`.

Шаги выполнения задания:

1. Скачайте датасет с описанием товаров (исходный файл — [product_description.csv](#)).
2. Импортируйте библиотеку `spacy` и загрузите модель языка `en_core_web_sm`.

Дополнительно для выполнения задания выполните импорт функций из библиотек `Pyth`

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
from sklearn.metrics.pairwise import cosine_similarity
```

3. Проведите предобработку текста: удалите стоп-слова, лемматизируйте слова, удалите пунктуацию.

Используйте следующий код для предобработки текста:

```
def preprocess_text(text):
```

```
    doc = nlp(text)
```

```
    tokens = [token.lemma_.lower() for token in doc if not token.is_stop and not token.is_punct]
```

```
    return " ".join(tokens)
```

```
data['processed_text'] = data['description'].apply(preprocess_text)
```

4. Создайте матрицу векторов для каждого товара.

Используйте следующий код для векторизации:

```
vectorizer = TfidfVectorizer()
```

```
vectors = vectorizer.fit_transform(data['processed_text'])
```

5. Рассчитайте косинусную меру угла между каждой парой товаров.
6. Отобразите топ-5 товаров, которые наиболее похожи друг на друга.

```
import pandas as pd
```

```
import numpy as np
```

```
# путь к файлу с данными
```

```
file_path = './product_description.csv'
```

```
data = pd.read_csv(file_path)
```

```
import spacy
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
from sklearn.metrics.pairwise import cosine_similarity
```

```
nlp = spacy.load('en_core_web_sm')
```

```
def preprocess_text(text):
```

```
    doc = nlp(text)
```

```
    tokens = [token.lemma_.lower() for token in doc if not token.is_stop and not token.is_punct]
```

```
    return " ".join(tokens)
```

```
data['processed_text'] = data['description'].apply(preprocess_text)
```

```
vectorizer = TfidfVectorizer()
```

```
vectors = vectorizer.fit_transform(data['processed_text'])
```

```
cosine_similarities = cosine_similarity(vectors, vectors)
```

```
## # Создайте DataFrame для удобства работы с результатами
```

```
cosine_similarity_df = pd.DataFrame(cosine_similarities, columns=data['product_name'],
```

```
index=data['product_name'])
```

```
# Выведите топ-5 похожих товаров для каждого товара
```

```
for product_name in data['product_name']:
```

```
    similar_products = cosine_similarity_df[product_name].sort_values(ascending=False)[1:6]
```

```
    print(f"\nТовар: {product_name}\nТоп-5 похожих товаров:\n{similar_products}")
```

Спасибо! Уверены, что вы хорошо постарались и поработали!

Предлагаем свериться возможным вариантом решения.

```
import spacy
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Загрузка модели языка
nlp = spacy.load("en_core_web_sm")

# Загрузка датасета с описанием товаров
data = pd.read_csv('product_description.csv')

# Предобработка текста
def preprocess_text(text):
    doc = nlp(text)
    tokens = [token.lemma_.lower() for token in doc if not token.is_stop and not token.is_punct]
    return " ".join(tokens)
data['processed_text'] = data['description'].apply(preprocess_text)

# Создание матрицы векторов для каждого товара
vectorizer = TfidfVectorizer()
vectors = vectorizer.fit_transform(data['processed_text'])

# Расчет косинусной меры угла между каждой парой товаров
cosine_similarities = cosine_similarity(vectors)

# Отображение топ-5 товаров, которые наиболее похожи друг на друга
for i, row in data.iterrows():
    similar_indices = cosine_similarities[i].argsort()[::-6:-1]
    similar_items = [(cosine_similarities[i][j], data['product_name'][j]) for j in similar_indices if j != i]
    print(f"Top 5 similar items for {row['product_name']}: \n{similar_items} \n")
```

Если у вас возникли вопросы по заданию, пожалуйста, обратитесь к преподавателю на ближайш семинаре.

Желаем продуктивного обучения!