

Scrapowanie wakacje.pl oraz przetwarzanie tekstu

Spis treści

1. [Wprowadzenie](#)
2. [Sprawozdanie](#)
 - [Cele i wymagania](#)
 - [Wykorzystane narzędzia](#)
 - [Zobrazowanie działania aplikacji](#)
 - [Podsumowanie](#)
3. [Dokumentacja techniczna](#)
 - [Instalacja](#)
 - [Uruchomienie aplikacji](#)

Wprowadzenie

Projekt obejmuje stworzenie aplikacji, która analizuje opinie dotyczące różnych hoteli ze strony wakacje.pl. Opracowano również narzędzia do przetwarzania tekstu, takie jak tokenizacja, lematyzacja, usuwanie stopwords oraz analizę częstości słów. Poniżej przedstawiono skrócony opis głównych komponentów projektu.

Sprawozdanie

Cele i wymagania

Celem projektu jest stworzenie systemu, który pozwala na analizę opinii o hotelach zebranej z witryny wakacje.pl. System ma za zadanie zbierać dane z różnych hoteli, przetwarzać je, a następnie generować statystyki i wizualizacje dotyczące częstości występowania słów oraz ogólnej struktury opinii. Algorytm ma za zadanie przeprowadzić poniższe operacje:

1. Utworzenie korpusu językowego z pobranych tekstów - korpus językowy to zbiór tekstów używany do analizy języka naturalnego. Może zawierać różnorodne teksty, takie jak artykuły, książki, czy rozmowy, służące do badania struktury językowej, słownictwa, czy innych właściwości języka.
2. Tokenizacja - proces podziału tekstu na pojedyncze jednostki, zwane tokenami. Tokeny mogą reprezentować pojedyncze słowa lub inne fragmenty tekstu, takie jak zdania. Tokenizacja jest często pierwszym krokiem w analizie tekstu.
3. Lematyzacja - proces zamiany słów na ich lematy, czyli formy podstawowe. Lemat to forma słowa, która reprezentuje jego podstawowy rdzeń. Ten proces pomaga w redukcji słów do ich podstawowej postaci, co ułatwia analizę tekstu.
4. Stemming - proces usuwania końcówek słów, aby uzyskać ich formę podstawową (tzw. rdzeń). Jest to bardziej agresywne podejście niż lematyzacja, ponieważ nie zawsze prowadzi do poprawnych form podstawowych.

5. Usuwanie słów z listy stopu - lista stopu zawiera słowa, które są często używane w języku, ale zazwyczaj nie niosą istotnego znaczenia, takie jak przyimki, spójniki, zaimki, etc. Usuwanie tych słów z tekstu pomaga skoncentrować się na bardziej istotnych aspektach analizy.
6. Analiza statystyczna oraz wizualizacja - analiza statystyczna obejmuje wykorzystanie metod statystycznych do zrozumienia i wyciągania wniosków z danych tekstowych. Wizualizacja danych tekstowych może obejmować tworzenie wykresów, diagramów, czy innych form graficznych prezentujących informacje zawarte w tekście.
7. Wektoryzacja - proces reprezentowania tekstu jako wektora liczb. W tekście każde słowo lub fragment tekstu jest przypisywane do odpowiedniego wektora numerycznego. Ten proces jest kluczowy w wielu technikach analizy tekstu, takich jak modelowanie tematów czy klasyfikacja dokumentów.

Wykorzystane narzędzia

1. **Język programowania Python** Python to język programowania wysokiego poziomu ogólnego przeznaczenia, o rozbudowanym pakiecie bibliotek standardowych, którego ideą przewodnią jest czytelność i klarowność kodu źródłowego. Jego składnia cechuje się przejrzystością i zwięzłością.
2. Biblioteka BeautifulSoup jest wykorzystywana do parsowania i analizy strony internetowej. Pozwala na wygodne ekstrakowanie danych z kodu HTML, co jest istotne podczas scrappingu informacji z witryny wakacje.pl.
3. Biblioteka spaCy została użyta do przetwarzania języka naturalnego. Zawiera modele do analizy gramatycznej, tokenizacji oraz lematyzacji tekstu w języku polskim. Umożliwia efektywne przetwarzanie dużych ilości danych tekstowych.
4. Biblioteka scikit-learn jest używana do analizy tekstu, zwłaszcza do wektorowania tekstu i tworzenia macierzy cech. Wykorzystuje się w niej moduły takie jak CountVectorizer oraz TfidfVectorizer do konwersji zbioru dokumentów tekstowych na reprezentację liczbową.
5. Biblioteka nltk dostarcza narzędzi do przetwarzania języka naturalnego, takich jak tokenizacja czy stemming. W projekcie użyto stemmera PorterStemmer do przekształcania słów na ich podstawową formę.
6. Biblioteka wordcloud została użyta do generowania chmur słów (word clouds) wizualizujących najczęściej występujące słowa w analizowanym tekście.
7. Biblioteka matplotlib jest wykorzystywana do generowania różnych wykresów, w tym słupkowych, co pozwala na wizualizację wyników analizy tekstu.
8. Opisy funkcji (docstrings) są wygenerowane przy użyciu dodatku do VSC opartego na AI.

Zobrazowanie działania aplikacji

1. Scraper wakacje.pl

Poprzez uruchomienie kodu z pliku scrap_data.py aplikacja przy użyciu pakietu bs4 pobiera opinie o chotelach z portalu wakacje.pl. Targetem scrapera jest strona <https://www.wakacje.pl/hotele/> z której dynamicznie pobierane są linki do 10 hoteli uznawanych przez platformę za najczęściej oceniane. Następnie skrypt iteruje przez wszystkie 10 hoteli zapisując opinie o nich do pliku 1_opinions.json. Skrypt uwzględnia

paginację występującą na portalu i iteruje przez wszystkie dostępne strony z opiniami (na jednej stronie znajduje się max 20 opinie).

2. Analiza języka

Po uruchomieniu pliku `main.py` program tworzy korpus językowy łącząc wszystkie opinie w jeden zbiorczy tekst i etap ten zapisywany jest do pliku `2_corpus.txt`. Następnie przeprowadzana jest tokenizacja czyli wydzielane są pojedyncze słowa i zapisywane do pliku `3_tokenized.txt` w formacie 1 linia = 1 token (słowo). Kolejnymi etapami są przeprowadzone kolejno lematyzacja czyli zamiana słów na formy podstawowe oraz stemming - usuwanie końcówek słów. Po zakończeniu powyższych procesów, dane po każdym etapie zapisywane są odpowiednio do plików `4_lemmatized.txt` oraz `5_stemmed.txt`. Do przeprowadzenia obu operacji wykorzystywana jest biblioteka `spacy` z zaimportowanymi paczkami do języka polskiego. Idąc dalej, skrypt tworzy plik `6_filtered.txt`, który zawiera tokeny z usuniętymi słowami, które powtarzają się zbyt często lub są zbędne dla analizy tekstu.

3. Wizualizacja

Tokeny poddane lematyzacji i filtrowaniu są przekazywane do funkcji która analizuje częstotliwość występowania poszczególnych słów w opiniach gości hotelowych.

W wyniku działania aplikacji, w konsoli wyświetlana jest liczba wszystkich słów oraz unikalnych, a następnie generowane są 2 pliki PNG, które są zamieszczone poniżej. Jeden z nich pokazuje częstotliwość występowania słów w formie wykresu typu wordcloud, druga zaś to tradycyjny wykres kolumnowy top 0,3% najpopularniejszych słów.

7_visualize_1 7_visualize_2

4. Wektoryzacja

Na koniec przeprowadzana jest wektoryzacja przy użyciu pakietu `sklearn` i importowanej funkcji `CountVecorized`, a wynik zapisywany jest do pliku `8_vectorized.txt`

Do wszystkich wyżej wymienionych etapów, przykładowe dane zostały zapisane w folderze `example_resultes` z odpowiadającymi nazwami poszczególnych plików.

Podsumowanie

Projekt skupia się na analizie opinii dotyczących hoteli zebranych z witryny `wakacje.pl`. Główne cele projektu obejmują skuteczne pobieranie danych z witryny, przetwarzanie języka naturalnego oraz generowanie statystyk i wizualizacji.

Scraper, oparty na bibliotekach `requests` i `BeautifulSoup`, zbiera opinie o hotelach ze strony `wakacje.pl`, a następnie dane są przetwarzane przy użyciu narzędzi do analizy tekstu. Biblioteka `spaCy` wspomaga tokenizację, lematyzację i usuwanie stopwords, natomiast `scikit-learn` umożliwia wektorowanie tekstu i analizę częstości słów. Dodatkowo, `nlTK` dostarcza stemmera do przetwarzania słów, a biblioteki `wordcloud` i `matplotlib` pomagają w wizualizacji wyników.

Dokumentacja techniczna

Instalacja

1. Pobierz repozytorium z kodem i otwórz folder z projektem przy użyciu dowolnego IDE (na przykład Visual Studio Code).
2. Utwórz wirtualne środowisko i aktywuj je.

```
# Utworzenie środowiska wirtualnego
python -m venv /path/to/new/virtual/environment
# Aktywacja środowiska wirtualnego
.\env\Scripts\activate
```

3. Używając menedżera pakietów [pip](#), zainstaluj wymagane biblioteki.

```
pip install -r requirements.txt
```

Uruchomienie aplikacji

1. Aktywuj środowisko wirtualne.

```
# Aktywacja środowiska wirtualnego
.\env\Scripts\activate
```

2. Przejdź do folderu, który zawiera plik main.py.
3. Uruchom scraper lub przetwarzanie tekstu używając poniższego polecenia :

```
python3 main.py
python3 scrap_data.py
```