Laboratorium

Przetwarzanie tekstu: bag of words, sentyment, sieci rekurencyjne

Zadanie 1

Zapoznaj się ze stronami NLTK: https://www.nltk.org/ oraz https://www.nltk.org/book/ . Na potrzeby tego zadania przejrzyj też poniższe samouczki.

- https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk
- https://realpython.com/python-nltk-sentiment-analysis/
- https://www.geeksforgeeks.org/tokenize-text-using-nltk-python/
- https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
- https://www.geeksforgeeks.org/python-lemmatization-with-nltk/

Punkty do wykonania:

a) Wybierz dowolny i niezbyt krótki artykuły z dowolnego portalu angielskojęzycznego (BBC, NBC, Nature lub inne) . Temat artykułu dowolny – może być polityczny, społeczny, naukowy. Skopiuj go (ręcznie) i zapisz w pliku txt.

b) Dokonaj tokenizacji dokumentu. Podaj liczbę słów po tym etapie.

c) Usuń stop-words z artykułu używając standardowej listy dla słów angielskich. Podaj liczbę słów po tym etapie.

d) Sprawdź czy w naszym zestawie słów (bag of words) są jeszcze jakieś pominięte niepotrzebne słowa. Wówczas dodaj do listy stopwords dodatkowe słowa ręcznie (np. za pomocą komendy append lub extend). Podaj liczbę słów po tym etapie.

e) Dokonaj lematyzacji dokumentu. Jaki lematyzer został wybrany? Alternatywnie: możesz dokonać stemmingu. Podaj liczbę słów po tym etapie.

f) Podaj przetworzony dokument w formie wektora zliczającego słowa. Następnie wyświetl na wykresie słupkowym 10 najczęściej występujących słów (oś X: słowa, oś Y: liczba wystąpień słowa w tekście).

g) Stwórz chmurę tagów (word cloud) dla Twojego dokumentu. Pomocne linki:
https://www.datacamp.com/community/tutorials/wordcloud-python
https://amueller.github.io/word_cloud/
https://pypi.org/project/wordcloud/

Zadanie 2

Wykorzystaj paczkę NLTK Vader (https://www.nltk.org/_modules/nltk/sentiment/vader.html, https://www.nltk.org/howto/sentiment.html) do sprawdzenia jak radzi sobie z analizą opinii/sentymentu. Następnie porównaj to z

a) Wejdź na stronę z hotelami (np. https://www.booking.com/ , https://www.tripadvisor.com/) i znajdź jedną pozytywną opinię o jakimś hotelu i jedną zdecydowanie negatywną. Wybierz opinie w języku angielskim składające się z przynajmniej kilku zdań.

b) Używając narzędzia Vader sprawdź w jakim stopniu obie opinie są pozytywne (pos), negatywne (neg), neutralne (neu) i jaki jest wynik zagregowany wszystkich opinii (compound), który waha się od -1 (negatywny) do 1 (pozytywny).

c) Teraz wykorzystaj paczkę Text2Emotion, żeby sprawdzić jak obie opinie są tagowane wg pięciu emocji.

d) Czy wyniki dla obu narzędzi są zgodne z oczekiwaniami?

e) Spróbuj dodać parę pikantnych słów do obu tych recenzji, tak aby oceny sentymentu były silniejsze i powtórz eksperyment dla obu narzędzi.

Zadanie 3

Analizowaliśmy pojedyncze teksty, ale żeby robić poważne badania trzeba mieć większe bazy dany tekstu lub narzędzia do pozyskiwania tekstów.

Szczególnie ciekawe do badań są tweety (twitter.com), które są zwięzłymi tekstami otagowanymi dodatkowo hasztagami. Można ich pobrać setki, mieć mnóstwo różnych opinii i punktów widzenia na przeróżne tematy.

Celem tego zadania jest automatyczne pobranie przynajmniej 100 tweetów na wybrany temat. Można to zrobić za pomocą różnych narzędzi:

- Tweepy (popularne, wykorzystujące Twitter Api, wymaga posiadania konta developera, niestety można ściągać tweety tylko z ostatnich dni)
- Snscrape (mniej popularna, nie wymaga posiadania konta developera, więc to „szara strefa" pozyskiwania danych, nie ma ograniczeń czasowych)

a) Wybierz temat tweetów (może być jakiś konkretny hasztag) np. #ukraine, #christmas, #blackfriday lub inny

b) Wykorzystaj Tweepy lub Snscrape do pozyskania około 100 tweetów na dany temat. Wyświetl te tweety.

c) Zbadaj czy można pobierać tweety z różnych okresów czasu, różnych lokalizacji. Spróbuj zebrać dodatkowo 50 tweetów z wybranego okresu czasu z okolic Gdańska.

Zadanie 4

Wykonaj następujące polecenia związane z sieciami rekurencyjnymi:

a) Uruchom i przeanalizuj pliki z wykładu o sieciach rekurencyjnych:
   - rnn01.py (prosta demonstracja)
   - rnn02.py (Badanie jak działa sieć rekurencyjna)
   - rnn03.py (Przewidywanie liczby plam na słońcu w danym miesiącu -RNN)
   - lstm01.py (Przewidywanie liczby plam na słońcu w danym miesiącu - LSTM)
   - lstm02.py (Uczenie generowania tekstu przez LSTM litera po literze)
   - lstm03.py (Uruchomienie generatora LSTM z poprzedniego zadania)
   - lstm04.py (Uczenie generowania tekstu przez LSTM słowo po słowie)
   - lstm05.py (Uruchomienie generatora LSTM z poprzedniego zadania)

   Uwaga: w zadaniach lstm02.py i lstm04.py uczenie może trwać wiele godzin. Znacznie zmniejsz liczbę epok, by trenowanie trwało parę minut.

b) Zademonstruj jak działają generatory tekstu (lstm03 i lstm05) po Twoim krótkim treningu.

c) Dotrenuj model z lstm02 i lstm04 o parę epok, wykorzystując jako bazę startową już wytrenowany przez Ciebie model zapisany w pliku hdf5. Następnie pokaż, czy generowany tekst jest trochę lepszy.