Laboratorium 3. Podobieństwo w poezji

Dzisiejsze laboratorium przedstawi od strony praktycznej zadanie wyszukiwania. Głównym elementem wyszukiwarki jest zdolność do oceny podobieństwa pomiędzy dokumentami. Dobrym sposobem jest budowa przestrzeni wektorowej. Podczas zajęć zademonstrowane zostaną możliwości biblioteki Gensim, która pozwala na projektowanie własnych wyszukiwarek.

Zadanie 3.1. Word2Vec – podobne słowa

1. Wczytaj wszystkie dokumenty z katalogów `adam`, `jan` oraz `juliusz` (zapisz nazwy plików na potem!)
2. Dokonaj podziału na tokeny z word_tokenize oraz usuń tokeny z listy `stopwordsPL.txt`.
3. W każdym tokenie pozostaw tylko znaki alfranumeryczne (`filter`, `str.isalnum`).
4. Utworz listę zawierającą listy z tokenami dla poszczególnych dokumentów
5. Zainicjuj model `Word2Vec` ustawiając parametry:
   - sentences=lista
   - vector_size=16
   - window=5
   - min_count=1
6. Model wytrenuj w **trzech** krokach tokenami z podziałem na 'adam', 'jan', 'juliusz' w **16** epokach.
7. Za pomocą `wv.similarity` wyznacz podobieństwa dla słów:
   - wiatr – fale
   - trawie – zioła
   - zbroja – szalonych
   - cichym – szeptem
8. Rezultat omów w pliku z kodem źródłowym za pomocą komentarzy.

Dokumentacja

- `word_tokenize`[1]
- `filter`[2]
- `str.isalnum`[3]
- `Word2Vec`[4]

Zadanie 3.2. Word2Vec – wektory nie tylko słów

1. Dla każdego dokumentu z poprzedniego zadania:
   - Wyznacz wektor każdego słowa za pomocą modelu z poprzedniego zadania (model.wv["słowo"]).
   - Wylicz średnią wartość wektorów dla jednego dokumentu
2. Wyznacz podobieństwo cosinusowe[5] pomiędzy dokumentami. Użyj do tego własnych obliczeń na macierzach.
3. Wyniki wyświetl w formie macierzy lub listy par nazw dokumentów i ich wartości podobieństwa.
4. Wskaż najbardziej i najmniej podobne dokumenty, pomijając podobieństwo do samego siebie.
5. Rezultat omów w pliku z kodem źródłowym za pomocą komentarzy.

Zadanie 3.3. Doc2Vec – nie jedna etykieta

1. Na podstawie danych z zadania pierwszego zbuduj model Doc2Vec (parametry `vector_size=32` oraz `window=5`)
2. Do funkcji trenowania przakaż listę obiektów `TaggedDocument`[6], jako tag użyj nazwy pliku.
3. Wyznacz wektory dla wszystkich dokumentów i wskaż najbardziej oraz najmniej podobne dokumenty, pomijając podobieństwo do samego siebie.
4. Rezultat porównaj z poprzednim zadaniem, a obserwacje zapisz w pliku z kodem źródłowym za pomocą komentarzy.
5. Sprawdź zmianę rezultatów, gdy oprócz nazwy dokumentu podane zostanie również imię autora jako dodatkowy tag oraz wektory zostaną wyznaczone w sposób inferencyjny.

Dokumentacja

- `Doc2Vec`[7]

[1] <https://www.nltk.org/api/nltk.tokenize.html>
[2] <https://docs.python.org/3/library/functions.html#filter>
[3] <https://docs.python.org/3/library/stdtypes.html#str.isalnum>
[4] <https://radimrehurek.com/gensim/models/word2vec.html>
[5] <https://en.wikipedia.org/wiki/Cosine_similarity>
[6] <https://radimrehurek.com/gensim/models/doc2vec.html#usage-examples>
[7] <https://radimrehurek.com/gensim/models/doc2vec.html>