Text mining
Pracownia 2
Zajęcia 28.04 i 12.05

Uwaga: Na zajęciach 5.05 będą **ćwiczenia**!

Zadanie 1. (5p) Dodaj do wyszukiwarki Wikipedyjki indeks pozycyjny. Każde zapytanie powinno być traktowane jako pytanie o frazę, przy czym powinieneś obsługiwać odmianę, tzn. pytanie *skoki narciarskie* powinno zwracać również dokumenty zawierające frazę *skoków narciarskich*. W wypisywanych fragmentach dokumentów powinny być wyróżnione (najlepiej kolorem) trafienia całej frazy. Kolejność wypisywania dokumentów może być dowolna.

Zadanie 2. (7p) Zadanie to jest rozwinięciem zadania poprzedniego. Wyszukiwarka, którą piszesz w tym punkcie powinna:

a) obsługiwać zapytania zwykłe i frazowe,

b) rozpoznawać typ zapytania po obecności cudzysłowów,

c) obsługiwać zapytania frazowe w dowolny sposób, niekoniecznie indeksem pozycyjnym,

d) tworzyć ranking dla zapytań zwykłych, który uwzględnia cechy z poprzedniej listy zadań,

e) premiując dodatkowo dokumenty zawierające frazy z zapytania.

Oznacza to w szczególności, że w przypadku zapytania zwykłego, będącego frazą, dokumenty zawierające tę frazę powinny być wysoko na liście odpowiedzi. Należy również inteligentnie obsługiwać takie pytania jak:

> *kodeks karny kara za morderstwo*

i premiować dokumenty zawierające np. takie zdanie

> *w ostatniej nowelizacji kodeksu karnego wprowadzono podwyższoną karę za morderstwo*,

a nie premiować takiego

> *Karni żołnierze mają kodeks, którego przekroczenie powoduje karę, a za jakiś czas nawet morderstwo .*

Zadanie 3. (5p) Napisz program, który przeglądając 1-gramy i 2-gramy (nkjp ngrams) z języka polskiego znajdzie możliwie najwięcej sytuacji, w których popełniono literówkę związaną z wstawieniem, bądź pominięciem spacji. Wystarczy, że znajdziesz w sumie około 10 tysięcy błędów, ale postaraj się, by były one możliwie jak najbardziej wiarygodne. Przykładowo, dla zadania wstawiania spacji:

- Powinieneś znaleźć takie przykłady jak: wielkiegopomorska, socjologiiuniwersytetu, otwartapracownia, przezgrzechy
- **Nie** powinieneś znajdować: antysystemową, supertygrysa, wewnątrzoddziałowego, wschodniokarpackiego

Zadanie 4. (6p) Napisz program, korygujący błędy, który czyta ze standardowego wejścia plik zawierający wiersze, w których mamy pary: (słowo-poprawne, słowo-wpisane). Przeczytawszy parę, dokonuje korekty i sprawdza, czy jest taka jak trzeba. Wypisuje błędy i podlicza na koniec ich procentowy udział we wszystkich korektach. Oczywiście korzysta z informacji o poprawnym słowie **tylko** do sprawdzenia, czy popełnił błąd. Dla zbioru testowego 1 powinien mieć skuteczność ponad 70%. Wymagana skuteczność dla zbioru testowego 2 zostanie podana później.

Uwaga: nie wolno spamiętywać żadnych danych z poprzednich korekt (dane są tak skontruowane, że taka strategia dawałaby „nieuczciwą" przewagę programowi z niej korzystającemu). Uwaga: na przyszłych zajęciach zrobimy osobno punktowany konkurs poprawiania literówek.

Zadanie 5. (4p) Napisz program, który **sprawdza**, jaka jest wielkość list postingowych dla słów z Wikipedyjki, gdy:

a) Kodujemy numery dokumentów za pomocą `int32`.

b) Kodujemy różnice numerów dokumentów za pomocą kodu VB.

c) Kodujemy różnice numerów dokumentów za pomocą kodu VB, operującego 4-bitowymi (a nie 8)- porcjami danych.

d) Kodujemy różnice numerów dokumentów za pomocą kodu Gamma (wybierz najbardziej wydajny wariant, biorąc pod uwagę że nie musisz kodować liczby 0). Zaokrąglaj wielkości całych list postingowych do liczby bitów podzielnej przez 8.

Ważne: wynikiem programu mają być 4 liczby, nie musisz zatem implementować tych kodów. Utożsamiaj małe i duże literki, ale możesz się nie przejmować tokenizacją (wystarczy split, choć możesz też zobaczyć, czy zmiana tokenizacji na wziętą z NLTK coś zmieni).

Zadanie 6. (1p) Zgłoś swoje propozycje korekt na SKOS.