# Text mining
# Pracownia 3
# Zajęcia przed Wielkanoca i drugie zajęcia po Wielkanocy

**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. (3p+1)** Zmodyfikuj schemat odpowiedzi na pytania z programu `baseline.py` w następujący sposób:

> Zadając pytania usuwaj nie najbardziej lewe słowo, tylko słowo o najmniejszym IDF

Porównaj skuteczność z oryginalnym schematem, tworząc raport, zawierający pytania (wraz z odpowiedziami), w których działanie obu schematów się różni (oczywiście powinno być zaznaczone, która odpowiedź jest właściwa.

Odpowiedz na pytanie: czy widzisz jakąś możliwość połączenia obu schematów (lub stworzenia innego schematu skracania pytań)? Opcjonalnie: zaimplementuj tę inną strategię za jeden punkt bonusowy i stwierdź, czy coś pomogła.

**Zadanie 3. (3p+1)** Wykorzystaj pytania frazowe (dowolnie zaimplementowane[1]) w systemie odpowiadania na pytania. Punk bonusowy jest za uzyskanie poprawy działania dzięki frazom.

**Zadanie 4. (2p+2)** Dodaj do Twojego systemu odpowiadania na pytania dowolną funkcjonalność związaną z udzielaniem odpowiedzi nie będących tytułami (lecz fragmentami treści). To mogą być na przykład pytania o kraj, ocean, czy kontynent (lub jakieś inne). Punkty bonusowe zależą od tego, jak zaawansowany będzie Twój system i powinny być jakoś skorelowane z wysiłkiem, jaki został włożony do jego utworzenia.

**Zadanie 5. (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 6. (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.

[1] To znaczy nie musisz rozwiązywać zadania 1. Oczywiście możesz z niego skorzystać, ale inne opcje – filtr treści, indeks dla fraz – są również akceptowalne