Eksploracja tekstów
ćwiczenia 3
17 i 18 maja 2022

Zadanie 1. Dysponujesz dużą liczbą par (słowo-wpisane, słowo-poprawne). Jak wykorzystać ten zbiór do ustalenia kosztów następujących operacji:

- Wpisania wariantu polskawego literki
- Omyłkowego podwojenia literki
- Wpisania *dbca* zamiast *abcd* (dla dowolnych a, b, c oraz d)

Zadanie 2. Wymień jak najwięcej różnych pod względem gramatycznym sytuacji, w których rzeczownik łączy się z innym wyrazem, tworząc frazę rzeczownikową o długości dwa (co najmniej 8). Frazy wymieniamy w formie „słownikowej", czyli „piękna pogoda", a nie „pięknej podody".

Zadanie 3. Piszesz wyszukiwarkę, korzystającą z list postingowych, obsługującą prostą koniunkcję (czyli znajdującą dokumenty z wszystkimi termami). Jak dodać do niej w sposób możliwie efektywny obsługę negacji – żebyśmy mogli pytać zadawać zapytania typu `+w1 -w2 +w3 -w4`, chcąc otrzymać dokumenty, które zawierają w_1 oraz w_3, ale nie zawierają w_2 oraz w_4.

Zadanie 4. Załóżmy, że mamy zaimplementowaną wyszukiwarkę bazującą na zwykłym *proximity search* (czyli znajdującą dokumenty, w których termy z zapytania mieszczą się w oknie o wielkości k). Jak wykorzystując ten mechanizm zaimplementować wyszukiwarkę, która:

a) Znajduje dokumenty, w których wszystkie termy z zapytania są w jednym zdaniu.

b) Znajduje dokumenty, w których premiowana jest obecność wszystkich termów w jednym wątku dokumentu (czyli spójnym kawałku na 1 temat).

Zadanie 5. Załóżmy, że słowa zapisujemy w drzewie trie. Jaką korzyść dla zapytań z „gwiazdką"[1] możemy otrzymać jeżeli do tego drzewa włożymy nie słowo *krowa*, lecz słowa: *$krowa*, *a$krow*, *wa$kro*, *owa$kr*, , *rowa$k?*

Zadanie 6. Wyjaśnij, jak drzewo trie może pomóc efektywnie znajdywać słowa w określonej odległości edycyjnej od danego słowa. Uwaga: drzewo powinno być używane podczas generowania, a nie jedynie jako filtr odpowiadający na pytania: „czy w jest poprawnym słowem".

Zadanie 7. Załóżmy, że mamy K rozłącznych zbiorów dokumentów (na przykład wiadomości sportowe, plotki dotyczące celebrytów, nowinki naukowe, artykuły polityczne, itp). Zapisujemy je jako wektory ze współrzędnymi tf-idf. Dla nowego dokumentu X (którego chcemy przypisać do któregoś z tych zbiorów) znajdujemy tę klasę, dla której średnie podobieństwo cosinusowe między X a elementami tej klasy jest największe.

Jaka jest złożoność tego algorytmu? Jak ją poprawić, przy założeniu, że możemy wykonać pewne obliczenia podczas „tworzenia indeksu", których czasu nie uwzględniamy?

Zadanie 8. Załóżmy, że mamy K rozłącznych zbiorów dokumentów (na przykład wiadomości sportowe, plotki dotyczące celebrytów, nowinki naukowe, artykuły polityczne, itp). Zapisujemy je jako wektory ze współrzędnymi tf-idf. Dla nowego dokumentu X (którego chcemy przypisać do któregoś z tych zbiorów) znajdujemy tę klasę, dla której średnie podobieństwo cosinusowe między X a elementami tej klasy jest największe.

Jaka jest złożoność tego algorytmu? Jak ją poprawić, przy założeniu, że możemy wykonać pewne obliczenia podczas „tworzenia indeksu", których czasu nie uwzględniamy?

Zadanie 9. ($\star$) Znajdź w Internecie informację na temat Vantage Point Trees i przedstaw tę strukturę danych. W szczególności wyjaśnij, jak można ją wykorzystać do znajdywania podobnych dokumentów do danego.

[1] Czyli takich, w których część wyrazu jest zamieniona na znak *