Text mining
Ćwiczenia 1
Zajęcia 3

Zadanie 1. (1p) Zdefiniuj formę superbazową dla słowa (tzn. powiedz, jak ją wyznaczać, mając daną formę bazową). Forma superbazowa ma spełniać następujące założenia:

i) Ma być jednoznaczna, tzn. każde słowo powinno mieć dokładnie jedną formę superbazową.

ii) Słowa o tym samym lemacie (formie bazowej) mają tę samą formę superbazową.

iii) Jak najwięcej słów o różnych lematach ma różne formy superbazowe.

Powiedz, dlaczego forma superbazowa może być użyteczna (lub uzasadnij, że do niczego się nie przyda).

Zadanie 2. (1p) Jak efektywnie wyznaczać formę superbazową? Wskazówka[1]: Gb mncrjar olłb an cbpmągxbjlpu jlxłnqnpu m NvFQ. Natvryfxn anmjn wrfg qjhpmłbabjn, qehtv pmłba gb fłbjb "manwqź".)

Zadanie 3. (1p) Zaproponuj algorytm generacji reguł dla stemmera, który korzysta z

a) dużej tabeli zawierającej pary (słowo, forma-bazowa)

b) tabeli zawierające powiązane formy bazowe, np. (nauczyciel, nauczycielka), (plaża, plażować), (czytać, czytanie).

Algorytm powinien generować reguły postaci:

```
<tekst-ze-słowa> -> <tekst-na-który-go-zastępujemy>
```

przy czym `<tekst-ze-słowa>` może zawierać sztuczne znaki początku i końca słowa. Reguły mogą zawierać też informacje o wymaganej wielkości słowa, dla którego stosujemy warunki. Zaproponuj jakąś metodę testowania, czy reguły są użyteczne (tzn. czy utożsamianie różnych słow o tym samym stemie poprawia komfort pracy z wyszukiwarką).

Zadanie 4. (1p) Załóżmy, że tokenizator jest programem, który dzieli tekst na części, spełniając następujący warunek:

```
''.join(txt.split()) == ''.join(tokenize(txt))
```

Zdefiniuj tokenizację „minimalną", czyli taką, w której mamy jak najmniejsze tokeny (aczkolwiek słowa pozostają jednym tokenem). Wymień kilka sytuacji (wystarczą 3), w których wydaje Ci się sensowne łączenie ciągów mikrotokenów w większe całości.

Zadanie 5. (1p) Dlaczego może być istotne nie tylko to, czy term znajduje się w dokumecie, ale również to, ile razy.

Zadanie 6. (1p) Załóżmy, że liczbę wystąpień danego termu w dokumencie musisz zapamiętać w jednym bajcie. Zaproponuj sposób, który to umożliwi (podaj sposób kodowania i odkodowywania, oczywiście niektóre liczby mogą nie być pamiętane dokładnie). Wykorzystaj ten sposób do stworzenia modyfikacji metody kompresji list postingowych[2], w której lista nadal jest jednym napisem, niemniej zawiera również informację o tym, ile razy dany term wystąpił w dokumencie.

Zadanie 7. (1p) Słownik używany w odwrotnym indeksie można kodować jako drzewo trie. Jest to szczególnie istotne, jeżeli będziemy chcieli umieszczać w nim nie tylko wyrazy, ale również frazy. Wyjaśnij, co to jest drzewo trie, jak je zaimplementować, w dwóch wariantach:

[1] rot13.com

[2] Na wykładzie była określona skrótem VB

a) wariant dla wyrazów, które pamiętamy po literce

b) wariant dla fraz, w którym najmniejszą jednostką jest słowo

Zadanie 8. (1p) Możemy rozważać 4 metody obsługi zapytań frazowych:

a) zwykły indeks + postprocessing,

b) indeks bigramowy (dwusłowowy),

c) indeks pozycyjny,

d) indeks frazowy.

Przypomnij krótko, jak te metody działają. W rzeczywistym systemie, metody te mogą być łączone. Zaproponuj dwie kombinacje ww. metod, powiedz, jak je zaimplementować i jakie mogą dać korzyści.

Zadanie 9. (1p) Opisz jak wykorzystać indeks pozycyjny, aby wykonać tzw. *proximity search*, czyli taki wariant wyszukiwania, w którym interesują nas dokumenty, które zawierają fragment o długości K, zawierający wszystkie termy z zapytania. Zwróć szczególną uwagę na to, czy potrzebne są jakieś zmiany w procedurze indeksowania oraz na to, jak powinno działać obsługiwanie takich zapytań.

Zadanie 10. (1p) W opisywanej na wykładzie wersji indeksu pozycyjnego konieczne jest jakaś dodatkowa struktura, umożliwiająca połączenie pozycji termu z dokumentem, na której ta pozycja się znajduje. Pokaż kod w dowolnym języku, który efektywnie znajduje nr dokumentu, przy założeniu, że mamy posortowaną listę pozycji pierwszego termu dla wszystkich dokumentów. Jak można to przyśpieszyć kosztem zużycia dodatkowej pamięci?

Zadanie 11. (1p) Na wykładzie była podana prosta metoda kompresji list postingowych (VB). W tym zadaniu powinieneś zaproponować inną metodę, która zapisuje liczby w niepodzielnej przez 8 liczbie bitów (i umożliwia potencjalnie większą kompresję). Idea metody jest następująca: długość liczby kodujemy unarnie, po czym kodujemy binarnie samą liczbę. Rozwiń tę metodę do pełnej definicji kodowania liczb. Opisz procedurę rozkodowywania (takie kody mają swoją nazwę, ale specjalnie nie jest ona tu podana).

Zadanie 12. (1p) W poprzednim zadaniu opis metody zawierał frazę: *długość liczby kodujemy unarnie*. Pokaż, jak można byłoby to poprawić (uzyskując potencjalnie większą kompresję), i pokaż przykład liczby, która rzeczywiście uzyskuje krótszy kod.