Instrukcja do zajęć laboratoryjnych

Język ANSI C (w systemie LINUX)

wersja: 1.0

Nr ćwiczenia:	12, 13	
Temat:	**Implementacja demonstracyjnego systemu do wyszukiwania informacji tekstowych w oparciu o tzw. reprezentację wektorową dokumentów (ang. Term-by-Document Matrix)**	
Cel ćwiczenia:	Celem ćwiczenia jest napisanie programu, który implementuje jedną z podstawowych metod tzw. eksploracja tekstu (ang. Test Mining).	
Wymagane przygotowanie teoretyczne:	Samodzielne zapoznanie się z problematyką pewnego wybranego fragmentu bardzo obszernej dziedziny nauki o nazwie *eksploracja danych (ang. data exploration, data mining)*. Należy korzystać z podanego spisu literatury oraz źródeł internetowych.	
Sposób zaliczenia:	Sprawozdanie w formie pisemnej.	[**X**]
	Pozytywna ocena ćwiczenia przez prowadzącego pod koniec zajęć.	[]

1. Uwagi wstępne

Zamieszczony w kolejnym punkcie opis zadania jest podany bardzo ogólnie i bez szczegółowego rozwinięcia. Student powinien samodzielnie zapoznać się z opisywanym zagadnieniem korzystając z podanego na końcu instrukcji spisu literatury oraz zasięgając informacji u prowadzącego.

2. Skrótowy opis problemu

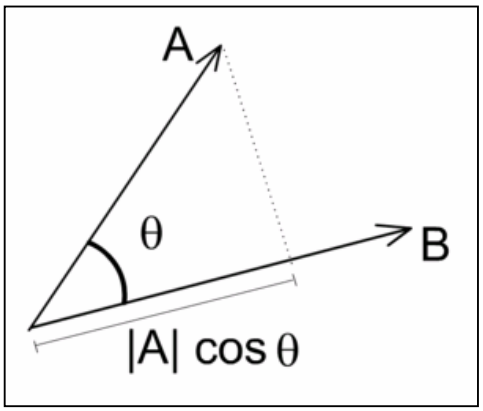
Celem ćwiczenia jest implementacja jednego z typowych algorytmów stosowanych w eksploracji danych tekstowych. W programie należy zaimplementować tzw. **reprezentację macierzową dokumentów (ang. Term-by-Document Matrix; TDM)** oraz zapewnić możliwość „odpytywania" utworzonej (w postaci tejże macierzy) „bazy danych".

Chodzi tutaj o możliwość wyszukiwania dokumentów (tekstowych) w oparciu o podane przez użytkownika zapytanie (na tej zasadzie działają praktycznie wszystkie wyszukiwarki internetowe). Użytkownik konstruuje zapytanie składające się z pewnej liczby słów kluczowych a wyszukiwarka zwraca dokumenty, które „najbardziej pasują" do tego zapytania. Obliczany jest również pewien liczbowy wskaźnik podobieństwa, który umożliwia ustawienie (wyświetlenie) dokumentów od tych najbardziej podobnych do najmniej podobnych czyli wg. tzw. rankingu. Wskaźnik ten to tzw. **miara kosinusowa**, która odzwierciedla podobieństwo pomiędzy dokumentami a zapytaniem

wprowadzonym przez użytkownika. Reprezentuje ona kąt pomiędzy dwoma **wektorami** reprezentującymi dokumenty. Dla przypadku 2-wymiarowego pokazano to na poniższym rysunku (uogólnienie na przypadek wielowymiarowy jest prawie natychmiastowe). Zachodzi oczywiście znana zależność:

$$\mathbf{A} \cdot \mathbf{B} = |\mathbf{A}| \cdot |\mathbf{B}| \cdot \cos \theta . \tag{1}$$

Dwa wektory są tym bardziej podobne do siebie, im bardziej kosinus kąta pomiędzy nimi zbliża się do 1. Analogicznie wektory stają się do siebie coraz bardziej niepodobne, gdy kosinus kąta zbliża się do 0.



Program powinien umożliwiać tworzenie macierzy TDM[1] na trzy sposoby:

- reprezentacja **boolowska** (wagi słów w wektorze dokumentu powinny przyjmować tylko dwie wartości: 0 lub 1),
- reprezentacja **ilościowa** (ang. Term Frequency TF; wagi słów w wektorze dokumentu powinny być równe liczbie wystąpień słowa `i` w dokumencie `j`),
- reprezentacja **ilościowa wg. schematu TFIDF[2]** (ang. Term Frequency Inverse Document Frequency). Człon TF oznacza częstość słowa `i` w dokumencie `j`, człon IDF oznacza tzw. odwrotna częstość słów i jest wyliczany jako $\log_2(N/n_i)$, gdzie N – łączna liczba dokumentów, n_i – liczba dokumentów zawierających słowo `i`. Waga słowa `i` w dokumencie jest wówczas wyliczana jako iloczyn czynnika TF oraz IDF.

Macierz TDM może zostać dodatkowo znormalizowana tak, aby wszystkie wektory (kolumny w macierzy) miały długość jednostkową. Normalizacja upraszcza wyznaczanie miary kosinusowej, gdyż nie trzeba wówczas obliczać długości (normy) wektorów (patrz wzór 1). Każdą element w macierzy TDM należy pomnożyć przez czynnik

$$\left(\sum_i (l_{ij})^2 \right)^{-1/2} \tag{2}$$

który skaluje poszczególne wektory. l_{ij} oznacza element w i-tym wierszu i j-tej kolumnie.

[1] Macierz ma rozmiar `i` x `j`, gdzie `i` oznacza całkowitą ilość słów a `j` całkowitą ilość dokumentów.

[2] Schemat TFIDF (zamiast TF) stosujemy w celu zmniejszenia efektu „przeszacowywania" wag słów, które znajdują się w dużych dokumentach. Po prostu w dużych dokumentach istnieje większe prawdopodobieństwo wystąpienia danego słowa. I wówczas dokumenty te stają się bardziej podobne do wydanego zapytania, niżby to wynikało z ich rzeczywistego podobieństwa

Uwaga: macierz TDM jest zwykle macierzą rzadką (zdecydowana większość elementów będzie miała wartości zerowe). Dlatego też używanie macierzy TDM w „czystej postaci" nie jest w praktyce stosowane, ze względu na jej wielkość i małą w związku z tym efektywność komputerowego przetwarzania. Niemniej jednak do celów demonstracyjnych można ją z powodzeniem wykorzystać.

3. Przykład[3]

Załóżmy, że mamy 4 dokumenty tekstowe, każdy z następującą zawartością:

Dokument 1	**„bazy relacyjne, bazy tekstowe, bazy inne"**
Dokument 2	**„bazy danych: przykłady, zastosowania"**
Dokument 3	**„bazy danych - zalety; bazy danych – wady"**
Dokument 4	**„składowanie danych"**

Macierz TDM z reprezentacją boolowską będzie wyglądała następująco (w nawiasach podano **wartości znormalizowane**):

	Dokument 1	Dokument 2	Dokument 3	Dokument 4
bazy	**1 (0.5000)**	**1 (0.5000)**	**1 (0.5000)**	**0**
danych	**0**	**1 (0.5000)**	**1 (0.5000)**	**1 (0.7071)**
inne	**1 (0.5000)**	**0**	**0**	**0**
przykłady	**0**	**1 (0.5000)**	**0**	**0**
relacyjne	**1 (0.5000)**	**0**	**0**	**0**
składowanie	**0**	**0**	**0**	**1 (0.7071)**
tekstowe	**1 (0.5000)**	**0**	**0**	**0**
wady	**0**	**0**	**1 (0.5000)**	**0**
zalety	**0**	**0**	**1 (0.5000)**	**0**
zastosowania	**0**	**1 (0.5000)**	**0**	**0**

Zwróćmy uwagę, że wszystkie znaki interpunkcyjne zostały pominięte jako nie wnoszące żadnej informacji merytorycznej. Pewne słowa, np. „inne", również takiej informacji nie wnoszą[4]. Powinny one zostać jakoś usunięte z finalnej macierzy TDM (w przykładzie nie zostały one usunięte). Ponadto, aby system miał bardziej praktyczne zastosowanie powinien w macierzy TDM umieszczać słowa w ich podstawowej formie, czyli np. zamiast słowa „składowanie" powinno być „składować", zamiast „bazy" powinno być „baza" itd. Automatyczne zamiana form odmienionych na ich formy podstawowe jest jednak dość trudnym zadaniem – nawet dla języka angielskiego, a tym bardziej dla języka polskiego.

Ta sama macierz z reprezentacją TF będzie miała postać (w nawiasach podano **wartości znormalizowane**):

	Dokument 1	Dokument 2	Dokument 3	Dokument 4
bazy	**3 (0.8660)**	**1 (0.5000)**	**2 (0.6325)**	**0**
danych	**0**	**1 (0.5000)**	**2 (0.6325)**	**1 (0.7071)**
inne	**1 (0.2887)**	**0**	**0**	**0**
przykłady	**0**	**1 (0.5000)**	**0**	**0**
relacyjne	**1 (0.2887)**	**0**	**0**	**0**
składowanie	**0**	**0**	**0**	**1 (0.7071)**

[3] Na stronie http://scgroup.hpclab.ceid.upatras.gr/scgroup/Projects/TMG/ można znaleźć program dla środowiska MATLAB (tzw. toolbox), w którym można sprawdzić/przetestować wykonany program. Umożliwia on m.in. tworzenie macierzy TDM w trzech wymienionych w instrukcji wariantach (oraz jeszcze dodatkowo w kilkunastu innych), jak również kierowanie zapytań do TDM z wykorzystaniem miary kosinusowej.

[4] Są to tzw. słowa wyłączone (ang. stop words).

tekstowe	**1 (0.2887)**	**0**	**0**	**0**
wady	**0**	**0**	**1 (0.3162)**	**0**
zalety	**0**	**0**	**1 (0.3162)**	**0**
zastosowania	**0**	**1 (0.5000)**	**0**	**0**

I wreszcie ta sama macierz z reprezentacją TFIDF będzie miała postać (w nawiasach podano **wartości znormalizowane**):

	Dokument 1	Dokument 2	Dokument 3	Dokument 4
bazy	**1.2451 (0.3382)**	**0.4150 (0.1437)**	**0.8301 (0.2711)**	**0**
danych	**0**	**0.4150 (0.1437)**	**0.8301 (0.2711)**	**0.4150 (0.2032)**
inne	**2.0000 (0.5433)**	**0**	**0**	**0**
przykłady	**0**	**2.0000 (0.6924)**	**0**	**0**
relacyjne	**2.0000 (0.5433)**	**0**	**0**	**0**
składowanie	**0**	**0**	**0**	**2.0000 (0.2032)**
tekstowe	**2.0000 (0.5433)**	**0**	**0**	**0**
wady	**0**	**0**	**2.0000 (0.6531)**	**0**
zalety	**0**	**0**	**2.0000 (0.6531)**	**0**
zastosowania	**0**	**2.0000 (0.6924)**	**0**	**0**

Wydając zapytanie do naszej macierzy TDM np. podając frazę: `bazy danych` otrzymujemy odpowiednio dla reprezentacji boolowskiej, TF oraz TFTDF następujące wyniki (miary kosinusowe):

reprezentacja boolowska
Dokument 3 – wsp. podobieństwa: 0.70711
Dokument 2 - wsp. podobieństwa: 0.70711
Dokument 4 - wsp. podobieństwa: 0.5
Dokument 1 - wsp. podobieństwa: 0.35355

reprezentacja TF
Dokument 3 - wsp. podobieństwa: 0.89443
Dokument 2 - wsp. podobieństwa: 0.70711
Dokument 1 - wsp. podobieństwa: 0.61237
Dokument 4 - wsp. podobieństwa: 0.5

reprezentacja TFIDF
Dokument 3 - wsp. podobieństwa: 0.38333
Dokument 1 - wsp. podobieństwa: 0.23918
Dokument 2 - wsp. podobieństwa: 0.20319
Dokument 4 - wsp. podobieństwa: 0.14368

4. Opis zadania do wykonania

W ramach ćwiczenia należy napisać program, który będzie umożliwiał:

A. Tworzenie macierzy TDM na podstawie wskazanych plików tekstowych. Z pliku powinny zostać usunięte wszystkie znaki przystankowe. W wersji uproszczonej znaki przystankowe mogą zostać ręcznie usunięte przez użytkownika, jeszcze zanim pliki zostaną wprowadzone do programu. Aby zbytnio nie komplikować obróbki plików wejściowych, pomijamy problem usuwania słów wyłączonych oraz problem odmiany słów. Gdy więc przykładowo w pliku będą słowa „komputer", „komputerami" oraz „komputery", to zostaną one potraktowane jako 3 całkowicie różne słowa. Podobnie, gdy będą słowa „ale", „lub" oraz „który", to mimo że nie niosą one żadnej treści merytorycznej, zostaną potraktowane przez system jako pełnoprawne słowa.

B. Powinna istnieć możliwość tworzenia macierzy TDM w trzech wersjach: boolowska, TF oraz TFIDF. Każda z nich powinna mieć 2 warianty: z normalizacją wartości oraz bez normalizacji.

C. Po utworzeniu macierzy TDM musi istnieć możliwość zadawania zapytań. System powinien wyświetlać wyniki (ranking dokumentów pod względem zgodności z podanym zapytaniem) w postaci podobnej do tej pokazanej w punkcie 3. Zwróć uwagę na to, że wektor zapytania musi uwzględniać to, czy macierz TDM jest znormalizowana, czy też nie.

5. Sprawozdanie

Sprawozdanie powinno zawierać następujące elementy:

- szczegółowy opis budowy macierzy TDM (dla reprezentacji boolowskiej, TF oraz TFIDF),
- ręczne obliczenia przygotowane dla prostych przykładowych dokumentów,
- 2-3 przykłady demonstrujące różnice pomiędzy różnymi wersjami macierz TDM (boolowska, TF oraz TFIDF). Chodzi tutaj o „konstruktywne" wychwycenie pewnych wad/zalet wszystkich wersji macierzy TDM.
- ew. inne uwagi i spostrzeżenia.

6. Literatura

1. Michale W. Berty, Murray Browne, *Understanding Search Engines. Mathematical Modeling and Text Retrieval*, SIAM, 1999 (książka w Polsce dość trudno dostępna. Można zamówić np. w księgarni internetowej amazon.com – koszt około 30-35 $. Można też pożyczyć ją od prowadzących).
2. Daniel T. Larose, *Odkrywanie wiedzy z danych ,* Wydawnictwo Naukowe PWN, 2006.
3. http://wazniak.mimuw.edu.pl/index.php (wykład „Eksploracja danych", rozdziały „Eksploracja tekstu I" oraz „Eksploracja tekstu II").
4. http://scgroup.hpclab.ceid.upatras.gr/scgroup/Projects/TMG/ - tzw. toolbox dla środowiska MATLAB implementujący tworzenie TDM.