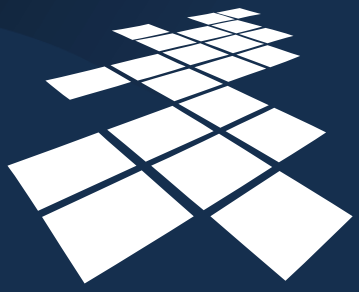
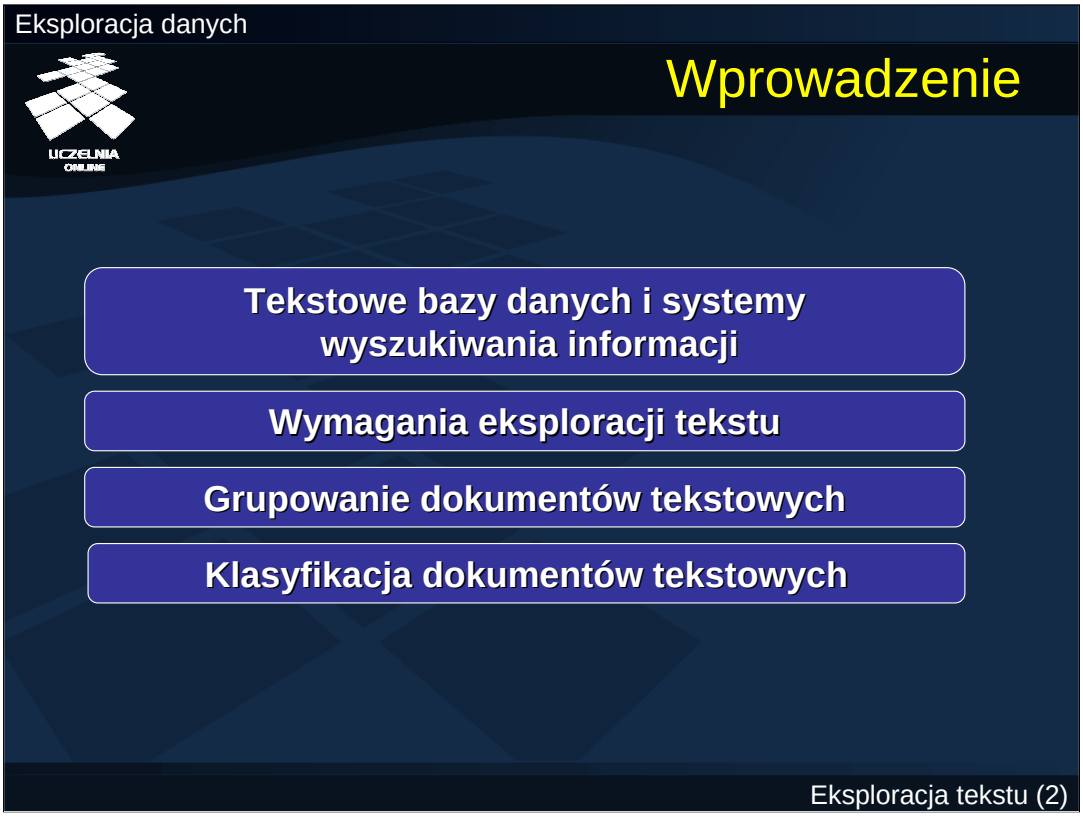
Eksploracja tekstu

Wprowadzenie
Wyszukiwanie dokumentów
Reprezentacje tekstu

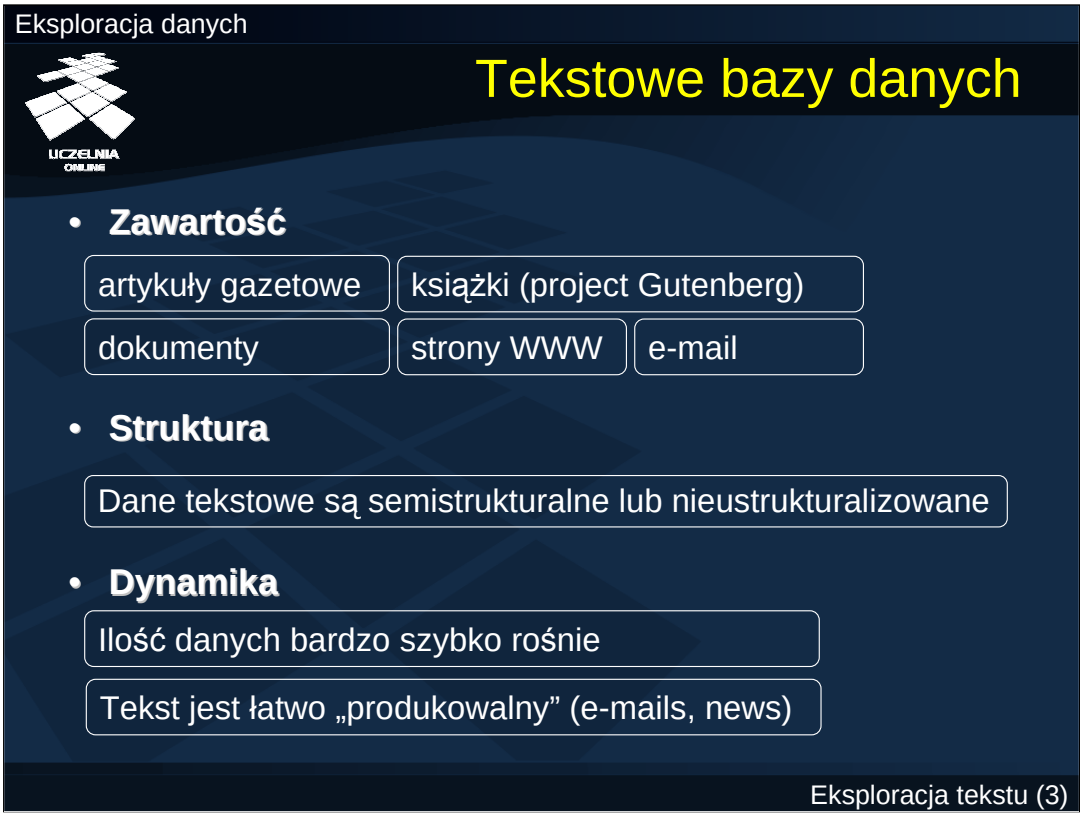


UCZELNIA
ONLINE

Tematem wykładu są zagadnienia związane z eksploracją tekstu. Rozpoczniemy od krótkiego wprowadzenia do problematyki przetwarzania tekstu. Krótko scharakteryzujemy systemy tekstowych baz danych oraz systemy wyszukiwania informacji, oraz porównany systemy wyszukiwania informacji i systemy baz danych. Następnie, przejdziemy do omówienia miar oceny wyszukiwania informacji oraz do zagadnienia reprezentacji tekstu. Przedstawimy i omówimy dwa podejścia do wyszukiwania dokumentów tekstowych: wyszukiwanie w oparciu o słowa kluczowe oraz wyszukiwanie w oparciu o reprezentację wektorową dokumentów tekstowych. Temu drugiemu zagadnieniu poświęcimy nieco więcej uwagi ze względu na jego znaczenie w odniesieniu wyszukiwania stron WWW w Internecie.

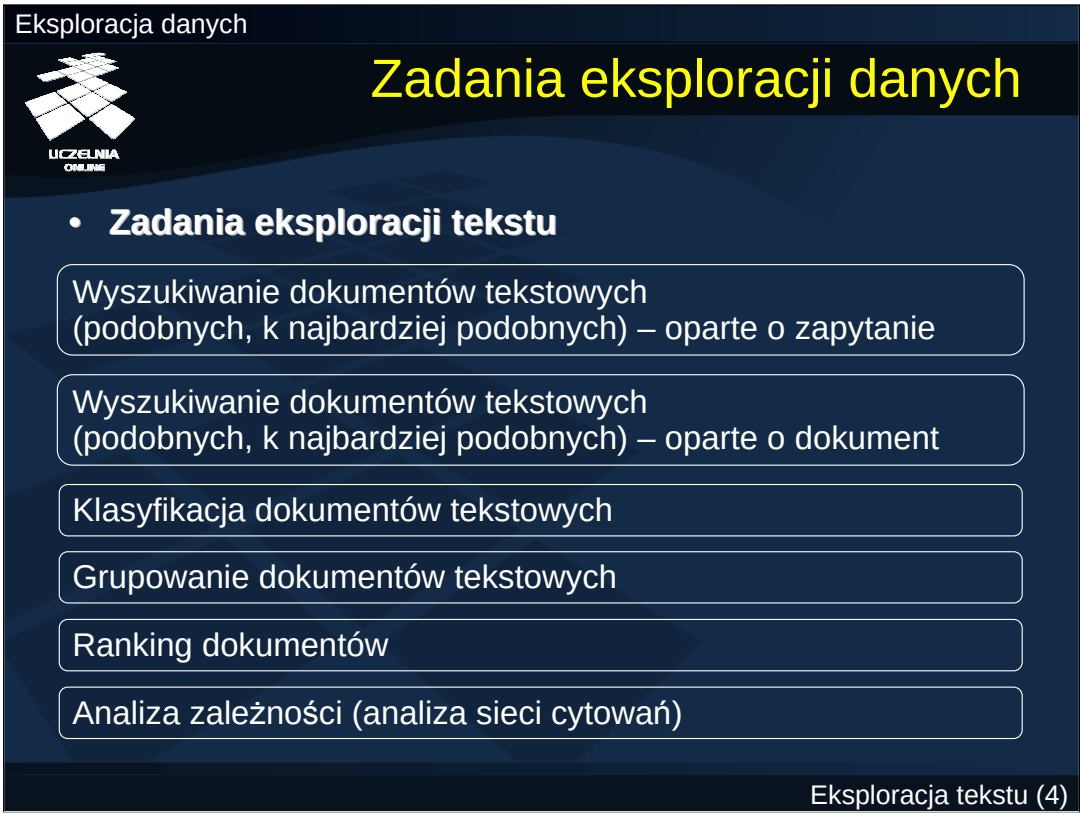


Dokumenty tekstowe są jedną z najpopularniejszych i najczęściej spotykanych form zapisu informacji. Systemy baz danych służące przechowywaniu dokumentów tekstowych nazywamy tekstowymi bazami danych. Systemy te są wyposażone w funkcjonalność umożliwiającą efektywne przechowywanie dokumentów tekstowych jak i ich efektywne wyszukiwanie (pod warunkiem, że potrafimy precyzyjnie wyspecyfikować poszukiwaną informację tekstową). Innym ważnym repozytorium służącym do przechowywania dokumentów tekstowych są tzw. systemy wyszukiwania informacji (ang. Information Retrieval Systems – w skrócie systemy IR). Dokumenty tekstowe, podobnie jak każdy inny typ danych, mogą być, i często są, źródłem ważnej i użytecznej wiedzy. Zatem, w odniesieniu do dokumentów tekstowych możemy zastosować metody eksploracji danych. Szczególnie przydatne i popularne, z praktycznego punktu widzenia, są dwie metody eksploracji tekstu: klasyfikacja i grupowanie. Niemniej, w odniesieniu do dokumentów tekstowych stosuje się również wiele innych technik, o których powiemy później.

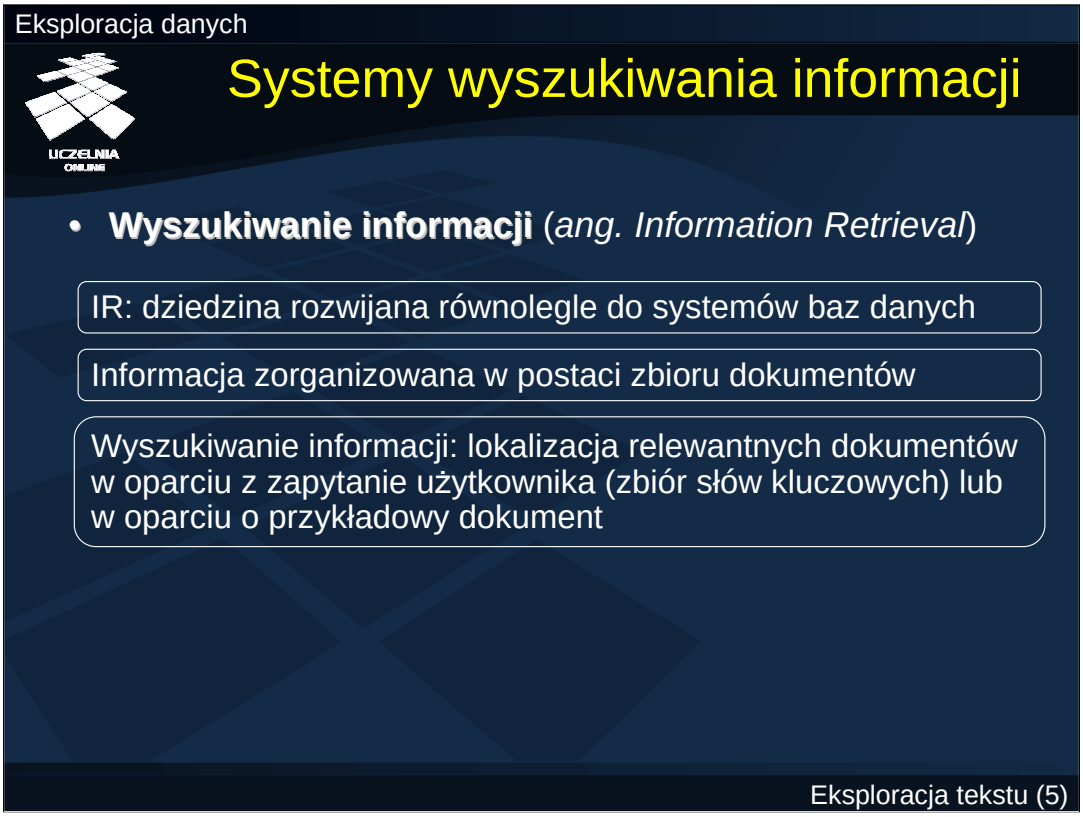


Tekstowe bazy danych, podobnie jak systemy IR, przechowują różnorodne typy dokumentów tekstowych, takie jak artykuły gazetowe, szeroko rozumiane dokumenty tekstowe, książki (e-book), wiadomości e-mail, strony WWW. Czym różnią się dokumenty tekstowe od innych typów danych? Zasadnicza różnica polega na tym, że dokumenty tekstowe nie posiadają, najczęściej, żadnej wewnętrznej struktury (mówimy, że dane są nieustrukturalizowane) lub struktura ta jest szczątkowa (mówimy, że dane są semistrukturalne). Przykładowo, struktura wiadomości e-mail zawiera informacje o nadawcy, odbiory, rozmiarze, temacie wiadomości, pozostała część wiadomości jest łańcuchem znaków pozbawionym wewnętrznej struktury.

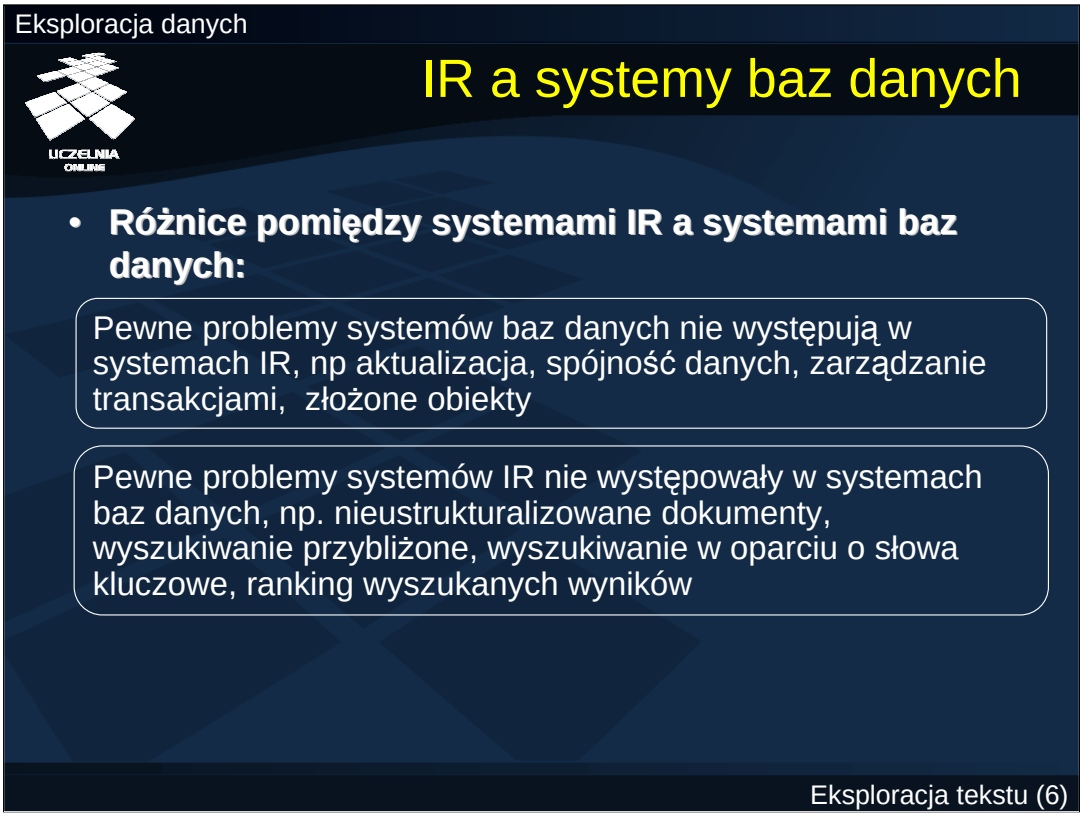
Czym charakteryzują się tekstowe repozytoria danych? Przede wszystkim charakteryzują się olbrzymią dynamiką przyrostu danych. Wynika to z faktu, że dane są bardzo łatwo „produkowalne”. Źródłem nowych danych są: poczta elektroniczna, media elektroniczne, serwisy internetowe, itp.



Jak wspomnieliśmy już wcześniej, dokumenty tekstowe, podobnie jak każdy inny typ danych, mogą być, i często są, źródłem ważnej i użytecznej wiedzy. Mogą więc podlegać eksploracji danych w celu ekstrakcji użytecznej wiedzy. Najpopularniejsze i najczęściej wykonywane zadania eksploracji tekstu to: eksploracja zawartości dokumentów tekstowych (tj. wyszukiwanie podobnych dokumentów tekstowych w oparciu o zapytania lub wyszukiwanie podobnych dokumentów tekstowych w oparciu o przykładowe dokumenty), klasyfikacja dokumentów tekstowych, grupowanie dokumentów tekstowych, kategoryzacja dokumentów tekstowych, ranking ważności dokumentów, oraz analiza zależności pomiędzy dokumentami (np. analiza sieci cytowań).



Zanim przejdziemy do omówienia technik eksploracji tekstu, szczególnie, technik wyszukiwania dokumentów tekstowych, krótko przedstawimy systemy wyszukiwania informacji. Systemy wyszukiwania informacji są rozwijane o wielu lat równolegle do systemów baz danych. Systemy te były przeznaczone, początkowo, do budowy systemów bibliotecznych oraz archiwów dokumentów tekstowych (np.. w systemach administracyjnych). Aktualnie systemy IR są wykorzystywane do budowy takich systemów jak: systemy biblioteczne, systemy zarządzania dokumentami, systemy zarządzania zawartościąCechą charakterystyczną tych systemów jest specyficzna organizacja danych – w systemach IR dane (informacja) są zorganizowane w postaci zbioru dokumentów tekstowych. Wyszukiwanie informacji e systemach IR polega lokalizacja relewantnych (istotnych i ważnych) dokumentów w oparciu z zapytanie użytkownika. Zapytanie może być zdefiniowane dwojako: w postaci zapytania składającego się ze słów kluczowych, opisujących poszukiwane dokumenty, lub w postaci przykładowego dokumentu, który charakteryzuje poszukiwane dokumenty.



Systemy wyszukiwania informacji przypominają, biorąc pod uwagę oferowaną funkcjonalność, systemy baz danych. Zapewniają możliwość przechowywania i wyszukiwania informacji - tym wypadku dokumentów tekstowych. Jednakże, występują istotne różnice pomiędzy technologią systemów wyszukiwania informacji a systemami baz danych. Pewne problemy, charakterystyczne dla systemów baz danych, nie występują w systemach wyszukiwania informacji. W systemach IR nie występuje problem spójności danych rozumiany jako konieczność zapewnienia ograniczeń integralnościowych. W systemach IR nie występują transakcje jako podstawowy mechanizm zapewniający poprawność współbieżnego dostępu do danych, spójność danych, poprawność odtwarzania danych po awarii czy, wreszcie, mechanizm gwarantujący trwałość danych. Zbiór typów przechowywanych danych jest również ograniczony – nie występują takie złożone typy danych jak: sekwencje, przebiegi czasowe, dźwięki, dane multimedialne, itp. Systemy IR nie dysponują, najczęściej, narzędziami do modelowania pojęciowego rzeczywistości, takich jak

schematy EER czy UML.

Z drugiej strony, pewne problemy systemów IR nie występowały, dotychczas, w systemach baz danych, np. nieustrukturalizowane dokumenty, wyszukiwanie przybliżone, wyszukiwanie w oparciu o słowa kluczowe, ranking wyszukanych wyników. W ostatnim czasie obserwujemy konwergencje obu technologii. Systemy baz danych umożliwiają, aktualnie, przechowywanie dokumentów tekstowych i dokumentów XML, ich indeksowanie, wyszukiwanie przybliżone oraz wyszukiwanie w oparciu o słowa kluczowe (tzw. keyword-based search).

Miary oceny wyszukiwania (1)

Precyzja (ang. *precision*): procent wyszukanych dokumentów, które są relewantne z punktu widzenia zapytania (t.j., “poprawne” odpowiedzi)

$$precision = \frac{|\{relevant\} \cap \{retrived\}|}{|\{retrived\}|}$$

Zwrot (ang. *recall*): procent relewantnych dokumentów, które zostały wyszukane

$$recall = \frac{|\{relevant\} \cap \{retrived\}|}{|\{relevant\}|}$$

W ramach systemów wyszukiwania informacji opracowano szereg miar, pozwalających na ocenę wyniku wyszukiwania dokumentów. Dwie najpopularniejsze i najszerzej stosowane miary oceny wyszukiwania to precyzja i zwrot. Precyzję definiujemy jako procent wyszukanych dokumentów, które są relewantne z punktu widzenia zapytania (t.j., są to “poprawne” dokumenty). 100% precyzja oznacza, że zbiór wyszukanych dokumentów zawiera wyłącznie „poprawne” (tj. relewantne) dokumenty. Drugą z miar, zwrot, definiujemy jako procent relewantnych dokumentów, które zostały wyszukane. 100% zwrot oznacza, że wyszukaliśmy wszystkie dokumenty relewantne z punktu widzenia zapytania. Oczywiście, określenie miary zwrotu wymaga znajomości całego zbioru ‘poprawnych” odpowiedzi.

Miary oceny wyszukiwania (2)

Fall-out: prawdopodobieństwo znalezienia dokumentu nierelewantnego wśród wyszukanych dokumentów

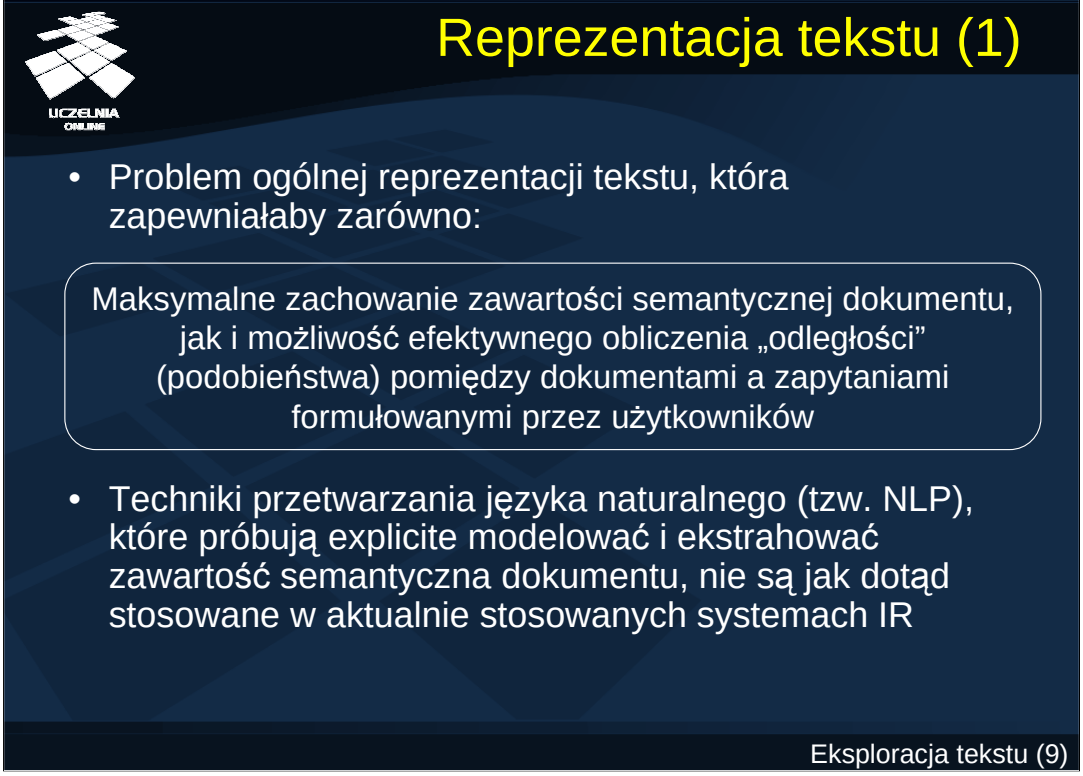
$$fall-out = \frac{\{\{irrelevant\} \cap \{retrieved\}\}}{\{irrelevant\}}$$

F-miara (ang. *F-measure*): średnia ważona precyzji i zwrotu

$$F\alpha = \frac{(1+\alpha) \cdot precision \cdot recall}{(\alpha \cdot precision \cdot recall)}$$

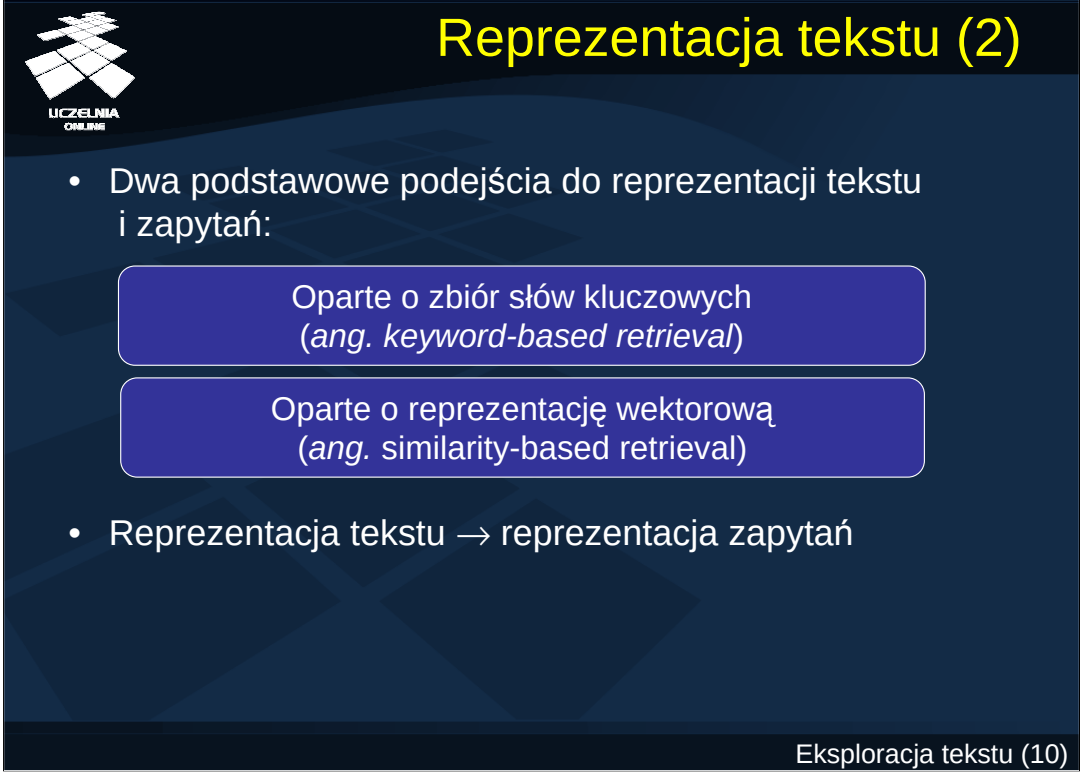
gdzie α przyjmuje najczęściej wartości: 1, 0.5 oraz 2

W ramach systemów wyszukiwania informacji opracowano szereg dodatkowych miar oceny wyszukiwania, które pozwalają, dodatkowo, charakteryzować jakość procesu wyszukiwania. Na slajdzie przedstawiono definicje dwóch innych popularnych miar oceny wyszukiwania: tzw. fall-out i F-miarę. Fall-out definiujemy jako procent nierelewantnych (tj. „błędnych" dokumentów), które zostały wyszukane. 100% fall-out oznacza, że zbiór wyszukanych dokumentów zawiera wyłącznie „błędne" (tj. nierelewantne) dokumenty. Wreszcie, F-miarę definiujemy jako średnia ważoną precyzji i zwrotu. Dla F-miary współczynnik alfa przyjmuje, najczęściej, 3 wartości: 1, 0.5 i 2.

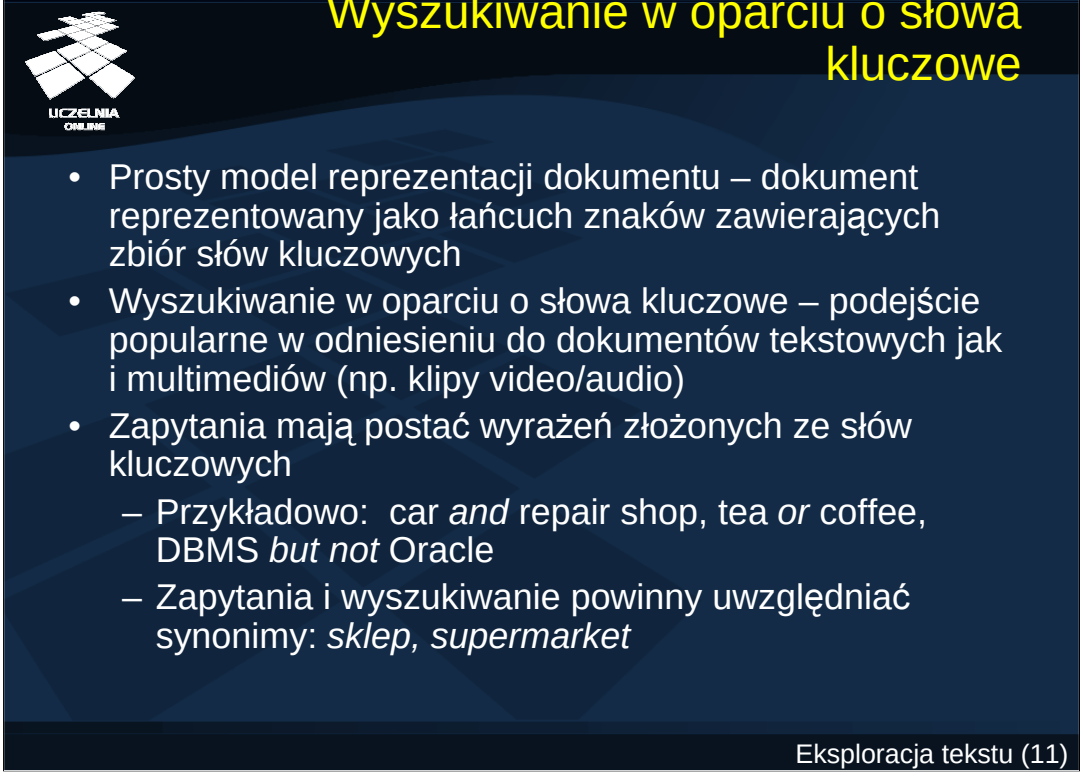


W przypadku problemu wyszukiwania dokumentów tekstowych fundamentalne pytanie dotyczy ogólnej reprezentacji dokumentu tekstowego, która zapewniałaby zarówno maksymalne zachowanie zawartości semantycznej dokumentu, jak i możliwość efektywnego obliczenia „odległości” (podobieństwa) pomiędzy dokumentami a zapytaniami formułowanymi przez użytkowników. Użytkownik korzystający z systemu IR chciałby wyszukać w systemie dokumenty, które są dla niego relewantne z punktu widzenia ich zawartości semantycznej. Oczywiście, najlepszym rozwiązaniem byłoby pełnotekstowe przechowywanie i wyszukiwanie dokumentów w oparciu o semantykę zapytania. Jednakże, w ogólnej postaci problem automatycznego wyszukiwania dokumentów jest niezwykle trudny.

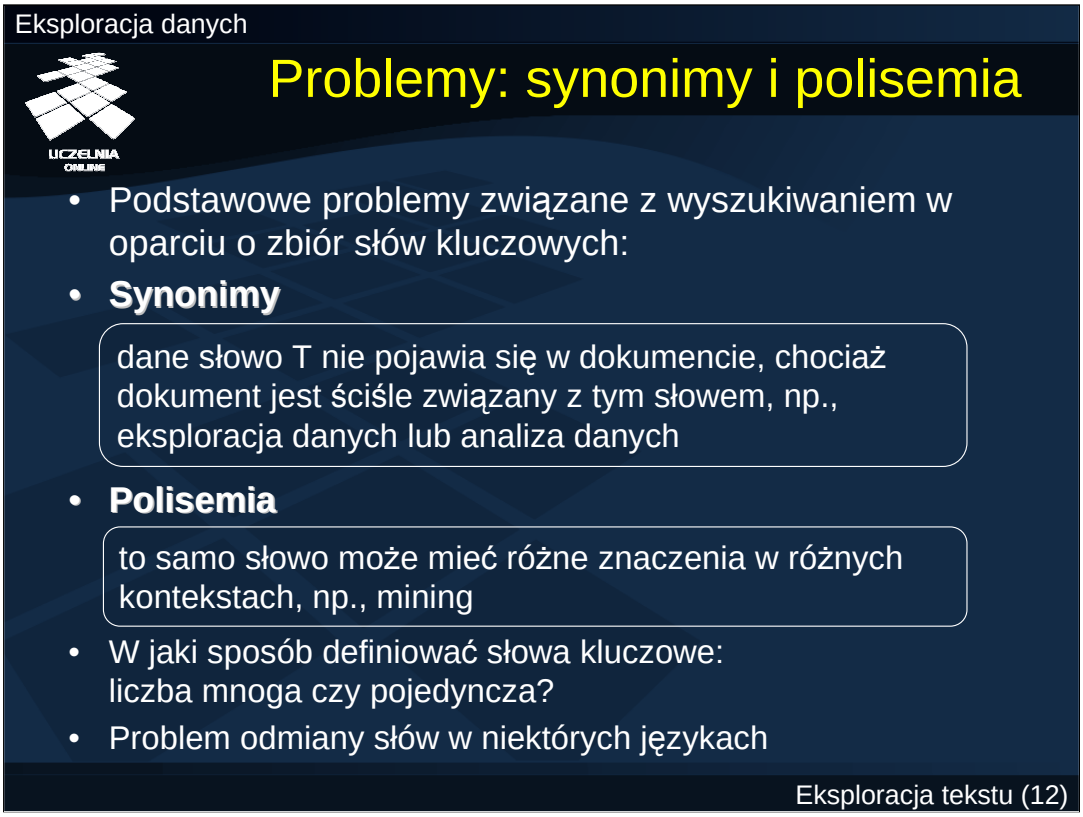
Podejmowane dotychczas próby budowy takich systemów, w których techniki przetwarzania języka naturalnego (tzw. NLP) (które próbują explicite modelować i ekstrahować zawartość semantyczna dokumentu) nie są jak dotąd stosowane w aktualnie dostępnych komercyjnych systemach IR i systemach tekstowych baz danych.



W chwili obecnej, większość systemów wyszukiwania informacji jak również systemów tekstowych baz danych opiera się na prostych technikach dopasowania i zliczania występowania słów kluczowych opisujących przechowywane dokumenty. W praktyce, mamy do czynienia z dwoma najpopularniejszymi podejściami do reprezentacji tekstu i zapytań: podejście oparte o zbiór słów kluczowych (ang. keyword-based retrieval) oraz podejście oparte o reprezentację wektorową dokumentów (ang. similarity-based retrieval). Przyjęcie określonej reprezentacji dokumentu tekstowego determinuje postać reprezentacji zapytania użytkownika.



Obecnie, krótko scharakteryzujemy pierwsze z wymienionych podejść. W podejściu tym zakłada się bardzo prosty model reprezentacji dokumentu tekstowego. Każdy dokument reprezentowany jest jako łańcuch znaków zawierający zbiór słów kluczowych opisujących dokument i przypisywanych dokumentowi przez „użytkownika" – najczęściej, osobę tworzącą dokument. Przykładowo, opis dokumentu może mieć następującą postać: „bazy danych, eksploracja, klasyfikacja, drzewo decyzyjne". Wyszukiwanie dokumentów jest również realizowane poprzez podanie słów kluczowych. Zapytania mają postać wyrażeń złożonych ze słów kluczowych. Przykładowo, zapytania mogą mieć postać następujących wyrażeń: „samochód *i* warsztat samochodowy", „herbata lub kawa", lub „DBMS *ale nie* Oracle". Przedstawione podejście znajduje również zastosowanie nie tylko w odniesieniu do modelowania i wyszukiwania dokumentów tekstowych, ale również w odniesieniu do innych typów danych takich jak klipy video/audio.



Główną zaletą przedstawionego podejścia jest jego prostota. Niestety, podejście posiada szereg istotnych wad związanych z precyzyjnym definiowaniem zapytań i realizacją procesu wyszukiwania. Pierwszym problemem jest synonimia. Pod pojęciem synonimii rozumiemy, że ta sama rzecz (dokument) może być opisany na wiele sposobów. Na przykład, synonimem słowa sklep może być słowo supermarket. W konsekwencji, okazuje się, że dane słowo T nie pojawia się w dokumencie, chociaż dokument jest ściśle związany z tym słowem, np., eksploracja danych lub analiza danych. System wyszukiwania powinien zatem wyszukać nie tylko te dokumenty, które explicite zawierają słowo lub zwrot „eksploracja danych”, ale również wszystkie dokumenty zawierające zwrot „analiza danych”. Innym problemem, odwrotnym do problemu synonimii, jest problem polisemii. Pod pojęciem polisemii rozumiemy, że to samo słowo może mieć różne znaczenia w różnych kontekstach, np., mining. Najczęściej, pod pojęciem „mining” rozumiemy wydobywanie węgla. W kontekście informatyki, hasło „mining” odnosi się do pojęcia „eksploracji danych”. Nie są to jedyne problemy związane z przedstawionym podejściem. Pojawia się problem w jaki sposób definiować słowa kluczowe - liczba mnoga czy liczba pojedyncza? Czy jeżeli zapytanie zawiera hasło „baza danych”, system wyszukiwania powinien wyszukać dokumenty zawierające hasło „bazy danych”? W przypadku niektórych języków narodowych pojawia się dodatkowa trudność związana z odmianą słów.

Problemy: zapytania i dokumenty

- Zapytanie jest reprezentowane jako wyrażenie logiczne zdefiniowane na podzbiorze słów kluczowych:

  data AND mining AND NOT (coal)

- Wyrażenie można uzupełnić o wagi, aby uwzględniać względne znaczenie jednych słów kluczowych w stosunku do innych
- Zasadnicze wady podejścia:
  - brak naturalnej semantyki miary odległości pomiędzy zapytaniem a zbiorem dokumentów
  - niemożność zdefiniowania rankingu odpowiedzi (dokumentów) w stosunku do zapytania
  - problemy z definicją zapytań w postaci wyrażeń boolowskich

Wróćmy do definicji zapytania. Zapytanie, w analizowanym podejściu, jest reprezentowane jako wyrażenie logiczne zdefiniowane na podzbiorze słów kluczowych: „data AND mining AND NOT (coal)”. Łatwo zauważyć, że wyrażenie można uzupełnić o wagi, tak aby uwzględniać względne znaczenie jednych słów kluczowych w stosunku do innych, i dzięki temu poprawić precyzję wyszukiwania. Niestety, nie zmienia to zasadniczo podstawowego problemu związanego z podejściem opartym o słowa kluczowe, a mianowicie, braku naturalnej semantyki miary odległości pomiędzy zapytaniem a dokumentami. W konsekwencji, uniemożliwia to przeprowadzenie rankingu dokumentów na podstawie ich zawartości w odniesieniu do zapytania. Dodatkowo, co wydaje się raczej dziwne i zaskakujące, użytkownicy mają często duże trudności z konstruowaniem zapytań w postaci wyrażeń boolowskich, które by odzwierciedlały ich intencje. Mimo swoich istotnych wad i ograniczeń, podejście oparte o zbiory słów kluczowych do reprezentacji dokumentów i definiowania zapytań jest bardzo popularne i szeroko stosowane w wielu praktycznych systemach wyszukiwania informacji z uwagi na swoją efektywność i prostotę.

Wyszukiwanie w oparciu o reprezentację wektorową

- Reprezentacja tekstu – macierz częstości występowania słów kluczowych (Frequency matrix):
  - **Term_Frequency_Matrix(d_i, t_i):** liczba wystąpień słowa t_i w dokumencie d_i.
  - Zbiór słów kluczowych może być bardzo duży (50 000 słów)
  - Każdy dokument d_i, $1 \leq i \leq N$, jest reprezentowany w postaci wektora słów
  - współczynnik d_{ij} – waga słowa d_i
- Reprezentacja boolowska wektora – waga przyjmuje dwie wartości 0 lub 1

Alternatywnym podejściem do reprezentacji dokumentu tekstowego jest podejście oparte o reprezentację wektorową dokumentu. Podstawowa idea reprezentacji wektorowej sprowadza się do tego, że dowolny dokument jest reprezentowany w postaci wektora częstości występowania słów kluczowych. Stąd, zbiór N przechowywanych dokumentów tekstowych można przedstawić w postaci macierzy, o nazwie *Term_Frequency_Matrix,* której element TFM[di, ti] reprezentuje liczbę wystąpień słowa kluczowego ti w dokumencie di. Dowolny dokument di, 1< i< N, jest reprezentowany w postaci wektora częstości występowania słów kluczowych. Element TFM[di, ti] nazywamy wagą słowa ti w dokumencie di. W najprostszej reprezentacji boolowskiej, wagi slow w wektorze dokumentu mogą przyjmować tylko dwie wartości: 0 lub 1. Waga słowa ti w dokumencie di równa 1 oznacza, że słowo ti występuje w danym dokumencie di; jeżeli słowo ti nie występuje w dokumencie di, waga słowa ti w dokumencie di jest równa 0. W pełnej reprezentacji, wagi słów odpowiadają częstości ich występowania w dokumentach. Zbiór słów kluczowych wykorzystywanych w podejściu może być bardzo duży (ok. 50 000 słów). Kolejny slajd ilustruje ideę macierzy TFM.

Macierz TFM (1)

	t1	t2	t3	t4	t5	t6
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

t1 – bazy_danych
t2 – SQL
t3 – indeks
t4 – regresja
t5 – wiarygodność
t6 - liniowa

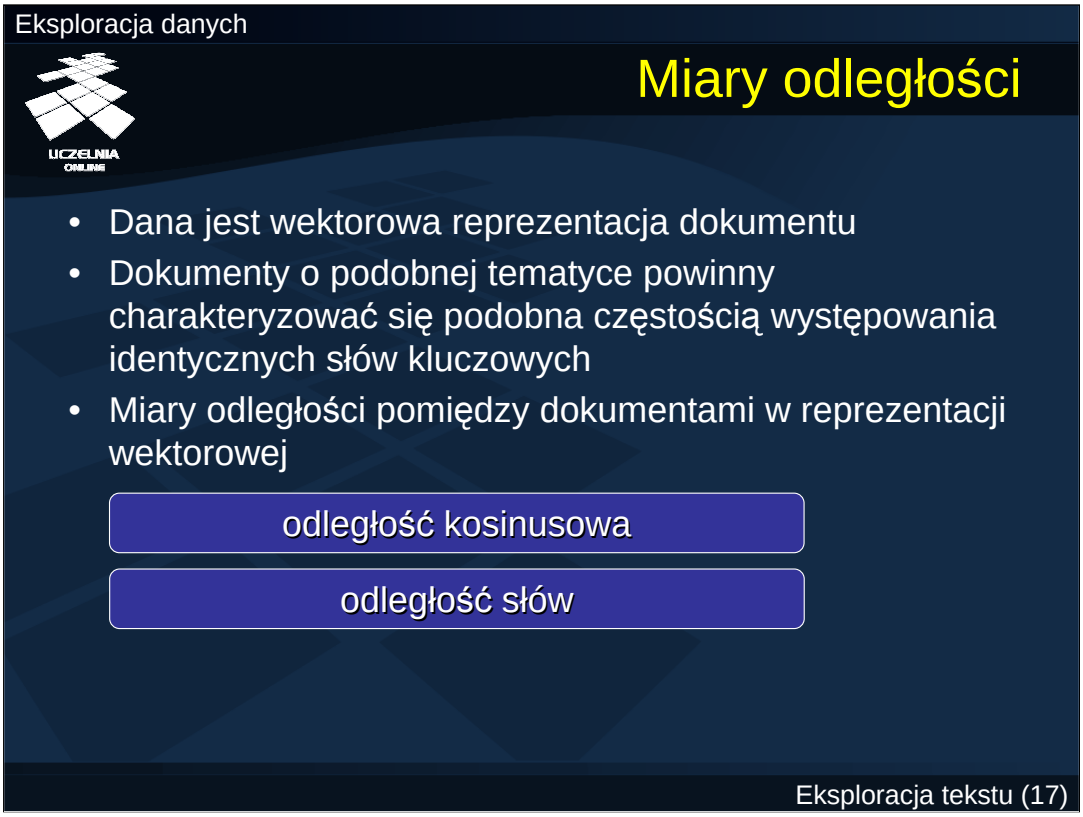
Reprezentacja dokumentu w postaci T-wymiarowego wektora słów powoduje utratę informacji o strukturze zdania jak i kolejności występowania słów w zdaniu

Niniejszy slajd przedstawia macierz TFM. Danych jest 10 dokumentów: d1, d2, ..., d10, oraz 6 słów kluczowych t1, t2, ..., t6. Znaczenie słów kluczowych jest następujące: t1 – bazy_danych, t2 – SQL, t3 – indeks, t4 – regresja, t5 – wiarygodność, oraz t6 – liniowa. Element {di, tj] macierzy reprezentuje częstość występowania słowa tj w dokumencie di. Przykładowo, TFM{d1, t2] = 21 oznacza, że słowo t2 = „SQL” wystąpiło 21 razy w dokumencie d1. Łatwo zauważyć, że reprezentacja dokumentu w postaci T-wymiarowego wektora słów powoduje utratę informacji o strukturze zdania jak i kolejności występowania słów w zdaniu. Klasycznym przykładem ilustrującym utratę semantyki zdania w wyniku transformacji zdania do reprezentacji wektorowej są dwa zdania: „John loves Mary” i „Mary loves John”. Łatwo zauważyć, że oba zdania posiadają identyczną reprezentacje wektorową, lecz kompletnie różne znaczenie semantyczne.

Macierz TFM (2)

- Każdy wektor stanowi D_i stanowi surogat oryginalnego dokumentu d_i
- Macierz TFM jest rzadka – większość macierzy jest wypełniona zerami
- W praktycznych implementacjach systemów IR, ze względu na rzadkość macierzy TFM, oryginalny zbiór dokumentów jest reprezentowany w postaci pliku odwróconego

Czym charakteryzuje się macierz TFM? Jak wspomnieliśmy wcześniej, zbiór słów kluczowych może być bardzo duży, gdyż musi „pokryć”, w przypadku dużych zbiorów dokumentów tekstowych, szeroki zakres tematyczny. W związku z tym, macierz TFM jest „rzadka”, co oznacza, że większość macierzy jest wypełniona zerami. Ze względu na „rzadkość” macierzy TFM, najczęściej, w praktycznych implementacjach systemów wyszukiwania informacji, zbiór dokumentów jest przechowywany w postaci pliku odwróconego, indeksowanego zbiorem słów kluczowych. Każde słowo kluczowe ti wskazuje na rekord w tablicy zawierający N liczb opisujących częstość występowania danego słowa dla każdego z N dokumentów (o pliku odwróconym powiemy w kolejnym wykładzie).



Główną zaleta reprezentacji wektorowej dokumentów, w stosunku do reprezentacji opartej o zbiór słów kluczowych jest możliwość zdefiniowania miary odległości pomiędzy dokumentami a zapytaniem użytkownika. Niech dana jest wektorowa reprezentacja dokumentu. Dokumenty o podobnej tematyce powinny charakteryzować się podobną częstością występowania tych samych słów kluczowych. Mając do dyspozycji wektorową reprezentację dokumentów, możemy interpretować każdy dokument jako punkt w T-wymiarowej przestrzeni, której wymiary odpowiadają poszczególnym słowom kluczowym. Zatem, do oceny odległości pomiędzy dokumentami, oraz pomiędzy dokumentami a zapytaniem, możemy wykorzystać dowolne miary stosowane do mierzenia odległości w przestrzeni euklidesowej. O miarach tych mówiliśmy szczegółowo na wykładach poświęconych metodom grupowania. Poza znanymi miarami odległości, stosowanymi w wielowymiarowych przestrzeniach euklidesowych, dla potrzeb systemów wyszukiwania informacji opracowano szereg specyficznych miar, takich jak: odległość kosinusowa (o której już mówiliśmy na wykładzie poświeconym algorytmom grupowania) czy miara odległości słów.

Odległość kosinusowa (1)

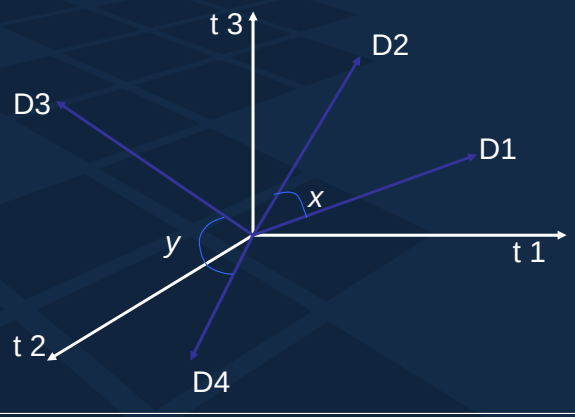
- Najpopularniejszą miara odległości (podobieństwa) dla reprezentacji wektorowej dokumentów jest odległość kosinusowa:

$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1||v_2|}$$

Miara kosinusowa reprezentuje kosinus kąta pomiędzy dwoma wektorami reprezentującymi dokumenty– znormalizowany iloczyn skalarny wektorów

Najpopularniejszą miarą odległości dla reprezentacji wektorowej dokumentów jest miara kosinusowa. Przypomnijmy, że termin „odległość” jest dla nas w pewnym uproszczeniu, synonimem terminu „podobieństwo”. Innymi słowy, dwa dokumenty leżące blisko siebie w przestrzeni słów kluczowych, prawdopodobnie, są do siebie „podobne” w tym sensie, że opisują podobne zagadnienia. Odległość kosinusowa dwóch dokumentów d1 i d2 jest zdefiniowana jako znormalizowany iloczyn skalarny wektorów d1 i d2, i reprezentuje kosinus kąta pomiędzy dwoma wektorami reprezentującymi dokumenty. Przypomnijmy, że tak zdefiniowana miara odległości nie spełnia pierwszego z aksjomatów metryki. Dwa dokumenty d1 i d2 są identyczne, zgodnie z miara kosinusową, wtedy i tylko wartość miary wynosi 1. jeżeli wartość miary wynosi 0, oznacza to, że dokumenty są do siebie niepodobne.

Odległość kosinusowa (2)



Dokumenty, które są „blisko” siebie w przestrzeni wektorowej, prawdopodobnie, dotyczą tej samej problematyki

Prezentowany slajd ilustruje ideę miary kosinusowej. Dane są 3 słowa kluczowe t1, t2, i t3, oraz 4 dokumenty reprezentowane przez wektory D1, D2, D3 i D4. Jeżeli kosinus kąta pomiędzy wektorami reprezentującymi dwa dokumenty jest bliski 0, wówczas kosinus kąta zmierza do 1. Takie dokumenty leżą „blisko” „blisko” siebie w przestrzeni wektorowej, i, prawdopodobnie, dotyczą tej samej problematyki.

Zapytania i dokumenty

- W reprezentacji wektorowej, zapytanie można przedstawić w postaci wektora wag słów kluczowych:

$$q = (q1, q2, ..., qT)$$

- W reprezentacji boolowskiej wagi mogą przyjmować wartość 0 lub 1
- Przykładowo:
  - Zapytanie: bazy_danych (t_1) - (1, 0, 0, 0, 0, 0)
  - Zapytanie : SQL (t_2) - (0, 1, 0, 0, 0, 0)
  - Zapytanie : regresja (t_4) - (0, 0, 0, 1, 0, 0)

Jak już wspominaliśmy, przyjęcie określonej reprezentacji dokumentu tekstowego determinuje postać reprezentacji zapytania użytkownika. W reprezentacji wektorowej, zapytanie q można przedstawić w postaci wektora wag słów kluczowych: q = (q1, q2, ..., qT), gdzie q1, ..., qT oznaczają wagi poszczególnych słów kluczowych w zapytaniu q. Podobnie jak w przypadku reprezentacji tekstu, podobnie w odniesieniu do reprezentacji zapytania, zapytanie może mieć prostą postać boolowską lub tzw. postać pełną. W reprezentacji boolowskiej, wagi slow w wektorze zapytania mogą przyjmować tylko dwie wartości: 0 lub 1. Waga słowa ti w zapytaniu równa 1 oznacza, że słowo ti powinno wystąpić w poszukiwanym dokumencie; jeżeli słowo ti nie występuje w zapytaniu, waga słowa ti w zapytaniu jest równa 0. W pełnej reprezentacji, wagi słów w zapytaniu oznaczają ważność danego słowa w stosunku do innych słów. Przykładowo, zapytanie do bazy danych o dokumenty zawierające słowo t1 ma postać: q1= (1, 0, 0, 0, 0, 0); zapytanie o dokumenty zawierające słowo SQL (t2) ma postać: q2= (0, 1, 0, 0, 0, 0), natomiast zapytanie o dokumenty zawierające słowo „regresja” (t4) ma postać: q3= (0, 0, 0, 1, 0, 0). Stosując odległość kosinusową w celu dopasowania wspomnianych wyżej zapytań do zbioru dokumentów przedstawionych w tablicy TFM ze slajdu nr 15, otrzymujemy, jako najbliższe dokumenty, odpowiednio, dokumenty d2, d3 i d9.

Zapytania do bazy danych: wagi (1)

- Podejście, w którym waga słowa przyjmuje wartość różna od 0, jeżeli słowo występuje gdziekolwiek w dokumencie preferuje duże dokumenty (niekoniecznie relewantne)
- Schemat nadawania wag **TF – IDF**

  TF oznacza - *term frequency* (częstość słów)

  IDF oznacza - *inverse document frequency* (odwrotna częstość dokumentu)

- Waga słowa j (idf_j) **$\ln(N/n_j)$**
  - gdzie N – łączna liczba dokumentów, n_j – liczba dokumentów zawierających słowo j

Zauważmy, że podejście wektorowe do reprezentacji dokumentów tekstowych, w którym waga słowa przyjmuje wartość różną od 0, jeżeli słowo występuje gdziekolwiek w dokumencie, preferuje duże dokumenty (niekoniecznie relewantne). Wynika to z tego, że istnieje większe prawdopodobieństwo wystąpienia danego słowa z zapytania w dokumencie, który jest większy. Ponadto, różne słowa mają różną wartość dyskryminacyjną. Niektóre słowa występują prawie we wszystkich dokumentach, inne tylko w niektórych. Te drugie, siłą rzeczy, lepiej opisują dany dokument – mówimy, że posiadają większą silę dyskryminacyjną (lepiej rozróżniają dokumenty). Rozważany przez nas model macierzy TFM nadaje większą wagę tym słowom, które często występują danym dokumencie, bez uwzględniania ich siły dyskryminacyjnej. W literaturze zaproponowano schemat nadawania wag dokumentom, który uwzględnia sile dyskryminacyjną słów kluczowych. Schemat ten nosi nazwę TF-IDF, gdzie człon TF oznacza wagę TF - częstość słów(term frequency), natomiast człon IDF oznacza wagę IDF - odwrotną częstość dokumentu (inverse document frequency). Waga TF słowa ti w dokumencie di oznacza, podobnie jak poprzednio, liczbę wystąpień słowa ti w dokumencie di. Waga IDF słowa ti w dokumencie di jest zdefiniowana jako logarytm (dziesiętny lub naturalny) z wyrażenia N podzielone przez ni, gdzie N oznacza łączną liczbę dokumentów, natomiast ni oznacza liczbę dokumentów zawierających słowo ti.

Zapytania do bazy danych: wagi (2)

- Wagi TF-IDF faworyzują słowa, które występują w niewielu dokumentach – mają zatem większa siłę dyskryminacyjną
- Waga słowa j w wektorze D_i jest iloczynem częstości występowania słowa w dokumencie d_i i wagi słowa j (idf_j)

	t1	t2	t3	t4	t5	t6
d1	2,54	14,6	4,6	0	0	2,07
d2	3,37	6,93	2,55	0	1,07	0
d3	1,26	11,1	2,55	0	0	0
d4	0,63	4,85	1,02	0	0	0
d5	4,53	21,5	10,2	0	1,07	0
d6	0,63	0	0	11,8	1,42	15,9
d7	0,21	0	1	22,2	4,28	0
d8	0,31	0	0	15,2	1,42	1,38
d9	1	0	0	23,6	9,63	17,3

Waga TF-IDF slowa ti w dokumencie di jest iloczynem wad Tf i IDF słowa ti w dokumencie di. Wagi TF-IDF faworyzują słowa, które występują tylko w niewielu dokumentach – mają zatem większą siłę dyskryminacyjną. Dla ilustracji rozważmy macierz wektorów dokumentów, przedstawioną na slajdzie 22, zawierającą wagi TF-IDF, dla macierzy TFM ze slajdu nr 15. Wagi IDF w prezentowanej macierzy są zdefiniowane jako logarytm naturalny. Zauważmy, że wagi niektórych słów znacząco uległy zmianie. Przykładowo, waga TF-IDF słowa t1 w dokumencie d1, poprzednio wynosząca 24, wynosi 2,54 i jest 6-krotnie mniejsza aniżeli waga TF-IDF słowa t2 w dokumencie d1, która poprzednio wynosiła 21. Wynika to stąd, że słowo t1 występuje praktycznie we wszystkich dokumentach, za wyjątkiem dokumentu d7, stąd jego siła dyskryminacyjna jest stosunkowo mała. Słowo t2 występuje tylko w połowie dokumentów, stąd jego siła dyskryminacyjna jest znacznie większa – stąd większa waga słowa t2.

Zapytania do bazy danych

- Klasyczne podejście do wyszukiwania dokumentów w oparciu o reprezentację wektorową:
  1) Przedstaw zapytanie w postaci wektora boolowskiego (1 - jeżeli słowo występuje w zapytaniu, 0 - jeżeli nie występuje), lub
  2) Przedstaw zapytanie w postaci wektora z wagami TF – IDF
  3) Oblicz odległość kosinusową zapytania od zbioru dokumentów – przeprowadź ranking dokumentów z punktu widzenia zapytania

Jak zatem wygląda realizacja zapytań w modelu wektorowym. Klasyczne podejście do wyszukiwania dokumentów w oparciu o reprezentację wektorową ma następujący schemat.

Zapytanie użytkownika ma postać wektora boolowskiego, w którym na pozycji słowa ti występuje 1, jeżeli słowo ti występuje w zapytaniu, lub 0 jeżeli nie występuje. Słowo ti występuje w zapytaniu, jeżeli użytkownik poszukuje dokumentów zawierających słowo ti. Jeżeli dokumenty są reprezentowane przez wektory z wagami TF-IDF, wówczas zapytanie użytkownika również musi mieć postać wektora z wagami TF – IDF. Następnie, obliczana jest odległość kosinusową zapytania od zbioru dokumentów i przeprowadzany jest ranking dokumentów biorąc pod uwagę odległość dokumentu od zapytania.

Ekstrakcja słów

- Stop lista:
  - Systemy IR często wiążą ze zbiorem dokumentów tzw. „stop listę”, zawierającą zbiór słów uznanych za nierelewantne, np., *a, the, of, for, with*, etc., nawet jeżeli występują one stosunkowo często w zbiorze dokumentów
  - Różne zbiory dokumentów mogą posiadać różne stop listy
- Trzon słowa (ang. *word stem*):
  - niektóre słowa stanowią wariant innego słowa, z którym dzielą wspólny trzon, np., *krowa, krowy, krowi*.

Na zakończenie tego wykładu wróćmy do pewnych problemów związanych z wyszukiwaniem dokumentów, a które wcześniej pominęliśmy. W jaki sposób tworzyć zbiór słów kluczowych? Najlepszym rozwiązaniem jest automatyczna ekstrakcja słów z danego zbioru dokumentów. Ekstrakcja wszystkich słów występujących w zbiorze dokumentów napotyka jeden problem – niektóre słowa często występują w dokumentach, chociaż z pewnością nie niosą ze sobą żadnej informacji semantycznej. Stąd, systemy IR często wiążą ze zbiorem dokumentów tzw. „stop listę”, zawierającą zbiór słów uznanych za nierelewantne. Na przykład, dla języka angielskiego są to następujące słowa: *a, the, of, for, with*, etc. Różne zbiory dokumentów mogą posiadać różne stop listy. Innym problemem jest problem nazywany z języka angielskiego problemem stemmingu. Word stem oznacza trzon słowa. Problem ten wynika stąd, że niektóre słowa stanowią wariant innego słowa, z którym dzielą wspólny trzon, np., *krowa, krowy, krowi*, itd. Stąd, dokonując automatycznej ekstrakcji słów, należałoby dokonać „ujednolicenia” słów, tak aby zbiór słów kluczowych składał się wyłącznie ze słów o różnym trzonie.

Zapytania do bazy danych: problemy

- **Idea wyszukiwania**:
  szukamy dokumentów „podobnych” do zapytania – podobne dokumenty charakteryzują się podobną częstością występowania identycznych słów kluczowych
- **Problemy:**
  - Liczba dokumentów (N) * liczba słów (T) – może być b. duża
  - Duża wymiarowość: bardzo rzadkie wektory dokumentów (trudno wykryć synonimy)
  - Użytkownicy mogą definiować zapytania korzystając z innej terminologii, aniżeli ta zastosowana do opisu dokumentów (odkrywanie wiedzy <> eksploracja danych)

Ostatni problem, o którym chcemy wspomnieć na zakończenie, to problem wymiarowości macierzy wektorów dokumentów, który ma istotny wpływ na jakość i efektywność wyszukiwania dokumentów. Zauważmy, że liczba dokumentów (N) * liczba słów (T) – może być bardzo duża. Jak już wspominaliśmy, macierz wektorów dokumentów charakteryzuje się bardzo dużą wymiarowością i bardzo rzadkimi wektorami dokumentów. W takiej macierzy trudno jest wykryć wszystkie synonimy. Z drugiej strony, użytkownicy mogą definiować zapytania korzystając z innej terminologii, aniżeli ta zastosowana do opisu dokumentów (odkrywanie wiedzy <> eksploracja danych). Naturalne pytanie, które można postawić brzmi: czy można zmniejszyć wymiarowość macierzy nie tracąc znacząco informacji? Odpowiedź na to pytanie udzielimy na następnym wykładzie.