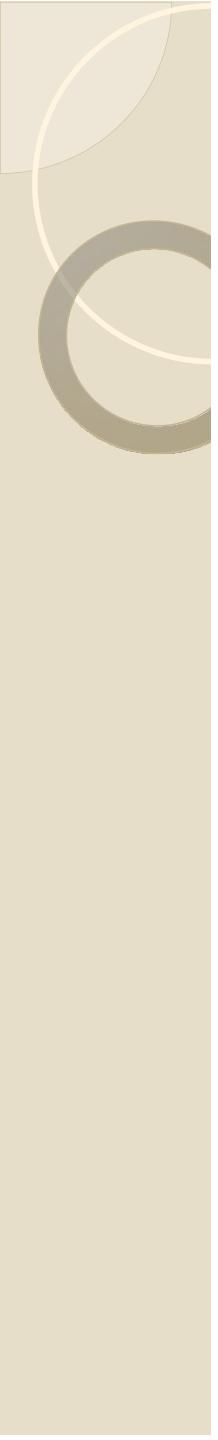


Eksploracja danych w Internecie

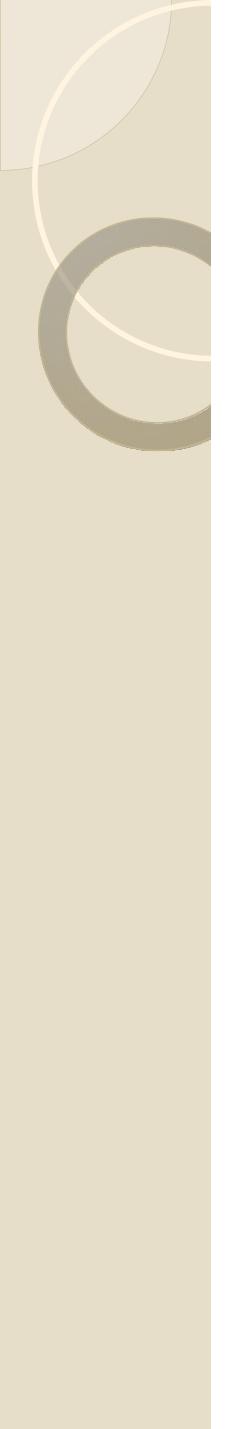
- Wykłady opracowano w oparciu o książkę Ricardo Baeza-Yates, Berthier Ribeiro-Neto, „*Modern Information Retrieval, the concepts and technology behind search*” 2nd edition, ACM Press Books, 2011
 - Z tego samego źródła zaczerpnięto także różne zadania i przykłady wykorzystywane w treści wykładu.
-



Eksploracja danych w Internecie

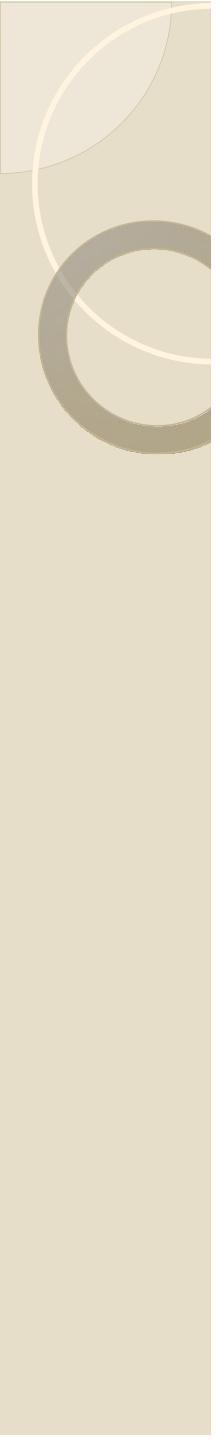
Literatura:

- Ricardo Baeza-Yates, Berthier Ribeiro-Neto, „Modern Information Retrieval, the concepts and technology behind search”, 2nd edition, ACM Press Books, 2011
 - Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, „An Introduction to Information Retrieval”, Online edition (c) 2009 Cambridge UP
 - Anand Rajaraman and Jeffrey D. Ullman „Mining of Massive Datasets”, <http://infolab.stanford.edu/~ullman/mmds/book.pdf>
 - „Eksploracja zasobów internetowych. Analiza struktury, zawartości i użytkowania sieci WWW.”, Zdravko Markov, Daniel T. Larose, PWN, Warszawa 2009
-



Eksploracja danych w Internecie

- Wprowadzenie
 - Interfejsy użytkownika
 - Modele dokumentów
 - Ocena jakości procesu wyszukiwania
 - Klasyfikacja tekstów
 - Indeksowanie przy wyszukiwaniu
 - Wyszukiwanie równoległe
 - Problemy wyszukiwania w sieci
 - Zagadnienia web crawlingu
 - Wyszukiwanie danych multimedialnych
-



Eksploracja danych w Internecie

- Wyszukiwanie informacji (*Information Retrieval*) dotyczy reprezentacji, zapamiętywania organizacji i dostępu do poszczególnych składników informacji
 - Elementy informacyjne to dokumenty, strony sieciowe, katalogi sieciowe, uporządkowane rekordy danych i obiekty multimedialne
 - Pierwotne cele wyszukiwania informacji to: indeksowanie tekstów i poszukiwanie pożądanych dokumentów w ich zbiorach
 - Obecnie wyszukiwanie informacji obejmuje zagadnienia :
 - modelowanie, przeszukiwanie sieci, klasyfikację tekstów, architektury systemów, interfejsy użytkownika, filtrowanie i wizualizację danych, tłumaczenie tekstów
-



Eksploracja danych w Internecie

- Przez ponad 5000 lat, człowiek porządkował informację dla późniejszego jej odzyskania po odpowiednim wyszukianiu
 - Odbywało się to przez zapisywanie, archiwizowanie i indeksowanie papirusów, glinianych tabliczek, węzełków-kipu, wampumów i oczywiście książek

 - W celu przechowywania nośników danych powstały specjalne budowle zwane bibliotekami
 - Najstarsza znana biblioteka powstała na wyspie Elba, pomiędzy 3000 i 2500 rokiem p.n.e.
 - Ok 300 roku p.n.e. powstała słynna Biblioteka Aleksandryjska
-



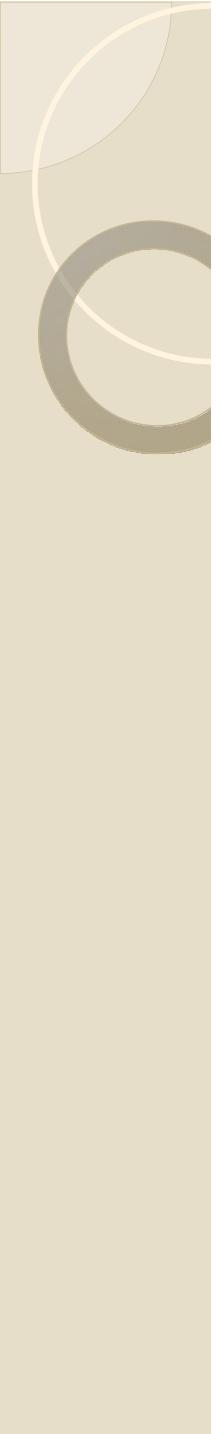
Eksploracja danych w Internecie

- Ponieważ ilość informacji przechowywanej w bibliotekach ciągle rośnie buduje się *indeksy* – specjalizowane struktury danych do szybkiego jej wyszukiwania
 - Dawniej indeksy tworzone manualnie jako zbiory kategorii, z dodatkowymi etykietami powiązanymi z każdą kategorią
 - Pojawienie się nowoczesnych komputerów pozwoliło na automatyczne budowanie dużych indeksów
 - W latach 50-tych badacze tacy jak Hans Peter Luhn, Eugene Garfield, Philip Bagley i Calvin Moores wypracowali pojęcie wyszukiwania informacji - *Information Retrieval (IR)*
-



Eksploracja danych w Internecie

- W 1963 roku Joseph Becker i Robert Hayes opublikowali pierwszą książkę na temat IR
 - W latach 60-tych badania w tym temacie prowadzili m. in. Karen Sparck Jones i Gerard Salton; doprowadziły one do utworzenia definicji *TF-IDF term weighting scheme* (Term Frequency - Inverse Document Frequency) określającego wzór wagi przypisanej poszczególnym pozycjom (pojęciom) w dokumencie
 - W 1971 r. Jardine i van Rijsbergen wprowadzili hipotezę klasteringu (*cluster hypothesis*)
 - W 1979 r. van Rijsbergen opublikował klasyczną pozycję książkową *Information Retrieval* omawiającą model probabilistyczny
-



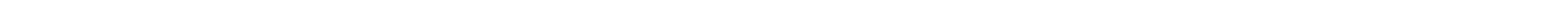
Eksploracja danych w Internecie

- W 1983 r. Salton i McGill opublikowali książkę *Introduction to Modern Information Retrieval* omawiającą model wektorowy pozyskiwania danych
 - Biblioteki były pierwszymi instytucjami wykorzystującymi systemy IR dla pozyskiwania informacji w formie przeszukiwania kart katalogowych
 - Następnie tę prostą funkcjonalność rozszerzono o:
 - analizę nagłówków, szukanie wg. słów kluczowych, specjalizowane operatory zapytań
 - Obecnie rozwój systemów IR koncentruje się na doskonaleniu interfejsów graficznych, wspomaganiu sprzętowym i cechach hipertekstowych dokumentów
-



Eksploracja danych w Internecie

- Przed epoką Internetu, wydobywaniem informacji interesowali się najczęściej bibliotekarze i eksperci od przetwarzania danych
- Internet stał się obecnie największym archiwum wiedzy w historii ludzkości
- Znalezienie pożądanej informacji w Internecie, z powodu gigantycznych rozmiarów tego repozytorium wymaga użycia nowej technologii wyszukiwania informacji





Problem wyszukiwania informacji

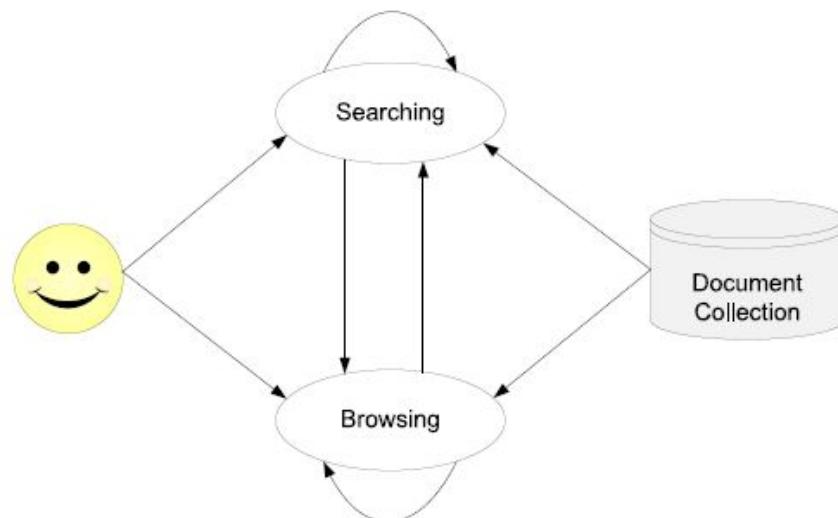
- Pełny opis wymaganej informacji podany przez użytkownika nie zawsze jest trafnym zapytaniem do systemu IR
- Użytkownik sieci może także sformułować swoje wymagania w formie zapytania
- Najistotniejszy dla wyszukiwania jest zbiór słów kluczowych (*keywords*) lub terminów indeksujących (*index terms*).
- Celem systemów IR jest jak najtrajniejsze wydobycie informacji istotnej dla użytkownika na podstawie podanego zapytania



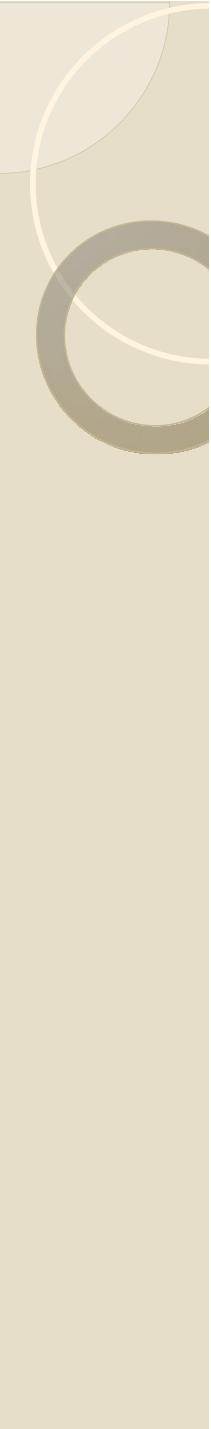
Problem wyszukiwania informacji

- System IR powinien uszeregować elementy informacji według ich istotności w zapytaniu użytkownika
 - Celem systemu IR jest wydobycie wszystkich elementów istotnych dla zapytania użytkownika i jak najmniejszej ilości elementów nieistotnych
 - Pojęcie istotności informacji w systemach IR jest kluczowe
-

Problem wyszukiwania informacji



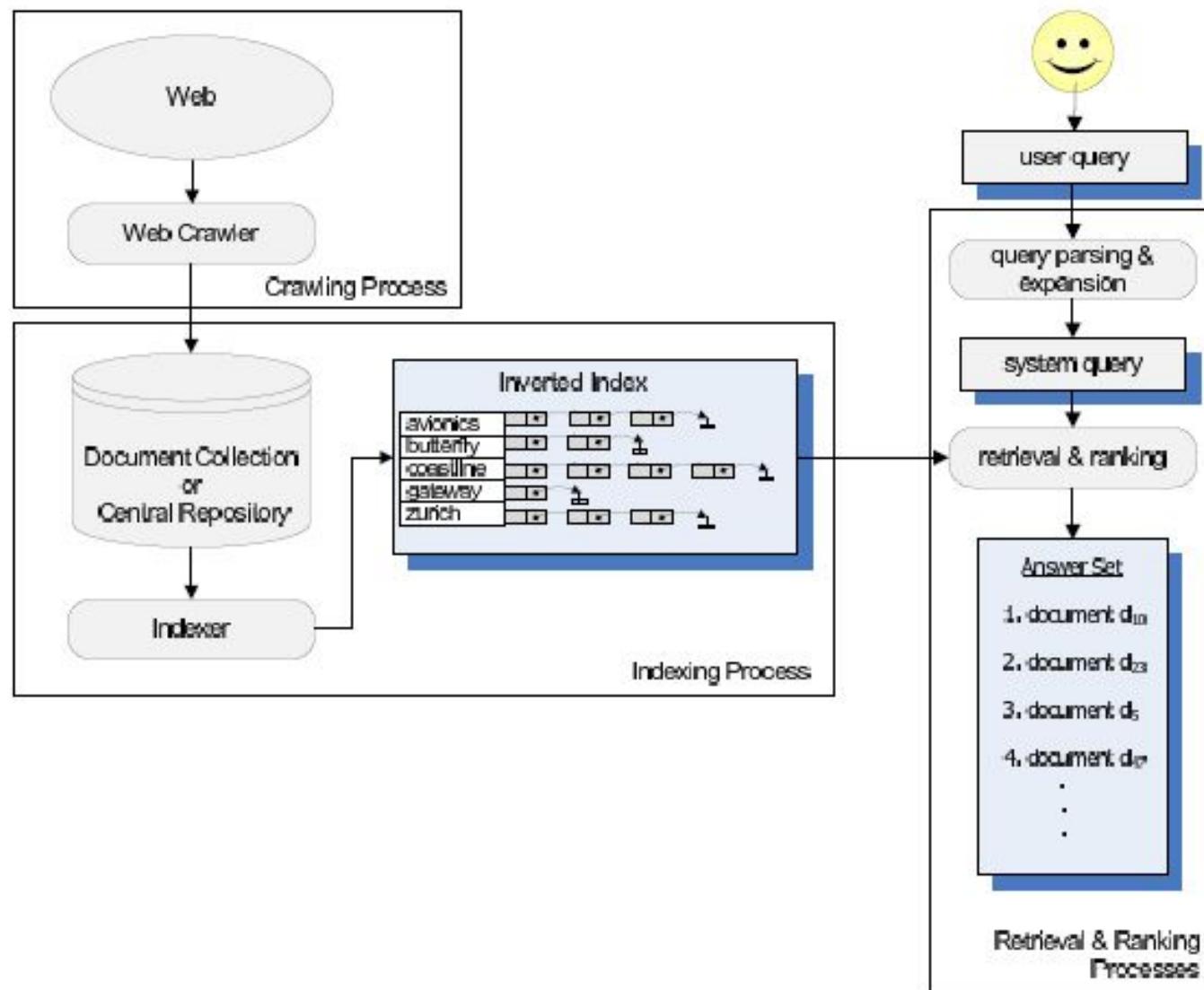
- Jeżeli użytkownik precyzuje konkretny temat, często w formie zapytania, to mówi się że poszukuje, wyławia (*searching*) lub pyta o informację (*querying*)
- Jeżeli użytkownik formułuje wymagania szeroko lub nieprecyzyjnie to mówi się o żeglowaniu (*navigating*) lub przeglądaniu (*browsing*) dokumentów w Internecie



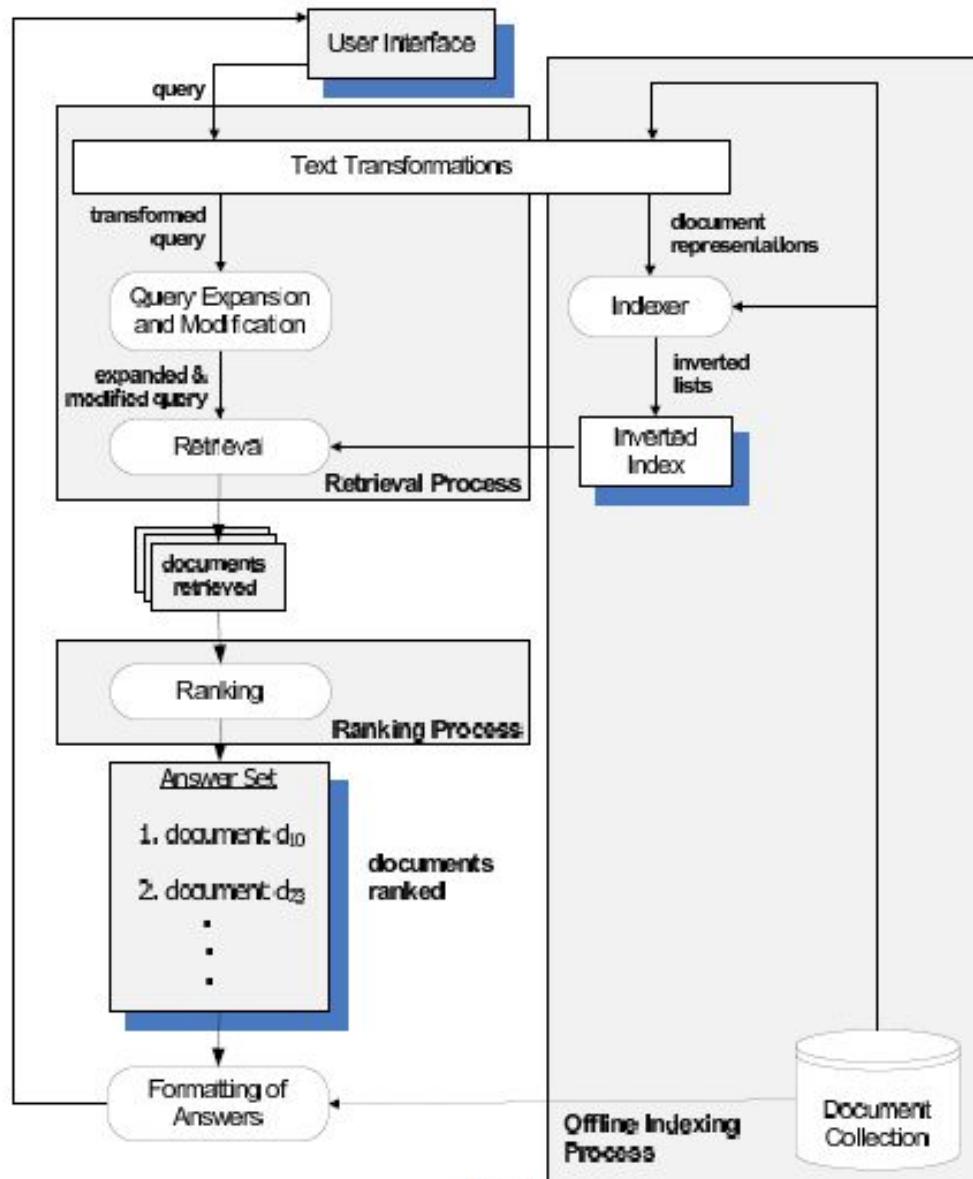
Wyszukiwanie danych i informacji

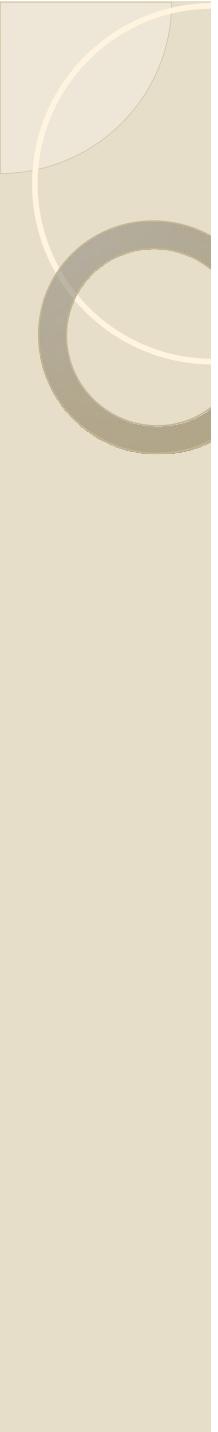
- Wyszukiwanie danych (*Data retrieval*): zadanie definiujące ściśle, które dokumenty ze zbioru zawierają słowa kluczowe z zapytania
- System wyszukiwania danych (*Data retrieval system*) to np. relacyjna baza z danymi o dokładnie określonej strukturze i semantyce. Zasadniczo nie dopuszcza żadnych błędów w wynikach wyszukiwania.
- Wyszukiwanie danych nie rozwiązuje problemu wydobywania informacji na dany temat

Architektura systemu IR



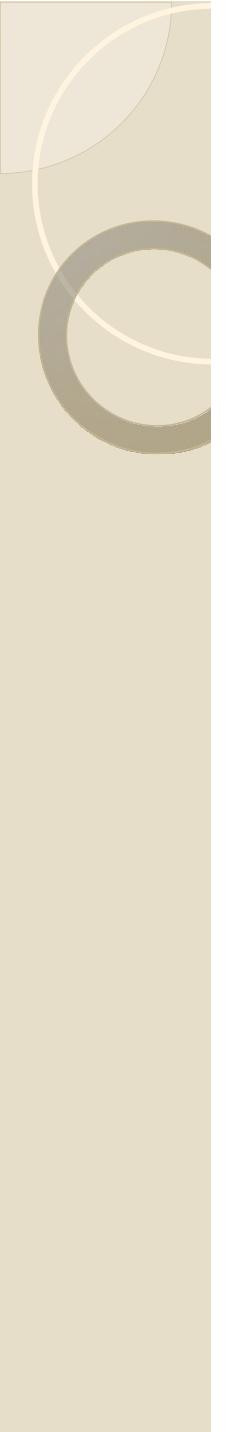
Proces wyszukiwania i szeregowania





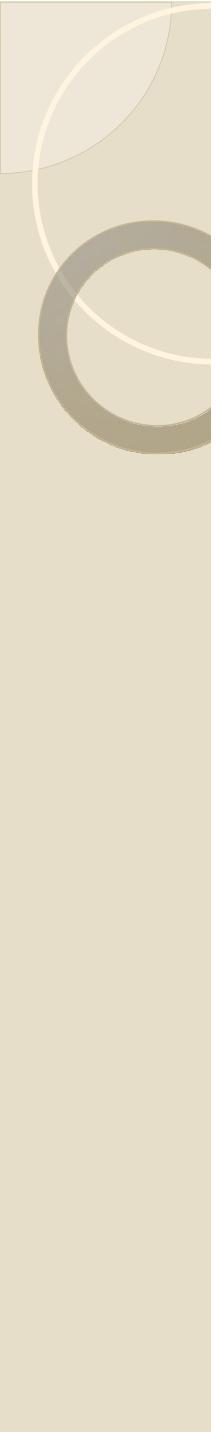
Sieć WWW - historia

- W 1990 roku Berners-Lee
 - opracował protokół HTTP,
 - zdefiniował język HTML,
 - napisał pierwszą przeglądarkę, którą nazwał World Wide Web,
 - opracował pierwszy serwer sieciowy.
 - W 1991, udostępnił serwer i przeglądarkę w Internecie
 - Tak narodziła się sieć
-



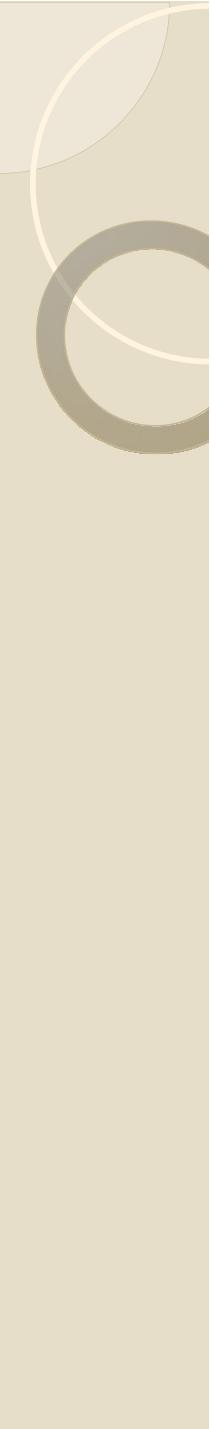
Wyszukiwanie w sieci

- Wyszukiwarki sieciowe są najpopularniejszymi aplikacjami stosującymi technologię IR wraz z jej zasadniczymi elementami: szeregowaniem i indeksowaniem.
- Sieć narzuca specyficzną charakterystykę wyszukiwania zbioru dokumentów – strony rozproszone w milionach witryn połączonych hiperlinkami; rozproszone dokumenty o pożądanych cechach są wydobywane i kopowane w jedno miejsce przed ich indeksowaniem. Taki sposób wyszukiwania stron w procesie IR nazywa się „pełzaniem (po stronach)” (*crawling*).



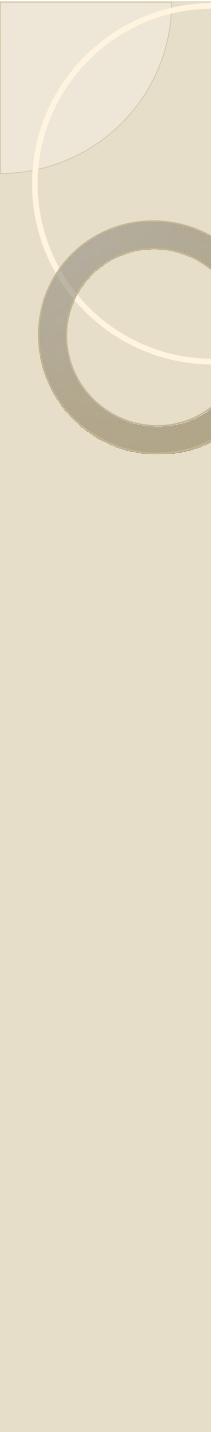
Wyszukiwanie w sieci

- Drugi wpływ sieci na wyszukiwanie to krytyczne znaczenie jakości i skalowalności procesu IR
- Wobec przeszukiwania dużych zbiorów w sieci przwidywanie istotności danych staje się bardzo istotne
- Sieć stanowi także medium do prowadzenia biznesu; strony zawierają linki do ładowania programów, adresy, numery telefonów instytucji itp.
- Przy wyszukiwaniu w sieci należy eliminować spamy
- Zapewnienie bezpieczeństwa, prywatności, praw autorskich i patentowych
- Skanowanie i rozpoznawanie pisma przy wyszukiwaniu w różnych językach



Interfejsy użytkownika

- Wykłady opracowano w oparciu o książkę Ricardo Baeza-Yates, Berthier Ribeiro-Neto, „*Modern Information Retrieval, the concepts and technology behind search*” 2nd edition, ACM Press Books, 2011
 - Z tego samego źródła zaczerpnięto także różne zadania i przykłady wykorzystywane w treści wykładu.
-



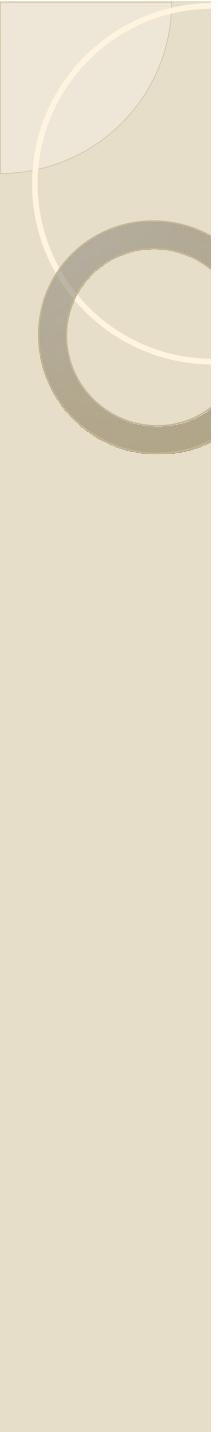
Interfejsy użytkownika

- Rolą interfejsu użytkownika jest pomoc w sformułowaniu zapytania i odebraniu wymaganych informacji
 - Interfejs powinien także umożliwić:
 - wybór źródła informacji,
 - zrozumienie wyników wyszukiwania,
 - śledzenie postępu w wyszukiwaniu
 - Interakcja użytkownika z interfejsem wyszukującym zależy od:
 - typu postawionego zadania,
 - dotychczasowej wiedzy o użytkowniku,
 - ilości czasu i wysiłku wkładanego w proces wyszukiwania.
 - Rozróżnia się wyszukiwanie informacji (*information lookup*) i wyszukiwanie rozpoznawcze (*exploratory search*)
-



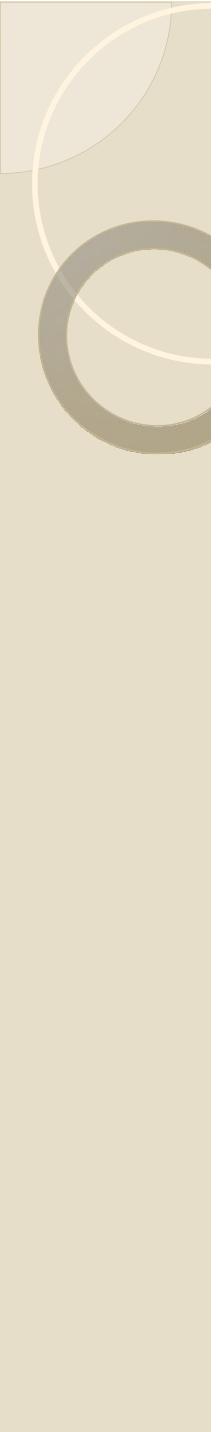
Interfejsy użytkownika

- Wyszukiwanie informacji (*information lookup*) :
 - jest podobne do wyszukiwania faktów i odpowiedzi na pytania,
 - może być wypełnione przez dyskretne informacje jak liczby, daty, nazwy lub strony sieciowe,
 - pracuje poprawnie w trybie standardowych interakcji z siecią.
- Wyszukiwanie rozpoznawcze (*exploratory search*) można podzielić na zadania śledzenia i uczenia się
- Wyszukiwanie uczące się (*learning search*) wymaga:
 - więcej niż jednej akcji zapytanie-odpowiedź,
 - czasu od wyszukiwacza,
 - odczytywania wielu porcji informacji,
 - złożenia różnych treści do sformowania odpowiedzi.



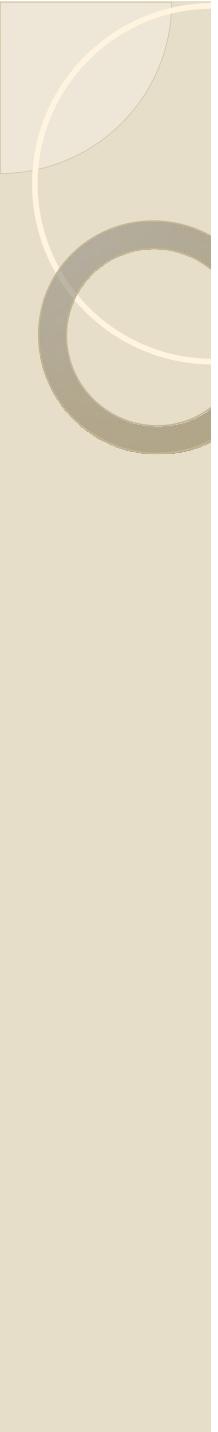
Interfejsy użytkownika

- Śledzenie (*investigating*) odnosi się do długotrwałych procesów które:
 - obejmują wiele iteracji w dłuższym czasie,
 - zwracają informacje podlegające ocenie przed uzupełnieniem bazy wiedzy o użytkowniku,
 - mogą być związane z wyszukiwaniem istotnych informacji w wielkich zbiorach odpowiedzi
- *Sensemaking* (Karl Weick 1969) -organizowanie wiedzy: proces iteracyjny tworzenia konceptualnej reprezentacji dużych zbiorów odpowiedzi.
- Przykładowe zadania (poza wyszukiwaniem) wymagające inteligentnego wyboru
 - badanie przypadków prawnych,
 - epidemiologia (*disease tracking*)
 - studiowanie uwag użytkowników dla poprawy obsługi,
 - analizy danych biznesowych,



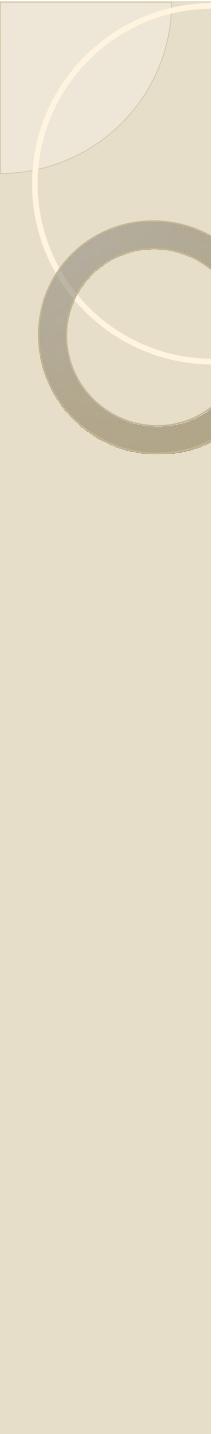
Interfejsy użytkownika

- Etapy klasycznego wyszukiwania informacji:
 - identyfikacja problemu,
 - sformułowanie potrzeb informacyjnych,
 - utworzenie zapytania,
 - ocena rezultatów wyszukiwania.
- Najnowsze modele uwzględniają dynamiczne aspekty wyszukiwania:
 - użytkownicy uczą się w trakcie wyszukiwania,
 - potrzeby informacyjne użytkowników zmieniają się podczas przeglądania wyników wyszukiwania.
- Model dynamicznego wyszukiwania nazywa się także „zbieraniem jagód” (*berry picking*) lub wyszukiwaniem zorientowanym (*orienteering*)



Interfejsy użytkownika

- Na podstawie analizy logów wyszukiwania Jansen stwierdził, że ok. 52% użytkowników dokonuje modyfikacji zapytań
- Modele wyszukiwania są formułowane w kategoriach dwóch zasadniczych strategii:
 - odzwierciedlającej przemyślane zachowanie specjalistów prowadzących wyszukiwanie,
 - odtwarzającej mniej planowe a bardziej odruchowe zachowania typowego poszukiwacza informacji
- Nawigacja: wyszukujący sprawdza strukturę informacji i przegląda dostępną informację (browsing)
- Taka strategia jest preferowana gdy struktura informacji jest dopasowana potrzeb użytkownika – dostępne są odpowiednie linki



Interfejsy użytkownika

- Jeżeli właściwe linki nie są dostępne takie przeglądanie może być denerwujące
- Przykład: poszukiwanie sterownika drukarki laserowej
- Użytkownik wybiera słowa: *printers*, *laser printers* i dalej sekwencję linków

HP laser printers

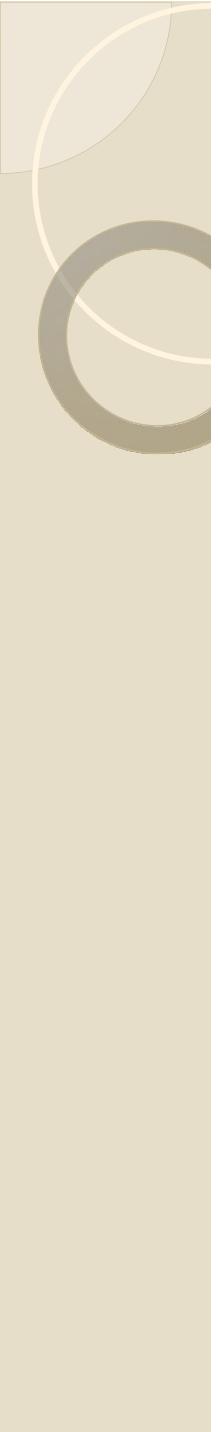
HP laser printers model 9750

software for HP laser printers model 9750

software drivers for HP laser printers model 9750

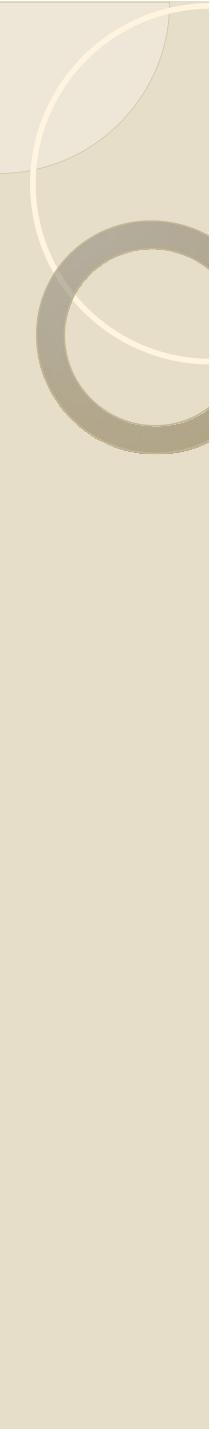
*software drivers for HP laser printers model 9750 for the
Win98 operating system*

- Taki rodzaj interakcji jest akceptowalny dokąd kolejne poprawianie dokładności wyszukiwania ma sens
-



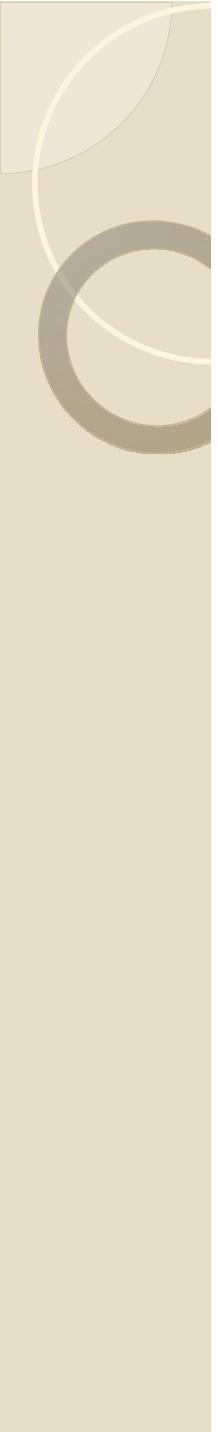
Interfejsy użytkownika

- Rozliczne studia prowadzone nad procesami wyszukiwania doprowadziły do wniosku, że użytkownicy często
 - reformułują zapytania z niewielkimi modyfikacjami,
 - ponownie szukają informacji znalezionych poprzednio.
- Badacze opracowali wsparcie interfejsów wyszukiwania z odwołaniem do historii zapytań i ponawianiem zapytań (*query history and revisitaiton*)
- Badania pokazują, że wyszukujący zwracają uwagę na kilka początkowych pozycji w rankingu wyszukanych odpowiedzi.
- Użytkownicy są zazwyczaj przekonani, że pierwsze wyniki w zbiorze odpowiedzi są lepsze niż pozostałe.



Aktualne interfejsy wyszukiwania

- Sposoby rozpoczętia sesji wyszukiwania informacji online :
 - Użycie wyszukiwarki internetowej,
 - wybór strony spośród stron ostatnio odwiedzonych (historia) lub zapamiętanych (zakładki),
- Wyszukiwarki z zakładkami online są stosowane przez mniejszą liczbę użytkowników. np. Delicious.com. Zakładki tego typu są gromadzone w sieci na odpowiednich portalach.
- Zapytania w wyszukiwarkach są zadawane w formie tekstowej
- Zapytania typowo są krótkie i zawierają jedno do trzech słów
 - jeśli wyniki nie są odpowiednie użytkownik reformuluje zapytanie,
 - jeśli wyniki są odpowiednie użytkownik nawiguje do stron, o których sądzi, że są istotne.



Specyfikacja zapytań

- Przed epoką sieci systemy wyszukujące typowo wspierały operatory boolowskie i składnię opartą o komendy.
- Badania Jansena (2007) na zbiorze 1.5M zapytań dały wyniki:
 - 2.1% zapytań zawiera operatory boolowskie ,
 - 7.6% zapytań ma inną składnię, głównie podwójne cudzysłowy.
- White (2007) badał logi interakcji ok. 600 000 użytkowników i stwierdził:
 - 1.1% zapytań zawiera jeden lub więcej operatorów,
 - 8.7% użytkowników używało zawsze operatorów.



Specyfikacja zapytań

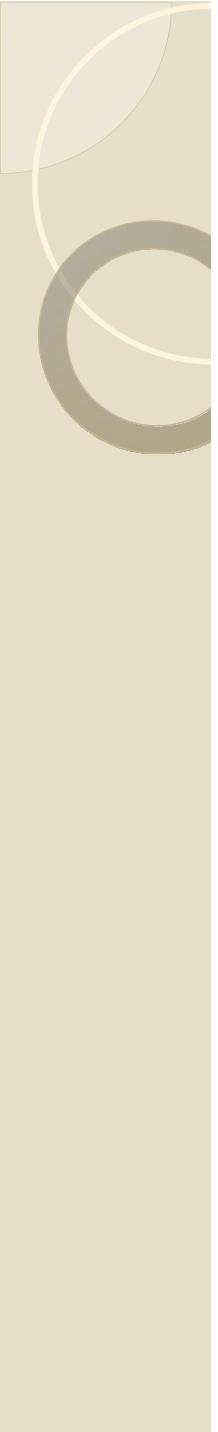
- Ok. 1997 roku Google wprowadził wyłącznie zapytania koniunktywne, które wkrótce stały się normą.
 - Google dodał pojęcie informacji przybliżonej i wprowadził ranking stron (PageRank).
 - W miarę rozwoju sieci pojawiły się poprawne odpowiedzi na dłuższe pytania stawiane w formie fraz.
 - Standardowym interfejsem wejściowym zapytania jest pole edycyjne (*search box entry*).
-

Specyfikacja zapytań

- W niektórych przeglądarkach formy wejściowe zapytań występują razem z filtrującymi np. yelp.com



- Przewiduje się także inne specjalizowane typy wejść:
 - Np. zvents.com rozpoznaje określenia czasu typu „tomorrow” i pozwala na wprowadzanie różnych formatów daty.
 - Wyszukiwanie dla zwrotu „comedy on wed” automatycznie wybiera najbliższą środę



Specyfikacja zapytań

- Niektóre interfejsy pokazują listy sugestii zapytań i typów zapytań
- Nazywa się to autouzupełnianiem, autosugerowaniem, lub dynamiczną sugestią zapytań (*auto-complete, auto-suggest, or dynamic query suggestions*)
- Sugestie dotyczą często uzupełnienia wprowadzonych znaków traktowanych jako prefiksy, rzadziej podają uzupełnienia środkowych liter
- Podpowiadane mogą być także synonimy dotychczas używanych słów

Specyfikacja zapytań

- Sugestie dynamicznych zapytań Netflix.com

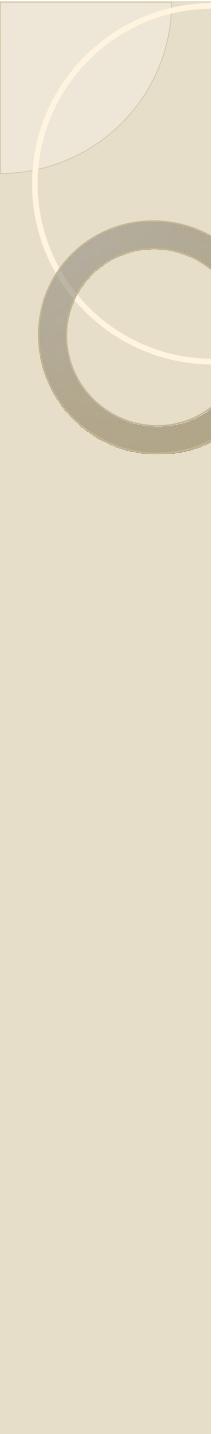


Specyfikacja zapytań

- Źródła sugestii zapytań dynamicznych:
 - Historia zapytań użytkownika,
 - Zbiór metadanych, które projektant strony internetowej uważa za istotne,
 - Cały tekst zawarty na stronie internetowej.
- Sugestie dynamiczne grupowane wg typów z NextBio.com

The screenshot shows the NextBio search interface. In the top left, the NextBio logo is visible. Below it, there are search fields and buttons: 'search > embr', 'experiments(0)', and 'lit'. A dropdown menu is open over the search bar, displaying suggestions starting with 'emb'. The suggestions are categorized by type: compound, gene, tissue, and compound again. The first suggestion is 'compound > EMB (Emb)'. Other suggestions include 'EMB (MGC71745, Gp70, AL022799, MGC21425)', 'EMB (Ethambutol)', 'EMB (Methylurethane)', 'Embl1', 'Embl2', 'EMBBA (Embba)', 'Embryo', 'Embarin (Allopurinol)', and 'Embutox (Butoxone)'. To the right of the dropdown, there is a 'search' button and a 'relevance by' link. The background of the slide features a decorative vertical bar on the left side with abstract circular patterns.

Suggestion Type	Suggestion
compound	EMB (Emb)
gene	EMB (MGC71745, Gp70, AL022799, MGC21425)
compound	EMB (Ethambutol)
compound	EMB (Methylurethane)
gene	Embl1
gene	Embl2
compound	EMBBA (Embba)
tissue	Embryo
compound	Embarin (Allopurinol)
compound	Embutox (Butoxone)



Prezentacja wyników wyszukiwania

- Przy prezentacji wyników wyszukiwania:
 - dokumenty muszą być pokazywane w całości lub wyszukującemu należy przedstawić pewną reprezentację ich zawartości
- Surogat dokumentu zawiera informację streszczającą ten dokument
 - Ta informacja jest kluczowa dla sukcesu wyszukiwania,
 - Projektowanie właściwych surogatów dla dokumentów jest aktywnym polem badań,
 - Jakość surogatów może wpływać na postrzegane znaczenie wyników wyszukiwania



Prezentacja wyników wyszukiwania

- Zawsze przedstawia się tytuł strony wraz z adresem URL i inne metadane
- Przy przeszukiwaniu kolekcji podawane są dane o dacie publikacji i autorze
- Wycięte fragmenty dokumentów zawierające wyszukiwane pojęcia lub streszczenia decydujące o ich rankingu są także istotne

- Standardowo wyniki wyświetla się jako pionowe listy *SERP* (*Search Engine Results Page*) podsumowań (streszczeń) tekstowych
- W pewnych przypadkach obok informacji tekstowej dołącza się specjalne metadane (*blended results*)

Prezentacja wyników wyszukiwania

■ Przykład -Google

[Koniunktura gospodarcza – Encyklopedia Zarządzania](#)

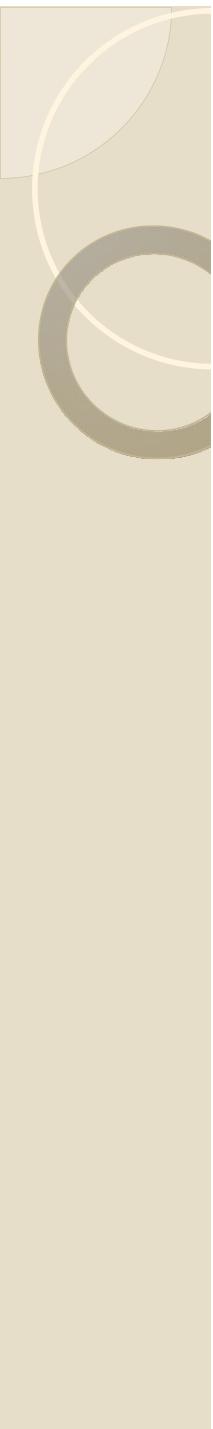
[mfles.pl/pl/index.php/Koniunktura_gospodarcza](#)

8 Lis 2009 – Definicja. **Koniunktura** gospodarcza to wszelkie zmiany aktywności gospodarczej przejawiające się w zmianach podstawowych wskaźników ...

[GUS - Główny Urząd Statystyczny - Koniunktura w przemyśle ...](#)

[www.stat.gov.pl](#) › Strona główna › Koniunktura

23 Lut 2012 – **Koniunktura** w przemyśle, budownictwie, handlu i usługach ... Ogólny klimat **koniunktury** w przetwórstwie przemysłowym w lutym oceniany jest ...



Prezentacja wyników wyszukiwania

- Np. zapytanie „rainbow” w Yahoo zwraca teksty i

The screenshot shows a web browser window with the Yahoo search interface. The search term "rainbow" is entered in the search bar. The results page displays various links and images related to rainbows.

Search Results:

- Rainbow - Image Results**:
- An image of a rainbow over clouds with the word "Rainbow" written below it.
- A photograph of a double rainbow in a blue sky.
- A vibrant, abstract image of a rainbow pattern.
[More rainbow images](#)
- Yahoo! Shortcut - About**
- Rainbow - Wikipedia, the free encyclopedia**
[Visibility](#) | [Scientific...](#) | [Variations](#) | [Scientific...](#)
A rainbow is an optical and meteorological phenomenon that causes a spectrum of light to appear in the sky when the Sun shines onto droplets of moisture in the Earth's atmosphere. They take the form of a multicoloured arc...

en.wikipedia.org/wiki/Rainbow - 121k - [Cached](#)
- Rainbow (band) - Wikipedia, the free encyclopedia**
[History](#) | [Member history](#) | [Discography \(studio albums\)](#) | [Other reading](#)
Rainbow were an English hard rock and heavy metal band formed by former Deep Purple guitarist Ritchie Blackmore in 1975. In addition to Blackmore, the band

Prezentacja wyników wyszukiwania

- Np. zapytanie o nazwę drużyny sportowej wydobywa wyniki ostatnich meczów i linki do stron zakupu biletów

Web Images Maps News Video Gmail more ▾

Google rockets Advanced Search Preferences

Web Video News Blogs Images Results 1 - 10 of about 22,800,000 for rockets [definition].

NBA.com - Houston Rockets
Official site containing news, scores, audio and video files, player statistics, and schedules.
www.nba.com/rockets/ - 7k - [Cached](#) - [Similar pages](#)

[Scores and Schedule](#) [Rockets Power Dancers](#)
[Tickets](#) [Stats](#)
[E-Brochure](#) [Giveaway Nights](#)
[Players](#) [Video Gallery](#)

[More results from nba.com »](#)

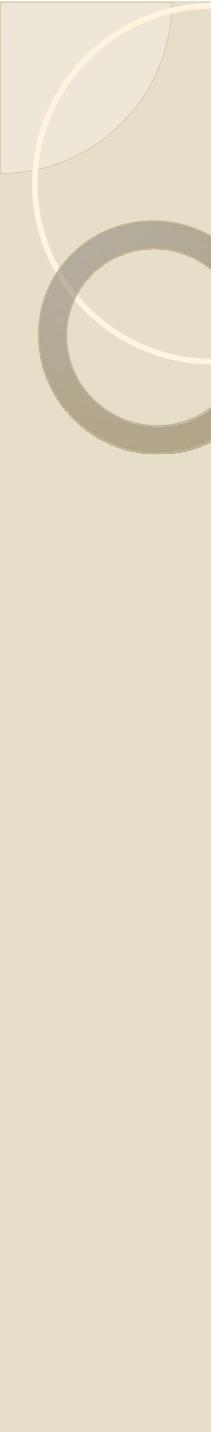
ROCKETS: 2008-09 ROCKETS SCHEDULE
Rocket Power Dancers · Clutch the Bear · Red Rowdies · Fan Photos · Launch Crew · Little Dippers · Recycle Item of the Month ... ROCKETS SCHEDULES & RESULTS ...
www.nba.com/rockets/schedule/ - 73k - [Cached](#) - [Similar pages](#)

Rocket - Wikipedia, the free encyclopedia
A rocket or **rocket** vehicle is a missile, aircraft or other vehicle which obtains thrust by the reaction of the **rocket** to the ejection of fast moving fluid ...
en.wikipedia.org/wiki/Rocket - 205k - [Cached](#) - [Similar pages](#)

Video results for rockets

 [Lakers vs rockets 11/9/2008](#)
[kobe bryant huge ...](#)
7 min
www.youtube.com

 [How To Make a Mentos Coke Rocket](#)
one.rever.com



Prezentacja wyników wyszukiwania

- Podkreślanie wyszukiwanych pojęć w odpowiedzi poprawia ocenę jakości wyników. Podkreślanie może być też stosowane w pełnych dokumentach.
- Wyzwanie stanowi dobór i rozmiar tekstu w streszczeniu dokumentu
- Należy wybrać pewną równowagę między pokazywaniem zdań zawierających poszukiwane termy a pokazaniem ciągłego fragmentu tekstu z tymi termami

- Najlepszy sposób wyświetlania wyników jest zawsze powiązany z intencją zapytania:
 - dla pewnych typów informacji dłuższe odpowiedzi są uważane za lepsze
 - w pytaniach nawigacyjnych o bieżące aktualności preferuje się zwięzłe odpowiedzi

Prezentacja wyników wyszukiwania

- Przykład równoległego wyciągania rysunków z artykułów

BioText SEARCH ENGINE

CXCR4 HIV-1

Search Over: Full Text & Abstracts Figure Captions (List) Figure Captions (Grid) Tables Sort By: Relevance Results/Page: 20

Results 1-20 of 168 searching full text < 1 2 3 4 >

ABSTRACTS FULL-TEXT EXCERPTS FIGURES

Down-regulation of cell surface CXCR4 by HIV-1
Choi, B., Gatti, P., Fermin, C., Vigh, S., Haislip, A., Garry, R. (2008) *Virology Journal*.

ABSTRACT
CXC chemokine receptor 4 (**CXCR4**), a member of the G-protein-coupled chemokine receptor family, can serve as a co-receptor along with CD4 for entry into the cell of T-cell tropic X4 human immunodeficiency virus type 1 (**HIV-1**) strains. Productive infection of T-lymphoblastoid cells by X4 **HIV-1** markedly reduces cell-surface expression of CD4, but whether or not the co-receptor **CXCR4** is down-regulated has not been conclusively determined. ... [Show Full Abstract](#)

FULL-TEXT EXCERPTS
...family function as coreceptors with the primary receptor CD4 to allow entry of various strains of human immunodeficiency virus type 1 (**HIV-1**) into the cells [5-8]. T-cell-tropic X4 **HIV-1** use CD4 and chemokine receptor **CXCR4** for entry into target cells, whereas macrophage-tropic R5 **HIV-1** use CD4 and chemokine receptor CCR5. Dual-tropic strains can use either CCR5 and **CXCR4** as co-receptors...
...manner [29,30]. Chemokine receptors, including CCR5 and **CXCR4**, can be... [Show Full Excerpts](#)

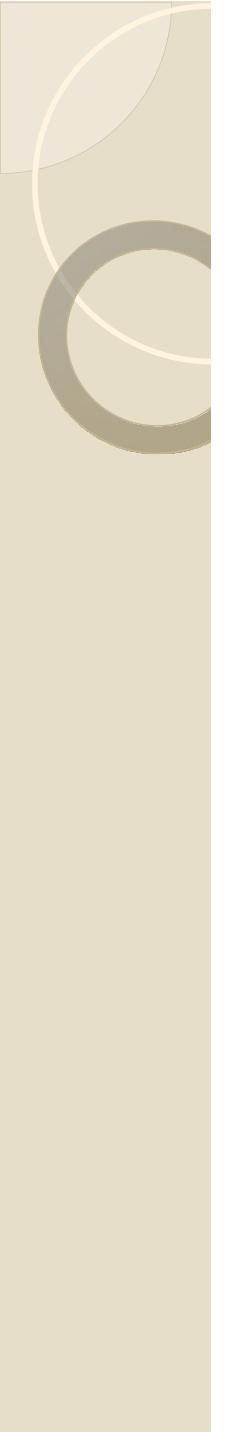
FIGURES FROM ARTICLE:

VIEW FULL ARTICLE: [HTML](#) | [PDF](#)

Differential control of CXCR4 and CD4 downregulation by HIV-1 Gag
Valiathan, R., Resh, M. (2008) *Virology Journal*.

ABSTRACT
The ESCRT (endosomal sorting complex required for transport) machinery functions to sort cellular receptors into the lumen of the multivesicular body (MVB) prior to lysosomal

FIGURES FROM ARTICLE:



Przeformułowanie zapytań

- Pomoc w reformatowaniu zapytania
 - pokazywane są termy zapytania i związane z wyszukanym dokumentami
 - specjalny przypadek to korekty pisowni lub sugestie określeń powiązanych
 - **Term expansion** – interfejsy wyszukiwania w coraz większym stopniu korzystają z podpowiedzi termów powiązanych z pierwotnym zapytaniem
-

Przeformułowanie zapytań

■ Wyniki zapytania „IMF” w przeglądarce IE

Live Search | MSN | Windows Live United States | Options | cashback | Sign in

Live Search IMF 

Web 1-10 of 9,500,000 results · Advanced
See also: [Images](#), [Video](#), [News](#), [Maps](#), [More](#) ▾

IMF -- International Monetary Fund Home Page
IMF Home page with links to News, About the IMF, Fund Rates, IMF Publications, What's New, Standards and Codes, Country Information and featured topics
www.imf.org/external/index.htm · [Cached page](#)

[IMF Country Page](#) [Publications](#)
[Data And Statistics](#) [What The IMF Does](#)
[About The IMF](#) [How To Contact Us](#)
[IMF Recruitment](#) [For Students](#)

[Show more results from www.imf.org](#)

Western Asset Inflation Management Fund Inc (IMF)

 **▲ 15.34**
+0.07 (0.46%)

Volume	12,567
P/E Ratio	NA
Market Cap.	NA

[Company Report](#) · [Financial Results](#) · [Earning Estimates](#) · Quotes by Comstock, 20 min delay - Data in US Dollars
· [Western Asset Inflation Management Fund Inc Announces...](#) BusinessWire 6 days ago
· [Western Asset Inflation Management Fund Inc. Announces...](#) BusinessWire 3/4/2009
Helpful? [Yes](#) | [No](#)

International Monetary Fund - Wikipedia, the free encyclopedia
The **International Monetary Fund (IMF)** is an international organization that oversees the global financial system by following the macroeconomic policies of its member countries, in particular those with an impact on exchange rates and the balance of payments. It is an organization formed to stabilize international exchange rates and facilitate development. [2]
[Organization and purpose](#) · [Data Dissemination ...](#) · [Membership qualifications](#)
en.wikipedia.org/wiki/IMF · [Cached page](#)

Related searches

[International Monetary Fund](#)
[The World Bank](#)
[International Music Feed](#)
[IMF Download](#)
[History IMF](#)
[International Ministerial Fellowship](#)
[Indian Mountaineering Foundation](#)
[IMF Archive Manager](#)

Sponsored sites

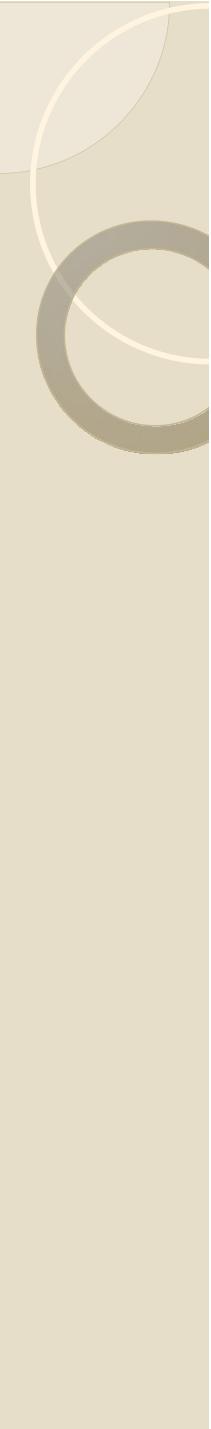
Sirana SpamCenter
Web-based application that enables administration of MS Exchange IMF.
www.sirana.com

See your message here...



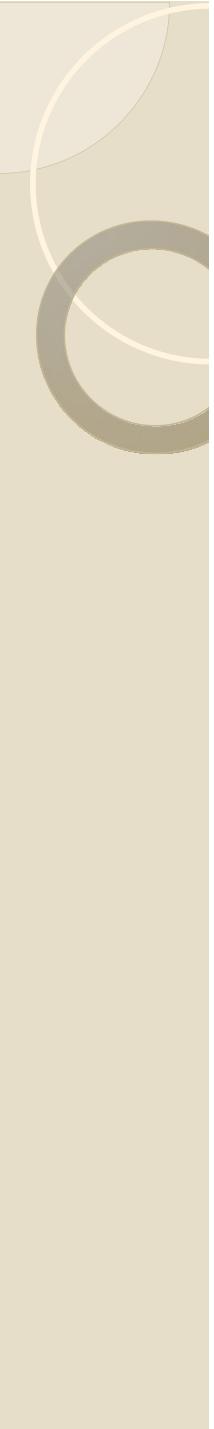
Przeforumułowanie zapytań

- Niektóre sugestie termów opierają się na danych całej sesji wyszukiwania danego użytkownika.
- Inne bazują na typowym zachowaniu innych użytkowników zadających podobne pytania
 - jedna strategia to wyświetlanie tych pytań
 - druga strategia: wyszukiwanie termów w dokumentach wyszukanych dla innych użytkowników
- Metoda „**relevance feedback**” zakłada, że użytkownik wskaże, które dokumenty wybrano trafnie lub które termy wybrane z dokumentów są istotne
- System wyszukiwania na tej podstawie formuluje nowe, poprawione zapytanie
- Metoda ta nie jest stosowana w aktualnych wyszukiwarkach ze względu na uczestnictwo użytkownika i jego ograniczoną wiedzę



Organizowanie wyników

- Pomaga użytkownikom ocenić wyniki i podjąć właściwe decyzje
- Wyróżnia się systemy kategorii (***category systems***) i klastering
- **Systemy kategorii:** istotne etykiety zorganizowane tak aby odwzorowywały ważne tematy w danej dziedzinie
 - powinny być możliwie spójne i kompletne,
 - ich struktura ma być przewidywalna i zgodna w całym zbiorze odpowiedzi
- Struktury kategorii: płaska, hierarchiczna lub fasetowa (***faceted***)
 - struktura płaska – lista tematów, dokumentów
 - hierarchiczna – stosowana do treści książek lub małych zbiorów



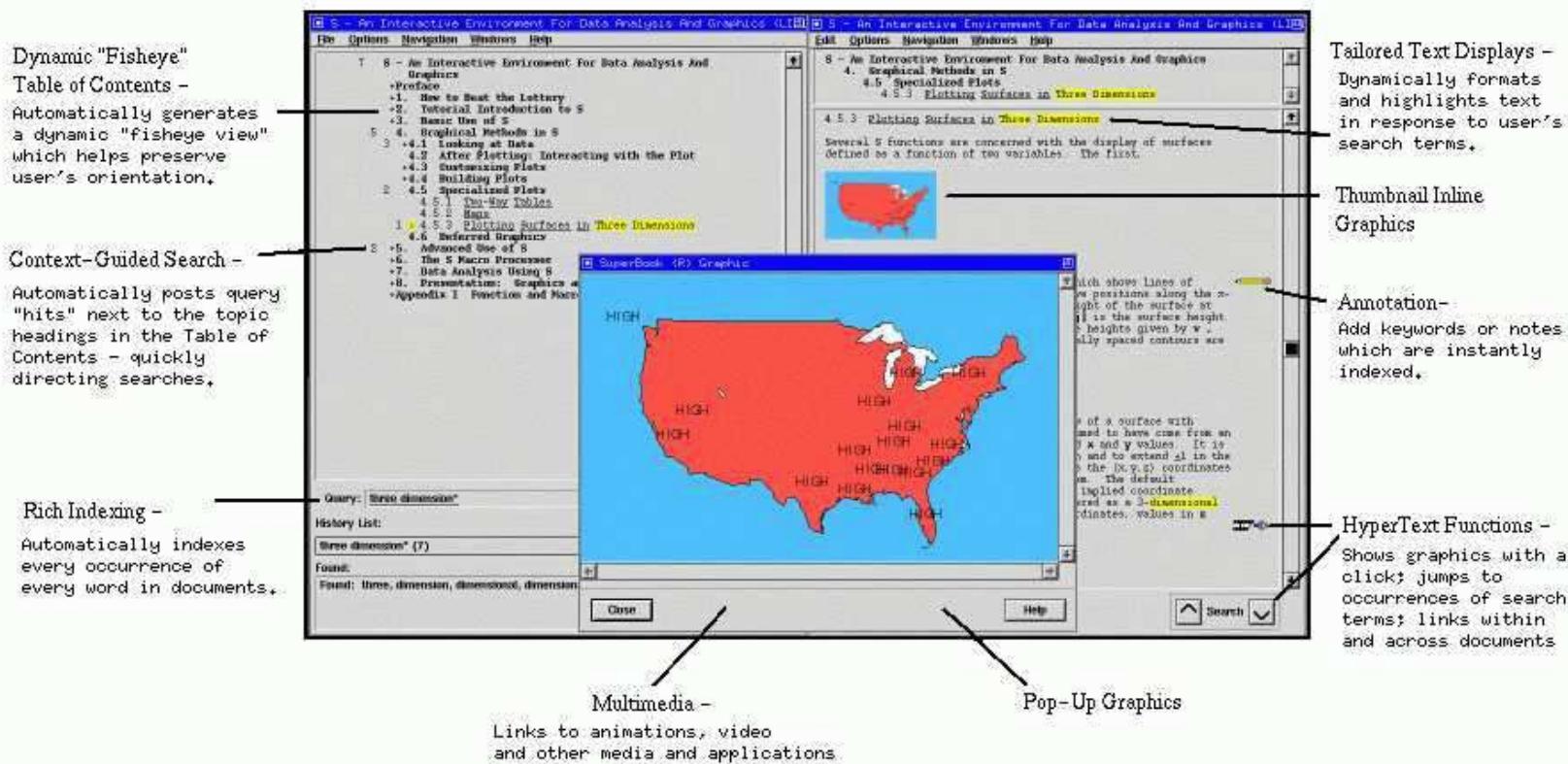
Organizowanie wyników

- Na zasadzie hierarchicznej działa m. in. system *Superbook*.
 - W tym systemie wyniki wyszukiwania są prezentowane w kontekście spisu treści
-

Organizowanie wyników

■ Interfejs Superbook

The SuperBook Document Browser Features





Organizowanie wyników

- **Reprezentacja fasetowa** metadanych pozwala na przypisanie wielu kategorii do pojedynczej pozycji wyników
- Każda z tych kategorii odpowiada innemu fasetowi (wymiarowi, typowi cechy) w zbiorze wyników

- **Klastering** to grupowanie pozycji wyników według pewnej miary podobieństwa
 - Grupuje razem dokumenty podobne do siebie i różne od pozostałych – np. dokumenty w języku japońskim w zbiorze publikacji głównie angielskich
- Klastering jest w pełni automatyczny, ale może dać wyniki grupowania niezgodne z intuicją użytkownika

Organizowanie wyników

■ Przykład organizacji fasetowej

Flamenco Fine Arts Search
Images from the Collections of the Fine Arts Museums of San Francisco:
Legion of Honor and de Young Museums, <http://www.thinker.org>

Powered by Flamenco
Save Search History and Settings Return to Search New Search Logout

These terms define your current search. Click the to remove a term.

keyword "castle"
LOCATION: Europe
MEDIA: Print

Refine your search within these categories:

MEDIA: all > Print

aquatint (4)	lithograph (21)
drypoint (10)	mezzotint (14)
engraving (60)	woodcut (12)
etching (77)	

LOCATION: all > Europe (group results)

Austria (1)	Italy (14)
Belgium / Flanders (5)	Scotland (5)
Bohemia (8)	Spain (1)
France (27)	Switzerland (2)
Germany (19)	more...
Holland (24)	

OBJECTS (group results)

Clothing (68)	Musical Instruments (4)
Containers (21)	Vehicles (56)
Food and Meals (45)	Weapons (27)
Fuel (2)	Writing Tools (13)
Lighting (2)	

BUILT_PLACES (group results)

Bridge (18)	Dwelling (197)
Building (56)	Part of Building (44)
Built Open Space (14)	Road (21)

ANIMALS AND PLANTS (group results)

Birds (19)	Mammals, Hoofed (43)
Creatures and Beasts (1)	Mammals, Other (39)
Fish and Molluscs (6)	Parts of Plants (4)
Flowers (5)	Trees (33)

197 items, grouped by MEDIA ([view ungrouped items](#))

aquatint (4)

Caernarfon Castle, Wales, 18th - 19th century	Dunstanburgh Castle, Northumbria, 1808	Edinburgh Castle, Scotland, 1801	Untitled (landscape), circa 1780

drypoint (10)

Lindesfarne Castle, Northumbria, 19th - 20th century	Stirling Castle, Scotland, 19th - 20th century	Castle Moyle, Northern Ireland, 19th - 20th century	Landscape with a castle, 19th - 20th century

Organizowanie wyników

■ Wyniki klasteringu Findex

The screenshot shows a Microsoft Internet Explorer window with the title "Findex search: jaguar - Microsoft Internet Explorer". The address bar contains "http://". The search query "jaguar" is entered in the search bar. The results page displays a sidebar titled "Categories" with links like "All results", "jaguar cars", "august", "club", "jaguar panthera onca", "mac jaguar", "atari jaguar", "apple", "largest", "cats", "information", "released", "reviews", "powerful", "virtual", and "endangered". The main content area shows several search results, each with a blue link, a snippet of text, and a count in parentheses. The results are as follows:

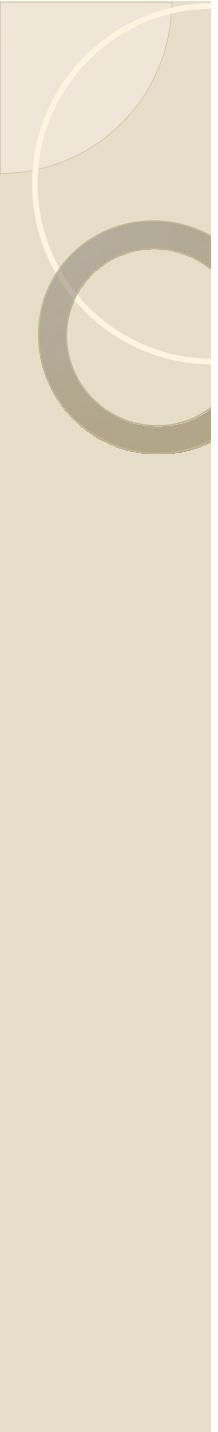
- Atari Jaguar FAQ (41)
Atari Jaguar FAQ. Atari Archives. ... Q. What was the Atari Jaguar/Jaguar64? ... (41)
<http://www.digiserve.com/eescape/showpage.phtml?page=a2>
- Jaguar Interactive II -- The Premier 24-Hour Atari Jaguar ... (59)
... 09:45 26/Jun/04, **Jaguar** Collector, ... 18:07 25/Jun/04, **Jaguar** Collector, ... (59)
<http://www.atarihq.com/interactive/>
- Atari Jaguar VLM (66)
Atari Jaguar VLM. Mucho thanks to Joe Britt for the pix and modification details. Atari's Virtual Light Machine (VLM), was developed ... (66)
http://www.audiovisualizers.com/toolshak/vidsynth/jag_vlm/jag_vlm.htm
- AtariAge - Atari Jaguar History (95)
... However, after the Summer CES that year, Atari announced that the Panther was cancelled so that they could concentrate on a new machine, the 64-bit **Jaguar**. ... (95)
<http://www.atariage.com/Jaguar/?SystemID=JAGUAR>
- Slashdot | New Atari Jaguar Game Running \$1,225 on eBay (100)
... New Atari Jaguar Game Running \$1,225 on eBay. Games. ... Bill Kendrick writes, "The long-awaited **Atari Jaguar** game Battle Sphere has finally been released. ... (100)
<http://slashdot.org/articles/00/03/02/1430232.shtml>
- Slashdot | New Atari Jaguar Game Running \$1,225 on eBay (101)
... New Atari Jaguar Game Running \$1,225 on eBay. Games. ... Bill Kendrick writes, "The long-awaited **Atari Jaguar** game Battle Sphere has finally been released. ... (101)
<http://slashdot.org/articles/00/03/02/1430232.shtml>

Organizowanie wyników

- Wynik klasterowania zapytania „senate” w Clusty.com (Yippy.com)

The screenshot shows a search results page from Clusty.com. The search bar at the top contains the query "senate". Below the search bar, a message states "Cluster Senate Committee contains 29 documents." The results are listed in a numbered format, each with a link, a small icon, and a brief description.

Rank	Result Title	Description
1.	U.S. Senate	Official site of "the living symbol of our union of states." Connect with Senators , and learn about Senate committees , legislation, records, art, history, schedules, news, tours ... www.senate.gov - [cache] - Live, Open Directory, Ask
2.	U.S. Senate Committee on Commerce, Science, & Transportation	Committee jurisdiction includes the Coast Guard, coastal management, communications, highway safety, waterways, interstate commerce, maritime commerce, fisheries, merchant marine ... commerce.senate.gov - [cache] - Live, Ask
3.	United States Senate Committee on Banking, Housing and Urban Affairs	United States Senate Committee on Banking, Housing and Urban Affairs banking.senate.gov - [cache] - Live
4.	Senate of the Kingdom of Cambodia	Information about legislative activities, laws, committees , senators and an historical timeline from 1998. www.senate.gov.kh - [cache] - Open Directory, Ask
5.	Kansas Senate	Senate Roster, ... Home > Senate ... Senate Committees www.kslegislature.org/legisv-senate/index.do - [cache] - Ask
6.	U.S. Senate Committee on Energy and Natural Resources	Has jurisdiction over energy policy, regulation, and research. Also deals with energy and mineral conservation, ports used for energy transport, irrigation, reclamation, mining ... energy.senate.gov - [cache] - Live



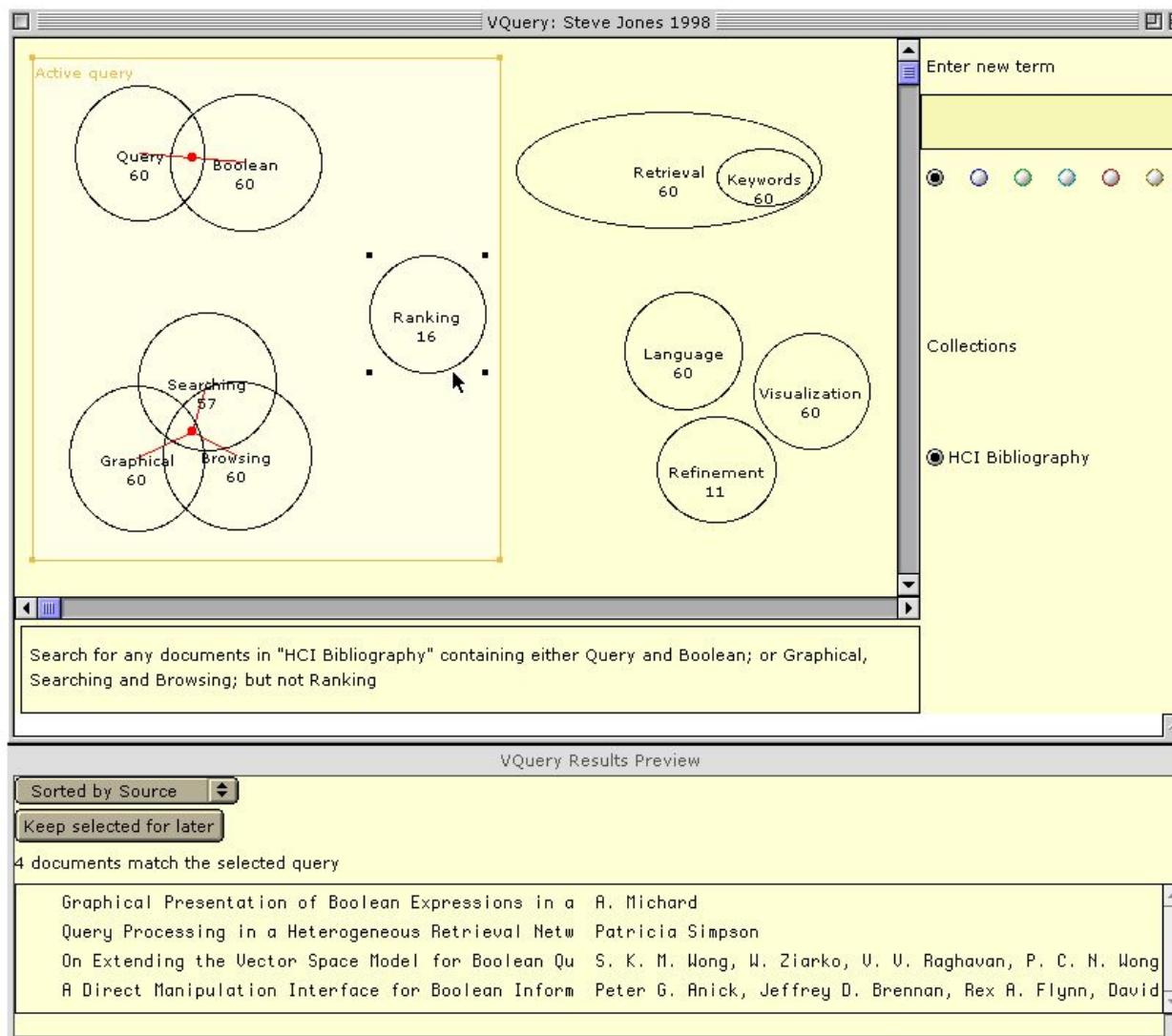
Wizualizacja w interfejsach wyszukiwania

- Wizualizacja wyników:
 - składni boolowskiej (rzadko używane),
 - termów zapytania w zbiorze wyników,
 - zależności między słowami i dokumentami
 - wizualizacja dla eksploracji tekstów

- Składnię boolowską przedstawia się przy pomocy tzw. diagramów Venna wykorzystywanych np. przez system **VQuery**.

Wizualizacja w interfejsach wyszukiwania

■ Interfejs VQuery





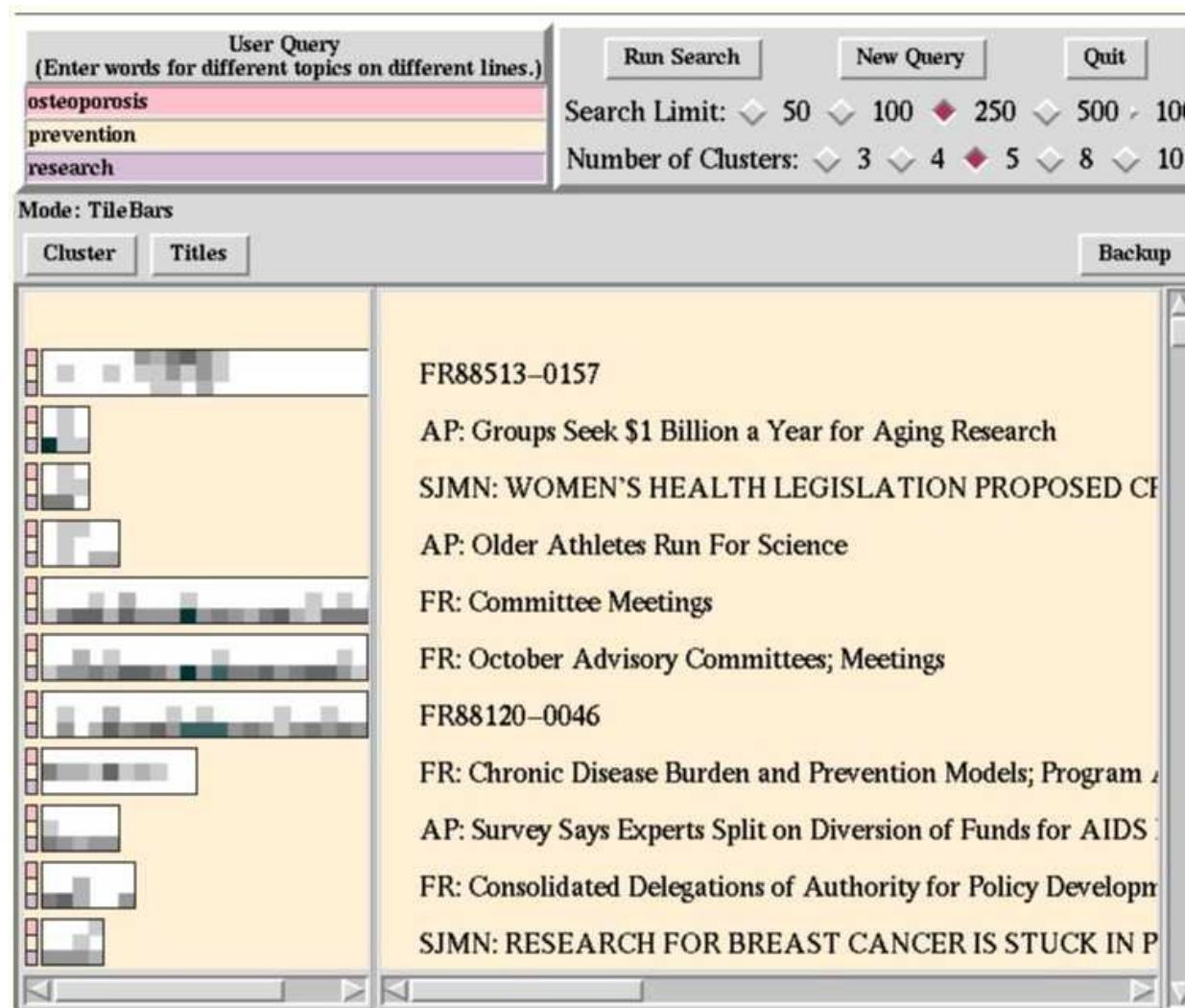
Wizualizacja termów zapytania

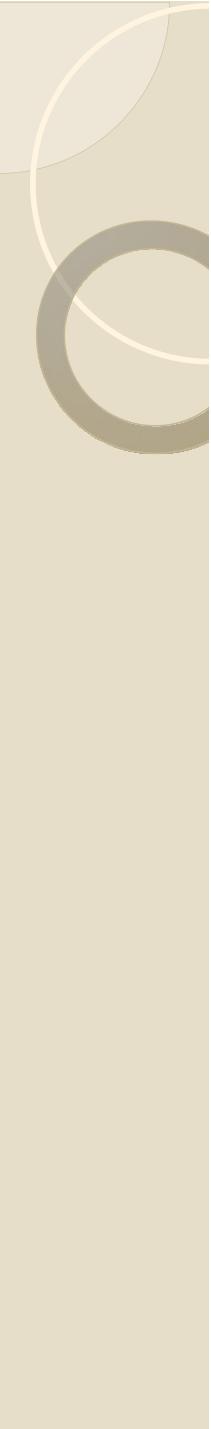
- W interfejsie TitleBar dokumenty są prezentowane jako poziome glify (*glyphs*)
- Położenia termów zapytania są zaznaczone wewnętrz nich
- Użytkownik jest zachęcany do rozbicia zapytania na różne fasety z jednym wariantem w linii
- Wtedy poziome wiersze graficznej reprezentacji dokumentów (*glyphs*) przedstawiają częstotliwość pojawiania się termów w ramach poszczególnych tematów



Wizualizacja termów zapytania

■ Interfejs *TitleBars*





Wizualizacja termów zapytania

- Inne podejścia to umieszczenie termów na wykresach słupkowych (bar charts), punktowych (scatter plots) i w tablicach
 - Uwypuklanie termów w miniaturach dokumentów
-

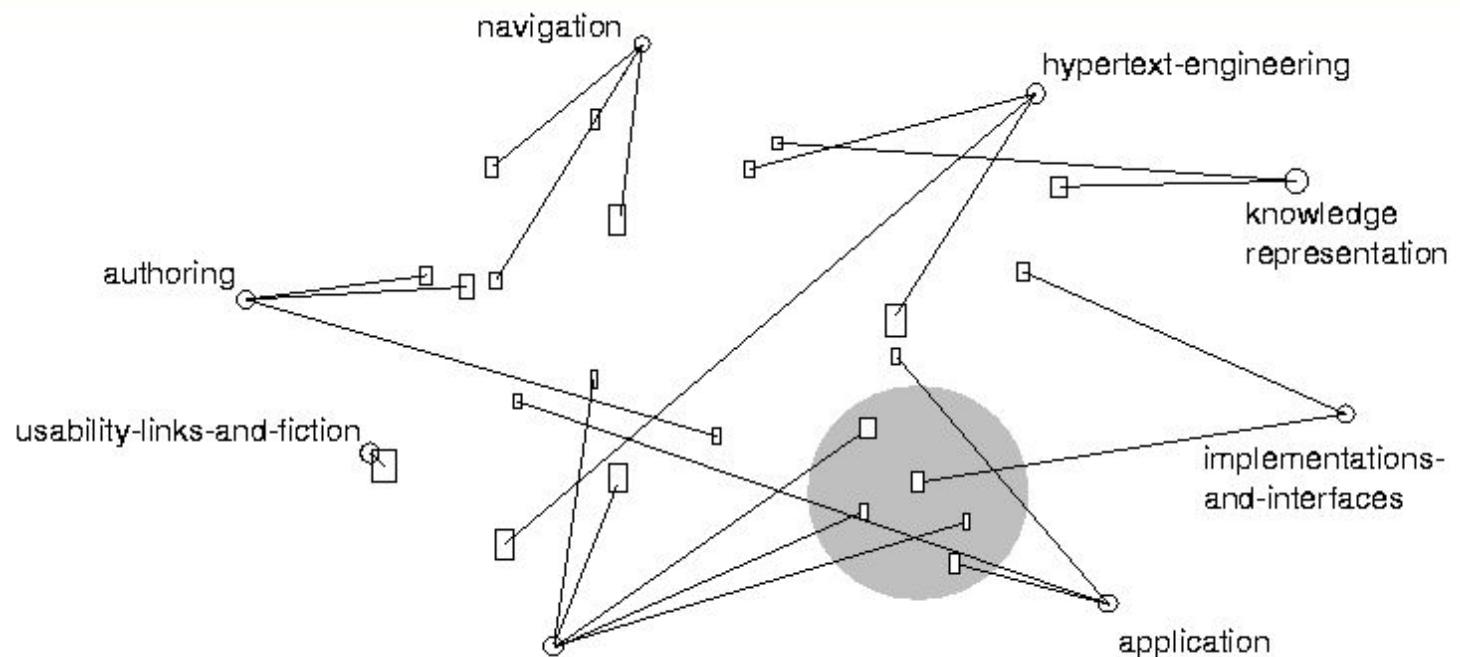


Powiązania termów i dokumentów

- W wielu pracach na temat interfejsów proponuje się połączenie słów (termów) i dokumentów na płótnie dwuwymiarowym
- Interfejs **VIBE** – bliskość glifów reprezentuje powiązania semantyczne termów i dokumentów; dokumenty z kombinacjami termów są umieszczone w pół drogi pomiędzy ikonami tych termów
- Projekty Aduna Autofocus i Lyberworld rozszerzają VIBE do 3 wymiarów

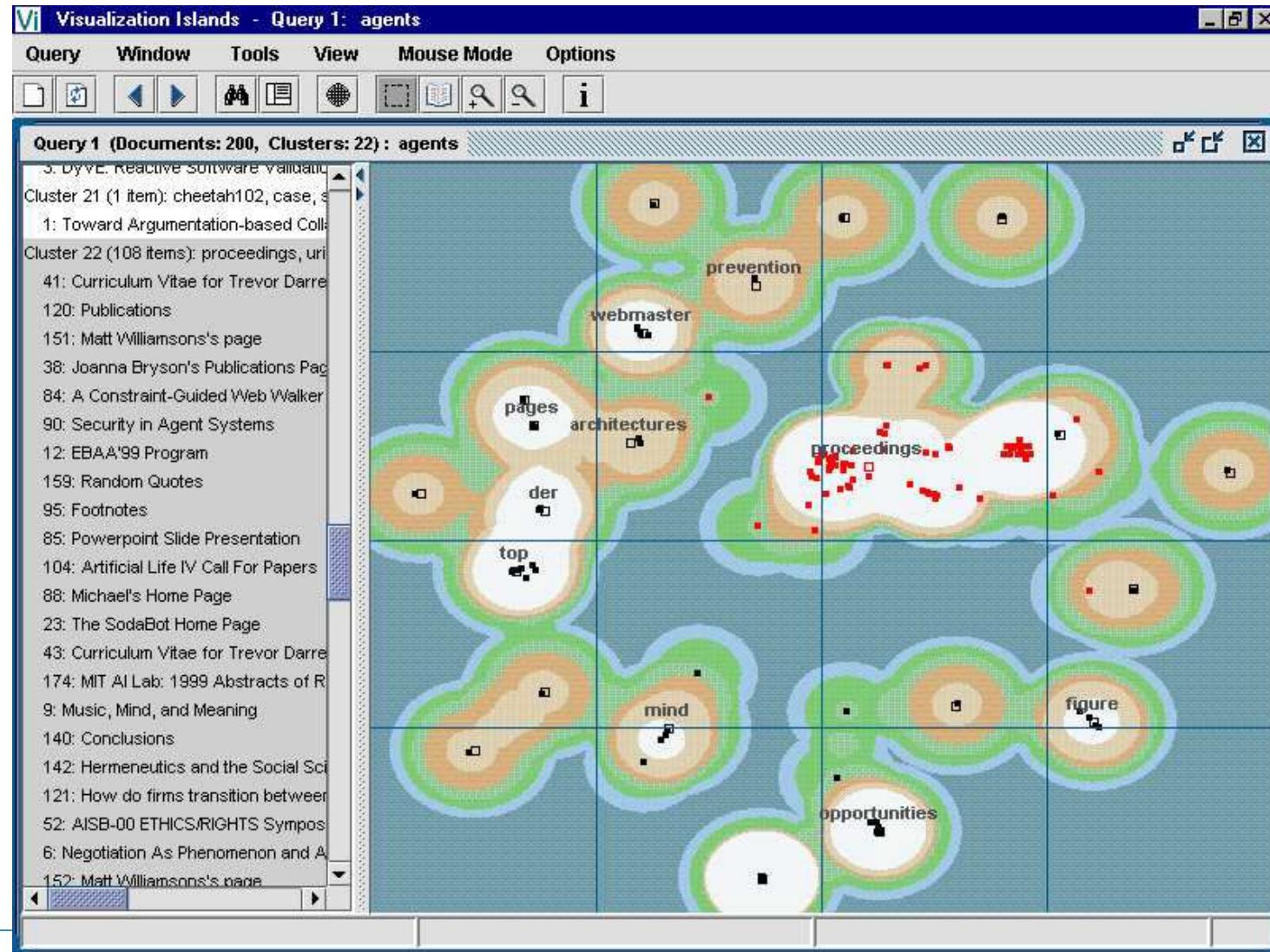
Powiązania termów i dokumentów

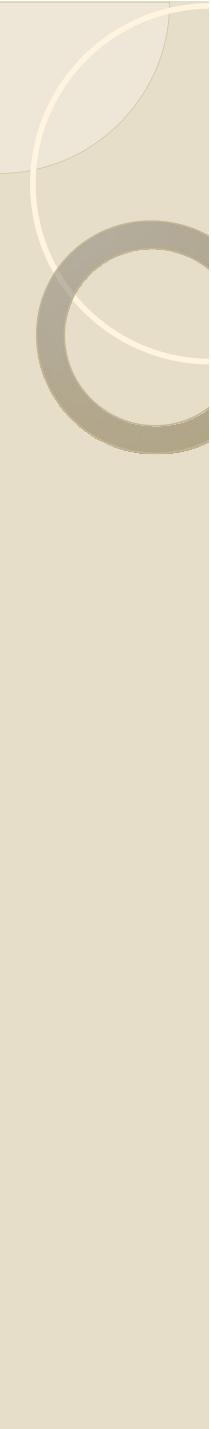
■ Okno VIBE



Powiązania termów i dokumentów

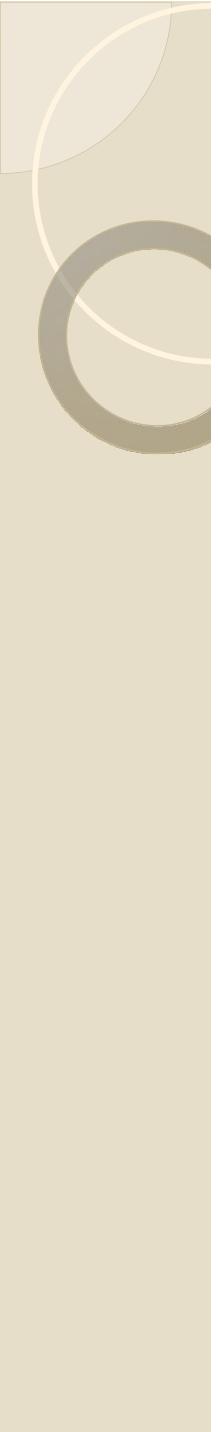
■ Wyświetlanie typu starfield (xFIND VisIslands)





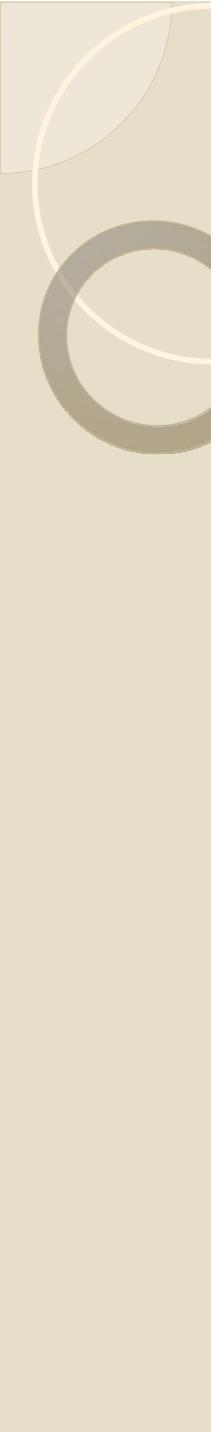
Modelowanie i ranking

- Wykłady opracowano w oparciu o książkę Ricardo Baeza-Yates, Berthier Ribeiro-Neto, „*Modern Information Retrieval, the concepts and technology behind search*” 2nd edition, ACM Press Books, 2011
- Z tego samego źródła zaczerpnięto także różne zadania i przykłady wykorzystywane w treści wykładu.



Modelowanie i ranking

- Modelowanie w wyszukiwaniu informacji ma na celu określenie sposobu budowania funkcji rankingu
 - **Funkcja rankingu:** przypisuje określoną ocenę ilościową do dokumentu w odniesieniu do danego zapytania
 - Modelowanie obejmuje dwa główne zadania:
 - Ustalenie założeń logicznych dla reprezentacji dokumentów i zapytań
 - Definicję funkcji rankingu pozwalającej na ilościowe wyrażenie podobieństwa zapytań i dokumentów zwracanych w odpowiedzi
 - Do indeksowania i wyszukiwania dokumentów służą termy indeksujące
-

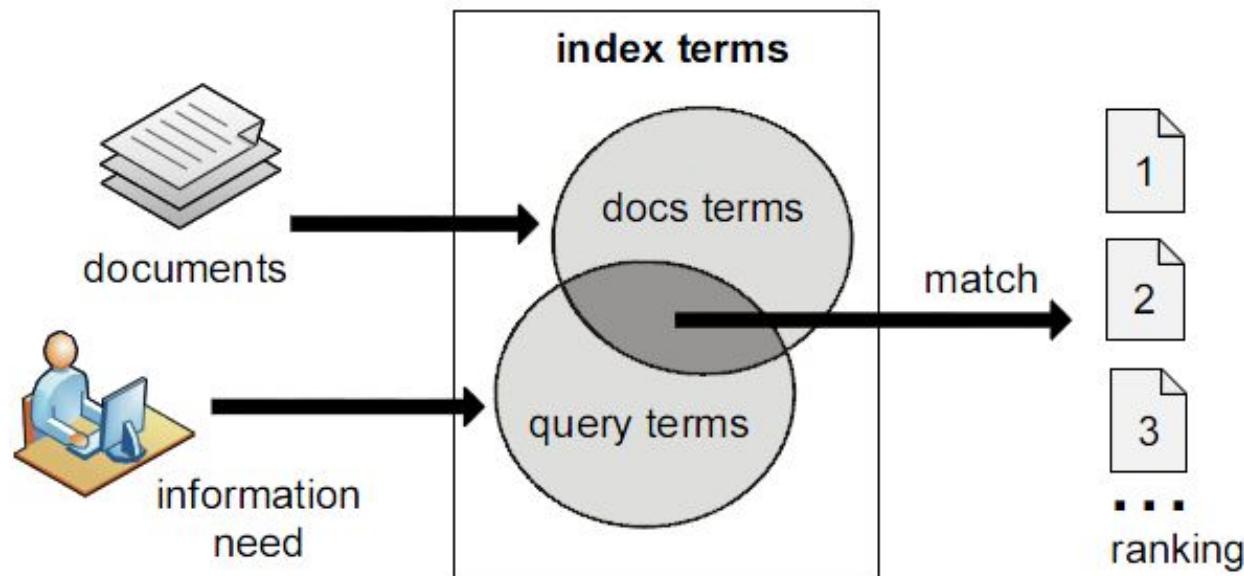


Model IR

- Term indeksujący:
 - W sensie ścisłym – jest to słowo kluczowe o pewnym znaczeniu, zazwyczaj rzeczownik
 - W sensie ogólnym – dowolne słowo pojawiające się w dokumencie
 - Wyszukiwanie danych opiera się na termach indeksujących
-

Modelowanie i ranking

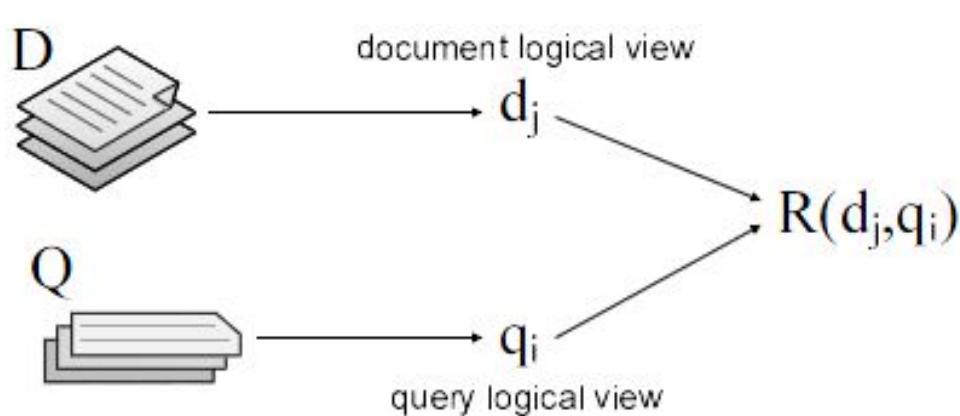
- Proces wyszukiwania informacji



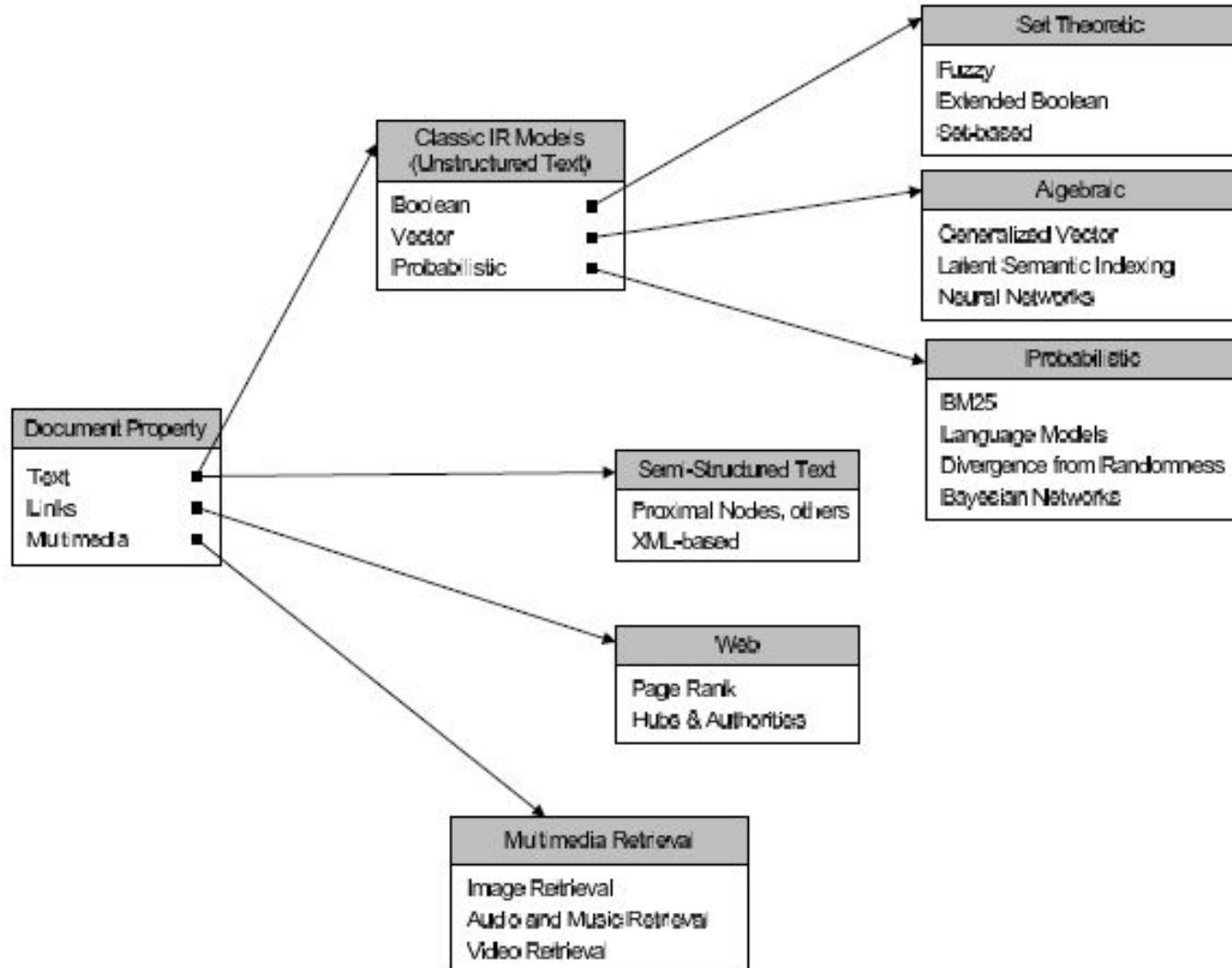
- **Ranking** to uporządkowanie dokumentów odzwierciedlające ich odpowiedniość względem zapytania
- System IR przewiduje które dokumenty są istotne dla użytkownika, z pewnym stopniem niepewności

Model IR

- Model IR jest czwórką $[D, Q, F, R(q_i, d_j)]$ gdzie:
 - D – zbiór logicznych perspektyw dokumentów
 - Q – zbiór logicznych perspektyw zapytań
 - F – założenia modelowania dokumentów i zapytań

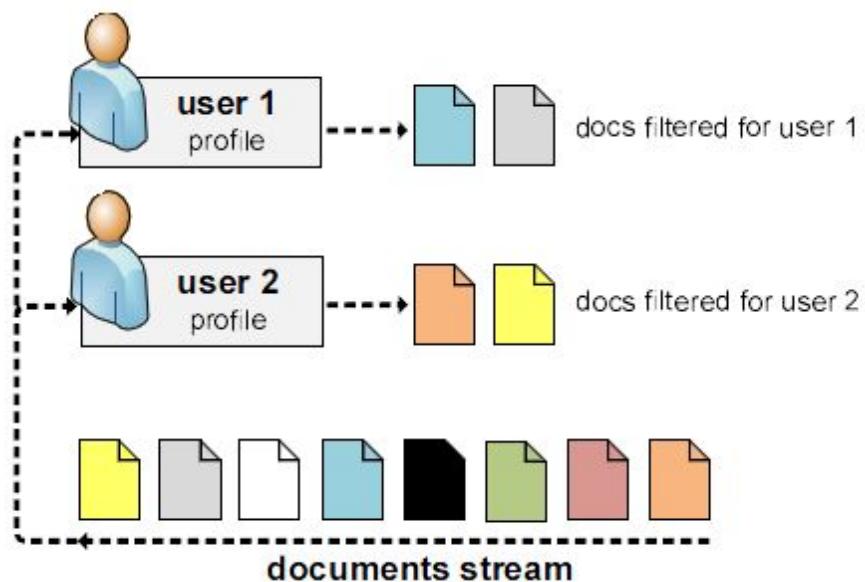
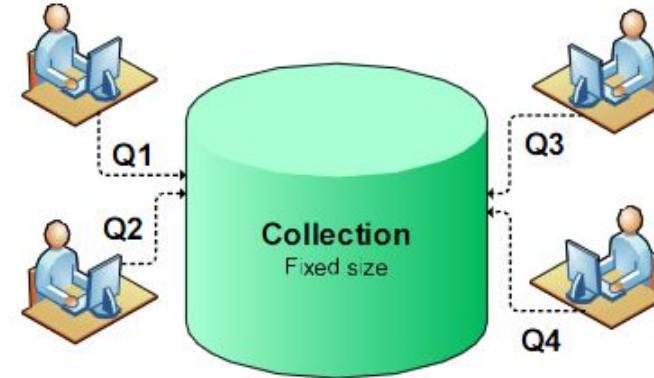


Klasyfikacja modeli



Wyszukiwanie ad-hoc i filtracja

- Każdy dokument jest reprezentowany przez termy indeksujące – słowa lub ciągi słów
- Wybrany zbiór termów reprezentuje dokument



Pojęcia podstawowe

- Słownik $V = \{k_1, \dots, k_t\}$ - zbiór różnych termów indeksujących

■ t - liczba termów w kolekcji dokumentów

■ k_i – elementarny term indeksujący

- $V = \boxed{k_1 \ k_2 \ k_3 \ \dots \ k_t}$
 $\boxed{1 \ 0 \ 0 \ \dots \ 0}$ pattern that represents documents (and queries) with the term k_1 and no other
 \vdots
 $\boxed{1 \ 1 \ 1 \ \dots \ 1}$ pattern that represents documents (and queries) with all index terms

- Każdy z powyższych wzorców współwystępowania termów jest **koniunktywnym komponentem termów**

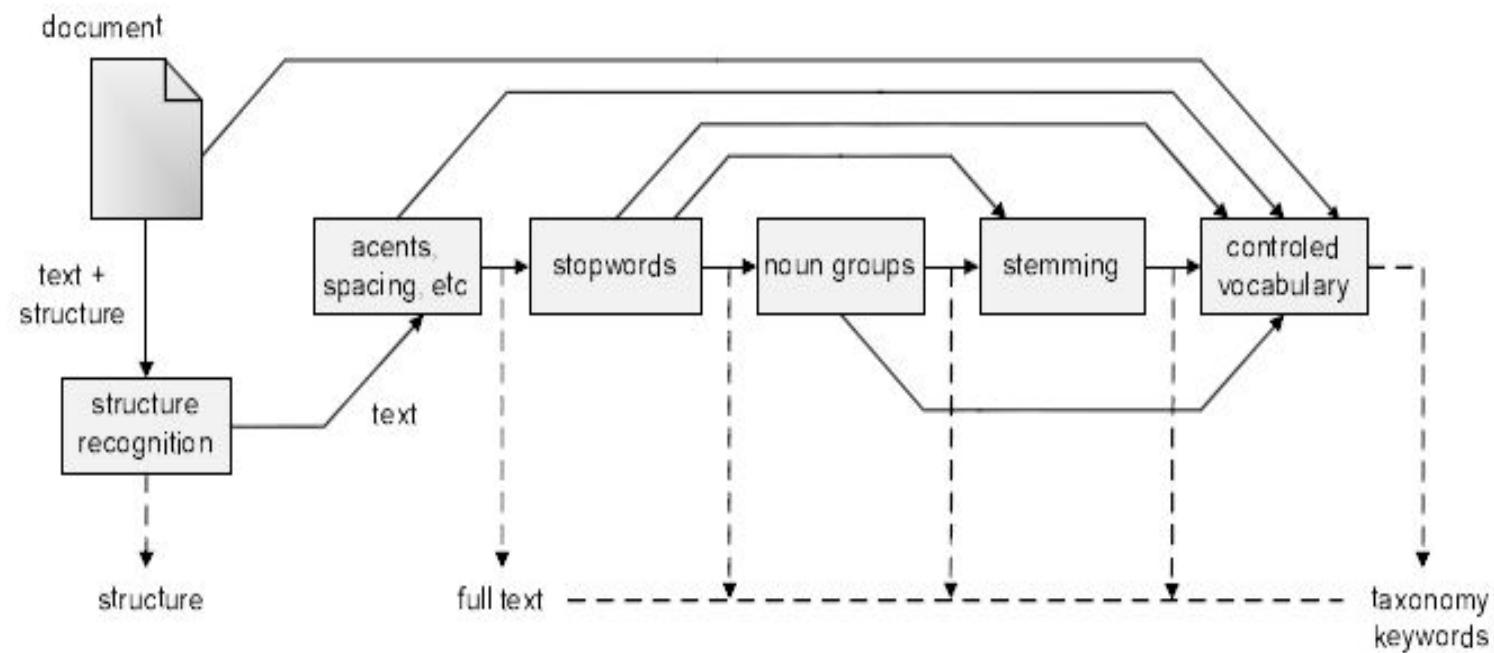
Pojęcia podstawowe

- Z każdym dokumentem d_j (lub pytaniem q) kojarzy się jednoznaczny komponent koniunktywny termu $c(d_j)$ (lub zapytania $c(q)$)
- Pojawienie się termu k_i w dokumencie d_j ustala relację między nimi, która jest opisana jako częstotliwość $f_{i,j}$ termu k_i w dokumencie d_j

$$\begin{array}{c} d_1 \quad d_2 \\ \begin{matrix} k_1 & \left[\begin{matrix} f_{1,1} & f_{1,2} \\ f_{2,1} & f_{2,2} \\ f_{3,1} & f_{3,2} \end{matrix} \right] \\ k_2 \\ k_3 \end{matrix} \end{array}$$

Pojęcia podstawowe

■ Struktura logiczna dokumentu





Model boolowski

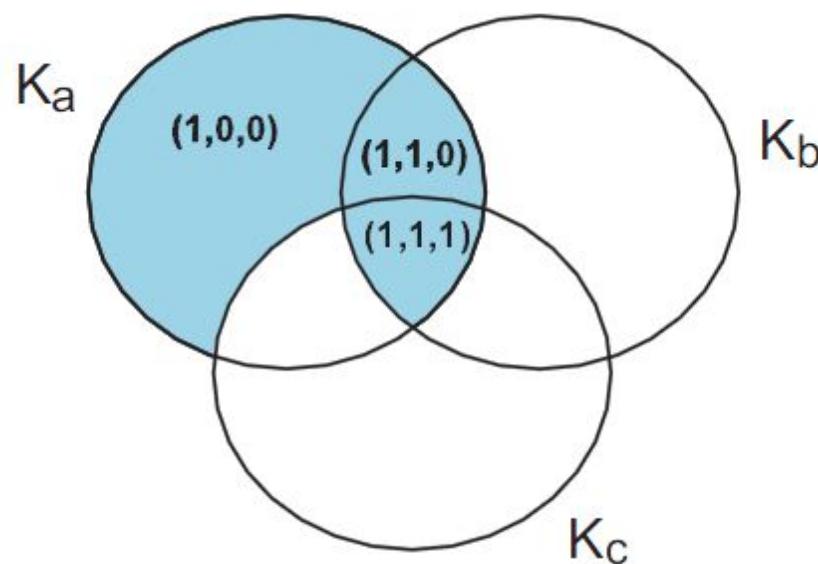
- Zapytania są specyfikowane jako wyrażenia boolowskie
 - intuicyjna i dokładna semantyka
 - zachowanie formalizmu
 - przykładowe zapytanie:

$$q = k_a \wedge (k_b \vee \neg k_c), \quad V = \{k_a, k_b, k_c\},$$

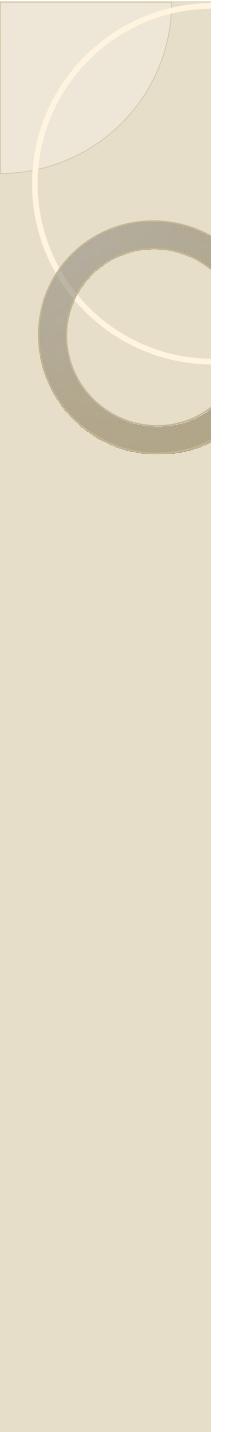
gdzie V oznacza słownik

- Stabilicowane zależności term-dokument są binarne
 - $w_{ij} \in \{0, 1\}$: waga związana z parą (k_i, d_j)
 - $w_{iq} \in \{0, 1\}$: waga związana z parą (k_i, q)
 - Koniunktywny komponent termów spełniający zapytanie q nazywa się koniunktywnym komponentem zapytania $c(q)$
-

Model boolowski



- Zapytanie q można zapisać w postaci dysjunktywnej
 $q_{DNF} = (1,1,1) \cup (1,1,0) \cup (1,0,0)$
- Jeżeli słownik $V=\{k_a, k_b, k_c, k_d\}$ a dokument d_j zawiera 3 pierwsze termy $c(d_j) = (1,1,1,0)$ to zapytanie q można także przedstawić w formie dysjunktywnej



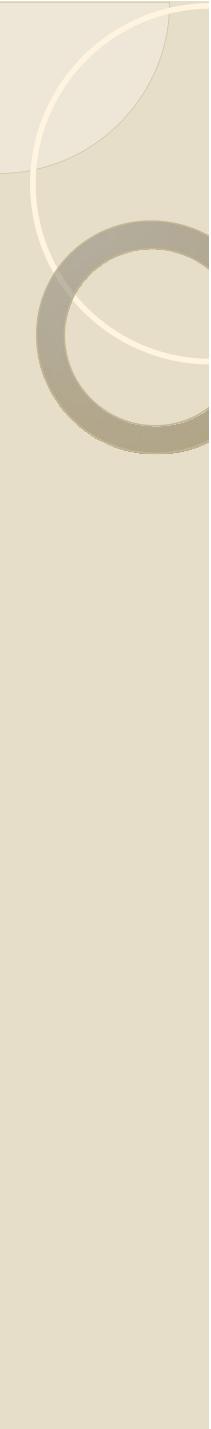
Model boolowski

$$\begin{aligned} q_{DNF} = & (1, 1, 1, 0) \vee (1, 1, 1, 1) \vee \\ & (1, 1, 0, 0) \vee (1, 1, 0, 1) \vee \\ & (1, 0, 0, 0) \vee (1, 0, 0, 1) \end{aligned}$$

- Podobieństwo dokumentu d_j do zapytania q można zdefiniować jako:

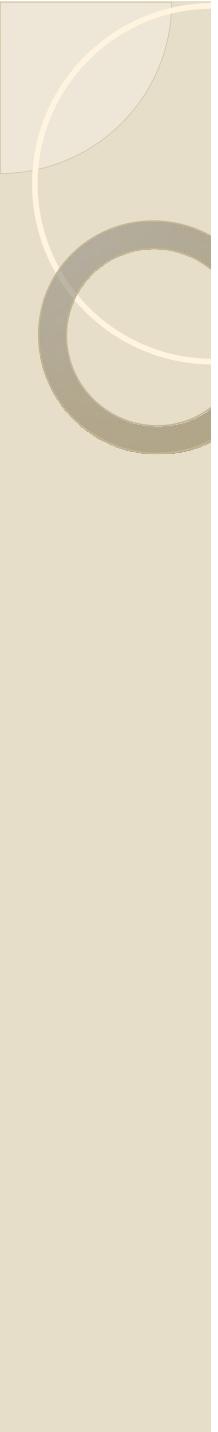
$$sim(d_j, q) = \begin{cases} 1 & \text{if } \exists c(q) | c(q) = c(d_j) \\ 0 & \text{otherwise} \end{cases}$$

- Model boolowski określa czy każdy dokument jest istotny lub nie dla zapytania, bez ustalania poziomu tej istotności



Model boolowski

- Dokumenty nie podlegają rankingowi
 - Zapytanie musi być tłumaczone na wyrażenie boolowskie
 - W praktyce model często uwzględnia za dużo lub za mało dokumentów
-



Wagi termów

- Termy w dokumentach nie są jednakowo użyteczne w rozróżnianiu ich treści
 - Np. słowo występujące we wszystkich dokumentach jest bezużyteczne dla wyszukiwania
- Z termem k_i w dokumecie d_j kojarzy się wagę $w_{i,j} > 0$ ($=0$) gdy term pojawia się (nie pojawia) w tym dokumencie
- Waga $w_{i,j}$ wyraża ważność termu k_i dla opisu zawartości dokumentu d_j ; jest istotna dla wyznaczenia rankingu dokumentu

Wagi TF-IDF

- Do ustalenia wag termów w dokumentach stosuje się formułę TF-IDF (term frequency – inverse document frequency)
- **Odwrotna częstotliwość dokumentu dla termu k_i**

$$idf_i = \log \frac{N}{n_i},$$

gdzie n_i – ilość dokumentów z termem k_i , N – wielkość kolekcji dokumentów

- **Częstotliwość termu k_i dla całej kolekcji dokumentów**

$$tf_i = 1 + \log \sum_{j=1}^N f_{i,j},$$

log –logarytm o podstawie 2

Wagi TF-IDF

■ wagi tf-idf

$$w_{i,j} = \begin{cases} (1 + \log f_{i,j}) \times \log \frac{N}{n_i} & \text{if } f_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases},$$

- To do is to be.
To be is to do.
- d_1
- To be or not to be.
I am what I am.
- d_2
- I think therefore I am.
Do be do be do.
- d_3
- Do do do, da da da.
Let it be, let it be.
- d_4

		d_1	d_2	d_3	d_4
1	to	3	2	-	-
2	do	0.830	-	1.073	1.073
3	is	4	-	-	-
4	be	-	-	-	-
5	or	-	2	-	-
6	not	-	2	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	2	-	-
10	think	-	-	2	-
11	therefore	-	-	2	-
12	da	-	-	-	5.170
13	let	-	-	-	4
14	it	-	-	-	4

Warianty TF-IDF

weighting scheme	document term weight	query term weight
1	$f_{i,j} * \log \frac{N}{n_i}$	$(0.5 + 0.5 \frac{f_{i,q}}{\max_i f_{i,q}}) * \log \frac{N}{n_i}$
2	$1 + \log f_{i,j}$	$\log(1 + \frac{N}{n_i})$
3	$(1 + \log f_{i,j}) * \log \frac{N}{n_i}$	$(1 + \log f_{i,q}) * \log \frac{N}{n_i}$

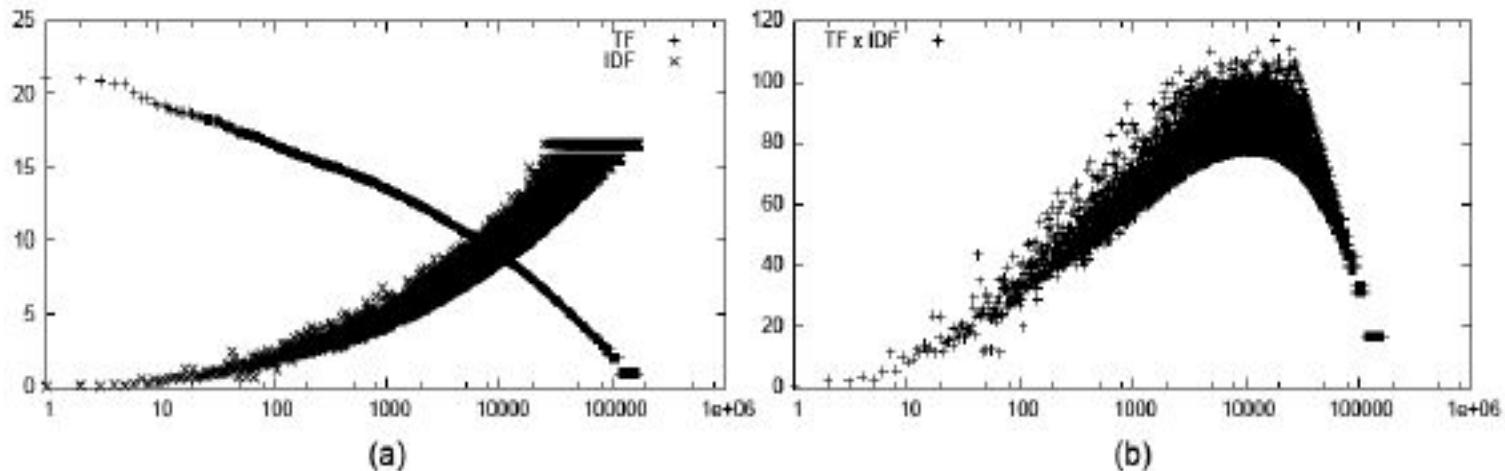
- Jeśli n_i jest częstotliwością dokumentu dla termu k_i , $n(r)$ – częstotliwością rzędu r w porządku malejącym częstotliwości to r jest rangą termu k_i oraz

$$n(r) = Nr^{-\alpha},$$

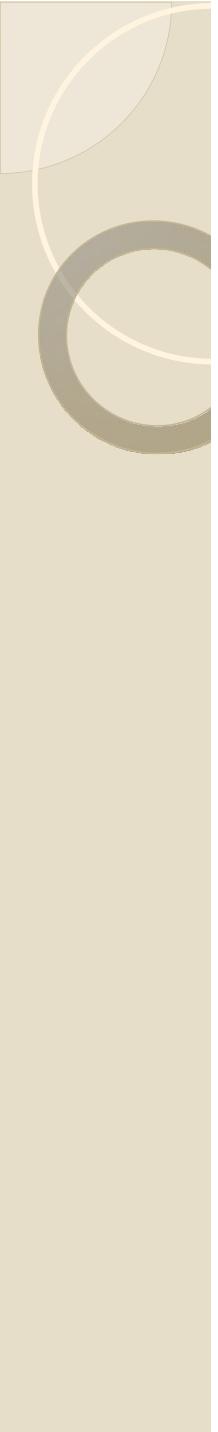
dla pewnej empirycznej stałej α (np. $\alpha = 1$).

Warianty TF-IDF

- Przykładowe wykresy $tf\text{-}idf$ (tf dla kolekcji dokumentów) względem rangi termów



- Termy dla pośrednich wartości idf oraz tf dają maksimum wag $tf\text{-}idf$ i są najbardziej odpowiednie dla rankingu

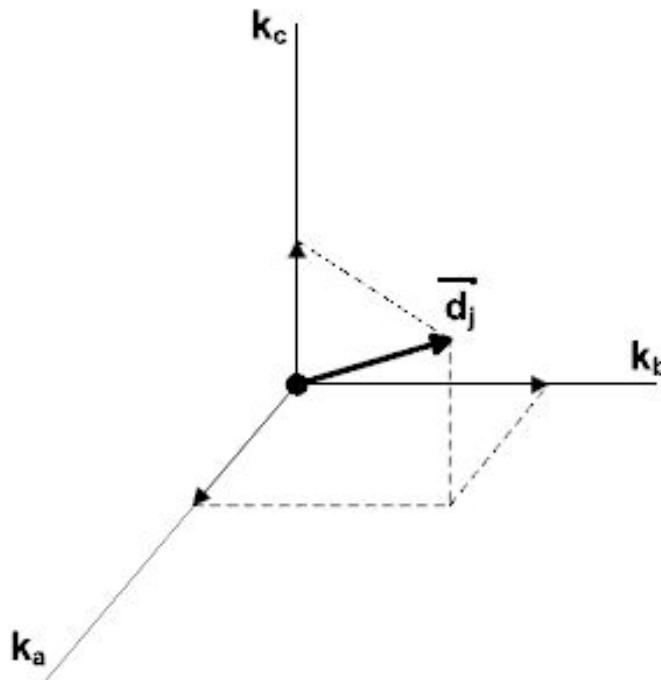


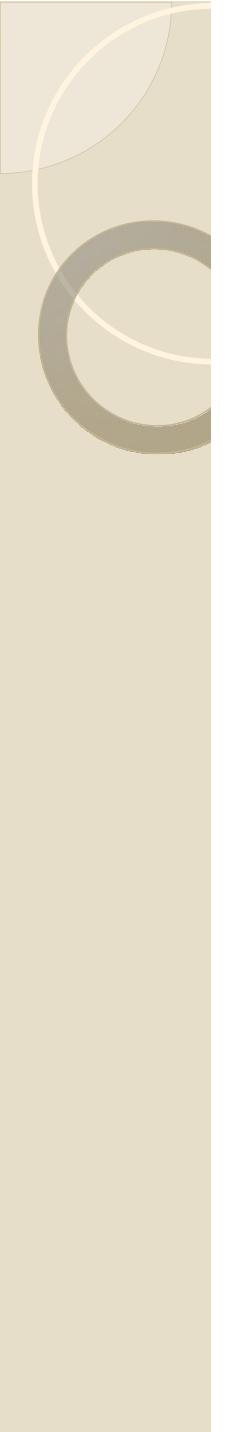
Normalizacja długości dokumentu

- Dokumenty posiadają różne długości; dłuższe z nich, a niekoniecznie bardziej istotne, mają większą szansę być wyszukane w danym zapytaniu
 - Aby usunąć ten niepożądany efekt dzieli się rangę każdego dokumentu przez jego długość; ten proces nazywamy **normalizacją długości dokumentu**
 - Metody normalizacji:
 - **Rozmiar w bajtach:** każdy dokument traktuje się jako strumień bajtów,
 - **Liczba słów:** dokument jest traktowany jako pojedynczy łańcuch słów do zliczenia
 - **Normy wektorowe:** dokumenty są reprezentowane jako wektory termów ważonych
-

Normalizacja długości dokumentu

- Każdy term w zbiorze dokumentów jest związany z wersorem \mathbf{k}_i w przestrzeni t-wymiarowej
- Term k_i w dokumencie d_j jest skojarzony ze składową $w_{i,j} \times k_i$ wektora tego dokumentu





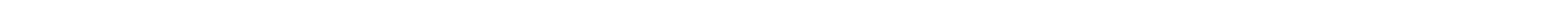
Normalizacja długości dokumentu

- wektor dokumentu:

$$\mathbf{d}_j = (w_{1,j}, w_{2,j} , \dots, w_{t,j}),$$

- długość dokumentu:

$$|\mathbf{d}_j| = \sqrt{\sum_i^t w_{i,j}^2},$$



Normalizacja długości dokumentu

- Trzy warianty długości dokumentu w przykładowej kolekcji:

To do is to be.
To be is to do.

d_1

To be or not to be.
I am what I am.

d_2

I think therefore I am.
Do be do be do.

d_3

Do do do, da da da.
Let it be, let it be.

d_4

	d_1	d_2	d_3	d_4
size in bytes	34	37	41	43
number of words	10	11	10	12
vector norm	5.068	4.899	3.762	7.738



Model wektorowy

- Model wektorowy uwzględnia nie tylko wystąpienie dopasowania, ale także stopień dopasowania dokumentów do zapytania poprzez wagi poszczególnych termów
- Dokumenty są uszeregowane w porządku malejącym stopnia zgodności z zapytaniem
- W modelu wektorowym:
 - Wagi $w_{i,j}$ związane z parami (k_i, d_j) są dodatnie i niebinarne
 - Termy indeksujące występują niezależnie od siebie (z założenia)
 - Termy są reprezentowane przez wersory w przestrzeni t-wymiarowej.

Model wektorowy

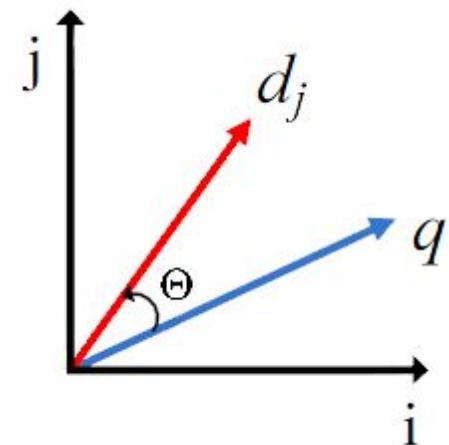
- W modelu wektorowym: reprezentacje dokumentu d_j oraz zapytania q są wektorami:

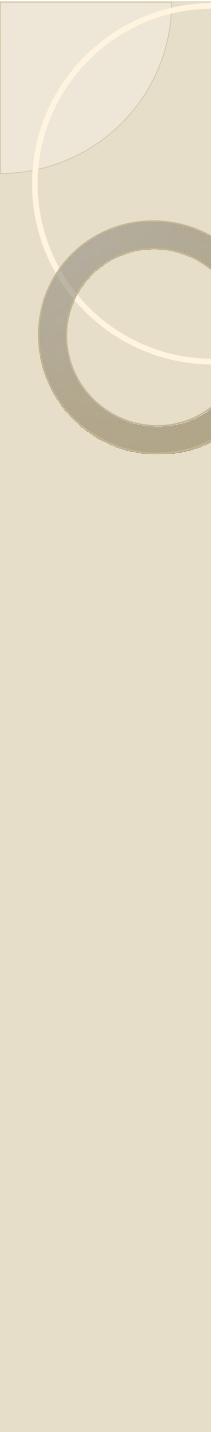
$$d_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j}), \quad q = (w_{1,q}, w_{2,q}, \dots, w_{t,q}),$$

- Podobieństwo dokumentu d_j oraz zapytania q :

$$\cos(\theta) = \frac{\mathbf{d}_j \bullet \mathbf{q}}{|\mathbf{d}_j| \times |\mathbf{q}|}$$

$$sim(d_j, q) = \frac{\sum_{i=1}^t w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,j}^2} \times \sqrt{\sum_{i=1}^t w_{i,q}^2}}$$





Model wektorowy

- Wagi w modelu wektorowym są typu tf-idf

$$w_{i,q} = (1 + \log f_{i,q}) \times \log \frac{N}{n_i},$$

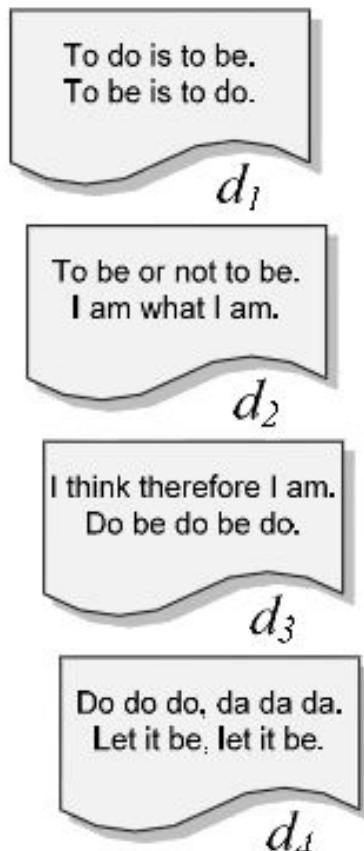
$$w_{i,j} = (1 + \log f_{i,j}) \times \log \frac{N}{n_i},$$

gdzie n_i – ilość dokumentów zawierających term k_i , N – ilość wszystkich dokumentów

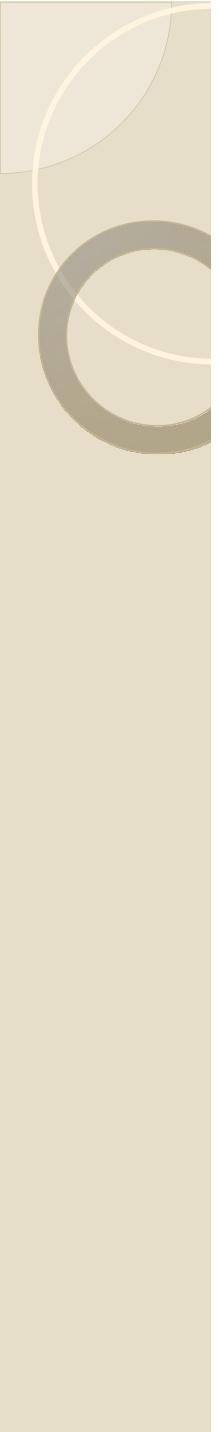
- Te równania stosuje się dla termów o częstotliwości większej od 0
- Jeżeli częstotliwość termu jest zerowa odpowiednie wagi są zerowe

Model wektorowy

- Rangi przykładowych dokumentów dla zapytania „*to do*”, w modelu wektorowym z wagami tf-idf



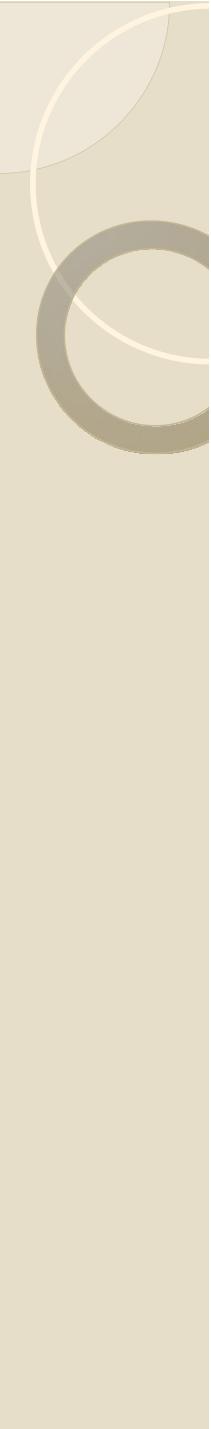
doc	rank computation	rank
d_1	$\frac{1*3+0.415*0.830}{5.068}$	0.660
d_2	$\frac{1*2+0.415*0}{4.899}$	0.408
d_3	$\frac{1*0+0.415*1.073}{3.762}$	0.118
d_4	$\frac{1*0+0.415*1.073}{7.738}$	0.058



Model wektorowy

- Zalety modelu wektorowego:
 - wprowadza wagi termów,
 - częściowe dopasowanie pozwala na przybliżone wyszukiwanie,
 - cosinusowa funkcja rankingu sortuje dokumenty wg. podobieństwa do zapytania,
 - normalizacja długości dokumentu wbudowana w ranking.

- Wady:
 - model zakłada niezależność termów



Model probabilistyczny

- Dla danego zapytania istnieje **idealny zbiór odpowiedzi** na to pytanie,
- Na podstawie tego zbioru ustala się istotne dokumenty
- Zapytanie traktuje się jako specyfikację właściwości tego idealnego zbioru
- Początkowy zbiór dokumentów wybiera się dowolnie,
- Po przejrzeniu przez użytkownika 10-20 pierwszych dokumentów korygowany jest zbiór idealny
- Powtarzanie powyższego procesu doskonali zbiór idealny

Ranking probabilistyczny

- Model probabilistyczny:
 - Estymuje prawdopodobieństwo, że dokument jest istotny dla zapytania użytkownika,
 - Zakłada, że to prawdopodobieństwo zależy jedynie od zapytania i reprezentacji dokumentów,
 - Idealny zbiór odpowiedzi R , maksymalizuje prawdopodobieństwo istotności dokumentu
- Podobieństwo dokumentu do zapytania:

$$\text{sim}(d_j, q) = \frac{P(R | d_j, q)}{P(\bar{R} | d_j, q)},$$

gdzie R – zbiór dokumentów istotnych dla zapytania q ,
– zbiór dokumentów nieistotnych

Ranking

- Ze wzoru Bayes'a:

$$\text{sim}(d_j, q) = \frac{P(\mathbf{d}_j | R, q) \times P(R, q)}{P(\mathbf{d}_j | \bar{R}, q) \times P(\bar{R}, q)} \sim \frac{P(\mathbf{d}_j | R, q)}{P(\mathbf{d}_j | \bar{R}, q)},$$

gdzie:

- $P(R, q)$ – prawdopodobieństwo, że dokument losowo wybrany z kolekcji jest istotny dla $\mathbf{zapytania} q$,
- $P(\mathbf{d}_j | R, q)$ – prawdopodobieństwo wybrania dokumentu d_j ze zbioru R ,

- Niech $p_{iR} = P(k_i | R, q)$, $q_{iR} = P(k_i | \bar{R}, q)$,

Ranking

- Po zlogarytmowaniu wzoru dla sim i przy założeniu

$$\forall k_i \notin q, \quad p_{iR} = q_{iR},$$

- uzyskuje się

$$\text{sim}(d_j, q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{p_{iR}}{1 - p_{iR}}\right) + \log\left(\frac{1 - q_{iR}}{q_{iR}}\right).$$

- Przyjmijmy:

- N – liczba dokumentów w kolekcji,
- R – liczba dokumentów istotnych dla pytania,
- n_i – liczba dokumentów z termem k_i ,
- r_i – liczba dokumentów istotnych z termem k_i .

Tablica kontyngencji

- Na podstawie powyższych zmiennych można sformułować tablicę kontyngencji

	relevant	non-relevant	all docs
docs that contain k_i	r_i	$n_i - r_i$	n_i
docs that do not contain k_i	$R - r_i$	$N - n_i - (R - r_i)$	$N - n_i$
all docs	R	$N - R$	N

- Jeśli informacje z tablicy są znane dla zadanego pytania, można zapisać

$$p_{iR} = \frac{r_i}{R}, \quad q_{iR} = \frac{n_i - r_i}{N - R},$$

Formuła rankingu

- Wówczas równanie dla obliczenia rankingu może być zapisane jako:

$$\text{sim}(d_j, q) \sim \sum_{k_i \in [q, d_j]} \log \left(\frac{r_i}{R - r_i} \times \frac{N - n_i - R + r_i}{n_i - r_i} \right),$$

- Dla małych wartości r_i dodajemy 0.5 do każdego termu we wzorze powyżej , przez co uzyskuje się równanie rankingu Robertsona-Sparcka Jonesa

$$\text{sim}(d_j, q) \sim \sum_{k_i \in [q, d_j]} \log \left(\frac{r_i + 0.5}{R - r_i + 0.5} \times \frac{N - n_i - R + r_i + 0.5}{n_i - r_i + 0.5} \right),$$

- Równanie powyżej wymaga znajomości początkowych przybliżeń r_i oraz R

Estymacja r_i i R

- Jedna możliwość to przyjęcie wartości $R = r_i = 0$,
 - Druga możliwość to estymacja R i r_i przez wstępne wyszukiwanie z 10-20 dokumentów i ponowne wykonanie zapytania dla estymowanych wartości.
-
- Mocny wzór wzajemnego równania logestycznego estymującego p_{iR} i q_{iR} oraz r_i
$$\log \left(\frac{p_{iR}}{1 - p_{iR}} \right) = \log \left(\frac{1 - q_{iR}}{q_{iR}} \right) + \sum_{k_i \in q \wedge k_i \in d_j} \log \left(\frac{n_i}{N - n_i} \right)$$
-
- $p_{iR} = 0.5$, $q_{iR} = n_i / N$ gdzie n_i to liczba dokumentów z termem k_i
 - powyższe przybliżenie umożliwia wyliczenie początkowego rankingu

Poprawa rankingu początkowego

- Ponowne przeliczenie estymat; dodajemy 0.5 aby uniknąć problemów przy $D_i = 1$ oraz $D_i = 0$:

$$p_{iR} = \frac{D_i + 0.5}{D + 1}, \quad q_{iR} = \frac{n_i - D_i + 0.5}{N - D + 1},$$

- gdzie
 - D – zbiór dokumentów wstępnie wyszukanych,
 - D_i – zbiór dokumentów zawierających term k_i .
- Powyższy proces można powtórzyć wielokrotnie



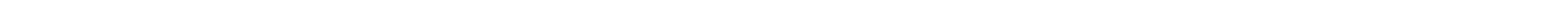
Plusy i minusy

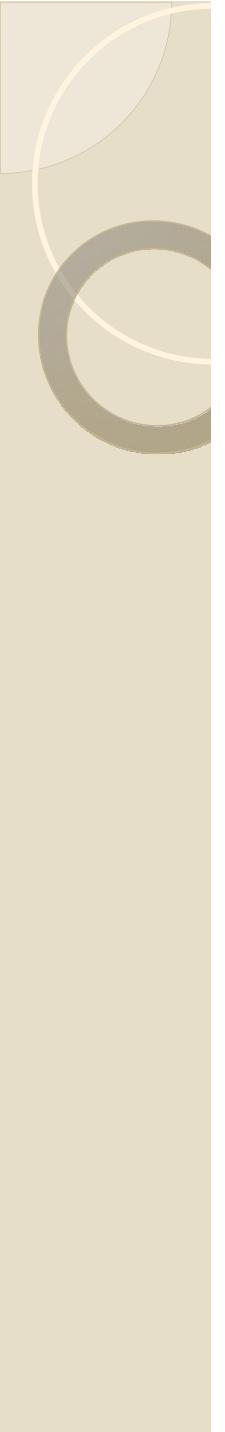
- Zalety:

- Dokumenty są szeregowane wg. malejącej istotności

- Wady:

- trzeba założyć początkową wartość p_{iR} ,
 - brak normalizacji długości dokumentu,
 - metoda nie stosuje wag tf





Uogólniony model wektorowy

- Modele klasyczne zakładają wzajemną niezależność termów indeksujących
- W modelu wektorowym przyjmuje się
$$\forall_{i,j} \Rightarrow \mathbf{k}_i \bullet \mathbf{k}_j = 0,$$
- W modelu uogólnionym wektory wektory termów indeksujących nie muszą być ortogonalne

- Założenia:
 - $w_{i,j}$ stanowi wagę binarną skojarzoną z $[k_i, d_j]$,
 - $V=\{k_1, k_2, \dots, k_t\}$ zbiór wszystkich termów.

Uogólniony model wektorowy

$$\begin{aligned} & (k_1, k_2, k_3, \dots, k_t) \\ m_1 &= (0, 0, 0, \dots, 0) \\ m_2 &= (1, 0, 0, \dots, 0) \\ m_3 &= (0, 1, 0, \dots, 0) \\ m_4 &= (1, 1, 0, \dots, 0) \\ & \vdots \\ m_{2^t} &= (1, 1, 1, \dots, 1) \end{aligned}$$

Minterm to term składający się z literałów połączonych logicznym symbolem koniunkcji, który dla dokładnie jednej kombinacji wejść danej funkcji przyjmuje wartość 1.

- Dla każdego dokumentu d_j istnieje minterm $m_r = c(d_j)$ zawierający tylko termy z tego dokumentu i żadnego innego
- Takie mintermy budują ortogonalne wersory \mathbf{m}_r przestrzeni 2^t wymiarowej

Uogólniony model wektorowy

- Ortogonalność wektorów \mathbf{m}_r nie oznacza niezależności termów indeksujących k_i , które są skorelowane w ramach wektorów \mathbf{m}_r .

$$on(i, m_r) = \begin{cases} 1 & \text{gdy } k_i \in m_r \\ 0 & \text{gdy } k_i \notin m_r \end{cases}$$

- Wektor związany z termem k_i oblicza się jako:

$$\mathbf{k}_i = \frac{\sum_{\forall r} on(i_m, r) c_{i,r} \mathbf{m}_r}{\sqrt{\sum_{\forall r} on(i_m, r) c_{i,r}^2}}, \quad c_{i,r} = \sum_{d_j | c(d_j) = m_r} w_{i,j}$$

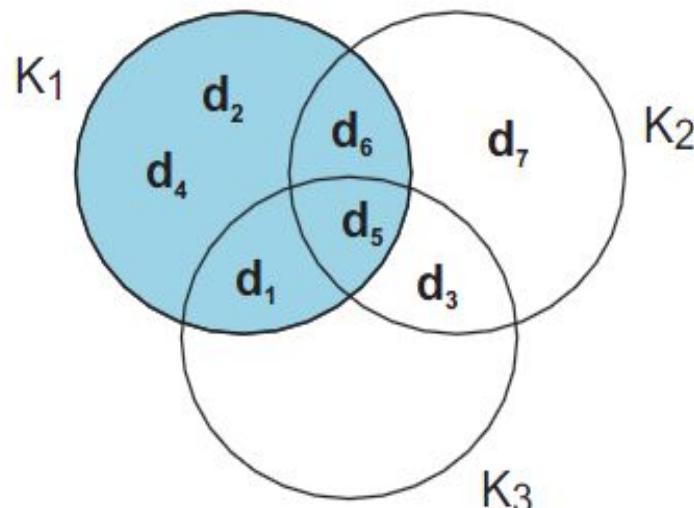
- Dla kolekcji N dokumentów tylko N mintermów (nie 2^t) uczestniczy w rankingu

Uogólniony model wektorowy

- Stopień korelacji termów k_i i k_j oblicza się jako

$$\mathbf{k}_i \bullet \mathbf{k}_j = \sum_{\forall r} on(i, m_r) \times c_{i,r} \times on(j, m_r) \times c_{j,r}$$

- Przykład:



	K_1	K_2	K_3
d_1	2	0	1
d_2	1	0	0
d_3	0	1	3
d_4	2	0	0
d_5	1	2	4
d_6	1	2	0
d_7	0	5	0
q	1	2	3

Uogólniony model wektorowy

■ Obliczenie $c_{i,r}$

	K_1	K_2	K_3
d_1	2	0	1
d_2	1	0	0
d_3	0	1	3
d_4	2	0	0
d_5	1	2	4
d_6	0	2	2
d_7	0	5	0
q	1	2	3

	K_1	K_2	K_3
$d_1 = m_6$	1	0	1
$d_2 = m_2$	1	0	0
$d_3 = m_7$	0	1	1
$d_4 = m_2$	1	0	0
$d_5 = m_8$	1	1	1
$d_6 = m_7$	0	1	1
$d_7 = m_3$	0	1	0
$q = m_8$	1	1	1

	$c_{1,r}$	$c_{2,r}$	$c_{3,r}$
m_1	0	0	0
m_2	3	0	0
m_3	0	5	0
m_4	0	0	0
m_5	0	0	0
m_6	2	0	1
m_7	0	3	5
m_8	1	2	4

Uogólniony model wektorowy

- Obliczenie \mathbf{k}_i $i=1,2,3$

- $\overrightarrow{k_1} = \frac{(3\vec{m}_2 + 2\vec{m}_6 + \vec{m}_8)}{\sqrt{3^2 + 2^2 + 1^2}}$

- $\overrightarrow{k_2} = \frac{(5\vec{m}_3 + 3\vec{m}_7 + 2\vec{m}_8)}{\sqrt{5^2 + 3^2 + 2^2}}$

- $\overrightarrow{k_3} = \frac{(1\vec{m}_6 + 5\vec{m}_7 + 4\vec{m}_8)}{\sqrt{1^2 + 5^2 + 4^2}}$

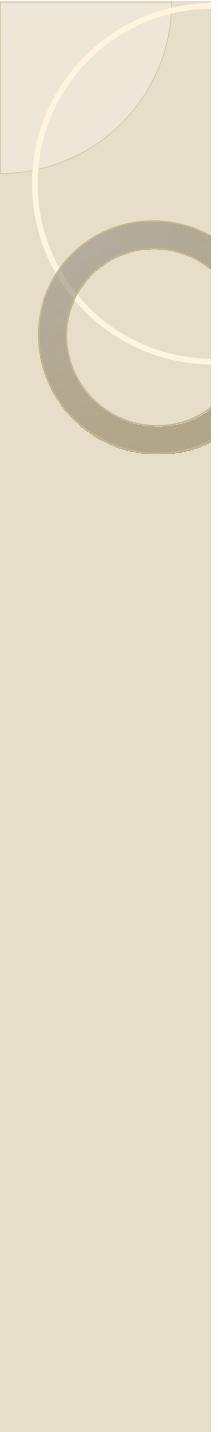
	$c_{1,r}$	$c_{2,r}$	$c_{3,r}$
m_1	0	0	0
m_2	3	0	0
m_3	0	5	0
m_4	0	0	0
m_5	0	0	0
m_6	2	0	1
m_7	0	3	5
m_8	1	2	4

Uogólniony model wektorowy

■ Obliczenie wektorów dokumentów

- $\vec{d}_1 = 2\vec{k}_1 + \vec{k}_3$
- $\vec{d}_2 = \vec{k}_1$
- $\vec{d}_3 = \vec{k}_2 + 3\vec{k}_3$
- $\vec{d}_4 = 2\vec{k}_1$
- $\vec{d}_5 = \vec{k}_1 + 2\vec{k}_2 + 4\vec{k}_3$
- $\vec{d}_6 = 2\vec{k}_2 + 2\vec{k}_3$
- $\vec{d}_7 = 5\vec{k}_2$
- $\vec{q} = \vec{k}_1 + 2\vec{k}_2 + 3\vec{k}_3$

	K_1	K_2	K_3
d_1	2	0	1
d_2	1	0	0
d_3	0	1	3
d_4	2	0	0
d_5	1	2	4
d_6	0	2	2
d_7	0	5	0
q	1	2	3



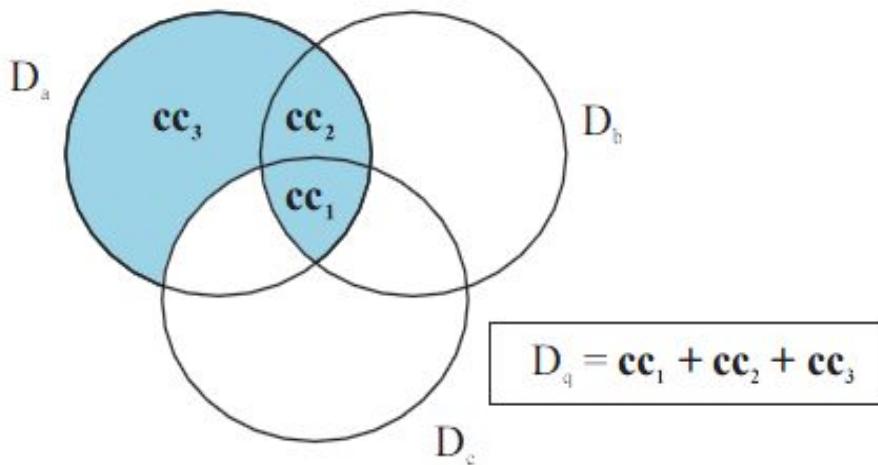
Model rozmyty

- Dopasowanie dokumentu do termów zapytania ma charakter przybliżony
 - Każdy term zapytania k_i definiuje zbiór rozmyty skojarzonych z nim dokumentów,
 - Każdy dokument posiada określony stopień przynależności $\mu_i \in [0,1]$ do takiego zbioru

 - Taka interpretacja jest podstawą wszystkich rozmytych modeli IR
-

Model rozmyty Ogawa, Morita, Kobayashi

- zapytanie boolowskie dla zbioru rozmytego dokumentów $q = k_a \wedge (k_b \vee \neg k_c)$



- D_a, D_b, D_c – zbiory rozmyte dokumentów odpowiednio dla termów k_a, k_b, k_c
- $cc_i, i=1,2,3$ – komponenty koniunktywne
- D_q –rozmyty zbiór zapytania



Model rozmyty

- Dla zwykłych zbiorów D_i dysjunktywna postać normalna zapytania składa się z 3 koniunktywnych komponentów cc

$$\begin{aligned} q_{dnf} &= (1, 1, 1) + (1, 1, 0) + (1, 0, 0) \\ &= cc_1 + cc_2 + cc_3 \end{aligned}$$

- W tym modelu przyjmuje się funkcję przynależności do iloczynu zbiorów rozmytych jako iloczyn a nie minimum z tych przynależności

$$\mu_{A \cap B}(u) = \mu_A(u)\mu_B(u)$$

- Niech $\mu_{a,j}, \mu_{b,j}, \mu_{c,j}$ oznaczają stopnie przynależności dokumentu d_j do zbiorów rozmytych D_a, D_b, D_c .



Model rozmyty

- Wówczas:

$$cc_1 = \mu_{a,j} \mu_{b,j} \mu_{c,j}$$

$$cc_2 = \mu_{a,j} \mu_{b,j} (1 - \mu_{c,j})$$

$$cc_3 = \mu_{a,j} (1 - \mu_{b,j}) (1 - \mu_{c,j})$$

- Przynależność dokumentu d_j do zapytania q :

$$\mu_{q,j} = \mu_{cc_1+cc_2+cc_3,j}$$

$$= 1 - \prod_{i=1}^3 (1 - \mu_{cc_i,j})$$

$$= 1 - (1 - \mu_{a,j} \mu_{b,j} \mu_{c,j}) \times \\ (1 - \mu_{a,j} \mu_{b,j} (1 - \mu_{c,j})) \times (1 - \mu_{a,j} (1 - \mu_{b,j}) (1 - \mu_{c,j}))$$



Model rozmyty

- Aby powiązać dokument d_j z termem k_i poprzez zbiór rozmyty buduje się słownik jako macierz korelacji C typu term-term

$$c_{i,l} = \frac{n_{i,l}}{n_i + n_l - n_{i,l}},$$

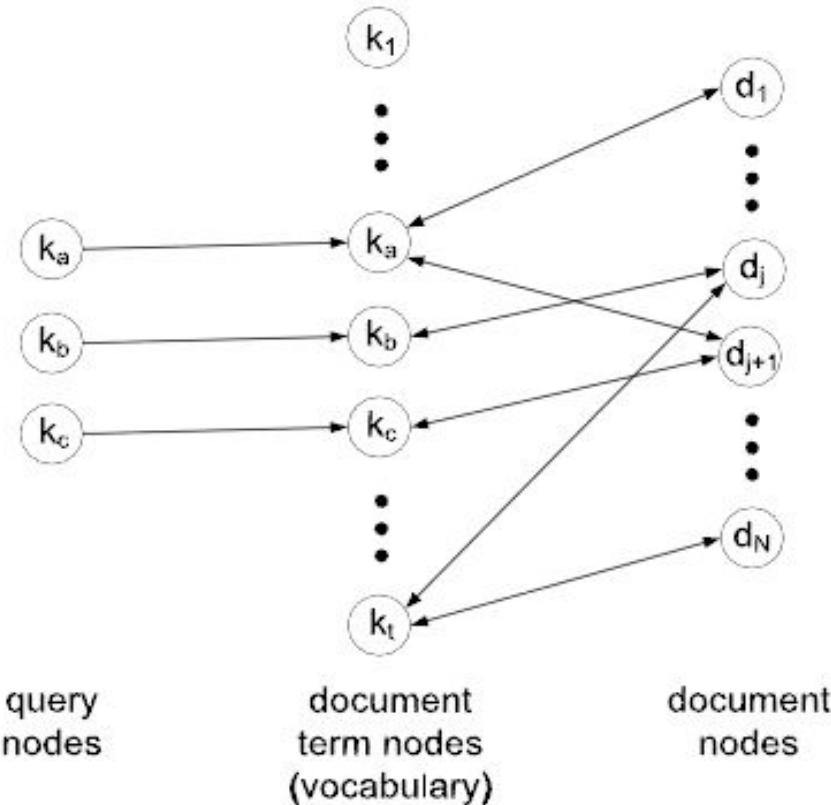
gdzie n_i – liczba dokumentów zawierających k_i , n_l – liczba dokumentów zawierających k_l , $n_{i,l}$ – liczba dokumentów zawierających zarówno k_i jak k_l

- W zbiorze rozmytym dokumentów z termem k_i dokument d_j posiada stopień uczestnictwa $\mu_{i,j}$ określony przez korelację k_i i innych termów indeksujących w dokumencie d_j

$$\mu_{i,j} = 1 - \prod_{k_l \in d_j} (1 - c_{i,l})$$

Model neuronowy

- Model sieci neuronowej dla wyszukiwania informacji



- Węzły dokumentów i ich termów mają wbudowane progi aktywacji



Model neuronowy

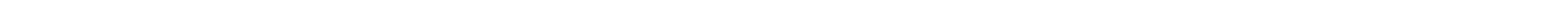
- siła sygnałów propagowanych między węzłami sieci jest wyrażona poprzez wagi połączeń synaptycznych
- Termy zapytań emitują sygnały jednostkowe
- Wagi powiązań między węzłami termów zapytań k_q i termów dokumentowych $w_{i,q} = \frac{k_i \cdot w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,q}^2}}$,
- Wagi powiązań węzła termu dokumentowego k_i z węzłem dokumentu $w_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_{i=1}^t w_{i,j}^2}}$,

Model neuronowy

- Poziom aktywacji węzła dokumentowego d_j odpowiadający modelowi wektorowemu

$$\sum_{i=1}^t \frac{w_{i,q}}{w_{i,j}} = \frac{\sum_{i=1}^t w_{i,q} w_{i,j}}{\sqrt{\sum_{i=1}^t w_{i,q}^2} \times \sqrt{\sum_{i=1}^t w_{i,j}^2}},$$

- Nowe sygnały mogą być wymieniane między węzłami dokumentów i termów dokumentowych w procesie uczenia się sieci, przy ustawieniu określonego minimalnego progu aktywacji



Model BM25 (Best Match 25)

- BM25 powstał w wyniku serii eksperymentów nad modelami probabilistycznymi
- Do wyznaczania wag termów wykorzystuje on:
 - odwrotną częstotliwość dokumentu,
 - częstotliwość termów,
 - normalizację długości dokumentu.
- Klasyczny model probabilistyczny BM1 uwzględnia tylko pierwszą z wymienionych pozycji
- Formuła BM1 rankingu dokumentu gdy brak informacji o istotności:

$$sim(d_j, q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log \frac{N - n_i + 0.5}{n_i + 0.5},$$

Model BM25 (Best Match 25)

- Równanie rankingu dla modelu BM25

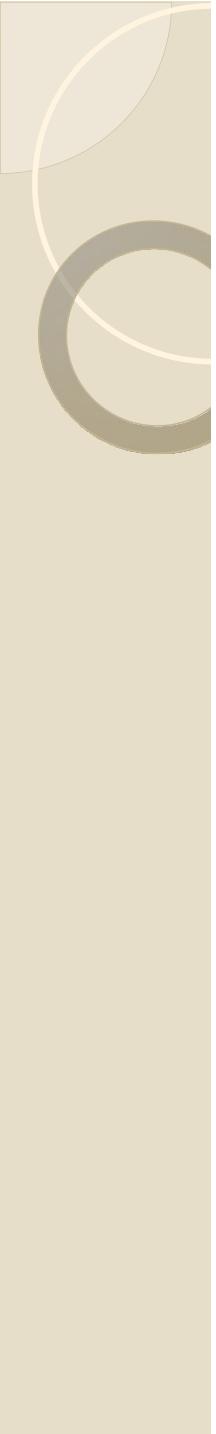
$$sim_{BM25}(d_j, q) \sim \sum_{k_i \in q \wedge k_i \in d_j} B_{i,j} \times \log \frac{N - n_i + 0.5}{n_i + 0.5},$$

- B_{ij} stanowi współczynnik określony wzorem

$$B_{i,j} = \frac{(K_1 + 1)f_{i,j}}{K_1 \left[(1 - b) + b \frac{\text{len}(d_j)}{\overline{\text{len}}} \right] + f_{i,j}},$$

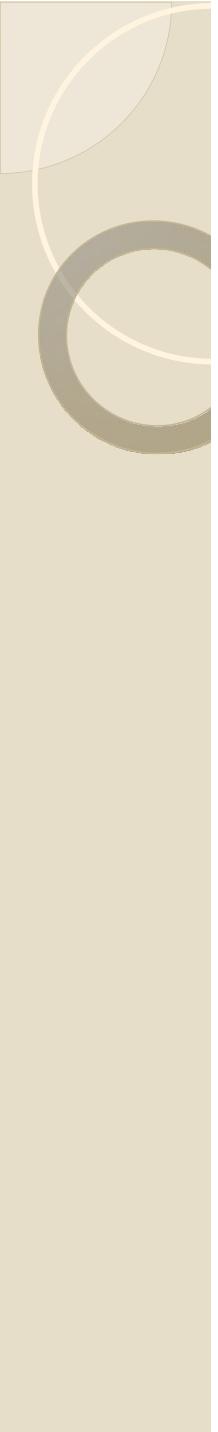
gdzie $b \in [0, 1]$ i K_1 – stałe empiryczne, zazwyczaj $b=0.75$ oraz $K_1=1$

- $b=0$ – redukuje model do BM15
- $b=1$ – redukuje model do BM11
- Stałe empiryczne można wyznaczyć z eksperymentów



Ocena wyszukiwania

- Ocena systemu wyszukiwania to miara tego, jak bardzo system spełnia potrzeby informacyjne użytkowników
 - Problem stanowi inna ocena jakości tego samego zbioru wynikowego przez różnych użytkowników
 - Aby rozwiązać problem wprowadzono różne miary jakości wyszukanego zbioru skorelowane z preferencjami różnych grup użytkowników
 - Aby oceniać systemy wyszukiwania (indeksowania) opracowano testowe kolekcje referencyjne bazujące na założeniach wynikających z badań Cranfielda (1958-1966)
 - Pozwalają one porównywać różne metody rankingu na tych samych zbiorach pytań i dokumentów
-

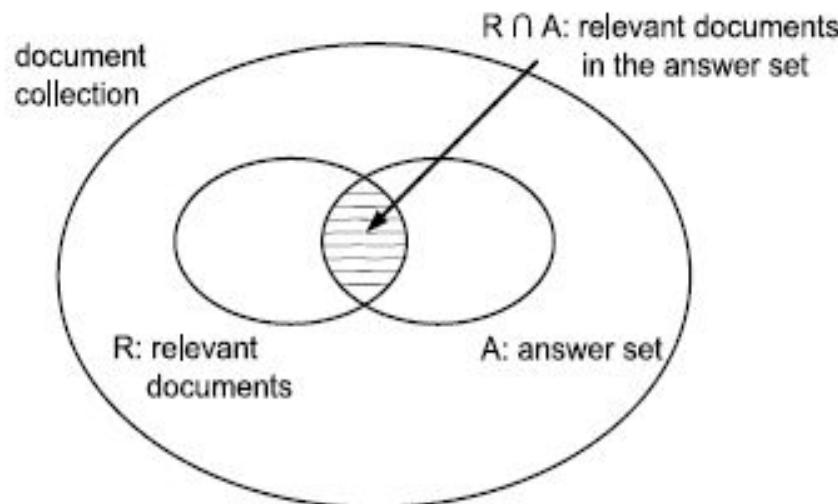


Ocena wyszukiwania

- Kolekcja referencyjna składa się z:
 - Zbioru D preselekcyjnych dokumentów,
 - Zbioru I opisów informacji używanej do testowania,
 - Zbioru ocen istotności związanych z każdą parą $[i_m, d_j]$ gdzie $i_m \in I$, $d_j \in D$.
 - Ocena istotności dokumentu ma wartość 0 gdy dokument d_j jest nieistotny dla i_m i 1 w przeciwnym przypadku
 - Taka ocena dokonywana jest przez specjalistów
-

dokładność i kompletność

- Dane są:
 - I : żądanie informacji,
 - R : zbiór dokumentów istotnych dla I ,
 - A : zbiór odpowiedzi dla I , wygenerowanych przez system wyszukiwania IR,
 - $R \cap A$: iloczyn zbiorów R i A .





dokładność i kompletność

- Kompletność jest częścią istotnych dokumentów (zbiór R), które zostały wyszukane:

$$Recall = \frac{|R \cap A|}{|R|}$$

- Dokładność jest częścią wyszukanych dokumentów (zbiór A), które są istotne:

$$Precision = \frac{|R \cap A|}{|A|}$$

- Definicja dokładności i kompletności zakłada, że wszystkie dokumenty w zbiorze A są testowane
- Użytkownik widzi uszeregowany zbiór dokumentów i sprawdza je od początku, więc dokładność i kompletność zmieniają się podczas przeglądania zbioru A .



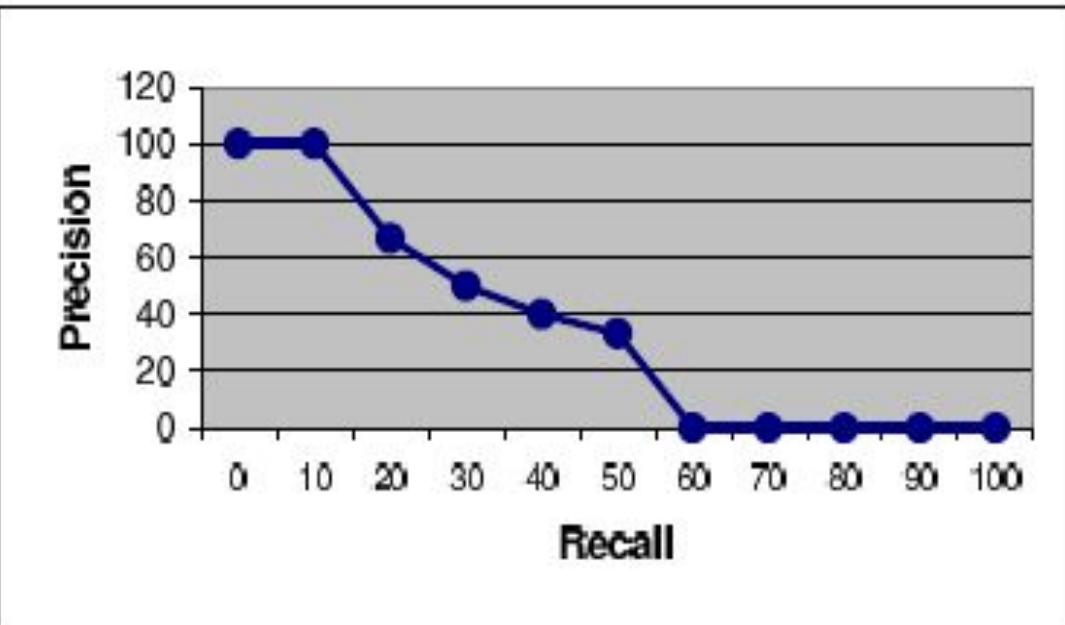
dokładność i kompletność

- Wskazane jest wykreślenie zależności dokładności (precision) od kompletności (recall)
- Niech R_{q_1} będzie zbiorem dokumentów istotnych dla zapytania q_1 :
$$R_{q_1} = \{d_3, d_5, d_9, d_{25}, d_{39}, d_{44}, d_{56}, d_{71}, d_{89}, d_{123}\}$$
- Algorytm wyszukiwania daje zbiór odpowiedzi (dokumenty istotne oznaczone kółkiem)

- | | | |
|-----------------|----------------|---------------|
| 01. d_{123} • | 06. d_9 • | 11. d_{38} |
| 02. d_{84} | 07. d_{511} | 12. d_{48} |
| 03. d_{56} • | 08. d_{129} | 13. d_{250} |
| 04. d_6 | 09. d_{187} | 14. d_{113} |
| 05. d_8 | 10. d_{25} • | 15. d_3 • |

dokładność i kompletność

- Wykres dokładności w funkcji kompletności dla kolejnych dokumentów odpowiedzi



Recall	Precision
0	100
10	100
20	66.6
30	50
40	40
50	33.3
60	0
70	0
80	0
90	0
100	0



dokładność i kompletność

- Rozważmy zapytanie q_2 dla którego zbiór istotnych odpowiedzi jest dany jako:

$$R_{q2} = \{d_3, d_{56}, d_{129}\}$$

- Algorytm IR przetwarza zapytanie i zwraca następujący ranking:

01.	d_{425}	06.	d_{615}	11.	d_{193}
02.	d_{87}	07.	d_{512}	12.	d_{715}
03.	$d_{56} \bullet$	08.	$d_{129} \bullet$	13.	d_{810}
04.	d_{32}	09.	d_4	14.	d_5
05.	d_{124}	10.	d_{130}	15.	$d_3 \bullet$

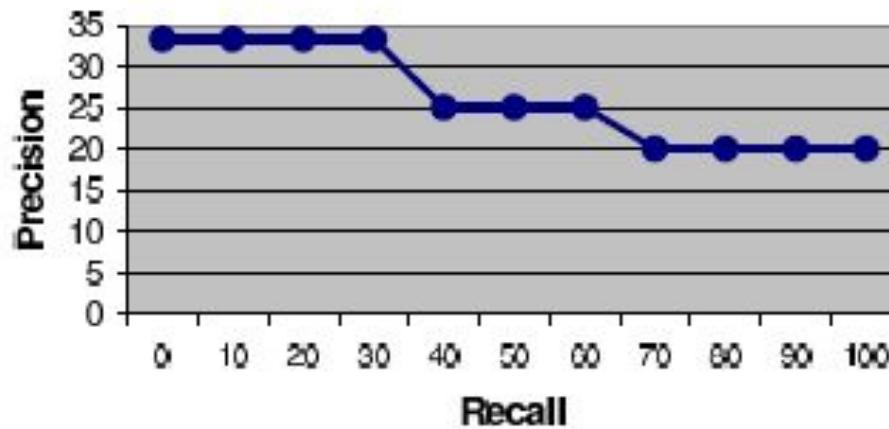
- Pierwszy istotny dokument d_{56} zapewnia dokładność i kompletność 33,3%, drugi d_{129} daje kompletność 66,6% i dokładność 25%, trzeci d_3 zapewnia 100% kompletności z dokładnością 20%

dokładność i kompletność

- Dla standardowych poziomów kompletności $r_j = \{0,1, \dots, 10\}$ dokładność jest interpolowana następująco

$$P(r_j) = \max_{\forall r \mid r \geq r_j} P(r)$$

- Ta zasada interpolacji umożliwia zbudowanie dla ostatniego przykładu następującego wykresu *Precision(Recall)*



Recall	Precision
0	33.3
10	33.3
20	33.3
30	33.3
40	25
50	25
60	25
70	20
80	20
90	20
100	20



dokładność i kompletność

- Zazwyczaj ocenia się średnią jakość wyszukiwania dla zbioru N_q pytań testujących

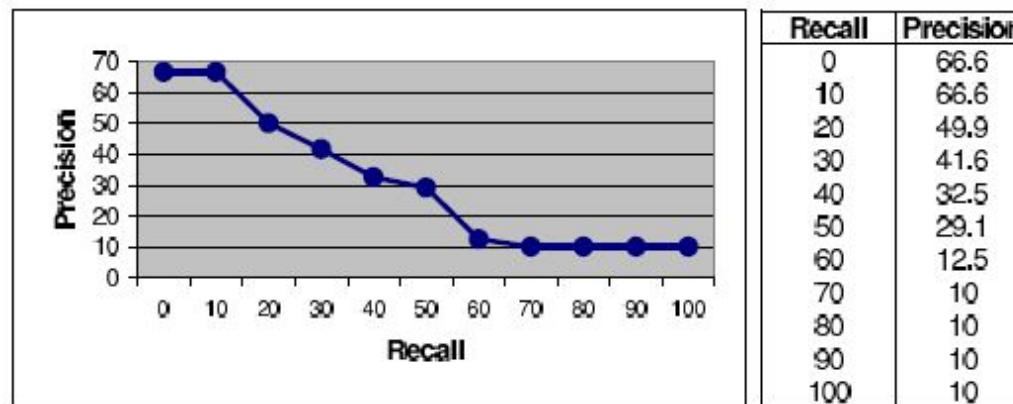
$$\overline{P(r_j)} = \sum_{i=1}^{N_q} \frac{P_i(r_j)}{N_q}$$

gdzie:

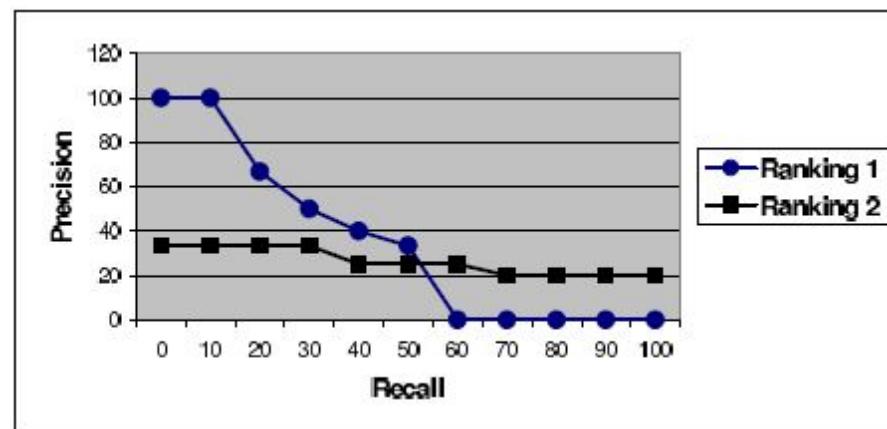
- $\overline{P(r_j)}$ - średnia dokładność na poziomie kompletności r_j ,
- $P_i(r_j)$ – dokładność odpowiedzi na pytanie q_i z poziomem kompletności r_j .

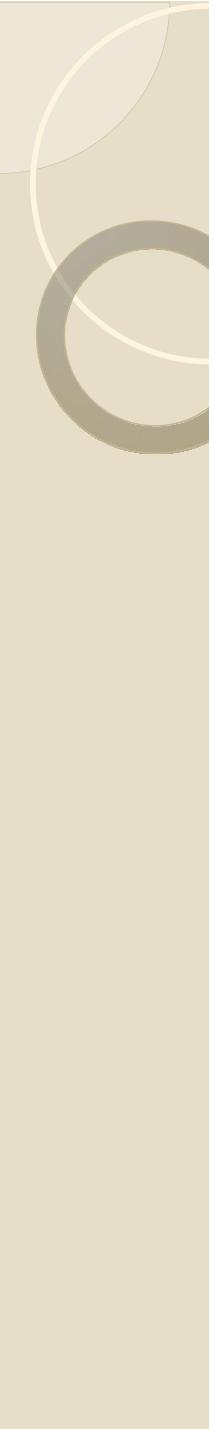
dokładność i kompletność

- Wykres *Precision(Recall)* uśredniony dla dwóch zapytań q_1 i q_2 :



- Uśrednione przebiegi *Precision(Recall)* dla 2 różnych algorytmów wyszukiwania:





Dokładność i kompletność

- Dokładność i kompletność są często stosowane do oceny jakości wyszukiwania przez algorytmy IR
- Poprawna estymacja kompletności zapytania wymaga znajomości wszystkich dokumentów kolekcji testowej
- Miara *Precision-Recall* nie jest skalarna
- Jakość algorytmu IR jest mierzona na zbiorze zapytań w trybie wsadowym
- Dla systemów IR wymagających słabego uporządkowania dokumentów miara *Presision-Recall* może nie być najlepsza



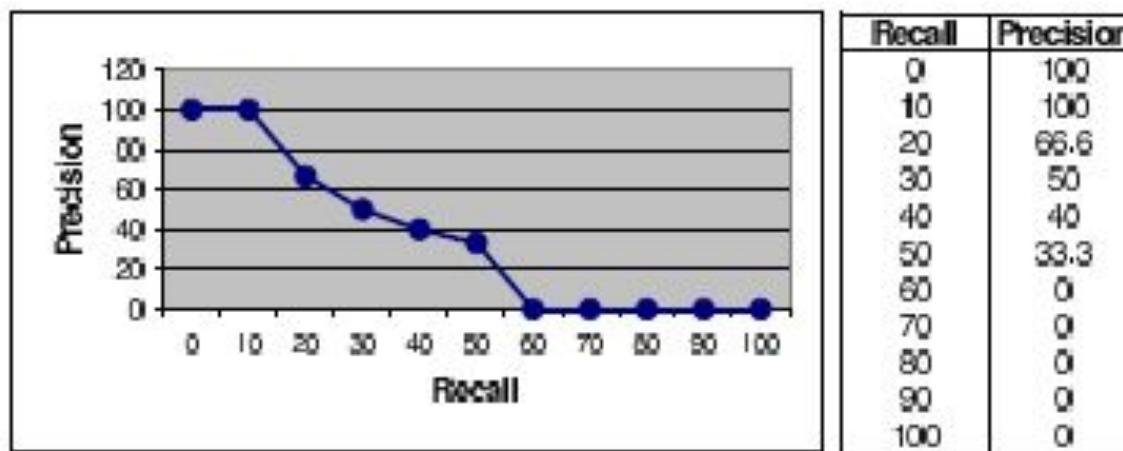
Miary $P@5$ i $P@10$

- Użytkownicy wyszukiwarek cenią przede wszystkim dużo istotnych dokumentów na początku rankingu
- Miary $P@5$ i $P@10$ wyznaczają dokładność odpowiednio dla 5 i 10 pierwszych dokumentów; pozwalają ocenić czy użytkownicy uzyskują istotne dokumenty na początku rankingu
- Rozpatrujemy listę dokumentów dla przykładowego pytania q_1 :

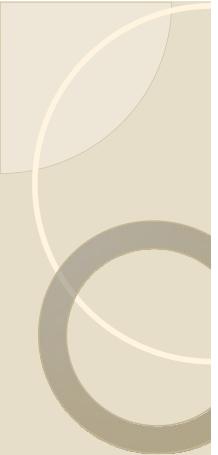
01. d_{123} •	06. d_9 •	11. d_{38}
02. d_{84}	07. d_{511}	12. d_{48}
03. d_{56} •	08. d_{129}	13. d_{250}
04. d_6	09. d_{187}	14. d_{113}
05. d_8	10. d_{25} •	15. d_3 •

MAP: Mean average precision

- Dla tego pytania $P@5 = 40\%$ i $P@10 = 40\%$
- $P@5$ i $P@10$ można uśredniać w zbiorze 100 zapytań
- MAP to średnia dokładność z wykresu *Precision-Recall*, przy standardowym zestawie kompletności
- Dla zapytania q_1 :



$$MAP_1 = \frac{1 + 0.66 + 0.5 + 0.4 + 0.33 + 0 + 0 + 0 + 0 + 0}{10} = 0.28$$



R-precision

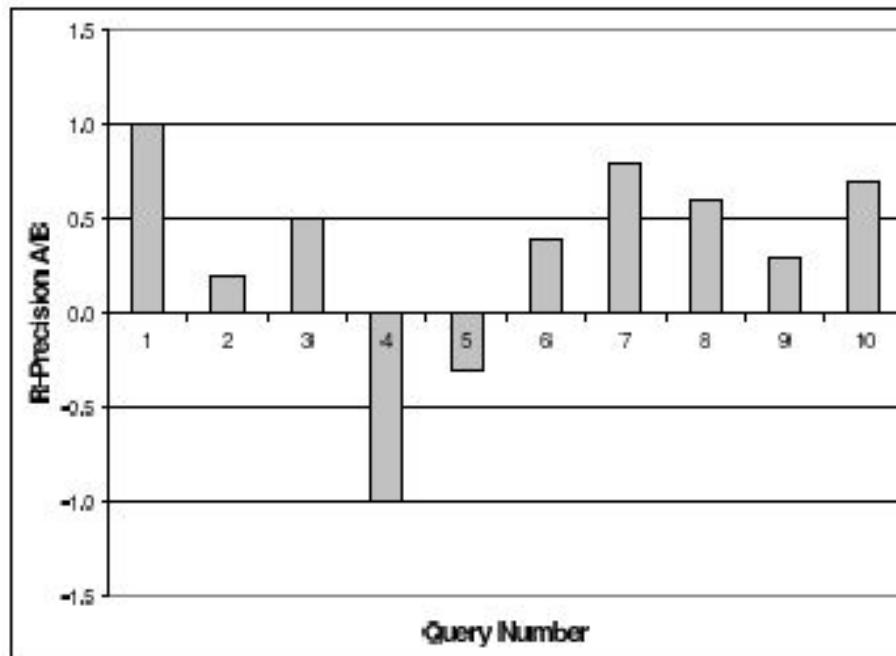
- R-dokładność to dokładność wyznaczona na r-tej pozycji w rankingu odpowiedzi
- Dla pytania q_1 są cztery dokumenty istotne wśród pierwszych 10-ciu zwróconych dokumentów więc R-dokładność wynosi $4/10=0,4$
- R-dokładność może być także uśredniana w zbiorze zapytań
- Może być stosowana do porównywania 2 algorytmów wyszukiwania dla każdego zapytania

$$RP_{A/B}(i) = RP_A(i) - RP_B(i)$$

- $RP_A(i)$ – R-dokładność algorytmu A dla i-tego zapytania
- $RP_B(i)$ – R-dokładność algorytmu B dla i-tego zapytania

Histogram dokładności

- Przykładowy histogram porównania algorytmów $RP_{A/B}(i)$ dla 10 różnych zapytań



- Algorytm A jest lepszy w 8 przypadkach na 10

MRR: Mean Reciprocal Rank

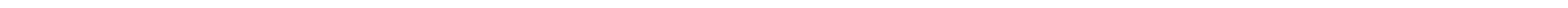
- MRR – dobra miara jakości wyszukiwania gdy interesuje nas pierwsza poprawna odpowiedź – np. adresy URL, strony główne w sieci
- Jeżeli przyjąć:
 - R_i : ranking odpowiedzi dla zapytania q_i ,
 - $S_{correct}(R_i)$: pozycja pierwszej poprawnej odpowiedzi w R_i ,
 - S_h : próg pozycji rankingu,
- to odwrotna ranga $RR(R_i)$ dla pytania q_i jest wyrażona jako

$$RR(\mathcal{R}_i) = \begin{cases} \frac{1}{S_{correct}(\mathcal{R}_i)} & \text{if } S_{correct}(\mathcal{R}_i) \leq S_h \\ 0 & \text{otherwise} \end{cases}$$

MRR: Mean Reciprocal Rank

- Średnia odwrotna ranga (MRR) dla zbioru Q z N_q zapytań:

$$MRR(Q) = \sum_i^{N_q} RR(\mathcal{R}_i)$$



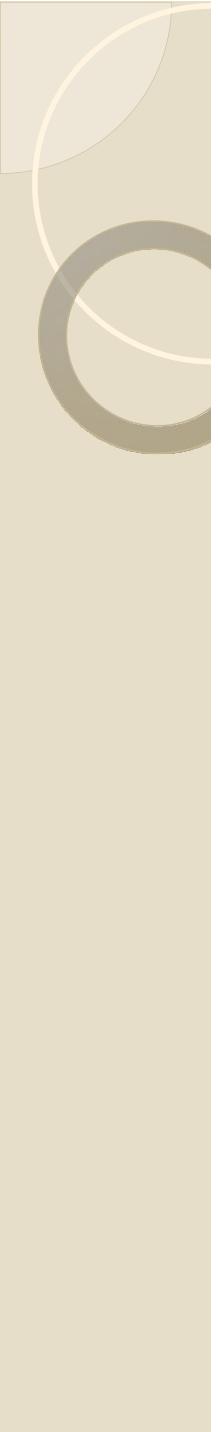


Miara E

- Łączy dokładność i kompletność w jednym wskaźniku skalarnym uwzględniającym ich proporcje:

$$E(j) = 1 - \frac{1 + b^2}{\frac{b^2}{r(j)} + \frac{1}{P(j)}}$$

- gdzie
 - $r(j)$ – kompletność na j-tej pozycji w rankingu,
 - $P(j)$ – dokładność na j-tej pozycji w rankingu,
 - $b \geq 0$ – ustalony parametr przewagi kompletności nad dokładnością; $b=0 \quad \Downarrow E(j) = 1-P(j)$ oraz $b \rightarrow \infty \quad \Downarrow E(j) = 1-r(j)$
 - $E(j)$ – E-metryka na j-tej pozycji w rankingu.

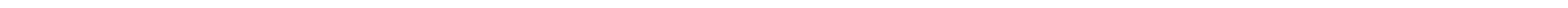


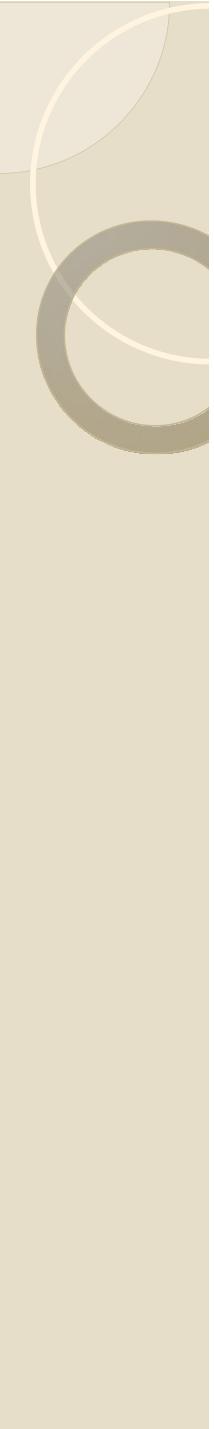
Miara F (średnia harmoniczna)

- Dla $b=1$ miara E staje się miarą średniej harmonicznej F :

$$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}}$$

- Funkcja F ma wartości w zakresie $[0,1]$; przyjmuje wartość 0 gdy żadne istotne dokumenty nie zostały znalezione lub wartość 1 gdy wszystkie dokumenty odpowiedzi są istotne
- Duże wartości F wiążą się z dużą dokładnością i kompletnością

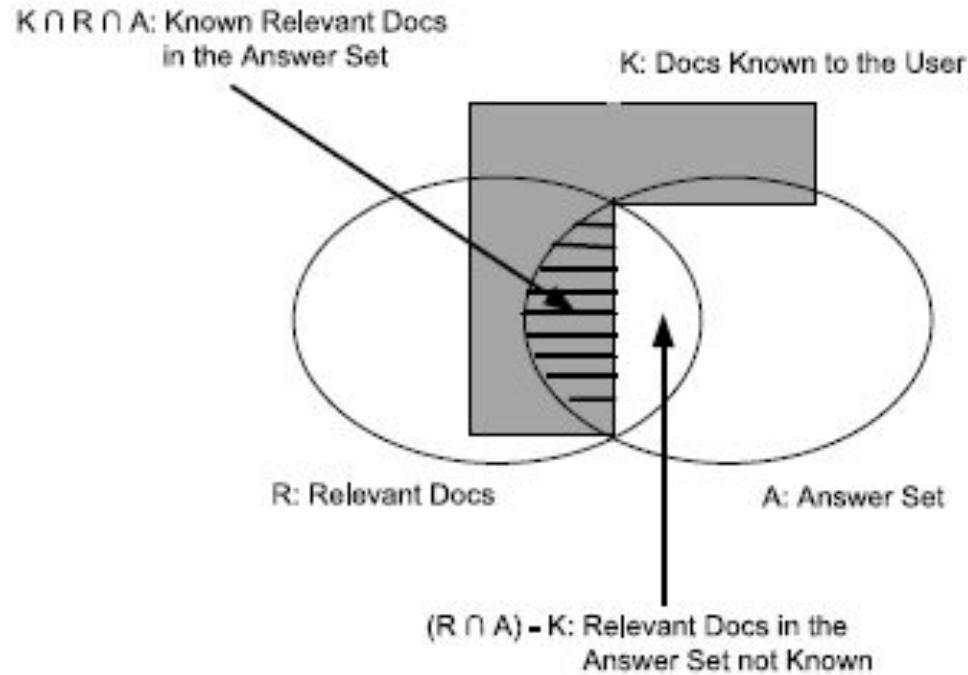




Miary zorientowane na użytkownika

- Dokładność i kompletność zakładają, że zbiór istotnych dokumentów zwróconych w odpowiedzi na zapytanie nie zależy od użytkowników
 - Różni użytkownicy mogą różnie rozumieć istotność dokumentów
-

Miary zorientowane na użytkownika



- K: zbiór dokumentów znanych użytkownikowi
- $K \cap R \cap A$: zbiór wyszukanych istotnych dokumentów znanych użytkownikowi
- $(R \cap A) - K$: zbiór wyszukanych istotnych dokumentów nieznanych użytkownikowi

Miary zorientowane na użytkownika

- Współczynnik pokrycia – frakcja dokumentów znanych i istotnych w zbiorze odpowiedzi

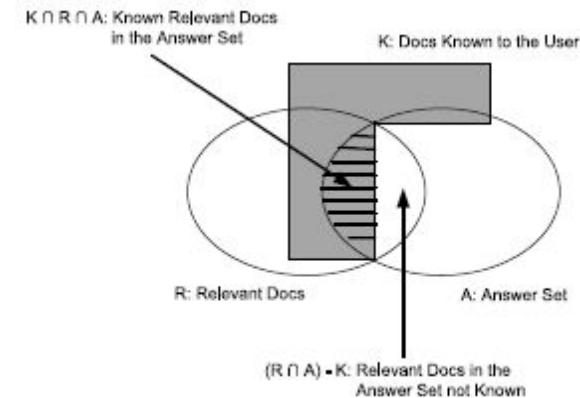
$$\text{coverage} = \frac{|K \cap R \cap A|}{|K \cap R|}$$

- Współczynnik nowości – frakcja istotnych dokumentów w zbiorze odpowiedzi nieznanych użytkownikowi

$$\text{novelty} = \frac{|(R \cap A) - K|}{|R \cap A|}$$

Wysokie pokrycie, wskazuje że system IR znalazł większość dokumentów oczekiwanych przez użytkownika

Wysoka nowość oznacza znalezienie wielu nowych, istotnych dokumentów





DCG — Discounted Cumulated Gain

- Dokładność i kompletność pozwalają jedynie na binarną ocenę istotności
- Nie ma rozróżnienia między bardzo i średnio ważnymi dokumentami
- Zredukowane skumulowane wzmacnienie DCG jest miarą uwzględniającą stopień istotności dokumentów
 - Bardzo istotne dokumenty są preferowane na początku rankingu,
 - Istotne dokumenty na końcu rankingu są mniej ważne
- Dla kilkunastu dokumentów $d[i]$ odpowiedzi na zapytanie q_j tworzy się wektor wzmacnienia $G_j[i]$ odpowiadający przyjętej skali istotności dokumentów np. 0-3.
- Wzmocnienie 0 mają dokumenty nieistotne.



DCG — Discounted Cumulated Gain

- Wektor skumulowany CG_j odpowiada sumowaniu wzmocnień w kolejności rankingu:

$$CG_j[i] = \begin{cases} G_j[1] & \text{if } i = 1; \\ G_j[i] + CG_j[i - 1] & \text{otherwise} \end{cases}$$

- Współczynnik dyskonta stanowi logarytm pozycji rankingu np. $\log_2(i)$,
- Dzieląc elementy wektora skumulowanego $CG_j[i]$ przez odpowiednie współczynniki dyskonta otrzymuje się współczynnik DCG – zdyskontowanego skumulowanego wzmocnienia.

DCG — Discounted Cumulated Gain

$$DCG_j[i] = \begin{cases} G_j[1] & \text{if } i = 1; \\ \frac{G_j[i]}{\log_2 i} + DCG_j[i - 1] & \text{otherwise} \end{cases}$$

- Dla przykładowych zapytań q_1 i q_2 uzyskuje się wektory G_1 , G_2 oraz DCG_1 i DCG_2 .

$$G_1 = (1, 0, 1, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 0, 3)$$

$$G_2 = (0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 3)$$

$$DCG_1 = (1.0, 1.0, 1.6, 1.6, 1.6, 2.8, 2.8, 2.8, 2.8, 3.4, 3.4, 3.4, 3.4, 3.4, 4.2)$$

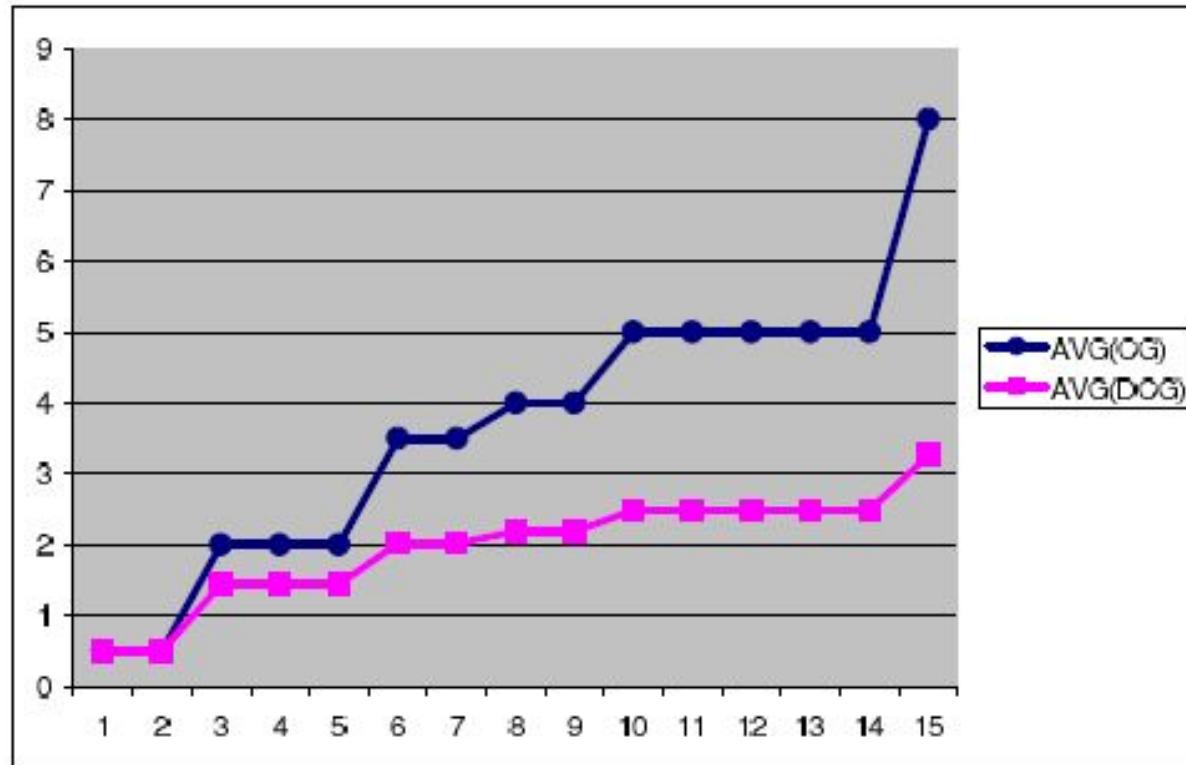
$$DCG_2 = (0.0, 0.0, 1.3, 1.3, 1.3, 1.3, 1.3, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 2.4)$$

- Dyskontowane skumulowane wzmacnienia są znacznie mniej wrażliwe na występowanie istotnych dokumentów pod koniec rankingu

DCG — Discounted Cumulated Gain

- Funkcje $\overline{CG}[i]$ i $\overline{DCG}[i]$ uśrednione w zbiorze N_q zapytań:

$$\overline{CG}[i] = \sum_{j=1}^{N_q} \frac{CG_j[i]}{N_q}; \quad \overline{DCG}[i] = \sum_{j=1}^{N_q} \frac{DCG_j[i]}{N_q}$$



DCG -znormalizowane

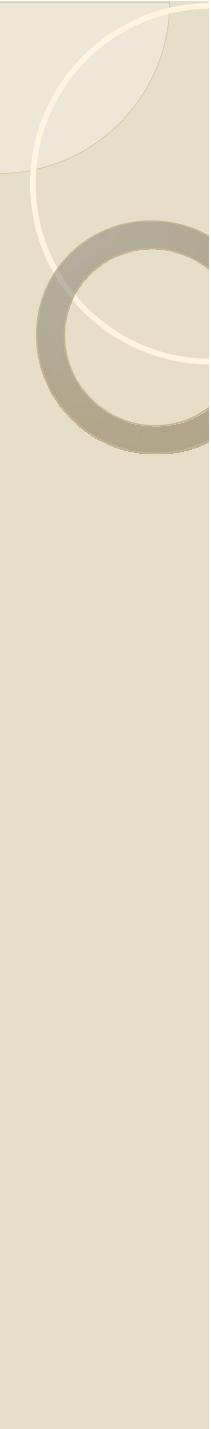
- Dokładność i kompletność są obliczane względem zbioru istotnych dokumentów
- CG i DCG nie mają odniesienia do żadnej konkretnej bazy, co może utrudniać użycie ich do porównania metod wyszukiwania
- Niech ICG i $IDCG$ będą idealnymi odpowiednikami CG i DCG - po wysortowaniu istotności dokumentów w porządku malejącym
- Wprowadza się znormalizowane wartości CG i DCG :

$$NCG[i] = \frac{\overline{CG}[i]}{\overline{ICG}[i]}; \quad NDCG[i] = \frac{\overline{DCG}[i]}{\overline{IDCG}[i]}$$



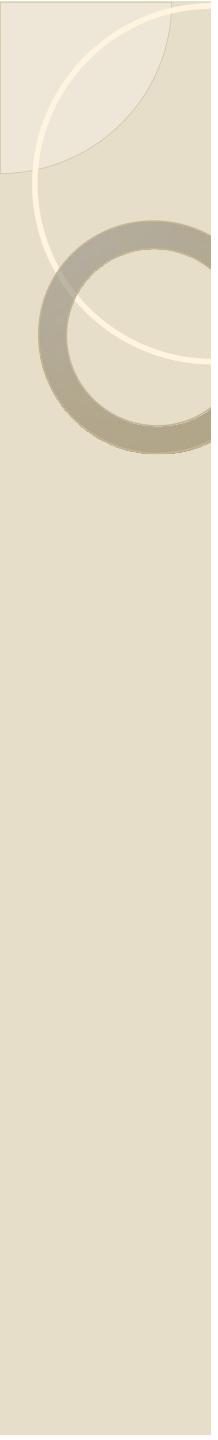
DCG -znormalizowane

- Pole pod krzywymi NCG i $NDCG$ reprezentuje jakość algorytmu wyszukiwania z rankingiem
 - Większe pole oznacza lepszy algorytm
 - Kumulowane wzmacnienie zapewnia skalarną miarę jakości wyszukiwania na dowolnej pozycji rankingu
 - Zdyskontowane wzmacnienie pozwala na kontrolę wpływu ważnych dokumentów pod koniec rankingu na ocenę jakości algorytmu
-



BPREF — preferencje binarne

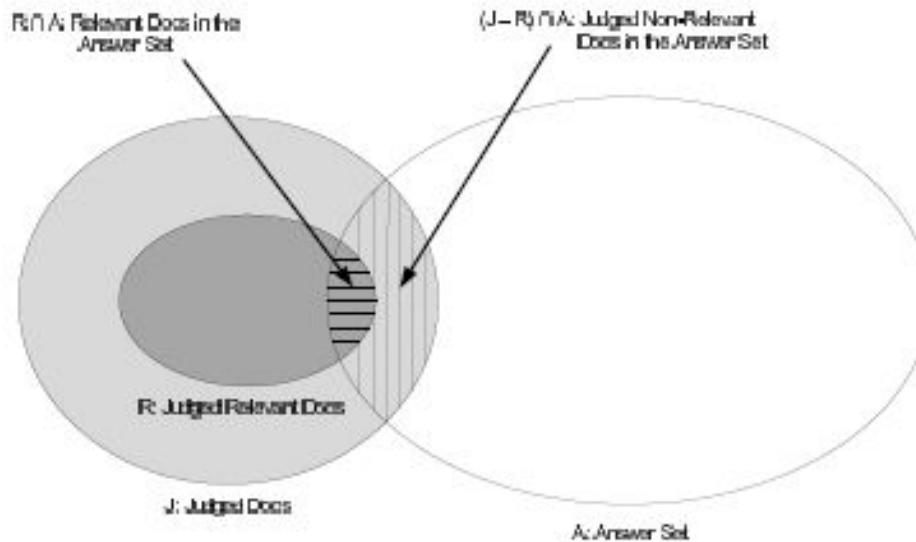
- Paradygmat Cranfielda zakłada ocenę wszystkich dokumentów testowych ze względu na każde zapytanie
- To może być spełnione tylko dla niewielkich zbiorów
- Dla dużych zbiorów stosuje się metodę grupowania (pooling)
 - Składa ona do puli tylko najtrajniejsze wyniki uzyskane przez różne algorytmy wyszukiwania,
 - Następnie dokumenty z puli są oceniane w celu porównania metod
- Miary typu *Precision-Recall* przyjmują za nieistotne wszystkie dokumenty nie wyszukane
- Jest to nie do przyjęcia dla kolekcji złożonych z miliardów dokumentów



BPREF — preferencje binarne

- Problem rozwiązuje się poprzez ustalenie preferencji między parami wyszukanych dokumentów zamiast bezpośredniego używania ich rangi - jak to ujęto w przypadku BPREF
- BPREF mierzy ilość dokumentów nieistotnych poprzedzających dokumenty istotne
- Ocenia się czy dokument d_j jest preferowany w parze z d_k dla danego zapytania q .
- Ponadto każdy dokument istotny jest preferowany w parze z nieistotnym

BPREF — preferencje binarne



- Dla wymaganej informacji I :
 - R_A : ranking wyliczony przez system IR w odniesieniu do I ,
 - $s_{A,j}$: pozycja dokumentu d_j w R_A ,
 - $[(J - R) \wedge A]_{|R|}$: zbiór pierwszych $|R|$ dokumentów w R_A ocenionych jako nieistotne

BPREF — preferencje binarne

- Ilość nieistotnych dokumentów pojawiających się w R_A przed d_j .

$$C(\mathcal{R}_A, d_j) = \parallel \{d_k \mid d_k \in [(J - R) \cap A]_{|R|} \wedge s_{A,k} < s_{A,j}\} \parallel$$

- Miara BPREF dla rankingu R_A :

$$Bpref(\mathcal{R}_A) = \frac{1}{|R|} \sum_{d_j \in (R \cap A)} \left(1 - \frac{C(\mathcal{R}_A, d_j)}{\min(|R|, |(J - R) \cap A|)} \right)$$

- Dla każdego istotnego dokumentu d_j w rankingu $Bpref$ akumuluje wagę zmieniającą się odwrotnie do ilości nieistotnych dokumentów poprzedzających
- Jeśli liczba znanych dokumentów istotnych jest bardzo mała (1 lub 2) stosuje się skorygowaną miarę BPREF-10

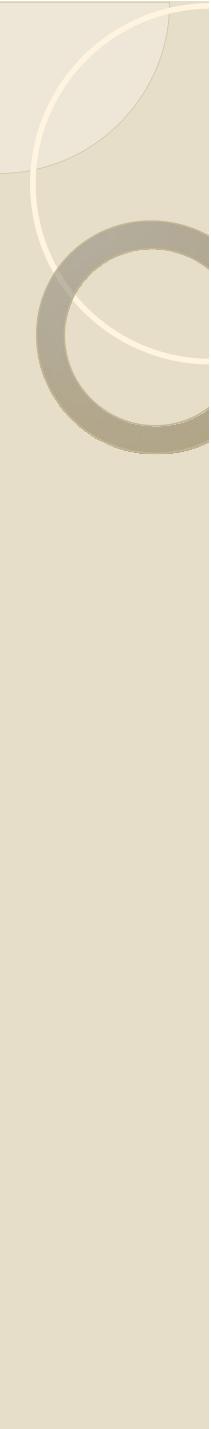
BPREF — preferencje binarne

- Zapewnia ona minimum 10 istotnych dokumentów

$$C_{10}(\mathcal{R}_A, d_j) = \left\| \{d_k \mid d_k \in [(J - R) \cap A]_{|R|+10} \wedge s_{A,k} < s_{A,j}\} \right\|$$

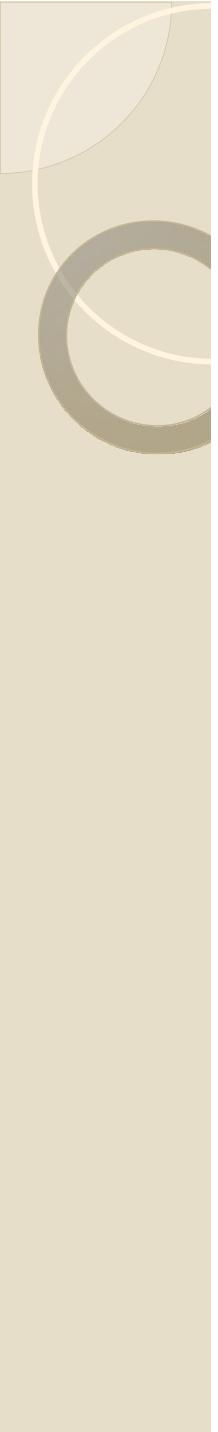
- Wówczas:

$$Bpref_{10}(\mathcal{R}_A) = \frac{1}{|R|} \sum_{d_j \in (R \cap A)} \left(1 - \frac{C_{10}(\mathcal{R}_A, d_j)}{\min(|R|+10, |(J-R) \cap A|)} \right)$$



Miary korelacji rangowej

- Dokładność i zupełność pozwalają na porównanie istotności wyników uzyskanych przez dwie funkcje rankingowe
- W pewnych sytuacjach:
 - Nie można bezpośrednio określić istotności dokumentów,
 - Należy ustalić jak bardzo dana funkcja rankingu różni się od drugiej znanej
- Wówczas interesuje nas porównanie uporządkowania dwóch rankingów
- Można tego dokonać poprzez funkcje statystyczne zwane miarami korelacji rangowej



Miary korelacji rangowej

- Miary korelacji rankingów R_1 i R_2 można wyrazić przez współczynnik korelacji $C(R_1, R_2)$ o następujących właściwościach
 - $-1 \leq C(R_1, R_2) \leq 1$,
 - Jeżeli $C(R_1, R_2) = 1$ zgodność pomiędzy rankingami jest pełna,
 - Jeżeli $C(R_1, R_2) = -1$ niezgodność pomiędzy rankingami jest pełna, tzn. są one odwrócone względem siebie,
 - Jeżeli $C(R_1, R_2) = 0$ dwa rankingi są kompletnie niezależne,
 - Zwiększenie wartości $C(R_1, R_2)$ implikuje zwiększenie zgodności rankingów

Współczynnik Spearman'a

- Współczynnik Spearman'a jest najczęściej używaną miarą korelacji rangowej – bazuje on na różnicach pozycji tego samego dokumentu w różnych rankingach

documents	$s_{1,j}$	$s_{2,j}$	$s_{i,j} - s_{2,j}$	$(s_{1,j} - s_{2,j})^2$
d_{123}	1	2	-1	1
d_{84}	2	3	-1	1
d_{56}	3	1	+2	4
d_6	4	5	-1	1
d_8	5	4	+1	1
d_9	6	7	-1	1
d_{511}	7	8	-1	1
d_{129}	8	10	-2	4
d_{187}	9	6	+3	9
d_{25}	10	9	+1	1
Sum of Square Distances				24

- Dane z 10 dokumentów wyszukanych przez 2 rankingi R_1 i R_2 . $s_{1,j}$ and $s_{2,j}$ są pozycjami j -tego dokumentu w tych rankingach

Współczynnik Spearman'a

- Przy porządkowaniu K dokumentów maksimum sumy kwadratów różnic pozycji rankingu wynosi

$$\frac{K \times (K^2 - 1)}{3}$$

- Przy pełnej niezgodności rankingów dla $K=10$ maksimum to wynosi $(10 \times (10^2 - 1))/3 = 330$
- Współczynnik Spearmana korelacji rang $S(R_1, R_2)$:

$$S(\mathcal{R}_1, \mathcal{R}_2) = 1 - \frac{6 \times \sum_{j=1}^K (s_{1,j} - s_{2,j})^2}{K \times (K^2 - 1)}$$

- zmienia się w zakresie $[-1, 1]$, K – rozmiar porównywanych zbiorów

Współczynnik Spearman'a

■ Przykład:

$$S(\mathcal{R}_1, \mathcal{R}_2) = 1 - \frac{6 \times 24}{10 \times (10^2 - 1)} = 1 - \frac{144}{990} = 0.854$$

documents	$s_{1,j}$	$s_{2,j}$	$s_{i,j} - s_{2,j}$	$(s_{1,j} - s_{2,j})^2$
d_{123}	1	2	-1	1
d_{84}	2	3	-1	1
d_{56}	3	1	+2	4
d_6	4	5	-1	1
d_8	5	4	+1	1
d_9	6	7	-1	1
d_{511}	7	8	-1	1
d_{129}	8	10	-2	4
d_{187}	9	6	+3	9
d_{25}	10	9	+1	1
Sum of Square Distances				24

Współczynnik tau Kendall'a

- Posiada on naturalną i intuicyjną interpretację

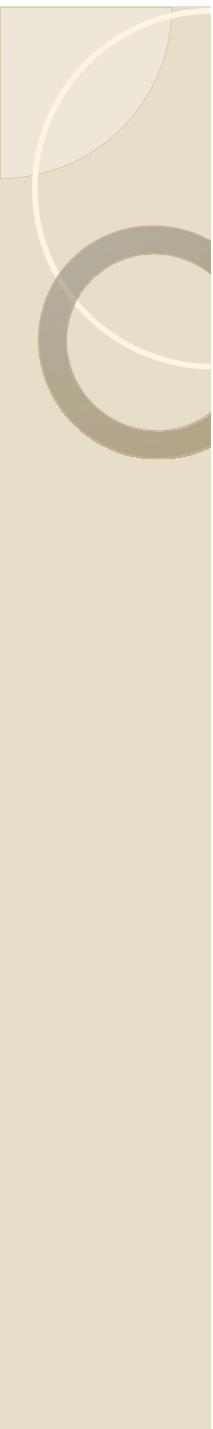
$$\tau(\mathcal{R}_1, \mathcal{R}_2) = P(\mathcal{R}_1 = \mathcal{R}_2) - P(\mathcal{R}_1 \neq \mathcal{R}_2)$$

- gdzie

- $P(R_1 = R_2)$ – znormalizowana ilość zgodnych par dokumentów (concordant), dla których różnice w pozycjach rankingu par dokumentów $[d_k, d_j]$: $s_{1,k} - s_{1,j}$ oraz $s_{2,k} - s_{2,j}$ są tego samego znaku dla 2 porównywanych metod
- $P(R_1 \neq R_2)$ – znormalizowana ilość niezgodnych par dokumentów (discordant), dla których różnice w pozycjach rankingu par dokumentów $[d_k, d_j]$: $s_{1,k} - s_{1,j}$ oraz $s_{2,k} - s_{2,j}$ są różnych znaków

- Jeżeli przyjąć

- $\Delta(R_1, R_2)$: liczba niezgodnych par dokumentów w rankingach R_1 i R_2 ,
- $K(K - 1) - \Delta(R_1, R_2)$: liczba zgodnych par dokumentów,



Współczynnik tau Kendall'a

- Wówczas:

$$P(\mathcal{R}_1 = \mathcal{R}_2) = \frac{K(K-1) - \Delta(\mathcal{R}_1, \mathcal{R}_2)}{K(K-1)}$$

$$P(\mathcal{R}_1 \neq \mathcal{R}_2) = \frac{\Delta(\mathcal{R}_1, \mathcal{R}_2)}{K(K-1)}$$

- co daje:

$$\tau(\mathcal{R}_1, \mathcal{R}_2) = 1 - \frac{2 \times \Delta(\mathcal{R}_1, \mathcal{R}_2)}{K(K-1)}$$

- Przykład: 5 dokumentów w 2 rankingach

documents	$s_{1,j}$	$s_{2,j}$	$s_{i,j} - s_{2,j}$
d_{123}	1	2	-1
d_{84}	2	3	-1
d_{56}	3	1	+2
d_6	4	5	-1
d_8	5	4	+1

Współczynnik tau Kendall'a

- Uporządkowane pary dokumentów w rankingu R_1

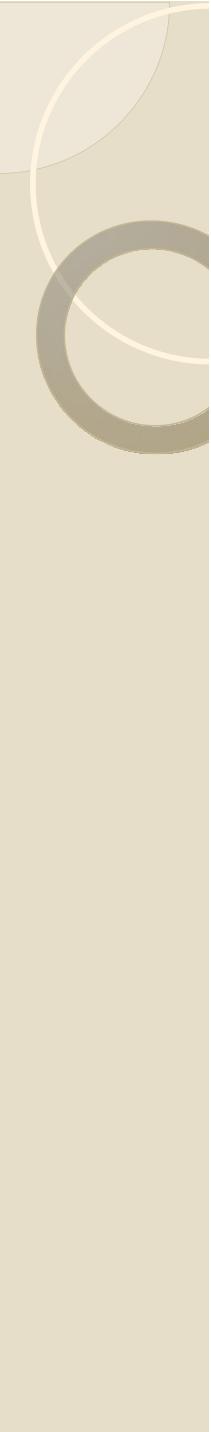
$[d_{123}, d_{84}]$, $[d_{123}, d_{56}]$, $[d_{123}, d_6]$, $[d_{123}, d_8]$,	$C, D, C, C,$
$[d_{84}, d_{56}]$, $[d_{84}, d_6]$, $[d_{84}, d_8]$,	$D, C, C,$
$[d_{56}, d_6]$, $[d_{56}, d_8]$,	$C, C,$
$[d_6, d_8]$	D

- Uporządkowane pary dokumentów w rankingu R_2 :

$[d_{56}, d_{123}]$, $[d_{56}, d_{84}]$, $[d_{56}, d_8]$, $[d_{56}, d_6]$,	$D, D, C, C,$
$[d_{123}, d_{84}]$, $[d_{123}, d_8]$, $[d_{123}, d_6]$,	$C, C, C,$
$[d_{84}, d_8]$, $[d_{84}, d_6]$,	$C, C,$
$[d_8, d_6]$	D

- Dla $K=5$ dokumentów $K(K-1)=20$ oraz

$$\begin{aligned}\tau(\mathcal{R}_1, \mathcal{R}_2) &= \frac{14}{20} - \frac{6}{20} \\ &= 0.4\end{aligned}$$





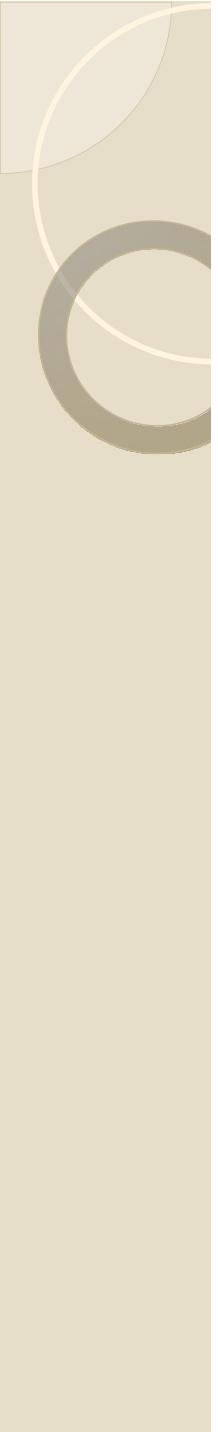
Eksploracja danych w Internecie

Klasyfikacja tekstów



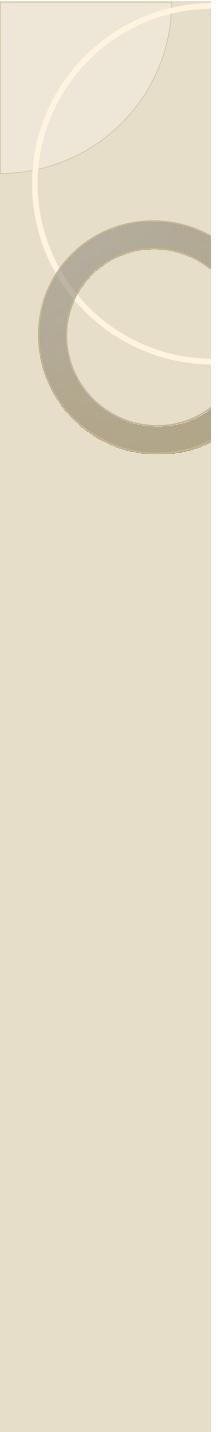
Klasyfikacja tekstów

- Aby umożliwić wyszukiwanie dokumentów na temat należy:
 - zgrupować dokumenty na podstawie tematów,
 - nadać tym grupom wyróżniające nazwy (etykiety),
- Klasyfikacja (kategoryzacja) tekstów:
Proces porządkowania informacji poprzez kojarzenie dokumentów tekstowych z klasami (kategoriami)
- Uczenie maszynowe:
 - Algorytmy, które uczą się wzorców danych,
 - Po wyuczeniu wzorców można przewidywać kategorie nowych danych,
 - Algorytmy uczące używają danych testowych przy uczeniu nadzorowanym i częściowo nadzorowanym.



Klasyfikacja tekstów

- Klasyfikator definiuje się następująco:
 - D : zbiór dokumentów,
 - $C = \{c_1, c_2, \dots, c_L\}$: zbiór L klas z odpowiednimi etykietami,
 - Klasyfikator tekstu jest funkcją binarną $F: D \times C \rightarrow \{0,1\}$ przypisującą każdej parze $[d_j, c_p]$, $d_j \in D$ i $c_p \in C$ wartość
 - 1, gdy d_j należy do klasy c_p ,
 - 0, gdy d_j nie należy do klasy c_p .
- Definicja ta objmuje algorytmy nadzorowane i nienadzorowane



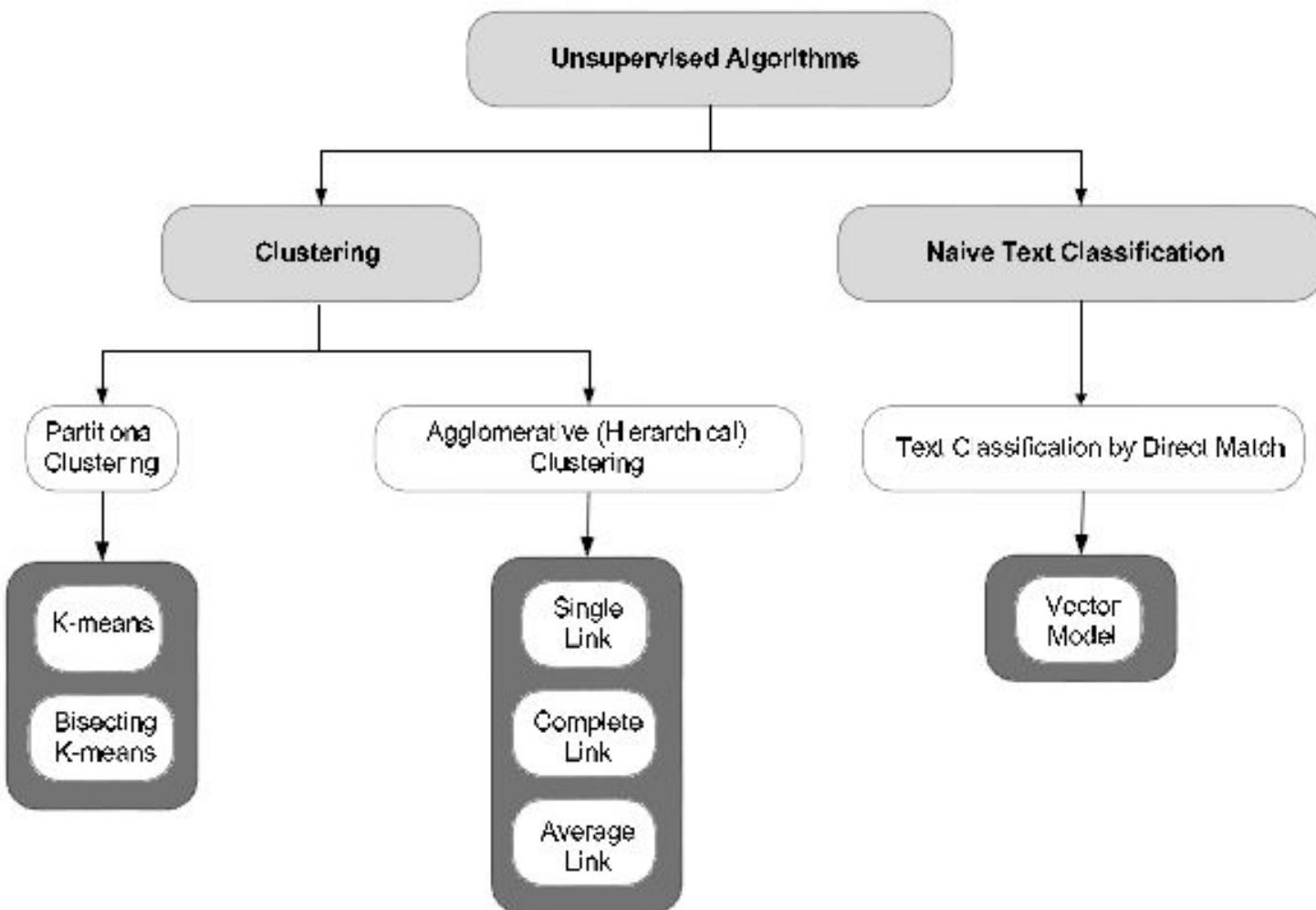
Klasyfikacja tekstów

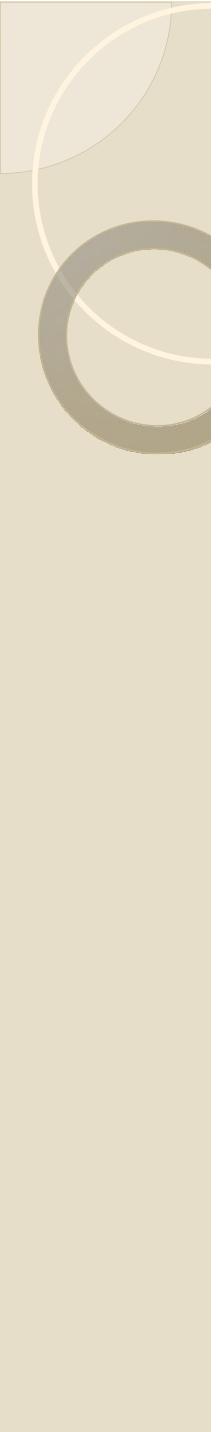
- Dokładniejsze są algorytmy nadzorowane:
 - jednoetykietowe – pojedyncza klasa przypisana do dokumentu,
 - wieloetykietowe – jedna lub więcej klas przypisanych do dokumentu

 - W ostatnim przypadku funkcja klasyfikacji $F(d_j, c_p)$ przestaje być binarna i zwraca stopień przynależności dokumentu d_j do klasy c_p .
-

Klasyfikacja tekstów

■ Algorytmy nienadzorowane



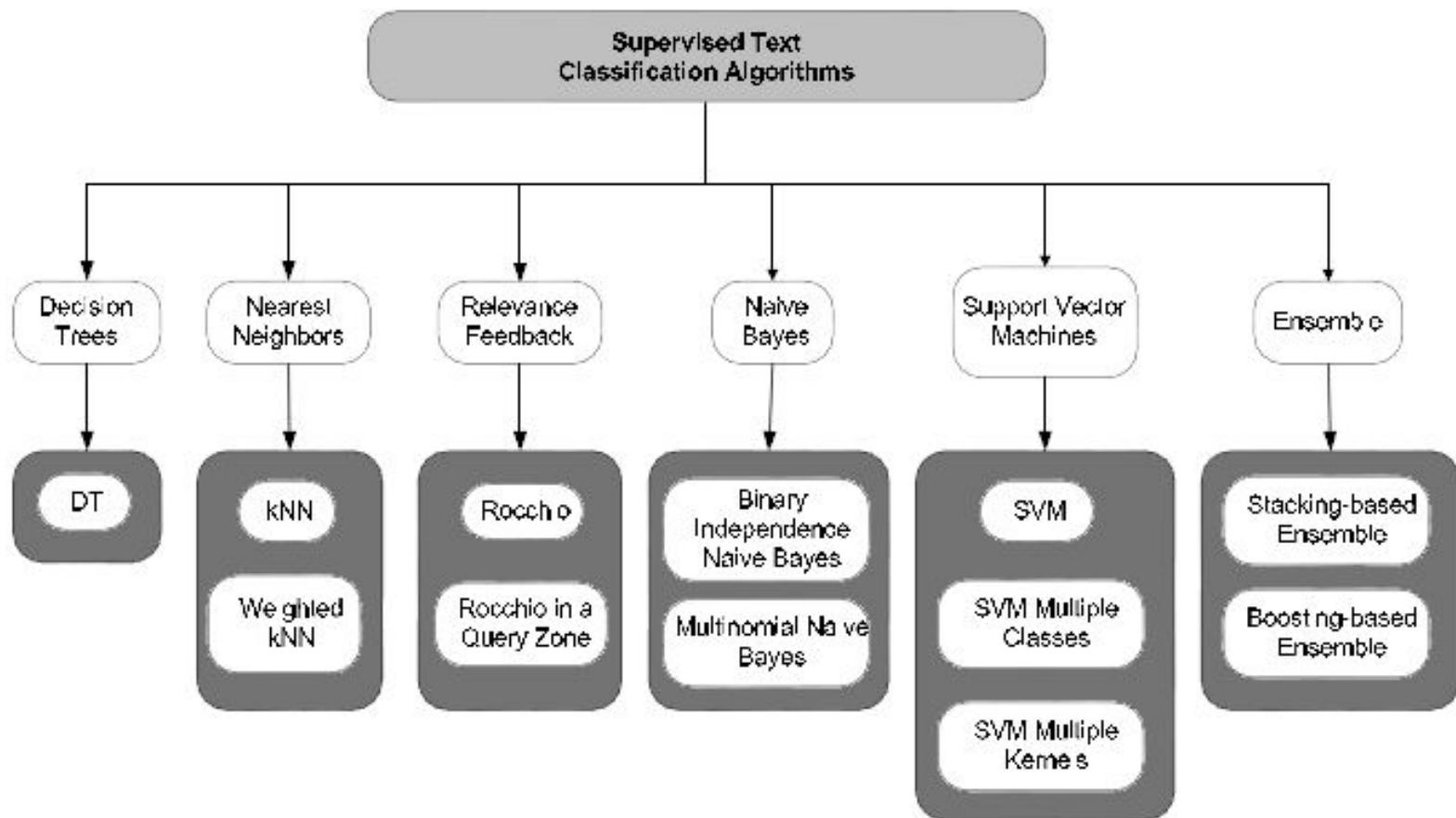


Klasyfikacja tekstów

- Algorytmy nadzorowane bazują na zbiorze testowym (uczącym)
 - Zbiór klas z przykładami dokumentów w każdej z nich
 - Przynależność do klas określają specjalści
 - Zbiór testowy służy do uczenia klasyfikatora
- Duża liczność zbioru uczącego lepiej dostraja klasyfikator i zapobiega przeuczeniu (*overfitting*)
- Klasyfikator podlega ocenie jakości (walidacja krzyżowa)

Klasyfikacja tekstów

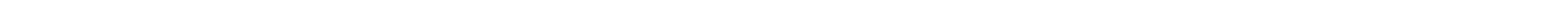
■ Nadzorowane algorytmy klasyfikacji

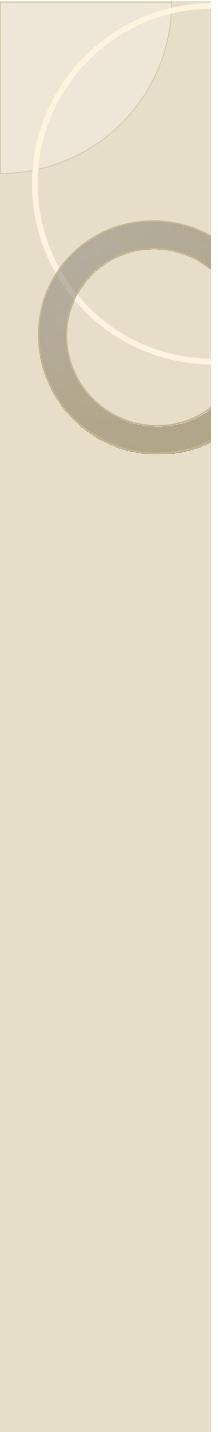




Klasyfikacja tekstów

Algorytmy nienadzorowane (klastering)





Klasyfikacja K-średnich

- W metodach nienadzorowanych etykiety klas są generowane automatycznie poprzez ustalenie środków skupień danych w odpowiednio dobranej przestrzeni atrybutów (termów) opisujących dokumenty
- Wyniki mogą być czasami niezadowalające lub różne od oczekiwanych przez użytkownika
- Metoda K-średnich:
 - Wejście – zadana liczba K klastrów,
 - Każdy klaster jest reprezentowany przez centroidę dokumentów,
 - Algorytm:
 - Przypisanie dokumentu do najbliższej centroidy,
 - Przeliczenie centroid,
 - Powtórzenie poprzednich kroków aż do stabilizacji położenia centroid

K-średnich – tryb wsadowy

- Wszystkie dokumenty są klasyfikowane przed przeliczeniem centroid
- Dokument d_j reprezentuje wektor \vec{d}_j
$$\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$$
- gdzie
 - $w_{i,j}$: waga termu k_i w dokumencie d_j ,
 - t : rozmiar słownika.
- Krok początkowy:
 - Wybierz losowo K dokumentów jako centroidy

$$\vec{\Delta}_p = \vec{d}_j$$

K-średnich – tryb wsadowy

■ Krok przypisania:

- przypisz każdy dokument do najbliższej centroidy
- oblicz funkcję odległości jako odwrotność funkcji podobieństwa d_i i c_p (formuła kosinusowa)

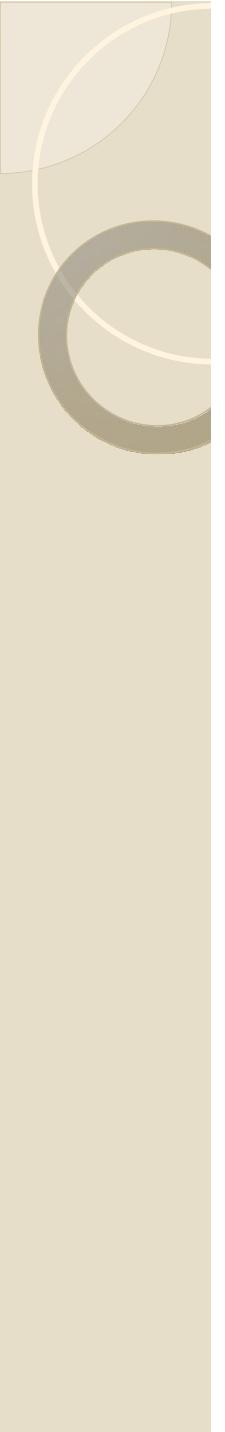
$$sim(d_j, c_p) = \frac{\vec{\Delta}_p \bullet \vec{d}_j}{|\vec{\Delta}_p| \times |\vec{d}_j|}$$

■ Krok poprawy – przeliczenie centroid każdego klastra

$$\vec{\Delta}_p = \frac{1}{size(c_p)} \sum_{\vec{d}_j \in c_p} \vec{d}_j$$

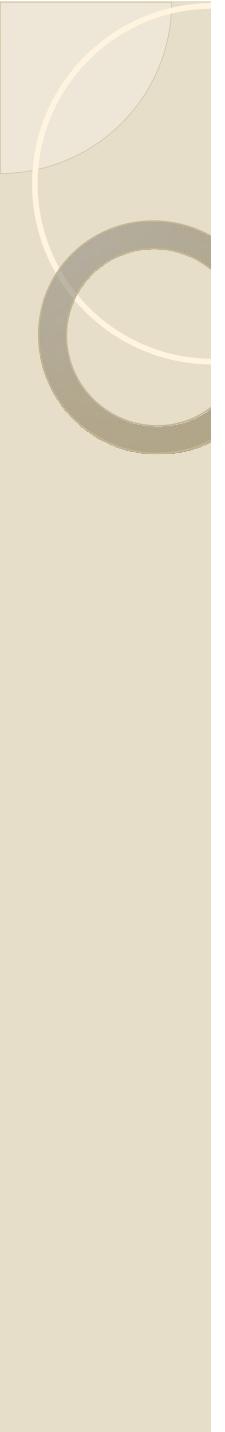
■ Powtarzaj kroki przypisania i poprawy, aż centroidy przestaną się zmieniać





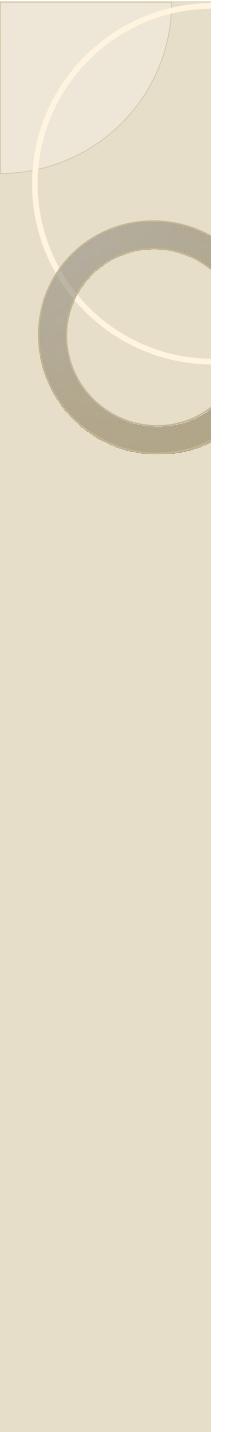
Bisekcja K-średnich

- Algorytm:
 - buduje hierarchię klastrów,
 - w każdym kroku dzieli na 2 klastry.
- Wielokrotne powtórzenie K-średnich dla $K=2$
- Krok początkowy – przypisanie dokumentów do jednego klastra,
- Krok podziału:
 - wybierz najliczniejszy klaster,
 - zastosuj do niego metodę K-średnich ($K=2$)
- Krok wyboru (decyzji):
 - zatrzymaj, jeśli wszystkie klastry mniejsze niż zadany rozmiar,
 - w przeciwnym razie wróć do kroku podziału



Klastering hierarchiczny

- Algorytm podstawowy:
- Początek:
 - start: zbiór N dokumentów do klasyfikacji
 - macierz podobieństwa (odległości) $N \times N$
- Przypisz każdy dokument do swojego klastra
 - Utwórz N klastrów, po jednym dla każdego dokumentu
- Znajdź dwa najbliższe klastry
 - połącz je w jeden klaster,
 - zredukuj liczbę klastrów do $N-1$
- Przelicz odległości między nowym klastrem i każdym ze starych
- Powtórz 2 ostatnie kroki, aż powstanie jeden klaster o rozmiarze N .



Klastering hierarchiczny

- Sposób obliczania odległości klastrów definiuje 3 warianty algorytmu:
 - *single-link*
 - *complete-link*
 - *average-link*
- $dist(c_p, c_r)$: odległość klastrów c_p and c_r
- $dist(d_j, d_l)$: odległość dokumentów d_j and d_l ,
- **Algorytm *Single-Link***

$$dist(c_p, c_r) = \min_{\forall d_j \in c_p, d_l \in c_r} dist(d_j, d_l)$$



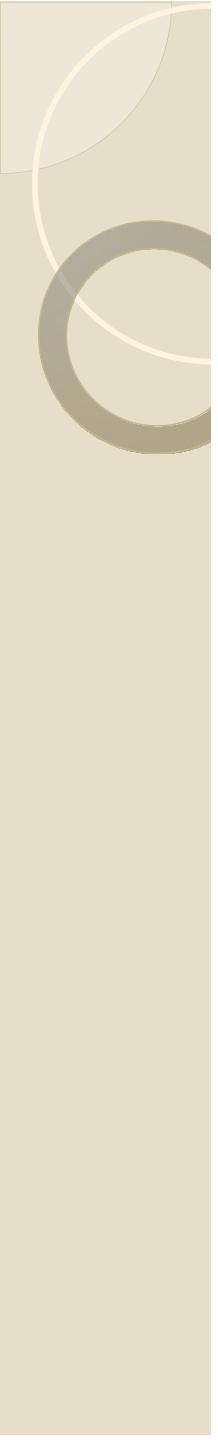
Klastering hierarchiczny

- Algorytm *Complete-Link*

$$dist(c_p, c_r) = \max_{\forall d_j \in c_p, d_l \in c_r} dist(d_j, d_l)$$

- Algorytm *Average-Link*

$$dist(c_p, c_r) = \frac{1}{n_p + n_r} \sum_{d_j \in c_p} \sum_{d_l \in c_r} dist(d_j, d_l)$$



Prosta klasyfikacja tekstu (naive)

- Wejście:
 - D : zbiór dokumentów,
 - $C = \{c_1, c_2, \dots, c_L\}$: zbiór L klas z odpowiednimi etykietami
- Algorytm: przypisz jedną lub więcej klas C do każdego dokumentu D .
 - dopasuj termy dokumentów do etykiet klas,
 - dopuść częściowe dopasowanie do klas
 - popraw pokrycie przez zdefiniowanie alternatywnych etykiet klas – synonimów



Prosta klasyfikacja tekstu (naive)

- **Klasyfikacja przez dopasowanie bezpośrednie**

- Wejście:

- D : zbiór dokumentów,
 - $C = \{c_1, c_2, \dots, c_L\}$: zbiór L klas z odpowiednimi etykietami

- Reprezentacja:

- dokument d_j jako ważony wektor \vec{d}_j
 - klasa c_p jako ważony wektor \vec{c}_p

- Dla każdego dokumentu $d_j \in D$

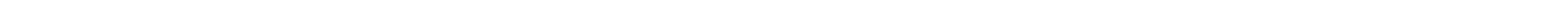
- wyszukaj klasy $c_p \in C$ których etykiety zawierają termy d_j ,
 - dla każdej wyszukanej pary $[d_j, c_p]$ oblicz wektor rankingu

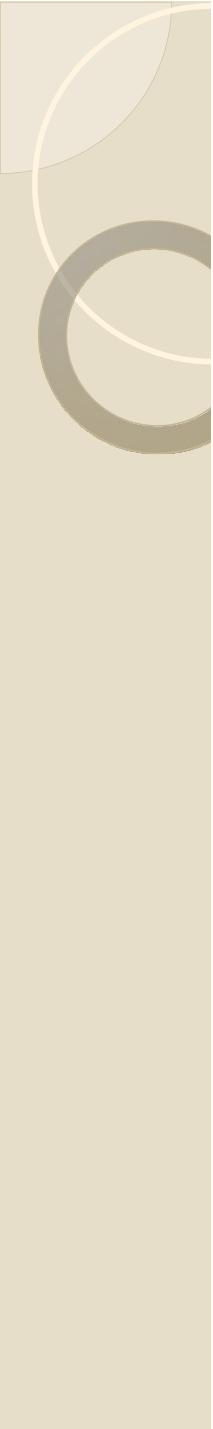
$$sim(d_j, c_p) = \frac{\vec{d}_j \bullet \vec{c}_p}{|\vec{d}_j| \times |\vec{c}_p|}$$



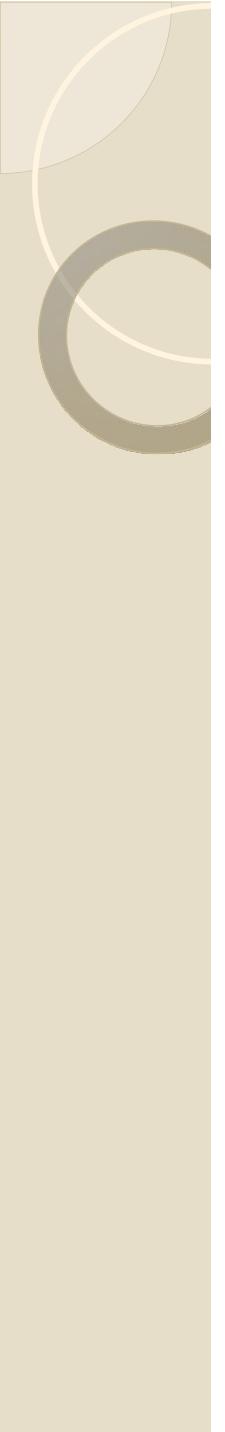
Prosta klasyfikacja tekstu (naive)

- Wybierz dla d_j klasy c_p z najwyższymi wartościami $sim(d_j, c_p)$





Algorytmy nadzorowane

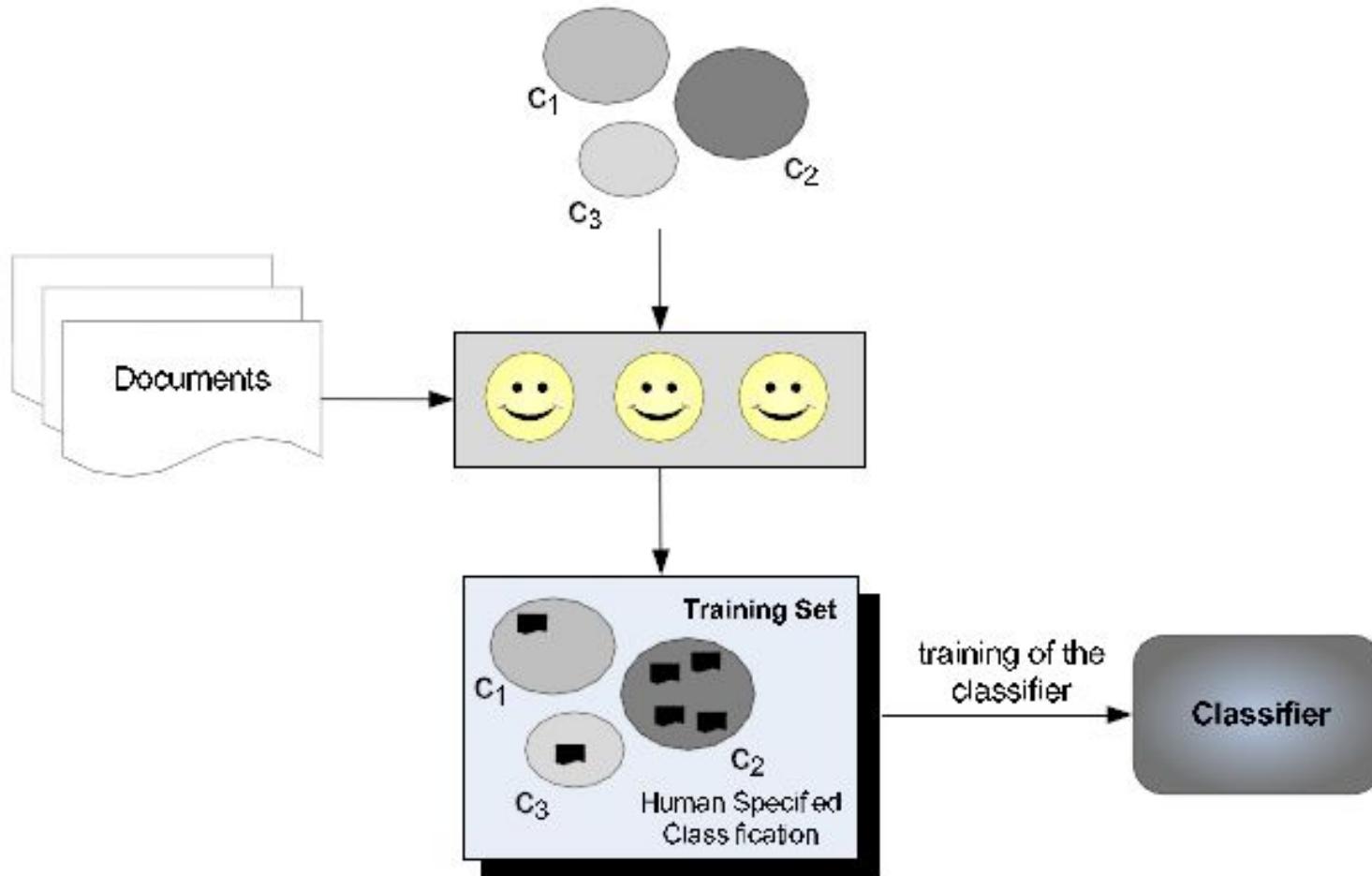


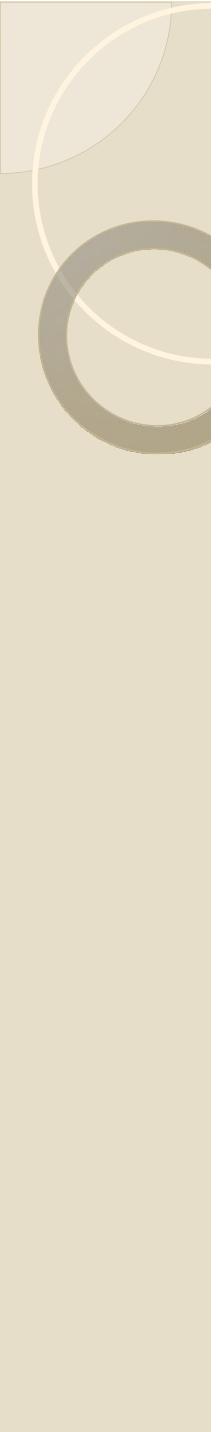
Algorytmy nadzorowane

- Wymagają zbiorów uczących
- $D_t \in D$: zbiór dokumentów uczących
- $T: D_t \times C \rightarrow \{0,1\}$: funkcja trenująca
Przypisuje do każdej pary $[d_j, c_p]$, $d_j \in D_t$ oraz $c_p \in C$ wartość:
 - 1, gdy $d_j \in c_p$ wg. oceny specjalistów,
 - 0, gdy $d_j \notin c_p$ wg. oceny specjalistów
- Funkcja ucząca T jest stosowana do dostrojenia klasyfikatora

Algorytmy nadzorowane

■ Faza treningu (uczenia się)





Algorytmy nadzorowane

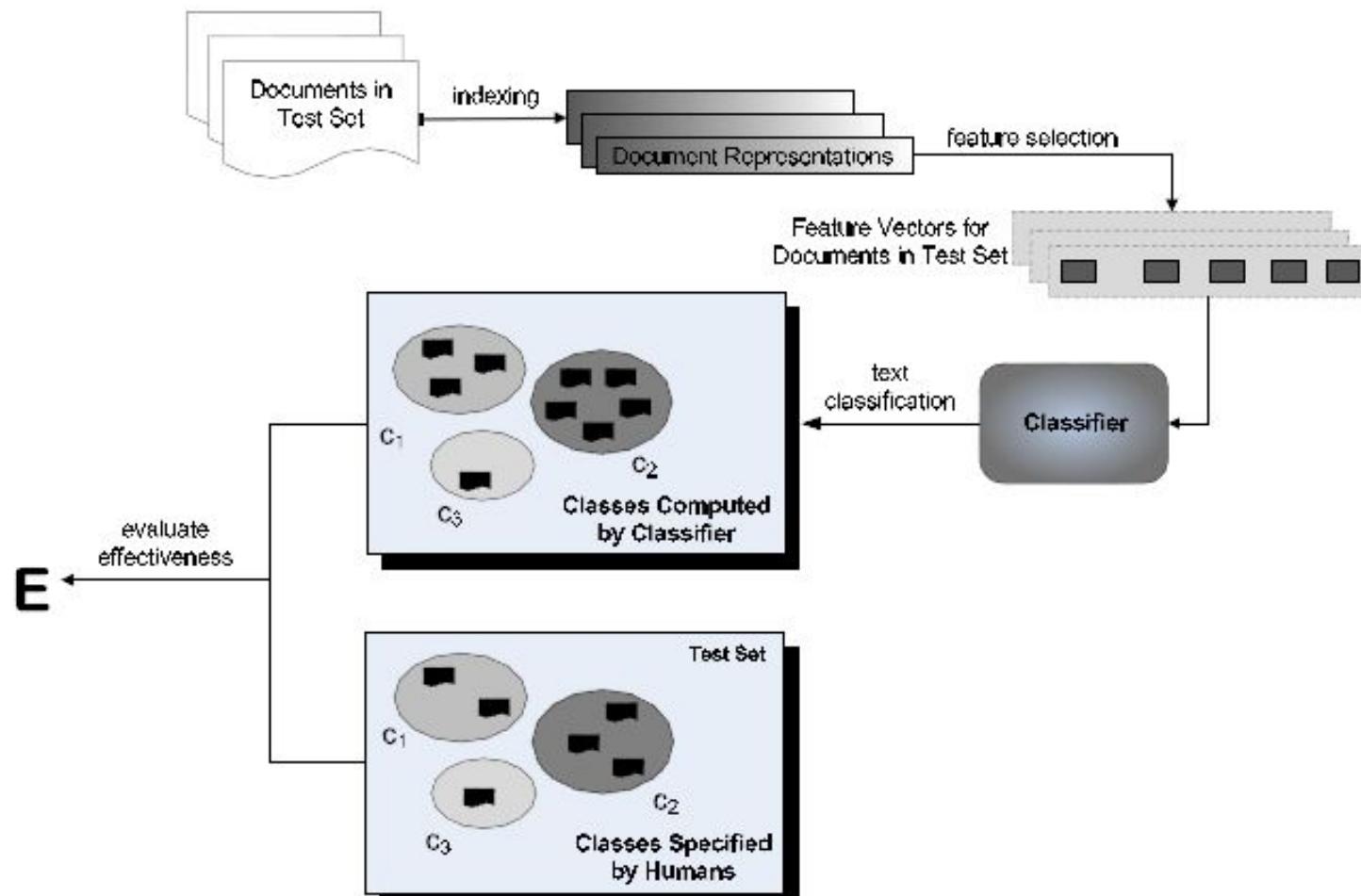
- Aby ocenić klasyfikator używa się zbioru testowego – podzbioru dokumentów nie występujących w zbiorze uczącym

- Klasy dokumentów określają specjalistyci

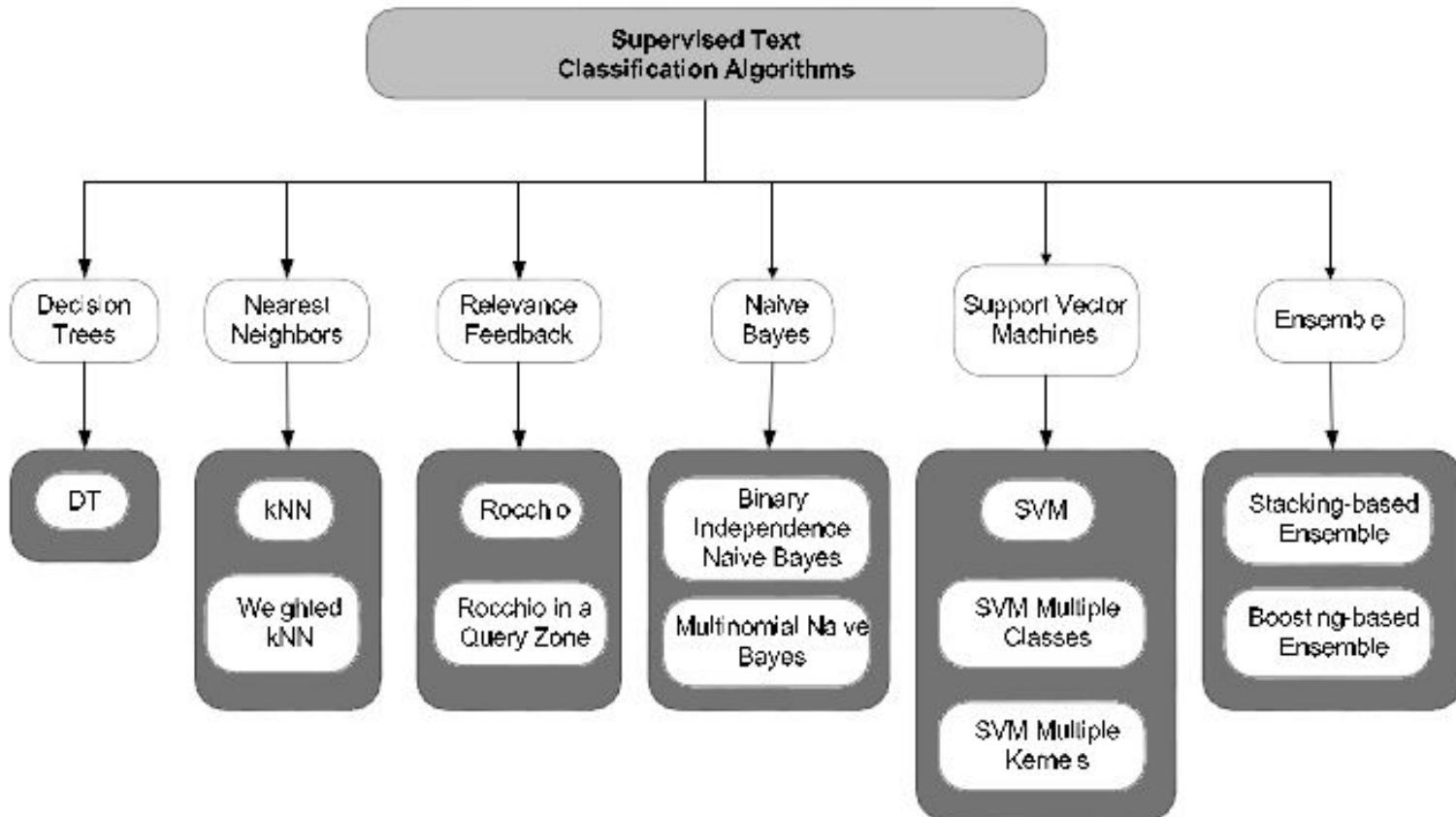
- Weryfikacja algorytmu polega na:
 - klasyfikacji zbioru testowego przy pomocy algorytmu,
 - porównaniu uzyskanych klas z podanymi przez specjalistów.

Algorytmy nadzorowane

■ Klasyfikacja i ocena



Algorytmy nadzorowane



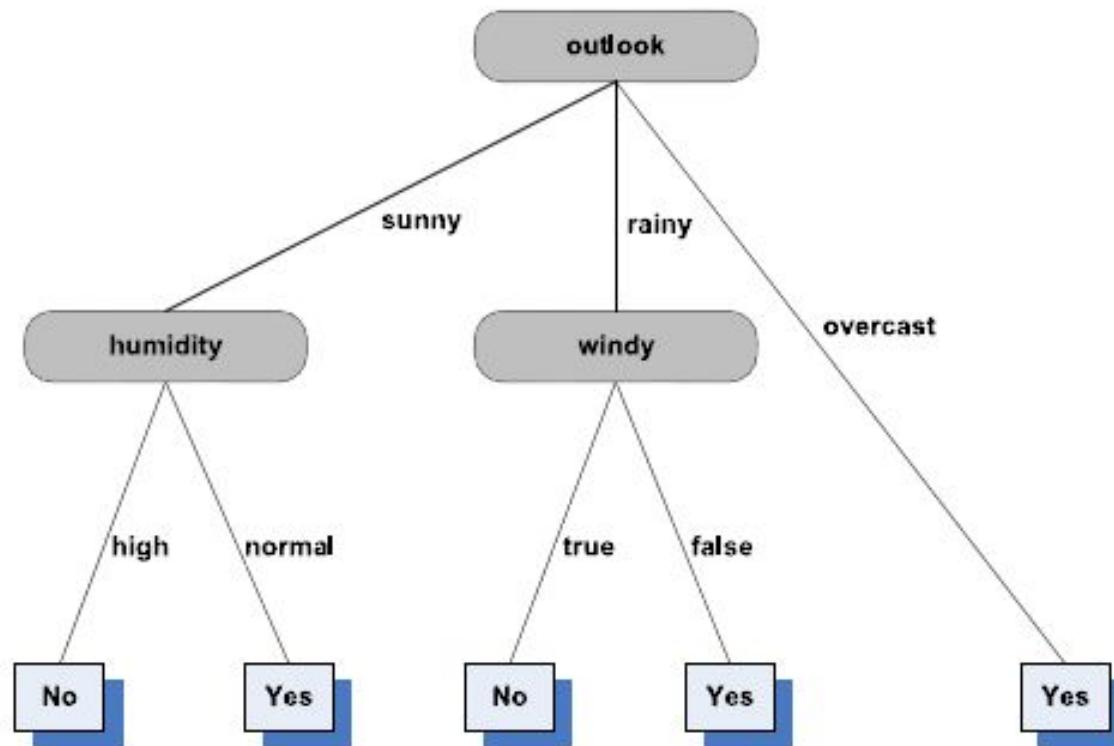
Drzewa decyzyjne

- Zbiór uczący jest używany do budowy reguł klasyfikacji
- Organizacja klasyfikatora w formie drzewa grafu
- Reguły klasyfikacji łatwe do interpretacji przez człowieka
- Przykładowa baza danych:

	Id	Play	Outlook	Temperature	Humidity	Windy
Training set	1	yes	rainy	cool	normal	false
	2	no	rainy	cool	normal	true
	3	yes	overcast	hot	high	false
	4	no	sunny	mild	high	false
	5	yes	rainy	cool	normal	false
	6	yes	sunny	cool	normal	false
	7	yes	rainy	cool	normal	false
	8	yes	sunny	hot	normal	false
	9	yes	overcast	mild	high	true
	10	no	sunny	mild	high	true
Test Instance	11	?	sunny	cool	high	false

Drzewa decyzyjne

- Przewidywanie atrybutu Play



Drzewa decyzyjne

- Węzły wewnętrzne → nazwy atrybutów
- Krawędzie → wartości atrybutów
- Trawers drzewa decyzyjnego → wartość atrybutu “Play”.
- $(\text{Outlook} = \text{sunny}) \wedge (\text{Humidity} = \text{high}) \rightarrow (\text{Play} = \text{no})$

	Id	Play	Outlook	Temperature	Humidity	Windy
Test Instance	11	?	sunny	cool	high	false

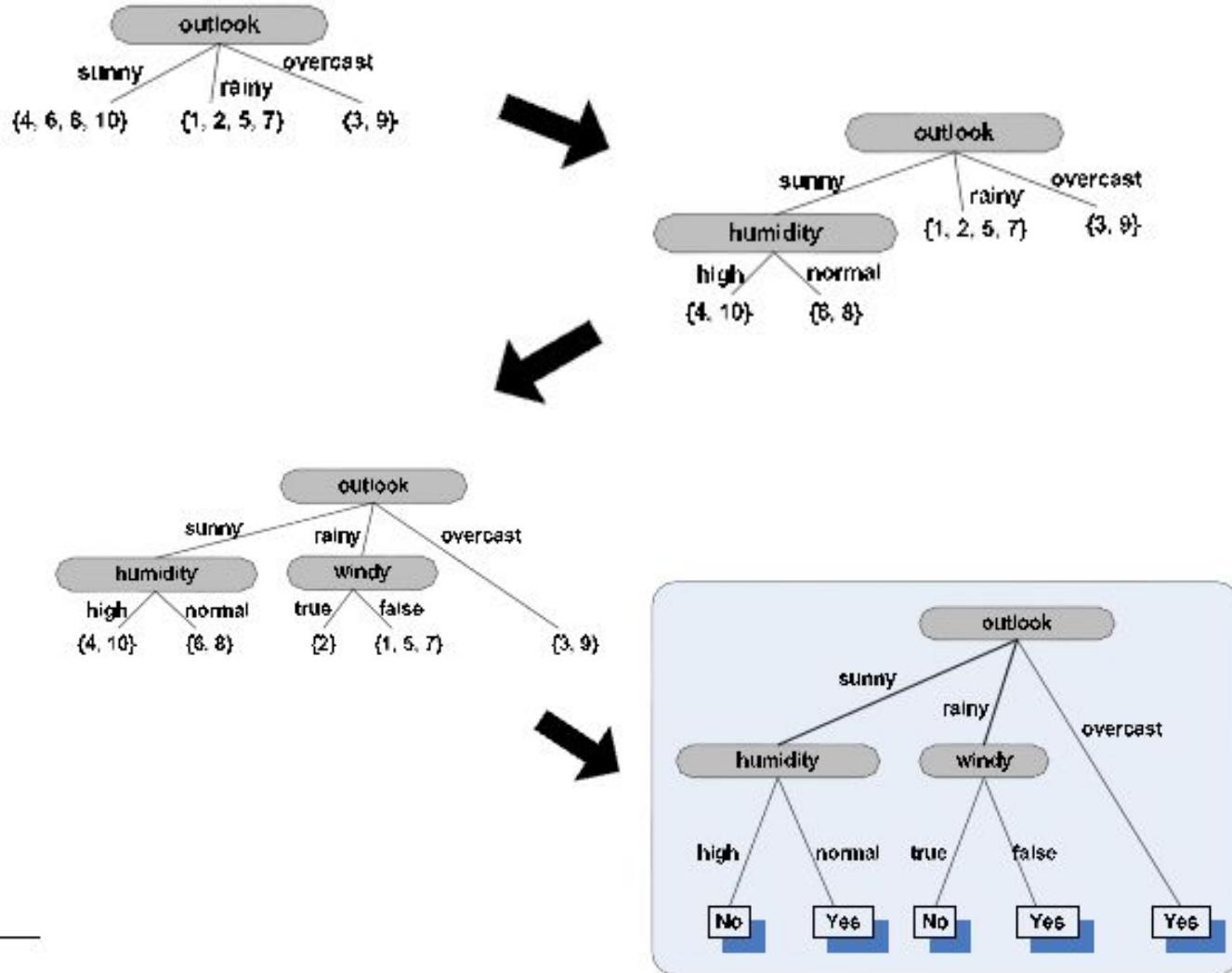
- Predykcje bazują na zadanych przykładach
- Nowe przykłady naruszające wzorzec prowadzą do błędnych przewidywań
- Przykładowa baza danych stanowi zbiór uczący, determinujący drzewo decyzyjne DT (Decision Tree)

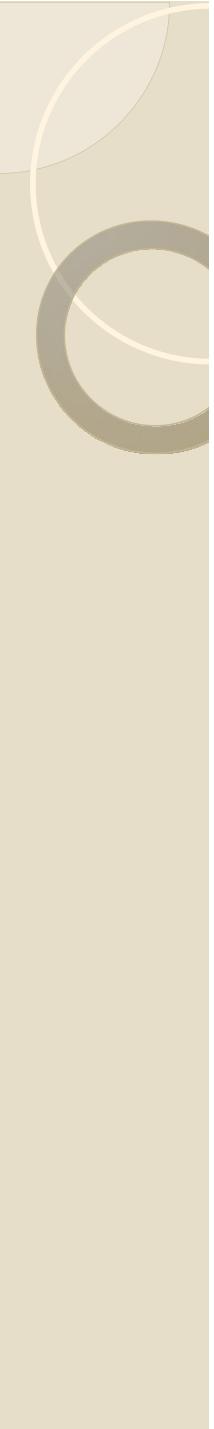


Drzewa decyzyjne

- DT buduje się z bazy danych poprzez rekursywne podziały
 - Wybiera się dowolny atrybut różny od „Play” jako korzeń drzewa
 - Pozostałe atrybuty dzielą krotki danych na podzbiory,
 - Dla każdego podzbioru krotek wybiera się kolejny atrybut dzielący
-

Drzewa decyzyjne





Drzewa decyzyjne

- Proces podziału zależy od kolejności rozpatrywania atrybutów
- Drzewa decyzyjne mogą być niezrównoważone; drzewa zrównoważone lepiej przewidują wartości atrybutów
- Reguła kciuka: wybieraj atrybuty, które pozwalają zmniejszyć średnią długość ścieżki
- Drzewa do klasyfikacji dokumentów:
 - term indeksujący jest skojarzony z węzłami wewnętrznymi,
 - klasy dokumentów są skojarzone z liśćmi,
 - binarne predykaty wskazujące obecność/nieobecność termów indeksujących są skojarzone z krawędziami.



Drzewa decyzyjne

- Niech:

- $K = \{k_1, k_2, \dots, k_t\}$: zbiór termów indeksujących dokumentów
- C : zbiór klas dokumentów
- P : zbiór predykatów logicznych termów indeksujących

- Drzewo klasyfikacji:

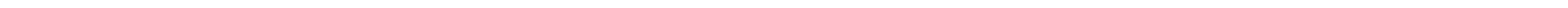
$$DT = (V, E; r; l_I, l_L, l_E)$$

- (V, E, r) : drzewo z korzeniem r ,
- $l_I: I \rightarrow K$: funkcja kojarząca termy indeksowe K z węzłami wewnętrznymi I ,
- $l_L: \bar{I} \rightarrow C$: funkcja kojarząca klasy $c_p \in C$ z liśćmi drzewa,
- $l_E: E \rightarrow P$: funkcja kojarząca predykaty logiczne P z krawędziami E .



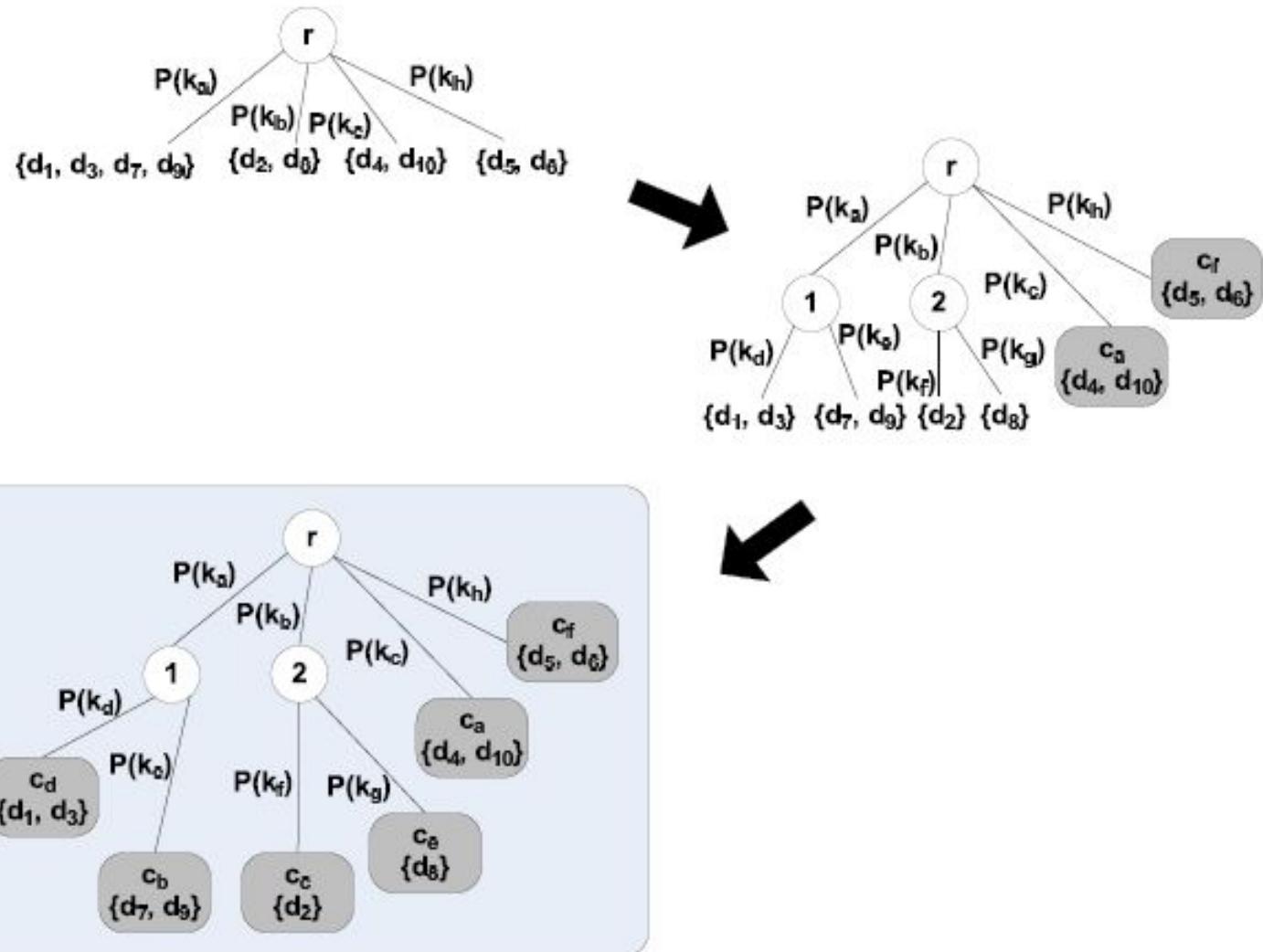
Drzewa decyzyjne

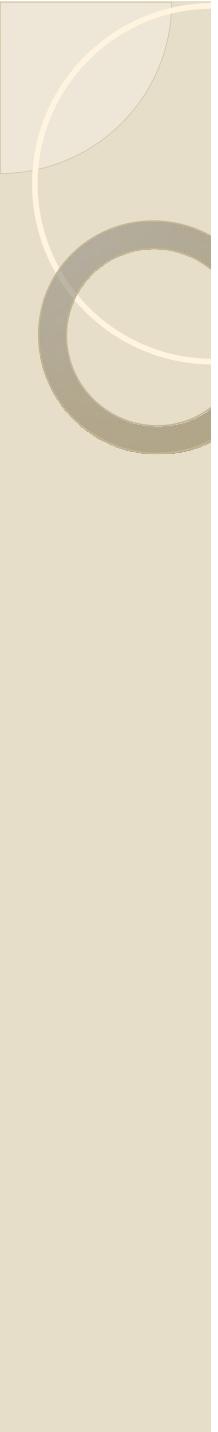
- Tworzenie drzewa przez rekursywne podziały:
 - **krok 1:** skojarzenie dokumentów z korzeniem,
 - **krok 2:** wybierz termy indeksujące zapewniające dobry podział
 - **krok 3:** powtarzaj aż drzewo rozwinie się w pełni



Drzewa decyzyjne

- Termy k_a, k_b, k_c, k_h - termy wybrane do 1 podziału



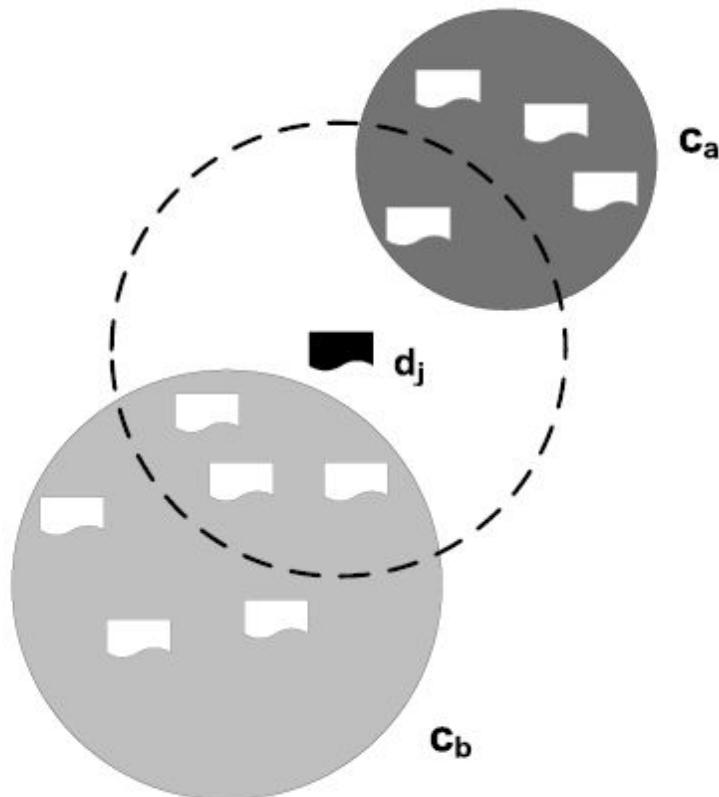


Drzewa decyzyjne

- Wybór termów rozdzielających odbywa się w oparciu o entropię lub tzw. wzmocnienie informacyjne
- Wybór termów z dużym wzmocnieniem informacyjnym
 - zwiększa liczbę gałęzi na danym poziomie drzewa,
 - zmniejsza liczbę dokumentów w podzbiorach wynikowych,
 - buduje mniejsze i mniej złożone drzewa decyzyjne.
- Jeżeli jakiś dokument nie zawiera termów używanych w budowie drzewa decyzyjnego nie wykorzystuje się go przy tej budowie

Klasyfikator k-NN

- Klasyfikacja na żądanie (zwłoczna) – wykonywana gdy prezentowany jest nowy dokument d_j .



Klasyfikator k-NN

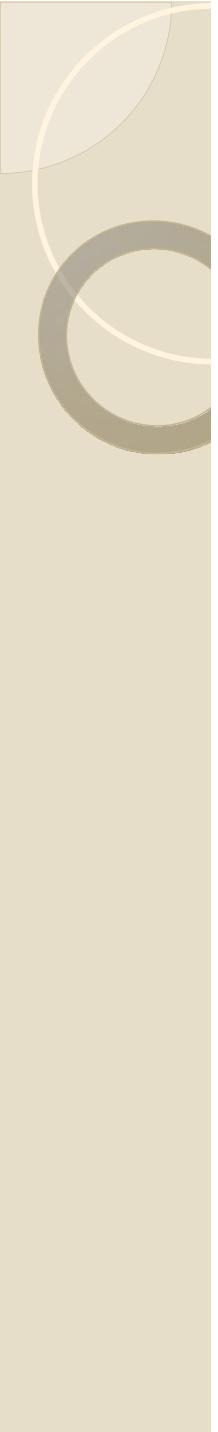
- Określa się k najbliższych sąsiadów dokumentu d_j w zbiorze uczącym.
- Klasy tych sąsiadów używa się do ustalenia klasy dokumentu d_j .
- Każdej parze dokument-klasa $[d_j, c_p]$ przypisuje się wartość:

$$S_{d_j, c_p} = \sum_{d_t \in N_k(d_j)} similarity(d_j, d_t) \times T(d_t, c_p)$$

gде

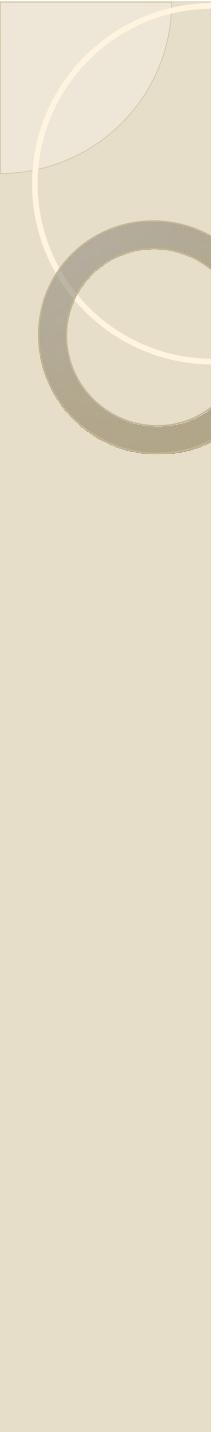
- $N_k(d_j)$: zbiór k najbliższych sąsiadów dokumentu d_j w zbiorze uczącym
- $similarity(d_j, d_t)$: formuła podobieństwa dokumentów,
- $T(d_t, c_p)$: funkcja zbioru uczącego równa 1 gdy $d_t \in c_p$ albo 0 w przeciwnym wypadku





Klasyfikator k-NN

- Klasyfikator przypisuje dokumentowi d_j klasę (klasy) c_p z najwyższym wynikiem
- Klasyfikator musi obliczyć odległości dokumentu d_j do każdego z elementów zbioru uczącego w zadanym otoczeniu
- Innym problemem jest wybór najlepszej wartości k



Klasyfikator Rocchio

- Klasyfikator Rocchio:
 - modyfikuje zapytanie na bazie odpowiedzi użytkownika
 - tworzy nowe zapytanie lepiej wyrażające potrzeby użytkownika
 - może być adaptowany do klasyfikacji tekstu
- Zbiór uczący pełni rolę informacji zwrotnej od użytkownika
 - termy dokumentów uczących z danej klasy c_p dają dodatnie sprzężenie zwrotne,
 - termy dokumentów uczących spoza danej klasy c_p dają ujemne sprzężenie zwrotne.
- Sprzężenie informacyjne zbierane jest przez wektor centroidy
- Nowy dokument jest klasyfikowany na bazie odległości od tej centroidy

Klasyfikator Rocchio

- Dokument d_j :

$$\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$$

- $w_{i,j}$: waga termu k_i w dokumencie d_j ,
- t : rozmiar słownika.

- Klasyfikator Rocchio dla klasy c_p :

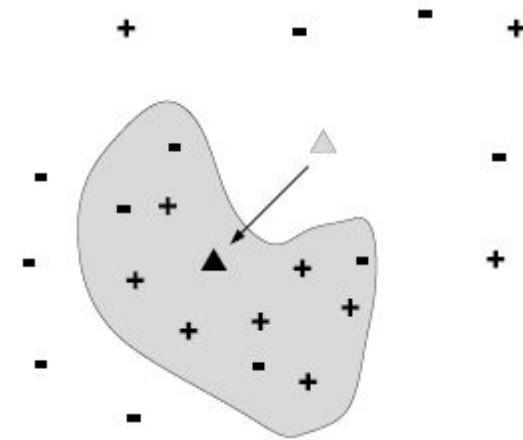
$$\vec{c}_p = \frac{\beta}{n_p} \sum_{d_j \in c_p} \vec{d}_j - \frac{\gamma}{N_t - n_p} \sum_{d_l \notin c_p} \vec{d}_l$$

gdzie

- n_p : liczba dokumentów w klasie c_p ,
- N_t : całkowita liczba dokumentów w zbiorze uczącym.

Klasyfikator Rocchio

- +: termy w klasie c_p ,
- : termy poza klasą c_p .



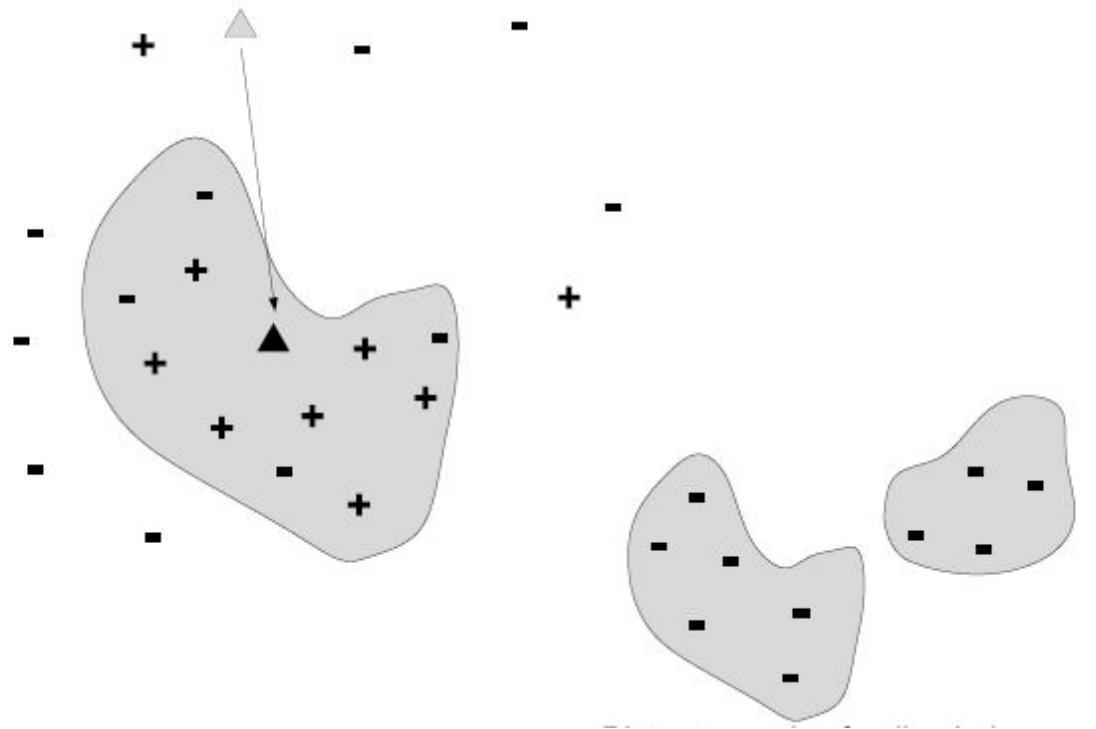
- Klasyfikator przypisuje każdej parze $[d_j, c_p]$ wartość:

$$S(d_j, c_p) = |\vec{c}_p - \vec{d}_j|$$

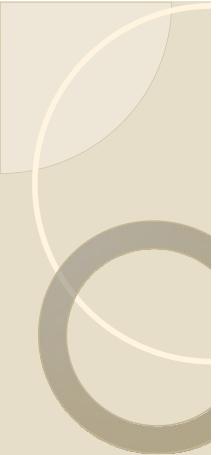
- Klasa z najmniejszą wartością S jest przypisana do d_j .

Klasyfikator Rocchio

- Problem: ujemne sprzężenie przemieszcza niekorzystnie centroidę



- Wykorzystuje się jedynie „najbardziej pozytywne” dokumenty z ujemnym sprzężeniem informacyjnym



Prosty klasyfikator Bayesa

- Każdej parze $[d_j, c_p]$ przypisuje się prawdopodobieństwo $P(c_p|\vec{d}_j)$ (tw. Bayesa)

$$P(c_p|\vec{d}_j) = \frac{P(c_p) \times P(\vec{d}_j|c_p)}{P(\vec{d}_j)}$$

- $P(\vec{d}_j)$: prawdopodobieństwo wylosowania dokumentu \vec{d}_j
- $P(c_p)$: prawdopodobieństwo wylosowania dokumentu w klasie c_p
- Nowym dokumentom przypisuje się klasy o największym prawdopodobieństwie
- Aby uprościć wyznaczenie $P(\vec{d}_j|c_p)$ zakłada się niezależność termów indeksujących

Prosty klasyfikator Bayesa

- Dokument d_j jest reprezentowany przez wektor wag binarnych

$$\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$$

$$w_{i,j} = \begin{cases} 1 & \text{if term } k_i \text{ occurs in document } d_j \\ 0 & \text{otherwise} \end{cases}$$

- Do każdej pary $[d_j, c_p]$ przypisuje się wartość

$$S(d_j, c_p) = \frac{P(c_p | \vec{d}_j)}{P(\bar{c}_p | \vec{d}_j)} \sim \frac{P(\vec{d}_j | c_p)}{P(\vec{d}_j | \bar{c}_p)}$$

Prosty klasyfikator Bayesa

- Przy założeniu niezależności termów

$$P(\vec{d}_j | c_p) = \prod_{k_i \in \vec{d}_j} P(k_i | c_p) \times \prod_{k_i \notin \vec{d}_j} P(\bar{k}_i | c_p)$$

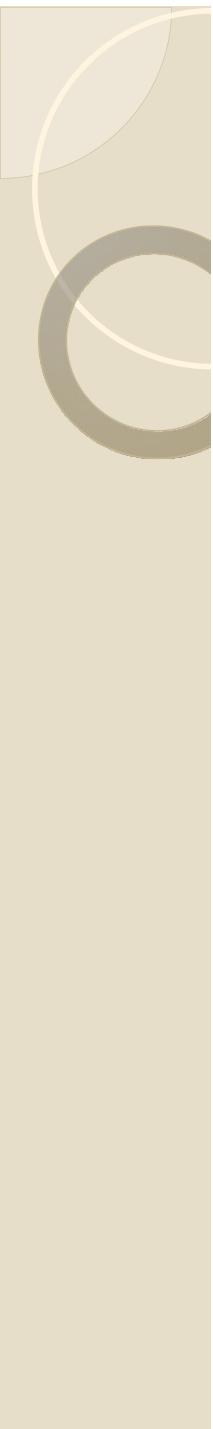
$$P(\vec{d}_j | \bar{c}_p) = \prod_{k_i \in \vec{d}_j} P(k_i | \bar{c}_p) \times \prod_{k_i \notin \vec{d}_j} P(\bar{k}_i | \bar{c}_p)$$

- Stąd

$$S(d_j, c_p) \sim \sum_{k_i} w_{i,j} \left(\log \frac{p_{iP}}{1 - p_{iP}} + \log \frac{1 - q_{iP}}{q_{iP}} \right)$$

$$p_{iP} = P(k_i | c_p)$$

$$q_{iP} = P(k_i | \bar{c}_p)$$



Prosty klasyfikator Bayesa

- Prawdopodobieństwa p_{iP} oraz q_{iP} są wyznaczane ze zbioru D_t dokumentów uczących.

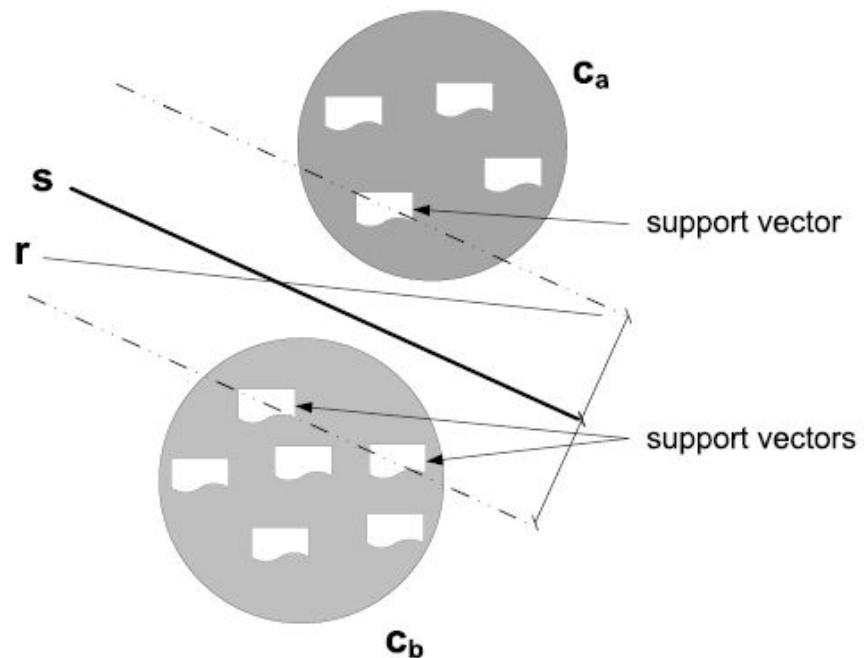
$$p_{iP} = \frac{1 + \sum_{d_j | d_j \in D_t \wedge k_i \in d_j} P(c_p | d_j)}{2 + \sum_{d_j \in D_t} P(c_p | d_j)} = \frac{1 + n_{i,p}}{2 + n_p}$$
$$q_{iP} = \frac{1 + \sum_{d_j | d_j \in D_t \wedge k_i \in d_j} P(\bar{c}_p | d_j)}{2 + \sum_{d_j \in D_t} P(\bar{c}_p | d_j)} = \frac{1 + (n_i - n_{i,p})}{2 + (N_t - n_p)}$$

- $n_{i,p}, n_i, n_p, N_t$: jak w modelu probabilistycznym (N_t – liczba dokumentów testowych, n_i – liczba dokumentów z i-tym termem, $n_{i,p}$ – liczba dokumentów z i-tym termem w klasie c_p).
- $P(c_p | d_j) \in \{0,1\}$ oraz $P(\bar{c}_p | d_j) \in \{0,1\}$: wzięte ze zbioru uczącego.
- Klasyfikator przypisuje każdemu dokumentowi d_j klasę z największą wartością $S(d_j, c_p)$.

Klasyfikator SVM

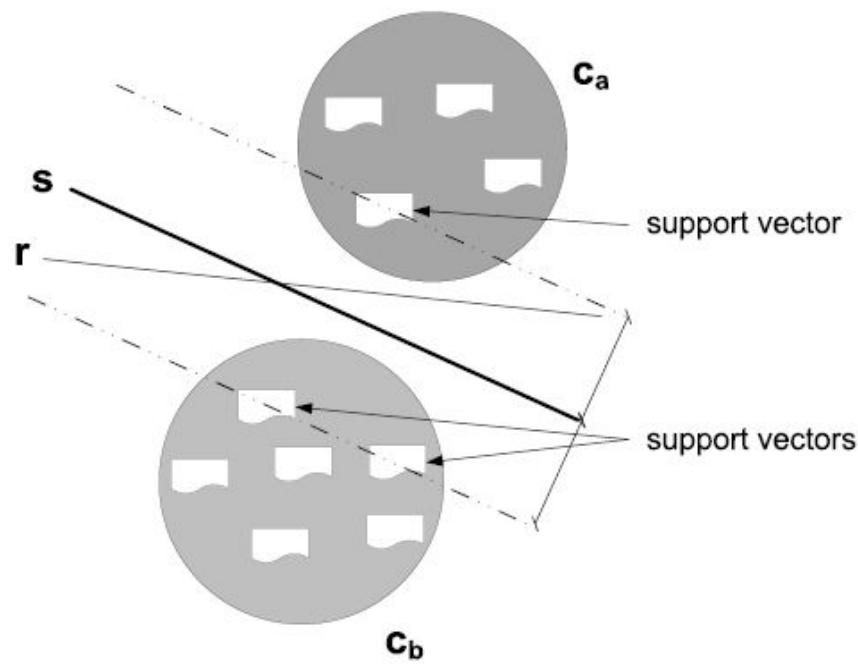
- SVM – metoda wektorów nośnych (*Support Vector Machine*)
- Hiperpłaczczyna s rozdzielająca klasy maksymalizuje odległości do najbliższych dokumentów w każdej z rozdzielanych klas (c_a, c_b na rysunku).

SVM – zakłada liniową
separowalność klas
dokumentów



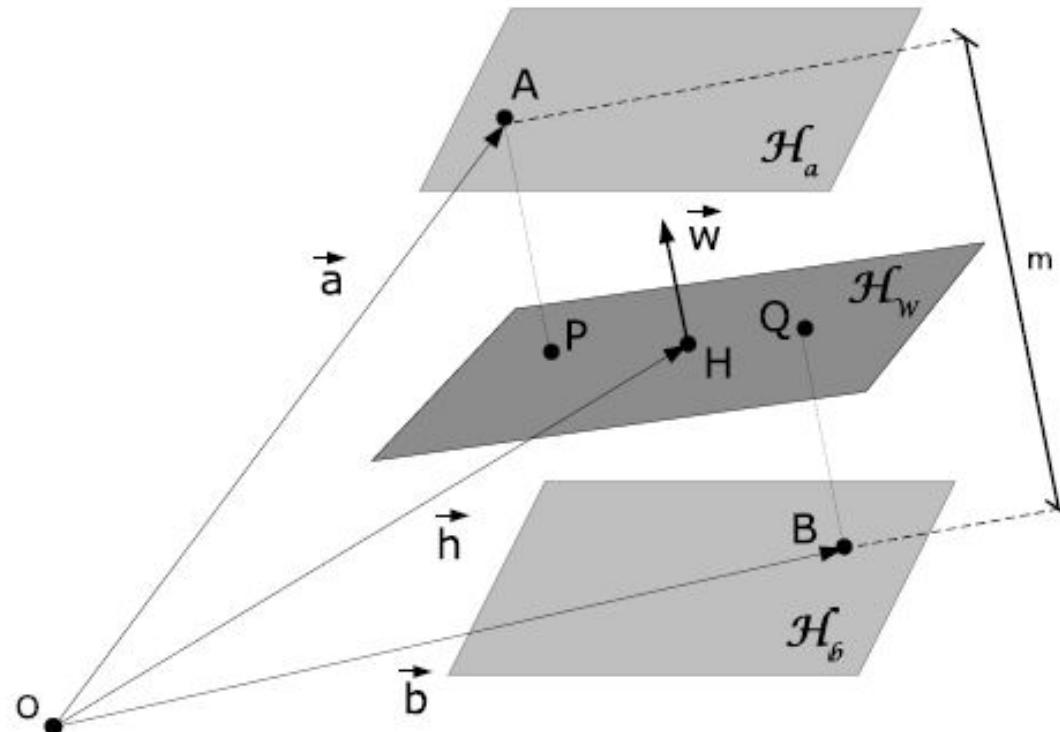
Klasyfikator SVM

- Hiperpłaszczyznę decyzyjną s wybiera się ze zbioru płaszczyzn równoległych do hiperpłaszczyzn ograniczających (support vectors) i umieszczonych pomiędzy nimi.



Klasyfikator SVM

- Problem optymalizacji SVM: dla danych wektorów \vec{a} i \vec{b} znaleźć płaszczyznę H_w , która maksymalizuje margines m .



Klasyfikator SVM

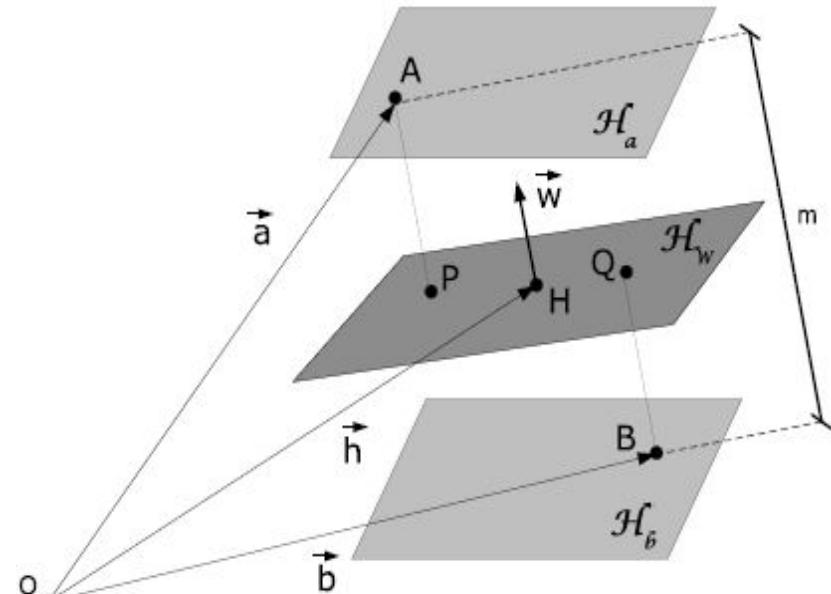
- Hiperpłaszczyzna H_w jest określona przez wektor \vec{h} i prostopadły wektor \vec{w} , które nie są z góry znane

\overline{AP} : Odległość punktu A do hiperpłaszczyzny H_w .

$$\overline{AP} = \frac{\vec{a}\vec{w} + k}{|\vec{w}|}$$

\overline{BQ} : Odległość punktu B do hiperpłaszczyzny H_w .

$$\overline{BQ} = -\frac{\vec{b}\vec{w} + k}{|\vec{w}|}$$



Klasyfikator SVM

- Margines m niezależny od rozmiaru \vec{w} :

$$m = \overline{AP} + \overline{BQ}$$

- Założenia ograniczające dla \vec{w} :

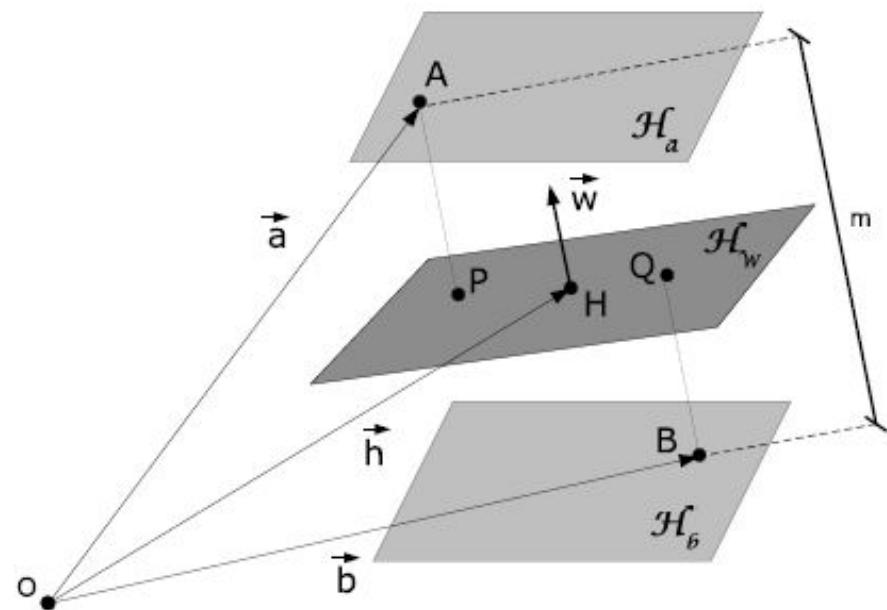
$$\vec{a}\vec{w} + k = 1$$

$$\vec{b}\vec{w} + k = -1$$

- Stąd:

$$m = \frac{1}{|\vec{w}|} + \frac{1}{|\vec{w}|}$$

$$m = \frac{2}{|\vec{w}|}$$



Klasyfikator SVM

- Oznaczenia:

- $T = \{\dots, [c_j, \vec{z}_j], [c_{j+1}, \vec{z}_{j+1}], \dots\}$: zbiór uczący,
- c_j : klasa związana z punktem \vec{z}_j reprezentującym dokument d_j .

- Problem optymalizacji SVM:

- maksymalizacja $m = 2/|\vec{w}|$
przy ograniczeniach

$$\vec{w}\vec{z}_j + b \geq +1 \text{ if } c_j = c_a$$

$$\vec{w}\vec{z}_j + b \leq -1 \text{ if } c_j = c_b$$

- wektory nośne spełniają warunki równości w powyższym układzie równań

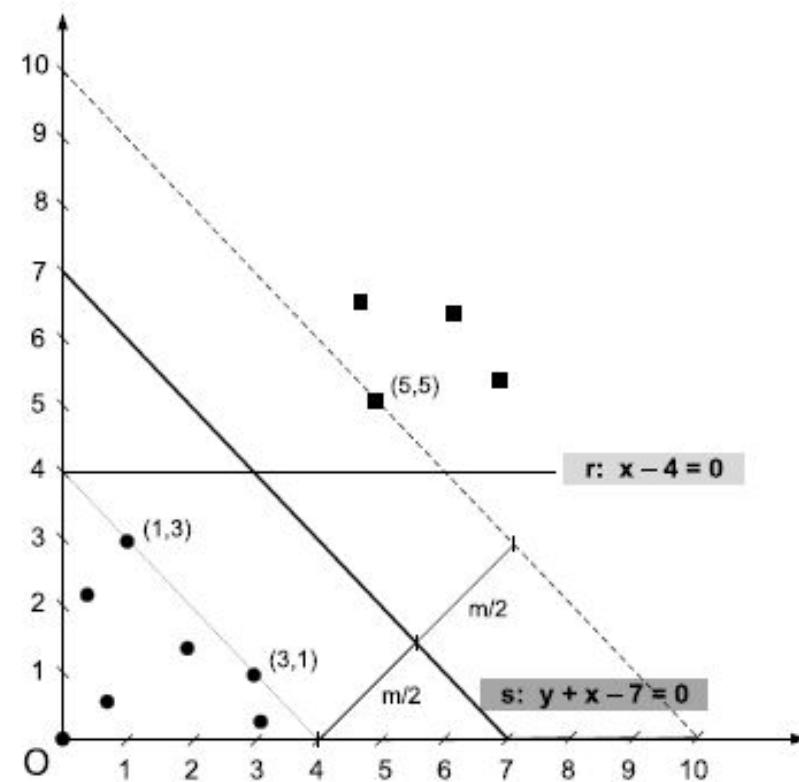
Klasyfikator SVM

- Przykład:
- maksymalizować wartość
 $m = 2/|\vec{w}|$
- przy ograniczeniach
 $\vec{w} \cdot (5, 5) + b = +1$
 $\vec{w} \cdot (1, 3) + b = -1$
- $m = 3\sqrt{2}$ - odległość między hiperpłaszczyznami,
- $|\vec{w}| = \sqrt{x^2 + y^2}$
- Stąd:

$$3\sqrt{2} = 2/\sqrt{x^2 + y^2}$$

$$5x + 5y + b = +1$$

$$x + 3y + b = -1$$



Rozwiązanie:

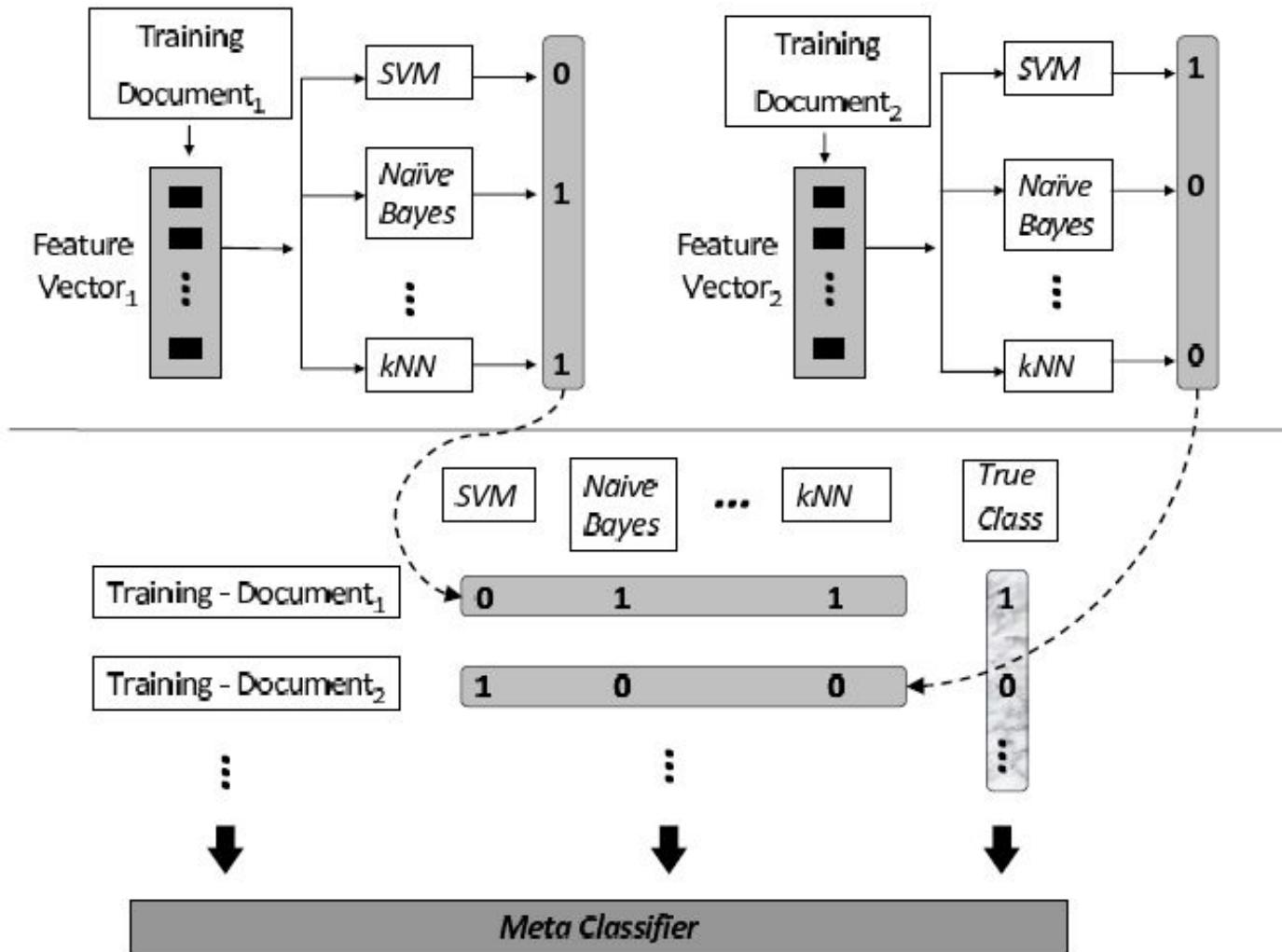
$$y + x - 7 = 0$$



Klasyfikator SVM

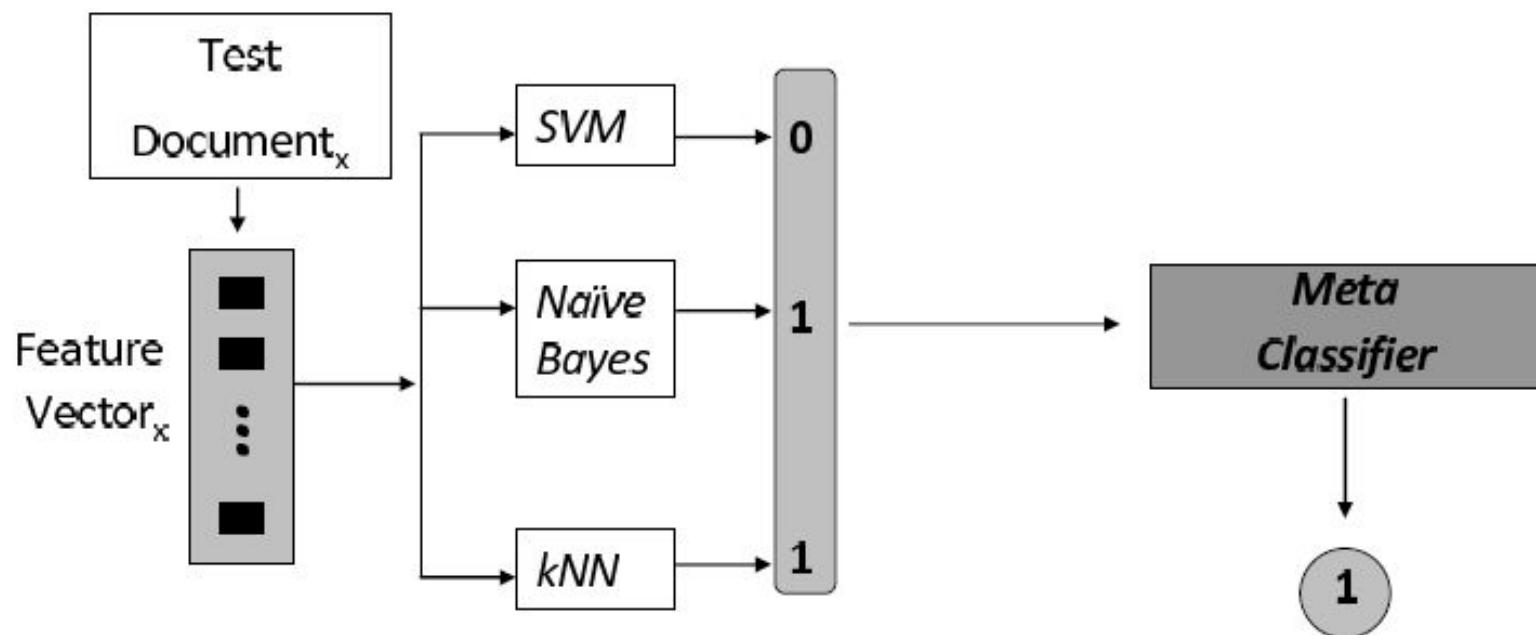
- Klasifikacja dokumentu d_j (wektora \vec{z}_j)
$$f(\vec{z}_j) = \text{sign}(\vec{w}\vec{z}_j + b)$$
- Nowy dokument d_j jest klasifikowany
 - do klasy c_a : gdy $\vec{w}\vec{z}_j + b > 1$,
 - do klasy c_b : gdy $\vec{w}\vec{z}_j + b < -1$.
- Przy wielu klasach każda wybrana klasa c_p jest oddzielana od pozostałych
- Wybrane klasy d_j – te, które zapewniają największe marginesy m względem innych klas

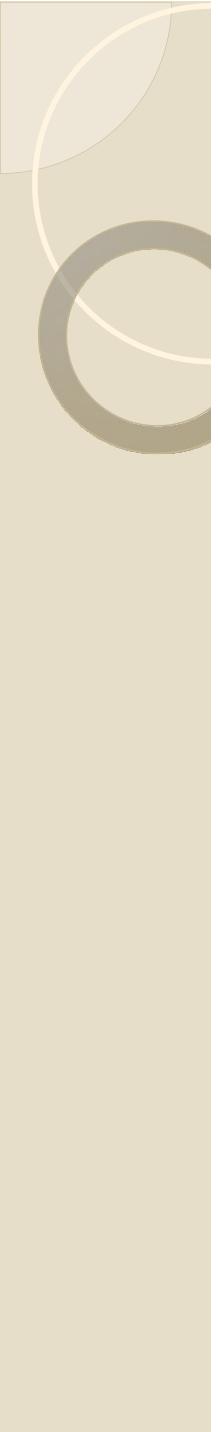
Klasyfikatory zespołowe (złożone)



Klasyfikatory zespołowe (złożone)

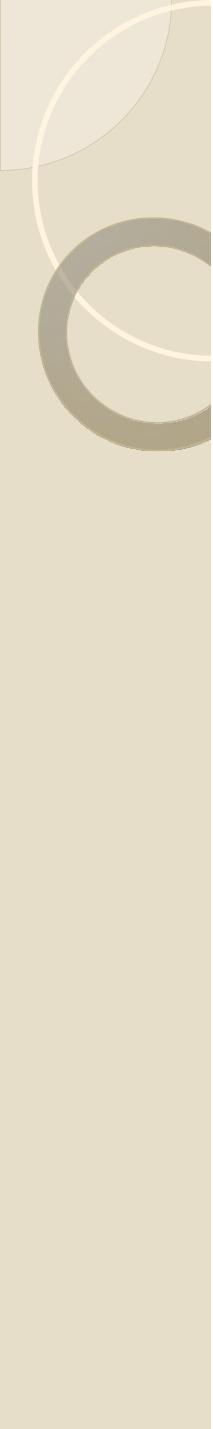
- Metoda stosowa: funkcja ucząca składa wyniki przewidływań poszczególnych klasyfikatorów składowych z określonymi wagami lub wybiera najlepszy





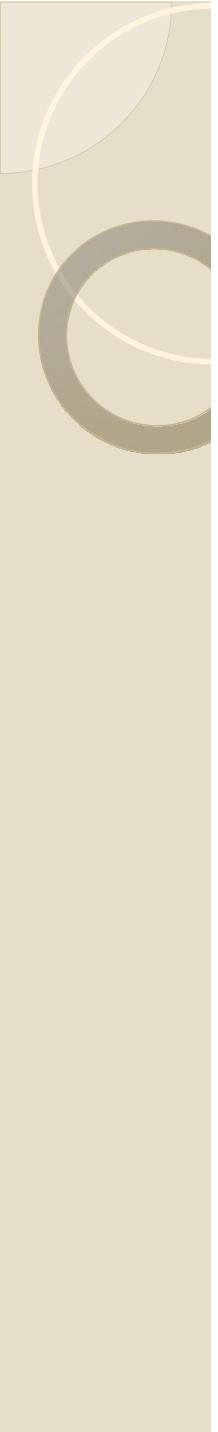
Klasyfikatory zespołowe (złożone)

- Metoda wzmacniania (boosting): łączone klasyfikatory są budowane w wielu iteracjach tą samą metodą uczenia
- W każdej iteracji
 - dokument w zbiorze uczącym ma przypisaną wagę,
 - wagi źle sklasyfikowanych dokumentów zwiększają się po kolejnych iteracjach
- Po n iteracjach
 - wyjścia klasyfikatorów są łączone w sumę ważoną,
 - wagi są związane z błędami estymacji każdego klasyfikatora



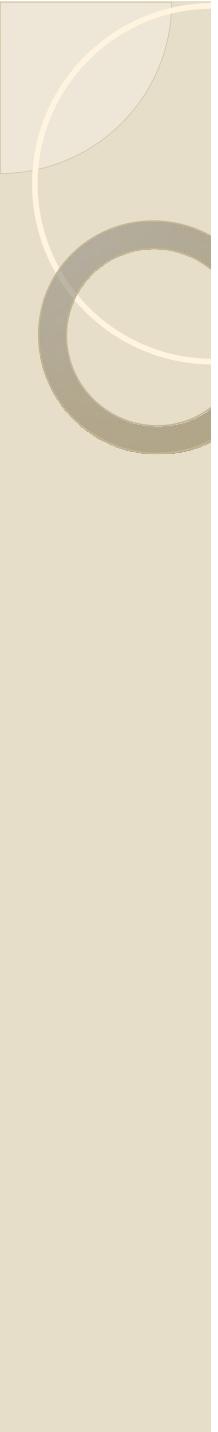
Eksploracja danych w Internecie

Indeksowanie i wyszukiwanie



Eksploracja danych w Internecie

- **Index:** struktura danych tworzona na podstawie tekstu aby przyspieszyć wyszukiwanie
- Wydajność indeksowania w systemach IR mierzy się przez:
 - **czas indeksowania:** czas niezbędny do budowy indeksu,
 - **przestrzeń indeksowania:** przestrzeń pamięci używana podczas generacji indeksu
 - **pamięć indeksu:** pamięć wymagana do zapisania indeksu
 - **opóźnienie zapytania (latency):** czas pomiędzy pojawieniem się zapytania i generacją odpowiedzi
 - **przepustowość zapytania:** średnia ilość pytań przetwarzanych na sekundę
- Podczas modyfikacji tekstu indeks także podlega zmianom



Eksploracja danych w Internecie

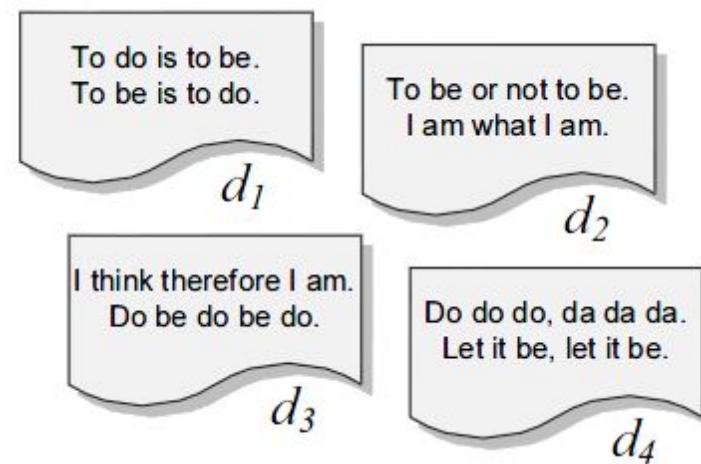
- Bieżąca technologia indeksowania nie jest dobrze przystosowana do bardzo częstych zmian w zbiorze tekstów
 - Zbiory pół-statyczne (*semi-static*) są modyfikowane w regularnych, stałych przedziałach czasowych (np. raz dziennie)
 - Większość zbiorów tekstowych w sieci jest indeksowana pół-statycznie
-
- **Indeks odwrócony:** mechanizm indeksowania zbiorów tekstowych przyspieszający wyszukiwanie
 - Struktura tego indeksu jest złożona z 2 elementów: **słownika i wystąpień (occurrences)**
 - Słownik: zbiór różnych słów tekstu; dla każdego elementu słownika indeks zapamiętuje zawierające go dokumenty (indeks odwrócony)

Eksploracja danych w Internecie

- Macierz **term-dokument** jest rzadka i wymaga dużo pamięci do przechowywania

Vocabulary	n_i	d_1	d_2	d_3	d_4
to	2	4	2	-	-
do	3	2	-	3	3
is	1	2	-	-	-
be	4	2	2	2	2
or	1	-	1	-	-
not	1	-	1	-	-
I	2	-	2	2	-
am	2	-	2	1	-
what	1	-	1	-	-
think	1	-	-	1	-
therefore	1	-	-	1	-
da	1	-	-	-	3
let	1	-	-	-	2
it	1	-	-	-	2

...słownika powiązać listę dokumentów; zbiór takich list to wystąpienia (*occurrences*)



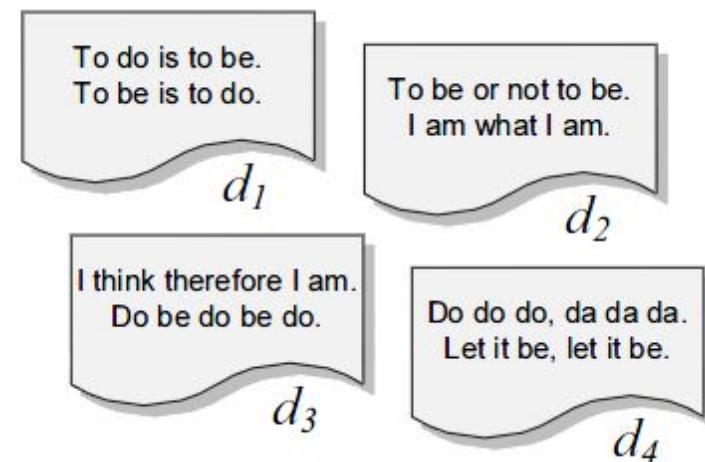
Eksploracja danych w Internecie

- Podstawowy indeks odwrotny – zbiór list zawierających identyfikatory dokumentów i ilości wystąpień :

Vocabulary	n_i
to	2
do	3
is	1
be	4
or	1
not	1
I	2
am	2
what	1
think	1
therefore	1
da	1
let	1
it	1

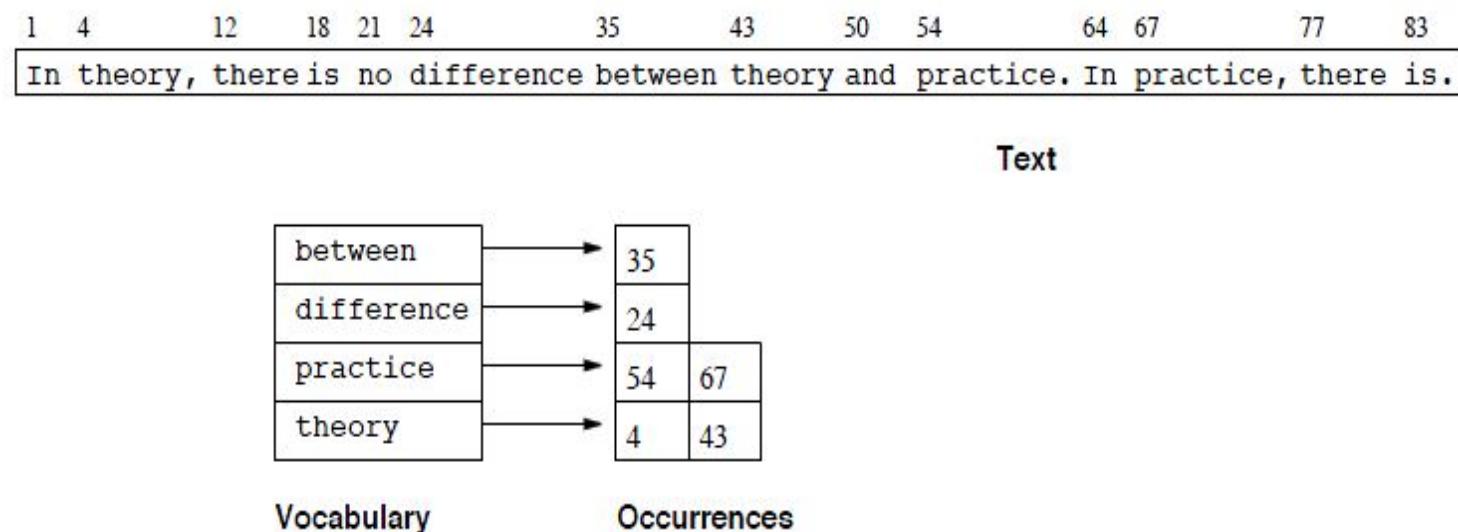
Occurrences as inverted lists

[1,4],[2,2]
[1,2],[3,3],[4,3]
[1,2]
[1,2],[2,2],[3,2],[4,2]
[2,1]
[2,1]
[2,2],[3,2]
[2,2],[3,1]
[2,1]
[3,1]
[3,1]
[4,3]
[4,2]
[4,2]



Indeksowanie i wyszukiwanie

- Podstawowy indeks odwrotny nie jest wystarczający do wyszukiwania odpowiedzi na pytania ze zwrotami (phrase) lub przybliżone (proximity queries)
- W tym celu należy go uzupełnić o pozycję każdego słowa w każdym dokumencie tworząc pełny indeks odwrotny



Indeksowanie i wyszukiwanie

- Pełny indeks odwrotny (adresujący słowa) – przykład:

Vocabulary	n_i
to	2
do	3
is	1
be	4
or	1
not	1
I	2
am	2
what	1
think	1
therefore	1
da	1
let	1
it	1

Occurrences as full inverted lists

[1,4,[1,4,6,9]], [2,2,[1,5]]
[1,2,[2,10]], [3,3,[6,8,10]], [4,3,[1,2,3]]
[1,2,[3,8]]
[1,2,[5,7]], [2,2,[2,6]], [3,2,[7,9]], [4,2,[9,12]]
[2,1,[3]]
[2,1,[4]]
[2,2,[7,10]], [3,2,[1,4]]
[2,2,[8,11]], [3,1,[5]]
[2,1,[9]]
[3,1,[2]]
[3,1,[3]]
[4,3,[4,5,6]]
[4,2,[7,10]]
[4,2,[8,11]]

To do is to be.
To be is to do.

d_1

To be or not to be.
I am what I am.

d_2

I think therefore I am.
Do be do be do.

d_3

Do do do, da da da.
Let it be, let it be.

d_4



Indeksowanie i wyszukiwanie

- Przestrzeń zajmowana przez słownik jest relatywnie mała w porównaniu z rozmiarem tekstu; rośnie jak $O(n^\beta)$, gdzie $\beta \in [0.4, 0.6]$, n – rozmiar zbioru.
- We wzorcowej kolekcji dokumentów TREC-3 1GB dokumentów posiada słownik rzędu zajmujący ok. 5MB
- Słownik można dodatkowo redukować poprzez różne techniki np. *stemming*.
- Indeksy adresujące tylko dokumenty typowo zajmują 20% do 40% rozmiaru tekstu, kiedy usuwa się tzw. wyrazy nieistotne (*stopwords*).

Indeksowanie i wyszukiwanie

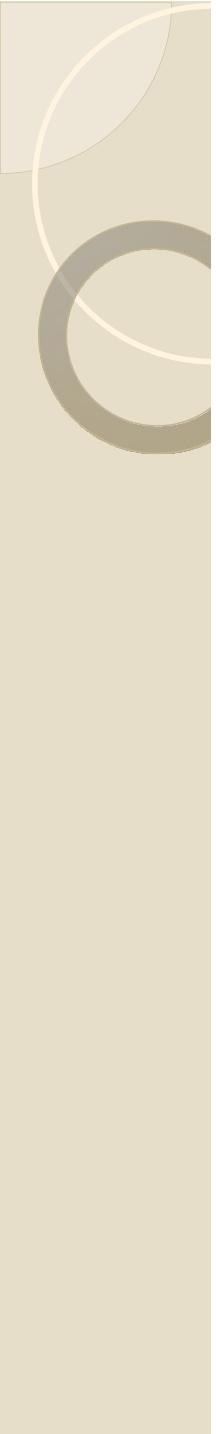
- Technika adresowania blokowego ograniczająca wymagania przestrzeni adresowej
- Dokument dzieli się na bloki i zapisuje tylko adresy bloków zawierające wyrazy, zamiast adresów poszczególnych wyrazów

Block 1	Block 2	Block 3	Block 4	
Text				Inverted Index
Vocabulary	Occurrences			
This is a text.	A text has many words. Words are made from letters.	letters	4...	
		made	4...	
		many	2...	
		text	1, 2...	
		words	3...	

Indeksowanie i wyszukiwanie

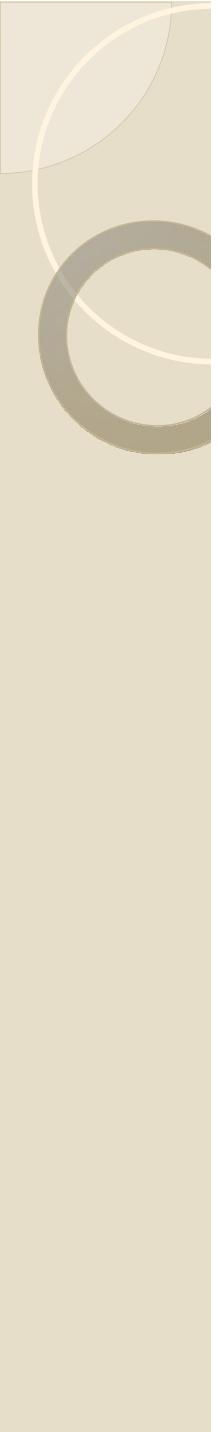
- Przykład: przestrzeń zajmowana przez indeksy odwrotne zależnie od rozmiaru dokumentów ; w każdej kolekcji lewa kolumna indeksuje tylko wyrazy istotne, prawa wszystkie wyrazy

Index granularity	Single document (1 MB)	Small collection (200 MB)	Medium collection (2 GB)		
Addressing words	45%	73%	36%	64%	35%
Addressing documents	19%	26%	18%	32%	26%
Addressing 64K blocks	27%	41%	18%	32%	5%
Addressing 256 blocks	18%	25%	1.7%	2.4%	0.5%
					0.7%



Indeksowanie i wyszukiwanie

- Podział tekstu na niewielkie bloki stałej wielkości zwiększa efektywność wyszukiwania – długie bloki są częściej wybierane i ich przeszukiwanie trwa dłużej
 - Powtarzające się referencje do tych samych słów, w tym samym kontekście sprowadza się do jednej referencji
-



Indeksowanie i wyszukiwanie

- Zapytania w formie pojedynczych słów (single word):
 - przeszukiwanie słownika może być prowadzone z użyciem haszowania, drzew trie lub B-drzew,
 - Pierwsze dwie metody dają koszt wyszukiwania $O(m)$, gdzie m – długość zapytania
 - Przyjmuje się, że słownik pozostaje w pamięci operacyjnej a lista wystąpień jest pobierana z dysku



Indeksowanie i wyszukiwanie

- Zapytania w formie wielu słów (multiple word):
- **Pytania koniunktywne (AND):** poszukuję wszystkich słów zapytania z listą indeksu odwrotnego dla każdego z nich. Poszukuje się iloczynu zbiorów list odwrotnych.
- **Pytania dysjunktywne (OR):** skleja się listy indeksów odwrotnych poszczególnych słów

- Najbardziej czasochłonna operacja na indeksach odwrotnych to łączenie list wystąpień
- m, n – rozmiary dwóch rozpatrywanych list w pamięci szeregowej
 - m << n to najlepiej m razy przeszukać listę n elementów aby dokonać wstawienia
 - m ≈ n można zastosować dwukrotnie algorytm przeszukiwania binarnego – wymaga średnio $m+n$ porównań

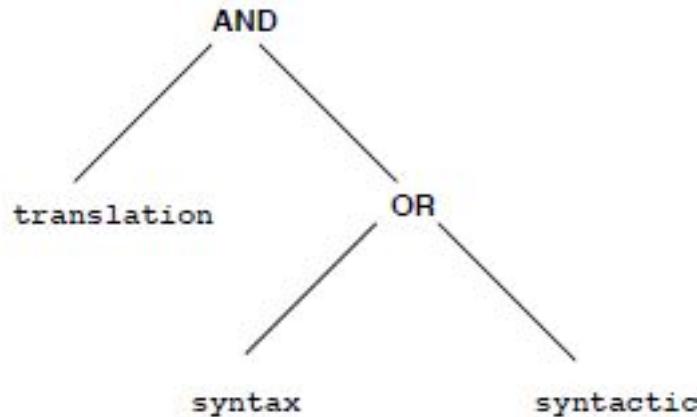


Indeksowanie i wyszukiwanie

- Pytania kontekstowe (frazy i zapytania przybliżone):
 - listy elementów są przeszukiwane aby znaleźć pozycje gdzie pojawia się ciągła sekwencja słów (frazy) lub sekwencja słów dostatecznie blisko siebie (pytania przybliżone)
 - Stosuje się algorytmy podobne jak przy dwóch łączeniu list
 - Dla wyszukania fraz można stosować także indeksowanie par słów przy pomocy zbliżonych algorytmów

Indeksowanie i wyszukiwanie

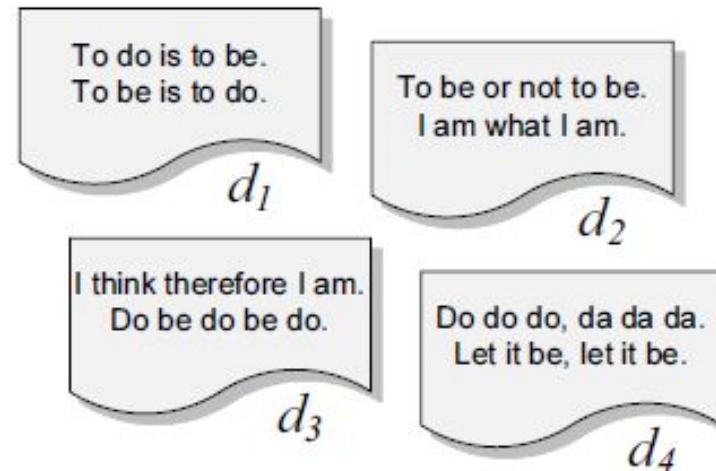
- Zapytania boolowskie: - składnia w formie drzewa

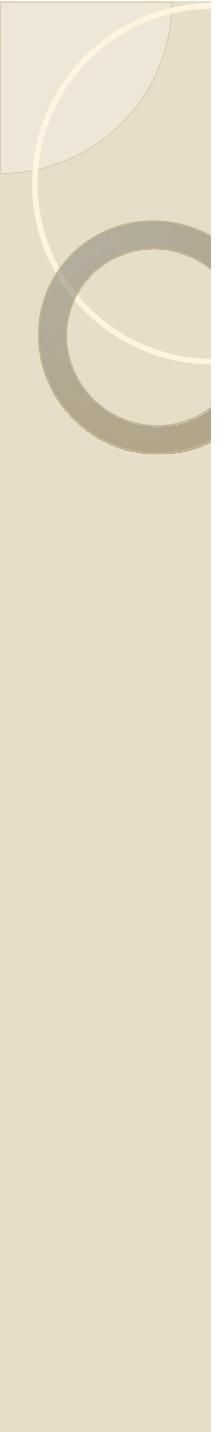


- w pierwszej fazie określa się które dokumenty są do porównania,
- w drugiej fazie ocenia się istotność dokumentów,
- w trzeciej fazie wyznacza się dokładne pozycje dopasowania

Indeksy odwrotne - ranking

- Jak znaleźć k pierwszych dokumentów mając listy odwrotne posortowane według wag?
- W przypadku pojedynczego słowa zapytania sortowanie już zostało wykonane
- Dla innych zapytań trzeba skleić odpowiednie listy
- Np. poszukujemy odpowiedzi na pytanie dysjunktywne „to do” w kolekcji dokumentów





Indeksy odwrotne - ranking

- W najprostszym i najkosztowniejszym podejściu wykonuje się ranking wszystkich dokumentów np. w modelu wektorowym aby wybrać te najbardziej istotne.
 - Przy wykorzystaniu indeksu odwrotnego można zmaksymalizować iloczyn TF-IDF używając następującego podejścia heurystycznego :
 - Przeszukujemy termy w porządku malejącym IDF i wybieramy najpierw „to” o większej wartości IDF,
 - przeszukujemy listę odwrotną termu „to” wg. malejących TF i wybieramy skończoną ilość najlepszych dokumentów np. dwa
 - przeszukujemy listę odwrotną termu „do” i zastępujemy dokumenty z listy „to” jeżeli wartości TF nowych dokumentów są lepsze od poprzednio wybranych
-



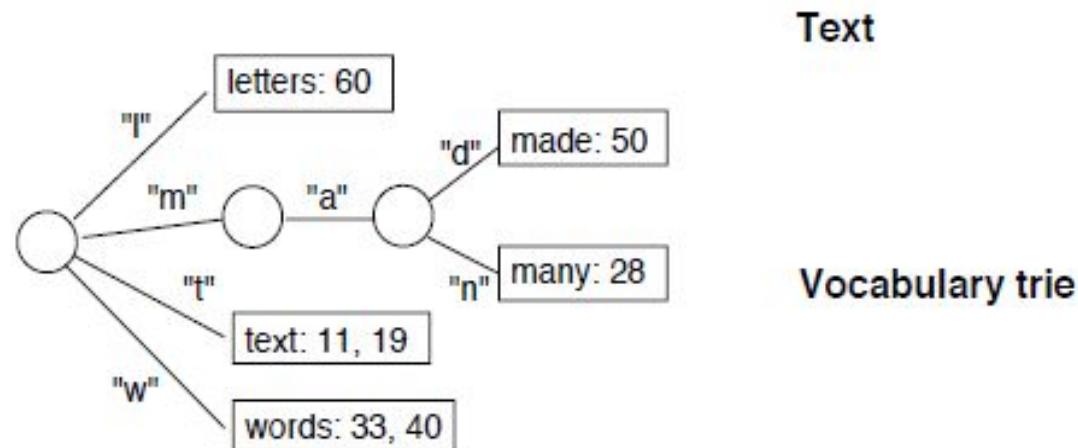
Indeksy odwrotne - ranking

- Budowa indeksu w pamięci RAM jest prosta i nie wymaga szczególnych nakładów obliczeniowych
- Tworzy się pustą strukturę danych do przechowywania słownika (B-drzewo, tablicę haszującą itp.)
- Podczas skanowania tekstu szuka się bieżącego słowa w słowniku
- Jeśli słowo jest nowe zostaje dodane do słownika przed dalszym przetwarzaniem
- Duża tablica indeksu jest alokowana tam gdzie zapisuje się identyfikatory kolejnych słów tekstu

Indeksy odwrotne - ranking

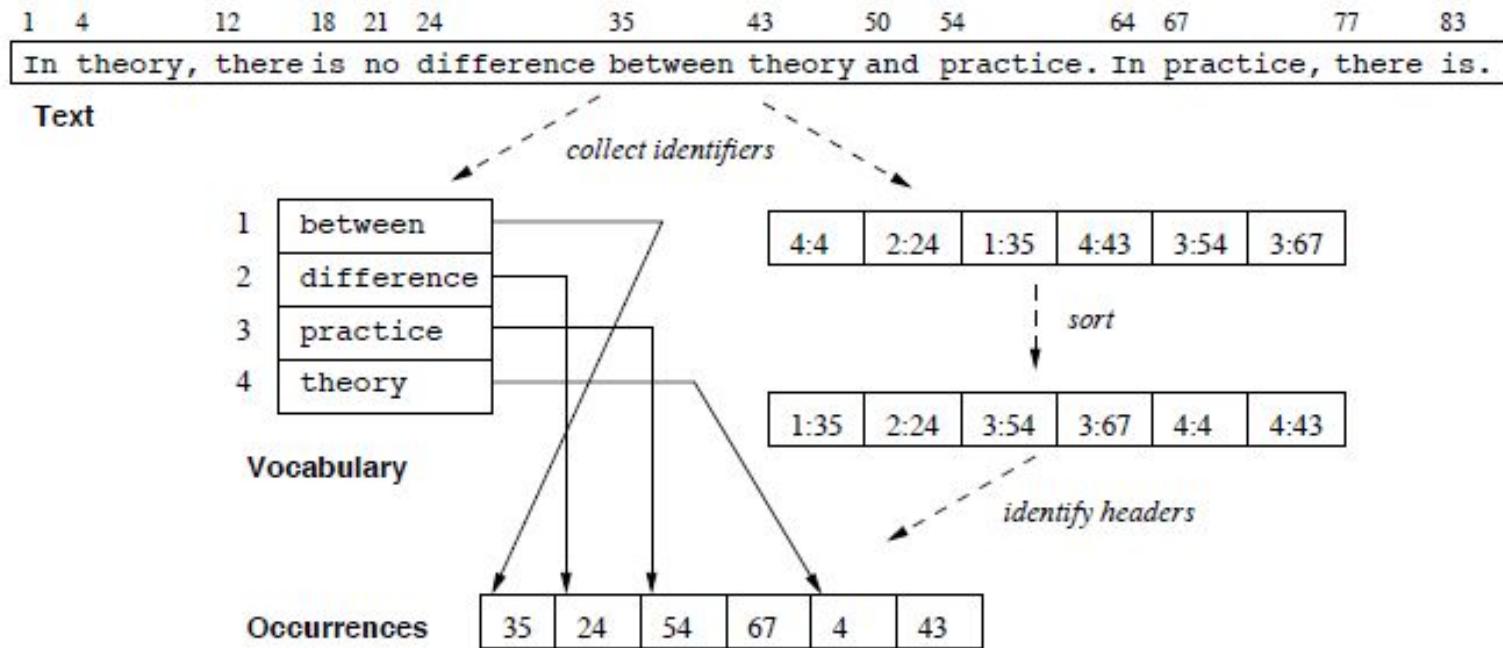
- Pełny indeks odwrotny dla przykładowego tekstu z algorytmem inkrementacyjnym:

1 6 9 11 17 19 24 28 33 40 46 50 55 60
This is a text. A text has many words. Words are made from letters.

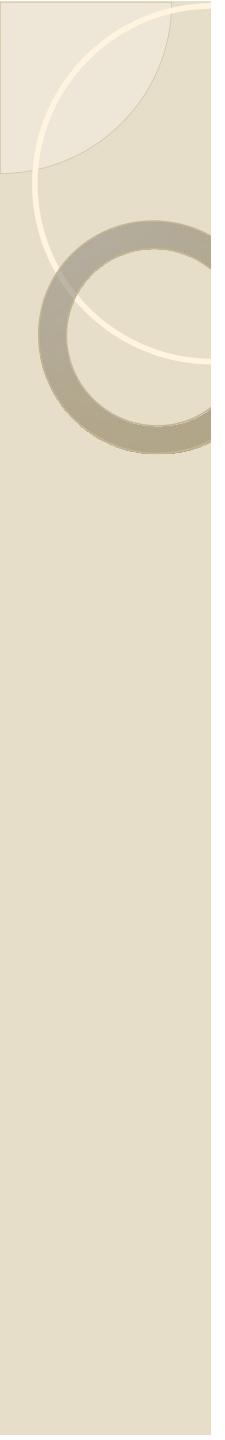


Indeksy odwrotne - ranking

- Pełny indeks odwrotny z algorytmem sortującym dla przykładowego tekstu :



- Aby uniknąć sortowania całości indeksu można tworzyć osobne listy dla każdego wyrazu w słowniku
- Z uwagi na ograniczenia pamięci preferowana jest lista bloków, każdy z wieloma elementami

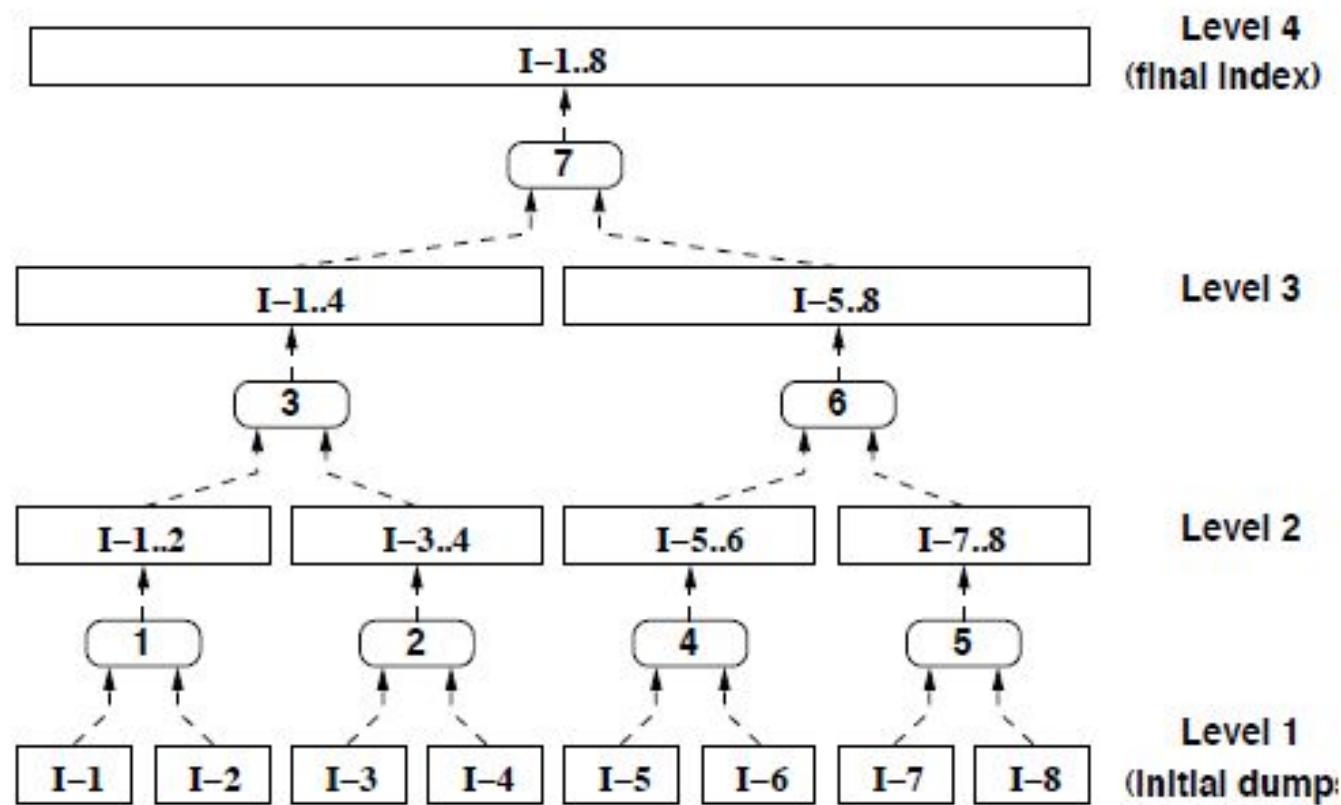


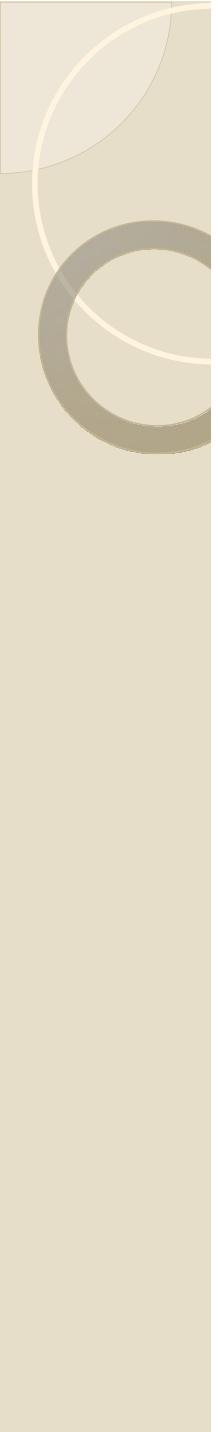
Indeksy odwrotne - ranking

- Kiedy zakończy się tworzenie listy wystąpień słownik i ta lista są zapisane w różnych plikach na dysku
- Słownik zawiera tylko wskaźniki do list odwrotnych dla każdego ze słów, co pozwala trzymać słownik w pamięci głównej (RAM)
- Po wyczerpaniu pamięci głównej uzyskany indeks częściowy jest przenoszony z pamięci głównej do dysku
- Takie indeksy częściowe są potem składane w sposób hierarchiczny

Indeksy odwrotne - ranking

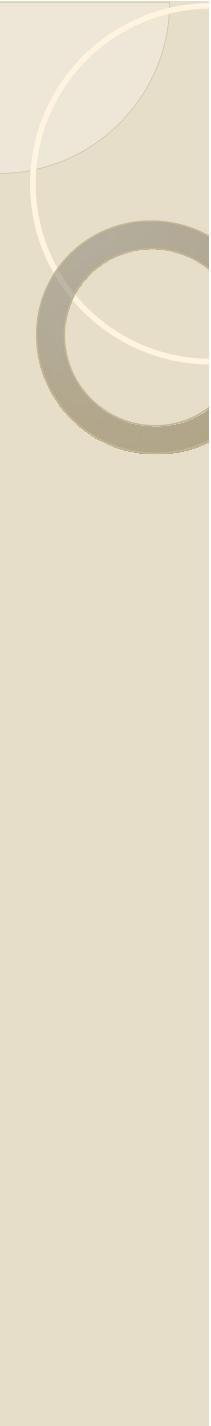
- Składanie binarne indeksów cząstkowych:





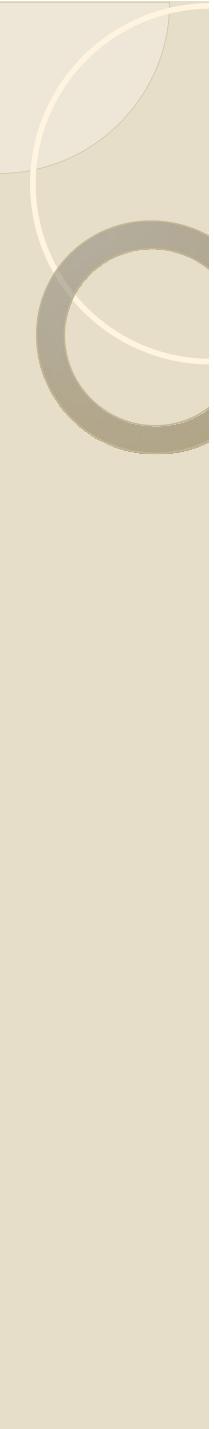
Indeksy odwrotne - ranking

- Indeks odwrócony można aktualizować na trzy sposoby:
 - przez przebudowywanie – gdy badany tekst nie jest zbyt długi,
 - przez przyrostowe aktualizacje – podczas przeszukiwania indeksu gdy wymagane są zmiany,
 - przez fragmentaryczne składanie – nowe dokumenty są oddzielnie indeksowane i ich indeks cząstkowy jest składany z indeksem głównym (najlepszy sposób w ogólnym przypadku)



Drzewa i tablice przyrostkowe

- Do implementacji systemów IR preferowane są indeksy odwrotne
- Sprawdzają się one gdy słowniki nie są zbyt duże, w przeciwnym razie efektywność ich użycia drastycznie maleje
- Indeksy odwrotne sprawdzają się w przypadku języków zachodnich, ale nie w językach niemieckim lub fińskim, które łączą krótkie sylaby w długie słowa; nie sprawdzają się także w językach dalekowschodnich
- Nie pyta się o słowa składane, ale o poszczególne składniki je tworzące
- W takich przypadkach używa się metody drzew i tablic przyrostkowych



Drzewa i tablice przyrostkowe

- Drzewa i tablice przyrostkowe umożliwiają przeszukiwanie indeksowe dowolnych podłańcuchów pasujących do łańcucha zapytania,
- Indeksy traktują tekst jako jeden długi łańcuch a każdy znak w tekście jest traktowany jako przyrostek tekstowy (**text suffix**)
- Przykład: suffiksy w tekście „missing missisipi”

```
missing mississippi
issing mississippi
sing mississippi
..
ppi
pi
i
```

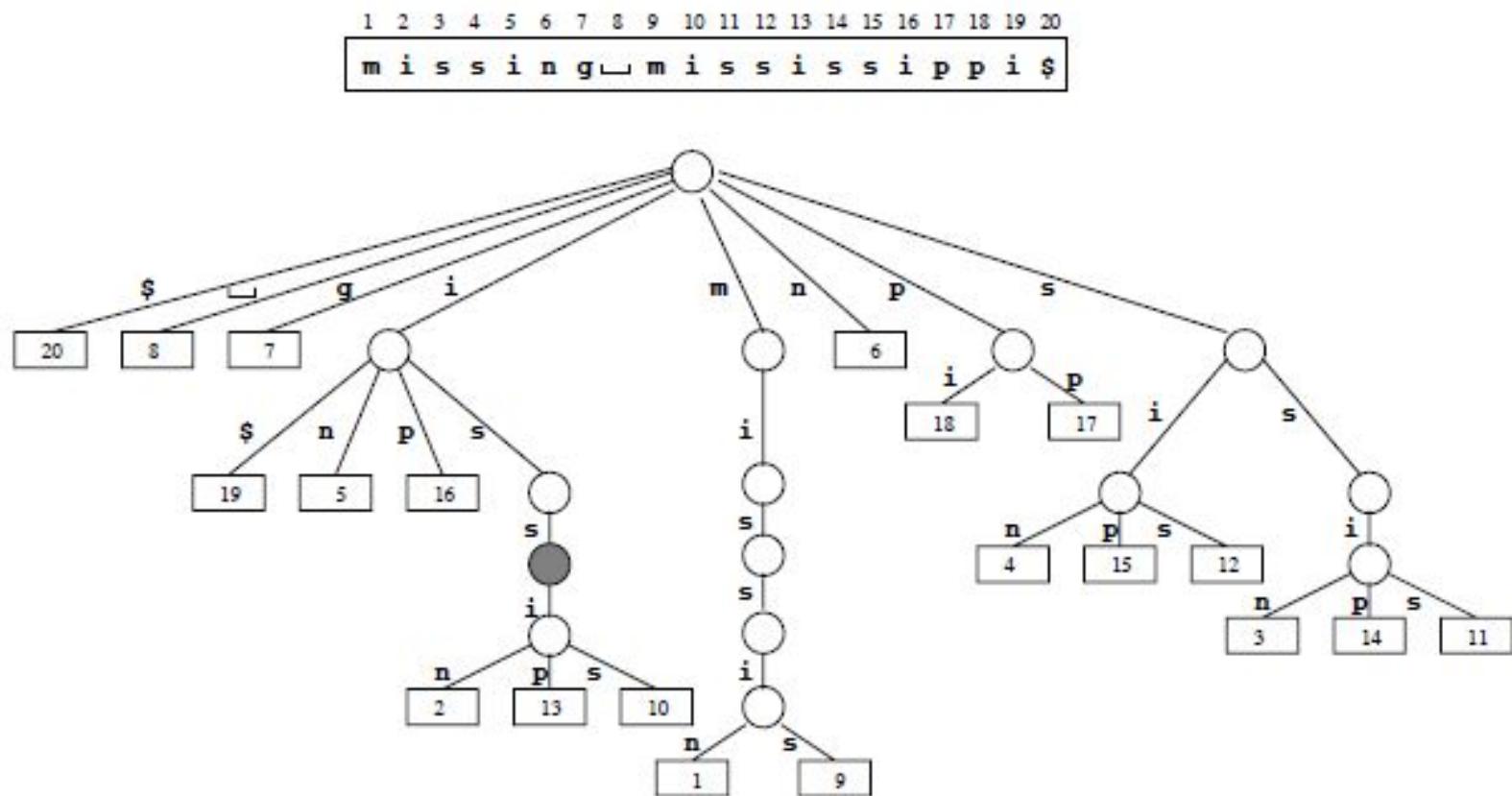


Drzewa i tablice przyrostkowe

- Strukturę przyrostków można opisać w oparciu o przyrostkowe drzewo wyszukiwania (*suffix trie - trie od retrieval*)
- Drzewa przyrostkowe to struktury danych, które zapisują zbioryłańcuchów tak, że można odtworzyć dowolnyłańcuch w czasie proporcjonalnym do jego długości, niezależnie od ilości zapisanychłańcuchów
- Zbiór przyrostków $P = \{P_1, \dots, P_r\}$ zapisanych jako drzewo wyszukiwania stanowi deterministyczny automat skończony (DFA) rozpoznający $P_1 | \dots | P_r$; poszukiwaniełańcucha przyrostkowego P jest równoznaczne z rozpoznaniem tegołańcucha przez DFA
- Drzewo przyrostkowe w tej wersji to struktura danychtrie zawierająca wszystkie przyrostki tekstu: $T = t_1 t_1 \dots t_n, \$$
- Liście drzewa trie zawierają wskaźniki do przyrostków $t_1 t_1 \dots t_n$

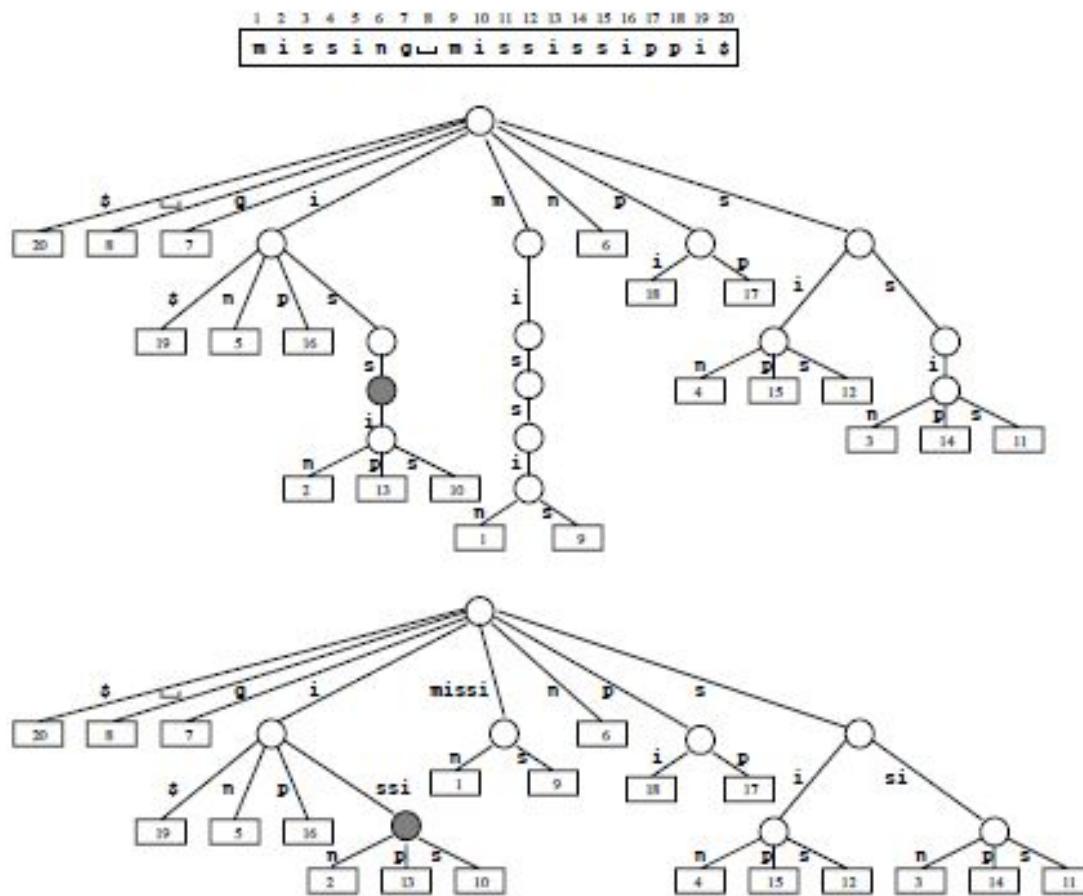
Drzewa i tablice przyrostkowe

- Drzewo wyszukiwania dla tekstu „missing missisipi”



Drzewa i tablice przyrostkowe

- Normalne drzewo przyrostkowe – krawędzie bez rozgałęzień są wyeliminowane



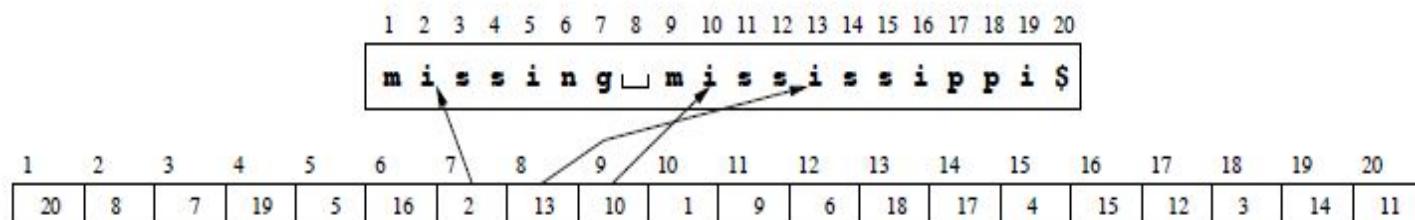


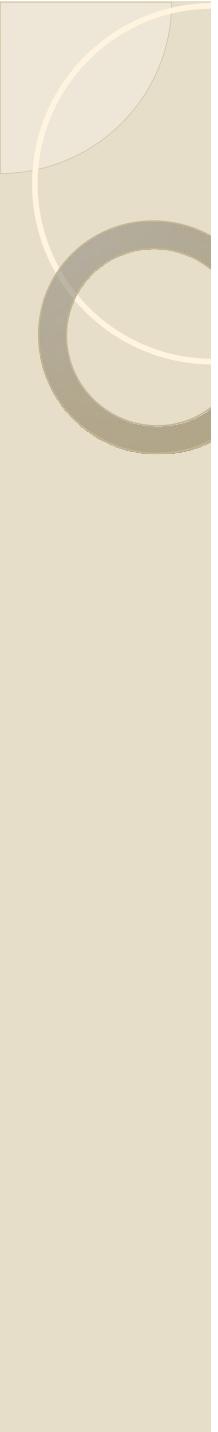
Drzewa i tablice przyrostkowe

- Drzewa przyrostkowe zajmują dużo miejsca w pamięci
 - Zależnie od implementacji zajmują 10 do 20 razy więcej miejsca niż pierwotny tekst
 - Ponadto indeksy jako drzewa przyrostkowe są efektywne przede wszystkim w pamięci RAM
-
- **Tablica przyrostkowa** dla tekstu T jest definiowana jako tablica wskaźników na wszystkie przyrostki T , które zostały uprzednio wysortowane leksykograficznie (jako liście drzева przyrostkowego od lewej do prawej)

Drzewa i tablice przyrostkowe

- Tablice przyrostkowe mają podobną funkcjonalność jak drzewa ale mniejsze wymagania pamięciowe – typowo są 4 razy większe od indeksowanego tekstu
- Tablice przyrostkowe nieco mają dłuższe czasy dostępu niż drzewa podczas wyszukiwania
- Tablica przyrostków dla tekstu „missing missisipi”

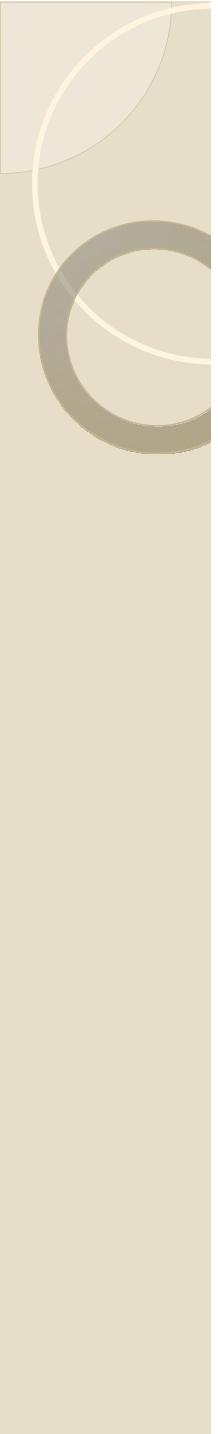




Drzewa i tablice przyrostkowe

■ Poszukiwanie łańcuchów:

- Dotyczy wszystkich łańcuchów (części przedrostkowych) pasujących do przyrostka $P = p_1p_1...p_m$
- Polega na przesuwaniu się w drzewie wyszukiwania po kolejnych znakach P .
- Możliwe są 3 sytuacje:
 - P nie pojawia się w tekście T – nie ma ścieżki odpowiadającej P w drzewie wyszukiwania
 - P zostaje znalezione przed osiągnięciem liścia w gałęzi drzewa – pojawi się na wszystkich pozycjach liści związanych z jego gałęzią
 - Liść drzewa pojawi się przed zakończeniem łańcucha P; należy kontynuować porównywanie tekstu aż do liścia aby zweryfikować obecność P



Drzewa i tablice przyrostkowe

- Jeżeli poszukiwanie dotyczy normalnego drzewa przyrostkowego krawędzie są skojarzone z całymi podłańcuchami
 - Wszystkie etykiety krawędzi odchodzące od danego węzła różnią się pierwszym znakiem
-

Drzewa i tablice przyrostkowe

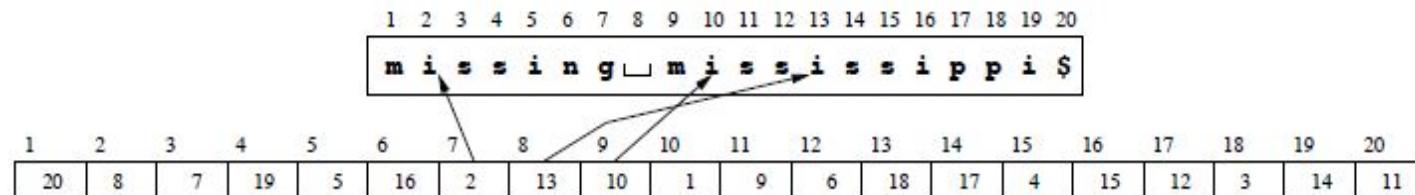
- Pseudokod przeszukiwania drzewa przyrostkowego:

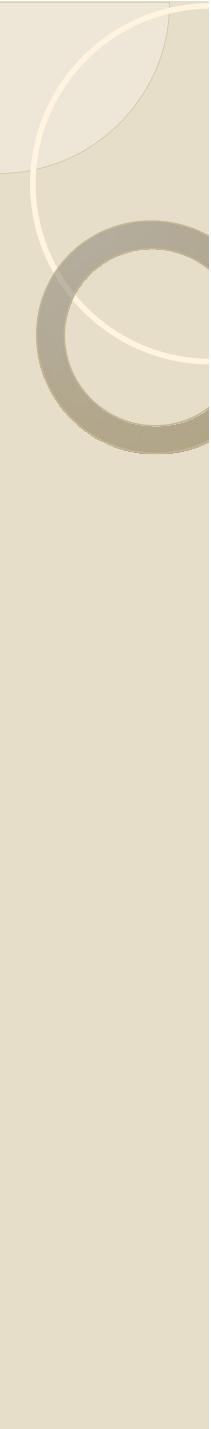
Suffix-Tree-Search (S , $P = p_1 p_2 \dots p_m$)

- (1) $i \leftarrow 1$
- (2) **while** $true$ **do**
- (3) **if** S is a leaf pointing to j **then**
- (4) **if** $p_i \dots p_m = t_{j+i-1} \dots t_{j+m-1}$
- (5) **then return** S
- (6) **else return** $null$
- (7) **if** there is an edge $S \xrightarrow{p'_1 \dots p'_s} S' \wedge p'_1 = p_i$ **then**
- (8) $j \leftarrow 0$
- (9) **while** $j < s \wedge i + j \leq m \wedge p'_{j+1} = p_{i+j}$ **do** $j \leftarrow j + 1$
- (10) $i \leftarrow i + j$
- (11) **if** $i > m$ **then return** S'
- (12) **if** $j < s$ **then return** $null$
- (13) $S \leftarrow S'$
- (14) **else return** $null$

Drzewa i tablice przyrostkowe

- Przeszukiwanie tablicy przyrostkowej jest typu binarnego z pośrednimi porównaniami
- Każdy krok binarnego wyszukiwania wymaga porównania P z przyrostkiem tekstu



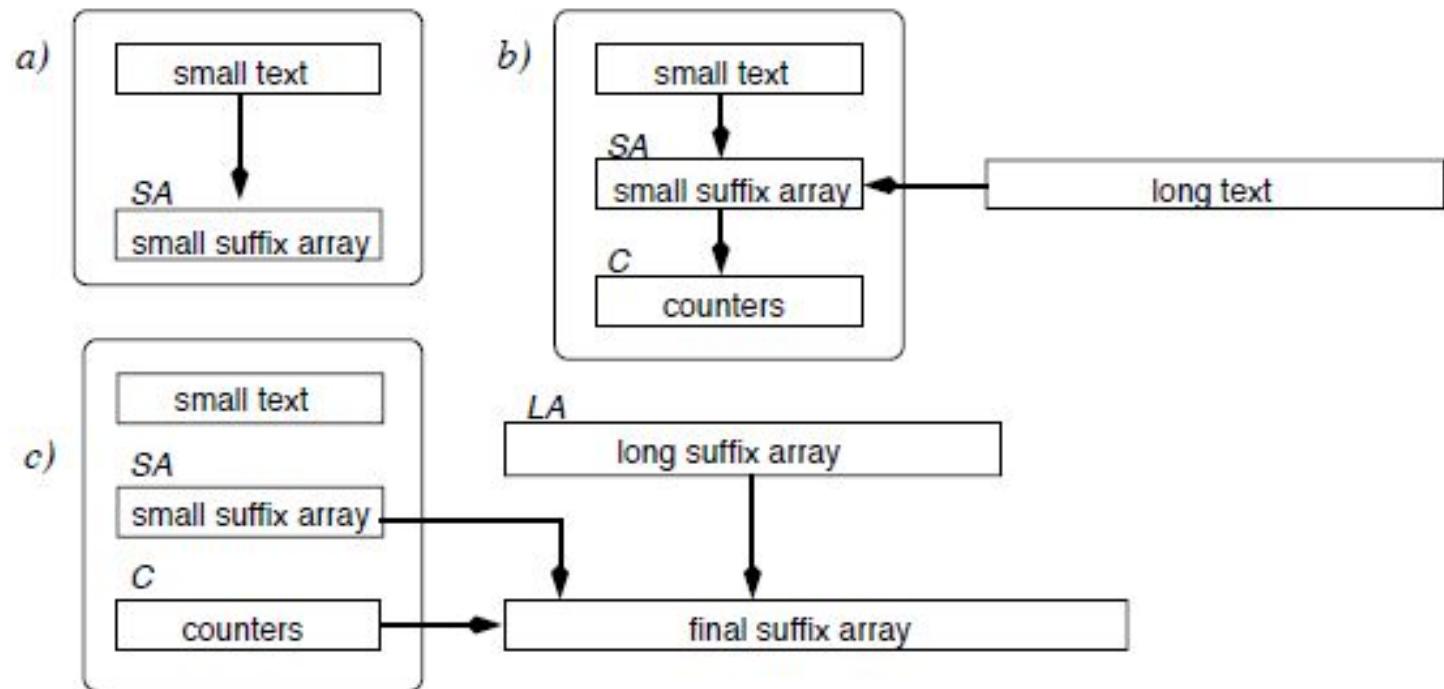


Drzewa i tablice przyrostkowe

- Drzewa i tablice przyrostkowe dla dużych tekstów
 - Kiedy dane tablic przyrostkowych lub tekstu nie mieszczą się w pamięci głównej specjalne algorytmy są wymagane do pracy z pamięcią zewnętrzną
 - Tekst dzieli się na bloki, które mogą być sortowane w pamięci głównej
 - Dla każdego bloku buduje się tablicę przyrostkową w pamięci głównej i składa się ją z resztą tej tablicy zbudowaną dla poprzednich bloków
-

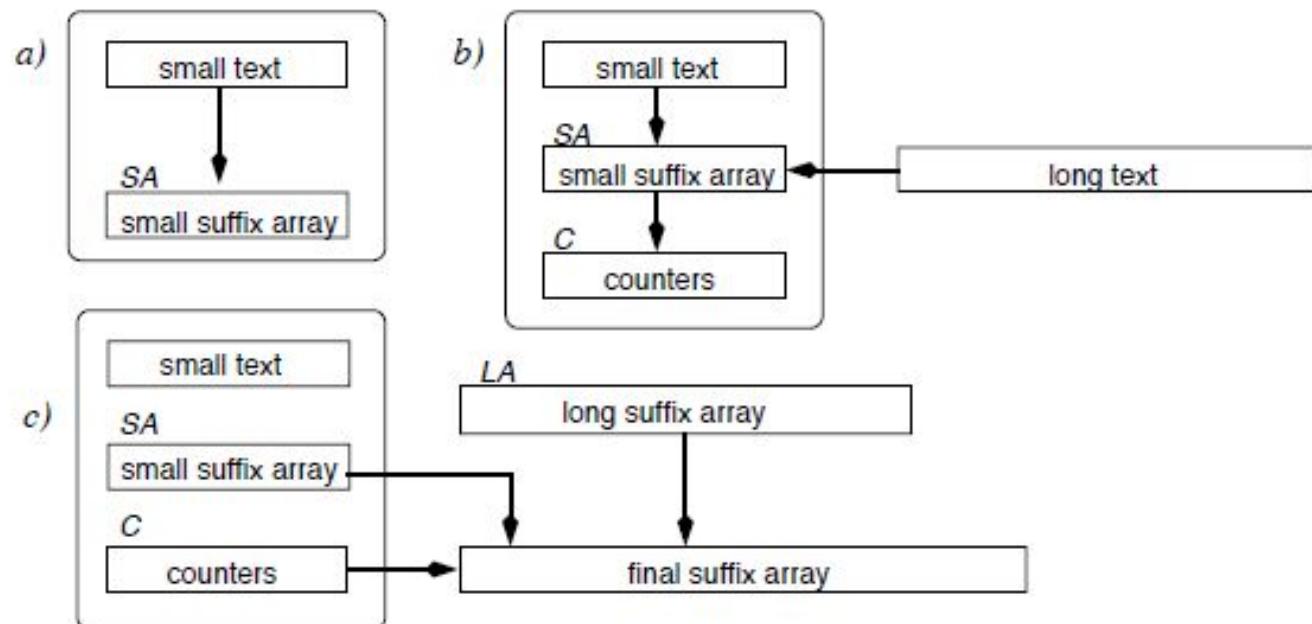
Drzewa i tablice przyrostkowe

- Krok budowy tablicy przyrostkowej dla dużych tekstów:
 - a) utworzenie małej tablicy przyrostków SA,
 - b) wypełnienie tablicy liczników C ,
 - c) sklejenie tablic LA dla bloków $1, 2, \dots, i-1$ oraz tablicy SA dla bloku i



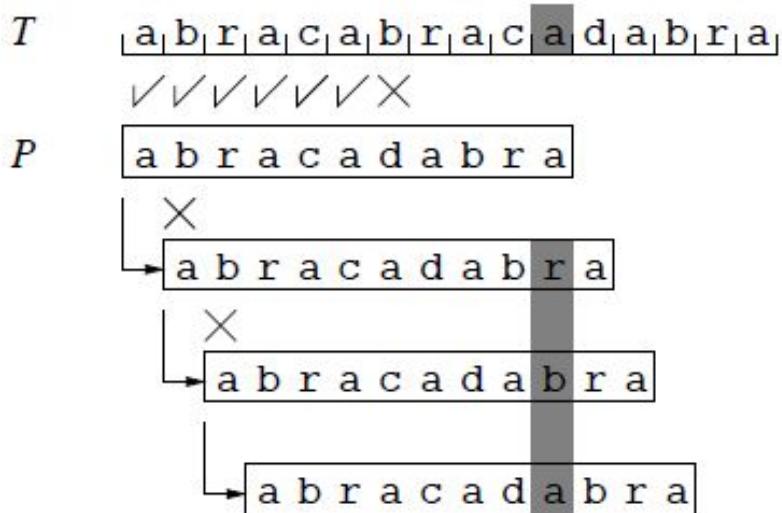
Drzewa i tablice przyrostkowe

- Tablica liczników C w każdej komórce $C[j]$ zapamiętuje ile przyrostków LA jest do wstawienia pomiędzy $SA[j]$ i $SA[j+1]$
- Obliczenie C nie wymaga dostępu do LA ; tekst odpowiadający LA jest sekwencyjnie wczytywany do pamięci głównej
- Każdy przyrostek tekstu jest poszukiwany w SA ; jeśli jest pomiędzy $SA[j]$ i $SA[j+1]$ to inkrementujemy $C[j]$



Wyszukiwanie sekencyjne

- Algorytm „brute force” polega na sprawdzeniu po kolejnych wszystkich możliwych pozycji wzorca w tekście
- Polega na przesuwaniu wzduż łańcucha tekstu okienka o długości m wzduż tekstu o długości m; $t_{i+2}...t_{i+m}$ dla $i \leq 0 \leq -m$.
- Wzorzec może wystąpić w każdym z okien, które muszą być po kolej sprawdzone
- *Przykład:*



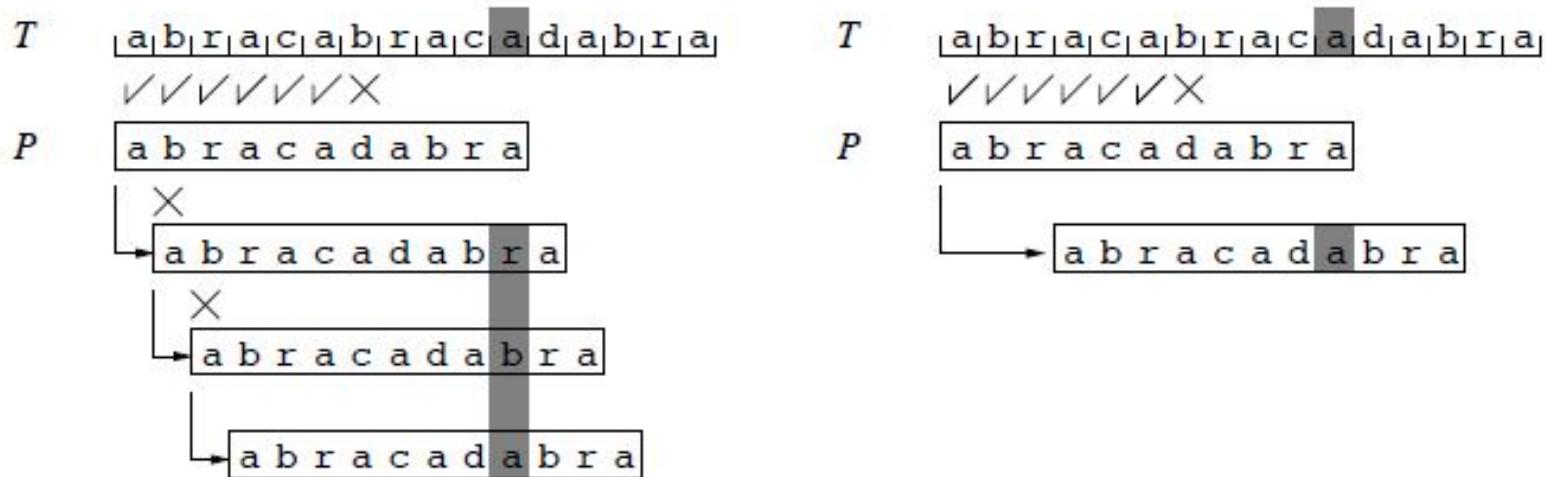


Wyszukiwanie sekencyjne

- Algorytm Horspool'a: najszybszy algorytm wyszukiwania łańcuchów dla języków naturalnych
- Wykorzystuje pomysł przesuwania okna ze wzorcem jak poprzednio
- Wstępnie przelicza tabelę d przesunięć okna indeksowaną znakami alfabetu; $d[c]$ zapisuje o ile pozycji trzeba przesunąć okno, jeżeli jego ostatnim znakiem jest c .
- Inaczej, $d[c]$ jest odległością od końca wzorca p_m , z wyłączeniem tego końca, do ostatniego wystąpienia c w P .

Wyszukiwanie sekencyjne

■ Algorytm Horspool'a – przykład



Wyszukiwanie sekencyjne

- Algorytm Horspool'a – pseudokod

Horspool ($T = t_1t_2 \dots t_n$, $P = p_1p_2 \dots p_m$)

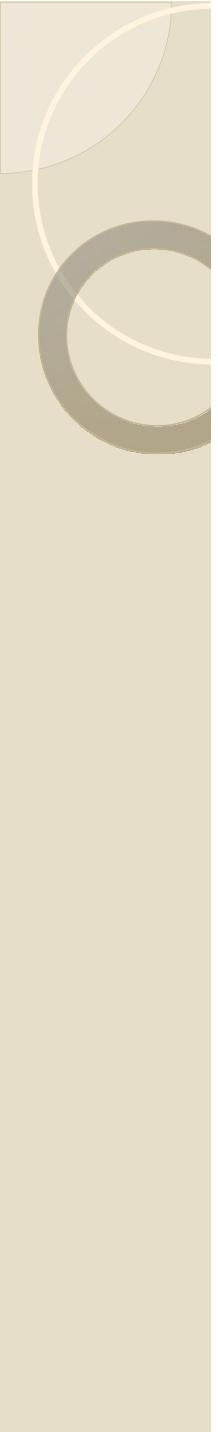
- (1) **for** $c \in \Sigma$ **do** $d[c] \leftarrow m$
- (2) **for** $j \leftarrow 1 \dots m - 1$ **do** $d[p_j] \leftarrow m - j$
- (3) $i \leftarrow 0$
- (4) **while** $i \leq n - m$ **do**
- (5) $j \leftarrow 1$
- (6) **while** $j \leq m \wedge t_{i+j} = p_j$ **do** $j \leftarrow j + 1$
- (7) **if** $j > m$ **then** report an occurrence at text position $i + 1$
- (8) $i \leftarrow i + d[t_{i+m}]$

- Przy długich wzorcach i krótkim alfabetie algorytm Horspool'a nie jest efektywny



Wyszukiwanie sekencyjne

- W przypadku ogólnym można przesuwać okno używając q końcowych znaków, a nie tylko ostatniego; jaka byłaby najlepsza wartość q ?
 - Jeżeli σ oznacza rozmiar alfabetu to wartość przesunięcia $\pi\sigma^q < m$, stąd $q = \log_\sigma(m)$ i średni czas szukania wynosi $O(n\log_\sigma(m)/m)$
-



Wyszukiwanie sekencyjne

- Programowanie dynamiczne:

- Klasyczna metoda przybliżonego dopasowaniałańcuchów wskazująca pozycję, gdzie wzorzec P pojawia się z co najwyżej k błędami.
- Stosuje się różne definicje błędów dopasowania:
 - odległość Hamminga – ilość niezbędnych podstawień znaków w tekście niezbędna dla dopasowania fragmentu tekstu do wzorca
 - odległość edycyjna (Levensteina) – ilość usunięć, wstawień i zamiany znaków w tekście niezbędna dla dopasowania fragmentu tekstu do wzorca



Wyszukiwanie sekencyjne

- Wylicza się tablicę $C[0 \dots m, 0 \dots n]$ elementów $C[i, j]$ reprezentujących minimalne ilości k błędów przy dopasowaniu wzorca $p_1 p_2 \dots p_i$ do tekstu $t_1 t_2 \dots t_j$.
- Tablicę wyznacza się następująco:

$$C[0, j] = 0,$$

$$C[i, 0] = i,$$

$$\begin{aligned} C[i, j] = & \text{ if } (p_i = t_j) \text{ then } C[i - 1, j - 1] \\ & \text{else } 1 + \min(C[i - 1, j], C[i, j - 1], C[i - 1, j - 1]), \end{aligned}$$

- Dopasowanie pojawia się na pozycjach j , takich że $C[m, j] \leq k$.
-

Wyszukiwanie sekencyjne

Przykład – programowanie dynamiczne poszukujące słowa „colour” w tekście „kolorama” z $k=2$ błędami; * opisuje wybraną pozycję wzorca

		k	o	l	o	r	a	m	a
	0	0	0	0	0	0	0	0	0
c	1	1	1	1	1	1	1	1	1
o	2	2	1	2	1	2	2	2	2
l	3	3	2	1	2	2	3	3	3
o	4	4	3	2	1	2	3	4	4
u	5	5	4	3	2	2	3	4	5
r	6	6	5	4	3	2*	3	4	5

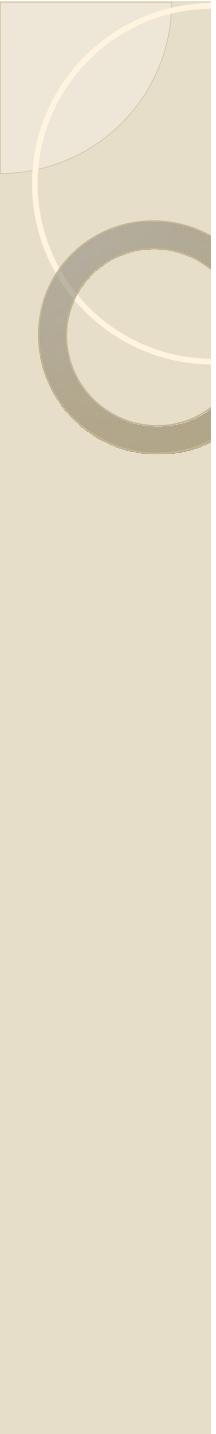
Odległość 2^* znaków jest spełniona w sensie Hamminga i Levensteina



Drzewa i tablice przyrostkowe

Programowanie dynamiczne w przypadku ogólnym wymaga $O(mn)$ czasu.

W przedstawionym wariantie korzystającym tylko z poprzedzającej kolumny $C[\cdot, j-1]$ do wyliczenia kolumny bieżącej $C[\cdot, j]$ algorytm może być implementowany w przestrzeni $O(m)$.



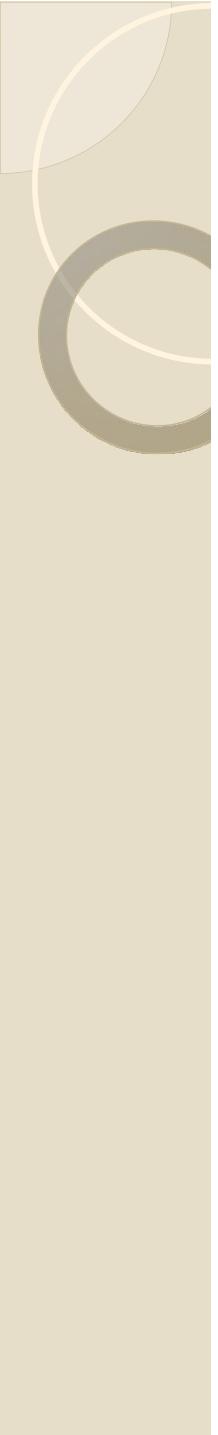
Drzewa i tablice przyrostkowe



Drzewa i tablice przyrostkowe



Drzewa i tablice przyrostkowe



Wyszukiwanie multimedów

Multimedia to dowolne dane cyfrowe video, dźwięk, zwykłe teksty, zazwyczaj bez struktury, używane do komunikacji i przechowywania informacji

Wyszukiwanie multimedów wymaga także ich rankingu według stopnia podobieństwa do zapytania

Dla celów wyszukiwania użytkownik może opisać scenę video np.: „Keanu Reeves unikający pocisków podczas zderzenia helikopterów w filmie Matrix”

Wyszukiwanie informacji multimedialnych obejmuje następujące tematy:

reprezentacja treści i obiektów multimedialnych,

wydobywanie cech,

formułowanie pytań odwzorowujących semantykę wysokopoziomową na cechy niskiego poziomu



Wyszukiwanie multimedów

zapytania przez przykłady
sprzężenie informacyjne, pytania interaktywne,
indeksowanie i katalogowanie cech,
zintegrowane wyszukiwanie typów searching i browsing,
przeszukiwanie multimedów w oparciu o ich treści.

W tekście słowa, znaki przestankowe i paragrafy tworzą jego strukturę

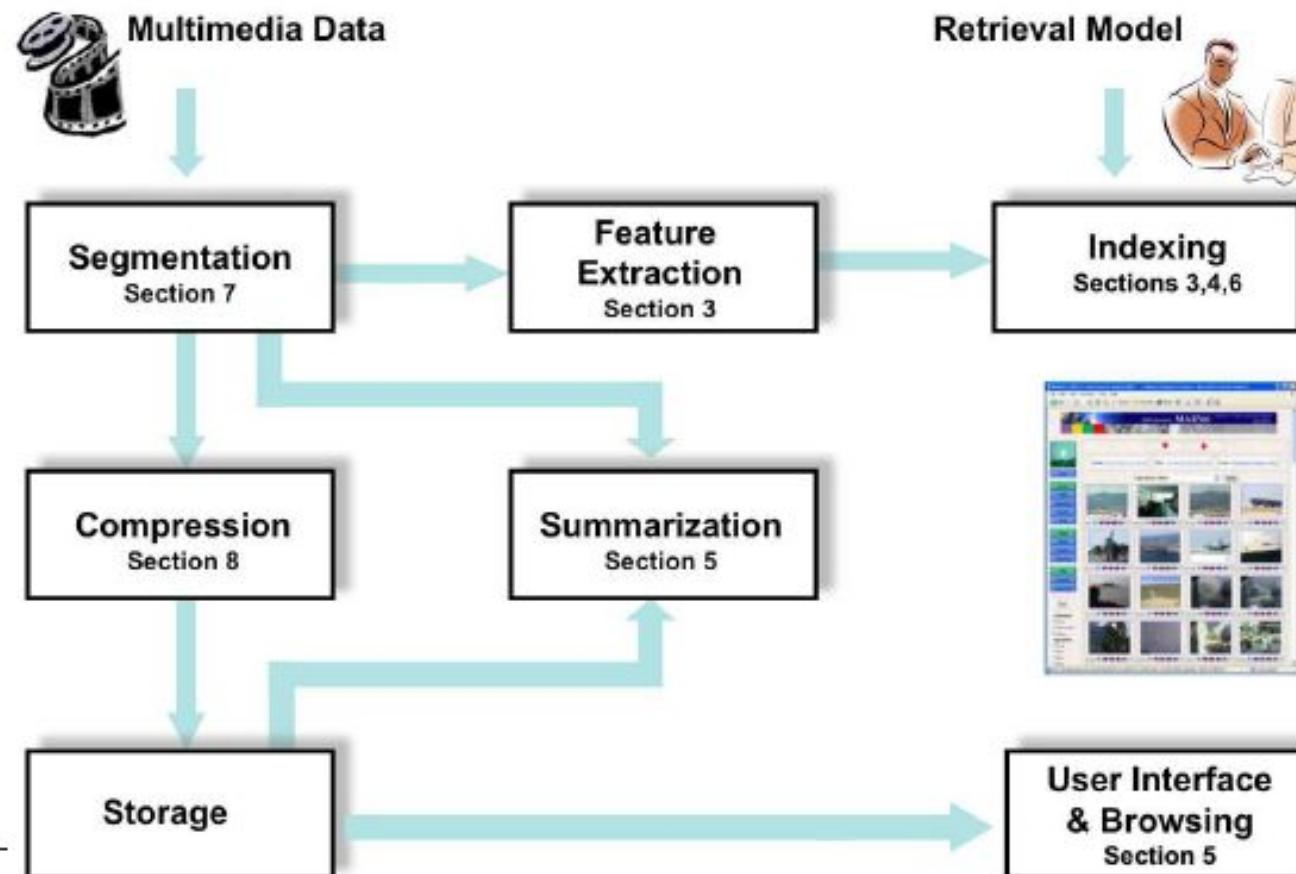
W multimediacach dane są zazwyczaj jednym, nieprzerwanym strumieniem; pożąданie jest zdefiniowanie jednostek semantycznych



Wyszukiwanie multimedów

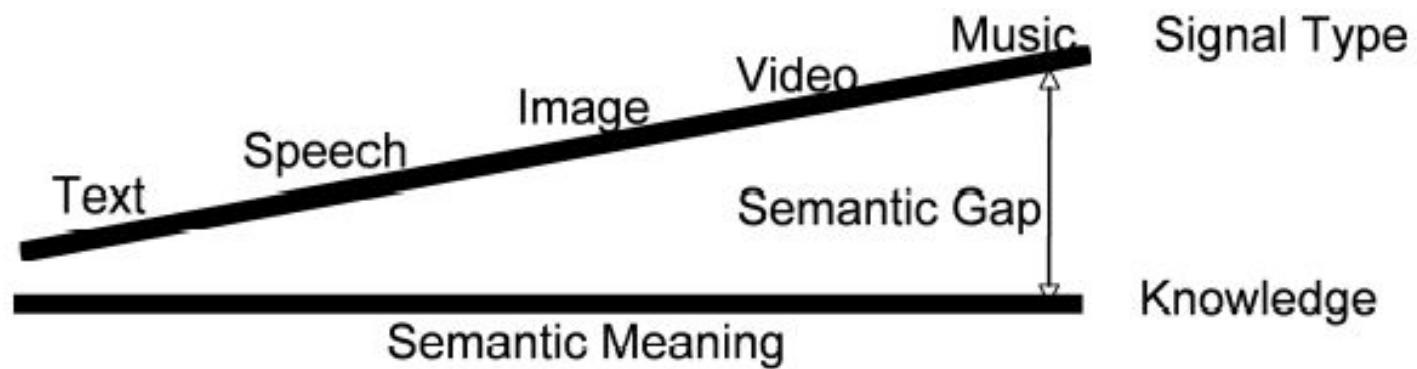
Przepływ informacji przy wyszukiwaniu multimedów:

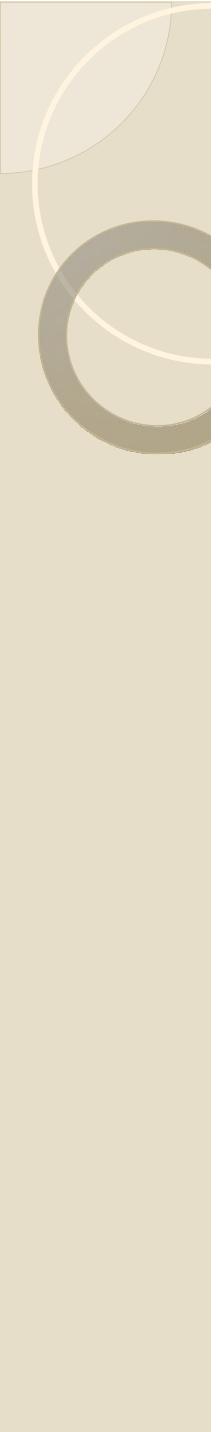
High-Level Multimedia IR Software Architecture



Wyszukiwanie multimedów

Między treścią sygnału multimedialnego i jego znaczeniem występuje duża luka semantyczna.





Wyszukiwanie multimedów

Identyfikacja obiektów: istotny problem w przetwarzaniu obrazu i dźwięku

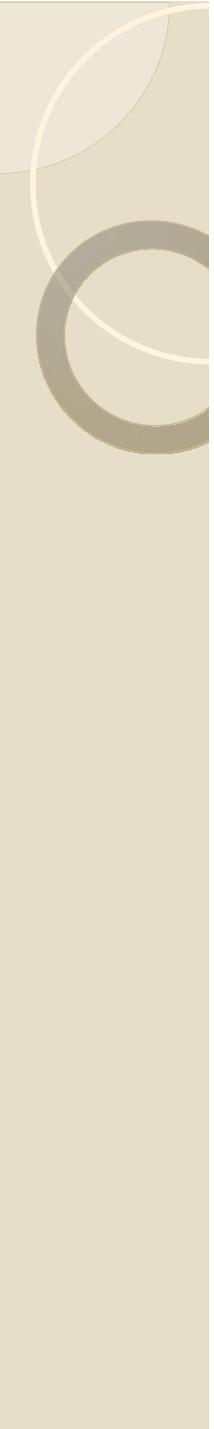
ludzie patrząc na obraz rozpoznają twarze i inne obiekty
automatyczne etykietowanie obiektów w obrazach lub
rozpoznawanie dźwięków w sygnałach audio są
problemami nadal nie rozwiązany globalnie

Dlatego systemy wyszukiwania multimedów często wykorzystują opisy tekstowe wytworzone przez człowieka

Systemy opisu multimedów często wykorzystują interpretacje emocjonalne

W przypadku mowy informacja niesemantyczna jest przekazywana przez tzw. prozodia sygnału – akcent, intonacja i iloczas.

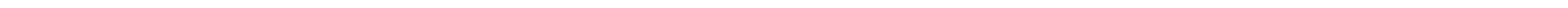
Np. prozodia pozwalają rozróżnić pomiędzy zdaniami „Nie zatrzymuj się” i „Nie zatrzymuj się!”

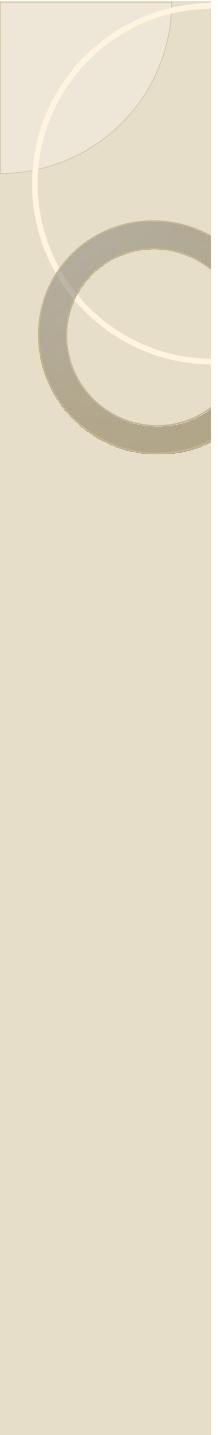


Wyszukiwanie multimedów

Apertura:

Ze względu na duże ilości danych ruch w sekwencji obrazów testuje się na małej porcji każdego z obrazów określonej przez aperturę; utrudnia to interpretację ruchu na obrazach video





Wyszukiwanie obrazów z treści

Polega na identyfikowaniu i wydobywaniu najważniejszych cech opisujących treść obrazu

Zapytanie przez przykład (*Query By Example - QBE*):

użytkownik posługuje się przykładowym obrazem jako zapytaniem o podobne do niego i ignoruje informację sementyczną związaną z obrazem

Najlepsze rankingi bazują na cechach obrazu takich jak: poza, ogniskowa kamery, oświetlenie, pozycja kamery, charakter ruchu.

Typowe cechy obrazów wyszukiwane przy zapytaniach QBE

średni kolor i jego rozkład na powierzchni obrazu,
 histogramy składowych koloru dla zadanej ilości słupków

Są to cechy niezależne od rozdzielczości i kąta pochylenia obrazu

Wyszukiwanie obrazów z treści

- Dla takich cech nie ma konieczności segmentacji obiektów i tła obrazu
- **Histogram koloru** c_i w obrazie I :

$$h_I(c_i) = P(\text{color}(p) = c_i | p \in I)$$

gdzie $P(\text{color}(p) = c_i | p \in I)$ – prawdopodobieństwo, że piksel p losowo wybrany z obrazu I posiada kolor c_i .

- **Autokorelogram** uzupełnienia histogram informacją o rozłożeniu koloru w polu obrazu

$$h_I(c_i, c_j, r) = P(\text{color}(p_1) = c_i \wedge \text{color}(p_2) = c_j | r = d(p_1 - p_2))$$

- gdzie p_1, p_2 – piksele losowo wybrane z obrazu I ,
- $d(p_1 - p_2)$ – odległość pikseli p_1, p_2 na obrazie.



Miary teksturalne

Problem:

człowiek rozpoznaje kolory obiektów niezależnie od padającego na nie światła – np. jabłko rozpoznaje się jako czerwone w świetle dziennym i przy oświetleniu sztucznym; histogramy kolorów nie spełniają tego warunku

Tekstura – obszar o powtarzającym się wzorze kolorów łatwo rozpoznawalny przez człowieka

Miary tekstury nie powinny zależeć od jasności ani orientacji obrazu

Macierz korelacji poziomów szarości (GLCM) – rejestruje zmiany jasności pomiędzy parami pikseli w obrazie; zapisuje w ten sposób informację o teksturowe jasności obrazu



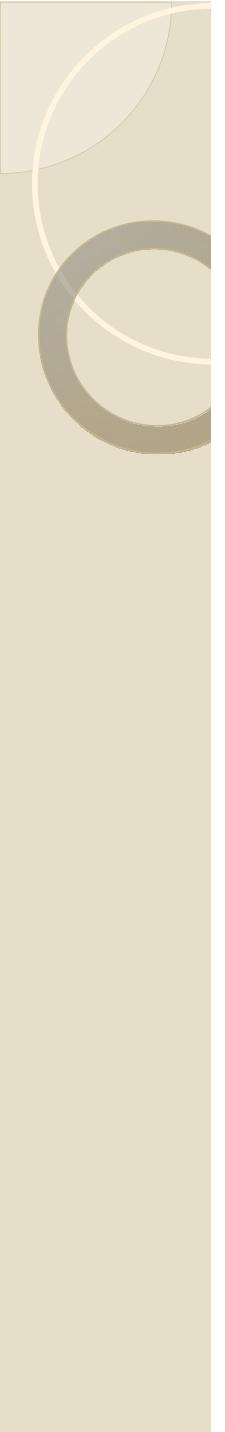
Miary teksturalne

- Przy obliczaniu macierzy GLCM rozważa się wszystkie v-związane pary pikseli p_1, p_2 odległe o wektor $\vec{v} = \vec{p}_2 - \vec{p}_1$
- Wyznacza się:

$$P_I(c_i, c_j, \vec{v}) = P(\text{color}(p_1) = c_i, \text{color}(p_2) = c_j | \vec{p}_2 - \vec{p}_1 = \vec{v})$$

gdzie $P_I(c_i, c_j, \vec{v})$ – prawdopodobieństwo znalezienia v-związkanych par pikseli związanych z kolorami c_i, c_j w obrazie I .

- Globalne wskaźniki tekstury obliczane na podstawie GLCM:
energia, entropia, kontrast, jednorodność.



Miary teksturalne

- **Energia** – miara jasności v-związanych pikseli

$$\mathcal{E}_I(c_i, c_j, \vec{v}) = \sum_i \sum_j P_I(c_i, c_j, \vec{v})^2$$

- **Entropia** – miara niejednorodności v-związanych pikseli

$$\Psi_I(c_i, c_j, \vec{v}) = \sum_i \sum_j P_I(c_i, c_j, \vec{v}) \log P_I(c_i, c_j, \vec{v})$$

- **Kontrast** – miara różnic jasności ϕ_i dla v-związanych par pikseli

$$\mathcal{C}_I(c_i, c_j, \vec{v}) = \sum_i \sum_j (\phi_i - \phi_j)^2 P_I(c_i, c_j, \vec{v})$$



Miary teksturalne

Jednorodność – miara podobieństwa v-związkanych pikseli

$$\mathcal{H}_I(c_i, c_j, \vec{v}) = \sum_i \sum_j \frac{P_I(c_i, c_j, \vec{v})}{1 + |\phi_i - \phi_j|}$$

Inteligentniejsze metody oparte są o modele cech obrazu – np. kombinacje koloru i częstotliwości przestrzennych wybranych regionów

Metoda punktów (regionów) odniesienia (*sailent points*): zbieranie cech obrazu niezależnych od skali, oświetlenia, pozycji kamery, rotacji obiektów; wykorzystuje się punkty odniesienia, unifikację orientacji i lokalną geometrię dla tekstury

Wyszukiwanie obrazów z treści

Punkty odniesienia są związane z rogami obrazu lub jego szczególnymi obszarami





Wyszukiwanie obrazów z treści

Podobieństwo obrazów oblicza się za pomocą sumarycznych statystyk punktów (obszarów) odniesienia

W regionach odniesienia obrazy są specjalnie filtrowane

Wartości charakterystyk teksturalnych są klasterowane metodą k-średnich aby określić słowa języka

Do dopasowania tak przetworzonych obrazów używa się algorytmu pLSA (*probabilistic Latent semantic analysis*)