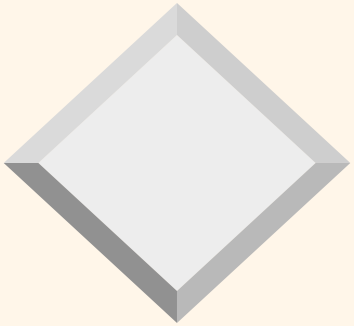


Dotazování nad databázemi textů

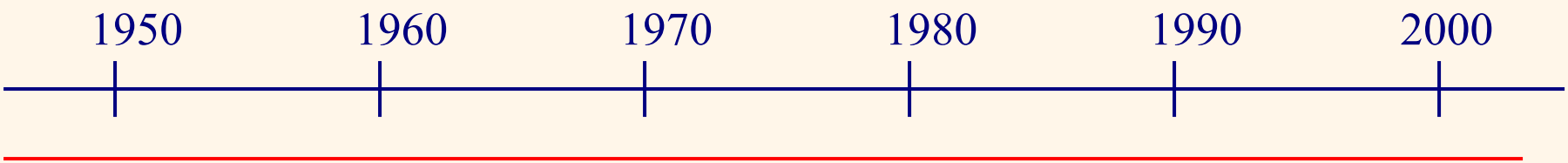
slajdy k přednášce NDBI006

Jaroslav Pokorný

MFF UK, Praha

pokorny@ksi.mff.cuni.cz

Vývoj DIS



systémy zpracování sekundárních informací

systémy zpracování úplných textů

digitální knihovny

Zdroje:

- vznik textů přímo v počítači
  - potřeba - vyhledávat, nejen listovat
  - ne vždy možné indexovat
- rozvoj velkých pamětí (CD ROM, WORM, hard disky*)
- rozvoj komunikací (Internet)

* dnes více než 10 TB

Obsah

1. Úvod
2. Měření relevance
3. Boolský model
4. Vektorový model
5. Zpětná vazba
6. Tezaurus
7. SQL/MM – Full-text
8. Závěr

Vyhledávání v textech

dotaz - požadavek formulovaný v nějakém jazyku bývá zadán vzorkem textu (slovo, výraz, část slova, nebo i celý text) nebo několika vzorky (*konjunktivní dotaz*)

Obecněji: Boolský výraz

*odpověď (*množina *hitů)* - texty vyhovující dotazu

relevance hitu - míra rozsahu, kterou se hit shoduje s požadavkem uživatele

omezení odpovědi - maximálně M

- maximálně M nejrelevantnějších
- zadání prahové hodnoty Θ

Vyhledávání v textech

Obor: **Information Retrieval**

(Vyhledávání informací)

IR je vše o vyhledávání toho, co chcete, když to, co chcete, je skryto v mase toho, co nechcete.

Přesněji: nalézt k dotazu relevantní dokumenty

Obor: **Information Filtering**

(Filtrování informací)

Přiřadit k dokumentu D profily tak, že D je pro ně relevantní.

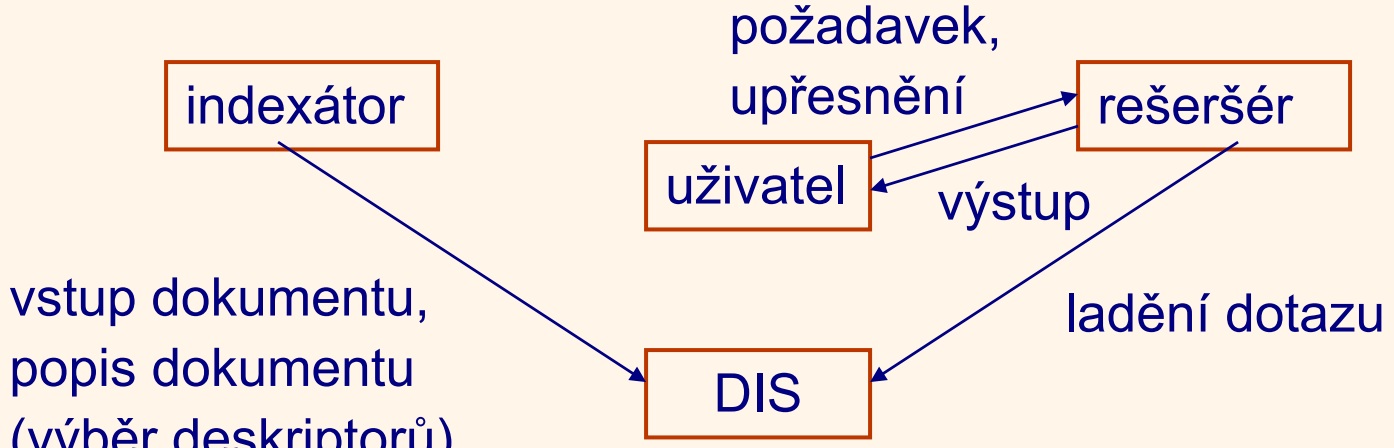
DIS - základní architektura

Subsystémy: zpřístupnění textu (1)

dodání textu (2)

(1) viz informační služby

sekundární informace vs. úplné texty



historický model

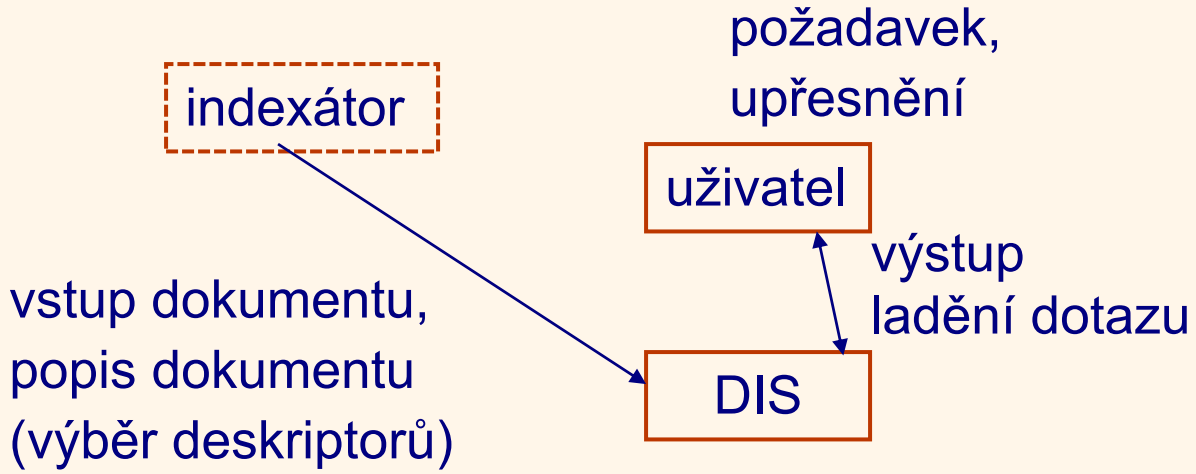
DIS - základní architektura

Subsystémy: zpřístupnění textu (1)

dodání textu (2)

(1) viz informační služby

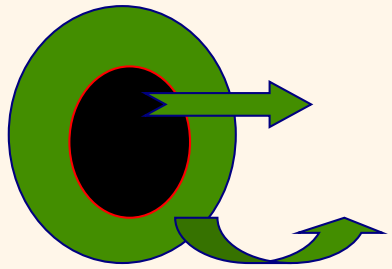
sekundární informace vs. úplné texty



současný model

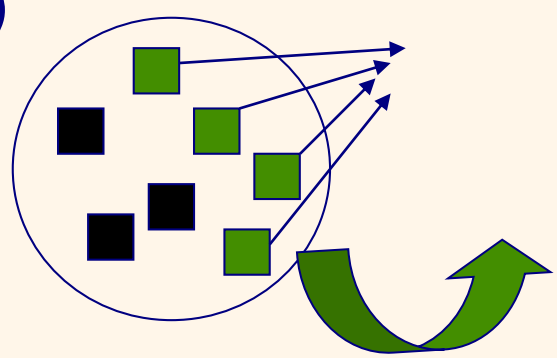
Měření relevance

koeficient úplnosti R (z angl. recall)



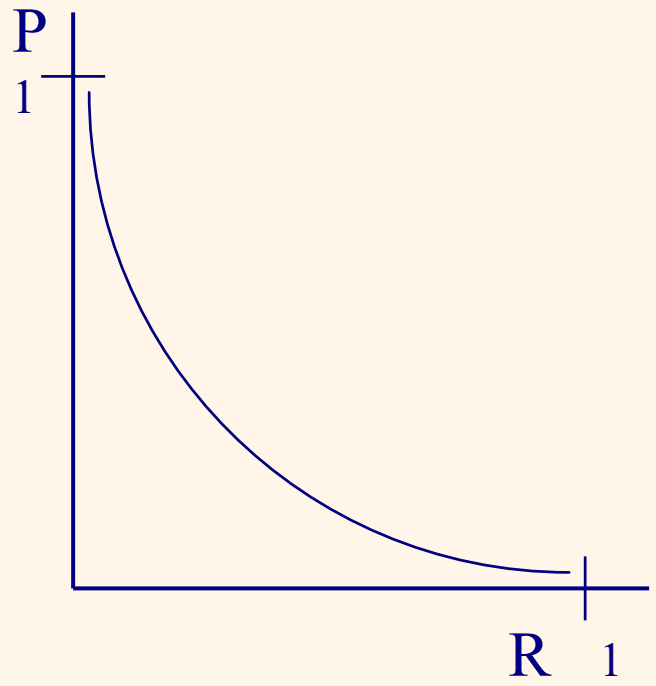
$$R = \frac{\#\text{vybraných relevantních záznamů}}{\#\text{relevantních záznamů v souboru}}$$

koeficient přesnosti P (z angl. precision)



$$P = \frac{\#\text{vybraných relevantních záznamů}}{\#\text{vybraných záznamů}}$$

Měření relevance



Boolský model

- reprezentace dokumentů: pomocí množin termů
- dotazování:
  - formálně: pomocí Boolských výrazů
  - způsob: na přesnou shodu
- nalezení termů - praxe:
  - odstranění nevýznamových slov (stop-words) z množiny termů

    výsledek: redukce 30-50% (C.J. van Rijsbergen)
  - lingvistické zpracování (tokenizace)

Boolský model

Jedna z možných syntaxí:

<term>

<jméno_atributu> = <hodnota_atributu> /porovnání/

<jméno_funkce>(<term>), /aplikace funkce/

X AND Y	vyber D, obsahující jak X, tak Y.
X OR Y	vyber D, obsahující buď X nebo Y.
X XOR Y	vyber D, obsahující buď X nebo Y ale ne X AND Y
NOT Y	vyber D, neobsahující Y
X adj Y	vyber D, ve kterých se vyskytuje X následovaný Y
X (n)words Y	vyber D, ve kterých se vyskytuje X následovaný Y nejdále ve vzdálenosti *n* slov
X sentence Y	vyber D, ve kterých se vyskytuje X a Y ve stejné větě

Boolský model

. odpovídá libovolnému znaku.

* znak následovaný * odpovídá libovolnému počtu výskytů (včetně nulového) tohoto znaku. Např. xy* odpovídá x, xy, xyy atd.

+ znak následovaný + odpovídá libovolnému počtu výskytů (kromě prázdného) tohoto znaku. Např. xy+ odpovídá xy, xyy, xyyy atd.

[] Znaky v [] odpovídají libovolnému jednomu znaku, který je v závorkách uveden, ale ne jinému. Např. [xyz] odpovídá x, y nebo z.

[^] ^ na začátku řetězce v [] znamená negaci (not). Např. [^xyz] odpovídá libovolnému znaku kromě x, y nebo z.

[-] – mezi znaky v [] označuje rozsah znaků. Např. [a-x] odpovídá libovolnému znaku od a do x.

Boolský model: P vs. R

- Upřesňováním dotazu v Boolském modelu získáváme větší P, ale menší R.

Př.: pokus (Blair, Maron,1985) - 40000 právnických textů

Cíl: nejen vysoké P, ale i R.

Výsledky: P $\rightarrow$ 80%, R $\rightarrow$ 20%

Problém synonym - obecný jazyk, nelze podchytit tezaurem.

Př.: nehoda, neštěstí srážka, karambol, „něco se tam stalo“, …

- automatická indexace neodstraní tyto problémy

Boolský model: problémy

Co ovlivňuje vztah P a R?

Problémy s ručně indexovanými systémy:

neurčitost

- v indexování — *vliv indexátora*
- ve výběru termů pro dotaz — *vliv tazatele*

Př.: p_1, p_2 pravděpodobnosti, že uživatel užije termy t_1, t_2

q_1, q_2 pravděpodobnosti, že termy t_1, t_2 se vyskytují v D

$\Rightarrow$ p, že tazatel zvolí t_1, t_2 a vyhledá se D s t_1, t_2, je

$$p_1 * p_2 * q_1 * q_2$$

např. R = 0,6 * 0,7 * 0,5 * 0,6 = 0,126 $\Rightarrow$ R < 13%

$\Rightarrow$ pro i=5, $p_i = q_i = 0,5 \Rightarrow$ R = 0,1%

$\Rightarrow$ je-li 1000 relevantních D, vybere se 1 !

Boolský model: problémy

kritérium predikce - jak zajistit shodu mezi výběrem termů pro dotaz a dokumenty (dnes: podobnost ontologií)

- metoda: odstraňování neurčitosti

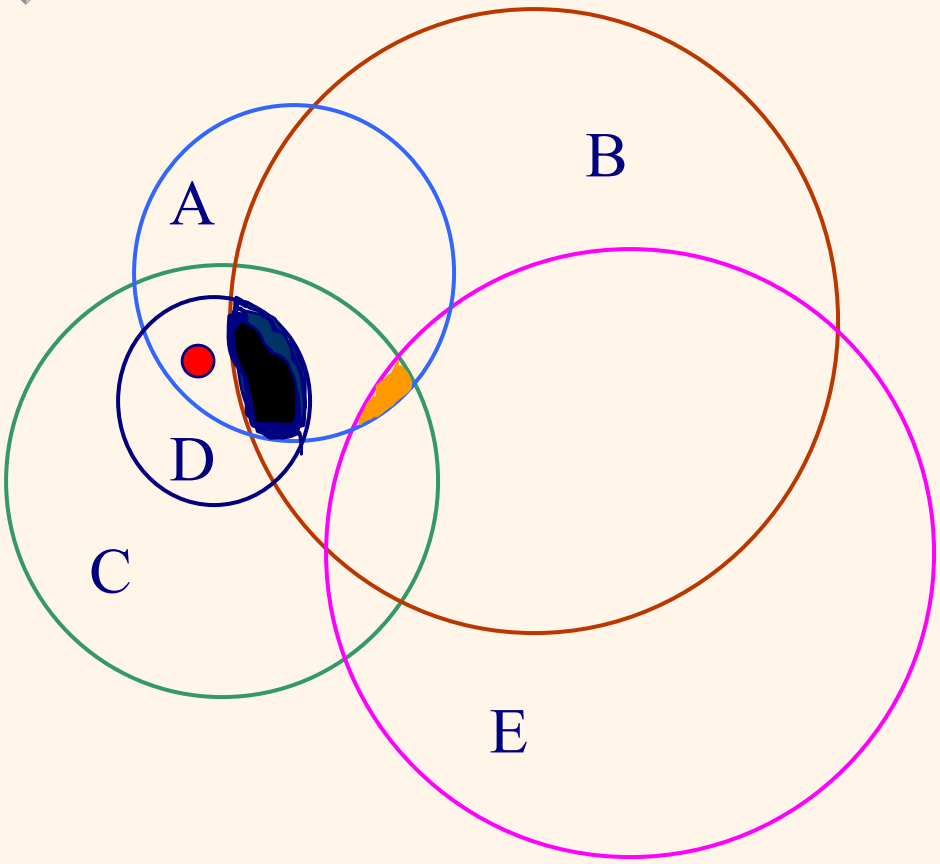
kritérium maxima - lze zvládnout 20-50 hitů

Problémy s db úplných textů:

- *velikost db* (vs. kritérium maxima)
- *výběr termů pro dotaz*
  - přecenění eliminace indexátorů
  - zůstává neurčitost tazatele
- *jednostranné chování tazatele* -

  tendence měnit poslední rozhodnutí, zachovávat první kroky

Boolský model: problémy



hit

$A \cap B \cap C \cap D$

$A \cap B \cap C \cap E$

Boolský model: problémy

Řešení neurčitosti ve výběru termů pro dotaz:

- najdeme D s vysokou relevancí pro uživatele (D je znám + je známo, že je v db),
- termy pro dotaz jsou vybrány z D,
- odstraňování termů resp. jejich nahrazování disjunkcemi.

$\Rightarrow$ zmenšování neurčitosti tazatele

Boolský model: problémy

Řešení jednostranného chování tazatele vážením:

Př.:	*termy*		*pravděpodobnost (váha)*
	Autor: Pokorný		0,3
	Datum: 1995-1999		0,7
	Časopisy:	CW	0,2
		Artificial Intelligence	0,5
		ERCIM News	0,2
	Předmětová hesla:	XML	0,6
		databáze	0,8
		dotazovací jazyky	0,9

Celkový počet konjunktivních dotazů je 255

Boolský model: problémy

Součiny pravděpodobností pro

2 termy	3 termy	max. pro 1, 2, ...
$p_{do} * p_{da} = 0{,}72$	$p_{do} * p_{da} * p_{dat} = 0{,}5$	0,9
$p_{do} * p_{dat} = 0{,}63$	$p_{do} * p_{dat} * p_{xm} = 0{,}38$	0,72
$p_{da} * p_{dat} = 0{,}56$	$p_{do} * p_{da} * p_{ar} = 0{,}4$	0,5
…	…	0,3
		0,15

Algoritmus:
- vytvoř skupiny pro všechny kombinace
- spočti pro skupiny maxima
- je splněno kriterium maxima?
- nabídka tazateli

Boolský model: další problémy

- Neintuitivní výsledky
  - A AND B AND C AND D AND E

    D neobsahující pouze jeden z uvedených termů nebude vybrán.
  - A OR B OR C OR D OR E

    D obsahující pouze jeden z uvedených termů jsou chápány jako stejně významné jako dokumenty obsahující všechny uvedené termy.
- Neumožňuje řízení velikosti výstupu.
- Všechny D vyhovující dotazu jsou chápány jako stejně důležité, není možné je uspořádat podle hodnoty relevance.

Boolský model: další problémy

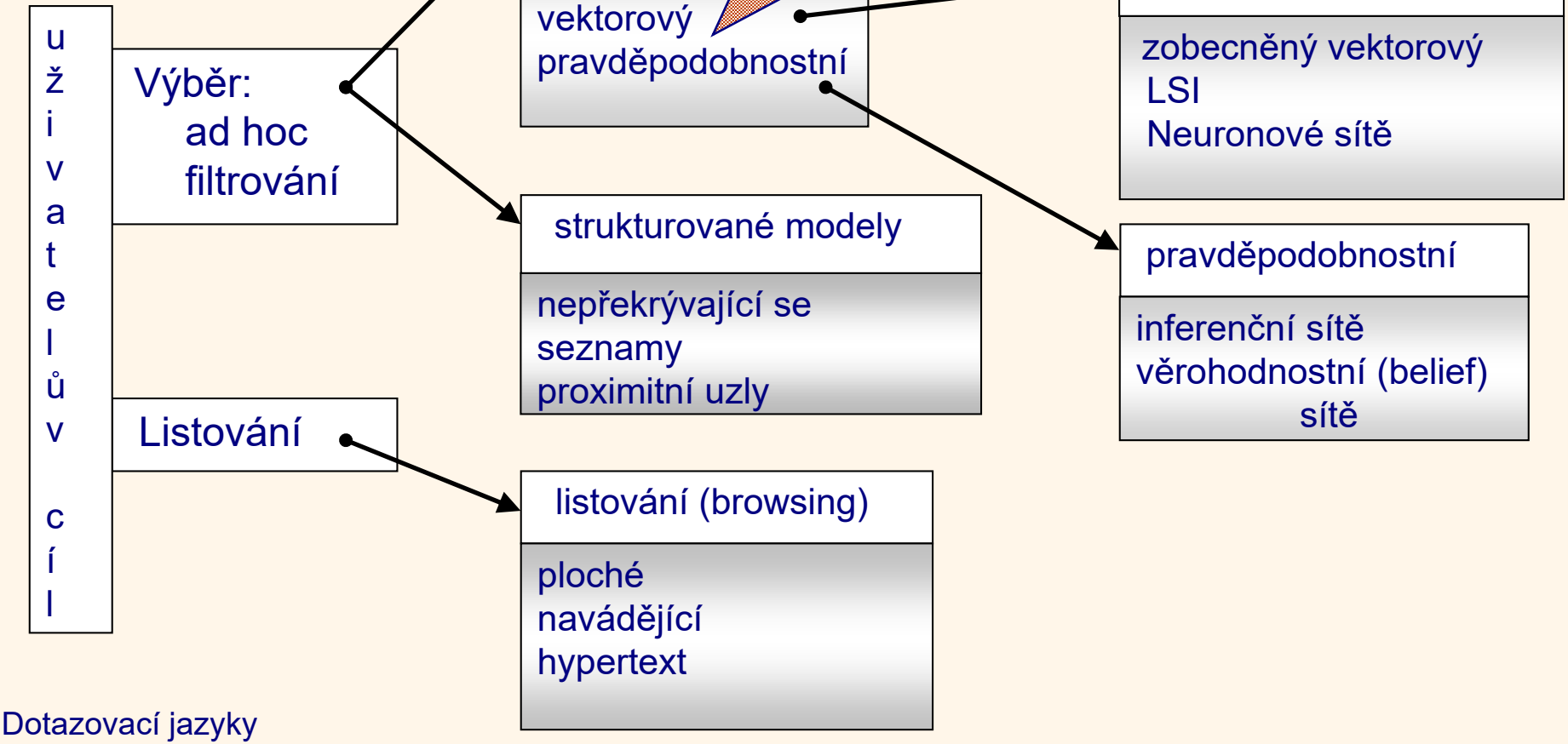
- Obtížně lze realizovat automatickou zpětnou vazbu, tj. na základě D označených v odpovědi za relevantní automaticky modifikovat dotaz.
- Vyjadřovací síla Boolského modelu je omezená. Jakákoliv množina {D} popsatelná pomocí termů, může být v principu vybrána vhodným Boolským dotazem. Není ale garantováno, že pro jakoukoliv množinu {D}, které jsou v uživatelově zájmu, je v praxi jednoduché formulovat Boolský dotaz.
- Spíše umění než věda.

Jak dál

Teze:

klasické Boolské systémy lze rozšířit o funkce ovlivňující kriterium maxima; nelze však současně dosahovat vysokého P i R bez přídavných informací.

Přehled IR modelů



Vektorový model

Předpoklad: kolekce *m* dokumentů **D**, *n* různých termů $t_1 ... t_n$

Každý dokument $D_i \in$ **D** je reprezentován vektorem

$$D_i = (w_{i1}, w_{i2}, ..., w_{in}), \text{ kde } w_{ij} \in \langle 0;1 \rangle$$

kde w_{ij} je váha náležející termu t_j v identifikaci dokumentu D_i.

D je reprezentovatelná maticí

$$\mathbf{D} = \begin{matrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \dots & & & \\ \dots & & & \\ w_{m1} w_{m2} & & \dots & w_{mn} \end{matrix}$$

Vektorový model

- dotazování:
  - formálně: pomocí vektoru dotazu
  - dotazování na částečnou shodu
    způsob: pomocí funkce (koeficientu) podobnosti

výraz dotazu Q ve vektorovém modelu

$$Q = (q_1, q_2, ..., q_n), \text{ kde } q_j \in \langle 0;1 \rangle.$$

Problém: jak počítat podobnost

Vektorový model

Úhel vs. vzdálenost

- Proč ne vzdálenost?
- Experiment: vezmeme dokument D a připojíme ho ještě jednou k D. Vznikne dokument D′.
- “Sémanticky” mají D a D′ stejný obsah.
- Euklidovská vzdálenost mezi body v prostoru mezi D a D′ (body prostoru) by byla velká
- Úhel mezi D a D′ (jako vektorů) je 0, koresponduje maximální podobnosti.
- Klíčová idea: Pořadí dokumentu D podle úhlu, který svírá s vektorem dotazu Q.
- Vhodná: cosinus - klesající funkce v intervalu [0^o, 180]

Vektorový model

koeficient podobnosti (angl. *similarity*) dotazu Q a dokumentu D_i

(a) $Sim(Q,D_i) = \sum_{k=1,..,n}(q_k * w_{ik})$ (*skalární součin*)

(b) $Sim(Q,D_i) = \sum_{k=1,..,n}(q_k * w_{ik}) / \sqrt{(\sum_{k=1,..,n}(w_{ik})^2 * \sum_{k=1,..,n}(q_k)^2)}$ (*kosinová míra*)

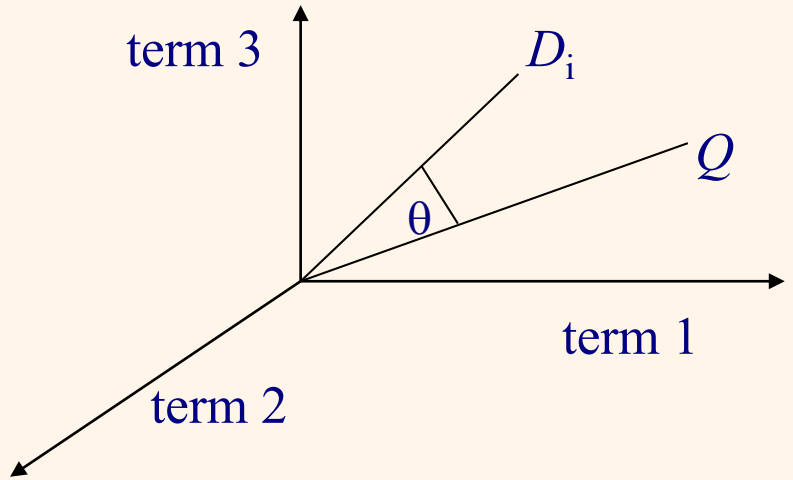
Dělitel v (b) je *normalizační faktor*,

(c) $Sim(Q,D_i) = 2\sum_{k=1,..,n}(q_k * w_{ik}) / (\sum_{k=1,..,n}(w_{ik})^2 + \sum_{k=1,..,n}(q_k)^2)$ (*Diceův koeficient*)

Postulát: Dokumenty, které jsou ve vektorovém prostoru „blízko sebe“ vypovídají o stejných věcech

Vektorový model

geometrická interpretace



Pz.: binární vektorový model (tj. jediné nenulové w_{ik} v D_i i Q jsou rovny 1).

Pro všechny tři případy *Sim* =

- $|Q \cap D_i|$
- $(|Q \cap D_i|)(\sqrt{|Q|} * \sqrt{|D_i|})$
- $2(|Q \cap D_i|)(|Q| + |D_i|)$

Vektorový model

Výhody: R i P lze zvýšit až o 20%.

Pragmatický přístup: jednoslovné termy + vhodná metoda vážení

TF_{ij} *frekvence termu* t_j *v dokumentu* D_i

NTF_{ij} *normalizovaná frekvence termu* t_j *v dokumentu* D_i

$((TF_{ij}/\max TF_{ik})+1)/2$

kde max je přes všechny termy v *i*-tém řádku matice **D**.

Nevýhoda: term s vysokou TF v mnoha D_i $\Rightarrow$ nízký P

Vektorový model

IDF *inverzní frekvence termu v dokumentech*

klesá se zvyšujícím se počtem dokumentů, ke kterým je term přiřazen.

IDF pro term t_j je definována jako

$$IDF_j = \log(m/DF_j) + 1$$

kde m je celkový počet dokumentů v **D** a DF_j je frekvence t_j v **D**, tj. počet dokumentů, ke kterým je term t_j přiřazen.

Pz.:

- Pro řazení dokumentů není základ logaritmu důležitý
- IDF je skutečně inverzní vzhledem k DF.

Vektorový model

Chování:

- term se vyskytuje ve všech dokumentech $\Rightarrow$ log(1) = 0 (term patří mezi nevýznamová slova)
- term se vyskytuje pouze v 1 dokumentu $\Rightarrow$
- $IDF = \log m + 1$
- Př.: pro $m = 10$ je $IDF = 2$, pro $m = 10\,000$ je $IDF = 5$ atd.

Vektorový model

TD rozlišení pomocí termů (vysoké TF i IDF)

$TD_{ij} = TF_{ij} * IDF_j$ nebo $TD_{ij} = NTF_{ij} * IDF_j$

Značení v literatuře: tf-idf, tf.idf, tf x idf

Pak: w_{ij} *se definuje jako* TD_{ij}

Pz.: nevyplatí se udržovat příliš malé váhy termů (k prahové hodnotě).

Nejlepší váhy v *Q*:

$q_k = (0{,}5 + (0{,}5 * TF_k)/\max TF) * IDF_k$

kde TF_k je frekvence termu t_k v *Q*, max *TF* je maximální frekvence nějakého termu v *Q* a IDF_k je *IDF* termu t_k v **D**.

Vektorový model

Speciální případy pro *Q* a **D**:

- zadána pouze množina termů $\Rightarrow q_k = IDF_k$
- dlouhé dotazy aproximace $q_k = TF_k$
- krátké dokumenty $\Rightarrow$ aproximace vah pomocí 0, 1
- dlouhé dokumenty $\Rightarrow$ jednotkou výběru *pasáž*

Vektorový model: problémy

- předpoklad: nezávislost termů
- chybějící syntaktická informace (fráze, pořadí slov, vzdálenosti)
- chybějící sémantika: polysémie (2 slova stejně zní a jejich významy mají nějakou genetickou souvislost), synonymita stále neřešeny

Historie: součást systému SMART (1970)

Dnes: open source software Apache Lucene (od r. 1999) – kombinuje vektorový a Boolský model

Vektorový model v Boolském systému - příklad implementace

Předpoklady:

- soubor indexů s invertovanými seznamy
- v invertovaných seznamech TF_{ji} (modelujeme jimi w_{ji})
- soubor obsahující IDF_j
- soubor SKÓRE[1:m]
- Váhy termů dotazu jsou rovny 1

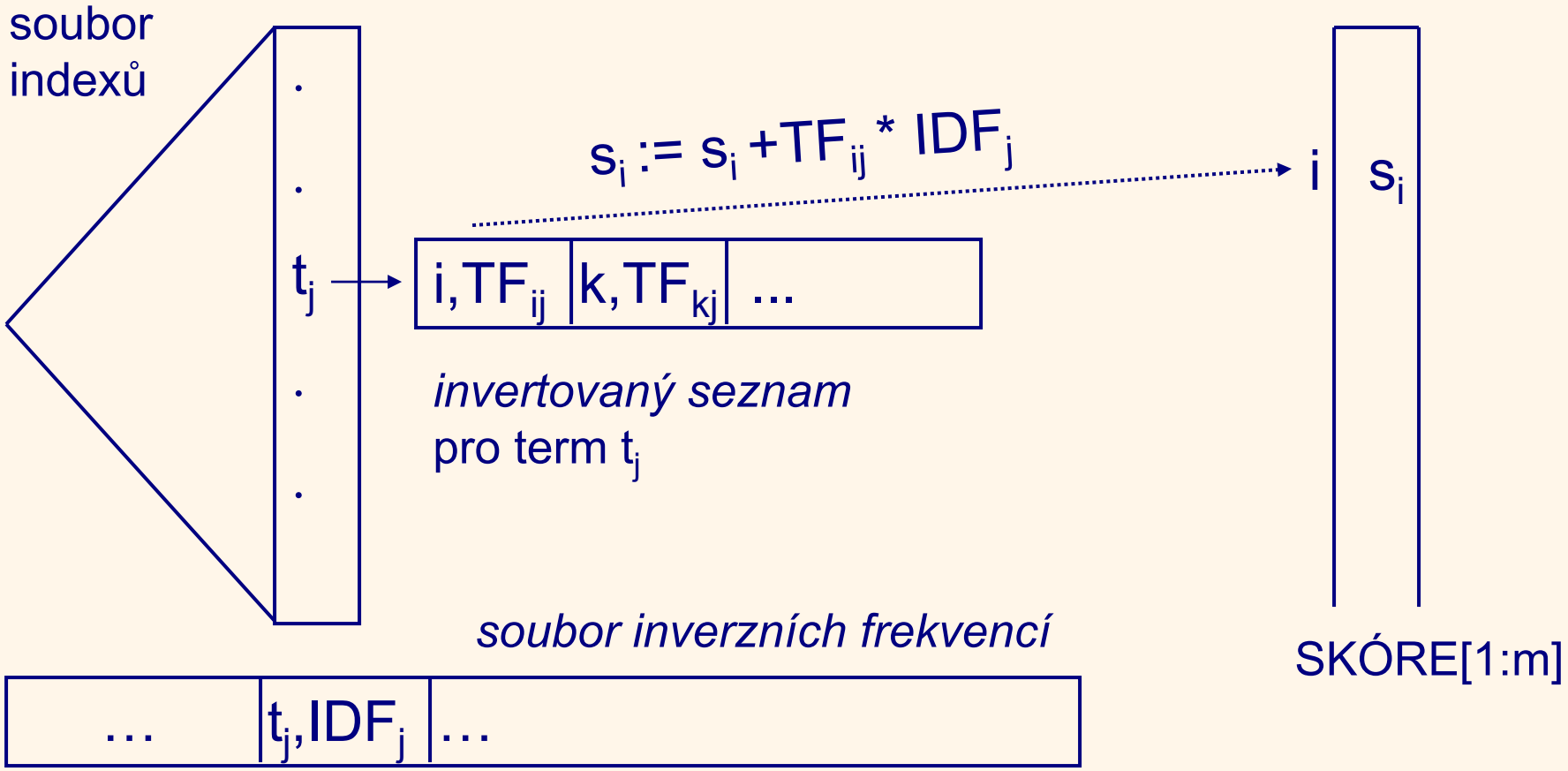
Algoritmus:

(1) podle termů dotazu přistupuj invertované seznamy.

(1.1) Oprav součty v SKÓRE

(2) Setřiď SKÓRE a vydej např. 20 nejvyšších.

Vektorový model v Boolském systému - příklad implementace



Vektorový model a signatury - příklad implementace

Předpoklady:

- D_j má b_j bloků, dotaz má Q termů
- soubor signatur - pro každý blok existuje signatura
- soubor obsahující IDF_i (modelujeme jimi q_i - stačí DF)
- soubor SKÓRE[1:20] (udržuje se 20 nevyšších)

Algoritmus: Pro všechny D proveď:

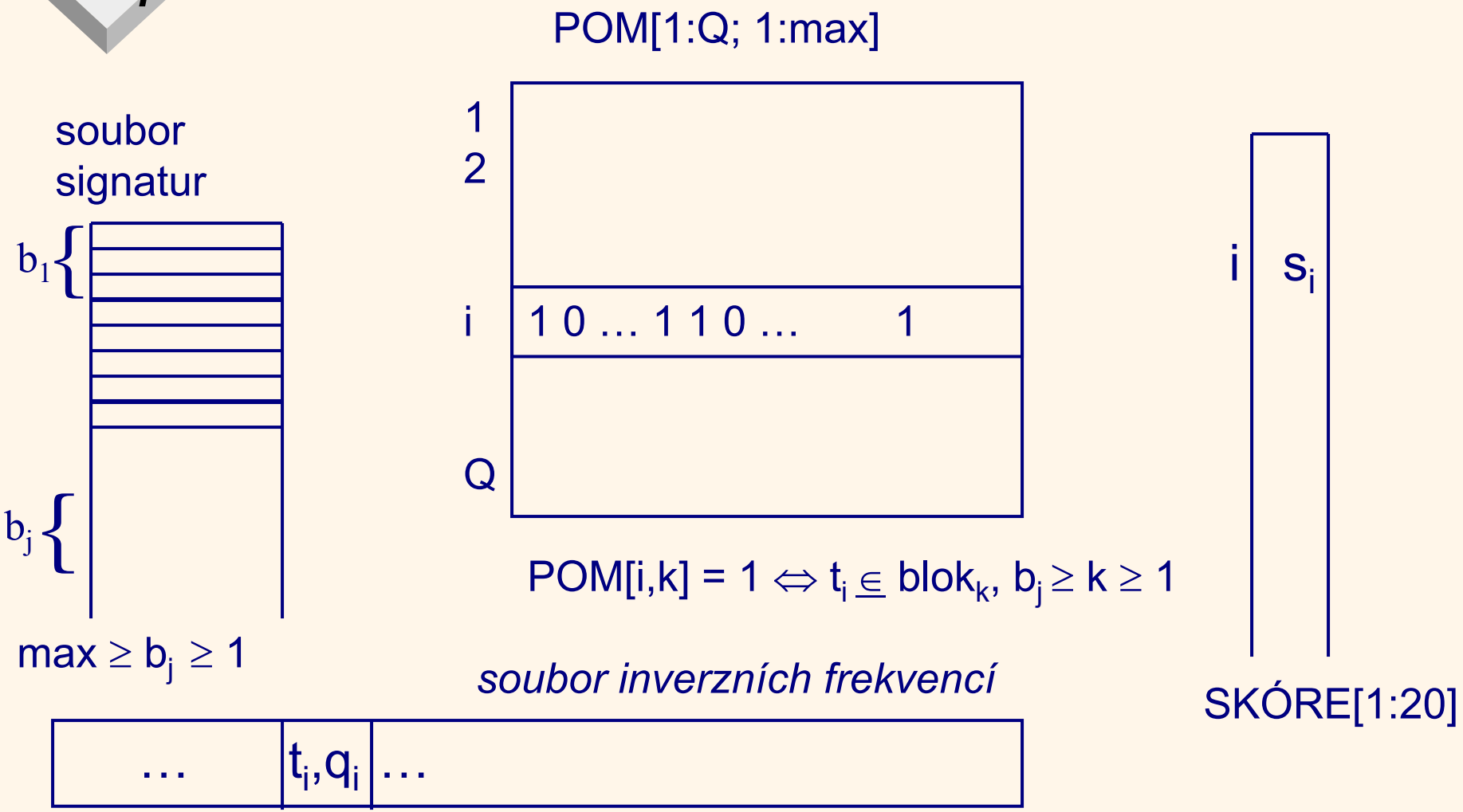
(1) Vynuluj POM.

(2) Signaturu každého z b bloků textu D porovnej s Q signaturami dotazu. Výsledky ulož do POM.

(3) Pro každý t_i dotazu spočti $bc_i = \Sigma_{j=1\ldots bmax} POM[i,j]$

(4) Spočti $s = \Sigma_{i=1\ldots Q}(bc_i * q_i)/b$

Vektorový model a signatury - příklad implementace



Složitost indexování vektorovým modelem

- vytváření vektorů a indexování dokumentu o *n* jednotkách je O(*n*).
- indexování *m* takových dokumentů je O(*m n*).
- počítání IDFs lze dělat při témže průchodu
- počítání délek vektorů je také O(*m n*).
- $\Rightarrow$ celková časová složitost je O(*m n*)

Techniky pro "inteligentní" IR

1. Zpětná vazba
   - přímá zpětná vazba
   - pseudo zpětná vazba
2. rozšiřování dotazu
   - „přirozeným“ tezaurem
   - „umělým“ tezaurem

Výhody: zvyšují R, ale jen zřídka P.

Zpětná vazba

Intuice:

- vektory relevantního dokumentu a dotazu si jsou podobné
- vektory nerelevantního dokumentu a dotazu si nejsou podobné;

$\Rightarrow$ *reformulace dotazu* na základě odpovědi na dotaz

Předpoklady: vektor dotazu $\vec{q}$

odpověď obsahuje relevantní $D_1^r, \ldots, D_{mr}^r$

nerelevantní $D_1^n, \ldots, D_{mn}^n$

Zpětná vazba

$$\vec{q}' = \alpha\vec{q} + \frac{\beta}{m_r}\Sigma_{i=1\ldots mr}\vec{D}_i^r - \frac{\gamma}{m_n}\Sigma_{i=1\ldots mn}\vec{D}_i^n$$

pro α=1 Rocchio 71

$$\vec{q}' = \alpha\,\vec{q} + \beta\,\Sigma_{i=1\ldots mr}\vec{D}_i^r - \gamma\,\Sigma_{i=1\ldots mn}\vec{D}_i^n$$

pro α= β= γ =1 Ide 71

$$\vec{q}' = \alpha\vec{q} + \beta\,\Sigma_{i=1\ldots mr}\vec{D}_i^r - \gamma\,\vec{D}_1^n$$

kde α, β, γ jsou vhodné konstanty

Zpětná vazba - inkrementálně

REPEAT

Systém vybere D s max. SIM(Q,D);

Tazatel označí D za relevantní nebo nerelevantní;

IF D je relevantní THEN D jde do výstupního seznamu;

$\vec{q}$ se modifikuje pomocí $\vec{D}$;

UNTIL φ

modifikace dotazu:

$$\vec{q}_{j+1} = \begin{cases} \alpha\vec{q}_j + \beta\vec{D}_j & D_j \text{ je relevantní} \\ \alpha\vec{q}_j - \gamma\vec{D}_j & D_j \text{ je nerelevantní} \end{cases}$$

Pz.: vybírá se vždy D, který ještě nebyl vybrán.

Zpětná vazba – další možnosti

převážení termů: zvýšení vah termů v relevantních dokumentech a snížená vah termů v nerelevantních dokumentech

pseudozpětná vazba: předpokládej k-prvních dokumentů jako relevantních a podle nich dej upravit dotaz.

Rozšíření dotazu pomocí tezauru

- *tezaurus* (též thesaurus, starořecky poklad, pokladnice) poskytuje informace o synonymech a sémanticky vztažených slovech a frázích.
- Př.: Eurovoc – pro oblast práva a legislativy, je od r. 2005 i pro češtinu.

Tezaurus

Výrazy s použitím tezauru (standard ISO-2788)

NT('text')	NARROWER TERM o úroveň užší term
NT('text',n)	užší pojmy o *n* úrovní
NT('text',*)	všechny užší pojmy
BT('text')	BROADER TERM o úroveň širší term
BT('text',n)	širší pojmy o *n* úrovní
BT('text',*)	všechny širší pojmy
TT('text')	TOP TERM - nejširší term
SYN('text')	SYNONYMS - synonyma
PT('text')	PREFERRED TERM preferovaný term
RT('text')	RELATED TERMS - příbuzné termy

Tezaurus

Další relace:

SN (scope note) - poznámka připojená k danému termu,

USE - k danému termu přiřazuje jeho preferovaný term,

UF - k danému termu přiřazuje jeho synonymní (nepreferovaný) term

Další standard (pro textové DB):

ANSI Z39.58 Common Command Language for Online Interactive Information Retrieval - vyvinuty institucí NISO (National Information Standards Organization) v r. 1992.

Pz: skutečné jazyky jsou pouze podobné těmto standardům

Příklad: Wordnet

- detailnější databáze semantických vztahů mezi slovy (pro angličtinu, …, češtinu).
- vyvinuta Prof. George Millerem a jeho týmem na univerzitě v Princetonu.
- okolo 150,000 anglických slov.
- Podstatná jména, přídavná jména, slovesa a příslovce seskupená do cca 110,000 synonymních množin zvaných *synsety*.

Příklad: Wordnet

Příklady typů vztahů:

- antonyma (opozita): vpředu → vzadu
- atributace: dobročinnost → dobrý (od podstatného jména k přídavnému)
- podobnost: bezpodmínečný → absolutní
- příčina: zabití → úmrtí
- holonyma: kapitola → text (být částí)
- meronyma: počítač → cpu (být částí)
- hyponyma (podřízené pojmy): strom → rostlina (specializace)
- hyperonyma (nadřazené pojmy): ovoce→ jablko (generalizace)

Příklad: Wordnet

- Měření sémantické podobnosti a vztaženosti zavedené pro WordNet Pedersonem, et al v r. 2005 – (software WordNet::Similarity)
- koeficienty podobnosti
  - založené na délkách cest:

    Lch, wup, Path
  - založené na informačním obsahu:

    res, lin, jcn
- koeficienty vztažnosti
  - hso, lesk, vector

Texty v SQL: Textový extender (v DB/2)

```
CREATE TABLE ČLÁNKY(
            časopis          VARCHAR(50),
            titul            VARCHAR(50),
            datum            DATE,
            text_článku      FULLTEXT)

SELECT časopis, datum, titul
FROM ČLÁNKY
WHERE CONTAINS(text_článku, '("databáze" AND
            ("SQL" | "SQL92") AND NOT "dBASE")') = 1;
```

D: Najdi všechny články v časopisech, které obsahují v textu „objektově-relační“ ve stejné větě jako slovo „databáze.“

```
SELECT  časopis, titul
FROM ČLÁNKY
WHERE CONTAINS(popis, '"databáze“ IN SAME
                      SENTENCE AS "objektově-relační“')
```

Texty v SQL: Textový extender (v DB/2)

Další funkce: NO_OF_MATCHES (kolikrát se zadaný vzorek vyskytoval v textu), RANK (hodnota pořadí v odpovědi na základě nějaké míry).

```
SELECT časopis, titul
FROM ČLÁNKY
WHERE NO_OF_MATCHES (text_článku, ‘databáze‘) > 10;
SELECT časopis, datum, titul, RANK(text_článku, ‘(“databáze”
                AND (“SQL” | “SQL92”) )’) AS relevantní
FROM ČLÁNKY
ORDER BY relevantní DESC;
```

možnost různých implementací

Texty v SQL: Fulltext v MySQL 5.1

Typy FT vyhledávání:

- Boolské
- FT s indexem

```
CREATE TABLE ČLÁNKY (
časopis      TEXT
text_článku VARCHAR(200)
FULLTEXT (časopis, text_článku)
) engine=MyISAM

SELECT *
FROM ČLÁNKY
WHERE MATCH(časopis, text_článku)
AGAINST('database' IN NATURAL LANGUAGE MODE);
```

FULLTEXT je typ indexu

paměťový stroj
další: InnoDB,…

Třídění výsledku: implicitně dle relevance

Texty v SQL: Fulltext v MySQL 5.1

Typy FT vyhledávání:

- Boolské
- FT s indexem

```
SELECT *
FROM ČLÁNKY
WHERE MATCH(časopis, text_článku)
AGAINST('+database –relational' IN BOOLEAN MODE);
```

Třídění výsledku:

- + (AND), - (NOT), žádný operátor (OR)
- implicitně žádné třídění

Texty v SQL: SQL/MM – Full-text

- Konstruktory
  - Řetězec znaků
  - Řetězec znaků + zadání jazyka
- Konverzi do běžných SQL znak. řetězců
  - FullText_to_Character
- Vyhledávací metody
  - Ano/Ne (CONTAINS)
  - Rank (RANK)

Texty v SQL: Full-text - vyhledávání

- Vzorek (+ wildcards)
- Odvozená slova (STEMMED)
- Slova s podobným nebo stejným významem (THESAURUS, SYNONYM)
- Stejně znějící slova (SOUNDS LIKE)
- Dle pozice v textu (NEAR, …)
- Dle konceptu textu (IS ABOUT)

Texty v SQL: Full-text - vyhledávání

```
CREATE TABLE INFORMACE (
  číslo_dokumentu   INTEGER,
  dokument          FULLTEXT
);
```

Texty v SQL: Full-text - vyhledávání

```
SELECT číslo_dokumentu
  FROM INFORMACE
  WHERE dokument.CONTAINS
  (
    STEMMED FROM OF “standard”
    IN SAME PARAGRAPH AS
    SOUNDS LIKE “sequel”’
  ) = 1;
```

Závěr

Současné (nové) aplikace:

- klasifikace textů
- extrakce (sumarizace) textů
- digitální knihovny
- vyhledávání na Webu
- multilingvální prostředí
- detekce spamu
- plagiátorství textů