



Információ-visszakereső technológiák

Góth Júlia

goth.julia@itk.ppke.hu

(4 kredit)

(Számonkérés módja: szóbeli vizsga)

4. ELŐADÁS

1. Általános információ-visszakereső rendszer architektúrája.
2. Objektumbázis-modul.
3. Indexelő modul.
4. Lekérdező modul.
5. Rangsoroló modul.
6. A rendszer kategoricitási tulajdonsága.

1. Általános információ- visszakereső rendszer architektúrája.

1. Az információ-visszakeresés részfolyamatai
2. Általános információ-visszakereső rendszer elemei

1. 1. Az információ-visszakeresés részfolyamatai -1.

Technikai oldalról az információ-visszakereső rendszer áll:

- egy objektumbázisból (pl. weboldalak), és
- egy keresőfelületből (a keresőfelületen keresztül lekérdezések végrehajtásával érhetjük el az objektumbázisban tárolt objektumokat).

A releváns információ hatékony visszakeresését:

- a felhasználói munka (az információigény megfelelő átalakítása keresőkérdéssé),
- és a dokumentumok logikai nézete (logical view) együttesen befolyásolják.

1. 1. Az információ-visszakeresés részfolyamatai -2.

Az információ-visszakeresési folyamat több
részfolyamatból áll:

1. Objektumbázis definiálása
2. Szövegfeldolgozó műveletek alkalmazása
3. Szöveg indexelése
4. Lekérdezés
5. Rangsorolás
6. Találati lista visszaadása.

1. 1. Az információ-visszakeresés részfolyamatai -3.

1. Objektumbázis (dokumentumbázis) definiálása:
 - dokumentumok,
 - szövegfeldolgozó műveletek,
 - szövegmodell (szöveg szerkezete, visszakereshető elemek) megadása.
2. A szövegfeldolgozó műveletek átalakítják a dokumentumokat, és elkészítik azok logikai nézetét.

1. 1. Az információ-visszakeresés részfolyamatai -4.

3. Ezután indexelni kell a szöveget, amit már az indexelő modul végez. Az index kritikus paraméter: nagy terjedelmű szöveg gyors keresését teszi lehetővé. Többféle fájl struktúra (file structure) használható a tárolására (a legelterjedtebb az inverz fájl (inverted file)).
4. A lekérdezés folyamán az indexek és a keresőkérdés kifejezéseinek összehasonlítása történik.

1. 1. Az információ-visszakeresés részfolyamatai -5.

5. A kapott eredmények alapján végzi el a rangsoroló modul a hasonlósági értékek kiszámítását,
6. és adja vissza a felhasználónak a találati listát.

1.2.Általános információ- visszakereső rendszer elemei -1.

Tehát egy általános információ-visszakereső rendszer az alábbi elemekből állhat:

- **OBJEKTUMBÁZIS:** tárolja a dokumentumok halmazát, amelyen a keresés történik. A dokumentumok az bázisba manuálisan, vagy speciális számítógépprogramok segítségével kerülnek.

1.2.Általános információ- visszakereső rendszer elemei -2.

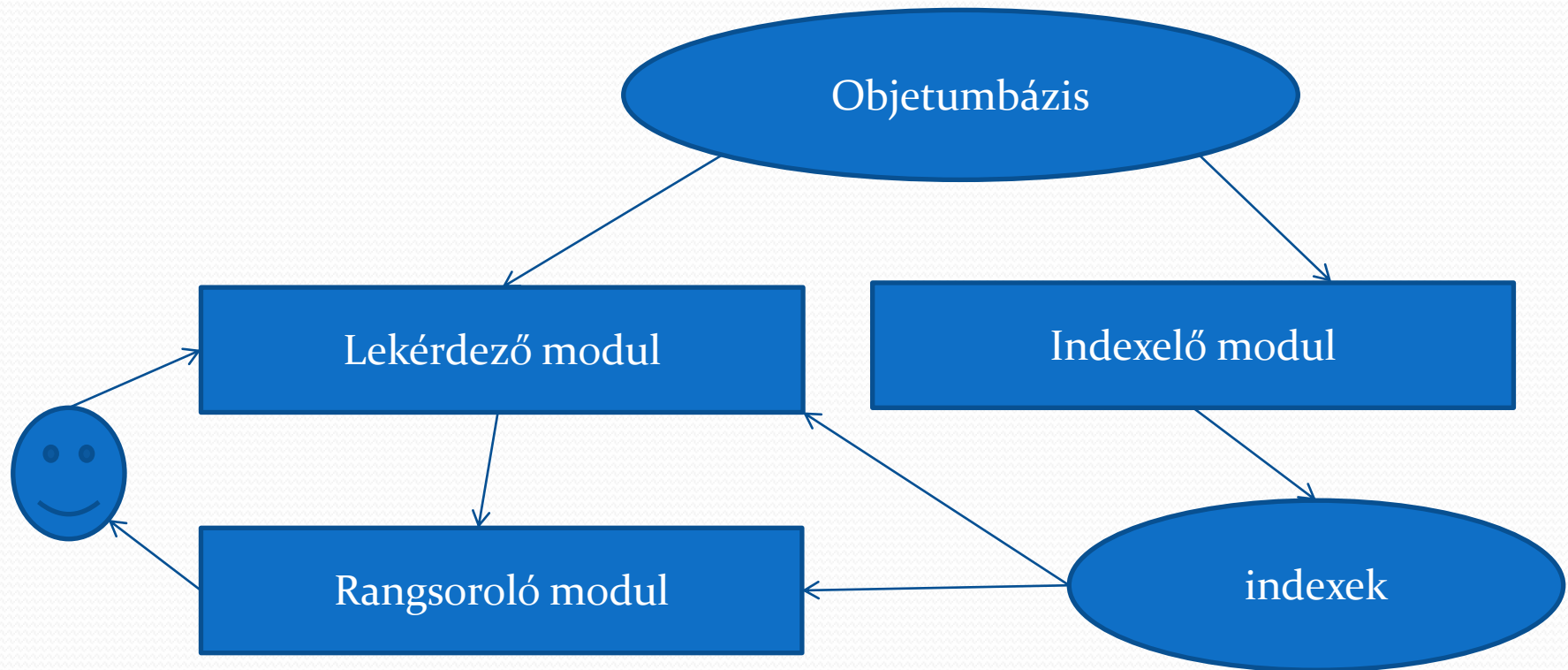
- INDEXELŐ MODUL. Az objektumbázisban lévő dokumentumokból az indexelő modul készíti az indexeket inverz fájl struktúrában. Ezeket a struktúrákat használja majd a lekérdező modul, hogy megtalálja a felhasználó által feltett keresőkérdésre a válasz dokumentumokat.
- LEKÉRDEZŐ MODUL. A lekérdező modul olvassa be a keresőkérdést, és alakítja át a felhasználható formára. A lekérdező modul az indexeket használja a felhasználói kérdésre megfelelő válaszok megtalálására. Ezután a talált dokumentumokat átadja a rangsoroló modulnak.

1.2.Általános információ- visszakereső rendszer elemei -3.

- RANGSOROLÓ MODUL. A rangsoroló modul kiszámítja a lekérdező modul által talált dokumentumokra a hasonlósági értékeket (az indexek felhasználásával). Majd a hasonlósági értékek alapján csökkenő sorrendbe rendezi a dokumentumokat, és ezt találati listaként megjeleníti a felhasználónak.

1.2.Általános információ- visszakereső rendszer elemei -4.

Általános információ-visszakereső rendszer elemei:



2.Objetumbázis-modul

2.Objektumbázis-modul -1.

Az információt objektumokban keressük.

$O = \{O_1; ::::; O_j ; ::::; O_m\}$ jelölje az objektumok véges halmazát, ahol az objektum lehet:

- szöveg (könyv, újságcikk, előadás jegyzet, címek, stb...)
- kép (fényképek, rajzok, stb...)
- hang (dalok, zenei művek, stb...)
- multimédia (szöveg kép és hang együttesen).

2. Objektumbázis-modul -2.

- Az információ-visszakeresés céljából minden O_j objektum megadható egy szöveges D_j dokumentumként.

A továbbiakban tekintsük a D_j -t az O_j objektum helyett, minden j esetben. Nyilvánvaló, hogy ezen a D_j dokumentumon az O_j objektumot értjük.

2. Objektumbázis-modul -3.

Nem szöveges objektum is megadható szöveges dokumentumként.

Ha O_j objektum pl. egy képi objektum (családi fénykép), akkor a hozzá tartozó D_j dokumentum lehet egy szöveg, ami tartalmazza például:

- a fényképen szereplő személyek nevét,
- a fénykép milyen eseményen készült, az esemény részletes leírása,
- a fénykép mikor készült,
- a fénykép hol készült,
- a fényképet ki készítette.

3. Az indexelő modul

- 3.1. Kulcsszó meghatározás
- 3.2. A szövegfeldolgozás lépései
- 3.3. Stoplista
- 3.4. Szótövesítés
- 3.5. Inverz file struktúra

Ismétlés-indexelés

- **Index:** ókori Rómába vezethető vissza: minden papirusztekercshez csatoltak egy kis kártyát, ami a mű címét tartalmazta, így egy papirusztekercset felesleges mozgatás nélkül lehetett beazonosítani. Idővel az index a mű rövid kivonatát is tartalmazta. Az első indexeknél a betűrendbeszedés még csak a szavak első betűje alapján történt. A teljes betűrendbeszedés csak a 18. században vált szabállyá.

Az *indexálás* egy olyan eljárás ami megadja:

- egy azonosítónak (szó, kifejezés, név, tárgy, kód),
- egy egységben (könyv, adatbázis) elfoglalt
- pontos helyét (fejezet, oldal, sor).

Ez az inverzfájl struktúra (inverted file structure) alapja.



3.1. Kulcsszó meghatározás -1.

- A szöveges dokumentumokban az információ többnyire strukturálatlan módon tárolódik.
- A dokumentumokat legtöbbször indexkifejezések bizonyos struktúráival reprezentáljuk.

Az indexkifejezéseket:

- vagy automatikusan, közvetlenül a dokumentumok szövegéből nyerjük,
- vagy szakértők határozzák meg.

Az indexkifejezések adják a dokumentumok logikai nézetét.

3.1. Kulcsszó meghatározás -2.

A manuális adatbázis-feltöltést emberek végzik. Ezáltal pontosak, ám ez:

- drága,
 - nehezen fenntartható eljárás,
 - nem minden témát érint,
 - továbbá :
 - nehezen frissíthetők és
 - szubjektívek
- az így előállított adatbázisok.

3.1. Kulcsszó meghatározás -3.

Gyorsabb információ- visszakeresést eredményez az index, amely:

- kiválasztott szavakból és
- kifejezésekből áll.
- Mutatók (pointerek) jelzik, hogy mely dokumentum(ok)ban található meg az indexelt szavak.

A számítógépek lehetővé teszik nagy terjedelmű indexek automatikus létrehozását.

3.1.Kulcsszómeghatározás -4.

Az automatikus indexeléssel az információ-visszakeresési probléma sokkal inkább magához a rendszerhez, mint a felhasználó igényéhez kapcsolódik:

- főként hatékony indexek létrehozásából,
- a felhasználói kérdés (keresőkifejezés) eredményes feldolgozásából
- és a találati lista minőségét javító rangsoroló algoritmusok fejlesztéséből áll.

3.1.Kulcsszómeghatározás -5.

- A számítógépek lehetővé teszik, hogy a dokumentumokat az azokat alkotó összes szóval reprezentáljuk.
- A visszakereső rendszer a dokumentumok teljes szövegű (full text) logikai nézetét (reprezentálásának módszerét) alkalmazza.
- Nagy dokumentumhalmazok esetén csökkenteni kell a reprezentált indexkifejezések számát, pl.: **szövegfeldolgozó** (test operation) műveletekkel.

3.1.Kulcsszómeghatározás -6.

A műveletek lehetővé teszik azt, hogy a teljes

- szöveg helyett
- indexkifejezésekkel

reprezentáljuk a dokumentumokat.

Valójában

- a teljes szövegű reprezentálás a legteljesebb logikai nézete a dokumentumoknak,
- ugyanakkor a leginkább számításigényes a használata.

3.1. Kulcsszó meghatározás -7.

- Az automatizált kulcsszó meghatározást más néven tartalomszűrésnek is nevezik, aminek egyik legfontosabb felfedezése a Zipf-törvény.
- Ennek az elméleti fontosságát az adja, hogy a nyelv teljes megértése felé vezető útnak egyik állomása mindenképpen a nyelvben fellelhető statisztikai szabályszerűségek felismerése kell, hogy legyen.
- A Zipf-törvény gyakorlati fontossága pedig abból adódik, hogy a számítógépes információ-visszakereső rendszerek által használt indexelési módszerek erre a törvényre épülnek.

ZIPF-törvény

- Angol nyelvű szövegekben (korpuszokban) a szavak f előfordulási gyakorisága a **Hatványtörvényt** követi:

$$f(r) = Cr^{-\alpha},$$

- ahol C korpusz függő konstans,
- r a szavak rangsora (az előfordulási szám szerinti csökkenő sorrendben elfoglalt hely).
- α a hatványfüggvény kitevője.
- Az $f(r) = Cr^{-1}$ hatványtörvényt Zipf törvénynek is nevezik. Mivel sok, eddig publikált kísérleti eredmény is igazolja, az a feltetelezés alakult ki, hogy minden nyelvre érvényes.

ZIPF-törvény

Zipf hipotézis: kimondja, hogy egy kulcsszó előfordulásának száma (frekvencia) és a frekvencia által létrehozott rangsor szorzata egy állandó érték körül mozog.

Ezt erősíti meg **Luhn és Hayes**, akik hasonló elmélettel álltak elő a dokumentum összes szavát vizsgálva:

a jellemző szavak egy bizonyos sávban találhatóak meg, amit

- a nagyon sűrűn és

- a nagyon ritkán

előforduló szavak határolnak.

3.2. A szövegfeldolgozás lépései -1.

1. A szavak, azaz a lexikai egységek azonosítása a szövegben (pl. írásjelek elhagyásával). A lexikai egység, azaz a szó karakterek sorozatából áll, amelyeket egymástól bizonyos speciális karakterek (pl. írásjelek, szünetjel) választanak el.
2. Stoplista alkalmazása. Stoplista egy olyan listát jelent ami azokat a szavakat tartalmazza, amelyek általában nem hordoznak jelentést egy dokumentumban.

3.2. A szövegfeldolgozás lépései -2.

- 3. Szótővesítő (stemming) algoritmus alkalmazása. Ez az algoritmus minden szót redukál vagy átranzformál a nyelvi szótőre.
- 4. Kiszámítjuk minden D_j dokumentumra a t_i kifejezés előfordulásainak számát: f_{ij}

- 5. Kiszámítjuk a t_i kifejezés összes előfordulását: t_{fi}

$$t_{fi} = \sum_{j=1}^m f_{ij}$$

- 6. t_{fi} szerint sorba rendezzük a kifejezéseket,
- 7. a nagyon magas értékűeket (ami nagyon gyakran előfordul, már nem mond semmit), és a nagyon alacsony előfordulásúakat (mert azok nem meghatározó jelentőségűek) kirekesztjük.

3.2. A szövegfeldolgozás lépései -3.

8. Az így megmaradó kifejezések az azonosítók vagy index kifejezések.
9. Az indexkifejezések felhasználásával kiszámítjuk minden D_j dokumentumra a w_{ij} súlyokat. Súlyszámok segítségével fejezzük ki, hogy egy kifejezés milyen mértékben tükrözi egy dokumentum tartalmát. A súlyszámok meghatározására számos módszer használható.

3.3. Stoplista -1.

Egy dokumentum szavakból áll, amelyek:

- előfordulhatnak többször is a dokumentumban,
- és vannak olyan szavak is amelyek csupán néhányszor
- vagy csak egyszer fordulnak elő a dokumentumban.

3.3. Stoplista -2.

A dokumentumfeldolgozás abból indul ki, hogy a:

- túlságosan gyakran előforduló szavak (amelyek előfordulási gyakorisága egy bizonyos küszöbérték feletti) többnyire nem hordoznak információt,
- akárcsak a túl ritkán előforduló kifejezések (amelyek előfordulási gyakorisága egy bizonyos küszöbérték alatti), amelyek valószínűleg azért fordulnak elő ritkán, mert a dokumentum szerzője nem tartotta azokat fontosnak.

3.3. Stoplista -3.

- Ezen szavak kezelésére találták ki a stoplistát, amelyek a túlságosan gyakori, illetve ritka kifejezéseket tartalmazza.
- Azokat a szavakat, amelyeket a stoplista tartalmaz kihagyjuk a további vizsgálatkor.
- A stoplista általában terület, illetve applikációfüggő.
- Stoplista elemei többek közt a kötőszavak, névelők.

3.3. Stoplista -4.

- Egy stoplista megalkotása lehet automatizált.
- Újabb stoplisták is generálhatók, pl. egy adott témához tartozó dokumentumokra, amikor a stoplistába olyan kifejezések is bekerülhetnek, amelyek egy általános stoplistának nem az elemei.
- Rendszerint egy általános stoplista a kiindulópont, majd a felhasználó:
 - módosítja,
 - bővítiaz adott témához kapcsolódóan.

3.3. Stoplista -5.

- Az angol nyelvű dokumentumok feldolgozására leggyakrabban használt stoplista a TIME stoplista.
- A magyar nyelvű stoplista néhány eleme:
 - A
 - és
 - Az
 - Van
 - Is
 - Mely
 - Ez
 - Hogy

3.4. Szótövesítés -1.

A keresők a felhasználó által megadott kulcsszavak alapján keresnek az adatbázisban.

- Azonban a kulcsszavak kinyerése a dokumentumból nem elegendő ahhoz, hogy összehasonlítást tudjunk végezni.
- A szavakhoz a mondatban különféle toldalékok csatlakoznak.
- Ezeket valamilyen módszerrel el kell távolítani (két szó egyezésének megállapításában zavaró lehet).

3.4. Szótövesítés -2.

- A toldalékok alakja erősen nyelvspecifikus.
- Porter algoritmus:
 - Kitűnő algoritmus angol nyelvre.
 - Az azonos szótövű angol nyelvű szavakat ugyanarra az alakra hozza.
 - Egyszerű megoldás a problémára (a weboldalak nagy hányada angol nyelvű).
- Több nyelvnek úgyszintén létezik szótövesítő algoritmus, így például a magyarnak is.

3.4. Szótövesítés -3.

Ezen algoritmusok ismeretében már elmondhatjuk:

- hogy a keresett kulcsszó és a
- dokumentumokban levő kulcsszavak között összehasonlítást tudunk végezni.

3.5. Inverz file struktúra -1.

A szövegfeldolgozási lépések:

- lexikai egységek azonosítása,
- stoplista alkalmazása,
- szótövesítés

során elkészített indexkifejezésekből készíti el az indexelő modul az indexet inverz fájl struktúrában.

3.5. Inverz file struktúra -2.

- Legyen O a feldolgozandó objektumok halmaza:

$$O = \{O_1, \dots, O_j, \dots, O_m\},$$

- és jelölje

$$D = \{D_1, \dots, D_j, \dots, D_m\},$$

az objektumoknak megfelelő dokumentumokat.

- Az első három lépés elvégzése után (szavak meghatározása, stoplista alkalmazása, majd szótövesítés) az alábbi indexkifejezéseket kapjuk:

$$T = \{t_1, \dots, t_i, \dots, t_n\}$$

3.5. Inverz file struktúra -3.

- A T indexkifejezéshalmazt használjuk az inverz fájl struktúra megalkotásához az alábbi módon:
 1. A $t_1, \dots, t_i, \dots, t_n$ indexkifejezéseket ábécé sorrendbe rendezzük (többféle gyors rendezési algoritmus ismert).
 2. Az I index tábla megalkotása, amelyben minden r_i sorban a t_i indexkifejezés van D_j dokumentumokra vonatkozó kódjaival, amely D_j dokumentumokban a t_i indexkifejezés előfordul.

3.5. Inverz file struktúra -4.

- Tehát, a t_i indexkifejezések ábécé sorrendben vannak, és mellettük azon dokumentumok kódjai, amelyek tartalmazzák az adott t_i indexkifejezést:

Indexkifejezés	Dokumentum kódja
t_1	$\{D_{11}, \dots, D_{1k}\}$
...	
t_i	$\{D_{i1}, \dots, D_{is}\}$
...	
t_n	$\{D_{n1}, \dots, D_{np}\}$

3.5. Inverz file struktúra -5.

Az inverz fájl struktúra (IF Inverted file Structure):

- egy I index táblából, és a
- hozzá tartozó MF (master file) objektumhalmazból áll.

Az MF az $O = \{O_1, ..., O_j, ..., O_m\}$, objektumokat tartalmazza:

Objektumok
O_1
...
O_j
....
O_m

3.5. Inverz file struktúra -6.

Az I indextáblában lévő kódok tartalmazhatják a megfelelő objektum Mfben elfoglalt címét is.

Az inverz fájl struktúra az alábbi módon alkalmazható:

- Jelölje t a keresőkérdést,
- majd a megfelelő keresőalgorithmus alkalmazása után
- az IF-ben megtaláljuk a t indexkifejezésre vonatkozó sort, azaz azon dokumentumok kódjait, amelyek tartalmazzák a t indexkifejezést.
- Felhasználva a dokumentumok kódjait a megfelelő O objektumok visszakereshetők az MF-ből.

3.3. Inverz file struktúra -7.

Az inverz fájl struktúrában egyéb más adatok is tárolhatók:

- t_i előfordulásainak a száma a D_j dokumentumban,
- t_i összes előfordulásainak a száma az összes dokumentumban
- stb...

4. A lekérdező modul

4.1. A lekérdezés lépései

4.2. A keresőkérdés megadása

4.1. A lekérdezés lépései -1.

A lekérdezési (információ-visszakeresési) folyamat az indexelés után kezdhető meg, és rendszerint a következő lépésekből áll:

1. Adott egy felhasználói információigény.
2. A felhasználó kiválaszt egy információ-visszakereső rendszert.

4.1. A lekérdezés lépései -2.

3. A felhasználó információigénye alapján megfogalmaz egy kérdést természetes nyelven.
4. A természetes nyelvű kérdést az információ-visszakereső rendszer értelmezhető keresőkifejezéssé alakítja.
5. A rendszer elemzi, átalakítja a szövegnél alkalmazott szövegfeldolgozó műveletekkel, majd feldolgozza a keresőkifejezést. Ezáltal a keresőkérdést is indexkifejezések halmazává alakítja.
6. A rendszer a keresőkifejezés alapján végrehajtja a visszakeresést. Majd a visszaadott dokumentumokat átadja a rangsoroló modulnak.

4.2. A keresőkérdés megadása -1.

- A különböző keresők egyes kereséseknél jobbak, mint más keresők.
- Általában elmondható, hogy a pontosság javítható, ha a keresőkérdés több kifejezést tartalmaz.
- A jelenleg elérhető keresők nagy része:
 - kulcsszó alapú keresést használ, és az
 - objektumbázis dokumentumait indexekkel jellemzi.

4.2. A keresőkérdés megadása -2.

- A felhasználó beírja a kulcsszavakat(kérdés), a kereső pedig azonosítja és felsorolja a kérdést kielégítő dokumentumokat.
- A keresők több tekintetben különböznek egymástól:
 - az indextábla mérete,
 - keresési lehetőségek,
 - a válaszok megadásának ideje,
 - a válaszok megjelenítése,
 - a válaszok relevanciája.

4.2. A keresőkérdés megadása -3.

A legmegfelelőbb kereső kiválasztása részben attól függ, hogy alaposan megértsük és ismerjük, hogy kereső a dokumentum mely elemeit indexeli:

- egyes keresők a teljes dokumentumot,
- míg mások csak bizonyos részeit (cím, fejléc).

Ezen különbségek miatt a különböző keresők nagyon eltérő válaszokat adnak vissza ugyanarra a kérdésre.

4.2. A keresőkérdés megadása -4.

A keresési lehetőségek (opciók) sem egyformák.

- Némely esetekben az OR (logikai VAGY) az alapértelmezett opció, és relevanciarangsoroló algoritmus rangsorolja a releváns dokumentumokat,
- máskor pedig több lehetőség közül is választhatunk: AND, OR, NOT, stb.
- Amennyiben a felhasználó nem ismeri a Boole-logikát, előfordulhat, hogy nem arra kap választ, amire szeretne. éppen ezért, a keresések nagy százalékában nem szerepel az ÉS és a VAGY, ami segítené a keresőt leszűkíteni a keresést, ezáltal lehetővé tenné, hogy sokkal pontosabb eredményekhez jusson a felhasználó.

4.2. A keresőkérdés megadása -5.

A keresők hasonlóképpen működnek, mégis más-más eredményeket adnak vissza.

Okai:

- adatbázis-feltöltés különböző módszerei,
- a felhasználó által megadott kulcsszavak értelmezése.

4.2. A keresőkérdés megadása -6.

Például egy egyszerű keresés két kulcsszóval, már bizonyos kérdéseket vet fel:

- A két kulcsszót egymás mellett kell keresni a dokumentumban?
- A két kulcs akármelyikének jelenléte a dokumentumban:
 - elegendő vagy
 - mindkettő jelen kell-e legyen?
- Az első kulcs fontosabb, mint a második?

4.2. A keresőkérdés megadása -7.

Egyes keresők ezt a problémát úgy oldják fel, hogy a

- kérdést más megfogalmazásban teheti fel a felhasználó,
- esetleg a felhasználói felületen ki lehet választani a kulcsok közötti kapcsolatot. (Pl. a szokás szerint a két kulcsszó közé alulhúzás jelet téve, vagy macskakörmök közé helyezve a keresendő szöveget a keresők egy darab kulcsszónak értelmezik és csak olyan válaszokat adnak eredményül, ahol ezek a kulcsszavak egymás mellett találhatóak.)

4.2. A keresőkérdés megadása -8.

- Egyes információ-visszakereső rendszerek lehetővé teszik azt, hogy egyszerű mondatot írjunk be keresőkifejezésnek, és kísérletek folynak összetett mondatok feldolgozására.
- A legtöbb kereső használatakor azonban a keresőkifejezést a rendszer számára feldolgozható formában, kulcsszavakkal és operátorokkal kell megadni.

4.2. A keresőkérdés megadása -9.

- Sok felhasználó nem ismeri az alkalmazható logikai operátorokat , és szövegfeldolgozó műveleteket



- a megadott keresőkifejezés nem tükrözi megfelelően információigényüket.



- rosszul meghatározott keresőkifejezés alacsony visszakeresési hatékonysághoz (retrieval effectiveness) vezet.

4.2. A keresőkérdés megadása -10.

A felhasználó gyakran nincs tisztában:

- a kereső működési elvével,
- és a keresőkérdés megadásának lehetőségeivel,
- ezáltal azzal sem, hogy egy megfelelőképpen megfogalmazott keresőkérdéssel lényegesen javítható a keresés hatékonysága .

Ennek érdekében hasznos lehet egy rövid tájékoztatást adni (vagy egy rámutató linket) a keresőkérdés megadásának lehetőségeiről.

4.2. A keresőkérdés megadása -11.

- A felhasználót meg kell tanítani a megfelelő kérdésfeltevésre. A kérdés feltevésének módja erősen befolyásolja a visszaadott eredmények minőségét.
- Egyes keresők kis és nagybetűre érzékenyek. (Pl., ha nagybetűvel írunk egy szót, akkor azt pontosan olyan formában keresi a kereső, és a kisbetűs változatával nem foglalkozik).
- Ugyanakkor a szó különböző alakjai miatt is a elveszhetnek releváns dokumentumok.

4.2. A keresőkérdés megadása -12.

- A mai keresőket gyakran ellátják ún. haladóknak (advanced) való kereséssel. (szűrési feltételeket lehet megadni).
- A kereső kiválasztásánál a keresési lehetőségek kifinomultsága mellett az indextábla méretét is figyelembe kell venni. Egyszerűbb keresési opciókkal, de nagy számú indexkifejezéssel is sikeres lehet a keresés.

Barcelona hópehely - Google Search - Google Chrome

← → ↻ 🔒 <https://www.google.hu/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=Barcelona+h%C3%B3pehely> ☆ ? ☰

Google Barcelona hópehely 🔍 +Júlia 📱 🔔 Share 🖼️

Web Images Videos News Maps More ▾ Search tools

About 49,100 results (0.46 seconds)

Kiderítették a világ egyetlen fehér gorillájának rejtélyét ...
www.erdekesvilag.hu/kideritettek-a-vilag-egyeten-feh... ▾ Translate this page
Jun 18, 2013 - A Hópehelynek nevezett fehér gorilla évtizedeken át **Barcelona** egyik ...
A BBC szerint feltételezhetően **Hópehely** (Copito de Nieve) volt a világ ...

Meghalt a világ egyetlen fehér gorillája - National Geographic
www.ng.hu/.../Meghalt_a_vilag_egyeten_feher_gorill... ▾ Translate this page
A Hópehelynek nevezett fehér gorilla évtizedeken át **Barcelona** egyik szimbóluma volt.
... A BBC szerint feltételezhetően **Hópehely** (Copito de Nieve) volt a világ ...






Hópehely
users.atw.hu/majomland/hopehely.htm ▾ Translate this page
COPITO DE NIEVE. A korabeli sajtó szerint egy ültetvényes agyonlőtt egy nőstény gorillát, mely a banánját dézsmálta. Áldozatához lépve meglepődve ...




Elpusztult a világ egyetlen albinó gorillája - Origo
www.origo.hu/tudomany/.../20031125hopelyhecske.h... ▾ Translate this page
Nov 25, 2003 - **Hópehely**, a világ egyetlen ismert albinó gorillája, a Barcelónai Állatkert sztárja hétfő reggel rákban pusztult el - közölték az állatkert.

Index - Tudomány - Kiderült, miért volt fehér az albinó gorilla
index.hu/.../2013/.../ezert_volt_feher_az_albino_gorill... ▾ Translate this page
Jun 5, 2013 - A Pompeu Fabra Egyetem, valamint a CSIC kutatóintézet szakemberei

Barcelona AND hópehely - Google Search - Google Chrome

← → ↻ <https://www.google.hu/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=Barcelona%20AND%20h%C3%B3pehely> ☆ ? ☰

Google Barcelona AND hópehely   +Júlia   Share 

Web Images Maps Videos News More ▾ Search tools   

About 21,400 results (0.56 seconds)

Kiderítették a világ egyetlen fehér gorillájának rejtélyét ...
www.erdekesvilag.hu/kideritettek-a-vilag-egyeten-feh... ▾ Translate this page
Jun 18, 2013 - A Hópehelynek nevezett fehér gorilla évtizedeken át Barcelona egyik ...
A BBC szerint feltételezhetően Hópehely (Copito de Nieve) volt a világ ...

Hópehely | Érdekes Világ
www.erdekesvilag.hu/tag/hopehely/ ▾ Translate this page
A hópehely jégkristályok összeállásából keletkezik, amely a Föld légkörében keletkezve ...
... A Hópehelynek nevezett fehér gorilla évtizedeken át Barcelona egyik ...

Meghalt a világ egyetlen fehér gorillája - National Geographic
www.ng.hu/.../Meghalt_a_vilag_egyeten_fehor_gorill... ▾ Translate this page
A Hópehelynek nevezett fehér gorilla évtizedeken át Barcelona egyik szimbóluma volt.
... A BBC szerint feltételezhetően Hópehely (Copito de Nieve) volt a világ ...

Barcelona városnézés - Vacansoleil
www.vacansoleil.hu/kirandulasok-utak/.../barcelona/ ▾ Translate this page
Itt élt Barcelona jelképe, Hópehely. Sajnos a gorilla már elpusztult, de a barcelonai állatkert még mindig ismert főemlőseiről, melyekből rengeteg faj él itt, például ...

Hópehely
users.atw.hu/majomland/hopehely.htm ▾ Translate this page
COPITO DE NIEVE. A korabeli sajtó szerint egy ültetvényes agyonlőtt egy nőstény

Barcelona OR hópehely - Google Search - Google Chrome

← → ↻ https://www.google.hu/search?hl=en&as_q=Barcelona+OR+hópehely&as_epq=&as_oq=&as_eq=&as_nlo=&as_nhi=&lr=lang_hu&cr=&as_qc ? ☰

+Júlia Search Images Maps Play YouTube Gmail Drive More ▾ Júlia Góth 0 Share... ⚙

Google Barcelona OR hópehely 🔍

Web Images Maps Videos News More ▾ Search tools

Any time ▾ All results ▾ Clear

Barcelona focicsapat - Focikatalógus
www.focikatalogus.hu › Focicsapatok ▾ Translate this page
Barcelona focicsapatának mérkőzései, videók összefoglalók, játékosok, góllövők, helyezések, drukkerek, tippjáték.

Barcelona - Booking.com
www.booking.com › Spanyolo. › Katalónia ▾ Translate this page
Hatalmas megtakarítások Barcelona városának szálláshelyein (Spanyolország). Nagyszerű árak és rengeteg szabad szoba. Nézze át korábbi vendégeink ...

Barcelona, Barca ajándéktárgyak, termékek - Football fanshop
www.football-fanshop.hu/barcelona ▾ Translate this page
Barcelona polár takaró 125 x 150 Bullseye · termékek ... Barcelona tábla kicsi 2 db-os öntapadós · termékek ... Barcelona mandzsettagomb stainless steel.

Hópehely | www.burgonyakutatas.hu
www.burgonyakutatas.hu/burgonyafajtak/hopehely/ ▾ Translate this page
Hópehely. Állami minősítés éve: 1997. Származás: Ke 27 x NDK 71.17/6 N+B; Tenyésztő: Középkorai; Gumójellemzők: Kerek-ovál alakú, nagy-középnagy ...

Látnivalók Barcelonában | index - balog.hu
www.balog.hu/barcelona/ ▾ Translate this page

5. A rangsoroló modul

5.1. A rangsoroló modul helye

5.2. A találati lista megadása

5.1. A rangsoroló modul helye -1.

A rangsoroló modul készíti el a felhasználónak visszaadott találati listát, amelyhez az alábbi megelőző lépésekre van szükség:

1. Az indexelő modul által meghatározott indexkifejezésekre a kifejezés-dokumentum mátrix generálása, ami az indexkifejezésekhez tartozó súlyszámokat adja meg.
2. A Q_k Keresőkérdés megadása, majd ebből a v_k keresővektor generálása a kiválasztott súlyszámítási séma alkalmazásával.

5.1. A rangsoroló modul helye -2.

Rangsoroló modul feladata:

3. A lekérdező modul által visszaadott dokumentumokra a hasonlósági értékek kiszámítása.
4. A visszakeresett dokumentumok rangsorolása hasonlósági értékük alapján,
5. A találati lista megjelenítése (relevanciaértékük alapján csökkenő sorrendben)
6. A felhasználó dönt a visszkapott dokumentumok (találati lista) relevanciájáról.

5.2. A találati lista megadása -1.

- A találati lista vizuális megjelenítése is különböző lehet.
- A weben található webkeresők gyakran súlyszámmal látják el az egyes keresések eredményeiként kapott weboldalakokat annak alapján, hogy a felhasználó első ránézésre relevánsnak találta-e az eredményt, vagyis rákattintott-e az eredményre.

5.2. A találati lista megadása -2.

- Ezáltal a visszajelzett válaszok között rangsort tud felállítani. Gyakran a rangsort csak az indexkifejezések előfordulási számával arányosan állítja össze.
- Vannak olyan webkeresők is, ahol pénz kérdése egy-egy weboldal súlysámának megállapítása. Ennek etikussága természetesen megkérdőjelezhető, főleg ha erről nem tájékoztatják kellőképpen a felhasználót, de tény, hogy ilyen létezik.

6. A rendszer kategoricitási tulajdonsága

6.1. Az U bizonytalanság definíciója

6.2. Példa az U bizonytalanság számítására

6.3. A kategoricitás definíciója

6.4. Példa a bizonytalanság és a kategoricitás közti párhuzamra

6.5. Változtatható kategoricitású rendszer

6.1. Az U bizonytalanság definíciója

Shannon-féle információ mennyiség fogalmának felhasználásával, a vektortér (VSM) modellben a találati listához kapcsolható U bizonytalanság :

$$U = - \sum_{j=1}^m p_j \log_2 p_j$$

ahol a p_j valószínűségek
($j=1\dots m$):

$$p_j = \frac{\rho_j}{\sum_{k=1}^m \rho_k}$$

ahol ρ_j a D_j dokumentumnak a Q kérdésre vonatkozó RSV (retrieval status value), relevanciaértéke

Shannon féle információ

- Az információelmélet az elektromos hírtovábbítás kutatása során jött létre 1948 körül.
- C. Shannon.
- Az információ latin eredetű szó, amely értesülést, hírt, üzenetet, tájékoztatást jelent.
- Az információelmélet szerint azonban az üzenet nem azonos az információval.
- Az átvitt adatmennyiség információtartalma attól függ, hogy a vétel helyén mennyire szünteti meg a bizonytalanságot.

6.2. Példa az U bizonytalanság számítására -1.

PÉLDA AZ U BIZONYTALANSÁG SZÁMÍTÁSÁRA:

Legyenek

- D_1, D_2 és D_3 , $m = 3$, dokumentumok,
- t_1 és t_2 , $n = 2$, kifejezések

A kifejezés-dokumentum mátrix legyen

$$W = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 3 & 2 \end{bmatrix}$$

6.2. Példa az U bizonytalanság számítására -2.

Jelölje Q a kérdést $(0, 1)$

Ha a visszakereső-függvény ρ a dot product,

- akkor a ρ_1 , ρ_2 és ρ_3 értékek a következők:

$(0, 3, 2)$

- A megfelelő valószínűségek:

$$\begin{bmatrix} 0 \\ 0.6 \\ 0.4 \end{bmatrix}$$

6.2. Példa az U bizonytalanság számítására -3.

Az U bizonytalanság: $U = -\sum_{j=1}^m p_j \log_2 p_j$
0.971

- a ρ_1 , ρ_2 és ρ_3 értékek:
(0, 3, 2)

a valószínűségek:

$$\begin{bmatrix} 0 \\ 0.6 \\ 0.4 \end{bmatrix}$$

Vektortér modell súly, hasonlóság

- Gyakoriság szerinti súlyozás: $w_{ij} = f_{ij}$

- Maxnormált: $w_{ij} = \frac{f_{ij}}{\max_{1 \leq k \leq n} f_{kj}}$

f_{ij} : t_i kifejezés előfordulásainak száma a D_j dokumentumban

- Koszinusz mérték (Cosine measure): c_{jk}

- $$s_{jk} = c_{jk} = \frac{\sum_{i=1}^n w_{ij} w_{ik}}{\sqrt{\sum_{i=1}^n w_{ij}^2 \sum_{i=1}^n w_{ik}^2}}$$

6.2. Példa az U bizonytalanság számítására -4.

- Ha a visszakereső függvény a Koszinusz mérték, és a súlyok max-normáltak,

$$W = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 3 & 2 \end{bmatrix}$$

- ρ_1, ρ_2 és ρ_3 : a valószínűségek: bizonytalanság:

$$\begin{bmatrix} 0 \\ 0.923 \\ 0.847 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0.521 \\ 0.479 \end{bmatrix}$$

0.997

6.2. Példa az U bizonytalanság számítására -6.

- Dot product, gyakoriság súly:

ρ_1, ρ_2 és ρ_3 : valószínűségek: bizonytalanság: **0.971**

(0, 3, 2)

$$\begin{bmatrix} 0 \\ 0.6 \\ 0.4 \end{bmatrix}$$

- Koszinusz mérték, max-normált súly:

ρ_1, ρ_2 és ρ_3 : a valószínűségek: bizonytalanság:

$$\begin{bmatrix} 0 \\ 0.923 \\ 0.847 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0.521 \\ 0.479 \end{bmatrix}$$

0.997

6.2. Példa az U bizonytalanság számítására -5.

Tehát a **második** esetben **nagyobb a bizonytalanság** arra nézve, hogy melyik válasz hasznosabb a felhasználónak



ugyanis a relevanciértékek közti különbség itt **kisebb**.

6.3. A kategoricitás definíciója -1.

Jelölje

- "A" és "B" két különböző és pozitív RSV-alapú (pozitív relevanciaértékeket adó) információ-visszakereső módszert.
- U_A , és U_B jelölje a találati listákhoz kapcsolódó bizonytalanságokat.

Ekkor az "A" módszeren alapuló információ-visszakereső rendszer sokkal kategorikusabb a válaszaiban, mint a "B",

hogy ha igaz: $U_A < U_B$

6.3. A kategoricitás definíciója -2.

Egy információ-visszakereső rendszer kategorikus a válaszaiban, hogy ha csökkenti a bizonytalanságot,



pontosan megállapítható, hogy melyiket is válasszuk ki a találatok közül.

Más szavakkal a kategoricitás azt jelenti, hogy a válaszok relevanciaértékükben mennyire szórnak.

6.4. Példa a bizonytalanság és a kategoricitás közti párhuzamra -1.

Adott

- Egy keresőkérdés
- két különböző modell: F_1 és F_2 ,
- válasz dokumentumhalmaz azonos, de a hozzájuk tartozó relevanciaértékek a különbözőek:
- **F_1 modell esetén:** az eseményválaszok (E_j) és a hozzájuk tartozó relevanciaértékek (ρ_j) a következők :
 $(E_1; \rho_1 = 0.5), (E_2; \rho_2 = 0.5), (E_3; \rho_3 = 0.5).$

Tehát a felhasználói kérdésre a válasz: E_1, E_2, E_3 .

Mivel az eseményválaszok azonos relevanciaértékekkel rendelkeznek, tehát egyformán hasznosak a felhasználói profilra nézve.

6.4. Példa a bizonytalanság és a kategoricitás közti párhuzamra -2.

- **F_2 modell esetén:** a válaszesemények (E_j) és a hozzájuk tartozó relevanciaértékek (ρ_j) a következők:
 $(E_1; \rho_1 = 0.7)$, $(E_2; \rho_2 = 0.6)$, $(E_3; \rho_3 = 0.5)$.

Tehát a felhasználói kérdésre a válaszok: E_1, E_2, E_3 .

Az eseményválaszok különböző relevanciaértékekkel rendelkeznek, tehát nem egyformán hasznosak a felhasználóra nézve.

6.4. Példa a bizonytalanság és a kategoricitás közti párhuzamra -3.

- Az első esetben a **válaszesemények kevésbé kategorikusak**, mivel **ugyanazokkal a relevanciaértékekkel** rendelkeznek, tehát a felhasználó sokkal bizonytalanabb abban, hogy a válaszokat milyen sorrendben is nézze meg.
- A második esetben, mivel a válaszesemények **különböző relevanciaértékekkel** rendelkeznek, ezért a felhasználó egyértelműen el tudja dönteni, hogy milyen sorrendben is nézze meg a válaszeseményeket, tehát a bizonytalansága sokkal kisebb.

6.4. Példa a bizonytalanság és a kategoricitás közti párhuzamra -4.

- F_1 modell esetén:

$$(E_1; \rho_1 = 0.5), (E_2; \rho_2 = 0.5), (E_3; \rho_3 = 0.5).$$

- F_2 modell esetén:

$$(E_1; \rho_1 = 0.7), (E_2; \rho_2 = 0.6), (E_3; \rho_3 = 0.5)$$

Az első esetben:

válaszesemények kevésbé kategorikusak,
nagyobb a bizonytalanság