



Az információ-visszakereső módszerek relevanciahatékonyságának mérése.

Szabványos tesztadatbázisok a mérésekben.

7. ELŐADÁS

1. Az információ-visszakereső módszerek relevanciahatékonyságának mérése.
2. Szabványos tesztadatbázisok a mérésekben.

1. Az információ-visszakereső módszerek relevancia- hatékonyságának mérése

- 1.1. A relevanciahatékonyság mérése
- 1.2. A Cranfield paradigma
- 1.3. Standard tesztkollekciók.
- 1.4. Mérőszámok a hatékonyság mérésére
- 1.5. A teljesség-pontosság grafikon
- 1.6. A teljesség-pontosság diagram.

1.1. A relevanciahatékonyság mérése -1.

- Információigény  keresőkérdés
(információ-visszakeresést megvalósító program által definiált formában)
- A visszakeresés: egy adott kérdésre keressük a választ egy adott objektumhalmazban

1.1. A relevanciahatékonyság mérése -2.

Az információ-visszakeresés (IR) formális definíciója:

Def. 1: Legyen IR az alábbi:

$$IR = (U; IN; Q; O) \rightarrow R;$$

ahol

- U = felhasználó (user),
- IN = információigény (information need),
- Q = keresőkérdés (query),
- O = keresendő objektumok halmaza,
- R = a Q keresőkérdésre válaszként visszaadott dokumentumok halmaza.

1.1. A relevanciahatékonyság mérése -3.

- Az információ-visszakereső rendszereknél azt vizsgáljuk:
 - milyen precíz a találati lista
(mennyire relevánsak a válaszok).
- Ez a vizsgálati módszer:
 - visszakeresési hatékonyság kiértékelése**
(retrieval performance evaluation, retrieval effectiveness evaluation)

1.1. A relevanciahatékonyság mérése -4.

- A relevanciát bináris skálán mérjük:
egy dokumentum
 - vagy releváns,
 - vagy irrelevánsaz információigény tekintetében.
- A visszaadott dokumentumok nem lehetnek pontos válaszok (a felhasználók információigényét nem tükrözik teljes mértékben a megadott kereső kifejezések)
- A válaszokat rangsorolni kell aszerint, hogy mennyire relevánsak a kérdésre.

1.1. A relevanciahatékonyság mérése -5.

A visszakeresési hatékonyság becslésénél figyelembe kell venni azt, hogy hogyan hajtjuk végre a visszakeresési folyamatot.

A visszakeresés állhat:

- egyszerűen egy kérdés **köteget** feldolgozásából (a felhasználó megad egy kereső kifejezést és megkapja a válaszokat),
- egy egész **interaktív** folyamatból (a felhasználó interaktív lépéseken keresztül adja meg információigényét a rendszernek),
- vagy lehet e két stratégia kombinációja is.

1.1. A relevanciahatékonyság mérése -6.

A köteget és az interaktív lekérdezés két teljesen különböző folyamat, ezért a kiértékelésük is különböző.

Interaktív lekérdezés esetében:

- a felhasználó fáradozása,
- a felhasználói interfész tulajdonságai,
- a rendszer által biztosított használati útmutató,
- a lekérdezés időtartama

mind fontos értékelési szempontok.

1.1. A relevanciahatékonyság mérése -7.

Kötegelt lekérdezés esetén ezek közül a szempontok közül egyik sem olyan fontos, mint a rendszer által generált találati lista minősége.

- Az információ-visszakereső módszereket a kérdések kötegelt feldolgozásával teszteljük.
- A kiértékelést a Cranfield paradigma alapján:
 - standard tesztkollekciókon,
 - standard mérőszámok (effectiveness measures) alkalmazásával végezzük el.

1.2.A Cranfield paradigma -1.

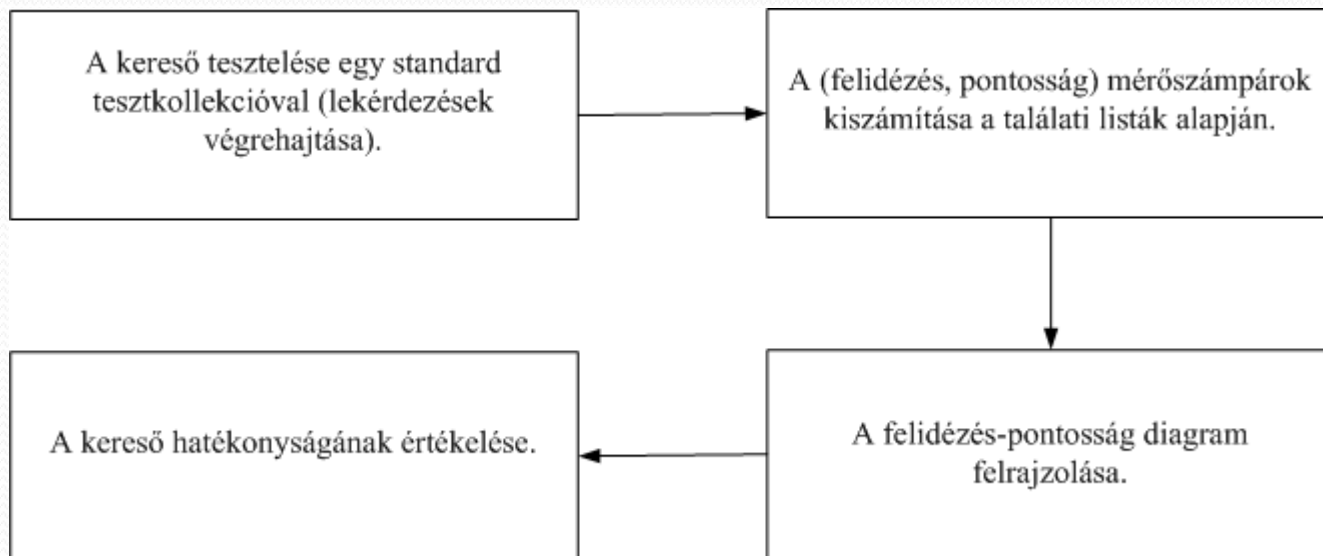
A Cranfield paradigma alapján a tesztelés a következőt jelenti:

Adottak:

- A vizsgált kereső.
- Egy standard tesztkollekció.
- Mérőszámok: teljesség (recall), pontosság (precision).
- A mérőszámokból meghatározható teljesség-pontosság diagram (Recall-precision graph).

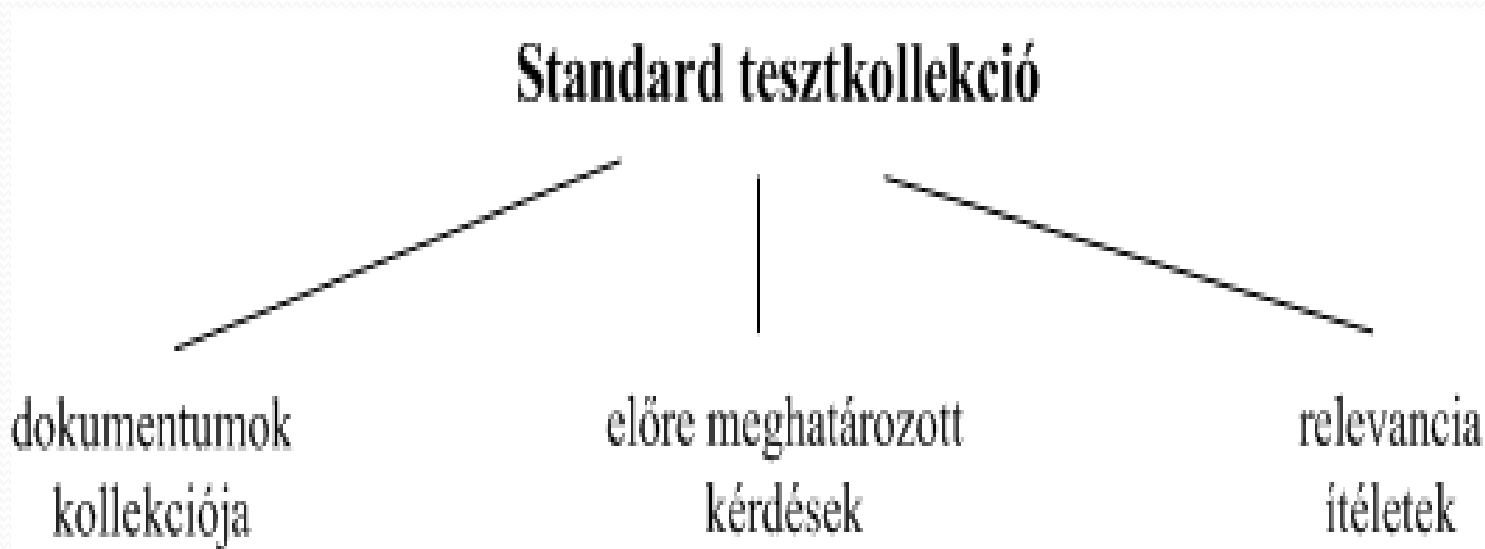
1.2.A Cranfield paradigma -2.

A tesztelés folyamata a Cranfield paradigma alapján:



1.3. Standard tesztkollekciók -1.

A standard tesztkollekció elemei:



1.3. Standard tesztkollekciók -2.

- A tesztkollekció mindhárom részét külön-külön szakértők határozzák meg manuálisan. Az egyes komponenseket előre, a vizsgált keresőtől függetlenül állítják össze.
- A tesztkollekciók lehetővé teszik:
 - az információ-visszakereső rendszerek információ-visszakeresési hatékonyságának gyors kiértékelését,
 - valamint az eredmények tükrében a különböző rendszerek közvetlen összehasonlítását.

1.4. Mérőszámok a hatékonyság mérésére -1.

Az információ-visszakereső rendszer által megadott válaszok relevanciáját leggyakrabban két standard mérőszámmal:

- a **teljességgel, felidézés (recall)**és
- a **pontossággal (precision)** jellemezzük.

1.4. Mérőszámok a hatékonyság mérésére -2.

- A standard mérőszámok minden egyes kérdés esetében számszerűen meghatározzák a hasonlóságot a keresési stratégia által visszaadott és a szakértők által meghatározott dokumentumok között.
- Ezek a mérőszámok lehetővé teszik a visszakeresési stratégia jóságának megbecslését.

1.4. Mérőszámok a hatékonyság mérésére -3.

A kiértékeléshez természetesen használhatók még további mérőszámok is pl:

- $\text{Selejt, Kiesés}(\text{Fallout}) = \frac{\text{visszakeresett_irreleváns_dokumentumok}}{\text{összes_irreleváns_dokumentum}}$
- Harmónikus közép:
$$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}}$$

A Cranfield paradigmában azonban csak a teljességet és a pontosságot mérjük.

1.4. Mérőszámok a hatékonyság mérésére -4.

A teljesség a megtalált releváns és az összes releváns dokumentum hányada (adott kérdés esetén).

$$\text{Felidézés, (Recall)Teljesség} = \frac{|Ra|}{|R|}$$

- R = az adott kérdésre (szakértők által meghatározott) releváns dokumentumok halmaza,
- $|R|$ = az R dokumentum halmaz számossága,
- Ra = az adott kérdésre a visszakeresési módszer által megtalált releváns dokumentumok halmaza,
- $|Ra|$ = az Ra halmaz számossága.

1.4. Mérőszámok a hatékonyság mérésére -5.

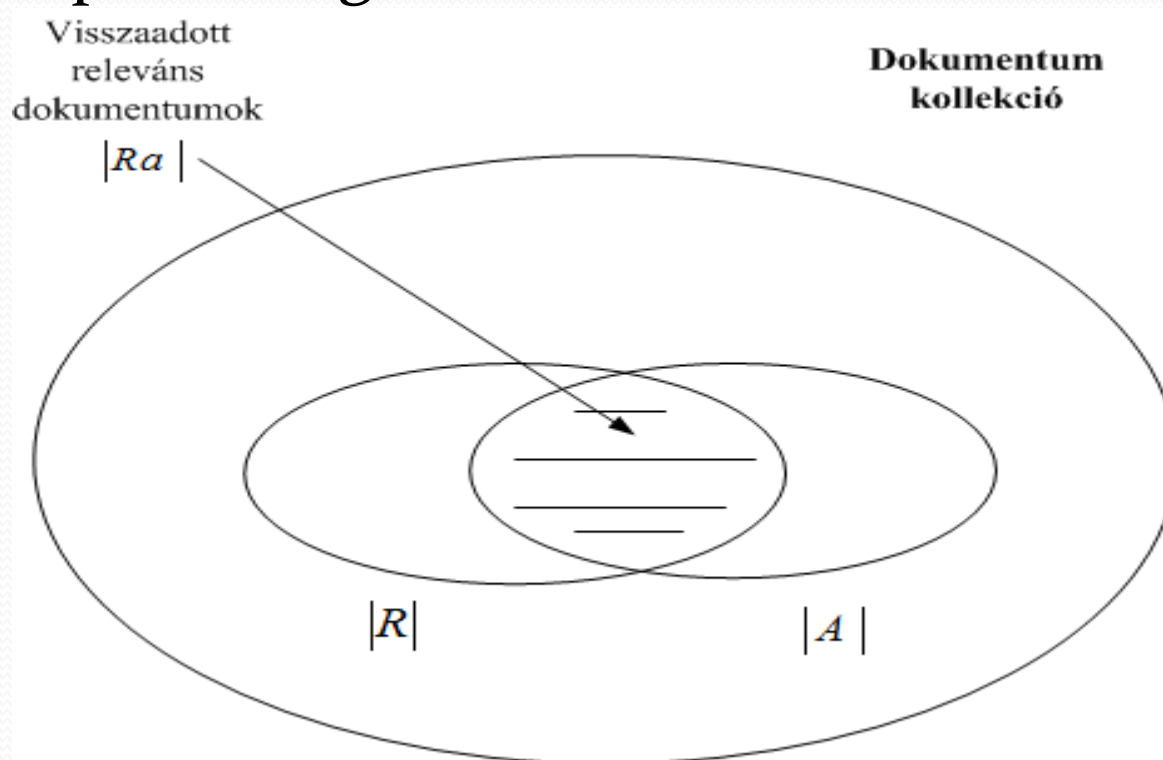
A pontosság a megtalált releváns és az összes visszakapott dokumentum hányada (adott kérdés esetén).

$$\text{Pontosság (Precision)} = \frac{|Ra|}{|A|}$$

- A = a visszakeresési módszer által visszaadott összes dokumentum halmaza,
- $|A|$ = az A dokumentum halmaz számossága,
- Ra = az adott kérdésre a visszakeresési módszer által megtalált releváns dokumentumok halmaza,
- $|Ra|$ = az Ra halmaz számossága.

1.4. Mérőszámok a hatékonyság mérésére -6.

A visszakeresési módszer egy adott kérdésre vonatkozó teljessége és pontossága:



1.4. Mérőszámok a hatékonyság mérésére -7.

- A teljesség és a pontosság elméleti definíciója azt feltételezi, hogy a visszakeresési módszer által visszaadott összes dokumentumot (A halmaz elemeit) megvizsgáltuk.
- A kialakult szokás szerint a tesztelésnél a következő számítási módszerrel állapítjuk meg a teljességet és a pontosságot (teljesség-pontossági diagram felrajzolásával):

1.5. Teljesség-pontosság grafikon

Tekintsünk:

- egy standard tesztkollekciót és
- a hozzá tartozó előre meghatározott kérdések halmazát.

Vizsgáljunk ezek közül egy kérdést, amelyhez meghatároztuk a megfelelő q kereső kifejezést.

Jelölje Rq a q kereső kifejezésre releváns (szakértők által meghatározott) dokumentumok halmazát.

1.5. Teljesség-pontosság grafikon

- A felidézés (teljesség) és a pontosság klasszikus mérőszámok az IR rendszerek hatékonyságának kiértékelésére.
- A (teljesség, pontosság) rendezett számpárokat grafikusán ábrázoljuk az ún. Teljesség – pontosság grafikonon.
- A teljesség-pontosság grafikon 11 standard teljességi szinten alapul: 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 100%. A 0%-os teljességi szinthez tartozó pontosság érték interpolációval határozható meg, megadása azonban nem feltétlenül szükséges.

PÉLDA-10 releváns dokumentum

Tegyük fel, hogy Rq a következő dokumentumokból áll:

- $Rq = \{d_3, d_5, d_9, d_{25}, d_{39}, d_{44}, d_{56}, d_{71}, d_{89}, d_{123}\}$

Vagyis a q keresőkifejezésre tíz (szakértők által meghatározott) releváns dokumentum van.

Tekintsünk egy visszakereső algoritmust. Tegyük fel, hogy az algoritmus a q kereső kifejezésre a következő rangsorolt találati listát adja vissza.

PÉLDA-10 releváns dokumentum

A találati lista:

(A q kereső kifejezésre releváns dokumentumokat pont jelöli)

1. d ₁₂₃ ·	2. d ₈₄	3. d ₅₆ ·
4. d ₆	5. d ₈	6. d ₉ ·
7. d ₅₁₁	8. d ₁₂₉	9. d ₁₈₇
10. d ₂₅ ·	11. d ₃₈	12. d ₄₈
13. d ₂₅₀	14. d ₁₁₃	15. d ₃ ·

PÉLDA-10 releváns dokumentum

Rq a következő dokumentumokból áll:

$$Rq = \{d3, d5, d9, d25, d39, d44, d56, d71, d89, d123\}$$

A találati lista:

(A q kereső kifejezésre releváns dokumentumokat pont jelöli)

1. d123 .	2. d84	3. d56.
4. d6	5. d8	6. d9.
7. d511	8. d129	9. d187
10. d 25.	11. d38	12. d48
13. d250	14. d113	15. d3 .

PÉLDA-10 releváns dokumentum

A találati lista jellemzői:

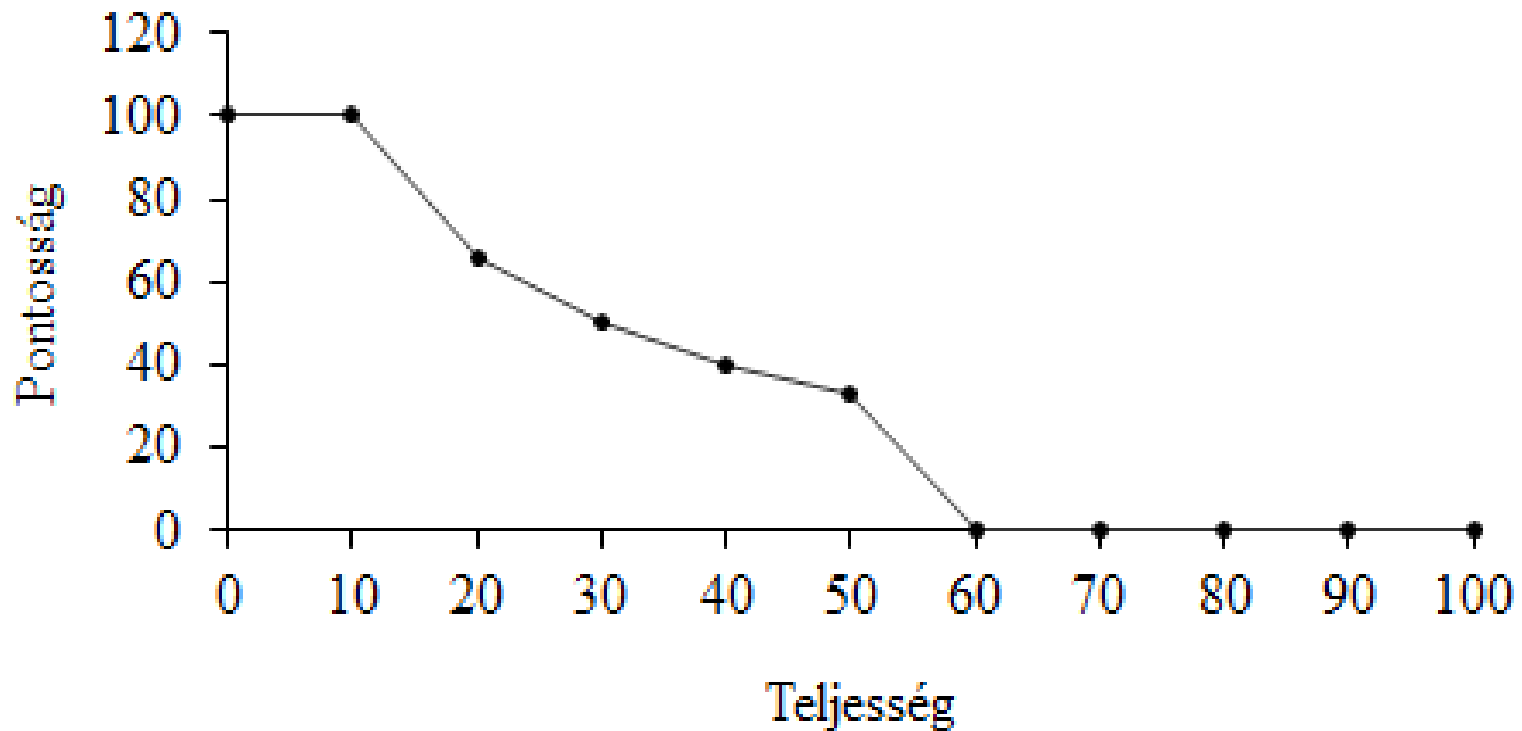
- Az **első releváns** dokumentum a **találati lista első** eleme (d_{123}). Ez a dokumentum az R_q összes releváns dokumentum halmazának 10%-át jelenti.
 - 10%-os felidézési szint esetén a pontosság 100%.
- A **második releváns** dokumentum a **találati lista harmadik** (d_{56}) dokumentuma. Ebben a pontban a pontosság megközelítőleg 66% (három dokumentumból kettő releváns) 20%-os felidézési szintnél (tízből két releváns dokumentumot vizsgáltunk meg).
 - 20 %-os felidézési (teljességi) szint esetén a pontosság 66%.

PÉLDA-10 releváns dokumentum

- d_{123} (találati lista 1. eleme) releváns,
• $\pi=1$ (pontosság), $\rho = 0.1$
- d_{56} (találati lista 3. eleme) is releváns
• $\pi=2/3 = 0.66$ (pontosság), $\rho = 2/10 = 0.2$
- d_9 (találati lista 6. eleme) releváns
• $\pi=3/6 = 0.5$ (pontosság), $\rho = 3/10 = 0.3$
- d_{25} (találati lista 10. eleme) releváns
• $\pi=4/10 = 0.4$ (pontosság), $\rho = 4/10 = 0.4$
- d_3 (találati lista 15. eleme) releváns
• $\pi= 5/15=0.33$ (pontosság), $\rho = 5/10 = 0.5$

PÉLDA-10 releváns dokumentum

A példához tartozó teljesség-pontosság grafikon:



PÉLDA-10 releváns dokumentum

- A visszakereső algoritmus a releváns dokumentumoknak csak az 50%-át találta meg, ezért a pontosság értéke nullára csökken az 50%-ot meghaladó teljességi szintek esetén.
- A fenti módszerrel meghatározhatók a standard tesztkollekció összes, előre meghatározott kérdéséhez tartozó különböző teljesség-pontosság grafikonok.

1.5. Teljesség-pontosság grafikon

- Általában a legtöbb kérdés esetén a felidézési szintek nem egyeznek meg a 11 standard szinttel, mert nem minden esetben van pontosan tíz releváns dokumentum az adott kérdésre.
- Interpolációs módszerrel határozzuk meg a standard felidézési szintekhez tartozó pontossági értékek:

$$P(r_j) = \max_{r_j \leq r \leq r_{j+1}} P(r)$$

ahol: $r_j, j \in \{1, 2, \dots, 10\}$ a j -edik standard felidézési szintet jelöli. (Pl. r_5 az 50%-os felidézési szintnek felel meg.)

1.5. Teljesség-pontosság grafikon

Az interpolációval meghatározott pontosság értéke a j -edik felidézési szintnél megegyezik a j -edik és a $j+1$ -edik felidézési szint közé eső legnagyobb ismert pontossággal.

- Pl. ha egy kérdésre három releváns válasz van, akkor a pontosságot csak a következő három felidézési szintnél tudjuk meghatározni: 33,3%, 66,6%, 100%.

Ekkor a standard felidézési szinteknél az interpolációval meghatározott pontosságok:

- 0%, 10%, 20%, 30%-nál a 33,3%-os felidézési szinthez tartozó pontosság,
- 40%, 50%, 60%-nál a 66,6%-os felidézési szinthez tartozó pontosság,
- 70%, 80%, 90%, 100%-nál a 100%-os felidézési szinthez tartozó pontosság.

Példa (3 rel. Dok.)

- Tegyük fel, hogy a releváns dokumentumok Rq száma: 3
- következő dokumentumokból áll:
- $Rq = \{ d9, d56, d123, \}$

1. d123 ·	2. d84	3. d56·
4. d6	5. d8	6. d9·
7. d511	8. d129	9. d187
10. d25	11. d38	12. d48
13. d250	14. d113	15. d3

Példa (3 rel. Dok.)

- d_{123} (találati lista 1. eleme) releváns,
• $\pi=1$ (pontosság), $\rho = 1/3 = 0.333$
- d_{56} (találati lista 3. eleme) is releváns
• $\pi=2/3 = 0.66$ (pontosság), $\rho = 2/3 = 0.666$
- d_9 (találati lista 6. eleme) releváns
• $\pi=3/6 = 0.5$ (pontosság), $\rho = 3/3 = 1$

Példa (3 rel. Dok.)

- d_{123} (találati lista 1. eleme) releváns,
- $\pi=1$ (pontosság), $\rho = 1/3$ azaz **33%** (0,10,20,30%)
- d_{56} (találati lista 3. eleme) is releváns
- $\pi=2/3 = 0.66$, $\rho = 2/3$ azaz **66%** (40,50,60%)
- d_9 (találati lista 6. eleme) releváns
- $\pi=3/6 = 0.5$, $\rho = 3/3$ azaz **100%** (70,80,90,100%)
- **Piros: standard teljességi szint**
- **Kék: valós teljességi szint**

Példa (16 rel. Dok.)

Tegyük fel, hogy a releváns dokumentumok Rq száma: 16:
következő dokumentumokból áll:

$Rq = \{d3, d5, d9, d25, d39, d44, d56, d71, d89, d123, d200, d201, d202, d203, d204, d205\}$

A találati lista:

1. d123 .	2. d84	3. d56 .
4. d6	5. d8	6. d9 .
7. d511	8. d129	9. d187
10. d 25 .	11. d38	12. d48
13. d250	14. d113	15. d3 .

Példa (16 rel. Dok.)

- *Ha egy kérdésre 16 releváns válasz van:*
 - d_{123} (találati lista 1. eleme) releváns,
 $\pi=1$ (pontosság), $\rho = 1/16 = 0.0625$ 6.25% (0%)
 - d_{56} (találati lista 3. eleme) is releváns
 $\pi=2/3 = 0.66$ (pontosság), $\rho = 2/16 = 0.125$, 12.5% (10%)
 - d_9 (találati lista 6. eleme) releváns
 $\pi=3/6 = 0.5$ (pontosság), $\rho = 3/16 = 0.1875$, 18.75%
 - d_{25} (találati lista 10. eleme) releváns
 $\pi=4/10 = 0.4$ (pontosság), $\rho = 4/16 = 0.25$, 25% (20%)
 - d_3 (találati lista 15. eleme) releváns
 $\pi= 5/15 = 0.33$ (pontosság), $\rho = 5/16 = 0.3125$,
31.25%(30%)

1.5. Teljesség-pontosság diagram

- Az algoritmusok visszakeresési hatékonyságát az összes kérdés figyelembevételével szokás kiértékelni,



- átlagolni kell a különböző kérdésekhez tartozó grafikonokat minden egyes felidézési szint esetén.

1.6. Teljesség-pontosság diagram

- Az átlagos pontossági értékeket a következő képlettel határozzuk meg:

$$\bar{P}(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q}$$

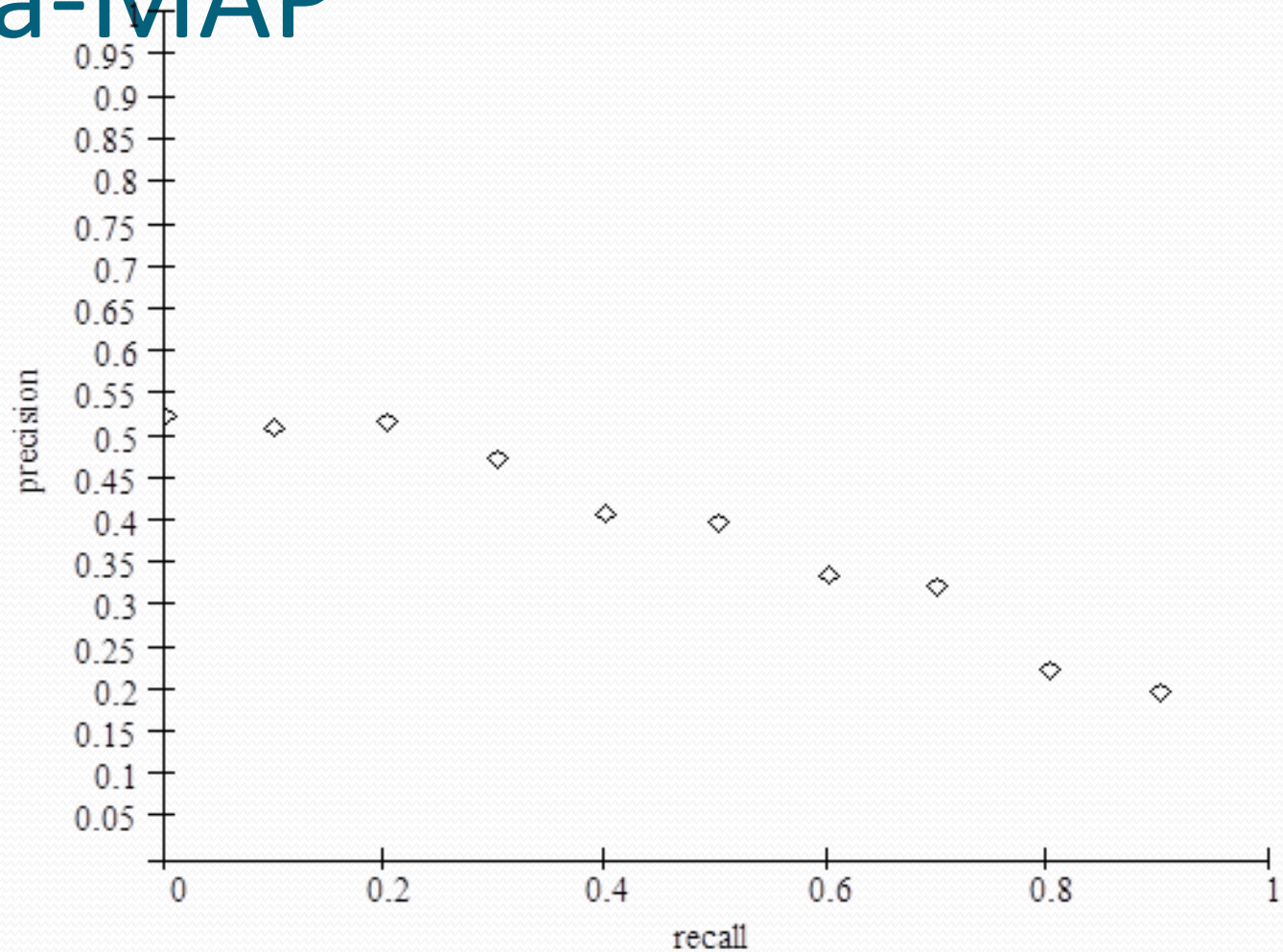
- $\bar{P}(r)$ = átlagos pontosság az r felidézési szintnél,
- N_q = a felhasznált kérdések száma,
- $P_i(r)$ = az i -edik kérdés pontossága az r felidézési szintnél.

MAP mérték

- Megadható a MAP (Mean Average Precision):
 - „átlagok átlaga”
 - Az r felidézési szinteken mért $\bar{P}(r)$ átlagos pontossági értékek átlaga

Példa-MAP

- MAP



Példa-MAP

• Az átlagos pontossági értékek az adott teljességi szinteken:

$$\bar{P}(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q}$$

ρ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
π	0.52	0.53	0.5	0.43	0.4	0.35	0.32	0.22	0.2

MAP (Mean Average Precision), azaz (átlagos pontossági értékek átlaga):

0.385

1.5. Teljesség-pontosság diagram

A teljességi-pontosság diagram:

- Megadása: a különböző kérdésekhez tartozó teljesség-pontosság grafikonok átlagolásával
- a visszakeresési algoritmusok hatékonyságának összehasonlítására használható.
- Az újonnan kifejlesztett algoritmusok hatékonyságát leggyakrabban a klasszikus vektor tér modell (SMART) visszakeresési hatékonyságával szokták összehasonlítani.

A keresés lehetséges kimenetei a találati lista és a relevancia szempontjából:

- TP: találati listában szereplő releváns dokumentumok száma
- FP: találati listában szereplő irreleváns dokumentumok
- FN: a találati listában nem szereplő releváns dokumentumok
- TN: a találati listában nem szereplő irreleváns dokumentumok

Selejt (fallout): $FN/(FN+TN)$

Teljesség: $TP/(TP+FN)$

Pontosság: $TP/(TP+FP)$

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

q_3 :

$$Rq_3 = \{3, 43, 45, 60\}$$

Vq_3 :

- | | | | |
|--------------|-------------|--------------|-----------|
| 1. 21 (15) | 3. 3 (11) * | 5. 60 (11) * | 7. 40 (6) |
| 2. 43 (15) * | 4. 10 (11) | 6. 45 (8) * | 8. 72 (6) |

Pontosság a lehetséges felidézési szinteknél:

Felidezés	25	50	75	100
Pontosság	50	66,6	60	66,6

0	10	20	30	40	50	60	70	80	90	100
10	10	10	15.4	15.4	15.4	0	0	0	0	0

q_{11} :

$$Rq_{11} = \{8, 43, 79, 81\}$$

Vq_{11} :

- | | | | |
|-----------|--------------|--------------|------------|
| 1. 62 (8) | 6. 6 (4) | 11. 55 (4) | 16. 36 (4) |
| 2. 30 (6) | 7. 9 (4) | 12. 47 (4) | 17. 19 (4) |
| 3. 39 (6) | 8. 12 (4) | 13. 79 (4) * | 18. 21 (4) |
| 4. 69 (6) | 9. 17 (4) | 14. 78 (4) | 19. 22 (4) |
| 5. 1 (4) | 10. 43 (4) * | 15. 38 (4) | 20. 26 (4) |

Pontosság a lehetséges felidézési szinteknél

Felidezés	25	50	75	100
Pontosság	10	15,4	0	0

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

q_{19} :

$$Rq_{19} = \{1, 2, 16, 39, 66\}$$

Vq_{19} :

- | | | |
|--------------|-------------|------------|
| 1. 66 (10) * | 3. 16 (6) * | 5. 34 (6) |
| 2. 70 (7) | 4. 22 (6) | 6. 2 (5) * |

Pontosság a lehetséges felidézési szinteknél

Felidezés	20	40	60	80	100
Pontosság	100	66,6	50	0	0

0	10	20	30	40	50	60	70	80	90	100
66.6	45	0	0	0	0	0	0	0	0	0

q_{27} :

$Rq_{27} = \{1, 2, 6, 8, 9, 11, 14, 15, 16, 18, 20, 22, 23, 26, 30, 32, 33, 34, 36, 37, 39, 40, 41, 48, 55, 57, 61, 62, 66, 68, 69, 70, 79\}$

Vq_{27} :

- | | | | |
|-------------|-----------|-------------|--------------|
| 1. 52 (11) | 4. 43 (5) | 7. 11 (4) * | 10. 70 (4) * |
| 2. 1 (10) * | 5. 3 (5) | 8. 17 (4) | 11. 20 (4)* |
| 3. 2 (5) * | 6. 65 (5) | 9. 50 (4) | |

Pontosság a lehetséges felidézési szinteknél:

Felidézés	3	6	9	12	15	18	21	24	27	30	33
Pontosság	50	66,6	43	40	45	0	0	0	0	0	0

Felidézés	36	39	42	45	48	51	54	57	60	63	66
Pontosság	0	0	0	0	0	0	0	0	0	0	0

Felidézés	70	73	76	78	81	85	88	91	94	97	100
Pontosság	0	0	0	0	0	0	0	0	0	0	0

Standard tesztkollekciók



	ADI	MEDLINE	TIME
Tárgykör	Információ-tudomány	Orvos-tudomány	újságcikkek
Dokumentumok száma	82	1033	423
Keresőkérdeések száma	30	30	83
Indexkifejezések száma	791	7744	13479

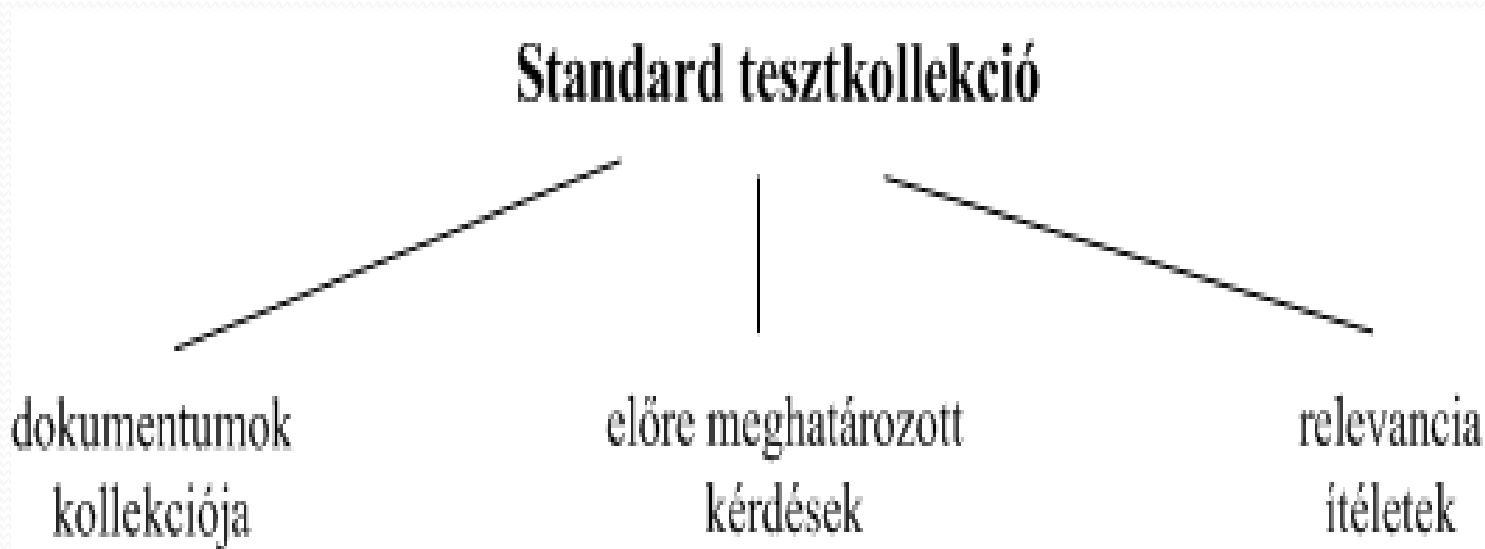
2. Szabványos tesztadatbázisok a mérésekben

2.1. Szabványos tesztkollekciók

2.2. ADI tesztkollekció felépítése

2.1. Szabványos tesztkollekciók -1.

- A standard tesztkollekció elemei:



2.1. Szabványos tesztkollekciók -2.

- A tesztkollekció mindhárom részét külön-külön szakértők határozzák meg manuálisan. Az egyes komponenseket előre, a vizsgált keresőtől függetlenül állítják össze.
- A tesztkollekciók lehetővé teszik:
 - az információ-visszakereső rendszerek információ-visszakeresési hatékonyságának gyors kiértékelését,
 - valamint az eredmények tükrében a különböző rendszerek közvetlen összehasonlítását.

2.1. Szabványos tesztkollekciók -3.

Legismertebb tesztkollekciók: TREC, ADI, MEDLINE, CACM, CISI, TIME, REUTERS,

- Néhány ezek közül szabadon hozzáférhető (http://ir.dcs.gla.ac.uk/resources/test_collections/)
- Ezeket használják az információ-visszakeresés területén történő kutatásokban.
- Eléggé különböznek a gyűjtemény méretét, illetve a kérdések számát tekintve.

2.2. ADI tesztkollekció -1.

ADI tesztkollekció a legkisebb
(http://ir.dcs.gla.ac.uk/resources/test_collections/adi/)
(it contains 82 homogeneous English articles from
computing journals with 2086 index terms and 35 queries).

A tesztkollekció az alábbi file-okat tartalmazza:

- adi.all : dokumentumok (documents),
- adi.que : (kérdések) queries,
- adi.rel : (relevancia ítéletek) relevance assessments,
- adi.blm : (boole kérdések listája) list of Boolean queries.

2.2. ADI teszkollekció -2.

Dokumentumok (adi.all) szöveges részt és további strukturált mezőt tartalmazna. A dokumentumok általában az alábbi mezőket tartalmazzák:

- .I : sorszám(serial number of the document)
- .T: cím (title of the document)
- .A: szerző (author/authors of the document)
- .W: szöveg (text of the document).

Ha valamelyik mező (pl. szerző) ismeretlen, akkor az a mező hiányzik a dokumentumból.

2.2. ADI teszkollekció -3.

Az I6 sorszámú dokumentum:

.I 6

.T

a new centralized information-retrieval system for the petroleum industry including a computer search system and two manual indexes

.A

E. H. BRENNER

B. H. WEIL

N. E. RAWSON

.W

an integrated system was developed cooperatively to include a current awareness manual index, a dual dictionary, and a search tape; all three indexes are produced from a master computer tape. updating, training, and advice will be provided companies for searching the abstracts and further indexing and merging of company internal information.

2.2. ADI tesztkollekció -4.

- Az ADI tesztkollekció 35 kérdést tartalmaz (adi.que).
- A kérdés struktúrája megegyezik a dokumentumével;
- Tehát tartalmaz sorszám ("serial number" (I)), és szöveg ("text" (W))mezőket.
- Kérdések típusai:
 - .I3 : a mixed query: kérdés és állítás
 - .I4: is a simple query: csak állítás
 - .I14: is a simple query: csak kérdés

2.2. ADI tesztkollekció -5.

.I 3

.W

What is information science? Give definitions where possible.

.I 4

.W

Image recognition and any other methods of automatically transforming printed text into computer-ready form.

.I 14

.W

What future is there for automatic medical diagnosis?

2.2. ADI tesztkollekció -6.

Az információ-visszakereső rendszer nem tudja feldolgozni a természetes nyelvű kérdést,



a kérdést a megfelelő formára kell hozni (pl. Boole-formára a Boole-féle információ-visszakereső rendszer számára)

2.2. ADI tesztkollekció -7.

Boole formában a kérdés:

- What is information science? Give definitions where possible.
 - #q3= #and ('information', #or ('science', 'definition'));
- #q4= #or (#and ('image', 'recognition'), #and (#or ('printed', 'text'), #or ('methods', 'automatically', 'transforming', 'computer-ready'))));
- What future is there for automatic medical diagnosis?
 - #q14= #and ('medical', #or ('future', 'automatic')) ;

2.2. ADI tesztkollekció -8.

- Az ADI tesztkollekció tartalmazza a relevanciaítéleteket (adi.rel) is az alábbi formában: (.I3, .I43, .I45, .I60 dokumentum releváns az .I3 kérdésre).

3	3
3	43
3	45
3	60
4	29
4	63
14	20
14	33

Medline adatbázis

- MEDLINE tesztkollekció orvosi folyóiratcikkeket tartalmaz
(http://ir.dcs.gla.ac.uk/resources/test_collections/med1/)
- (it contains 1033 homogeneous English articles from medical journals and 30 queries).
- A tesztkollekció az alábbi file-okat tartalmazza:
- med.all : dokumentumok (documents),
- med.que : (kérdések) queries,
- med.rel : (relevancia ítéletek) relevance assessments,

3. Web keresőmotorok

- 2.1. Web keresők fajtái
- 2.2. Web keresők tulajdonságai
- 2.3. Kulcsszavak
- 2.4. Rangsor
- 2.5. Keresési módszerek
- 2.6. Web keresők moduljai
- 2.7. Web keresők története
- 2.8. Keresők népszerűségi listája
- 2.9. Példa webkeresőkre

3. Web keresőmotorok -1.

Az Internetre az emberek úgy tekintenek ma, mint egy könyvtárra,



ám a könyvtárban se lehet eligazodni könyvtáros nélkül.

3. Web keresőmotorok -2.

A webkeresők szolgáltatják a könyvtáros funkcióját az Interneten.

A felhasználó oldaláról a probléma a következőképpen fogalmazódik meg:

- tudja mire kíváncsi,
- és meg szeretné keresni a hálózat káoszában a számára megfelelő dokumentumot, amely választ ad kérdésére,
- kizárva a nem releváns vagy téves információkat.

3. Web keresőmotorok -3.

- Az internet hálós szerkezetéből adódóan egy weblap olvasása közben könnyedén rátalálhatunk más minket érdeklő információkra.
- A weben található lapok ugyanis hiperhivatkozásokat tartalmaznak, melyek más webes tartalmakra mutatnak.
- Így konkrét cél nélkül bolyongva is rengeteg információhoz juthatunk használatával.

BÖNGÉSZÉS-BOLYONGÁS-BROWSING

3. Web keresőmotorok -4.

Ám ez

- az internet méretei és a
 - mai emberek idő hiánya miatt
- egyáltalán nem hatékony módszer.
- Ha nem csak egyszerűen böngészünk a világhálón, hanem egy konkrét információ igényünket szeretnénk hatékonyan kielégíteni webes keresőket használunk.

KERESÉS-SEARCHING

3. Web keresőmotorok -5.

- A webkeresők webszervereken keresztül elérhető szoftverek, melyek segítségével a kívánt kulcsszavak alapján megtalálhatjuk a megfelelő dokumentumokat.
- Egy keresőrendszer esetében alapvetően két fő komponensről beszélhetünk:
 - a kereső motor (search engine): ez, a felhasználó elől rejtett modul végzi magát a keresést.
 - a felhasználói felület (user interface): az amin keresztül a felhasználók kereshetnek az adatbázisban.

3. Web keresőmotorok -6.

A ma elterjedt keresők működésének lényege:

1. a végiglátogatott oldalak dokumentumait egy adatbázisban valamilyen szempont szerint indexelik.
2. Ebből az indextáblából keresnek a felhasználó által megadott kulcsszó / kérdés alapján.

Ám az hogy maga ez az adatbázis, hogy épül fel, keresőnként változik.

3.1. Web keresők fajtái -1.

A web keresők működését több alapelve is helyezték már.

- Egyes web keresők arra törekszenek, hogy minél több weboldal tartalmát értelmezzék egy webet pásztázó program segítségével és a dokumentumot kielemezve annak jellemzőit tárolják egy központi helyen.

Ezt az eljárást keresőrobotok más szóval spiders, crawlerek végzik.

3.1. Webkeresők fajtái -2.

A keresőrobot egy olyan program:

1. mely a webet pásztázza, releváns dokumentumok után kutatva
2. ezeket letölti,
3. rangsorolja,
4. értékeli,
5. és ez alapján tárolja el adatbázisában.

Ez a módszer sokkal aktuálisabb képet tud adni a web tartalmáról, mert a robotok folyamatosan végzik munkájukat.

Példa: Google, Bing

3.1. Web keresők fajtái -3.

Léteznek olyan keresők, amelyeknél:

1. a weboldal szerkesztői regisztrálják a weboldalukat egy bizonyos hierarchikus struktúrába sorolva,
2. és ezen adatok alapján csoportosítva eltárolják a dokumentum jellemzőit.

3.1. Web keresők fajtái -4.

Katalógus rendszerek (megpróbálják kategorizálni a weben tárolt anyagokat):

- az adatbázist manuálisan töltik fel és kézileg rangsorolják.
- Előnyük hogy megbízhatóbbak, pontosabbak a tekintetben hogy az adott weboldal tényleg releváns e az adott kulcsszóra.
- Ám a web méretei és áttekinthetetlensége miatt a manuális katalogizálás bonyolult, hosszadalmas folyamat.
- Ebben az esetben lehetőség van arra is, hogy az oldal készítője maga regisztrálja a kereső adatbázisban saját weblapját.
- Példa: Yahoo!
- Magyar példa: Startlap

3.2. Web keresők tulajdonságai -1.

A web keresők közös tulajdonsága:

- egy lokális adatbázisból való lekérés során visszaadott hivatkozásokat adják vissza a felhasználónak.
- az adatbázisok indexelve vannak a web kereső alapelvétől függően különböző szempontok alapján.

Az indexelés:

- általában kulcsszavak alapján történik,
- de másodlagosan további szempontok alapján (pl. URL, utolsó frissítés, kategória) is rendezhetik az adatokat.
- Ily módon pillanatok alatt kiolvasható a felhasználó igényéhez igazodó dokumentum fellelési helye.

3.2. Web keresők tulajdonságai -2.

- Az, hogy egy kereső kellően nagyméretű indexelt adatbázissal rendelkezik még nem elegendő a hatékony kereséshez.
- Az eredményességet az is befolyásolja, hogy a felhasználó által feltett kérdést hogyan értelmezik.
- A különböző nyelvek más - más nyelvtani sajátosságai miatt nyelvenként eltérő ez a módszer.

3.3. Rangsor -1.

- Az is fontos hogy milyen módszerrel rendezi sorba a kereső a találatokat, hogy határozza meg az oldalak fontosságát.
- Statisztikák bizonyítják, hogy a felhasználók többsége csak az első 11 találatot tekinti meg, a többit már figyelmen kívül hagyja.

3.3. Rangsor -2.

- A keresők gyakran súlyszámmal látják el az egyes keresések eredményeiként kapott weboldalakat annak alapján, hogy a felhasználó:
 - első ránézésre relevánsnak találta-e az eredményt, vagyis
 - rákattintott-e az eredményre. Ezáltal a visszajelzett válaszok között rangsort tud felállítani.
- Gyakran a rangsort csak a kulcs előfordulási számával arányosan állítja össze.
- Vannak olyan web keresők, ahol pénz kérdése egy-egy weboldal súlyszámának megállapítása. Ennek etikussága természetesen megkérdőjelezhető, főleg ha erről nem tájékoztatják kellőképpen a felhasználót.

3.3. Rangsor -3.

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

- az adatbázis-feltöltés különböző módszereinek,
- és annak is, ahogy a felhasználó által megadott kulcsszavakat értelmezik.

3.4. Keresési módszerek -1.

A felhasználók különböző keresési módszereket alkalmazhatnak:

- az egyszerű keresés, és
- az összetett keresés.
- Egy átlagos felhasználó alapvetően az egyszerű keresést alkalmazza. Ennek során visszakapjuk az keresett kulcsszót tartalmazó összes dokumentum listáját, esetleg ezeket kulcs előfordulási gyakoriság alapján rangsorolva.
- Az összetett keresés már feltételez valamiféle informatikai tudást, hiszen itt boole kifejezések (AND, OR, NOT) segítségével tehetjük eredményesebbé a keresésünket.

3.4. Keresési módszerek -2.

- A kereső rendszerek bizonyos mértékig abban is eltérést mutatnak, ahogy megjelenítik a találataikat. Tehát, hogy milyen módszer alapján történik a rangsorolás, vagy van e lehetőségünk kategóriákba rendezni a találatokat (pl. képek keresése).

3.4. Keresési módszerek -3.

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?

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.

Például 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 weboldalakat adnak eredményül, ahol ezek a kulcsszavak egymás mellett találhatóak.

3.4. Keresési módszerek -4.

- Előfordulhat az is, hogy ha több szót adunk meg, akkor nem számít, hogy mindkét szó benne legyen a dokumentumban, hanem inkább az számít, hogy a két szó minél többször forduljon elő.

Vagyis ha az első szó 10-szer előfordul, viszont a második egyszer sem, akkor annak a weboldalnak előnye van ahhoz képest, ahol az első 4-szer és a második 5-ször fordul elő.

3.4. Keresési módszerek -5.

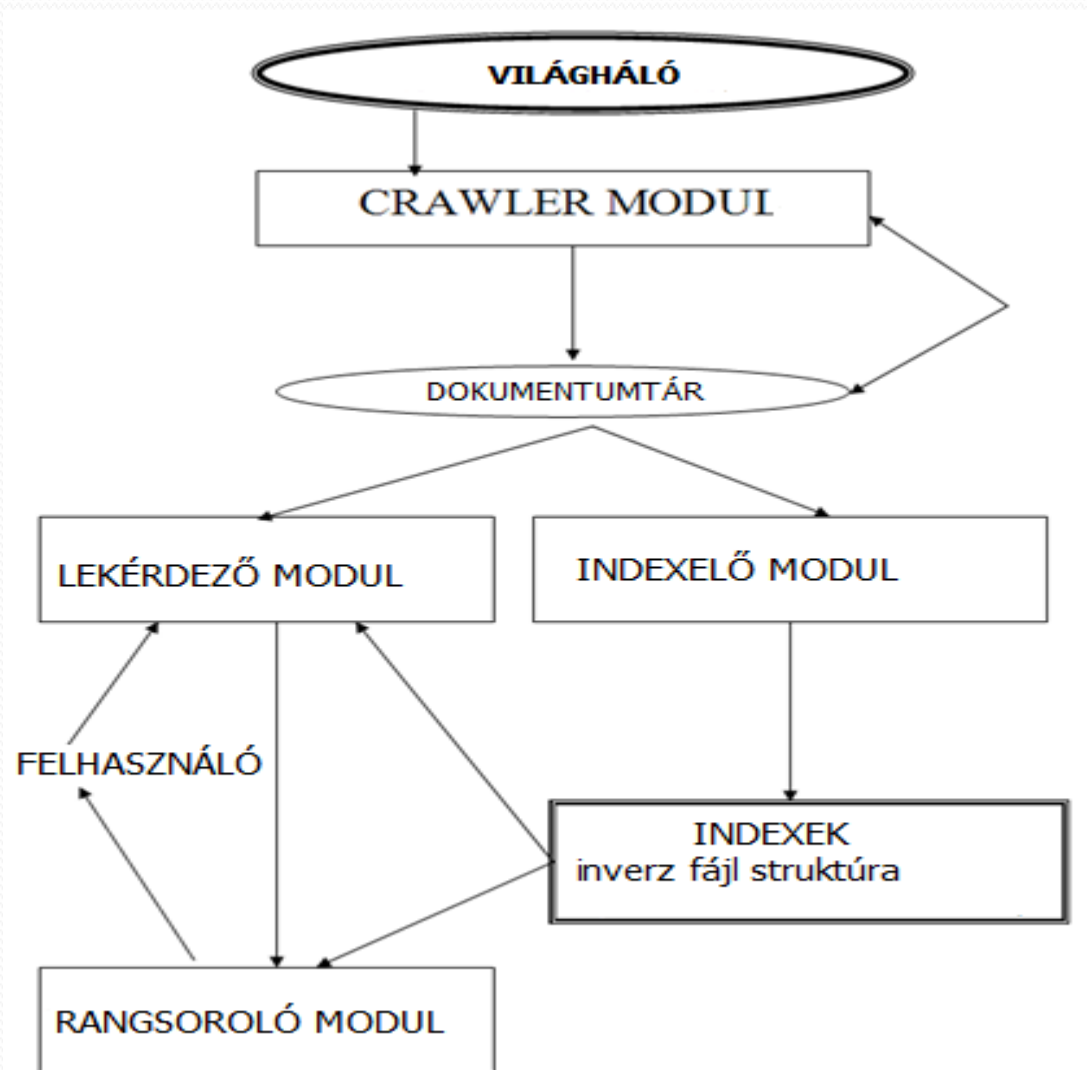
- A Google kereső figyelembe veszi ezt is és különbséget tesz a kulcsszavak között. A Google súlyszámítási és rangsorolási eljárása (ranking) az ún. legtöbbször idézett (most cited) vagy szavazási elven (voting principle) alapszik (PageRank eljárás): minél több weboldal hivatkozik valamely weboldalra, annál nagyobb súlyszámot kap ez a weboldal.

Mindkettőnek megvan az előnye is és a hátránya, a kettő közötti választás a felhasználó igényének a függvénye.

3.4. Keresési módszerek -6.

- A weben rengeteg keresőt találhatunk. Sok web szerver nyújt olyan jellegű szolgáltatást, hogy a saját maga által az Internetre kihelyezett weboldalai között lehet keresést eszközölni.

3.5. Web kereső moduljai



3.5. Web kereső moduljai -2.

- A hagyományos információ-visszakereső rendszerben a dokumentumokat egy központi tárban tároljuk, számítógép diszkeken, általában egy adott intézménynél (egyetemi könyvtár, bank számítóközpontja).
- A weboldalakat decentralizáltan, a világ számos számítógépén tároljuk. Ennek egyik előnye, hogy nincs földrajzi korlátozás a dokumentumok között. Ugyanakkor azt is jelenti, hogy a webes keresőknek össze kell gyűjteni a dokumentumokat a világ minden tájáról.

3.5. Web kereső moduljai:

CRAWLER MODUL: (robot, spider)

- Virtuális robotok, úgynevezett pókok pásztázzák a webet lapról lapra járva és a meglátogatott oldalakat letöltik a dokumentumtárba. („böngészi” a webet, ahhoz hasonlóan, ahogyan az emberek követik a linkeket, hogy átjussanak új oldalakra.)
- Adott az URL-eknek egy kezdő halmaza, amelyeket meglátogat. Ezekről az oldalakról kigyűjti a linkek URL-jeit, és átadja őket a *crawler control modul*nak.
- A crawler folyamatosan böngészi a webet (éjjel-nappal futnak), amíg a helyi erőforrások ki nem merülnek (pl. betelik a tárház).
- **Crawler control modul:** eldönti, hogy mely URL-eket keresse fel legközelebb a crawler, és visszaadja ezt a listát a crawlernek. Számon tartja, hogy mely linkek újak, melyeket látogatott már meg. Lehetnek speciális keresők, ilyenkor a *crawler control modul* ügyel arra, hogy mely linkeken folytatódjon a munka. (Pl. a SZTAKI kereső csak a .hu-t tölti le.)

3.5. Web kereső moduljai: crawler

A Crawler :

1. az URL-ek egy kezdeti S halmazából indítja a keresést
2. Ebből a crawler kivesszi az első URL-t,
3. letölti az oldalt,
4. kiszedi belőle a linkek URL-jeit és
5. beteszi őket a sor végére.
6. Ez a folyamat addig ismétlődik, amíg a crawler úgy nem dönt, hogy leáll. Sok kérdés vetődik fel ezzel kapcsolatban:

3.5. Web kereső moduljai: crawler

- **Mely oldalakat töltse le a crawler?** A legtöbb esetben a crawler nem tudja az összes oldalt letölteni a webről. A legnagyobb webkeresők is csak néhány százalékát tudják indexálni a teljes webnek. Ezért fontos, hogy figyelmesen osztályozzuk a weboldalakat és a jó oldalakat keressük fel először. Ennek eléréséhez az URL-eket elsőbbségi sorba tesszük, és a jó URL-eknek nagyobb súlyt adunk.

3.5.1. Fontossági mértékek

- **Népszerűség:** Az oldalak fontossága függ a népszerűségüktől.
 - Egy mód a népszerűség mérésére, ha megszámoljuk az oldalra mutató linkek számát. Tehát a P oldal népszerűségi mértéke megegyezik azon linkek számával, amelyek P -re mutatnak.

3.5.1. Fontossági mértékek

- **Cím:** Egy oldal fontosságát jól mutatja a címe is.
 - Például ha az URL .com-ra végződik, akkor azt hasznosabbnak tekintjük, mint az egyéb végződésű URL-eket,
 - vagy ha az URL-ben szerepel a „home” szó, azt is fontosabbnak tekintjük.
 - Más mérték szerint a kevés „/”-t tartalmazó URL-ek a fontosabbak, mivel ezek a könyvtárstruktúra tetején helyezkednek el. Ezek könnyen számíthatóak.

3.5. Web kereső moduljai: crawler

- **Hogyan frissítsük a weboldalakat?** Ha már sok oldalt letöltöttünk, el kell kezdenünk újra meglátogatni őket, hogy észleljük a változásokat és frissítsük az adatbázist. Mivel a weboldalak nagyon különböző mértékben változnak, ezért a crawlernek körültekintően kell döntenie arról, hogy mely oldalakat látogassuk meg újra, melyeket csak később. Ennek a döntésnek jelentős hatása van az adatbázis frissességére.

3.5. Web kereső moduljai: crawler

- **Hogyan tudjuk az erőforrás felhasználást egy website-on minimalizálni?** Amikor a crawler begyűjti az oldalakat, akkor más szervezetek erőforrásait is használja. Például, amikor a crawler a p oldalt tölti le az S siteről, akkor a sitenak ki kell keresnie a p -t a saját filerendszerében, ezáltal használva a lemezt és a processzort. A fílet a megkeresése után át kell küldeni a hálózaton, ami újabb szervezetek erőforrásait használja. A crawlernek minimalizálnia kell a hatását ezekre az erőforrásokra. Különben a website fenntartói panaszkodni fognak és néha teljesen megtiltják a crawlerek hozzáférését.

3.5. Web kereső moduljai: crawler

- **Hogyan tudjuk párhuzamosítani a letöltést?** A web hatalmas méretei miatt a crawlerek gyakran több gépen futnak és párhuzamosan töltik le a weboldalakokat. Ez a párhuzamosítás szükséges is, ha sok oldalt akarunk letölteni ésszerű idő alatt. Világos, hogy ezeket a párhuzamosan működő crawlereket megfelelően koordinálni kell, hogy ugyanazt a website-ot ne keressék fel sokszor.

3.5. Web kereső moduljai: crawler

Ha a crawler már kiválasztotta és letöltötte a fontos oldalakat, akkor ezeket frissítenie is kell, hogy a dokumentumtár ne legyen elavult. Ennek többféle módja lehet. A két természetesen adódó eljárás:

- **Egységes frissítés:** Minden oldalt ugyanolyan gyakorisággal töltünk le újra és újra.
- **Arányos frissítés:** Minél gyakrabban változik egy oldal, annál gyakrabban frissítjük. Ha egy elem tízszer gyakrabban változik meg, mint egy másik, akkor azt tízszer gyakrabban frissítjük, mint a másikat.

A crawlernek tudnia kell, hogy melyik oldal milyen gyakran változik. Ez megbecsülhető a korábbi változásaiból, vagy használható a „last modified date”.

3.5. Web kereső moduljai: crawler

Robot Exclusion –t (a robotokra vonatkozó előírások) a crawlernek be kell tartania:

- a robotoknak minden esetben meg kell vizsgálnia, hogy engedélyezett-e az adott oldal letöltése. Ehhez meg kell néznie a robots.txt fület és a weboldal meta-tag-ját. Ezeket a file-okat a weboldalak készítőinek és a site fenntartóinak kell megírniuk, hogy szabályozzák a robotok működését.
- A robots.txt file tartalmazza, hogy a site mely oldalait nem töltheti le a webkereső.
- Másik fontos korlátozás, hogy egy site-ra csak 60 másodperc elteltével szabad újabb kérést küldenie a crawlernek, hogy ne terhelje nagyon az adott site hálózatát.

3.5. Web kereső moduljai: crawler

- A meta-tagek a html oldalak fejlécében helyezkednek el, a következő alakban:
- `<META NAME="ROBOTS" CONTENT="robots-terms">` A következő utasítások állhatnak a robotok számára:
 - ALL (bármit lehet),
 - NONE(mindent tilos),
 - INDEX(az oldal bekerülhet a keresőbe),
 - NOINDEX(az oldal nem kerülhet a keresőbe),
 - FOLLOW(követhetőek a linkek az oldalról),
 - NOFOLLOW(nem követhetőek a linkek az oldalról).

3.5. Web kereső moduljai - dokumentumtár.

DOKUMENTUMTÁR (page repository):

A pókok által letöltött dokumentumokat a dokumentumtárban tároljuk (fizikailag számítógép diszkeken, a számítógépek a keresőt üzemeltető cég tulajdonában vannak). A dokumentumtárban található oldalak az indexelő modulhoz kerülnek további feldolgozásra. A fontos vagy népszerű oldalak hosszabb ideig tárolódnak.

3.5. Web kereső moduljai - indexelő.

INDEXELŐ MODUL:

- A dokumentumtárban található weblapokat dolgozza fel (HTML címkék szűrése, index kifejezések azonosítása, stb.). Kiszedi a szavakat az oldalakról, és mindegyik szóhoz rögzíti, hogy az mely URL-ben fordult elő. Ennek eredménye egy hatalmas táblázat, amelyet „text index”-nek nevezünk. Ez a feladat különösen nehéz, a web hatalmas mérete és gyors változása miatt

3.5. Web kereső moduljai -indexelő

- **Text Index:** Minden szóhoz karbantartja az adott szót tartalmazó URL-ek listáját.
- **Structure Index:** A webet, mint irányított gráfot tekintjük. A csúcsoknak megfelelnek a weboldalak, az irányított élek pedig a hiperhivatkozások. Ebben az indexben a gráf szerkezetét tartjuk nyilván, azaz hogy melyik oldalról mely oldalakra mutat link.
- **Utility index:** Többféle rangsort tartalmazhat, például fontossági sorrendet, képek száma szerinti sorrendet, hosszúság szerinti sorrendet stb. Ezen listák elkészítéséhez felhasználja a text indexet és a structure indexet is a következő elem:

3.5. Web kereső moduljai -indexelő

- Azaz a weblapok egy tömörített verzióját hozza létre oly módon, hogy meghatározzuk és kivonatoljuk a fontos információkat. Az indexek logikailag inverz fájl struktúrába vannak szervezve (fizikailag tömörítve vannak tárolva memória megtakarítási célból). Jellemzően számos részstruktúrára vannak felosztva
- A pók a link struktúrát használja még felfedezetlen oldalak megtalálására.

3.5. Web kereső moduljai - lekérdező

LEKÉRDEZŐ MODUL:

- beolvassa,
- analizálja és
- a megfelelő formátumba (pl. numerikus kód) transzformálja a felhasználó kérdését.

A lekérdező modul az indexek segítségével meghatározza, mely dokumentumok illeszkednek a felhasználó kérdésére (pl. azok az oldalak, amelyek tartalmazzák a kérdés kifejezést). Az illeszkedő oldalak a rangsoroló modulhoz kerülnek.

3.5. Web kereső moduljai - rangsoroló.

RANGSOROLÓ MODUL:

Végzi a lekérdező modul által kiválasztott oldalak sorba rendezését hasonlósági értékük szerint csökkenő sorrendben. Az így kapott listát találati listának hívjuk. A felhasználó a találati listában a weboldalak URL-jét kapja vissza. Ezután a felhasználó az URL linkre kattintva megtekintheti a teljes oldalt.

3.5. Web kereső moduljai - rangsoroló

A hasonlósági érték kiszámítása számos kritérium figyelembevételével, különböző módszerekkel történik.:

- A számítás egyrészt a hagyományos információ-visszakereső módszereken alapul,
- másrészt figyelembe veszi a webre jellemző speciális tényezőket. A jellemző tényezők a következők:
 - oldal tartalmának jellemzői (kifejezések előfordulási gyakorisága),
 - oldalon belüli jellemzők (a kifejezés elhelyezkedése az oldalon belül, kifejezés betűmérete),
 - link információ (mely oldalak mutatnak az adott oldalra, az adott oldal mely oldalakra mutat).

3.6. Webkeresők története

1. Az első keresőmotor a (ma már nem működő) Wandex volt, amit Matthew Gray fejlesztett ki a MIT-en 1993-ban. Indexet hozott létre amit Wandex-nek hívtak. Keresőrobotját a web akkori méretének meghatározására használták 1995-ig.
2. Egy másik 1993-ban készült kereső, az Aliweb (*Archie Like Indexing for the Web*) még ma is elérhető. Ez a keresőgép nem használt robotot az oldalak begyűjtésére, hanem kizárólag a webhely adminisztrátorok visszajelzésére épült azzal kapcsolatban, hogy létezik-e valamilyen indexállomány az adott webhelyen.

3.6. Webkeresők története

3. 1993-ban kezdte el működését a *JumpStation*, amely már keresőrobotot alkalmazott az oldalak meglátogatására és az indexének létrehozására. Ez volt az első olyan kereső, amely egy keresőgép mindhárom alapvető jellemzőjét tartalmazta (a begyűjtést, az indexelést és a keresést). A szűkös erőforrások miatt a JumpStation indexelése és keresése a begyűjtött weboldalak címére korlátozódott.
4. 1994: *WebCrawler* : az első olyan kereső, amelynek a keresőrobotja már a begyűjtött weboldalak teljes szövegét vette figyelembe. Elődjeitől eltérően megengedte használóinak, hogy a weboldalakon lévő bármelyik szóra keressenek, ami ettől fogva alapvető elvárásként fogalmazódott meg a keresőknél.
5. 1994-ben indult a Lycos, ami az egyik első példa volt arra, hogy egy kereső komoly üzleti vállalkozássá váljon.

3.6. Webkeresők története

6. A következő néhány évben sok nagy keresőmotor szállt be a versenybe. (a [Hotbot](#), az [Excite](#), az [Infoseek](#), az [Inktomi](#), az [Open Text](#), a [Northern Light](#) és az [AltaVista](#)).
7. 1996-ban a Stanford Egyetemen Larry Page és Sergey Brin megalkotta a [Google](#)-t, ami a hivatkozási népszerűséget vizsgáló PageRank algoritmuson alapuló, a korabeli keresőkhöz képest nagyon fejlett sorrendezésének (és kisebb részben a puritán felhasználói felületnek) köszönhetően gyorsan kiszorította a vetélytársait, és
8. 2001 körül egyeduralmódóvá vált a Google (elemezni tudja a különböző oldalak közötti kapcsolatokat, és ennek segítségével sokkal relevánsabb találatokat tud visszaadni, mint más keresők).
9. 2002-ben a [Yahoo!](#) megvásárolta az Inktomit, majd 2003-ban az [AlltheWebet](#) és az [AltaVistát](#), és 2004-ben megindította saját keresőszolgáltatását.

3.6. Keresők története -MS

- 1998 őszén a *Microsoft* elindította az *MSN* keresőjét, amely az *Inktomi*-ból származó keresési találatokra támaszkodott.
- 1999 elején az *MSN* a *Looksmart* és az *Inktomi* keresési eredményeit jelenítette meg a találati listájában.
- 2004-ben a *Microsoft* elkezdett áttérni a saját keresőtechnológiájára, amelyet a keresőrobotjával, az ún. *msnbot*-tal támogatott.
- 2009 júniusában a *Microsoft* egy új, *Bing* nevű keresővel jelent meg a piacon.
- 2009 júliusában a *Yahoo!* és a *Microsoft* olyan egyezséget kötött egymással, amely szerint a *Yahoo!* is a *Microsoft* *Bing* keresőtechnológiájára épül.

3.6. Webkeresők története

Néhány újabb (de egyelőre még nem túl népszerű) megközelítés a keresési technológiában:

- a tárgyspecifikus hivatkozási népszerűség (ezt használja a [Teoma](#) keresőmotor, amire az Ask épül): pl. egy weboldal fontossága nagyobb ha ugyanolyan témájú weboldalak mutatnak rá.
- és a metakeresők (magyar fejlesztésű [Polymeta](#)):
 - Tetszőleges számú **adatforrás egyidejű keresése** (web keresők, híroldalak, rejtett web források: könyvtári katalógusok, publikációs oldalak stb.). **Egyetlen egyszerű keresőfelület**, amely áttekinthető eredménylistát ad, akár különböző típusú találatokkal is (szöveg, kép, videó).

Jelenleg a legnépszerűbb a [Google](#).