# Információ visszakeresés vizsga 2015

A jegyzetet Dr. Góth Júlia Krisztina Tanárnő diái alapján készítettem magamnak összefoglalóként, tehát nem helyettesíti azokat.

Bartha András
users.itk.ppke.hu/~baran16

## Az információ-visszakeresés története, kialakulása.

Információ visszakeresés angolul information retrieval (IR)
Nem keverendő az információ kinyeréssel (information extraction (IE)) ami a struktúrálatlan szövegből strukturáltat állít elő.

### Az információ visszakeresés története:

- **Tartalomjegyzék:**
  Jegyzet arról hogy mely fontos rész hol található az adott műben.
- **Betűrend:**
  A művek címük alapján betűrendbe helyezése.
- **Index:**
  A művekhez csatoltak egy tőle különálló cetlit, ami tartalmazza a mű címét, főbb fejezeteit, rövid kivonatát, és hogy az adott mű pontosan hol helyezkedik el a könyvtárban. Így a művek mozgatása nélkül meg lehetett keresni a számunkra relevánsat.

A **világháló** az interneten működő, egymással hivatkozásokkal összekötött weblapok hálózata, melyet böngészőprogram segítségével lehet elérni. Ezek a weblapok a világ különböző helyein lévő számítógépeken vannak melyeket webhelyeknek hívunk. Egy webhelyet az internetcíme azonosítja. Az internet globális számítógépes hálózat, ami lehetővé teszi olyan elosztott rendszerek működtetését, mint pl a web.

- Internet: fizikai eszközök hálózata
- web: egy, az internetet kihasználó szolgáltatás
  - jelenleg több mint egy milliárd weboldal van
  - több mint 4 milliárdan használják az interneten elérhető szolgáltatásokat személyes vagy üzleti célokra.

### A világháló megjelenésének mérföldkövei (T. Berners-Lee → az internet atyja):

- 1989: Javaslat egy rendszer hálózat kialakítására.
- 1991. augusztus. 6: T. Berners-Lee elérhetővé teszi az első weboldalt, ami tartalmazta a világháló koncepcióját, és a böngészőkkel, webszerverekkel kapcsolatos alap információkat.
- 1993. április. 30: A világhálót mindenki számára elérhetővé és ingyenessé teszik.

### Weboldalak számának alakulása:

- 1993: 130 weboldal
- 1996: 100 000 weboldal
- 1997: 1 millió weboldal
- 2000: 10 millió weboldal
- 2015: több mint 1 milliárd weboldal

### A web kihasználtsága az Internet Live stats adatai szerint 2014 szeptemberében:

- 3.1 milliárd keresés a Google-el 24 óra alatt
- 170 milliárd e-mail üzenet a világban 24 óra alatt (több mint 90%-a spam :) )
- Az internetezők száma több mint 4 milliárd fő (föld lakossága kb 7 milliárd fő)

**Web története:**

- **Web 1.0 1990-től 2001-ig.**
  A web hőskora, a weboldalak statikusak voltak, és ritkán frissültek. A papír világot volt hivatott leképezni, ahol a weboldal csak információt közölt, a kommunikációra e-mail, vagy telefon szolgált.
  Legfőbb előfordulása:
  - cégek prospektusai
  - magánszemélyek bemutatkozó oldalai
  - híroldalak

  Kulcsszavak: You an I; Bring the web into our lives;
- **Web 2.0 2001-től napjainkig (2015)**
  a web 2.0 mint fogalomról a 2001-es dotcom válság után kezdtek el beszélni. (dotcom válság: egyre több tisztán web szolgáltatásokra épülő, részvény társaság indult, melyek csak fiktív értéket képviseltek valódit termék hiányában. Egyre kelendőbbek lettek az ilyen részvények, aztán amikor összeomlott a Nasdaq technológiai részvények tőzsdéje, sok ilyen jellegű cég tűnt el. ?)
  Az első témával kapcsolatos konferencia 2004-ben volt, ezután vált ismertté a web 2.0 fogalom Tim O'Reilly-net köszönhetően.
  Kulcsszavak: Us; Bring our lives into the web;
- **Web 3.0 majd valamikor**
  Szemantikus web? azaz a weben található információkat a keresőrendszerek valódi, jelentéssel bíró tartalomként kezelik. Célja egy olyan infrastruktúra létrehozása, amely lehetővé teszi az adatok integrálását, a közöttük levő kapcsolatok definiálását, jellemzését, valamint az adatok értelmezését. Tehát ne kulcsszavakra keressünk, és a válasz azon dokumentum amelyben szerepel az adott kulcsszó, hanem kérdést tehessünk fel a keresőnek, és az válaszol rá (részben ma (2015) is így működik pl: Wolfram alpha)
  Érzékelő web?
  Szociális web?
  Mobil web?
  Virtuális valóság egy formája?
  („Mindezek egyszerre, de még annál is több” (O'Reilly-Battele, 2009))

**Böngészés vs keresés:**

- **Böngészés:**
  Egy adott webcím megadásával felkeresünk egy oldalt, majd onnan linkek segítségével további oldalakra jutunk. Linkeket követünk, és azt reméljük, hogy előbb-utóbb megtaláljuk a keresett információt.
- **Keresés:**
  Nem egy adott weboldalt keresünk fel, hiszen nem tudjuk a címét, hanem egy keresőprogram segítségével keressük meg a számunkra hasznos weboldalakat. Ezt a kereső eszközt használjuk, hogy megtaláljuk a számunkra hasznos információkat tartalmazó weboldalakat.
  **Keresés típusai:**
  - Tematikus keresés (pl: startlap):
    A különböző weblapok témájuk szerint témakörökbe, majd alcsoportokba kerülnek besorolásba. Keresés során először az átfogó témakörök listájából választunk pl: tudomány, kultúra, sport, majd ezeken belül egyre részletesebb témák vannak, végül a weboldalak linkjei közül választhatunk.
  - Kulcsszavas keresés: (pl: google)

Megadunk egy keresőkérdést, majd a keresés eredménye a keresőkérdésnek megfelelő weblapok listája lesz, amely általában tartalmazza az egyes weblapok címét, tartalmának elejét, rövid tartalmi kivonatát stb...

**Az információ-visszakeresés kapcsolódó területei:**

- **Adattípusok:**
  - **Strukturálatlan adat:**
    Elektronikus formában tárolt adat, amelyre nem illeszthető jól használható adatmodell, emiatt adatelemzés szempontjából ezek elemzése a legnehezebb. Pl.: videó, audió, könyv, stb…
  - **Félig-strukturált adat:**
    Átmenet a strukturálatlan, és a jól strukturált adat között. Tárolási formát tekintve tartalmaznak olyan formális elemeket, amelyek elkülönítik az egyes tartalmi részeket egymástól, de ettől még nem válik strukturálttá. Pl: táblázatok, ahol meghatározott az adatok felosztása, vagy e-mail, ahol néhány adat strukturált mondjuk a feladó, címzett, tárgy, de az üzenet tartalma strukturálatlan.
  - **Strukturált adat:**
    Az információ elemi szintű adatokra bomlik, és ezek egy bizonyos modell alapján meghatározott kapcsolatrendszer szerint kötődnek egymáshoz. Általában táblázatos formában tároljuk, ahol az egyes oszlopok az objektumokat leíró tulajdonságokat tartalmazzák, a sorokat pedig az egyes objektumoknak feleltetjük meg.
- **Adatbázis kezelő vs információ-visszakereső rendszerek**
  - **Adatbázis kezelő:**
    Strukturált adatokon dolgozik, ahol jól definiált a struktúra, és a szemantika, valamint a lekérdezés is strukturált formában van megadva.
  - **Információ-visszakereső rendszer:**
    Strukturálatlan adatokon dolgozik, ahol természetes nyelvű szövegek vannak, és a szemantika is többértelmű, valamint a keresőkérdés is természetes nyelvű szöveg.
- **Szövegbányászat vs információ-visszakeresés**
  - **Szövegbányászat:**
    Új, a felhasználó számára eddig ismeretlen összefüggések, ismeretek kinyerése. Olyan tudásra kívánunk szert tenni, ami nincs közvetlen a rendelkezésre álló dokumentum állományban, csak elrejtve a tartalomban.
  - **Információ-visszakeresés:**
    A felhasználónak konkrét információigénye van, és a dokumentumokban fellelhető információk alapján próbáljuk meg kielégíteni azt.
- **Nyelvtechnológia:**
  A természetes nyelvű szövegek számítógépes feldolgozásával foglalkozik.
  A gép számára egy nyelv alapvetően értelmezhetetlen, ezért a nyelvtechnológia célja, hogy ezekből a strukturálatlan szövegekből a számítógép számára is értelmezhető adatokat/metaadatokat nyerjen ki.

**Az információ-visszakeresés modelljeinek áttekintése, csoportosítása.**

**Klasszikus információ-visszakereső modellek.**

Első, hagyományos modellek ilyenek. Matematikai módszereken alapulnak, melyek a kérdés, és a dokumentum távolságának matematikai mérésén alapszik. Könnyű implementálni, emiatt sok kereskedelmi kereső ezeken alapszik.

**Nem klasszikus modellek.**

Nem csak magát a dokumentumot vizsgálja, hanem a dokumentumok egymáshoz való viszonyát is.

Csoportosításuk:

- Információs logika alapú
- Szituációelmélet alapú
- **Kölcsönhatás alapú**

**A Boole-féle információ-visszakereső modell.**

Ez volt az első modell. Széles körben elterjedt, sok kereskedelmi kereső alapját képezi. A Boole logikára, és a klasszikus halmazelméletre épül. A visszakeresés lényege, hogy egy dokumentum tartalmazza-e a keresőkérdésben megadott kifejezéseket, vagy sem, így a Boole algebra segítségével megadhatóak a dokumentumok közti kapcsolatok.

**A vektortér modell.**

Széles körben kutatott és használt klasszikus modell, amelyet szöveges objektumok feldolgozására és információ-visszakeresésre már régóta használnak. Lényege, hogy a dokumentumokat, és a kérdést a tér egy-egy pontjaiként értelmezzük. Ez a tér egy ortonormált euklideszi tér, amelyben a tengelyek páronként egymásra merőlegesek. A tér dimenzióját az indexkifejezések száma határozza meg, és a visszakeresés azon alapszik, hogy a kérdés vektor, és a dokumentum vektorok milyen közel helyezkednek el a térben.

**Kifejezések kiválasztása, és a súlyok meghatározása a vektortér modellben.**

Nehéz elméleti és gyakorlati probléma, számos lehetséges megoldása van. A legnyilvánvalóbb az, hogy az indexkifejezéseket magukban a dokumentumokban keressük, és az előfordulások száma lesz a kifejezés súlya az adott dokumentumban.

**Hatványtörvény.**

**Fokszám:**

A hálózat egy elemének fokszáma, a hálózaton belüli kapcsolatainak száma.

**Fokszámeloszlás:**

A hálózat elemeinek számát tünteti fel, a fokszámuk függvényében.

- **Véletlen hálózat:**
  Ahol a hálózat elemeit véletlenszerűen kötjük össze, Poisson-eloszlást követ (viselkedésében hasonlít a haranggörbére). A legtöbb csomópontnak azonos számú kapcsolata van, így nem létezik kiemelkedően sok kapcsolatú csomópont.
- **Skálafüggetlen gráfok:**
  A legtöbb csomópontnak csupán kevés kapcsolata van, amelyeket néhány nagy súllyal összekapcsolt központ tart össze. Itt az eloszlás hatványfüggvényt követ. A hatványtörvénnyel leírt hálózatok esetében az elemek fokszámeloszlása szabályszerű:
  - Van kevés alkalommal előforduló elem, aminek nagyon sok kapcsolata van.
  - Majd ahogy a kapcsolatok száma csökken, az előfordulás nő.
  - Végül nagyon sok olyan elem fordul elő, aminek csak néhány kapcsolata van.

**Zipf-törvény.**

Minden szövegekben a szavak előfordulási gyakorisága Hatványtörvényt követ.
$f(r) = C*r^{-a}$
Ahol:

- C: az adott nyelvű dokumentumhalmazra (korpusz) jellemző konstans.
- r: a szavak helye az előfordulásuk száma szerinti rendezésben.

- a: a hatványtörvény kitevője.

Továbbá egy szó előfordulási gyakorisága fordítva arányos az előfordulási táblában szereplő rangjával. Emiatt a leggyakrabban szereplő szó közel kétszer gyakoribb, mint a második leggyakrabban szereplő, és közel háromszor gyakoribb mint a harmadik leggyakrabban szereplő. Ennek következménye, hogy ha egy korpuszból csak a leggyakoribb szavakat tartjuk meg, a többit töröljük, akkor a korpusz nagy része megmarad.

## Indexelési technikák, hasonlósági mértékek, visszakeresés a vektortér modellben.

### Automatikus indexkifejezés kinyerés lépései:

- **Lexikai egységek azonosítása:**
  Egy program megkülönbözteti a szavakat.
- **Stoplista alkalmazása:**
  A stoplista olyan szavakat tartalmaz, amelyek általában nem hordoznak jelentést az adott dokumentumban, így ezeket a szavakat kihagyjuk a további vizsgálatkor. A stoplista általában függ a felhasználási területtől, és az indexelés céljától.
- **Szótövesítés:**
  A szavakról eltávolítjuk a toldalékokat, hogy csak a szótő maradjon.
- **Gyakoriság számítás:**
  Ha megvannak a kifejezések, kiszámítjuk az egyes dokumentumokra, hogy hány alkalommal fordulnak bennük elő.
- **Inverz gyakoriság számítás:**
  Kiszámítjuk, hogy a kifejezések összesen hány alkalommal fordulnak elő.
- Végül inverz gyakoriság szerint sorba rendezzük őket, és a nagyon gyakran előfordulókat eltávolítjuk, mert nem hordoznak információt, és a legritkábban előfordulókat is töröljük, mert a jelentőségük elhanyagolható.

### Súlyszámok meghatározása:

Minden dokumentumra meg kell határozni, hogy a benne szereplő indexkifejezés milyen mértékben tükrözi az adott dokumentum tartalmát. Erre több technika is létezik:

- **Bináris súlyozás:**
  1: ha a kifejezés szerepel a dokumentumban, 0: ha nem. Ekkor a súlyfüggvény megegyezik a kifejezések előfordulási gyakoriságával.
- **Normalizált gyakoriság szerinti súlyozás:**
  - **Maxnormált:**
    Az előfordulások számát elosztjuk a leggyakoribb előfordulás számával, így ami a legtöbbször szerepel a dokumentumban 1 súlyt kap, a többi pedig 0 és 1 közöttit.
  - **Hossznormált:**
    Az előfordulások számát elosztjuk az indexkifejezések által alkotott vektor hosszával. (előfordulások négyzetösszegének gyöke)
- **Inverz dokumentum gyakoriság:**
  Az egyes indexek előfordulásának számát megszorozzuk az inverz dokumentum gyakoriság kettes alapú logaritmusával.
  $(\log_2(2 * {}^{\text{összes dokumentum}}/_{\text{amiben szerepel}}))$
- **Hosszra normalizált inverz dokumentum gyakoriság:**
  Az inverz dokumentum gyakorisággal kapott súlyokat osztjuk az

ugyanígy kapott súlyok hosszával.

**A visszakeresés lépései.**

A visszakeresés azon alapszik, hogy a kérdés vektor, és a dokumentum vektorok mennyire vannak közel egymáshoz.
A felhasználó által feltett kérdésben szereplő indexkifejezéseket is ugyanazzal a súlyszámítási módszerrel meghatározzuk, mint a többi dokumentumét, majd ezek alapján vizsgálunk hasonlósági mértéket. Ha a mérés eredménye nagyobb egy előre definiált küszöbértéknél, akkor a dokumentum válasz a kérdésre.

**Hasonlósági mértékek:**

- **Skalár szorzat (Dot product):**
  A kérdés vektoron, és a dokumentum vektoron skalárszorzatot számolunk.
- **Koszinusz mérték:**
  Skalárszorzat, osztva a vektorok hosszának szorzatával. Ha a súlyszámítás során hossznormált módszert használtunk, a nevező 1 lesz, tehát ez meg fog egyezni a kalárszorzattal.
- **Dice együttható:**

$$\frac{2\sum_{i=1}^{n} w_{ij} w_{ik}}{\sum_{i=1}^{n} \left(w_{ij} + w_{ik}\right)}$$

- **Jackard együttható:**

$$\frac{\sum_{i=1}^{n} w_{ij} w_{ik}}{\sum_{i=1}^{n} \frac{w_{ij} + w_{ik}}{2 w_{ij} w_{ik}}}$$

(Itt a nevezőben a nevező, kettőnek a hatványkitevője)

**Rangsortartás:**

Két hasonlósági mérték rangsortartó, ha az általuk számolt találtai listák sorrendje megegyezik, bármely keresőkérdés esetén. Ha két hasonlósági mérték rangsortartó, akkor ekvivalensnek tekinthetjük, és egymással helyettesíthetjük. Vektortér modellben, általában a hasonlósági mértékekre ez nem jellemző.

**Az információ-visszakeresés részfolyamatai.**

- **Objektumbázis definiálása:**
  Meghatározzuk a dokumentumok halmazát, amelyben keresni szeretnénk, meghatározzuk a szövegfeldolgozó műveleteket, és kijelöljük a szövegmodellt.
- **Szövegfeldolgozó műveletek alkalmazása:**
  Ezek átalakítják a dokumentumokat, és elkészítik azok logikai nézetét.
- **Szöveg indexelése:**
  Ez egy kritikus lépés a nagy terjedelmű szövegekben való gyors keresés szempontjából. Többféle struktúra létezik az indexek tárolására, a legelterjedtebb az inverz fájl struktúra.
- **Lekérdezés:**
  Itt a keresőkérdés, és az indexek összehasonlítása történik.

- **Rangsorolás:**
  A kapott eredményekből, a hasonlósági mértékek segítségével elkészít egy rangsort, majd visszaadja a felhasználónak.

**Általános információ-visszakereső rendszer architektúrája.**

- **Objektumbázis:**
  Tárolja a dokumentumok halmazát, amelyeken a keresés történik. A dokumentumok a bázisba manuálisa, vagy program segítségével kerülhetnek.
- **Indexelő modul.**
  Az objektumbázisban lévő dokumentumokból készít indexet, melyet rendszerint inverz fájl struktúrában tárol.
- **Lekérdező modul.**
  Beolvassa a keresőkérdést, és átalakítja a gép számára felhasználható formára. Az indexek segítségével kiválasztja azokat a dokumentumokat, amelyek megfelelő válaszok a kérdésre, és ezeket átadja a rangsoroló modulnak.
- **Rangsoroló modul.**
  Kiszámítja a kapott dokumentumokra a hasonlósági értékeket, majd ezek alapján csökkenő sorrendbe rendezi, és ezt megjeleníti a felhasználónak, mint találati lista.

**Kategoricitás, és bizonytalanság.**

**Shannon -féle információ:**

Az információelmélet szerint az üzenet nem azonos az információval. Az átvitt adatmennyiség információtartalma attól függ, hogy a vétel helyén mennyire szünteti meg a bizonytalanságot.
Szerinte a vektortér modellben a találati listához kapcsolódó bizonytalanság:

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

ahol $P_j$:

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

ahol ρ az adott dokumentumnak, a kérdésre vonatkozó relevanciaértéke.

**Kategoricitás:**

Egy információ-visszakereső rendszer kategorikusabb a többinél, ha a találati listájához kapcsolódó bizonytalanság kisebb, mint a többié.

**Az információ-visszakereső módszerek relevanciahatékonyságának mérése.**

A relevanciahatékonyság mérésekor azt vizsgáljuk, hogy milyen precíz egy találati lista, mennyire relevánsak a válaszai. Többek köz ezzel a módszerrel is értékelhetjük egy visszakeresés hatékonyságát.
A visszakeresési hatékonyság becslésénél figyelembe kell venni azt, hogy hogyan hajtjuk végre a visszakeresési folyamatot. Ez lehet:

- **Interaktív folyamat:**
  Ekkor a felhasználó interaktív lépéseken keresztül adja meg az információigényét a rendszernek. Itt a hatékony keresést több tényező is befolyásolja pl: a felhasználó ügyessége, az interfész tulajdonságai stb...
- **A kérdések kötegelt feldolgozása:**

Itt a legfontosabb, a rendszer által generált találati lista minősége, melyet kiértékelhetünk pl.: **Cranfield paradigma** alapján, melyet **standard tesztkollekciókon**, **standard mérőszámok** alkalmazásával végzünk.

**Cranfield paradigma:**

Egy rendszer hatékonyságát kiértékelhetjük egy **standard tesztkollekció** alapján meghatározott teljesség-pontosság grafikon alapján.

**Standard tesztkollekció:**

Elemei:

- Dokumentumok
- Előre meghatározott kérdések
- Relevancia ítéletek

A tesztkollekciók mindhárom részét szakértők határozzák meg manuálisan, és az egyes komponenseket a vizsgált keresőtől függetlenül állítják össze. Ezek segítségével lehetőségünk nyílik információ-visszakereső rendszerek hatékonyságának gyors kiértékelésére, valamint az eredmény tükrében különböző rendszereket közvetlen összehasonlíthatunk.

**Standard mérőszámok:**

- Teljesség (felidézés): $^{\text{megtalált releváns}}/_{\text{összes releváns}}$
- Pontosság: $^{\text{megtalált releváns}}/_{\text{találati lista számossága}}$

Ezek a mérőszámok meghatározzák a hasonlóságot a kereső rendszer által visszaadott, és a szakértők által meghatározott dokumentumok között, ezzel becslést adva a keresési stratégia hatékonyságára.
A kiértékelésben természetesen nem csak ezek használhatók, vannak más mérőszámok is pl:

- Selejt: $(^{\text{megtalált irreleváns}}/_{\text{összes irreleváns}})$
- Harmonikus közép:

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

Azonban a Cranfield paradigmában csak a teljességet, és a pontosságot mérjük.

**Teljesség-pontosság grafikon:**

A találati listában szereplő releváns dokumentumokat ábrázoljuk egy grafikonon, melyen a vízszintes tengely az adott dokumentumra a teljesség, a függőleges pedig a pontosság értékét jelöli. Általában a felidézési szintek nem egyeznek meg a 11 standard szinttel (0-tól 100-ig 10-es léptékkel zárt tartományként) mert nem minden esetben van pontosan 10 releváns dokumentum az adott kérdésre. Ilyenkor az úgynevezett **interpolációs módszer**rel határozzuk meg a teljesség értékekhez tartozó pontosság értékeket.

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

Az így meghatározott pontosság értéke a j-edik felidézési szintnél megegyezik a j és j+1 szint közé eső legnagyobb pontossággal. (lépcsős lesz a grafikon)

**Átlagos pontossági értékek:**

Minden egyes felidézési szinthez (teljesség érték) átlagoljuk a különböző kérdésekhez tartozó pontossági értékeket. Képlete:

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

$\bar{P}(r)$= 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 érték** (Mean Average Precision)

Egy adott felidézési szinten mért átlagos pontossági értékek átlaga.

**Szabványos tesztadatbázisok a mérésekben.**

- **TREC**
- **ADI**

  Ez a legkisebb tesztkollekció, tartalma: dokumentumok, kérdések, relevancia ítéletek, boole kérdések listája.
- **MEDLINE**
- **CACM**
- **CISI**
- **TIME**
- **REUTERS**

**Webes technológiák, Webkeresőmotorok.**

Ha a web-en próbálunk információhoz hozzájutni, többféle módon tehetjük.

- **Böngészés (browsing)**

  Az egyet weblapokon lévő hiperhivatkozások segítségével weblapról weblapra ugrálhatunk. Így egy csomó információhoz hozzájuthatunk, de lassú, és lehet, hogy soha nem találunk rá az eredetileg keresett dokumentumra.
- **Keresés**

  Webszervereken keresztül elérhető szoftverek segítségével, kívánt kulcsszavak alapján megtalálhatjuk a megfelelő dokumentumokat.

**Web keresőrendszerek:**

Felhasználói szempontból két fő komponensről beszélhetünk:

- **Felhasználói felület:** A felhasználó ezen keresztül végezhet keresést.
- **Kereső motor:** A felhasználó elől rejtve végzi a keresést.

A ma elterjedt keresők működésének lényege, hogy végiglátogat oldalakat, majd miután ezeket indexelte valamilyen szempont szerint, az indexeket eltárolja egy adatbázisban. Majd a felhasználó által megadott kérdés alapján ebben az indextáblában végez keresést. Az oldalak végiglátogatását az úgynevezett crawler végzi.

Léteznek azonban ettől eltérő kereső is, pl. az úgynevezett **katalógus rendszerek.** Ezek megpróbálják kategorizálni a weben tárolt anyagokat. Ezek adatbázisát manuálisan töltik fel, és kézzel rangsorolják. Előnyük, hogy pontosabb találatot adnak egy adott kulcsszóra, hátrányuk, hogy a web méretéből, és

áttekinthetetlenségéből adódóan a kategorizálás hosszadalmas folyamat. Általában lehetőség van arra, hogy az oldal készítője maga regisztrálja a kereső adatbázisába a weboldalát. Ilyen például a startlap.

**Crawler:**

Pásztázza a webhelyeket releváns dokumentumok után kutatva, ezeket letölti, rangsorolja, és az értékelése alapján eltárolja az adatbázisába. Majd ha végzett, az oldalon található linkek alapján megy tovább más webhelyre. Ez a módszer viszonylag aktuális képet tud adni a web tartalmáról, mivel a robotok folyamatosan végzik a munkájukat. Így működik pl.: a Google, Bing, stb…

Tartozik hozzá egy **crawler control modul** mely eldönti, hogy mely URL-eket keresse fel legközelebb a crawler.

**Robot exclusion:**

A robotoknak megadhatók előírások, amiket be kell tartaniuk.
Minden webhely esetén meg kell vizsgálniuk a robots.txt fájlt és a weboldalak meta-tag-eit. Ezekkel a weboldal készítői szabályozhatják a robotok működését. A robots.txt fájl tartalmazza, hogy mely oldalakat nem tölthet le a robot. Továbbá általános korlátozás, hogy egy webhelyre percenként csak egyszer szabad kérést küldenie a crawlernek, hogy ne terhelje a szervert.

**Kulcsszavak:**

Az egyes weboldalakhoz kulcsszavakat rendelnek, melyek tükrözik az oldal tartalmát, ezzel gyorsítva a visszakeresést. Ez történhet automatikusan, vagy manuálisan is. Az automatikus előnye, hogy olcsó, gyors, és fenntartható, a manuálisé pedig hogy megbízható, és pontos. Ennek folyamata, hogy a crawler letölti a weboldalt, átadja az értelmezőnek, ami elvégzi a tartalomszűrést.

**Luhn és Hayes elmélete:**

Szerintük a dokumentumra jellemző szavak egy bizonyos sávban találhatóak meg, a nagyon sűrűn, és a nagyon ritkán előforduló szavak között.

**Webkereső moduljai:**

- **Dokumentumtár**

  Itt tárolódnak ideiglenesen a crawlerek által letöltött oldalak. Innen kerülnek az indexelő modulhoz további feldolgozásra, azonban a keresés során a válasz nem az itt szereplő dokumentum lesz, hanem az eredeti helyen lévő weboldal. A fontosabb vagy népszerű oldalak hosszabb ideig tárolódnak, a többire azonban nincs szükség.
- **Indexelő modul**

  A dokumentumtárban lévő weblapokat dolgozza fel, kiszedi a szavakat az oldalakról, és mindegyikhez rögzíti, hogy mely URI-n fordult elő. Ennek eredménye egy hatalmas táblázat, az úgynevezett **text index**. A webhelyekhez rendelhet még számos részstruktúrát.
- **Lekérdező modul**

  Beolvassa, és a megfelelő alakra transzformálja a felhasználó kérdését, majd az indexek segítségével meghatározza, mely dokumentumok illeszkednek a felhasználó kérdésére. Ezeknek az oldalaknak a listája kerül tovább a rangsoroló modulhoz.
- **Rangsoroló modul**

A kapott lista elemeit hasonlósági értékük szerint csökkenő sorrendbe rendezi, ez lesz a találati lista. Azonban a felhasználó számára megjelenített listában nem a tárolt dokumentumok szerepelnek, hanem az eredeti weboldalak linkjei.

**Web-metakeresőmotorok. Keresőkkel szemben felmerülő felhasználói igények.**

A metakeresők más keresőknek vagy adatbázisoknak a találati listája alapján állítják össze a saját keresési eredményeiket. Ezt úgy csinálják, hogy a keresett kérdést elküldik több webkeresőnek, webtárnak és adatbázisnak, majd összegyűjtik és egy bizonyos eljárással egyesítik az eredményeket. Általában a fellelhető metakeresők abban különböznek egymástól, hogy más módszereket alkalmaznak a begyűjtött eredmények egyesítéséhez, és más keresőket használnak.

Előnyeik, hogy több keresőt tudunk elérni egyetlen interfésszel, így használatuk során relevánsabb találati listát kaphatunk.

Egy metakereső annál jobb, minél több webes keresőt, adatbázist használ, de minősége nem kizárólag ezen múlik. Sok esetben lehetőségünk van kiválasztani milyen webes keresőket használjon a keresés során, illetve hogy hány találatot jelenítsen meg az egyes keresőktől.

Vannak metakeresők, melyek a visszakapott címekről letöltik az oldalt, és tartalom alapján rangsorolják azt. Ez természetesen időigényesebb, de így kiszűrhetőek a már nem elérhető weboldalak, és jobb rangsorolást kapunk az aktuális tartalom szűrése miatt.

Általános felépítése:

- **Interfész modul:**
  Beolvassa a felhasználó kérdését, majd szótövesítés után elküldi az általa használt keresőknek.
- **Metakereső motor:**
  A visszakapott találati listák első n elemét letölti, majd azonosítja a kulcsszavakat, stoplistáz, szótövesít, és a feldolgozott oldalakat a dokumentumtárba teszi.
- **Dokumentumtár:**
  Az interfész modul által letöltött és feldolgozott weboldalakat tartalmazza.
- **Rangsoroló modul:**
  Az eredményeket különböző szempontok szerint rangsorolja, majd visszaadja az interfész modulnak, hogy megjelenítse mint találati lista.

**Felmerülő felhasználói igények:**

Fontos, hogy az eredmények esztétikusan, átláthatóan jelenjenek meg.
A mai keresőket gyakran ellátják haladóknak való funkciókkal pl.: egy adott oldalon lehet keresni, egy bizonyos dátum után frissített oldalakon lehet keresni, stb.
A felhasználót meg kell tanítani helyesen feltennie a kérdését a pontos találatok érdekében.

**Webes keresők és metakeresők kiértékelése.**

Egy webes kereső rendszert a találati listája alapján minősíthetjük. Pl.:

- Felállítunk egy elfogulatlan kérdéssort úgy, hogy kijelölünk egy témát, majd pontosan meghatározzuk a keresőkérdéseket. Az elfogulatlanság érdekében általános, és speciális kérdéseket is választani kell.
- Ezután, a tesztelés végrehajtása előtt meg kell határoznunk, hogy a válaszokat hogyan fogjuk minősíteni. Fontos hogy a válaszok megismerése előtt alakítsuk ki a módszert, különben szubjektív lesz az értékelés. Meg kell határoznunk azt is, hogy

egy választ mikor tekintünk relevánsnak.

- Ezután, a találatokat kategóriákba soroljuk. Ezt végezhetjük binárisan, tehát a két kategória releváns, vagy irreleváns.
  - **Releváns:**
    Általában akkor tekintünk egy dokumentumot relevánsnak, ha az oldal címe, és az oldal is tartalmazza a keresett kifejezést, vagy a kérdés által meghatározott témához tartozik az oldal.
  - **Irreleváns:**
    Ha nem tesz eleget a releváns feltételeinek,
    vagy ha dupla link, azaz többedjére szerepel a találati listában,
    vagy inaktív link, tehát az oldal mád nem elérhető.
- Végül a találati lista elemeit a fentebb meghatározott kategóriákba soroljuk. Webes keresőknél nem lehetne végignézni az egész találati listát, ezért csak az első 10 vagy 20 választ vesszük figyelembe. (tanulmányok szerint amúgy csak az első 11 találatot nézzük meg)

**Leighton módszer:**

Egy módszer a webes keresők relevancia hatékonyságának kiértékelésére. A módszer a mérőszám meghatározásához csak az első 5, vagy 10 találatot veszi fegyelembe. A mérőszám értéke 0 és 1 közé esik, mely a pontosságot adja meg a hatékonysággal súlyozva. 0 ha nem kapunk választ, vagy az összes válasz irreleváns, és 1, ha az összes válasz releváns, és megfelelő számú választ kaptuk vissza.
Tulajdonságai:

- A relevanciát bináris skálán mérjük: ha a linket az 1. kategóriába soroltuk, akkor 1 pontot ér, ha a 0. kategóriába, akkor pedig 0 pontot ér. A dupla linket a mérőszám meghatározásánál vesszük figyelembe.
- Figyelembe vesszük a releváns linkek rangsorbeli elhelyezkedését. Minél előrébb helyezkednek el a releváns linkek, annál nagyobb lesz a mérőszám értéke.
- A mérőszám tükrözi, ha a kereső kevesebb találatot ad vissza ugyanannyi jó találat esetén, vagyis magasabb a precizitása. Így persze könnyebb megtalálni a releváns linkeket. (ez persze nem jelenti azt hogy az a legjobb kereső, ami egy választ sem ad vissza)

Lépései:

- A mérni kívánt keresőmotor kiválasztása.
- Relevancia kategóriák definiálása.
- Csoportok definiálása.
- Súlyok definiálása.
- Mértékek megadása.
- Számolás.

Képletek 5 és 10 elem mérése esetén:

$$\frac{(Pontok_{\ 1.-2.link} \cdot 10) + (Pontok_{\ 3.-5.link} \cdot 5)}{35 - \left[(5 - találatok\_száma_{\ első\ 5!}) \cdot 5\right]}$$

$$\frac{(Pontok_{1.-2.link} \cdot 20) + (Pontok_{3.-5.link} \cdot 17) + (Pontok_{6.-10.link} \cdot 10)}{141 - \left[(10 - találatok\_száma_{első\,10!}) \cdot 10\right]}$$

**RP módszer** (relatív pontosság)

Egy találati lista pontossága mérhető, erre alapozva megadható egy webes metakereső relatív pontossága úgy, hogy a metakereső találati listáját összevetjük a használt keresők találati listájával.
Ezt úgy kapjuk meg hogy minden keresés alkalmával elosztjuk azon találatok számát, amik szerepelnek a metakereső találati listájában, és az egyik használt kereső első 5 (vagy 10) találata között, azzal a számmal, ahány találatot összesen visszaadott a metakereső.

## Kapcsolatelemzésű információ-visszakereső módszerek. Az $I^2R$ módszer.

### Klaszterezés:

A visszakeresendő dokumentumokat diszjunkt halmazokba csoportosítjuk, ezeket nevezzük klaszteréknek. Ezekben a klaszterekben, valamilyen értelemben hasonló dokumentumok állnak. Minden klaszternek van egy képviselője (reprezentáns), aki nem feltétlen tagja a klaszternek. A visszakereső rendszer a kereső kérdéshez egy reprezentánst társít, és a válasz, a reprezentáns által képvisel klaszter összes tagja lesz.
Alapja a **klaszter hipotézis,** miszerint a szorosan asszociált dokumentumok ugyanarra a kérdésre relevánsak. Kialakítása **priori** azaz a keresés előtt, a kérdéstől függetlenül történik a csoportok kialakítása.
Az algoritmus stabil

- **növekedésre:** tehát új dokumentum hozzáadása nem változtatja meg jelentősen a klaszterek szerkezetét.
- **leírási módra:** tehát a dokumentumok kismértékű megváltozása alig befolyásolja a klaszter szerkezetet.
- **rendezésre nézve:** tehát a szerkezet független a dokumentumok rendezésétől.

Módjai:

- Fix klaszterezési mód:
  Még a kérdés feltevése előtt, a kérdéstől függetlenül alakulnak ki a klaszterek, így a kérdések nincsenek hatással a klaszter kialakítására.
- Adaptív klaszterezési mód:
  A kérdés feltevése után, a kérdés hatására alakulnak ki a klaszterek, így a kérdések befolyásolják a klaszterek kialakítását

### Kölcsönhatás-alapú információ visszakeresés ($I^2R$):

A felhasználó kérdése hatást gyakorol az eredeti dokumentumhalmazra azáltal, hogy megváltoztatja a dokumentumok közti kapcsolatokat. Megvalósításának módja, hogy a dokumentumok nem egymástól elszigetelt egységeket képeznek, hanem egy összekötött hálózatot alkotnak, amit a felhasználó kérdése részlegesen átalakít. Matematikai modellje a mesterséges neuronhálózat állapotegyenletén alapszik, tehát az objektumok hálózata megfelel egy neuronhálózatnak, ahol az objektumok a neuronok. Az objektumok közti kapcsolatokat minden alkalommal újraértelmezzük, amikor új objektumok kapcsolunk a hálózathoz pl.: kérdést. Ezek alapján a kérdést is objektumnak tekintjük.

A dokumentumok között súlyozott és irányított kapcsolat van, amit számolhatunk a:

- **Gyakoriság alapján:**
  Ez megadja, hogy egy közös indexkifejezés hányszor fordul elő a másik

dokumentumban, osztva a másik dokumentum indexeinek számával.

- **Inverz gyakoriság:**
  Ez megadja, hogy egy kifejezés mennyire különbözteti meg dokumentum tartalmát a többi dokumentumétól.

**Az aktiváció terjedése:**

A kérdéstől indul, és mindig a legerősebb kapcsolat mentén terjed neuronról-neuronra. Minden neuront csak egyszer látogatunk meg, és ez addig tart, amíg körbe nem érünk.

**A PageRank módszer.**

A Google keresője ezen alapszik. Larry Page és Sergey Brin találta ki.
Alapja a link népszerűség, tehát hogy egy dokumentum annál fontosabb, minél többen hivatkoznak rá. A PageRank ennek egy továbbfejlesztett változata: nem egyforma súllyal veszi figyelembe a dokumentumokra való hivatkozásokat, hanem súlyozza a hivatkozó oldal fontosságával is.
Tekintsük a webet egy irányított gráfnak. A csúcsok a weboldalak, az irányított élek pedig a weboldalak közötti irányított hivatkozások. Az oldalak „elosztják” a fontosságukat a hivatkozásaikkal, k kimenő hivatkozású oldal minden hivatkozása 1/k-t ér.
A képletben szerepel még egy csillapító tényező. Ez annak a valószínűsége, hogy a felhasználó bármely lépésben tovább folytatja a szörfölést. Ennek értéke: 0.85
A PageRank képlete:

$$\mathrm{PageRank}(i) = (1-d) + d \sum_{j \in M(i)} \frac{\mathrm{PageRank}(j)}{L(j)}$$

A PageRank értéket egy rekurzív eljárással közelítjük. Ez úgy történik, hogy addig számoljuk egy hálózatban a PageRank értékeket, amíg az előző érték és az új érték közti különbség kisebb nem lesz egy küszöbszámnál. Ha ezt elértük, a PageRank érték az aktuális hálózatra tekintve végleges.

**Lógó link:**

Hivatkozás, ami olyan dokumentumra mutat, ami nem tartalmaz linket.

A lógó linkeket nem vesszük figyelembe PageRank számításkor, mert akkor a rendszerben lévő összes szavazat kevesebb lenne az oldalak számánál.