Tartalomjegyzék

4. fejezet....2
Információ visszakereső módszerek relevanciahatékonyságának mérése....2
4.1 A relevanciahatékonyság mérése....2
4.2 A standard tesztkollekció....3
4.3 A hatékonyság mérése, mérőszámok....3
4.4 A teljesség-pontosság diagram felrajzolásának menete....5
5. fejezet....8
Webes technológiák....8
5.1 A világháló....8
5.2 Webkeresőmotorok....8
5.2 Web metakereső motorok....13
5.3 A keresőkkel szemben felmerülő felhasználói igények....18
5.4 A Google kereső használata....19
5.4.1 Kép keresése....19
5.4.2 Speciális keresési funkciók....21
5.4.3 Találatok szűrése....23
6. fejezet....26
Webes keresők és metakeresők relevanciahatékonyságának mérése....26
6.1 Webes visszakeresés relevanciahatékonyságának mérése....26
6.2 A Leighton-módszer....27
6.3 A relatív pontosság mérésének RP módszere....31
7. fejezet....33
Kapcsolatelemzésű információ visszakereső módszerek....33
7.1 I2R Adaptív Klaszterező Technika Információ–visszakereséshez....33
7.1.1 Klaszterezés....33
7.1.2 Asszociatív Kölcsönhatás alapú Információ Visszakeresés....33
7.2 PageRank....38
7.3 HITS....44
7.2 SALSA....49
Irodalomjegyzék....52

4. fejezet

Információ visszakereső módszerek relevanciahatékonyságának mérése

4.1 A relevanciahatékonyság mérése

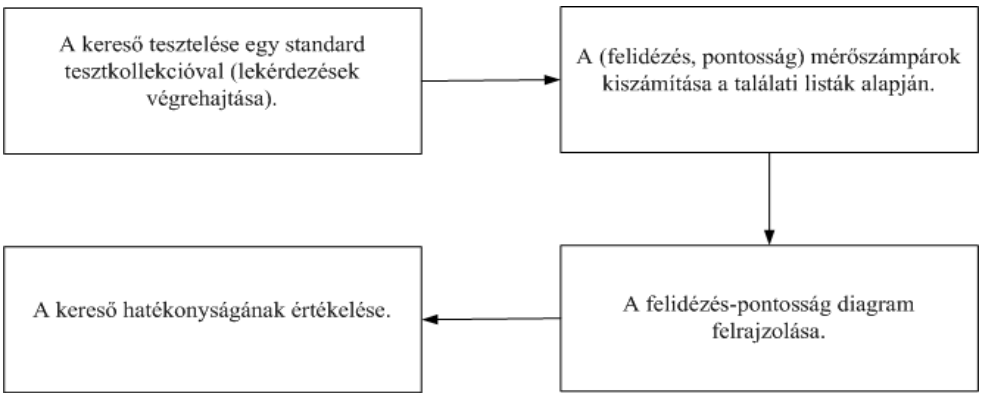
Az információ–visszakereső rendszereknél azt vizsgáljuk, hogy milyen precíz a találati lista, azaz mennyire relevánsak a válaszok. Ezt a vizsgálati módszert visszakeresési hatékonyság kiértékelésnek (retrieval performance evaluation, retrieval effectiveness evaluation) nevezzük. A relevanciát alapvetően bináris skálán (binary relevance) mérjük, ekkor egy dokumentum vagy releváns, vagy irreleváns az információigény tekintetében. A releváns dokomentumokat tovább osztályozhatjuk aszerint, hogy mennyire elégítik ki a keresőkérdést. Ebben az esetben osztályozott relevanciáról (graded relavance) beszélünk, és például a következő osztályokat használhatjuk: meglehetősen releváns, nagyon releváns [52]. Nem várható el, hogy a visszaadott dokumentumok pontos válaszok legyenek, ha a felhasználók információigényét nem tükrözik teljes mértékben az információ–visszakereső rendszerrel közölt kereső kifejezések. Ekkor a válaszokat rangsorolni kell aszerint, hogy mennyire relevánsak a kérdésre.

A visszakeresési hatékonyság becslésénél figyelembe kell venni azt, hogy hogyan hajtjuk végre a visszakeresési folyamatot. A visszakeresés állhat egyszerűen egy kérdés kötegelt 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. A kötegelt 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. 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 [24.]. 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).

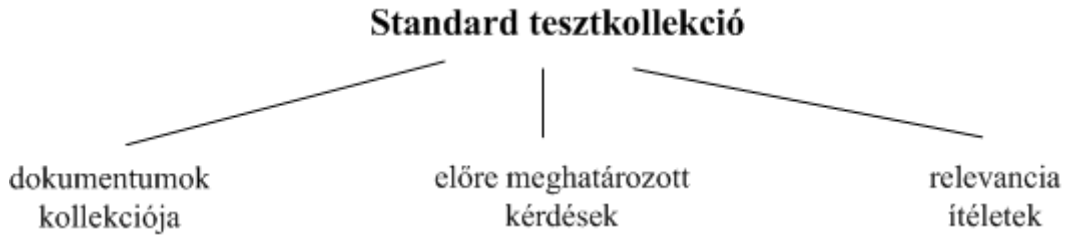
A tesztelés folyamatát a következő folyamatábra szemlélteti:



1. ábra. Tesztelés a Cranfield paradigma alapján

4.2 A standard tesztkollekció

A standard tesztkollekció a következő ábrán látható komponensekből áll.



2. ábra. A standard tesztkollekció

A tesztkollekció mindhárom részét külön-külön szakértők határozzák meg manuálisan. Az egyes komponenseket előre, a vizsgált keresőtől függetlenül állítják össze.

A legismertebb standard tesztkollekciók: TREC, ADI, MED, CISI, CACM. A tesztkollekciók lehetővé teszik az információ-visszakereső rendszerek információ-visszakeresési hatékonyságának gyors kiértékelését, valamint az eredmények tükrében a különböző rendszerek közvetlen összehasonlítását.

4.3 A hatékonyság mérése, mérőszámok

Az információ-visszakereső rendszer által megadott válaszok relevanciáját leggyakrabban két standard mérőszámmal, a teljességgel és a pontossággal jellemezzük. 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.

A kiértékeléshez természetesen használhatók még további mérőszámok is, mint pl. kiesés (fallout), R-pontosság (R-Precision), harmonikus középérték (harmonic mean), stb.. A Cranfield paradigmában azonban csak a teljességet és a pontosságot mérjük.

A teljességet és a pontosságot a következő módon határozzuk meg:

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

$$Teljesség = \frac{|Ra|}{|R|}$$

Ahol:

- 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.

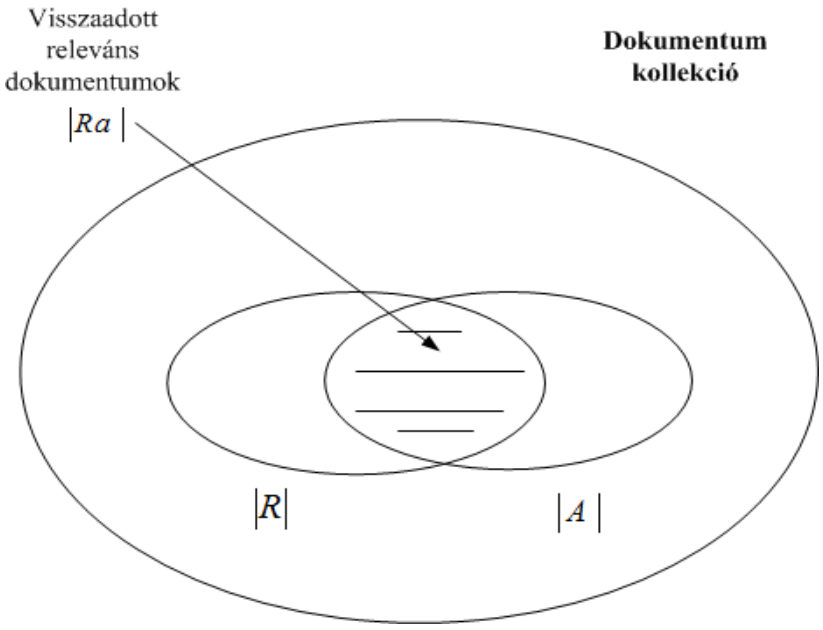
A pontosság a megtalált releváns és az összes visszakapott dokumentum hányada (adott kérdés esetén).

$$Pontosság = \frac{|Ra|}{|A|}$$

Ahol:

- 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.

A visszakeresési módszer egy adott kérdésre vonatkozó teljességét és pontosságát a következő ábra szemlélteti.



3. ábra. Teljesség és pontosság

A teljesség és a pontosság fenti, 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.

4.4 A teljesség-pontosság diagram felrajzolásának menete

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. 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* kereső kifejezésre tíz (szakértők által meghatározott) releváns dokumentum van.

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

A találati lista:

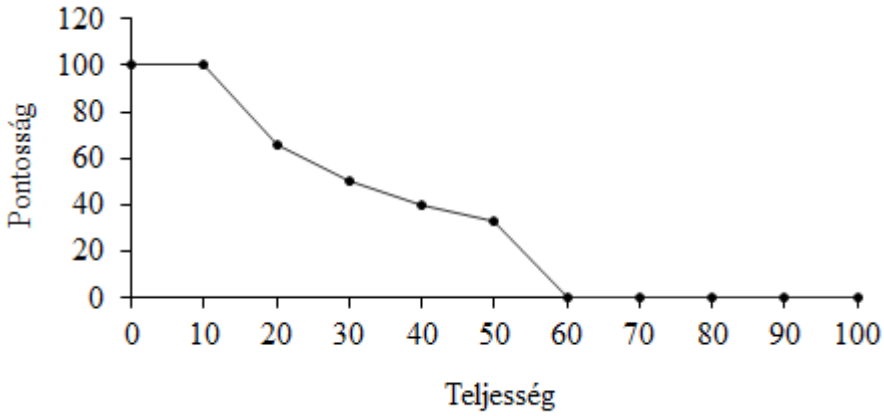
1. D_{123} •
2. D_{84}
3. D_{56}•
4. D_6
5. D_8
6. D_9•
7. D_{511}
8. D_{129}
9. D_{187}
10. D_{25}•
11. D_{38}
12. D_{48}
13. D_{250}
14. D_{113}
15. D_3 •

A *q* kereső kifejezésre releváns dokumentumokat pont jelöli. A találati listát sorrendben, az első dokumentumtól vizsgálva a következőket figyelhetjük meg.

- A találati lista első, D_{123}–as dokumentuma releváns. Továbbá ez a dokumentum az *Rq* összes releváns dokumentum halmazának 10%–át jelenti. Tehát azt mondjuk, hogy 10%–os teljességi szint esetén a pontosság 100%.
- A találati lista harmadik, D_{56}–os dokumentum a következő releváns dokumentum. Ebben a pontban azt mondjuk, hogy a pontosság megközelítőleg 66% (három dokumentumból kettő releváns) 20%–os teljességi szintnél (tízből két releváns dokumentumot vizsgáltunk meg).
- Ha tovább folytatjuk a találati lista vizsgálatát, akkor megkapjuk a pontosságot a teljességi szinteknél.

A teljesség és a pontosság klasszikus mérőszámok az IR rendszerek hatékonyságának kiértékelésére. A (teljesség, pontosság) rendezett számpárokat grafikusan ábrázoljuk az ún. Teljesség – pontosság grafikonon. A teljesség-pontosság grafikon a következő 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. A példához tartozó grafikont a következő ábra mutatja.



4. ábra. Pontosság a 11 standard teljességi szintnél

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.

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

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

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

Az interpolációval meghatározott pontosság értéke a j–edik teljességi szintnél megegyezik a j–edik és a j+1–edik teljességi szint közé eső legnagyobb ismert pontossággal. Pl. ha egy kérdésre három (szakértők által meghatározott) releváns válasz van, akkor a pontosságot csak a következő három teljességi szintnél tudjuk meghatározni: 33,3%, 66,6%, 100%. Ekkor a standard teljességi szinteknél az interpolációval meghatározott pontosságok a következők lesznek: 0%, 10%, 20%, 30%–nál a 33,3%–os teljességi szinthez tartozó pontosság, 40%, 50%, 60%–nál a 66,6%–os teljességi szinthez tartozó pontosság, míg 70%, 80%, 90%, 100%–nál a 100%–os teljességi szinthez tartozó pontosság. Az interpolációval meghatározott értékekből felrajzolható a teljesség–pontosság grafikon.

Az algoritmusok visszakeresési hatékonyságát az összes kérdés figyelembevételével szokás kiértékelni, ezért átlagolni kell a különböző kérdésekhez tartozó grafikonokat minden egyes felidézési szint esetén. Az átlagos pontossági értékeket a következő képlettel határozzuk meg:

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

ahol:

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

A különböző kérdésekhez tartozó teljesség–pontosság grafikonok átlagolásával kapott grafikont teljességi–pontosság diagramnak hívják és a visszakeresési algoritmusok információ–visszakeresési hatékonyságának összehasonlítására használják. Az újonnan kifejlesztett algoritmusok hatékonyságát leggyakrabban a klasszikus vektor tér modell (SMART) visszakeresési hatékonyságával szokták összehasonlítani. Az információ–visszakereső rendszerek kiértékelésére jelenleg használt standard módszer a teljesség–pontosság diagram, amely lehetővé teszi a teljes válasz halmaz és az információ–visszakereső algoritmus minőségének számszerű meghatározását (értékelését).

5. fejezet

Webes technológiák

5.1 A világháló

A világháló (World Wide Web) elektronikus dokumentumok hálózata, amelyeket speciális számítógépeken, szervereken tárolunk. A dokumentumok különböző típusú adatokat tartalmazhatnak, mint pl. szöveg, kép, hang. A dokumentumok tárolási egységét weblapnak hívjuk. Minden lapnak van egy egyedi azonosítója, az URL (Universal Resource Locator), ez azonosítja a lap helyét a szerveren.

A weblapok számát a web méretének is nevezzük. (jelenleg több, mint 14 milliárd weblap van). A legtöbb web dokumentum HTML (Hypertext Mark Up Language) formátumú, és számos ún. címkét (tag) tartalmaz (a címkék fontos információt szolgáltatnak az oldalról).

Pl. a következő címke : <b>, amely a félkövér betűtípust jelöli, általában növeli a hivatkozott kifejezés fontosságát.

A weboldalak általában kevésbé strukturáltak (nincs előírás a formátumra vonatkozóan), és változatosak. Tetszőleges nyelvűek lehetnek, sőt egy oldalon belül többféle nyelv is használható. Az oldalak tartalma, helyesírása gyakran nem ellenőrzött. Az oldalak nyelvi stílusa is jelentős eltéréseket mutat. Az oldalak hossza elméletileg nem korlátozott. A web oldalak különböző adattípusokat tartalmazhatnak: szöveg, kép, hang, videó, végrehajtható kód. Számos formátum használható: HTML, XML, PDF, MSWord, mp3, avi, mpeg, stb..

A klasszikus információ-visszakeresés a dokumentumokat statikusnak tekinti. A web oldalak azonban dinamikusak, vagyis tartalmuk gyakran frissülhet, törölhetik őket és újakat hozhatnak létre, dinamikusan generálódhatnak. A weboldalak összekapcsolhatók hiperlinkek segítségével, ezáltal weboldalak egy hálózata jön létre. A weboldalaknak ezt a kapcsolat alapú fontosságát különböző tényezők befolyásolhatják:

- URL egyik weboldalon egy másikra,
- horgony (anchor) szöveg,
- aláhúzott, „kattintható” szöveg.

5.2 Webkeresőmotorok

Az Internetre az emberek úgy tekintenek ma, mint egy könyvtárra, ám a könyvtárban se lehet eligazodni könyvtáros nélkül. 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.

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 a linkek használatával. Ám ez az Internet méretei és a mai emberek időhiánya miatt egyáltalán nem hatékony módszer. Abban az esetben viszont, ha nem csak egyszerűen böngészünk a világhálón, hanem egy konkrét információ igényünket szeretnénk hatékonyan kielégíteni webes keresőket használunk.

A webkeresők webszervereken keresztül elérhető szoftverek, melyek segítségével a kívánt kulcsszavak alapján megtalálhatjuk a megfelelő weboldalakat. Egy keresőrendszer esetében

alapvetően két fő komponensről beszélhetünk. Az egyik a kereső motor (search engine). Ez, a felhasználó elől rejtett modul végzi magát a keresést. A másik pedig a felhasználói felület. Az, az interfész, amin keresztül a felhasználók kereshetnek az adatbázisban. A ma elterjedt keresők működésének lényege hogy a végiglátogatott oldalak dokumentumait egy adatbázisban valamilyen szempont szerint indexelik. Ebből az indextáblából keresnek a felhasználó által megadott kulcsszó / kérdés alapján. Ám az hogy maga ez az adatbázis, hogy épül fel, keresőnként változik.

A webkeresők működését több alapelvre is helyezték már. Egyes webkeresők arra törekszenek, hogy minél több weboldal tartalmát értelmezzék egy webet pásztázó program segítségével és a dokumentumból kielemezve annak jellemzőit tárolják egy központi helyen. Ezt az eljárást keresőrobotok más szóval spiderek, crawlerek végzik. A keresőrobot egy olyan program mely a webet pásztázza, releváns dokumentumok után kutatva ezeket letölti, rangsorolja, értékeli, és ez alapján tárolja el adatbázisában. Ez a módszer sokkal aktuálisabb képet tud adni a web tartalmáról, mert a robotok folyamatosan a végzik munkájukat. Ilyenek például a Google vagy az Altavista keresők. Magyar példaként a Heurékát és egy új magyar keresőt a Bluu-t említhetjük. Léteznek olyan keresők, amelyeknél a weboldal szerkesztői regisztrálják a weboldalukat egy bizonyos hierarchikus struktúrába sorolva, és, ezen adatok alapján csoportosítva, eltárolják a dokumentum jellemzőit. Egyes webkeresőknél az adatbázist manuálisan töltik fel és kézileg rangsorolják. Ezeket katalógus rendszereknek nevezzük, hiszen megpróbálják katalogizálni a weben tárolt anyagokat. 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. Ilyen a Yahoo! és a MiningCo webkereső is. Magyar példaként talán a Magyar Elektronikus Könyvtárat lehetne említeni, mint katalógus rendszer, ez azonban nem csak az adatbázist, hanem a vizsgált dokumentumokat is tartalmazza. Szintén magyar nyelvű fejlesztés a viszonylag új keletű Netkezdő.hu katalógusrendszer.

A webkeresők közös tulajdonsága mégis az, hogy egy lokális adatbázisból való lekérés során visszaadott hivatkozásokat adják vissza a felhasználónak. Ezek az adatbázisok indexelve vannak a webkereső 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. 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.

Felmerül annak a kérdése is, hogy a kulcsszavak megfelelőek-e, vagyis hűen tükrözik-e a weboldal tartalmát. További fontos kérdés az is, hogy ezek a kulcsszavak hűen tükrözik-e a felhasználó információs igényét. Ezeket a szavakat kétféleképpen lehet meghatározni: manuálisan és automatizálva.

A manuális adatbázis-feltöltést emberek végzik. Ezáltal pontosak, ám ez drága, nehezen fenntartható eljárás.

Az automatizált kulcsszó-keresés előnye, hogy könnyen fenntartható, olcsó, és gyors. Hátránya, hogy nem teljesen megbízható, hiszen könnyen félrevezethető. Az emberi leleményesség határtalan ilyen esetben. Vannak olyan weboldalak, ahol háttérszínnel írt szöveggel a gyakran keresett szavak tömegét írják, ezzel azt okozva, hogy egy kereső relevánsnak vélje a dokumentumot, és minél többen tévedjenek erre az oldalra.

Az automatizált kulcsszó meghatározást más néven tartalomszűrésnek is nevezik. A spider letölti a weboldalt és átadja az értelmezőnek, amely elvégzi a tartalomszűrést. Zipf hipotézise kimondja, hogy egy kulcsszó előfordulásának száma a dokumentum összes kulcsszójának számához képest (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. Hipotézisük kimondja, hogy a jellemző szavak egy bizonyos sávban találhatóak meg, amit a nagyon sűrűn és a nagyon ritkán előforduló szavak határolnak. Ennek megbízhatósága eléggé nyelvspecifikus.

A keresők a felhasználó által megadott kulcsszavak alapján keresnek az adatbázisban. Arra is gondolnunk kell, hogy a kulcsszavak kinyerése a dokumentumból nem elegendő ahhoz, hogy összehasonlítást tudjunk végezni. Az azonban biztos, hogy a szavak a mondatokban alapvetően nem szótári alakban szerepelnek, hanem valamilyen ragokkal, toldalékokkal kiegészítve. Ezektől tehát mindenképp meg kell szabadulnunk a kereső kifejezések értelmezéséhez. Erre megoldásként létezik egy angol nyelvre specifikus algoritmus, az úgynevezett Porter algoritmus. Ez az algoritmus stemming-el tehát szótői alakra hozza a szavakat, és így már könnyen felismerhetők az egyezések a keresett kifejezés és a vizsgált dokumentumok között.

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.

A keresők hasonlóképpen működnek, mégis más-más eredményeket adnak vissza. Ez betudható az adatbázis-feltöltés különböző módszereinek, de annak is, ahogy a felhasználó által megadott kulcsszavakat értelmezik.

A felhasználók különböző keresési módszereket alkalmazhatnak a webes keresők használata során ezek az egyszerű keresés, és az összetett keresés. Egy átlagos felhasználó alapvetően az egyszerű keresést alkalmazza. Ennek során visszakapjuk a keresett kulcsszót tartalmazó összes dokumentum listáját, esetleg ezeket kulcs előfordulási gyakoriság alapján rangsorolva. Az összetett keresés már feltételez valamiféle felhasználói tudást, hiszen itt boole kifejezések (AND, OR, NOT) segítségével tehetjük eredményesebbé a keresésünket. 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).

Az Altavista keresőnél például, 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. Míg például 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.

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

Magyar keresők:

- www.goliat.hu
- www.heureka.hu
- www.kurzor.hu
- keres.sztaki.hu
- www.ok.hu

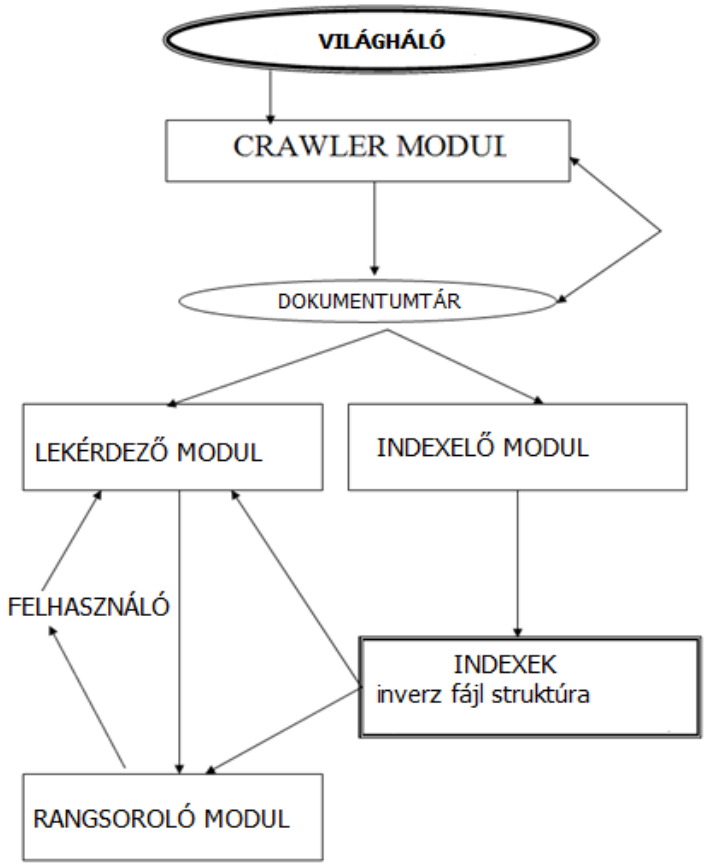
- www.startlapkereso.hu
- www.bluu.hu
- http://katalogus.netkezdo.hu/

Nemzetközi keresők:

- www.bing.com
- www.google.com
- www.yahoo.com
- www.search.msn.com
- www.allsearchengines.co.uk
- www.altavista.com
- www.excite.com
- www.gigablast.com
- www.exalead.com/search
- www.lycos.com

A fent felsorolt internetes keresőkön kívül a kereso.lap.hu rengeteg egyéb keresőt kínál kategóriákban rendezve. Országok szerinti bontásban, illetve a keresett tartalom szerint csoportosítva, mint például: kép keresők, könyv keresők, videó keresők, cég keresők, zenei keresők, FTP keresők, szoftver keresők .

Összefoglalásként tekintsük a webes információ-visszakereső rendszerek általános felépítését, melyet a következő ábra szemléltet.



5. ábra. Webes információ-visszakereső rendszerek felépítése

Crawler modul

A hagyományos információ-visszakereső rendszerben a dokumentumokat egy központi tárban tároljuk, számítógép diszkeken, általában egy adott intézménynél (egyetemi könyvtár, bank számítóközpontja). Ezzel szemben 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. Ezt a feladatot speciális programok végzik, amelyek együttesen az úgynevezett CRAWLER MODUL-t alkotják. Ezek a programok mindig, éjjel-nappal futnak. Virtuális robotok, úgynevezett pókok (spiders) pásztázzák a webet lapról lapra járva és letöltik az oldalakat a dokumentumtárba.

Dokumentumtár

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.

Indexelő modul

A dokumentumtárban található weblapokat az indexelő modul dolgozza fel (HTML címkék szűrése, index kifejezések azonosítása, stb.). Más szavakkal, a weblapok egy tömörített verzióját hozzuk 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:

- Tartalom struktúra: az oldalak szövegét, horgony szövegét stb. tárolja.

- Link struktúra: az oldalak közti kapcsolatokat tartalmazza (melyik oldal melyik másikra tartalmaz linket).

A pók a link struktúrát használja még felfedezetlen oldalak megtalálására.

Lekérdező modul

A 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.

Rangsoroló modul

A lekérdező modul által kiválasztott oldalak sorba rendezése 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. 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:

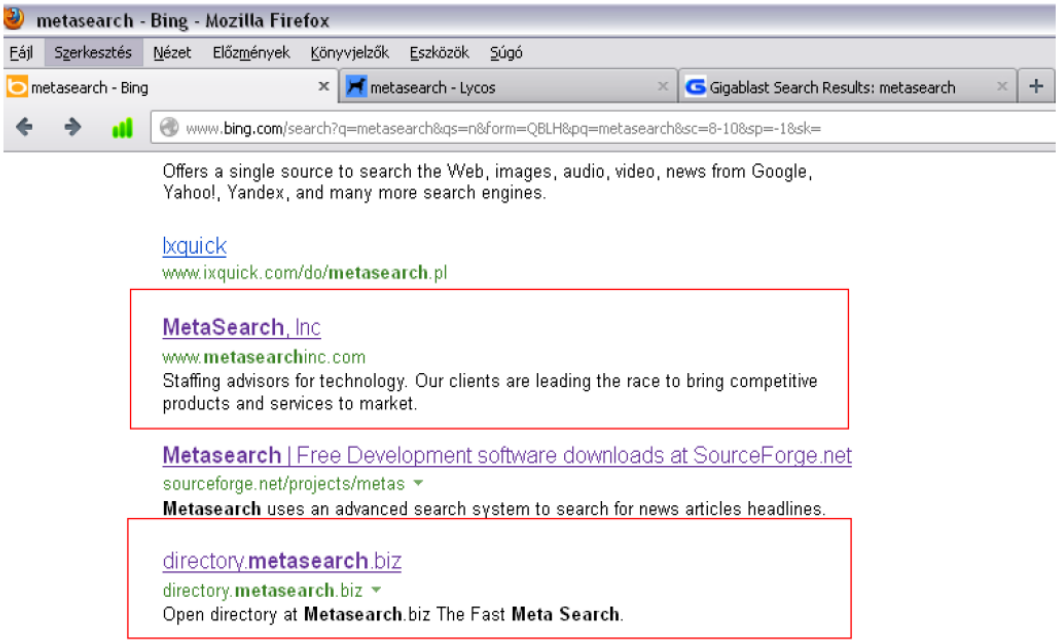
- Oldal tartalmának jellemzői (kifejezések előfordulási gyakorisága),
- Oldalon belüli jellemzők (a kifejezés elhelyezkedése az oldalon belül, kifejezés betűmérete),
- link információ (mely oldalak mutatnak az adott oldalra, az adott oldal mely oldalakra mutat).

5.2 Web metakereső motorok

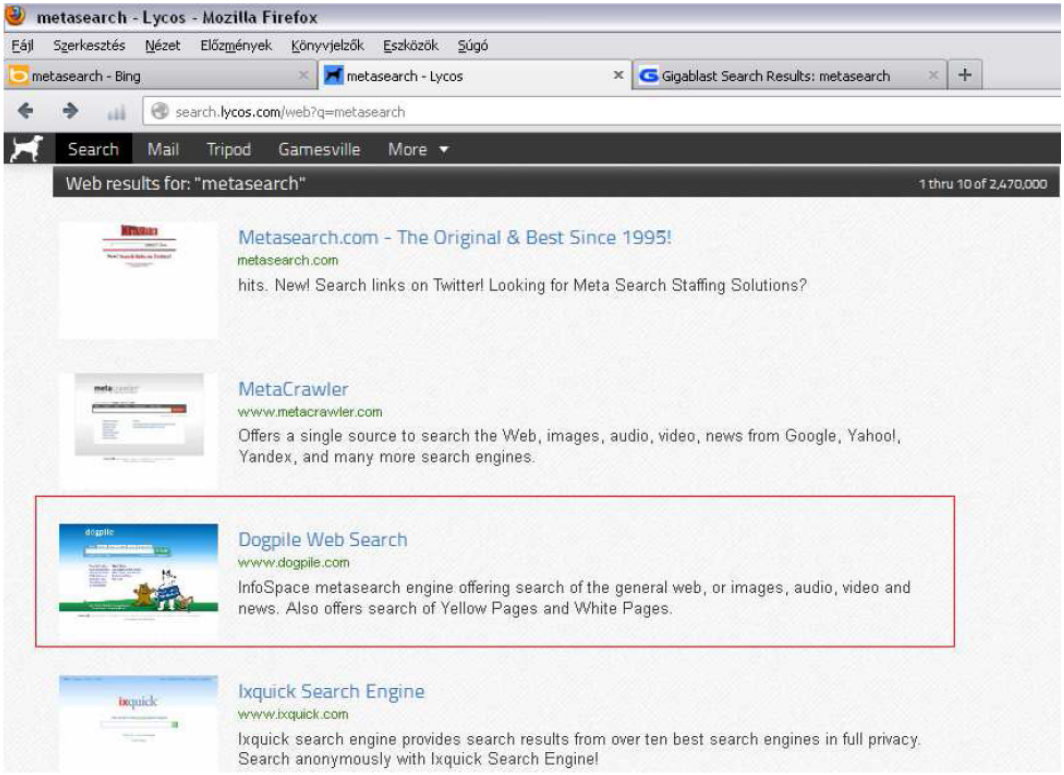
A weben történő kereséshez rengeteg gyors és jó kereső áll rendelkezésünkre, ám előfordulhat, hogy bizonyos esetekben még sem találják meg a felhasználó által keresett tartalmat. Abban az esetben, ha a számunkra hasznos információval tér vissza az éppen használt kereső, akkor is érdemes más keresőket is kipróbálni, hiszen a találati listájuk között általában igen kicsi az átfedés.

Példa

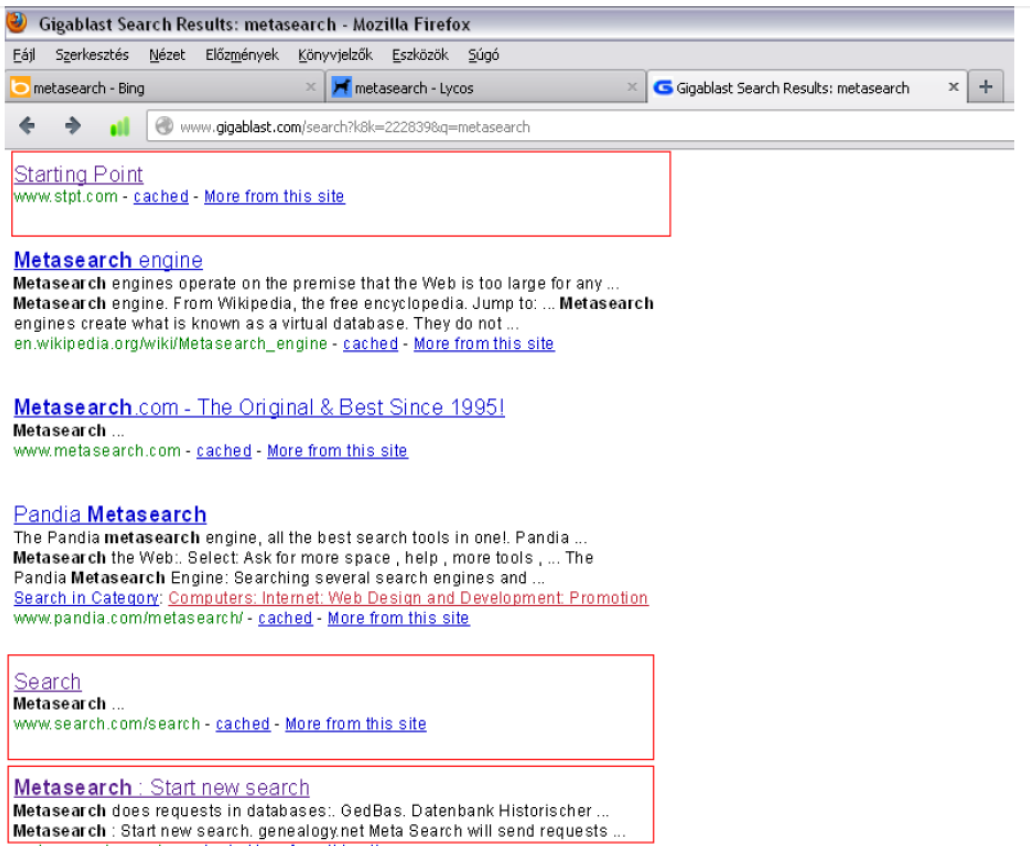
Vizsgáljuk a következő három keresőt: Bing, Lycos, Gigablast. Mindhárom keresőbe beírva a ’metasearch’ kereső kifejezést igen eltérő találati listát adnak, ahogy azt az ábrák is mutatják.



6. ábra. A Bing egyedi találatai



7. ábra. A Lycos egyedi találatai



8. ábra. Ábra: A Gigablast egyedi találatai

Az első eltérés rögtön a találatok számában mutatkozik meg, de nem ez a leglényegesebb, hiszen a felhasználók sosem nézik végig az összes találatot. Abban az esetben, ha az első 10 találatot vizsgáljuk, látszólag mindhárom keresővel jól járunk egyenként is, hiszen többségében metakereső oldalakat ajánlanak fel. Ám ha egyesével végignézzük a keresési eredményeket az egyezéseken kívül egyedi találatok is kapcsolódnak mindhárom keresőhöz. Példának okáért, ha releváns találatként csak a tényleges metakereső rendszerekre mutató linkeket tekintjük a Bing 2, a Lycos 1, a Gigablast pedig 3, egyedi (a másik kettőtől különböző) találattal rendelkezik csak az első 10 találata között. Éppen ezért célszerű több kereső párhuzamos használata. Ez azonban a nagyszámú kereső rendszer, illetve a találatok körülményes összehasonlítása miatt igen hosszadalmas feladat. Ennek megoldására találták ki a metakeresőket melyek más keresőknek és/vagy adatbázisoknak a találatai listája alapján állítják össze a saját keresési eredményeit. Ezzel megkönnyítik és egyben eredményesebbé is teszi a felhasználók munkáját.

A metakeresők olyan, webszervereken keresztül elérhető szoftverek, melyek egy adott kérdést elküldenek több webkeresőnek, webtárnak és adatbázisnak. Összegyűjtik, és valamilyen eljárással egyesítik az eredményeket. A metakereső legfőbb előnye, hogy több keresőt tudunk elérni egyetlen, egyszerű interfésszel. A fellelhető metakeresők egymástól annyiban különböznek, hogy mindegyikük más-más módszert alkalmaz a begyűjtött eredmények egyesítéséhez.

A metakeresők előnye nem csak abban rejlik, hogy egyszerre több webkereső eredményeit érhetjük el segítségükkel. Használatuk során sokkal relevánsabb találati listát kaphatunk, hiszen a találatokat különböző módszerek alapján rangsorolják és a redundánsakat is szűrik. Egy metakereső annál jobb minél több webes keresőt, adatbázist használ, ám a minősége nem kizárólag a mennyiségen múlik. A keresőktől visszakapott dokumentumok száma korlátos így egy metakeresőt használva nem feltétlenül kapunk vissza minden, a keresésnek megfelelő oldalt, de a visszakapott találatok a legnagyobb pontosságúak. Ezt biztosítja a talált dokumentumok tartalom alapján történő

rangsorolása is. A keresők általában már a keresett kulcsszavakban végrehatanak némi változtatást (a gyakran előforduló szavak például a névelők, kérdőszavak, kötőszavak, toldalékok eltávolítása) ezzel is pontosítva a keresést.

Ennek ellenére előfordulhat akár az az eset is, hogy a metakereső egyáltalán nem rangsorol. Ebben az esetben is előnnyel bír a webes keresőkkel szemben, hiszen ezek adatbázisa az internetnek különböző részeit fedi le, más-más adatokat tárol az egyes weboldalakról, ha ezeket összességében láthatjuk átfogóbb képet kapunk. Egy metakereső oldal általában igen áttekinthető, könnyen használható, sok esetben lehetőségünk van kiválasztani mely webes keresőket használja a keresés során illetve, hogy hány találatot jelenítsen meg az egyes keresőktől. A következő táblázatban öt ismertebb metakeresőt láthatunk, elérési címükkel.

Metakereső	Elérési cím
Dogpile	http://www.dogpile.com
Vivisimo	http://vivisimo.com/
Kartoo	http://www.kartoo.com
Mamma	http://www.mamma.com
Surfwax	http://www.surfwax.com

9. ábra. A Dogpile metakereső

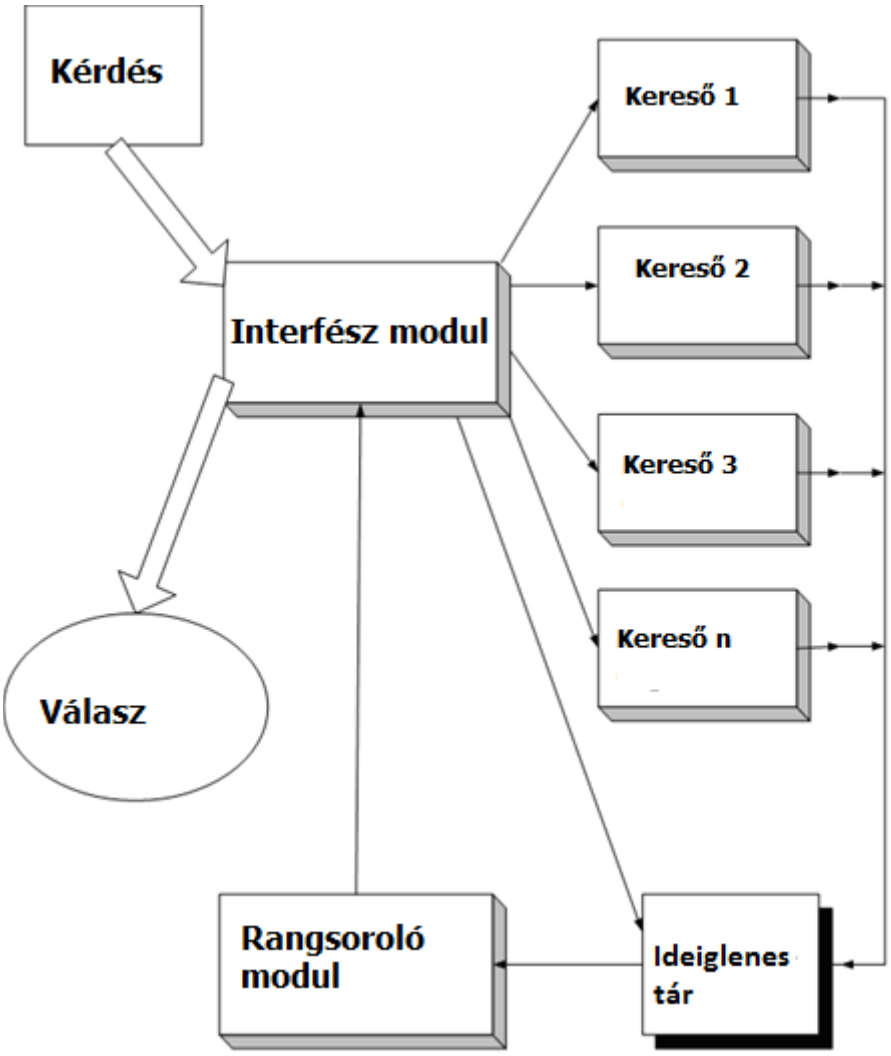
Léteznek olyan metakeresők, amelyek a kliens gépen futó applikációk, és a web szerverekről a kliens hostra töltik le az eredményeket. Ilyen például a Copernic, EchoSearch, WebFerret, WebCompass és a WebSeeker. Míg más metakeresők a forrás alapján külön ablakokban jelenítik meg az eredményeket, összevonás nélkül. Ilyenek a All4One, a OneSeek, a Proteus és a Search Spaniel rendszerek.

A metakeresők előnye továbbá, hogy különböző szempontok szerint rangsorolhatjuk, rendezhetjük a tulajdonságaik alapján a weboldalakat, mint például host, kulcsszó, dátum stb. Így sokkal információdúsabb a keresés egy egyszerű keresőhöz képest. Gyakran egyszerűbb és kezelhetőbb a

felület. Továbbá a metakeresők nem feltétlenül jelenítenek meg minden oldalt, amely megfelel a keresésnek, mivel a keresők által visszaadott eredmények száma korlátos. A felhasználó természetesen a keresőket használva ezeket meg tudja adni, de akkor is véges számú eredményt ad vissza. A metakereső célja az, hogy az általa visszaadott weboldalak a lehető legnagyobb pontosságúak legyenek.

Vannak metakeresők, melyek minden visszaadott címről letöltik az eredményt és a tartalom alapján rangsorolják azokat. Ez természetesen időigényesebb megoldás, ám, megfelelő megjelenítési technikával, nem lép túl az emberi türelem időkorlátján. Ezzel a technikával eleve kiszűrődnek azok a weboldalak, melyek nem elérhetőek, és ami fontosabb, hogy egy jobb rangsorolást kapunk a tartalomszűrés segítségével. A hátránya ennek, hogy sok hálózati erőforrást igényel, ezáltal a rendszer lassul a hálózati kapcsolat sebességének a függvényében.

A fentiek összefoglalásaként tekintsük a webes metakereső rendszerek általános felépítését, melyet a következő ábra szemléltet.



10. ábra. Ábra Webes metakerső általános felépítése

A metakereső beolvassa a felhasználó kérdését. A kérdést elküldi további keresőknek. A keresők által megtalált oldalak közül néhányat letölt az ideiglenes tárba. Ezek az oldalak lesznek a kérdésre a válaszok. A válasz oldalakból a metakereső létrehozza saját találati listáját.

Interfész modul

Feladata a felhasználó kérdésének beolvasása. A kérdés kifejezések egy halmaza. A modul a kérdést szótövesíti és elküldi kereskedelmi webes keresőknek API használatával.

Metakereső

Minden egyes kereső találati listájának első *n* eleme kerül letöltésre, párhuzamosan. Minden weboldal átesik a következő feldolgozáson: címkék eltávolítása, kulcsszavak azonosítása, stoplistázás, szótövesítés. Az eredmény a feldogozott oldalak dokumentumtára, amelyet a továbbiakban a rangsoroló modul dolgoz fel.

Rangsoroló modul

Ez a modul online működik. Feladata az eredmények rangsorolása valamely előre definiált értékelési módszer szerint.

5.3 A keresőkkel szemben felmerülő felhasználói igények

A napjaink keresői között válogatva a felhasználó szívesen használja a külalakra esztétikus megjelenésű weboldalakat, esztétikus alatt itt nem csupán a weboldal designja jut kifejezésre, hanem annak praktikussága is. Fontos, hogy az eredmények miként jelennek meg a felhasználó előtt. Például ugyanazon host eredményei közül a gyökér URL-hez legközelebb álló jelenik meg először és utána sorbehúzással azok a weboldalak, amelyek ebből a könyvtárból nyílnak. Így meg tudja a felhasználó különböztetni, hogy valóban egy új oldalt talált a kereső, vagy csak egy az előzőhöz szorosan kapcsolódó HTML dokumentumot. A megjelenítést kezelhetővé teszi, ha az URL-t nem csak egyszerű szövegként kiírva, hanem a hypertext lehetőségeit kihasználva hivatkozásként is láthatjuk, hogy lehetőségünk adódjon a címre kattintva rögtön letölteni a keresés eredményeként visszaadott weboldalt.

Egy keresőnél fontos szempont a minél egyszerűbb használat, a minél kielégítőbb és információban dús válasz. A felhasználó elvár egy kivonatot a weboldalból, amelyet a kereső felajánl neki, ugyanis egy URL cím keveset árul el a weboldal tartalmáról. Ebben a kivonatban a keresett kulcsszavak félkövér szedetben jelennek meg, hogy a felhasználó lássa, hogy a kulcsszót körülbelül milyen szövegkörnyezetben lelte meg a kereső. A kiemelés eredményeképpen sokkal áttekinthetőbbek az eredmények. A mai keresőket gyakran ellátják ún. haladóknak (advanced) való kereséssel. A felhasználók nagy része mindennapi alkalmazásban csak egyszerű kulcsszó szerinti keresést igényel. Gyakran előfordul, hogy a felhasználó a host, esetleg a HTML oldal utolsó frissítése vagy Boole-kifejezésű kérdés szerint szeretné sorrendbe tenni vagy szűrni az eredményeket, ilyenkor nyújt neki segítséget a haladóknak szánt keresés. Itt a felhasználó szűrési feltételeket tud megadni, ami alapján a keresést végrehajtja a szoftver.

A felhasználó gyakran nincs tisztában a kereső működési elvével. Ennek érdekében egy tájékoztatóra mutató linket célszerű elhelyezni a weboldalon. Ezáltal egy rövid leírást kap a használatról. A felhasználóknak meg kell tanulniuk kérdéseket feltenni. A kérdés feltevésének módja erősen befolyásolja a visszaadott eredmények minőségét.

Egyes keresők kis és nagybetűre érzékenyek. Például, ha nagybetűvel írunk egy szót, akkor azt pontosan olyan formában fogja keresni a kereső. Más esetekre vonatkozóan pedig kimutatták, hogy kis- és nagybetű érzékeny keresőknél már a szó első betűjének megváltoztatása jelentősen befolyásolja a szó értelmét. Gondolhatunk itt arra is, hogy a szó különböző alakjai miatt akár a 10-20%-át is elveszthetjük a releváns dokumentumoknak. A szavak félregépelve (például idegen nyelvű nevek stb.), akár 50%-nyi releváns válasz elvesztését is okozhatja.

A felhasználó a feltett kérdéseiben más értelemben használja az és (AND) és a vagy (OR) kulcsszavakat. A leggyakrabban előforduló hiba a kérdés feltevésében vagy természetes nyelvben használt szerepe szerint a Boole-logikának megfelelő kizáró vagy felel meg. Amennyiben a felhasználó nem ismeri a Boole-logikát, előfordulhat, hogy nem arra kap választ, amire szeretne.

Éppen ezért, mint ahogy ki is mutatták, a keresések 80%-ában nem szerepel az ÉS és a VAGY, ami segítené a keresőt leszűkíteni a keresést, ezáltal lehetővé tenné, hogy sokkal pontosabb eredményekhez jusson a felhasználó a kérdésével kapcsolatban.

A weboldal esztétikus megjelenése is fontos a mai felhasználó szemében, hiszen egy kellemes környezetben sokkal jobban érzi magát az ember. A weboldal színeinek megválasztásában fontos szerepe van az olvashatóságnak. A mai webes társadalom időközben kialakult igénye a szép weboldalakra egyre nagyobb méreteket ölt, a webszerkesztők pedig egyre szebb és érdekesebb megoldásokat találnak a HTML oldalak különlegessé tételéhez, mely rengeteg felhasználót vonz.

5.4 A Google kereső használata

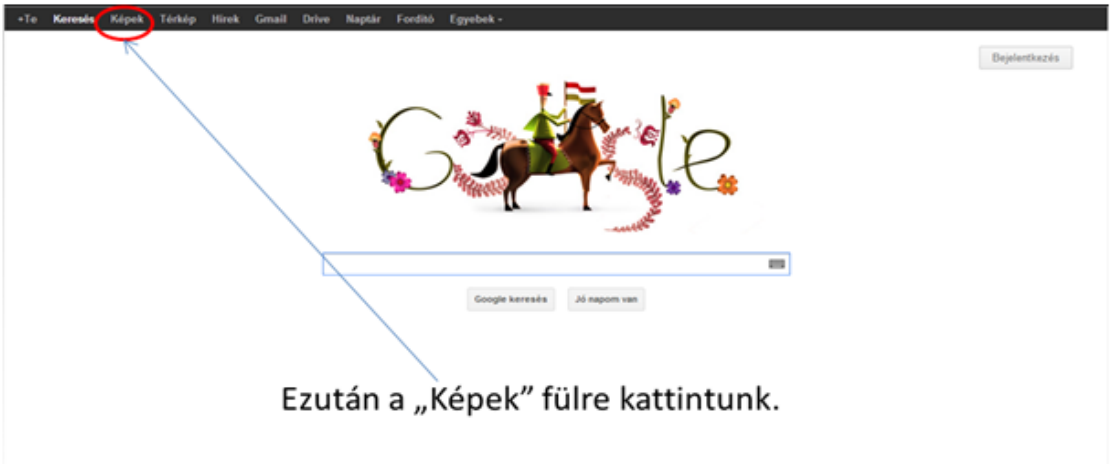
A Google a világ legnépszerűbb keresője. Naponta több mint kétszázmillió felhasználói kérést válaszol meg. A hagyományos kulcsszó alapú keresés mellett lehetőséget nyújt képek, hírek, speciális fájlformátumok keresésére. Kapcsolt alkalmazásait tekintve használhatjuk térképként, útvonaltervezőként, mértékegység átváltóként, számológépként és folytathatnánk a felsorolást számos további alkalmazás megemlítésével. A Google kereső néhány speciális keresési funkciója a következő néhány példán keresztül kerül bemutatásra.

5.4.1 Kép keresése

Azt a feladatot kapjuk, hogy tudjunk meg mindent erről az állatról, de csak ezt a képet kapjuk. Mit tegyünk?

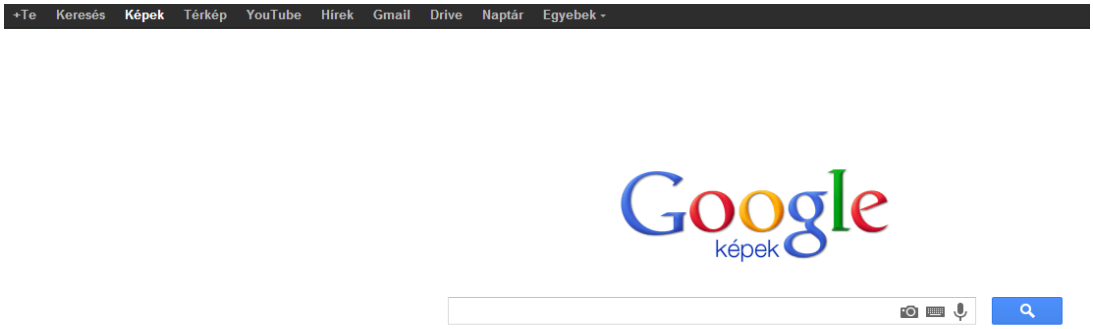
11. ábra. Mit ábrázol a kép?

Első lépésben nyissuk meg a böngészőt és írjuk be a www.google.hu címet. Azután kattintsunk a képek fülre, ahogy az ábra mutatja.



12. ábra. Képkereső fül

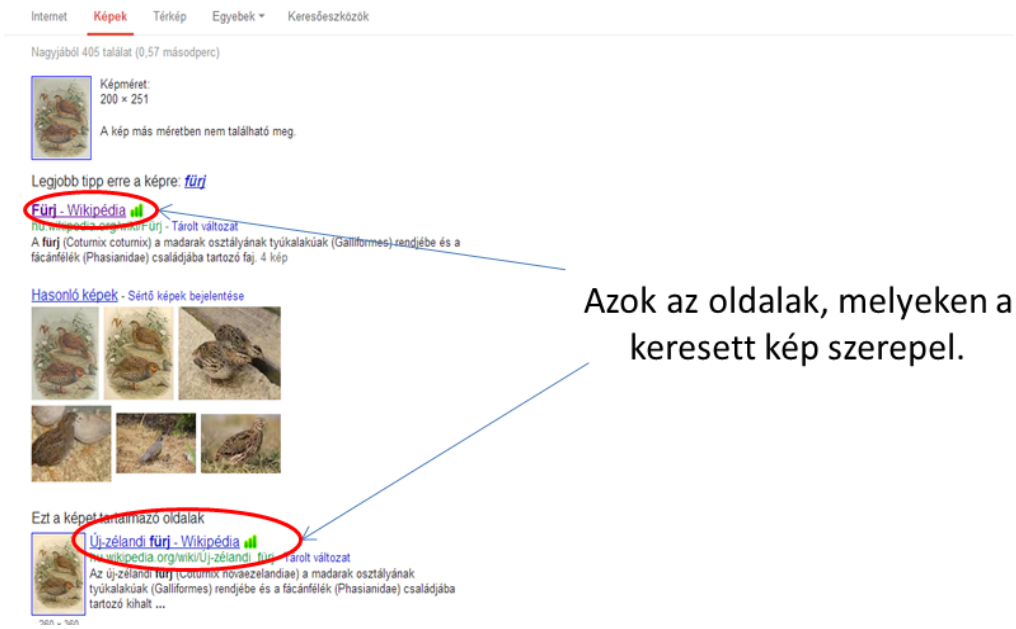
Ekkor a következő oldalt kapjuk. A keresőbe egyszerűen csak húzzuk be a keresendő képet a „fogd és vidd” módszerrel.



13. ábra. Képkereső

14. ábra. Keresés kép alapján

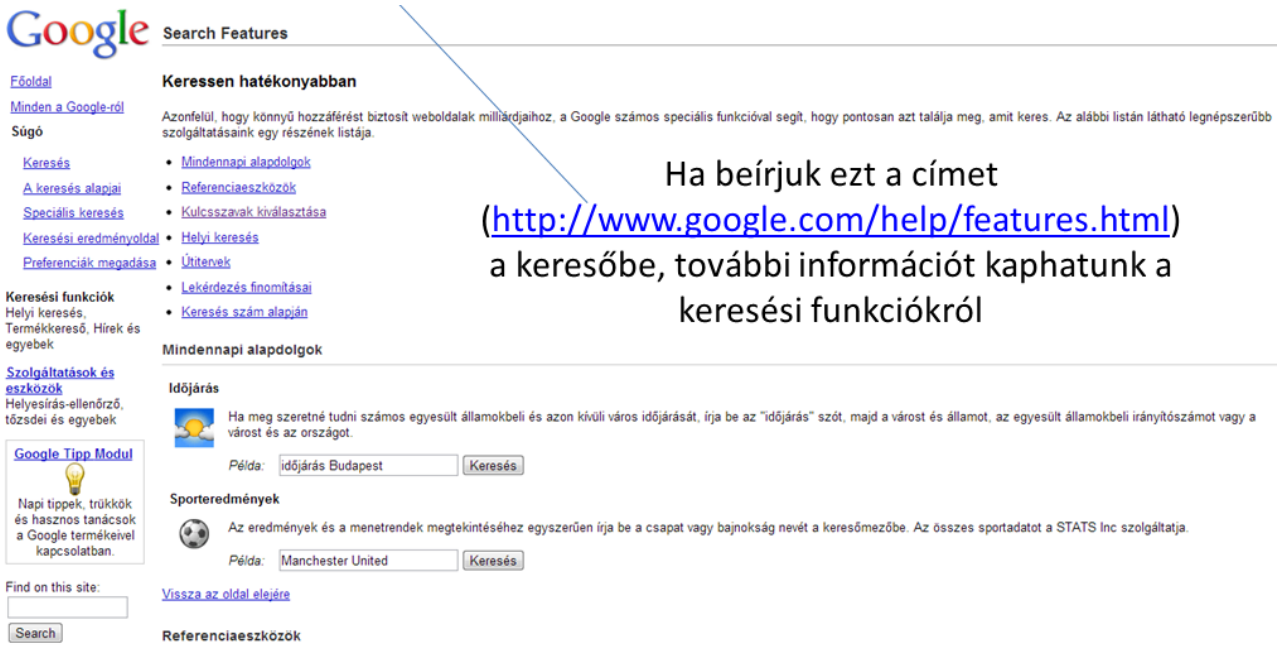
Néhány másodperc várakozás után a Google megkeresi azokat az oldalakat, melyeken a keresett kép szerepel. Ezzel a módszerrel szinte minden képről megtudhatunk minden információt a Google segítségével.



15. ábra. Találatok

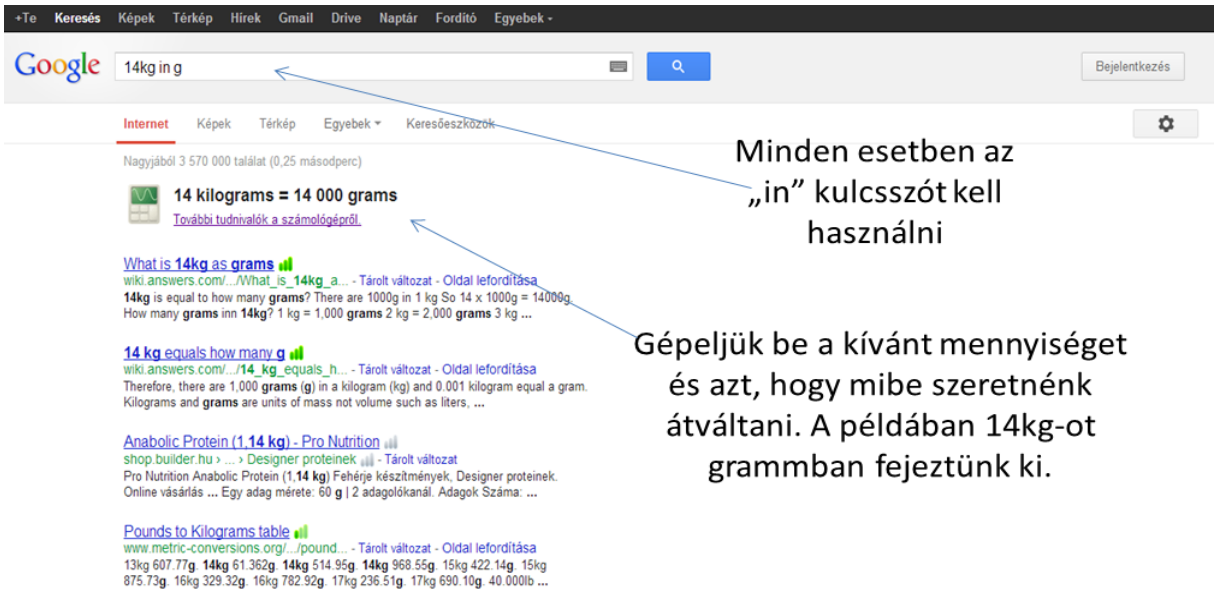
5.4.2 Speciális keresési funkciók

A Google speciális keresési funkcióiról a www.google.com/help/features.html oldalon kaphatunk részletes tájékoztatást.

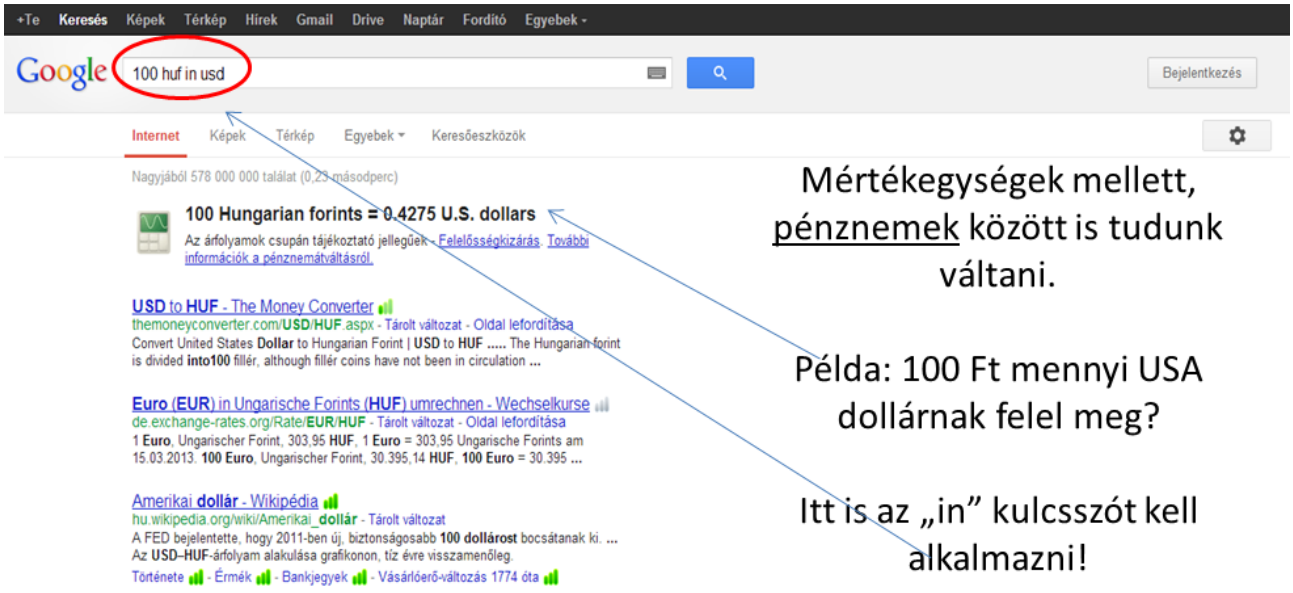


16. ábra. Ábra Google speciális keresési funkciók

Egy nagyon hasznos keresési funkció a konvertálás és számolás. Ennek az alkalmazásnak a segítségével pillanatok alatt megoldható a különböző mértékegységek és pénznemek közötti átváltás. A kulcsszó minden esetben az *IN*.

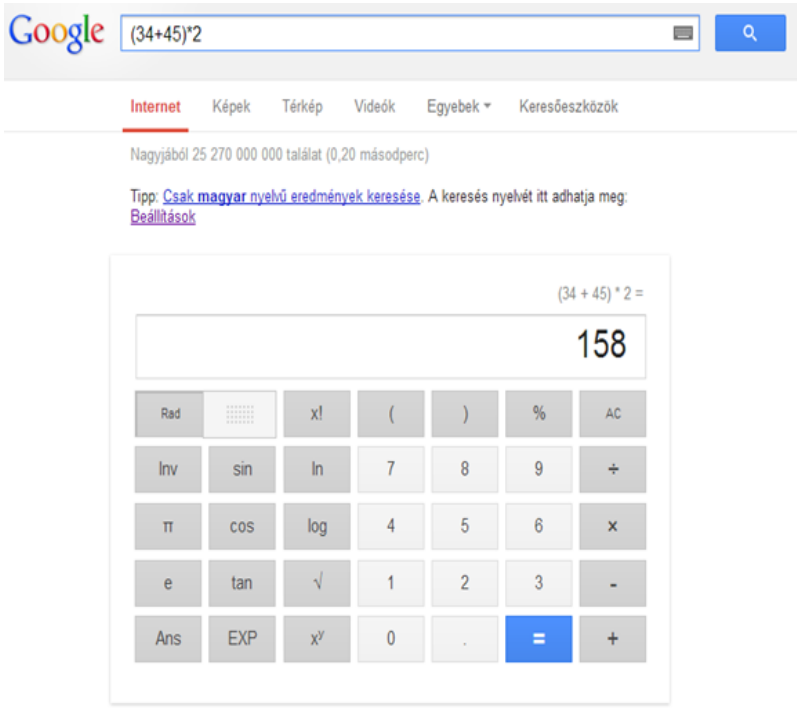


17. ábra. Mértékegység átváltás a Google segítségével



18. ábra. Pénznemek közti átváltás a Google segítségével

A keresővel számítási feladatok is végezhetünk. Csak be kell írni a keresőbe a kívánt műveleteket és a Google kiszámítja nekünk a végeredményt.



19. ábra. Google számológép

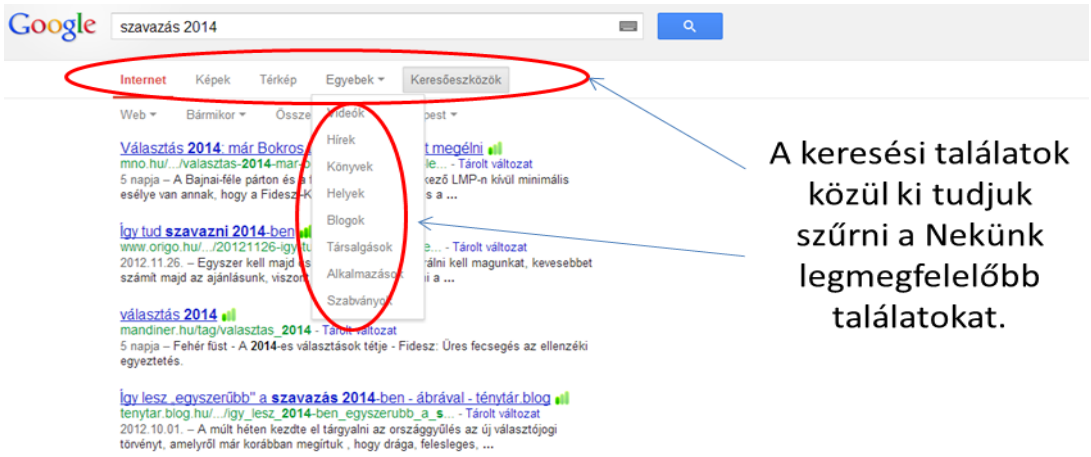
Matematikai műveletek mellett függvények ábrázolására is használhatjuk a keresőt. Egyszerűen beírjuk a keresőmezőbe a függvényt, mint például sin(x)+cos(x^2) és a Google kirajzolja a grafikont.

20. ábra. Google, mint függvényábrázoló

5.4.3 Találatok szűrése

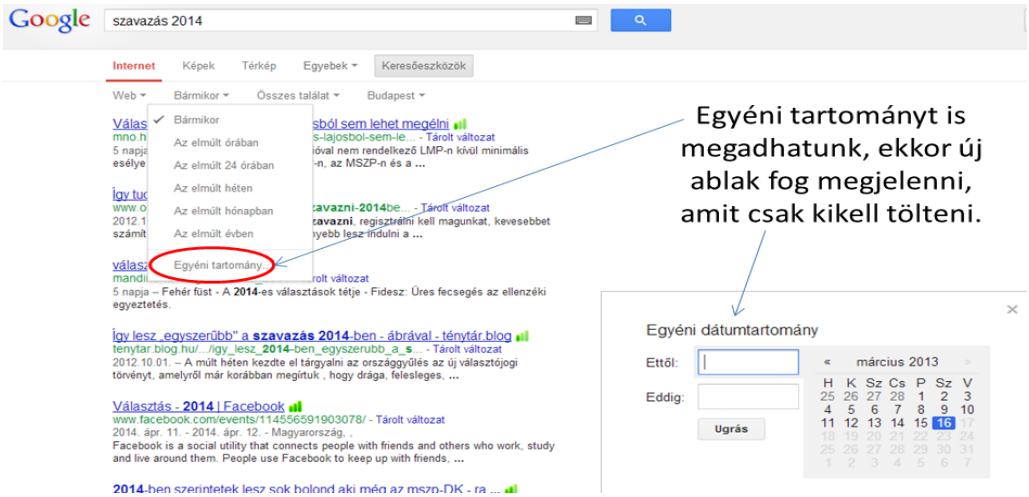
A Goggle számos keresőeszközt és operátort biztosít, amelyek segítségével egy adott keresőkifejezésre kapott találati listát tovább szűkíthetünk, még egyedibb, személyre szabott

találatok megjelenítésére. Egyik lehetőség a tartalom típusára történő szűrés. Ebben a menüben megkülönböztetünk például videó, hír, blog stb. kategóriákat.



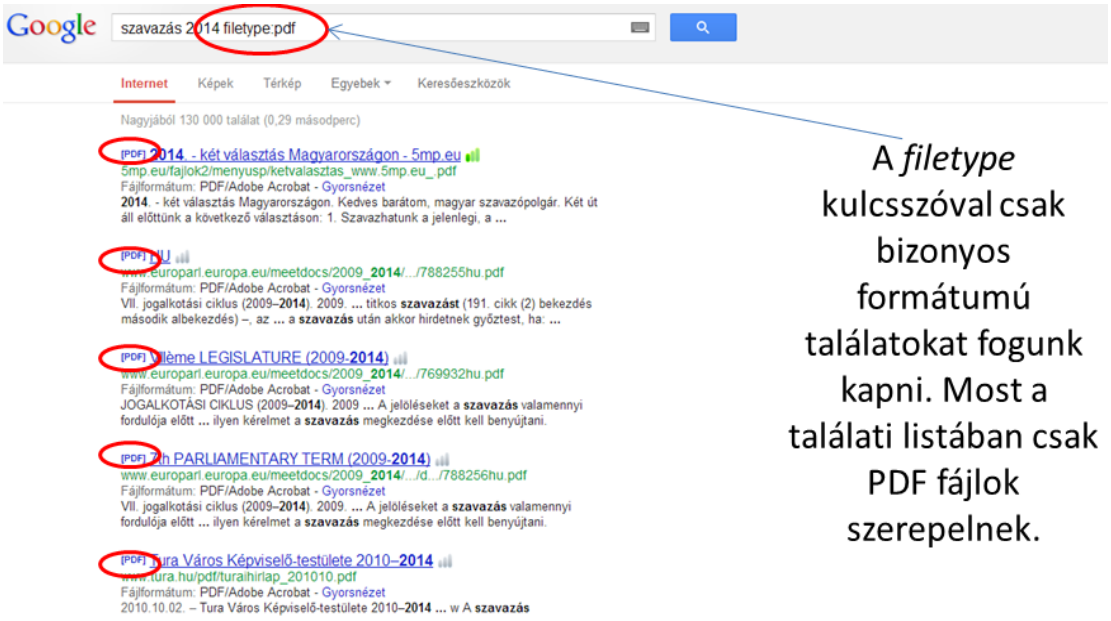
21. ábra.Kategória szűrés

A *Bármikor* fül segítségével időlimitet tudunk beállítani. Például beállíthatjuk, hogy csak azokat a találatokat jelenítse meg a kereső, amiket az elmúlt héten töltöttek fel az internetre. Természetesen *Egyéni tartományt* is megadhatunk, ekkor egy új ablak fog megjelenni, amit csak ki kell töltenünk.



22. ábra. Időlimit beállítása

Végül, de nem utolsósorban tekintsük az állomány típus szerinti szűrést. A *filetype* operátor segítségével csak a megadott formátumú állományok jelennek meg a találati listában.



23. ábra. Állomány típus szűrés

6. fejezet

Webes keresők és metakeresők relevanciahatékonyságának mérése

6.1 Webes visszakeresés relevanciahatékonyságának mérése

Az információ-visszakereső módszereket és rendszereket, mint bármely szoftverrendszert, minősíteni szokás. Az információ-visszakereső rendszereknél elsősorban azt vizsgáljuk, hogy milyen precíz a találati lista, azaz mennyire relevánsak a válaszok. A találati lista a keresőkérdésre a kereső által visszaadott linkeket (oldalakat) tartalmazza. Ezt a vizsgálati módszert visszakeresési teljesítmény kiértékelésnek (becslésnek) hívjuk.

A vizsgálat első lépéseként egy elfogulatlan kérdéssort kell felállítani a rendszer teszteléséhez. A tesztkérdések kiválasztása két lépésből áll. Elsőként ki kell jelölni a témát, amelyet keresni kívánunk, majd pontosan meg kell határozni a keresőkérdéseket. Az elfogult értékelés elkerülésére általános és speciális kérdéseket is választani kell.

A tesztelés végrehajtása előtt kell meghatározni, hogy a kérdésekre kapott válaszokat hogyan fogjuk minősíteni. Fontos, hogy a válaszok megismerése előtt alakítsuk ki az értékelési módszert. Ellenkező esetben elfogultabban, szubjektívebben értékeljük a keresőt, hiszen ismerjük erős és gyenge pontjait. Megtudjuk, melyik kérdésre ad több választ, mennyire pontos válaszokat ad, stb.

Az információ-visszakereső rendszereket nem minősíthetjük az alapján, hogy egy kérdésre hány választ adnak vissza. A válaszok közt általában inaktív és a felhasználók számára felhasználhatatlan oldalak is vannak. A felhasználók által megfogalmazott információ igény sokszor határozatlan, így a visszakeresett dokumentumok nem lehetnek pontos válaszok. Másrészt különböző felhasználóknak más a megítélésük, hogy melyik válasz releváns és melyik nem. A kiértékelés előtt meg kell határozni, hogy a keresőkérdésekere kapott válaszokat mikor tekintjük relevánsnak. A relevancia két entitáshoz kapcsolódik: az egyik az információforrás valamilyen formája, a másik az információigény egy formája. Az információforrás magát a dokumentumot, vagy az azt helyettesítő tartalmi összegzést jelenti. Az információigény magában foglalja a felhasználó szavakkal megfogalmazott igényét, és az információ–visszakereső rendszerrel közölt keresőkérdést. A találati oldalakat kategóriákba soroljuk. A kategóriák meghatározzák, hogy egy választ mikor tekintek relevánsnak. A kategóriák kialakítása a következő módon történik.

- 1. kategória: Releváns oldalak (linkek). Nehéz eldönteni, hogy melyik oldalt tekintsük relevánsnak, hiszen a különböző felhasználóknak más a megítélésük, hogy melyik válasz releváns és melyik nem. Általában akkor tekintjük relevánsnak a találatot, ha
    - az oldal címe tartalmazza a keresett kifejezést,
    - az oldalon szerepel a keresett kifejezés vagy
    - kérdés által meghatározott tématerülethez tartozik az oldal.

- 0. kategória: Az oldal irreleváns, semmilyen szempontból nem tesz eleget a kereső kérdésnek, vagyis nem sorolható az 1.kategóriába.
- dupla link: A keresőkérdésre visszakapott találati listában többször is előfordul ugyanaz az oldal.

- inaktív link: Egy linkre kattintva különböző hibaüzeneteket kaphatunk: File not found, Forbidden, Server not responding.

A kategóriák által az oldalak besorolása egyértelmű, minden egyes oldal vagy benne van egy kategóriában, vagy nincs benne. Természetesen több relevancia kategóriát választhatunk. A releváns oldalakat különböző szempontok szerint további kategóriákba sorolhatjuk.

A kategóriák meghatározása után el kell dönteni, hogy a kereső által visszaadott találati listából hány dokumentumot soroljunk be a kategóriákba. Ez is fontos kérdés, hiszen Webes keresőknél a visszakapott találatok száma tízezres nagyságrendű is lehet. Gyakorlatilag lehetetlen több ezer linket elolvasni, és megvizsgálni, hogy releváns-e a kérdésre vagy sem. A létező tanulmányok általában az első 10 vagy 20 választ veszik figyelembe, ezzel is az átlagos felhasználót utánozva, aki csak az első, vagy az első két oldal találatait nézi meg.

Amint azt már láttuk, visszakereső módszer relevancia-hatékonyságának laboratóriumi mérése alapmértékeken (pontosság, teljesség) nyugszik. A világhálón élesben működő webes visszakereső rendszer relevanciahatékonyság-mérésének a következő sajátosságai vannak a laboratóriumi méréshez viszonyítva, mégpedig:

- Magának a Világhálónak a jellemzői:
  - a Web-oldalak sokféle formátuma (HTML stb.),
  - a Web-oldalak „laza” szerkezete (nem olyan értelemben precízen definiáltak, mint pl. egy relációs adatbázis esetében a táblázatok),
  - a Web-oldalak dinamikusak (megjelennek, eltűnnek, módosulnak),
  - a Web-oldalak hálózatot alkothatnak (linkek révén, Web-gráf),
  - a Web-felhasználók száma óriási, a felhasználók különböznek egymástól beszélt nyelv, érdeklődési terület stb. szempontjából,
  - a Web-oldalak száma exponenciálisan növekszik.

- Élesben működő visszakereső rendszer esetében a keresőkérdés előre nem ismert, nem ismerjük a web tartalmát, ezért nem tudjuk, hogy egy kérdésre hány releváns válasz van ezért a felidézés nem számítható ki, relevancialista nem adható meg. Ha nagyon sok választ kapunk a kérdésre, akkor a pontosságot sem tudjuk meghatározni. Gyakorlatilag lehetetlen az összes linket elolvasni, és megvizsgálni, hogy melyik releváns és melyik nem.

6.2 A Leighton-módszer

Web-es keresők relevancia hatékonyságának kiértékelésére nagyon jól használható a Leighton-módszer [20.]. A módszer a mérőszám meghatározásához csak az első 5 (vagy 10) $\frac{9}{8}$ a kísérlet szempontjait figyelembe véve tetszőlegesen megválasztható számú $\frac{9}{8}$ találatot veszi figyelembe.

Leighton módszere olyan mérőszámot használ, amely a pontosságot méri a rendezési hatékonysággal súlyozva. A két mérőszámot egy értékben egyesíti. A mérőszám értéke egy 0 és 1 közé eső szám. 0 a mérőszám értéke, ha nem kapunk választ a kérdésre, vagy az összes válasz irreleváns. 1 a mérőszám értéke, ha az összes válasz releváns, és megfelelő számú választ kaptunk vissza. Ennek a módszernek a következő fontos tulajdonságai vannak:

- Minden kérdés esetén megvizsgáljuk a kereső által visszaadott első 5 (vagy 10) választ, és a vizsgált kritériumnak megfelelően a fent említett kategóriákba soroljuk őket.
- A relevanciát bináris skálán mérjük: ha a linket az 1. kategóriába soroltuk, akkor 1 pontot, ha a 0. kategóriába, akkor 0 pontot ér a válasz. A dupla linkeket 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 linket. (Ez persze nem jelentheti azt, hogy csak nagyon kevés találatot adjon vissza egy jó kereső, netán egyet se.)

A rangsor figyelembe vételére a pontosság kiszámításakor súlytényezőket használunk. A visszakapott találatokat csoportokba soroljuk. A csoportokhoz súlyszámokat rendelünk. Az azonos csoportban található linkek azonos súlyozással szerepelnek a mérőszám értékének kiszámításakor. A csoportok kialakítása és a súlytényezők megválasztása természetesen többféleképpen történhet.

Csoportok megadása

Az 5-pontosság esetében az első öt találatot osztjuk fel 2 csoportra a következő módon:

- 1. csoport: az első 2 találat. .
- 2. csoport: a következő 3 találat.

Az első két találat azért került az első csoportba, mert általában ezek vannak a kezdőképernyőn a találatok megjelenítésekor.

A 10-pontosság esetében az első tíz találatot osztjuk fel 3 csoportra a következő módon:

- 1. csoport: az első 2 találat. .
- 2. csoport: a következő 3 találat.
- 3. csoport: a következő 5 találat.

A felhasználó számára fontos, hogy a találati lista elején legyenek a releváns találatok. Feltételezhetjük, hogy a felhasználó számára általában az első 5 link a legfontosabb, közülük kiemelten fontos az első kettő.

Súlyszámok megadása

Az ugyanabban a csoportban lévő linkek ugyanolyan súlyozást kapnak.

Az 5-pontosság esetében a következő módon súlyozunk:

- Az első csoport, vagyis az első 2 link súlya: 10.
- A második csoport, vagyis a következő 3 link súlya: 5.

A súlyszámok alapján elégedettebbek vagyunk, ha rögtön a kezdő képernyőn megkapjuk a releváns linkeket. Sok felhasználó nem vizsgálja meg a további találati oldalakat.

A 10-pontosság esetében a következő módon súlyozunk:

- Az első csoport, vagyis az első 2 link súlya: 20.
- A második csoport, vagyis a következő 3 link súlya: 17.
- A harmadik csoport, vagyis a további 5 link súlya 10.

A kitartó felhasználó ugyan több linket megvizsgál, de számára is fontos, hogy a találati lista elején legyenek a releváns linkek. Azt feltételezzük, hogy a kitartó felhasználó számára az első 5 link a legfontosabb. Közülük is kiemelten fontosak az első találati oldalon található linkek. Ezt a szemléletet tükrözi a csoportok kialakítása, és a súlyszámok meghatározása.

Mértékek meghatározása

A csoportok és súlyszámok ismeretében határozzuk meg minden egyes kérdés esetén a mérőszámok értékét. Az 5-pontosság mértékét a következő módon számítjuk ki. A mérőszám egy tört, ahol a számláló az első öt válasz súlyozott összege.

Példa

Ha egy *K* kérdés esetén az első 3 választ volt releváns, a számláló: (2 x 10) + (1 x 5) = 25.

Az első két válasz az 1. csoportba tartozik, így a hozzájuk rendelt súlytényező 10. A harmadik válasz a 2. csoportba tartozik, így a hozzá rendelt súlytényező 5. Ne felejtsük el, hogy csak a releváns linkek kapnak 1 pontot. Az irreleváns linkek 0 pontot kaptak (nem számítanak bele a számlálóba).

A nevező a visszaadott linkek (csak az első 5 figyelembe vételével!) súlyozásával számítható ki az előző módszer szerint. A nevező kiszámításakor azonban nem vesszük figyelembe, hogy a válasz releváns-e vagy sem. Azt vesszük figyelembe, hogy hány választ kaptunk.

Példa

Ha 5 vagy több találatot kapunk, akkor (2 x 10) + (3 x 5) = 35 lesz a nevező.

A nevező kiszámításakor több probléma is felmerülhet:

- Mi történjék, ha kevesebb, mint 5 találat van?
- Ha nincs találat, akkor 0 a nevező? (Bár ekkor már a számláló is az lenne, ami még nem baj, de máris feleslegessé tenné a további számolást.)
- Ha nem büntetjük valamilyen módon a hiányzó találatokat, az azt jelenti, hogy inkább egy vagy egy választ se szeretnénk kapni?

A következő lehetőséget érdemes választani: ha 5 találat van, a nevező legyen 35, egyéb esetekben pedig minden hiányzó találatért (ami az 5-ből hiányzik) vonjunk le a nevezőből 5 pontot, bármelyik találat hiányzik is. Így csak akkor büntetjük a keresőt, ha a számunkra fontosabbnak ítélt első két találatból hiányzik valamelyik. Már két válasz esetén (ha az első kettő válasz) megfelelőnek ítéljük a válaszok számát. Természetesen az irreleváns linket is büntetjük. Dönteni kell még a többszörös linkek kezeléséről is. A mérőszám meghatározásánál ha büntetjük a dupla linkeket, akkor a számlálóból kell levonni őket. Úgy tekintjük, mintha a válasz irreleváns lett volna.

Példa

A *K* kérdés esetén 3 találatot kaptunk, vagyis: 35 – (2 x 5) = 25. lesz a nevező.

A *K2* kérdésre öt releváns választ kaptuk. A harmadik és az ötödik válasz azonban azonos volt (dupla link). Ebben az esetben:

- A mérőszám értéke: ((2 x 10) + (2 x 5)) / (35 – 1 x 5) = 1. ha a dupla linket nem büntetjük.
- A mérőszám értéke: ((2 x 10) + (2 x 5)) / 35 = 0,857. ha dupla linket büntetjük

A mérőszámok különbsége megmutatja, hogy a dupla linkek előfordulása mennyire lerontja a kereső teljesítményét.

Végül a mérőszámot megkapjuk, ha a számlálót elosztjuk a nevezővel. Vagyis az 5-pontosság mérőszáma:

$$P_5 = \frac{releváns-link-száma_{1.-2.link} \times 10 + releváns-link-száma_{3.-5.link} \times 5}{(35-((5-linkek-száma_{1.-5.link}) \times 5))}$$

A 10-pontosság mértékét hasonló módon, a következő képlettel számítjuk ki:

$$P_{10} = \frac{releváns-link-száma_{1.-2.link} \times 20 + releváns-link-száma_{3.-5.link} \times 17 + releváns-link-száma_{6.-10.link} \times 10}{(141-((10-linkek-száma_{1.-10.link}) \times 10))}$$

Web-es keresők relevancia hatékonyságának kiértékelésére az 5 (vagy 10) pontosság mérőszámát kérdések egy csoportjára kiszámoljuk és a kapott értékeket átlagoljuk.

6.3 A relatív pontosság mérésének RP módszere

Az információ-visszakeresés alapvető mértékegysége a pontosság. A pontosság megadja a találati listában a releváns válaszok hányadát.

$$Pontosság = \frac{|Ra|}{|A|}$$

Ahol:

- 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.

A webes metakereső gépek más webes keresőgépek találati listáit használják fel saját találati listájuk előállítására. Ha egy találati lista jósága, pontossága mérhető a pontosság mérőszámmal, akkor erre alapozva, mérték adható meg webes metakeresőgép relatív pontosságának felhasználó-független mérésére. Az ötlet a következő. A metakereső találati listáját összevetve a használt keresők listájával, relatív pontosság definiálható. Ha a webes keresőgépek találati listáinak első *m* eleméhez viszonyítjuk az őket használó metakereső válaszait, akkor ez utóbbi számára relatív, a Web-keresőgépekhez viszonyított pontosságot lehet értelmezni és kiszámítani.

Legyen:

- *q* egy kérdés,
- *V*: a vizsgált metakereső által visszaadott találatok száma.
- *T*: *V* azon elemeinek száma, amelyeket a használt keresők közül legalább az egyik, találati listájának első *m* helyén belül rangsorolt.

Ekkor a metakereső $RP_{q,m}$ relatív pontossága a következőképp számolható:

$$RP_{q,m} = \frac{T}{V}$$

m értéke lehet például 10 vagy 5, vagy más érték, különböző szempontokat figyelembe véve (pl. a mérés tartománya).

A relatív pontosság értéket több kérdésre ki kell számolni, és ezeket az értékeket átlagolni kell. A relatív pontosság átlaga a következő:

$$\frac{\sum_{i=1}^{n} RP_{qi,m}}{n}$$

Az RP eljárás erősen függ attól a feltevéstől, hogy a keresőmotorok találati listája tartalmaz-e releváns találatokat. Más szavakkal, az RP mérés csak annyira jó, mint a találati listák.

Példa

- Tegyük fel, hogy a mérendő metakereső négy webes keresőmotort használ.
- Legyen *q* egy kérdés
- Továbbá tegyük fel, hogy öt találatot ad vissza a metakereső vagyis, *V*=5.

Vizsgáljuk meg a keresőmotortok találati listáit:

- a metakereső első találata az első keresőmotor listáján a harmadik,
- a második találat a második motor első találata,
- a harmadik találat a negyedik motor harmadik találata,
- a negyedik találat a negyedik motor második találata,
- az utolsó találat a második motor harmadik találata.

Ezért $T = 5$, és $m = 10$ esetén a relatív pontosság:

$$RP_{q,10} = 5 / 5 = 1.$$

7. fejezet

Kapcsolatelemzésű információ visszakereső módszerek

7.1 I^2R Adaptív Klaszterező Technika Információ–visszakereséshez

7.1.1 Klaszterezés

A klaszterezés egy jól ismert technika az információ-visszakeresésben. A visszakeresendő dokumentumokat — általában — diszjunkt halmazokba csoportosítjuk. E halmazokat klaszthereknek nevezzük. Minden klaszter, bizonyos értelemben, hasonló dokumentumokból áll.

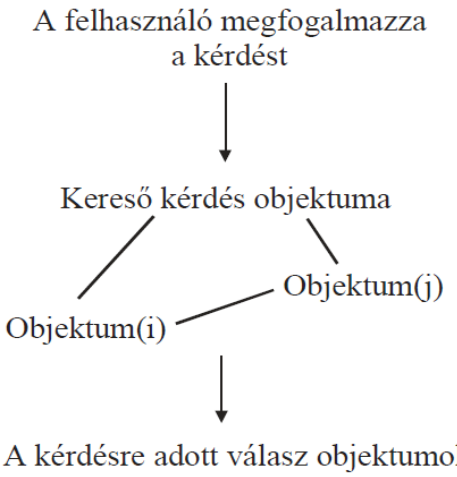
A visszakeresés egy klaszter–képviselő (reprezentáns) alapján történik. A reprezentánsnak nem kell feltétlenül klaszter–tagnak lennie. A visszakereső módszer minden egyes kérdéshez (keresőkérdés) egy reprezentánst társít. A kérdésre visszakapott válasz a reprezentáns által képviselt klaszter minden tagját tartalmazza.

Ez a szemlélet a klaszter hipotézisen alapul: a szorosan asszociált dokumentumok ugyanarra a kérdésre relevánsak. A klaszter hipotézis a priori, vagyis a kérdéstől függetlenül, a keresés végrehajtása előtt történik meg a csoportok kialakítása. Az algoritmus stabil a növekedésre (új dokumentum hozzáadásával valószínűleg nem változik jelentősen a klaszterek szerkezete), a leírási módra (a dokumentumok reprezentálási módjának kis mértékű megváltoztatása alig befolyásolja a klaszter szerkezetet), a rendezésre nézve (a szerkezet független a dokumentumok rendezésétől, ha egyáltalán van).

A fent említett fix klaszterezési módnál hatékonyabbnak bizonyult az adaptív módszert alkalmazó információ-visszakeresés. Az adaptív klaszterezés során a felhasználó által feltett kérdések hatására alakulnak ki a klaszterek.

7.1.2 Asszociatív Kölcsönhatás alapú Információ Visszakeresés

A kölcsönhatás-alapú (interakciós) információ-visszakeresésnek rövidítése I^2R (Interaction Information Retrieval). A felhasználó által feltett kérdés hatást gyakorol az eredeti dokumentumhalmazra: részben megváltoztatja a dokumentumok közötti kapcsolatokat (kölcsönhatás), és ez a kölcsönhatás indítja el a visszakeresési folyamatot.



24. ábra. A visszakeresés folyamata

A kölcsönhatás megvalósításának egyik módja, hogy a dokumentumok, amelyeket a későbbiekben objektumoknak fogunk hívni, nem egymástól elszigetelt egységeket képeznek, hanem egy összekötött hálózatot, amit a felhasználó kérdése a válasz megadása előtt részlegesen átalakít (kölcsönhatásba lép vele). Az I^2R matematikai modellje a mesterséges neuronhálózat alapvető állapotegyenletén alapszik. Így az objektumok megfelelnek egy neuronhálózatnak, ahol az egyes objektumok egy-egy neuronnak felelnek meg, melyek képesek különböző szintű aktivitást produkálni.

A dokumentumokat objektumok rugalmasan összekapcsolt hálózatával reprezentáljuk. Az objektumok közti kapcsolatokat minden alkalommal újra értékeljük, amikor új objektumot kapcsolunk a hálózathoz. A kérdés is kölcsönhatásba lép az objektumokkal: összekapcsolódik a többi már összekapcsolt objektummal. Egyrészt új kapcsolatok jönnek létre, másrészt a már meglevőek némelyike megváltozhat. Ezek alapján a kérdést is objektumnak tekintjük.

Minden objektumhoz rendeljünk hozzá egy vektort, amivel azonosítjuk. Ez a vektor az objektumra jellemző kulcsokat tartalmazza. Vagyis bármely o_i, i = 1, 2, ..., M, dokumentum objektumhoz t_{ik}, k = 1, 2, ..., n_i azonosítókat (kulcsszavak, index kifejezések) rendelünk hozzá. Bármely (o_i, o_j) objektum-pár között súlyozott és irányított kapcsolat van. Az egyik jellemző súlyszám a gyakoriság, mely az f_{ijp} szám és az o_i objektum n_i hosszának a hányadosa, ahol: f_{ijp} = a t_{jp} kifejezés hányszor fordul elő az o_j objektumban, n_i = az o_i -ben található összes kifejezés száma.

A gyakoriság:

$$w_{ijp} = \frac{f_{ijp}}{n_i},\ p = 1, \ldots, n_j$$

Mivel a w_{ijp} azt a gyakoriságot jelöli, amellyel az o_i objektum a t_{jp} azonosítót szolgáltatja, a hozzá tartozó kapcsolatot $o_i \rightarrow o_j$ irányúnak tekinthetjük.

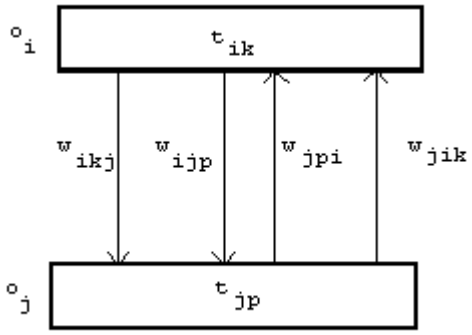
A másik jellemző súlyszám a w_{ikj}, inverz gyakoriság, amely megadja, hogy egy adott kifejezés mennyire tükrözi egy dokumentum tartalmát, ahol: f_{ikj} = a t_{ik} kifejezés hányszor fordul elő az o_j objektumban, df_{ik} = azon dokumentumok száma, melyekben a t_{ik} kifejezés előfordul.

Az inverz gyakoriság:

$$w_{ikj} = f_{ikj} \log\left(\frac{2M}{df_{ik}}\right)$$

A w_{ikj} annak a mértéke, hogy az o_j objektum tartalmát mennyire fejezi ki a t_{jp} kifejezés, ezért a hozzá tartozó kapcsolatot $o_i -> o_j$ irányúnak tekinthetjük.

A másik kettő, ellentétes irányú w_{ijp}, w_{jpi} kapcsolatot hasonlóan értelmezzük. A kapcsolatokat a következő ábra szemlélteti.



25. ábra. Az objektum-párok közti kapcsolatok

Példa
Legyen három dokumentumunk:

D_1 = Az információ-visszakeresés egy nagyon gyorsan és dinamikusan fejlődő tudományág.

D_2 = Napjainkban a kereskedelmi keresők a Boole-féle információ-visszakeresés modelljét használják leggyakrabban. Ez a megoldás implementálható a legkönnyebben.

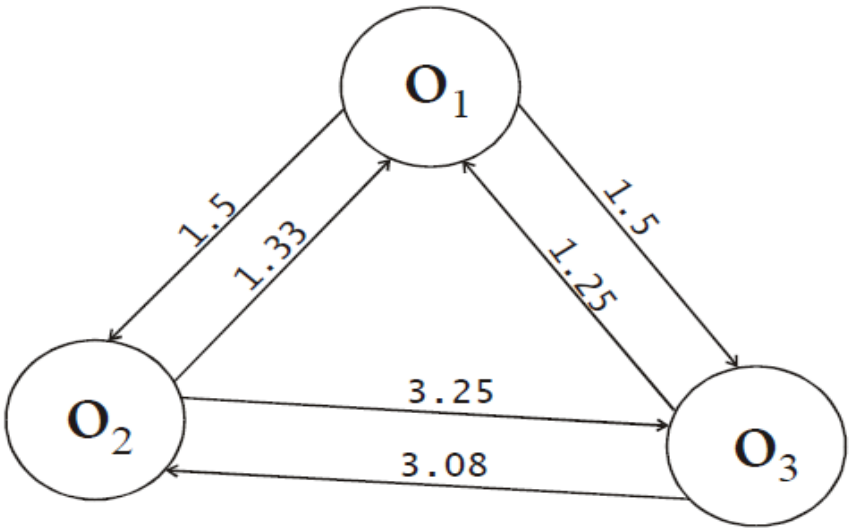
D_3 = Az implementálás során fontos a memória és a lemez megfelelő használata. Az információ-visszakeresés implementálása nem egyszerű feladat.

Az objektumokat kifejezések n-eseként fogjuk fel, azaz indexeljük őket a következő módon:

o_1 = (t_{11} = információ-visszakeresés, t_{12} = tudományág)

o_2 = (t_{21} = információ-visszakeresés, t_{22} = Boole-féle, t_{23} = implementálás)

o_3 = (t_{31} = információ-visszakeresés, t_{32} = implementálás, t_{33} = memória, t_{34} = lemez)



26. ábra. A kiindulási objektumok hálózata

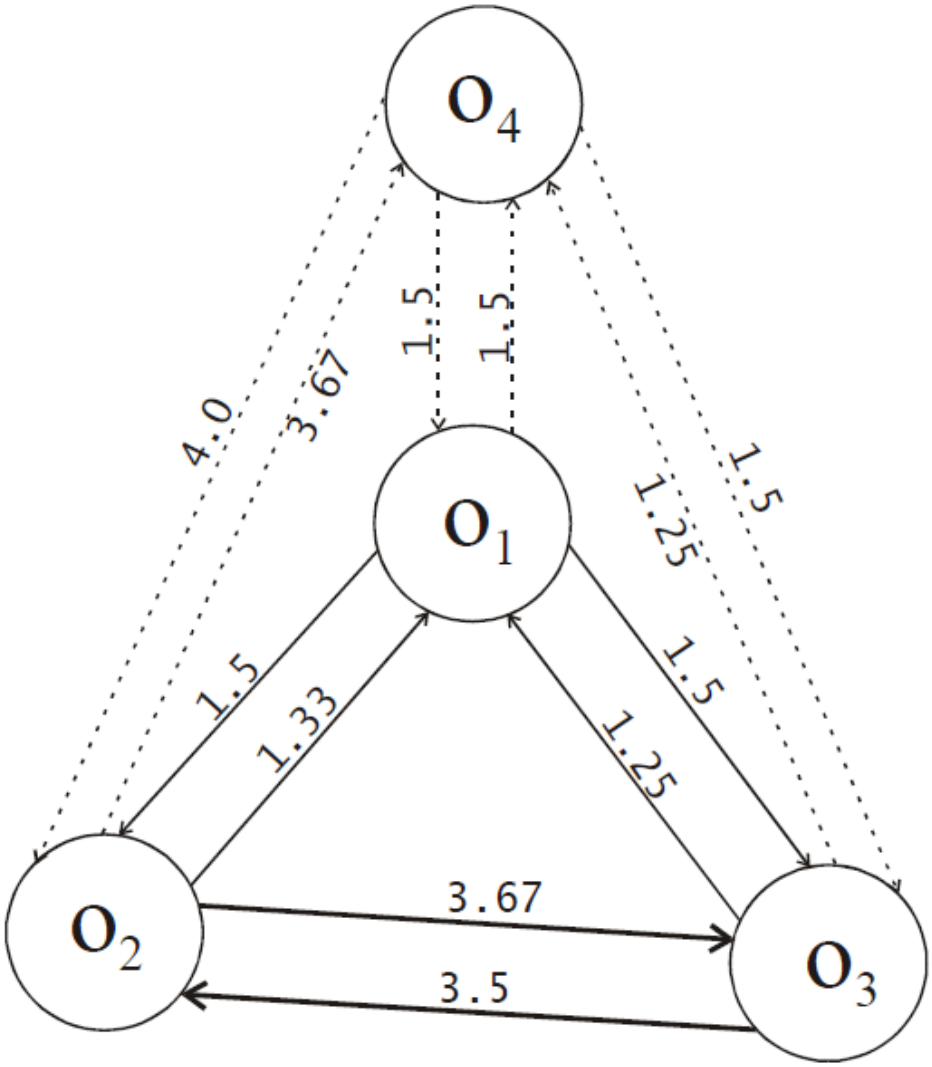
A kérdés legyen a következő:

Q = Használják a Boole-féle információ-visszakeresés modelljét?

Q is a szerkezet része lesz, összekapcsolódik az objektumok hálózatával. A kérdésből indexeléssel a következő objektumot kapjuk, ami beépül az objektum hálózatba. (Itt a két szót semmiféle Boole-operátor nem köti össze.)

o_4 = (t_{41} = információ-visszakeresés, t_{42} = Boole-féle)

Az ábra a kérdés beépülése után mutatja a hálózatot. Egyrészt új kapcsolatok jöttek létre , másrészt néhány régi súlyszám megváltozott. Az új súlyokat szaggatott, a régi, de megváltozott súlyokat vastag vonallal jelöli.



27. ábra. A kérdés beépülése utáni hálózat

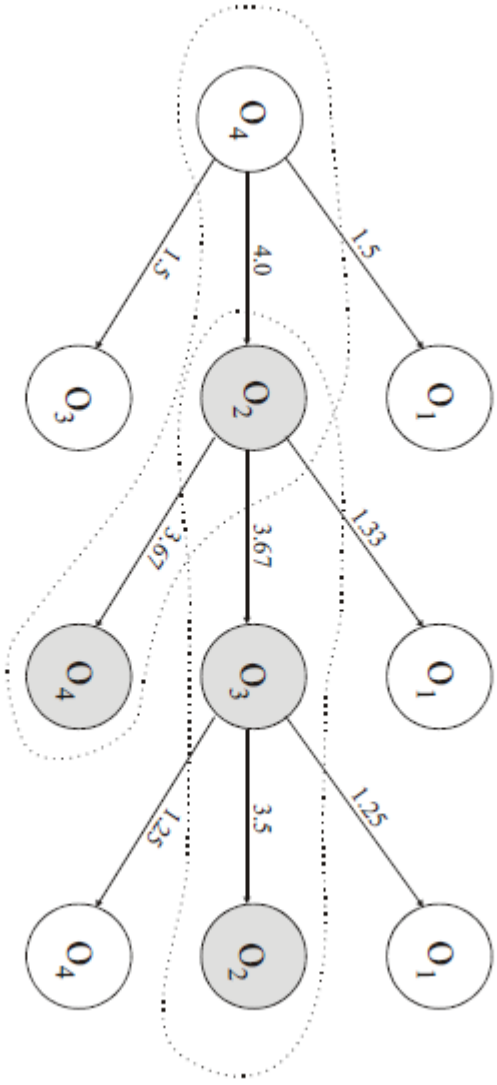
Az objektumokat mesterséges neuronok hálózatának tekinthetjük. Az aktiváció az o_4 kérdéstől indul, és a legerősebb kapcsolat mentén terjed neuronról-neuronra. A kérdés és a többi o_i objektum közti kapcsolat erősségét a következőképpen definiáljuk:

$$\sum_{p=1}^{n_j} w_{jpi} + \sum_{k=1}^{n_i} w_{jik}$$

Az összegzést a w_{jpi} és w_{jik} előzőekben megfogalmazott jelentése teszi lehetővé. Néhány lépés után az aktiváció egy már érintett objektumhoz ér, vagyis öningerlő (reverberáló) kör alakul ki: az aktiváció objektumok egy körében terjed. Ez megfelel a kérdés által visszahívott lokális memóriának. Az öningerlő kört a kérdéshez rendelt adaptív klaszterként értelmezhetjük. Egy kérdésre azokat az objektumokat kapjuk meg válaszként, amelyek ugyanahhoz a reverberáló körhöz tartoznak. A válaszokat a maximális aktiváció alapján rangsoroljuk

Példánkban az aktivitás a kérdés objektumtól, o_4, indul és halad objektumról objektumra. A folyamatot az a következő ábra szemlélteti. Az folyamat az o_4 objektumtól, azaz a kérdéstől indul. A

legnagyobb súllyal az o_2 objektum kapcsolódik, így a WTA (winner takes all) stratégia elvén ez lesz a következő objektum, amerre az aktivitás terjed. Ezután vizsgáljuk az o_2 kapcsolatait a többi objektummal. Itt két objektum is azonos súllyal szerepel (o_3 és o_4). Itt egy elágazás történik. Azonban o_4 objektum már volt kiválasztva, így megvan az első öngerjesztő kör. A másik irányba tovább terjed az aktivitás: o3 objektum vizsgálatakor o2 objektum lesz a győztes, ő visz mindent. Ezzel bezárul egy újabb kör és elhal az aktivitás. Lefutott a visszakeresés. A kölcsönhatás eredménye két öngerjesztő kör, melynek tagjai adják a kérdésre a választ. Így a válasz-objektumok: o_2 és o_3. (o_4 nem szerepel a válaszok között, hiszen az maga a kérdés.) o_3 nem tartalmaz Q-beli szavakat, mégis válasz, mivel más kulcsok révén erősen kapcsolódik o_2 objektumhoz. Ez a tulajdonság sajátja az I^2R modellnek.



28. ábra. A keresés folyamata

7.2 PageRank

A számítógépek korszakában élünk. Az Internet a mindennapi életünk része lett. Csak megnyitjuk a kedvenc keresőprogramunkat, mint például a Google, beírunk egy kulcsszót és néhány szempillantás alatt megkapjuk a kérdésre releváns oldalakat. Az elmúlt évek során a Google kereső a legtöbbet használt keresőoldallá vált világszerte. Sikerét egyedi találati rangsorának köszönheti, amelynek alapja egy, a weboldalak rangsorolására kidolgozott kifinomult metódus.

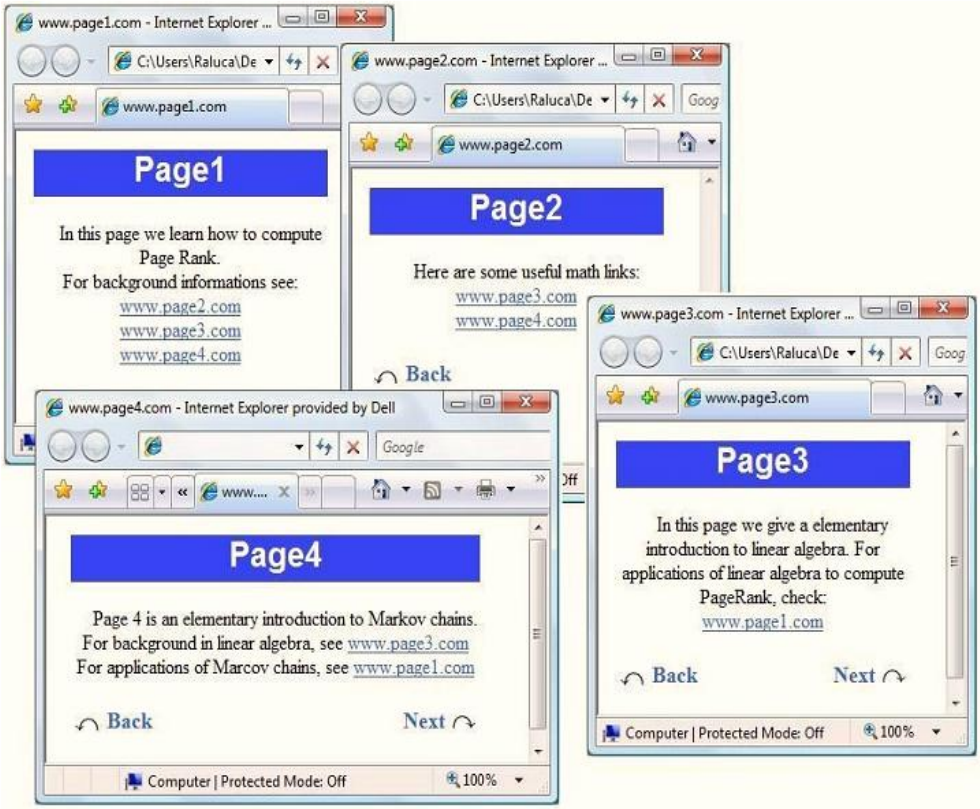
A különböző webes keresők különböző megoldásokat fejlesztettek ki a weboldalak rangsorolására. A Google megjelenéséig gyakorlatilag az összes webes kereső rangsorolási technikái számára a keresett kifejezés előfordulásának száma volt az egyik meghatározó faktor. Ezáltal a keresett kifejezés előfordulásának számát súlyozták a dokumentum hosszával (kulcsszó-sűrűség alapú rangsorolás), vagy azt vizsgálták, hogy a keresett kifejezés hol, milyen kiemelés jellegű HTML elemben található (címsor, szövegkiemelés). A jobb találati eredmények érdekében a kifinomultabb algoritmusok figyelembe veszik az úgynevezett linknépszerűséget (link popularity). Eszerint a bejövő hivatkozások száma határozza meg egy adott web dokumentum általános értelemben vett fontosságát. Vagyis, minél több oldal hivatkozik egy weblapra, annál fontosabbnak tekinthetjük az adott lapot.

A PageRank a Google internetes keresőmotor alapja, amit a Google alapítói, Larry Page és Sergey Brin fejlesztettek ki 1998-ban a Stanford Egyetemen [21.]. A PageRank egy ún. rekurzív algoritmus, de magát a weboldalhoz rendelt számot is PageRanknek nevezik. A Google keresőgép keretén belül a PageRank módszer a nyilvános Web (public Web) pásztázott részének hivatkozási gráfját használja fel arra, hogy a weboldalak relatív fontosságának egy mértékét számítsa ki. A linknépszerűséggel ellentétben a PageRank érték nemcsak egyszerűen a bejövő linkek számától függ. Az alapelv az, hogy minél több weboldal hivatkozik az adott weblapra, annál fontosabb, viszont a bejövő hivatkozások nem egyenértékűek. Összességében egy weblapnak magas a PageRank értéke, ha más magas PageRank értékű dokumentumokról mutatnak rá hivatkozások. Ha létrehozunk egy *i* weboldalt, amely egy hivatkozást tartalmaz a *j* weboldalra, akkor ez azt jelenti, hogy a *j* weboldalt fontosnak és relevánsnak tartjuk weboldalunk témájának tekintetében. Ha sok weboldal hivatkozik *j*-re, akkor ez azt jelenti, hogy a közösen hisszük, hogy a *j* oldal fontos. Az is előfordulhat, hogy csak egy link mutat egy adott oldalra. Azonban, ha ennek az oldalnak nagy a tekintélye (mint, pl. a www.google.com), akkor ez az oldal átadja tekintélyét a hivatkozott oldalnak, így az szintén fontossá válik. Akár tekintélyről, akár népszerűségről beszélünk, minden weboldalhoz egy rangot, egy értéket rendelhetünk, méghozzá iteratív módon. Egy weboldal értéke a rá mutató weboldalak értékétől függ.

A fenti modell alkalmazása céljából tekintsük a webet egy irányított gráfnak. A gráf csúcsai a weboldalak, a gráf élei pedig a weboldalak közti irányított linkek.

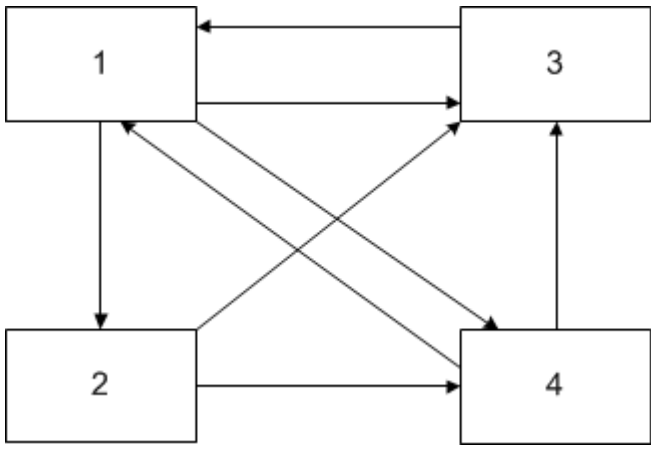
Példa

Tegyük fel, hogy van egy mini webünk, amely mindössze négy weboldalból áll: www.page1.com, www.page2.com, www.page3.com, www.page4.com. A weboldalak közti kapcsolatot a következő ábra szemlélteti.



29. ábra. Mini web

A mini web ábrázolható egy irányított gráffal. A négy weboldal négy csúcs szemlélteti. Ha az *i* weboldalról link mutat a *j* weboldalra, akkor adjunk a gráfhoz egy irányított élt az *i* és *j* csúcsok között. A weboldalak elemzése eredményeképpen a következő gráfot kapjuk.



30. ábra. Mini web gráf

Ebben a modellben, minden weboldal egyenlő módon adja át fontosságát a hivatkozott oldalnak.

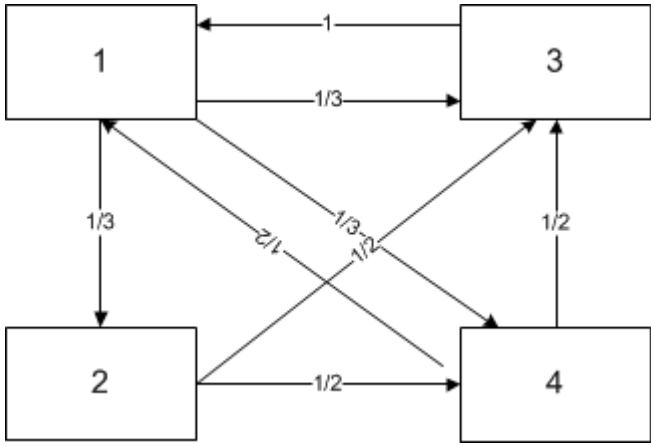
Példa

Az egyes csúcsnak három kimenő éle van, ezért fontosságának 1/3-ad részét adja át a hivatkozott másik három weboldalnak.

Általában, ha egy csúcsnak *k* kimenő éle van, akkor fontosságának 1/k-ad részét adja tovább a hivatkozott weboldalaknak. Hogy jobban szemléltessük a folyamatot, rendeljünk az élekhez súlyokat a következő módon.

Jelölje $W_1, \ldots, W_i, \ldots W_N$ weboldalak egy halmazát. Ha a W_i weboldalról irányított link mutat a W_j weboldalra, akkor azt a következő módon jelöljük: $W_i \overset{23}{11} W_j$. A weboldalak alkotta web gráf $A = (a_{ij})_{NxN}$ átmeneti mátrixa pedig legyen a következő:

$$a_{ij} = \begin{cases} \frac{1}{L_j}, & W_i \to W_j \\ 0, & \textit{egyébként} \end{cases}$$



31. ábra. Súlyozott mini-web gráf

A fentiek alapján a mini web gráf *A* átmeneti mátrixa a következő alakban írható fel:

$$A = \begin{pmatrix} 0 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix}$$

Tegyük fel, hogy kezdetben a fontosság egyenletes eloszlású a weboldalak közt, vagyis mindegyik weboldal fontossága ¼..Minden, a weboldalra mutató link növeli az adott weboldal fontosságát. Tehát első lépésként frissítsük minden weboldal rangját oly módon, hogy megnöveljük a fontosság értékét az oldalra mutató weboldalak fontosságának értékével. Vagyis az első lépésben az új fontosság vektor: $v_1 = Av$. A folyamatot iterálva kapjuk, hogy a második lépésben a fontosság vektor : $v_2 = A(Av) = A^2v$

$$v = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix} \qquad Av = \begin{pmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{pmatrix} \qquad A^2 v = A(Av) = A\begin{pmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{pmatrix} = \begin{pmatrix} 0.43 \\ 0.12 \\ 0.27 \\ 0.16 \end{pmatrix}$$

Az iterációk v, Av, $...A^k v$ sorozata egy v^{PR} egyensúlyi érték felé tart. Ezt a vektort a web gráf PageRank vektorának nevezzük.

$$v^{PR} = \begin{pmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{pmatrix}$$

A PageRank szerint az 1-es weboldal a legfontosabb oldal. Ez meglepőnek tűnhet, hiszen az 1-es weboldalra csak két link mutat szemben a 3-as weboldallal, melyre három link is mutat. Miért mégis az 1-es a legfontosobb oldal? Vizsgáljuk meg alaposan a webgráfot. A 3-as weboldal kifoka mindössze egy, ezért fontosságának egészét átadja az 1-es weboldalnak. Hasonlóan, ha egy weben szörfölő felhasználó a 3-as weboldalt nézegeti, akkor innen csak az 1-es weboldalra léphet tovább. Figyeljük meg azt is, hogy egy weboldal fontossága nem csupán a rá mutató linkek súlyának az összegéből adódik. Első lépésben megkapja a szomszédjai alapján a fontossági értékeket, majd a szomszédok szomszédjai alapján, és így tovább.

A fenti elgondolás az úgynevezett véletlenszerűen szörfölő felhasználó modellje. Brin és Page PageRank algoritmus működését ahhoz hasonlították, mint amikor az interneten szörfölő felhasználó a hivatkozott tartalom figyelembevétele nélkül, véletlenszerűen klikkelget az egyes linkekre. Képzeljünk el egy személyt, aki a weben böngészik, vagy, más szófal szörfözik. Elindul egy tetszőleges weboldalról, majd az ott található hivatkozások segítségével egy másik oldalra ugrik. Ott megint található egy érdekes hivatkozás. Ennek segítségével ismét egy másik oldalra kerül, és így tovább. Azonban a szörfölés sem tart korlátlan ideig. Egy idő után a felhasználó elunja magát, és teljesen véletlenszerűen felkeres egy új weboldalt, ahonnan tovább folytatja az előbbi szörfölést. A találomra klikkelgető internetező egy adott oldalra meghatározott valószínűséggel jut el, és ez a weblap PageRank értékével függ össze. Annak a valószínűsége, hogy a szörfölő egy adott linkre klikkeljen, az oldalon található linkek számától függ. Ezért van az, hogy a hivatkozó oldal PageRank értékét nem adja át teljes egészében egy hivatkozott oldalnak, hanem elosztásra kerül a hivatkozó oldalon található összes hivatkozás számával.

A fentieket figyelembe véve tekintsük, hogyan történik a Google keresőben a felhasználó által megadott keresőkérdésre releváns weboldalak visszakeresése és rangsorolása. A lépések a következők:

- a keresőkérdés egyes kifejezéseit tartalmazó weboldalak megtalálása (keresés inverz file szerkezetben);
- a Web-oldalak relatív fontosságának kiszámítása;
- a Web-oldalak rangsorolása.

Az oldalak relatív fontosságának kiszámítása több tényezőtől függ:

- a keresőkérdés kifejezéseinek előfordulási helye az oldalon belül (pl. címben, URL-ben), vagy a kifejezések egymástól való –szavakban mért –távolsága (on page factors);

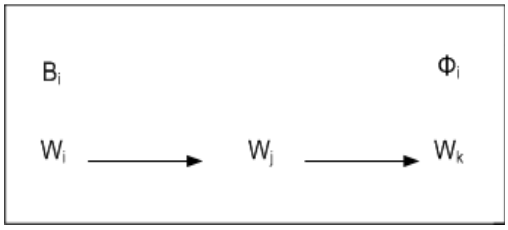
  a kifejezések alaki megjelenése, mint pl. betűtípus, -méret, -szín;
- a kifejezések előfordulásainak száma a Web-oldalon;
- az oldal PageRank értéke;

- egyéb tényezők (szponzorok, fizetett reklám, oldalak manuális letiltása, a relatív fontosság manuális módosítása).

Bár nyilvánosan nem ismert, feltételezhető, hogy a keresőkérdés és a weboldal számára egy-egy numerikus vektort számítanak ki (a fenti tényezők alapján), majd ezek pontszorzatának és az oldal PageRank értékének valamilyen súlyozott összefüggése adja az oldal relatív fontosságát. Egy oldal PageRank értéke a rámutató oldalak PageRank értékeitől valamint az ezekből az oldalakból kiinduló linkek számától függ. A PageRank módszer a hivatkozáselemzés (link analysis) elméletén alapszik. Ennek alapelve az, hogy a hivatkozások száma a fontosság egy mértéke. A PageRank módszer annyiban módosította ezt az alapelvet, hogy a hivatkozások számát nem tekintette többé abszolútnak, hanem relativizálta azt: egy weboldal fontossága függ a többi oldalétól is.

A PageRank módszer a következőképpen értelmezett G gráfként modellezi a web szerkezetét:

- $\Omega = \{W_1, W_2, \ldots, W_i, \ldots, W_N\}$ jelöli a Web-oldalak halmazát,
- $\Phi_i = \{W_k \mid k = 1, \ldots, n_i\}$ jelöli azoknak a Web-oldalaknak a halmazát, amelyekre rámutat a W_i oldal, $\Phi_i \subseteq \Omega$,
- $B_i = \{W_j \mid j = 1, \ldots, m_i\}$ jelöli azoknak a Web-oldalaknak a halmazát, amelyek rámutatnak a W_i oldalra, $B_i \subseteq \Omega$.



32. ábra. Web-oldalak *G* gráfja a PageRank módszer számára.

Jelölje R_i a W_i Web-oldal PageRank értékét. Ezt a következő egyenlet adja meg:

$$R_i = \sum_{W_j \in B_i} \frac{R_j}{L_j}$$

ahol L_j a W_j oldalból kiinduló linkek számát jelöli. Az egyenlet homogén lineáris egyenletrendszer. Ennek mindig van triviális megoldása, a null vektor.

Ha a *G* gráf erősen összefüggő, hurokmentes, akkor van nemtriviális megoldás. Ha a *G* gráf erősen összefüggő, akkor az $M = (1/L_j)_{N \times N}$ mátrixnak oszlopösszegei 1-gyel egyenlőek. Mivel az *M* mátrix főátlója zérus, az *M*−*I* mátrix (*I* az egységmátrix), ami egyben az egyenlet mátrixa is, oszlopösszegei nullák. jelölje $D = |M-I|$ ennek a determinánsát. Ha a *D* determináns első sorához hozzáadjuk a többi sorát, akkor az első sor zérus lesz, így D = 0. Tehát a Rouché-tétel alapján az egyenletrendszernek van triviálistól különböző megoldása is.

Mivel $|M-I| = 0$, az 1 szám az M mátrix sajátértéke, a PageRank értékek az ennek megfelelő sajátvektor. Az egyenlet az $M \times R = R$ mátrixegyenlet alakban is írható, ahol $R = [R_1,...,R_i,...,R_N]^T$ a PageRank értékek vektora. A gyakorlatban a PageRank értékeket az $R_{i+1} = M \times R_i$ rekurzív közelítő numerikus számítási mód révén számítják ki.

A gyakorlatban a PageRank értékek kiszámítása valamely numerikus közelítő eljárás segítségével történik (annak tekintetében, hogy N az nagy szám). Az alábbi közelítési módszer alkalmazható:

$$M \times R_k = R_{k+1},\ k = 0,\ 1,\ \ldots,\ K$$

$$R_0 = \left[\frac{1}{N} \ldots \frac{1}{N}\right]$$

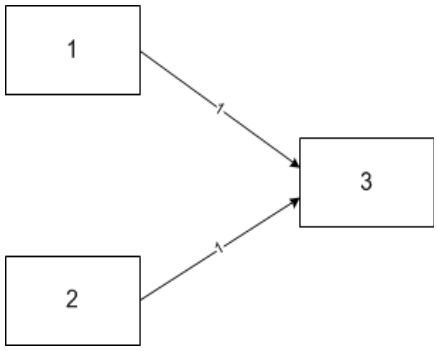
ahol K értéke általában 50, vagy rekurzívan adjuk meg, amíg el nem érjük az előre megadott ε hiba küszöbértéket.

$$\text{Max} \mid R_{k+1} - R_k \mid < \varepsilon$$

A valóságban a Web gráfja általában nem erősen összefüggő, vannak:

- elszigetelt oldalak (sem be-, sem kimenő link nem szerepel),
- nyelő oldalak (csak bemenő link van),
- forrás oldalak (csak kimenő linkvan).

Az ábrán a 3-as weboldal nyelő oldal. Csak bemenő linkjei vannak.



33. ábra. Nyelő oldal

A fenti tulajdonságok miatt az egyenletnek a gyakorlatban több változata is használatos.

7.3 HITS

A HITS (Hypertext Induced Topic Search = hiperhivatkozásos indukált témakeresés) egy kapcsolat elemzés alapú algoritmus nagyjából a PageRank-kel egy időben került kifejlesztésre [22.]. Ez az algoritmus egyrészt meghatározza az oldalak tartalmának fontosságát, amit tekintély értéknek (authorithy value) hívunk. Másrészt meghatározza az oldalra mutató linkek fontosságát, amit elosztó értéknek (hub value) hívunk. Mindkét érték kiszámítása csupán a kapcsolatok szerkezete alapján, iteratív algoritmussal történik, hasonlóan, mint a PageRank esetében. A HITS algoritmus, szemben a PageRank-kel, a tekintély és elosztó értékeket egy adott kérdésre visszakapott weboldalak egy halmazára számolja ki. Az elosztó és a tekintély kifejezések weboldal típusokat jelölnek, ezek kölcsönösen erősítő kapcsolatban állnak egymással. Egy weboldal tekintély típusú, ha sok elosztó típusú weboldal mutat rá, és elosztó típusú, ha sok tekintély oldalra mutat. Az egyes

weblapok közti kapcsolódás nemcsak arra mutathat rá, hogy melyik oldalak elismertebbek, de arra is, hogy melyek tartoznak többé-kevésbé egy témakörbe. A HITS módszer, a Teoma keresőgép (www.teoma.com) rangsorolási eljárásának része.

34. ábra. A Teoma kereső

Ez a módszer az interneten elkülöníthető kisebb közösségeket próbálja felismerni. Közösség alatt azokat a honlap-csoportosulásokat értjük, melyek nagyjából hasonló témakörrel foglalkoznak, s ezért jellemzően egymáshoz kapcsolódnak hivatkozásokkal. Nem csak az számít, hogy nagy tekintéllyel, vagy más szóval fontossággal rendelkező oldalakról mutassanak hivatkozások oldalainkra, de az is számít, hogy a hivatkozások hasonló témakörű oldalakról érkezzenek, mint amilyen a miénk.

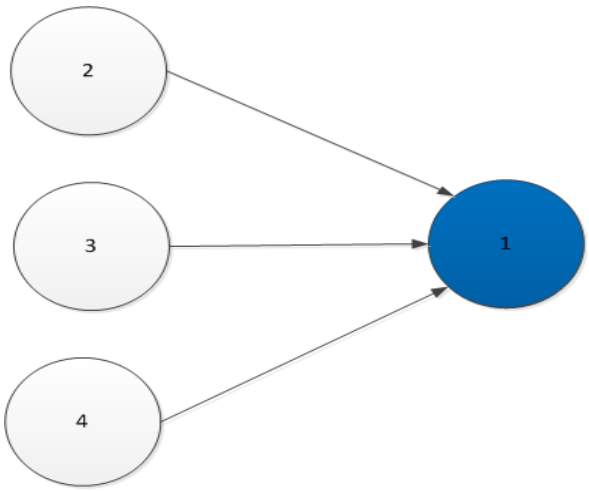
Adott egy p weblap, amelynek a tekintélysúlya $x^{<p>}$, az elosztósúlya pedig $y^{<p>}$. Ha p sok olyan hivatkozást tartalmaz, amelyek nagy tekintélysúlyú oldalakra mutatnak, akkor nagy elosztósúllyal rendelkezik. És ha a p oldalra sok nagy elosztósúllyal rendelkező oldal hivatkozik, akkor az ő tekintélysúlya szintén nagy lesz.

Az elosztó súlyokat a következő iteratív módon számítjuk ki. E a web gráf éleinek halmazát jelöli.

$$x^{\langle p \rangle} \leftarrow \sum_{q:(q,p)\in E} y^{\langle q \rangle}$$

Példa

Az elosztó súlyok számítását az ábra szemlélteti.



x(1)= y(2) + y(3)+ y(4)

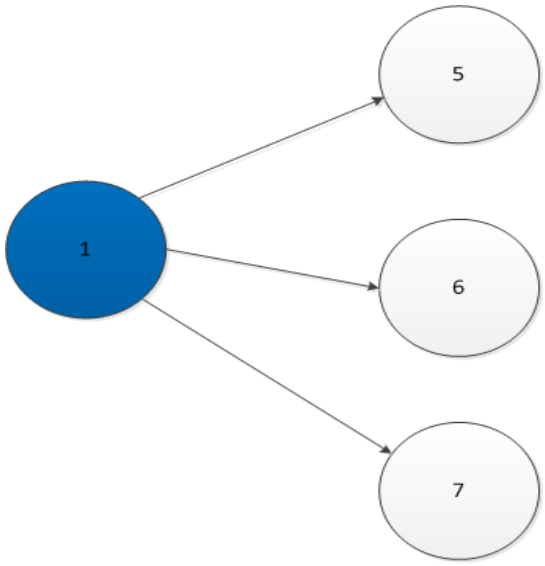
35. ábra. Elosztó súlyok számítása

A tekintély súlyokat a következő iteratív módon számítjuk ki. *E* a web gráf éleinek halmazát jelöli.

$$y^{\langle p \rangle} \leftarrow \sum_{q:(p,q)\in E} x^{\langle q \rangle}$$

Példa

A tekintély súlyok számítását az ábra szemlélteti.



y(1)= x(5) + x(6)+ x(7)

36. ábra. Ábra Tekintély súlyok számítása

Legyen *M* a web gráf adjacencia mátrixa. Ekkor az előző egyenletek megadhatók a következő alakban:

$$x^{\langle k \rangle} = M^T M x^{\langle k-1 \rangle}$$

$$y^{\langle k \rangle} = MM^T y^{\langle k-1 \rangle}$$

Az M^TM az elosztó mátrix, míg a MM^T a tekintély mátrix. Vagyis a HITS módszer megfelel a következő sajátvektor probléma megoldásának.

$$M^T Mx = \lambda x$$
$$MM^T y = \lambda y$$

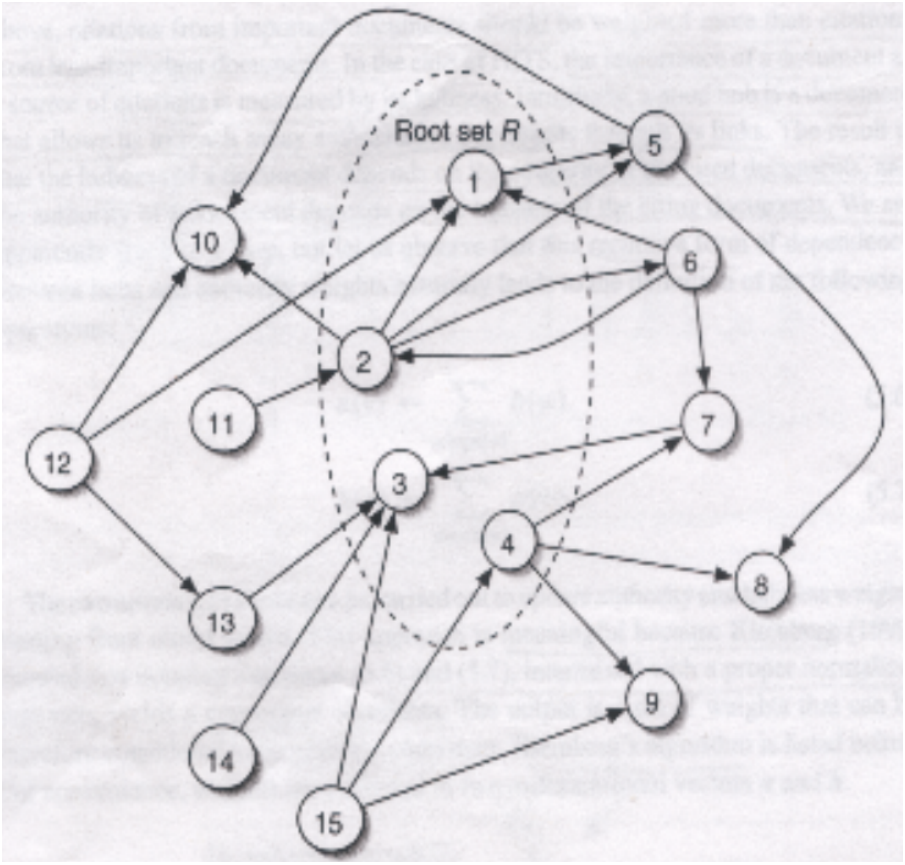
A súlyokat numerikus közelítő eljárással számítják ki az x_0=(1,…,1) és y_0=(1,…,1) kezdőértékektől indulva. Kimutatható, hogy x az M^TM sajátvektora, y pedig az MM^T-jé , ahol M a web gráf adjacencia mátrixa. A tekintély illetve elosztó súlyok csökkenő sorrendje szerint rangsorolják az oldalakat a tekintély illetve elosztó típuson belül.

Az $A = M^TM = (a_{ij})$ mátrix jól ismert a bibliometriában együtthivatkozási mátrix néven (cocitation matrix): az a_{ij} azoknak az oldalaknak a száma, amelyek mindegyike egyidejűleg mutat az i és j oldalakra. A $H = MM^T = (h_{ij})$ mátrix pedig a bibliometriai kapcsolás mátrix (bibliometric coupling matrix): a h_{ij} azoknak az oldalaknak a száma, amelyekre mind az i, mind a j oldalak mutatnak.

A gyakorlatban a tekintély és elosztó vektorok kiszámítása a következő lépéssorozat végrehajtásával történik.

1. lépés: Adjuk meg weblapok egy R gyökérhalmazát (root set), például a következő módon: hajtsunk végre egy keresést valamilyen témában valamely keresőmotorral és tekintsük a kereső által visszaadott első L találatot.

2. lépés: Bővítsük az R halmazt azokkal a weblapokkal, amelyek az R-beli weblapokra mutatnak, illetve azokkal, amelyekre R-ből mutat link. Az így kapott halmaz lesz a T alaphalmaz. A T halmazt az ábra szemlélteti.



37. ábra. Az R gyökérhalmaz és T alaphalmaza

3. lépés: Távolítsuk el az azonos domain névvel rendelkező oldalakat.

4. lépés: Adjuk meg a T halmazban maradt weboldalak web gráfját.

5. lépés: Hajtsunk végre megfelelő számú iterációt x és y meghatározására a következő kezdeti értékektől indulva az alábbiak szerint:

$$x_0 = [1,\ldots,1]^T$$

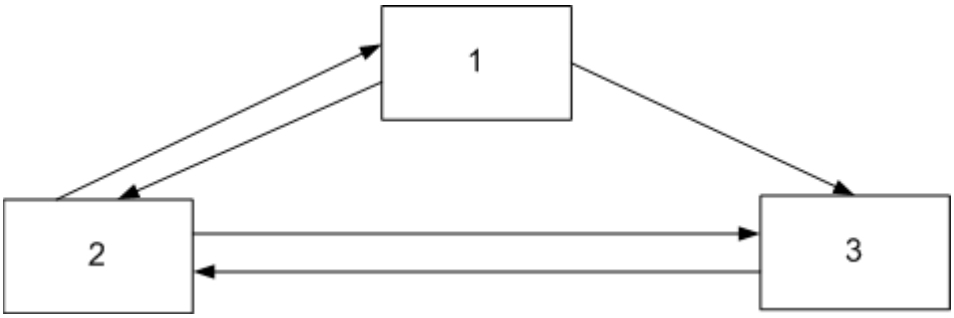
$$y_0 = [1,\ldots,1]$$

$$x_{j+1} = M^T y_i$$

$$y_{i+1} = M x_{i+1}$$

Az egyes iterációk után az x és y vektort normalizáljuk hosszra normalizálás módszerével. Belátható, hogy az x a domináns sajátvektora az M^TM mátrixnak, és y a domináns sajátvektora az MM^T mátrixnak.

Példa



38. ábra. Mini-web

Tekintsük az ábrán látható mini web gráfot. Legyen M a gráf adjacencia mátrixa.

$$M = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Ekkor az M^TM elosztó mátrix a következő alakban írható fel:

$$MM^T = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

Az MM^T a tekintély mátrix a pedig a következő lesz:

$$M^T M = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

A sajátértékek: 1.5, 0.2, 3.2.

Hajtsuk végre a $x_{i+1} = M^T y_i$ és $y_{i+1} = M x_{i+1}$ műveleteket egészen addig, amíg az x és y vektorok értéke nem változik jelentősen. A példánkban három iterációs lépés végrehajtása után a következő értékeket kapjuk:

$$x = [0.309; 0.619; 0.722]^T$$

$$y = [0.744; 0.573; 0.344]^T$$

7.2 SALSA

A SALSA (Stochastic Approach for Link Structure Analyis) módszer a tekintély és elosztó jelleg megállapításának másik módszere [23.]. A PageRank és a HITS módszerek egy kombinációja. Érvényes benne a HITS alapelve, de a tekintély és elosztó jelleget kifejező súlyszámokra a HITS értékektől általában különböző értékeket szolgáltat. A SALSA módszer is egy szomszédossági gráfot alkalmaz. A gráfban tekintély és elosztó csúcsok vannak élekkel összekötve. A SALSA algoritmus egy véletlen sétát valósít meg az alább definiált gráfokon, amely az eredeti gráf pontjainak gyűjtőlap és tekintély tulajdonságait emeli ki .

A súlyoknak a kiszámítási módja a következő. Jelölje $W=(w_{ij})_{n\times n}$ a tekintett Web-gráf adjacencia mátrixát. Legyenek $W_r = (r_{ij})$ illetve $W_c = (c_{ij})$ azok a mátrixok, amelyeket a W-ből nyerünk úgy, hogy a W sorait illetve oszlopait elosztjuk azok összegével, azaz

$$r_{ij} = \frac{w_{ij}}{\sum_{j=1}^{n} w_{ij}} \quad ; \quad \sum_{j=1}^{n} w_{ij} \neq 0 \,; i,\ j = 1, \ldots, N$$

$$c_{ij} = \frac{w_{ij}}{\sum_{i=1}^{n} w_{ij}} \quad ; \quad \sum_{i=1}^{n} w_{ij} \neq 0 \,; i,\ j = 1, \ldots, N$$

Vezessük be a H és az A, elosztó- illetve tekintély-mátrixokat (*hub* and *authority matrix*) a következőképpen:

$$H = W_r \times W_c^T$$

$$A = W_c^T \times W_r$$

Az elosztó súlyok és a tekintély súlyok a H illetve az A mátrixok domináns sajátvektorai. Az oldalakat a tekintély típuson belül tekintély súly szerint csökkenő, az elosztó típuson belül pedig elosztó súly szerint csökkenő sorrendben rangsorolják.

Példa

A Mini-web ábrát felhasználva kapjuk a következő mátrixokat:

$$W = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad W_r = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0 & 1 & 0 \end{pmatrix} \quad W_c = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 1 & 0 & 0.5 \\ 0 & 0.5 & 0 \end{pmatrix}$$

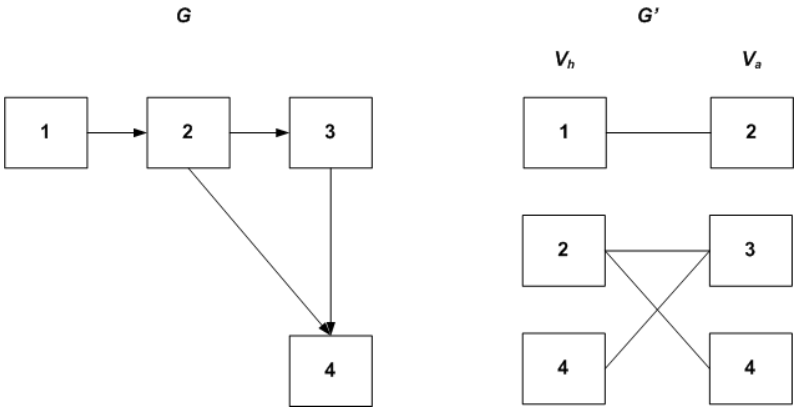
Ekkor a H és A mátrixok a következők:

$$H = \begin{pmatrix} 0.5 & 0.25 & 0.25 \\ 0.25 & 0.75 & 0 \\ 0.5 & 0 & 0.5 \end{pmatrix} \quad A = \begin{pmatrix} 0.5 & 0 & 0.5 \\ 0 & 0.75 & 0.25 \\ 0.25 & 0.25 & 0.5 \end{pmatrix}$$

A H mátrix domináns sajátértéke 1. Az elosztó súlyok pedig: $[0.577\ 0.577\ 0.577]^{T.}$ Az A mátrix domináns sajátértéke 1. A tekintély súlyok pedig: $[0.577\ 0.577\ 0.577]^{T.}$ Látható, hogy ebben a példában a SALSA módszer szerint valamennyi oldal egyformán elosztó és tekintély típusú.

A H és az A mátrixok elemeinek eredetileg javasolt kiszámítási módja a következő. A szomszédossági $G = (V, E)$ Web-gráf alapján $G' = (V_h, V_a, E')$ páros gráfot képezünk a következőképpen. Egy halmaz tartalmazza az elosztó oldalakat, egy másik halmaz pedig a tekintély oldalakat. Egy weboldal mindkét halmazban előfordulhat. A halmazok megadása legyen a következő:

- $V_h = \{s \mid s \in V, \text{ki-fok}(s) > 0\}$, elosztó oldal (hub side);
- $V_a = \{s \mid s \in V, \text{be-fok}(s) > 0\}$, tekintély oldal (authority side);
- $E' = \{(s, r) \mid (s, r) \in E\}$.



39. ábra. G gráf és G' páros gráf

Feltételezzük, hogy a G gráf összefüggő (jóllehet ennek hiánya a gyakorlatban nem jelent különösebb akadályt, mert a G'-t a G valamennyi összefüggő részgráfjára meg lehet építeni külön-

külön). A páros gráf alapján két mátrix definiálható. Az A mátrix egy tekintély típusú Markov lánc. A H mátrix egy elosztó típusú Markov lánc.

Az A mátrix a_{ij} elemét a következőképpen értelmezzük:

$$a_{ij} = \sum_{\{k|(k,i),(k,j)\in E'\}} \left(\frac{1}{deg(i \in V_a)} \times \frac{1}{deg(i \in V_h)}\right)$$

A H mátrix h_{ij} elemét a következőképpen értelmezzük:

$$h_{ij} = \sum_{\{k|(i,k),(j,k)\in E'\}} \left(\frac{1}{deg(i \in V_h)} \times \frac{1}{deg(i \in V_a)}\right)$$

Irodalomjegyzék

20. LEIGHTON, H. Vernon; SRIVASTAVA, Jaideep. First 20 precision among World Wide Web search services(search engines). *Journal of the American Society for Information Science*, 1999, 50.10: 870-881.
21. BRIN, Sergey; PAGE, Lawrence. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 1998, 30.1: 107-117.
22. KLEINBERG, Jon M. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 1999, 46.5: 604-632.
23. LEMPEL, Ronny; MORAN, Shlomo. SALSA: the stochastic approach for link-structure analysis. *ACM Transactions on Information Systems* (TOIS), 2001, 19.2: 131-160.
24. CLEVERDON, Cyril. The Cranfield tests on index language devices. In: *Aslib proceedings*. MCB UP Ltd, 1967. p. 173-194.
25. KEKALAINEN, Jaana. Binary and graded relevance in IR evaluations — Comparison of the effects on ranking of IR systems. *Information Processing and Management,* 2005, 41: 1019–1033.