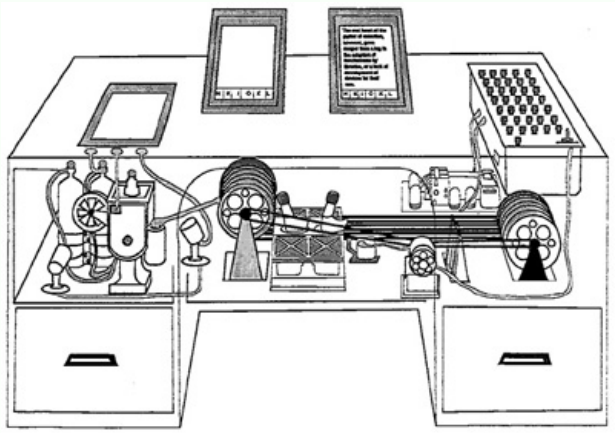
Információ-visszakereső rendszerek

Bodó Zalán



Tartalom

Információ-visszakeresés

Adottak:

- adathalmaz természetes nyelven (cikkek, szövegek)
- kereső-kifejezés, kérés, kérdés (angolul *query*)

Feladat:

- keressük a kereső-kifejezésnek megfelelő, releváns szövegeket, dokumentumokat

Webes keresőmotorok:

- Google
- Bing (Microsoft)
- Yahoo!
- ...

Rendszerek osztályozása

- Boole-féle modell
- Vektortér modell
- (Valószínűségi modellek)
- ...

Boole-féle modell

- csak azt veszi figyelembe, hogy egy szó szerepel-e vagy sem: 0 – nem szerepel, 1 – szerepel
- naiv módszer

Jelölések:

- indexelő szavak: $T = \{t_1, t_2, \ldots, t_d\}$
- dokumentumok halmaza: $D = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\}$
- $\Rightarrow$ dokumentum: $\boldsymbol{d}_i = (0\ 1\ 0\ \ldots 1)'$ vagy $\boldsymbol{d}_i = \{t_i \mid t_i \in T \cap d_i\}$
- kérés: $Q = \bigwedge_{k \in K} \left(\bigvee_{j \in J_k} q_j \right)$, ahol $q_j \in \{t_j, \overline{t_j}\}$

Visszakeresés:

1. meghatározzuk a dokumentumhalmazokat, melyekben q_j szerepel:
$$S_j = \{\boldsymbol{d}_i \mid q_j \in \boldsymbol{d}_i\}$$
2. Meghatározzuk a visszatérítendő dokumentumokat:
$$\bigcap_{k \in K} \left(\bigcup_{j \in J_k} S_j \right)$$

Példa

$d_1 =$ 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 indexelő szavak pedig legyenek:
$T = \{t_1, t_2, t_3, t_4, t_5, t_6\} = \{$információ-visszakeresés, tudományág, Boole-féle, implementálás, memória, lemez$\}$
A dokumentumok:

$$\begin{aligned} \boldsymbol{d}_1 &= \{t_1, t_2\} \\ \boldsymbol{d}_2 &= \{t_1, t_3, t_4\} \\ \boldsymbol{d}_3 &= \{t_1, t_4, t_5, t_6\} \end{aligned}$$

A kérés: $Q =$ információ-visszakeresés $\wedge$ Boole-féle
A halmazok meghatározása:

$$\begin{aligned} S_1 &= \{\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3\} \\ S_2 &= \{\boldsymbol{d}_2\} \end{aligned}$$

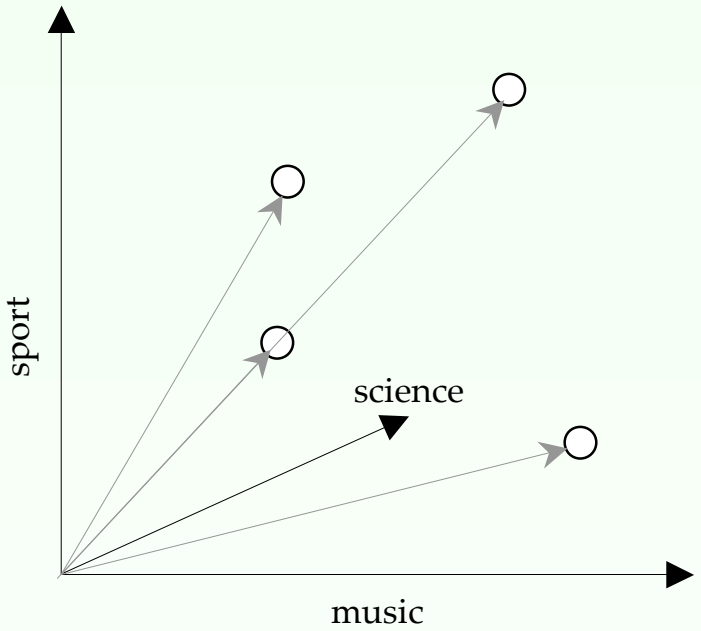
A visszatérített dokumentum(ok):

$$\{\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3\} \cap \{\boldsymbol{d}_2\} = \{\boldsymbol{d}_2\}$$

A modell hátrányai

- nem képes “megközelítő” eredményt szolgáltatni: a kereső-kifejezésben szereplő minden szónak szerepelnie kell a visszatérített dokumentumban
- az eredményeket nem lehet rangsorolni: minden egyes visszatérített dokumentum ugyanolyan “jó”

Vektortér modell



Jelölések:

- indexelő szavak: $T = \{t_1, t_2, \ldots, t_d\}$
- dokumentumok halmaza: $D = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\}$
- $\Rightarrow$ dokumentum: $\boldsymbol{d}_i = (w_{1i}\ w_{2i}\ \ldots\ w_{di})'$
  - w_{ki} = a k-adik szó súlya az i-edik dokumentumban
- kérés: $\boldsymbol{q} = (w_{1q}\ w_{2q}\ \ldots\ w_{dq})'$
- súly: w_{ij} = hányszor fordult elő az i-edik szó a j-edik dokumentumban
- megjegyzés: vannak más, jobb súlyozási módszerek

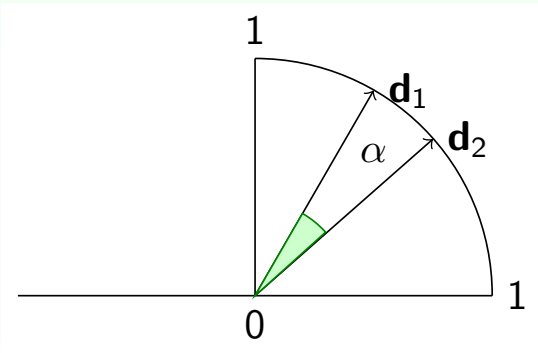
Visszakeresés:

- vektoros hasonlósági metrika alapján
- például *koszinusz-hasonlóság*:

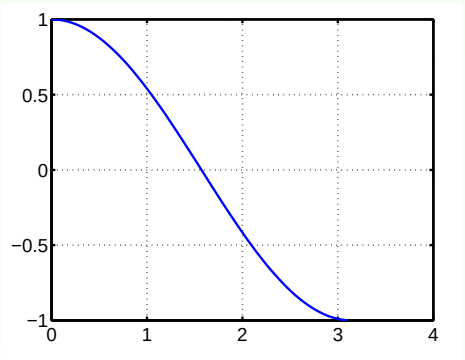
$$s(\boldsymbol{d}_i, \boldsymbol{q}) = \frac{\boldsymbol{d}_i' \cdot \boldsymbol{q}}{\|\boldsymbol{d}_i\| \cdot \|\boldsymbol{q}\|}$$

ahol $\boldsymbol{d}_i' \cdot \boldsymbol{q} = \sum_{j=1}^{d} w_{ji} \cdot w_{jq}$ és $\|\boldsymbol{d}_i\| = \sqrt{\sum_{j=1}^{d} w_{ji}^2}$

Koszinusz hasonlóság



(a)



(b)

(a) (normalizált) skalárszorzat = a dokumentumvektorok által bezárt szög koszinusza;
(b) koszinusz függvény a $[0; \pi/2]$ intervallumon

Más hasonlósági metrikák

	halmazok esetén	**vektorok esetén**
illeszkedési együttható	$\lvert A \cap B \rvert$	$\boldsymbol{x}' \cdot \boldsymbol{y}$
Jaccard-együttható	$\frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$	$\frac{\boldsymbol{x}' \cdot \boldsymbol{y}}{\lVert \boldsymbol{x} \rVert^2 + \lVert \boldsymbol{y} \rVert^2 - \boldsymbol{x}' \cdot \boldsymbol{y}}$
átfedési együttható	$\frac{\lvert A \cap B \rvert}{\min(\lvert A \rvert, \lvert B \rvert)}$	$\frac{\boldsymbol{x}' \cdot \boldsymbol{y}}{\min(\lVert \boldsymbol{x} \rVert^2, \lVert \boldsymbol{y} \rVert^2)}$
Dice-együttható	$\frac{2 \cdot \lvert A \cap B \rvert}{\lvert A \rvert + \lvert B \rvert}$	$\frac{2 \cdot \boldsymbol{x}' \cdot \boldsymbol{y}}{\lVert \boldsymbol{x} \rVert^2 + \lVert \boldsymbol{y} \rVert^2}$

Példa

$d_1 =$ 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 indexelő szavak:
$T = \{t_1, t_2, t_3, t_4, t_5, t_6\} =$ {információ-visszakeresés, tudományág, Boole-féle, implementálás, memória, lemez}
A dokumentumok:

$$\begin{aligned} \boldsymbol{d}_1 &= (1\ 1\ 0\ 0\ 0\ 0)' \\ \boldsymbol{d}_2 &= (1\ 0\ 1\ 1\ 0\ 0)' \\ \boldsymbol{d}_3 &= (1\ 0\ 0\ 2\ 1\ 1)' \end{aligned}$$

A kérés: $Q =$ információ-visszakeresés $\wedge$ rendszer $\wedge$ implementálás
$\Rightarrow \boldsymbol{q} = (1\ 0\ 0\ 1\ 0\ 0)'$
(Koszinusz hasonlóságot használunk)

$$\begin{aligned} s(\boldsymbol{d}_1, \boldsymbol{q}) &= \frac{1}{\sqrt{2}\sqrt{2}} = \frac{1}{2} = 0.5 \\ s(\boldsymbol{d}_2, \boldsymbol{q}) &= \frac{2}{\sqrt{3}\sqrt{2}} = \frac{2}{\sqrt{6}} = 0.8165 \\ s(\boldsymbol{d}_3, \boldsymbol{q}) &= \frac{3}{\sqrt{7}\sqrt{2}} = \frac{3}{\sqrt{14}} = 0.8018 \end{aligned}$$

Megjegyzések

- a visszatérített dokumentumok rangsorolhatók
- szavak egymástól való függetlenségének feltételezése
- szinonímák, hiponímia, hiperonímia stb. figyelmen kívül hagyása
- nem keressük a mondatok értelmét/szemantikáját

Webes keresés

- információ-visszakeresési feladat
- általában két hasonlósággal dolgozik:
  - tartalom alapú hasonlóság
  - népszerűségi érték (nem függ a tartalomtól)
- keresőrendszerek elemei:
  - crawler
  - szövegfeldolgozó, indexelő
  - kereső
  - rangsoroló

Crawlerek

- más megnevezések: searchbot, spider, knowbot, walker, . . .
- feladatuk: bejárni a webet (inkább annak egy részét) és az oldalak tartalmát átadni a szövegfeldolgozó egységnek
- lásd: `robots.txt` a weboldal könyvtárhierarchiájában (REP – *Robots Exclusion Protocol*)
  Példa:
  User-agent: * # bármilyen robot
  Disallow: /cgi-bin/ # ezeket nem engedjük indexelni
  Disallow: /images/
  Disallow: /tmp/
  Disallow: /private/

Szövegek feldolgozása, invertált index

Szövegek feldolgozása:

- a dokumentum szövegének szavakra bontása
- írásjelek, speciális szimbólumok elhagyása
- szótövesítés = szavak szótövének meghatározása, és azzal való helyettesítese
- szavak törlése a *stopszavak listája* alapján
  Stopszavak lehetnek:
  ahogy, ahol, aki, akik, akkor, alatt, által, általában, amely, amelyek, amelyekben, amelyeket, amelyet, amelynek, ami, amit, amolyan, amíg, amikor, át, abban, ahhoz, annak, arra, arról, az stb.

Példa:
Az információ-visszakeresés egy nagyon gyorsan és dinamikusan fejlődő tudományág
$\Rightarrow$ információ-visszakeresés gyors dinamikus fejlődik tudományág

Invertált index:

- a kérés és a dokumentum/oldal hasonlóságának gyors kiszámításához
- egyszerű modell:

  $$\text{szó/szótő} \longrightarrow \text{oldalszám}_1, \text{oldalszám}_2, \ldots, \text{oldalszám}_k$$

  Például: dinamikus $\longrightarrow 11, 126, 11\,985, 1\,005\,256$
- "okosabb" modell:

  $$\text{szó/szótő} \longrightarrow \text{oldalszám}_1[\#\text{címben},\#\text{leírásban},\#\text{oldalon}], \ldots$$

  Például: dinamikus $\longrightarrow 11[1,0,7], 126[0,1,2], 11\,985[2,1,5]$

Keresés

- tartalom alapú hasonlósági metrika használata

Példa:
dinamikus $\longrightarrow 3, 15, 19, 94, 101, 673, 1199$
rendszer $\longrightarrow 3, 31, 56, 94, 673, 909, 11\,114, 253\,791$
Egyszerű (bináris) keresés:

3. oldal	2
94. oldal	2
673. oldal	2

Okosabb keresés:
dinamikus $\longrightarrow 3[1, 1, 27], 94[1, 0, 7], 673[0, 0, 3]$
rendszer $\longrightarrow 3[1, 1, 10], 94[0, 0, 5], 673[1, 1, 14]$

3. oldal	$(1 + 1 + 27) \cdot (1 + 1 + 10) = 348$
94. oldal	$(1 + 0 + 7) \cdot (0 + 0 + 5) = 40$
673. oldal	$(0 + 0 + 3) \cdot (1 + 1 + 14) = 48$

Rangsorolás

- rang = népszerűségi érték (nem függ a tartalomtól)
- citációs elemzés: ki mennyit hivatkozik az adott oldalra/dokumentumra
- híres algoritmus: PageRank (Larry Page, Sergey Brin 1998 [Google])

PageRank

- a rangok rekurzív definíciója: egy oldalnak magas rangja van, ha sok oldal vagy kevés de magas rangú oldal mutat rá

$$\boldsymbol{r}_i = \sum_{j \in N^{-1}(i)} \frac{\boldsymbol{r}_j}{|N(j)|}$$

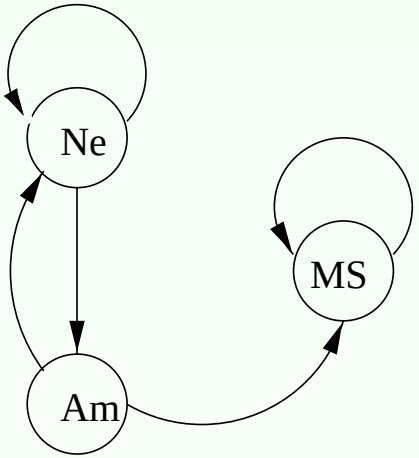
  ahol $\boldsymbol{r}$ a rangok vektora, $\boldsymbol{r}_i$ annak i-edik eleme, $N^{-1}(i)$ az i-edik oldalra mutató oldalak száma
- egyszerűbb formában:

$$\boldsymbol{r} = \boldsymbol{P} \cdot \boldsymbol{r}$$

  ahol $\boldsymbol{P}$ egy súlyozott szomszédsági mátrix, $P_{ij} = 1/|N(j)|, \forall j \in N^{-1}(i)$
- véletlen bolyongás egy gráfon; a rangok annak valószínűségei, hogy k-adik lépésben az adott oldalon vagyunk
- ha k elégségesen nagy, akkor egy egyedi eloszláshoz tart
- probléma: csomópontok vagy csoportok, ahonnan nincsenek kifelé vezető linkek
- megoldás: véletlenszerűség hozzáadása a szörfözéshez:

$$\boldsymbol{r} = c\boldsymbol{P}'\boldsymbol{r} + (1-c)\boldsymbol{1}, \quad c \in (0,1]$$

Példa



$$\begin{pmatrix} n \\ m \\ a \end{pmatrix} = \begin{pmatrix} 1/2 & 0 & 1/2 \\ 0 & 1 & 1/2 \\ 1/2 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} n \\ m \\ a \end{pmatrix}$$

"Buta" PageRank (40 iteráció):

$$\begin{array}{lllllll} n & = & 1 & 0.75 & 0.625 & \ldots & 0.0002 \\ m & = & 1.5 & 1.75 & 2 & \ldots & 2.9996 \\ a & = & 0.5 & 0.5 & 0.375 & \ldots & 0.0002 \end{array}$$

"Okos" PageRank ($c = 0.8$, 30 iteráció):

$$\begin{array}{lllllll} n & = & 1 & 0.84 & 0.776 & \ldots & 0.6364 \\ m & = & 1.4 & 1.56 & 1.688 & \ldots & 1.9091 \\ a & = & 0.6 & 0.6 & 0.536 & \ldots & 0.4545 \end{array}$$

Rangsorolás = értékek összevonása:

tartalom alapú hasonlóság $\times$ népszerűségi érték

Köszönöm a figyelmet!