Vektortér modell (egy klasszikus információ-visszakereső modell)

3. ELŐADÁS

1. A vektortér modell.
2. Indexelési technikák, hasonlósági mértékek a vektortér modellben.
3. A rangsortartás.

1. Vektortér modellek

1.1. Vektortér modell helye az információ-visszakereső modellek közt

1.2. Vektortér modell formális leírása

1.3. Hatványtörvény-Zipf törvény

Vektorok normalizálása (*Normalization*)

- vektor iránya nem változik,
- viszont a hossza egy lesz.
- A normalizált vektort úgy kaphatjuk meg, hogy az eredeti vektort elosztjuk a hosszával.

1.1. Vektortér modell az információ-visszakereső modellek közt -1

Az információ-visszakeresés alapelemei:

- dokumentum (document),
- kérdés (query),
- relevancia (relevance),
- visszakeresés (retrieval).

1.1. Vektortér modell az információ-visszakereső modellek közt -2

Attól függően, hogy :

- a dokumentumokat,
- a kérdést, és
- a visszakeresést

hogyan modellezzük többféle információ-visszakereső modellt (information retrieval models) különböztetünk meg.

1.1. Vektortér modell az információ-visszakereső modellek közt -3

Klasszikus modellek tulajdonságai:

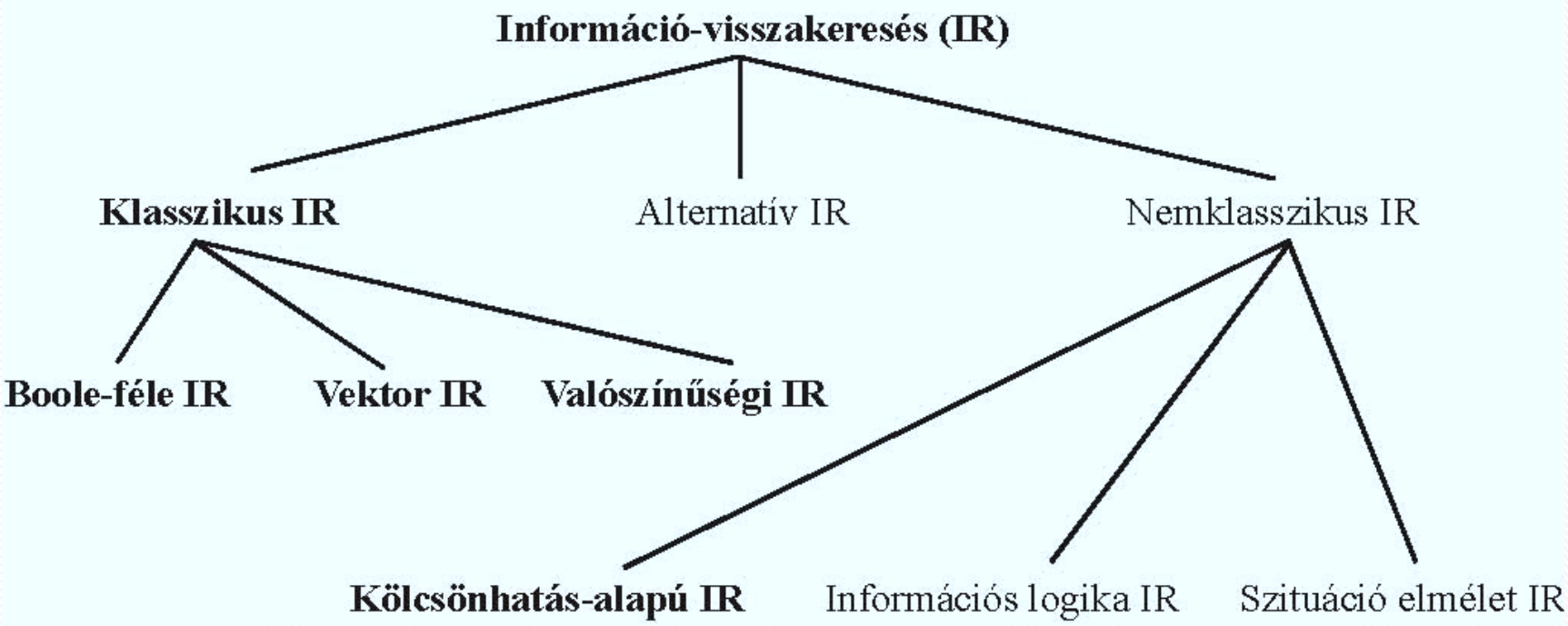
- Első (hagyományos) modellek,
- Matematikai módszereken alapulnak,
- *Kérdés* (*Q*)és a *Dokumentum (D)* távolságának matematikai mérésén alapul
- Mintaillesztési, illetve távolság-alapú modellek
- Könnyű implementálási lehetőség
- Kereskedelmi keresők ezek speciális változatain alapulnak

1.1. Vektortér modell az információ-visszakereső modellek közt -4

Klasszikus információ-visszakereső modellek :

- Boole modell (Boolean Model), matematikai logikán és halmazelméleten alapul
- **Vektortér modell (Vector Space Model), a lineáris algebrán alapul**
- Valószínűségi modell (Probablistic Model), a valószínűségszámításon és a Bayes-statisztikán alapul

1.1. Vektortér modell az információ-visszakereső modellek közt -5



1.2. Vektortér modell formális leírása -1.

A vektortér modell:

- egy fontos,
- jól érthető, és
- széles körben kutatott és használt klasszikus modell
- amelyet szöveges objektumok feldolgozására, és információ-visszakeresésre már régóta használnak (Salton, 1966).

1.2. Vektortér modell formális leírása -2.

Ezt vektortér modellnek nevezik,

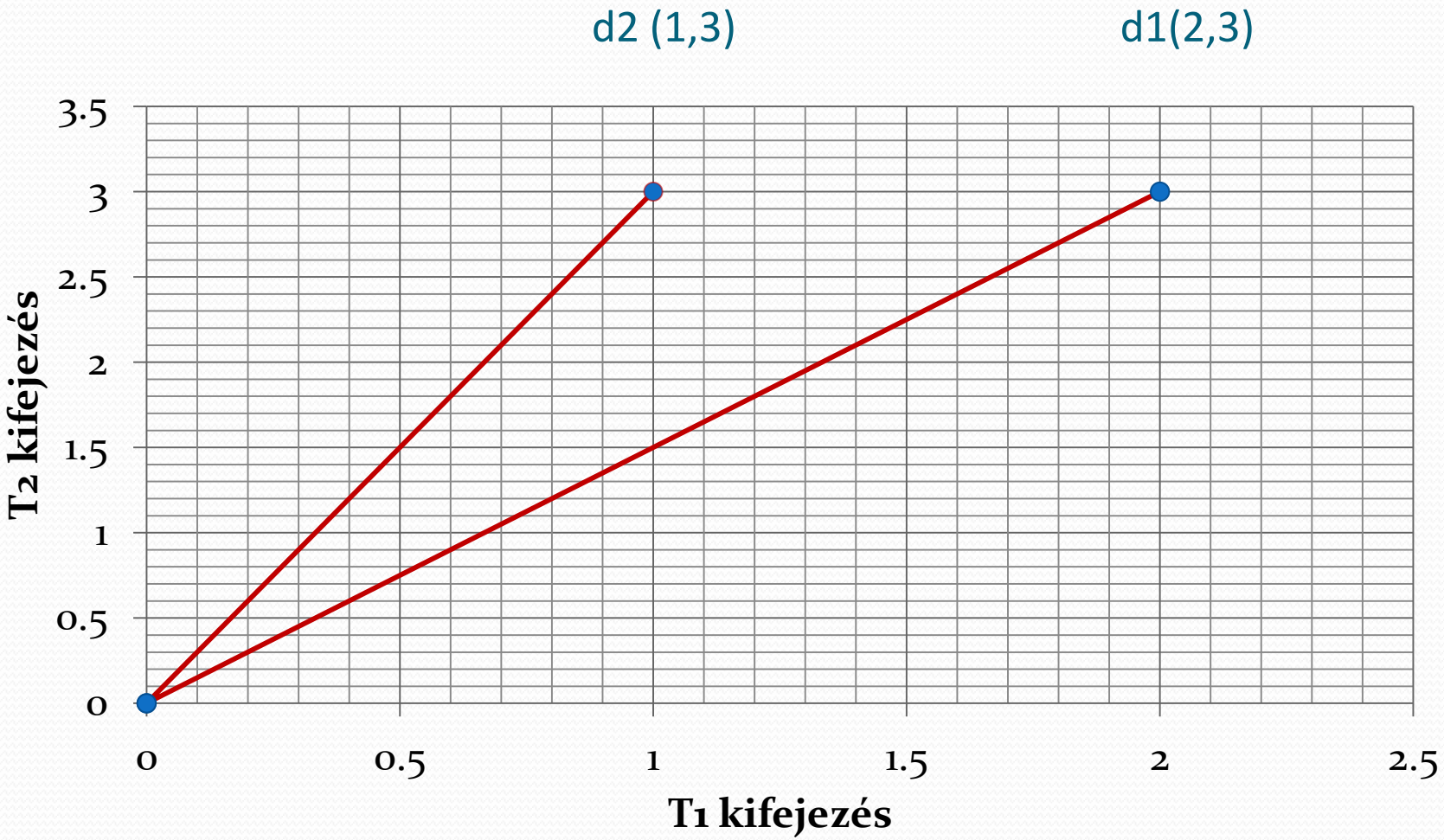
- mert minden
  - dokumentum és a
  - kérdés is a

  tér egy pontjába van leképezve,
- amely tér alapját a dokumentumokban található kifejezések adják.

1.2. Vektortér modell formális leírása -3.

- A tér matematika modellje:
  - egy orthonormált euklideszi tér,
  - amelyben a tengelyek páronként egymásra merőlegesek.
- A tér dimenzióit az indexkifejezések adják.
- A visszakeresés azon alapul, hogy a
  - kérdés-vektor és a
  - dokumentum-vektor

  mennyire van „közel” egymáshoz.


d2 (1,3)
d1(2,3)
3.5
3
2.5
2
1.5
1
0.5
0
T2 kifejezés
0
0.5
1
1.5
2
2.5
T1 kifejezés

1.2. Vektortér modell formális leírása -4.

Legyen

- D egy véges halmaz, melynek elemei a dokumentumok:

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

- T egy véges halmaz, melynek elemei az indexkifejezések:

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

- Minden D_j dokumentumhoz hozzárendelünk egy n hosszú $\boldsymbol{v}_j$ súlyvektort. A vektor elemeit súlyoknak nevezzük:

  $$\mathbf{v}_j = (w_{ij})_{i=1,\ldots,n} = (w_{1j}, \ldots, w_{ij}, \ldots, w_{nj})$$

  - általában $0 \leq w_{ij} \leq 1$
  - a w_{ij} súllyal azt fejezzük ki, hogy a t_i kifejezés milyen mértékben tükrözi a D_j dokumentum tartalmát.

1.2. Vektortér modell formális leírása -4.

A $\boldsymbol{v}_j$ súlyvektorokból megadható a

TxD (term-by-document) kifejezés-dokumentum mátrix:

- m (dokumentumok száma) oszlopa van
- n (indexkifejezések száma) sora van
- amelynek elemei a súlyok,
- TD= (w_{ij})nxm , ahol i=1…n, j=1…m

1.2. Vektortér modell formális leírása -5.

- TD Mátrix

$$\begin{pmatrix} w_{11} & \cdots & w_{1j} & \cdots & w_{1m} \\ \vdots & & \vdots & & \vdots \\ w_{i1} & \cdots & w_{ij} & \cdots & w_{im} \\ \vdots & & \vdots & & \vdots \\ w_{n1} & \cdots & w_{nj} & \cdots & w_{nm} \end{pmatrix}$$

1.2. Vektortér modell formális leírása -6.

A kifejezések kiválasztása és a súlyok meghatározása :

- nehéz elméleti (nyelvészeti, szemantikai) és
- gyakorlati probléma.

Ennek számos lehetséges megoldása van.

- A legnyilvánvalóbb az, hogy az index- kifejezéseket magukban a dokumentumokban keressük.
- Feltételezzük, hogy a szavak előfordulási gyakorisága a dokumentumokban jelentőséggel bír, és ezért azonosítóként használhatók.

1.2. Vektortér modell formális leírása -7.

Indexkifejezések meghatározása:

- automatikus (a dokumentumból)
- manuális (szakértők által).

Hatványtörvény

- Fokszám: hálózat egy elemének fokszáma a hálózaton belüli kapcsolatainak a száma
- Fokszámeloszlás: a hálózat összes, adott fokszámú elemének számát tünteti fel a fokszám függvényében
  - Random-gráfok (olyan hálózat, amelynek elemeit véletlenszerűen kötjük össze): Poisson-eloszlás
  - Skálafüggetlen: hálózat fokszámeloszlása hatványfüggvényt követ

Hatványtörvény

- **Véletlen hálózat** fokszáma Poisson-eloszlást követ, (haranggörbe).
  - A legtöbb csomópontnak azonos számú kapcsolata van,
  - nem létezik kiemelkedően sok kapcsolatú csomópont
- **Skálafüggetlen** hálózat hatványfüggvényű fokszámeloszlású
  - legtöbb csomópontnak csupán kevés kapcsolata van,
  - amelyeket néhány nagymértékben összekapcsolt középpont tart össze.(Barabási, 2003 után)

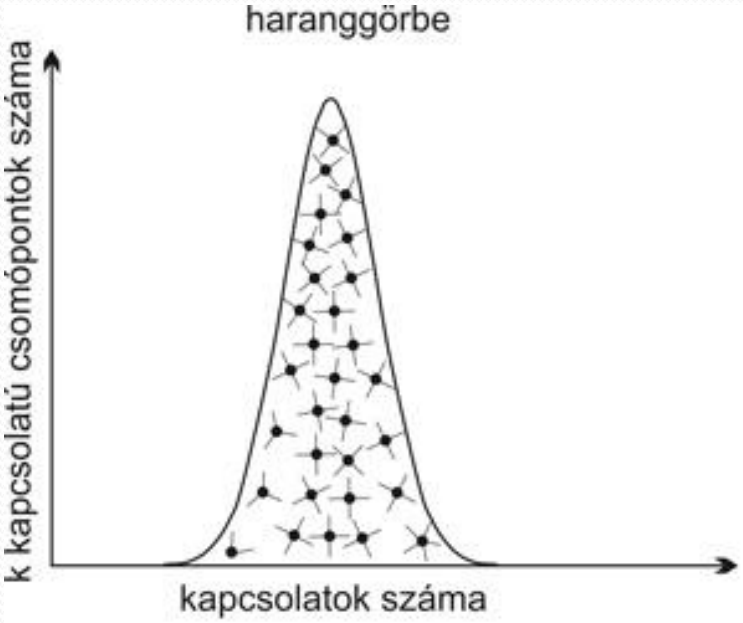
Hatványtörvény

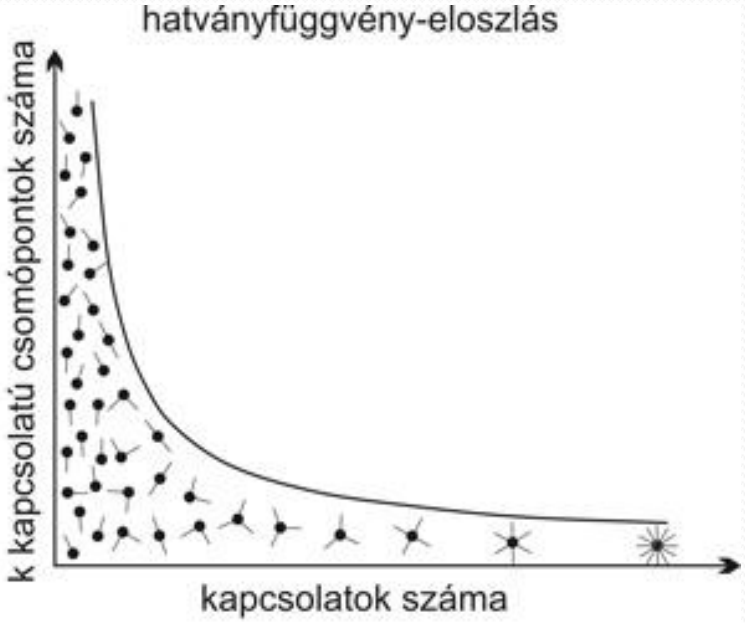
Ha egy esemény változása valamely jellemzőjének hatványával arányos, akkor azt mondjuk, hogy a hatványtörvény szerint viselkedik.

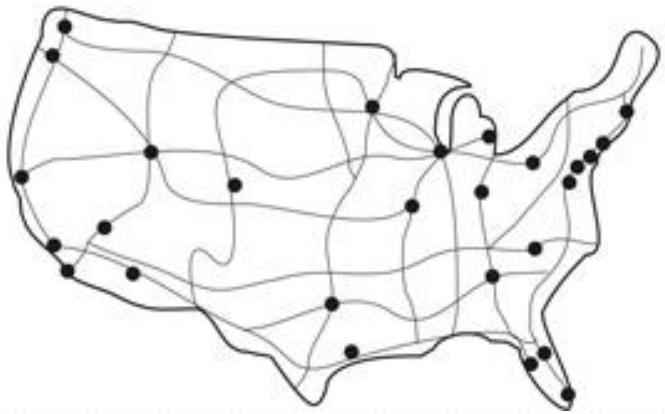
A hatványtörvénnyel leírt hálózatok esetében az elemek fokszám-eloszlása szabályszerű:

- van kis számú elem, aminek nagyon sok kapcsolata van (pl. 2 db 100 kapcsolattal rendelkező), (kevés amire sok link mutat)
- majd a kapcsolatok számának csökkenésével növekszik az adott kapcsolattal rendelkező elemek száma (pl. 4 db 80 kapcsolattal, 10 db 50 kapcsolattal, 30 db 20 kapcsolattal 80 db 5 kapcsolattal, 130 db 4 kapcsolattal, stb.) (sok amire kevés link mutat).

Azon hálózatokat, amelyeknél az elemek fokszám-eloszlása a hatványtörvényt követi, skálafüggetlen hálózatoknak nevezzük (Barabási 2003, Newman 2005).


haranggörbe
k kapcsolatú csomópontok száma
kapcsolatok száma


hatványfüggvény-eloszlás
k kapcsolatú csomópontok száma
kapcsolatok száma



Hatványtörvény

Pl. Barabási:

203 millió weboldal vizsgálata (bejövő linkek alapján):

- Oldalak 90 %-a **10 vagy kevesebb** beérkező linkkel rendelkezik
- Néhányat (3) közel **1 millió** másik oldalon hivatkoznak

ZIPF-törvény

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

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

  - ahol *C* korpusz függő konstans,
  - *r a szavak rangsora (*az előfordulási szám szerinti csökkenő sorrendben elfoglalt hely *)*.
  - α a hatványfüggvény kitevője.
- Az $f(r) = Cr^{-1}$ hatványtörvényt Zipf törvénynek is nevezik.

ZIPF-törvény

Sok, eddig publikált kísérleti eredmény is igazolja, azt a feltételezést, hogy minden nyelvre érvényes.

A leggyakoribb szó

- közel kétszer gyakoribb, mint a második leggyakoribb szó, és
- háromszor gyakoribb, mint a harmadik helyen lévő, stb.

ZIPF-törvény példa

Szakirodalmi, hivatalos példa:

- az úgynevezett Brown-gyűjteményt (Brown University-ben kb. 500 angol szöveget vizsgáltak meg a nyelvészek), ahol:
  - a “the” a leggyakrabban előforduló szó (az összes előforduló szó 7%-a) .
  - “and” a második leggyakoribb szó, amelynek az előfordulási gyakorisága 3,5%.

ZIPF-törvény

- A törvény érdekes következménye, ha egy korpuszból csak a leggyakoribb szavakat tartjuk meg, a többit töröljük, a korpusz nagy része akkor is megmarad.

Pl. 30 000 különböző szavunk van, és $\alpha=1.1$, és

- a 15 000 leggyakrabban előforduló szót tartjuk meg (szótár a felére csökken) , akkor a korpuszban levő szavak több,mint 96%-át megtartottuk.
- 1 000 leggyakoribb szót tartjuk meg (szótár a 30-adára csökken) akkor ugyanez az arány majdnem 80%.

2. Indexelési technikák, hasonlósági mértékek a vektortér modellben

2.1. Automatikus Indexkifejezés- kinyerés lépései

2.2. Indexelési technikák

2.3. A visszakeresés lépései

2.4. Hasonlósági mértékek

2.1. Automatikus Indexkifejezés-kinyerés lépései -1.

A kifejezések, és azok jelentőségének meghatározására a következő egyszerű automatikus módszer használható:

1. Lexikai egységek azonosítása. Egy számítógépes program kell szavak felismerésére (szó = karaktersorozat, amelyet szóköz, írásjel előz meg és követ).
2. Stoplista alkalmazása (azokat a szavakat tartalmazza, amelyek általában nem hordoznak jelentést egy dokumentumban, pl.: a, az, egy, ez, ...). Azokat a szavakat, amelyeket a stoplista tartalmaz, kihagyjuk a további vizsgálatkor. A stoplista általában terület- és applikáció-függő.

2.1. Automatikus Indexkifejezés-kinyerés lépései -2.

3. Szótővesítő (stemming) algoritmus alkalmazása. Ez az algoritmus minden szót redukál vagy áttranszformál a nyelvi szótőre.
4. Kiszámítjuk minden D*j* dokumentumra a t*i* kifejezés előfordulásainak számát: f*ij*
5. Kiszámítjuk a t*i* kifejezés összes előfordulását: t_{fi}

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

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

2.1. Automatikus Indexkifejezés-kinyerés lépései -3.

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

2.2. Indexelési technikák -1.

- A súlyszámok meghatározásának technikái:
- Bináris:

$$w_{ij} = \begin{cases} 1, ha\ a\ t_i kifejezés\ szerepel\ a\ D_j dokumentumban \\ 0, egyébként \end{cases}$$

- Gyakoriság szerinti súlyozás (*TF*: Term-frequency):

A súlyfüggvény megegyezik a kifejezések előfordulási gyakoriságával.

$$w_{ij} = f_{ij}$$

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

2.2. Indexelési technikák -2.

- Normalizált gyakoriság szerinti súlyozások:
  - Maxnormált (*MaxNorm*):

$$w_{ij} = \frac{f_{ij}}{\max\limits_{1 \le k \le n} f_{kj}}$$

  - Hossznormált (*tfn*: term-frequency normalised):

$$w_{ij} = \frac{f_{ij}}{\sqrt{\sum_{k=1}^{n} {f_{kj}}^2}}$$

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

2.2. Indexelési technikák -3.

- *TF-IDF* (Term-frequency-inverse document frequency):

$$w_{ij}=f_{ij} \times log\left(\frac{m}{F_i}\right), ahol:$$

- f_{ij}: t_i kifejezés előfordulásainak száma a Dj dokumentumban
- m: dokumentumok száma
- F_i: azon dokumentumok száma, amelyekben előfordul a t_i indexkifejezés

Indexelési technikák példa

- D1 (2,4); D2(1,4); D3(0,1)
- Bináris:

  v1(1,1); v2(1,1); v3(0,1)
- Gyakoriság:

  v1(2,4); v2(1,4); v3(0,1)
- Maxnormált: $w_{ij} = \frac{f_{ij}}{\max\limits_{1 \le k \le n} f_{kj}}$

  v1(0.5,1); v2(0.25,1); v3(0,1)

Indexelési technikák példa

- D1 (2,4); D2(1,4); D3(0,1)
- Hossz-normált: $w_{ij} = \frac{f_{ij}}{\sqrt{\sum_{k=1}^{n} {f_{kj}}^2}}$

  $v1(^2/_{\sqrt{20}}, ^4/_{\sqrt{20}})$; $v2(^1/_{\sqrt{17}}; ^4/_{\sqrt{17}})$; v3(0,1)
- TF-IDF: $w_{ij}=f_{ij} \times log\left(\frac{m}{F_i}\right)$

  v1(0.34,0); v2(0.17,0); v3 (0,0)

2.3. A visszakeresés lépései-1.

- A visszakeresés azon alapul, hogy a kérdés-vektor és a dokumentum-vektor mennyire van „közel” egymáshoz.
- A felhasználó által feltett Q_k keresőkérdéshez is megadható egy $\mathbf{v}_k$ vektor, amelynek elemeit ugyanolyan súlyszámítási séma alapján adjuk meg, mint a dokumentumokét:

$$\mathbf{v}_k = (w_{ik})_{i=1,\ldots,n} = (w_{1k}, \ldots, w_{ik}, \ldots, w_{nk}),$$ ahol:

w_{ik} : *a* t_i indexkifejezés Q_k keresőkérdésre vonatkozó súlyszáma

2.3. A visszakeresés lépései-2.

- A visszakeresés egy hasonlóság meghatározásán alapul.
  - A dokumentumoknak megfelelő $\boldsymbol{v}_i$ vektorokat összevetjük a kérdés vektorral $\boldsymbol{v}_k$ (hasonlóság-mérés). Ennek eredménye s_{ik}.
  - Ha ez a mérési eredmény egy küszöbértéknél nagyobb, akkor $\boldsymbol{v}_i$ vektorral azonosított dokumentum válasz a kérdésre.

$$S_{ik} = s(\mathbf{v}_i, \mathbf{v}_k) > K$$

2.3. A visszakeresés lépései -3.

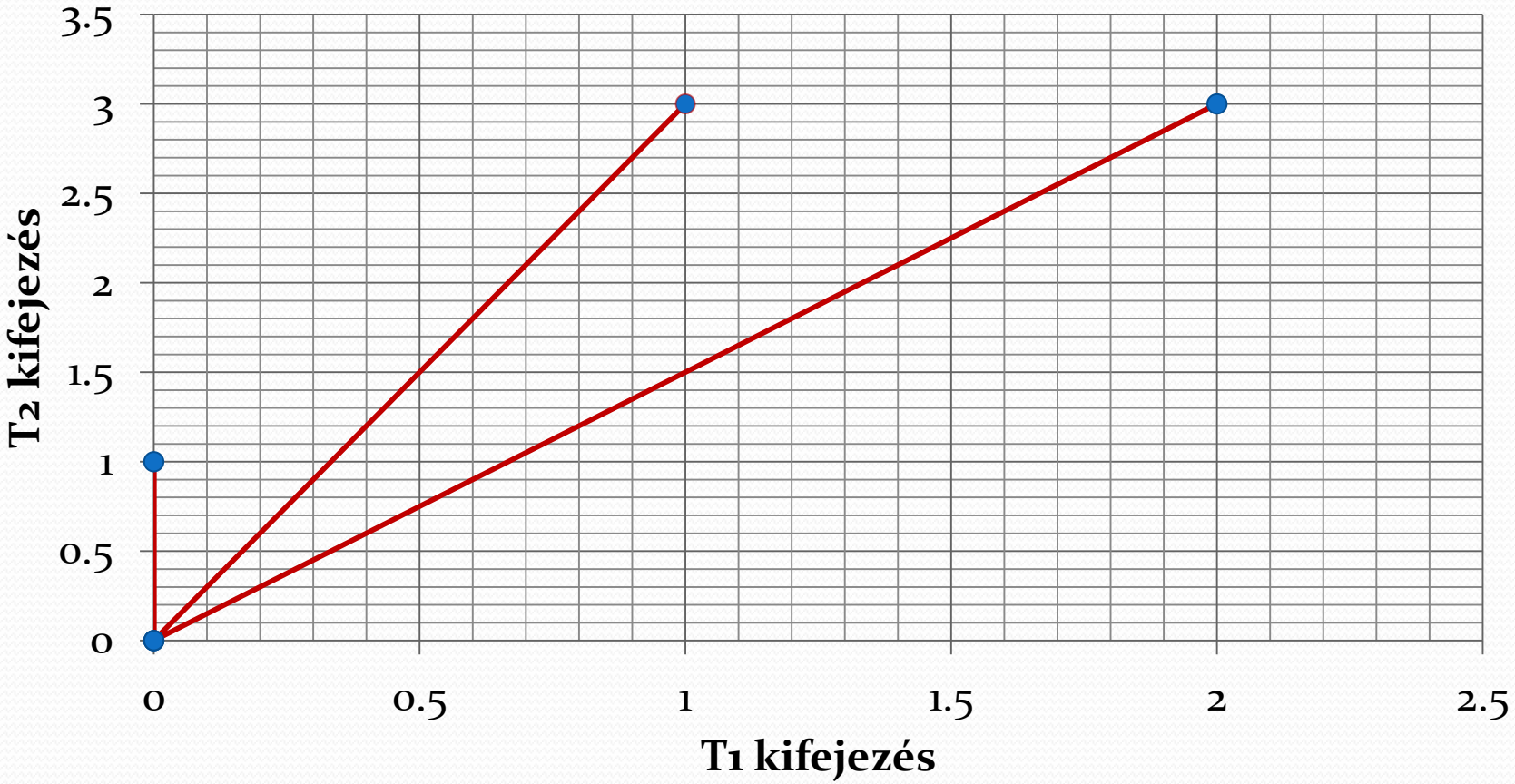
Tehát a visszakeresés lépései a következők:

1. A Q_k keresőkérdés megadása.
2. A t_i indexkifejezések w_{ik} súlyának meghatározása a Q_k kérdésben (hasonlóan, mint a dokumentumokban).
3. Minden D_j dokumentumra a hasonlósági mérték értékeinek kiszámítása.
4. A találati lista (visszakapott dokumentumok) megadása hasonlósági érték szerint csökkenő sorrendben.

q(0,1)

d2 (1,3)

d1(2,3)



2.4. Hasonlósági mértékek -1.

- **Pont, vagy skalár szorzat (Dot product):**

$$s_{jk} = (\mathbf{v}_j, \mathbf{v}_k) = \Sigma_{i}^{n}{}_{=1}\, w_{ij} w_{ik}$$

2.4. Hasonlósági mértékek -2.

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

$$s_{jk} = c_{jk} = (\mathbf{v}_j, \mathbf{v}_k)/(||\mathbf{v}_j|| \cdot ||\mathbf{v}_k||) =$$

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

2.4. Hasonlósági mértékek -3.

- **Dice együttható (Dice's coefficient): d_{jk}**

$$s_{jk} = d_{jk} = 2 \cdot (\mathbf{v}_j, \mathbf{v}_k) \; / \; \Sigma_{i=1}^{n} (w_{ij} + w_{ik})$$

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

2.4. Hasonlósági mértékek -4.

-
- **Jaccard együttható (Jaccard's coefficient): J_{jk}**

$$s_{jk} = J_{jk} = (\mathbf{v}_j, \mathbf{v}_k) \,/\, (\Sigma_i{}^n{}_{=1}\,(w_{ij} + w_{ik})/2^{w_{ij}w_{ik}}) =$$

$$= \frac{\Sigma_i{}^n{}_{=1}\, w_{ij}w_{ik}}{\Sigma_i{}^n{}_{=1}\, \frac{w_{ij} + w_{ik}}{2^{w_{ij}w_{ik}}}}$$

3. Rangsortartás

3.1. A rangsortartás definíciója

3.2. A rangsortartás jelentősége

3.1 A rangsortartás definíciója

Adott:

- két tetszőleges dokumentum (objektum): D_1 and D_2,
- és két hasonlósági mérték: σ_1 and σ_2.

Ha a két dokumentum sorrendje megegyezik mindkét hasonlósági mértékkel számolva bármely tetszőleges Q keresőkérdésre, azaz:

$$\sigma_1(\mathbf{w}_1, \mathbf{q}) \leq \sigma_1(\mathbf{w}_2, \mathbf{q}) \Leftrightarrow \sigma_2(\mathbf{w}_1, \mathbf{q}) \leq \sigma_2(\mathbf{w}_2, \mathbf{q}), \forall D_1, D_2, Q,$$

Akkor azt mondjuk, hogy a σ_1 és σ_2 hasonlósági mérték rangsortartó.

3.1. A rangsortartás jelentősége

Rangsortartás:

- a két hasonlósági mérték egymással ekvivalens, azaz:
- a két hasonlósági mérték egymással helyettesíthető, mivel:

➢ ugyanazokat a találatokat (dokumentumokat) adja vissza,

➢ ugyanabban a sorrendben.

A vektortér modellben általában a hasonlósági mértékek egymással nem helyettesíthetők, tehát nem rangsortartók.

Példa vektortér modellre -1.

- Adott könyvcímek egy kis gyűjteménye (7 dokumentum D_i)

Terms		Documents		Query
T1	Baby	**D1**	Infant and Toddler First Aid	
T2	Child	**D2**	Babies and Children's Room (For your Home)	Child
T3	Guide	**D3**	Child Safety at Home	
T4	Health	**D4**	Your Baby's Health and Safety: From Infant to Toddler	
T5	Home	**D5**	Baby Proofing Basics	Home
T6	Infant	**D6**	Your Guide to Easy Rust Proofing	Infant
T7	Proofing	**D7**	Babies Collectors Guide	Proofing
T8	Safety			Safety
T9	Toddler			

Table 3.1. Collection of book titles with the index terms

Példa vektortér modellre -2.

- m=7 dokumentum (mátrixnak 7 oszlopa van), azaz D=*{D1;D2;D3;D4;D5;D6;D7}*
- n=9 indexkifejezés (a mátrixnak 9 sora van), azaz T =*{t1; t2; t3; t4; t5; t6; t7; t8; t9}*,
- a dokumentumok:
  - *D1 = {t6; t9}*
  - *D2 = {t1; t2; t5}*
  - *D3 = {t2; t5; t8}*
  - *D4 = {t1; t4; t6; t8; t9}*
  - *D5 = {t1; t7}*
  - *D6 = {t3; t7}*
  - *D7 = {t1; t2}*

Példa vektortér modellre -2.

- *tfn* súlyszámítási sémával a TD mátrix:
- $$D := \begin{pmatrix} 0 & 0.5774 & 0 & 0.4472 & 0.7071 & 0 & 0.7071 \\ 0 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7071 & 0.7071 \\ 0 & 0 & 0 & 0.4472 & 0 & 0 & 0 \\ 0 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7071 & 0.7071 & 0 \\ 0 & 0 & 0.5774 & 0.4472 & 0 & 0 & 0 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \end{pmatrix}$$

Példa vektortér modellre -3.

- *tfn* sémával a *term-by-query* :

$$Q := \begin{pmatrix} 0 \\ 0.4472 \\ 0 \\ 0 \\ 0.4472 \\ 0.4472 \\ 0.4472 \\ 0.4472 \\ 0 \end{pmatrix}$$

Példa vektortér modellre 4.

- Cosine, Dice and Jaccard mértékekkel számított értékek:

Document	Similarity measure		
	Cosine	Jaccard	Dice
D1	0.316	0.092	0.174
D2	0.516	0.142	0.264
D3	0.775	0.224	0.39
D4	0.4	0.094	0.178
D5	0.316	0.092	0.174
D6	0.316	0.092	0.174
D7	0	0	0

Péda vektortér modellre 5.

- 1. Adja meg a hasonlósági mértékek által visszaadott találati listákat!
- 2. Vizsgálja meg a hasonlósági mértékek rangsortartását!

MORFÉMÁK - SZUFFIXEK

A **morféma (morpheme)** a nyelv legkisebb olyan egysége, amely önálló jelentést vagy strukturális szerepet hordoz; a szó legkisebb értelmezhető része.
A **toldalék** *(affixum)* jelentésváltoztató, -módosító vagy viszonyjelentést hordozó szórész, morféma. Közvetlen környezete a szótő. A toldalékok részben helyzetük, részben szerepük szerint csoportosíthatók. A szótőhöz viszonyított helyük szerint lehetnek:

- **szuffixumok**, ha a szótő mögé kerülnek (pl. *erdő.**ben***);
- **prefixumok**, ha a szótő elé kerülnek (pl. ***meg**.eszik*);
- **infixumok**, ha beékelődnek a szótőbe (ez a magyartól teljesen idegen);
- **cirkumfixumok**, ha körülveszik a szótövet (pl. ***leg**.jo.**bb***)

Ex.: connect + ion=connection

MORFÉMÁK - SZUFFIXEK

- A magyar nyelv szuffixumai változatos feladatokat láthatnak el, így funkciójuk szerint három alcsoportba sorolhatók:
- a képző megváltoztathatja szótári szó jelentését, rendszerint új szavakat hoz létre, és szófajváltást is eredményezhet: *ég.i,* Egy szótőhöz több is járulhat.
- a jel valamilyen viszonyjelentéssel (többek közt mód, idő, hasonlítás, többség, birtoklás) módosítja a fogalmi jelentést, gyakran további jelek vagy ragok felvételét kívánja: *fiú.é* (birtokjel), *áll.t.unk* (múlt idő jele+ igei személyrag);
- a rag a szavak mondatban betöltött szerepét, a mondat más szavaihoz való viszonyát jelöli. A szavakban csak egyetlen rag található, amely lezárja a szóalakot. Utána már semmilyen más toldalék nem állhat. *kert.ben* (határozórag),
- A szótő és a képző a szótári szavak létrehozásában játszik szerepet(lexikológiai természetű), a jel és a rag ellenben a mondatok felépítésben nélkülözhetetlen (grammatikai szerepű) .

The Porter Stemming Algorithm

- The Porter stemming algorithm (or 'Porter stemmer') is a process for removing the suffixes from words in English.
  - Ex.: connection → connect
- Complex suffixes are removed step by step. Thus:
  - GENERALIZATIONS is stripped to GENERALIZATION (Step 1),
  - then to GENERALIZE (Step 2),
  - then to GENERAL (Step 3), and
  - then to GENER (Step 4).

The *rules* for removing a suffix are given in the form:

(condition) S1 -> S2

This means that if a word ends with the suffix S1, and the stem before S1 satisfies the given condition, S1 is replaced by S2.

Ex #1.: (*S or *T) ION -> .

Here S1 is `ION' and S2 is null. This would map ADOPTION to ADOPT because the word ends with letter S or T.

Rules:

SSES -> SS **caresses -> caress**
IES -> I **ponies -> poni**
ties -> ti
S -> **cats -> cat**

ATIONAL->ATE **relational -> relate**
TIONAL ->TION **conditional -> condition**
IZER ->IZE **digitizer -> digitize**
ATION ->ATE **predication -> predicate**

Stemming Hungarian

Néhány szabály a magyar nyelvű szőtövesítésből:

Rule:	**Example:**
Z -> Z	**ráz -> rázz**
Z -> ZÁL	**ráz -> rázzál**
[ÓUÚ] -> VAL	**manó -> manóval, hamu ->hamuval, bú -> búval**
. -> KÉNT	**kutya -> kutyaként, okos -> okosként**
A -> -A,ÁNAK	**macska -> macskának**

A HelyesLem lemmatizáló, azaz szótővesítő program egy adott nyelv egy tetszőleges toldalékolt szóalakjára a szó tövének (vagy töveinek) szótári alapalakját (vagy alakjait) adja vissza. Használata főként szövegekben történő keresésnél, illetve a keresési index elkészítésénél fontos, mivel a magyar nyelvben a toldalékolás során a szótő gyakran megváltozik, és ezekben az esetekben az egyszerű, betű szerinti keresés nem találja meg az összes keresett alakot.

Magyar hiedelem-szövegek adatbázisa

- Az MHAB (Magyar Hiedelemszövegek Adatbázisa) 2704 magyar hiedelem-szöveget tartalmaz (Darányi, 2001). Vannak köztük rövid szövegek, amelyek tömören egy hiedelmet tartalmaznak, és vannak olyanok is, amelyek az adott hiedelmet kis történeten keresztül mutatják be. Néhány példa a hiedelmekből:
- *- Aki máskor vet, az ne szóljon senkinek.*
- *- Aki aratáskor a tarlót vizzel leönti s a vizes helyre lép, akkor seb lesz a lábán.*
- *- Ha azt akarod, hogy a bajszod hama nyôjön, akkor rëgge amind kinyitod a szëmed, minnyá a két újjadda éjnyalló bekenyéd a bajszod helit.*

Magyar hiedelemszöveg-bázis

- *Alig lëhettem 10 éves de úgy emlikszëm mintha most történt volna. A tehenünk véres tejet adott azt monták, hogy a boszorkányok mëgfejik. Hát jóvan gondotam magamban majd én mëglesem, hogy ki feji mëg a tehenünket. Szótam a szogánok a bátyámnok és igy hárman kifeküttünk az istallóba. Ëgyszër aztán mëgcsapott engém ëgy hideg levegô és abban a pillanatban erôs nyomást érëztem a mellemën és jól hallottam hogy a tehén zúg vagyis a teje. Fölakartam keni de nem tudtam még mëgmozduni së, szóni akartam de azt se tudtam hát vártam. Ëgyszërcsak hallom, hogy fölugrik a bátyám és kapja a vasvillát és belevágja az istálló ajtóba és ekezd kiabányi, hogy megvan a boszorkány. Kigyüttek a szüleim a lármáró kerestük, de sëhol sëm találtunk sënkit az egész istállóban, és a tehén tôgye üres vót és nedves. Másnap aztán együtt hozzánk az öreg Tolam néni ëgy kis sóér më hagymáér, mer a boszorkánynak ha valahol megszúrják el köll mënni oda másnap sóért és foghagymáér mer csak akkor gyógyul më a sebe. De bizony az öreg Tolam néni mëgjártó mert amint e panaszkotta hogy a hatábó esett a vasvilla mingyá tudtuk hogy ô fejte më a tehenünket, és a só më a foghagyma mellé bátyám jó everte, de többet nem is gyütt még felénk se.*

Magyar hiedelem-szövegek adatbázisa

- Számítógépes nyelvtechnológiai szempontból az MHAB sajátosságai közül ki kell emelni a következőket. :
- A szövegek ASCII (szöveg) formátumban szerepelnek, ezért az MHAB viszonylag könnyen adaptálható (lehetővé teszi a kívánt számítógépes adatszerkezetek kialakítását és szükséges algoritmusok alkalmazását):
- Mind mai írásmód, pl.
- *Ha kis gyermeknek komoly baja van, akkor szenes vízzel mossák meg. A meleg vízbe 9 db. szenet tesznek, megkenik a vízzel a gyermek homlokát, és ezt mondják: Ha férfi, kalap alá; ha leány, párta alá; ha asszony, fejkötő alá, az atya, fiú, szentlélek nevében. Ámen.*
- mind régebbi vagy tájszólás jellegű írásmód és szóhasználat, pl.
- *Ha a tehenet merrontya a boszorkány, vësznek egy új fëlliteres cserepbëgrét; abba belëtësznek ecs csomaócskát a tehen gannajjábó. Azután szöget vernek a kény belsejébe s erre felakasztyák a bëgrét. Etteô aszt meggyön a tehen haszna.*"
- jellemző az MHAB állományára.

Magyar hiedelem-szövegek adatbázisa

Viszonylag sok a szóalak (pl. a számítógépes nyelvtechnológiában elterjedten vizsgált angol nyelvvel összehasonlítva):

- *asszony, asszonnak, asszony, zasszony, háziasszony, asszonyról, fehírníp, asszonyhoz, gazdasszony, kisasszony, gazdasszonyok, fehérnép, asszonyt, asszonnyal, asszonyok, fehérnépek, asszon, háziasszonyok, vászoncselédeknek, asszonyokhoz, gazdaasszony, asszonyokról, gazdasszonynak, gazdasszonya, ételvivôasszony, asszonya, asszonynak, asszonyoknak, háziasszonynak, asszonyai, asszonyra*

Magyar hiedelem-szövegek adatbázisa

- Az automatikus szövegfeldolgozás első lépéseként a stop-lista meghatározására került sor.
- 1,551 stop-szó azonosítása történt meg, manuálisan, azaz olyan szóé (névmás, határozó, jelző, ige, múlt idejű alak, ritkán használt szó, ragozott alak), amely nem vagy alig hordoznak jelentést a hiedelemre nézve. A stop-lista néhány részlete a következő:
- *abba, abban, abbó, abból, abbú, abbüó, addig, ahány, ahanyadik, ahányadik, ahányan, ahányat, ahányszor, ahányszori, ahhoz, ahogy, ahol, ahon, ahonneét, ahonnét, ahova, ahová, ahun, ahuon, ajánlatos, ajánlják, akár, akárhogy, akármelyik, akármilyen, akármit, aképen, …, aki, akié, akiébe, akiért, akihez, akijé, akik, akiknek, akin, akinek, akinél, akire, akiről, akit, akitől, akivel, akki, akkinek, akkire , …, zén, zett, zije, zik, zis, zisnagyon, zni, zsémb, ztem, zzen, zönt, örvend, örvendetes, örvendetesebbet, összefügg, összefüggő, övéket*

Magyar hiedelem-szövegek adatbázisa

- Az eltávolítás után 14.421 szóalak maradt az eredetileg szereplő összes 15.972 szóalakból.
- Az automatikus nyelvtechnológia következő lépésekét az azonos jelentéssel felruházható, de különböző alakú szavaknak azonos tőre való redukálása (lemmatizálás, stemming) képezi.
- Jóllehet léteznek a magyar nyelvre kifejlesztett stemmerek (Morphologic, Szószablya), a hiedelemszövegek változatos, különleges (fentebb érzékeltetett) szóhasználata, régi homonimák miatt azokat az MHAB-ra nem vagy csupán igen alacsony hatékonysággal lehet alkalmazni. Ezért a szótőre való visszavezetés manuálisan valósult meg.

Magyar hiedelem-szövegek adatbázisa

- A szótő-lista néhány részlete a következő:
  #agyon
- *#ajak*
- *#ajtó*
- *#ajándék*
- *#akadály*
- *#akar*
- *#alsónemű*
- *#csal*
- *#család*
- *#csütörtök*
- *#úrfelmutatás*
- *#úrnapja*
- *#úrvacsora*
- *#üszô*
- *#üt*

Magyar hiedelem-szövegek adatbázisa

- A stop-listán szereplő szavaknak automatikus, C++ programozási nyelven írt számítógépes programok segítségével való törlése után maradt szavak tövesítése 2.602 szótőt eredményezett.
- Ezek a szótövek képezik az *index-kifejezéseket*, amelyek segítségével minden hiedelemszöveget a benne előforduló index-kifejezések előfordulási számainak számtömbjeként, 'vektoraként' ábrázoltunk.
- A valamennyi hiedelem-vektort oszlopokba és egymás mellé rendezve kapjuk a *kifejezés-dokumentum* (term-by-document) *matrix*ot, TD-t. A TD mátrixnak az *i*-ik sorában és *j*-ik oszlopában szereplő eleme az *i*-ik index-kifejezésnek a *j*-ik hiedelemszövegben való előfordulási száma.

Magyar hiedelem-szövegek adatbázisa

- A TD mátrixot automatikusan, C++ programozási nyelven írt számítógépes program segítségével állítottuk elő, 2.602 sora és 2.704 oszlopa van (Sorszám=index-kifejezés sorszáma, oszlopszám=hiedelemszöveg sorszáma)

TD =

	21	22	23	24	25	26	27	28	29	30
1	0	1	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0

Magyar hiedelem-szövegek adatbázisa

Hiedelemszövegek száma	2.704
Index-kifejezések száma	2.602
Index-kifejezések maximális száma/szöveg	263
Index-kifejezések minimális száma/szöveg	1
Index-kifejezések átlagos száma/szöveg	12
Index-kifejezések átlagos számának szórása	11
Index-kifejezés maximális előfordulási száma/szöveg	16
Szövegek maximális száma ugyanazon index-kifejezéssel	386
Szövegek minimális száma ugyanazon index-kifejezéssel	0