Gradus Vol 3, No 2 (2016) 73-79
ISSN 2064-8014

A VEKTORTÉR MODEL HASZNÁLATA A SZÖVEGBÁNYÁSZATBAN

THE USAGE OF THE VECTOR-SPACE MODEL IN TEXT MINING

Subecz Zoltán[1*]

[1]Informatika Tanszék, Gépipari és Automatizálási Műszaki Főiskolai Kar, Pallasz Athéné Egyetem, Magyarország

Kulcsszavak:
természetes nyelvfeldolgozás
szövegbányászat
adatbányászat
vektortér modell
programozás

Keywords:
natural language processing
text mining
data mining
vector space model
programming

Cikktörténet:
Beérkezett 2016. szeptember 6.
Átdolgozva 2016. november 14.
Elfogadva 2016 november 15.

Összefoglalás

A természetes nyelvi feldolgozás egy új interdiszciplináris terület. Ennek a területnek a célja a számítógép segítségül hívása olyan feladatokban, mint a természetes szövegek feldolgozása, az ember-gép közötti kommunikáció elősegítése és egyéb szöveg feldolgozási feladatok. A nyelv ismerete az, ami megkülönbözteti a nyelvi szövegeket feldolgozó alkalmazásokat más adatfeldolgozó rendszerektől. A természetes nyelvi feldolgozás számos formális modellt és elméletet használ. A modellek a számítástechnika, matematika és a nyelvészet eszközeivel dolgoznak, ezek között található a Vektortér modell is. A vektortér modell a lineáris algebra alapjaira épül, és segítségére van sokfajta információkinyerési módszernek. A cikkben áttekintettem a Vektortér modell elméletét, és feldolgoztam természetes nyelvi szövegeket az internetről a vektortér modell segítségével.

Abstract

The natural language processing is a new interdisciplinary field. The goal of this new field is to get computers to perform useful tasks involving human language, tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech. What distinguishes language processing applications from other data processing systems is their use of knowledge of language. The natural language processing uses some formal models or theories. These models and theories are all drawn from the standard toolkits of computer science, mathematics, and linguistics. Among these models are the vector-space models. Vector-space models, based on linear algebra, underlie information retrieval and many treatments of word meanings. In this article I reviewed Vector-space models theory and processed 100 natural language texts from Internet with Vector-space models.

* Kapcsolattartószerző. Tel.: +36 56/511-750
E-mail cím: subecz@szolf.hu

1. Bevezetés

A kutatási témám keretében az interneten megtalálható ingatlanközvetítői hirdetések szövegét dolgoztam fel. Ennek keretében a kiválasztott szövegeket több részlépésben kellett átalakítani és a szükséges információt kigyűjteni belőlük. Ezek alapján készítettem el a Vektortér modellt, amit szöveges tartalmak hatékony reprezentációjához dolgoztak ki. Ezekhez a feladatokhoz Java programozási nyelv segítségével írtam programokat. Az elkészített *Vektortér modell* segítségével részletesen elemeztem a kiválasztott szövegeket és az ezeket alkotó szavakat. A feladatot 100 internetes cikken végeztem el, de mivel a módszer automatizált, ezért több ezer dokumentumra is ugyanúgy alkalmazható.

1.1. Az információkinyerés és szövegbányászat

Az *információkinyerés* (IE, Information Extraction) technológiájának kutatása dinamikusan fejlődő terület a természetesnyelv-feldolgozásban. Az interneten megjelenő hatalmas információtömeg gépi feldolgozása és a kívánt információ tömör formában történő összegyűjtése napi szükséglet, amelyre a gazdaság, a tudomány, a politika területén is van igény. Míg az *információ visszakeresés* (IR, Information Retrieval), amely a webes kereső programok jellemző tevékenysége, arra irányul, hogy a felhasználó igényeinek megfelelő dokumentumokat változatlan formában bocsássa rendelkezésre, addig az információkinyerés célja a megtalált dokumentumokban a lényeges információ megjelölése, majd összegyűjtése. [1]

Az utóbbi évtizedekben az adatok tárolása egyre olcsóbbá vált (a tárolókapacitások rohamosan fejlődtek, míg az árak csökkentek), ezáltal az elektronikus eszközök és adatbázisok elérhetővé váltak mindennapi életünkben. Az egyre olcsóbb adattárolási lehetőségek az adatok tömeges felhalmozását eredményezték, ám ezeket az adatokat nem dolgozzák fel, ezáltal a döntéshozók intézkedéseiket nem hozhatták meg információ-gazdag adatok alapján.

Az utóbbi években az informatika egyik leggyorsabban fejlődő részterülete az *adatbányászat* lett. Ez az új tudományág szolgál a nagy mennyiségű adatokban rejlő információk automatikus feltárására *mesterséges intelligencia algoritmusok* alkalmazásával (például neurális hálók, szabálygenerálók, asszociációs modellek). A fejlődés egyik motorja a pénzügyi haszon, hiszen a kibányászhatatlannak vélt, vagy csak nagyon erőforrás igényesen elérhető információk, összefüggések nagyon sokat érhetnek. [2]

A köznyelv és a különböző informatikai cégek sok mindent neveznek adatbányászásnak, de a szigorúbb szakmai terminológia szerint nem tekinthető adatbányászatnak az adatokból lekérdezésekkel, aggregálásokkal, illetve alapstatisztikai vizsgálatokkal történő információ kinyerése. Az *adatbányászat* egy már meglévő, valamilyen egyéb célból összegyűlt adathalmazban keres megbúvó, rejtett és számunkra hasznos, releváns összefüggéseket, ismereteket, információkat. Az adatbányászat egyik igen fontos részterülete a *szövegbányászat*, amely a strukturálatlan (vagy részben strukturált), elektronikus szöveges állományokban megbúvó, nem triviális információk kinyerését jelenti. [6]

Jól mutatja a probléma jelentőségét, hogy például az üzleti információk 85%-a strukturálatlan, illetve részben strukturált adat formájában áll rendelkezésre (e-mailek, emlékeztetők, üzleti és kutatási beszámolók, prezentációk, hírek, reklámanyagok, weboldalak, ügyfélszolgálati tevékenység jegyzetei, stb.).

Az adatbányászathoz hasonlóan a *szövegbányászat* is a látens összefüggések és rejtett ismeretanyagok automatikus kinyerésére hivatott, de inputja folyó szövegek, azaz strukturálatlan, vagy részben strukturált dokumentumokból áll. A szövegbányászati problémák megoldásai eltérő eszközöket igényelnek, itt modellezni kell az emberek által írt szövegek szintaktikai, szemantikai szerkezetét, sőt a pontos megértéshez azok stílusát is. [3]

Napjainkra a tudásmenedzsment egyik legfontosabb információtechnológiai eszközévé vált a szövegbányászat, melynek segítségével üzleti versenyelőny szerezhető. Az új alkalmazási lehetőségek közül a web-bányászat az egyik legígéretesebb, mivel a világ legnagyobb és leggyorsabban bővülő adattárát, az internetet használja. A *web-bányászat* célja, hogy az internethez kapcsolható dokumentumokból (honlapok, e-mailek, blogok, fórumok stb.) hasznos információkat automatikusan összegyűjtsön. Ilyen feladat lehet például állásajánlatok automatikus összegyűjtése vállalati honlapokról, vagy újsághírekben egy vállalkozásról fellelhető információk kinyerése. Az így

nyert információ strukturált szerkezetű (mezőértékekből áll), azaz például betölthető egy adatbázisba.

2. A Vektortér modell

A *szövegbányászatban* a tartalmak tömör reprezentációjára a *vektortér modell* (VTM) nyújtja a legszélesebb körben használt megoldást. A modell minden egyes dokumentumot egy vektorral ír le, amelyben minden elem az egyes *termek* (általában szavak) előfordulását jelenti. Termek alatt a reprezentáció egységeit, alapesetben az írásjelek által határolt szavakat (unigram) értjük. [7]

Adva van egy *dokumentumgyűjtemény*, amelynek elemein valamilyen rendszerezési műveletet kívánunk végrehajtani. Ehhez olyan modellt kell felépítenünk, amiben a dokumentumok távolságát, vagy hasonlóságát egyszerűen meg tudjuk határozni. Intuitív módon nyilván azok a dokumentumok hasonlítanak egymásra, amelyeknek a szókészlete átfedi egymást, és a hasonlóság mértéke az átfedéssel arányos. Ezt a megfigyelést használja fel az információ-visszakeresésben széles körben használt vektortér modell. [4]

A *vektortér modellben* a $D=\{d_1,\ldots,d_N\}$ dokumentumgyűjteményt a *szó-dokumentum mátrixszal* (term-document mátrix) reprezentáljuk ($D \in R^{MxN}$), ahol a mátrix d_{ki} eleme a k-adik szó (t_k) relevanciáját reprezentálja az i-edik dokumentumban, d_i-ben. A d_i dokumentumot reprezentáló dokumentumvektort $d_i=<d_{1i},\ldots,d_{Mi}>$- vel jelöljük. A D mátrixban a sorok száma, M megegyezik az egyedi szavak számával. N pedig a dokumentumok száma. Ezt láthatjuk az 1. ábrán.

$$D = \begin{pmatrix} d_{11} & d_{12} & \ldots & d_{1N} \\ d_{21} & d_{22} & \ldots & d_{2N} \\ \ldots & \ldots & \ldots & \ldots \\ d_{M1} & d_{M2} & \ldots & d_{MN} \end{pmatrix}$$

1. ábra. Vektortér modell

A dokumentumvektorok tehát a mátrix oszlopai lesznek. A mátrix egy sora azon pozíciókban tartalmaz nullától különböző értéket, amelyekhez tartozó dokumentumokban a szó nem nulla relevanciájú. Ez általában ekvivalens azzal, hogy a szó nem szerepel a dokumentumban, de ettől eltérő esetekben is lehet a relevancia nulla.

Az egyedi szavak összességét *szótárnak, vagy lexikonnak* nevezzük, jelölése: T. A szótár mérete tehát |T|=M. Mivel általában egy dokumentumban a szótár szavainak csak egy kis töredéke fordul elő, ezért a D mátrix ritka. Ugyanakkor az egyedi szavak száma rendkívül nagy lehet, akár a milliós nagyságrendet is elérheti. [5]

Ahhoz, hogy a Vektortér modell mátrixába be tudjuk tölteni az értékeket, a szövegeket előre fel kellett dolgozni. Így először a következő lépéseket készítettem el programmal:

- A kiválasztott hirdetések letöltése az Internetről
- HTML oldalakról a feldolgozandó szövegek kigyűjtése szövegfájlokba
- A bemeneti szövegek szavakra bontása, kisbetűsre alakítása
- A dokumentumhalmaz egyedi szavainak kigyűjtése egy fájlba (3317 db egyedi szó)

3. A Vektortér modell elkészítése és vizsgálata

A Vektortér modellnek több fajtája is van, attól függően, hogy a mátrix celláiba milyen értékeket teszünk. Ezek közül a következő hármat készítettem el:

- bináris mátrix
- szógyakorisági TD mátrix
- súlyozott TD mátrix

3.1. A bináris mátrix

A lehetséges mátrixok közül ez a legegyszerűbb. Ebben az esetben azt tároljuk a mátrixban, hogy az adott dokumentumban előfordult-e az adott szó vagy nem. A bináris reprezentációnál a szó-dokumentum mátrixban a d_{ki} értékeket a következő képen határozzuk meg:

$$d_{ki} = \begin{cases} 1, ha\ n_{ki} > 0 \\ 1, ha\ n_{ki} = 0 \end{cases} \quad (1)$$

ahol n_{ki} a t_k szó előfordulásainak száma (másképpen támogatottsága) a d_i dokumentumban. Vagyis, ha egy adott szó előfordul legalább egyszer a dokumentumban, akkor ott a mátrixban az érték 1, különben 0. Ez a mátrix azon az igényen alapul, hogy egy szónak annál nagyobb a jelentősége az adott dokumentumhalmazon, minél több dokumentumban szerepel. A bináris mátrixban csak egyesek és nullák szerepelnek.

A Vektortér modellt felhasználhatjuk például *osztályozásra* is. Például, ha a dokumentumhalmaz elemei két csoportba sorolhatóak (A és B csoportok), akkor a mátrix vizsgálatával kiválaszthatóak olyan szavak, amelyek inkább az A csoport dokumentumaira jellemzőek és olyanok, amelyek a B csoport dokumentumaira. Ha új, ismeretlen dokumentumot akarunk besorolni az A, vagy a B csoportba, akkor meg kell vizsgálni, hogy az egyes csoportokat jellemző szavak közül melyikből található meg több a vizsgált új dokumentumban. Vannak olyan szavak, amelyek minden dokumentumban gyakran szerepelnek (például a névelők, és, vagy stb.), így ezeknek az osztályozás szempontjából nincs jelentősége. Ezeket nevezik *stop szavaknak*.

A mátrix jellemzése:

- Sorok száma: 3317
- Oszlopok száma: 100
- Vektortér celláinak száma: 331700
- Nem 0 értékű cellák: 9349
- A mátrix kitöltöttsége: 2,82%

Vagyis a mátrix celláinak csak 2,82%-ában van nullától különböző érték. A cellák 97,18%-ában nulla érték áll. Ez mutatja, hogy ez egy ritka kitöltöttségű mátrix.

A bináris mátrixnak a gyakorlati jelentősége kisebb, mint a következő szógyakorisági TD mátrixnak, mert abból több információ kiolvasható.

3.2. A Szógyakorisági TD mátrix

Ebben az esetben $d_{ki} = n_{ki}$. Vagyis a d_{ki} itt azt adja meg, hogy az adott szó hányszor fordul elő az i. dokumentumban. Egy szónak annál nagyobb a súlya egy dokumentumban, minél többször fordul elő. Ennek a mátrixnak is 3317 sora és 100 oszlopa van, így a celláinak száma: 331.700.

Statisztika a dokumentumhalmaz szavairól a szógyakorisági mátrix alapján.

Kigyűjtöttem, hogy melyik szó hány dokumentumban és összesen hányszor szerepel. Ez alapján a leggyakrabban előforduló 10 szó látható az 1. táblázatban.

1. Táblázat. A leggyakrabban előforduló 10 szó

Szavak	*Összesen*	*Hány fájlban?*
a	969	96
és	329	85
az	244	84
egy	163	76
is	149	73
lakás	112	55
van	99	49
található	93	49
ház	75	45
helyezkedik	69	42

Mint ahogy az várható volt, a *stop szavak* (névelők és egyéb rövid kötőszavak: a, és, az, egy, is) sokszor szerepelnek a dokumentumokban. Az viszont érdekesség, hogy nincs olyan szó, ami minden dokumentumban szerepel. A leggyakoribb az ˝a˝ névelő is csak 96 dokumentumban szerepel. A gyakorisági sorrend elején találhatóak az ingatlanhirdetésekben gyakran előforduló szavak (pl. lakás, van, található, ház, jó, eladó, nm, ingatlan…).

Ha azt vizsgáljuk, hogy melyik szó összesen hányszor szerepel a dokumentumhalmazon, akkor látszik, hogy az első helyezett, az ˝a˝ névelő jelentősen megelőzi a másodikat. Majdnem háromszor annyiszor szerepel, mint az ˝és˝ kötőszó. Ha viszont azt vizsgáljuk, hogy melyik szó hány dokumentumban szerepel, akkor látható, hogy itt már nincsen akkora különbség az első és a második helyezett között: az ˝a˝ névelő 96 dokumentumban, az ˝és˝ kötőszó 85 dokumentumban szerepel.

A mátrix egyéb jellemzése:

- Sorok száma: 3317
- Vektortér celláinak száma: 331700
- Nem 0 értékű cellák: 9349
  így a mátrix kitöltöttsége: 2,82%
- Legtöbb szót tartalmazó fájl: 438 db szó
- Legkevesebb szót tartalmazó fájl: 16 db szó
- Azon szavak száma, amelyek csak egy dokumentumban fordulnak elő: 2010

Így a 3317 szó hatvan százaléka csak egyszer fordul elő.

Ennek a mátrixnak is 2,82%-os a kitöltöttsége

3.3. A súlyozott TD mátrix elkészítése a szógyakorisági TD mátrixból

Gyakorlati szempontból ennek a mátrixnak van a *legnagyobb jelentősége* az előző kettővel összehasonlítva. Egy-egy cellájában arra ad értéket, hogy egy adott szónak mekkora a jelentősége egy adott dokumentumban. A szógyakorisági TD mátrix egy cellája azt tartalmazta, hogy egy adott szó hányszor szerepel egy adott dokumentumban. A súlyozott TD mátrixban ezt az értéket két lényeges szempont szerint módosítjuk.

Az eddig ismertetett súlyozási sémák figyelmen kívül hagyták a dokumentumok hosszát, noha egy 100 szavas dokumentumban egy szó tízszeri előfordulása nyilván sokkal jelentősebb, mint egy 10 000 szavasban. Ezt a szempontot a dokumentumok hossz szerinti normálásával vehetjük számításba. A súlyozott TD mátrixban tehát nem azt vizsgáljuk, hogy egy szó hányszor szerepelt egy dokumentumban, hanem ezt a számot viszonyítjuk az adott dokumentum szavainak a számához.

A dokumentum szavainak számát jelöljük:

$$|d_i| = \sum_{k=1}^{M} n_{ki} \tag{2}$$

akkor a gyakoriság alapú súlyozás: $f_{ki} = n_{ki} / |d_i|$

Az így definiált f_{ki} a t_k szó d_i dokumentumbeli gyakorisága, vagy frekvenciája, amit az angol elnevezés rövidítése alapján *TF-súlyozásnak (term frequency)* is hívnak. A súlyozott TD mátrixnál e mellett még egy szempontot figyelembe veszünk. Mégpedig azt, hogy az adott szó hány dokumentumban szerepel. Minél több dokumentumban szerepel egy szó, annál kisebb a jelentősége (stopszavak pl. névelők). Mindeddig a dokumentumokban (szótárban) előforduló összes szót egyenrangúnak tekintettük, holott a dokumentumok tartalmi jellemzését illetően a szavak jelentősége eltérő. Például egy dokumentumon belül nem sok tartalmi jelentősége van a névelőknek, mert azok sok dokumentumban előfordulnak. Ha van két szó, amelyik a 100 dokumentumban ugyanannyiszor fordul elő, akkor a két szó közül az a fontosabb, amelyik koncentráltan, kevés dokumentumban, de azokon belül nagy gyakorisággal fordul elő, semmint az, amelyik sok dokumentumban alacsony gyakorisággal.

Jelöljük n_k-val azon dokumentumok számát, amelyben a t_k szó előfordul. Ekkor az n_k/N hányados, amit *dokumentum gyakoriságnak (document frequency, df)* neveznek, jól jellemzi a szó ritkaságát a korpuszban. Ez az érték megadja, hogy mekkora megkülönböztető ereje van, avagy mennyire tekinthető indikátornak a szó jelenléte (és előfordulásainak a száma) a dokumentum tartalmára vonatkozóan.

A súlyozási sémákban inkább a dokumentumgyakoriság inverzével számolnak (idf ,inverse document frequency): $idf(t_k) = \log(N/n_k)$

Így kapjuk a leggyakrabban használt *td-idf súlyozást. (term frequency & inverse document frequency)*: $d_{ki} = f_{ki} * idf(t_k)$

A *tf -idf súlyozás* értéke tehát:

1. Magas lesz azon szavak esetében, amelyek az adott d_i dokumentumban gyakran fordulnak elő, míg a teljes korpuszban ritkán (nagy a megkülönböztető képességük).
2. Alacsonyabb lesz azon szavak esetén, amelyek a d_i dokumentumban ritkábban, vagy a korpuszban gyakrabban fordulnak elő.
3. és kicsi lesz azon szavakra, amelyek szinte a korpusz összes dokumentumában előfordulnak.

A súlyozott TD mátrix egy-egy cellájában arra ad értéket, hogy egy adott szónak mekkora a jelentősége egy adott dokumentumban.

Elemzésül kiválasztottam 3 szót (lakás, séta, otthon), és több szempont szerint összehasonlítottam ezeket. A három szót elsősorban az első dokumentum alapján vizsgáltam. A következő eredményeket kaptam:

- A lakás szó:
  - 55 dokumentumban 112-szer fordul elő.
  - kétszer fordul elő az első dokumentumban.
  - td-idf súlya az első dokumentumban: 0.00358
- A séta szó:
  - 4 dokumentumban 5-ször fordul elő.
  - egyszer fordul elő az első dokumentumban.
  - td-idf súlya az első dokumentumban: 0,00964
- Az otthon szó:
  - 14 dokumentumban 18-szor fordul elő.
  - kétszer fordul elő az első dokumentumban.
  - td-idf súlya az első dokumentumban: 0,01177

Látjuk, hogy jóval nagyobb a jelentősége a séta szónak az első dokumentumban, mint a lakás szónak. Mert séta bár csak egyszer fordul elő a dokumentumban (a lakás kétszer), de jóval kevesebb dokumentumban szerepel, mint a lakás szó.

Látjuk, hogy jóval nagyobb a jelentősége az otthon szónak az első dokumentumban, mint a lakás szónak. Mert, bár mindkét szó kétszer fordul elő a dokumentumban, de az ott hon szó jóval kevesebb dokumentumban fordul elő.

A séta szó kevesebbszer szerepel az első dokumentumban, mint az otthon szó, ezért súlya kisebb lesz ott, annak ellenére, hogy a séta szó jóval kevesebbszer fordul elő a dokumentumokban.

A mátrixot tovább vizsgálva kikerestem a benne szereplő legnagyobb és legkisebb td-idf értékeket. Legnagyobb súlya az adott dokumentumban azoknak a szavaknak van, amelyek csak egyszer szerepeltek összesen és olyan fájlban, amelyek a legkevesebb szót tartalmazzák. Ilyen szavak voltak például: beállt, debrecen, fodrászüzlet, hajdúszoboszlón, János, mini. Mindegyik szónak 0,125-ös súlya volt az adott dokumentumban.

A legkisebb súlyú szavakat a 2. táblázatban láthatjuk.

2. Táblázat. A leggyakrabban előforduló 8 szó

Szavak	*Súly*
a	0,0002728
egy	0,0004082
is	0,0004104
és	0,0004496
az	0,0004733
jó	0,000814
szobás	0,0008368
szoba	0,0010409

Ezeknek a szavaknak kicsi a jelentősége az adott dokumentumban (például: névelő). Ezek azok a szavak, amelyek valamelyik dokumentumban nagyon kis súllyal szerepelnek, mert sok dokumentumban szerepelnek.

4. Összefoglalás

Ebben a cikkben áttekintettem és példákon keresztül bemutattam a Vektortér modell használatát a szövegbányászatban. Jellemeztem azokat a lépéseket, amiket egy nagyobb dokumentumhalmaz esetén végre kell hajtani, hogy megkapjuk eredményként a vektortér mátrixot. Láttuk, hogy a Vektortér modell segítségével, számítógépes módszerekkel meg lehet állapítani szövegekben a szavainak a fontosságát. A későbbiekben a Vektortér modellt felhasználom majd dokumentum halmazokon végzett osztályozási feladatokra.

Irodalomjegyzék

[1] Martin Daniel Jurafsky and James H. Martin: Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition. Stanford, CA: Pearson Prentice Hall, 2009.
[2] Tikk Domonkos: Szövegbányászat. Budapest, Typotex kiadó, 2007
[3] Tóth Ágoston: Vektortér alapú szemantikai szóhasonlósági vizsgálatok. Magyar Számítógépes nyelvészeti Konferencia, 2013.
[4] Peter D. Turney, Patrick Pantel: From Frequency to Meaning: Vector Space Models of Semantics. Journal of Artificial Intelligence Research 37 (2010) 141-188
[5] D.L. Lee: Document ranking and the vector-space model. IEEE Software > Volume: 14 Issue: 2, 1997
[6] Pablo Castells: An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval, IEEE Transactions on Knowledge and Data Engineering, Volume: 19 Issue: 2, 2007
[7] Haizhou Li: A Vector Space Modeling Approach to Spoken Language Identification, IEEE Transactions on Audio, Speech, and Language Processing, 2007