

Iskanje po zbirski dokumentov - Latentno  
semantično indeksiranje  
Matematično modeliranje, projektna naloga

Nedžad Beus, Gašper Spagnolo, Tilen Ožbot

Junij 2022

# 1 Opis naloge

Naloga je izdelati program, ki bo v zbirki dokumentov za dane ključne besede poiskal najbolj relevantne dokumente, s pomočjo metode *latentnega semantičnega indeksiranja* (LSI).

LSI je metoda za indeksiranje in iskanje, ki uporablja dekompozicijo singularnih vrednosti (SVD) za prepoznavanje vzorcev v odnosih med izrazi in pojmi v nestrukturirani zbirki besedil. Metoda temelji na načelu, da imajo besede, ki se uporabljajo v istem kontekstu, podoben pomen. Ključna značilnost LSI je, da lahko izlušči konceptualno vsebino besedila z vzpostavljanjem povezav med izrazi, ki se pojavljajo v podobnih kontekstih.

## 2 Resitev

Nalogo razdelimo na več korakov:

### 2.1 Izdelava začetne matrike

Iz zbirke dokumentov zgradimo matriko  $A$  povezav med besedami in dokumenti. Vsakemu dokumentu v zbirki ustreza stolpec v matriki, vsaki besedi v zbirki pa vrstica. Element  $a_{ij}$  naj v začetku predstavlja frekvenco  $i$ -te besede v  $j$ -tem dokumentu.

**Enostaven primer:**

Imejmo tri dokumente:

- $d1$  : Jogurt je v vrecki.
- $d2$  : V vrecki imam jogurt.
- $d3$  : Zunaj piha veter.

Najprej prestejemo pojavitve besed v vseh dokumentih.

beseda	d1	d2	d3
"jogurt"	1	1	0
"je"	1	0	0
"v"	1	1	0
"vrecki"	1	1	0
"imam"	1	0	0
"zunaj"	0	0	1
"piha"	0	0	1
"veter"	0	0	1

Nato lahko zgradimo matriko  $A$ :

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Metodo in s tem rezultate lahko izboljšamo, če elemente matrike  $a_{ij}$  izračunamo z bolj kompleksnimi merami kot so npr. entropija. Element matrike  $a_{ij}$  lahko zapisemo kot

$$a_{ij} = l_{ij} \cdot g_i,$$

kjer je  $L_{ij}$  lokalna mera za pomembnost besede v dokumentu,  $G_i$  pa globalna mera pomembnosti besede.

$$L_{ij} = \log(f_{ij} + 1)$$

$$G_i = 1 - \sum_j \frac{p_{ij} \log(p_{ij})}{\log n}$$

$$p_{ij} = \frac{f_{ij}}{gf_i},$$

kjer je  $f_{ij}$  frekvenca  $i$ -te besede v  $j$ -tem dokumentu,  $gf_i$  pa globalna frekvenca  $i$ -te besede v bazi dokumentov.

**Primer matrike  $A$  z uporabo entropije:**

$$A = \begin{bmatrix} 1.6309 & 1.6309 & 0 \\ 1.0000 & 0 & 0 \\ 1.6309 & 1.6309 & 0 \\ 1.6309 & 1.6309 & 0 \\ 0 & 1.0000 & 0 \\ 0 & 0 & 1.0000 \\ 0 & 0 & 1.0000 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

## 2.2 Razcep matrike

Matriko  $A$  razcepimo z odrezanim SVD razcepom  $A = U_k S_k V_k^T$ , ki obdrži le  $k$  največjih singularnih vrednosti. Stolpci matrike  $U_k$  nam predstavljajo t.i. vektorje izrazov, stolpci matrike  $V_k$  pa t.i. vektorje dokumentov. S odstranitvijo najmanjših singularnih vrednosti skušamo aproksimirati originalno matriko  $A$  in s tem zmanjšati šum. Število  $k$  je odvisno od velikosti baze podatkov (npr. pri 1000+ dokumentih je  $k=100$  dobra aproksimacija).

**Primer razcepa matrike  $A$  z uporabo entropije:**

$$A = \begin{bmatrix} 0.5601 & 0 & -0.0000 \\ 0.1717 & 0.0000 & 0.7071 \\ 0.5601 & 0 & -0.0000 \\ 0.5601 & 0 & -0.0000 \\ 0.1717 & -0.0000 & -0.7071 \\ -0.0000 & 0.5774 & 0.0000 \\ -0.0000 & 0.5774 & 0.0000 \\ -0.0000 & 0.5774 & 0.0000 \end{bmatrix} \begin{bmatrix} 4.1182 & 0 & 0 \\ 0 & 1.7321 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix} \begin{bmatrix} 7.0711e^{-01} & 7.8505e^{-17} & 7.0711e^{01} \\ 7.0711e^{-01} & 1.9626e^{-16} & -7.0711e^{01} \\ -2.3551e^{-16} & 1.0000 & -7.8505e^{-17} \end{bmatrix}$$

### 2.3 Vektor poizvedbe

Besede po katerih želimo iskati oz. iskalni niz lahko zapišemo z vektorjem  $q$ , ki je enake dolžine kot število vrstic v matriki  $A$ . Iskalni niz tretiramo kot dokument. Iz iskalnega niza generiramo vektor v prostoru dokumentov po naslednji formuli:

$$\hat{q} = q^T U_k S_k^{-1},$$

ki jo izpeljemo iz naslednjih formul:

$$\begin{aligned} A &= U S V^T, \\ A^T &= (U S V^T)^T = V S U^T \\ A^T U S^{-1} &= V S U^T U S^{-1}, \\ V &= A^T U S^{-1}, \\ \hat{q} &= q^T U S^{-1}. \end{aligned}$$

Dobljeni vektor  $\hat{q}$  je v enakem prostoru kot stolpci v matriki  $V_k$ . Dokumenti oz. stolpci matrike  $V_k$ , ki najbolj ustrezajo poizvedbi so tisti, ki so dovolj blizu vektorju  $\hat{q}$ . Za razdaljo uporabimo kosinus kota med vektorjem.

$$\cos(\alpha) = \frac{\hat{q} \cdot \vec{v}_i}{\|\hat{q}\| \cdot \|\vec{v}_i\|}$$

## 3 Rezultati

Za bazo podatkov smo se odločili za recepte jedi.

## 4 Dodajanje dokumentov in besed

Predpostavimo, da je začetna matrika že zgrajena in SVD te matrike izračunan. Če želimo dodati nove dokumente oz. nove besede, obstajata dve možnosti za dodajanje:

1. ponovno generiranje začetne matrike in izračunavanje SVD,
2. metoda *folding-in* ali zgibanje, ki je hitrejša.

## 4.1 Ponovno izračunavanje SVD

Za to možnost se odločimo, ko dodamo veliko število dokumentov oz. besed. Ponovno zgradimo začetno matriko in SVD razcep. Ponovni izračun SVD-ja omogoča, da nove besede in dokumenti neposredno vplivajo na latentno semantično strukturo, saj se ustvari nova začetna matrika in s tem drugačen SVD razcep. Slabost je, da ponovno izračunavanje SVD velike matrike zahteva veliko časa in pomnilnika.

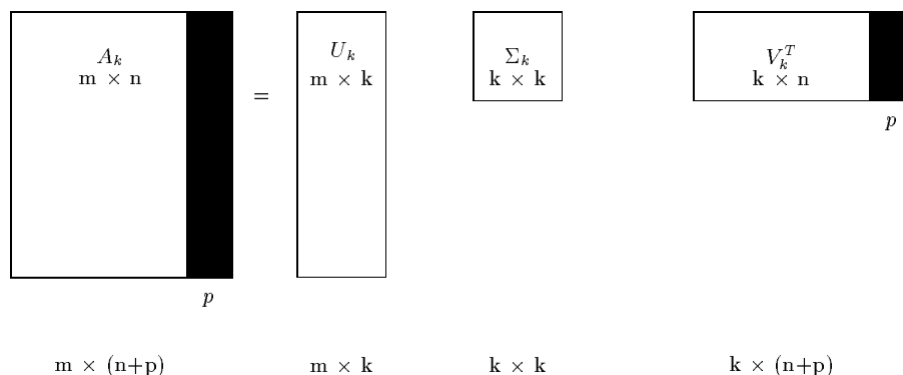
## 4.2 Metoda folding-in ali zgibanje

Metoda zgibanja temelji na obstoječi latentni semantični strukturi, trenutni matriki, zato nove besede in dokumenti ne vplivajo na predstavitev že obstoječih izrazov in dokumentov. Ta metoda zahteva manj časa in pomnilnika, vendar lahko poslabša predstavitev novih izrazov in dokumentov.

Postopek je enak postopku za vektor poizvedbe. Vsak nov dokument je predstavljen kot utežena vsota njegovih sestavnih vektorjev izrazov.

$$\hat{d} = d^T U_k S_k^{-1}$$

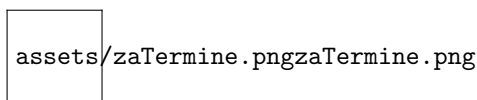
Izračunan vektor je dodan naboru obstoječih vektorjev dokumentov ali stolpcev v matriki  $V_k$ .



Podobno lahko nove besede izrazimo kot uteženo vsoto vektorjev dokumentov, v katerih se pojavljajo.

$$\hat{t} = t V_k S_k^{-1}$$

Izračunani vektor je dodan naboru obstoječih vektorjev besed ali stolpcev v matriki  $U_k$ .



## 5 Viri in literatura

1. M. W. Berry, S.T. Dumais, G.W. O'Brien, Michael W. Berry, Susan T. Dumais, and Gavin. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573–595, 1995.
2. Susan T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments, & Computers*, 23(2):229– 236, 1991.