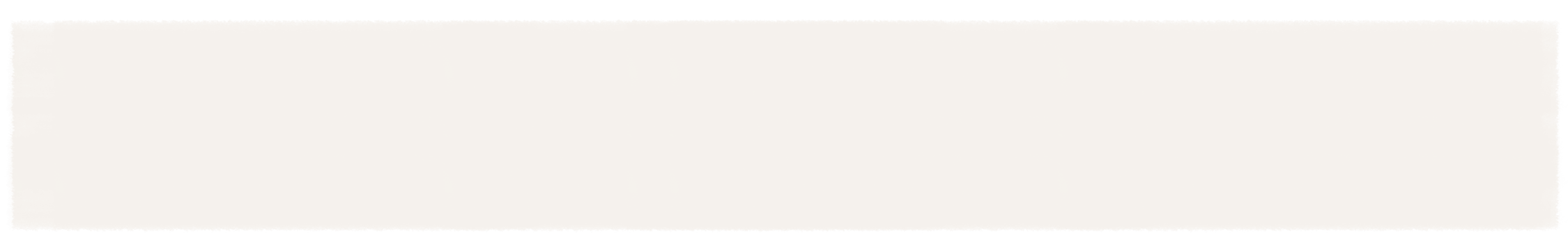
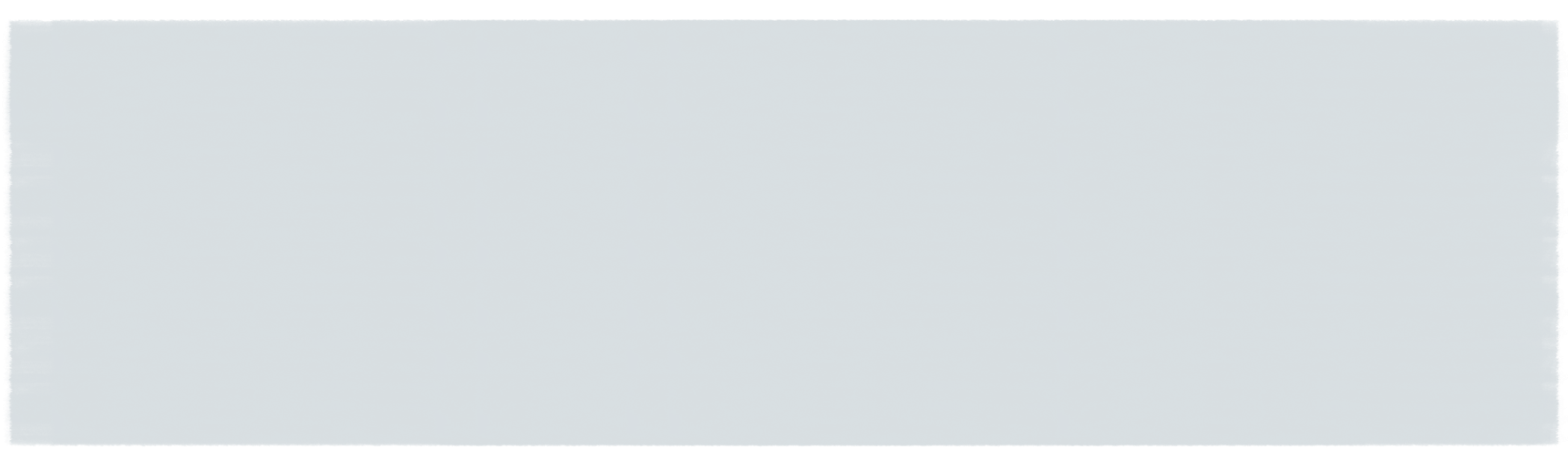
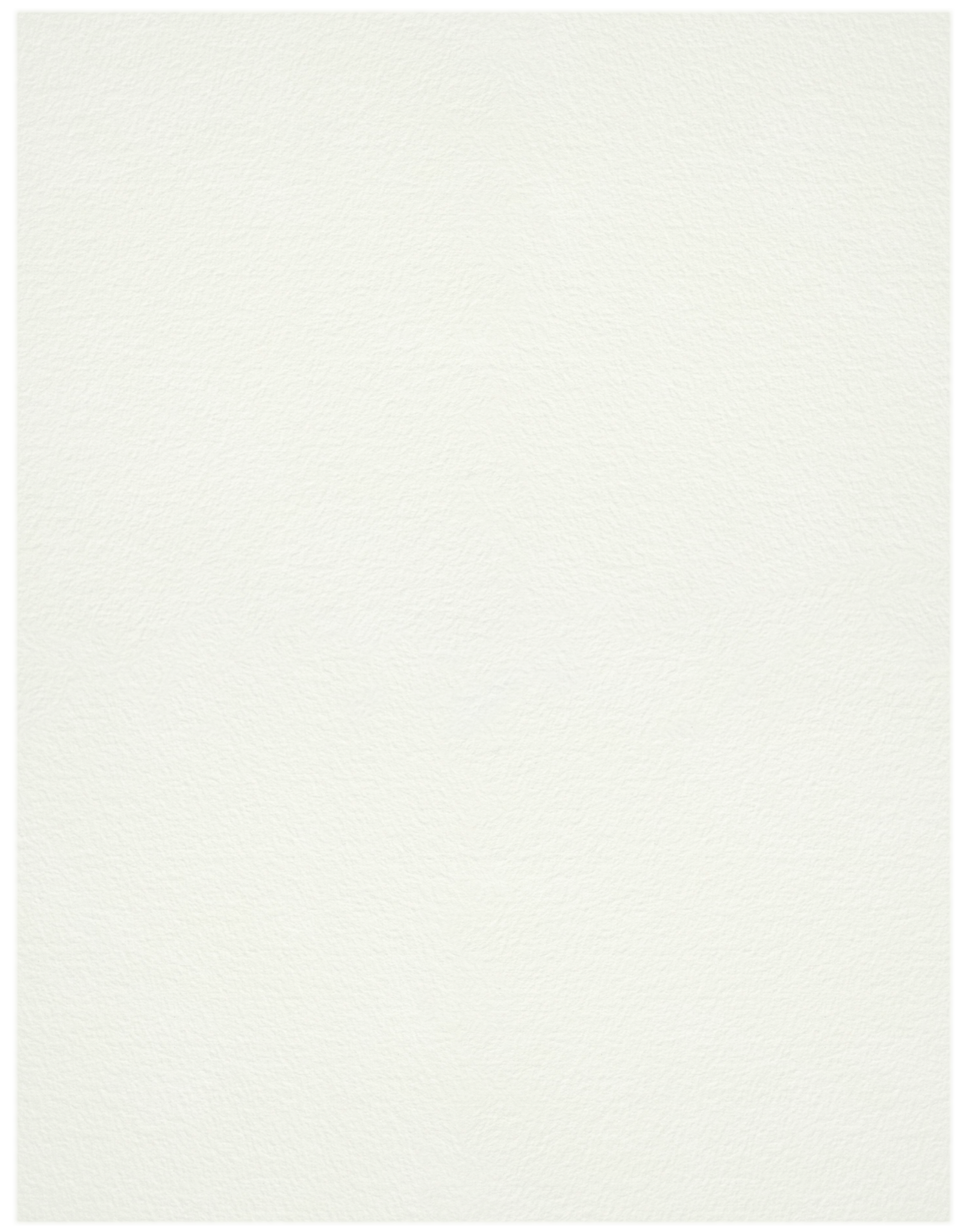
**Poročilo**



**Semantični spletni portal**

**Skupina "FERImdb"**

Vodja: Anja Hauptman

Člani:

Dominik Šbüll

Simona Siljanovska

Urška Nemet

Kazalo

UVOD 4

TF-IDF 5

# UVOD

Za delo pri projektu Semantični spletni portal smo si izbrali tematiko priporočanja filmov uporabnikom. V okviru predmetov Prevajanje programskih jezikov, Sistemska administracija, Spletno programiranje in Umetna inteligenca smo ustvarili spletni portal, na katerem ima uporabnik možnost, da vpiše naslov nekega filma, ki si ga je že ogledal, ali pa vpiše ključne besede, ki opišejo skupino filmov, ki ga zanimajo, in mu tako mi s pomočjo uporabe algoritmov iz umetne inteligence in podatkov, ki smo jih shranili v podatkovno bazo, priporočamo filme, ki bi uporabniku lahko bili všeč in bi si jih želel ogledati. Uporabnik ima tudi možnost izbire brskanja po filmih, ki so razdeljeni v kategorije, če ga takšen način izbire bolj zanima. V načrtu je bilo tudi to, da bi lahko prijavljenim uporabnikom glede na njihov profil in nastavitve, ki so si jih tam izbrali, priporočili filme, ki si še jih niso ogledali, na podlagi njihovih lastnih favoritov ali pa mu ponudili seznam filmov, ki bi ga utegnili zanimati glede na podobnosti okusa z drugimi uporabniki.

Za priporočanje filmov glede na druge uporabnike smo si izbrali algoritem Apriori, a se je članica, ki ji je bil algoritem dodeljen (Urška Nemet), odločila, da ga ne bo implementirala.

Podobno se je zgodilo z priporočanjem glede na gledane filme uporabnika samega, za kar smo si izbrali Pearsonov algoritem, a se je ponovno dodeljeni član (Dominik Šbüll) odločil proti njegovi implementaciji v spletni portal.

Za predlaganje filmov glede na posamezen že ogledan film, smo uporabili TF-IDF rankirno metriko. Z implementacijo le-tega se je ukvarjala Anja Hauptman, na podlagi podatkov, prenesenih iz spletne strani www.kolosej.si, kjer so polek vseh filmov napisani tudi njihovi krajši opisi. Iz teh podatkov so se vzele posamezne besede, za katere sta se izračunali tf in idf vrednosti, na podlagi katerih smo se lahko odločili, kako pomembne so posamezne besede za vsak film. Uporabnikom na podlagi teh podatkov shranjenih v podatkovni bazi nato priporočamo tiste filme, ki se po prej omenjenih besedah in njihovi pomembnosti najbolj ujemajo s filmom, ki ga je vpisal. Na podobnem principu sestoji tudi iskanje po ključnih besedah o filmu.

Za gručenje podatkov v bolj smiselne kategorije smo si izbrali Bayesov algoritem, ki ga je implementirala Simona Siljanovska. Uporabila je tf-idf podatke, ki jih je pred tem Anja že naložila v podatkovno bazo in na podlagi besed in kako so bile porazdeljene glede na filme, ustvarila okoli dvajset različnih kategorij, v katere je porazdelila filme. Še večja uporabnost algoritma pa je ta, da se bodo, ob dodajanju novih filmov v podatkovno bazo, novi filmi sami primerno razporedili v določene kategorije.

Natančneje sta algoritma opisana v nadaljevanju dokumenta.

Pri pisanju poročila sva sodelovali Anja in Simona.

# TF-IDF

Izvedba tf-idf rangirnega sistema je potekala po naslednjih točkah:

1. V okviru predmeta PPJ sem iz Kolosejeve spletne strani snela podatke o vseh filmih, ki jih imajo v bazi znanja. Ti podatki so vsebovali tudi opise posameznih filmov, ki sem jih uporabila za to nalogo.

Ker so bili podatki že naloženi v podatkovno bazo, sem iz baze s SQL poizvedbo v XML datoteko shranila ID-je vseh filmov in njihove opise. Tako sem si olajšala delo in sem lahko za vsak opis shranila posamezne besede in shranila tudi kateremu filmu pripadajo.

1. Opise sem najprej želela preoblikovati v bolj splošno obliko, zato sem z lematizacijo preoblikovala besede v njihove osnovne oblike (primer: besedo »obiskovalci« proces lematizacije preoblikuje v »obiskovalec«. Enako stori z besedo »obiskovalcev« ipd.). Za to sem uporabila orodje LemmaGen (<http://lemmatise.ijs.si/>), ki sem ga implementirala v svoj C# projekt.
2. Ko so bile vse besede v njihovih osnovnih oblikah, sem opise filmov razdelila na posamezne besede in za njih izračunala vrednosti tf (Term Frequency) – torej število kolikokrat se posamezna beseda pojavi v opisu filma.
3. Ko sem končala s preštevanjem vseh besed, sem izračunala še idf (Inverse Document Frequency) vrednosti za besede – pri tem vzamem število vseh dokumentov (v mojem primeru število opisov filmov - N) in jo delim z številom, v koliko dokumentih se beseda še pojavi (X). Nato to vrednost še logaritmiram (s poljubno osnovo, jaz sem uporabila ln).
4. Vrednost tf-idf se izračuna z množenjem vrednosti tf in idf. Višje kot je to število, bolj pomembna je beseda za opis filma.
5. Pridobljene in izračunane podatke sem dodala v podatkovno bazo spletnega portala, od kod jih uporabljamo pri rangiranju iskalnih rezultatov.

Primer:

Imejmo stavek »Na ta lep sončen dan sem videla lepo rožo.«.

S procesom lematizacije se je stavek preoblikoval v: »na to lepo sončno dan sem videti lepo roža«.

Če ta stavek razdelimo na posamezne besede dobimo:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Beseda* | **Na** | **To** | **Lepo** | **Sončno** | **Dan** | **Sem** | **Videti** | **roža** |
| *Tf* | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 |
| *Idf* |  |  |  |  |  |  |  |  |
| *idf* | 0,92 | 0,92 | 1,6 | 0,92 | 1,6 | 0,51 | 0,22 | 0,92 |
| *Tf-idf* | **0,92** | **0,92** | **3,2** | **0,92** | **1,6** | **0,51** | **0,22** | **0,92** |

Nato preštejemo kolikokat se posamezne besede pojavijo v stavku in to zapišemo kot vrednost tf.

Vrednost idf izračunamo glede na ostale stavke v sklopu obdelave. Recimo da se besede pojavijo še v drugih stavkih. Recimo da imamo še 5 drugih stavkov (N = 5) in X predstavlja v koliko stavkih se določena beseda še pojavi.

Po izračunu obeh vrednosti za vse besede, moramo vrednosti tf in idf zmnožiti.

Po izračunih opazimo, da sta za naš primer najpomembnejši besedi »lepo« in »dan«, saj imata najvišji tf-idf vrednosti, najmanj pomembna pa je beseda »videti«, saj je njena tf-idf vrednost najmanjša.

Po izračunu vseh podatkov je sledila njihova implementacija v spletni portal.

Ker sem podatke v podatkovno bazo že dodala, je večina implementacije potekala v obliki SQL poizvedb.

V portalu sem rangirno metriko TF-IDF uporabila na dveh mestih: ko uporabnik vpiše film, ki mu je všeč in mu z metriko predlagamo filme, ki so temu podobni in tudi pri iskanju filmov, ki so povezani s ključnimi besedami, ki jih uporabnik vpiše.

Najprej sem implementirala prvi problem. Na spletni strani imamo omogočeno iskanje po imenu filma, kamor uporabnik vpiše ime filma, ki si ga je že ogledal in mu je všeč. S tem imenom opravim poizvedbo v SQL bazo, kjer dobim ID filma in s to vrednostjo, lahko opravim klic procedure, ki sem jo napisala za pridobivanje podatkov.

V proceduri si najprej ustvarim začasno tabelo v katero shranim lematizirane besede iz opisa filma (ki jih dobim iz TF-IDF tabele). Nato s pomočjo te tabele izvedem poizvedbo po TF-IDF tabeli in iščem, kateri drugi filmi vsebujejo besede v tabeli. Te filme združim v skupine, da vidim, kateri filmi vsebujejo največ besed iz tabele in jih uredim po padajoči vrednosti (vrednost ujemanja dobim s seštevkom tf-idf vrednosti besed, ki se ujemajo).

Rezultat procedure je izvleček iz tabele, ki vsebuje podatke o filmih. Na spletni strani prikažem po 6 filmov, od tistega, ki se najbolj ujema, do tistega z najmanjšim ujemanjem.

Implementacija drugega dela algoritma je potekala po podobnem postopku kot prvi del. Tukaj od uporabnika zahtevamo ključne besede, za katere želi, da jih film opiše ali vsebuje, ločene s presledki. String z besedami pošljem kot vhod v drugo, a podobno, proceduro.

Ta najprej vhodni podatek razdeli glede na presledke, da iz njega dobi posamezne besede. Te besede nato lematiziram s poizvedbo v bazi, saj imamo tam shranjen slovar za lematizacijo, in njihove lematizirane oblike damo v enako razpredelnico kot v prejšnjem koraku. Nato ponovim korake od prej - torej poizvedbo po TF-IDF tabeli in urejanje po tf-idf vrednosti.

# Viri

[http://en.wikipedia.org/wiki/Tf–idf](http://en.wikipedia.org/wiki/Tf%E2%80%93idf)

<http://www.tfidf.com>

<http://stevenloria.com/finding-important-words-in-a-document-using-tf-idf/>

<http://lemmatise.ijs.si/>