Sistem vsebinskega priporočanja dokumentov kot izboljšava funkcionalnosti v digitalni knjižnici Univerze v Mariboru

Mladen Borovič, Milan Ojsteršek

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko,
Smetanova 17, 2000 Maribor
{mladen.borovic, ojstersek}@uni-mb.si

Povzetek

Prispevek predstavlja sistem vsebinskega priporočanja dokumentov v digitalni knjižnici Univerze v Mariboru. S pomočjo takšnega sistema lahko uporabnikom ponudimo več vsebine, hkrati pa izboljšamo nekatere že obstoječe funkcionalnosti v digitalni knjižnici. Sistem priporočanja obdeluje vsebino dokumentov in pri razvrščanju zadetkov upošteva tudi uporabniške aktivnosti. V prispevku podrobneje opišemo kako sistem obdela besedila in razvrsti zadetke po podobnosti. Podamo rezultate meritev vplivov različnih obdelav besedila na kvaliteto priporočil. Na koncu podamo nekaj idej za izboljšave funkcionalnosti v digitalni knjižnici.

A content-based recommender system and its role in functionality improvements of the Digital Library of University of Maribor

This article presents a content-based document recommender system in the digital library of University of Maribor. By using this kind of a system, we can offer users more content while simultaneously enhancing some established features of the digital library. In the system workflow, the documents are processed and user activities are also taken into account in the document similarity ranking process. The article describes in detail, the document processing, as well as the document similarity ranking process. We provide the measurement results which show the effect of different document processing techniques on recommendation quality. Some ideas where to use the functionality of this system in the digital library, apart from recommending documents, are also provided.

1. Uvod

Sistemi priporočanja so del vsakdanje izkušnje na spletu, predvsem pri spletnih trgovinah in spletnih iskalnikih. Glavni cilj teh sistemov je ponuditi uporabniku vsebine, ki bi ga najbolj zanimale. Motivacija za to je lahko različna. Spletne trgovine uporabljajo takšne sisteme, da bi povečale predstavitev ponudbe. Pri spletnih iskalnikih iskalnikih je cilj nuditi poosebljeno iskanje in tako uporabniku dostaviti najbolj ustrezne zadetke. Sistem priporočanja mora torej na inteligentni način razpoznati uporabnikovo zanimanje zgolj na podlagi njegovih aktivnosti na spletni strani.

Pristopi priporočanja se delijo na dve skupini. Prva skupina pristopov deluje nad uporabniškimi aktivnostmi. Vanjo spadajo sodelujoče filtriranje *(angl. collaborative filtering)*, pristopi z binarnimi vektorji (Melville in Sindhwani, 2010) in algoritmi Slope One (Lemire in Maclachlan, 2005). Druga skupina pristopov priporočanja deluje zgolj nad vsebino, uporabniške aktivnosti pa nimajo poglavitne teže in se tako navadno uporabljajo za dodatne uteži pri razvrščanju rezultatov. Te metode se uporabljajo tudi na področju pridobivanja informacij *(angl. information retrieval)*. Med bolj znane spadajo metoda BM25 (Garcia, 2011), latentna semantična analiza (LSA) (Deerwester et al., 1990) in druge, ki so bile izpeljane iz podobnih predpostavk.

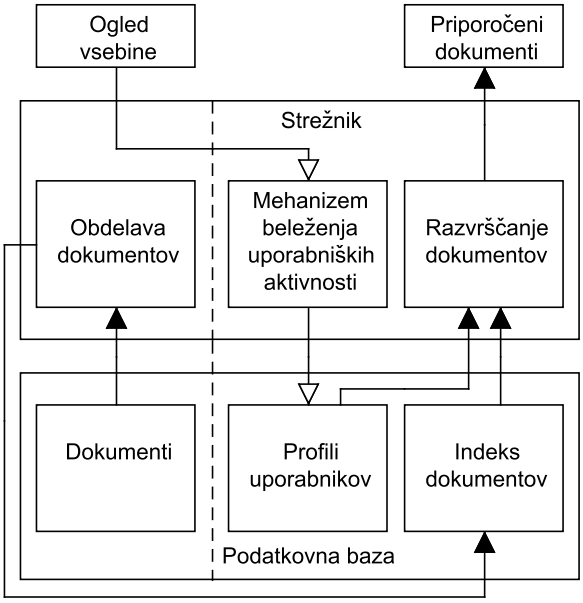
V tem članku predstavimo sistem vsebinskega priporočanja dokumentov v digitalni knjižnici Univerze v Mariboru (v nadaljevanju DKUM) in pokažemo kako posredno in neposredno izboljšuje njene trenutne funkcionalnosti. Pokažemo tudi, da se lahko sistemi priporočanja uporabijo tudi v druge namene. Izhajamo iz dejstva, da je delovanje sistema priporočanja podobno delovanju spletnega iskalnika. Obema je skupno, da vračata najbolj ustrezne rezultate glede na vhodni vnos. Razlikujeta se le v načinu podajanja vhodnega vnosa. Iskalni pojem kot vhod v spletni iskalnik največkrat vnese uporabnik, medtem ko se pri sistemu priporočanja kot vhod ovrednoti uporabnikova aktivnost. Kadar izvajamo vsebinsko priporočanje, pravzaprav tvorimo značilke iz besedila. Slednje počnejo tudi spletni iskalniki. Čeprav imajo uporabnikove aktivnosti pri tem stransko vlogo pa jih lahko še vedno uporabimo kot dodatne uteži, da bi še bolje filtrirali ustrezne vsebine in se s tem približali ideji poosebljenega iskanja.

V prispevku najprej opišemo sistem vsebinskega priporočanja dokumentov v DKUM. Podrobneje opišemo strukturo sistema, uporabljene obdelave besedila in postopek razvrščanja zadetkov. Sledi analiza vpliva različnih obdelav besedila na kvaliteto priporočil. Zatem opišemo možne uporabe sistema priporočanja v druge namene - natančneje pri iskanju potencialnih plagiatov in svetovanju pri izbiri mentorjev zaključnih del. V zaključku podamo smernice za nadaljnje delo.

2. Sistem vsebinskega priporočanja

Naš sistem tvorijo strežnik in spletne storitve za dostop do strežnika. Strežnik je sestavljen iz treh ključnih komponent, ki omogočajo priporočanje (slika 1). Z modulom za

beleženje uporabniških aktivnosti pridobimo informacije o uporabniških aktivnostih kot so število ogledov, število prenosov in ocene gradiv. Modul obdelave dokumentov zagotavlja poenoteno predstavitev vseh dokumentov v obliki naslova, ključnih besed in povzetka. Hkrati se v tem modulu izvaja izračun ocen BM25, ki tvori indeks dokumentov. Slednjega lahko predstavimo kot matriko podobnosti vseh dokumentov. Strežnik izvaja obdelavo dokumentov periodično, saj se v zbirko dokumentov dodajajo novi dokumenti. Na tak način posodabljamo indeks dokumentov. Podatki o uporabnikih in izračunane podobnosti med dokumenti so vhod v modul razvrščanja dokumentov, kjer izberemo podobne dokumente glede na dokument, ki si ga uporabnik trenutno ogleduje.



Slika 1: Struktura sistema priporočanja dokumentov.

2.1. Obdelava besedila

Vhodne dokumente smo predstavili z naslovi, ključnimi besedami in povzetki. Iz besedila vhodnih dokumentov smo najprej odstranili pogoste besede, ki ne vsebujejo informacije o njegovem pomenu. To so slovnične besedne zveze med katere spadajo vezniki, predlogi, členki in medmeti. Seznam pogostih besed smo tvorili sami s štetjem besed v dokumentih, ki so trenutno v DKUM. Nadalje smo uporabljali dve vrsti obdelave - lematizacijo in avtomatsko pridobivanje ključnih besed s pomenskim označevanjem. Tako lematizator kot tudi pomenski označevalnik sta plod dela naše razvojne skupine in ju uporabljamo po principu črne škatle.

2.1.1. Lematizacija

Lematizacija (*angl. lemmatisation*) ali geslenje je postopek določanja osnovne slovarske oblike (leme) besedam v besedilu (Brezovnik, 2009). Lematizaciji zelo podoben postopek je krnjenje. Glavna razlika med lematizacijo in krnjenjem je v tem, da krnjenje besede ne pretvori v slovarsko obliko, ampak preprosto odreže končnico besede tako, da ostane le krn. Pri besedilnem rudarjenju se lematizacija lahko uporablja pri odkrivanju kontekstov besedil. Tako si z lematizacijo olajšamo delo, saj sklepamo, da bo lema obdržala pomen besede.

2.1.2. Avtomatsko pridobivanje ključnih besed s pomenskim označevanjem

Za opis tematike dokumenta ponavadi vzamemo ključne besede, podane s strani avtorja ali knjižničarja. S slednjimi lahko uvrstimo dokument v ustrezne kategorije. Nekatere ključne besede pa niso vedno primerne, saj so preveč splošne ali pa preveč specifične. V tem primeru poznamo rešitev v obliki avtomatskega pridobivanja ključnih besed. Postopek deluje tako, da kot vhod podamo celoten dokument ali pa le en odsek, na izhodu pa dobimo besede, ki opisujejo tematiko vhodnega besedila. Uporabili smo postopek, ki uporablja pomensko označevanje s pomočjo Wikipedije (Burjek, 2011). Pridobivanje ključnih besed poteka v naslednjih korakih:

I. Z algoritmom za iskanje besed najdi primerne besede za pomensko označevanje.

II. Razvrsti besede glede na podatek o temah, na katere je posamezna beseda kazala v Wikipediji.

III. S pomočjo algoritma za izračun sorodnosti, razločevalnika in klasifikatorja naučenega z algoritmom C4.5, izračunaj splošnosti besed.

IV. Za vsako besedo izračunaj vrednost za verjetnost, da bi bila beseda povezava na temo, če bi se pojavila v Wikipediji.

V. Vrni seznam besed, urejenih po verjetnostih izračunanih v koraku IV.

Tabela 1: Primer rezultata pridobivanja ključnih besed s pomenskim označevanjem.

Vhod
V diplomskem delu obravnavamo problematiko pisanja tehničnih vsebin in objavljanj le-teh na spletu. V teoretičnem delu predstavimo slovnico označevalnih jezikov LaTeX in MathML. V praktičnem delu smo izdelali spletni urejevalnik WYSIWYG, ki omogoča vnos matematičnih in kemijskih formul, zapisanih v označevalnem jeziku MathML.
Izhod
MathML, WYSIWYG, LaTeX, Jezik, Slovnica, Splet, Tehnika

V koraku I pred izbiranjem primernih besed za pomensko označevanje uporabimo parameter p_{min}, ki predstavlja minimalno zahtevano verjetnost, da je beseda označena kot povezava v Wikipediji. Ta parameter lahko nastavljamo

poljubno, avtor pa svetuje vrednost 0.85, kar pomeni 85% verjetnost, da gre za povezavo v Wikipediji. Verjetnosti se izračunajo z enačbo 1, kjer $n_l(w)$ pomeni število pojavitev kot beseda, $n_W(w)$ pa število vseh pojavitev v Wikipediji.

$$p_l(w) = \frac{n_l(w)}{n_W(w)} \tag{1}$$

Po koraku I so znane besede, ki so se pojavile v besedilu, ne pa tudi njihov pomen. Zato v koraku II ugotavljamo pomen besed na podlagi tematik, v katerih so se pojavile. Besede lahko imajo več pomenov, zato je treba ugotoviti katere so enopomenske in katere večpomenske. To storimo tako, da ovrednotimo povezavo med besedami in tematikami glede na pogostost uporabe. Tukaj igrajo glavno vlogo verjetnosti, da so besede povezave v Wikipediji. V kolikor je verjetnost besede večja od minimalne zahtevane verjetnosti, jo smatramo kot enopomensko in tako določimo pomen besede.

V koraku III s pomočjo razločevalnika izračunamo verjetnosti mišljenih pomenov pri ostalih besedah. Razločevalnik izračuna verjetnost s pomočjo splošnosti *angl. commonness*, sorodnosti in podatkov s konteksta. Pri kontekstu se upošteva kvaliteta konteksta; če so enopomenske besede v istem dokumentu sorodne, je pri določanju pomena večpomenskih besed bolj pomembna sorodnost, v primeru slabe sorodnosti pa je bolj pomembna splošnost. Splošnost je dana z enačbo 2 in predstavlja inverz verjetnosti, da je beseda povezava v Wikipediji.

$$c(w) = \frac{n_W(w)}{n_l(w)} \tag{2}$$

Sledi še korak IV, kjer se za vsako besedo izračuna vrednost za verjetnost, da bi bila beseda povezava na temo, če bi se pojavila v Wikipediji. Izračun poteka s pomočjo algoritma C4.5, ki tvori odločitveno drevo s katerim klasificiramo primerne besede. Uporabljen pomenski označevalnik je zasnovan na odprtokodni rešitvi Wikipedia Miner, kjer se prav tako uporablja algoritem C4.5, zanimivo pa bi bilo preizkusiti tudi kakšen drug algoritem za gradnjo odločitvenih dreves. Vhod v ta algoritem so torej pojavitve besed, izhod razločevalnika (verjetnost, da je pomen pravilen), sorodnost z ostalimi tematikami, verjetnost povezave, globina tematike v hierarhiji in pozicija besede v besedilu (pojavitev na enem mestu ali razpršena pojavitev). Vse verjetnosti za besede, ki so manjše od p_{min}, se odstranijo iz seznama. S korakom V se seznam preostalih besed uredi po verjetnostih.

2.2. Razvrščanje zadetkov

Za razvrščanje zadetkov smo uporabili metodo razvrščanja BM25 skupaj z dodatnimi utežmi, ki so pridobljene z metapodatki dokumentov in opazovanjem aktivnosti uporabnikov.

2.2.1. BM25

BM25 (*Best Match 25*) je metoda razvrščanja, ki omogoča razvrščanje dokumentov po podobnosti na podlagi besed, ki se pojavljajo v dokumentih. BM25 v bistvu ni samo ena funkcija, temveč družina več funkcij, ki se razlikujejo po utežnih shemah in vrednostih parametrov pomembnosti za uteži. Največkrat se uporabljata uteži tf in idf. Utež tf *(angl. term frequency)* predstavlja frekvenco določene besede v dokumentu, utež idf *(angl. inverse document frequency)* pa pomembnost besede glede na celotno zbirko dokumentov.

$$tf(t,d) = ||n : t \in d|| \tag{3}$$

$$n(t) = ||d \in D : t \in d|| \tag{4}$$

$$idf(t) = \log \frac{||D|| - n(t) + 0.5}{n(t) + 0.5} \tag{5}$$

$||D||$ v enačbi 5 predstavlja kardinalnost (velikost) zbirke D, $n(t)$ dan z enačbo 4 pa število dokumentov, ki vsebujejo besedo t. Ocena BM25 ($s(d,q)$) je odvisna od uteži tf in idf ter parametrov k_1 in b. Splošna enačba izračuna ocene BM25 za dokument d glede na zahtevo q z besedami q_i je podana z enačbo 6.

$$s(d,q) = \sum_{i=1}^{||q||} idf(q_i) \cdot \frac{tf(q_i,d) \cdot (k_1 + 1)}{tf(q_i,d) + k_1 \cdot B}, \quad q_i \in q \tag{6}$$

2.2.2. Parametra k_1 in b

Ključno vlogo imata parametra k_1 in b, ki uravnavata težo uteži tf in težo dolžine dokumentov v končnem izračunu. Dolžina dokumentov se meri s številom besed. Parametra sta bila utemeljena zaradi dveh predpostavk o značilnostih, ki se pojavljajo pri pisanju dokumentov. Predpostavka o širini vsebine dokumenta (*angl. verbosity hypothesis)* govori o tem, da je lahko dokument daljši zaradi uporabe nepomembnih ali redundantnih besed, medtem ko predpostavka o obsegu dokumenta (*angl. scope hypothesis*) govori o daljših dokumentih zaradi dejanske koristne vsebine. V praksi gre za kombinacijo teh dveh predpostavk, zato potrebujemo ustrezno normalizacijo. Dolžino vsakega dokumenta lahko normaliziramo s povprečno dolžino dokumentov (enačba 8). Nadalje lahko to normalizacijo reguliramo s parametrom b kot kaže enačba 9.

$$||d|| = \sum_{i=1}^{n} tf(t_i,d), \quad t_i \in d \tag{7}$$

$$A = \frac{\sum_j^{||D||} ||d_j||}{||D||} \tag{8}$$

$$B = 1 - b + b \cdot \frac{||d_j||}{A} \tag{9}$$

Parameter k_1 uravnava pomembnost uteži tf, parameter b pa pomembnost dolžine dokumentov. Vrednosti parametrov k_1 in b lahko dobimo z optimizacijskimi postopki, navadno pa se uporabijo vrednosti $k_1 \in [1.2, 2.0]$ in $b = 0.75$.

2.2.3. Dodatne uteži

Vsak dokument je ob vsebini predstavljen tudi z metapodatki. Ti vsebujejo podatke o organizaciji, mentorju, študijskem programu, letu izida, primarnem jeziku ipd. Skupaj s temi podatki smo uporabili tudi uporabniške aktivnosti, ki se odražajo v številu ogledov, številu prenosov in

Tabela 2: Obdelava besed in tokenizacija besedne zveze ob lematizaciji

obdelava besed	porazdeljenega → porazdeljen računalniškega → računalniški sistema → sistem
obdelava besednih zvez	porazdeljenega računalniškega sistema → porazdeljen_računalniški_sistem

povprečni oceni dokumenta. Tako uporabljamo več kriterijev za razvrščanje, ki jih združimo v strategijo razvrščanja, v kateri ima vsak kriterij svojo prioriteto. Slednja je predstavljena s tabelo 3.

Tabela 3: Strategija razvrščanja rezultatov priporočanja.

pomembnost	**utež**
1	ocena BM25
2	leto izida
3	število prenosov
4	število ogledov
5	povprečna ocena
6	fakulteta
7	mentor
8	študijski program

2.2.4. Odstranjevanje neustreznih zadetkov z dinamično mejo

Včasih se lahko zgodi, da za rezultat ne bomo dobili dobrih priporočil. Razlog je v raznolikosti gradiv, ki nam ne zagotavlja, da obstaja toliko podobnih dokumentov, kolikor je naše minimalno želeno število priporočil. V tem primeru je potrebno razviti naknadno filtriranje z dinamično mejo. Če je v seznamu priporočil zaznana velika razlika med vrednostmi ocen BM25, lahko iz seznama odstranimo vse zadetke, ki se pojavijo za zaznano veliko razliko. S tem zagotovimo le pomensko ustrezne zadetke. Primer filtriranja z dinamično mejo je podan v tabeli 5, kjer je vrednost minimalnega želenega števila zadetkov enaka 5. Dinamična meja je postavljena na polovico največje ocene, kar lahko interpretiramo kot upoštevanje vseh zadetkov, ki so za več kot 50% pomensko povezani z najustreznejšim zadetkom. Nato izračunamo razlike Δ_i med največjo oceno in vsako naslednjo oceno. Iz seznama odstranimo tiste zadetke, kjer je razlika ocen večja od dinamične meje.

3. Vpliv različnih obdelav besedila

Pri našem delu smo preučevali obnašanje sistema pri različnih obdelavah vhodnih besedil. Preverjali smo kvaliteto rezultatov ob obdelavi na nivoju besed in besednih zvez. Ker so metode, ki smo jih uporabljali, namenjene obdelavi besed, smo upoštevanje besednih zvez omogočili tako, da smo besedno zvezo prepoznali in jo tokenizirali. Tako smo dobili tokenizirano besedno zvezo, ki se je v metodi za izračun podobnosti obnašala kot beseda. Primerjavo obdelave besed in tokenizacije besednih zvez prikazuje tabela 2.

Hkrati smo preučevali vpliv lematizacije. Podobno smo preučevali tudi vpliv avtomatskega pridobivanja ključnih besed s pomenskim označevanjem. Uspešnost je bila zmerjena na 80 naključno izbranih dokumentih v DKUM, kjer smo se omejili na gradiva iz Fakultete za elektrotehniko, računalništvo in informatiko Univerze v Mariboru (v nadaljevanju UM-FERI). Za vsako gradivo smo vrnili 5 priporočil in ugotavljali ustreznost priporočila za človeka. Ustreznost smo merili tako, da je skupina izbranih ljudi ocenjevala seznam priporočil z oceno med 0 in 5. Ta ocena pomeni število za človeka ustreznih zadetkov v seznamu priporočil. Ocene smo za vsako kombinacijo obdelav sešteli in izračunali odstotek ustreznosti. Med ocenjevalci so bili študenti zaključnih letnikov študijskih programov 1. in 2. bolonjske stopnje, kot tudi nekateri zaposleni na UM-FERI. Te ljudi smo smatrali kot domenske strokovnjake. Ocenjevalci so za vsak dokument dobili le informacije o naslovu in ključnih besedah.

Tabela 4: Rezultati kvalitete rezultata pri različnih obdelavah vhodnih dokumentov.

	brez lem.	**z lem.**
besede	51.80%	**55.70**%
besedne zveze	52.60%	**59.70**%
besede + pom. ozn.	47.40%	46.20%
besedne zveze + pom. ozn.	48.60%	**55.50**%

Dobljeni rezultati so pokazali, da se z vidika kvalitete priporočanja bolje obnese obdelava besedil na nivoju besednih zvez, kot tudi obdelava z lematizacijo. Slednja je bistveno pripomogla k boljšemu odstotku ustreznosti. Avtomatsko pridobivanje ključnih besed se je v večini primerov izkazalo kot dober način bogatenja konteksta v opisu dokumenta. Ob podrobnejši analizi spreminjanja vrednosti parametra p_{min} pri pomenskem označevalniku menimo, da bi lahko še povečali faktor vpliva na končen rezultat.

Tabela 5: Filtriranje z dinamično mejo; neustrezen zadetek izpade iz seznama zadetkov, saj ima meja vrednost 6.32.

Naslov vhodnega dokumenta			
Izdelava WYSIWYG urejevalnika s podporo matematičnim in kemijskim znakom			
Seznam zadetkov			
i	**Naslov dokumenta**	**Ocena BM25**	Δ_i
1	Izdelava e-dokumentov s programskim jezikom LaTeX	12.64	0
2	Objava znanstvenih dokumentov na spletnih straneh	10.91	1.73
3	Urejanje in obdelava besedil	7.05	5.59
4	Analiza uporabnosti urejevalnikov besedil v poslovnih sistemih	6.85	5.79
5	Virtualna radijska novinarska redakcija	5.93	**6.71**

4. Uporaba priporočanja v druge namene

Priporočanje dokumentov lahko uporabimo tudi za druge namene. Zaradi narave rezultata (to je seznam podobnih dokumentov) lahko priporočanje dokumentov uvrstimo v delovni tok kot korak obsežnejšega procesa. Primer takšnega procesa sta recimo odkrivanje potencialnih plagiatov in svetovanje pri izbiri mentorjev zaključnih del.

4.1. Izboljšava iskalnika

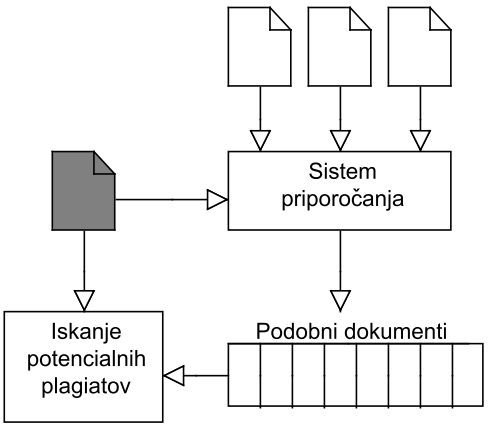
Trenutni iskalnik na DKUM deluje na podlagi ujemanja iskalnega niza z naslovi, ključnimi besedami in avtorji gradiv. Pri tem nekatere ustrezne zadetke uvršča nižje, saj ne upošteva sinonimov in drugih pomenskih informacij. Trenutni iskalnik prav tako ne smatra sklanjatev vhodnih besed kot ene besede, temveč kot različne. Posledično vrača le tista gradiva, ki vsebujejo točno podano sklanjatev ali pa jo imajo vsebovano v naslovu in ključnih besedah kot podniz. Iskalnik bi lahko izboljšali z upoštevanjem zadetkov, ki jih najde sistem priporočanja. Slednji je odporen na problem sklanjatev zaradi uporabe lematizacije in upošteva pomen dokumenta, ne le ujemanja nizov.

4.2. Iskanje potencialnih plagiatov

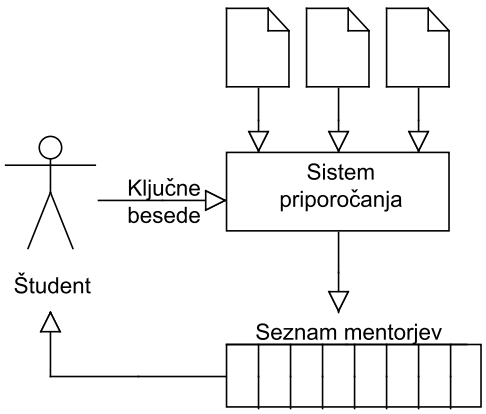
Pri iskanju potencialnih plagiatov navadno delamo primerjave med vsemi dokumenti. Z vključitvijo delovanja priporočanja dokumentov bi lahko eliminirali dokumente, ki si niso podobni, kar bi zmanjšalo obdelovano množico. S tem bi ustvarili seznam filtriranih dokumentov, katere naj sistem za iskanje potencialnih plagiatov upošteva pri nadaljnji obdelavi. Priporočanje dokumentov služi kot predhodna obdelava dokumentov, ki vrne le najbolj podobne dokumente in s tem pohitri dejanski postopek odkrivanja potencialnih plagiatov nad njimi (slika 2).

4.3. Svetovanje pri izbiri mentorja zaključnih del

Še ena aplikacija priporočanja bi lahko bila v obliki orodja, ki študentom svetuje izbiro mentorja pri zaključnih delih. Študent bi v aplikacijo vpisal ključne besede iz področja, ki ga zanima, aplikacija pa bi med diplomskimi, magistrskimi in doktorskimi nalogami našla najbolj ustreznega mentorja (slika 3). To bi potekalo tako, da bi najprej poiskali dokumente, ki so podobni vhodnim ključnim besedam, nato pa bi prešteli, kolikokrat se mentorji pojavljajo v seznamu. V primeru izenačenja bi lahko uporabili še dodatno utež glede na to, kateri dokument ima največjo podobnost z vhodnimi ključnimi besedami.



Slika 2: Vloga sistema priporočanja znotraj procesa iskanja potencialnih plagiatov.



Slika 3: Uporaba sistema priporočanja pri svetovanju mentorja.

5. Zaključek

Prispevek predstavlja sistem vsebinskega priporočanja dokumentov v digitalni knjižnici Univerze v Mariboru. S pomočjo takšnega sistema lahko uporabnikom ponudimo več vsebine, hkrati pa izboljšamo nekatere že obstoječe funkcionalnosti v digitalni knjižnici. Sistem priporočanja obdeluje vsebino dokumentov in pri razvrščanju zadetkov upošteva tudi uporabniške aktivnosti. Predstavili smo tudi notranje funkcije sistema priporočanja v smislu obdelave besedil, pomenskega označevanja in razvrščanja zadetkov, katerih delovanje smo tudi podrobneje opisali. Pokazali smo, da se je najbolje obnesla obdelava besedil na nivoju besed v navezi s pomenskim označevanjem in lematizacijo. V splošnem je obdelava z lematizacijo doprinesla k boljši ustreznosti priporočil. Opisali smo kako lahko predstavljeni sistem priporočanja izboljša trenutne funkcionalnosti DKUM v smislu izpopolnjenega iskanja in potencialnega iskanja plagiatov ter omogoča izdelavo aplikacije za svetovanje mentorjev pri zaključnih delih. Razvit sistem priporočanja se od junija 2012 aktivno uporablja na straneh DKUM.

Nadaljnje delo obsega preučevanje drugih metod za iskanje vsebinske podobnosti. Med te metode spada latentna semantična analiza, katero bi v prihodnosti radi vključili v delovanje sistema, saj omogoča boljšo razpoznavo sinonimov. Rezultat bi tako bila hibridna dvofazna funkcija razvrščanja, ki bi v prvi fazi grobo filtrirala zadetke s pomočjo BM25, v drugi fazi pa fino filtrirala z latentno semantično analizo in upoštevanjem uporabniških aktivnosti. Prav tako bi želeli preizkusiti kvaliteto delovanja sistema, če bi dokumente predhodno gručili in klasificirali glede na tematiko. Trenutno delovanje bi morda lahko še izboljšali z optimizacijo parametrov BM25 glede na korpus besedil. Želeli bi preveriti vpliv parametrov BM25 na rezultat priporočanja in izvesti uglaševanje teh parametrov s pomočjo nekaterih optimizacijskih postopkov. Predvsem želimo zasnovati še objektivno metriko za ocenjevanje priporočanja, na podlagi števila obiskanih priporočil.

6. Literatura

J. Brezovnik in M. Ojsteršek. 2011. Textproc - a natural language processing framework and its use as plagiarism detection system. *International Journal of Education and Information Technologies*, 1(5):293–300.

J. Brezovnik. 2009. Programsko orodje za procesiranje besedil v naravnem jeziku. Magistrsko delo, Fakulteta za elektrotehniko, računalništvo in informatiko Maribor, Univerza v Mariboru.

M. Burjek. 2011. Wikifikacija vsebin v digitalni knjižnici UM. Diplomsko delo, Fakulteta za elektrotehniko, računalništvo in informatiko Maribor, Univerza v Mariboru.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, in R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

S. T. Dumais. 2004. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38:188–230.

E. Garcia. 2011. A tutorial on Okapi BM25. http://www.miislita.com/information-retrieval-tutorial/okapi-bm25-tutorial.pdf.

G. Hrovat. 2010. Izdelava oblikoslovnega označevalnika za slovenski jezik in primerjava z drugimi rešitvami. Diplomsko delo, Fakulteta za elektrotehniko, računalništvo in informatiko Maribor, Univerza v Mariboru.

D. Lemire in A. Maclachlan. 2005. Slope one predictors for online rating-based collaborative filtering. V: *Proceedings of SIAM Data Mining (SDM'05)*, str. 471–475.

Y. Lv in C. Zhai. 2012. A log-logistic model-based interpretation of TF normalization of BM25. V: *Proceedings of the 34th European Conference on Information Retrieval*.

P. Melville in V. Sindhwani. 2010. Recommender systems. V: *Encyclopedia of Machine Learning*, str. 829–838.

F. Ricci, L. Rokach, in B. Shapira. 2011. Introduction to recommender systems. V: *Recommender Systems Handbook*, str. 1–35.