Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Doktorska disertacija

Hibridno priporočanje vrstilcev univerzalne decimalne klasifikacije

Maribor, november 2023

Mladen Borovič

Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Doktorska disertacija

Hibridno priporočanje vrstilcev univerzalne decimalne klasifikacije

Maribor, november 2023

Mladen Borovič
Mentor: izr. prof. dr. Damjan Strnad
UDK: [004.032.26+81’322.2]:025.45UDC(043.3)

Hibridno priporočanje vrstilcev univerzalne decimalne klasifikacije

Doktorska disertacija

Študent:	Mladen Borovič, mag. inž. rač. in inf. tehnol.
Študijski program:	doktorski študijski program
	Računalništvo in informatika
Mentor:	izr. prof. dr. Damjan Strnad

ZAHVALA

Zahvaljujem se mentorju izr. prof. dr. Damjanu Strnadu za vodenje in pomoč pri izdelavi doktorske disertacije. Hvala tudi dr. Milanu Ojsteršku in sodelavcem iz Laboratorija za heterogene računalniške sisteme in Laboratorija za geoprostorsko modeliranje, multimedijo in umetno inteligenco za vse koristne nasvete. Nazadnje se zahvaljujem še družini in prijateljem za podporo in potrpljenje.

Hibridno priporočanje vrstilcev univerzalne decimalne klasifikacije

Ključne besede: hibridni priporočilni sistemi, univerzalna decimalna klasifikacija, vsebinsko filtriranje, globoke nevronske mreže, obdelava naravnega jezika

UDK: [004.032.26+81'322.2]:025.45UDC(043.3)

Povzetek

V doktorski disertaciji predlagamo hibridni pristop za priporočanje vrstilcev univerzalne decimalne klasifikacije (UDK) za elektronske dokumente, ne glede na globino hierarhije UDK. Razvit hibridni pristop priporočanja vrstilcev UDK temelji na metodah vsebinskega filtriranja in uporablja strukturirane metapodatke v slovenskem jeziku za klasifikacijo področja znanosti in priporočanje ustreznih vrstilcev. Ker se dokumenti pogosto nanašajo na več področij znanosti, mora biti pristop sposoben identificirati interdisciplinarnost in vrniti več ustreznih vrstilcev UDK. Predlagani hibridni pristop uporablja kaskadno hibridizacijo in je razdeljen na dva kaskadna koraka. Najprej z rangirno funkcijo BM25 zagotovimo začetni seznam vrstilcev UDK. V prvem kaskadnem koraku začetni seznam vrstilcev UDK preuredimo s seznamom, ki je rezultat večznačnega klasifikatorja. Večznačni klasifikator temelji na globoki nevronski mreži BERT in je prilagojen na hierarhično topologijo UDK. V drugem kaskadnem koraku s pomočjo seznama najbolj pogostih vrstilcev UDK v organizaciji, iz katere izvira dokument, preuredimo seznam iz prvega koraka. Za kaskadno hibridizacijo se izvedejo postopki naknadne obdelave, ki preuredijo sezname priporočil glede na vrhnje področje in glede na specifičnost, omogočajo pa tudi rezanje seznama. Disertacija vključuje vrednotenje na množici zaključnih del v slovenskem jeziku, ki so del repozitorijev slovenskih univerz in že imajo ročno določene vrstilce UDK s strani knjižničarjev. Na testni množici dokumentov s predlaganano metodo po metriki HR@K dosežemo povprečne vrednosti 0,574 (K = 1), 0,869 (K = 3) in 0,892 (K = 5). Po metriki NDCG@K dosežemo povprečne vrednosti 0,993 (K = 1), 0,921 (K = 3) in 0,916 (K = 5), po metrikah MRR in MAP pa povprečne vrednosti vrednosti 0,782 (MRR) in 0,785 (MAP). V primerjavi z obstoječimi pristopi pokažemo, da uporaba predlaganega pristopa vodi v statistično značilne izboljšave.

Hybrid recommendation of universal decimal classification codes

Keywords: hybrid recommender systems, Universal Decimal Classification, content-based filtering, deep neural networks, natural language processing

UDC: [004.032.26+81'322.2]:025.45UDC(043.3)

Abstract

In the doctoral dissertation, we propose a hybrid approach for recommending Universal Decimal Classification (UDC) notations for electronic documents, regardless of the field of science or the depth of the UDC hierarchy. The developed hybrid approach for recommending UDC notations is based on content filtering methods and uses structured metadata in the Slovenian language for classifying the field of science and recommendation of the appropriate notations. Since documents often relate to multiple fields of science, the approach must be able to identify interdisciplinarity and return multiple relevant UDC notations that can represent different fields of science. The hybrid approach uses the cascade hybridization approach and is divided into two cascading steps. First, the BM25 ranking function is used to provide the initial list of recommended notations for a new document. In the first cascade step, the initial list of recommended notations is re-ranked using the list obtained with a multi-label classifier. The multi-label classifier is based on the deep neural network BERT and is adapted to the hierarchical topology of UDC. In the second cascade step, the resulting list from the first cascade step is re-ranked using a list of most common notations used in the document's source organization. Following the cascade hybridization are the post-processing procedures that re-rank and alter the recommendation lists based on the top-level branches of the UDC hierarchy and specificity, as well as cut-off rate. The dissertation includes an evaluation on a set of theses in the Slovenian language that are part of the repositories of Slovenian universities and have UDC notations manually catalogued by librarians. In the evaluation on a test set of documents we achieve mean values for the metric HR@K of 0,574 (K = 1), 0,869 (K = 3) and 0,892 (K = 5). For the metric NDCG@K we achieve values 0,993 (K = 1), 0,921 (K = 3) and 0,916 (K = 5). For metrics MRR and MAP we achieve values 0,782 (MRR) and 0,785 (MAP). Compared to existing approaches, we show that the use of the proposed approach leads to statistically significant improvements.

Univerza v Mariboru

Fakulteta za elektrotehniko, računalništvo in informatiko
(ime članice UM)

IZJAVA O AVTORSTVU IN ISTOVETNOSTI TISKANE IN ELEKTRONSKE OBLIKE DOKTORSKE DISERTACIJE

Ime in priimek študenta/-ke: Mladen Borovič

Študijski program: RAČUNALNIŠTVO IN INFORMATIKA

Naslov doktorske disertacije: Hibridno priporočanje vrstilcev univerzalne decimalne klasifikacije

Mentor/-ica: izr. prof. dr. Damjan Strnad

Somentor/-ica:

Podpisani/-a študent/-ka Mladen Borovič

- izjavljam, da je zaključno delo rezultat mojega znanstvenoraziskovalnega dela;
- izjavljam, da sem pridobil/-a vsa potrebna soglasja za uporabo podatkov in avtorskih del v zaključnem delu in jih v zaključnem delu jasno in ustrezno označil/-a;
- na Univerzo v Mariboru neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico ponuditi zaključno delo javnosti na svetovnem spletu preko DKUM in drugih informacijskih zbirk in ponudnikov; sem seznanjen/-a, da bodo dela, deponirana/objavljena v DKUM, dostopna široki javnosti pod pogoji licence **Creative Commons BY-NC-ND**, kar vključuje tudi avtomatizirano indeksiranje preko spleta in obdelavo besedil za potrebe tekstovnega in podatkovnega rudarjenja in ekstrakcije znanja iz vsebin; uporabnikom se dovoli reproduciranje brez predelave avtorskega dela, distribuiranje, dajanje v najem in priobčitev javnosti samega izvirnega avtorskega dela, in sicer pod pogojem, da navedejo avtorja in da ne gre za komercialno uporabo;
- dovoljujem objavo svojih osebnih podatkov, vezanih na zaključek študija (ime, priimek, leto zaključka študija, naslov zaključnega dela) na spletnih straneh Univerze v Mariboru in v publikacijah Univerze v Mariboru;
- izjavljam, da je tiskana oblika zaključnega dela istovetna elektronski obliki zaključnega dela, ki sem jo oddal/-a za objavo v DKUM;
- izjavljam, da sem seznanjen s pogoji Proquesta za oddajo in javno objavo doktorske disertacije v podatkovno zbirko ProQuest Dissertations & Theses Global (http://contentz.mkt5049.com/lp/43888/382619/PQDTauthoragreement.pdf).

Uveljavljam permisivnejšo obliko licence Creative Commons: _____
(*navedite obliko*)

Kraj in datum: Podpis študenta/-ke:

Maribor, 04. 08. 2023 _____

KAZALO VSEBINE

KAZALO SLIK . . . . . **III**

KAZALO TABEL . . . . . **VI**

KAZALO ALGORITMOV . . . . . **VII**

SEZNAM KRATIC . . . . . **VIII**

SEZNAM SIMBOLOV . . . . . **IX**

1 Uvod **1**
1.1 Opredelitev raziskovalnega problema 1
1.2 Cilji doktorske disertacije 2
1.3 Predpostavke in omejitve 3
1.4 Teza in hipoteze 4
1.5 Izvirni znanstveni prispevki 4
1.6 Struktura doktorske disertacije 5

2 Metodologije in sorodna dela **6**
2.1 Univerzalna decimalna klasifikacija 6
2.1.1 Struktura UDK 7
2.1.2 Primeri izrazov UDK 10
2.2 Obdelava naravnega jezika 12
2.2.1 Klasifikacija besedil 14
2.2.2 Predstavitev besedila v statističnih metodah 16
2.2.3 Rangirna funkcija BM25 17
2.2.4 Predstavitev besedila v globokih nevronskih mrežah 19
2.2.5 Arhitektura globokih nevronskih mrež transformer 20
2.2.6 Globoke nevronske mreže BERT 28
2.3 Priporočilni sistemi 31
2.3.1 Sodelovalno filtriranje 33
2.3.2 Vsebinsko filtriranje 34
2.3.3 Hibridni priporočilni sistemi 36
2.3.4 Priporočilni sistemi v digitalnih knjižnicah 38
2.3.5 Vrednotenje priporočilnih sistemov 39

2.4 Sorodna dela 48

3 Hibridni pristop za priporočanje vrstilcev UDK 51
3.1 Struktura hibridnega priporočilnega sistema 51
3.2 Pridobivanje začetnega seznama relevantnih vrstilcev UDK 53
3.3 Večznačni klasifikator 55
3.3.1 Arhitektura večznačnega klasifikatorja 57
3.3.2 Glajenje oznak 58
3.4 Kaskadna hibridizacija 59
3.4.1 Prvi kaskadni korak 60
3.4.2 Drugi kaskadni korak 61
3.5 Naknadna obdelava priporočil 62
3.5.1 Preurejanje na podlagi vrhnjega področja 63
3.5.2 Preurejanje na podlagi specifičnosti 64
3.5.3 Rezanje seznama 66

4 Eksperiment in rezultati 68
4.1 Eksperimentalno okolje 68
4.1.1 Podatkovna zbirka 68
4.1.2 Učenje večznačnega klasifikatorja 71
4.1.3 Zasnova primerjav 72
4.2 Rezultati eksperimenta 72
4.2.1 Primerjava 1 73
4.2.2 Primerjava 2 75
4.2.3 Primerjava 3 78
4.2.4 Primerjava 4 80
4.3 Analiza parametrov predlaganega pristopa 81

5 Interpretacija rezultatov in razprava 86
5.1 Potrjevanje zastavljenih hipotez 89

6 Zaključek 92
6.1 Izvirni prispevki k znanosti 93

VIRI IN LITERATURA 94

KAZALO SLIK

Slika 2.1: Primer enostavnega izraza UDK za matematični učbenik. 10
Slika 2.2: Primer enostavnega izraza UDK za diplomsko delo. 11
Slika 2.3: Primer kompleksnega izraza UDK za doktorsko disertacijo. 11
Slika 2.4: Primer kompleksnega izraza UDK za turistični zemljevid. 12
Slika 2.5: Različni tipi klasifikacije besedil s primeri. 15
Slika 2.6: Primeri različnih delitev besedila na besedne n-grame. 16
Slika 2.7: Primer pomikanja med različnimi kontekstnimi vložitvami. 19
Slika 2.8: Arhitektura transformer na najvišjem nivoju. 20
Slika 2.9: Obdelava vhoda skozi sklada kodirnikov (modra) in dekodirnikov (oranžna). 21
Slika 2.10: Podrobna struktura kodirnika (levo) in dekodirnika (desno). 21
Slika 2.11: Prehod vložitev skozi sklad kodirnikov. 23
Slika 2.12: Prikaz izračuna matrik Q, K in V znotraj plasti samopozornosti. 24
Slika 2.13: Prikaz izračuna vektorjev Q_i, K_i in V_i pri izračunu večglave samopozornosti. 25
Slika 2.14: Prikaz združevanja rezultatov večglave samopozornosti. 26
Slika 2.15: Struktura kodirnika in dekodirnika z dodano normalizacijo plasti in rezidualnimi povezavami. 27
Slika 2.16: Pretvorba vektorja nazaj v besedilo z linearno in softmax plastjo. 27
Slika 2.17: Povezane komponente in plasti znotraj transformerjev. 28
Slika 2.18: Učenje modela BERT in prilagoditev modela BERT na specifično nalogo. . 29
Slika 2.19: Komponente priporočilnega sistema, povezane v delovni tok. 31
Slika 2.20: Prikaz delovanja sodelovalnega filtriranja. 34
Slika 2.21: Prikaz delovanja vsebinskega filtriranja. 35
Slika 2.22: Delovni tok v utežni hibridizaciji. 36
Slika 2.23: Delovni tok v preklopni hibridizaciji. 36
Slika 2.24: Delovni tok v mešani hibridizaciji. 37
Slika 2.25: Delovni tok v hibridizaciji s kombinacijo značilk. 37
Slika 2.26: Delovni tok v hibridizaciji z obogatenjem značilk. 37
Slika 2.27: Delovni tok v kaskadni hibridizaciji. 38
Slika 2.28: Delovni tok v hibridizaciji na meta ravni. 38
Slika 2.29: Primer izračuna metrike HR@K pri K = [1; 3; 5] za tri sezname priporočil. 41
Slika 2.30: Prvi del primera izračuna metrike MAP za tri sezname priporočil. 43
Slika 2.31: Drugi del primera izračuna metrike MAP za tri sezname priporočil. . . . 44

Slika 2.32: Primer izračuna metrike MRR za tri sezname priporočil. 45
Slika 2.33: Prvi del primera izračuna metrike NDCG@5 za tri sezname priporočil. . . 46
Slika 2.34: Graf funkcije diskontiranja pri metriki NDCG@K. 47
Slika 2.35: Drugi del primera izračuna metrike NDCG@5 za tri sezname priporočil. . 47
Slika 2.36: Tretji del primera izračuna metrike NDCG@5 za tri sezname priporočil. . 48

Slika 3.1: Arhitektura predlaganega hibridnega pristopa za priporočanje vrstilcev UDK. 52
Slika 3.2: Pridobivanje seznama relevantnih vrstilcev UDK z rangirno funkcijo BM25. 53
Slika 3.3: Primer določanja smiselnih vrstilcev UDK z večrazrednim (levo) in večznačnim (desno) klasifikatorjem za dokument z interdisciplinarno vsebino. . . 56
Slika 3.4: Arhitektura večznačnega klasifikatorja. 57
Slika 3.5: Primer razpoznavanja izraza UDK in njegove delitve na vrstilce UDK. . . . 58
Slika 3.6: Primer glajenja oznak (vrstilcev UDK). 59
Slika 3.7: Postopek kaskadne hibridizacije razdeljene na dva kaskadna koraka. . . . 60
Slika 3.8: Delovanje prvega kaskadnega koraka. 60
Slika 3.9: Delovanje drugega kaskadnega koraka. 62
Slika 3.10: Uporaba in zaporedje postopkov naknadne obdelave priporočil v hibridnem pristopu za priporočanje vrstilcev UDK. 63
Slika 3.11: Delovanje postopka preurejanja na podlagi vrhnjega področja. 64
Slika 3.12: Delovanje postopka preurejanja na podlagi specifičnosti. 65
Slika 3.13: Delovanje postopka rezanja seznama. 66

Slika 4.1: Deleži dokumentov v podatkovni množici glede na vrhnja področja hie-rarhije UDK. 69
Slika 4.2: Deleži dokumentov v podatkovni zbirki glede na dolžino razpoznanih vrstilcev UDK. 70
Slika 4.3: Deleži dokumentov v podatkovni zbirki glede na število razpoznanih vrstilcev UDK. 70
Slika 4.4: Grafična ponazoritev delitve podatkov za večznačni klasifikator in eksperiment. 71
Slika 4.5: Primerjava povprečnih vrednosti metrik HR@K, NDCG@K, MRR in MAP za pristopa v primerjavi 1. 74
Slika 4.6: Primerjava povprečnih vrednosti metrik HR@K, NDCG@K, MRR in MAP za pristope v primerjavi 2. 77
Slika 4.7: Primerjava povprečnih vrednosti metrik HR@K, NDCG@K, MRR in MAP za pristopa v primerjavi 3. 79

Slika 4.8: Primerjava povprečnih vrednosti metrik HR@K, NDCG@K, MRR in MAP za pristopa v primerjavi 4. 81
Slika 4.9: Vrednosti izbranih metrik pri različnih vrednostih parametra $_1$. 83
Slika 4.10: Vrednosti izbranih metrik pri različnih vrednostih parametra $_2$. 84
Slika 4.11: Vrednosti izbranih metrik pri različnih vrednostih parametra 85

KAZALO TABEL

Tabela 2.1: Hierarhična razčlenitev vrstilca 004.738.52 - “Orodja za iskanje v inter-netu.”. 7
Tabela 2.2: Vrstilci vrhnjih področij univerzalne decimalne klasifikacije. 8
Tabela 2.3: Vrstilci v področju 6 - “Uporabne znanosti. Medicina. Tehnika.”. 8
Tabela 2.4: Vrstilci za področje 68 - “Industrije, obrti in rokodelstva za sestavljanje in dodelavo izdelkov.”. 9
Tabela 2.5: Splošni privesni vrstilci v univerzalni decimalni klasifikaciji s primeri. . 9
Tabela 2.6: Znaki za povezovanje v univerzalni decimalni klasifikaciji s primeri. . . 10
Tabela 2.7: Primerjava struktur različnih modelov, ki uporabljajo podobno arhitekturo kot BERT. 30

Tabela 3.1: Pomeni končnih priporočenih vrstilcev v uporabljenem zgledu. 67

Tabela 4.1: Uporabljene vrednosti hiperparametrov pri učenju večznačnega klasifikatorja. 72
Tabela 4.2: Dosežene povprečne vrednosti in standardni odkloni (v oklepajih) metrik HR@K, NDCG@K, MRR in MAP za pristopa v primerjavi 1. 73
Tabela 4.3: Dobljene vrednosti statistične značilnosti p v primerjavi 1. 75
Tabela 4.4: Dosežene povprečne vrednosti in standardni odkloni (v oklepajih) metrik HR@K, NDCG@K, MRR in MAP za pristope v primerjavi 2. 76
Tabela 4.5: Dobljene vrednosti statistične značilnosti p v primerjavi 2. 77
Tabela 4.6: Dosežene povprečne vrednosti in standardni odkloni (v oklepajih) metrik HR@K, NDCG@K, MRR in MAP za pristopa v primerjavi 3. 78
Tabela 4.7: Dobljene vrednosti statistične značilnosti p v primerjavi 3. 80
Tabela 4.8: Dosežene povprečne vrednosti in standardni odkloni (v oklepajih) metrik HR@K, NDCG@K, MRR in MAP za pristopa v primerjavi 4. 80
Tabela 4.9: Parametri predlaganega hibridnega pristopa s priporočenimi vrednostmi. 82

KAZALO ALGORITMOV

Algoritem 3.1: Določanje dolžine seznamov za vrhnja področja hierarhije UDK. . . 54
Algoritem 3.2: Omejitev seznama glede na najpogostejše vrhnje področje v začetnem seznamu priporočil. 55

SEZNAM KRATIC

AUC	—	Area Under the Curve (ROC)
BERT	—	Bidirectional Embedding Representations from Transformers
BM25	—	Best Match 25
BOW	—	vreča besed *(ang. Bag Of Words)*
CBOW	—	zvezna vreča besed *(ang. Continuous Bag Of Words)*
DistilBERT	—	Distilled BERT
FFNN	—	naprej usmerjena nevronska mreža *(ang. Feed-Forward Neural Network)*
GloVe	—	Global Vectors
GPT	—	Generative Pre-trained Transformer
GRU	—	Gated Recurrent Unit
HR	—	razmerje zadetkov *(ang. Hit Ratio)*
LLM	—	veliki jezikovni model *(ang. Large Language Model)*
LSTM	—	dolgi kratkoročni spomin *(ang. Long Short-Term Memory)*
MAP	—	srednja povprečna natančnost *(ang. Mean Average Precision)*
MRR	—	srednji recipročni rang *(ang. Mean Reciprocal Rank)*
NDCG	—	normaliziran diskontiran kumulativni dobiček *(ang. Normalized Discounted Cumulative Gain)*
NLP	—	obdelava naravnega jezika *(ang. Natural Language Processing)*
ReLU	—	usmerjena linearna enota *(ang. Rectified Linear Unit)*
ROC	—	Receiver Operating Characteristic
RoBERTa	—	Robustly optimized BERT approach
SBERT	—	Sentence BERT
SPLADE	—	SParse Lexical AnD Expansion model
SVM	—	metoda podpornih vektorjev *(ang. Support Vector Machines)*
UDK	—	Univerzalna Decimalna Klasifikacija *(ang. Universal Decimal Classification)*
Word2Vec	—	Word to Vector
XMTC	—	večznačna klasifikacija besedil z velikim številom oznak *(ang. eXtreme Multi-label Text Classification)*

SEZNAM SIMBOLOV

Rangirna funkcija BM25		
tf	—	število pojavitev (frekvenca) termina *(ang. term frequency)*
idf	—	število pojavitev (frekvenca) termina v dokumentu *(ang. inverse document frequency)*
d	—	dokument
D	—	korpus dokumentov
$\|D\|$	—	velikost korpusa dokumentov D
Q	—	vhodni iskalni niz
q_i	—	beseda v vhodnem iskalnem nizu Q
$\|Q\|$	—	dolžina vhodnega iskalnega niza Q
k_1	—	parameter rangirne funkcije BM25
B	—	normalizacijski faktor v rangirni funkciji BM25
b	—	parameter normalizacijskega faktorja v rangirni funkciji BM25
l_d	—	dolžina dokumenta d
l_{avg}	—	povprečna dolžina dokumentov v korpusu D
$s(d; Q)$	—	ocena rangirne funkcije BM25
k_3	—	parameter različice rangirne funkcije BM25
qtf	—	število pojavitev (frekvenca) termina v vhodnem iskalnem nizu *(ang. query term frequency)*
$s^0(d; Q)$	—	ocena različice rangirne funkcije BM25
Arhitektura globokih nevronskih mrež transformer		
$PE_{(pos;2i)}$	—	položajno kodiranje besed s sodim indeksom položaja
$PE_{(pos;2i+1)}$	—	položajno kodiranje besed z lihim indeksom položaja
X	—	matrika vložitev
W_Q	—	matrika uteži poizvedb *(ang. query weight matrix)*
Q	—	matrika poizvedb *(ang. query matrix)*
W_K	—	matrika uteži ključev *(ang. key weight matrix)*

	Arhitektura globokih nevronskih mrež transformer *(nad.)*	
K	—	matrika ključev *(ang. key matrix)*
W_V	—	matrika uteži vrednosti *(ang. value weight matrix)*
V	—	matrika vrednosti *(ang. value matrix)*
Z	—	matrika vektorjev samopozornosti *(ang. self-attention vector matrix)*
d_k	—	dimenzija vektorjev v matriki K
W_0	—	matrika uteži za večglavo samopozornost
	Predlagan hibridni pristop priporočanja vrstilcev UDK	
t	—	vrhnje področje hierarhije UDK
S	—	asociativno polje z števili dokumentov za vrhnja področja
P	—	asociativno polje z deleži dokumentov za vrhnja področja
L	—	asociativno polje z dolžinami seznamov za vrhnja področja
k	—	polje z možnimi dolžinami seznamov
b^{min}	—	polje s spodnjimi mejami za deleže dokumentov
b^{max}	—	polje z zgornjimi mejami za deleže dokumentov
U	—	urejen seznam vrstilcev UDK
U^0	—	seznam vrstilcev UDK U, reduciran na vrhnja področja
t_{max}	—	najbolj pogosto vrhnje področje v U^0
UR_0	—	dolžina začetnega seznama vrstilcev UDK
R_1	—	začetni seznam vrstilcev UDK, omejen na elementov
$r^{(R_0)}$	—	urejen začetni seznam relevantnih vrstilcev UDK
$^{(R_1)}_{r}$	—	urejen seznam relevantnih vrstilcev UDK po prvem kaskadnem koraku
L_{hier}	—	vrstilec UDK v seznamu
l_i	—	ocena vrstilca UDK r v seznamu R_0
$sim_{jw}(r; l_i)$	—	ocena vrstilca UDK r v seznamu R_1
$sim_j(a; b)$	—	seznam oznak, ki ga vrne večznačni klasifikator v prvem kaskadnem koraku
‘	—	oznaka v seznamu oznak L_{hier}
p	—	Jaro-Winklerjeva razdalja med vrstilcem r in oznako l_i
L_{org}	—	Jarova razdalja med nizoma a in b
R_2	—	dolžina predpone na začetku niza
org	—	skalirni faktor pri izračunu Jaro-Winklerjeve razdalje
L_{max} (R_2)	—	seznam najbolj pogostih vrstilcev UDK za organizacijo
jR_2j	—	urejen seznam relevantnih vrstilcev UDK po drugem kaskadnem koraku
	—	ocena vrstilca UDK v L_{org}
	—	vrednost največje ocene v L_{org}
	—	ocena vrstilca UDK r v seznamu R_2
	—	dolžina seznama R_2

Predlagan hibridni pristop priporočanja vrstilcev UDK *(nad.)*		
$f_r^{(T)}$	—	faktor dominance pri preurejanju na podlagi vrhnjega področja
r_{rank}^{1}	—	ocena vrstilca UDK r po izvedbi preurejanja na podlagi vrhnjega področja
r_{max}^{r}	—	parameter pri preurejanju na podlagi vrhnjega področja
$l_r^{(S)}$	—	rang vrstilca UDK r
$l_{max}^{(S)}$	—	dolžina vrstilca UDK r v seznamu priporočil
$s_r^{(S)}$	—	dolžina najdaljšega vrstilca UDK v seznamu priporočil
r_{spec}^{r}	—	ocena vrstilca UDK r po izvedbi preurejanja na podlagi specifičnosti
T_{spec}	—	parameter pri preurejanju na podlagi specifičnosti
T_{rel}	—	parameter pri rezanju seznama
n_{hit}	—	ocena najvišje uvrščenega vrstilca UDK r v seznamu priporočil
n_{rel}	—	prag pri rezanju seznama
Metrike		
$P@K$	—	število pravilno napovedanih vrstilcev UDK v seznamu priporočil
$rel(k)$	—	število vseh pravilnih vrstilcev UDK
$AP@K$	—	natančnost pri K *(ang. precision at K)*
N	—	funkcija relevantnosti za k-ti element v seznamu priporočil
$rank_i$	—	povprečna natančnost pri K *(ang. average precision at K)*
$DCG@K$	—	število seznamov priporočil, ki jih obravnavamo z metriko
$IDCG@K$	—	rang prvega ustreznega priporočila v i-tem seznamu priporočil
	—	diskontiran kumulativni dobiček pri K
	—	idealni diskontiran kumulativni dobiček pri K

1 Uvod

Sodobne digitalne knjižnice in digitalni repozitoriji dandanes vsebujejo ogromno dokumentov z različnih znanstvenih področij. Te dokumente je smiselno klasificirati v znanstvena področja z namenom, da bi poenostavili in pohitrili iskanje relevantne literature. Klasifikacija znanstvenega področja dokumenta se tako zgodi v procesu katalogizacije, tj. med obravnavo novega dokumenta v digitalni knjižnici. V ta namen se uporabljajo različni knjižnični klasifikacijski sistemi, v Sloveniji pa uporabljamo sistem univerzalne decimalne klasifikacije (UDK). Slednja opiše razred dokumenta z izrazom, ki je sestavljen iz vrstilcev, predstavljenih z decimalnimi števili. Ker je proces katalogizacije možno podpreti z informacijskimi sistemi, se v tej disertaciji ukvarjamo z idejo o uvedbi priporočanja vrstilcev univerzalne decimalne klasifikacije, ki služi kot podporni sistem pri delu knjižničarjev.

1.1 Opredelitev raziskovalnega problema

Katalogizacija novih dokumentov v knjižničnih sistemih je standardni proces beleženja metapodatkov o dokumentih, ki zajema tudi določanje vrstilcev UDK [1]. Katalogizacijo opravljajo knjižničarji ročno tako, da pregledajo dokument in izluščijo za katalogizacijo relevantne podatke. Pri določanju vrstilcev UDK morajo iz naslova, ključnih besed in povzetka določiti najustreznejši vrstilec UDK, kar v določenih primerih zajema pregled hierarhije kataloga UDK. Določanje vrstilca UDK je tako lahko počasno, površno, presplošno ali pa v skrajnih primerih tudi napačno.

Zaradi teh izzivov so bile razvite metode avtomatske klasifikacije dokumentov z ustreznimi vrstilci knjižničnih klasifikacijskih sistemov [2]. Obstoječe metode najpogosteje zajemajo univerzalno decimalno klasifikacijo in Deweyjevo decimalno klasifikacijo (DDK) [3, 4]. Uveljavljeni pristopi najprej izvedejo pretvorbo dokumentov v obliko vreče besed (ang. bag of words) [5], pri čemer uporabljajo naslove, povzetke, ključne besede ali celotno besedilo dokumenta [6–8]. Tej pretvorbi sledi uporaba različnih metod strojnega učenja, ki izvajajo klasifikacijo. Pogosto uporabljene metode strojnega učenja v tem primeru so naivni Bayesov klasifikator, linearna regresija ali večplastne nevronske mreže [9]. Boljše rezultate dosežemo z uporabo metode podpornih vektorjev [10, 11] in z uporabo globokih nevronskih mrež [12], ki temeljijo na arhitekturi transformer, pri tem pa se dokumenti pretvorijo v obliko kontekstnih vložitev (ang. contextual embeddings) [13, 14]. Ker so nekateri dokumenti interdisciplinarne narave, imajo lahko kompleksne sestavljene vrstilce, ki niso omejeni samo na eno znanstveno področje. Naštete uveljavljene metode

večrazredne klasifikacije se uporabljajo tako, da vrnejo zgolj en vrstilec kot rezultat. To vodi v pomanjkljivo avtomatsko klasifikacijo interdisciplinarnih dokumentov, zaradi česar se manjša zaupanje knjižničarjev vanjo. Prav tako obstoječe metode ne upoštevajo hierarhične strukture kataloga UDK. V digitalnih knjižnicah se tako določanje vrstilcev UDK še vedno izvaja ročno.

Digitalne knjižnice se danes na veliko uporabljajo predvsem zaradi možnosti preprostega iskanja in pridobivanja relevantne literature. V zadnjem času se na tem področju predvsem uporabljajo sistemi za priporočanje podobnih dokumentov [15–20]. V splošnem so se v preteklosti priporočilni sistemi na različnih področjih realizirali bodisi s sodelovalnim filtriranjem (ang. collaborative filtering) [21–23], bodisi z vsebinskim filtriranjem (ang. content-based filtering) [24, 25]. Danes so bolj aktualni hibridni priporočilni sistemi (ang. hybrid recommender systems) [26–28], ki odpravljajo pomanjkljivosti in omejitve metod sodelovalnega in vsebinskega filtriranja [29–33]. V digitalnih knjižnicah se sicer povečini uporablja vsebinsko filtriranje, saj gre za okolja, kjer je sledenje aktivnostim uporabnikov omejeno in sodelovalno filtriranje ne pride v poštev.

Ker je priporočanje dokumentov zaradi svoje uveljavljenosti na področju digitalnih knjižnic knjižničarjem dobro poznan koncept, se je pojavila ideja o polavtomatski katalogizaciji dokumentov [10], ki v proces določanja vrstilcev vključuje uporabo priporočilnih sistemov. Priporočilni sistem v tem scenariju priporoči seznam vrstilcev, knjižničar pa ta seznam uporabi pri ročnem določanju vrstilcev za dokument. Dve pomembni prednosti uvedbe polavtomatske katalogizacije sta pohitritev procesa katalogizacije in zmanjšanje napak pri določanju vrstilcev. Glavni izziv obstoječih pristopov je torej zasnova celovitega priporočilnega sistema, ki knjižničarjem med procesom katalogizacije priporoča seznam ustreznih vrstilcev izbranega knjižničnega klasifikacijskega sistema, pri tem pa uporablja različne načine filtriranja. Sorodna dela obravnavajo ta problem z uporabo vsebinskega filtriranja, različnih klasifikatorjev strojnega učenja ali ontologij, pri čemer se omejujejo le na izbrane veje znanosti, kot tudi na vnaprej določeno globino v hierarhiji izbranih knjižničnih klasifikacijskih sistemov.

1.2 Cilji doktorske disertacije

Cilj doktorske disertacije je zasnova in razvoj hibridnega pristopa priporočanja vrstilcev UDK, ki temelji na metodah vsebinskega filtriranja in se lahko uporabi za polavtomatsko določanje ustreznih vrstilcev univerzalne decimalne klasifikacije za neklasificirane elektronske dokumente, brez omejitev na vejo znanosti ali globino hierarhije UDK. Najprej smo analizirali obstoječe pristope za avtomatsko in polavtomatsko določanje vrstilcev UDK. Sledil je razvoj hibridnega pristopa priporočanja vrstilcev UDK, ki deluje nad do-

kumenti v slovenskem jeziku, obogatenimi s strukturiranimi metapodatki. Zaradi interdisciplinarnih dokumentov mora biti hibridni pristop sposoben interdisciplinarnost razpoznati in vrniti več ustreznih vrstilcev UDK, ki lahko predstavljajo različne veje znano-sti. Hibridni pristop priporočanja uporablja kaskadno hibridizacijo in je tako razdeljen v dva kaskadna koraka. Vhod v hibridni pristop je dokument, za katerega želimo priporo-čati vrstilce UDK. V prvem koraku smo osnovni seznam priporočenih vrstilcev zagotovili z uporabo rangirne funkcije BM25 [34–39]. Tega smo preoblikovali z novim seznamom priporočenih vrstilcev, ki je bil rezultat večznačnega klasifikatorja (ang. multi-label classifier). Le-ta temelji na globoki nevronski mreži BERT, ki smo jo prilagodili na hierarhično topologijo UDK. Glede na organizacijo, s katere izvira vhodni dokument, smo v drugem kaskadnem koraku uporabili seznam iz prvega kaskadnega koraka in ga preoblikovali s seznamom najpogosteje uporabljenih vrstilcev UDK v organizaciji. Za kaskadno hibridizacijo smo uporabili postopke naknadne obdelave. Med te spadajo preurejanje na podlagi vrhnjega področja, preurejanje na podlagi specifičnosti in rezanje seznama. Kvaliteto priporočil smo preverili z eksperimentom na množici zaključnih del v slovenskem jeziku, ki so del repozitorijev slovenskih univerz in že imajo ročno določene vrstilce UDK s strani knjižničarjev. Pri tem smo uporabili tradicionalne metrike za vrednotenje priporočilnih sistemov [40–43], tj. razmerje zadetkov pri K (HR@K, ang. hit-ratio at K), normaliziran diskontiran kumulativni dobiček pri K (NDCG@K, ang. normalized discounted cumulative gain), srednja povprečna natančnost (MAP, ang. mean average precision) in srednji recipročni rang (MRR, ang. mean reciprocal rank).

1.3 Predpostavke in omejitve

V okviru doktorske disertacije nismo postavili nobene predpostavke. Pri izdelavi disertacije smo upoštevali naslednje omejitve (O):

- **O1**: Priporočanje vrstilcev univerzalne decimalne klasifikacije smo omejili na dokumente v slovenskem jeziku.
- **O2**: V eksperimentalnem delu (učenje, testiranje, vrednotenje in analiza) smo se omejili na podatkovno množico OpenScience metadata dataset [44] in brezplačen omejen obseg kataloga univerzalne decimalne klasifikacije [45].
- **O3**: Pri izvedbi hibridnih priporočilnih sistemov smo se omejili na kaskadno hibridizacijo.
- **O4**: Pri izvedbi vsebinskega filtriranja znotraj hibridnih priporočilnih sistemov smo se omejili na rangirno funkcijo BM25 in arhitekturo globokih nevronskih mrež transformer.

- **O5**: Za ocenjevanje kvalitete priporočil smo uporabili standardne metrike HR@K, NDCG@K, MAP in MRR, pri čemer smo se omejili na vrednosti K = [1; 3; 5].

1.4 Teza in hipoteze

Glavni cilj doktorske disertacije strnemo v naslednjo tezo:

S hibridnim pristopom priporočanja, ki kombinira metodologije vsebinskega filtriranja in večznačne klasifikacije, izboljšamo določanje vrstilcev univerzalne decimalne klasifikacije v primerjavi z obstoječimi pristopi.

Tezo doktorskega dela smo razčlenili na naslednje hipoteze (H):

- **H1**: *Predlagani hibridni pristop priporočanja vrstilcev univerzalne decimalne klasifikacije vrne statistično značilno boljše rezultate od pristopov priporočanja, ki temeljijo izključno na vsebinskem filtriranju, glede na metriki HR@K in NDCG@K za standardne vrednosti parametra K ter metriki MAP in MRR.*
- **H2**: *Predlagani hibridni pristop priporočanja vrstilcev univerzalne decimalne klasifikacije vrne statistično značilno boljše rezultate od pristopov priporočanja, ki temeljijo izključno na večrazredni klasifikaciji, glede na metriki HR@K in NDCG@K za standardne vrednosti parametra K ter metriki MAP in MRR.*
- **H3**: *Upoštevanje metapodatkov o izvoru elektronskega dokumenta statistično značilno izboljša delovanje hibridnega pristopa priporočanja vrstilcev univerzalne decimalne klasifikacije glede na metriki HR@K in NDCG@K za standardne vrednosti parametra K ter metriki MAP in MRR.*

1.5 Izvirni znanstveni prispevki

Pri izdelavi doktorske disertacije se pričakujejo naslednji izvirni znanstveni prispevki (IZP):

- **IZP1**: Nov pristop za reševanje problema polavtomatskega določanja vrstilcev univerzalne decimalne klasifikacije elektronskim dokumentom.
- **IZP2**: Razvoj postopkov rerangiranja znotraj kaskadnega tipa hibridnega priporočilnega sistema z namenom izboljšave postopka priporočanja.
- **IZP3**: Razvoj metode za dinamično določanje števila osnovnih relevantnih dokumentov v prvem kaskadnem koraku priporočilnega sistema.

- **IZP4**: Nova metoda glajenja oznak pri učenju klasifikacijskega modela, ki upošteva hierarhično topologijo univerzalne decimalne klasifikacije.
- **IZP5**: Podrobna analiza parametrov kaskadnega tipa hibridnega priporočilnega sistema.

1.6 Struktura doktorske disertacije

Doktorska disertacija je sestavljena iz šestih poglavij. Po uvodu v prvem poglavju sledi drugo poglavje, ki opisuje univerzalno decimalno klasifikacijo, področje obdelave naravnega jezika in področje priporočilnih sistemov. Zatem sledi pregled sorodnih del. V tretjem poglavju predstavimo predlagani hibridni pristop za priporočanje vrstilcev univerzalne decimalne klasifikacije. Na tem mestu podrobneje opišemo delovanje rangirne funkcije BM25 in večznačnega klasifikatorja, ki temelji na globoki nevronski mreži BERT v sklopu predlaganega hibridnega pristopa. Nato opišemo kaskadno hibridizacijo in postopke naknadne obdelave seznamov priporočil. V četrtem poglavju opišemo postavitev eksperimentalnega okolja in predstavimo rezultate eksperimenta glede na izbrane metrike za vrednotenje priporočilnih sistemov. V petem poglavju izvedemo analizo in interpretacijo rezultatov, na podlagi katere potrdimo zastavljene hipoteze. Disertacijo zaključimo s šestim poglavjem, v katerem povzamemo ključne ugotovitve in izpostavimo glavne znanstvene prispevke doktorske disertacije.

2 Metodologije in sorodna dela

Razvoj digitalnih knjižnic je posledica interdisciplinarnega sodelovanja med raziskovalci s področij računalništva in knjižničarstva. Raziskovalci s področja knjižničarstva so zaslužni za vzpostavitev knjižničarskih procesov znotraj digitalnih knjižnic, raziskovalci s področja računalništva pa za tehnično zasnovo, izvedbo in vzdrževanje digitalnih knjižnic. Aktivno sodelovanje predstavnikov obeh področij vodi v nove ideje in tehnično realizacijo teh idej, ki bodisi olajšajo delo knjižničarjem, bodisi izboljšajo uporabniško izkušnjo. S področja knjižničarstva v tem poglavju predstavljamo ozadje univerzalne decimalne klasifikacije, ki je primarni knjižnični klasifikacijski sistem v Sloveniji. V sklopu orodij in funkcionalnosti sodobnih digitalnih knjižnic so nekatere tesno povezane z umetno inteligenco, ki je veja računalništva. Priporočilne sisteme kot eno izmed tehnologij umetne inteligence danes srečujemo praktično vsepovsod na spletu, njihova uporaba pa je našla mesto tudi v digitalnih knjižnicah. Ker digitalne knjižnice vsebujejo predvsem velike količine besedil, pa veliko orodij temelji na razvoju še enega zelo aktivnega področja umetne inteligence – obdelave naravnega jezika.

2.1 Univerzalna decimalna klasifikacija

Leta 1895 sta belgijska bibliografa Paul Otlet in Henri La Fontaine ustvarila katalog v obliki šifranta, katerega namen je bil klasificirati objavljene dokumente v področja znanosti. Prva različica, v francoščini poimenovana *Répertoire Bibliographique Universel*, je bila kmalu zatem spremenjena in dopolnjena s pomočjo francoskega prevoda Deweyjeve decimalne klasifikacije, ki ga je z dovoljenjem Melvila Deweyja ustvaril Otlet [46]. Glavna novost glede na takrat obstoječe knjižnične klasifikacijske sisteme je bila zmožnost označevanja dokumentov z odnosi med različnimi področji znanosti [47]. V prvem priročniku iz leta 1905 s francoskim naslovom *Manuel du Répertoire bibliographique universel* se tako že pojavijo dodatne tabele s splošnimi privesnimi vrstilci, kot tudi notacijski sistem s povezovalnimi simboli in sintaktičnimi pravili [48]. Skozi čas je prišlo do preimenovanja v univerzalno decimalno klasfikacijo (ang. Universal Decimal Classification), standardizacije in prevodov v druge jezike. Univerzalno decimalno klasifikacijo (UDK) danes vzdržuje in dopolnjuje konzorcij Universal Decimal Classification Consortium (UDCC) [49]. Trenutno se uporablja v 130 državah na svetu in služi kot primarni knjižnični klasifikacijski sistem v približno 30 državah. Med te države spada tudi Slovenija, ki je s 3,5 milijona klasificiranimi dokumenti trenutno tretja na svetovni lestvici po množičnosti uporabe UDK [50].

Od leta 1993 naprej se vzdržuje standardna različica UDK, ki se hrani v podatkovni bazi (UDC Master Reference File - UDC MRF) in se redno posodablja. Trenutno najsodobnejša je različica UDC MRF 2011, ki je bila izdana leta 2012 in vsebuje preko 70,000 vrstilcev. Ta različica je dostopna s plačljivo licenco in je s tem omejena na uporabo v narodnih knjižničnih sistemih. Od leta 2011 je na spletu dostopna tudi prosto dostopna različica UDK [45], ki je omejena na okoli 2600 vrstilcev in je na voljo v 57 jezikih, med drugim tudi v slovenščini.

2.1.1 Struktura UDK

V univerzalni decimalni klasifikaciji se dokumenti označujejo z izrazi, ki so sestavljeni iz vrstilcev in simbolov za povezovanje. Vrstilec je sestavljen iz arabskih števil, ki jih loči decimalna pika na mestu, kjer pride do delitve področja na več podpodročij. Vsako podpodročje se lahko na enak način deli še dlje, kar vodi v hierarhično strukturo. UDK je torej hierarhično izrazna. To pomeni, da z daljšim vrstilcem bolj specifično opredelimo področje. V tabeli 2.1 je prikazana hierarhična razčlenitev vrstilca 004.738.52 - "Orodja za iskanje v internetu.".

Tabela 2.1: Hierarhična razčlenitev vrstilca 004.738.52 - "Orodja za iskanje v internetu.".

Vrstilec	**Opis področja**
0	Znanost in znanje. Organizacije. Informacije. Dokumentacija. Bibliotekarstvo. Institucije. Publikacije.
00	Prolegomena. Splošne osnove znanosti in kulture.
004	Računalniška znanost in tehnologija. Računalništvo. Obdelava podatkov.
004.7	Računalniške komunikacije. Računalniška omrežja.
004.73	Omrežja glede na prostranost.
004.738	Medsebojno povezovanje omrežij. Medomrežanje.
004.738.5	Internet.
004.738.52	Orodja za iskanje v internetu.

Vrstilci UDK so urejeni v glavne tabele, ki se hierarhično razvijajo v podpodročja. Vrhnja področja (tabela 2.2) so definirana z devetimi arabskimi števili in predstavljajo vsa področja človeškega znanja. Število 4 je prosto zaradi načrtovane razširitve. Predhodno je bilo namenjeno področju jezikoslovja, ki pa se je v 1960-ih premaknilo pod število 8 zaradi razmaha na področjih naravoslovja in tehnologije.

Tabela 2.2: Vrstilci vrhnjih področij univerzalne decimalne klasifikacije.

Vrstilec	**Opis področja**
0	Znanost in znanje. Organizacije. Informacije. Dokumentacija. Bibliotekarstvo. Institucije. Publikacije.
1	Filozofija. Psihologija.
2	Teologija. Verstva.
3	Družbene vede. Politika. Ekonomija. Pravo. Izobraževanje.
4	*prosto*
5	Matematika. Naravoslovje.
6	Uporabne znanosti. Medicina. Tehnika.
7	Umetnost. Arhitektura. Fotografija. Glasba. Šport.
8	Jezik. Književnost.
9	Geografija. Biografija. Zgodovina.

Tabela 2.3: Vrstilci v področju 6 - “Uporabne znanosti. Medicina. Tehnika.”.

Vrstilec	**Opis področja**
60	Biotehnologija.
61	Medicina.
62	Inženirstvo. Tehnologija na splošno.
63	Kmetijstvo ter sorodne vede in tehnologije.
64	Gospodinjstvo. Stanovanje.
65	Komunikacije in transport. Knjigovodstvo. Poslovni menedžment. Stiki z javnostjo.
66	Kemijska tehnologija. Kemijske in sorodne industrije.
67	Razne industrije, obrti in rokodelstva.
68	Industrije, obrti in rokodelstva za sestavljanje in dodelavo izdelkov.
69	Gradbeništvo. Gradbeni materiali. Gradbene obrti in dela.

V tabelah 2.3 in 2.4 je prikazana hierarhična struktura UDK za področje 6 - “Uporabne znanosti. Medicina. Tehnika.”. V tabeli 2.4 se vidi podaljšanje dolžin vrstilcev in bolj specifični opisi področij.

Tabela 2.4: Vrstilci za področje 68 - “Industrije, obrti in rokodelstva za sestavljanje in dodelavo izdelkov.”.

Vrstilec	**Opis področja**
681	Fina mehanika in instrumenti.
682	Kovaštvo. Podkovstvo. Ročno kovano železje.
683	Železnina. Ključavničarstvo. Polnjenje steklenic. Svetila. Grelni aparati.
684	Pohištvene in podobne industrije. Proizvodnja pohištva. Tapetništvo.
685	Sedlarstvo. Čevljarstvo. Rokovičarstvo. Potovalna in športna oprema. Predmeti za igre.
686	Knjigoveštvo. Metaliziranje. Izdelovanje ogledal. Pisalne potrebščine.
687	Oblačilna industrija. Kozmetična industrija in podobne obrti.
688	Galanterija. Igrače. Okrasni predmeti.
689	Amaterske spretnosti. Tehnični hobiji.

Ob vrstilcih za področja se uporabljajo tudi splošni privesni vrstilci, ki so označeni na poseben način in tako v kompleksnem izrazu UDK izstopajo od ostalih vrstilcev. Splošni prives-ni vrstilci se vedno začnejo s posebnimi znaki, ki definirajo pomen splošnega privesnega vrstilca. Različni tipi splošnih privesnih vrstilcev so navedeni v tabeli 2.5.

Tabela 2.5: Splošni privesni vrstilci v univerzalni decimalni klasifikaciji s primeri.

Znak	**Pomen splošnega privesnega vrstilca**	**Primer**
=...	Jezik	=163.6 *za slovenski jezik*
(0...)	Oblika	(043) *za zaključno delo*
(1/9)	Kraj	(497.4) *za Slovenijo*
(=...)	Rase, narodi in etnične skupine	(=011) *za evropejce*
”...”	Čas	”198” *za leta med 1980 in 1989*
-0...	Lastnosti, materiali, osebe	-035 *za kovine*

UDK je tudi sintaktično izrazna, saj omogoča združevanje več vrstilcev UDK z znaki za povezovanje. Slednji definirajo odnose med različnimi vrstilci, kar je uporabno pri zelo specifičnem določanju tematike dokumenta. Različne podprte znake za povezovanje z njihovimi primeri prikazuje tabela 2.6. Znak za priredno razširitev (+) povezuje definirana področja v UDK tako, da ponazarja enakovredno združevanje med njimi. Znak za zaporedno razširitev (/) povezuje definirana področja v UDK tako, da ponazarja zaporedno združevanje med njimi. Znak za enostaven odnos (:) povezuje definirana področja v UDK tako, da definira rabo enega področja znotraj drugega področja oziroma v povezavi z drugim področjem. Znak za stalno zaporedje (::) povezuje definirana področja v UDK tako, da trdno

določi zaporedje vrstilcev UDK. Z oglatimi oklepaji ([ ]) lahko v izrazih UDK določamo podskupine več vrstilcev UDK. Z zvezdico (*) lahko v izrazih UDK določamo oznako, ki ni del UDK in iz nje ne izvira. UDK podpira tudi neposredno abecedno določitev, ki omogoča določitev lastnih imen in kratic.

Tabela 2.6: Znaki za povezovanje v univerzalni decimalni klasifikaciji s primeri.

Znak	Pomen znaka	Izraz UDK	Pomen izraza UDK
+	Priredna razširitev	004+02	računalništvo in knjižničarstvo
/	Zaporedna razširitev	548/549	kristalografija in mineralogija
:	Enostaven odnos	004:330	računalništvo v ekonomiji
::	Stalno zaporedje	77.04::355.4	vojna fotografija
[ ]	Podrobna delitev	[378+53](497.4)	visoko šolstvo in fizika v Sloveniji
*	Oznaka, ki ne izvira iz UDK	796.8*kg85	borilne veščine, kategorija 85 kg
A-Z	Neposredna abecedna določitev	(497.4Maribor)	Slovenija, mesto Maribor

2.1.2 Primeri izrazov UDK

Če upoštevamo strukturo UDK in njene značilnosti, vidimo, da je s kombinacijo vrstilcev za različna področja, splošnih privesnih vrstilcev in znakov za povezovanje možno sesta-viti tako enostavne, kot tudi zelo kompleksne izraze UDK. V nadaljevanju je podanih ne-kaj enostavnih in kompleksnih primerov izrazov UDK z utemeljitvami, kako smo sestavili končne izraze UDK (slike 2.1-2.4).

Primer 1 - enostaven izraz UDK

Cilj: sestaviti želimo izraz UDK za matematični učbenik.

Končni izraz UDK: 51(075)

Uporabljeni vrstilci:

51 (področje: matematika)

(075) (splošni privesni vrstilec za obliko: učbenik)

Slika 2.1: Primer enostavnega izraza UDK za matematični učbenik.

Primer 2 - enostaven izraz UDK

Cilj: sestaviti želimo izraz UDK za diplomsko delo, ki govori o sistemu umetne inteligence za avtomatsko razpoznavanje govora.

Končni izraz UDK: 004.8:004.934(043.2)

Uporabljeni vrstilci:

004.8 (področje: umetna inteligenca)

004.934 (področje: avtomatsko razpoznavanje govora)

(043.2) (splošni privesni vrstilec za obliko: diplomsko delo)

Slika 2.2: Primer enostavnega izraza UDK za diplomsko delo.

Primer 3 - kompleksen izraz UDK

Cilj: sestaviti želimo izraz UDK za doktorsko disertacijo, ki govori o meritvah mišičnih aktivnosti in analizi površinskih elektromiogramov.

Končni izraz UDK: [621.37+681.5.015]:612.7-073.7(043.3)

Uporabljeni vrstilci:

[621.37+681.5.015] (podrobna delitev in združevanje dveh področij)

621.37 (področje: elektronska vezja)

681.5.015 (področje: mehke regulacije)

612.7 (področje: motorične funkcije, organi za premikanje, glas, koža)

-073.7 (splošni privesni vrstilec za lastnosti: elektromiografija)

(043.3) (splošni privesni vrstilec za obliko: doktorska disertacija)

Slika 2.3: Primer kompleksnega izraza UDK za doktorsko disertacijo.

Primer 4 - kompleksen izraz UDK

Cilj: sestaviti želimo izraz UDK za turistični zemljevid Ljubljane iz leta 2021 v digitalni obliki datoteke PDF.

Končni izraz UDK: 338.48(497.4Ljubljana)”2021”(084.3)(0.034.2PDF)

Uporabljeni vrstilci:

338.48 (področje: turizem)

(497.4Ljubljana) (splošni privesni vrstilec za kraj: Slovenija, mesto Ljubljana)

”2021” (splošni privesni vrstilec za čas: leto 2021)

(084.3) (splošni privesni vrstilec za obliko: zemljevid)

(0.034.2PDF) (splošni privesni vrstilec za obliko: digitalna datoteka, PDF)

Slika 2.4: Primer kompleksnega izraza UDK za turistični zemljevid.

2.2 Obdelava naravnega jezika

Znanstveno področje, ki združuje računalništvo, umetno inteligenco in jezikoslovje, ter se ukvarja z razumevanjem človeškega jezika v obliki besedila, zvoka in slike (pisave), je splošno znano kot obdelava naravnega jezika (ang. natural language processing). Ker se v tej doktorski disertaciji osredotočamo na dokumente, kjer je človeški jezik v obliki besedila, se v tem poglavju ne bomo poglabljali v metode na področju obdelave naravnega jezika, ki so vezane na zvok in slike. S tega stališča lahko to področje definiramo s petimi osrednjimi nalogami, ki rešujejo širok nabor problemov:

- klasifikacija besedil (ang. text classification),
- luščenje informacij (ang. information extraction),
- pogovorni agenti (ang. conversational agents),
- informacijsko poizvedovanje in iskalniki (ang. information retrieval and search engines),
- sistemi za odgovarjanje na vprašanja (ang. question answering systems).

Klasifikacija besedil vključuje ustrezno razvrstitev besedil glede na njihovo tematiko. Na ta način lahko gručimo gradiva v sorodne gruče in s tem tvorimo skupine podobnih bese-dil [51]. To je uporabno pri iskalnikih, detekciji podobnih vsebin in priporočilnih sistemih.

Zelo razširjena oblika klasifikacije je analiza sentimenta (ang. sentiment analysis), ki ugotavlja ali ima besedilo pozitiven ali negativen kontekst [52–54]. To je še posebej uporabno pri ugotavljanju sovražnega govora [55].

Naloga luščenja informacij se ukvarja s pretvorbo besedila iz nestrukturirane oblike, v strukturirano obliko, kjer so zaznani zanimivi in koristni deli besedila. Nestrukturirano besedilo je neorganizirano surovo besedilo, brez posebnih označb, ki bi omogočale enostavno strojno branje besedila. Na drugi strani je strukturirano besedilo organizirano in opremljeno z dodatnimi označbami, vse skupaj pa je ponavadi shranjeno v strojno berljivi obliki različnih formatov kot so CSV (ang. comma-separated values), JSON (ang. JavaScript Object Notation) ali XML (ang. Extensible Markup Language). Med oblike luščenja informacij spadajo razpoznavanje imenskih entitet (ang. named entity recognition) [56], luščenje povezav (ang. relationship extraction) [57] in odkrivanje koreferenčnosti (ang. coreference resolution) [58].

Pogovorni agenti [59] omogočajo uporabnikom, da s pogovorom upravljajo z računalniškim sistemom ali pa za izvajanje ukazov uporabljajo naravni jezik (npr. OpenAI ChatGPT, Google Bard, You.com in OpenAssistant.io). Sodobni pogovorni agenti so na voljo na pametnih telefonih v obliki pametnih asistentov (npr. Apple Siri, Google Assistant in Microsoft Cortana), kot tudi v obliki pametnih hišnih asistentov (npr. Amazon Alexa).

Na področju pridobivanja informacij kljub vsem napredkom na področju obdelave naravnega jezika še vedno govorimo o iskalnikih in priporočilnih sistemih. Bistvo te naloge je, da se iz ogromne množice informacij filtrirajo le tiste, ki so relevantne in uporabne za končnega uporabnika [60].

Sistemi za odgovarjanje na vprašanja [61] so tesno povezani z vsemi prej naštetimi osrednjimi nalogami, saj omogočajo uporabniku, da postavi vprašanje, nato pa z uporabo različnih podsistemov pridobijo ustrezne informacije, ki jih preoblikujejo v odgovor.

Področje obdelave naravnega jezika je v preteklosti prehajalo skozi različna obdobja, ki so močno zaznamovala razvoj metod. V prvem obdobju (med 1950 in 1980) so bile zgodnje metode za obdelavo naravnega jezika v obliki besedila zasnovane na podlagi kompleksnih ročno določenih pravil [62]. V tem obdobju so nastale najbolj enostavne metode, ki pa niso bile razširljive in so bile posledično uporabne samo za specifične probleme. Skozi čas je prišlo tudi do pionirskih implementacij različnih sistemov, najbolj atraktivni za tisti čas pa so bili pogovorni agenti (npr. ELIZA [63]). V drugem obdobju (med 1980 in 2010) se je razširila uporaba metod strojnega učenja, s tem pa na področju povzročila t.i. statistično revolucijo [64, 65]. Veliko metod iz tega obdobja temelji na verjetnostnih modelih [66, 67], seveda pa se je s tem izboljšala uporabnost implementiranih sistemov v praksi.

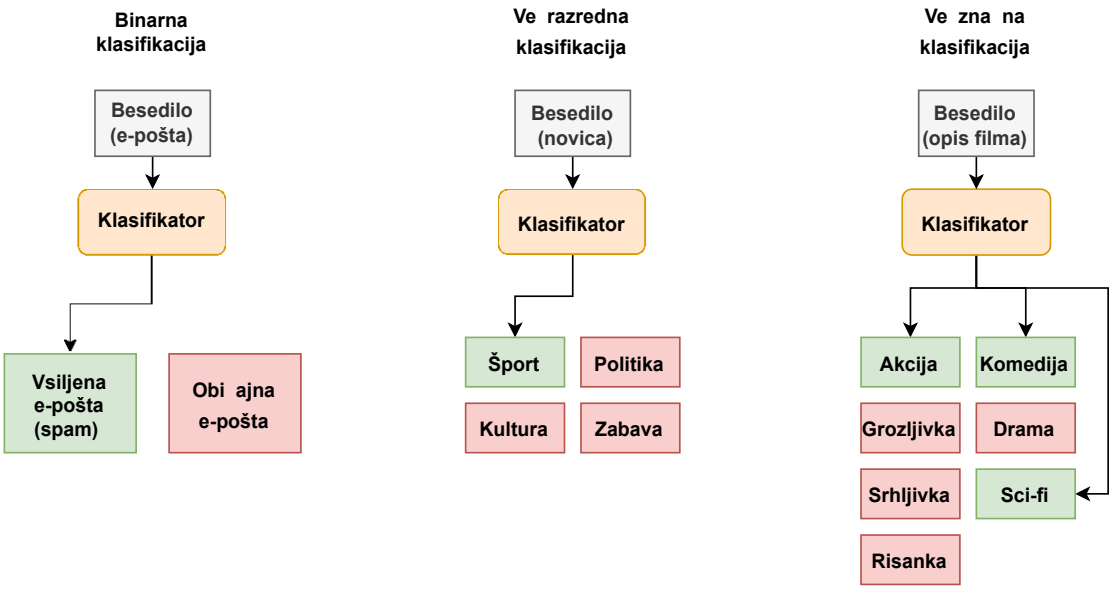
V tem obdobju se srečamo s prvimi zares uporabnimi sistemi, ki jih masovno uporabljajo navadni uporabniki (npr. iskalnik Google). Nekje od leta 2010 naprej govorimo o tretjem obdobju - t.i. nevronski revoluciji - kjer so različne izvedbe nevronskih mrež v veliki meri nadomestile prejšnje metode. Posledica tega je uporaba drugačnih značilk imenovanih kontekstne vložitve, ki so se v večini primerov izkazale za boljše kot prej uporabljene značilke iz vreče besed [68]. Med leti 2015 in 2019 so na področju obdelave naravnega jezika prednjačile arhitekture povratnih globokih nevronskih mrež (ang. recurrent neural networks), predvsem tiste, ki so uporabljale dolgi kratkoročni spomin LSTM (ang. long short-term memory) ali celice GRU (ang. gated recurrent unit) [69]. Leta 2017 se je pojavila arhitektura globokih nevronskih mrež transformer [13], ki je omogočila ogromen napredek na področju obdelave naravnega jezika. Trenutno najbolj uveljavljeni izvedbi globokih nevronskih mrež transformer sta BERT [14] (ang. Bidirectional Encoder Representations from Transformers) in GPT [70] (ang. Generative Pre-trained Transformer). Značilnost globokih nevronskih mrež BERT je uporaba mehanizma pozornosti (ang. attention mechanism) za ohranjanje konteksta v daljšem odseku besedila, pri čemer se za določeno besedo upošteva kontekst pred njo in za njo. Značilnost globokih nevronskih mrež GPT je zmožnost napovedovanja naslednje najbolj verjetne besede, kar omogoča generiranje novega besedila z upoštevanjem konteksta predhodnih besed. Gre za kompleksne globoke nevronske mreže, ki so vnaprej naučene predstavitev naravnih jezikov, v praksi pa jih lahko prilagodimo (ang. fine-tuning) na druge tipe nalog s prenosnim načinom učenja (ang. transfer learning). Te metode so trenutno najuspešnejše pri reševanju različnih nalog na področju obdelave naravnega jezika.

2.2.1 Klasifikacija besedil

Kot ena izmed osrednjih nalog področja obdelave naravnega jezika, se klasifikacija besedil ukvarja z razporejanjem besedil v specifične razrede ali kategorije. Klasifikacijo besedil lahko delimo na tri različne tipe: binarno, večrazredno in večznačno. Pri binarni klasifikaciji se besedilo uvrsti v enega izmed dveh možnih razredov. Primer binarne klasifikacije besedil je filtriranje vsiljene elektronske pošte (ang. spam e-mail), kjer z uporabo algoritmov strojnega učenja besedilo klasificiramo v razred vsiljene elektronske pošte, ali pa v razred običajne elektronske pošte [71–73].

Pri večrazredni klasifikaciji se besedilo uvrsti v enega izmed več možnih razredov. Takšen tip klasifikacije se pogosto uporablja za uvrščanje posamezne novice v kategorijo (npr. šport, kultura, zabava, politika) [74–76]. Tudi pri večznačni klasifikaciji se besedilo uvrsti v več možnih razredov, vendar lahko pri tem tipu klasifikacije besedil besedilo uvrstimo v več razredov hkrati. Primer uporabe večznačne klasifikacije besedil je označevanje (ang. tagging) besedil z različnimi smiselnimi oznakami, kar se pogosto uporablja v objavah na

družbenih omrežjih, spletnih trgovinah in priporočilnih sistemih [77–80]. Slika 2.5 prikazuje grafično ponazoritev razlik med tipi klasifikacije besedil na različnih primerih.



Slika 2.5: Različni tipi klasifikacije besedil s primeri.

Za binarno klasifikacijo se v primeru na sliki 2.5 izvaja detekcija vsiljene e-pošte. Klasifikator se odloča med dvema možnima razredoma: vsiljena e-pošta in običajna e-pošta. Rezultat klasifikatorja je le en izmed teh dveh razredov, tj. razred vsiljena e-pošta. Za večrazredno klasifikacijo se izvaja kategorizacija novic. Klasifikator se odloča med štirimi možnimi razredi: šport, politika, kultura in zabava. Rezultat klasifikatorja je le en izmed teh štirih razredov, tj. razred šport. Za večznačno klasifikacijo se izvaja določanje filmskega žanra. Klasifikator se odloča med sedmimi možnimi razredi: akcija, komedija, grozljivka, drama, srhljivka, sci-fi in risanka. Rezultat klasifikatorja je več razredov, in sicer akcija, komedija in sci-fi.

Pri specifičnih problemih večznačne klasifikacije besedil se lahko zgodi, da imamo opravka z velikim številom možnih oznak (ang. extreme multi-label text classification - XMTC). Gre za posebno različico večznačne klasifikacije besedil, kjer so možne oznake lahko med seboj v hierarhičnih odnosih, pri tem pa je število možnih oznak lahko več kot 100.000 [81]. Zaradi ogromne računske zahtevnosti so metode za reševanje takšnih problemov zasnovane tako, da razvrstijo možne oznake v interno drevesno strukturo [82–84]. V zadnjem času so se začele pojavljati tudi metode, ki uporabljajo različne tipe globokih nevronskih mrež [85–87].

2.2.2 Predstavitev besedila v statističnih metodah

Priljubljena statistična metoda za obdelavo naravnega jezika v obliki besedil je utežna shema tf-idf (ang. term frequency, inverse document frequency). Pri tem je besedilo predstavljeno kot vreča besed, ki jo sestavlja množica besednih n-gramov. To so deli besedila, razdeljeni na n besed. Slika 2.6 prikazuje primer različnih delitev na n-grame za stavek "*Miha je šel v trgovino in kupil kruh.*". V primeru so prikazani unigrami (n = 1), bigrami (n = 2) in trigrami (n = 3).

Miha je šel v trgovino in kupil kruh.	
Tip delitve	**Rezultat delitve**
unigrami (n = 1)	Miha, je, šel, v, trgovino, in, kupil, kruh
bigrami (n = 2)	Miha je, je šel, šel v, v trgovino, trgovino in, in kupil, kupil kruh
trigrami (n = 3)	Miha je šel, je šel v, šel v trgovino, v trgovino in, trgovino in kupil, in kupil kruh

Slika 2.6: Primeri različnih delitev besedila na besedne n-grame.

Na podlagi pojavitev n-gramov znotraj dokumenta se izračunata uteži tf in idf za vsak n-gram in dokument. Utež tf predstavlja število pojavitev (frekvenco) $f_{t;d}$ n-grama t v dokumentu d. Obstaja več načinov kako izračunati utež tf, najpogosteje uporabljena načina pa upoštevata zgolj frekvenco n-gramov (enačba 2.1) oziroma njen logaritem (enačba 2.2).

$$tf(t; d) = f_{t;d} \tag{2.1}$$

$$tf(t; d) = \log(1 + f_{t;d}) \tag{2.2}$$

Utež idf predstavlja pojavitev n-grama v posameznem dokumentu glede na njegovo pogostost v zbirki dokumentov D. Tudi utež idf lahko izračunamo na več načinov, najpogosteje uporabljen način (enačba 2.3) pa v izračunu uporablja velikost zbirke vseh dokumentov $|D|$ in število dokumentov $n(t)$, ki vsebujejo n-gram t.

$$idf(t) = \log \frac{|D| \quad n(t) + 0;5}{n(t) + 0;5} \tag{2.3}$$

Z množenjem uteži tf in idf dobimo utež tf-idf (enačba 2.4). Višja vrednost uteži tf-idf pomeni večjo pomembnost n-grama, z nižjo pomembnostjo n-grama pa se vrednosti približujejo 0. Zaradi te lastnosti lahko za vsak dokument filtriramo pomembne n-grame od

nepomembnih. S tem dobimo nabor tistih, ki dobro kontekstno opisujejo vsebino besedila, odstranimo pa manj pomembne dele besedila, kot so npr. vezniki.

$$tf\text{-}idf(t; d) = tf(t; d)\ idf(t) \tag{2.4}$$

Z izračunom uteži tf-idf za besedilo v dokumentu se tvorijo značilke dokumenta, ki se uporabljajo v algoritmih strojnega učenja in iskalnih algoritmih. Najpogosteje uporabljeni algoritmi strojnega učenja so naivni Bayesov klasifikator, linearna regresija, večplastna nevronska mreža in metoda podpornih vektorjev SVM (ang. support vector machine). Iskalni algoritmi zajemajo predvsem rangirno funkcijo BM25 (ang. Best Match 25), ki je podrobneje opisana v nadaljevanju.

2.2.3 Rangirna funkcija BM25

Rangirna funkcija BM25 je statistična metoda za ugotavljanje podobnosti med dokumenti. Nastala je v 1970-ih kot del ogrodja za informacijsko poizvedovanje [34] in se po tem ustalila kot metoda, ki jo še danes uporabljajo moderni iskalniki. Gre v bistvu za družino metod, ki se razlikujejo po utežnih shemah in vrednostih parametrov. Vse različice te rangirne funkcije uporabljajo uteži tf in idf. Skozi čas se je izkazalo, da se nekatere različice obnesejo bolje v specifičnih primerih [37–39], čeprav se v splošnem še vedno uporablja osnovna različica. Z enačbo 2.5 izračunamo vrednost BM25, ki predstavlja podobnost $s(d; Q)$ med dokumentom d in iskalnim nizom Q.

$$s(d; Q) = \sum_{i=1}^{|Q|} idf(q_i)\ \frac{tf(q_i; d)\ (k_1 + 1)}{tf(q_i; d) + k_1\ B} \tag{2.5}$$

Rangirno funkcijo BM25 lahko uporabimo za izračun podobnosti med dvema dokumentoma tako, da namesto iskalnega niza Q podamo besedilo drugega dokumenta. V tem primeru besedilo novega dokumenta dobi vlogo iskalnega niza Q, s katerim se izračuna podobnost z že katalogiziranim dokumentom d (enačba 2.5). Člen q_i označuje posamezno besedo iskalnega niza, člen $tf(q_i; d)$ pa je vrednost uteži tf, ki je enaka številu pojavitev besede q_i v dokumentu d. Parameter k_1 je parameter, ki ima v literaturi običajno vredno-sti med 1 in 3. Na primer, v [5] avtorji priporočajo vrednost $k_1 = 1;2$, v [34] predlagajo vrednost $k_1 = 2;0$, v [88] pa vrednost $k_1 = 2;5$. Nadalje je člen $idf(q_i)$ vrednost uteži idf, ki se izračuna z enačbo 2.3. Nazadnje je tukaj še člen B, ki je normalizacijski faktor in se izračuna z enačbo 2.6:

$$B = 1 \quad b + b \ \frac{l_d}{l_{avg}} \tag{2.6}$$

kjer člen l_d predstavlja dolžino dokumenta d, člen l_{avg} pa povprečno dolžino dokumentov v korpusu D. Za izračun normalizacijskega faktorja B uporabimo parameter b, ki ima vrednost med 0 in 1. Vrednosti tega parametra se v literaturi razlikujejo. V [5] in [34] priporočajo vrednost $b = 0;75$, v [88] pa vrednost $b = 0;8$. Tudi nekatere prve različice BM25 so nastale zgolj s spremembo tega parametra. To sta na primer BM11, ki uporablja vrednost $b = 0$ in BM15, ki uporablja vrednost $b = 1$.

Obstaja še ena različica rangirne funkcije BM25, ki uporablja dodatni parameter k_3 in dodatno utež qtf. Ta različica rangirne funkcije se izračuna z enačbo 2.7. Dodatna utež $qtf(q_i)$ predstavlja število pojavitev besede q_i v vhodnem iskalnem nizu Q. Dodatni parameter k_3 je parameter, za katerega najdemo v literaturi zelo različne vrednosti. Največkrat se uporablja vrednost $k_3 = 0$, v [89] uporabijo vrednost $k_3 = 1000$, v [88] pa vrednost $k_3 = 7$. Vredno je omeniti tudi, da je enačba 2.5 poseben primer enačbe 2.7, če je $k_3 = 0$, tj. ko se utež qtf ignorira.

$$s^0(d;Q) = s(d;Q) \ \frac{(k_3 + 1) \ qtf(q_i)}{k_3 + qtf(q_i)} \tag{2.7}$$

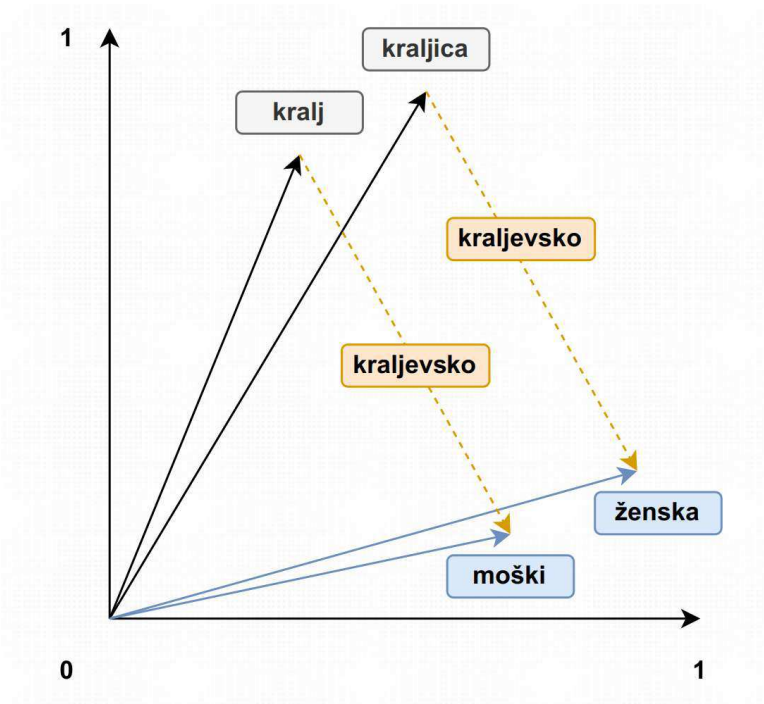
Praktična uporaba rangirne funkcije BM25 je največkrat realizirana tako, da se namesto besed upoštevajo n-grami. Na podlagi pojavitev n-gramov znotraj dokumenta se izračunata uteži tf in idf za vsak n-gram in dokument. Rezultat je iskalni indeks, ki se uporabi za iskanje podobnih dokumentov na podlagi pojavitev n-gramov znotraj besedila. Rangirna funkcija BM25 to delovanje razširi z upoštevanjem porazdelitve n-gramov, predpostavko o gostobesednosti dokumenta in predpostavko o obsegu dokumenta. Predpostavka o gostobesednosti dokumenta govori o tem, da se v besedilu dokumenta lahko pojavijo določeni različni pojmi, ki na široko določajo kontekst dokumenta. Rangirna funkcija BM25 poskuša takšne pojme ustrezno obtežiti na način, ki določa najbolj verjeten kontekst dokumenta. Predpostavka o obsegu dokumenta govori o tem, da je lahko besedilo dokumenta zelo dolgo, najbolj kontekstno relevantni pojmi znotraj besedila pa so tako slabo zaznavni. Rangirna funkcija BM25 poskuša takšne dokumente ustrezno obdelati na način, ki kontekstno nepomembnim pojmom daje zelo nizko utež. Na področju razvoja iskalnikov se tradicionalno uporabljajo statistične metode kot sta tf-idf in BM25, ki delujejo nad redkimi vektorji, pri katerih ima veliko komponent vektorja vrednost 0. Z razvojem globokih nevronskih mrež so se v zadnjem času začele pojavljati metode, ki uporabljajo vložitve. To so vektorji, ki vsebujejo pomenske informacije in kontekst. Trenutno najbolj aktualni takšni metodi sta SBERT [90] in SPLADE [91, 92].

2.2.4 Predstavitev besedila v globokih nevronskih mrežah

Moderne metode za reševanje nalog obdelave naravnega jezika, ki temeljijo na arhitekturi globokih nevronskih mrež transformer, besedilo pretvorijo v vektorje v kontekstnem vektorskem prostoru. Te vektorje imenujemo vložitve (ang. embeddings) in se uporabljajo za iskanje podobnosti med konteksti v besedilih. Vložitve so smiselne, saj v vektorskem prostoru omogočajo gručenje podobnih kontekstov, prav tako pa so bistveno manjših dimenzij kot predstavitve, ki jih uporabljajo statistične metode.

Vložitve se ustvarijo s pomočjo nevronskih mrež, ki poskušajo napovedati manjkajoče besede iz množice stavkov. Dva najbolj razširjena načina učenja vložitev sta Word2Vec [93, 94] in GloVe [95]. Vložitve Word2Vec uporabljajo model neprekinjene vreče besed CBOW (ang. continuous bag of words) ali model skip-gram, vložitve GloVe pa uporabljajo matriko sopojavnosti besed (ang. co-occurence matrix). Na srečo danes obstajajo že naučene vložitve za skoraj vse jezike in lahko v praksi ta korak preskočimo.

Z vložitvami lahko izvajamo zanimive operacije znotraj kontekstnega vektorskega prostora. Z odštevanjem ali seštevanjem dveh vložitev se lahko pomikamo v bližino drugih vložitev, ki predstavljajo soroden kontekst. Zelo popularen primer za prikaz te lastnosti vložitev je prikazan na sliki 2.7 in je predstavljen z 2D projekcijo vložitev, ki so tipično dimenzije nekaj 1000.



Slika 2.7: Primer pomikanja med različnimi kontekstnimi vložitvami.

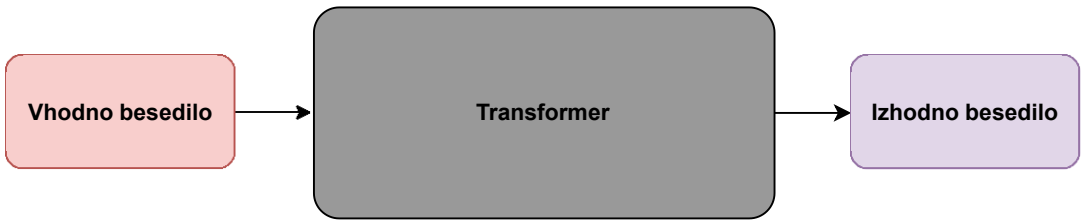
Če od vložitve za besedo “kralj” odštejemo vložitev za besedo “kraljevsko”, potem dobimo vektor, ki je v bližini vložitve za besedo “moški”. Podobno, če od vložitve za besedo “kraljica” odštejemo vložitev za besedo “kraljevsko”, potem dobimo vektor, ki je v bližini vložitve za besedo “ženska”.

Te lastnosti vložitev s pridom uporabljajo različne arhitekture globokih nevronskih mrež. Na področju obdelave naravnega jezika v obliki besedila je danes uporaba globokih nevronskih mrež samoumevna, saj te dosegajo najboljše rezultate za različne probleme, sploh tiste v okviru prej opisanih osrednjih nalog.

2.2.5 Arhitektura globokih nevronskih mrež transformer

V zadnjih letih je na področju razvoja globokih nevronskih mrež nastalo kar nekaj različnih arhitektur. Med najnovejše danes spada arhitektura transformer, ki se tudi v praktičnih aplikacijah obnese zelo dobro. Gre za kompleksno strukturo globoke nevronske mreže, saj je sestavljena iz številnih različnih komponent. Transformerji so še posebej pomembni za obdelavo naravnega jezika, saj so doprinesli k največjim izboljšavam pri številnih nalogah tega področja.

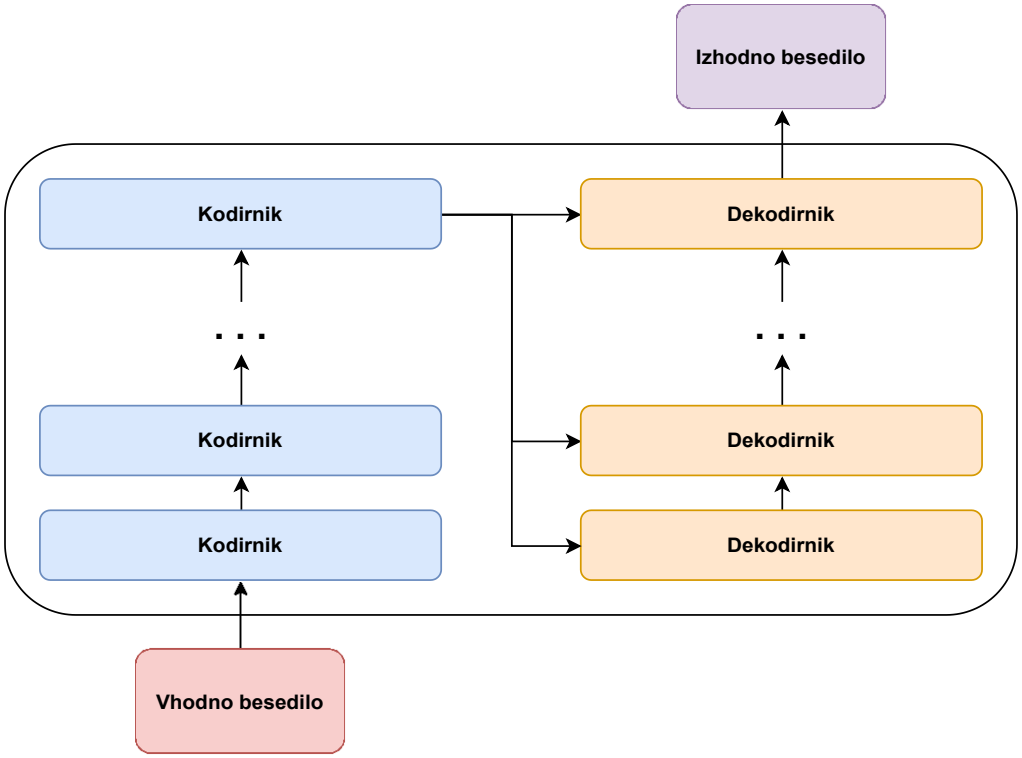
V nadaljevanju bomo najprej podali visokonivojski opis arhitekture transformerjev, nato pa se poglobimo v podrobnosti posameznih komponent. Kot pri tradicionalnih nevronskih mrežah, je tudi pri arhitekturi transformerjev možno model na najvišjem nivoju predstaviti kot črno škatlo, ki prejme vhodno besedilo, vrne pa izhodno besedilo (slika 2.8).



Slika 2.8: Arhitektura transformer na najvišjem nivoju.

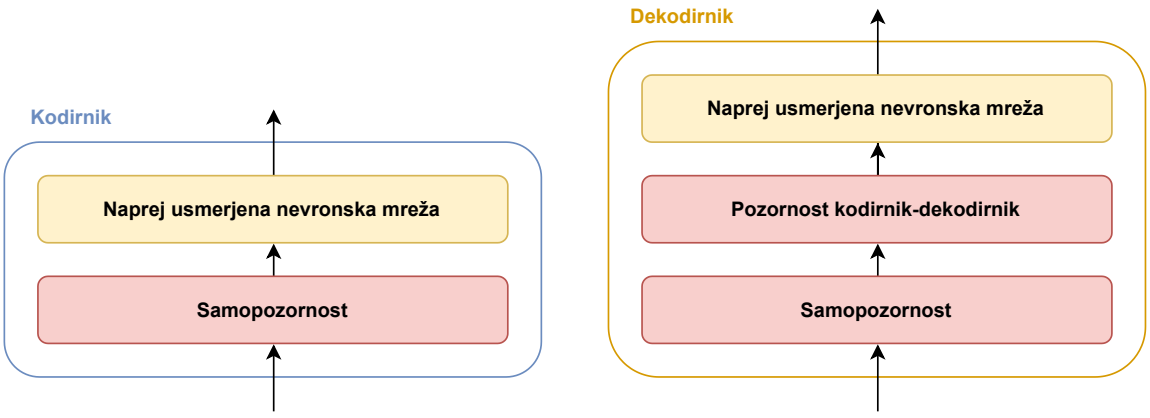
Če podrobneje pogledamo vsebino črne škatle, lahko transformer predstavimo z dvema večjima komponentama. To sta sklada kodirnikov in dekodirnikov. Vhod se obdela tako, da se pošlje skozi sklad kodirnikov, rezultat, ki je vektor konteksta, pa se nato pošlje še skozi sklad dekodirnikov (slika 2.9).

Rezultat sklada dekodirnikov predstavlja izhod nevronske mreže. Število kodirnikov in dekodirnikov je lahko poljubno, v literaturi pa se sicer velikokrat uporablja število 6, saj so to število uporabili tudi avtorji izvornega članka [13].



Slika 2.9: Obdelava vhoda skozi sklada kodirnikov (modra) in dekodirnikov (oranžna).

Vsak kodirnik je nadalje sestavljen iz plasti samopozornosti (ang. self-attention) in naprej usmerjene nevronske mreže (ang. feed-forward neural network). Struktura dekodirnikov je enaka, le da vsebuje še vmesno plast pozornosti med kodirnikom in dekodirnikom. Ta dodatna plast pozornosti v dekodirniku pripomore k boljšem zaznavanju relevantnega dela besedila na vhodu v dekodirnik. Slika 2.10 prikazuje podrobno strukturo kodirnika in dekodirnika.



Slika 2.10: Podrobna struktura kodirnika (levo) in dekodirnika (desno).

Na vhod kodirnika se besedilo ne pošlje v berljivi obliki, temveč v obliki vložitev posameznih členov (ang. token). Členi so običajno manjši deli besed, lahko tudi posamezne črke, ki so določeni z uporabljenim slovarjem. V nadaljevanju bomo zaradi lažje predstave v primerih namesto členov uporabljali besede. Pretvorba v vložitev se tako zgodi pred prehodom skozi prvi kodirnik na skladu. V besedilu je vrstni red besed pomemben. Pri pretvorbi v vložitve se vrstni red besed izgubi, zato je potrebno preslikati tudi vrstni red besed v vložitve. Za to poskrbi položajno kodiranje (ang. positional encoding), s katerim tvorimo položajne vektorje in jih prištejemo vektorjem besednih vložitev. Pri položajnem kodiranju se za vse besede s sodim indeksom položaja pos uporabi trigonometrična funkcija sinus (enačba 2.8), za vse besede z lihim indeksom položaja pa trigonometrična funkcija kosinus (enačba 2.9).

$$PE_{(pos;2i)} = \sin \frac{pos}{10000^{2i/D}} \tag{2.8}$$

$$PE_{(pos;2i+1)} = \cos \frac{pos}{10000^{2i/D}} \tag{2.9}$$

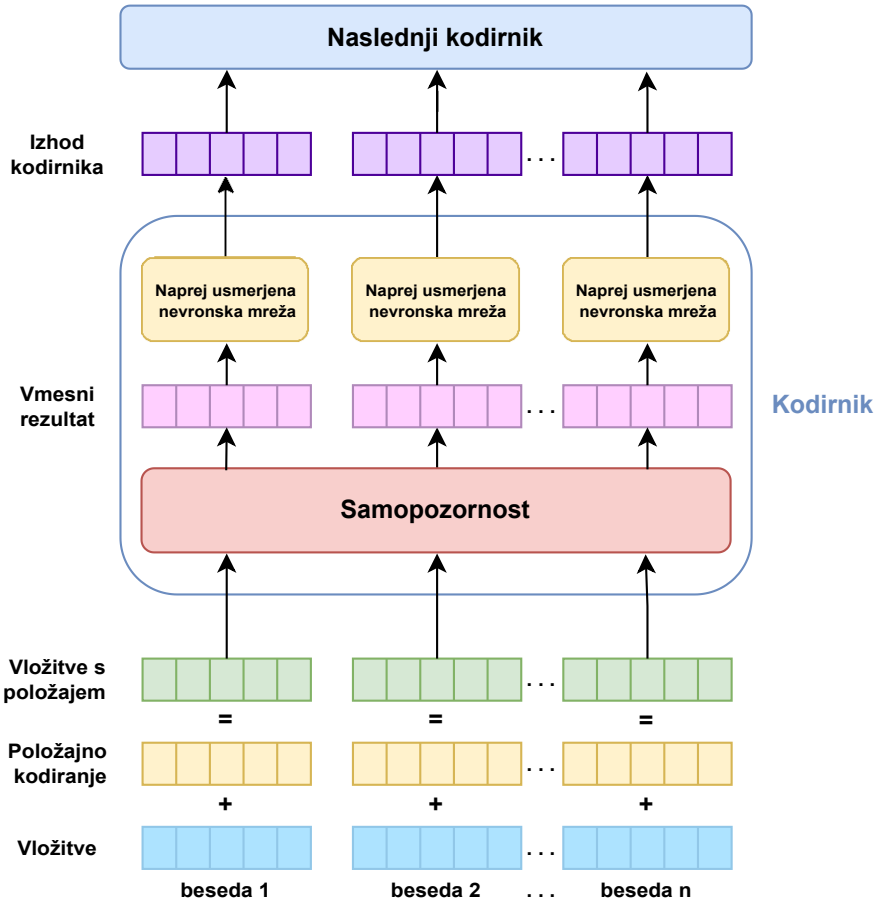
Vložitev nato potuje skozi vse komponente kodirnika, ki vrne vmesni rezultat. Rezultat vsakega kodirnika je tako vhod naslednjemu kodirniku na skladu vse do zadnjega, katerega izhod predstavlja vhod prvega dekodirnika. Na skladu dekodirnikov je postopek enak. Vsak rezultat dekodirnika je vhod naslednjemu dekodirniku na skladu vse do konca, ko se izhodni vektorji glede na aplikacijo uporabijo v nadaljnjih procesih ali pa se pretvorijo nazaj v besedilo. Slika 2.11 prikazuje prehod vložitev skozi sklad kodirnikov.

Na tem mestu lahko opazimo eno izmed glavnih značilnosti transformerjev. Vsaka predstavitev besede v obliki vložitve potuje po ločeni vzporedni poti skozi kodirnik. Na plasti samopozornosti se obravnavajo odvisnosti med vložitvami (tj. odvisnosti med besedami v vhodnem besedilu), skozi naprej usmerjeno nevronsko mrežo pa se vložitve pošljejo sočasno. Prehod vložitev skozi sklad dekodirnikov je podoben. Vhoda v posamezen dekodirnik sta izhod prejšnjega dekodirnika in izhod zadnjega kodirnika.

Samopozornost je zmožnost povezovanja ene besede v besedilu z drugo besedo v besedilu v smislu konteksta. Vzemimo za primer stavek:

Miha je želel z Matjažem igrati košarko, ta pa je hotel igrati nogomet.

V podanem stavku je jasno, da se beseda “ta” nanaša na Matjaža, prav tako pa razumemo povezavo med Miho in košarko ter Matjažem in nogometom. V arhitekturi transformer je zmožnost takšnega povezovanja realizirana z mehanizmom samopozornosti. Ko transformer obdeluje vsako besedo in s tem vsako pozicijo znotraj vhodnega besedila, pri tem upošteva druge položaje te besede (in drugih besed) z namenom, da zanjo ustvari boljšo kodirano vrednost.



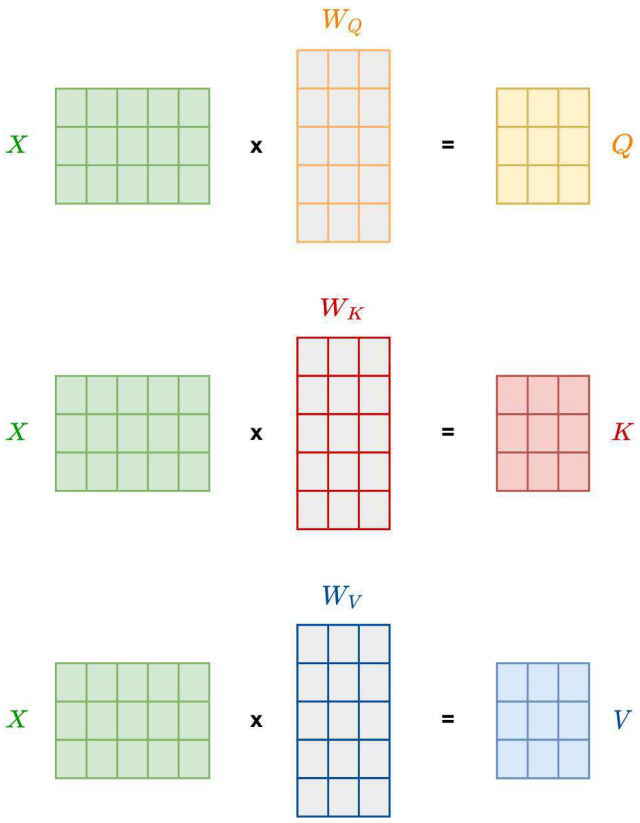
Slika 2.11: Prehod vložitev skozi sklad kodirnikov.

Samopozornost torej definira povezave ostalih relevantnih besed na vhodu s trenutno besedo, ki je v postopku obdelave. Če bi pogledali stanje povezav za podan stavek takoj po obdelavi s plastjo samopozornosti, bi dobili sledeče stanje:

Miha je želel z Matjažem igrati košarko, ta pa je hotel igrati nogomet.

Opazimo, da sta povezana pojma Miha in košarka (označeno z rdečo barvo), prav tako pa Matjaž in nogomet (označeno z modro barvo). Beseda “ta” se nanaša na Matjaža in

je tudi vzrok za povezavo z besedo “nogomet”. Vidimo torej, da z uporabo mehanizma samopozornosti pridemo do podobnih povezav, ki jih tvorimo ljudje z bralnim razumevanjem jezika. Poglejmo si še podrobneje, kaj se dogaja v plasti samopozornosti (slika 2.12). Vložitve, ki pridejo v plast samopozornosti, najprej preoblikujemo v matriko X .

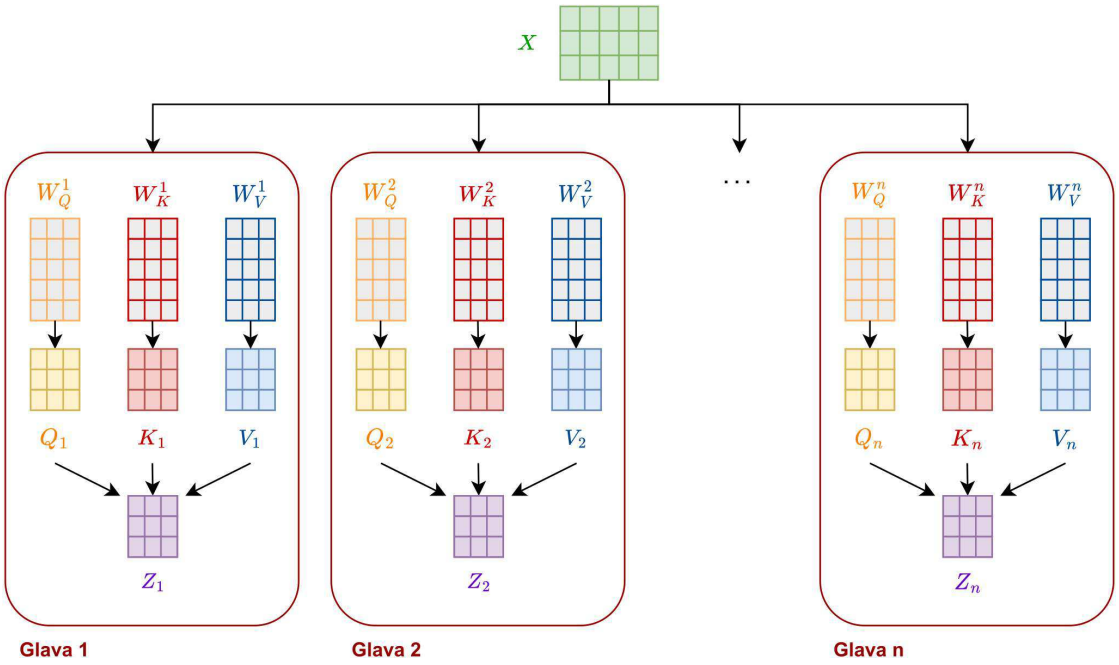


Slika 2.12: Prikaz izračuna matrik Q, K in V znotraj plasti samopozornosti.

Vsaka vrstica v matriki X je torej vektor, ki predstavlja vložitev posamezne besede. Zatem izračunamo matrike Q (ang. queries), K (ang. keys) in V (ang. values) tako, da matriko X pomnožimo z matrikami uteži W_Q, W_K in W_V, ki predstavljajo učljive parametre modela. Vsaka vrstica v matrikah Q, K in V se ujema z vrstico v matriki X . Te matrike predstavlja-jo abstrakcije posameznih kontekstov besed, ki tvorijo osnovo za izračun pozornosti. Na koncu matrike Q, K in V uporabimo za izračun matrike Z , ki vsebuje vektorje samopozornosti za vsako vrstico vhodne matrike X (enačba 2.10).

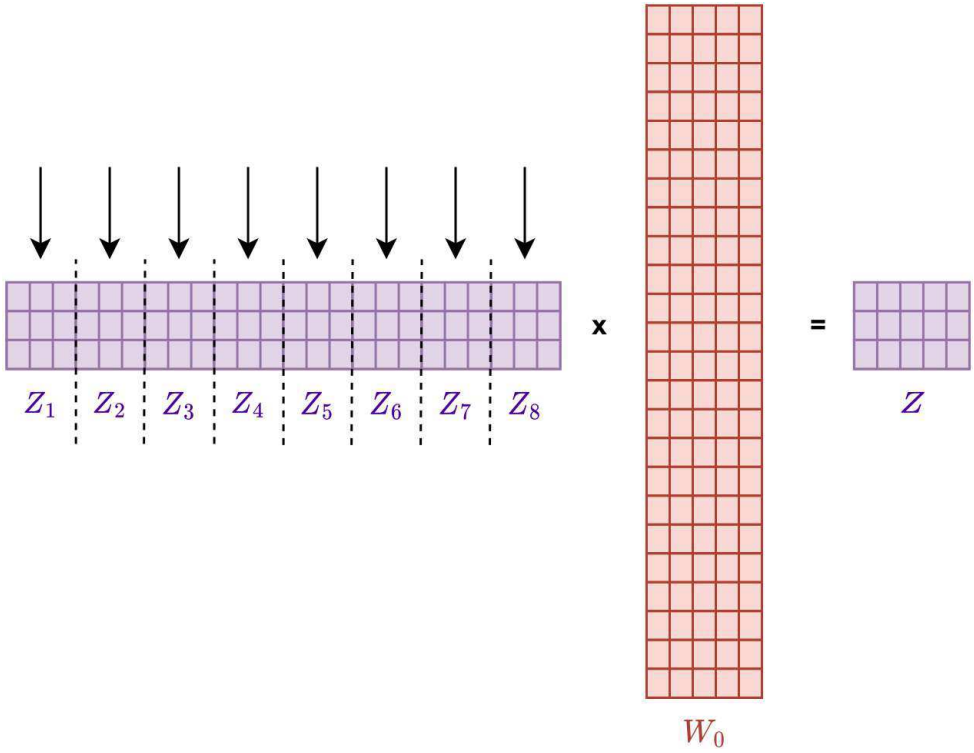
$$Z = \text{softmax}\left(\frac{QK}{\sqrt{d_k}}\right)V \tag{2.10}$$

Opazimo uporabo funkcije softmax. Ta funkcija normalizira vse vmesne rezultate tako, da so pozitivni in je njihova vsota enaka 1. Za vložitve dobimo tako numerične vrednosti, ki predstavljajo vrednosti samopozornosti in jih uporabimo pri odločanju o povezavah med različnimi besedami. Omeniti je potrebno še člen $\sqrt{d_k}$ v enačbi 2.10, kjer je d_k dolžina vrstice matrike K (tj. dimenzija vektorjev k). Ta člen se uporablja za stabilizacijo učenja. V izvirnem članku [13] so uporabili vrednost $d_k = 64$, kar je privzeta vrednost v implementacijah transformerjev. Vrednost je seveda lahko drugačna, smiselno pa je izbirati vrednosti, ki so potence števila 2. Izvirna arhitektura transformerjev uporablja mehanizem večglave samopozornosti (ang. multi-headed self-attention), pri kateri vložitve na vhodu uporabimo z več glavami, ki izvajajo ločene izračune samopozornosti (slika 2.13).



Slika 2.13: Prikaz izračuna vektorjev Q_i, K_i in V_i pri izračunu večglave samopozornosti.

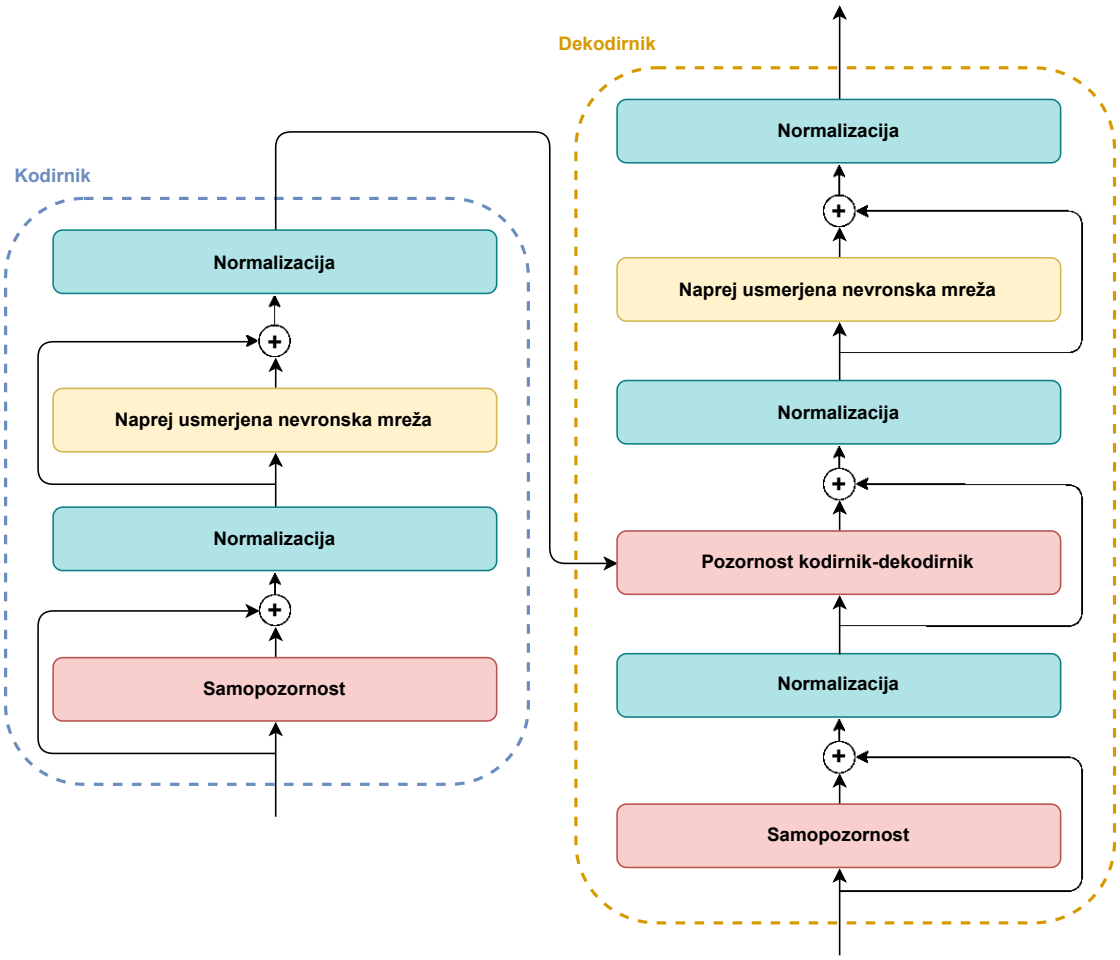
Mehanizem večglave samopozornosti omogoča, da se pozornost preverja na različnih položajih v besedilu. Ker vsaka glava izvaja neodvisen izračun samopozornosti, kot rezultat dobimo več skupin matrik Q, K in V. V izvirnem članku so uporabili 8 glav, kar je privzeta vrednost v implementacijah transformerjev, tudi tukaj pa je smiselno izbirati vrednosti, ki so potence števila 2. Rezultat večglave samopozornosti je več matrik Z, ki jih je potrebno pred posredovanjem naprej usmerjeni nevronski mreži združiti. Združena matrika se v izvirni arhitekturi transformerjev množi z matriko W_0, ki je naučena skupaj z modelom. Ta proces prikazuje slika 2.14.



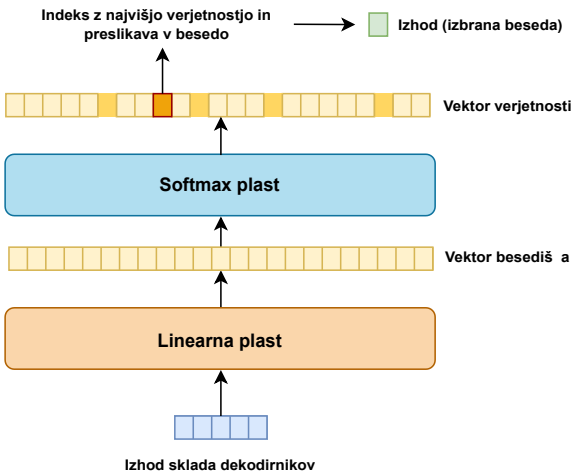
Slika 2.14: Prikaz združevanja rezultatov večglave samopozornosti.

Na koncu dobimo matriko Z , ki je ustreznih dimenzij za naslednji korak sočasne obdelave z naprej usmerjeno nevronsko mrežo. V kodirniku in dekodirniku se med vsako plastjo izvede tudi normalizacija plasti (ang. layer normalization), vsaka plast pa ima tudi rezidualno povezavo. Ideja tega je, da se vhod v posamezne komponente kodirnika ali dekodirnika prišteje izhodu komponente nato pa se rezultat normalizira (slika 2.15). To je dobro poznana tehnika na področju globokega učenja in pripomore k bolj stabilnim gradientom, hitrejšemu učenju in višji natančnosti modela.

Rezultat zadnjega kodirnika na skladu kodirnikov se pošlje v vsak dekodirnik na skladu dekodirnikov, kjer se uporabi v plasti pozornosti kodirnik-dekodirnik, ki je komponenta dekodirnika. Vsak dekodirnik izvaja enako samopozornost med vhodnimi vektorji, dodatno pa še pozornost glede na vhod iz zadnjega kodirnika. Slednja deluje kot plast večglave pozornosti z matrikama K in V , pri čemer se matriki K in V tvorita iz vektorjev kodirnika, matrika Q pa iz izhoda prejšnje plasti dekodirnika. Zadnji dekodirnik na skladu dekodirnikov vrača rezultat v obliki vektorja, ki se glede na aplikacijo pošlje skozi dodatno procesiranje ali pa se pretvori nazaj v besedilo. Ta pretvorba se v primeru napovedovanja naslednje besede (tj. pri generiranju besedila ali strojnem prevajanju) zgodi z uporabo dveh dodatnih plasti (slika 2.16). Najprej gre vektor v linearno plast, ki ga preslika v vek-

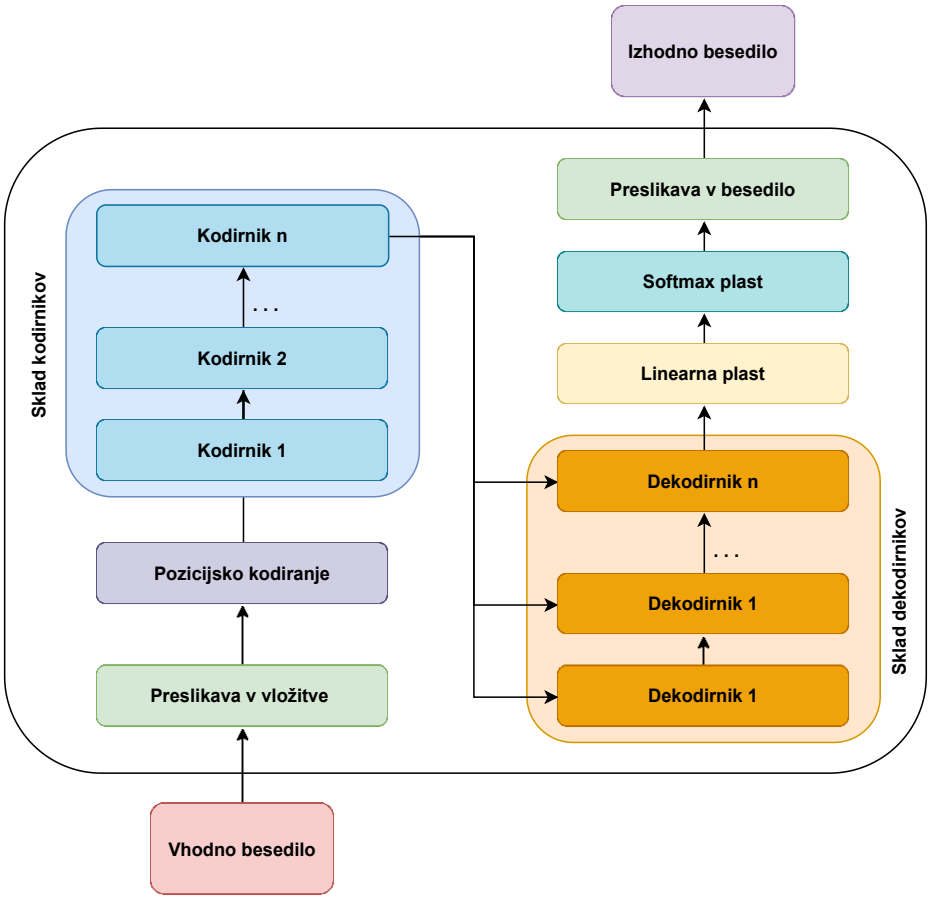


Slika 2.15: Struktura kodirnika in dekodirnika z dodano normalizacijo plasti in rezidualnimi povezavami.



Slika 2.16: Pretvorba vektorja nazaj v besedilo z linearno in softmax plastjo.

tor dimenzije besedišča (tj. vseh besed). Zatem gre ta vektor v plast s funkcijo softmax, ki pretvori vrednosti vektorja v verjetnosti. Te so pozitivne in njihova vsota je enaka 1. Iz-bere se komponenta vektorja z najvišjo verjetnostjo, pripadajoča beseda pa se na koncu vrne kot rezultat. Dogajanje na izhodu je povsem odvisno od končne aplikacije, saj lahko transformerje uporabljamo za napovedovanje naslednjih besed, strojno prevajanje, analizo sentimenta ali drugo obliko klasifikacije. Kot lahko opazimo, ima arhitektura transformerjev veliko plasti in komponent, ki so med seboj povezane na kompleksen način. Slika 2.17 prikazuje celotno arhitekturo transformerjev vključno s povezanimi plastmi in njihovimi komponentami.



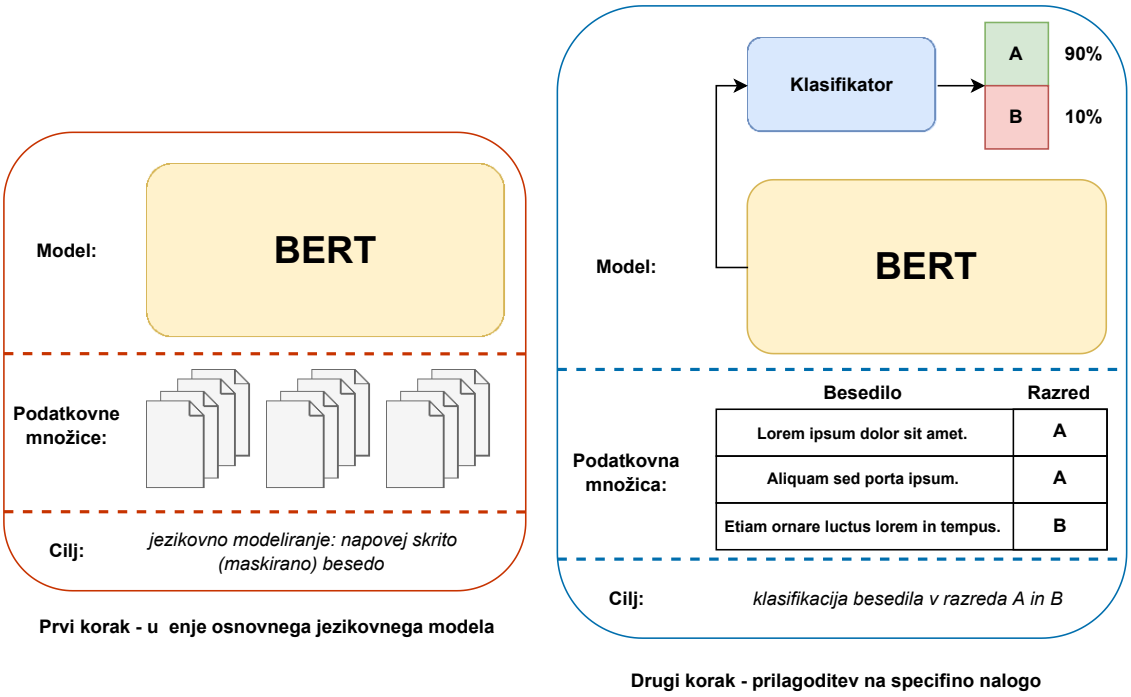
Slika 2.17: Povezane komponente in plasti znotraj transformerjev.

2.2.6 Globoke nevronske mreže BERT

Z razvojem transformerjev so se sčasoma začele pojavljati nove ideje, ki to arhitekturo uporabljajo pri svojem delovanju. Transformerji vsebujejo kodirni in dekodirni del, oba pa sta bila v osnovi zasnovana za naloge strojnega prevajanja. Pozneje so se pojavile aplikacije, ki uporabljajo samo kodirni del ali samo dekodirni del transformerjev. Posledično so se začeli pojavljati različni tipi velikih jezikovnih modelov (ang. large language models, LLM), ki so naučeni na ogromnih množicah besedil in omogočajo osnovno razu-

mevanje slovničnih pravil in pomenov besed naravnega jezika. Med najpogostejše aplikacije, ki uporabljajo kodirni del transformerjev, štejemo analizo sentimenta, razpoznavanje imenskih entitet in klasifikacijo besedil, medtem ko je uporaba dekodirnega dela transformerjev značilna za aplikacije generiranja besedila, strojnega prevajanja in samodejnega dopolnjevanja besed.

Leta 2018 so pri Google-u [14] predstavili globoko nevronsko mrežo BERT, ki za svoje delovanje uporablja samo kodirni del arhitekture transformer, hkrati pa so omogočili širši javnosti prost dostop do naučenega jezikovnega modela. To je tudi pomembna točka v zgodovini razvoja metod na področju obdelave naravnega jezika, saj je prvič nastal zelo velik in učinkovit jezikovni model, ki ga lahko uporabi kdorkoli za reševanje nalog na področju obdelave naravnega jezika. Celotno idejo globokih nevronskih mrež BERT si lahko predstavljamo v dveh korakih (slika 2.18).



Slika 2.18: Učenje modela BERT in prilagoditev modela BERT na specifično nalogo.

Prvi korak je učenje osnovnega jezikovnega modela BERT. Pri tem gre za samonadzorovano učenje (ang. self-supervised) modela z ogromnimi količinami besedil. Cilj tega učenja je napovedovanje skritih (maskiranih) besed v vhodnem besedilu. Rezultat učenja je osnovni jezikovni model, ki je sposoben zaznati pomenske vzorce v besedilih. Ta osnovni jezikovni model lahko uporabimo kot začetno točko pri reševanju nalog na področju obdelave naravnega jezika, saj je izhod iz globokih nevronskih mrež BERT latentna predstavitev vsebine vhodnega besedila, ki jo je mogoče uporabiti kot vhod v nadaljnje postopke kla-

sifikacije. Ta predstavlja drugi korak uporabe globokih nevronskih mrež BERT. Ideja tukaj je, da uporabimo vnaprej naučen jezikovni model BERT, ki ga s pomočjo nadzorovanega učenja prilagodimo na specifično nalogo obdelave naravnega jezika. Pri tem uporabimo lastne učne, testne in validacijske množice, v proces pa za jezikovnim modelom BERT dodamo še glavo, ki rešuje izbrano nalogo. Največkrat govorimo o klasifikacijski glavi, tj. polno povezani nevronski mreži, ki predstavlja klasifikator.

Enojezični in večjezični modeli BERT so na voljo v več različicah. Najpogosteje uporabljeni sta različici $\text{BERT}_{\text{BASE}}$ in $\text{BERT}_{\text{LARGE}}$, ki se razlikujeta po številu kodirnikov, velikosti vložitev in glav v plasteh samopozornosti. Obstajajo tudi drugi prosto dostopni modeli, ki temeljijo na arhitekturi, ki jo uporabljajo modeli BERT. Najbolj znani so modeli RoBERTa [96], DistilBERT [97] in XLNet [98]. Ti modeli izboljšajo BERT v smislu hitrosti učenja in uspešnosti pri nalogah na področju obdelave naravnega jezika. Leta 2021 je nastal model SloBERTa [99], ki je bil naučen izključno na slovenskih besedilih. Ta model je po strukturi zelo podoben francoskemu modelu Camembert [100], ki je zasnovan na modelu RoBERTa. Primerjavo različnih struktur teh modelov prikazuje tabela 2.7.

Tabela 2.7: Primerjava struktur različnih modelov, ki uporabljajo podobno arhitekturo kot BERT.

Jezikovni model	**Število kodirnikov**	**Velikost vložitev**	**Število glav samopozornosti**	**Število parametrov**
$\text{BERT}_{\text{BASE}}$	12	768	12	110M
$\text{BERT}_{\text{LARGE}}$	24	1024	16	340M
$\text{RoBERTa}_{\text{BASE}}$	12	768	12	125M
$\text{RoBERTa}_{\text{LARGE}}$	24	1024	16	355M
DistilBERT	6	768	12	66M
$\text{XLNet}_{\text{BASE}}$	12	768	12	110M
$\text{XLNet}_{\text{LARGE}}$	24	1024	16	340M
SloBERTa	12	768	12	125M

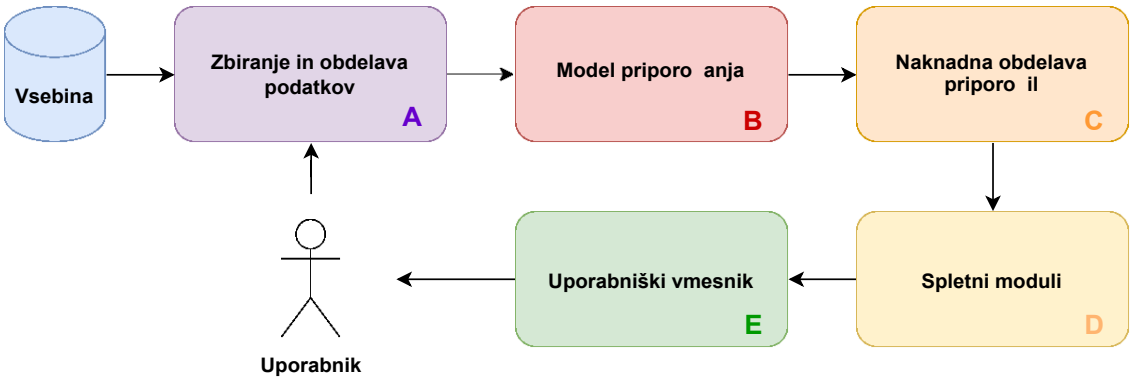
Doktorska disertacija se ukvarja s področjem obdelave naravnega jezika v sklopu osrednjih nalog klasifikacije besedil in pridobivanja informacij. Obe osrednji nalogi obdelave naravnega jezika sta ključni komponenti v sodobnih priporočilnih sistemih, ki je še eno aktivno, razvijajoče se znanstveno področje, kjer je uporaba globokih nevronskih mrež v zadnjem času vse bolj prisotna.

2.3 Priporočilni sistemi

Priporočilni sistemi (ang. recommender systems) so sistemi, ki uporabnikom vračajo njim prilagojene rezultate iskanja [101]. Slednji so uporabnikom podani kot seznami priporočil, ki temeljijo na uporabniških interakcijah z elementi priporočanja (npr. kliki, ogledi, ocene in recenzije) ali pa na njihovi vsebinski podobnosti, kjer je delovanje priporočilnega sistema zelo podobno delovanju iskalnika [102]. Pravzaprav lahko sodobne priporočilne sisteme v veliki meri danes definiramo kot naslednike iskalnikov [103]. Raziskovalno področje priporočilnih sistemov se smatra kot veja umetne inteligence, saj so sodobni pristopi priporočanja izvedeni s pomočjo algoritmov umetne inteligence.

Uporaba sodobnih priporočilnih sistemov v različnih storitvah na spletu je danes neizogibna. Priporočilni sistemi so integrirani v spletnih trgovinah (npr. Amazon in AliExpress), multimedijskih storitvah (npr. YouTube, Netflix in Spotify) in družbenih omrežjih (npr. Facebook, Instagram, Twitter in TikTok). Najpogostejši razlog za vključevanje priporočilnega sistema v delovanje storitve je prilagoditev na uporabnikovo interakcijo s storitvijo z namenom posredovanja relevantnih vsebin. Tako je uporabnik izpostavljen širšemu naboru vsebin in s tem tudi možnosti oglaševanja, kar je v določenih primerih tudi osrednji poslovni model storitve.

Priporočilne sisteme definiramo s 5 komponentami (slika 2.19), povezanimi v delovni tok [104], ki poteka vse od zalednega delovanja do uporabniškega vmesnika [105].



Slika 2.19: Komponente priporočilnega sistema, povezane v delovni tok.

Prva komponenta se ukvarja z zbiranjem in obdelovanjem podatkov (A). Na tem mestu se nestrukturirani podatki pretvorijo v strukturirano in povezano obliko. Podatki se prav tako po potrebi obdelajo in preoblikujejo v obliko primerno za uporabo v modelu priporočanja. V tem delu delovnega toka govorimo o prečiščevanju in normalizaciji podatkov ter o tvorbi in izbiri značilk. Naslednji korak v delovnem toku priporočilnih sistemov velja za najpomembnejšega. Gre za model priporočanja (B), ki zna iz vhodnih podatkov napo-

vedati ali izbrati tiste, ki so relevantni za končnega uporabnika. Rezultat je največkrat v obliki rangiranega seznama, urejenega po oceni relevantnosti. Bolj relevantni elementi so na začetku seznama, manj relevantni pa na koncu.

Dobljeni seznami gredo v tretjem koraku delovnega toka skozi proces naknadne obdelave (C). Ta komponenta priporočilnega sistema izvaja različne spremembe nad dobljenim rangiranim seznamom. Tipično se znotraj te komponente seznami spreminjajo zaradi upoštevanja demografskih, geografskih ali kontekstnih omejitev. Če npr. govorimo o priporočanju artiklov v spletni trgovini, se na tem mestu iz seznama odstranijo vsi artikli, ki so sicer priporočeni, vendar za njih ne obstaja možnost dostave na uporabnikovo geografsko lokacijo. Seznam se lahko tudi obogati tako, da se dodajo novi elementi, ki so posledica poslovne logike – v seznam priporočil se npr. na eno izmed pozicij doda sponzoriran artikel, ker imata spletna trgovina in proizvajalec tistega artikla dogovor o oglaševanju. V seznamu lahko pride tudi do preurejanja zaradi kontekstnih omejitev, npr. določeni artikli v seznamu, ki jih je uporabnik že kupil v preteklosti, se glede na kontekst artikla uvrstijo nižje ali višje v seznamu. Nižje uvrščanje bi se izvedlo za artikel, ki ga uporabniki navadno ne kupujejo redno oziroma večkrat zapored v krajšem časovnem intervalu (npr. tiskalnik). Višje uvrščanje bi se uporabilo za artikel, ki ga uporabniki navadno kupujejo redno ali ko gre za potrošni material (npr. kartuša za tiskalnik).

V četrtem koraku delovnega toka priporočilnega sistema se uporabljajo spletni moduli (D), ki služijo posredovanju rezultatov v uporabniški vmesnik. Največkrat so ti moduli implementirani kot spletne storitve, ki imajo ob primarni nalogi posredovanja rezultatov tudi sekundarno nalogo beleženja zahtevkov in odgovorov v dnevnik. To je pomembno za analizo delovanja kvalitete priporočilnega sistema. Na tem mestu se ponavadi vključijo različni pristopi vrednotenja priporočilnih sistemov.

Zadnji, peti korak delovnega toka priporočilnega sistema predstavlja uporabniški vmesnik za prikaz rezultatov uporabniku (E). Ta komponenta priporočilnega sistema definira, kako bodo uporabniki videli rezultat priporočanja in kako bodo priporočilnemu sistemu z interakcijami posredovali povratno informacijo. Z vidika uporabnosti priporočilnega sistema je dobra praksa, da uporabniki dobijo utemeljitev rezultata priporočil. Na primer: ob ogledu nekaj artiklov v spletni trgovini uporabnik dobi priporočilo za artikel X z utemeljitvijo “Priporočamo artikel X, ker ste kupili artikla Y in Z”.

Posamezne zgoraj opisane komponente so lahko bolj dovršeno implementirane kot druge. Največkrat gre pri razvoju priporočilnih sistemov za osredotočanje na komponenti predobdelave podatkov (A) in algoritma oziroma modela priporočanja (B). Preostale komponente (C, D in E) so ponavadi zapostavljene do trenutka, ko je potrebno priporočilni sistem izvesti v produkcijskem okolju. Z vidika razvoja modela priporočanja so se v preteklosti

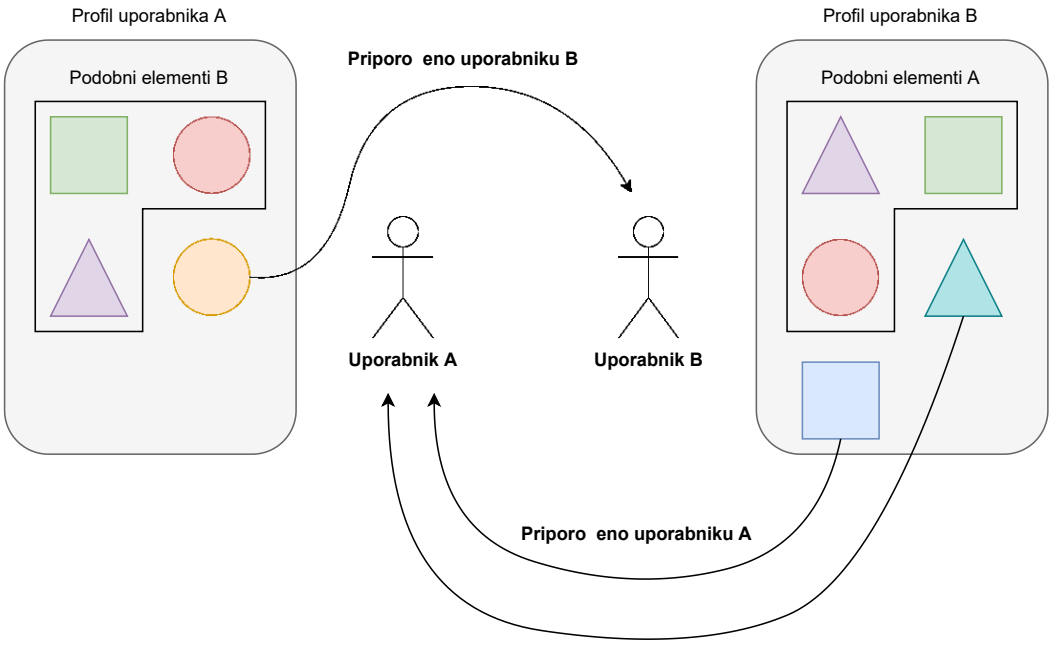
priporočilni sistemi razvijali bodisi s sodelovalnim filtriranjem, bodisi z vsebinskim filtriranjem, danes pa so bolj pogosti hibridni priporočilni sistemi.

2.3.1 Sodelovalno filtriranje

Tehnika priporočanja, ki jo imenujemo sodelovalno filtriranje, se pri svojem delovanju osredotoča na uporabnika in njegovo aktivnost. Pri tem se uporabljajo specifične aktivnosti posameznega uporabnika, ki določajo kako pomembni so zanj elementi priporočanja. Med elemente priporočanja štejemo vso vsebino, ki jo želimo priporočati (npr. če priporočamo filme, je element priporočanja posamezen film). Za določanje, ali je nek element priporočanja za uporabnika zanimiv, najpogosteje uporabljamo število ogledov, prenosov, ocene zadovoljstva, čas ogleda in položaj miškinega kurzorja. Naštete interakcije se smatrajo kot pozitivne interakcije uporabnika s sistemom priporočanja, saj se je na priporočila odzval na načine, ki potrjujejo zanimanje zanje. S temi informacijami se tvori profil uporabnika v obliki vektorja. S primerjavo profilov uporabnikov v vektorski obliki se nato poiščejo drugi uporabniki, ki so podobni trenutnemu uporabniku. V ta na-men se uporabljajo mere podobnosti kot so kosinusna razdalja, Pearsonova korelacija in Jaccardov indeks [101].

V gruči profilov podobnih uporabnikov se nato poiščejo razlike. To so tisti elementi priporočanja, ki so bili za podobne uporabnike zanimivi, uporabnik, za katerega se izvaja priporočanje, pa še ni prišel v stik z njimi. Ti elementi priporočanja so torej kandidati za priporočila, saj glede na podobnost profila uporabnika z drugimi podobnimi uporabniki obstaja velika verjetnost, da mu bodo ti elementi priporočanja zanimivi.

Slika 2.20 prikazuje delovanje sodelovalnega filtriranja, kjer se izvaja priporočanje med uporabnikoma A in B. Iz slike sta razvidna profila obeh uporabnikov. Opazimo lahko, da obstajajo nekateri elementi priporočanja, ki so skupni obema uporabnikoma (zeleni kvadrat, rdeči krog in vijolični trikotnik). Oblika elementov priporočanja (kvadrat, krog in trikotnik) ponazarja različne tipe elementov priporočanja, barva elementov pa ponazarja vsebinsko različico določenega tipa elementa priporočanja (npr. oblika kvadrata ponazarja žanr akcijskih filmov, barva kvadrata pa specifičen akcijski film). Označena množica skupnih elementov je dovolj velika, da lahko smatramo uporabnika A in B kot uporabnika s podobnima profiloma. Hkrati opazimo, da je uporabnik A imel pozitivno interakcijo z elementom priporočanja označenim z oranžnim krogom. Tega elementa ni v profilu uporabnika B, zato je ta element kandidat za priporočanje uporabniku B. Uporabnik B je imel pozitivno interakcijo z elementoma priporočanja označenima z modrim kvadratom in turkiznim trikotnikom. Teh elementov ni v profilu uporabnika A, zato sta ta dva elementa kandidata za priporočanje uporabniku A.



Slika 2.20: Prikaz delovanja sodelovalnega filtriranja.

Med prednosti sodelovalnega filtriranja štejemo enostavnost algoritmov, kot največjo slabost pa lahko izpostavimo problem hladnega začetka (ang. cold-start problem). To je situacija, ko nimamo dovolj velikega števila uporabniških interakcij z elementi priporočanja ali dovolj velikega števila uporabnikov, da bi lahko izvajali priporočanje. V tem primeru je ena izmed možnosti priporočanje naključnih elementov, seveda pa je za razreševanje tega problema bolj smiselno uporabiti metode, ki upoštevajo opise vsebine elementov priporočanja. V tem primeru govorimo o metodah vsebinskega filtriranja.

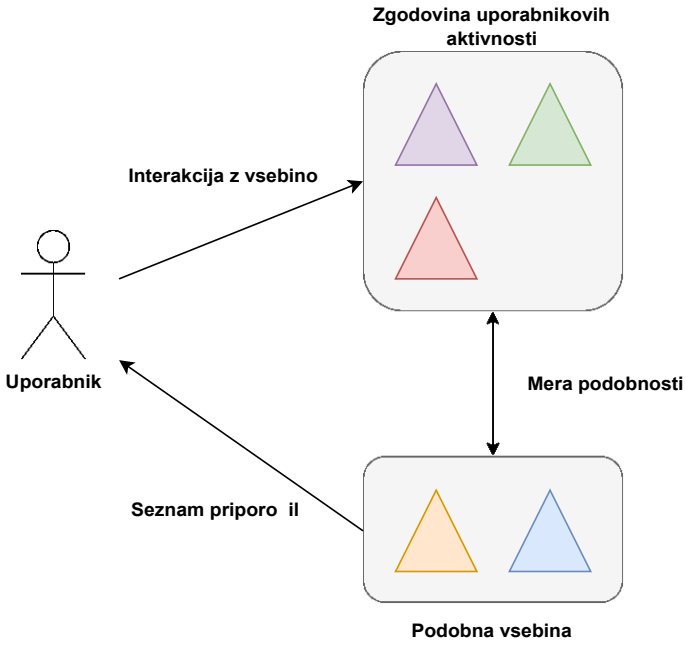
2.3.2 Vsebinsko filtriranje

Vsebinsko filtriranje je alternativen način priporočanja, pri katerem se upoštevajo podobnosti med strukturiranimi ali nestrukturiranimi opisi elementov priporočanja. Med strukturirane opise štejemo vnaprej določene lastnosti z njihovimi vrednostmi, nestrukturirani opisi elementov pa so običajno podani v obliki besedila. Priporočanje z metodami vsebinskega filtriranja poteka tako, da se uporabniku priporočajo elementi, ki so vsebinsko najbolj podobni drugim elementom, ki jih imamo na voljo za priporočanje. Metode vsebinskega filtriranja, ki uporabljajo strukturirane opise elementov priporočanja, tako z merami podobnosti iščejo najpodobnejše druge elemente priporočanja in jih vrnejo uporabniku v obliki seznama, urejenega po podobnosti.

Najpogosteje uporabljena mera podobnosti je kosinusna razdalja, kar pomeni, da so strukturirani opisi (podobno kot pri sodelovalnem filtriranju) v obliki vektorjev značilk. Pri nestrukturiranih opisih elementov se uporabljajo metode obdelave naravnega jezika, s

katerimi pridobimo značilke za primerjavo elementov priporočanja med seboj.

Priporočilni sistemi, ki uporabljajo vsebinsko filtriranje, so pravzaprav razširitev iskalnikov in s tem tudi njihovi moderni nasledniki. Slika 2.21 prikazuje delovanje vsebinskega filtriranja za uporabnika. Zgodovina uporabnikovih aktivnosti zajema pozitivno interakcijo z elementi priporočanja, označenimi z vijoličnim, zelenim in rdečim trikotnikom. Tudi na tej sliki oblika elementa priporočanja določa tip, barva elementa priporočanja pa vsebin-sko različico elementa priporočanja. Ker gre za vsebinsko filtriranje, vidimo, da se mera podobnosti izvaja nad istim tipom elementov (tj. vsi elementi so ponazorjeni z obliko trikotnika). Vsebinsko so ti elementi seveda drugačni (tj. vsi elementi so označeni z drugo barvo). Z uporabo mere podobnosti vsebinski filter najde dva elementa s podobno vsebino, ki sta ponazorjena z oranžnim in modrim trikotnikom. Ta dva elementa sta kandidata za priporočanje uporabniku.



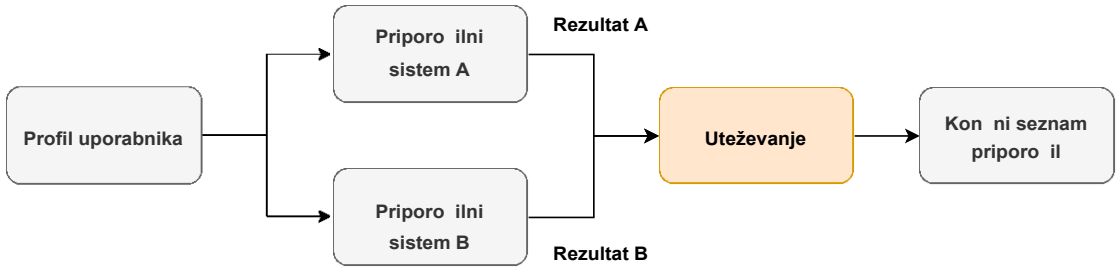
Slika 2.21: Prikaz delovanja vsebinskega filtriranja.

Z uporabo vsebinskega filtriranja je uporabnikom možno priporočati vsebine tudi takrat, ko imamo malo množico vsebin in uporabnikov. Težava se pojavi, če so vsebine razporejene v vsebinsko strogo ločene gruče. V tem primeru bo rezultat vsebinskega priporočanja omejen na fiksno množico, ki jo določa mera podobnosti značilk. V praksi to pomeni pomanjkanje raznolikosti priporočil, kar seveda ni vedno zaželjeno.

2.3.3 Hibridni priporočilni sistemi

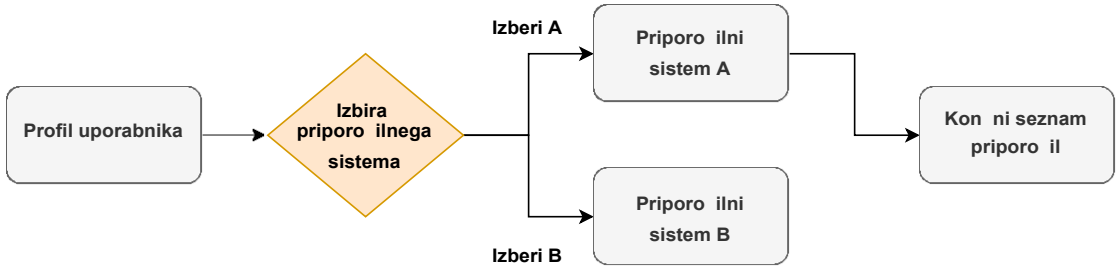
Hibridni priporočilni sistemi se ukvarjajo z reševanjem pomanjkljivosti sodelovalnega filtriranja in vsebinskega filtriranja. Uporabljajo kombinacijo obeh načinov priporočanja ali kombinacijo več različnih tehnik sodelovalnega oziroma vsebinskega filtriranja. Izbira načina filtriranja je predvsem odvisna od ciljev priporočilnega sistema in značilnosti podatkov, ki jih uporabljamo pri priporočanju. Obstaja več vrst izvedb hibridnih priporočilnih sistemov [26–28].

Utežna hibridizacija (ang. weighted hybridization) je izvedba, pri kateri se rezultati dveh ali več načinov priporočanja na podlagi uteževanja združijo v končni seznam priporočil (slika 2.22). V procesu uteževanja se pogosto uporabi kar linearna kombinacija [29].



Slika 2.22: Delovni tok v utežni hibridizaciji.

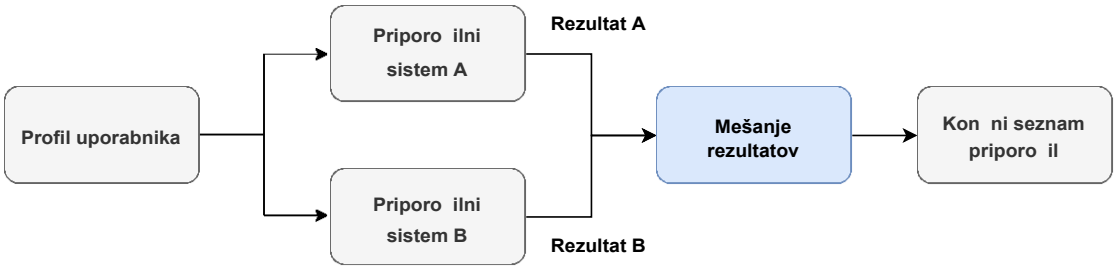
Pri preklopni hibridizaciji (ang. switching hybridization) priporočilni sistem preklaplja med vključenimi načini priporočanja, kar se pogosto uporablja za reševanje problema hladnega začetka, pri čemer se preklaplja med sodelovalnim in vsebinskim filtriranjem. Izbira priporočilnega sistema je pomembna komponenta v preklopni hibridizaciji (slika 2.23), ki je lahko definirana z naborom vnaprej določenih odločitvenih pravil, lahko pa se za izbiranje uporabi tudi bolj kompleksen model odločanja [24].



Slika 2.23: Delovni tok v preklopni hibridizaciji.

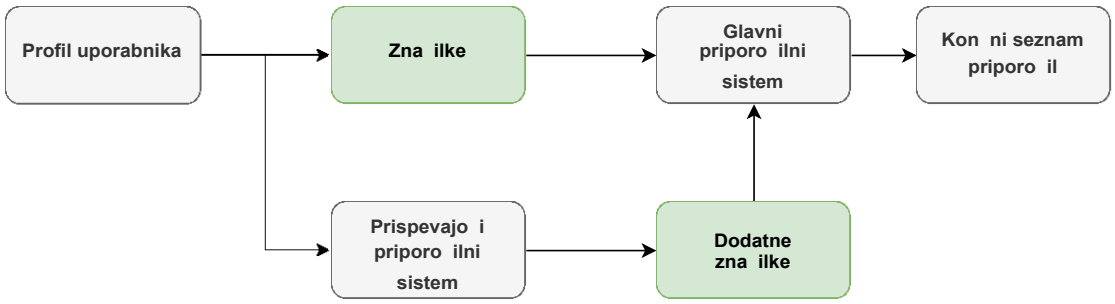
Z mešano hibridizacijo (ang. mixed hybridization) se rezultati dveh ali več načinov priporočanja prikažejo v mešanem vrstnem redu v končnem seznamu priporočil (slika 2.24). Določanje vrstnega reda v končnem seznamu priporočil lahko v procesu mešanja rezultatov

nadziramo z željenim številom priporočil iz vsakega vmesnega seznama priporočil [33].



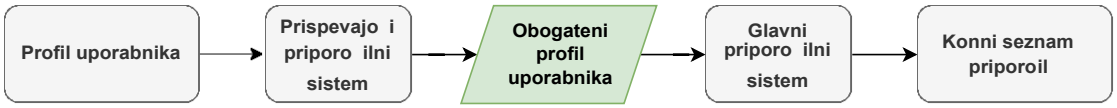
Slika 2.24: Delovni tok v mešani hibridizaciji.

Pri hibridizaciji s kombinacijo značilk (ang. feature combination hybridization) se z uporabo prispevajočih priporočilnih sistemov tvorijo ločene značilke, ki se uporabijo v kombinaciji z značilkami glavnega priporočilnega sistema (slika 2.25). V primeru uporabe sodelovalnega filtriranja kot prispevajočega priporočilnega sistema in vsebinskega filtriranja kot glavnega priporočilnega sistema lahko z dodatnimi značilkami v vsebinsko filtriranje vrinemo značilnosti uporabnika, ki so bile zaznane s sodelovalnim filtriranjem [31].



Slika 2.25: Delovni tok v hibridizaciji s kombinacijo značilk.

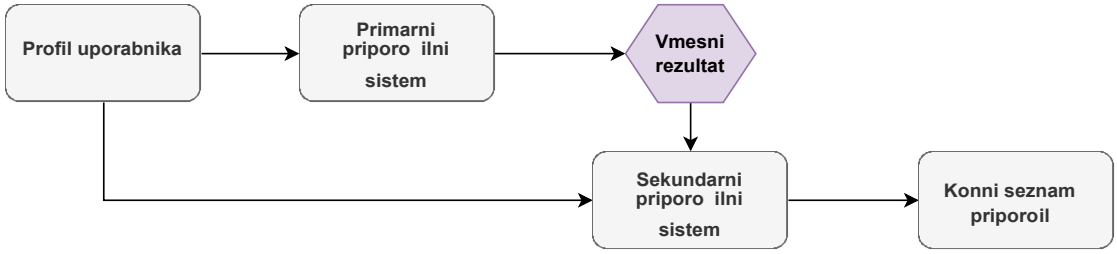
Hibridizacija z obogatenjem značilk (ang. feature augmentation hybridization) je podoben pristop, le da se v tem primeru prispevajoči priporočilni sistem uporabi za pridobivanje značilk, ki obogatijo profil uporabnika [25]. Ta se v obliki obogatenih značilk uporabi kot vhod v glavni priporočilni sistem (slika 2.26).



Slika 2.26: Delovni tok v hibridizaciji z obogatenjem značilk.

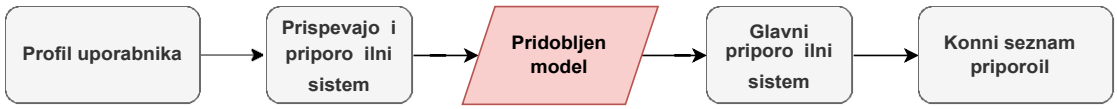
Glavna razlika med kombinacijo značilk in obogatenjem značilk je v tem, da se pri kombinaciji značilk uporablja več značilk iz različnih priporočilnih sistemov, pri obogatenju značilk pa prispevajoči priporočilni sistem tvori značilke za glavni priporočilni sistem.

V kaskadni hibridizaciji (ang. cascade hybridization) se več načinov priporočanja izvaja v striktnem vrstnem redu [30], pri tem pa nastajajo vmesni rezultati, ki se uporabijo za preurejanje končnega seznama priporočanja (slika 2.27).



Slika 2.27: Delovni tok v kaskadni hibridizaciji.

Pri hibridizaciji na meta ravni (ang. meta-level hybridization) se s prispevajočim priporočilnim sistemom zgradi model, ki je vhod glavnemu priporočilnemu sistemu (slika 2.28). Ta način je zelo podoben hibridizaciji z obogatenjem značilk. Bistvena razlika je v tem, da se pri hibridizaciji z obogatenjem značilk obogatijo samo značilke, ki so potem vhod glavnemu priporočilnemu sistemu, pri hibridizaciji na meta ravni pa je vhod v glavni priporočilni sistem kar celoten model, pridobljen s prispevajočim priporočilnim sistemom [32].



Slika 2.28: Delovni tok v hibridizaciji na meta ravni.

2.3.4 Priporočilni sistemi v digitalnih knjižnicah

Priporočanje dokumentov v digitalnih knjižnicah in repozitorijih je zahtevno opravilo, ki uporabnikom pomaga pri raziskovanju in iskanju pomembne literature in drugih virov [18, 21]. Priporočilni sistemi so uporabni tudi v akademskih družbenih omrežjih, kot je recimo Mendeley [15]. Za doseganje smiselnega priporočanja je potreben računalniški algoritem, ki analizira besedilo in metapodatke dokumenta ter tvori seznam njemu najbolj podobnih in relevantnih dokumentov [16, 17]. V Sloveniji je bil v okviru nacionalne infrastrukture odprtega dostopa implementiran priporočilni sistem za vse digitalne knjižnice in repozitorije slovenskih univerz [19]. To vključuje Digitalno knjižnico Univerze v Mariboru (DKUM) [106], Repozitorij Univerze v Ljubljani (RUL) [107], Repozitorij Univerze na Primorskem (RUP) [108], Repozitorij Univerze v Novi Gorici (RUNG) [109], Digitalni repozitorij raziskovalnih organizacij Slovenije (DiRROS) [110] in Repozitorij samostojnih visokošolskih in višješolskih izobraževalnih organizacij (ReVIS) [111].

Priporočanje v digitalnih knjižnicah je ponavadi implementirano z metodami vsebinskega filtriranja, saj algoritem na ta način poskuša zajeti kontekst dokumenta in poiskati njemu

podobne dokumente. V digitalnih knjižnicah je mogoče uporabiti tudi sodelovalno filtriranje, vendar ob pogoju, da imajo te digitalne knjižnice možnost prijavljenih uporabnikov, za katere se v ozadju gradijo uporabniški profili. Moderne digitalne knjižnice sicer omogočajo registracijo uporabnikov, vendar so v veliki večini namenjene širši javnosti in za osnovne funkcionalnosti ne koristijo prijavnih sistemov. Uporabniki s prijavo torej dobijo le možnost nalaganja novih dokumentov, ki pa se v veliki večini prav tako naložijo preko drugih repozitorijev in zbirk dokumentov. Delo s sodelovalnim filtriranjem v okolju digitalnih knjižnic je torej omejeno in iz tega razloga se največkrat uporabi zgolj vsebinsko filtriranje ali pa hibridno priporočanje, ki združuje več vsebinskih filtrov.

Kadar izvajamo priporočanje dokumentov v digitalnih knjižnicah, imamo opravka tako s strukturiranimi kot z nestrukturiranimi opisi dokumentov. Med strukturirane opise dokumentov štejemo metapodatke kot so naslov, povzetek, ključne besede, avtorji in leto izida dokumenta. Med nestrukturirane opise dokumentov štejemo polno besedilo, ki predstavlja dejansko vsebino dokumenta. Pri uporabi vsebinskega filtriranja vsak dokument obdelamo z metodami obdelave naravnega jezika, da pridobimo značilke, s katerimi nato primerjamo dokumente med seboj. Priporočanje z vsebinskim filtriranjem torej poteka tako, da se značilke trenutnega dokumenta, ki ga uporabnik preglejuje, primerjajo z značilkami vseh ostalih dokumentov. Nato se poiščejo trenutnemu dokumentu najbližji dokumenti, ki se ovrednotijo z mero podobnosti in uredijo po padajoči podobnosti. Rezultat je seznam dokumentov, ki se priporočijo uporabniku. Glavni problem vsebinskega filtriranja je, da bodo priporočeni dokumenti omejeni na fiksno množico, ki jo določa mera podobnosti značilk. V praksi to pomeni priporočanje enakih dokumentov ne glede na uporabnika. Gre pravzaprav za omejitev na delovanje iskalnikov, kjer se za vhodni iskalni niz vrne enaka množica rezultatov iskanja. Iz tega razloga je smiselna uporaba hibridnih pristopov za priporočanje. Priporočilni sistemi v digitalnih knjižnicah pa nimajo uporabne vrednosti samo za končne uporabnike, temveč tudi za knjižničarje, ki izvajajo različne dokumentne procese v ozadju, preden so dokumenti na voljo javnosti. Ena izmed pomembnih nalog knjižničarjev je uvrščanje dokumentov v tematska področja s knjižničnim klasifikacijskim sistemom. Pri tej nalogi si pomagajo tako z iskalniki, vgrajenimi v digitalne knjižnice, kot tudi z drugimi orodji, ki se še vedno razvijajo.

2.3.5 Vrednotenje priporočilnih sistemov

Priporočilni sistemi se lahko uporabljajo z različnimi nameni, zato tudi za njihovo vrednotenje obstajajo različni pristopi in metrike [43, 112–114]. S toliko raznolikimi možnostmi je pri vrednotenju priporočilnega sistema potrebno najprej definirati končni rezultat, ki ga želimo doseči z vključitvijo priporočilnega sistema [42]. Intuitivno se obrnemo k ustaljenim metrikam kot sta natančnost in priklic [40, 41], vendar te metrike niso najboljša izbira,

kadar želimo ovrednotiti uporabniško izkušnjo [115]. Osredotočili smo se na uporabo priporočilnega sistema v okolju digitalnih knjižnic med procesom katalogizacije dokumentov. Knjižničarji se med tem procesom srečujejo s problemom prebijanja skozi veliko množico vrstilcev UDK, na koncu pa izberejo manjše število vrstilcev UDK, ki predstavljajo končno klasifikacijo dokumenta. Gre torej za vrednotenje pridobivanja najbolj relevantnih vrstilcev UDK, ki morajo biti visoko rangirani v končnem seznamu priporočil. V nadaljevanju je podrobneje opisanih nekaj metrik s primeri na področju priporočilnih sistemov, ki so smiselne za takšen scenarij. To so metrike HR@K, MAP, MRR in NDCG@K, ki se pogosto uporabljajo v industriji in na raziskovalnem področju priporočilnih sistemov.

Metrika HR@K

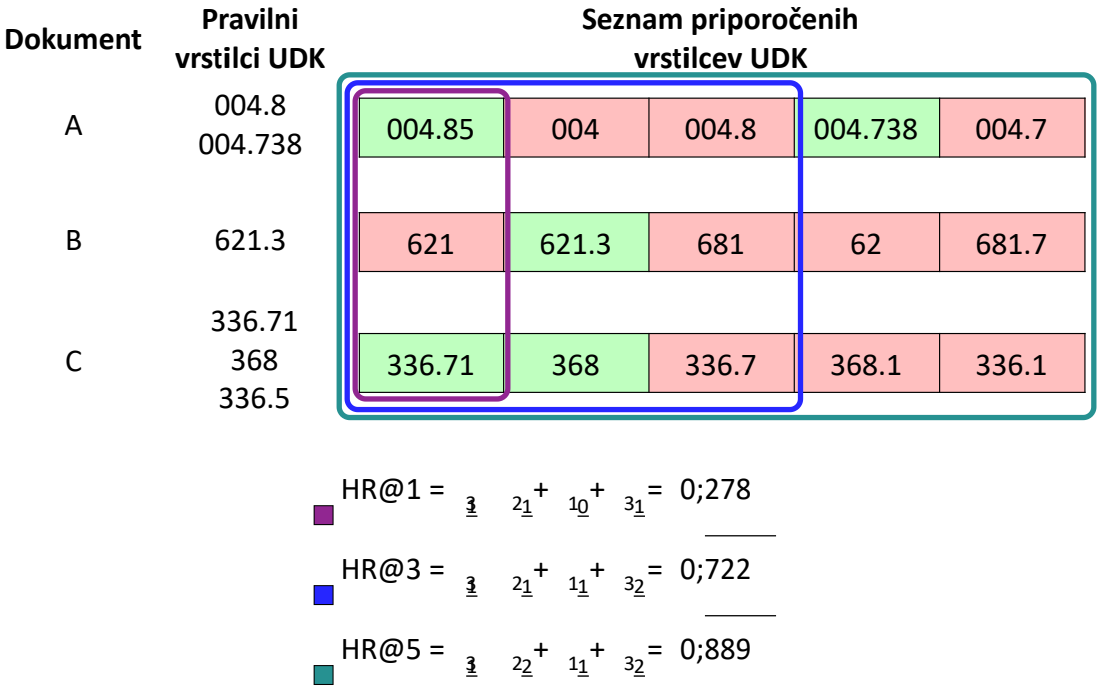
Z vidika vrednotenja priporočanja vrstilcev UDK za dokument, kjer bo uporabnik imel zadnjo odločitev o uporabi vrstilca UDK, je pomembno, da bodo v vrnjenem seznamu tak-šni vrstilci, ki pravilno določajo področje dokumenta. Uspešnost priporočilnega sistema lahko določimo z metriko razmerja zadetkov (ang. hit ratio - HR). Nadalje se nekatere me-trike za vrednotenje priporočilnih sistemov ponavadi tudi omejijo na prvih K priporočil v seznamu, kar v tem primeru vodi v prilagojeno metriko HR@K, ki vrednosti uspešnost za prvih K priporočil. Metriko $HR@K_i$ za i-ti seznam priporočil izračunamo z enačbo 2.11, kjer je n_{hit} število pravilno napovedanih vrstilcev UDK v seznamu priporočil, n_{rel} pa je število vseh možnih pravilnih vrstilcev UDK. Pri tem je seznam priporočil omejen na K priporočenih vrstilcev UDK.

$$HR@K_i = \frac{n_{hit;i}}{n_{rel;i}} \tag{2.11}$$

V primeru več dokumentov, izračunamo metriko HR@K za vsak dokument in izračunamo povprečno vrednost metrike (enačba 2.12).

$$HR@K = \frac{1}{N} \sum_{i=1}^{N} HR@K_i \tag{2.12}$$

Slika 2.29 prikazuje primer s tremi dokumenti (A, B in C), njihovimi pravilnimi vrstilci UDK in seznami priporočenih vrstilcev UDK. Z vijolično, modro in turkizno barvo so obrobljeni deli seznamov, ki se upoštevajo z metrikami HR@1, HR@3 in HR@5. Z zeleno barvo so označeni pravilno napovedani vrstilci UDK v seznamu priporočil, z rdečo pa napačno napovedani vrstilci UDK.



Slika 2.29: Primer izračuna metrike HR@K pri K = [1; 3; 5] za tri sezname priporočil.

Pri izračunu metrike HR@1 (vijolična obroba) vidimo, da smo obravnavali le prve vrstilce v seznamih priporočil in tako pravilno napovedali vrstilca za dokumenta A in C, za dokument B pa nismo dobili pravilne napovedi vrstilca. V izračunu smo upoštevali tudi vse možne pravilne vrstilce UDK za posamezen dokument in tako za dokument A dobili vrednost $\frac{1}{2}$, za dokument B vrednost $\frac{0}{1}$, za dokument C pa vrednost $\frac{1}{3}$. Ko izračunamo povprečje metrike HR@1 za dokumente A, B in C, dobimo vrednost HR@1 = 0,278.

Za izračun metrike HR@3 (modra obroba) upoštevamo prve tri vrstilce v seznamih priporočil. V tem primeru smo pravilno napovedali vrstilec za dokumenta A in B, za dokument C pa smo pravilno napovedali dva vrstilca. Z upoštevanjem vseh možnih pravilnih vrstilcev UDK za posamezen dokument dobimo za dokument A vrednost $\frac{1}{2}$, za dokument B vrednost $\frac{1}{1}$, za dokument C pa vrednost $\frac{2}{3}$. Po izračunu povprečne vrednosti metrike HR@3 za dokumente A, B in C, dobimo vrednost HR@3 = 0,722.

Podobno je pri izračunu metrike HR@5 (turkizna obroba), kjer upoštevamo prvih pet vrstilcev v seznamih priporočil. Ker to sovpada z dolžino seznama priporočil, obravnavamo pravzaprav kar celoten seznam priporočil. V tem primeru smo pravilno napovedali dva vrstilca za dokumenta A in C in en vrstilec za dokument B. Ko izračunamo vrednosti me-

trike dobimo za dokument A vrednost $\frac{2}{2}$, za dokument B vrednost $\frac{1}{1}$, za dokument C pa vrednost $\frac{2}{3}$. Opazimo, da smo za dokumenta A in B uspešno napovedali vse možne pravilne vrstilce, za dokument C pa smo pravilno napovedali dva vrstilca od treh. Po izračunu povprečne vrednosti metrike HR@5 za dokumente A, B in C, dobimo vrednost HR@5 = 0,889.

Število K za metriko HR@K lahko določimo na podlagi dolžine seznama priporočil. V industriji in na raziskovalnem področju se za K največkrat uporabljajo vrednosti 3, 5 in 10. V posebnih primerih je za K smiselno uporabiti tudi vrednosti 1 in 15. V prvem primeru govorimo o zelo strogem vrednotenju, kjer pričakujemo, da bo najvišje rangirano priporočilo dejansko pravilno. Iz primera na sliki 2.29 lahko vidimo, da izbira K = 1 lahko predsta-vlja pomanjkljivost takšne metrike, saj je lahko pravilnih vrstilcev več, ne samo eden. V drugem primeru govorimo o bolj ohlapnem vrednotenju za situacije, kjer bomo uporabniku pokazali relativno veliko število zadetkov (npr. priporočamo več podobnih artiklov v spletni trgovini).

Metrika MAP

Srednja povprečna natančnost (ang. mean average precision) se označuje s kratico MAP in se uporablja za vrednotenje seznamov priporočil za več dokumentov. Najprej se za vsak seznam priporočil izračunajo vrednosti metrike natančnosti pri K (ang. precision at K, P@K), ki se izračuna po enačbi 2.13:

$$P@K_i = \frac{n_{hit;i}}{K} \tag{2.13}$$

V enačbi 2.13 $n_{hit;i}$ predstavlja število pravilno napovedanih vrstilcev UDK, K pa število upoštevanih vrstilcev v i-tem seznamu priporočil. Vrednosti P@K se uporabijo za izračun povprečne natančnosti pri K (ang. average precision, AP@K) za vsak seznam priporočil po enačbi 2.14:

$$AP@K_i = \frac{\sum_{k=1}^{K} P@k_i \; rel(k)_i}{n_{rel;i}} \tag{2.14}$$

$P@k_i$ je vrednost metrike natančnosti pri rangu k, $rel(k)_i$ pa je funkcija, ki vrača 1, če je vrstilec UDK pri rangu k pravilen in 0 sicer (enačba 2.15). Ker metrike MAP pri vrednotenju nismo posebej omejevali, bodo vrednosti K enake kar dolžinam seznamov priporočil.

$$rel(k)_i = \begin{cases} 1; \text{če vrstilec UDK pri rangu k pravilen} \\ 0; \text{sicer} \end{cases} \quad (2.15)$$

V imenovalcu enačbe 2.14 upoštevamo še $n_{rel;i}$, ki je število vseh možnih pravilnih vrstilcev UDK za dokument. Vrednost metrike MAP (enačba 2.16) se nato izračuna kot povprečje vrednosti AP@K za sezname priporočil N dokumentov.

$$MAP = \frac{1}{N} \sum_{i=1}^{N} AP@K_i \quad (2.16)$$

Na sliki 2.30 so podani dokumenti A, B in C z njihovimi pravilnimi vrstilci UDK in pripadajočimi seznami priporočenih vrstilcev UDK. Z zeleno barvo so označeni pravilno napovedani vrstilci, z rdečo pa napačno napovedani vrstilci.

Dokument	**Pravilni vrstilci UDK**	**Seznam priporočenih vrstilcev UDK**				
A	004.8 004.738	004.85	004	004.8	004.738	004.7
B	621.3	621	621.3	681	62	681.7
C	336.71 368 336.5	336.71	368	336.7	368.1	336.1

Slika 2.30: Prvi del primera izračuna metrike MAP za tri sezname priporočil.

Na sliki 2.31 so za vsak dokument podane pripadajoče vrednosti $P@K_i$ z vrednostmi $rel(k)_i$ in izračunom AP, ki je enak AP@5, saj so vsi seznami v primeru dolžine 5. Pri vrednostih $P@K_i$ so z zeleno označene tiste vrednosti, kjer je bil vrstilec UDK pravilno napovedan.

Pri vrednostih $rel(k)_i$ so z modro označene tiste vrednosti, kjer je funkcija zaradi pravilne napovedi vrnila vrednost 1. Ko uporabimo enačbo 2.14, dobimo za dokument A vrednost 0,75, za dokument B dobimo vrednost 0,5, za dokument C pa vrednost 0,667. Na koncu uporabimo enačbo 2.16 za izračun metrike MAP in dobimo vrednost 0,639.

		$P@k_i$ in $rel(k)_i$					$AP@5_i$
A	$P@k_A$	$\frac{1}{1}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{2}{4}$	$\frac{2}{5}$	$\frac{\frac{1}{1}+\frac{2}{4}}{n_{rel;i}} = \frac{1;5}{2} = 0;75$
	$rel(k)_A$	1	0	0	1	0	
B	$P@k_B$	$\frac{0}{1}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{\frac{1}{2}}{n_{rel;i}} = \frac{0;5}{1} = 0;5$
	$rel(k)_B$	0	1	0	0	0	
C	$P@k_C$	$\frac{1}{1}$	$\frac{2}{2}$	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{2}{5}$	$\frac{\frac{1}{1}+\frac{2}{2}}{n_{rel;i}} = \frac{2}{3} = 0;667$
	$rel(k)_C$	1	1	0	0	0	

$$MAP = \frac{1}{3}\ (0;75 + 0;5 + 0;667) = \underline{0;639}$$

Slika 2.31: Drugi del primera izračuna metrike MAP za tri sezname priporočil.

Metrika MRR

Metrika povprečni recipročni rang (ang. mean reciprocal rank) se označuje s kratico MRR in je ena izmed preprostejših metrik za vrednotenje priporočilnih sistemov, ki upošteva vrstni red priporočil. To metriko v literaturi zasledimo tudi pod nazivom povprečni recipročni rang zadetka (ang. average reciprocal hit rank) oziroma ARHR. Pri tej metriki je pomemben le prvi relevanten zadetek v seznamu priporočil, za katerega se izračuna recipročna vrednost ranga. Nad naborom večih seznamov priporočil nato izračunamo povprečje recipročnih vrednosti ranga, ki je končna vrednost metrike. Izračun vrednosti metrike je podan z enačbo 2.17, kjer je N število vseh seznamov priporočil, $rank_i$ pa je rang prvega relevantnega priporočila v i-tem seznamu priporočil.

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_i} \qquad (2.17)$$

Slika 2.32 prikazuje izračun metrike MRR za tri sezname priporočil. V seznamih priporočil so z zeleno barvo označeni najvišje rangirani pravilno napovedani vrstilci UDK, z modro pa vsi preostali nižje uvrščeni pravilno napovedani vrstilci UDK. Z rdečo barvo so označeni nepravilni vrstilci UDK.

Opazimo, da imajo vsi seznami priporočil pravilno napovedane vrstilce UDK, seznama priporočil za dokumenta A in C pa imata še druge pravilno napovedane vrstilce UDK, vendar ju metrika MRR ne upošteva. Za vsak seznam priporočil so podane vrednosti $rank_i$, ki predstavlja rang prvega pravilno napovedanega vrstilca UDK. Na koncu je podan še izračun metrike MRR, ki ga izračunamo kot povprečje vrednosti recipročnih vrednosti

Dokument	Pravilni vrstilci UDK	Seznam priporočenih vrstilcev UDK					$rank_i$
A	004.8 004.738	004.85	004	004.8	004.738	004.7	1
B	621.3	621	621.3	681	62	681.7	2
C	336.71 368 336.5	336.71	368	336.7	368.1	336.1	1

MRR = $\frac{3}{3}$ $\frac{1}{1}$ + $\frac{1}{2}$ + $\frac{1}{1}$ = 0;833

Slika 2.32: Primer izračuna metrike MRR za tri sezname priporočil.

ranga ($rank_i$). Metrika upošteva zgolj rang prvega pravilno napovedanega vrstilca UDK. Ker se metrika omejuje na prvi pravilno napovedan vrstilec UDK, je s tem pravzaprav opisana kvaliteta celotnega seznama priporočil, kar je omejitev te metrike.

Metrika NDCG@K

Metrike s področja informacijskega poizvedovanja, kot so na primer natančnost (ang. precision), priklic (ang. recall) in F1, so omejene v smislu vrednotenja vrstnega reda v seznamu priporočil. Za vrednotenje priporočilnih sistemov se zato pogosto uporablja normaliziran diskontiran kumulativni dobiček (ang. normalized discounted cumulative gain, NDCG), ki upošteva vrstni red generiranih priporočil. Tudi ta metrika se lahko prilagodi tako, da jo omejimo na prvih K zadetkov v seznamu. V tem primeru govorimo o metriki NDCG@K. Metrika NDCG deluje tako, da izračunamo diskontiran kumulativni dobiček (DCG) za obravnavan seznam priporočil (enačba 2.18), nato pa za isti seznam izračunamo idealni diskontiran kumulativni dobiček (IDCG). IDCG (enačba 2.19) je torej vrednost, ki bi jo dobili, če bi imeli seznam z idealnim vrstnim redom. Vrednost metrike NDCG izračunamo kot količnik med DCG in IDCG, kot je prikazano v enačbi 2.20.

$$DCG@K_i = \sum_{k=1}^{K} \frac{2^{rel(k)_i} \quad 1}{\log_2(k + 1)} \tag{2.18}$$

$$IDCG@K_i = \sum_{k=1}^{min(K;n_{rel;i})} \frac{2^{rel(k)_i} \quad 1}{\log_2(k + 1)} \tag{2.19}$$

$$NDCG@K_i = \frac{DCG@K_i}{DCG@K_i} \tag{2.20}$$

Pri izračunu vrednosti metrike NDCG@K se uporablja vrednost $rel(k)_i$, ki predstavlja relevantnost k-tega zadetka v i-tem seznamu in se izračuna enako kot pri metriki MAP (enačba 2.15). V enačbah 2.18 in 2.19 se za izračun DCG@K in IDCG@K uporablja različica metrike [116], ki v števcu uporablja izraz $2^{rel(k)_i}$ 1. Originalna oziroma tradicionalna različica [117] sicer uporablja v števcu zgolj $rel(k)_i$. Ta sprememba poudarja oceno relevantnih zadetkov. V primeru več dokumentov, izračunamo metriko NDCG@K za vsak dokument in izračunamo povprečno vrednost metrike (enačba 2.21).

$$NDCG@K = \frac{1}{N} \sum_{i=1}^{N} NDCG@K_i \tag{2.21}$$

V nadaljevanju je podan primer izračuna metrike NDCG@K za tri sezname priporočil pri K = 5. Na sliki 2.33 so ponovno podani dokumenti A, B in C s pripadajočimi pravilnimi vrstilci UDK in seznami priporočenih vrstilcev UDK.

Dokument	**Pravilni vrstilci UDK**	**Seznam priporočenih vrstilcev UDK**				
A	004.8 004.738	004.85	004	004.8	004.738	004.7
B	621.3	621	621.3	681	62	681.7
C	336.71 368 336.5	336.71	368	336.7	368.1	336.1

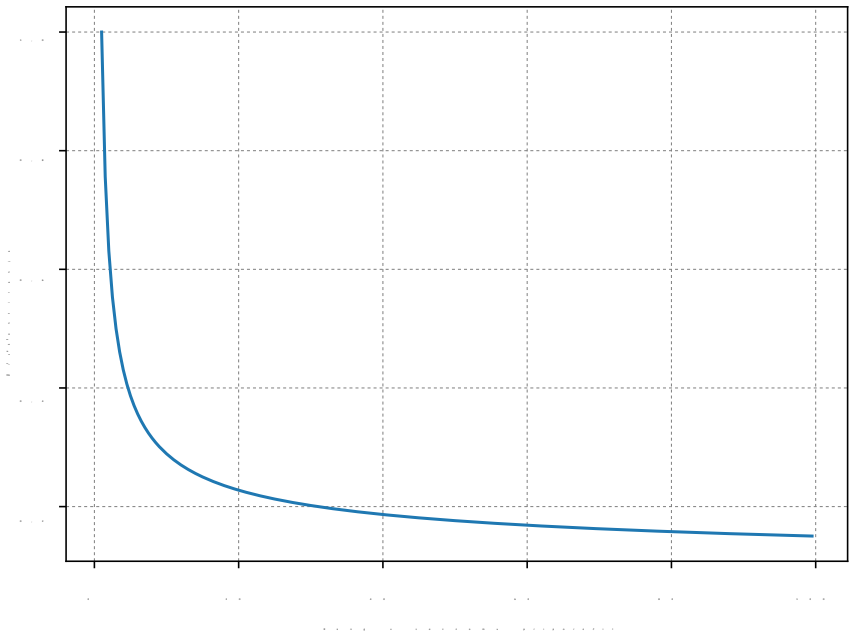
k	1	2	3	4	5
$\frac{1}{\log_2(k+1)}$	1	0,631	0,5	0,431	0,387

Slika 2.33: Prvi del primera izračuna metrike NDCG@5 za tri sezname priporočil.

Z zeleno barvo so označeni pravilno napovedani vrstilci UDK, z rdečo pa napačno napovedani vrstilci. Zatem so podane vrednosti uteži funkcije diskontiranja ($\frac{1}{\log_2(i+1)}$) za range k v seznamu priporočil. Graf funkcije diskontiranja je podan s sliko 2.34.

Na sliki 2.35 so podane vrednosti $rel(k)_i$ za izračun $DCG@5_i$ in $IDCG@5_i$. Z zeleno barvo so označene vrednosti, ki se upoštevajo pri $DCG@5_i$, z modro pa vrednosti, ki se upošte-vajo pri $IDCG@5_i$. V skrajno desnem stolpcu so podane izračunane vrednosti $DCG@5_i$ in $IDCG@5_i$ za dokumente A, B in C.

S sliko 2.36 so podane še izračunane vrednosti NDCG@5 za dokumente A, B in C. Za doku-ment A smo izračunali vrednost 0,877, za dokument B smo izračunali vrednost 0,631, za dokument C pa vrednost 1. Končna vrednost metrike NDCG@5, ki je povprečna vrednost izračunanih vrednosti NDCG@5 vsakega dokumenta (enačba 2.21), tako znaša 0,836.



Slika 2.34: Graf funkcije diskontiranja pri metriki NDCG@K.

							$DCG@5_i$ in $IDCG@5_i$
A	$rel(k)_A$	1	0	0	1	0	$DCG@5_A$ = 1;431
	$rel(k)_A$	1	1	0	0	0	$IDCG@5_A$ = 1;631
B	$rel(k)_B$	0	1	0	0	0	$DCG@5_B$ = 0;631
	$rel(k)_B$	1	0	0	0	0	$IDCG@5_B$ = 1
C	$rel(k)_C$	1	1	0	0	0	$DCG@5_C$ = 1;631
	$rel(k)_C$	1	1	0	0	0	$IDCG@5_C$ = 1;631

Slika 2.35: Drugi del primera izračuna metrike NDCG@5 za tri sezname priporočil.

$$NDCG@5_A = \frac{DCG@5_A}{IDCG@5_A} = \frac{1;431}{1;631} = 0;877$$

$$NDCG@5_B = \frac{DCG@5_B}{IDCG@5_B} = \frac{0;631}{1} = 0;631$$

$$NDCG@5_C = \frac{DCG@5_C}{IDCG@5_C} = \frac{1;631}{1;631} = 1$$

$$NDCG@5 = \frac{1}{3}(0;877 + 0;631 + 1) = 0;836$$

Slika 2.36: Tretji del primera izračuna metrike NDCG@5 za tri sezname priporočil.

Opisane metrike imajo različne lastnosti, ki so lahko koristne pri analizi in vrednotenju priporočilnih sistemov. Metrika HR@K je dober pokazatelj, ali priporočilni sistem sploh vrača relevantne vrstilce UDK v seznamu priporočil. Metrika MAP je uporabna, kadar želimo ugotoviti ali priporočilni sistem uvršča pravilne vrstilce UDK na začetek seznama priporočil. Z metriko MRR se bolj posvetimo vrednotenju prvega pravilnega vrstilca UDK v seznamu priporočil. Ta metrika je najbolj primerna za vrednotenje priporočilnih sistemov, ki stremijo k temu, da vrnejo najbolj relevaten zadetek na prvem mestu v seznamu. Po drugi strani metrika NDCG@K, zelo dobro ovrednoti vrstni red zadetkov v seznamu priporočil, hkrati pa omogoča vpeljavo lastne funkcije za ocenjevanje relevantnosti. V podrobnejši analizi [118] so ugotovili, da je metrika NDCG@K zmožna konsistentno določiti, kateri priporočilni sistemi so boljši.

2.4 Sorodna dela

Digitalne knjižnice imajo tradicionalno vlogo vmesnika med človekom in zbirko dokumentov. Iz tega razloga je za njih ključnega pomena, da omogočajo kvalitetno iskanje in brskanje po zbirki dokumentov z namenom, da človek čim hitreje najde vsebine, ki ga zanimajo. Sčasoma so tudi knjižničarji v sklopu procesa katalogizacije novih dokumentov začeli uporabljati vgrajene iskalnike v digitalnih knjižnicah, s katerimi je moč najti dokumente s podobno vsebino. Iskalniki so tako postali orodje, ki se uporablja pri procesu katalogizacije, čez čas pa so se pojavile tudi prve ideje o uporabi metod strojnega učenja.

Raziskovalci po svetu so problem klasifikacije dokumentov s knjižničnimi klasifikacijskimi sistemi načeloma reševali z avtomatsko klasifikacijo dokumentov [3, 4, 6, 7]. V njihovih pristopih se dokumenti najprej obdelajo z metodami obdelave naravnega jezika [5, 8]. S tem se pridobijo značilke o dokumentih, ki se nato uporabijo v različnih metodah strojnega učenja [9]. Slednje izvajajo večrazredno klasifikacijo, ki vrne ustrezen vrstilec izbranega knjižničnega klasifikacijskega sistema. Po pregledu literature smo se osredotočili na štiri najbolj sorodna dela, ki se ukvarjajo s problemom avtomatske klasifikacije dokumen-

tov v univerzalno decimalno klasifikacijo in Deweyjevo decimalno klasifikacijo. Ti pristopi so podrobneje opisani v nadaljevanju.

Kragelj in Kljajić Borštnar v svojem delu [11] predstavljata postopek avtomatske klasifikacije starejših neklasificiranih dokumentov v UDK. Avtorja sta ugotovila, da je s postopki strojnega učenja dokumentom možno določiti enostavne vrstilce UDK. Kot najboljšo metodo strojnega učenja sta izpostavila metodo podpornih vektorjev. Za avtomatsko klasifikacijo uporabljata večrazredni klasifikator, kar pomeni, da je rezultat klasifikacije en vrstilec UDK. Pri tem ni bilo omejitev na nabor vrstilcev in znanstvenih področij v UDK. Sorodno delo uporablja dokumente v slovenskem jeziku, za katere se pridobijo značilke z uporabo utežne sheme tf-idf.

Nevzorova in Almukhametov se svoji raziskavi [119] ukvarjata s problemom avtomatske klasifikacije UDK za dokumente v ruskem jeziku z znanstvenega področja matematike. Njuna rešitev je torej domensko specifična in jezikovno omejena. Klasifikacijo izvajata z uporabo ontologije OntoMathPro, ki podpira ruski in angleški jezik. Rezultat klasifikacije dokumenta v njunem pristopu je en vrstilec UDK, za vrednotenje uspešnosti pa sta zasnovala lastno metriko na podlagi mehke logike. Sorodno delo uporablja ročno izbrane matematične publikacije v ruskem jeziku.

Golub in drugi so v študiji [10] predstavili primerjavo pristopov avtomatske klasifikacije v Deweyjevo decimalno klasifikacijo (DDK) nad dokumenti v švedskem jeziku. Sorodno delo ne uporablja knjižničnega klasifikacijskega sistema UDK, vendar je zaradi podobno-sti UDK z DDK kljub temu relevantno. Ugotovili so, da so rezultati avtomatske klasifikacije najboljši, če uporabijo metodo podpornih vektorjev. Njihove ugotovitve so zelo podobne ugotovitvam v [11], kar nakazuje na dejstvo, da so pristopi za avtomatsko klasifikacijo UDK in DDK sorodni. Rezultat klasifikacije dokumenta v tem pristopu je en vrstilec DDK, značilke dokumentov pa so pridobljene z uporabo utežne sheme tf-idf. V raziskavi so izpostavili, da je omejitev rezultata na en vrstilec DDK lahko problematična in da je potrebno nadaljnje raziskovalno delo, ki bi omogočilo določanje več relevantnih vrstilcev DDK.

Schrumpf in drugi so v svojem delu [12] predstavili pristop avtomatske klasifikacije v DDK z globokimi nevronskimi mrežami, ki temeljijo na arhitekturi transformer. Uporabili so globoko nevronsko mrežo BERT [14], ki so jo prilagodili s prenosnim načinom učenja, pri tem pa so uporabljali akademske dokumente v nemškem jeziku. Z rezultati so pokazali izboljšavo v primerjavi z metodo podpornih vektorjev. To sorodno delo je tudi prvo, pri katerem so za avtomatsko klasifikacijo v knjižnični klasifikacijski sistem DDK uporabili globoke nevronske mreže. Rezultat klasifikacije dokumenta je ponovno en vrstilec DDK, av-

torji pa so se prav tako omejili na klasifikacijo do nivojev, ki so dolgi največ štiri znake. V UDK se takšna omejitev odraža kot upoštevanje vrhnjega področja in prvih podpodročij, ki razširjajo vrhnje področje.

V sorodnem delu [10] so avtorji izpostavili izzive avtomatske klasifikacije dokumentov v knjižnične klasifikacijske sisteme. Navedli so, da je uporaba priporočilnih sistemov kot del polavtomatske klasifikacije dokumentov smiseln pristop, saj dopolnjuje proces katalogizacije in ga bistveno ne spreminja. Ideja je torej, da bi priporočilni sistem deloval kot podporni sistem, pri tem pa se knjižničar sam odloči, ali bo upošteval njegove rezultate ali ne. Iz tega izvira motivacija za naše delo, ki temelji na hibridnem priporočilnem sistemu. Izziv, ki ga naslavljamo v disertaciji, je upoštevanje celotne hierarhije knjižničnega klasifikacijskega sistema UDK brez omejitev na specifična področja. Dodatno s predlaganim pristopom vpeljujemo vračanje rezultata v obliki urejenega seznama večih vrstilcev UDK namesto omejitve rezultata na samo enega. Ostala sorodna dela opisujejo različne zasnove hibridnih priporočilnih sistemov [26–28]. Ti priporočilni sistemi se tako ukvarjajo s priporočanjem filmov [22, 23, 33, 120], restavracij [121], knjig, [25, 31], dokumentov [21, 122], novic [24] in glasbe [30, 32, 123]. V teh hibridnih priporočilnih sistemih je najpogostejša kombinacija sodelovalnega in vsebinskega filtriranja [101, 124]. V omenjenih sorodnih delih so se pogosto odločili za utežno, mešano in preklopno hibridizacijo. Večina teh sorodnih del uporablja globoke nevronske mreže BERT in besedilo preoblikujejo v kontekstne vložitve, nekatera pa uporabljajo tudi različne značilke, ki jih pridobijo iz besedila z utežno shemo $tf\text{-}idf$ [35, 125] ali z uporabo rangirne funkcije BM25 [36–39, 126, 127].

3 Hibridni pristop za priporočanje vrstilcev UDK

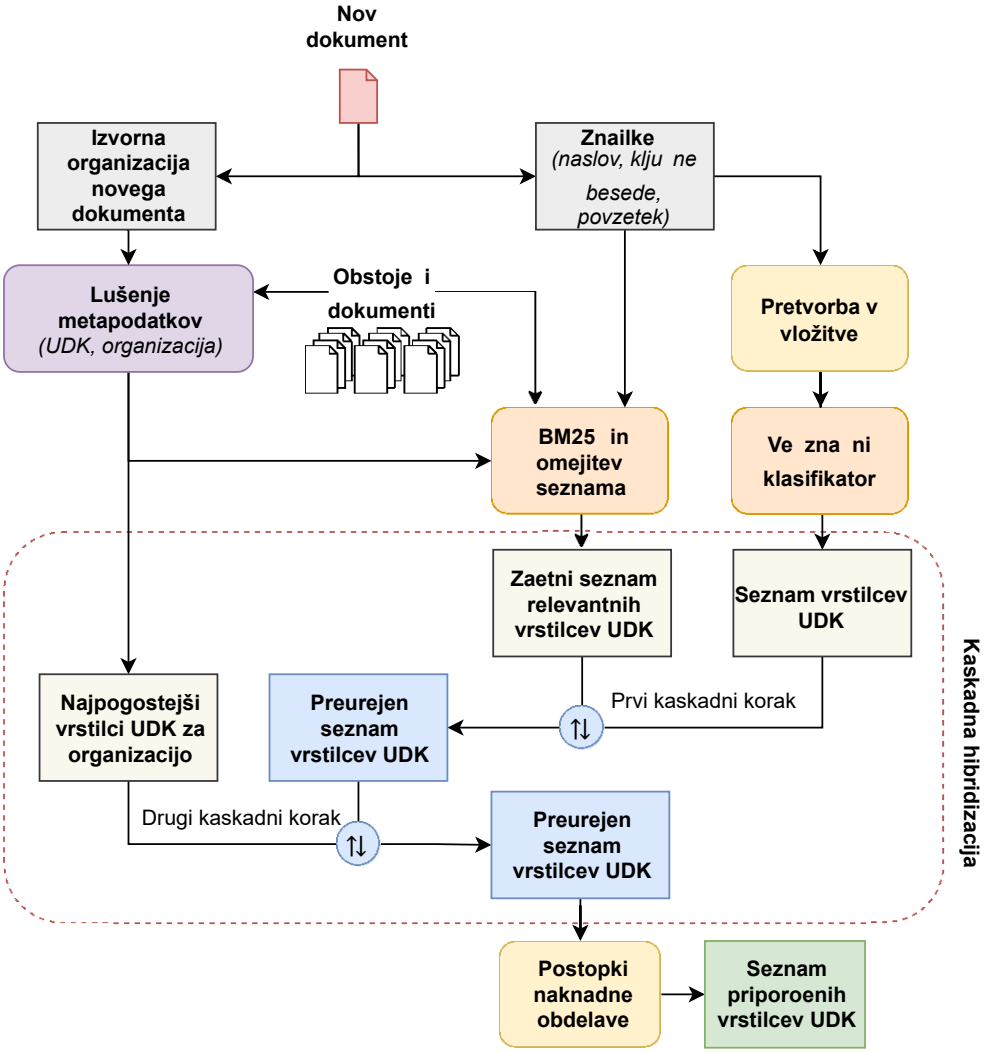
V tem poglavju je podrobneje predstavljen predlagan hibridni pristop za priporočanje vrstilcev UDK. Najprej opišemo strukturo predlaganega priporočilnega sistema, zatem pa so podrobneje opisane posamezne komponente priporočilnega sistema. Te vključujejo pridobivanje začetnega seznama relevantnih vrstilcev UDK, večznačni klasifikator, kaskadno hibridizacijo in naknadno obdelavo priporočil.

3.1 Struktura hibridnega priporočilnega sistema

Ideja predlaganega hibridnega priporočilnega sistema je, da najprej na podlagi obstoječih indeksiranih dokumentov in njihovih metapodatkov tvorimo začetni seznam relevantnih vrstilcev UDK za nov dokument, nato pa ga uporabimo kot osnovo za naslednje korake v delovnem toku, ki jih izvajajo posamezne komponente priporočilnega sistema. Pri tem so posamezne komponente povezane v kaskadno hibridizacijo, ki preuredi začetni seznam relevantnih vrstilcev UDK. Na koncu se izvedejo še postopki naknadne obdelave, ki tvorijo izhodni seznam priporočenih vrstilcev UDK. Delovni tok predlaganega hibridnega pristopa za priporočanje vrstilcev UDK temelji na več komponentah (slika 3.1).

Pri pridobivanju začetnega seznama relevantnih vrstilcev UDK se uporabijo obstoječi dokumenti, metapodatki dokumentov, rangirna funkcija BM25 in algoritem za omejitev rezultata rangirne funkcije BM25. Obstoječi dokumenti so indeksirani na podlagi naslovov, ključnih besed in povzetkov. Ločeno so shranjeni metapodatki, ki vsebujejo informacijo o vrstilcih UDK in izvornih organizacijah dokumentov. Vhod v rangirno funkcijo BM25 so značilke novega dokumenta v obliki združenega besedila naslova, ključnih besed in povzetka. Rangirna funkcija BM25 vrne urejen seznam vsebinsko podobnih dokumentov skupaj z njihovimi metapodatki. Na ta način dobimo začetni seznam vrstilcev UDK, ki so relevantni za novi dokument.

Del predlaganega hibridnega priporočilnega sistema je večznačni klasifikator, ki za podano vhodno besedilo vrne nabor vrstilcev UDK, ki so kontekstno ustrezni glede na vsebino vhodnega besedila. Večznačni klasifikator uporablja prilagojen jezikovni model BERT, ki je prednaučen na učni množici že katalogiziranih dokumentov, ki jih uporablja tudi ran-girna funkcija BM25 za pridobivanje začetnega seznama vrstilcev UDK. Gre za večznačno klasifikacijo, ki lahko vrne tudi več vrstilcev UDK z različnih globin v isti veji hierarhije UDK.



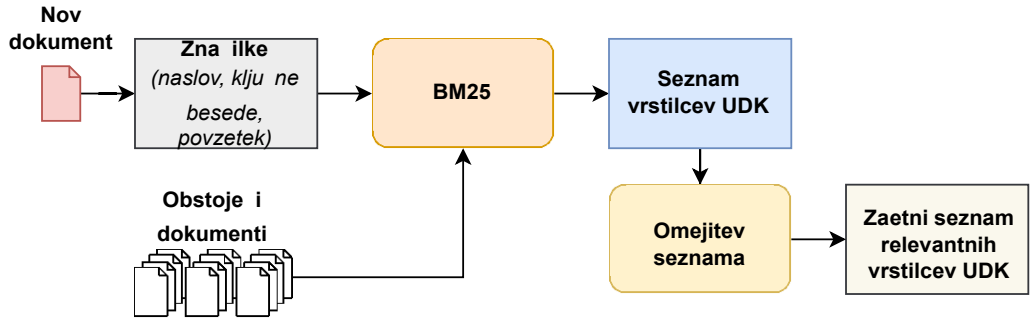
Slika 3.1: Arhitektura predlaganega hibridnega pristopa za priporočanje vrstilcev UDK.

Po izvedbi pridobivanja začetnega seznama relevantnih vrstilcev UDK in večznačnega klasifikatorja sta na voljo dva seznama vrstilcev UDK. Tvori se še tretji seznam vrstilcev UDK, ki vsebuje najbolj značilne vrstilce UDK za obstoječe dokumente z isto izvorno organiza-cijo kot nov dokument. Na tej točki se prične postopek kaskadne hibridizacije. Pristop kaskadne hibridizacije je bil izbran zaradi enostavne razširitve z vključitvijo drugih metod priporočanja v zaporednem delovnem toku. V splošnem se kaskadna hibridizacija smatra kot učinkovit, fleksibilen, robusten in skalabilen pristop za realizacijo hibridnih priporočilnih sistemov [26, 28]. V predlaganem pristopu kaskadno hibridizacijo izvajamo v dveh kaskadnih korakih. V prvi kaskadni korak vstopata začetni seznam relevantnih vrstilcev UDK in seznam, ki ga tvori večznačni klasifikator. Izvede se preurejanje začetnega seznama, rezultat pa je nov preurejen seznam vrstilcev UDK. Sledi drugi kaskadni korak, v

katerega vstopata preurejen seznam vrstilcev UDK in seznam najbolj značilnih vrstilcev UDK za izvorno organizacijo novega dokumenta, pri čemer se preuredi izhodni seznam prvega kaskadnega koraka. Na koncu se izvedejo še postopki naknadne obdelave priporočil, preden se rezultat prikaže končnemu uporabniku.

3.2 Pridobivanje začetnega seznama relevantnih vrstilcev UDK

V delovnem toku predlaganega hibridnega priporočilnega sistema najprej pridobimo začetni seznam relevantnih vrstilcev UDK. Pri priporočanju vrstilcev UDK za nov dokument uporabimo že katalogizirane sorodne dokumente. Ideja je, da za nov dokument najdemo podobne dokumente in pridobimo njihove vrstilce UDK, ki bodo služili kot osnova za določanje vrstilcev UDK novemu dokumentu (slika 3.2). Značilke novega dokumenta, ki so v obliki združenega besedila naslova, ključnih besed in povzetka, uporabimo kot vhodni iskalni niz v rangirno funkcijo BM25. Ta vrne po oceni BM25 padajoče urejen seznam, ki vsebuje 25 najbolj podobnih dokumentov z njihovimi metapodatki, iz katerih izluščimo in ohranimo samo vrstilce UDK. V primeru večkratnih pojavitev vrstilca UDK, zanj upoštevamo višjo oceno BM25 pripadajočega dokumenta.



Slika 3.2: Pridobivanje seznama relevantnih vrstilcev UDK z rangirno funkcijo BM25.

Z vrstilci UDK so določena področja znanosti dokumentov. Ker za nekatera področja znanosti obstaja več dokumentov kot za druga, se na podlagi števila dokumentov določi dolžina seznama relevantnih vrstilcev UDK. Ločeno od postopka pridobivanja začetnega seznama relevantnih vrstilcev UDK se določijo dolžine seznamov za vrhnja področja hierarhije UDK (algoritem 3.1), pri čemer vrhnje področje pomeni prvi znak vrstilca UDK (npr. za področje 004 je vrhnje področje 0). Za dobro zastopana področja znanosti določimo daljši seznam, za manj zastopana področja znanosti pa krajšega. To pripomore k zajema-nju relevantnih vrstilcev UDK manj zastopanih področij znanosti.

Z algoritmom 3.1, ki se izvede vnaprej oziroma ob spremembi v bazi dokumentov, se najprej izračunajo deleži dokumentov, ki pripadajo posameznim vrhnjim področjem. Nato se

Algoritem 3.1: Določanje dolžine seznamov za vrhnja področja hierarhije UDK.

Vhod: Asociativno polje S, kjer je ključ t vrhnje področje hierarhije UDK, vrednost S[t] pa število dokumentov v korpusu z vrhnjim področjem t

Izhod: Asociativno polje L, kjer je ključ t vrhnje področje hierarhije UDK, vrednost L[t] pa dolžina seznama priporočil za vrhnje področje t.

```
initialize P  [ ]
foreach t in S do
    P[t]  delež dokumentov, ki so v vrhnjem področju t
end
initialize L  [ ]
initialize k  [5; 10; 15; 20; 25]                    . dolžine seznama
initialize b^min  [0; 16; 31; 61; 86]                . spodnje meje za deleže
initialize b^max  [15; 30; 60; 85; 100]              . zgornje meje za deleže
foreach t in P do
    for i  1 to |k| do
        if b_i^min  P[t]  b^max_i then
            L[t]  k_i      . določi dolžino seznama za vrhnje področje t
            break
        end
    end
end
return L
```

vnaprej določijo možne dolžine seznamov, ki so lahko dolgi 5, 10, 15, 20 ali 25 elementov. Določijo se tudi spodnje (b^{min}) in zgornje (b^{max}) meje za vrednosti deležev dokumentov, ki se uporabijo pri določanju dolžine seznama za vrhnje področje t. Na koncu se na pod-lagi deleža dokumentov pripadajočega vrhnjega področja določijo dolžine seznamov za vrhnja področja t, ki se vrnejo v obliki asociativnega polja L. Z določenimi dolžinami seznamov za posamezna vrhnja področja se izvede omejitev vhodnega seznama U, ki smo ga tvorili z rangirno funkcijo BM25 (algoritem 3.2). V seznamu U so vrstilci UDK urejeni po padajoči vrednosti BM25. Vrstilci UDK v vhodnem seznamu se najprej reducirajo na vrhnja področja in shranijo v pomožni seznam U^0, nato pa se v njem poišče najbolj pogosto vrhnje področje t_{max}. V primeru več vrhnjih področij z enako pogostostjo pojavitve se izbere tisto vrhnje področje, ki je del višje uvrščenega vrstilca UDK v vhodnem seznamu U. Na podlagi izbranega vrhnjega področja t_{max} se z vpogledom v vhodno asociativno polje L določi dolžina seznama . Rezultat je omejen seznam vrstilcev UDK U, ki se pridobi z omejitvijo vhodnega seznama U na elementov.

Algoritem 3.2: Omejitev seznama glede na najpogostejše vrhnje področje v začetnem seznamu priporočil.

Vhod: Urejen seznam vrstilcev UDK U in asociativno polje L z dolžinami seznamov za vrhnja področja t

Izhod: Omejen seznam vrstilcev UDK U

```
U^0      seznam vrstilcev UDK U, reduciran na vrhnja področja
t_max    najbolj pogosto vrhnje področje v U^0
         L[t_max]  . določi dolžino z vpogledom v asociativno polje L
U        omejen seznam U z elementi
return U
```

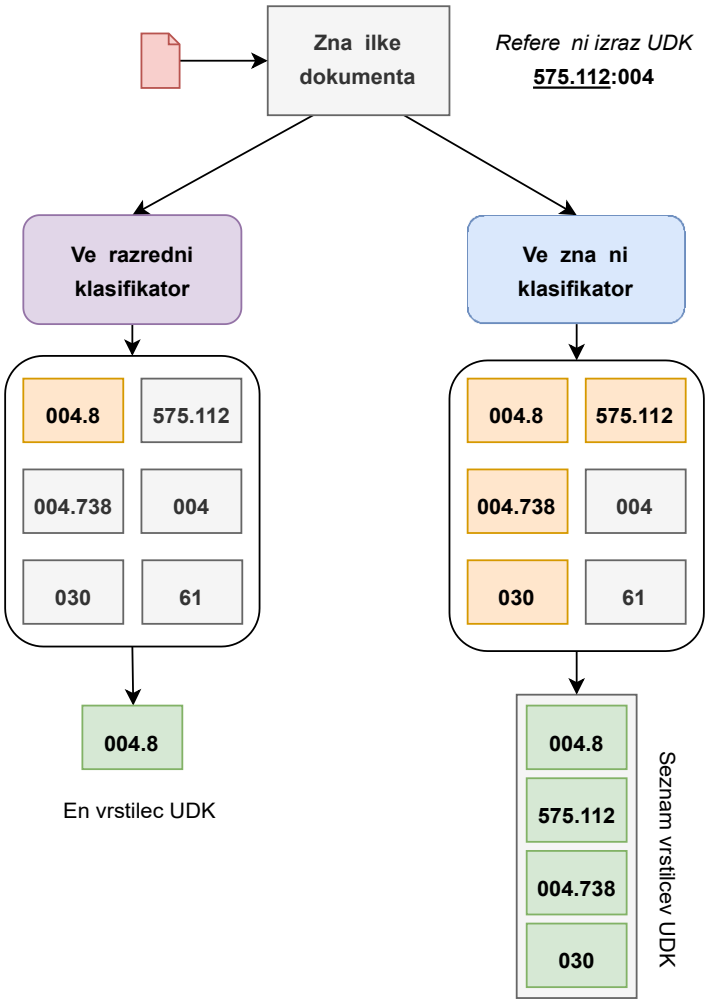
Po izvedbi omejitve seznama dobimo seznam vrstilcev UDK, ki je urejen padajoče po ocenah BM25 in bi ga v enostavnejši izvedbi priporočilnega sistema lahko že uporabili kot končni seznam vrstilcev UDK za prikaz uporabniku. V tem primeru gre za metodo vsebinskega filtriranja z uporabo rangirne funkcije BM25. Dobljen seznam označimo z R_0, pripadajoče ocene BM25 za posamezne vrstilce UDK v seznamu pa $s_r^{(R_0)}$. V predlaganem hibridnem pristopu za priporočanje vrstilcev UDK ta seznam predstavlja začetni seznam relevantnih vrstilcev UDK, ki se bo še spreminjal skozi proces kaskadne hibridizacije. Za začetek procesa kaskadne hibridizacije v predlaganem hibridnem pristopu uporabimo še seznam vrstilcev UDK, ki je rezultat večznačnega klasifikatorja.

3.3 Večznačni klasifikator

Večznačni klasifikator je potrebno za uporabo v predlaganem priporočilnem sistemu predhodno naučiti. Pri tem uporabimo prednaučen jezikovni model BERT, ki ga priučimo na nalogo večznačne klasifikacije vrstilcev UDK. Za razliko od sorodnih del, v katerih so avtorji izvajali večrazredno klasifikacijo, predlagani klasifikator izvaja večznačno klasifikacijo. To je pomembna razlika, saj se v scenariju večrazredne klasifikacije vhodno besedilo klasificira v enega izmed izbranih razredov. Izhod v tem pristopu je torej en vrstilec UDK. V primeru večznačne klasifikacije pa klasifikator učimo tako, da je zmožen klasificirati vhodno besedilo v več oznak, pri čemer so oznake vrstilci UDK. Vhod v klasifikator je združeno besedilo naslova, ključnih besed in povzetka dokumenta, izhod klasifikatorja pa je seznam vrstilcev UDK.

Ta pristop je tudi bolj naraven za uporabo znotraj priporočilnih sistemov, saj imamo tukaj opravka s seznami priporočenih elementov. To velja tudi za priporočanje vrstilcev UDK. Iz primerov ročno določenih izrazov UDK namreč vidimo, da se v njih pojavlja več različ-

nih vrstilcev UDK. To je še posebej razvidno na primerih ročno določenih izrazov UDK za dokumente z interdisciplinarno vsebino. Iz tega razloga smo preoblikovali problem klasifikacije iz večrazredne v večznačno. Primer takšne situacije prikazuje slika 3.3, kjer z večrazrednim in večznačnim klasifikatorjem poskušamo določiti smiselne vrstilce UDK.



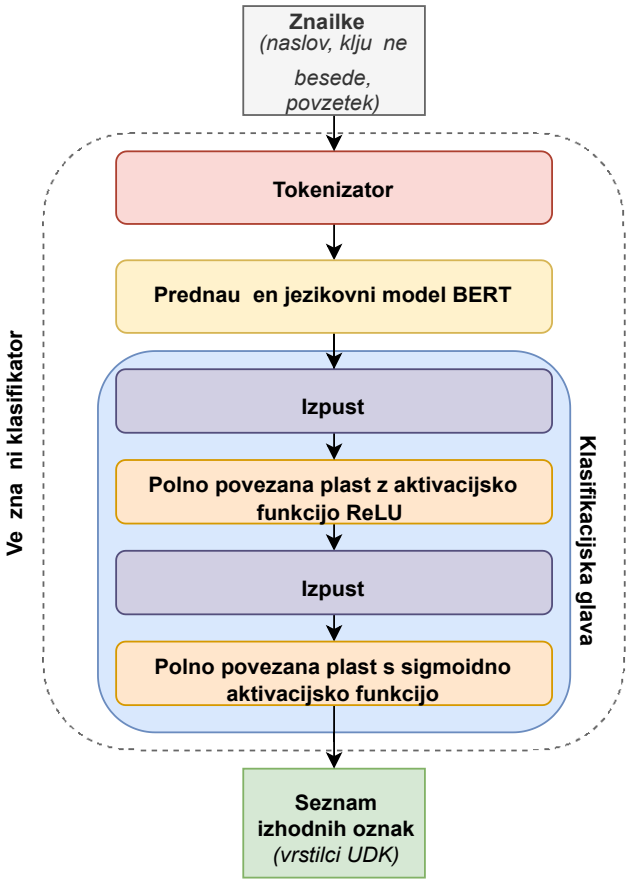
Slika 3.3: Primer določanja smiselnih vrstilcev UDK z večrazrednim (levo) in večznačnim (desno) klasifikatorjem za dokument z interdisciplinarno vsebino.

V primeru je referenčni izraz UDK obravnavanega dokumenta enak 575.112:004, kar ga uvršča na področji bioinformatike (vrstilec 575.112) in računalništva (vrstilec 004). Dokument govori o izdelavi spletne aplikacije za odkrivanje trendov v biomedicinski znanstveni literaturi. Večrazredni klasifikator je v tem primeru izbiral med več vrstilci UDK in izbral vrstilec 004.8, ki predstavlja področje umetne inteligence. Podobnost z referenčnim izrazom je zgolj polovična, saj je v rezultatu zajeto samo področje računalništva (vrstilec 004). Večznačni klasifikator je za isti primer izbiral med več vrstilci UDK in izbral vrstilce 004.8, 575.112, 004.738 in 030.

3.3.1 Arhitektura večznačnega klasifikatorja

Ideja za uporabo predlaganega klasifikatorja je v tem, da med postopkom priporočanja pripomore pri preurejanju vrstilcev UDK na podlagi hierarhije UDK z določeno mero specifičnosti. Na ta način se v končni seznam priporočenih vrstilcev UDK uvrstijo bolj specifični vrstilci UDK, ki bolje posnemajo ročno določanje vrstilcev UDK in s tem izboljšajo proces katalogizacije dokumentov.

Struktura predlaganega večznačnega klasifikatorja je prikazana na sliki 3.4. Pri učenju in uporabi klasifikatorja se na začetku vhodno besedilo s pomočjo tokenizatorja razčleni, zatem pa se členi pretvorijo v kontekstne vložitve v prvi plasti prednaučenega jezikovnega modela BERT. Kontekstne vložitve gredo skozi preostale plasti prednaučenega jezikovnega modela BERT. Kot prednaučen jezikovni model BERT smo uporabili model SloBERTa, ki je bil naučen na besedilih v slovenskem jeziku. Za temi plastmi sledijo plasti, ki predstavljajo klasifikacijsko glavo.



Slika 3.4: Arhitektura večznačnega klasifikatorja.

Najprej nastopi plast izpusta (ang. dropout layer), zatem pa polno povezana plast nevronov, ki uporablja aktivacijsko funkcijo ReLU. Nato ponovno sledi plast izpusta, za njo

pa še ena polno povezana plast nevronov, ki uporablja sigmoidno aktivacijsko funkcijo. Parameter verjetnosti izpusta p je v obeh primerih nastavljen na p = 0;5. Na koncu (tj. po prehodu skozi sigmoidno aktivacijsko funkcijo) se vse izhodne oznake (vrstilci UDK) z vrednostjo nad pragom = 0;5 shranijo v seznam in uredijo v padajočem vrstnem redu.

3.3.2 Glajenje oznak

Značilke dokumentov so bile sestavljene iz naslovov, ključnih besed in povzetkov, za vsak dokument pa je bil znan ročno določen izraz UDK, ki smo ga z razpoznavalnikom razdelili na vrstilce UDK. Primer takšnega razpoznavanja izraza UDK in njegove delitve na vrstilce UDK prikazuje slika 3.5, iz katere je razvidno, da se privesni vrstilci ignorirajo. Prav tako je s podčrtanim besedilom označen specifičen vrstilec, ki je zaradi uporabe omenjenega prostodostopnega nabora vrstilcev UDK [45] bil razpoznan na višjem nivoju.

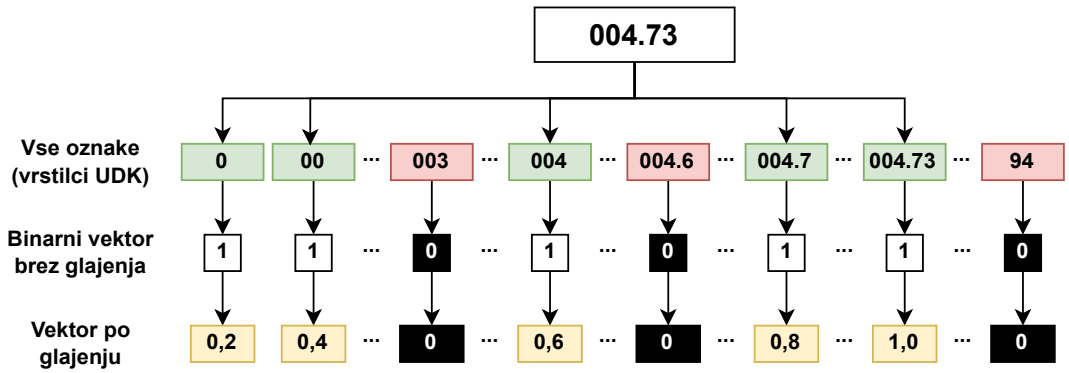
Vhod	**Izhod**
[004.94:621.952.8]+658.8(043.2)	004.94 621.9 658.8
711.4:711.1:158.937:003.63(497.4Koper)(043.2)	711.4 711.1 158.937 003.63

Slika 3.5: Primer razpoznavanja izraza UDK in njegove delitve na vrstilce UDK.

Po obdelavi značilk dokumentov z razpoznavalnikom ustvarimo binarni vektor dimenzije vseh oznak, ki v našem primeru predstavljajo vse možne vrstilce UDK na vseh nivojih hierarhije UDK. Vrednost binarnega vektorja je za posamezno oznako enaka 1, kjer je vrstilec UDK za dokument uspešno razpoznan z razpoznavalnikom. Pri tem se upošteva tudi hierarhija vrstilca. Na primer, za vrstilec 004 so enice v binarnem vektorju dodeljene tudi za vrstilca 0 in 00, saj ta dva vrstilca predstavljata višja nivoja določenega vrstilca 004 v hierarhiji UDK. Vrednost binarnega vektorja je enaka 0, kjer vrstilec UDK za dokument ni razpoznan z razpoznavalnikom. Binarni vektorji z določenimi vrednostmi za posamezno oznako pri učenju določajo pravilno stanje.

Kadar so vrstilci UDK zelo specifični, je posledično njihova dolžina daljša, saj se tako nahajajo globlje v hierarhiji UDK. V takšnih primerih so vrstilci UDK, ki predstavljajo vrhnja področja hierarhije UDK, lahko preveč splošni, da bi jih bilo smiselno uporabiti kot ciljne vrednosti za večznačno klasifikacijo. Takšne primere razpoznamo na podlagi dolžine vrstilca UDK in priredimo pripadajoče vrednosti vsem vrstilcem, ki so njegovi predniki v

hierarhiji UDK. Pripadajoča vrednost vrstilca UDK na posamezni globini v hierarhiji UDK se izračuna kot razmerje med trenutno globino vrstilca UDK in največjo globino vrstilca UDK. Primer glajenja oznak (vrstilcev UDK) prikazuje slika 3.6. V primeru obravnavamo vrstilec 004.73. Z zeleno barvo so označeni vsi vrstilci v hierarhiji UDK, ki so bili razpoznani kot del obravnavanega vrstilca. Z rdečo barvo so označeni vrstilci, ki niso bili razpoznani kot del obravnavanega vrstilca. V sredini je podan binarni vektor brez uporabe glajenja, kjer vidimo, da se razpoznani vrstilci UDK preslikajo v enice, nerazpoznani pa v ničle. Nižje je podan vektor po glajenju z vrednostmi uteži, ki so bile določene glede na hierarhično globino vrstilca 004.73.



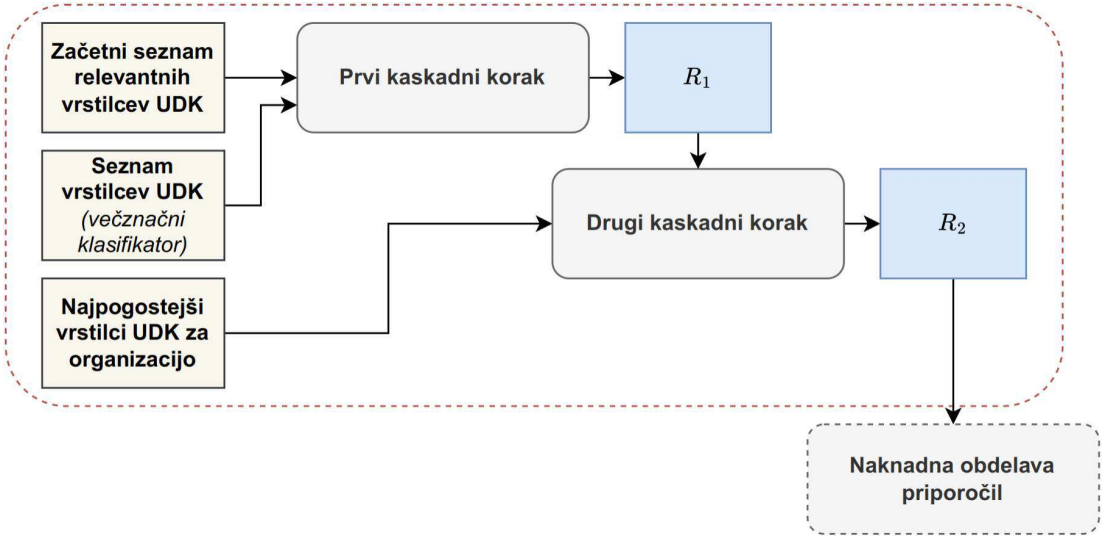
Slika 3.6: Primer glajenja oznak (vrstilcev UDK).

Vrednosti vektorja po glajenju odražajo hierarhijo vrstilca UDK. V primerjavi z binarno predstavitvijo pripadnosti, vrednosti vektorja po glajenju bolj natančno določajo specifično vejo hierarhije UDK, v katero spada posamezen vrstilec UDK. Pri učenju modela lahko uporabimo kar vektor z vrednostmi po glajenju, ali pa ga z določanjem pragovne vrednosti preslikamo v nov binarni vektor. Tak vektor ne predstavlja več tistih vrstilcev v hierarhiji UDK, ki so bližje vrhnjim področjem. Naučen večznačni klasifikator v predlaganem hibridnem pristopu priporočanja uporabimo v prvem koraku kaskadne hibridizacije.

3.4 Kaskadna hibridizacija

Vsak postopek kaskadne hibridizacije lahko opišemo v več korakih, saj se procesi znotraj nje izvajajo zaporedno. V predlaganem pristopu se kaskadna hibridizacija izvede z dvema zaporednima kaskadnima korakoma (slika 3.7), ki se izvajata nad seznami relevantnih vrstilcev UDK.

Namen prvega kaskadnega koraka je bolje razvrstiti relevantne vrstilce UDK iz začetnega seznama. Razvrščanje v tem koraku temelji na podlagi korpusa dokumentov z ročno označenimi vrstilci UDK in se izvede z večznačnim klasifikatorjem. Namen drugega kaskadnega

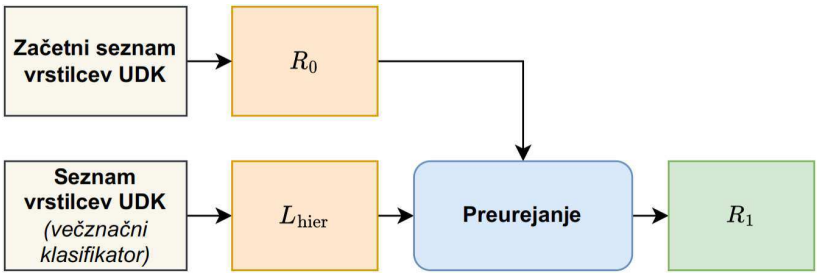


Slika 3.7: Postopek kaskadne hibridizacije razdeljene na dva kaskadna koraka.

koraka je izboljšati oceno relevantnim vrstilcem UDK na podlagi najpogostejših vrstilcev UDK dokumentov, ki izvirajo iz iste organizacije kot dokument, za katerega želimo priporočiti vrstilce UDK. Gre torej za kombinacijo uporabe kontekstnih značilk dokumentov v prvem koraku in metapodatkov v drugem koraku.

3.4.1 Prvi kaskadni korak

Vhod v prvi kaskadni korak (slika 3.8) je začetni seznam relevantnih vrstilcev UDK R_0, ki je urejen po padajoči vrednosti$_r^{(R_0)}$. Značilke dokumenta, za katerega tvorimo priporočila, se najprej obdelajo s tokenizatorjem. Členi, pridobljeni s to obdelavo, predstavljajo vhod v večznačni klasifikator, ki vrne seznam oznak L_{hier}, ta pa se uporabi za preurejanje seznama R_0.



Slika 3.8: Delovanje prvega kaskadnega koraka.

Ocene posameznih vrstilcev UDK v R_0, ki so podobni oznakam v L_{hier}, se ponovno izračunajo z enačbo 3.1, in tvori se nov urejen seznam R_1.

$$(R_1)_r = \sum_{i=0}^{|L_{hier}|} (R_0)_r (1 + sim_{jw}(r; l_i)) \quad (3.1)$$

V enačbi 3.1 je $(R_1)_r$ nova ocena vrstilca UDK r, $(R_0)_r$ pa predstavlja njegovo staro oceno. Vrednost $sim_{jw}(r; l_i)$ je Jaro-Winklerjeva podobnost [128] med vrstilcem UDK r iz R_0 in oznako l_i iz L_{hier}, ki se za poljubno dolga niza znakov a in b izračuna z enačbo za Jarovo podobnost:

$$sim_j(a; b) = \begin{cases} 0 & , \text{če je } = 0 \\ \frac{1}{3}\left(\frac{}{|a|} + \frac{}{|b|} + \frac{}{}\right) & , \text{sicer} \end{cases} \quad (3.2)$$

kjer je število ujemanj znakov, število transpozicij znakov, |a| in |b| pa sta dolžini nizov a in b. Jaro-Winklerjeva podobnost se nato izračuna z enačbo:

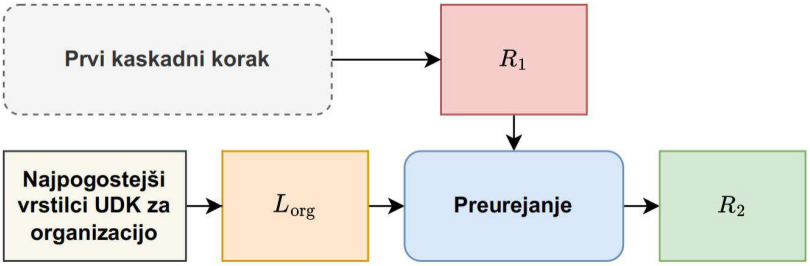
$$sim_{jw}(a; b) = sim_j(a; b) + \ell p(1 - sim_j(a; b)) \quad (3.3)$$

kjer je ℓ dolžina predpone na začetku niza, p pa skalirni faktor za spremembo dodatka h končni vrednosti razdalje. Standardni vrednosti za parametra sta $\ell = 3$ in $p = 0{;}1$ [128]. Vrednost Jaro-Winklerjeve razdalje je na intervalu $[0; 1]$, kjer 0 pomeni popolno različnost, 1 pa pomeni popolno ujemanje. Vrednost Jaro-Winklerjeve podobnosti je višja za nize znakov, ki se ujemajo od začetka (z leve proti desni), kar ustreza naši nalogi primerjave vrstilcev UDK po globini hierarhije UDK. Izhod prvega kaskadnega koraka je torej preurejen seznam vrstilcev UDK in njihovih pripadajočih ocen (R_1) v padajočem vrstnem redu, glede na $(R_1)_r$.

3.4.2 Drugi kaskadni korak

Vhod v drugi kaskadni korak (slika 3.9) je urejen seznam vrstilcev UDK R_1, ki je rezultat prvega kaskadnega koraka. Na podlagi metapodatka o izvorni organizaciji dokumenta, za katerega želimo priporočiti vrstilce UDK, se pridobi vnaprej pripravljen seznam najbolj pogostih vrstilcev UDK za ugotovljeno izvorno organizacijo dokumenta (L_{org}). Ta seznam se nato uporabi za preurejanje seznama R_1.

Ocene posameznih vrstilcev UDK v R_1 se ponovno izračunajo z enačbo 3.4, kjer se tvori nov urejen seznam R_2.



Slika 3.9: Delovanje drugega kaskadnega koraka.

$$r^{(R_2)} = r^{(R_1)} + \frac{org}{L_{max}} \quad (3.4)$$

V enačbi 3.4 je $r^{(R_2)}$ nova ocena vrstilca UDK r, $r^{(R_1)}$ pa predstavlja staro oceno vrstilca UDK. Člen org predstavlja število pojavitev vrstilca UDK pri dokumentih, ki izvirajo iz organizacije L_{org}, L_{max} pa predstavlja največje število pojavitev katerega od vrstilcev pri dokumentih iz L_{org}. Preurejanje vodi v nov urejen seznam R_2, ki je urejen po $r^{(R_2)}$ (ocenah vrstilcev UDK) v padajočem vrstnem redu in predstavlja izhod drugega kaskadnega koraka. S tem korakom se postopek kaskadne hibridizacije zaključi.

3.5 Naknadna obdelava priporočil

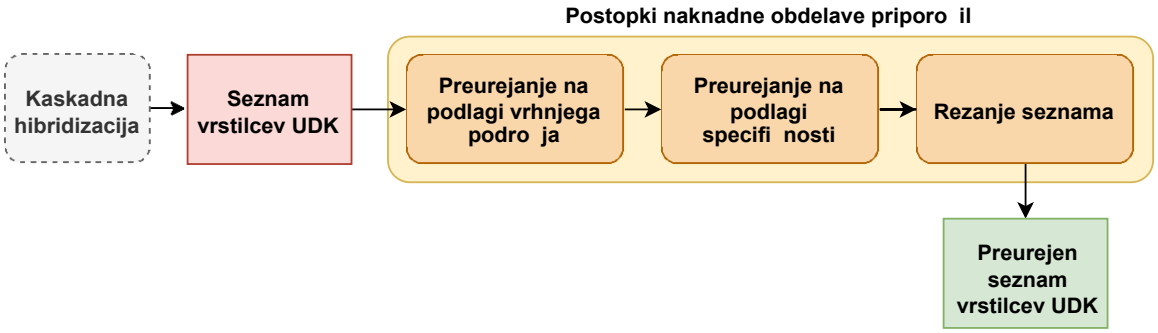
S postopkom kaskadne hibridizacije dobimo seznam priporočenih relevantnih vrstilcev UDK. Slednjega lahko že prikažemo uporabnikom, vendar je splošna praksa v modernih priporočilnih sistemih, da se priporočila pred prikazom uporabniku še naknadno obdelajo. V splošnem se postopki naknadne obdelave priporočil uporabljajo za to, da se vključi logika, ki izvede dodatno poosebitev rezultatov. V našem pristopu priporočanja se poslužimo postopkov naknadne obdelave z namenom, da bi bil končni rezultat bolj podoben ročnemu določanju vrstilcev UDK. Ročno določeni vrstilci UDK so ponavadi dokaj specifični, kar se vidi iz globine hierarhije UDK. Zasnovali smo tri naknadne obdelave priporočil, ki poskušajo posnemati ročno določanje vrstilcev UDK:

- *Preurejanje na podlagi vrhnjega področja* se uporabi za doseganje enotnosti vrstilcev UDK v seznamu priporočil.

- *Preurejanje na podlagi specifičnosti* se uporabi za višjo uvrstitev bolj specifičnih vrstilcev UDK v seznamu priporočil.

- *Rezanje seznama* se uporabi za odstranjevanje manj relevantnih vrstilcev UDK iz seznama priporočil.

Vse omenjene naknadne obdelave priporočil so zasnovane tako, da jih lahko poljubno vključujemo v delovni tok, prav tako pa jih lahko vključujemo v poljubnem vrstnem redu. Slika 3.10 prikazuje uporabo in zaporedje postopkov naknadne obdelave priporočil v predlaganem hibridnem pristopu za priporočanje vrstilcev UDK.



Slika 3.10: Uporaba in zaporedje postopkov naknadne obdelave priporočil v hibridnem pristopu za priporočanje vrstilcev UDK.

3.5.1 Preurejanje na podlagi vrhnjega področja

V končnem seznamu priporočenih vrstilcev UDK lahko dobimo več različnih vrstilcev UDK iz različnih področij znanosti. Kadar priporočamo vrstilce UDK za dokument z interdisciplinarno vsebino je to povsem pričakovano, za dokumente z domensko specifično vsebino pa si želimo priporočiti smiselne vrstilce UDK glede na področje znanosti, ki ga dokument opisuje. V tem primeru s preurejanjem na podlagi vrhnjega področja poskušamo seznam priporočenih vrstilcev UDK preurediti tako, da so višje uvrščeni vrstilci UDK, ki so del iste ali zelo sorodne veje hierarhije UDK. S tem želimo ustvariti občutek enotnosti priporočil, kar vodi v boljšo uporabniško izkušnjo.

Preurejanje na podlagi vrhnjega področja deluje tako, da se najprej zaznajo vrhnja področja za tri najvišje uvrščene vrstilce UDK v seznamu priporočil. Nato se izračuna razlika med oceno prvega in drugega vrstilca UDK v seznamu priporočil. Razlika tako predstavlja dominanco najboljše uvrščenega vrstilca UDK. Nadalje se po enačbi 3.5 izračunajo ocene vrstilcev UDK v seznamu priporočil, ki pripadajo dominantnemu vrhnjemu področju.

$$^{(T)}_{r} = {}^{(R_2)}_{r} \qquad 1 + \frac{}{jR_2j} \quad + {}_1 \log_2(r_{rank} + 1) \tag{3.5}$$

V enačbi 3.5 je $^{(R_2)}_{r}$ ocena vrstilca UDK r v seznamu R_2, r_{rank} je rang vrstilca UDK r v seznamu R_2, jR_2j pa predstavlja dolžino urejenega seznama priporočil R_2. S parametrom $_1$, definiranim na intervalu [0; 1], lahko določamo vpliv trenutnega ranga vrstilca UDK na

preurejanje. Po izračunu nove ocene $^{(T)}_{r}$ se seznam priporočil preuredi v padajočem vrstnem redu. Na sliki 3.11 je prikazan primer preurejanja na podlagi vrhnjega področja pri vrednosti parametra $_1$ = 0;1.

Vhodni seznam

Priporočen vrstilec UDK	$^{(R_2)}_{r}$
004.85	18,224
81‘32	14,306
004.8	13,23
81‘322	12,984
004.93	12,548
004.032.26	12,384
519.76	10,678

Preurejanje

$^{(T)}_{r}$	= 3;918 $_1$ = 0;1
28,524	
14,306	H-3
20,835	N+1
12,984	H-2
19,830	N+2
19,596	N+2
10,678	

Izhodni seznam

Priporočen vrstilec UDK	$^{(T)}_{r}$
004.85	28,524
004.8	20,835
004.93	19,830
004.032.26	19,596
81‘32	14,306
81‘322	12,984
519.76	10,678

Slika 3.11: Delovanje postopka preurejanja na podlagi vrhnjega področja.

V vhodnem seznamu sta bili med vrhnjimi področji prvih treh priporočil zaznani področji UDK 0 in 8. Dominantno vrhnje področje je 0, saj sta med prvimi tremi priporočili dva vrstilca UDK, ki spadata v to področje. Razlika med oceno prvega in drugega vrstilca UDK v vhodnem seznamu je 3,918. Z upoštevanjem enačbe 3.5 se izračunajo ocene $^{(T)}_{r}$ za vrstilce UDK v seznamu, ki pripadajo dominantnemu vrhnjemu področju. V fazi preurejanja se določijo višje uvrstitve vrstilcev UDK označenih z rumeno barvo, ki se upoštevajo v izhodnem seznamu. Vrstilci z višjim položajem v izhodnem seznamu so označeni z zeleno, z nižjim položajem pa z rdečo barvo. Iz vrstnega reda v izhodnem seznamu je moč opaziti, da so vrstilci, ki so del dominantnega področja UDK (področje 0), uvrščeni višje v seznamu.

3.5.2 Preurejanje na podlagi specifičnosti

Izkaže se, da med ročnim določanjem vrstilcev UDK knjižničarji stremijo k temu, da za dokumente določijo čim bolj specifične vrstilce UDK. V sami hierarhiji UDK to pomeni daljše nize znakov vrstilcev UDK, ki predstavljajo ozko specializirana področja znanosti. V seznamu priporočenih vrstilcev UDK lahko dobimo vrstilce UDK na različnih globinah hierarhije UDK. Namen tega postopka naknadne obdelave je, da bolj specifične vrstilce UDK uvrstimo višje v seznam. S tem se želimo približati rezultatu ročnega določanja vrstilcev UDK. Med preurejanjem na podlagi specifičnosti se ocene vsakega vrstilca UDK v seznamu priporočil ponovno izračunajo po enačbi 3.6. Novo oceno $^{(S)}_{r}$ izračunamo s

pomočjo stare ocene $^{(T)}_{r}$, ki je bila izračunana v postopku preurejanja na podlagi vrhnjega področja.

$$^{(S)}_{r} = {}^{(T)}_{r} + {}_{2}{}^{r} \frac{\log (r_{rank} + 1)}{\log_{max} 2} \tag{3.6}$$

V enačbi 3.6 je z_r predstavljena dolžina vrstilca UDK v seznamu priporočil, z $_{max}$ pa dolžina najdaljšega vrstilca UDK v seznamu priporočil. Enako kot pri preurejanju na podlagi vrhnjega področja je del enačbe r_{rank}, ki predstavlja rang vrstilca UDK r. Podobno kot pri preurejanju glede na vrhnje področje, nastopa še parameter $_2$, ki ima vrednosti omejene na intervalu [0; 1], s katerim določamo vpliv trenutnega ranga vrstilca na preurejanje.

Preurejanje na podlagi specifičnosti se izvede takoj za preurejanjem na podlagi vrhnjega področja, rezultat pa je nov seznam vrstilcev UDK, ki je urejen po padajočih ocenah $^{(S)}_{r}$.

Na sliki 3.12 je prikazan primer preurejanja na podlagi specifičnosti. V vhodnem seznamu je najdaljši vrstilec UDK 004.032.26 z dolžino $_{max}$ = 10. Z upoštevanjem vrednosti $_2$ = 0;8 se z enačbo 3.6 za vsak vrstilec UDK v seznamu izračunajo ocene $^{(S)}_{r}$.

Vhodni seznam		**Preurejanje**		**Izhodni seznam**	
Priporočen vrstilec UDK	$^{(T)}_{r}$	$^{(S)}_{r}$	$_2$ = 0;8	**Priporočen vrstilec UDK**	$^{(S)}_{r}$
004.85	28,524	29,004		004.85	29,004
004.8	20,835	21,469		004.8	21,469
004.93	19,830	20,79	H-1	004.032.26	21,454
004.032.26	19,596	21,454	N+1	004.93	20,79
81‘32	14,306	15,34		81‘32	15,34
81‘322	12,984	14,332		81‘322	14,332
519.76	10,678	12,118		519.76	12,118

Slika 3.12: Delovanje postopka preurejanja na podlagi specifičnosti.

V fazi preurejanja se pri vrstilcih UDK, označenih z rumeno barvo, zgodijo spremembe v rangih. Spremembe v vrstnem redu v izhodnem seznamu so označene z zeleno barvo za tiste vrstilce UDK, ki so bili uvrščeni višje glede na njihovo prejšnjo uvrstitev, in z rdečo barvo za tiste vrstilce UDK, ki so bili uvrščeni nižje glede na njihovo prejšnjo uvrstitev. Iz vrstnega reda v izhodnem seznamu je moč opaziti, da je vrstilec 004.032.26 pridobil mesto, krajši vrstilec 004.93 pa je izgubil mesto. Prav tako se je zmanjšala razlika med

ocenama vrstilcev 81‘32 in 81‘322, saj je vrstilec 81‘322 pridobil več na oceni kot vrstilec 81‘32.

3.5.3 Rezanje seznama

V nekaterih primerih lahko pride do velikega razhajanja med ocenami vrstilcev UDK v seznamu priporočil. Lahko se zgodi, da imajo najvišje uvrščeni vrstilci UDK neprimerno višjo oceno kot preostali priporočeni vrstilci UDK. Pojavi se vprašanje, ali je sploh smiselno obdržati nižje uvrščene vrstilce UDK v seznamu priporočil, ki ga bomo prikazali končnemu uporabniku. Z namenom filtriranja takšnih vrstilcev UDK izvedemo rezanje seznama glede na prag T, ki ga izračunamo po enačbi 3.7.

$$T = \max_r {}^{(S)}{}_r \tag{3.7}$$

V enačbi 3.7 je $\max_r {}^{(S)}{}_r$ ocena najvišje uvrščenega vrstilca UDK v seznamu priporočil, pa je hiperparameter z vrednostjo med 0 in 1. Ta predstavlja odstotek najvišje ocene, ki ga mora doseči posamezen vrstilec UDK, da ga obdržimo v seznamu. Vsi vrstilci UDK, katerih vrednosti ocen so pod pragom T, se odstranijo iz seznama. Na sliki 3.13 je prikazan primer rezanja seznama z vrednostjo $= 0;5$, ki določa prag $T = 14;502$. V prikazanem primeru se iz seznama odstranita vrstilca 81‘322 in 519.76.

Vhodni seznam

Priporočen vrstilec UDK	$^{(S)}_r$
004.85	29,004
004.8	21,469
004.032.26	21,454
004.93	20,79
81‘32	15,34
81‘322	14,332
519.76	12,118

Rezanje seznama

$= 0;5$
$T = 14;502$
3
3
3
3
3
7
7

Izhodni seznam

Priporočen vrstilec UDK	$^{(S)}_r$
004.85	29,004
004.8	21,469
004.032.26	21,454
004.93	20,79
81‘32	15,34

Slika 3.13: Delovanje postopka rezanja seznama.

Primer seznama priporočil, ki je uporabljen za prikaz vseh treh postopkov naknadne obdelave priporočil (slike 3.11, 3.12 in 3.13), je rezultat priporočanja vrstilcev UDK za interdisciplinarno gradivo, ki govori o uporabi nevronskih mrež za klasifikacijo besedil. Gre torej za interdisciplinarno gradivo, ki je vezano na področje računalništva in jezikoslovja.

To je razvidno tudi iz izhodnega seznama po izvedbi rezanja seznama. Prvo izrazito vrhnje področje je področje 0 (znanost in znanje) oziroma konkretneje področje 004 (računalništvo). Drugo izrazito vrhnje področje je področje 8 (jezikoslovje, filologija, književnost) oziroma konkretneje področje 81 (jezikoslovje in jeziki).

V tabeli 3.1 so podani pomeni posameznih vrstilcev UDK, ki so v seznamu priporočil. Vidimo, da so priporočeni vrstilci UDK smiselni za izbrano interdisciplinarno gradivo, prav tako pa je iz seznama priporočil možno razbrati primarno in sekundarno področje. Opaziti je možno tudi različne nivoje specifičnosti priporočenih vrstilcev UDK.

Tabela 3.1: Pomeni končnih priporočenih vrstilcev v uporabljenem zgledu.

Priporočen vrstilec UDK	**Pomen**
004.85	Strojno učenje
004.8	Umetna inteligenca
004.032.26	Nevronske mreže
004.93	Obdelava informacij v vzorcih
81‘32	Matematično jezikoslovje

4 Eksperiment in rezultati

Da bi preverili zastavljene hipoteze doktorske disertacije, smo v sklopu eksperimenta predlagani pristop za hibridno priporočanje vrstilcev univerzalne decimalne klasifikacije ovrednotili z uveljavljenimi metrikami za vrednotenje priporočilnih sistemov. Primernost predlaganega pristopa smo ocenili na podlagi dobljenih vrednosti ocen metrik, ki smo jih uporabili za vrednotenje. Za potrebe eksperimenta smo vzpostavili eksperimentalno okolje, v katerem smo določili vrednosti parametrov, izbrali podatkovno zbirko, določili metrike za vrednotenje in izvedili primerjave različnih pristopov.

4.1 Eksperimentalno okolje

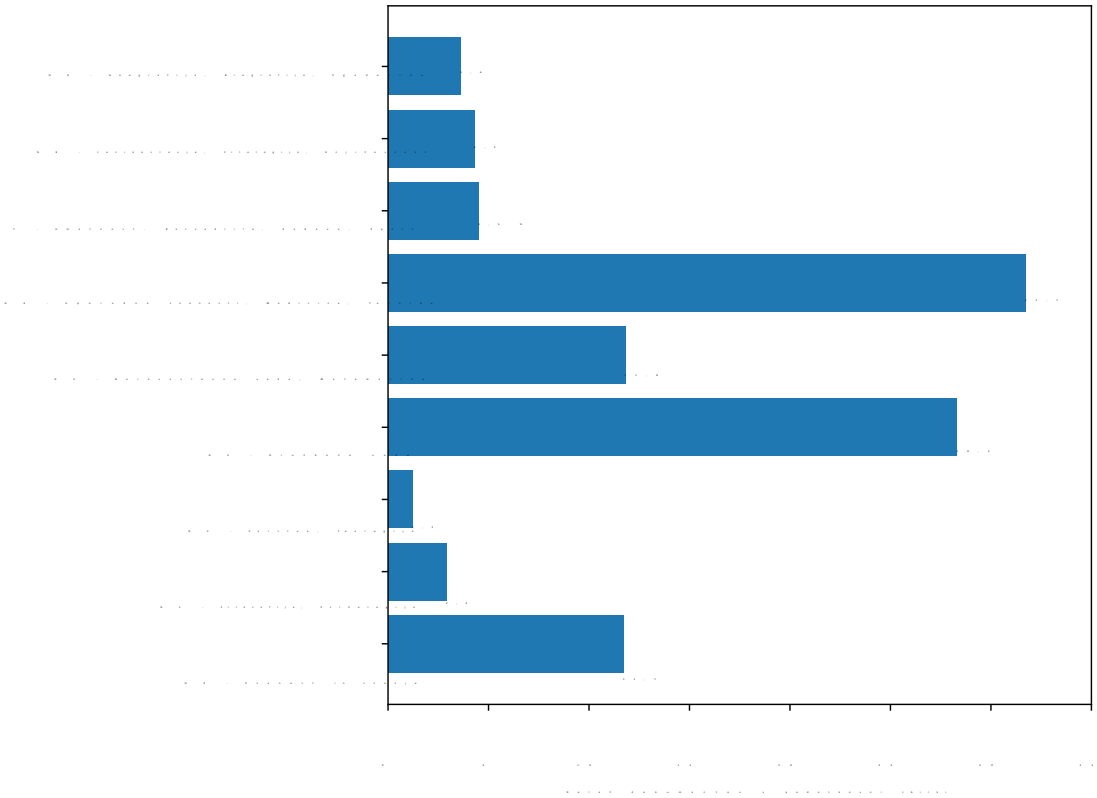
V sklopu eksperimenta smo zasnovali štiri scenarije, ki podrobneje primerjajo uporabo različnih pristopov za izvedbo priporočanja vrstilcev UDK. Pri uporabi rangirne funkcije BM25 smo za vrednosti parametrov uporabili uveljavljene in s strani avtorjev priporočene vrednosti (k_1 = 1;2 in b = 0;75) [34]. Preizkusili smo več vrednosti za različne parametre postopkov naknadne obdelave in izbrali tiste, pri katerih smo glede na izbrane metrike za vrednotenje predlaganega pristopa dobili najboljše vrednosti metrik na validacijski množici. Za postopek preurejanja na podlagi vrhnjega področja smo tako uporabili vrednost parametra $_1$ = 0;2, za postopek preurejanja na podlagi specifičnosti smo uporabili vrednost parametra $_2$ = 0;8, za postopek rezanja seznama pa vrednost parametra = 0;5. Utemeljitev izbire teh vrednosti za te tri parametre podrobneje opišemo v podpoglavju 4.3. Pri metrikah HR@K in NDCG@K smo se omejili na vrednosti K = [1; 3; 5], ki predstavljajo standardne vrednosti za omenjene metrike pri vrednotenju priporočilnih sistemov [115, 118].

4.1.1 Podatkovna zbirka

Za vrednotenje predlaganega pristopa smo izbrali podatkovno zbirko OpenScience metadata dataset [44], ki vsebuje dokumente z različnih področij znanosti in njihove metapodatke v slovenskem jeziku. Vsi dokumenti so prosto dostopni v sklopu slovenske infrastrukture odprtega dostopa. Podatkovna zbirka vsebuje 114.485 dokumentov in njihovih metapodatkovnih zapisov.

V sklopu eksperimenta smo uporabljali segmentirane metapodatke, ki so zajemali naslove, povzetke, ključne besede, izraze UDK in šifro organizacije po šifrantu eVŠ. Izrazi UDK

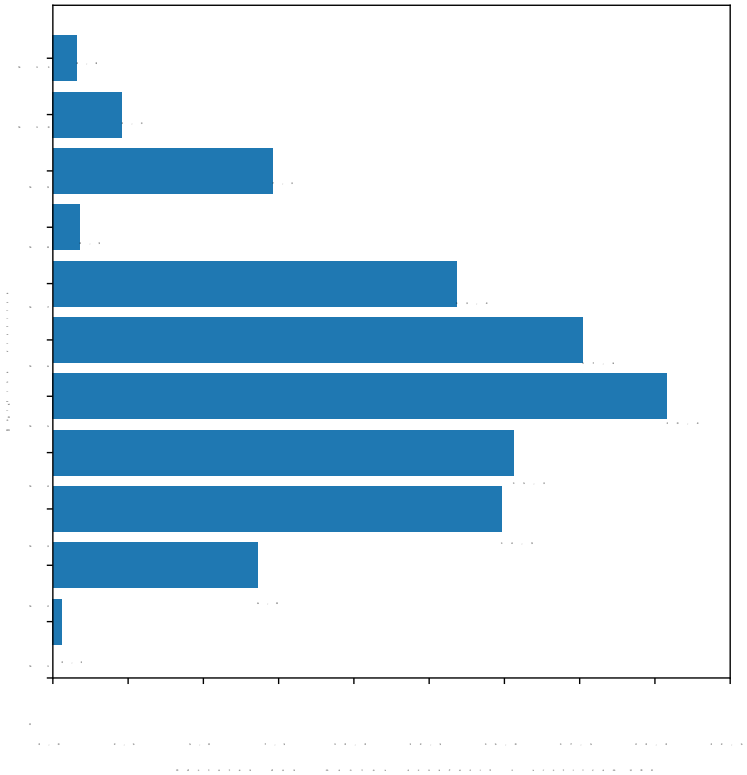
so bili obdelani z enakim razpoznavalnikom vrstilcev UDK kot pri učenju večznačnega klasifikatorja, zato se privesni vrstilci niso upoštevali kot razpoznani vrstilci. Na podlagi ujemajočih vrstilcev UDK smo razpoznane vrstilce povezali s prosto dostopnim katalogom UDK [45]. Deleže dokumentov v podatkovni zbirki glede na vrhnja področja hierarhije UDK prikazuje slika 4.1.



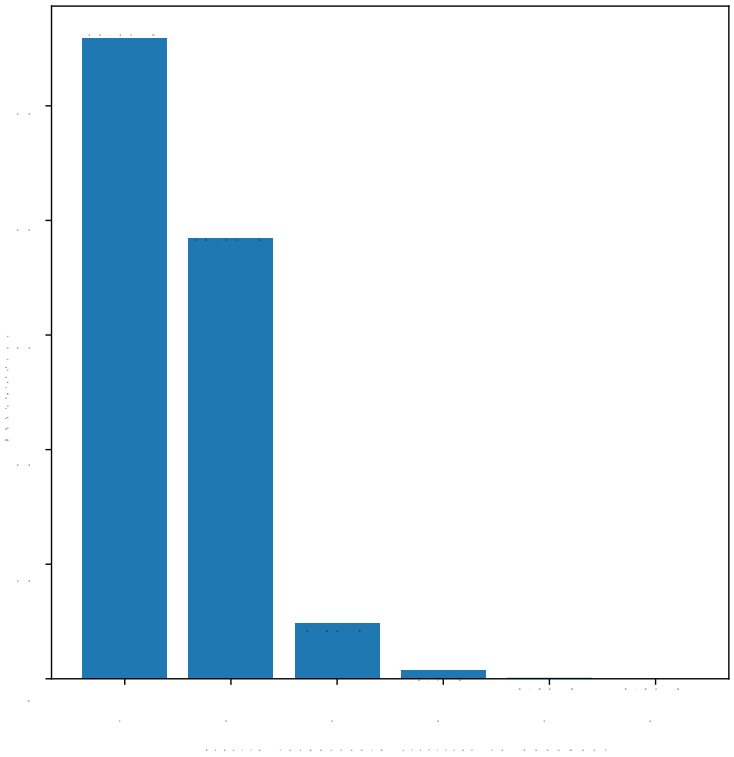
Slika 4.1: Deleži dokumentov v podatkovni množici glede na vrhnja področja hierarhije UDK.

Za podatkovno zbirko smo preverili tudi porazdelitev dolžin razpoznanih vrstilcev UDK, ki definirajo področja dokumentov. Dolžina vrstilcev UDK se preslika v njihovo globino v hierarhiji UDK. Deleže dokumentov v podatkovni množici glede na dolžino razpoznanih vrstilcev UDK prikazuje slika 4.2.

Nazadnje smo preverili tudi koliko vrstilcev UDK na dokument je razpoznanih. Na sliki 4.3 vidimo, da ima 55,91 % dokumentov en razpoznan vrstilec, 38,41 % dokumentov pa dva razpoznana vrstilca UDK. Tri ali več razpoznane vrstilce UDK ima razpoznanih 5,67 % dokumentov.

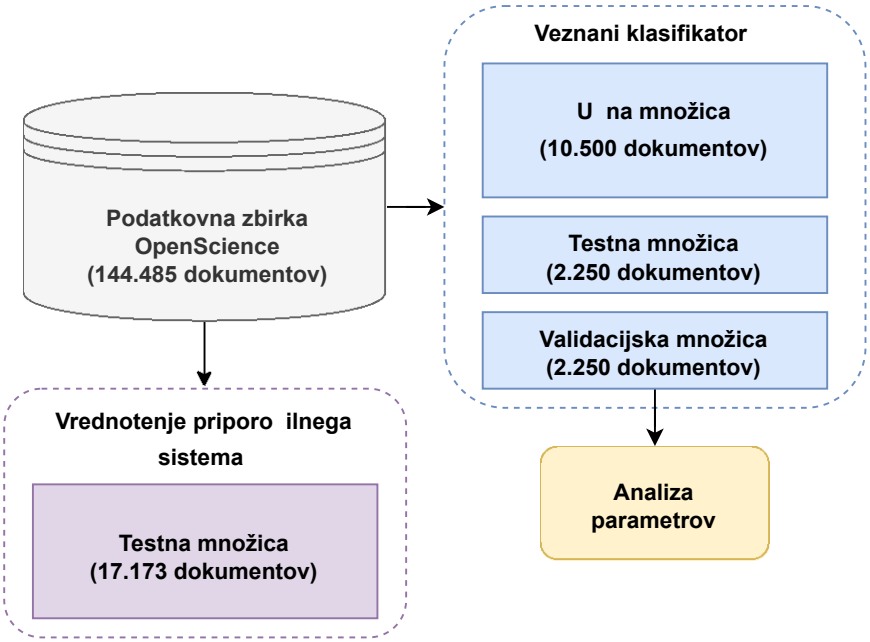


Slika 4.2: Deleži dokumentov v podatkovni zbirki glede na dolžino razpoznanih vrstilcev UDK.



Slika 4.3: Deleži dokumentov v podatkovni zbirki glede na število razpoznanih vrstilcev UDK.

Pri vrednotenju smo se omejili na testno podatkovno množico, ki predstavlja del podatkovne zbirke OpenScience in ne vsebuje dokumentov, ki so bili uporabljeni za učenje in validacijo večznačnega klasifikatorja. Ta podatkovna množica vsebuje 15 % celotne podatkovne zbirke kar znaša 17.173 dokumentov. Za analizo parametrov predlaganega hibridnega priporočilnega sistema smo uporabili enako validacijsko množico kot pri validaciji večznačnega klasifikatorja. Delitev podatkov je grafično ponazorjena s sliko 4.4.



Slika 4.4: Grafična ponazoritev delitve podatkov za večznačni klasifikator in eksperiment.

4.1.2 Učenje večznačnega klasifikatorja

Predlagani večznačni klasifikator smo učili na vseh vejah hierarhije UDK, ki pokriva vsa področja znanosti in je prosto dostopna v povzetku UDK [45]. Za učenje in testiranje večznačnega klasifikatorja smo uporabili del podatkovne zbirke OpenScience [129], ki je zajemal 15.000 dokumentov v slovenskem jeziku in smo ga razdelili v razmerju 70:15:15. Učna množica je tako zajemala 10.500 dokumentov, testna in validacijska množica pa vsaka po 2.250 dokumentov. Te množice so bile pripravljene s stratificiranim vzorčenjem tako, da so vrstilci za vrhnja področja hierarhije UDK zastopani v enakih deležih.

Model smo učili s prednaučenim jezikovnim modelom SloBERTa, ki je bil naučen na besedilih v slovenskem jeziku. Pri tem smo uporabili vrednosti hiperparametrov, ki so podane v tabeli 4.1. Pri učenju smo uporabljali optimizacijski algoritem AdamW [130] in linearno ogrevanje s kosinusnim ohlajanjem (ang. linear warmup with cosine annealing) vrednosti hitrosti učenja med postopkom učenja [131].

Tabela 4.1: Uporabljene vrednosti hiperparametrov pri učenju večznačnega klasifikatorja.

Hiperparameter	Vrednost
max_len	512
batch_size	8
lr	2e-5
warmup	0,2
wd	1e-5
epochs	4

4.1.3 Zasnova primerjav

V skladu s postavljenimi hipotezami smo zasnovali primerjave različnih pristopov za izvedbo priporočanja vrstilcev UDK. Tako smo za potrebe Hipoteze 1 zasnovali primerjavo med predlaganim hibridnim pristopom za priporočanje vrstilcev UDK in pristopom, ki temelji izključno na vsebinskem filtriranju in se izvede z rangirno funkcijo BM25. Za potrebe Hipoteze 2 smo zasnovali primerjavo, med predlaganim hibridnim pristopom za priporočanje vrstilcev UDK in pristopom, ki temelji izključno na večrazredni klasifikaciji. Na podlagi rezultatov v sorodnih delih smo izbrali pristop s podpornimi vektorji, ki se je v teh sorodnih delih izkazal za najboljšega. Dodatno smo v tej primerjavi z omenjenima pristopoma primerjali tudi samostojno večznačno različico pristopa s podpornimi vektorji in samostojni večznačni klasifikator, ki ga uporablja predlagani hibridni pristop za priporočanje vrstilcev UDK. Za potrebe Hipoteze 3 smo zasnovali primerjavo med dvema izvedbama predlaganega hibridnega pristopa za priporočanje vrstilcev UDK. V prvi izvedbi niso upoštevani metapodatki o izvoru dokumenta (tj. izvedba brez drugega kaskadnega koraka), v drugi izvedbi pa so metapodatki o izvoru dokumenta upoštevani (tj. izvedba z drugim kaskadnim korakom). Dodatno smo zasnovali še četrto primerjavo, v sklopu katere smo preučevali vpliv vključevanja postopkov naknadne obdelave priporočil na rezultate predlaganega hibridnega pristopa za priporočanje vrstilcev UDK. V vseh primerjavah smo rezultate vrednotili z metrikami HR@K, MAP, MRR in NDCG@K, ki se pogosto uporabljajo v industriji in na raziskovalnem področju priporočilnih sistemov.

4.2 Rezultati eksperimenta

Z vzpostavljenim eksperimentalnim okoljem smo z izbrano podatkovno množico, definiranimi primerjavami in izbranimi metrikami izvedli vrednotenje predlaganega hibridnega pristopa za priporočanje vrstilcev UDK. Pri vrednotenju dobljene sezname priporočenih vrstilcev UDK primerjamo s seznami vrstilcev UDK, ki so bili za vsak dokument v testni množici razpoznani iz njihovega dejanskega katalogiziranega izraza UDK. Pri določanju

statistično značilnih razlik smo primerjali rezultate metrik za posamezne pare pristopov z Wilcoxonovim statističnim testom [132], kjer smo mejo statistične značilnosti nastavili na = 0;05 [133]. V nadaljevanju predstavimo rezultate vrednotenja za posamezno definirano primerjavo. Pri vrednotenju smo uporabili testno množico, ki obsega 17.173 dokumentov (slika 4.4).

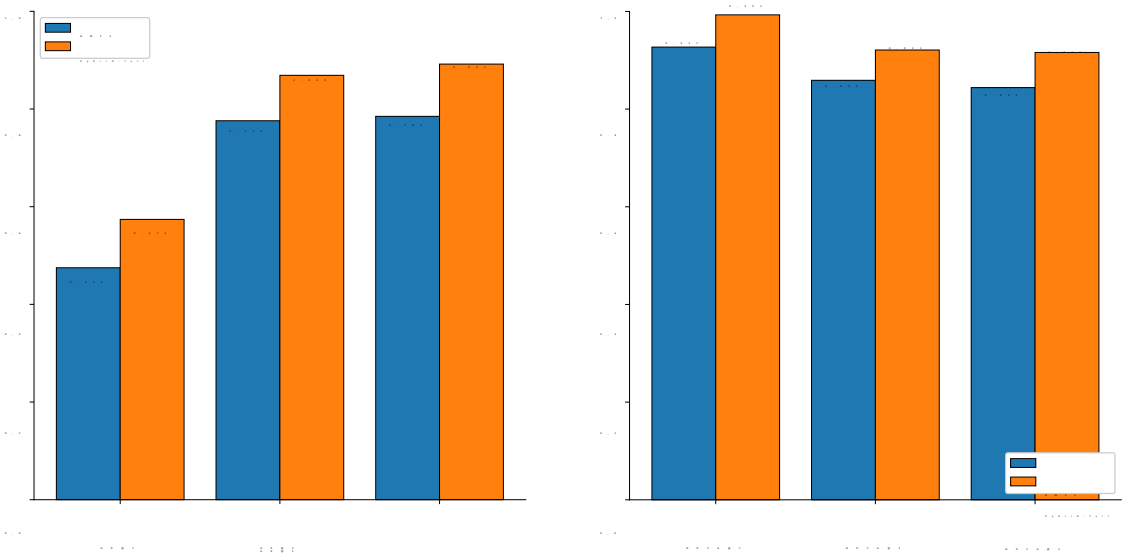
4.2.1 Primerjava 1

Ta primerjava vključuje predlagan hibridni pristop za priporočanje vrstilcev UDK in pristop, ki temelji izključno na vsebinskem filtriranju in se izvede z rangirno funkcijo BM25 (slika 4.5). Podoben pristop z rangirno funkcijo BM25 so za pridobivanje vrstilcev DDK uporabili v [10]. Tabela 4.2 prikazuje dosežene povprečne vrednosti metrik HR@K, NDCG@K, MRR in MAP. Pristop, ki temelji izključno na vsebinskem filtriranju, je označen z "BM25", predlagan hibridni pristop za priporočanje vrstilcev UDK pa s "hybrid-full".

Tabela 4.2: Dosežene povprečne vrednosti in standardni odkloni (v oklepajih) metrik HR@K, NDCG@K, MRR in MAP za pristopa v primerjavi 1.

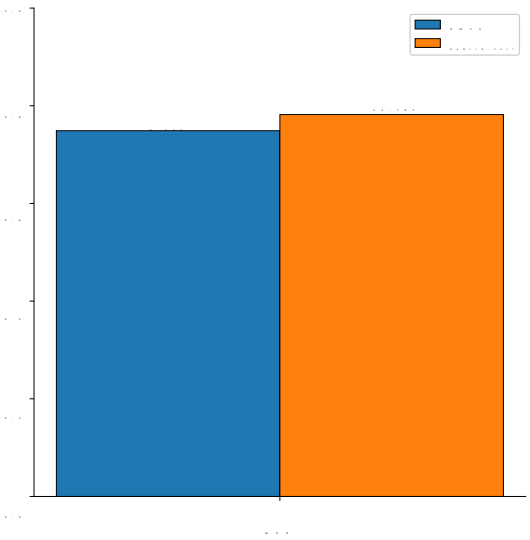
	HR@K			**NDCG@K**				
Pristop	K = 1	K = 3	K = 5	K = 1	K = 3	K = 5	**MRR**	**MAP**
BM25	0,475 (0,152)	0,776 (0,132)	0,785 (0,148)	0,927 (0,121)	0,859 (0,134)	0,844 (0,167)	0,749 (0,127)	0,728 (0,114)
hybrid-full	**0,574** (0,131)	**0,869** (0,128)	**0,892** (0,136)	**0,993** (0,108)	**0,921** (0,115)	**0,916** (0,118)	**0,782** (0,104)	**0,785** (0,098)

Opazimo, da vrednosti metrike HR@K rastejo, če povečujemo K, saj z daljšimi seznami priporočil zajamemo večje število pravilnih vrstilcev UDK. Predlagan pristop hibridnega priporočanja se za matriko HR@K za vse vrednosti K izkaže za boljšega od pristopa BM25. V najstrožji obliki metrike (K = 1) pristop BM25 doseže 47,5 %, predlagan pristop pa 57,4 %. V manj strogih oblikah metrike (K = 3 in K = 5) pristop BM25 doseže 77,6 % in 78,5 %, predlagan pristop pa 86,9 % in 89,2 %. Predlagan pristop hibridnega priporočanja torej zajame več pravilnih vrstilcev UDK v seznamih priporočil dolžine K kot pristop BM25. V nasprotju z metriko HR@K, vrednosti metrike NDCG@K padajo, če povečujemo K. To pomeni, da se z daljšim seznamom priporočil slabša kvaliteta rangiranja priporočil, saj smo za pravilno napoved vrstilca UDK potrebovali daljši seznam priporočil. Prav tako je lahko pravilen vrstilec UDK uvrščen nižje v seznamu priporočil, kar vpliva na vrednost metrike NDCG@K. Tudi pri tej metriki se predlagan pristop hibridnega priporočanja za vse vrednosti K izkaže za boljšega od pristopa BM25. V najstrožji obliki metrike (K = 1) pristop BM25 doseže 92,7 %, predlagan pristop hibridnega priporočanja pa 99,3 %. V manj strogih oblikah metrike (K = 3 in K = 5) pristop BM25 doseže 85,9 % in 84,4 %,

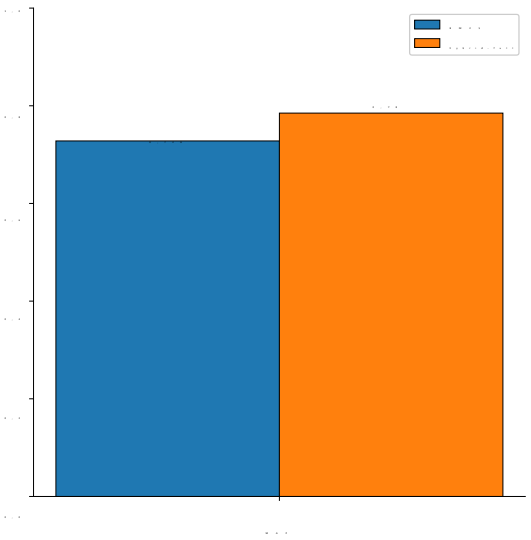


(a) HR@K

(b) NDCG@K



(c) MRR



(d) MAP

Slika 4.5: Primerjava povprečnih vrednosti metrik HR@K, NDCG@K, MRR in MAP za pristopa v primerjavi 1.

predlagan pristop pa 92,1 % in 91,6 %. Predlagan pristop hibridnega priporočanja torej vrača bolje rangirane sezname priporočil dolžine K kot pristop BM25.

Povprečne vrednosti metrike MRR znašajo 74,9 % za pristop BM25 in 78,2 % za predlagan pristop. Iz tega je moč zaključiti, da predlagan pristop rangira pravilne vrstilce UDK višje v seznamih priporočil. Vrednosti metrike MAP znašajo 72,8 % za pristop BM25 in 78,5 % za predlagan pristop, kar kaže na to, da predlagan pristop v povprečju vrača več pravilnih vrstilcev UDK na začetku seznamov priporočil kot pristop BM25.

Za vsako metriko smo statistično značilne razlike preverili z Wilcoxonovim statističnim testom z mejo statistične značilnosti = 0;05. Ničelna hipoteza H_0 izraža stanje brez razlik med pristopoma v primerjavi. Na podlagi dobljenih vrednosti statistične značilnosti p (tabela 4.3) zavržemo ničelno hipotezo H_0 in zaključimo, da predlagan hibridni pristop vrne statistično značilno boljše rezultate od pristopa priporočanja, ki temelji izključno na vsebinskem filtriranju.

Tabela 4.3: Dobljene vrednosti statistične značilnosti p v primerjavi 1.

	HR@1	**HR@3**	**HR@5**	**NDCG@1**	**NDCG@3**	**NDCG@5**	**MRR**	**MAP**
p	0,002	0,001	0,008	0,002	0,004	0,007	0,003	0,003

4.2.2 Primerjava 2

Ta primerjava vključuje predlagan hibridni pristop za priporočanje vrstilcev UDK in pristop, ki za večrazredni klasifikator uporablja metodo podpornih vektorjev (MC-SVM). Večrazredni klasifikator z metodo podpornih vektorjev so za klasifikacijo vrstilcev DDK uporabili v [10], za klasifikacijo vrstilcev UDK pa v [11]. Za predstavitev besedila pa so oboji upora-bili utežno shemo `tf-idf`. Za potrebe primerjave smo večrazredni klasifikator z metodo podpornih vektorjev prilagodili tako, da kot rezultat ne vrne zgolj vrstilca s pripadajočo najvišjo vrednostjo verjetnosti, kot je to značilno za večrazredno klasifikacijo, temveč vrne seznam vrstilcev s pripadajočimi vrednostmi verjetnosti, katerega nato omejimo na K najbolj verjetnih vrstilcev.

V primerjavo smo vključili tudi večznačno izvedbo klasifikatorja z metodo podpornih vektorjev (ML-SVM) in večznačni klasifikator s priučenim jezikovnim modelom SloBERTa (ML-UDC-SloBERTa). Podoben jezikovni model so za klasifikacijo vrstilcev DDK uporabili v [12]. Večznačno izvedbo klasifikatorja z metodo podpornih vektorjev smo implementirali z uporabo strategije “en proti ostalim” (ang. one-vs-rest strategy, OvR). S to strategijo se za vsako možno oznako v problemu večznačne klasifikacije ustvari binarni klasifikator. Pri napovedi se vhod pošlje skozi vse ustvarjene binarne klasifikatorje, ki določajo posamezne vrstilce UDK. Tudi tukaj smo za predstavitev besedila uporabili utežno shemo `tf-idf`. Dodatno smo v primerjavo vključili še metodo XR-Transformer, ki se uporablja za večznačno klasifikacijo z velikim številom oznak [86]. Večznačni klasifikator XR-Transformer smo implementirali s prosto dostopno programsko kodo avtorjev, ki je na voljo v ogrodju Amazon PECOS [87]. Rezultat je seznam dolžine K z vrstilci UDK, ki so urejeni po vrednosti verjetnosti. Tabela 4.4 prikazuje dosežene povprečne vrednosti metrik HR@K, NDCG@K, MRR in MAP, slika 4.6 pa prikazuje primerjavo med njimi.

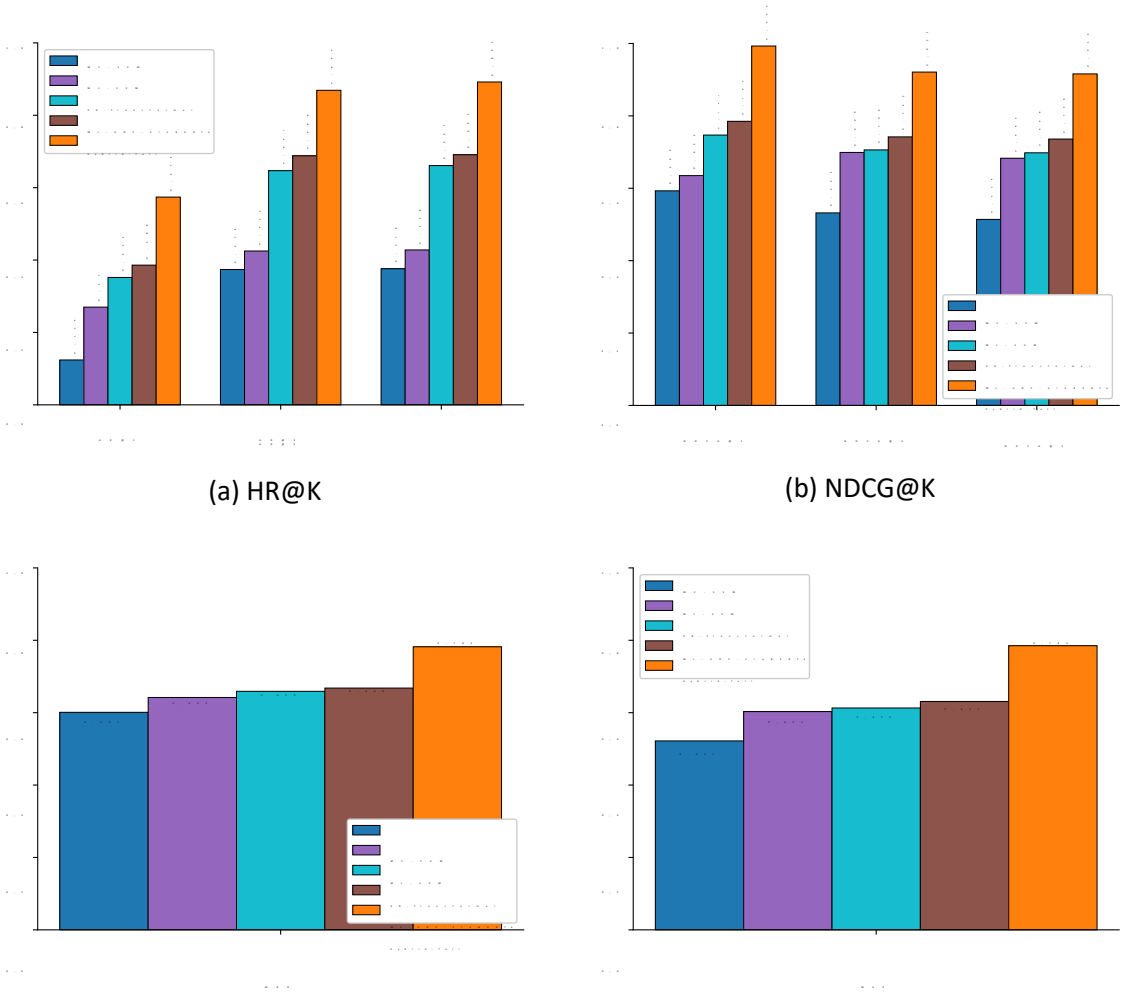
Tabela 4.4: Dosežene povprečne vrednosti in standardni odkloni (v oklepajih) metrik HR@K, NDCG@K, MRR in MAP za pristope v primerjavi 2.

	HR@K			**NDCG@K**				
Pristop	K = 1	K = 3	K = 5	K = 1	K = 3	K = 5	**MRR**	**MAP**
MC-SVM	0,124 (0,116)	0,374 (0,118)	0,376 (0,097)	0,593 (0,175)	0,532 (0,146)	0,514 (0,122)	0,601 (0,138)	0,522 (0,103)
ML-SVM	0,270 (0,141)	0,425 (0,126)	0,428 (0,114)	0,635 (0,188)	0,699 (0,157)	0,683 (0,102)	0,642 (0,104)	0,603 (0,112)
XR-Transformer	0,352 (0,163)	0,647 (0,121)	0,661 (0,109)	0,747 (0,168)	0,706 (0,154)	0,698 (0,139)	0,659 (0,106)	0,613 (0,115)
ML-UDC-SloBERTa	0,386 (0,167)	0,688 (0,116)	0,691 (0,103)	0,785 (0,161)	0,742 (0,153)	0,736 (0,157)	0,668 (0,095)	0,631 (0,117)
hybrid-full	**0,574** (0,131)	**0,869** (0,128)	**0,892** (0,136)	**0,993** (0,108)	**0,921** (0,115)	**0,916** (0,118)	**0,782** (0,104)	**0,785** (0,098)

Predlagan pristop hibridnega priporočanja vrstilcev UDK se po vseh metrikah izkaže za boljšega od vseh ostalih pristopov. Med večznačnimi klasifikatorji se za boljšega izkaže pristop ML-UDC-SloBERTa, za najslabšega pa se izkaže večrazredni klasifikator SVM (MC-SVM). Povprečne vrednosti metrike HR@K za predlagan pristop so od povprečnih vrednosti za pristop MC-SVM višje med 45 % in 51,6 %, vrednosti metrike NDCG@K pa so višje med 38,9 % in 40,2 %. Predlagan pristop v primerjavi s pristopom MC-SVM doseže 18,1 % višjo vrednost za metriko MRR in 26,3 % višjo vrednost za metriko MAP.

Povprečne vrednosti metrike HR@K za predlagan pristop so od vrednosti za pristop ML-SVM višje med 30,4 % in 46,4 %, povprečne vrednosti metrike NDCG@K pa so višje med 22,2 % in 35,8 %. Predlagan pristop v primerjavi s pristopom ML-SVM doseže 14 % višjo vrednost za metriko MRR in 18,2 % višjo vrednost za metriko MAP. Povprečne vrednosti metrike HR@K za predlagan prestop so od vrednosti za pristop XR-Transformer višje med 22,2 % in 23,1 %, povprečne vrednosti metrike NDCG@K pa so višje med 21,5 % in 24,6 %. Predlagan pristop v primerjavi s pristopom XR-Transformer doseže 12,3 % višjo vrednost za metriko MRR in 17,2 % višjo vrednost za metriko MAP. Povprečne rednosti metrike HR@K za predlagan pristop so od povprečnih vrednosti za pristop ML-UDC-SloBERTa višje med 18,1 % in 20,1 %, povprečne vrednosti metrike NDCG@K pa so višje med 17,9 % in 20,8 %. Predlagan pristop v primerjavi s pristopom ML-UDC-SloBERTa doseže 11,4 % višjo vrednost za metriko MRR in 15,4 % višjo vrednost za metriko MAP.

Iz rezultatov primerjave je razvidno, da hibridni pristop priporočanja vrstilcev UDK vrača boljše sezname priporočil v primerjavi s samostojnim večrazrednim klasifikatorjem, kot tudi v primerjavi z obema samostojnima večznačnima klasifikatorjema. Za vsako metriko



(a) HR@K (b) NDCG@K

(c) MRR (d) MAP

Slika 4.6: Primerjava povprečnih vrednosti metrik HR@K, NDCG@K, MRR in MAP za pristope v primerjavi 2.

smo statistično značilne razlike preverili z Wilcoxonovim statističnim testom z mejo statistične značilnosti = 0;05. Ničelna hipoteza H_0 izraža stanje brez razlik med predlaganim hibridnim pristopom in pristopom z večrazrednim klasifikatorjem. Na podlagi dobljenih vrednosti statistične značilnosti p (tabela 4.5) zavržemo ničelno hipotezo H_0 in zaključimo, da predlagan hibridni pristop vrne statistično značilno boljše rezultate od pristopa priporočanja, ki temelji izključno na večrazredni klasifikaciji.

Tabela 4.5: Dobljene vrednosti statistične značilnosti p v primerjavi 2.

	HR@1	**HR@3**	**HR@5**	**NDCG@1**	**NDCG@3**	**NDCG@5**	**MRR**	**MAP**
p	0,008	0,002	0,006	0,009	0,004	0,0006	0,0003	0,007

4.2.3 Primerjava 3

V tej primerjavi se primerjata izvedbi predlaganega hibridnega pristopa za priporočanje vrstilcev UDK z in brez upoštevanja metapodatkov o izvoru dokumenta (slika 4.7). Tabela 4.6 prikazuje dosežene povprečne vrednosti metrik HR@K, NDCG@K, MRR in MAP. Izvedba predlaganega hibridnega pristopa za priporočanje vrstilcev UDK, ki ne upošteva metapodatkov o izvorni organizaciji dokumenta je označena z nazivom "hybrid-no-org". Izvedba predlaganega hibridnega pristopa za priporočanje vrstilcev UDK, ki upošteva tudi metapodatke o izvorni organizaciji dokumenta, je označena z nazivom "hybrid-full".

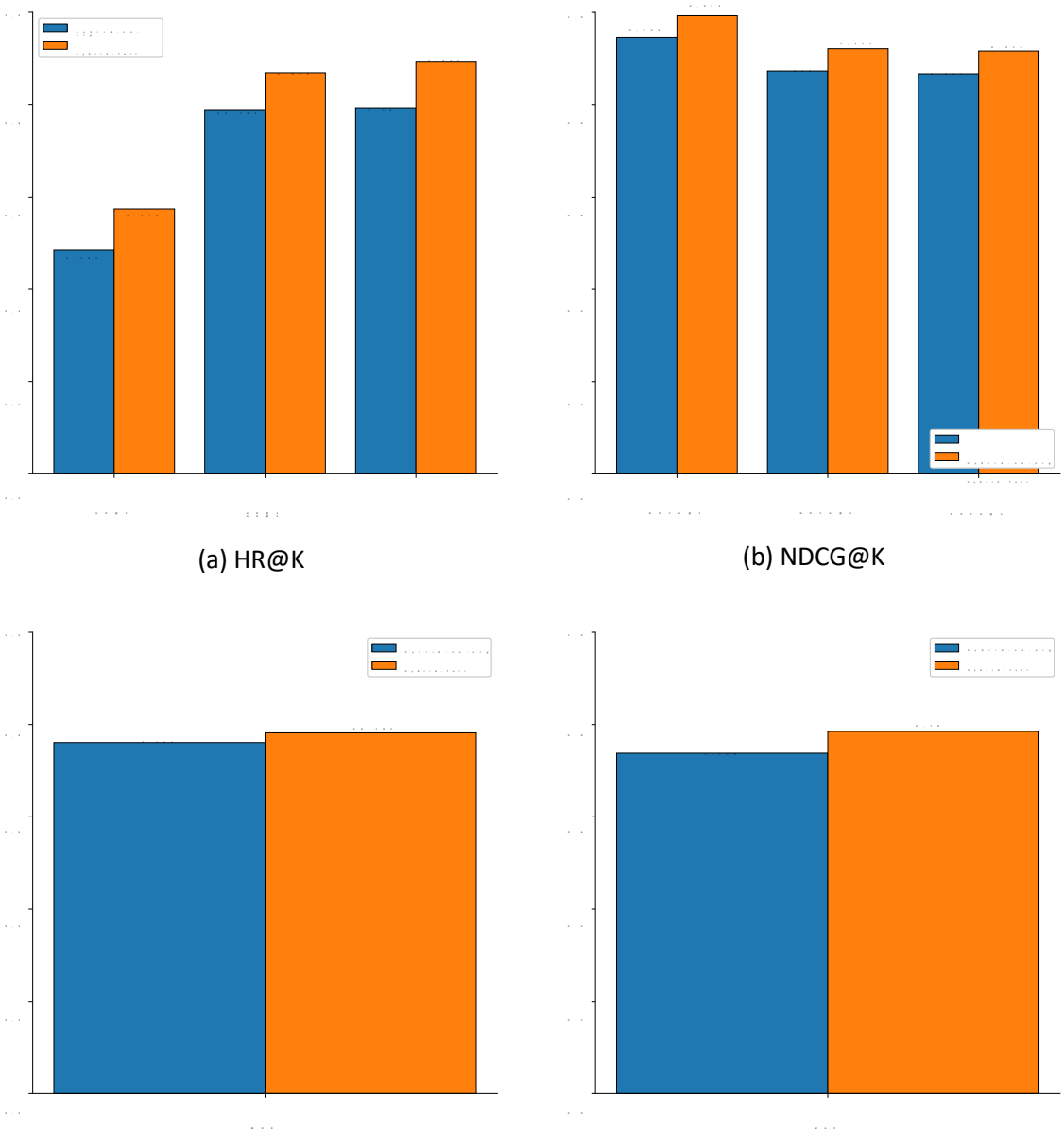
Tabela 4.6: Dosežene povprečne vrednosti in standardni odkloni (v oklepajih) metrik HR@K, NDCG@K, MRR in MAP za pristopa v primerjavi 3.

	HR@K			**NDCG@K**				
Pristop	K = 1	K = 3	K = 5	K = 1	K = 3	K = 5	**MRR**	**MAP**
hybrid-no-org	0,484 (0,138)	0,789 (0,114)	0,793 (0,115)	0,946 (0,112)	0,873 (0,119)	0,867 (0,113)	0,761 (0,102)	0,738 (0,102)
hybrid-full	**0,574** (0,131)	**0,869** (0,128)	**0,892** (0,136)	**0,993** (0,108)	**0,921** (0,115)	**0,916** (0,118)	**0,782** (0,104)	**0,785** (0,098)

Predlagan pristop hibridnega priporočanja, ki upošteva metapodatke o izvorni organizaciji dokumentov se po metriki HR@K za vse vrednosti K izkaže bolje od predlaganega hibridnega pristopa, ki metapodatkov ne upošteva. V najstrožji obliki metrike (K = 1) predlagan hibridni pristop z metapodatki o izvorni organizaciji dokumentov doseže za 9 % višjo vrednost. V manj strogih oblikah metrike (K = 3 in K = 5) predlagan hibridni pristop z metapodatki doseže za 8 % in 9,9 % višjo vrednost.

Predlagan pristop hibridnega priporočanja z metapodatki o izvorni organizaciji dokumenta torej zajame več pravilnih vrstilcev UDK v seznamih priporočil dolžine K kot predlagan pristop hibridnega priporočanja brez metapodatkov o izvorni organizaciji dokumenta. Tudi po metriki NDCG@K se predlagan pristop hibridnega priporočanja z metapodatki o izvorni organizaciji dokumenta za vse vrednosti K izkaže za boljšega. V najstrožji obliki metrike (K = 1) pristop z upoštevanjem metapodatkov o izvorni dokumentaciji doseže 4,7 % višjo vrednost, v manj strogih oblikah metrike (K = 3 in K = 5) pa 4,8 % in 4,9 % višjo vrednost.

Primerjava povprečnih vrednosti metrik MRR in MAP pokaže izboljšavo hibridnega pristopa, ki uporablja metapodatke o izvorni organizaciji za 2,1 % pri vrednosti metrike MRR in za 4,7 % pri vrednosti metrike MAP. Iz rezultatov primerjave zaključimo, da upoštevanje metapodatkov o izvorni organizaciji dokumentov izboljša delovanje hibridnega pristopa za priporočanje vrstilcev UDK glede na izbrane metrike.



(a) HR@K

(b) NDCG@K

(c) MRR

(d) MAP

Slika 4.7: Primerjava povprečnih vrednosti metrik HR@K, NDCG@K, MRR in MAP za pristopa v primerjavi 3.

Za vsako metriko smo statistično značilne razlike preverili z Wilcoxonovim statističnim testom z mejo statistične značilnosti = 0;05. Ničelna hipoteza H_0 izraža stanje brez razlik med pristopoma v primerjavi. Na podlagi dobljenih vrednosti statistične značilno-sti p (tabela 4.7) zavržemo ničelno hipotezo H_0 in zaključimo, da izvedba predlagane-ga hibridnega pristopa, ki upošteva metapodatke o izvorni organizaciji dokumenta vrne statistično značilno boljše rezultate od izvedbe predlaganega hibridnega pristopa, ki teh metapodatkov ne upošteva.

Tabela 4.7: Dobljene vrednosti statistične značilnosti p v primerjavi 3.

	HR@1	**HR@3**	**HR@5**	**NDCG@1**	**NDCG@3**	**NDCG@5**	**MRR**	**MAP**
p	0,003	0,002	0,007	0,021	0,005	0,004	0,035	0,023

4.2.4 Primerjava 4

V tej dodatni primerjavi se primerjata izvedbi predlaganega hibridnega pristopa za priporočanje vrstilcev UDK z in brez vključenih postopkov naknadnih obdelav (slika 4.8). V tabeli 4.8 so podane pripadajoče dosežene povprečne vrednosti metrik HR@K, NDCG@K, MRR in MAP.

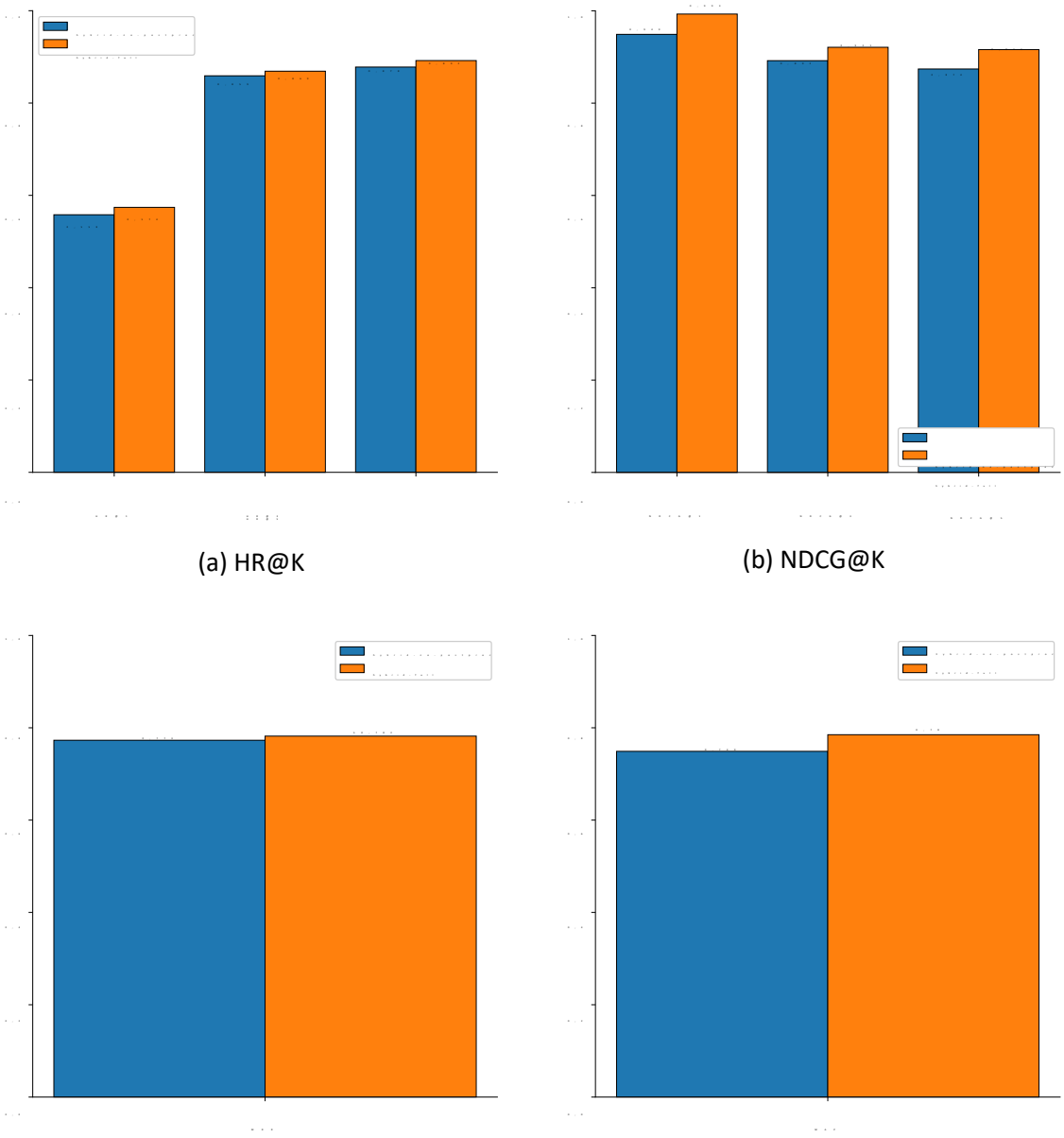
Tabela 4.8: Dosežene povprečne vrednosti in standardni odkloni (v oklepajih) metrik HR@K, NDCG@K, MRR in MAP za pristopa v primerjavi 4.

	HR@K			**NDCG@K**				
Pristop	K = 1	K = 3	K = 5	K = 1	K = 3	K = 5	**MRR**	**MAP**
hybrid-no-postproc	0,558 (0,133)	0,859 (0,119)	0,878 (0,132)	0,949 (0,114)	0,892 (0,112)	0,874 (0,113)	0,773 (0,101)	0,749 (0,105)
hybrid-full	**0,574** (0,131)	**0,869** (0,128)	**0,892** (0,136)	**0,993** (0,108)	**0,921** (0,115)	**0,916** (0,118)	**0,782** (0,104)	**0,785** (0,098)

Predlagan pristop hibridnega priporočanja, ki vključuje postopke naknadne obdelave se po metriki HR@K za vse vrednosti K izkaže bolje od predlaganega pristopa, ki ne vključuje postopkov naknadne obdelave. V najstrožji obliki metrike (K = 1) predlagan pristop, ki vključuje postopke naknadne obdelave, doseže za 1,6 % višjo vrednost. V manj strogih oblikah metrike (K = 3 in K = 5) predlagan hibridni pristop, ki vključuje postopke naknadne obdelave, doseže za 1 % in 1,4 % višjo vrednost.

Po metriki NDCG@K se predlagan pristop hibridnega priporočanja, ki vključuje postopke naknadne obdelave, za vse vrednosti K izkaže za boljšega. V najstrožji obliki metrike (K = 1) pristop, ki vključuje postopke naknadne obdelave, doseže 4,4 % višjo vrednost, v manj strogih oblikah metrike (K = 3 in K = 5) pa 2,9 % in 4,2 % višjo vrednost.

Primerjava povprečnih vrednosti metrik MRR in MAP pokaže izboljšavo hibridnega pristopa, ki vključuje postopke naknadne obdelave, za 0,9 % pri vrednosti metrike MRR in za 3,6 % pri vrednosti metrike MAP. Iz rezultatov primerjave zaključimo, da vključevanje postopkov naknadne obdelave izboljša delovanje hibridnega pristopa za priporočanje vrstilcev UDK glede na izbrane metrike.



(a) HR@K

(b) NDCG@K

(c) MRR

(d) MAP

Slika 4.8: Primerjava povprečnih vrednosti metrik HR@K, NDCG@K, MRR in MAP za pristopa v primerjavi 4.

4.3 Analiza parametrov predlaganega pristopa

V predlaganem hibridnem pristopu za priporočanje vrstilcev UDK nastopa več parametrov, ki se uporabijo v različnih posameznih komponentah. Predlagan hibridni pristop vpeljuje nove parametre, z uporabo nekaterih že obstoječih metod pa posledično uporablja tudi njihove parametre. Vrednosti teh parametrov so bile predmet preteklih raziskav, nekatere pa so se uveljavile tudi kot standardne vrednosti za te parametre. Zaradi tega smo se med razvojem predlaganega hibridnega pristopa na podlagi teh raziskav odločili za uporabo priporočenih vrednosti teh parametrov (tabela 4.9).

Tabela 4.9: Parametri predlaganega hibridnega pristopa s priporočenimi vrednostmi.

Parameter	Priporočena vrednost	Reference	Uporaba v predlaganem hibridnem pristopu
k_1	1;2	[5], [34], [134]	pridobivanje začetnega seznama (BM25)
b	0;75		
‘	0;5	[135], [136]	predlagan večznačni klasifikator
p	3	[128], [137]	Jaro-Winklerjeva podobnost (prvi kaskadni korak)
	0;1		

Priporočeni vrednosti parametrov k_1 in b sta zanimivi za nadaljnjo podrobnejšo analizo, saj se lahko zaradi jezika besedila razlikujeta od priporočenih vrednosti. Čeprav se priporočeni vrednosti pogosto uporabljata kot privzeti vrednosti v primeru kateregakoli jezika [5], so možne optimizacije, ki vodijo do vrednosti parametrov prilagojenih na specifičen korpus besedil. V literaturi lahko med drugim zasledimo poskuse optimizacije teh parametrov z mrežnim iskanjem (ang. grid search) [138] in diferencialno evolucijo (ang. differential evolution) [139].

Priporočena vrednost parametra je praktično standard kjerkoli v literaturi, saj je to smiselna vrednost za prag kadar model izvaja odločitve na podlagi verjetnosti. Tako je priporočena vrednost dobro izhodišče za nadaljnje optimizacije, ki se največkrat izvajajo v primerih neuravnoteženih podatkovnih množic [135].

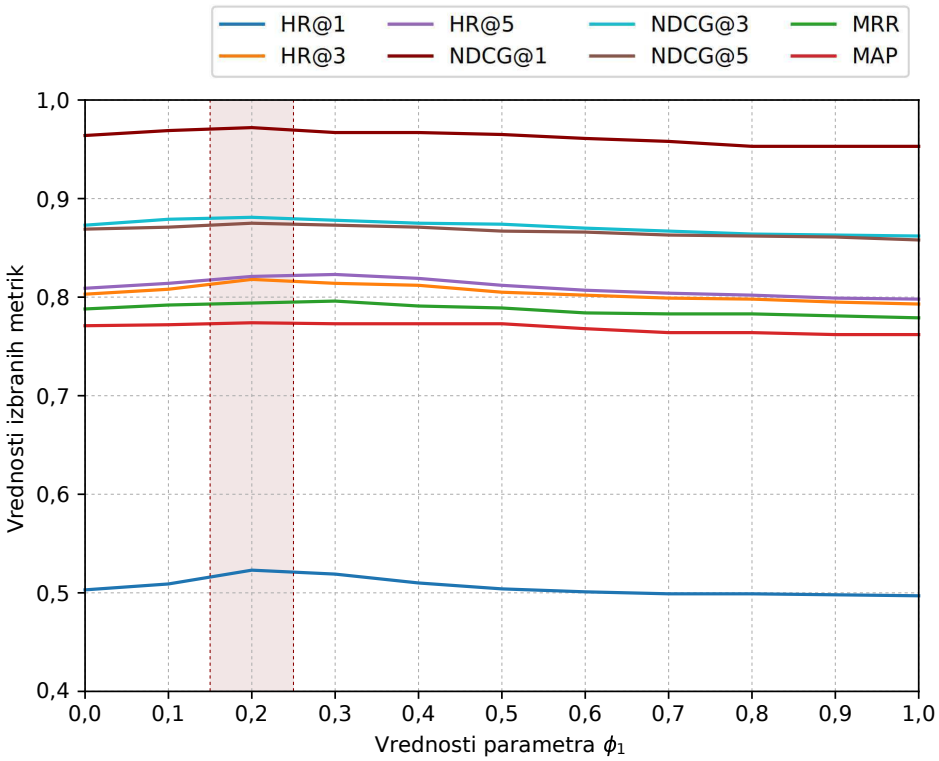
Priporočeni vrednosti parametrov ‘ in p, ki nastopita pri uporabi Jaro-Winklerjeve podobnosti v prvem kaskadnem koraku, se izkažeta za ustrezni za uporabo z vrstilci UDK. Parameter ‘, ki je dolžina predpone (tj. začetni del vrstilca UDK), z vrednostjo 3 upošteva ustrezen del hierarhije UDK. Vrstilci UDK se po vsakih treh decimalnih številih ločijo s znakom pika (“.”), ki eksplicitno določa nadaljnjo razširitev področij na podpodročja. Tako se pri izračunu Jaro-Winklerjeve podobnosti uporabi ustrezno število znakov (števil) vrstilca UDK, ki določajo vrhnja področja hierarhije UDK in njihova podpodročja do globine 3. Pri tem se upošteva, da sta si dva vrstilca UDK bolj podobna, če vsebujeta enake znake na začetku niza v smeri iz leve proti desni. Po predlogu avtorja Jaro-Winklerjeve podobno-sti [128] je sicer največja dovoljena vrednost parametra ‘ = 4, vrednost parametra p pa naj ne bi presegala vrednosti $\frac{1}{'}$. V nasprotnem primeru se lahko zgodi, da vrednost Jaro-Winklerjeve podobnosti preseže 1 [140], kar ni zaželjeno.

V analizi smo se osredotočili na parametre $_1$, $_2$ in , ki jih vpeljuje predlagan hibridni pristop in se uporabljajo v postopkih naknadne obdelave. Parameter $_1$ se uporabi pri preurejanju na podlagi vrhnjega področja v izračunu novih ocen vrstilcev UDK v seznamu

priporočil in tako vpliva na vrstni red preurejenih vrstilcev UDK. Podobno se parameter $_2$ uporabi pri preurejanju na podlagi specifičnosti. Parameter predstavlja odstotek ocene najvišje uvrščenega vrstilca UDK v seznamu priporočil in se uporabi pri rezanju seznama. Ta parameter vpliva na vrednost praga T , s katerim se izvede rezanje seznama. Vrednosti vseh treh parametrov so definirane na intervalu [0; 1].

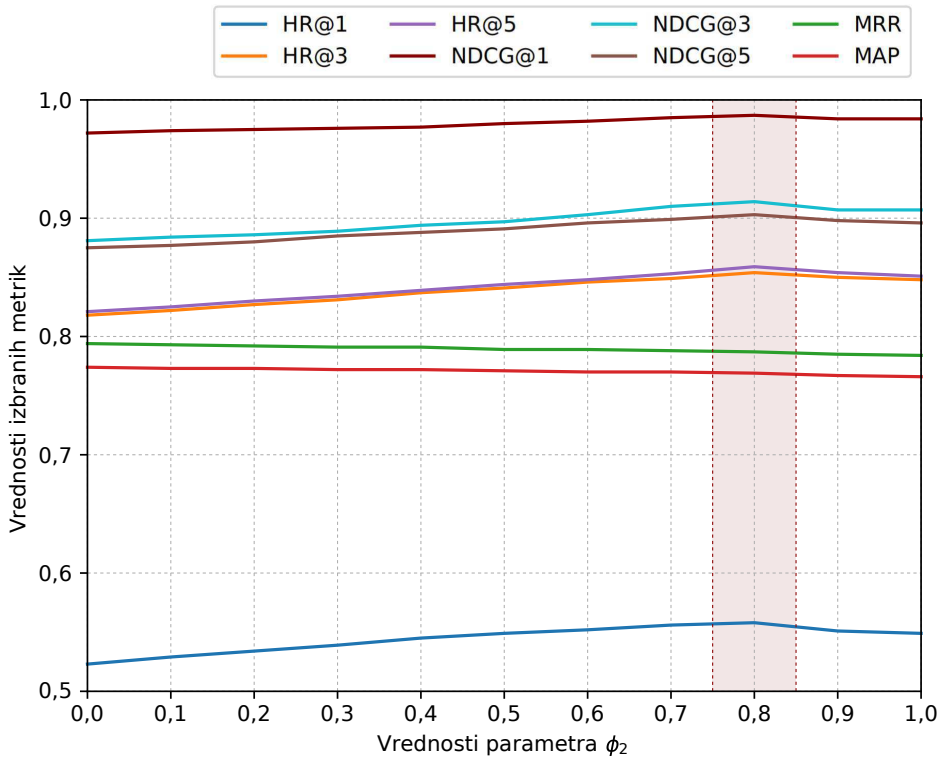
Ker se v postopkih naknadne obdelave preurejanje na podlagi vrhnjega področja izvede pred preurejanjem na podlagi specifičnosti, slednje pa se izvede pred rezanjem seznama, smo z validacijsko podatkovno množico in izbranimi metrikami najprej poiskali najboljšo vrednost parametra $_1$. To vrednost smo nato uporabili pri iskanju najboljše vrednosti parametra $_2$. Zatem smo obe dobljeni vrednosti za $_1$ in $_2$ uporabili pri iskanju najboljše vrednosti parametra . Vrednosti vseh parametrov smo na intervalu [0; 1] spreminjali s korakom 0;1.

Na sliki 4.9 so podani grafi vrednosti metrik HR@K, NDCG@K (K = [1; 3; 5]), MRR in MAP pri različnih vrednostih parametra $_1$. Uporaba vrednosti parametra $_1$ = 0 pomeni, da obstoječi rang vrstilca v seznamu ne vpliva na vrednost nove ocene, kot je razvidno iz enačbe 3.5. Vidimo, da se najbolje obnese izbira vrednosti parametra $_1$ = 0; 2.



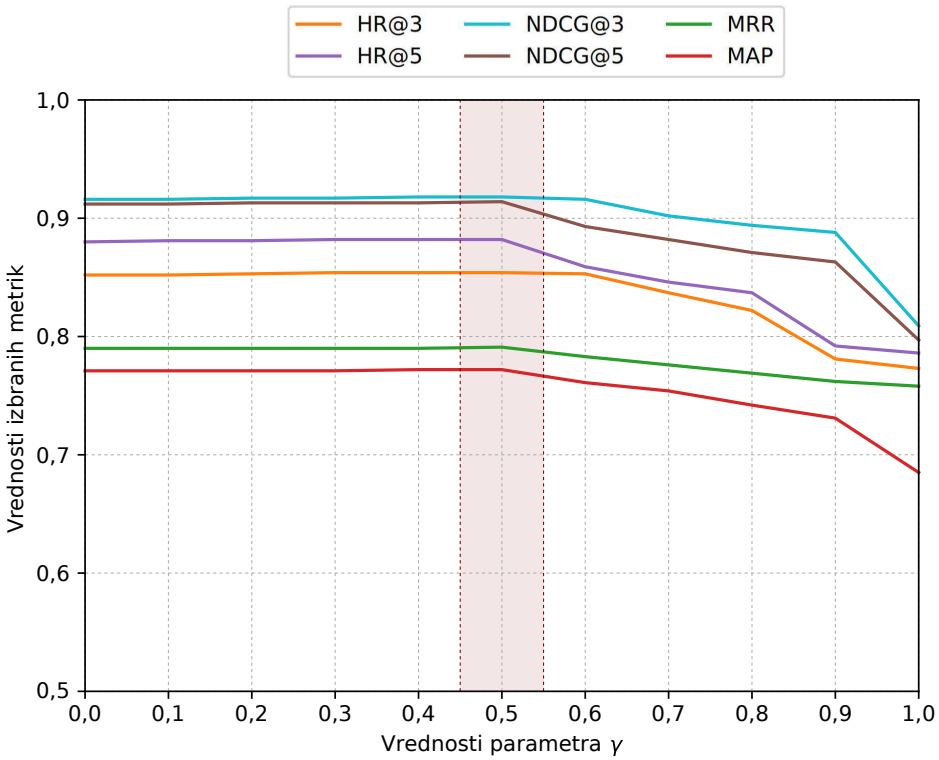
Slika 4.9: Vrednosti izbranih metrik pri različnih vrednostih parametra $_1$.

Na sliki 4.10 so podani grafi vrednosti metrik HR@K, NDCG@K (K = [1; 3; 5]), MRR in MAP pri različnih vrednostih parametra $_2$. Uporaba vrednosti parametra $_2$ = 0 pomeni, da se preurejanje na podlagi specifičnosti ne zgodi, kot je razvidno iz enačbe 3.6. Vidimo, da se najbolje obnese izbira vrednosti parametra $_2$ = 0;8.



Slika 4.10: Vrednosti izbranih metrik pri različnih vrednostih parametra $_2$.

Na sliki 4.11 so podani grafi vrednosti metrik HR@K, NDCG@K (K = [3; 5]), MRR in MAP pri različnih vrednosti parametra . Metrik HR@1 in NDCG@1 v tem primeru nismo upo-števali, saj delujeta le nad seznami velikosti 1, kjer rezanje seznama ni smiselno. Kot je razvidno iz enačbe 3.7, uporaba vrednosti parametra = 0 pomeni, da se rezanje sezna-ma ne bo izvedlo. Uporaba vrednosti parametra = 1 pomeni, da se bo zgodilo rezanje vseh vrstilcev UDK v seznamu, razen najvišje uvrščenega. Vidimo, da se najbolje obnese izbira vrednosti parametra = 0;5. Opazimo tudi, da je ta parameter smiselno nasta-vljati na vrednosti 0;1 0;5, saj v nasprotnem primeru prihaja do prekomernega rezanja seznama, kar se odraža v nižjih vrednostih izbranih metrik.



Slika 4.11: Vrednosti izbranih metrik pri različnih vrednostih parametra .

5 Interpretacija rezultatov in razprava

V primerjavi 1 primerjamo predlagan hibridni pristop za priporočanje vrstilcev UDK s pristopom, ki temelji na vsebinskem filtriranju. Pri tem smo za vsebinsko filtriranje uporabili rangirno funkcijo BM25. Rezultati primerjave pokažejo, da se glede na izbrane metrike bolje izkaže predlagan hibridni pristop. Razlike v vrednostih metrike HR@K med pristopoma kažejo na to, da se v končnih seznamih priporočil, ki jih tvori predlagan hibridni pristop, pojavi več pravilnih vrstilcev UDK. To velja tako za zelo omejeno dolžino (K = 1), kot za bolj standardno dolžino (K = 3 in K = 5) seznama priporočil. Smiselno je tudi, da se z daljšim seznamom poveča število vrnjenih pravilnih vrstilcev UDK.

Razlike v vrednostih metrike NDCG@K med pristopoma nakazujejo na to, da predlagan hibridni pristop tvori sezname priporočil z boljšim vrstnim redom. Pri tej metriki je pričakovano, da se z daljšim seznamom priporočil vrednost metrike manjša, saj je v teh primerih potrebno pravilno rangirati več vrstilcev UDK v seznamu priporočil. Opazimo, da je rangirna funkcija BM25 po tej metriki sicer dobra, vendar je predlagan hibridni pri-stop boljši. Pri metriki MRR je vidna izboljšava v korist predlaganega hibridnega pristopa. Ker ta metrika meri, kako dobro pristop uvrsti pravilne vrstilce UDK na začetek seznama, lahko ugotovimo, da so v seznamih priporočil predlaganega hibridnega pristopa pravilni vrstilci UDK večkrat uvrščeni bolje v primerjavi s pristopom z rangirno funkcijo BM25. Po pregledu vrednosti metrike MAP pridemo do podobnih ugotovitev. Na podlagi teh rezultatov lahko zaključimo, da predlagan hibridni pristop tvori bolj kvalitetne sezname priporočenih vrstilcev UDK.

V sklopu primerjave 2 primerjamo predlagan hibridni pristop za priporočanje vrstilcev UDK s pristopom, ki temelji na večrazredni klasifikaciji. Pri tem smo izbrali večrazredni klasifikator z metodo podpornih vektorjev (MC-SVM), ki se je v sorodnih delih izkazal za najboljšega. Dodatno smo primerjali, kako se obnesejo pristopi, ki temeljijo na večznačni klasifikaciji. Pri tem smo izbrali prilagojen večznačni klasifikator z metodo podpornih vektorjev (ML-SVM), večznačni klasifikator XR-Transformer, in večznačni klasifikator, ki uporablja jezikovni model SloBERTa (ML-UDC-SloBERTa). Rezultati primerjave pokažejo, da se za izbrane metrike bolje izkaže predlagan hibridni pristop. Razlike v vrednostih me-

trike HR@K nakazujejo na to, da se v končnih seznamih priporočil, ki jih tvori predlagan hibridni pristop, pojavi več pravilnih vrstilcev UDK.

Najslabše se obnese večrazredni klasifikator z metodo podpornih vektorjev (MC-SVM). To je posledica delovanja takšnega klasifikatorja, saj je klasifikator namenjen klasifikaciji v en razred, s prilagoditvijo pa vračamo K razredov, ki so v našem primeru vrstilci UDK. Večznačni klasifikator z metodo podpornih vektorjev (ML-SVM) se izkaže za boljšega od večrazrednega klasifikatorja z metodo podpornih vektorjev (MC-SVM), kljub temu pa se glede na metrike izkaže za slabšega tako od večznačnih klasifikatorjev XR-Transformer in ML-UDC-SloBERTa, kot tudi od predlaganega hibridnega pristopa. Glede na to, da ta klasifikator uporablja značilke pridobljene z utežno shemo `tf-idf`, predlagani večznačni klasifikator pa uporablja kontekstne vložitve, lahko ugotovimo, da so kontekstne vložitve boljša metoda za transformacijo besedila v vektorski prostor, saj bolje zajamejo pomen besedila in medsebojno povezanost besed v njem. Posledično se to vidi pri klasifikaciji besedil v ustrezne oznake, ki so v našem primeru vrstilci UDK.

Predlagan hibridni pristop uporablja tako utežno shemo `tf-idf`, kot tudi kontekstne vložitve. Glede na rezultate primerjave 1 lahko vidimo, da so na področju priporočilnih sistemov metode za informacijsko poizvedovanje in iskalnike, ki temeljijo na utežni shemi `tf-idf`, zaenkrat boljše kot metode, ki temeljijo na globokih nevronskih mrežah. Podobno ugotovitev najdemo tudi v literaturi [127]. Iz tega dejstva in na podlagi rezultatov primerjave 2 lahko zaključimo, da predlagan hibridni pristop tvori bolj kvalitetne sezname priporočenih vrstilcev UDK v primerjavi z večrazrednim klasifikatorjem z metodo podpornih vektorjev in obeh večznačnih klasifikatorjev, ki sta bila del primerjave.

V primerjavi 3 primerjamo predlagan hibridni pristop za priporočanje vrstilcev UDK s pristopom, ki se razlikuje samo v tem, da ne uporablja metapodatkov o izvornih organizacijah dokumentov. Iz rezultatov te primerjave je razvidno, da upoštevanje metapodatkov o izvornih organizacijah dokumentov vodi v bolj kvalitetne sezname priporočenih vrstilcev UDK. Predlagan hibridni pristop uporablja vnaprej pripravljene sezname značilnih vrstilcev UDK za vsako izvorno organizacijo. Seznami značilnih vrstilcev UDK za vsako izvorno organizacijo se sicer prekrivajo v nekaterih vrstilcih UDK, vendar so v splošnem zelo raznoliki. Še več, raznoliki so do te mere, da zajemajo le del hierarhije UDK, ki je kontekstno vezan na področje znanosti, s katerim se izvorna organizacija ukvarja. To je smiselno, saj so izvorne organizacije generatorji domensko specifičnih dokumentov.

Za primer vzemimo fakulteto za računalništvo in pravno fakulteto. Če obravnavamo dokument, ki je nastal na fakulteti za računalništvo, je večja verjetnost, da bo označen z vrstilci UDK, ki v hierarhiji UDK opisujejo področje računalništva. Prav tako je manjša verjetnost, da bo tak dokument označen z vrstilci UDK, ki v hierarhiji UDK opisujejo področje prava. Seveda velja tudi obratno, če bi obravnavali dokument, ki je nastal na pravni fakulteti. Med postopkom preurejanja v predlaganem hibridnem pristopu za izvorne organizacije značilni vrstilci UDK pripomorejo k boljši uvrstitvi smiselnih vrstilcev v seznamu priporočenih vrstilcev UDK.

V sklopu dodatne primerjave 4 primerjamo vpliv vključevanja postopkov naknadne obdelave na predlagan hibridni pristop. Rezultati pokažejo, da z uporabo postopkov naknadne obdelave znotraj predlaganega hibridnega pristopa dosežemo manjše izboljšave. Glede na to, da so največje izboljšave vidne pri metriki NDCG@K, zaključimo, da postopki naknadne obdelave najbolj vplivajo na vrstni red vrstilcev UDK v seznamih priporočil. To je tudi pričakovano, saj v postopkih naknadne obdelave nastopita preurejanje na podlagi vrhnjega področja in preurejanje na podlagi specifičnosti.

S preurejanjem na podlagi vrhnjega področja preurejamo seznam priporočil tako, da višje rangiramo tiste vrstilce UDK, ki spadajo v dominantno vrhnje področje v vmesnem seznamu priporočil. Glede na to, da so dokumenti največkrat vezani na eno domensko specifično področje, s tem preurejanjem še dodatno povečamo vrednost ocen tistih vrstilcev UDK, ki v hierarhiji UDK to domensko specifično področje definirajo in opisujejo. To velja tudi za dokumente z interdisciplinarno vsebino, saj se v teh primerih največkrat prekrivata zgolj dve domensko specifični področji.

S preurejanjem na podlagi specifičnosti preurejamo seznam priporočil tako, da višje rangiramo tiste vrstilce UDK, ki so daljši in se tako v hierarhiji UDK nahajajo globlje. Doprinos tega postopka naknadne obdelave je manjši kot doprinos preurejanja na podlagi vrhnjega področja, saj smo omejeni na prosto dostopni katalog UDK, v katerem ni na voljo vseh vrstilcev UDK. V izbrani podatkovni zbirki vidimo, da je večina vrstilcev v izrazih UDK kataloziranih dokumentov tudi prisotnih v prosto dostopnem katalogu. Iz tega sledi, da tudi knjižničarji, ki ročno določajo vrstilce in izraze UDK za dokumente, v veliki meri ne izbirajo tistih najbolj specifičnih vrstilcev. Kljub temu imajo pri svojem delu na voljo celoten katalog UDK in jim je to omogočeno. Obstaja torej možnost povečanja doprinosa tega postopka naknadne obdelave, če ne bi bili omejeni na prosto dostopni katalog UDK.

Rezanje seznama kot postopek naknadne obdelave je lahko v določenih primerih tudi nezaželjen, če iz seznama priporočil odstrani tudi pravilne vrstilce UDK. V tem primeru je potrebno podrobneje analizirati specifikacije priporočilnega sistema in glede na to določiti stopnjo rezanja. Če bi uporabniku prikazali le nekaj najboljših priporočil, potem rezanje seznama kot postopek naknadne obdelave morda niti ni potreben, saj se bo uporabniku prikazal le del seznama in rezanje seznama izvajamo pravzaprav na nivoju uporabniškega vmesnika. V praksi se rezanje seznama velikokrat izvaja kot odstranjevanje določenih elementov v seznamu priporočil zaradi določenih omejitev uporabnika. Med te spadajo geografska lokacija, posebne uporabniške nastavitve (npr. uporabnik ne želi videti vsebin povezanih z X) in že videna vsebina. Kot nadgradnjo postopka rezanja seznama v predlaganem pristopu bi bilo v sklopu nadaljnjega dela zanimivo preučiti druge načine odstranjevanja vrstilcev UDK iz seznama priporočil.

5.1 Potrjevanje zastavljenih hipotez

Glavni cilj doktorske disertacije je bil zasnovati in razviti hibridni pristop priporočanja vrstilcev UDK, ki temelji na metodah vsebinskega filtriranja in se lahko uporabi za polavtomatsko določanje ustreznih vrstilcev UDK, brez omejitev na področje znanosti ali globino hierarhije UDK. V sklopu doktorske disertacije smo predlagan hibridni pristop priporočanja vrstilcev UDK primerjali z drugimi pristopi, ki temeljijo zgolj na vsebinskem filtriranju, ali večrazredni klasfikaciji. Za potrebe primerjave smo vzpostavili eksperimentalno okolje, v katerem smo definirali scenarije primerjave, podatkovno zbirko in metrike.

Za priporočanje vrstilcev UDK je pomembno, da so seznami priporočenih vrstilcev UDK smiselni in dosegajo stopnjo kvalitete, ki je primerna za implementacijo v produkciji. Takšno priporočanje lahko izvedemo z metodami vsebinskega filtriranja, pri tem pa uporabimo rangirno funkcijo BM25, ki je ena izmed najpogosteje uporabljenih metod na področju in se zelo dobro izkaže tudi v produkciji. Zato smo zastavili naslednjo hipotezo:

H1: *Predlagani hibridni pristop priporočanja vrstilcev univerzalne decimalne klasifikacije vrne statistično značilno boljše rezultate od pristopov priporočanja, ki temeljijo izključno na vsebinskem filtriranju, glede na metriki HR@K in NDCG@K za standardne vrednosti parametra K ter metriki MAP in MRR.*

Glede na rezultate primerjave 1 (podpodpoglavje 4.2.1), kjer primerjamo predlagan hibridni pristop in pristop, ki uporablja rangirno funkcijo BM25, zaključimo, da predlagan

hibridni pristop vrne statistično značilno boljše rezultate od pristopa priporočanja, ki temelji izključno na vsebinskem filtriranju. Glede na rezultate Wilcoxonovega statističnega testa hipotezo H1 potrdimo.

Priporočanje vrstilcev UDK lahko izvedemo tudi z metodami večrazredne klasifikacije. Pri tem se je v sorodnih delih za najboljši pristop izkazal večrazredni klasifikator z metodo podpornih vektorjev. Zato smo zastavili naslednjo hipotezo:

H2: *Predlagani hibridni pristop priporočanja vrstilcev univerzalne decimalne klasifikacije vrne statistično značilno boljše rezultate od pristopov priporočanja, ki temeljijo izključno na večrazredni klasifikaciji, glede na metriki HR@K in NDCG@K za standardne vrednosti parametra K ter metriki MAP in MRR.*

Glede na rezultate primerjave 2 (podpodpoglavje 4.2.2), kjer primerjamo predlagan hibridni pristop in pristop, ki uporablja večrazredni klasifikator z metodo podpornih vektorjev, zaključimo, da predlagan hibridni pristop vrne statistično značilno boljše rezultate od pristopa priporočanja, ki temelji izključno na večrazredni klasifikaciji. Glede na rezultate Wilcoxonovega statističnega testa hipotezo H2 potrdimo.

Pri izvedbi hibridnega pristopa za priporočanje vrstilcev UDK lahko upoštevamo tudi metapodatke o izvorni organizaciji dokumenta. Vpliv teh metapodatkov je zaradi domenske specifičnosti dokumentov lahko pomemben. Zato smo zastavili naslednjo hipotezo:

H3: *Upoštevanje metapodatkov o izvoru elektronskega dokumenta statistično značilno izboljša delovanje hibridnega pristopa priporočanja vrstilcev univerzalne decimalne klasifikacije glede na metriki HR@K in NDCG@K za standardne vrednosti parametra K ter metriki MAP in MRR.*

Glede na rezultate primerjave 3 (podpodpoglavje 4.2.3), kjer primerjamo izvedbo predlaganega hibridnega pristopa z upoštevanjem metapodatkov o izvorni organizaciji dokumenta in izvedbo predlaganega hibridnega pristopa brez upoštevanja metapodatkov o izvorni organizaciji dokumenta, zaključimo, da izvedba predlaganega hibridnega pristopa, ki upošteva metapodatke o izvorni organizaciji dokumenta vrne statistično značilno boljše rezultate od izvedbe predlaganega hibridnega pristopa, ki teh metapodatkov ne upošteva. Glede na rezultate Wilcoxonovega statističnega testa hipotezo H3 potrdimo.

S potrditvijo vseh treh zastavljenih hipotez, potrdimo tudi tezo doktorske disertacije, ki se glasi:

S hibridnim pristopom priporočanja, ki kombinira metodologije vsebinskega filtriranja in večznačne klasifikacije, izboljšamo določanje vrstilcev univerzalne decimalne klasifikacije v primerjavi z obstoječimi pristopi.

6 Zaključek

V doktorski disertaciji smo predstavili nov pristop za hibridno priporočanje vrstilcev univerzalne decimalne klasifikacije. Predlagali smo pristop, ki s kaskadno hibridizacijo združuje rangirno funkcijo BM25 in večznačni klasfikator. Za razliko od sorodnih del je rezultat našega predlaganega pristopa seznam priporočenih vrstilcev UDK. Z uporabo kaskadne hibridizacije smo omogočili fleksibilnost pristopa, saj lahko v takšnem tipu hibridizacije poljubno dodajamo ali odvzemamo metode, ki izvajajo preurejanje seznamov priporočil.

Predstavili smo univerzalno decimalno klasifikacijo, ki je primarni knjižnični sistem v Sloveniji. Podali smo pregled uveljavljenih metod na področjih obdelave naravnega jezika in priporočilnih sistemov, pri tem pa se osredotočili na metodo vsebinskega filtriranja, rangirno funkcijo BM25 in arhitekturo globokih nevronskih mrež transformer, ki je osnova za velike jezikovne modele kot je BERT. Podrobno smo opisali postopek kaskadne hibridizacije, ki združuje metodo vsebinskega filtriranja in večznačni klasifikator. V nadaljevanju doktorske disertacije smo podrobneje opisali algoritem za pridobivanje začetnega seznama vrstilcev UDK, ki vodi v osnovni seznam priporočenih vrstilcev UDK. Ti se v nadaljevanju preurejajo znotraj dveh kaskadnih korakov, ki uporabljata večznačni klasifikator in metapodatke o izvornih organizacijah dokumentov. Pri učenju večznačnega klasifika-torja smo izvedli glajenje oznak, ki upošteva hierarhično topologijo UDK. Nazadnje smo predlagali tri postopke naknadne obdelave, ki se lahko poljubno vključijo v postopek priporočanja. Ti postopki so preurejanje na podlagi vrhnjega področja, preurejanje na podlagi specifičnosti in rezanje seznama. Z rezultati smo pokazali, da uporaba teh postopkov naknadne obdelave dodatno izboljša delovanje hibridnega priporočanja glede na metrike HR@K, NDCG@K, MRR in MAP. Z zasnovo, razvojem in vrednotenjem hibridnega priporočilnega sistema smo pokazali, da se preurejanje priporočil s kaskadno hibridizacijo izkaže za boljši pristop za priporočanje vrstilcev UDK kot do zdaj znani sorodni pristopi.

Uporaba predlaganega hibridnega pristopa za priporočanje vrstilcev UDK pa ni omejena zgolj na priporočanje vrstilcev UDK. Predlagan pristop vrača seznam vrstilcev UDK, ki dobro definirajo področje dokumenta. S tem je definirana tudi tematika in posledično lahko takšen pristop uporabimo pri detekciji podobnih vsebin v postopku izbire kandidatov za

preverjanje podobnosti. Na ta način ni potrebno primerjati novega dokumenta z vsakim dokumentom, temveč samo s tistimi, ki so po tematiki sorodni. S tem lahko dosežemo pohitritev delovanja detekcije podobnih vsebin, kar je ob konstantni rasti novih dokumentov zelo zaželjeno. Prav tako se določanje vrstilcev in preverjanje podobnosti izvajata sočasno ob dodajanju novih dokumentov v knjižnični sistem, kar je priročno za vključitev predlaganega pristopa v že obstoječe in uveljavljene knjižnične sisteme.

6.1 Izvirni prispevki k znanosti

V doktorski disertaciji smo predlagali hibridni pristop za priporočanje vrstilcev UDK, ki s kaskadno hibridizacijo združuje metode vsebinskega filtriranja in večznačne klasifikacije. Dodatno smo zasnovali tri postopke naknadne obdelave, ki se lahko vključijo v predlagani hibridni pristop priporočanja vrstilcev UDK. Kot ključne izpostavimo naslednje izvirne znanstvene prispevke (IZP):

- **IZP1**: Nov pristop za reševanje problema polavtomatskega določanja vrstilcev univerzalne decimalne klasifikacije elektronskim dokumentom (poglavje 3).

- **IZP2**: Razvoj postopkov rerangiranja znotraj kaskadnega tipa hibridnega priporočilnega sistema z namenom izboljšave postopka priporočanja (podpoglavji 3.4 in 3.5).

- **IZP3**: Razvoj metode za dinamično določanje števila osnovnih relevantnih dokumentov v prvem kaskadnem koraku priporočilnega sistema (podpoglavje 3.2).

- **IZP4**: Nova metoda glajenja oznak pri učenju klasifikacijskega modela, ki upošteva hierarhično topologijo univerzalne decimalne klasifikacije (podpodpoglavje 3.3.2).

- **IZP5**: Podrobna analiza parametrov kaskadnega tipa hibridnega priporočilnega sistema (podpoglavje 4.3).

VIRI IN LITERATURA

[1] A. Slavic, "UDC implementation: From library shelves to a structured indexing language," *International cataloguing and bibliographic control*, vol. 33, pp. 60–65, 2004.

[2] K. Yi, "Automated Text Classification Using Library Classification Schemes: Trends, Issues, and Challenges," *International cataloguing and bibliographic control*, vol. 36, pp. 78–82, 2007.

[3] J. Wang, "An extensive study on automated Dewey Decimal Classification," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 11, pp. 2269–2286, 2009.

[4] B. Lund and D. Agbaji, "Use of Dewey Decimal Classification by Academic Libraries in the United States," *Cataloging & Classification Quarterly*, vol. 56, no. 7, pp. 653–661, 2018.

[5] Manning, Christopher D. and Raghavan, Prabhakar and Schütze, Hinrich, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[6] C. J. Godby and J. Stuler, "The Library of Congress Classification as a Knowledge Base for Automatic Subject Categorization," in *Subject Retrieval in a Networked Environment*, pp. 163–169, 2003.

[7] E. Frank and G. W. Paynter, "Predicting Library of Congress Classifications from Library of Congress Subject Headings," *Journal of the American Society for Information Science and Technology*, vol. 55, pp. 214–227, Feb 2004.

[8] V. M. Nisha and K. R. Ashok, "Implementation on Text Classification Using Bag of Words Model," in *Proceedings of the Second International Conference on Emerging Trends in Science and Technologies For Engineering Systems (ICETSE-2019)*, 2019.

[9] B. Baharudin, L. H. Lee, and K. Khan, "A Review of Machine Learning Algorithms for Text-Documents Classification," *Journal of Advances in Information Technology*, vol. 1, pp. 4–20, 2010.

[10] K. Golub, J. Hagelbäck, and A. Ardö, "Automatic Classification of Swedish Metadata Using Dewey Decimal Classification: A Comparison of Approaches," *Journal of Data and Information Science*, vol. 5, no. 1, pp. 18–38, 2020.

[11] M. Kragelj and M. Kljajić Borštnar, "Automatic classification of older electronic texts into the Universal Decimal Classification–UDC," *Journal of Documentation*, vol. 77, pp. 755–776, Jan 2021.

[12] J. Schrumpf, F. Weber, and T. Thelen, "A Neural Natural Language Processing System for Educational Resource Knowledge Domain classification," in *DELFI 2021* (A. Kienle, A. Harrer, J. M. Haake, and A. Lingnau, eds.), (Bonn), pp. 283–288, Gesellschaft für Informatik e.V., 2021.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 6000–6010, Curran Associates Inc., 2017.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *ArXiv*, vol. abs/1810.04805, 2019.

[15] S. Vargas, M. Hristakeva, and K. Jack, "Mendeley: Recommendations for Researchers," in *RecSys '16 Proceedings of the 10th ACM Conference on Recommender Systems*, (Boston, MA, USA), pp. 365–365, 2016.

[16] P. Carlos, M. Jose Maria, and H.-V. Enrique, "A multi-disciplinar recommender system to advice research resources in University Digital Libraries," *Expert Systems with Applications*, vol. 36, no. 10, pp. 12520 – 12528, 2009.

[17] B. Joeran, A. Akiko, B. Corinna, and G. Bela, "Mr. DLib: Recommendations-as-a-Service (RaaS) for Academia," in *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pp. 1–2, Jun 2017.

[18] B. Xiaomei, W. Mengyang, L. Ivan, Y. Zhuo, K. Xiangjie, and X. Feng, "Scientific Paper Recommendation: A Survey," *IEEE Access*, vol. 7, pp. 9324–9339, 2019.

[19] M. Ojsteršek, J. Brezovnik, M. Kotar, M. Ferme, G. Hrovat, A. Bregant, and M. Borovič, "Establishing of a Slovenian open access infrastructure: a technical point of view," *Program*, vol. 48, no. 4, p. 394–412, 2014.

[20] M. Borovič, M. Ferme, J. Brezovnik, S. Majninger, K. Kac, and M. Ojsteršek, "Document Recommendations and Feedback Collection Analysis within the Slovenian Open-Access Infrastructure," *Information*, vol. 11, no. 11, 2020.

[21] I. Liao, W. Hsu, M. Cheng, and L. Chen, "A library recommender system based on a personal ontology model and collaborative filtering technique for English collections," *Electronic Library*, vol. 28, no. 3, pp. 386–400, 2010.

[22] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," *Expert Systems with Applications*, vol. 92, pp. 507 – 520, 2018.

[23] Y. Koren, "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, (New York, NY, USA), p. 426–434, Association for Computing Machinery, 2008.

[24] D. Billsus, M. J. Pazzani, and J. Chen, "A Learning Agent for Wireless News Access," in *Proceedings of the 5th International Conference on Intelligent User Interfaces*, IUI '00, (New York, NY, USA), p. 33–36, Association for Computing Machinery, 2000.

[25] R. J. Mooney and L. Roy, "Content-Based Book Recommending Using Learning for Text Categorization," in *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, (New York, NY, USA), p. 195–204, Association for Computing Machinery, 2000.

[26] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, pp. 331–370, Nov 2002.

[27] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intelligent Data Analysis*, vol. 21, pp. 1487–1524, 2017.

[28] I. Folasade Olubusola, F. Yetunde, and O. Bolanle Adefowoke, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015.

[29] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales, "Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks," *International Journal of Approximate Reasoning*, vol. 51, no. 7, pp. 785–799, 2010.

[30] A. S. Lampropoulos, P. S. Lampropoulou, and G. A. Tsihrintzis, “A Cascade-Hybrid Music Recommender System for Mobile Services Based on Musical Genre Classification and Personality Diagnosis,” *Multimedia Tools and Applications*, vol. 59, p. 241–258, Jul 2012.

[31] P. Bedi, P. Vashisth, P. Khurana, and Preeti, “Modeling user preferences in a hybrid recommender system using type-2 fuzzy sets,” in *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–8, 2013.

[32] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, “An Eficient Hybrid Music Recommender System Using an Incrementally Trainable Probabilistic Generative Model,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 435–447, 2008.

[33] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro, “A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition,” *Information Sciences*, vol. 180, no. 22, pp. 4290–4311, 2010.

[34] S. Robertson and H. Zaragoza, “The Probabilistic Relevance Framework: BM25 and Beyond,” *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[35] B. He and I. Ounis, “A Study of Parameter Tuning for Term Frequency Normalization,” in *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM ’03, (New York, NY, USA), pp. 10–16, ACM, 2003.

[36] B. He and I. Ounis, “Term Frequency Normalisation Tuning for BM25 and DFR Models,” in *Advances in Information Retrieval* (D. E. Losada and J. M. Fernández-Luna, eds.), (Berlin, Heidelberg), pp. 200–214, Springer Berlin Heidelberg, 2005.

[37] Y. Lv and C. Zhai, “Adaptive Term Frequency Normalization for BM25,” in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM ’11, (New York, NY, USA), pp. 1985–1988, ACM, 2011.

[38] Y. Lv and C. Zhai, “Lower-bounding Term Frequency Normalization,” in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM ’11, (New York, NY, USA), pp. 7–16, ACM, 2011.

[39] A. Trotman, A. Puurula, and B. Burgess, “Improvements to BM25 and Language Models Examined,” in *Proceedings of the 2014 Australasian Document Computing Symposium*, ADCS ’14, (New York, NY, USA), pp. 58:58–58:65, ACM, 2014.

[40] L. Derczynski, “Complementarity, F-score, and NLP Evaluation,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, (Portorož, Slovenia), pp. 261–266, European Language Resources Association (ELRA), May 2016.

[41] D. Hand and P. Christen, “A note on using the F-measure for evaluating record linkage algorithms,” *Statistics and Computing*, vol. 28, no. 3, pp. 539–547, 2018.

[42] S. Rendle, L. Zhang, and Y. Koren, “On the Dificulty of Evaluating Baselines: A Study on Recommender Systems,” *ArXiv*, vol. abs/1905.01395, 2019.

[43] M. Bogaert, J. Lootens, D. V. den Poel, and M. Ballings, “Evaluating multi-label classifiers and recommender systems in the financial service sector,” *European Journal of Operational Research*, vol. 279, no. 2, pp. 620 – 634, 2019.

[44] M. Borovič, M. Ferme, J. Brezovnik, S. Majninger, A. Bregant, G. Hrovat, and M. Ojsteršek, “The OpenScience Slovenia metadata dataset,” *Data in Brief*, vol. 28, p. 104942, 2020.

[45] UDC Consortium (UDCC), *Multilingual Universal Decimal Classification Summary (UDCC Publication No. 088)*, 2012.

[46] W. Rayward, *From the index card to the World City: knowledge organization and visualization in the work and ideas of Paul Otlet*, pp. 1–41. Egon Verlag, Jan 2013.

[47] I. C. McIlwaine, “The Universal Decimal Classification: Some factors concerning its origins, development, and influence,” *Journal of the American Society for Information Science*, vol. 48, no. 4, pp. 331–339, 1997.

[48] V. Broughton, *Essential Classification*. London, UK: Facet, 2015.

[49] “Universal Decimal Classification Consortium.” https://udcc.org/index.php.

[50] “Collections indexed by UDC.” https://udcc.org/index.php/site/page?view=collections.

[51] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press, 1999.

[52] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[53] D. M. E.-D. M. Hussein, “A survey on sentiment analysis challenges,” *Journal of King Saud University - Engineering Sciences*, vol. 30, no. 4, pp. 330–338, 2018.

[54] A. Pelicon, M. Pranjić, D. Miljković, B. Škrlj, and S. Pollak, “Zero-Shot Learning for Cross-Lingual News Sentiment Classification,” *Applied Sciences*, vol. 10, no. 17, 2020.

[55] A. Pelicon, R. Shekhar, B. Škrlj, M. Purver, and S. Pollak, “Investigating cross-lingual training for offensive language detection,” *PeerJ Computer Science*, vol. 7, p. e559, Jun 2021.

[56] V. Yadav and S. Bethard, “A Survey on Recent Advances in Named Entity Recognition from Deep Learning models,” in *Proceedings of the 27th International Conference on Computational Linguistics*, (Santa Fe, New Mexico, USA), pp. 2145–2158, Association for Computational Linguistics, Aug 2018.

[57] S. Pawar, G. K. Palshikar, and P. Bhattacharyya, “Relation Extraction : A Survey,” 2017.

[58] H. Zhang, X. Zhao, and Y. Song, “A Brief Survey and Comparative Study of Recent Development of Pronoun Coreference Resolution in English,” in *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*, (Punta Cana, Dominican Republic), pp. 1–11, Association for Computational Linguistics, Nov 2021.

[59] S. Singh and H. Beniwal, “A survey on near-human conversational agents,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, Part A, pp. 8852–8866, 2022.

[60] S. Büttcher, C. Clarke, and G. V. Cormack, *Information Retrieval: Implementing and Evaluating Search Engines*. The MIT Press, 2010.

[61] M. Caballero, “A Brief Survey of Question Answering Systems,” *International Journal of Artificial Intelligence and Applications*, vol. 12, pp. 01–07, Sep 2021.

[62] Y. Wilks, “Themes in the work of Margaret Masterman,” in *Proceedings of Translating and the Computer 10: The translation environment 10 years on*, (London, UK), Aslib, Nov 1988.

[63] J. Weizenbaum, “ELIZA — a Computer Program for the Study of Natural Language Communication between Man and Machine,” *Commun. ACM*, vol. 9, p. 36–45, Jan 1966.

[64] M. Lesk, “Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone,” in *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC ’86, (New York, NY, USA), p. 24–26, Association for Computing Machinery, 1986.

[65] K. Sparck Jones, *Synonymy and Semantic Classification*. GBR: Edinburgh University Press, 1986.

[66] K. Sparck Jones, “Index term weighting,” *Information Storage and Retrieval*, vol. 9, no. 11, pp. 619–633, 1973.

[67] K. Sparck Jones, *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*, p. 132–142. GBR: Taylor Graham Publishing, 1988.

[68] F. Almeida and G. Xexéo, “Word Embeddings: A Survey,” 2023.

[69] A. Graves, “Generating Sequences With Recurrent Neural Networks,” 2014.

[70] A. Radford and K. Narasimhan, “Improving language understanding by generative pre-training,” in *public OpenAI papers*, 2018.

[71] F. Jáñez-Martino, E. Fidalgo, S. González-Martínez, and J. Velasco-Mata, “Classification of Spam Emails through Hierarchical Clustering and Supervised Learning,” 2020.

[72] M. F. Abdul Kadir, A. Abidin, M. A. Mohamed, and N. Abd. Hamid, “Spam detection by using machine learning based binary classifier,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, p. 310, Apr 2022.

[73] M. Sethi, N. Tyagi, P. S. Kalsi, and P. Atchuta Rao, “Deep Learning-based Binary Classification for Spam Detection in SMS Data: Addressing Imbalanced Data with Sampling Techniques,” in *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, pp. 1–9, 2023.

[74] M. I. Rana, S. Khalid, and M. U. Akbar, “News classification based on their headlines: A review,” in *17th IEEE International Multi Topic Conference 2014*, pp. 211–216, 2014.

[75] A. Barua, O. Sharif, and M. M. Hoque, “Multi-class Sports News Categorization using Machine Learning Techniques: Resource Creation and Evaluation,” *Procedia Computer Science*, vol. 193, pp. 112–121, 2021.

[76] E. Shushkevich, M. Alexandrov, and J. Cardiff, “Improving Multiclass Classification of Fake News Using BERT-Based Models and ChatGPT-Augmented Data,” *Inventi-ons*, vol. 8, no. 5, 2023.

[77] B. Iancu, G. Mazzola, K. Psarakis, and P. Soilis, “Multi-label Classification for Automatic Tag Prediction in the Context of Programming Challenges,” 2019.

[78] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, “Deep Learning for Extreme Multi-Label Text Classification,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’17, (New York, NY, USA), p. 115–124, Association for Computing Machinery, 2017.

[79] A. Fiallos and K. Jimenes, “Using Reddit Data for Multi-Label Text Classification of Twitter Users Interests,” in *2019 Sixth International Conference on eDemocracy and eGovernment (ICEDEG)*, pp. 324–327, 2019.

[80] D. Zhang, S. Zhao, Z. Duan, J. Chen, Y. Zhang, and J. Tang, “A Multi-Label Classification Method Using a Hierarchical and Transparent Representation for Paper-Reviewer Recommendation,” *ACM Transactions on Information Systems*, vol. 38, Feb 2020.

[81] A. Dasgupta, S. Katyan, S. Das, and P. Kumar, “Review of Extreme Multilabel Classification,” 2023.

[82] Y. Prabhu, A. Kag, S. Harsola, R. Agrawal, and M. Varma, “Parabel: Partitioned Label Trees for Extreme Classification with Application to Dynamic Search Advertising,” in *Proceedings of the 2018 World Wide Web Conference*, WWW ’18, (Republic and Canton of Geneva, CHE), p. 993–1002, International World Wide Web Conferences Steering Committee, 2018.

[83] R. You, Z. Zhang, Z. Wang, S. Dai, H. Mamitsuka, and S. Zhu, “AttentionXML: Label Tree-Based Attention-Aware Deep Model for High-Performance Extreme Multi-Label Text Classification,” in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, (Red Hook, NY, USA), Curran Associates Inc., 2019.

[84] W.-C. Chang, D. Jiang, H.-F. Yu, C. H. Teo, J. Zhang, K. Zhong, K. Kolluri, Q. Hu, N. Shandilya, V. Ievgrafov, J. Singh, and I. S. Dhillon, “Extreme Multi-Label Learning for Semantic Matching in Product Search,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD ’21, (New York, NY, USA), p. 2643–2651, Association for Computing Machinery, 2021.

[85] H.-F. Yu, K. Zhong, I. S. Dhillon, W.-C. Wang, and Y. Yang, "X-BERT: eXtreme multi-label text classification using bidirectional encoder representations from transformers," in *NeurIPS 2019 Workshop on Science Meets Engineering of Deep Learning*, 2019.

[86] J. Zhang, W.-C. Chang, H.-F. Yu, and I. S. Dhillon, "Fast Multi-Resolution Transformer Fine-tuning for Extreme Multi-label Text Classification," in *NeurIPS 2021 Conference on Neural Information Processing Systems*, 2021.

[87] H.-F. Yu, K. Zhong, J. Zhang, W.-C. Chang, and I. S. Dhillon, "PECOS: Prediction for enormous and correlated output spaces," *Journal of Machine Learning Research*, 2022.

[88] T. Qin, T.-Y. Liu, J. Xu, and H. Li, "LETOR: A benchmark collection for research on learning to rank for information retrieval," *Information Retrieval*, vol. 13, pp. 346–374, Aug 2010.

[89] K. Sparck Jones, S. Walker, and S. Robertson, "A probabilistic model of information retrieval: development and comparative experiments: Part 2," *Information Processing and Management: an International Journal*, vol. 36, pp. 809–840, Nov 2000.

[90] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Nov 2019.

[91] T. Formal, B. Piwowarski, and S. Clinchant, *SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking*, p. 2288–2292. New York, NY, USA: Association for Computing Machinery, 2021.

[92] T. Formal, C. Lassance, B. Piwowarski, and S. Clinchant, "SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval," 2021.

[93] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Eficient Estimation of Word Representations in Vector Space," 2013.

[94] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013.

[95] J. Pennington, R. Socher, and C. Manning, “GloVe: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct 2014.

[96] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemo-yer, and V. Stoyanov, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *ArXiv*, vol. abs/1907.11692, 2019.

[97] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” *ArXiv*, vol. abs/1910.01108, 2019.

[98] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized Autoregressive Pretraining for Language Understanding,” in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, (Red Hook, NY, USA), Curran Associates Inc., 2019.

[99] M. Ulčar and M. Robnik-Šikonja, “SloBERTa: Slovene monolingual large pretrained masked language model,” in *Proceedings of the SiKDD Data Mining and Data Warehouses 2021 conference*, 2021.

[100] L. Martin, B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, É. de la Clergerie, D. Seddah, and B. Sagot, “CamemBERT: a Tasty French Language Model,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 7203–7219, Association for Computational Linguistics, Jul 2020.

[101] P. Lops, M. de Gemmis, and G. Semeraro, “Content-based Recommender Systems: State of the Art and Trends,” in *Recommender Systems Handbook*, (Boston, MA), pp. 73–105, Springer US, 2011.

[102] M. Tennenholtz and O. Kurland, “Rethinking Search Engines and Recommendation Systems: A Game Theoretic Perspective,” *Communications of the ACM*, vol. 62, p. 66–75, Nov 2019.

[103] I. Balush, V. Vysotska, and S. Albota, “Recommendation System Development Based on Intelligent Search, NLP and Machine Learning Methods,” in *MoMLeT+DS*, 2021.

[104] D. C. Liu, S. Rogers, R. Shiau, D. Kislyuk, K. C. Ma, Z. Zhong, J. Liu, and Y. Jing, “Related Pins at Pinterest: The Evolution of a Real-World Recommender System,” 2017.

[105] A. Tejeda-Lorente, J. Bernabe-Moreno, J. Herce-Zelaya, C. Porcel, and E. Herrera-Viedma, “Adapting Recommender Systems to the New Data Privacy Regulations,” in *Proceedings of the 17th International Conference on Intelligent Software Methodologies, Tools and Techniques (SoMeT 2018)*, (Granada, Spain), pp. 373–385, Sep 2018.

[106] “Digitalna Knjižnica Unvierze v Mariboru.” https://dk.um.si.

[107] “Repozitorij Univerze v Ljubljani.” https://repozitorij.uni-lj.si.

[108] “Repozitorij Univerze na Primorskem.” https://repozitorij.upr.si.

[109] “Repozitorij Univerze v Novi Gorici.” https://repozitorij.ung.si.

[110] “Digitalni repozitorij raziskovalnih organizacij Slovenije.” https://dirros.openscience.si.

[111] “Repozitorij samostojnih visokošolskih in višješolskih izboraževalnih organizacij.” https://revis.openscience.si/.

[112] G. Shani and A. Gunawardana, “Evaluating Recommendation Systems,” in *Recommender Systems Handbook*, (Boston, MA), pp. 257–297, Springer US, 2011.

[113] D. Monti, E. Palumbo, G. Rizzo, and M. Morisio, “Sequeval: An Ofline Evaluation Framework for Sequence-Based Recommender Systems,” *Information*, vol. 10, p. 174, May 2019.

[114] C. Krauss, A. Merceron, and S. Arbanowski, “The Timeliness Deviation: A Novel Approach to Evaluate Educational Recommender Systems for Closed-Courses,” in *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, LAK19, (New York, NY, USA), pp. 195–204, ACM, 2019.

[115] S. M. McNee, J. Riedl, and J. A. Konstan, “Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems,” in *CHI ’06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’06, (New York, NY, USA), p. 1097–1101, Association for Computing Machinery, 2006.

[116] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, “Learning to rank using gradient descent,” in *Proceedings of the 22nd International Conference on Machine Learning*, ICML ’05, (New York, NY, USA), p. 89–96, Association for Computing Machinery, 2005.

[117] K. Järvelin and J. Kekäläinen, "Cumulated Gain-Based Evaluation of IR Techniques," *ACM Trans. Inf. Syst.*, vol. 20, p. 422–446, Oct 2002.

[118] Y. Wang, L. Wang, Y. Li, D. He, and T.-Y. Liu, "A Theoretical Analysis of NDCG Type Ranking Measures," in *Proceedings of the 26th Annual Conference on Learning Theory* (S. Shalev-Shwartz and I. Steinwart, eds.), vol. 30 of *Proceedings of Machine Learning Research*, (Princeton, NJ, USA), pp. 25–54, PMLR, 12–14 Jun 2013.

[119] O. Nevzorova and D. Almukhametov, "Towards a Recommender System for the Choice of UDC Code for Mathematical Articles," in *DAMDID/RCDL 2021* (A. Pozanenko, S. Stupnikov, B. Thalheim, E. Mendez, and N. Kiselyova, eds.), (Moscow, Russia), pp. 54–62, 2021.

[120] M. Polignano, C. Musto, M. de Gemmis, P. Lops, and G. Semeraro, "Together is Better: Hybrid Recommendations Combining Graph Embeddings and Contextualized Word Representations," in *Proceedings of the 15th ACM Conference on Recommender Systems*, RecSys '21, (New York, NY, USA), p. 187–198, Association for Computing Machinery, 2021.

[121] C. Channarong, C. Paosirikul, S. Maneeroj, and A. Takasu, "HybridBERT4Rec: A Hybrid (Content-Based Filtering and Collaborative Filtering) Recommender System Based on BERT," *IEEE Access*, vol. 10, pp. 56193–56206, 2022.

[122] X. Zhao, H. Kang, T. Feng, C. Meng, and Z. Nie, "A Hybrid Model Based on LFM and BiGRU Toward Research Paper Recommendation," *IEEE Access*, vol. 8, pp. 188628–188640, 2020.

[123] C. N. Dang, M. N. Moreno-García, and F. De la Prieta, "Using Hybrid Deep Learning Models of Sentiment Analysis and Item Genres in Recommender Systems for Streaming Services," *Electronics*, vol. 10, no. 20, 2021.

[124] P. Melville and V. Sindhwani, "Recommender Systems," in *Encyclopedia of Machine Learning and Data Mining*, (Boston, MA), pp. 1056–1066, Springer US, 2017.

[125] Y. Wang, Y. Allouache, and C. Joubert, "A Stafing Recommender System based on Domain-Specific Knowledge Graph," in *2021 Eighth International Conference on Social Network Analysis, Management and Security (SNAMS)*, pp. 1–6, 2021.

[126] G. Penha and C. Hauff, "What Does BERT Know about Books, Movies and Music? Probing BERT for Conversational Recommendation," in *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, (New York, NY, USA), p. 388–397, Association for Computing Machinery, 2020.

[127] J. Zhu, B. Patra, and A. Yaseen, “Recommender system of scholarly papers using public datasets,” in *AMIA 2021 Virtual Informatics Summit*, vol. 2021, pp. 672–679, 05 2021.

[128] W. E. Winkler, “String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage.,” in *Proceedings of the Section on Survey Research*, pp. 354–359, 1990.

[129] M. Ojsteršek, J. Brezovnik, M. Ferme, G. Hrovat, A. Bregant, and M. Boro-vič, “OpenScience Slovenia Dataset.” http://www.openscience.si/OpenData.aspx, 2014. Accessed: 2021-08-15.

[130] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, USA, May 6-9, 2019, Conference Track Proceedings*, 2019.

[131] I. Loshchilov and F. Hutter, “SGDR: stochastic gradient descent with warm restarts,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[132] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[133] R. L. Wasserstein, A. L. Schirm, and N. A. Lazar, “Moving to a world beyond “p<0.05”,” *The American Statistician*, vol. 73, pp. 1–19, 2019.

[134] L. Rostami, “Investigating Search Algorithms for Shorter Documents : A study on how to search for titles,” Master’s thesis, KTH, School of Electrical Engineering and Computer Science (EECS), 2022.

[135] H. Fallah, P. Bellot, E. Bruno, and E. Murisasco, “Adapting Transformers for Multi-Label Text Classification,” in *CIRCLE (Joint Conference of the Information Retrieval Communities in Europe) 2022*, (Samatan, France), Jul 2022.

[136] I. Nejadgholi, S. Kiritchenko, K. C. Fraser, and E. Balkır, “Concept-Based Explanations to Test for False Causal Relationships Learned by Abusive Language Classifiers,” 2023.

[137] K. Dreßler and A.-C. Ngonga Ngomo, “On the Eficient Execution of Bounded Jaro-Winkler Distances,” *Semantic Web*, vol. 8, p. 185–196, Jan 2017.

[138] Z. Zeng and T. Sakai, “BM25 Pseudo Relevance Feedback using Anserini at Waseda university,” *CEUR Workshop Proceedings*, vol. 2409, pp. 62–63, Jan 2019.

[139] D. Bollegala, N. Noman, and H. Iba, "RankDE: Learning a Ranking Function for Information Retrieval Using Differential Evolution," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, (New York, NY, USA), p. 1771–1778, Association for Computing Machinery, 2011.

[140] M. Borovič and J. Dugonik, "Razdalje urejanja 2. del - Jarova in Jaro-Winklerjeva razdalja," *Presek*, vol. 49, no. 6, p. 23–27, 2022.

[141] P. Pu, L. Chen, and R. Hu, "A User-centric Evaluation Framework for Recommender Systems," in *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, (New York, NY, USA), pp. 157–164, ACM, 2011.

[142] A. Schuth, F. Sietsma, S. Whiteson, and M. de Rijke, "Optimizing Base Rankers Using Clicks," in *Advances in Information Retrieval* (M. de Rijke, T. Kenter, A. P. de Vries, C. Zhai, F. de Jong, K. Radinsky, and K. Hofmann, eds.), (Cham), pp. 75–87, Springer International Publishing, 2014.

[143] C. Duchene, H. Jamet, P. Guillaume, and R. Dehak, "A benchmark for toxic comment classification on Civil Comments dataset," 2023.

[144] I. Price, J. Gifford-Moore, J. Fleming, S. Musker, M. Roichman, G. Sylvain, N. Thain, L. Dixon, and J. Sorensen, "Six Attributes of Unhealthy Conversation," 2020.