# KRNJENJE SLOVENSKIH BESEDIL S PODROČJA BIBLIOTEKARSTVA[1]

**Polona Vilar**

**Jasna Maver**

Oddano: 11.04.2002 – Sprejeto: 06.05.2002

Izvirni znanstveni članek

UDK 001.4 : 02 : 004.021

## Izvleček

Prispevek predstavlja pripravo algoritma za krnjenje slovenskih besedil s področja bibliotekarstva, ki je potekal v treh fazah: učni, testni in evalvacijski. Predstavljena je priprava oz. učenje Optimalnega algoritma za krnjenje bibliotekarskih besedil, njegovo testiranje in primerjava z dvema drugima algoritmoma za krnjenje, imenovanima Popovičev in Generični. Za učenje je bil uporabljen korpus 790.000 besed s področja bibliotekarstva. Zgrajeni so bili seznami krnov, besednih končnic ter blokiranih besed. Testna faza je obsegala testiranje algoritma, predvsem zgrajenih sestavnih delov, z dodatnim korpusom, obsegajočim 167.000 besed. V evalvacijski fazi je bila izvedena primerjava delovanja omenjenih treh algoritmov na istem korpusu. Rezultati delovanja algoritmov so primerjani z intelektualno pripravljenim kontrolnim rezultatom. V njem so množice semantično povezanih besed, zastopane s krni. Spremljano je premalo natačno krnjenje – koliko krnov za semantično povezane besede izdela posamezen algoritem. Rezultati so statistično obdelani s Kruskal-Wallisovim testom. Optimalni algoritem daje najboljše rezultate. Največkrat doseže popolno ujemanje s kontrolnim rezultatom in hkrati izdela najmanj krnov za en pomen. Sledi Popovičev z majhnim odstopanjem. Najmanj natančen je Generični. Opisani postopki lahko predstavljajo izhodišče za nadaljnjo gradnjo orodij za avtomatsko indeksiranje dokumentov s področja bibliotekarstva in poizvedovanje po njih.

**Ključne besede:** avtomatsko krnjenje, algoritmi, slovenski jezik, bibliotekarstvo

1 *Prispevek opisuje eksperiment, ki je v letih 2000 in 2001 potekal na Oddelku za bibliotekarstvo, informacijsko znanost in knjigarstvo na Filozofski fakulteti v Ljubljani, in je podrobneje opisan v magistrskem delu "Krnjenje slovenskih besedil s področja bibliotekarstva" (Vilar, 2001).*

VILAR, Polona; Jasna MAVER: Stemming of Slovenian library science texts. Knjižnica, Ljubljana, 46(2002)1-2, 111-136

Original scientific article
UDC 001.4 : 02 : 004.021

## Abstract

The theme of the article is the preparation of a stemming algorithm for Slovenian library science texts. The procedure consisted of three phases: learning, testing and evaluation. The preparation of the optimal stemmer for Slovenian texts from the field of library science is presented, its testing and comparison with two other stemmers for the Slovenian language: the Popovič stemmer and the Generic stemmer. A corpus of 790.000 words from the field of library science was used for learning. Lists of stems, word endings and stop-words were built. In the testing phase, the component parts of the algorithm were tested on an additional corpus of 167.000 words. In the evaluation phase, a comparison of the three stemmers processing the same word corpus was made. The results of each stemmer were compared with an intellectually prepared control result of the stemming of the corpus. It consisted of groups of semantically connected words with no errors. Understemming was especially monitored – the number of stems for semantically connected words, produced by an algorithm. The results were statistically processed with the Kruskal-Wallis test. The Optimal stemmer produced the best results. It matched best with the reference results and also gave the smallest number of stems for one semantic meaning. The Popovič stemmer followed closely. The Generic stemmer proved to be the least accurate. The procedures described in the thesis can represent a platform for the development of the tools for automatic indexing and retrieval for library science texts in Slovenian language.

**Key words:** stemming, stemming algorithms, Slovenian language, library and information science

# 1 Uvod

## 1.1 Poizvedovanje in avtomatsko indeksiranje

Tehnika, ki jo opisuje prispevek, sodi na področje avtomatskega indeksiranja, oz. širše gledano, na področje poizvedovanja. Oboje sta v preglednem članku opisala Vilarjeva in Dimec. Poizvedovanje definirata kot: "Prevod angleškega termina *information retrieval*, za katerega se v slovenskem prostoru pojavljajo različna poimenovanja, npr. iskanje informacij, iskanje in priklic informacij, itd. Gre za sistematično preiskovanje indeksiranih informacijskih virov z odkrivanjem, izbiranjem *in pridobivanjem* podatkov, zapisov iz njih. V širšem, v svetu najbolj uveljavljenem pomenu, information retrieval pomensko vključuje tudi predhodne postopke gradnje zbirke dokumentov, še posebej postopke opisovanja njihove vsebine" (Vilar in Dimec, 2000, str. 7).

Avtomatsko indeksiranje je ena najpogostejših tehnik obdelave dokumentov s polnimi besedili na področju poizvedovanja. Gre za proces algoritmične obdelave besedil z namenom določanja seznama indeksnih izrazov. Sodi med derivativne metode, saj indeksni izrazi izvirajo iz dokumenta. Prednosti, ki jih ponuja avtomatsko indeksiranje, so odprava nekaterih pomanjkljivosti intelektualnega indeksiranja, denimo zmanjševanje stroškov in količine intelektualnega dela, in predvsem večja hitrost. Poleg tega je bolj izčrpno, ima bolj specifično terminologijo, in vsaj teoretično natančno predstavlja vsebino dokumenta. Njegova pomanjkljivost pa je, da je izrazito vezano na besedilo dokumenta (tako glede postopkov indeksiranja kot tudi iskanja), ne uvaja semantičnih povezav med indeksnimi izrazi (kot npr. v tezavru) in proizvede obsežne sezname indeksnih izrazov.

Avtomatsko indeksiranje uporabljamo skoraj izključno za analizo besedil. Avtomatske tehnike na področju analiziranja slik, video in avdio dokumentov so trenutno še v eksperimentalni fazi, čeprav znanstveniki na njih intenzivno delajo. Cilj, h kateremu težijo sistemi za poizvedovanje po polnih besedilih, je doseganje enakih rezultatov tako iskanja kot tudi vsebinske obdelave, kot bi ti bili, če bi se jih lotil človek. Seveda je dejansko stanje še precej oddaljeno od tega cilja. Potrebno je opozoriti, da je moč računalnikov pri konceptualnem indeksiranju zelo omejena, saj so le v redkih primerih sposobni interpretiranja. To zaenkrat še vedno ostaja domena človeka, prav tako pa tudi prepoznavanje zatipkanih besed ali oblikoslovnih različic besed, ki so posledica slovničnih značilnosti jezika.

Za avtomatski izbor vsebinskih predstavnikov besedila obstajata dva osnovna pristopa: jezikoslovni in statistični. Prvi temelji na semantičnih in sintaktičnih metodah, drugi pa na pogostosti pojavljanja besed v besedilu (Popovič, 1990). Pri sodobnih avtomatskih tehnikah za opisovanje vsebine gre navadno za statistično analizo pogostosti pojavljanja besed v tekstu. Včasih so jim dodane še metode za razpoznavanje besednih korenov, fraz ali celo semantičnih pomenov besed. Sistemi tehtajo pomembnost posameznih izrazov glede na njihovo pozicijo v dokumentu, kontekst, v katerem se pojavljajo, ali glede na njihovo sintaktično pozicijo, mnogi pa tudi glede na vgrajene tezavre (Vilar in Dimec, 2000, str. 11).

## 1.2 Postopek avtomatskega indeksiranja

V procesu avtomatskega indeksiranja je potrebno opraviti nekaj zaporednih postopkov:

1. Leksikalna analiza

   To je avtomatska analiza besedila dokumenta (naslov, povzetek ali celotno besedilo), v kateri predvsem določamo potencialne kandidate za indeksne izraze. Drugače povedano, gre za pretvarjanje vhodnega besedila v niz posameznih besed v stolpcih, te pa nato obdelamo z avtomatskimi postopki.

2. Blokiranje

   To je primerjava kandidatov za gesla s t. i. seznamom blokiranih besed, oziroma negativnim slovarjem, v katerem so besede, ki v besedilu nosijo zelo malo ali nič pomena, in so zato nepotrebne.

3. Izbor izrazov, ki predstavljajo vsebino dokumenta

4. Krnjenje

To je poenotenje morfoloških različic izrazov, ki so bili izbrani za gesla, in na ta način zlitje besed v enotno obliko, ki nato zastopa vse različice. Je ključnega pomena za uspeh poizvedovanja.

## 1.3 Krnjenje

### 1.3.1 Oblikoslovje in semantika

Jezik je živ organizem in nanj vplivajo različne zakonitosti, zato obstajajo za isti semantični pomen različne oblike besed. Mednje sodijo sinonimi, kratice, pa tudi morfološke oziroma oblikoslovne različice, ki nastajajo z dodajanjem najrazličnejših končnic, oz. pon (pripon ali predpon). Te bodisi spreminjajo besedno vrsto, spol, sklon, sklanjatev, spregatev bodisi katero drugo značilnost posamezne besede.

Ujemanje iskalnih zahtev in dokumentov v podatkovni zbirki je odvisno od števila in pogostosti izrazov, ki se pojavljajo v obojih. "V nasprotju z jezikoslovjem, ki različne besedne vrste, izpeljanke, zloženke in podobno obravnava ločeno, je za proces poizvedovanja bistvenega pomena semantična interpretacija besed. Oblikoslovne različice, ki so posledica obnašanja naravnega jezika, zato v procesu poizvedovanja predstavljajo oviro, kajti z vidika semantike jih lahko obravnavamo kot sinonime" (Vilar in Dimec, 2000, str. 12).

### 1.3.2 Zlivanje besed

Ker bi iskalni algoritmi omenjene oblikoslovne različice prepoznali kot različne besede, je za besedila v naravnem jeziku potrebno procesiranje, katerega končni cilj je odstraniti omenjene oblikoslovne različice iz besedila, oz. jih poenotiti. To najpogosteje storimo tako, da jih zreduciramo na koren, ki je sicer dejanski nosilec pomena besede. Ta potem nastopa namesto vseh različnih oblik. Velikokrat pa se zgodi, da preostanek ne ustreza korenu v jezikoslovnem pomenu, zato namesto tega izraza uporabljamo izraz krn, iz česar izvira tudi izraz krnjenje.

"Krnjenje povzroči "zlivanje" oziroma združevanje pomensko sorodnih, a različnih besed v enotno obliko. Lahko se izvaja ročno ali avtomatsko. Ročno krnjenje imenujemo tudi ročni odrez. Navadno se izvaja v fazi iskanja, ponavadi z desnim krnjenjem, ki je lahko nastavljeno kot privzeta vrednost, a ga pogosteje izvaja iskalec. Za avtomatsko krnjenje pa se uporabljajo algoritmi. V nasprotju z ročnim se avtomatsko krnjenje ne izvaja v fazi iskanja, temveč na besedah, ki sestavljajo iskalne izraze ter na dokumentih, ki sestavljajo podatkovno zbirko" (Vilar in Dimec, 2000, str. 13).

Med najpogostejše težave pri krnjenju sodita premočno in prešibko krnjenje. Prvo povzroči zlivanje besed, ki imajo različne pomene, in bi torej ne smele biti zlite na en krn. Zaradi tega pri poizvedovanju trpi natančnost. Prešibko krnjenje pa se pojavi, kadar algoritem ne uspe zliti vseh različic besed za en pomen na isti krn, zaradi česar ostane za en pomen več različnih krnov, kar zmanjšuje odziv.

### 1.3.3 Vrste algoritmov za krnjenje

Obstaja veliko različnih vrst algoritmov, za katere avtorji predlagajo različne delitve. Vilarjeva in Dimec (2000, str. 14) predlagata delitev na:

1. algoritme, ki zlivajo besede na osnovi soodvisnosti črk v besedi;
2. algoritme, ki združujejo semantično sorodne besede v šope na osnovi njihove statistične sorodnosti;
3. algoritme, ki zlivajo besede z odstranjevanjem pon;
4. algoritme, ki pri zlivanju uporabljajo slovarje.

Poleg opisov nekaterih algoritmov za tuje jezike natančno predstavita tudi štiri algoritme, ki so bili izdelani za slovenščino (Ibidem, str. 22-26):

1. Preprost algoritem, izdelan na Medicinski fakulteti v Ljubljani;
2. Algoritem Mirka Popoviča;
3. Generični algoritem;
4. Optimalni algoritem.

## 2 Optimalni algoritem

Za eksperimente, ki so opisani v nadaljevanju, smo uporabili algoritem, ki so ga avtorji (Dimec et al., 1999) preprosto poimenovali Optimalni algoritem, kajti njegov namen je, naj bi v vsakem trenutku oblikoval najbolj optimalen krn. Podrobno sta ga opisala Vilarjeva in Dimec (2000), tu bomo omenili le nekaj najpomembnejših značilnosti.

Algoritem sodi v družino algoritmov, ki odstranjujejo pone, natančneje besedne končnice. Deluje po načelu najdaljšega ujemanja, a tako, da poskuša vedno ohraniti najdaljši možni niz znakov, ki nato zastopa oblikoslovne različice semantično povezanih besed. Pri najkrajši možni različici namreč prehitro lahko pride do premočnega krnjenja. Dodatna ideja avtorjev je bila, da je potrebno krnjenje bolj prilagoditi strokovnim podjezikom. Iz tega drugega razloga je za delovanje algoritma vedno v začetku potrebna t. i. učna faza, v kateri pridobimo nabor krnov, ki izvirajo iz terminologije nekega področja. Tako zgradimo sestavne dele algoritma, ki jih moramo v nadaljnjih postopkih seveda redno dopolnjevati in vzdrževati.

V sistemu tako sodelujejo trije seznami krnov:

- veljavni krni;
- imenski krni;
- novi krni.

Prvi seznam vključuje optimalne oblike krnov podjezika, v drugem se nahajajo krni, ki ne sodijo v podjezik, a so pomembni kot vsebinski predstavniki dokumentov (osebna, geografska imena, kratice...). V tretjem se nahajajo vsi krni, ki nastanejo v postopku, in jih še ni med veljavnimi in imenskimi. Zato se imenuje seznam novih krnov.

Delovanje algoritma je naslednje (Vilar in Dimec, 2000): "V nasprotju z algoritmi z najdaljšim ujemanjem, Optimalni algoritem načne besedo na koncu. Od desne proti levi išče pare soglasnik-samoglasnik. Pri vsakem paru opravi popolni postopek krnjenja – odrez končnice, obdelavo soglasniških parov in uporabo pravil za popravljanje krnov. Krn, ki nastane, sistem poskuša najti v seznamih veljavnih in imenskih krnov. Če ga najde, je postopek opravljen, sicer poišče naslednji par soglasnik-samoglasnik v smeri proti začetku besede in ponovi krnjenje. Algoritem torej deluje iterativno, z vedno daljšimi in daljšimi končnicami." Če se iskanje krna v seznamih veljavnih in imenskih krnov nikoli ne izteče pozitivno, je rezultat zadnji krn, ki je nastal v postopku." Slika 1 na simboličen način predstavlja delovanje Optimalnega algoritma.

```
1. Begin.
   STEM = word;
2. Search for the STEM in LEGAL_STEMS and in OTHER_STEMS list;
3. If found go to 11;
4. If MOVING_ALLOWED move cursor in STEM one character to the left
   else go to 11;
5. Search in the CV_ENDINGS list for the STEM's ending beginning
   at the cursor;
6. If found cut it off the STEM
   else go to 4;
7. Iteratively process the STEM consonant pairs with the use of
   CONS_PAIRS list;
8. Process the STEM with rules from the RECOD_RULES list;
9. Search for the STEM in LEGAL_STEMS and OTHER_STEMS list;
10.If found go to 11
   else go to 4
11.End
   Pass the STEM to next indexing steps.

Legenda:
LEGAL_STEMS – seznam veljavnih krnov
OTHER_STEMS – seznam drugih krnov
MOVING ALLOWED – pravila o najkrajšem dovoljenem krnu
CV-ENDINGS – seznam končnic, ki cepijo besedo na stiku soglasnik-
samoglasnik
CONS-PAIRS – seznam parov soglasnikov
RECOD-RULES – pravila za popravljanje krnov
```

**Slika 1:** Delovanje Optimalnega algoritma (Dimec et al., 1999)

Srce algoritma so torej trije koraki (5-8), ki se izvedejo za vsak premik kazalca. Krnjenje ni možno na vsakem premiku, kadar pa je uspešno opravljeno, algoritem poskuša najti izdelani krn v seznamih veljavnih in drugih krnov. Če ga najde, se postopek konča, sicer se ponovi za naslednji premik kazalca. Končnim pogojem je zadoščeno, ko je krn najden v enem izmed seznamov, ali pa naslednji premik kazalca ni mogoč zaradi pravil o najkrajšem možnem krnu (Dimec et al., 1999).

Krn, ki ga izdela algoritem, in rezultati vseh vmesnih stopenj krnjenja se zapišejo v seznam novih krnov. Administrator sistema nato ob pregledu seznama krne označi z eno izmed treh možnih oznak:

- krn je lahko optimalen – takrat dobi oznako "v" (iz angl. valid);
- lahko izvira iz kratice, geografskega ali osebnega imena – takrat dobi oznako "n" (iz angl. name);
- ali pa izvira iz napačno zapisane ali iz kateregakoli drugačnega razloga nesprejemljive besede – takrat dobi oznako "b" (iz angl. bad).

Z uporabo posebnega programa, ki so ga avtorji poimenovali "Program za čiščenje", in ki vse spremembe takoj odslika v indeksni datoteki zbirke doku

mentov, dosežemo, da se označeni krni iz tega seznama preslikajo v ustreznega izmed prvih dveh. Optimalni krni se vključijo v seznam veljavnih krnov, imenski v seznam imenskih krnov, slabe pa zavržemo.

Najpomembnejše v celotnem postopku je vzdrževanje seznama veljavnih krnov, ki vsebuje nabor besed oz. krnov podjezika, in seznama imenskih krnov. Seznama naraščata z vsakim obdelanim dokumentom in tako se algoritem "uči" podjezika. Dela s seznami je z naraščanjem zbirke dokumentov vedno manj, kajti vse več izrazov se že nahaja v enem izmed njiju, zato jih je vse manj v seznamu novih krnov. Administrator sistema določa najustreznejšo obliko krnov. Ni nujno, da se pri pregledovanju seznama novih krnov odloči za obliko krna, ki jo za določeno besedo predlaga algoritem. S pomočjo spiska vseh krnov, ki so nastali v zaporednih stopnjah krnjenja, izbere tistega, ki je po njegovem najustreznejši. Če ima občutek za jezik, obstaja velika verjetnost, da bo tak krn res optimalen.

# 3 Krnjenje besedil s področja bibliotekarstva

## 3.1 Kratek opis eksperimentov

Z eksperimenti smo skušali ugotavljati način in hitrost učenja algoritma na bibliotekarskem podjeziku ter njegovo učinkovitost in posledično uporabnost rezultatov. Zato smo v našem postopku zasnovali tri faze.

1. Učna faza

Ker je bil Optimalni algoritem razvit za področje medicine, mi pa smo potrebovali besedišče s področja bibliotekarstva, smo učno fazo pričeli s praznima seznamoma veljavnih in imenskih krnov. Oba smo, skupaj s seznamom blokiranih besed in dopolnjenima seznamoma končnic, zgradili za bibliotekarski podjezik. Za to smo uporabili besedilni korpus, obsegajoč 790.000 besed, ki smo ga pred tem morali pripraviti za obdelavo.

2. Testna faza

Po končani učni fazi smo testirali izdelani algoritem z dodatno, testno množico besedil. Ta je bila manjša, obsegala je 167.000 besed. V testni fazi smo preverjali uspešnost delovanja in zmanjševanje besedne mase v posamezni fazi v postopku krnjenja: blokiranju in samem krnjenju.

3. Faza vrednotenja

V tej fazi smo primerjali delovanje "našega" algoritma z dvema drugima algoritmoma, izdelanima za slovenščino. Primerjali smo ga z Generičnim algoritmom (Dimec et al., 2000) ter algoritmom, ki ga je izdelal Popovič (1991a).

Rezultate obdelave testnega korpusa z vsemi tremi algoritmi smo statistično obdelali z neparametričnim Kruskal-Wallisovim testom. Ocenjevali smo, ali algoritmi dajo enake rezultate. Poskušali smo tudi ugotoviti, kateri algoritem da najboljše rezultate brez ročnega poseganja, oz. se kar najbolj približa idealnemu rezultatu krnjenja. To smo ugotavljali iz izračunanega povprečnega števila krnov, ki jih izdela vsak algoritem v primerjavi z referenčnim rezultatom, standardno deviacijo in kumulativno vsoto krnov.

## 3.2 Pridobivanje in priprava besedil

### 3.2.1 Besedila za učno fazo

Odločili smo se, da besedila s področja bibliotekarstva poiščemo v temeljni slovenski bibliotekarski reviji "Knjižnica", ki jo izdaja Zveza bibliotekarskih društev Slovenije. Izbirali smo samo novejše članke v slovenskem jeziku, torej nismo upoštevali objav v drugih jezikih (hrvaščina, angleščina).

Po ugotovitvi, da ne obstaja digitalni arhiv revije, smo se odločili za naslednji postopek: procesiranje besedil z optičnim čitalnikom in nato s programom za optično razpoznavanje znakov. Določene napake, ki so se pri tem pojavile, smo popravljali ročno. Izločali smo tudi izvlečke, ključne besede, bibliografijo. Na ta način smo pridobili besedila štirinštiridesetih člankov iz let 1998 in 1999.

Drugi obsežni sklop besedil so predstavljala besedila prispevkov v zbornikih s posvetovanj Sekcije za specialne knjižnice pri Zvezi bibliotekarskih društev Slovenije (takih smo pridobili 55) ter besedilo zbornika ob petdeseti obletnici Narodne in univerzitetne knjižnice (10 člankov). Pridobili smo jih v digitalni obliki iz grafičnega studia, zato tu ni bilo potrebe za optično čitanje in razpoznavanje ter ročno popravljanje. V učno množico smo vključili tudi besedila 28 diplomskih nalog z Oddelka za bibliotekarstvo.

Datoteke smo shranjevali v tekstovnem formatu. Besedilo smo nato v procesu leksikalne analize pretvorili v stolpec besed (tokenov), istočasno izločali nekatere odvečne znake (ločila, oklepaje, narekovaje, znake "*", "+", "<", ">", "=") in številke, ter pretvorili vse črke v male. Učni korpus je obsegal preko 790.000 besed, ki smo jih zaradi enostavnejše obdelave razdelili v 53 segmentov po 15.000 besed.

### 3.2.2 Besedila za testno fazo

Tudi besedila za testno fazo smo pridobili iz grafičnega studia, zato postopki digitalizacije niso bili potrebni. Odločili smo se za drugo temeljno revijo s področja bibliotekarstva, "Šolska knjižnica", ki jo izdaja Zavod Republike Slovenije za šolstvo. V celoti smo jih pridobili iz grafičnega studia, kjer besedila pripravljajo za tisk, in si s tem prihranili postopke digitalizacije. Testni korpus je bil manjši od učnega, obsegal je 167.000 besed. Tudi to besedilo smo prevedli v ascii format in pretvorili v tokene, a smo ga ohranili kot en korpus v eni datoteki. Ker so bila besedila pripravljena z računalnikom tipa Apple-Macintosh, so bili pred uporabo potrebni določeni ročni posegi, npr. pretvarjanje znakov za šumnike.

## 3.3 Obdelava

### 3.3.1 Učna faza

3.3.1.1 Krnjenje

Tako pripravljene datoteke smo po vrsti procesirali z Optimalnim algoritmom. Rezultat krnjenja je bil na vsakem koraku seznam novih krnov, iz katerega so bile razvidne tudi vse stopnje krnjenja, ki jih je algoritem opravil pri oblikovanju določenega krna.

Ker sta bila v začetku oba seznama, veljavni in imenski krni, prazna, so bili v prvem koraku v datoteko "Novi krni" uvrščeni vsi krni, ki jih je izdelal algoritem. Kasneje pa se je z naraščanjem obeh seznamov obseg datoteke "Novi krni" na vsakem koraku zmanjšal, izjemoma pa ne, kadar je v obdelavo prišel nov vir krnjenih besedil.

Krnjenju je sledila faza, v kateri je imela ključno vlogo administratorka. Njena naloga je bilo pregledovanje in ročno označevanje predlaganih krnov z ustreznimi oznakami "v" za veljavne oziroma optimalne, "n" za imenske in "b" za neustrezne.

Pri veljavnih in imenskih krnih je morala biti administratorka poleg dodeljevanja oznak pozorna tudi na to, da z izbiro prekratkega krna ni povzročila premočnega krnjenja. V ta namen ji je bila v pomoč datoteka z izvornimi besedami za vsak predlagan krn "New_stem_origins". Lahko se je namreč odločila za katero koli izmed stopenj krnjenja, ne nujno zadnjo. S tem se doseže nedvoumnost krna (ta bo zastopal vse različice besed za en pomen). Za besede, ki pomensko ne sodijo zraven, pa se lahko oblikuje nov krn, ali jih ne upošteva. Struktura zapisov v datotekah je naslednja:

Datoteka novih krnov:

`krn/identifikacijska številka/stopnje krnjenja`
(na prvem mestu je cela beseda, sledijo pa koraki, ki so privedli do takega krna)

Datoteka izvornih besed:

`identifikacijska številka/krn/izvorne besede, ki so dale ta krn`

Identifikacijska številka služi povezovanju zapisov v obeh datotekah, in je še posebej pomembna, kadar se administrator ne odloči za zadnjo stopnjo krnjenja. Naslednji primer prikazuje rezultat krnjenja oblikovno podobnih, a semantično različnih besed *žarnica* in *žarčenje*.

Zapis iz datoteke novih krnov:

`žar/6464/žarnice,žarnic,žarn,žar`

Zapis iz izvorne datoteke:

`6464/žar/žarnice,žarčenja`

V datoteki novih krnov vidimo oblikovan krn in vse stopnje krnjenja besede *žarnice*. Krn preverimo v datoteki izvornih besed. Ugotovimo, da sta besedi, iz katerih je algoritem izdelal krn *žar*, dve. Vidimo, da gre za premočno krnjenje, ker besedi *žarnice* in *žarčenje* ne smeta imeti istega krna. Zato se odločimo za predzadnjo stopnjo krnjenja, torej *žarn* za žarnico, in ročno ustvarimo nov krn *žarč* za žarčenje, ki mu dodamo identifikacijsko številko 0. Oba označimo kot veljavna. Po opravljeni nalogi izgledata zapisa v datoteki takole:

```
v/žarn/6464/žarnice,žarnic,žarn,žar

v/žarč/0/žarčenje
```

Seveda se je bilo potrebno v določenih primerih sprijazniti tudi s prešibkim krnjenjem. V določenih primerih se zgodi, da en koncept neizogibno zastopata dva ali celo več krnov, katerih vzrok je besedotvorna in oblikoslovna pestrost slovenskega jezika, in ki jih ne moremo poenotiti, včasih niti s pravili za rekodiranje krnov. Naslednji primer ilustrira take dogodke:

Zapis iz datoteke novih krnov:

```
zadel/5472/zadelo,zadel
zadet/891/zadeti,zadet
zadetk/3691/zadetkov,zadetk
```

Zapis iz izvorne datoteke:

```
5472/zadel/zadelo,
891/zadet/zadeti,
3691/zadetk/zadetke,zadetki,zadetkih,zadetkov,
```

3.3.1.2 Čiščenje seznama novih krnov

Po opravljenem označevanju smo seznam novih krnov vsakič obdelali s programom za "čiščenje". Program deluje tako, da krne označene z "v" preslika v seznam veljavnih krnov (Valid_stems), z "n" v seznam imenskih krnov (Name_stems), tiste z "b" pa zavrže. Enako se zgodi z izvornimi besedami za vsak posamezen krn. Datoteka z besedami, iz katerih izvirajo novi krni, se po opravljenem čiščenju krnov iz datoteke novih krnov razdeli glede na oznako, ki smo jo dodelili posameznemu krnu. Nastaneta torej dve novi datoteki: izvorna datoteka veljavnih krnov (Valid_stem_origins) in izvorna datoteka imenskih krnov (Name_stem_origins), ki pripadata ustreznima datotekama z veljavnimi in imenskimi krni. Poleg tega je rezultat procesiranja vsakič tudi datoteka s poročilom o postopku in naraščanju števila krnov v vseh seznamih. Oba seznama smo občasno tudi kontrolirali oz. popravljali morebitne napake administratorke (npr. napačno uvrščene besede).

3.3.1.3 Dopolnjevanje seznamov končnic in blokiranih besed

Na podlagi rezultatov krnjenja in opaženih napak smo po potrebi dopolnjevali tudi seznama končnic (moje_koncnice in sogl_koncnice), kot prikazuje naslednji primer:

Zapis iz datoteke novih krnov:

```
arhiv/5386/arhivskega,arhivskeg,arkivsk,arhiv
arhivir/5947/arhiviranje,arhiviranj,arhivir
```

Zapis iz izvorne datoteke:

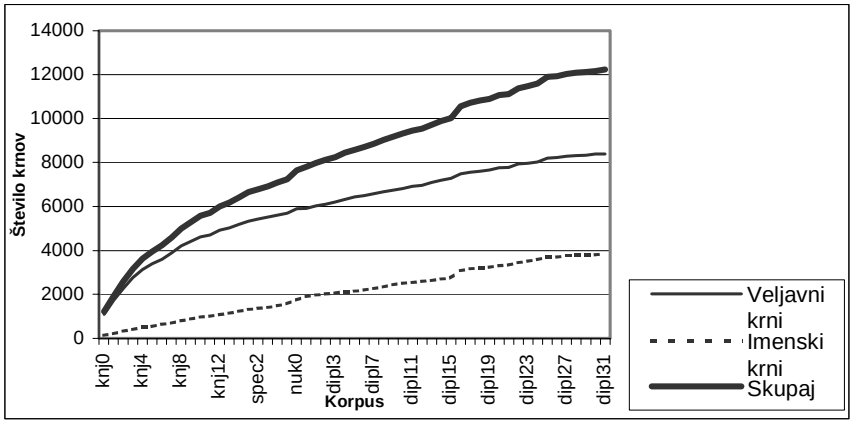
```
5386/arhiv/arhiv,arhivih,arhivskega,arhivskem,
arhivski,arhivskih,arhivu,
5947/arhivir/arhiviranje,
```

Algoritem je izdelal dva različna krna za besede, ki pomensko sodijo v isto skupino. Vidimo, da gre tu dejansko za prešibko krnjenje, ki izvira iz dejstva, da v seznamu končnic manjka *–iranje*. Zato smo jo dodali v seznam končnic.

Dopolnjevali smo tudi seznam blokiranih besed. Ta je v začetku obsegal 2.201 besed, na koncu pa je število naraslo na 2.393. Dodajali smo besede kot idr., joj, hm, prva, dvojen, enem, dveh...

3.3.1.4 Učenje algoritma

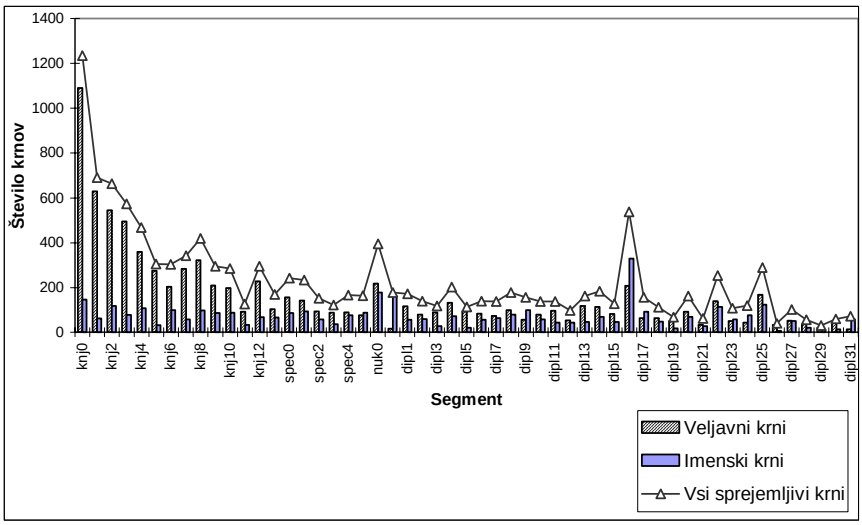
Naraščanje števila veljavnih in imenskih krnov v učnem korpusu prikazuje slika 3. Po zaključeni učni fazi je bilo v datoteki veljavnih 8.389 krnov, datoteka imenskih krnov je obsegala 3.854 krnov, torej je bilo celotno število sprejemljivih krnov 12.243. Na Grafikonu 1 opazimo naraščanje števila veljavnih krnov, ki je sprva strmo, a se nato prične umirjati. Nasprotno pa je naraščanje števila imenskih krnov skoraj linearno.




**Grafikon 1:** Naraščanje števila sprejemljivih krnov

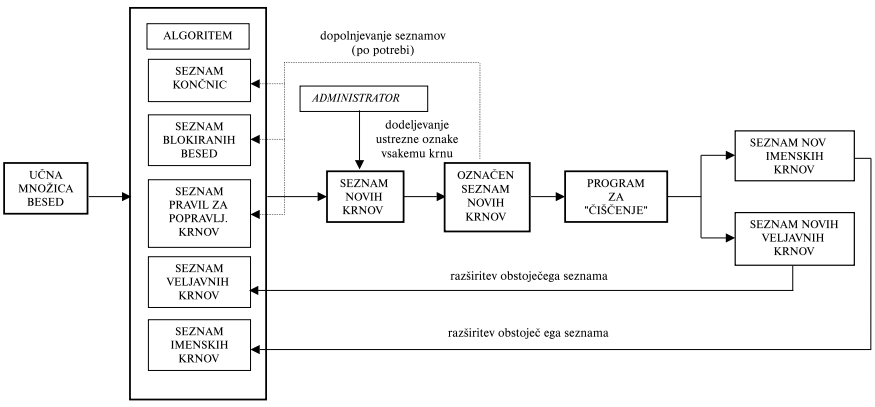
Na Grafikonu 2 vidimo potek učenja bibliotekarskega podjezika z dodano krivuljo trenda. V prvem besedilnem segmentu 15.000 besed, je bilo število sprejemljivih krnov 1.235 (8,23%), v zadnjem segmentu pa le še 71 (0,47%). To pomeni, da je algoritem ob obdelavi našel vse več krnov že med veljavnimi ali imenskimi, in jih zato ni uvrščal v seznam novih krnov. Največje razlike opazimo pri veljavnih krnih – teh je bilo v začetku 1.090 (7,27%), nato je število novih strmo padalo do šestega segmenta. Nato se je ustalilo okrog enega odstotka, v zadnjem segmentu pa je bilo veljavnih krnov le še 14 (0,093%). Pri imenskih krnih je trend bolj enakomeren: v začetku jih je bilo 145 (0,97%), nato je število ostajalo bolj ali manj konstantno v vseh segmentih, na koncu jih je bilo 57 (0,38%). Opazimo lahko nenadno povečanje števila novih sprejemljivih krnov

ob menjavi vira besedila (spec1, nuk0, dipl0). Še posebej pa je zanimiv trend ob obdelavi besedil diplomskih nalog, kjer nekajkrat opazimo vidno povečanje števila krnov. Sklepamo lahko, da gre za vsebinsko izrazito heterogen korpus.




**Grafikon 2:** Učenje bibliotekarskega podjezika

Slika 2 vsebuje grafični prikaz mehanizma delovanja algoritma v fazi učenja. Učenje poteka v iteracijah, torej se ponovi za vsak segment s 15.000 besedami.




**Slika 2:** Mehanizem delovanja optimalnega algoritma v fazi učenja

### 3.3.2 Testna faza

S seznami, pripravljenimi v učni fazi, smo pričeli testno fazo optimalnega algoritma. V njej smo uporabili dodaten, posebej za to pridobljen korpus besedil iz revije Šolska knjižnica, ki je obsegal 166.969 besed. Želeli smo testirati, kako učinkovita bo obdelava tega korpusa z algoritmom s sestavnimi deli, zgrajenimi v učni fazi.

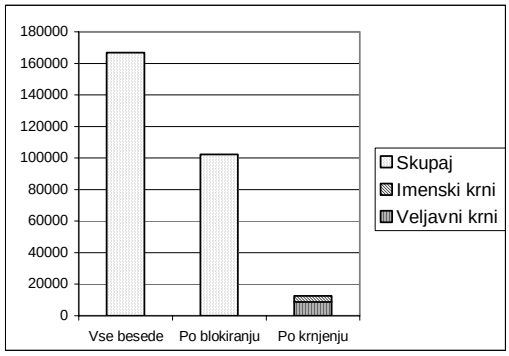
Obseg seznamov, s katerimi smo vstopili v testno fazo, je naslednji:

- seznam veljavnih krnov: 8.389 krnov;
- seznam imenskih krnov: 3.854 krnov;
- seznam blokiranih besed: 2.393 besed.

Najprej smo želeli testirati učinkovitost blokiranja. Po navedbah nekaterih raziskovalcev (Popovič, 1991b) se pri jezikih z zapleteno oblikoslovno strukturo besedna masa po opravljenem blokiranju zmanjša celo do 50%. V našem primeru je algoritem na podlagi seznama blokiranih besed izločil 64.626 pomensko praznih besed, kar predstavlja 38,7% korpusa. V postopek krnjenja je bilo tako uvrščenih 102.343 besed.

Drugi del testne faze je predstavljalo krnjenje testnega korpusa. Po končanem krnjenju je algoritem izdelal 8.588 veljavnih in 3.973 imenskih krnov, torej skupaj 12.561 sprejemljivih krnov iz testnega korpusa. V datoteko novih krnov, torej takih, ki jih ni našel na nobenem seznamu, pa je uvrstil 5.910 krnov. Po pregledu tega seznama je administratorka ugotovila, da jih je bilo 5.561 slabih (94%), 214 imenskih in 135 veljavnih. Vidimo torej, da je bilo le 349 krnov takih, ki še niso bili uvrščeni na sezname sprejemljivih krnov, s katerimi je sicer delal algoritem.

Po zaključenem krnjenju je tako od besed, ki so bile po blokiranju uvrščene v krnjenje, ostalo 12,27%. Če gledamo celoten korpus, predstavljajo izdelani krni 7,52% besedila. Tak rezultat predstavlja občutno zmanjšanje obsega datoteke. Slika 3 prikazuje zmanjševanje besedne mase v posamezni fazi postopka.




**Slika 3:** Zmanjševanje besedne mase

Potrebno je opozoriti, da gre v tem primeru za drugačno situacijo kot v fazi učenja. Tam so bili produkt krnjenja le tisti krni, ki jih algoritem ni uspel najti v seznamih veljavnih ali imenskih krnov, in jih je bilo zato potrebno ročno označevati kot veljavne, imenske ali nesprejemljive, s čimer smo dograjevali seznama veljavnih in imenskih krnov. Tu pa ni šlo več za dograjevanje seznamov. Teh 12.561 krnov so vsi krni, ki so nastali iz korpusa 166.969 besed. 349 krnov pa je bilo za algoritem novih.

Gre za to, da so se besede, ki so ostale po opravljenem postopku blokiranja, "zlile" na krne, ki sedaj zastopajo njihov semantični pomen, in ki bi jih uporabili v nadaljnjih postopkih gradnje seznamov predstavnikov vsebine in tudi v postopkih poizvedovanja.

# 4 Vrednotenje

Po zaključeni učni fazi je bil Optimalni algoritem izoblikovan in pripravljen za nadaljnje delo. S testnim korpusom smo testirali uspešnost učenja. Na njegovi osnovi smo želeli izvesti tudi vrednotenje. V fazi vrednotenja smo torej na testnem korpusu primerjali delovanje treh algoritmov za slovenski jezik, Optimalnega, Generičnega in Popovičevega.

## 4.1 Referenčni rezultat krnjenja

Za namene vrednotenja smo najprej oblikovali referenčni rezultat krnjenja, katerega značilnost je bila, da so se v njem nahajali krni, ki so brez napake zastopali besede v testnem korpusu. V testni fazi dobljene sezname končnic

in krnov smo ročno popravili in dopolnili in se poskušali kar najbolj približati idealnemu rezultatu krnjenja. V njem je bil torej prisoten samo po en krn za en semantičen pomen, in hkrati zastopani vsi semantični pomeni v korpusu, kar je idealen rezultat krnjenja. Enako velja tudi za krne, ki izvirajo iz lastnih imen. Istočasno pa smo poskrbeli, da v referenčnem rezultatu ni bilo nesprejemljivih krnov.

Naša odločitev je, da v tem primeru ne bomo spremljali ustreznosti samih krnov, marveč smo se posvetili besedam, ki jih določen krn zastopa. Skupine besed, ki dajo določen krn, smo oblikovali ročno, oziroma intelektualno (želeli smo se namreč kar najbolj približati idealnemu rezultatu krnjenja) in jih poimenovali referenčni razredi. Pri oblikovanju referenčnih razredov smo upoštevali nekatere dogovorjene principe.

### 4.1.1 Principi oblikovanja referenčnih razredov

V posameznem referenčnem razredu se nahajajo besede, ki sodijo skupaj glede na semantični pomen, ne glede na obliko. V naravnem jeziku, še zlasti pri oblikoslovno razgibanih jezikih, seveda naletimo na besede enakega pomena, ki so popolnoma različne (npr. *mečem – vržem, človek - ljudje*), zato se z avtomatskimi metodami obdelave naravnih jezikov idealnemu rezultatu vedno le približamo.

Pri oblikovanju referenčnih razredov smo zato upoštevali naslednje principe, pri katerih smo se opirali na semantiko:

- v en referenčni razred sodijo besede oziroma besedne vrste, ki imajo enak semantičen pomen, ne glede na vrsto tvorjenja oziroma oblikoslovne značilnosti;
- v primeru, da določena beseda zaradi semantike ne ustreza, jo uvrstimo v drug referenčni razred;
- po potrebi tak referenčni razred, oz. njemu pripadajoči krn, ustvarimo intelektualno.

Oblikovanje referenčnega razreda ilustrira naslednji primer:

Algoritem je izdelal naslednji referenčni razred, pripadajoč krnu *avtor*:

```
107/avtor/avtor,avtoric,avtorica,avtorice,avtorici,
avtorizacija,avtorizacije, avtorizacijo,avtorja,
avtorje,avtorjem,avtorjema,avtorjev,avtorjeva,avtorjeve,
avtorjevega,avtorjevem,avtorjevo,avtorji,avtorjih,
avtorju,avtorske,avtorskega,avtorski, avtorskih,avtorsko,
```

Vidimo, da so na seznamu besede, ki zaznamujejo vse, kar je v zvezi z osebo, ki je ustvarila neko delo, torej avtorjem (to so različne oblikoslovne oblike besede avtor in iz nje izpeljanih pridevnikov). V seznamu besed pa so se znašle tudi besede avtorizacija, avtorizacije in avtorizacijo, ki semantično ne ustrezajo ostalim. Zato jih izločimo, in oblikujemo nov ekvivalenčni razred:

```
0/avtoriz/avtorizacija,avtorizacije,avtorizacijo,
```

## 4.2 Metodologija vrednotenja

Isti korpus, iz katerega smo oblikovali referenčni rezultat, smo nato obdelali z vsakim izmed treh algoritmov. Za potrebe vrednotenja smo izhod obdelave testnega korpusa, ki ga je izdelal posamezen algoritem, združili v eno datoteko. To pomeni, da so se v njej nahajali vsi krni, ki jih je izdelal posamezen algoritem.

Datoteke z rezultati vsakega algoritma smo nato primerjali z referenčnim rezultatom. Ugotavljali smo, koliko se rezultat krnjenja korpusa s posameznim algoritmom približa referenčnemu – idealnemu. Šlo je za to, da ugotovimo, koliko ekvivalenčnih razredov z besedami enega referenčnega razreda izdela posamezen algoritem. V praksi so možne tri situacije; popolno ujemanje, premalo natančno ali premočno krnjenje. To pomeni, da določen algoritem lahko izdela en krn za vse semantično povezane besede, in se s tem izenači z referenčnim rezultatom. Algoritem lahko izdela tudi več kot en krn za en semantičen pomen, kar pomeni, da množico semantično enakih besed razdeli na nekaj manjših. Mogoče pa je seveda tudi, da se v množici besed, ki jih algoritem okrni na en krn, znajde beseda (ali več besed), ki bi zaradi svojega pomena sodila v drugo množico besed.

Naslednji primer prikazuje situacijo, ko algoritem za en referenčni razred izdela tri ekvivalenčne:

Referenčni razred:
```
/žel/želel,6;želela,5;želele,5;želeli,11;želene,1;
želenem,1; želenih,1; želeno,5;želi,24;želijo,31;
želite,1;želno,1; želim,6;želimo,57; želja,13; željah,3;
željami,2;želje,12;želji,5;željno,1;željo,9;
```
Ekvivalenčni razredi:
```
/žel/želel,6;želela,5;želele,5;želeli,11;želene,1; želenem,1;
želenih,1;želeno,5;želi,24;želijo,31;želite,1;želno,1;
/želim/želim,6;želimo,57;
/želj/želja,13;željah,3; željami,2; želje,12;želji,5;
željno,1;željo,9;
```

Pri vrednotenju je zato mogoče spremljati več dejavnikov, predvsem pa dva:

- število krnov (oz. množic besed), ki jih algoritem izdela za en semantičen pomen – kar je premalo natančno krnjenje;
- število napačno uvrščenih besed – oz. premočno krnjenje.

Odločili smo se, da se bomo pri vrednotenju delovanja algoritmov predvsem omejili na premalo natančno krnjenje. Le-to pri poizvedovanju vodi v zmanjševanje odziva, kar pomeni, da med zadetki ne bo vseh relevantnih dokumentov v podatkovni zbirki. Nismo testirali premočnega krnjenja, ki je sicer pomembno za natančnost. Pomeni, da so med zadetki tudi nerelevantni dokumenti. V primeru, da sistem npr. okrni besedi *avtor* in *avtorizacija* na isti krn, bi pri poizvedovanju dobili dokumente, ki govorijo o obojem, česar pa ne želimo. Zavedamo se, da prihaja tudi do tega pojava, ker pa smo izkustveno ugotovili, da je premalo natančno krnjenje veliko bolj izrazito, nas je to vodilo do odločitve, da ga izberemo kot kriterij vrednotenja.

Potrebno je sicer še enkrat opozoriti, da gre pri referenčnem rezultatu za ročno oz. intelektualno oblikovane semantično povezane skupine besed, kar je v realnem okolju živega jezika nemogoče doseči z avtomatskimi metodami krnjenja. Zato smo domnevali, da bomo pri pregledovanju rezultatov delovanja posameznega algoritma ugotovili, da se posamezen rezultat lahko idealnemu približa, lahko mu je enak, ne more pa ga preseči. Naša domneva je bila torej, da bodo med rezultati delovanja algoritmov značilne razlike, ki smo jih poskušali tudi statistično dokazati.

## 4.3 Rezultati primerjave algoritmov

### 4.3.1 Primerjava Optimalnega, Generičnega in Popovičevega algoritma

V statistični analizi smo uporabili neparametrični Kruskal-Wallisov test ali krajše H test, namenjen primerjavi več vzorcev (Adamič, 1995). Številu ekvivalenčnih razredov (oz. krnov), ki jih algoritem izdela za en referenčni razred, je bila izbrana naključna spremenljivka. Ničelna hipoteza $H_0$ trdi, da med algoritmi ni razlik, torej da se ne razlikujejo glede na število proizvedenih ekvivalenčnih razredov. Dobljene razlike so zgolj naključne. Nasprotno tej postavimo osnovno hipotezo $H_1$, da se algoritmi razlikujejo, oz. da število ekvivalenčnih razredov, ki jih izdelajo, ni enako.

Tabela 1 prikazuje rezultate delovanja vsakega izmed algoritmov. Prikazuje, koliko krnov oz. ekvivalenčnih razredov je izdelal posamezen algoritem v primerjavi z referenčnim rezultatom krnjenja. Dodani so tudi rangi posameznih rezultatov. Tako je npr. Optimalni algoritem v 5.563 primerih izdelal rezultat,

enak referenčnemu, torej en krn, ki zastopa vse semantično enake besede. Generični algoritem je to storil v 3.814 primerih, Popovičev pa v 5.016 primerih. Po dva krna, ki zastopata besede, uvrščene v en ekvivalenčni razred v referenčnem rezultatu, so izdelali: Optimalni algoritem v 463 primerih, Generični v 892 primerih in Popovičev v 769 primerih. Največje število izdelanih krnov za en semantičen pomen je 32, iz tabele vidimo, da jih je izdelal Generični algoritem.

**Tabela 1:** Rezultati delovanja algoritmov

| število ekv. razredov | Optimalni | Generični | Popovič | skupaj | rang |
|---|---|---|---|---|---|
| 1 | 5.563 | 3.814 | 5.016 | 14.393 | 7.197 |
| 2 | 463 | 892 | 769 | 2.124 | 15.456 |
| 3 | 104 | 471 | 195 | 770 | 16.903 |
| 4 | 42 | 269 | 69 | 380 | 17.477 |
| 5 | 23 | 185 | 29 | 237 | 17.786 |
| 6 | 3 | 133 | 14 | 150 | 17.979 |
| 7 | 4 | 94 | 8 | 106 | 18.108 |
| 8 | 3 | 73 | 3 | 79 | 18.200 |
| 9 | 1 | 51 | 0 | 52 | 18.265 |
| 10 | 1 | 46 | 0 | 47 | 18.315 |
| 11 | 0 | 28 | 0 | 28 | 18.353 |
| 12 | 0 | 24 | 0 | 24 | 18.379 |
| 13 | 0 | 33 | 0 | 33 | 18.407 |
| 14 | 0 | 14 | 0 | 14 | 18.431 |
| 15 | 0 | 13 | 0 | 13 | 18.444 |
| 16 | 0 | 18 | 0 | 18 | 18.460 |
| 17 | 0 | 14 | 0 | 14 | 18.476 |
| 18 | 0 | 8 | 0 | 8 | 18.486 |
| 19 | 0 | 9 | 0 | 9 | 18.495 |
| 20 | 0 | 4 | 0 | 4 | 18.502 |
| 21 | 0 | 2 | 0 | 2 | 18.504 |
| 22 | 0 | 2 | 0 | 2 | 18.506 |
| 23 | 0 | 2 | 0 | 2 | 18.509 |
| 25 | 0 | 1 | 0 | 1 | 18.510 |
| 26 | 0 | 1 | 0 | 1 | 18.511 |
| 27 | 0 | 2 | 0 | 2 | 18.513 |
| 28 | 0 | 2 | 0 | 2 | 18.515 |
| 31 | 0 | 1 | 0 | 1 | 18.516 |
| 32 | 0 | 1 | 0 | 1 | 18.517 |

| | | | |
|---|---|---|---|
| $T_v$ | 5,0311 x $10^7$ | 6,7683 x $10^7$ | 5,3454 x $10^7$ |
| $n_v$ | 6.207 | 6.207 | 6.103 |

$T_v$ – vsota rangov v posameznem vzorcu

$n_v$ – število enot v posameznem vzorcu

Izračunana vrednost $H$ = 930,4835

Zaradi velikega števila skupnih rangov se odločimo tudi za korekcijo. Izračunamo korekcijski koeficient $k_C$, s katerim delimo izračunano vrednost $H$.

$k_C$ = 0,5288

$H_C$ = 1759,6

Kritična vrednost hi-kvadrat pri 1% tveganju in dveh stopinjah prostosti je $\chi^2$ = 9,21. Dobljena vrednost $H_C$ je znatno večja kot kritična vrednost.

Ničelno hipotezo $H_0$ zavrnemo in sprejmemo osnovno hipotezo $H_1$, ki trdi, da rezultati krnjenja niso enaki, oz. da se število ekvivalenčnih razredov, ki jih proizvedejo algoritmi za en referenčni razred, značilno razlikuje.

V primerjavi z referenčnim rezultatom, ki obsega 6.209 krnov, jih Optimalni izdela 7.642, Popovičev 7.860 in Generični 15.930. Izračunamo še povprečno število ekvivalenčnih razredov za referenčni razred in standardno deviacijo. Rezultate prikazuje tabela 2.

**Tabela 2:** Povprečno število in standardne deviacije ekvivalenčnih razredov na referenčni razred

| Algoritem | $\bar{x}$ | $\delta$ |
|---|---|---|
| Optimalni | 1,2307 | 0,5381 |
| Popovičev | 1,2659 | 0,6995 |
| Generični | 2,5656 | 2,9044 |

Iz izračuna povprečnega števila in standardnih deviacij je očitno, da Generični algoritem izrazito odstopa od Popovičevega in Optimalnega, medtem ko sta si slednja dva glede na rezultate dokaj blizu. Zato smo izvedli še primerjavo Popovičevega in Optimalnega algoritma.

### 4.3.2 Primerjava Optimalnega algoritma s Popovičevim

Ničelna hipoteza $H_0$ tokrat trdi, da se algoritma glede na število ekvivalenčnih razredov ne razlikujeta. Preverimo jo s Kruskal-Wallisovim testom.

Iz tabele 3 vidimo, da Optimalni algoritem doseže natančno ujemanje z referenčnim rezultatom v 5.563 primerih, Popovičev pa v nekaj več kot pet tisoč. Popovičev algoritem izdela največ osem krnov za en pomen, medtem ko jih Optimalni izdela največ deset, kar pomeni, da se v tej značilnosti Popovičev algoritem izkaže celo kot boljši.

**Tabela 3:** Rezultati delovanja Optimalnega in Popovičevega algoritma

| **Število krnov** | **Optimalni** | **Popovič** | **skupaj** | **rang** |
|---|---|---|---|---|
| 1 | 5.563 | 5.016 | 10.579 | 5.290 |
| 2 | 463 | 769 | 12.32 | 11.196 |
| 3 | 104 | 195 | 299 | 11.961 |
| 4 | 42 | 69 | 111 | 12.166 |
| 5 | 23 | 29 | 52 | 12.248 |
| 6 | 3 | 14 | 17 | 12.282 |
| 7 | 4 | 8 | 12 | 12.296 |
| 8 | 3 | 3 | 6 | 12.306 |
| 9 | 1 | 0 | 1 | 12.309 |
| 10 | 1 | 0 | 1 | 12.310 |

| | | |
|---|---|---|
| $T_v$ | $3{,}6796 \times 10^7$ | $3{,}8978 \times 10^7$ |
| $n_v$ | 6.207 | 6.103 |

$H = 51{,}2455$

Korekcija

$k_C = 0{,}3643$
$H_C = 140{,}6703$

Kritična vrednost hi-kvadrat pri 1% tveganju in eni stopinji prostosti je $\chi^2 = 6{,}63$. Dobljena vrednost $H_c$ je večja kot kritična vrednost. Značilna razlika tudi v tem primeru kaže, da lahko ničelno hipotezo $H_0$ zavrnemo in sprejmemo osnovno hipotezo $H_1$, pri čemer je tveganje 1%. Ta trdi, da sta rezultata krnjenja obeh algoritmov različna.

## 4.4 Povzetek vrednotenja algoritmov

Z vrednotenjem algoritmov smo prišli do zaključka, da so algoritmi različni. Krnjenje enakega besedilnega korpusa proizvede tri izrazito različne rezultate.

Optimalni algoritem kaže najboljše rezultate, oz. se najbolj približa idealnemu rezultatu krnjenja. Generični algoritem se izkaže kot najmanj natančen. Popolno ujemanje z referenčnim rezultatom najdemo pri Optimalnem algoritmu v 5.563 primerih, pri Popovičevem v 5.016 primerih in pri Generičnem v 3.814 primerih.

Ob spremljanju največjega števila krnov za en semantičen pomen, smo ugotovili, da se tukaj najbolj izkaže Popovičev algoritem, saj v najslabšem primeru izdela osem krnov za en pomen. Optimalni algoritem jih sicer izdela devet oz. deset, vendar le v dveh primerih (0,032% vseh krnov), kar je zanemarljivo. Sicer pa je veliko bolj natančen od Popovičevega. Generični algoritem pa v 225 primerih izdela 10 ali več krnov za en semantičem pomen, od tega v dvanajstih več kot 20, v najslabšem primeru pa celo 32. Dejstvo postane razumljivo, če se spomnimo, da je glavna naloga Generičnega algoritma preprečevanje premočnega krnjenja (reduciranja besed z različnimi pomeni na enak krn). Sklepali smo, da izvrševanje te naloge rezultira v izrazito premalo natančnem krnjenju, ki je še posebej očitno v jeziku z bogato oblikoslovno strukturo, kot je slovenščina.

Omeniti je sicer potrebno, da so bili tako izraziti rezultati v prid optimalnega algoritma gotovo tudi posledica omejenega besedilnega korpusa, s katerim smo delali. Res je tudi, da ima intelektualno oblikovan referenčni rezultat krnjenja značilnosti, ki se jih ne da doseči z avtomatskimi metodami krnjenja, saj referenčni razredi vsebujejo tudi sinonime in druge oblike semantično povezanih besed, ki jih algoritem brez vgrajenih leksičnih pripomočkov kot npr. slovar ali tezaver ne more doseči.

## 5 Zaključek

Cilj dela, opisanega v prispevku, je bil razvoj in testiranje postopka avtomatskega krnjenja besedil slovenskega bibliotekarskega podjezika. Ta je potekal v treh fazah. V prvi fazi postopka je šlo za učenje algoritma z imenom Optimalni - na besedilnem korpusu iz temeljne slovenske bibliotekarske revije Knjižnica, referatov z nekaterih bibliotekarskih posvetovanj in diplomskih nalog s 790.000 besedami, ki je bil razdeljen na segmente po 15.000 besed. Cilj prve faze je bil zgraditi datoteke z veljavnimi in imenskimi krni, ki izvirajo iz besedišča s področja bibliotekarstva, in ki bi jih bilo mogoče uporabiti v nadaljnjih postopkih gradnje orodij za avtomatsko indeksiranje in pri poizvedovanju.

Po opravljeni učni fazi je sledila testna faza, v kateri smo testirali delovanje algoritma z vsemi pripadajočimi datotekami (seznami veljavnih in imenskih

krnov, seznamoma blokiranih besed in pravil za popravljanje krnov) na dodatnem, sicer manjšem besedilnem korpusu. Ta je obsegal 167.000 besed. Ugotovili smo, da se s krnjenjem besedna masa korpusa zmanjša za 92,48% in da je le 349 krnov takih, ki še niso uvrščeni na seznama sprejemljivih krnov.

Nato smo izvedli vrednotenje. V tej fazi smo primerjali tri algoritme, Optimalnega, Generičnega in Popovičevega. Primerjali smo rezultate njihovega delovanja z referenčnim rezultatom. To je Optimalni rezultat krnjenja, ki smo ga namenoma izdelali ročno, da bi dosegli njegovo maksimalno pravilnost. Posebno pozornost smo posvetili skupinam besed, ki so semantično povezane, zato se lahko zlijejo na določen krn. Za kriterij vrednotenja delovanja algoritmov smo si izbrali premalo natančno krnjenje. Spremljali smo, koliko krnov (oz. skupin besed, zastopanih z določenim krnom) izdela posamezen algoritem za en semantičen pomen v primerjavi z referenčnim rezultatom. Tako skupino besed, ki dajo krn, smo poimenovali ekvivalenčni razred. Možno je bilo popolno ujemanje z referenčnim rezultatom, torej da algoritem izdela en sam ekvivalenčni razred ali pa da jih izdela več. Hkrati smo se zavedali, da je mogoče spremljati tudi premočno krnjenje, a smo ugotovili, da se pojavlja manj pogosto. Zato ga nismo spremljali. Rezultate smo statistično obdelali, za statistično obdelavo pa smo izbrali neparametrični Kruskal-Wallisov test ali krajše H test.

Z eksperimenti in vrednotenjem smo prišli do zaključka, da najboljše rezultate med omenjenimi tremi algoritmi daje Optimalni algoritem. Največkrat doseže popolno ujemanje z referenčnim rezultatom in hkrati izdela najmanj krnov za en semantičen pomen. Sledi Popovičev algoritem, za tretjega, Generičnega, pa se je izkazalo, da je premalo natančno krnjenje njegova izrazita slabost. Ko rezultate optimalnega algoritma primerjamo z rezultati krnjenja medicinskega podjezika (Vilar in Dimec, 2001), lahko ugotovimo številne podobnosti. Zato lahko sklepamo na možnost uspeha algoritma tudi pri obdelavi besedil z drugih področij, seveda pa tudi za nadaljnje delo na področju obdelave bibliotekarskih besedil.

S pomočjo eksperimentov smo zgradili nabor krnov, ki zastopajo semantično povezane besede iz besedil, s katerimi smo delali. Menimo, da tak nabor lahko predstavlja osnovo za nadaljnjo gradnjo besedišča podjezika s področja bibliotekarstva. Seveda pa se zavedamo, da je za to potreben dodaten, mnogo večji korpus besedil. Z obdelavo dodatnih besedil lahko zgradimo relativno ažuren besednjak, uporaben v orodjih za avtomatsko indeksiranje, iskalnih orodjih za dokumente v elektronski obliki. Možna je celo uporaba za nekatera druga področja, npr. terminologijo ali vsebinsko obdelavo.

V tako zgrajenem naboru besed, pod pogojem, da je ažurno vzdrževan, se odraža dejansko stanje sodobnega znanstvenega in strokovnega jezika, kakršnega najdemo v pisnih virih na področju bibliotekarstva. Tako besedišče

vključuje tudi mnoge druge, z vidika bibliotekarstva, "mejne" vede in stroke, vse od računalništva in informatike, preko upravljanja in vodenja, do družbenih ved, pedagogike in psihologije.

Z eksperimenti smo tudi ugotovili, da je mogoče slovenski jezik, kljub njegovi razgibanosti, s kvalitetnim algoritmom in zadostno količino dodatnih pravil za preoblikovanje krnov dokaj uspešno modelirati.

Naša metoda, s svojim relativno enostavnim načinom dograjevanja in dopolnjevanja besednjaka, lahko predstavlja osnovo za nadaljnje delo na področju avtomatske obdelave naravnega slovenskega jezika. Spričo hitrosti razvoja bibliotekarske znanosti in stroke in spričo dejstva, da je jezik živ, nenehno se razvijajoč organizem, se zavedamo, da je na tem področju potrebnega še mnogo dela, in da lahko rezultati pričujočega magistrskega dela služijo kot pomoč pri načrtovanju nadaljnjih postopkov pri gradnji orodij za avtomatsko indeksiranje slovenskega jezika.

### Zahvala

Zahvaljujemo se vsem, ki so omogočili nastanek magistrskega dela, na katerem temelji prispevek. V prvi vrsti mentorici dr. Jasni Maver in avtorju uporabljene programske opreme in mnogih idej dr. Juretu Dimcu. Zahvala pa gre tudi uredništvoma revij Knjižnica in Šolska knjižnica, Zvezi bibliotekarskih društev Slovenije ter Narodni in univerzitetni knjižnici, ki so omogočili uporabo besedil v raziskovalne namene.

## Citirani viri

1. **Adamič, Š.** (1995). *Temelji biostatistike*. Ljubljana: Medicinska fakulteta.
2. **Dimec, J., Todorovski, L., Hristovski, D., & Džeroski, S.** (1999). The personalized search engine for Slovenian and English medical documents. V *Managing multimedia collections*. 23rd Library systems seminar, Bled, 21-23 April 1999 (str. 56-63). Ljubljana: National and University Library..
3. **Dimec, J., Todorovski, L., Hristovski, D., & Džeroski, S.** (2000). *Three new stemmers for Slovenian language.* Pridobljeno 29.2.2000 s spletne strani: http://www.mf.uni-lj.si/ds/new-stemmers.html
4. **Popovič, M.** (1991). *Implementation of a Slovene language free-text retrieval system: a study submitted in fulfilment of the requirements for the degree of Doctor of Philosophy at the University of Sheffield*. Sheffield: Department of Information Studies.

5. **Popovič, M., & Willett, P.** (1992). The effectiveness of stemming for natural language access to Slovene textual data. *Journal of the American Society for Information Science*, *43* (5), 384-390.

6. **Vilar, P., & Dimec, J.** (2000). Krnjenje kot osnova nekaterih nekonvencionalnih metod poizvedovanja. *Knjižnica*, *44* (4), 7-31.

7. **Vilar, P.** (2001). *Krnjenje slovenskih besedil s področja bibliotekarstva*. Magistrsko delo. Ljubljana: Oddelek za bibliotekarstvo, informacijsko znanost in knjigarstvo.

---

**Mag. Polona Vilar** je zaposlena kot asistentka na Oddelku za bibliotekarstvo, informacijsko znanost in knjigarstvo, Filozofska fakulteta, Univerza v Ljubljani
Naslov: Aškerčeva 2, 1000 Ljubljana
Naslov elektronske pošte: polona.vilar@ff.uni-lj.si

**Dr. Jasna Maver** je zaposlena kot docentka na Oddelku za bibliotekarstvo, informacijsko znanost in knjigarstvo, Filozofska fakulteta, Univerza v Ljubljani
Naslov: Aškerčeva 2, 1000 Ljubljana
Naslov elektronske pošte: jasna.maver@ff.uni-lj.si