Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Žiga Zupanec

Sistem za (pol)avtomatsko označevanje medicinskih izvidov z MKF kodami

diplomsko delo

visokošolski strokovni študijski program prve stopnje računalništvo in informatika

Ljubljana, 2013

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Žiga Zupanec

Sistem za (pol)avtomatsko označevanje medicinskih izvidov z MKF kodami

diplomsko delo

visokošolski strokovni študijski program prve stopnje računalništvo in informatika

Mentor: doc. dr. Luka Šajn

Ljubljana, 2013

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00417/2013
Datum: 05.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ŽIGA ZUPANEC**

Naslov: **SISTEM ZA (POL)AVTOMATSKO OZNAČEVANJE MEDICINSKIH IZVIDOV Z MKF KODAMI**

SEMI-AUTOMATIC SYSTEM FOR ANNOTATING MEDICAL REPORTS WITH ICF CODES

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Študent naj izdela sistem za (pol)avtomatsko označevanje medicinskih izvidov pacientov, ki prihajajo v rehabilitacijski center Soča. Uporabi naj oznake za mednarodno klasifikacijo funkcioniranja, zmanjšane zmožnosti in zdravja (MKF). Sistem naj omogoča uporabo spletnega portala, ki deluje na različnih spletnih brskalnikih in podpira sočasno uporabo različnim zdravnikom specialistom. Podatki naj se zapisujejo v bazo, ki jo lahko nadrejeni zdravnik pregleduje in potrjuje morebitno dodane nove kode.

Mentor:

doc. dr. Luka Šajn

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Nikolaj Zimic

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Žiga Zupanec, z vpisno številko **63090334**, sem avtor diplomskega dela z naslovom:

Sistem za (pol)avtomatsko označevanje medicinskih izvidov z MKF kodami

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Luke Šajna,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 4. julija 2013 Podpis avtorja:

Za strokovno vodenje, usmerjanje in strokovne nasvete se zahvaljujem svojemu mentorju, doc. dr. Luki Šajnu. Zahvaljujem se zdravnikom na Univerzitetnem rehabilitacijskem inštitutu Republike Slovenije – Soča, predvsem prof. dr. Heleni Burger, dr. med. Za pomoč in nasvete pri uporabi orodij za obdelavo naravnega jezika se zahvaljujem članom Odseka za tehnologije in znanja na Institutu Jožef Stefan.

Kazalo

Slike

Kratice

AJAX Asynchronous JavaScript and XML

CSS Cascading Style Sheets

EFRR European Federation for Research and Rehabilitation

HTML HyperText Markup Language

ICD International Statistical Classification of Diseases and Related Health Problems

ICF International Classification of Functioning, Disability and Health

JSON JavaScript Object Notation

MKB mednarodna statistična klasifikacija bolezni in sorodnih zdravstvenih problemov

MKF Mednarodna klasifikacija funkcioniranja, zmanjšane zmožnosti in zdravja

MPK Model-pogled-krmilnik

MVC Model-view-controller

PDF Portable Document Format

SPA single-page application

SZO Svetovna zdravstvena organizacija

UDP User Datagram Protocol

URL Uniform resource locator

WHO World Health Organization

Povzetek

Naš cilj je razbremeniti zdravnike pri delu, ki ni v neposredni povezavi z njihovim poklicem. Takšno opravilo je označevanje (kodiranje) medicinsko-tehnične dokumentacije – izvidov po klasifikaciji MKF (angl. ICF). V ta namen smo v sodelovanju z Univerzitetnim rehabilitacijskim inštitutom Republike Slovenije – Soča razvili spletno aplikacijo, ki z uporabo metod umetne inteligence pomaga zdravnikom pri kodiranju izvidov. Problem rešujemo z našo spletno aplikacijo, ki čas kodiranja izvidov zmanjša, postopek kodiranja pa postane bolj preprost.

Abstract

Our goal is to reduce the time physicians spend for tasks that are not closely related to their primary practice. Annotating medical reports is one of those tasks. We developed in cooperation with the University Rehabilitation Institute Republic of Slovenia – Soča a web application that helps doctors annotate medical reports using methods of artificial intelligence. Our web application reduces the time and eases the annotation process.

Poglavje 1

Uvod

Zdravniki se pri svojem delu srečujejo z obilico opravil. Da bi lahko več časa namenili bolnikom, jih je treba razbremeniti na birokratskem področju. Med takšna opravila spada tudi označevanje izvidov po klasifikaciji MKF. Klasifikacija je plod večletnih prizadevanj Svetovne zdravstvene organizacije SZO (angl. WHO), da se poenoti opis posameznikovega funkcioniranja, posameznikovih dejavnosti in posameznikovega okolja. Klasifikacija služi kot ogrodje, po katerem zdravniki določajo kode. Kodiranje je proces, pri katerem se opisi izvidov preslikajo v kode ujemajočih se definicij [5].

Prednosti enotnega označevanja prinašajo koristi tako zdravstvenem osebju kot tudi pacientu. Klasifikacija MKF se ne osredotoča zgolj na pacientovo diagnozo, saj zajema tudi posameznikovo širše okolje in delovanje. To pripomore k natančnejšemu odkrivanju vzrokov in k boljši razlagi pacientove diagnoze. Poenotenje opisov prinaša tudi boljše sodelovanje in večje razumevanje pacientovega stanja med različnimi vejami medicine. Ker dobijo opisi posameznikov kodo, ki je mednarodno usklajena, se poenostavi sodelovanje med medicinskim osebjem preko meja držav [5].

Za pomoč pri označevanju zdravniki uporabljajo priročnik MKF (v nadaljevanju priročnik). Zdravnik kodo določi na podlagi definicije, kar zahteva poznavanje vseh definicij in ustrezne kode. Priročnik je neke vrste slovar, kjer zbrane kode predstavljajo indeks, poleg njih pa je opis kode (definicija).

Iskanje ustrezne kode je tako za novince kot tudi izkušene zdravnike časovno zelo zamudno opravilo [2], ker je miselni proces določanja kod ravno nasproten. Zdravnik kodo določi na podlagi definicije, kar zahteva poznavanje vseh definicij in ustrezne kode.

Da bi zdravnikom olajšali in pospešili proces kodiranja, smo v sodelovanju z Univerzitetnim rehabilitacijskim inštitutom Republike Slovenije – Soča (v nadaljevanju URI – Soča) razvili aplikacijo, ki z uporabo metod umetne inteligence, natančneje z uporabo metod za obdelavo naravnega jezika, pri obdelovanju izvidov predlaga kode, ki so potencialni kandidati v opisu posameznikovega stanja.

V drugem poglavju predstavimo problem in podatke, ki jih uporabljamo v aplikaciji. Tretje poglavje opisuje delovanje metod in tehnik, ki jih uporabljamo pri obdelavi naravnega jezika. V četrtem poglavju so predstavljeni rezultati – spletna aplikacija in z njo povezane pomožne komponente. Delovanje aplikacije na konkretnih izvidih orišemo v petem poglavju. V zaključku predstavimo prostor za izboljšave in komentiramo pridobljene rezultate.

Poglavje 2

Opis problema in podatkov

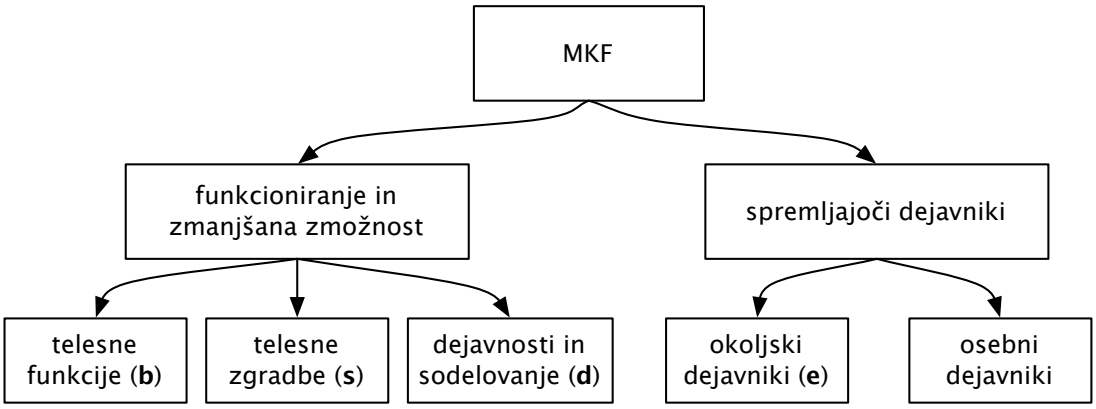
2.1 Opis problema

Zaradi velikega števila kod (preko 1420) uporabniki pri kodiranju uporabljajo priročnik. Priročnik je v slovenščini na voljo le v knjižni obliki, zato je listanje po iskanju definicij kod zelo nepraktično. Iskanje ustrezne kode je tudi časovno zelo zamudno opravilo, ker miselni proces določanja kod temelji na poznavanju definicij. Te so zaradi preglednosti hierarhično urejene (slika 2.2). Takšna ureditev je za uporabnika še vedno zamudna in ne ustreza vsem uporabnikom. Priročnik sicer vsebuje abecedno kazalo, ki pa pri takem opravilu ne pripomore veliko. Dober pripomoček, ki bi ga zdravniki potrebovali pri procesu kodiranja, je tako nemogoče ponuditi v knjižni obliki. Določitev vseh ustreznih kod za en izvid lahko traja tudi več kot eno uro [8].

2.2 Opis podatkov

2.2.1 Priročnik

Priročnik vsebuje Mednarodno klasifikacijo funkcioniranja, zmanjšane zmožnosti in zdravja (v nadaljevanju MKF). Razdeljen je na dva dela, ki se nadalje delita na razdelke (slika 2.1). Razdelki se delijo na področja, ta pa

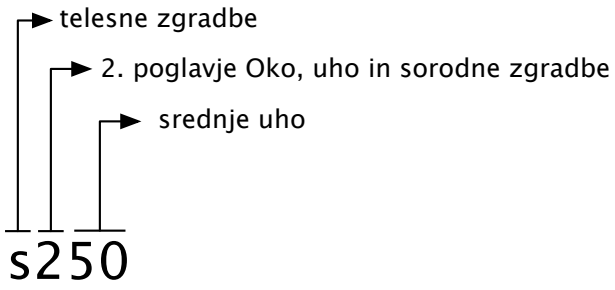


Slika 2.1: Zgradba MKF, povzeto po [5].

na kategorije, ki so klasifikacijska enota. Posamezna kategorija ima do štiri ravni. Globlja kot je raven, bolj specialno je določena kategorija. Na četrti in najgloblji ravni je končnih kod 1424. V okviru diplomskega dela napovedujemo kategorije na drugi ravni, ki obsega 362 kod. Druga raven povsem zadošča za praktično rabo klasifikacije v medicinskih ustanovah. Tretja oziroma četrta raven se običajno uporablja v specialističnih zavodih, prva raven pa je dovolj za vodenje statistike ali za populacijske raziskave. Vsaka koda je sestavljena iz črke, ki predstavlja razdelek, in iz niza številk (slika 2.2). Prva številka pomeni številko poglavja, sledi pa ji dvomestna številka, ki predstavlja drugo raven [5]. V diplomskem delu tretjo in četrto raven, ki jo predstavljata številki na mestu tri in štiri, uporabljamo za povečanje točnosti pri napovedovanju kod na drugi ravni.

2.2.2 Medicinsko-tehnična dokumentacija

Izvid je sestavljen iz imena, priimka in datuma rojstva pacienta, diagnoze, opisa pacientove težave (anamneze), trenutnega stanja pacienta ter mnenja zdravnika. Izvid je v procesu označevanja vhodni dokument, nad katerim izvedemo postopek označevanja. Primer izvida je na sliki 2.3.



Slika 2.2: Zgradba kode.

Medicinsko tehnična dokumentacija

[redacted]

Diagnoze: Parapareza po mielomeningokeli

Anamneza: Deklica je drugo rojeni otrok. V začetku nosečnosti je mati prebolela virusno vnetje zgornjih dihalnih poti, sicer je bila zdrava in v nosečnosti ni imela težav. Deklica se je rodil dva dni pred rokom. Med porodom ni bilo zapletov. Ocenjena je bila z Apgard 9/ 9/ 9. Takoj ob rojstvu so opazili mielomeningokelo in parezo spodnjih udov. V starosti 1 meseca je bila operirana.

b310-0 b730-1

Deklica sicer lepo napreduje, nadzoruje glavo, se obrača, govori. Pri plazenju se opira le na roki, nogi pasivno vleče za seboj. Bila je vključena v nevrorazvojno fizioterapijo. Prihajajo zaradi možnosti oskrbe z ortozo s katero bi se deklica začela postavljati in hoditi.

s760

Status: Je živahna in pri pregledu lepo sodeluje. Hrbtenica je ravna, ledvene lordoze ni, morda rahla kifoza. Vidna postoperativna brazgotina, ki je cela.

d455 s810 d445 d410

Deklica se samostojno plazi, premika se le z rokami, samostojno se obrača, nadzor glave in zgornjega dela trupa je dober. V spodnjih udih ni aktivne gibljivosti, pasivna popolna. Prisotne so motnje zaznavanja, ki jih je težko natančno oceniti.

b730 b710 b156

Vode ne nadzoruje, blata tudi ne.

b620 b525

Mnenje: Starše seznanimo z ortotičnimi možnostmi, deklici predpišemo ustrezno ortozo in gre na odvzem mere.

e310 e410

Slika 2.3: Izvida iz URI – Soča.

Poglavje 3

Metode in orodja

V tem poglavju predstavimo metode, tehnike in orodja, ki se uporabljajo pri predobdelavi besedila. Četrto poglavje predstavlja konkretno implementacijo in podrobnosti.

3.1 Predobdelava besedila

Predobdelava besedila se začne z razbitjem besedila na posamezne povedi in nadalje na posamezne besede. Besedam se določi besedno vrsto in stavčni člen.

3.1.1 Lematizacija besed

Lematizacija je postopek, pri katerem določimo besedam njihove osnovne oblike. Z lematizacijo zmanjšamo čas, potreben za nadaljnjo obdelavo besede, pri čemer ohranimo njen pomen. Zmanjša se tudi prostorska kompleksnost, kar se pozna predvsem pri pregibnih jezikih. Tako ni potrebno več shranjevati vseh skladenjskih ali spregatvenih oblik besede.

Za lematizacijo besed smo prvotno uporabili prosto dostopni spletni servis JOS ToTaLe. Rezultat, ki ga po analizi vrne servis prikazuje slika 3.1.

Pozneje smo zaradi zahteve po hermetično zaprtem sistemu prešli na Označevalnik Obeliks (ob-likoslovni označ-e-valnik za s-lovenščino), ki je bil

```
beseda    oznaka  osnovna oblika
--------+-------+---------------
Po        Dm      po
operaciji Sozem   operacija
še        L       še
vedno     Rsn     vedno
pada      Ggnste  padati
desno     Ppnset  desni
stopalo   Soset   stopalo
.         .       .
```

Slika 3.1: Izpis oblikoskladenjskih oznak servisa JOS ToTaLe.

razvit na Institutu Jožef Stefan. Označevalniku kot vhod določimo besedilo za označevanje, model za označevanje in model za lematizacijo. Označevalnik izpiše podano besedilo v formatu TEI-XML. Prednost Oblikoslovnega označevalnika je tudi ta, da je prosto dostopen in za delovanje ne potrebuje zunanje povezave.

3.1.2 Odstranjevanje nepotrebnih besed

S tehniko odstranjevanja nepotrebnih besed (angl. stopwords) odstranimo besede, ki ne prispevajo s tematiko povezane informacije. Nepotrebne besede v grobem delimo v dve kategoriji. Generične nepotrebne besede so besede, ki ne nosijo informacije ne glede na tematiko. V to skupino spadajo slovnične besedne vrste: predlog, veznik, členek in medmet. Druga kategorija so besede, ki se pogosto pojavljajo le na domensko specifičnem področju. Domensko specifične nepotrebne besede so z razliko od generičnih nepotrebnih besed v tipičnih besedilih zelo redke [7]. Njihova relativna redkost naraste šele v besedilih znotraj določene stroke. Pogostejša kot je beseda, manjša je njena vloga pri ločevanju posameznih razredov znotraj dokumenta.

```
beseda      koren
----------+--------
interval    inter
intervalih  inter
intervalov  inter
pulzira     pulz
veliko      velik
```

Slika 3.2: Besede in koreni besed.

3.1.3 Korenjenje besed

Korenjenje besed je tehnika, pri kateri besedam, navadno že pretvorjenim v osnovne oblike, odstranimo končnice. Na ta način dobimo koren besede, kar zmanjša časovno in prostorsko kompleksnost ter hkrati zmanjšuje verjetnost, da bi bil relevanten dokument izpuščen pri iskanju. S tem se povečuje priklic R, ki je definiran kot delež pravih dokumentov a (tistih, ki jih uporabnik želi) med vsemi pomembnimi dokumenti n [3], kar posledično poveča uspešnost iskanja.

$$R = a/n. \tag{3.1}$$

Pri korenjenju ni nujno, da dobljeni koren predstavlja smiselno, slovnično pomensko besedo. Dovolj je, da se ohrani bistvo besede, po katerem se beseda loči od drugih.

Za korenjenje smo uporabili programsko knjižnico libstemmer (`http://snowball.tartarus.org/`), ki implementira jezik za korenjenje besed Snowball. Jeziku za korenjenje besed Snowball določimo seznam pravil, po katerih se koreni besede.

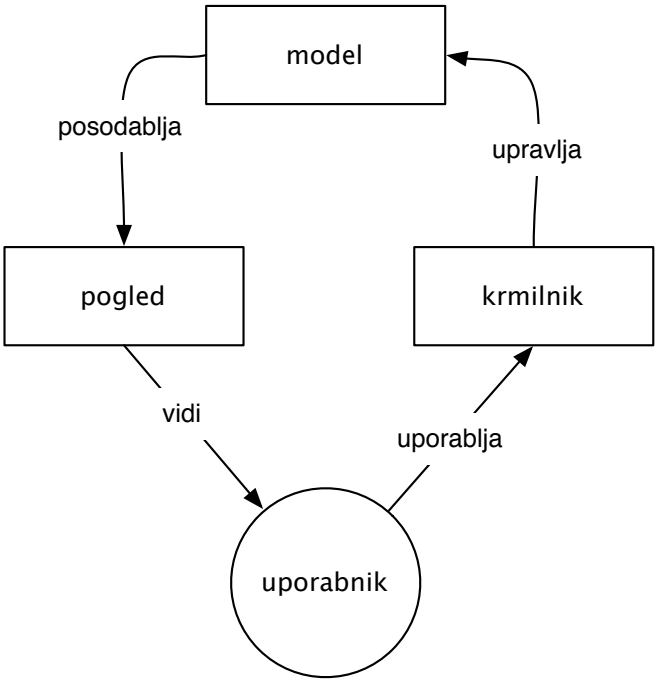
3.2 Orodja

3.2.1 Podatkovna baza

Podatkovna baza je mehanizirana, večuporabniška, formalno definirana in centralno nadzorovana zbirka logično povezanih podatkov [6]. Uporabili smo sistem za upravljanje z objektno-relacijskimi podatkovnimi bazami postgreSQL (`http://www.postgresql.org/`). Baza predstavlja model v arhitekturi MPK (angl. MVC) (slika 3.3). Model sestoji iz strukturno organiziranih podatkov, potrebnih pri analizi medicinskih izvidov.

3.2.2 Spletno aplikacijsko ogrodje

Uporabili smo spletno aplikacijsko ogrodje Django, napisano v programskem jeziku Python (`https://www.djangoproject.com/`). Ogrodje Django je zasnovano po principu arhitekture MPK (slika 3.3). Podatkovna baza predstavlja model, sistem za spletne predloge predstavlja pogled, krmilnik pa predstavlja upravljavca naslovov URL. Ogrodje Django vsebuje preprost spletni strežnik, sistem za branje in pisanje podatkov po standardu JSON, sistem za upravljanje spletnih predlog in sistem za krajevno prilagajanje. Ogrodje Django ima tudi sistem za urejanje uporabnikov, sistem za prijavo in odjavo ter sistem za vodenje sej. Aplikacije, ki temeljijo na ogrodju Django, omogočajo sočasno uporabo več uporabnikom.



Slika 3.3: Povezave v modelu MPK.

Poglavje 4

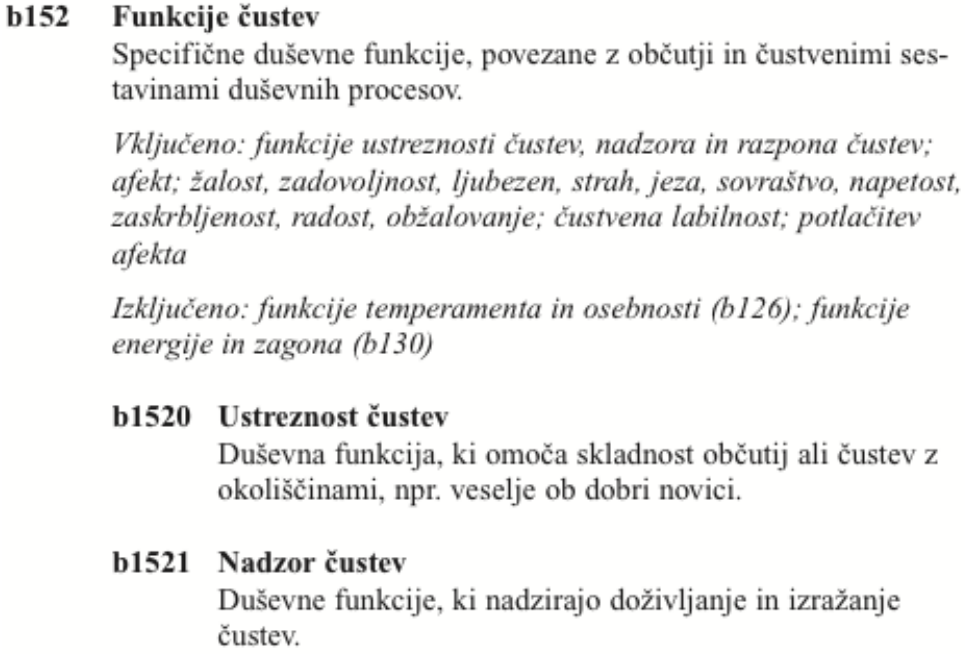
Rezultati

4.1 Razčlenjevalnik dokumentov PDF

Priročnik je bilo treba pretvoriti v elektronsko obliko (slika 4.1), primerno za nadaljnjo obdelavo. Iz dobljenega dokumenta v formatu PDF smo izluščili golo besedilo. Za to smo uporabili program, ki smo ga ob uporabi zunanjih knjižnic napisali v jeziku C. Program je prosto dostopen na naslovu `https://github.com/agiz/pdftotxt`. V dobljenem besedilu se navadno pojavi nekaj napak, ki jih je treba popraviti ročno. Gre za drobne, a pričakovane napake, na primer podobnost med 1 (ena), l (mali L) in I (veliki i) ipd. Tako pripravljeno besedilo je osnova za izgradnjo korpusa.

4.2 Korpus

Korpus smo zgradili na osnovi priročnika (slika 4.1). Nad besedilom, pretvorjenim v elektronsko obliko, smo opravili sintaktično analizo, katere podrobnosti so predstavljene v tem poglavju. Semantika besed presega obseg tega diplomskega dela in je predlagana v zaključku. Obdelano besedilo smo shranili v podatkovno bazo PostgreSQL (sliki 4.2 in 4.3). Vsako besedo v besedilu pretvorimo najprej v osnovno obliko ter nato v koren besede, da bi se izognili kompleksnosti in drobljenju informacije. Vsak koren besede v



b152 Funkcije čustev
Specifične duševne funkcije, povezane z občutji in čustvenimi sestavinami duševnih procesov.

Vključeno: funkcije ustreznosti čustev, nadzora in razpona čustev; afekt; žalost, zadovoljnost, ljubezen, strah, jeza, sovraštvo, napetost, zaskrbljenost, radost, obžalovanje; čustvena labilnost; potlačitev afekta

Izključeno: funkcije temperamenta in osebnosti (b126); funkcije energije in zagona (b130)

b1520 Ustreznost čustev
Duševna funkcija, ki omoča skladnost občutij ali čustev z okoliščinami, npr. veselje ob dobri novici.

b1521 Nadzor čustev
Duševne funkcije, ki nadzirajo doživljanje in izražanje čustev.

Slika 4.1: Primer zapisa besedila v knjigi.

besedilu skupaj s kodo tvori indeks, poleg pa shranimo še število pojavitev korena te besede v definiciji dane kode (slika 4.4).

4.3 Servis za korenjenje

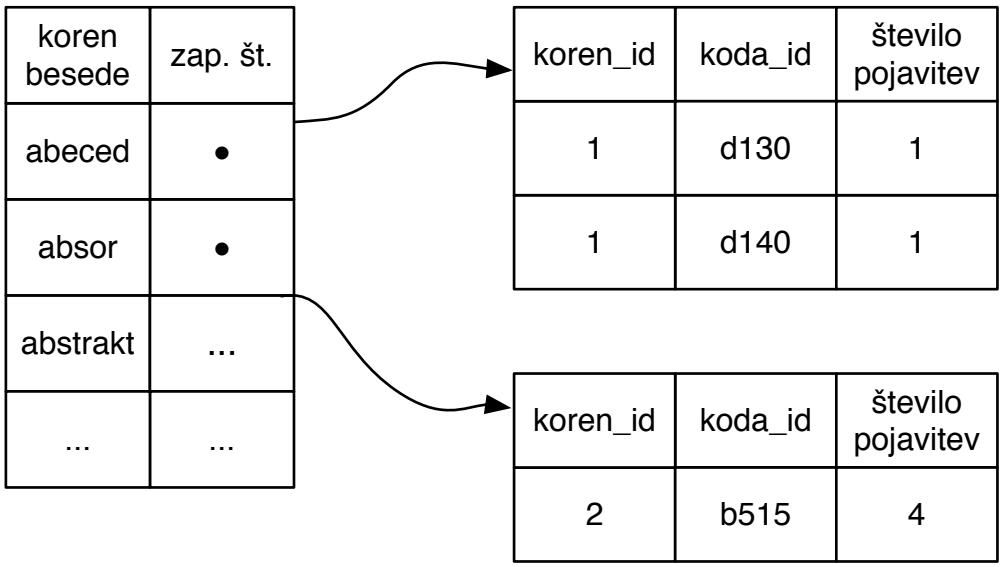
Servis za korenjenje je samostojna komponenta, napisana v programskem jeziku C. Kot vhod podamo datoteko s seznamom besed v osnovni obliki, na izhodu pa dobimo novo datoteko, ki vsebuje korene besed. Ker knjižnica libstemmer ne vsebuje pravil za korenjenje besed v slovenščini, smo pravila za korenjenje dodali po predlogi, dobljeni na naslovu `http://snowball.tartarus.org/archives/snowball-discuss/0670.html` [1]. Primer pravila za odstranjevanje končnice $-ovski$ je na sliki 4.5.

```
ICF_1=# \d api_code
               Table "public.api_code"
     Column     |          Type           | Modifiers
----------------+-------------------------+-----------
 tag            | character varying(16)   | not null
 tag_int        | integer                 | not null
 parent_tag     | character varying(16)   | not null
 parent_tag_int | integer                 | not null
 title          | character varying(255)  | not null
 description    | character varying(2048) | not null
 chapter        | integer                 | not null
 subchapter     | integer                 | not null
 subsubchapter  | integer                 | not null
```

Slika 4.2: Organizacija atributov v tabeli *api_code*.

```
ICF_1=# select tag, parent_tag, title from api_code;
  tag   | parent_tag | title
--------+------------+--------------------------------
 b152   | 0          | Funkcije čustev
 b1520  | b152       | Ustreznost čustev
 b1521  | b152       | Nadzor čustev
 b1522  | b152       | Razpon čustev
 b1528  | b152       | Funkcije čustev, drugo neopredeljeno
 b1529  | b152       | Funkcije čustev, neopredeljeno
```

Slika 4.3: Vrednost nekaterih atributov v tabeli *api_code*.



Slika 4.4: Logična predstavitev korpusa v podatkovni bazi.

```
define remove_ovski as (
  [substring] among (
    'ovski' (delete)
  )
)
```

Slika 4.5: Pravilo za odstranjevanje končnice $-ovski$.

4.4 Označevalni servis

Označevalni servis je samostojna komponenta, ki temelji na Označevalniku Obeliks. Označevalnik smo predelali tako, da preko protokola UDP sprejme ime datoteke, v kateri je besedilo pripravljeno na označevanje. Ko Označevalnik konča proces označevanja, zapiše rezultat v izhodno datoteko v formatu TEI-XML, prvotnega prosilca pa po protokolu UDP obvesti, da je proces končan. Označevalnik smo prilagodili za delo na večjedrnih napravah. Zaradi načina delovanja procesa označevanja se čas, potreben za označevanje, zmanjšuje obratno sorazmerno s številom jeder. Servis omogoča hkratno procesiranje več zahtev. Prednost delovanja v obliki servisa je, da se inicializacija opravi samo enkrat – na začetku. Tako se zmanjša čas, ki se drugače porabi za izgradnjo struktur in logike za označevanje.

4.5 Spletna aplikacija

4.5.1 Uporabniški vmesnik

Izgled spletne aplikacije je moral zadostiti določenim pogojem zaradi tehničnih omejitev opreme, ki se uporablja v medicinskih zavodih. Aplikacija mora biti odzivna, delovati mora na starejših računalnikih z brskalnikom Internet Explorer 6.0. Izgled aplikacije mora biti pregleden tudi na monitorjih z ločljivostjo 800 x 600. Uporabniški vmesnik mora biti preprost za uporabo. Krivulja učenja uporabe mora biti prilagojena uporabnikom, ki niso vešči upravljanja z računalnikom. Tipična raba aplikacije mora biti mogoča samo s tipkovnico ali pa samo z miško.

Uporabniški vmesnik sledi principu »one size fits all«, kar pomeni, da je izgled na različnih napravah enak. Spletni del aplikacije je napisan v jeziku za oblikovanje večpredstavnostnih dokumentov HTML. Spletni del aplikacije uporablja filozofijo »vse na eni strani« (angl. SPA). To pomeni, da se zamenjave pogledov opravljajo, ne da bi zato bilo potrebno ponovno nalaganje strani. Ta tehnika je mogoča z uporabo spletne tehnologije AJAX. Posledica

takega pristopa je, da se bistveno poveča odzivnost aplikacije, uporabnikom pa ponudi izkušnjo, podobno namiznim aplikacijam.

Krmilni del spletne aplikacije je napisan v programskem jeziku JavaScript. Ker različni brskalniki uporabljajo različne interpreterje jezika JavaScript, kar se v praksi kaže v različnih ukazih za dosego enakega učinka, smo za poenotenje programske kode uporabili knjižnico jQuery. jQuery je zbirka funkcij, ki same prepoznajo brskalnik in uporabijo ustrezne ukaze. Posledica uporabe knjižnice je bolj pregledna koda in hitrejše pisanje kode, saj programerju ni treba poznati vsake posebnosti različnih brskalnikov.

Povečano preglednost aplikacije ter postavitev in obliko elementov smo dosegli z uporabo prekrivnih slogov (angl. CSS). Kompaktnost zagotovimo tako, da na istem mestu uporabljamo različne elemente, primerne trenutnem opravilu. Polje za vnos besedila po obdelavi besedila zamenja vsebnik, ki ločeno prikazuje povedi. Prikazovalnik predlaganih kod (slika 5.2), preglednica kod za izbrano besedo (slika 5.6) in dodajanje nove kode (slika 5.5) so kot zavihki združeni v enotnem vsebniku.

4.5.2 Razčlenjevalnik TEI-XML

Za analizo in razčlenjevanje smo zgradili razčlenjevalnik TEI-XML. Razčlenjevalnik iz vhodnega besedila zgradi drevesno strukturo, ki je razumljiva programskemu jeziku Python.

4.5.3 Odstranjevanje nepotrebnih besed

Vsako besedo se po njeni osnovni obliki primerja z bazo nepotrebnih besed. Besede, ki so označene kot nepotrebne, se odstrani. Seznam tipičnih generičnih besed smo zgradili iz nepotrebnih besed, zbranih na naslovu `http://nl.ijs.si/GNUsl/lex/stop/`. Za določitev besed, ki se pogosto pojavljajo, smo napisali pomožno aplikacijo v programskem jeziku Python. Aplikacija zgradi seznam besed z največjo frekvenco v korpusu. Primer take pogoste besede v korpusu je beseda *funkcije*. Seznamu generičnih nepotrebnih

besed smo dodali dvajset najpogostejših domensko specifičnih nepotrebnih besed.

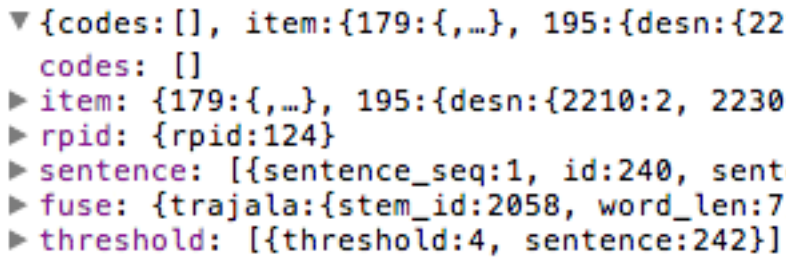
4.5.4 Ujemanje

Ujemanje se v celoti opravi na odjemalčevi strani. Ko odjemalec pošlje novo vnešeno besedilo v obdelavo, po obdelavi, ki traja največ nekaj sekund, nazaj dobi podatke v formatu JSON (slika 4.6). Podatki sestojijo iz:

- codes: seznam potrjenih kod;
- item: vsebuje indeks korena, koren besede in seznam kod ter frekvenco korena v definiciji kode. Ker klasifikacijo opravljamo na drugi ravni, združimo definicije iz tretje in četrte ravni s prvotno definicijo;
- rpid: identifikacijska številka izvida;
- sentence: seznam povedi;
- fuse: besedam v povedih poišče njihove korene;
- threshold: za vsako poved hrani trenutno nastavljen prag senzitivnosti.

Iz dobljenih podatkov se sestavi naslednje strukture (slika 4.7):

- Slovar ključnih besed, kjer posamezna beseda v povedi predstavlja indeks, vrednost pa je kazalec na seznam korenskih besed.
- Seznam korenskih besed, kjer ima vsaka korenska beseda svoj indeks.
- Urejen seznam kod, kjer vrednost predstavlja padajoče urejen seznam kod po številu ponovitev korenske besede v definiciji te kode.
- Neurejen seznam kod, kjer je vrednost slovar kod s številom pojavitev korenske besede z danim indeksom.



Slika 4.6: Podatki za izgradnjo struktur.

Nato se za vsako posamezno poved po korenih posameznih besed v neurejenem seznamu kod poišče vse kode in število njihovih ponovitev. Rezultate se shrani v podatkovno strukturo, kjer se za vsako poved hrani padajoče urejen seznam kod po kumulativnem številu ponovitev vseh korenskih besed v povedi za posamezno kodo (slika 4.8). Sestavi se tudi slovar kod s kumulativnim številom pojavitev v definiciji posamezne kode za vse besede v povedi.

4.5.5 Določanje praga za prikaz kod

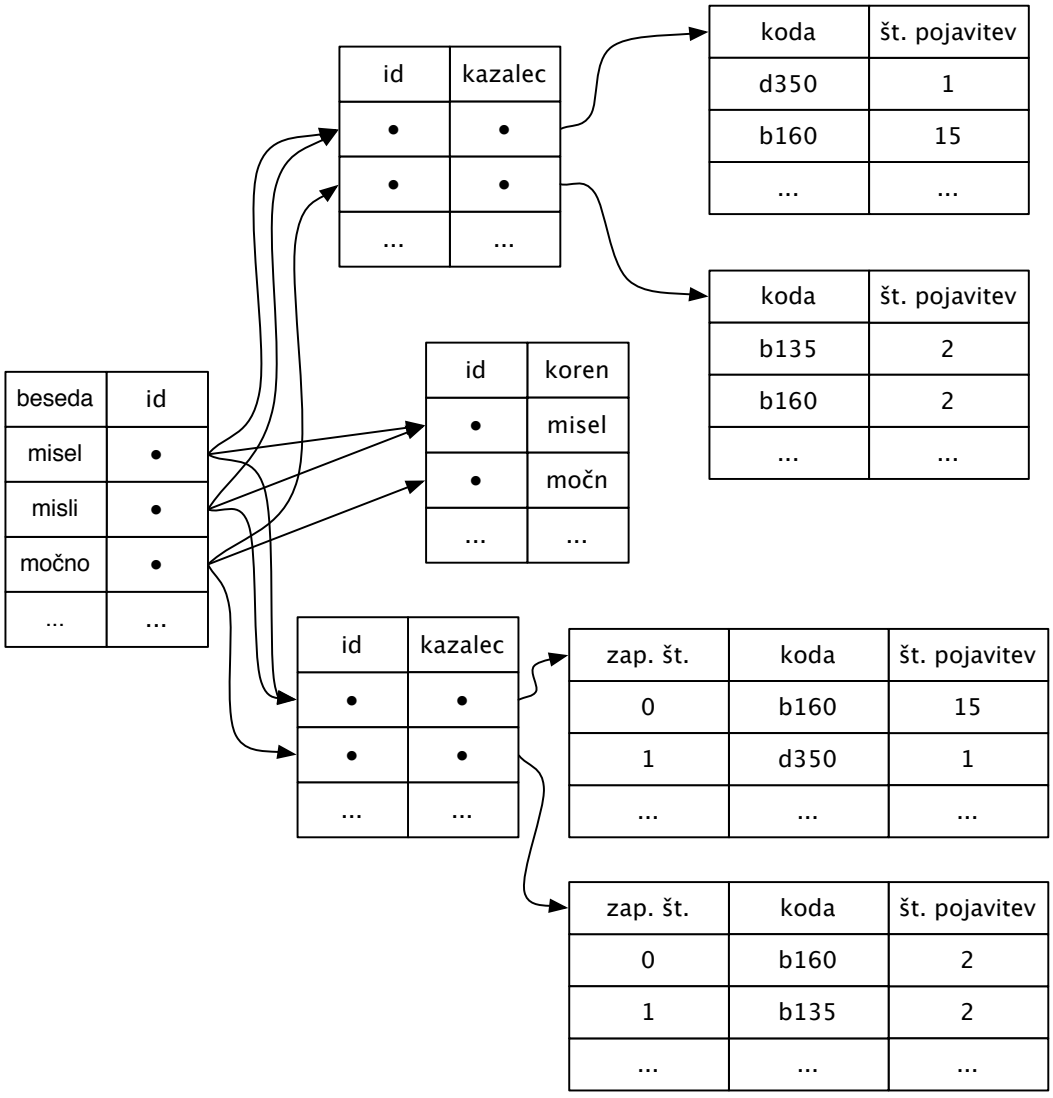
Prag je ločnica na padajoče urejenem seznamu kod v povedi. S spreminjanjem praga vplivamo na število prikazanih kod, pri čemer je m število ponujenih kod, od tega a relevantnih (prikazanih). Prag je definiran kot

$$P = a/m. \tag{4.1}$$

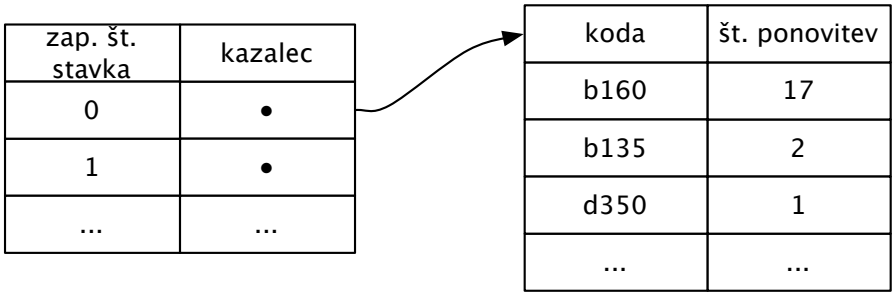
Nižji kot je prag, več je prikazanih kod.

4.6 Delovni tok sistema

Sistem je modularen, sestavljen iz posameznih komponent. Posamezna komponenta sistema je logično zaključena celota za določen namen in jo je možno uporabiti povsem ločeno in neodvisno od ostalih komponent.

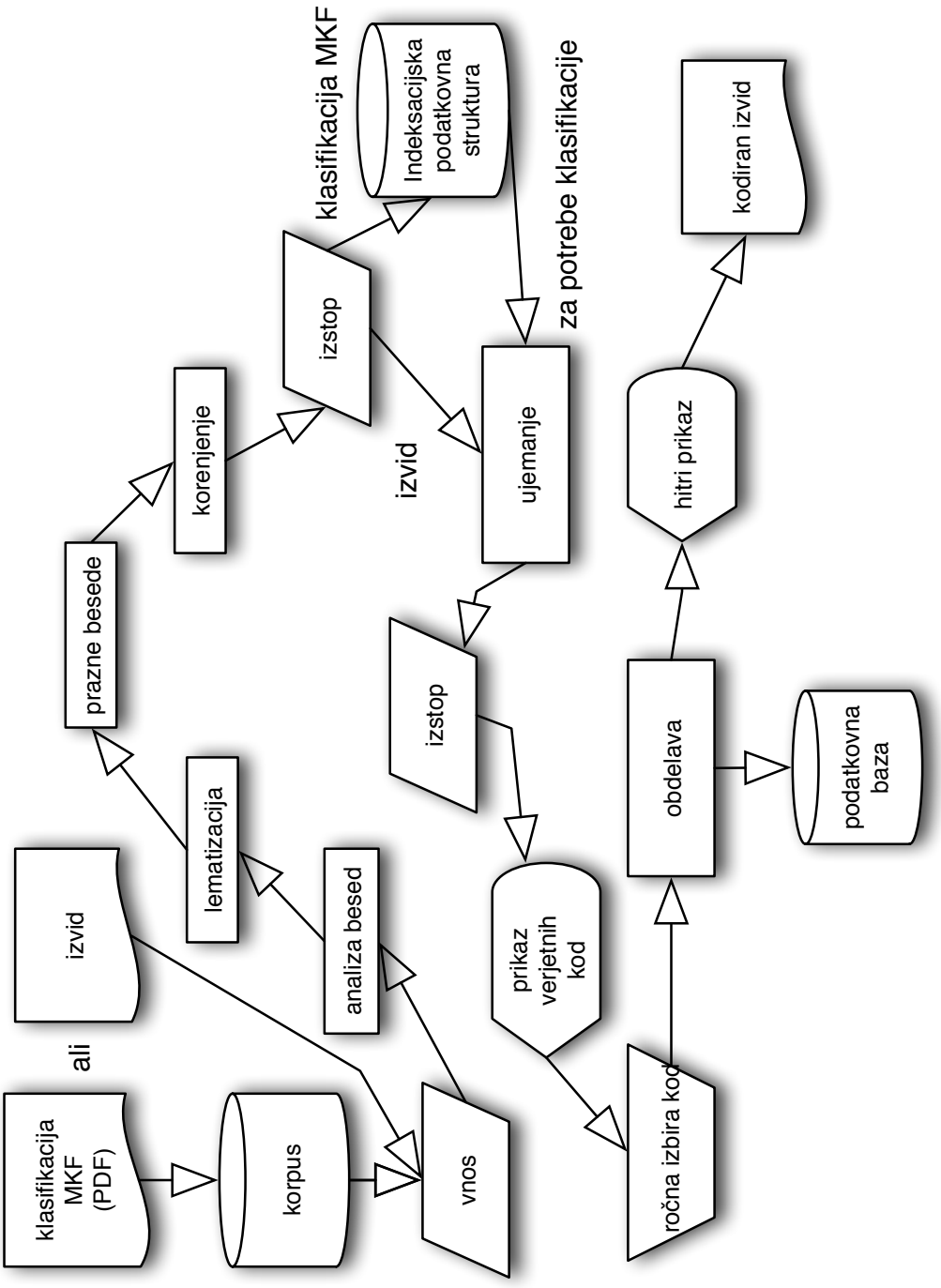


Slika 4.7: Strukture za ujemanje in relacije med njimi.



Slika 4.8: Struktura ujemanjočih kod po posameznih povedih.

Uporabnik vnese izvid v za to namenjeno vnosno polje v brskalniku. Podatki se pošljejo na strežnik, kjer vsebina izvida vstopi v proces lematizacije. Na izhodu se iz dobljenega besedila, strukturiranega v formatu TEI-XML, po razčlembi odstrani nepotrebne besede. Ostale besede se korenijo. Znanim korenom se pripišejo ustrezni indeksi. Indeksom se poišče kode in število pojavitev indeksa v definiciji posamezne kode. Besede, koreni besed s kodami in število pojavitev se pošljejo nazaj brskalniku. Ta zamenja pogled (slika 4.10), kjer prikaže kode kot potencialne kandidate za posamezne povedi.



Slika 4.9: Diagram delovnega toka sistema.

Slika 4.10: Prikaz kod v posamezni povedi.

Poglavje 5

Uporaba spletne aplikacije

5.0.1 Vnos besedila in kodiranje

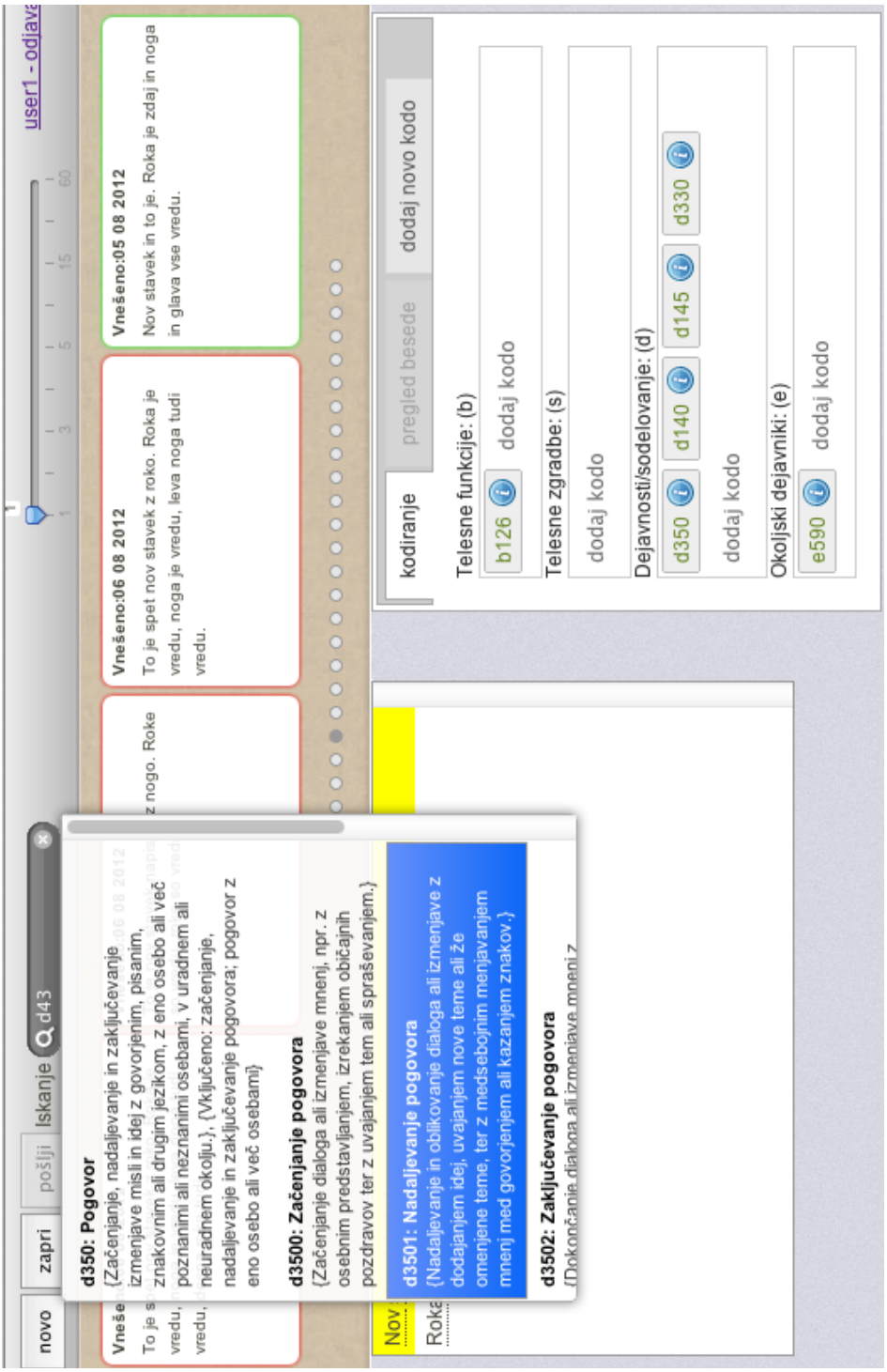
Uporabnik vnese besedilo izvida v vnosno polje na levi strani in pritisne gumb pošlji. Ko se na strežniški strani opravi obdelava, se rezultati obdelave pošljejo nazaj k uporabniku. Sestavijo se potrebne strukture, polje za vnos besedila pa se zamenja z vsebnikom za povedi z osvetljeno prvo povedjo. Uporabnik se po povedih premika s smernimi tipkami ali pa s kolescem na miški. Ko se poved zamenja, se posodobi tudi kontekst ponujenih in/ali izbranih kod, prikazanih v zavihku na desni strani (slika 5.2). Uporabnik lahko z nižanjem praga manjša stopnjo preciznosti predlaganih kod, kar posledično pomeni več prikazanih kod. Kode, ki so pravilne, uporabnik izbere s klikom nanje. Izbranim kodam se ozadje spremeni v zeleno barvo. Besede, ki nosijo pomen, so podčrtane. S klikom na podčrtano besedo se na desni strani zamenja zavihek v zavihek za pregled besede. Tu vidimo, v definicij katerih kod in kolikokrat se pojavlja izbrana beseda (slika 5.6). Če uporabnik meni, da je določena beseda ali besedna zveza tesno povezana z neko kodo, a je sistem ne predlaga, lahko tako povezavo ustvari v zavihku »dodaj novo kodo« (slika 5.5). Sprememba kateregakoli parametra se avtomatsko zabeleži in je za uporabnika nevidna. Do že kodiranih izvidov se pride s klikom na gumb odpri. Prikaže se povzetek izvidov, ki jih je vnesel trenutni uporabnik. Nadzorni zdravnik ima dostop do vseh izvidov (slika 5.1).

5.0.2 Hitro iskanje

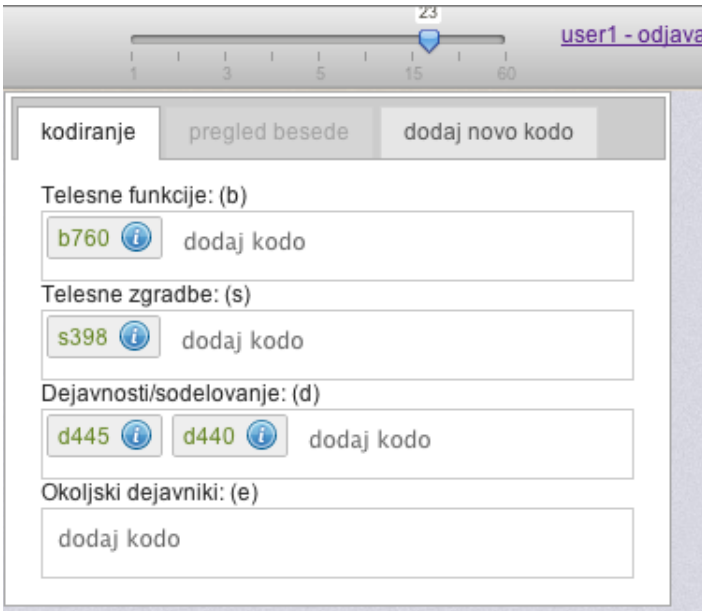
Relevantno kodo, ki je aplikacija ne predlaga, je potrebno vnesti ter označiti besede, ki služijo kot asociacija na to kodo. Uporabniku, ki kode ne zna točno umestiti, aplikacija ponudi nabor možnih kod za izbrani sklop (slika 5.3). Isto okno je mogoče uporabiti tudi kot abecedno kazalo, ki je identično tistemu v priročniku za iskanje po ključnih besedah (slika 5.4).

5.0.3 Potrjevanje novih besed k ustrezni kodi

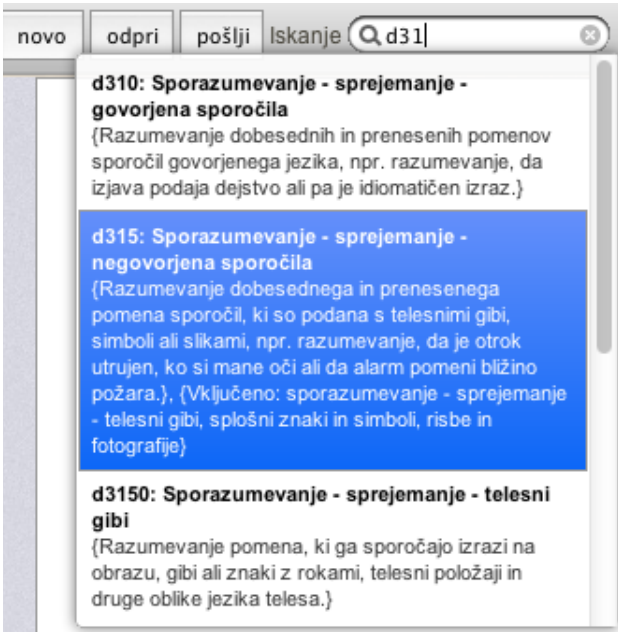
Pogled potrjevanja besed k ustreznim kodam prikaže, kateri uporabnik je predlagal novo ključno besedo, čas predloga, koren besede, kodo in poved, v kateri se nahaja ključna beseda (slika 5.7). Nadzorni zdravnik predlagano besedo zavrne ali potrdi. Ko je beseda potrjena, se koda, na katero je beseda vezana, predlaga ob naslednji pojavitvi te besede v novih izvidih.



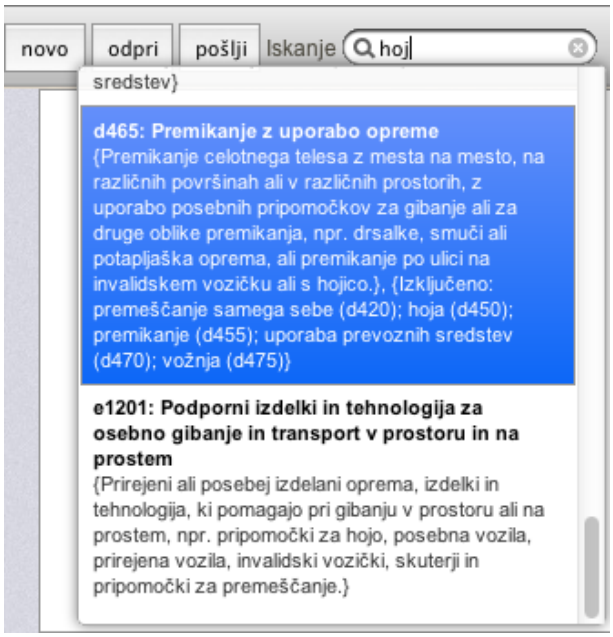
Slika 5.1: Izgled spletne aplikacije v brskalniku.



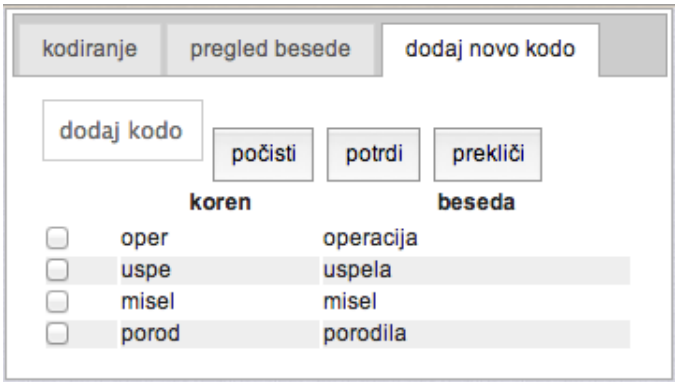
Slika 5.2: Zgoraj prag, spodaj nabor ponujenih kod za nesmiselno poved: *Roki in nogi sta vredu.*



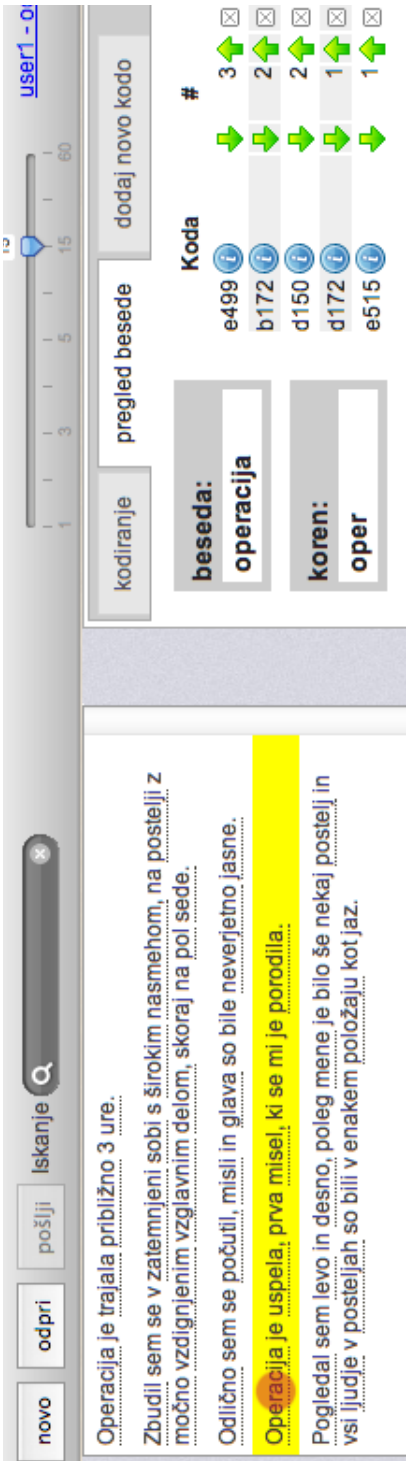
Slika 5.3: Seznam možnih kod za izbrani sklop.



Slika 5.4: Iskanje po ključnih besedah.



Slika 5.5: Dodajanje kode izbranim besedam.



Slika 5.6: Pregled vseh kod v izbrani besedi. Rdeči krog označuje klik miške.

Slika 5.7: Potrjevanje kod in vnos novih ključnih besed.

Poglavje 6

Sklepne ugotovitve

6.1 Diskusija

V diplomskem delu smo predstavili metode in tehnike za obdelavo naravnega jezika. Te metode smo uporabili v končnem izdelku, spletni aplikaciji, ki zdravnikom olajša kodiranje izvidov. Aplikacija se že uporablja v URI - Soča. Aplikacijo je možno razmeroma enostavno prilagoditi za obdelavo izvidov v jeziku, za katerega je na voljo klasifikacija MKF. Trenutno sta podprta slovenski in agleški jezik. Angleška verzija aplikacije je bila predstavljena na XI. Kongresu EFRR leta 2011 v Italiji [9].

6.2 Nadaljnje delo

Prostor za izboljšave je predvsem v razumevanju pomena besed in semantičnih povezav med njimi. To bi dosegli z uporabo tezavrov. Točnejši prikaz kod bi dosegli tudi s preučevanjem že kodiranih izvidov in na podlagi teh sestavili dodatna pravila za predlagane kode. Ogrodje aplikacije in posamezne komponente, ki smo jih predstavili v tem delu, bi bilo možno uporabiti za druge klasifikacije SZO, na primer MKB (angl. ICD) [4].

Literatura

[1] B. Jenko, "Samodejno razvrščanje izvlečkov objav v slovenskem jeziku", magistrsko delo, Fakulteta za elektrotehniko, Univerza v Ljubljani, Ljubljana, 2005.

[2] R. Kukafka, M. E. Bales, A. Burkhardt, C. Friedman, "Human and Automated Coding of Rehabilitation Discharge Summaries According to the International Classification of Functioning, Disability, and Health", Journal of the American Medical Informatics Association, št. 13, zv. 5, str. 508-515, 2006.

[3] I. Kononenko, M. Robnik Šikonja, "Inteligentni sistemi", Založba FE in FRI, Ljubljana, 2010, pogl. 11.

[4] "Mednarodna klasifikacija bolezni in sorodnih zdravstvenih problemov za statistične namene : MKB-10", Inštitut za varovanje zdravja Republike Slovenije, Ljubljana, 1995.

[5] "Mednarodna klasifikacija funkcioniranja, zmanjšane zmožnosti in zdravja : MKF", Inštitut za varovanje zdravja Republike Slovenije, Inštitut Republike Slovenije za rehabilitacijo, Ljubljana, 2006.

[6] T. Mohorič, "Podatkovne baze", Bi-tim, Ljubljana, 2002, str. 15-20.

[7] K. P. Soman, R. Loganathan, "Machine Learning with SVM and Other Kernal Methods", Prentice-Hall of India Pvt. Ltd, New Delhi, 2009.

[8] G. Stucki, T. Ewert, A. Cieza, "Value and application of the ICF in rehabilitation medicine", Disabil Rehabil., št. 25, zv. 11-12, str. 628-634, 2003.

[9] Ž. Zupanec, L. Šajn, "Computer aided ICF classification of medical reports", v zborniku Proceedings of the XI European Congress of the European Federation for Research in Rehabilitation, Riva Del Garda, Italy, may 2011, str. 72-75.