UNIVERZITET U NOVOM SADU
TEHNIČKI FAKULTET
„MIHAJLO PUPIN“
ZRENJANIN

MODELOVANJE I PRETRAŽIVANJE NAD NESTRUKTUIRANIM PODACIMA I DOKUMENTIMA U E-UPRAVI REPUBLIKE SRBIJE

DOKTORSKA DISERTACIJA

Mentor
dr Branko Markoski

Kandidat
mr Vojkan Nikolić

Zrenjanin, 2016. godine

UNIVERZITET U NOVOM SADU
TEHNIČKI FAKULTET
„MIHAJLO PUPIN“
ZRENJANIN

MODELOVANJE I PRETRAŽIVANJE NAD NESTRUKTUIRANIM PODACIMA I DOKUMENTIMA U E-UPRAVI REPUBLIKE SRBIJE

DOKTORSKA DISERTACIJA

Mentor
dr Branko Markoski

Kandidat
mr Vojkan Nikolić

Zrenjanin, 2016. godine

UNIVERZITET U NOVOM SADU ● **TEHNIČKI FAKULTET „MIHAJLO PUPIN”**
23000 ZRENJANIN, Đure Đakovića bb

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj, **RBR**:	
Identifikacioni broj, **IBR**:	
Tip dokumentacije, **TD**:	Monografska dokumentacija
Tip zapisa, **TZ**:	Tekstualni štampani materijal
Vrsta rada, **VR**:	Doktorska disertacija
Autor, **AU**:	Mr Vojkan Nikolić
Mentor, **MN**:	Dr Branko Markoski, vanredni profesor
Naslov rada, NR:	Modelovanje i pretraživanje nad nestruktuiranim podacima i dokumentima u e-Upravi Republike Srbije
Jezik publikacije, **JP**:	srpski (latinica)
Jezik izvoda, **JI**:	srpski/engleski
Zemlja publikovanja, **ZP**:	Srbija
Uže geografsko područje, **UGP**:	Vojvodina
Godina, **GO**:	2016.
Izdavač, **IZ**:	Tehnički fakultet „Mihajlo Pupin“ Zrenjanin
Mesto i adresa, **MA**:	23000 Zrenjanin, Đure Đakovića bb
Fizički opis rada, **FO**: (poglavlja/strana/citata/tabela/slika/priloga)	7 poglavlja /156 strana /130 citata /44 tabela /63 slika
Naučna oblast, **NO**:	Informacione tehnologije
Naučna disciplina, **ND**:	Information retrieval, Data mining
Predmetna odrednica/Ključne reči, **PO**:	e-Uprava; dubinska analiza teksta; *Natural Language Processing*; *Bag of Words*; nestruktuirani dokumenti; *Apach Lucene*;
UDK	
Čuva se, **ČU**:	Biblioteka Tehničkog fakulteta „Mihajlo Pupin“ u Zrenjaninu
Važna napomena, **VN**:	nema

Izvod, **IZ**:		Danas, servisi e-Uprave u različitim oblastima koriste *question answer* sisteme koncepta u pokušaju da se razume tekst i da pomognu građanima u dobijanju odgovora na svoje upite u bilo koje vreme i veoma brzo. Automatsko mapiranje relevantnih dokumenata se ističe kao važna aplikacija za automatsku strategiju klasifikacije: upit-dokumenta. Ova doktorska disertacija ima za cilj doprinos u identifikaciji nestruktuiranih dokumenata i predstavlja važan korak ka razjašnjavanju uloge eksplicitnih koncepata u pronalaženju podataka uopšte. Najčešća reprezentativna šema u tekstualnoj kategorizaciji je BoW pristup, kada je u pozadini veliki skup znanja. Ova disertacija uvodi novi pristup ka stvaranju koncepta zasnovanog na tekstualnoj prezantaciji i primeni kategorizacije teksta, kako bi se stvorile definisane klase u slučaju sažetih tekstualnih dokumenata. Takođe, ovde je prikazan algoritam zasnovan na klasifikaciji, modelovan za upite koji odgovaraju temi. Otežavajuća okolnost u slučaju ovog koncepta, koji prezentuje termine sa visokom frekvencijom pojavljivanja u upitma, zasniva se na sličnostima u prethodno definisanim klasama dokumenata. Rezultati eksperimenta iz oblasti Krivičnog zakonika Republike Srbije, u ovom slučaju i studija, pokazuju da prezentacija teksta zasnovana na konceptu ima zadovoljavajuće rezultate i u slučaju kada ne postoji rečnik za datu oblast.
Datum prihvatanja teme, **DP**:		18.11.2015.
Datum odbrane, **DO**:		
Članovi komisije, **KO**:	Predsednik:	Dr Dragica Radosav, redovni profesor, TF „Mihajlo Pupin“ Zrenjanin
	Član:	Dr Miodrag Ivković, redovni profesor, TF „Mihajlo Pupin“ Zrenjanin
	Član:	Dr Dragana Glušac, vanredni profesor, TF „Mihajlo Pupin“ Zrenjanin
	Član:	Dr Srđan Popov, docent, FTN Novi Sad
	Član:	Dr Zdravko Ivanković, docent, TF „Mihajlo Pupin“ Zrenjanin
	Član, mentor:	Dr Branko Markoski, vanredni profesor, TF „Mihajlo Pupin“ Zrenjanin

UNIVERSITY OF NOVI SAD ● **TECHNICAL FACULTY "MIHAJLO PUPIN"**
23000 ZRENJANIN, Đure Đakovića bb

KEY WORDS DOCUMENTATION

Accession number, **ANO**:	
Identification number, **INO**:	
Document type, **DT**:	Monographic publication
Type of record, **TR**:	Textual printed article
Contents code, **CC**:	Doctoral Thesis
Author, **AU**:	Vojkan Nikolić, M.Sc.
Mentor, **MN**:	Professor Branko Markoski, Ph.D.
Title, **TI**:	Modeling and searching over unstructured data and documents in e-Government of the Republic of Serbia
Language of text, **LT**:	Serbian
Language of abstract, **LA**:	Serbian/English
Country of publication, **CP**:	Serbia
Locality of publication, **LP**:	Vojvodina
Publication year, **PY**:	2016.
Publisher, **PB**:	
Publication place, **PP**:	23000 Zrenjanin, Đure Đakovića bb
Physical description, **PD**: (chapters/pages/ref./tables/pictures/appendixes)	7 chapters /*156 pages* /130 ref. /44 tables /63 pictures
Scientific field, **SF**:	Information technology
Scientific discipline, **SD**:	Information retrieval, Data mining
Subject/Key words, **S/KW**:	e-Government; text mining; Natural Language Processing; Bag of Words; unstructured documents; *Apach Lucene*;
UC	
Holding data, **HD**:	Library of the Faculty
Note, **N**:	none

Abstract, **AB**:		Nowadays, the concept of Question Answering Systems (QAS) has been used by e-government services in various fields as an attempt to understand the text and help citizens in getting answers to their questions promptly and at any time. Automatic mapping of relevant documents stands out as an important application for automatic classification strategy: query-document. This doctoral thesis aims to contribute to identification of unstructured documents and represents an important step towards clarifying the role of explicit concepts within Information Retrieval in general. The most common scheme in text categorization is BoW approach, especially when, as a basis, we have a large set of knowledge. This thesis introduces a new approach to the creation of text presentation based concept and applying text categorization, with the aim to create a defined class in case of compressed text documents.Also, this paper discusses the classification based algorithm modeled for queries that suit the theme. What makes the situation more complicated is the fact that this concept is based on the similarities in previously defined classes of documents and terms with a high frequency of appearance presented in queries. The results of the experiment in the field of the Criminal Code, and this paper as well, show that the text presentation based concept has satisfactory results even in case where there is no vocabulary for certain field.
Accepted by the Scientific Board on, **ASB**:		18.11.2015.
Defended on, **DE**:		
Defended Board, **DB**:	President:	Dragica Radosav, PhD, Full Time Professor, TF Zrenjanin
	Member:	Miodrag Ivković, PhD, Full Time Professor , TF Zrenjanin
	Member:	Dragana Glušac, PhD, Associate Professor, TF Zrenjanin
	Member:	Srđan Popov, PhD, Associate Professor, FTS, Novi Sad
	Member:	Zdravko Ivanković, PhD, Associate Professor, TF Zrenjanin
	Member,Menthor:	Branko Markoski, PhD, Assistant Professor, TF Zrenjanin

Lista slika

Slika 2.1: ESB – Enterprise Service Bus 24
Slika 2.2: Evropski okvir interoperabilnosti [12] 27
Slika 2.3: Konceptualni model [3] 29
Slika 2.4 Interoperabilnost GSB – ESB [16] 33
Slika 2.5: Nivoi interoperabilnosti (prilagođena verzija EIF v2.0) 36
Slika 2.6: Skup tehnologija veb-servisa [33] 38
Slika 2.7: ESB i veb-servisi 39
Slika 2.8: Portal e-Uprava u okviru MDO [34] 40
Slika 2.9: Realizacija koncepta e-Uprave Republike Srbije [35]. 42
Slika 2.10: Arhitektura Ekstraneta MUP-a Republike Srbije 43
Slika 2.11: Provera podataka u bazama MUP-a RS od strane spoljašnih korisnika 45
Slika 2.12: Korišćenje spoljašnih izvora podataka od strane OS MUP Republike Srbije 46
Slika 2.13: Korišćenje aplikacija (servisa) drugih informacionih sistema od strane OS MUP Republike Srbije 47
Slika 2.14: Upis podataka u baze intraneta MUP Republike Srbije od strane spoljašnih korisnika 48
Slika 2.15: Slojeviti model za kategorizaciju standarda [40] 50
Slika 2.16: . Evolucija koncepta e-Uprave 52
Slika 3.1: Generalna šema metode 56
Slika 3.2: Arhitekturu DSS-a zasnovana na TM-u za e-Upravu 60
Slika 3.3: Tok podataka u procesu ekstrakcije odgovora 62
Slika 3.4: Izgled QALL-ME QA sistema [75] 66
Slika 3.5: Polje za dijalog u AQUALOGQA sistemu 67
Slika 3.6: Arhitektura AQUALOG [77] 67
Slika 3.7: QANUS sistem [79] 68
Slika 3.8: Arhitetura FALCON sistema [81] 68
Slika 4.1: Glavne komponente aplikacije za pretraživanje 71
Slika 4.2: Osnovne komponente pretraživača 73
Slika 4.3: Indeksiranje pomoću Lucene 74
Slika 4.4: Glavne komponente aplikacije za pretraživanje teksta 78
Slika 4.5: A Lucene dokument: clas dijagram 81
Slika 4.6: Koncept procesa Lucene analize 82

Slika 4.7: Struktura klase Analizer 83
Slika 4.8: Dekorator obrazac TokenStream 83
Slika 4.9: Tokenizer 85
Slika 4.10: Struktura TokenFilter-a 85
Slika 4.11: Arhitektura factory atributa tokena 86
Slika 4.12: Proces tokenizacije (FMC Notation) 87
Slika 4.13: Osnovne komponente aplikacije za pretraživanje 88
Slika 4.14: IndexWriter i interna zavisnost 88
Slika 4.15: Interna zavisnost IndexReader-a 90
Slika 4.16: Pristup indeksu koristeći IndexReader 90
Slika 4.17: Indeks algoritam Lucene. 92
Slika 4.18: Pretraživanje sa korisničkim upitom. 92
Slika 4.19: Lucene QueryParser 93
Slika 4.20. Lucene IndexSearch. 94
Slika 4.21: GUI Luke 96
Slika 4.22: Pregled sadržaja dokumenta 97
Slika 4.23: Prikaz rangiranih rezultata 97
Slika 4.24: Struktura upita 98
Slika 5.1: Interakcija između građana i sistema predstavljena ADVANSE sistemom [97] 100
Slika 5.2: Logički tok formiranja klastera 102
Slika 5.3: Normalizacija 4-gram 104
Slika 5.4: Aplikacija VebRanka [107] 104
Slika 5.5: Prikaz procesa klasifikacije u n-dimenzionalnom prostoru 114
Slika 5.6: Klasterizacija podataka 114
Slika 5.7: Postupak klasterizacije 115
Slika 5.8: plug-in WIRIS editora u softveru Ckeditor 115
Slika 5.9: Izgled veb- strane sa matematičkim formulama u tekstu [119] 116
Slika 5.10: Portal PRO BONO 119
Slika 5.11: Reprezentacija BoC u domenu kriminala za deo Krivičnog zakonika 120
Slika 5.12: Deo liste stop-reči 121
Slika 5.13: Vektori dokumenata (tri člana Krivičnog zakonika) u prostoru 122
Slika 5.14: QA sistem baziran na BoC modelu 123
Slika 5.15: Algoritam za dela sistema za klasifikaciju upita za članove krivičnih dela Krivičnog zakonika 124

Lista tabela

Tabela 2.1: Tela standarda veb-servisa 39
Tabela 3.1: Paketi otvorenog koda za TM analizu 65
Tabela 5.1: Dobijeni rezultati korišćenjem Uniteks platforme 106
Tabela 5.2: Najbitnije reči iz Krivičnog zakonika RS za članove 121., 122. i 135. 108
Tabela 5.3: Normalizacija pomoću morfološkog rečnika 111
Tabela 5.4: Normalizacija odsecanjem na N-grame dužine četiri 112
Tabela 5.5: Inverzni indeks dokumenata 112
Tabela 6.1: Analiza najfrekventnijih reči u tri dokumenta 129
Tabela 6.2: Rezultati frekventnosti reči 130
Tabela 6.3: Reprezentativni skupovi za tri člana krivičnog zakonika 131
Tabela 6.4: Deset pripremljenih upita 132
Tabela 6.5: Vektori upita KP 1 i člana 121 Krivičnog zakonika 132
Tabela 6.6: Vektori upita KP 1 i člana 122 Krivičnog zakonika 132
Tabela 6.7: Vektori upita KP 1 i člana 135 Krivičnog zakonika 132
Tabela 6.8: Vektori upita KP 2 i člana 121 Krivičnog zakonika 133
Tabela 6.9: Vektori upita KP 2 i člana 122 Krivičnog zakonika 133
Tabela 6.10: Vektori upita KP 2 i člana 135 Krivičnog zakonika 133
Tabela 6.11: Vektori upita KP 3 i člana 121 Krivičnog zakonika 133
Tabela 6.12: Vektori upita KP 3 i člana 122 Krivičnog zakonika 134
Tabela 6.13: Vektori upita KP 3 i člana 135 Krivičnog zakonika 134
Tabela 6.14: Vektori upita KP 4 i člana 121 Krivičnog zakonika 134
Tabela 6.15: Vektori upita KP 4 i člana 122 Krivičnog zakonika 134
Tabela 6.16: Vektori upita KP 4 i člana 135 Krivičnog zakonika 134
Tabela 6.17: Vektori upita KP 5 i člana 121 Krivičnog zakonika 135
Tabela 6.18: Vektori upita KP 5 i člana 122 Krivičnog zakonika 135
Tabela 6.19: Vektori upita KP 5 i člana 135 Krivičnog zakonika 135
Tabela 6.20: Vektori upita KP 6 i člana 121 Krivičnog zakonika 136
Tabela 6.21: Vektori upita KP 6 i člana 122 Krivičnog zakonika 136
Tabela 6.22: Vektori upita KP 6 i člana 135 Krivičnog zakonika 136
Tabela 6.23: Vektori upita KP 7 i člana 121 Krivičnog zakonika 136
Tabela 6.24: Vektori upita KP 7 i člana 122 Krivičnog zakonika 137
Tabela 6.25: Vektori upita KP 7 i člana 135 Krivičnog zakonika 137

Tabela 6.26: Vektori upita KP 8 i člana 121 Krivičnog zakonika 137
Tabela 6.27: Vektori upita KP 8 i člana 122 Krivičnog zakonika 137
Tabela 6.28: Vektori upita KP 8 i člana 135 Krivičnog zakonika 138
Tabela 6.29: Vektori upita KP 9 i člana 121 Krivičnog zakonika 138
Tabela 6.30: Vektori upita KP 9 i člana 122 Krivičnog zakonika 138
Tabela 6.31: Vektori upita KP 9 i člana 135 Krivičnog zakonika 138
Tabela 6.32: Vektori upita KP 10 i člana 121 Krivičnog zakonika 139
Tabela 6.33: Vektori upita KP 10 i člana 122 Krivičnog zakonika 139
Tabela 6.34: Vektori upita KP 10 i člana 135 Krivičnog zakonika 139
Tabela 6.35: Zbirna tabela 140
Tabela 6.36: Merenje ocene rezultata istraživanja: Pregled klasifikacija 141
Tabela 6.37: Rezultati 142

Sadržaj

Lista slika **5**

Lista tabela **7**

Sadržaj **9**

Korišćene oznake **12**

1. Uvod **14**

1.1. Potrebe istraživanja 15
1.2. Problem istraživanja 16
1.3. Predmet i cilj istraživanja 17
1.4. Zadaci istraživanja 18
1.5. Hipoteze istraživanja 18
1.6. Program istraživanja 18
1.7. Metodološki koncept 19
1.7.1. Način izbora, veličina i konstrukcija uzorka 19
1.7.2. Tehnike, postupci i merni instrumenti istraživanja 20
1.7.3. Mesto eksperimentalnog istraživanja 20
1.8. Struktura disertacije 20

2. e-Uprava Republike Srbije **22**

2.1. Interoperabilnost e-Uprave u kontekstu Evropskog okvira interoperabilnosti 22
2.1.1. Interoperabilnost u e-Upravi 24
2.1.2. Evropski okvir interoperabilnosti 26
2.1.3. Konceptualni model EIF 29
2.1.4. Nivoi interoperabilnosti u EIF 29
2.1.5. Realizacija interoperabilnosti pomoću GSB u e-Upravi Republike Srbije 32
2.2. Nacionalni okvir interoperabilnosti Republike Srbije i Servisno orijentisana arhitektura 34
2.2.1. Nacionalni okvir interoperabilnosti Republike Srbije 35
2.2.2. SOA na bazi veb-servisa i interoperabilnost 37
2.2.3. Koncept e-Uprave Republike Srbije 40
2.3. G2G integracija MUP-a Republike Srbije sa portalom e-Uprava 42
2.3.1. Ekstranet MUP-a Republike Srbije 42
2.3.2. Slučajevi korišćenja Ekstraneta MUP-a Republike Srbije 44
2.4. Bezbednost e-Uprave Republike Srbije 49
2.4.1. „Lista standarda interoperabilnosti“ e-Uprave Republike Srbije 49
2.4.2. Bezbednost e-Uprave Republike Srbije 51

3. *Question answering* sistemi **54**

3.1. TM i e-Uprava 57
3.1.1. TM aplikacije u e-Upravi 57
3.1.2. Dubinska analiza podataka 58
3.1.3. Tehnike TM 60
3.2. Analiza maksimalne učestalosti sekvenci reči (*eng. Mining Maximal Frequent Word Sequences*) 63
3.3. Rangiranje rezultata pretrage (*eng. ranking score*) 63
3.4. Analiza podataka sadržanih u višejezičnim tekstovima (*eng. Multilingual Text Mining* – MLTM) 64
3.5. Softverski paketi za TM 64
3.6. Postojeći okviri za QA sisteme 66
4. Apache Lucene 69
4.1 Arhitektura *Apache Lucene* 70
4.1.1 Koncept Apache Lucene 72
4.1.2 Pregled kompozicione strukture Lucene 77
4.2 *Lucene* dokument 80
4.2.1 Package org.apache.lucene.document 80
4.3 Lucene *analiza* 81
4.3.1 Package org.apache.lucene.analysis 82
4.3.2 Analizer 84
4.3.3 Tokenizer 84
4.4 *Lucene* indeksiranje 87
4.5 *Lucene* pretraživanje 92
4.5.1 Package org.apache.lucene.QueryParser 93
4.5.2 Package org.apache.lucene.Search 94
4.5.3 Algoritam *Lucene* za pretraživanje indeksa 95
4.6 LUKE 95
5. Modelovanje sistema za dobijanje brzih odgovora za servise e-Uprave Republike Srbije u oblasti Krivičnog zakonika 99
5.1. Princip korišćenja sistema e-Uprava 99
5.1.1. Teorijske osnove za razvoj sistema brzih odgovora 102
5.1.2. Prikaz stanja u Krivičnom zakoniku Republike Srbije 106
5.1.3. Koncepti poređenja kratkog teksta: poređenje članova Krivičnog zakonika 108
5.1.4. Princip pretraživanja primenom MathML standarda 115
5.2. Predloženi okvir za realizaciju mogućeg QA veb-servisa 118
5.2.1. Stop-reči 121
5.2.2. QA sistem baziran na BoC modelu 121
6. Analiza eksperimentalnih rezultata 126
6.1. Prikaz tekstualnih dokumenata u vektorskom obliku 127
6.2. Provera preciznosti predloženog sistema 131
7. Zaključak 144
Literatura 148

Korišćene oznake

Oznaka	Značenje
ADF	*Application Development Framework*
BoC	*Bag of Concept*
BoW	*Bag of Words*
BPEL	*Business Process Execution Language*
BPM	*Business process management*
BPMN	*Business Process Modeling and Notation*
CLTR	*Cross-lingual text retrieval*
DM	*Data mining*
DTM	*Matica termina dokumenta*
DSS	*Decision support system*
EA	*Enterprise Architecture*
EAI	*Enterprise Application Integration*
EIF	*European Interoperability Framework*
EPAN	*European Public Administration Network*
ESB	*Enterprise Service Bus*
EU	*Evropska Unija*
G2B	*Government to Businesses*
G4B	*Government for Businesses*
G2C	*Government to Citizens*
G4C	*Government for Citizens*
G2E	*Government to Employees*
G2G	*Government to Governments*
GIF	*eGovernment Interoperability Framework*
GSB	*eGovernment Servise Bus*
HTML	*HyperText Markup Language*
HTTP	*HyperText Transfer Protocol*
IE	*Informations Extraction*
IEEE	*Institute of Electrical and Electronics Engineers*
IKT	*Informacionih i telekomunikacionih tehnologija*

IR	*Information Retrieval*
IT	*Information Technology*
JMS	*Java Messaging System*
LDAP	*Lightweight Directory Access Protocol*
MLTM	*Multilingual Text Mining*
MOM	*Message-Oriented Middleware*
MUP	*Ministrastvo Unutrašnjih poslova*
NIF	*Nacionalni okviri interoperabilnosti*
NLP	*Natural language processing*
OFM	*Oracle Fusion Middleware*
ORB	*Object Request Broker*
OS	*Ovlašćeni službenik*
OWL	*Web Ontology Language*
RDF	*Resource Description Format*
RCU	*Repository Creation Utility*
SAS	*Statistical Analysis System*
SEO	*Search Engine Optimization*
SOA	*Servisno orijentisana arhitektura*
SOAP	*Simple Object Access Protocol*
TF-IOF	*Frekvencija termina i inverzna frekvencija dokumenta*
UDDI	*Universal Description, Discovery, and Integration*
UIMA	*Arhitektura za upravljanje nestruktuiranim informacijama*
W3C	*World Wide Web Consortium*
WLS	*WebLogic Server*
WSDL	*Web Service Description Language*
XML	*Extensible Markup Language*

1. Uvod

Reforma i modernizacija javnog sektora na osnovu široke primene informacionih i komunikacionih tehnologija (IKT) smatra se jednim od ključnih elemenata u daljem razvoju informacionog društva u Republici Srbiji. Razvoj e-Uprave Republike Srbije zasniva se na usvojenim direktivama i preporukama Evropske unije, kao i na određenim strategijama i akcionim planovima koje je usvojila Vlada Republike Srbije. U okviru razvoja e-Uprave u Republici Srbiji razvijen je centralni portal, koji predstavlja mesto pristupa za sve građane. Prilikom razvoja portala e-Uprave Republike Srbije primenjen je princip interoperabilnosti i implementirana je servisno orijentisana arhitektura bazirana na veb-servisima. Na portalu su postavljene razne informacije i aplikacije koje su namenski razvijene za portal u obliku veb-servisa i prdstavljaju servis e-Uprave Republike Srbije. Građani ovde informacije mogu dobiti pomoću uobičajenih pretraživača, gde se prikazuju linkovi koji ukazuju na veliki skup dokumenata i ne mogu dobiti odgovor. Kako bi se unapredila usluga e-Uprave Republike Srbije neophodno je objediniti veliku količinu podataka, dokumenata i usluga na jednom mestu i organizovati ih po temama i/ili određenim grupama. Veoma je značajno kreirati sistem za pretraživanje baziran na indeksima i pretraživanje indeksa. Način dosadašnjeg razvoja e-Uprave Republike Srbije i primenjene tehnologije mogu da podrže i napredne servise koji se baziraju na savremenim principima dubinske analize teksta, *question answering* (QA) i „Vreća reči“ (*eng. Bag of Words* – BoW) konceptima.

Potrebu za ovakvim, naprednim servisima e-Uprave povećava činjenica da se u Vladi Republike Srbije svakodnevno stvara velika količina tekstualnih dokumenata i pronalaženje informacija od strane građana postaje veoma teško. Kako bi se olakšalo pronalaženje informacija građanima i predstavnicima privrednih subjekata, a i zaposlenima u javnoj upravi, potrebno je raditi na iznalaženju adekvatnog pristupa za pronalaženje informacija. Jedan od njih je realizacija QA sistema, koji predstavlja vrstu sistema za pretraživanje informacija koji obrađuje upite postavljene na prirodnom jeziku i vraća ili ekstrahuje odgovore iz struktuiranih ili nestrukturiranih izvora. U principu, složenost ovih sistema leži u uspostavljanju podrazumevanih odnosa između upita i odgovora i oni daju konkretne odgovore na upite građana.

Da bi se došlo do zadovoljavajućeg krajnjeg rešenja, koje omogućava bržu i efikasniju pretragu i dobijanje mogućih odgovora na upite građana, korišćeni su:

Dubinska analiza podataka (*eng. Data mining* - DM) - koncipirana kao sredstvo za rešavanje problema analiziranja ogromne količine podataka, koji se svakodnevno stvara i koji se povećava u kontinuitetu;

Dubinska analiza teksta (*eng. Text mining* – TM) – koja ima za cilj da se odbaci nebitan materijal da bi se identifikovalo ono što korisnik traži, u kontekstu pretrage teksta, gde korisnik zna šta traži i taj tekstualni materijal već postoji;

Natural language processing (NLP) - koji se može posmatrati kao skup tehnika i metoda za automatsko generisanje tekstova u prirodnom jeziku;

Question Answer - davanje odgovora od strane QA, koji se bavi pronalaženjem najboljeg odgovora na upit preko servisa e-Uprave i može da koristi više od jedne tehnike TM (tekst kategorizaciju, tekst klasterovanje, koncept/ekstrahovanja entiteta, sumarizaciju dokumenta, i sl).

Pristup BoW - posmatra dokument kao skup reči bez obzira na redosled reči i gramatiku, grupiše podatke prema ključnim rečima i uspostavlja niz značajnih reči i rečenica na osnovu faktora statističke analize kao što su učestalost pojavljivanja termina i distribucija;

N-gram analiza – kojom se mogu prevazići problemi oko leksičkih resursa za srpski jezik i pomoću koje je moguće doći do prihvatljivih rezultata, s obzirom da srpski jezik sadrži mnoga pravila i odstupanja od tih pravila i zahteva znatan skup leksičkih resursa kako bi analiza tekstualnih dokumenata bila prihvatljiva;

Apache Lucene - skalabilna biblioteka za pretraživanje koja predstavlja osnovu na kojoj se mogu razviti aplikacije za pretraživanje i koja može da analizira i indeksira tekstualni sadržaj i da obezbedi pretraživanje unutar kreiranih indeksa i prikaže rezultate pretraživanja za određeni upit;

U ovom istraživanju glavni zadatak je da se izvrši poređenje kratkog teksta, koji je dat kao vektor, sa upitima koje postavlju građani, takođe, vektorski predstavljeni. Ovde su, kao kratki tekstovi, korišćeni određeni članovi Krivičnog zakonika Republike Srbije koji se odnose na krivična dela u vezi sa nanošenjem telesnih povreda.

U ovoj doktorskoj disertaciji su realizovani i predstavljeni: okvir za predloženi veb-servis, model sistema za dobijanje brzih odgovora za servise e-Uprave, QA sistem baziran na „vreći koncepata“ (*eng. Bag of Concept* – BoC) modelu, algoritam dela sistema za klasifikaciju upita za članove Krivičnog zakonika Republike Srbije i rezultati koji su dobijeni korišćenjem predloženog sistema.

1.1. Potrebe istraživanja

Većina portala e-Uprave zasniva se na jednosmernoj komunikaciji u kojoj Vlada određene države proizvodi i isporučuje informacije koje koriste građani. Ove informacije su grupisane ili se mogu dobiti kroz jednostavan pretraživač na osnovu ključne reči. Rezultat takve pretrage može biti veliki broj dokumenata koje građanin mora da pregleda kako bi pronašao željene informacije. Ako je njegovo znanje o temi, na primer pravu i politici, ograničeno, pretraživanje bi moglo trajati satima dok ne pronađe odgovarajuće informacije.

Većina zemalja Evropske unije ima pristup „odvojenih portala“ za svoje informacije, servise i učešće građana u raznim ponudama. Međutim, odnedavno, trend u mnogim zemljama je da uspostave portale koji objedinjuju velike količine informacija i usluga na jednom mestu. Zajednički pristup uključuje organizovanje sadržaja po temama i/ili

određenim grupama. Ovi portali omogućavaju pretrage koje mogu koristiti indeksni sadržaj drugih sajtova Vlada određenih zemalja.

S obzirom da u državnim organima Republike Srbije postoji značajna količina podataka i informacija u elektronskom obliku, koja će se sa vremenom sve više povećavati, neophodano je u e-Upravi Republike Srbije uvesti nove pristupe u pretraživanju podataka i dokumenata kako bi građani, predstavnici privrednih subjekata, a i sami zaposleni u državnim organima brzo i efikasno dolazili do potrebnih podataka i informacija.

1.2. Problem istraživanja

Pristup građana i pravo informisanja na nivou javne uprave su neophodni elementi za uspešno funkcionisanje Vlade određene zemlje. Pristup informacijama javne uprave omogućava građanima da donose brze i adekvatne odluke, dok u isto vreme te odluke predstavljaju kritičku povratnu informaciju Vladi koja ima obavezu da zadovolji potrebe građana i poboljša kvalitet njihovog života. Trebalo bi da Vlada aktivno nastoji da prepozna trendove prateći povratne informacije građana i da isporuči visok kvalitet usluga za građane i privredne subjekte. Upotreba savremenih informacionih tehnologija omogućava lakši pristup informacijama javne uprave i Vladinim servisima, što predstavlja neophodan uslov za efikasnu državnu upravu. Međutim, neophodno je prevazići niz izazova kako bi se u potpunosti iskoristile prednosti dostupne tehnologije.

Informacije u javnoj upravi u vezi sa zakonima, propisima, izmenjenim odredbama, pravnim presedanima i tumačenjima uputstava distribuirane su na mnogim državnim portalima, kako bi građani mogli da pregledaju, pretražuju i preduzimaju određene akcije. Neki od tih portala opremljeni su pretraživačima koji pružaju tekst zasnovana pretraživanja dokumenata. Međutim, Vladini dokumenti su često veoma obimni i sa mnogim unakrsnim referencama između srodnih dokumenata. Ovi dokumenti su najčešće nestruktuirani ili polustruktuirani sa sličnim i često dvosmislenim sadržajem i terminologijom. Kao takve, državne evidencije dokumenata predstavljaju ozbiljnu prepreku u realizovanju jednostavnog pretraživanja teksta razumljivog za građane.

Uprkos značajnoj pažnji na uvođenju informacionih i telekomunikacionih tehnologija (IKT) u automatizovanju poslova Vlade, najrazvijene zemlje i zemlje u razvoju su se do sada fokusirale na jednostavnije faze u razvoju e-Uprave: razvoj sajtova, pilot projekata za nekoliko aplikacija i postavljanja ovih usluga na internet. Razvijene zemlje su sposobnije da investiraju u IKT infrastrukturu i poboljšanje usluga, dok zemlje u razvoju moraju pažljivo da procene granice korisnosti takve investicije. Dok je globalni trend stalno poboljšanje servisa e-Uprave isti za sve zemlje, već postoji jaz u razvoju e-Uprave između razvijenih zemalja i zemalja u razvoju.

Razvoj e-Uprave u Republici Srbiji je značajno unapređen u poslednjih nekoliko godina i u skladu je sa Strategijom i akcionim planom za razvoj elektronske uprave do 2013. godine. Republika Srbija je odnedavno značajno povećala performanse svoje

elektronske uprave. Direkcija za elektronsku upravu predstavlja organ odgovoran za uvođenje *on-line* usluga za poboljšanje kvaliteta usluga koje se pružaju građanima na osnovu principa „sve usluge na jednom mestu“. Ovaj organ je realizovao portal e-Uprava, (http://www.euprava.gov.rs), koji objedinjuje usluge i informacije iz više od 27 državnih organa, uključujući i organe lokalne samouprave.

Postoji velika količina podataka i dokumenata nagomilanih tokom poslednjih decenija u sistemu e-Uprava Republike Srbije. Oni su struktuirani, formatirani i skladišteni na različite načine. To proizvodi veliku kompleksnost za implementaciju zajedničkih usluga koje mogu ponuditi rezultate pronalaženja informacija (eng. Information Retrieval-IR) iz različitih izvora. Postoji nekoliko pristupa razvijenih za bavljenje takvim ciljem DM, online analitička obrada i drugi sistemi zasnovani na poslovnoj inteligenciji su reprezentativni primeri. U svima njima jedna od faza pripreme za obradu podataka je klasterovanje. IR se može značajno poboljšati upotrebom odgovarajućih metoda klasterovanja.

1.3. Predmet i cilj istraživanja

Na osnovu prethodno iznete problematike, predmet istraživanja je:

- Pregled dostupnih tehnologija za razvoj sistema e-Uprava;
- Odabir najpogodnije tehnologije za izradu modela za brzo pronalaženje korisnih informacija;
- Vrednovanje sistema.

Cilj istraživanja je pronalaženje efikasnijeg načina za organizovanje sadržaja nedovoljno razvijene strukture teksta koji se nalaze kao vrednosti polja tabela u bazama podataka ili kao dokumenti u fajl sistemu. To je pozadina formirana za napredna pretraživanja i pronalaženje informacione podrške za savremene servise e-Uprave koji se nude građanima kako bi dobili veću transparentnost institucija.

Jedan od način je da se deo sadržaja dokumenata grupiše odvojeno od izvornih dokumenata. Na primer, metapodaci kojima se opisuje dokument na digestivan i precizan način mogu se klasterovati umesto klasterovanja celog dokumenta, dok su kratke poruke koje se sastoje od jedne ili nekoliko rečenica mogu klasterovati u celinu. Klaster tehnike i algoritmi obezbeđuju veliku fleksibilnost i raznovrsnost implementacije.

Klasterovanje kao proces grupisanja podataka bez nadzora na osnovu prepoznavanja oblika među njima predstavlja jedan od neophodnih delova naprednog pretraživanja i informacionih sistema za pretraživanje. To je primenljivo na nestrukturiranim tekstualnim podacima. Postoje različiti pristupi i primene.

Osnovne komponente istraživanja su:

- Utvrđivanje postojećih klaster tehnika i algoritama;
- Definisanje objekata klasterovanja i utvrđivanje njihovih ograničenja;
- Merenja sličnosti teksta.

1.4. Zadaci istraživanja

Na osnovu predmeta i postavljenog cilja istraživanja, zadatak istraživanje bio je:

1. Sistematizacija znanja iz oblasti TM, posebno primenom NLP koji se odnosi na srpski jezik;
2. Klasifikovanje naučnih saznanja iz oblasti *question answering* sistema;
3. Kreiranje modela sistema za brze odgovore građanima baziranog na tehnici učestalosti pojavljivanja termina;
4. Implementiranje modela sistema za brze odgovore građanima u oblasti Krivičnog zakonika Republike Srbije;
5. Eksperimentalna provera modela sistema za brze odgovore građanima;
6. Ispravljanje eventualno uočenih nedostataka;
7. Ponovna provera modela;
8. Evaluacija modela.

1.5. Hipoteze istraživanja

Glavna hipoteza:

Moguće je kreirati model za pretraživanje nestruktuiranih podataka i dokumenata u e-Upravi Republike Srbije.

Pomoćne hipoteze:

- Moguće je grupisati nestruktuirane dokumente na osnovu traženja podataka u dokumentima (metadata).
- Moguće je realizovati QA sistem baziran na BoC modelu koji daje brze odgovore na upite građana u vezi članova Krivičnog zakonika Republike Srbije.

1.6. Program istraživanja

Programi istraživanja (faze) i orijentacioni sadržaj doktorske disertacije:

FAZE ISTRAŽIVANJA

Faza 1 - Prikupljanje i analiza informacija o tehnologijama za pretraživanje podataka. Odnosi se na prikupljanje literaturnih izvora i građe za navedenu oblast.

Faza 2 – Istraživanje o vrstama ekstrahovanja informacija koje se koriste u e-Upravi Republike Srbije.

Faza 3 – Analiza postojećih sistema za obradu teksta na prirodnom jeziku, pronalaženje i vizuelno predstavljanje konteksta i koncepata.

Faza 4 – Analiza postojećih modela za grupisanje dokumenata na osnovu fazi koncepta i primene različitih merenja sličnosti teksta.

Faza 5 – Modelovanje sistema za brze odgovore Vlade Republike Srbije na pitanja građana.

Faza 6 – Provera i vrednovanje kreiranog sistema kroz prototip slučaja krivičnih dela.

1.7. Metodološki koncept

Primenjene metode su:

- Metoda teorijske analize je korišćena kod proučavanja teorijskih saznanja i najnovijih nalaza u oblastima od interesa.

Pri analizi i interpretaciji podataka, sintetizovanju teorijskog koncepta i izvođenju zaključaka su kombinovane:

- Metoda analize, sinteze, deduktivna metoda pri određivanju teorijskog koncepta.
- Metoda sistemske analize, metoda poređenja da bi se sagledale mogućnosti predloženog teorijskog koncepta u odnosu na postojeće tehnike.
- Eksperiment.

Deskriptivna metoda je korišćena za implementaciju, opis i vrednovanje sistema. Njena primena je neophodna kod eksperimentalog dela.

Eksperimentalni deo bi se odnosi na definisanje:

- Veb-sistema;
- Parametara vrednovanja predloženog sistema;
- Metoda i instrumenata eksperimentalog istraživanja (kreiranje prototipa sistema i testiranje).

1.7.1. Način izbora, veličina i konstrukcija uzorka

Za potrebe istraživanja biće prikupljeni i analiziran određeni zakon Republike Srbije u elektronskom obliku, pitanja građana na portalu PRO BONO i repozitorijum Blica.

1.7.2. Tehnike, postupci i merni instrumenti istraživanja

Pri istraživanju je primenjena tehnika paralelnih grupa relativno ujednačenih po značaju sadržaja. Na ovaj način moguće je uporediti rezultate dve grupe: jedne sa konvencionalnim načinom provere znanja, a druge u eksperimentalnom načinu rada (automatsko prezentovanje sadržaja u vektorskom obliku). Grupe su formirane tako da budu ujednačene po broju, računarskoj pismenosti, starosti i polu. Od postupaka su korišćeni analiza dokumentacije, posmatranje, testiranje i anketiranje.

1.7.3. Mesto eksperimentalnog istraživanja

Kriminalističko-policijska akademija u Beogradu, Srbija.

1.8. Struktura disertacije

U uvodnom poglavlju ukazano je na glavne probleme koji su prisutni prilikom modelovanja sistema za brzo pronalaženja korisnih informacija iz nestruktuiranih tekstualnih dokumenata, opisan je predmet istraživanja i postavljen je cilj disertacije.

U drugom poglavlju je predstavljen koncept e-Uprave sa posebnim osvrtom na e-Upravu Republike Srbije. Detaljno su predstavljene usvojene direktive i preporuke Evropske unije, kao i određene strategije i akcioni planovi koje je usvojila Vlada Republike Srbije, a na kojima se zasniva razvoj e-Uprave u Republici Srbiji. Prikazan je osnivni koncept e-Uprave Republike Srbije, predstavljeni principi na kojima se ona razvija i dat je pregled tehnologija koje podržavaju ovaj razvoj. Pored toga, dati su i konkretni primeri povezivanja određenih državnih organa sa mrežom državnih organa Republike Srbije, gde je detaljno predstavljen specifičan vid povezivanja Ministarstva unutrašnjih poslova Republike Srbije sa portalom e-Uprava preko Ekstraneta. Ovde se mogu videti i specifični načini korišćenja Ekstraneta MUP-a Republike Srbije zajedno sa konkretnim primerima servisa e-Uprave Republike Srbije. Ekstranet ujedno predstavlja i veoma specifičan i visoko sofisticiran način zaštite informacionog sistema MUP-a Republike Srbije.

U trećem poglavlju je dat pregled literature i razmatraju se *question answering* sistemi, koji obrađuju upite postavljene na prirodnom jeziku i vraćaju ili ekstrahuju odgovor iz struktuiranih (baze podataka) ili nestrukturiranih (tekstualnih) izvora i mogućnosti njihove primene u realizaciji servisa e-Uprave Republike Srbije. Ovde se sagledavaju mogućnosti dubinske analize podataka i dubinske analize teksta, kao stredstva za analiziranje ogromne količine podataka i dokumenata, kao i mogućnosti njihove primene kao softverske tehnologije koja se može koristiti u sadašnjim i budućim zahtevnim servisima e-Uprave. Predstavljene su: „Analiza maksimalne učestalosti sekvenci reči“, „Rangiranje rezultata pretrage“ i „Analiza podataka sadržanih u višejezičnim tekstovima“. U ovom poglavlju je dat pregled: softvrerskih paketa za dubinsku analizu teksta, korišćenih aplikacija u e-Upravi brojnih zemalja kao i postojećih okviri QA sistema otvorenog koda, koji su razvijeni i u upotrebi su.

U četvrtom poglavlju dat je detaljan pregled jedne od najkorišćenijih biblioteka za pretraživanje, *Apache Lucene*. Ova biblioteka otvorenog koda predstavlja jednu skalabilnu biblioteku za pretraživanje i čvrstu osnovu na kojoj se razvijaju aplikacije za pretraživanje. Takođe, data je arhitektura i koncept *Apache Lucene*, opisane su sve komponente i dva najvažnija procesa: indeksiranje i pretraživanje. Pored toga predstavljen je GUI LUKE, njegove mogućnosti pregleda sadržaja fajla Lucene indeksa i pokretanje upita preko indeksa.

U petom poglavlju je prikazan predloženi QA sistem baziran na BoC modelu, kao i okvir za predloženi veb-servis za dobijanje brzih odgovora u okviru servisa e-Uprave, algoritam za klasifikaciju upita za deo članova Krivičnog zakonika Republike Srbije u oblasti teških telesnih povreda. Problem pronalaženja određene informacije, u primeru adekvatnog odgovora, predstavljenog u ovoj doktorskoj disertaciji, razmatra se sa dva aspekta: BoW i BoC. Ključna stvar u oba slučaja jeste proces izdvajanja ključnih reči iz dokumenta, pa su u ovom poglavlju predstavljene tehnike normalizacija reči sa posebnim osvrtom na N-gram analizu.

Šesto poglavlje prikazuje rezultate istraživanja i daje analizu dobijenih rezultata. Izbor odgovarajuće mere sličnosti je od ključnog značaja za klaster analizu tekstualnih dokumenata gde svaki vektor prezentuje jedan dokument. Način izrade klastera za skup članova Krivicnog zakona Republike Srbije, kao i odabir odgovarajuce funkcije slicnosti su uključeni u datoj studiji slučaja. Prezentovani su i rezultati pretraživanja, koji je automatski predloženi sistem dao na postavljena nova pitanja građana. Algoritam rangiranja pitanja prema sličnosti sa članovima zakona, prikazuje efikasan način primene ovog algoritma u sistemima za određivanje sličnosti kratkih tekstova na srpskom jeziku. Kratki tekstovi na srpskom jeziku su u širokoj upotrebi na Internetu u vidu tvitova, komentara i utisaka, mogu biti elementi od neprocenjive vrednosti za socijalni inženjering koji predstavlja najveći rizik za sigurnost podataka.

Sedmo, završno poglavlje, sumira rezultate rada, daje osvrt na mogućnost primene predloženog sistema u realnim okolnostima, kao što je Kancelarija za brze odgovore, prikazuje zaključna razmatranja i ukazuje na dalje pravce mogućeg istraživanja. Sistem za davanje brzih odgovora, baziran na NLP tehnologijama, Kao i softverskim agentima koji pomažu korisnicima da izaberu objekte od interesa korišćenjem postojećeg tekstualnog repozirorijuma samo su neki od primera upotrebe predlozenog QA sistema koji su dati u ovoj doktorskoj disertaciji.

2. e-Uprava Republike Srbije

Elektronska uprava (e-Uprava) se zasniva na digitalnim interakcijama između Vlade određene zemlje i građana (*eng. Government to Citizens - G2C*), Vlade i privrednih subjekata (*eng. Government to Businesses - G2B)*, Vlade i zaposlenih u javnoj upravi (*eng. Government to Employees - G2E*). Pored toga, e-Uprava se zasniva i na interakciji između Vlade i Vlada i agencija drugih zemalja (*eng. Government to Governments - G2G*).

2.1. Interoperabilnost e-Uprave u kontekstu Evropskog okvira interoperabilnosti

Evropska Komisija je kreirala definiciju na osnovu koje e-Uprava predstavlja upotrebu kombinacije informacija, tehnologije, organizacionih promena i novih veština u javnoj upravi, tako da se javni servisi mogu poboljšati i ojačati demokratski procesi kako bi mogli da podrže procedure javne uprave [1].

Na osnovu iskaza predstavnika Evropske Komisije „ako servisi e-Uprave treba da podrže jedinstveno tržište i aktivnosti svih učesnika u borbi za slobodom, ne samo da je interoperabinost potrebna unutar i između organizacionih i administrativnih granica već i između nacionalnih granica sa javnim upravama drugih zemalja članica. Ovo će rezultirati i kao razvoj interoperabilnosti u privrednom sektoru“ [1]. Interoperabilnost ima određeni značaj za razvoj e-Uprave ukoliko su težnje razvoja usmerene ka postignuću potencijala u potpunosti. Važnost interoperabilnosti se najjasnije vidi tamo gde nisu primenjeni principi interoperabilnosti i u situacijama gde njen nedostatak dovodi do nemogućnosti razmene podataka, informacija i dokumenata, kao i neadekvatnog kreiranja veze, gubitka vremena i mogućnosti građana [2]. Interoperabilnost omogućava skladnu e-Upravu u Evropskoj Uniji (EU).

Predstavnici Evropske Komisije su iskazali da je „interoperabilnost informacionih i komunikacionih tehnologija sistema, deljenje i ponovna upotreba informacija i spajanje administrativnih poslovnih procesa, pre svega unutar organizacija javnog sektora, a zatim između njih, krucijalno za obezbeđivanje visokog kvaliteta, inovacija i sklada servisa e-Uprave koji su orijentisani ka korisniku.“[2]

Evropski okvir interoperabilnosti (*eng. European Interoperability Framework* - EIF) je definisan nizom legalnih aktova, standarda i preporuka koje opisuju način na koji su se države članice EU u evropskim javnim servisima sporazumele ili način na koji bi trebalo da se sporazumeju kako bi se uspostavila saradnja. Evropski okvir interoperabilnosti [3] ima sledecu svrhu:

- Promocija/popularizacija i podrška u izlaganju evropskih javnih servisa podsticanjem prekogranične i međusektorske interoperabilnosti;
- Davanje smernica u naporima javnog sektora kako bi se obezedili evropski javni servisi privrednim subjektima i građanima;
- Obezbeđivanje kompatibilnost i popvezanost raznih Nacionalnih okvira interoperabilnosti (NIF) u širi okvir kako bi se postigla evropska dimenzija.

Niz preporuka i smernica za servise u e-Upravi definisano je okvirom za Evropsku interoperabilnost, tako da bi javni sektor, privredni subjekti i građani mogli da dođu u međusobni odnos van granica u panevropskom kontekstu. EIF obuhvata sledeće slojeve interoperabilnosti:

- Politički;
- Zakonski;
- Organizacioni;
- Semantički i
- Tehnički.

Tri osnovna interakciona scenarija koja prikazuju načine interakacije u evropskim javnim servisima između država članica su predstavljena u EIF. EIF promoviše konceptualni model javnog sektora koji se sastoji od tri sloja: osnovne javne funckije, sigurne razmena podataka i agregatni javni servisi.

Pretpostavka da svaka zemlja članica ima ili je u procesu izrade svog NIF je osnova EIF. Kao rezultat toga, EIF se koncentriše više na dodatak vrednosti postojećim rešenjima nego na zamenu i na nove nacionalne okvire za interoperabilnost koji obezbeđuju panevropsku dimenziju [3].

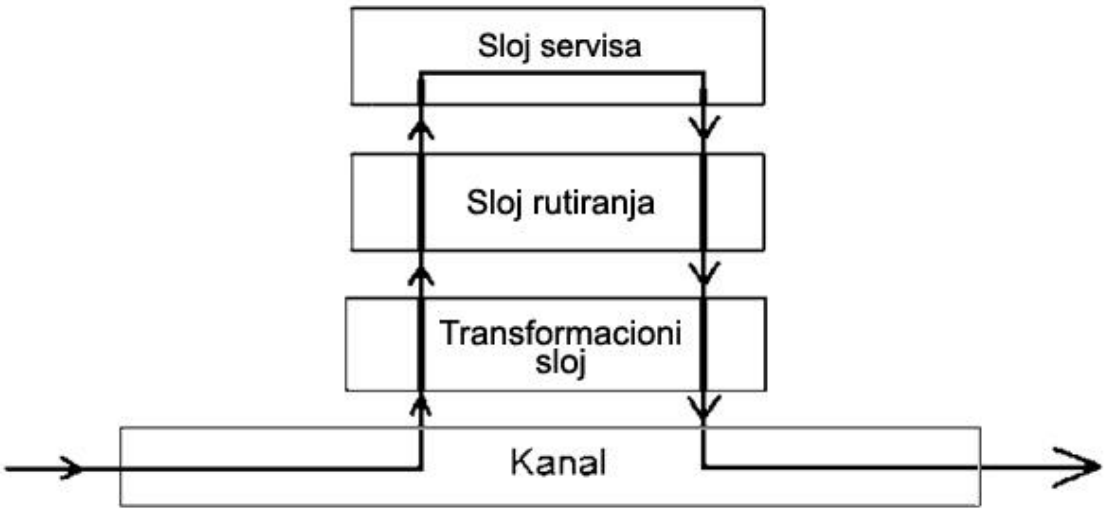
EIF je imao poseban značaj za razvoj e-Uprave u Republici Srbiji zato što je Republici Srbiji trebalo dugo vremena da usvoji NIF (10.01.2014). Razvoj e-Uprave u Republici Srbiji je zasnovan na Strategiji i akcionom planu za razvoj e-Uprave do 2013, dok je EIF bio jedini dokument koji se odnosio na interoperabilnost e-Uprave. Na osnovu ova dva dokumenta, koncept e-Uprava u Republici Srbiji je počeo da se realizije tamo gde je pristupna tačka za građane i poslovne korisnike, a to je portal e-Vlade. Poslovni procesi koji podržavaju servise e-Uprave se izvršavaju na portalu e-Uprava i razmenjuju potrebne podatke i informacije sa drugim državnim organima direktno preko veb-servisa.

Pošto do sada nije postignut značajan stepen interoperabilnost u razvoju e-Uprave Republike Srbije, razmena podataka i informacija direktno preko veb-servisa bi zadovoljila potrebe. Da bi se obezbedila jednostavnija i sigurnija, ali i složenija razmena

podataka između portala e-Uprave Republike Srbije i drugih državnih organa Republike Srbije, neophodna je implementacija *eGovernment Servise Bus* (GSB).

Građani Republike Srbije imaju prilike da preko veb-sajta e-Uprave koriste veb-servisa koji su razvijeni u Ministarstva unutrašnjih poslova (MUP) Republike Srbije. MUP Republike Srbije je, u nekoliko poslednjih godina, jedan od glavnih nosilaca razvoja informacionog društva u Republici Srbiji, a sasvim sigurno, najznačajniji nosilac razvoja e-Uprave Republike Srbije.

Informacioni sistem MUP-a Republike Srbije se zasniva na servisno orijentisanoj arhitekturi (SOA) koja se bazira na veb-servisima. Najkorišćeniji pristup, zasnovan na *Enterprise Service Bus* (ESB) tehnologiji kao na slici 2.1, koji podrazumeva uvođenje interne komunikacione magistrale (BUS) sa implementiranom logikom upravljanja čitavim postupkom integracije, implementiran je u SOA arhitekturi kao nezavisna komponenta koja omogućava veb-servisima da međusobno komuniciraju. Sve pojedinačne aplikacije sistema koje pružaju servise celokupnom sistemu su povezane na zajedničku magistralu i komuniciraju pomoću poruka univerzalnog formata. ESB se tipično implementira u skladu sa principima SOA i veb-servisa kao komunikacionih interfejsa. Sa fizičkog aspekta ESB je softverski proizvod koji interaguje sa pojedinačnim aplikacijama i pruža im jedinsveni interfejs za komunikaciju.



Slika 2.1: ESB – Enterprise Service Bus

2.1.1. Interoperabilnost u e-Upravi

Evropska Komisija je kreirala definiciju na osnovu koje e-Uprava predstavlja upotrebu kombinacije informacija, tehnologije, organizacionih promena i novih veština u javnoj upravi, tako da se javni servisi mogu poboljšati i ojačati demokratski procesi kako bi mogli da podrže procedure javne uprave [1]. Cilj je poboljšanje kvaliteta javnih usluga, podsticanje demokratskih procesa i podrška osnovnih ciljeva zajednice.

Na osnovu definicije inicijative Komisije, e-Uprava je:

- Otvorena i transparentna: javna uprava je u stanju da shvati očekivanja građana, odgovorna je i otvorena za demokratsko učešće.
- Nije isključiva: korisnički centralizovana, javna uprava mora biti dostupna svima sa personalizovanim uslugama.
- Efikasna javna uprava: radi efikasno, štedi vreme i troškove u cilju prikupljanja novca od poreskih obveznika.

Postoji različiti broj definicija za interoperabilnost. Četiri definicije date od strane IEEE [4] [5] su:

1. „Sposobnost dva ili više sistema ili elemenata da razmenjuju informacije i koriste razmenjene informacije“;
2. „Sposobnost jedinica da efikasno rade zajedno kako bi pružile korisne funkcije“;
3. „Sposobnost - promovisana ali ne garantovana - postiže se usaglašenim standardima koji omogućavaju heterogenoj opremi, uglavnom proizvedenoj od strane različitih proizvođača, zajednički rad u mrežnom okruženju“;
4. „Sposobnost dva ili više sistema ili komponenti za razmenu i korišćenje razmenjenih informacija u heterogenoj mreži“.

Prema izveštaju „European Interoperability Framework for pan-European eGovernment services“: ,,Interoperabilnost predstavlja sposobnost IKT sistema i poslovnih procesa da podrže razmenu podataka i da omoguće razmenu informacija i znanja.“ [6]

Izveštaj „eGovernment Working Group of the European Public Administration Network“ (EPAN) predlaže definiciju gde je,,Interoperabilnost sposobnost sistema ili procesa da koriste informacije i/ili funkcionalnost drugog procesa sistema pridržavajući se zajedničkih standarda.“ Izveštaj „The Role of eGovernment for Europe’s Future“, ukazuje da je Interoperabilnost„ … način na koji se realizuje interno povezivanje sistema, informacija i načina rada: unutar ili između uprava, država ili širom Evrope ili sektora preduzeća.“ [7]. Interoperabilnost e-Uprave, u širem smislu, je sposobnost državnih organa da rade zajedno. Na tehničkom nivou, to je sposobnost dva ili više različitih Vladinih IKT sistema ili komponenti da smisleno i neprimetno razmenjuju informacije i da koriste već razmenjene informacije [8].

Interoperabilnost e-Uprave je veoma važna za unapređenje državnih organa i efikasnosti za isporuku osnovnih javnih usluga svim građanima i poslovnim korisnicima. Interoperabilnost e-Uprave omogućava bolje odluke i bolje upravljanje u javnom sektoru. Ova vrsta upravljanja omogućava građanima i poslovnim korisnicima lakši i brži pristup vladinim informacijama i uslugama. Većina vlada EU je prihvatila dizajn strategije nacionalne e-Uprave i sprovode prioritetne programe. Interoperabilnost e-

Uprave je važna kako za Vlade zemalja članica EU, tako i za države koje treba da postanu deo EU.

Interoperabilnost e-Uprave se ostvaruje usvajanjem standarda i arhitekture. Standardi su obezbeđeni od strane vladinog okvira za upravljanje interoperabilnošću (*eng. eGovernment Interoperability Framework* - *GIF*) koji predstavlja skup standarda i politika koje Vlade raznih zemalja koriste da bi odredile način na koji javni sektor, građani i privredni subjekti stupaju u interakciju jedni sa drugima. GIF uključuje tehničke specifikacije koje svi javni sektori koji su uključeni u realizaciju implementacije e-Uprave treba da usvoje. Ovi standardi se odnose na:

- Poslovne procese i organizacionu interoperabilnost;
- Informacije ili semantičku interoperabilnost;
- Tehničku interoperabilnost.

Interoperabilnost arhitekture e-Uprave obezbeđuju Arhitektura preduzeća (*eng. Enterprise Architecture* - EA) i SOA. IEEE definiše arhitekturu kao "osnovu organizacije sistema, kao sastvni deo svojih komponenti i njihovih međusobnih odnosa i odnosa sa okruženjem, po principima koji usmeravaju njen dizajn i aktivnost." [9]

EA je strateški okvir za planiranje koji se odnosi na IKT i usklađuje IKT sa Vladinim funkcijama koje podržava [9]. Danska Vlada je definisala EA kao "zajednički okvir koji osigurava opštu povezanost između sistema informacionih tehnologija (*eng. Information Technology* - IT) u javnom sektoru u isto vreme kada su sistemi optimizovani u odnosu na lokalne potrebe." [10]

SOA je "IT arhitektura unutar preduzeća koja promoviše „labavu“ vezu, ponovnu upotrebu i interoperabilnost između sistema" [11]. Servisna orijentacija definiše potrebe i ishod e-Uprave u pogledu usluga, nezavisno od tehnologije (hardverske platforme, operativnog sistema i programskih jezika) koja ih sprovodi.

Ono što razlikuje SOA-u je njena primena "servisne platforme koji se sastoji od mnogih servisa koji označavaju elemente poslovnih procesa koji se mogu kombinovati i ponovo kombinovati u različita rešenja i scenarije, što je određeno na osnovu poslovnih potreba" [10]. Ova mogućnost da se integrišu i ponovo kombinuju servisi je ono što daje servisno orijentisanom preduzeću potrebnu agilnost da brzo i efikasno reaguje na nove situacije i zahteve.

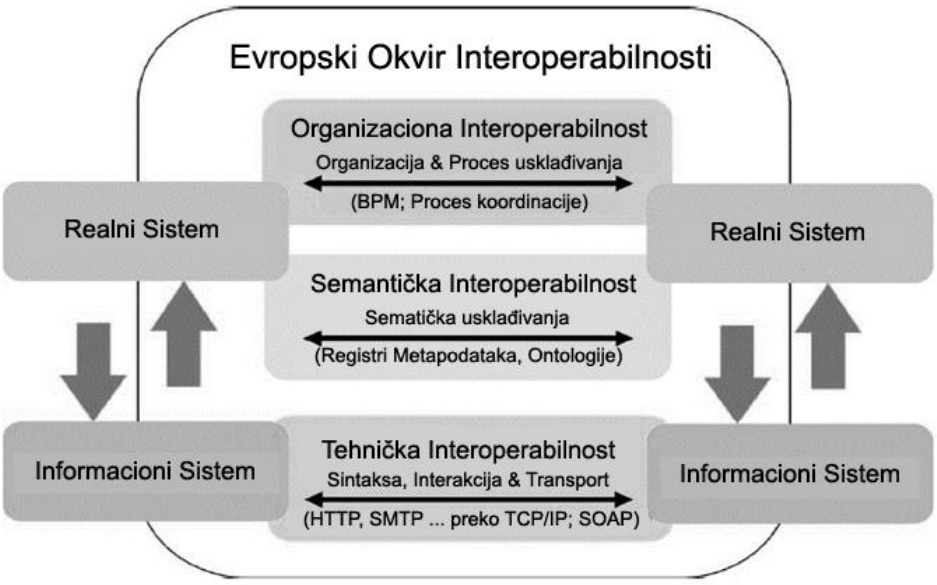
2.1.2. Evropski okvir interoperabilnosti

EIF je definisan kao "okvir za interoperabilnost gde je usvojen interoperabilni pristup za organizacije koje žele da rade zajedno u pravcu zajedničkog pružanja javnih usluga. U okviru svog delokruga primenjivosti, postoji niz zajedničkih elemenata: vokabular, koncepti, principi, politike, smernice, preporuke i prakse " [3].

EU strategija interoperabilnosti za cilj ima da pruži smernice i dodeli prioritet aktivnostima neophodnim za poboljšanje interakcije, razmene i saradnje između evropskih javnih servisa, preko graničnih servisa i između sektora u cilju pružanja javnih usluga u EU. Ova strategija je zasnovana na EIF i odgovorna je za centralizovano pružanje usluga korisnicima e-Uprave, kao i da olakša panevropski nivo interoperabilnosti usluga i sistema između državnih organa i između državnih organa i građanstva (građani, privreda). Zajednički, jedinstven pristup interoperabilnosti sa usvojenom vizijom je da do 2015. godine interoperabilnost omogući visok stepen pružanja javnih usluga u Evropi:

- Odgovarajuće organizacije i procesi vlasti u skladu sa politikom i ciljevima EU;
- Bezbedna razmena informacija omogućena kroz zajednički usvojenu, ujedinjenu i koordinisanu inicijativu za interoperabilnost prilikom kreiranja pravnog okruženja, stvaranje okvira za interoperabilnost i sporazuma o standardima i pravilima interoperabilnost.

U strategiji je naglašeno da aktivnosti treba da budu koordinirane na nivou EU i država članica, a upravljanje interoperabilnosti treba uspostaviti na nivou EU. Interoperabilnost se smatra ključnim segmentom za efikasno i efektivno pružanje javnih usluga u Evropi u svim okvirima politike EU.



Slika 2.2: Evropski okvir interoperabilnosti [12]

EIF se zasniva na pretpostavci da svaka zemlja članica EU ima ili je u postupku stvaranja nacionalnog okvira za upravljanje interoperabilnosti. Imajući to u vidu, EIF se više fokusira na dodavanje procesa nego na zameni nacionalnih okvira interoperabilnosti, dajući im panevropsku dimenzija. EIF se uglavnom bavi panevropskom dimenzijom interoperabilnosti i osim toga, ona ima značaj na nacionalnom nivou. Slika 2.2 predstavlja EIF.

Interoperabilnost je i preduslov za nešto što olakšava efikasnu isporuku servisa evropskih državnih službi. Interoperabilnost se odnosi na:

- saradnju između državnih uprava koje imaju za cilj uspostavljanje javnih servisa,
- razmenu informacija između javnih uprava kako bi se ispunili zakonski uslovi ili političke obaveze,
- razmena i ponovno korišćenje informacija između javnih uprava u cilju povećanja administrativne efikasnosti i smanjenja administrativnih opterećenja za građane i preduzeća,

što vodi ka:

- poboljšanju pružanja javnih usluga građanima i preduzećima obezbeđujući „one-stop shop“ isporuku javnih usluga,
- smanjenje troškova za javne uprave, preduzeća i građane kroz efikasno i delotvorno pružanje javnih usluga. [3]

EIF definiše konceptualni model koji opisuje organizacioni princip koji je osnova za izgradnju i funkcionisanje evropskih javnih servisa i ističe pristup „building block“ u izgradnji evropskih javnih službi, dozvoljavajući međusobno povezivanje i ponovno korišćenja komponenti pri izgradnji novih servisa.

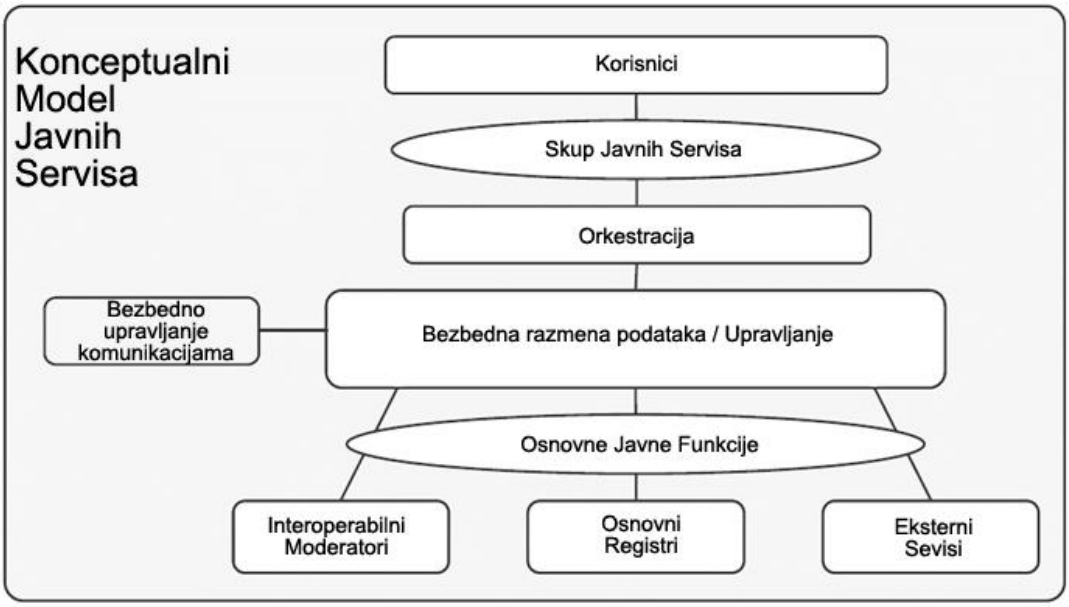
EIF predstavlja dvanaest važnih principa:

- Prvi princip postavlja okvir za aktivnost zajednice u oblasti evropskih javnih servisa:
    1. Supsidijarnost (*eng. subsidiarity*) i proporcionalnost.
- Drugi princip postavlja niz generičkih potreba i očekivanja:
    2. Centralizaciju;
    3. Inkluzije i pristupačnost;
    4. Bezbednost i privatnost;
    5. Višejezičnost;
    6. Administrativno pojednostavljenje;
    7. Transparentnost;
    8. Zaštitu informacija.
- Treći princip postavlja temelj za saradnju između javnih uprava:
    9. Otvorenost;
    10. Ponovno korišćenje;
    11. Tehnološku neutralnost, prilagodljivost;
    12. Efektivnost i efikasnost.

2.1.3. Konceptualni model EIF

EIF v2.0 promoviše konceptualni model, kao model koji promoviše ponovnu upotrebu informacija, koncepata, obrazaca, rešenja i standarda u zemljama članicama i na evropskom nivou za evropske javne službe. Evropski javni servisi su zasnovani na podacima i informacijama na različitim lokacijama i administrativnim nivoima u različitim državama članicama. Pored toga, oni kombinuju osnovne usluge izgrađene nezavisno od javnih organa u različitim državama članicama.

Konceptualni model počinje od činjenice da je neophodno da se obezbedi modularnost, „labavo“ povezane servisne komponente koje su međusobno povezane preko potrebne infrastrukture, gde svi rade zajedno tako da bi mogli da obezbede isporuku evropskih javnih servisa. Prema konceptualnom modelu, servisna orijentacija ka konceptu sistema i razvoju usluga je u prvom planu. Servisna orijentacija je specifičan stil kreiranja i korišćenja poslovnih procesa. Poslovni proces se realizuje kao skup servisa. Slika 2.3 predstavlja konceptualni model EIF.



Slika 2.3: Konceptualni model [3]

Posebne preporuke koje se odnose na konceptualni model su preporuke broj 8 i broj 9:

- "Javni organi treba da razviju servisno orijentisan model baziran na komponentama, što omogućava uspostavljanje evropskih javnih servisa za ponovnu upotrebu, koliko je to moguće, postojećih servisnih komponenti."
- "Javni organi treba da se dogovore o zajedničkoj šemi kako bi međusobno povezali „labavo“ vezane komponente i uspostavili potrebnu infrastrukturu pri uspostavljanju evropskih javnih servisa." [3]

2.1.4. Nivoi interoperabilnosti u EIF

EIF razmatra nekoliko nivoa interoperabilnosti:

- Politička interoperabilnost;
- Zakonska interoperabilnost;
- Organizaciona interoperabilnost;
- Semantička interoperabilnost;
- Technička interoperabilnost.

Politička interoperabilnost

Politička interoperabilnost se odnosi na saradnju partnera sa kompatibilnim vizijama, usklađenim prioritetima i fokusiranim ciljevima. Preporuka 13: „Javni organi treba da dobiju političku podršku za svoje napore koji se tiču interoperabilnosti koja je potrebna za uspostavljanje evropskih javnih servisa.“ [3]

Zakonska interoperabilnost

Zakonska Interoperabilnost se odnosi na usklađivanje zakonodavstva tako da su razmenjeni podaci dobili pravilno svoj pravni značaj. Preporuka 14: „Javni organi treba pažljivo da razmotre sve relevantne zakone vezane za razmenu informacija, uključujući Zakon o zaštiti podataka o ličnosti, u trenutku kada predviđaju uspostavljanje evropskih javnih servisa.“ [3]

Organizaciona interoperabilnost

Organizaciona interoperabilnost se odnosi na koordinaciju procesa u kojima različite organizacione celine postižu prethodno dogovoreni i uzajamno korisni cilj.

Organizaciona interoperabilnost definiše poslovne ciljeve, koordinira poslovne procese, daje mogućnosti za saradnju u organizacijama koje žele da razmenjuju informacije a imaju različite unutrašnje strukture i poslovne procese. Cilj organizacione interoperabilnosti je da zadovolji zahteve građana i poslovnih korisnika tako što servise čini dostupnim, lakim za identifikaciju, pristupačnim i korisnički orijentisanim. To je sposobnost organizacija da pruže servise jedni drugima, kao i korisnicima ili klijentima ili široj javnosti u slučaju javnih organizacija.

Uobičajeni organizacioni problemi koje treba rešiti u umreženom preduzeću (*eng. enterprise networking*) na organizacionom nivou uključuju, ali nisu ograničeni na: različito ljudsko i organizaciono ponašanje, različite organizacione strukture, organizaciju različitih poslovnih procesa i menadžerski pristup, različite načine kreiranja vrednosti mreža, različite poslovne ciljeve, različite zakonske osnove, kulture ili metode rada i različite pristupe odlučivanja [12].

U cilju postizanja organizacione interoperabilnosti, potrebno je koordinirati poslovne procese administrativnih celina koje međusobno sarađuju, definisati sinhronizovane korake i poruke i definisati mehanizme za koordinaciju i kolaboraciju za međuorganizacione procese. Upravljanje poslovnim procesima (*eng. Business process management - BPM*) je realizovano pomoću alata za upravljanje poslovnim procesima i metoda potrebnih za modeliranje i kontrolu ovih poslovnih procesa, mašine za tok rada (*eng. workflow engine*) za koordinaciju izvršenja koraka procesa koji se definišu kao poslovni servisi, alata za kolaboraciju i portala preduzeća kako bi obezbedili „*user-friendly*“ pristup poslovnim servisima i stranice sa informacijama učinili dostupne krajnjim korisnicima. Standardni jezik za modeliranje i analizu poslovnih procesa na poslovnom nivou je Business Process Modeling and Notation (BPMN) verzija 2.0.

Semantička interoperabilnost

Semantička interoperabilnost se odnosi na precizno značenje razmene informacije koje se čuvaju i razmenjuju od strane svih učesnika. Semantička interoperabilnost se definiše kao sposobnost deljenja, spajanja ili sinhronizacije podataka i informacija preko heterogenih informacionih sistema. Semantička interoperabilnost se bavi podacima i integracijom informacija, kao i pitanjima doslednosti kako bi se podržala kooperacija i kolaboracija, a posebno prenos znanja i informacija. Neophodno je obezbediti da dva sistema koja sigurno sarađuju, interpretiraju zajedničke ili podeljene (*eng. shared*) podatke i informacije na konzistentan način.

Semantičke prepreke i problemi koje je potrebno rešiti su: semantički heterogene informacije, semantički jaz, integracija šeme baze podataka, označavanje (npr. homonima i sinonima), logičke strukturne nedoslednosti i slično.

Neophodno je obezbediti sisteme koji tumače značenje podataka, informacija i znanja. Najjednostavnije rešenje je da se izgradi skladište za metapodatke. Skladišta za metapodatke opisuju sadržaj i svrhu podataka koji se čuvaju u različitim informacionim sistemima, a koriste se u preduzeću ili u drugim administrativnim entitetima. Na primer, *Lightweight Directory Access Protocol* (LDAP) za korisnike, IT resurse metapodataka i *Universal Description, Discovery, and Integration* (UDDI) skladišta za veb-usluge registara i tezaurusa (rečnika sinonima) [12]. Ontološki modeli su predstavljeni pomoću *Resource Description Format* (RDF) i *Web Ontology Language for Service* (OWL-S) u skladu sa *World Wide Web Consortium* (W3C) preporukama. "Ontologija se koristi kao ključni jezik za mapiranje koncepata koji se koriste u jednom sistemu sa konceptima drugog sistema i rešava semantičku impedansu neslaganja." [12]

Preporuka 18: „Javna uprava treba da podrži osnivanje specifičnog sektora i međusektorsku zajednicu u cilju olakšavanja semantičke interoperabilnosti, kao i da podstiče razmenu rezultata dobijenih u takvim zajednicama kroz nacionalne i evropske platforme“ [3]

Tehnička interoperabilnost

Tehnička interoperabilnost se odnosi na planiranje tehničkih pitanja koja su uključena u povezivanju računarskih sistema i servisa. Tehnička interoperabilnost (sintaksna interoperabilnost) daje tehničke osnove. Cilj je olakšavanje komunikacije i razmene u pogledu komunikacionih protokola i razmenu podataka i poruka koje prolaze između sistema aplikacija. Ovaj aspekt interoperabilnosti se razvija veoma brzo zahvaljujući razvoju IKT.

Kako bi se obezbedila izgradnja „labavo" vezanih sistema u kojima aplikacije podržavaju administrativne poslovne procese napravljene od veb-servisa, razmenjene poruke (u sinhronizovanom ili asinhronizovanom režimu rada) koriste neutralne formate (XML ili XML-based) i jednostavne protokole za prenos (npr.: HTTP/HTTPS, SMTP, MIME, JMS ili SOAP preko TCP/IP) [12].

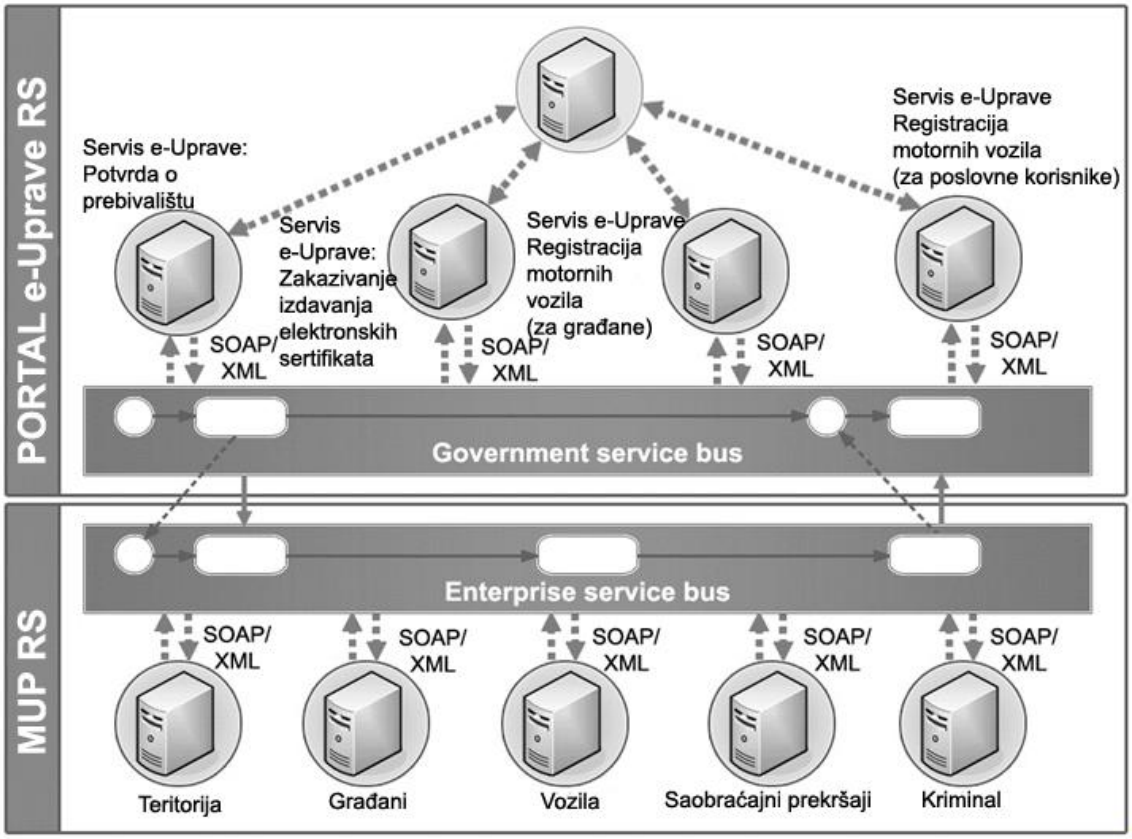
Prikaz stanja tehnika za izgradnju integrisanih ili interoperabilnih sistema je SOA koja se zasniva na servisima. SOA obezbeđuje korišćenje postojećih sistema kao servise gde su ti servisi „upakovani" nasleđeni sistemi i izložene funkcije. S druge strane, SOA omogućava izgradnju novih sistema kao kompoziciju veb-servisa koji se izvršavaju na raznim udaljenim serverima, koji komuniciraju preko interneta. SOA arhitektura obezbeđuje dugoročno korišćenje rešenja: *Object Request Broker* (ORB) i *Enterprise Application Integration* (EAI) su zamenjeni novim rešenjem *Enterprise Service Bus*. Zbog ESB upotreba novih tehnologija koje se zasnivaju na novim jezicima i standardima, odnosno HTTP, SMTP ili *Java Messaging System* (JMS) preko TCP/IP na nivou trasnporta podataka, *Simple Object Access Protocol* (SOAP) ili *RosettaNet* na nivou poruka, *Web Service Description Language* (WSDL) na nivou opisa servisa, UDDI skladišta na nivou objavljenih usluga i na nivou otkrivanja i *Business Process Execution Language* (BPEL) na nivou kompozicije servisa je omogućena [13].

Preporuka 19: „Javne uprave treba da se dogovore oko standarda i specifikacija koje bi se koristile da bi se osigurala tehnička interoperabilnost pri uspostavljanju evropskih javnih servisa." [3]

2.1.5. Realizacija interoperabilnosti pomoću GSB u e-Upravi Republike Srbije

Većina javnih ustanova u Republici Srbiji ima svoj informacioni sistem. Neki IS su u Mreži Državnih Organa (MDO), dok drugi nisu. IS koji nisu u MDO i imaju svoju sopstvenu mrežu moraju da se poveže sa MDO u cilju podrške poslovnih porcesa e-Uprave. Na primer, IS Ministarstva unutrašnjih poslova nije u MDO i kako bi se povezao sa MDO i portalom e-Uprava razvijen je Ekstranet MUP-a Republike Srbije [14][15]. Da bi se obezbedio viši nivo interoperabilnosti i bezbednija razmena podataka, informacija i znanja, neophodno je implementirati GSB u e-Upravi Republike Srbije.

ESB u Extranet zoni već je implementiran u informacionom sistemu MUP-a Republike Srbije kako bi obezbedio jednostavnije i sigurnije povezivanje sa drugim državnim organima. Slika 2.4 predstavlja realizaciju interoperabilnosti u Vladi Republike Srbije, odnosno vezu između portala e-Uprava i informacionog sistem MUP-a Republike Srbije kroz ESB (GSB i ESB). Pored toga, dati su primeri povezivanja servisa portala e-Uprave sa servisima i podacima koji se nalaze u MUP-u Republike Srbije.



Slika 2.4 Interoperabilnost GSB – ESB [16]

Realizacija GSB obezbeđuje:

- Platformu za visok nivo interoperabilnosti informacionih sistema državnih organa Republike Srbije;
- Platformu za standardizovanu integraciju javnih organa Republike Srbije;
- Sigurnu razmenu podataka između državnih organa Republike Srbije;
- Jednostavnu registraciju usluga na portalu e-Uprave;
- Čvrsto spajanje sa modulom za generisanje elektronskih usluga na portalu e-Uprave.

2.2. Nacionalni okvir interoperabilnosti Republike Srbije i Servisno orijentisana arhitektura

Reforma i modernizacija državne uprave zasnovana na širokoj primeni IKT sistema smatra se jednim od ključnih elemenata daljeg razvoja informacionog društva u Republici Srbiji [17].

Kako bi se obezbedile bolje usluge građanima i privrednim subjektima Republike Srbije usvojeno je nekoliko važnih dokumenta (strategija i akcionih planova):

- Strategija reforme državne uprave i Akcioni plan za sprovođenje reforme državne uprave 2009. do 2013. godine [18];
- Strategija i akcioni plan za razvoj elektronske uprave do 2013. godine [19];
- Strategija i akcioni plan za razvoj širokopojasnog pristupa do 2012. godine [20];
- Strategija razvoja elektronskih komunikacija u Republici Srbiji do 2020. godine [21];
- Strategija razvoja informacionog društva u Republici Srbiji do 2020. godine [22];

Strategije daju smernice razvoja, a akcioni planovi precizno definišu zadatke i izvršioce zadataka. Vlada Republike Srbije je 10. 01. 2014. godine usvojila Nacinalni okvir interoperabilnosti (NOI) kojim se utvrđuju smernice za uspostavljanje i primenu interoperabilnosti u organima javne uprave u Republici Srbiji. NOI Republike Srbije treba da obezbedi usklađenost poslovnih procesa unutar i između organa javne uprave.

NOI se uspostavlja u skladu sa evropskom praksom pružanja javnih usluga, poštujući politiku bezbednosti, privatnosti, čuvanja i arhiviranja javnih usluga i elektronskih zapisa. NOI je u skladu sa EIF verzija 2.0 [23].

Pružanje boljih javnih usluga prilagođenih potrebama građana i privrednim subjektima iziskuje nesmetan protok informacija na nivou čitave javne uprave. To se može postići kroz interoperabilnost, koja predstavlja „sposobnost sistema informacionih i komunikacionih tehnologija i podržanih poslovnih procesa, da razmenjuju podatke i omoguće zajedničko korišćenje informacija i znanja” [24].

Interoperabilnost elektronske uprave je vrlo važna za unapređenje javnog sektora i efikasnosti pri isporuci osnovnih javnih servisa svim građanima i poslovnim korisnicima. Interoperabilnost e-Uprave obezbeđuje bolje odluke i bolje upravljanje unutar javnog sektora. Ova vrsta upravljanja omogućava građanima i privrednim subjektima lakši i brži pristup informacijama javnog sektora i servisima. U svom širem smislu, interoperabilnost e-Uprave je sposobnost komponenti javnog sektora da rade zajedno. Na tehničkom nivou to je sposobnost dva ili više različitih informacionih sistema i

komunikacionih tehnologija javnog sektora ili komponenti da razmenjuju informacije kao da ne postoje granice između sistema [25].

Interoperabilnost e-Uprave se realizuje usvajanjem standarda i arhitekture. Standarde obezbeđuje NOI koji predstavlja skup standarda i politika koje Vlada koristi kako bi odredila način na koji javni sektor, građani i partneri međusobno sarađuju. NOI uključuje tehničke specifikacije koje bi javni sektor uključen u implementaciju e-Uprave trebalo da usvoji. Kao što je već rečeno u poglavlju 2.1.4. standardi interoperabilnosti odnose se na:

- Organizacionu interoperabilnost;
- Semantičku interoperabilnost;
- Tehničku interoperabilnost.

SOA bazirana na veb-servisima je arhitektura koja omogućava realizaciju koncepta interoperabilnosti. SOA obezbeđuje visok stepen interoperabilnosti zahvaljujući posebnoj komponenti ESB, a njena implementacija na nivou države, kako bi obezbedila interoperabilnost između državnih organa, predstavlja GSB.

2.2.1. Nacionalni okvir interoperabilnosti Republike Srbije

NOI predstavlja dokument politike organa javne uprave kojim se definišu pravila i način korišćenja interoperabilnosti u Republici Srbiji. On definiše moguće zajedničke infrastrukture i usluge, koje mogu da doprinesu interoperabilnosti i olakšaju interakciju na više nivoa, kao i moguće ponovno korišćenje aplikacija i informacija.

Prilikom izrade NOI uzeta su u obzir sledeća dokumenta:

- Strategija i akcioni plan za razvoj elektronske uprave do 2013. godine [18];
- Strategija razvoja informacionog društva u Republici Srbiji do 2020. godine [23];

NOI, takođe, uzima u obzir preporuke Evropske unije EIF verzija 2.0, tehnološku osposobljenost u različitim organima državne uprave, postojeće elektronske usluge u organima državne uprave, primenu otvorenih standarda i aplikacije koje su u širokoj upotrebi među građanima [17].

Interoperabilnost omogućava javnoj upravi da bolje upravlja svojim internim poslovima. Ona takođe promoviše međunarodnu saradnju poznatu kao međudržavna interoperabilnost, koja može pomoći da se izgradi infrastruktura kakva je potrebna za rešavanje prekograničnih problema. Interoperabilnost ove ciljeve postiže tako što obezbeđuje niz konkretnih povoljnosti:

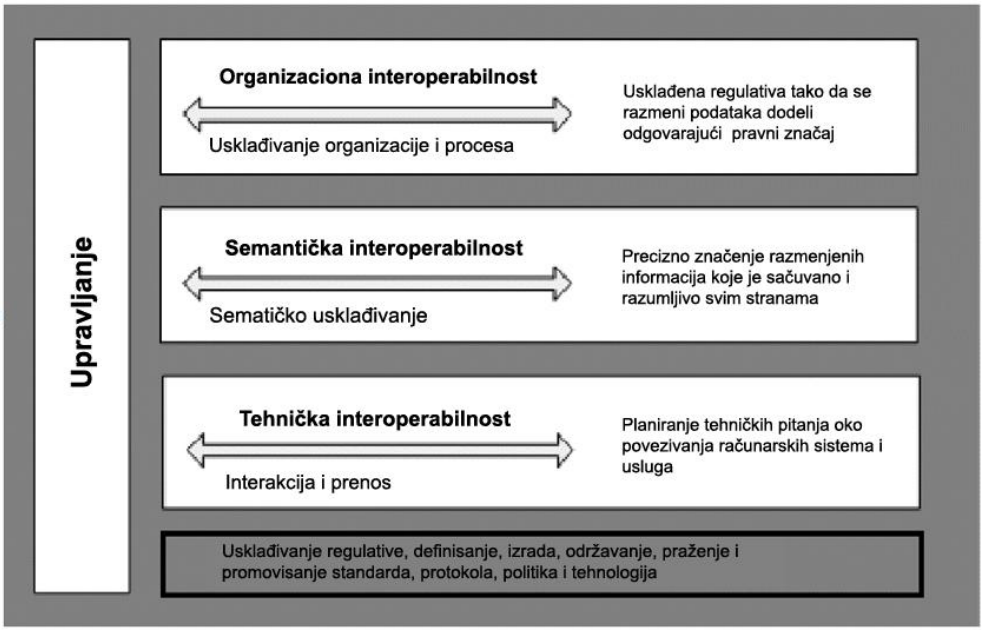
- Povećava fleksibilnost, obezbeđujući različita kombinovanja komponenti;

- Povećava ekonomičnost, obezbeđujući ponovno korišćenje postojećih komponenti i mogućnosti;
- Stvara virtuelno integrisane sisteme koji su jednostavniji za upotrebu;
- Omogućava stvaranje novih mogućnosti, kroz izradu novih funkcija, koristeći postojeće.

NOI se primenjuje u skladu sa ovim opštim načelima koji su od značaja za uspostavljanje interoperabilnosti u svim organima državne uprave:

- Bezbednost i privatnost;
- Transparentnost;
- Čuvanje informacija;
- Otvorenost i mogućnost višestrukog korišćenja;
- Nezavisnost od tehnologije i dobavljača.

NOI definiše konceptualni model javnih usluga gde se predlaže način da se organizuje kreiranje i funkcionisanje elektronskih usluga koje izrađuje, primenjuje i održava Vlada Republike Srbije i različiti organi javne uprave. Predstavljen je generički model, koji se može primeniti u svim organima javne uprave u kojima se pružaju javne usluge, bez obzira na njihovu hijerarhiju, položaj, delokrug poslova i mesto njihovog obavljanja, pri čemu model ukazuje na činjenicu da bilo koji organ javne uprave može biti pružalac kako osnovnih, tako i objedinjenih javnih usluga.



Slika 2.5: Nivoi interoperabilnosti (prilagođena verzija EIF v2.0)

Ovaj model pomaže da se sačini zajednički rečnik i u svim upravnim telima, uspostavi razumevanje glavnih elemenata javnih usluga. Njime se naglašava pristup gradivnih blokova prilikom uspostavljanja javnih usluga, čime se omogućava unutrašnja

povezanost i mogućnost višestrukog korišćenja komponenti usluga, prilikom stvaranja novih usluga [17].

Kao što je već rečeno, u NOI se interoperabilnost razmatra na tri nivoa: tehnološkom, semantičkom i organizacionom (Slika 2.5). Pored toga, razmatra se i interoperabilnost nivoa upravljanja koji podrazumeva pravni i politički kontekst interoperabilnost.

Interoperabilnost upravljanja

Interoperabilnost upravljanja prate ostale tri dimenzije interoperabilnosti, a obuhvata političku, pravnu, upravljačku, ekonomsku i tehničku oblast interoperabilnosti. Ona pruža stalnu podršku interoperabilnosti između pravnih instrumenata, organizacionih poslovnih procesa, razmene informacija, usluga i komponenti koje podržavaju pružanje javnih usluga.

Organizacona interoperabilnost

Organizaciona interoperabilnost se „odnosi na koordinaciju i usklađivanje poslovnih procesa i informatičkih arhitektura koje se prostiru i unutar i između organizacionih granica" [26]. Ona ima za cilj da uspostavi „saradnju uprava koje žele da razmenjuju informacije i mogu imati različite interne strukture i procese" [27]. Konkretno, poslovni procesi ili organizaciona interoperabilnost „bavi se zajedničkim metodama, procesima i zajednički korišćenim uslugama za saradnju, zajedno sa radnim tokom, odlučivanjem i poslovnim transakcijama" [28].

Semantička interoperabilnost

Informaciona ili semantička interoperabilnost se „odnosi na obezbeđivanje toga da precizno značenje informacije koja se prenosi bude razumljivo bilo kom licu ili aplikaciji koja prima podatke" [29]. Informaciona interoperabilnost „omogućava sistemima da kombinuju primljene informacije sa drugim izvorima informacija i da ih obrađuju na smisleni način" [30]. Ona takođe „obezbeđuje zajedničku metodologiju, definisanje i strukturu informacija, zajedno sa zajednički korišćenim uslugama za učitavanje" [31].

Tehnička interoperabilnost

Tehnička interoperabilnost se „odnosi na tehnička pitanja oko povezivanja računarskih sistema za potrebe razmene informacija ili korišćenja funkcionalnosti" [30]. Ona se odnosi na standarde i specifikacije koji treba da omoguće doslednu razmenu informacija između računarskih sistema, a obuhvata definisanje načela, standarda i smernica za mehanizme uobičajenog prenosa, izradu standardizovanih metapodataka i korišćenje zajedničkog jezika.

2.2.2. SOA na bazi veb-servisa i interoperabilnost

World Wide Web Konzorcijum je dao sledeću definiciju veb-servisa [32]: " Veb-servis je softverska aplikacija koju identifikuje URL; čiji se interfejs i veze mogu identifikovati,

opisati i otkriti pomoću XML artifakta i podržavaju direktne interakcije sa drugim softverskim aplikacijama koristeći XML zasnovan na porukama preko Internet protokola.".

Jedan od najznačajnijih aspekt veb-servisa je upravo interoperabilnost. Da bi se ona postigla vodeće softverske i hardverske kompanije u svetu su fokusirane na razvoj tehnologija za razvoj i implementaciju veb-servisa (Slika 2.6).

Opis sloja	Implementacija	Ostali koncerni			
Standard za razmenu poruka	Electronic Business XML Initiative (ebXML)	Kvalitet servisa	Upravljanje	Zaštita	Razvoj servisa
Servis za građenje kompozicije servisa	Business Process Execution Service for Web Services (BPEL4WS)				
Servis za registar	Universal Description, Discovery and Integation (UDDI) ebXML Registries				
Servis za opis	Web Services Description Language (WSDL)				
Servis za razmenu poruka	Simple Object AccessProtocol (SOAP)/Extensible Markup Language (XML)				
Servis za transport	Hypertext Transfer Protocol (HTTP) Simple Mail Transfer Protocol (SMTP) File Transfer Protocol (FTP)				

Slika 2.6: Skup tehnologija veb-servisa [33]

U razvoju zasnovanom na servisima interoperabilnost je regulisana usvajanjem veb-standarda za interoperabilnost (Tabela 2.1). Veb-servisi koriste HTTP protokol za prenos podataka i XML za format podataka. To dozvoljava zahtevima servisa da se lako kreću kroz deo kompjuterskog sistema ili mrežu koja je dizajnirana da blokira neautorizovan pristup (*eng. firewall*). Razvoj zasnovan na servisima je omogućen pomoću Interneta, WWW i otvorenih tehnologija. Realizacijom servisa postignuto je da se između delova aplikacije skrivaju informacije. Ovo je obezbeđeno razdvajanjem interfejsa servisa od njegove implementacije. Veb-servisi omogućavaju sistemima da komuniciraju jedni sa drugima koristeći standardne veb-tehnologije.

Veb-servisi promovišu okruženje za sisteme koji koriste „labave“ parove i interoperabilni su. Ovi koncepati veb-servisa dolaze od konceptualne arhitekture, tj. servisno orijentisana arhitektura. SOA konfiguriše entitete (servise, registre, ugovore i proksije) kako bi obezbedila „labave“ parove i ponovno korišćenje.

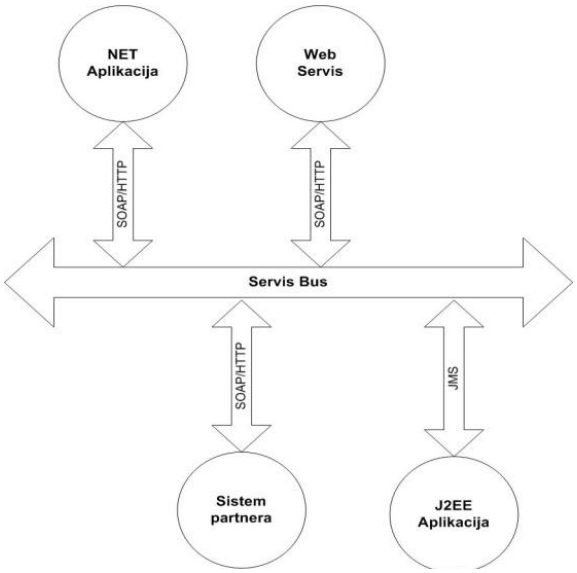
Najvažniji aspekt SOA je da ona odvaja implementaciju servisa od interfejsa servisa. Drugim rečima, ona odvaja „šta“ od „kako“. Korisnici servisa vide servis jednostavno kao krajnju tačku koja podržava format određenog zahteva ili ugovora. Korisnici servisa se ne brinu kako servis radi pri izvršavanju svojih zahteva. SOA obezbeđuje veliki akcenat na interoperabilnosti, tj. mogućnosti sistema da koriste različite platforme i jezike da bi komunicirali jedni sa drugima. Svaki veb-servis obezbeđuje interfejs koji se

može pozvati kroz neku vrstu konektora. Interoperabilni konektor se sastoji od protokola i formata podataka koji svako od potencijalnih klijenata servisa razume.

Organizacija	Standard
World Wide Web Consortium (W3C)	XML, SOAP, HTTP
Web Services Interoperability Organization (WS-I)	BPEL4WS WS-Security WS- Transaction WS-Coordination WS-Attachments WS-Inspection WS-Referral WS-Routing
Organization for the Advancement of Structured Information Standards (OASIS)	UDDI
UN/CEFACT (United Nations Centre for Trade Facilitation)	ebXML
The Internet Engineering Task Force (IETF)	DIME

Tabela 2.1: Tela standarda veb-servisa

Tehnike za podržavanje standardnih protokola i formata podataka se sastoje od mapiranja karakteristika svake platforme i jezika za specifikaciju medijacija. Specifikacija medijacija se mapira između formata interoperabilnih podataka i formata specifičnih podataka platforme. Ovo ponekad zahteva mapiranje nizova karaktera kao što su ASCII na EBCDIC kao i mapiranje drugih vrste podataka. Na primer, veb-servisi su medijacija specifikacije za komunikaciju između sistema.

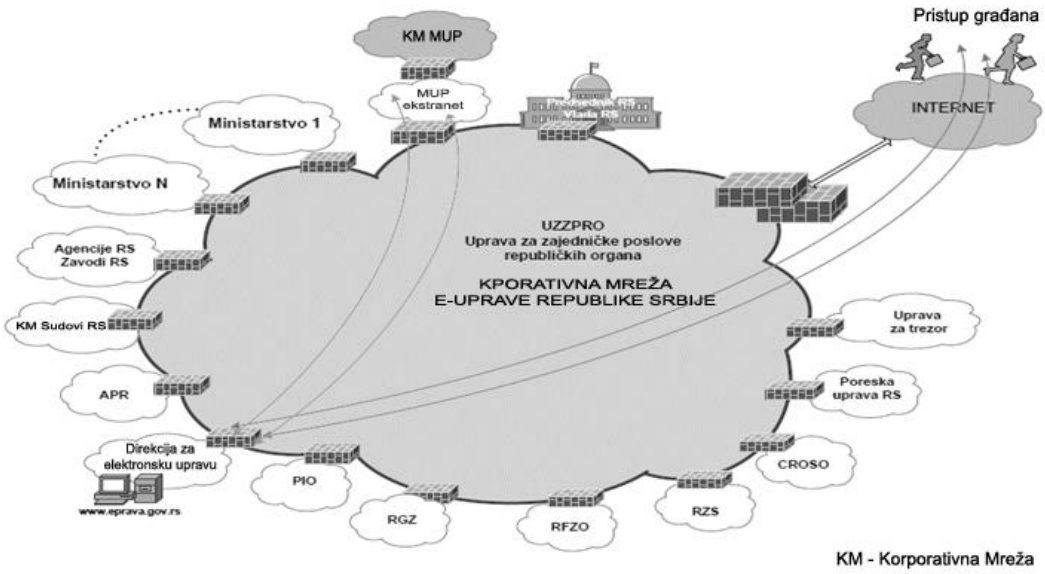


Slika 2.7: ESB i veb-servisi

ESB obezbeđuje veb-servisima da komuniciraju međusobno pomoću nezavisne komponente (Slika 2.7). Ova infrastrukturna komponenta kombinuje najbolju praksu iz *Enterprise Application Integration* (EAI), kao što je *Message-Oriented Middleware* (MOM), veb-servis, rutiranja i XML obradu u cilju obezbeđivanja, korišćenja i komponovanja veb-servisa.

2.2.3. Koncept e-Uprave Republike Srbije

Strategija razvoja e-Uprave u periodu od 2009. do 2013. godine i akcionog plana koji prati ovu strategiju zasniva se na mogućnosti primene IKT u javnom sektoru Republike Srbije. U skladu sa ovom strategijom, razvoj e-Uprave u Republici Srbiji se smatra preovlađujuće decentralizovanim modelom sektora e-Uprave sa jednom pristupnom tačkom servisima e-Uprave. Portal e-Uprava predstavlja tačku gde građani i predstavnici privrednih subjekata pristupaju servisima e-Uprave kako bi realizovali poslovne aktivnosti u zemlji, gde je svaki javni organ Republike Srbije zadužen za pružanje usluga i zadržava ukupnu odgovornost za kvalitet servisa i podataka (Slika 2.8).



Slika 2.8: Portal e-Uprava u okviru MDO [34]

Na samom portalu e-Uprava se izvršavaju elektronski servisi, tj. poslovni procesi koji podržavaju usluge e-Uprave. Poslovni procesi koji imaju potrebu za podacima ili uslugama nekog drugog državnog organa to obezbeđuju preko veb-servisa [22].

Portal e-Uprava je realizovan na Microsoft BizTalk Server 2010 platformi. Ova platforma obezbeđuje [21]:

- Automatizaciju i monitoring poslovnih procesa;
- *Enterprise* integraciju;
- B2B komunikaciju;

- Komunikaciju sa sistemima izvan granica organizacije;
- *Content-based publish/subscribe* arhitekturu;
- Čvrstu vezu sa SQL serverom: *dehydratation* = *serialization*, sprečavanje gubitka podataka;
- Administrativnu konzolu: postavljanje (*eng. deployment*), nadgledanje, operacije;
- Automatizaciju poslovnih procesa korišćenjem orkestracija, *long-running* transakcije;
- Veliki broj adaptera za komunikaciju sa različitim sistemima:
    - EDI, File, HTTP, SFTP,FTP SMTP, POP3, SOAP, SQL, MSMQ i drugi;
    - Microsoft SharePoint Server, IBM mainframe zSeries (CICS i IMS), iSeries (AS/400) server, IBM DB2, IBM WebSphere MQ adapters;
    - WCF adapteri;
    - BizTalk adapter *pack*;
    - Adapteri treće strane i prilagođeni (*eng. custom*) adapteri;
- Akceleratore za industrijska rešenja: HIPAA, HL7, RosettaNet, SWIFT;
- Business rules engine (BRE);
- Business activity monitoring (BAM), dashboard (daje prikaz izvršavanje poslovnih procesa i procesiranja poruka);
- EDI podršku: X12 i EDIFACT;
- XML šemu;
- Podršku za RFID;

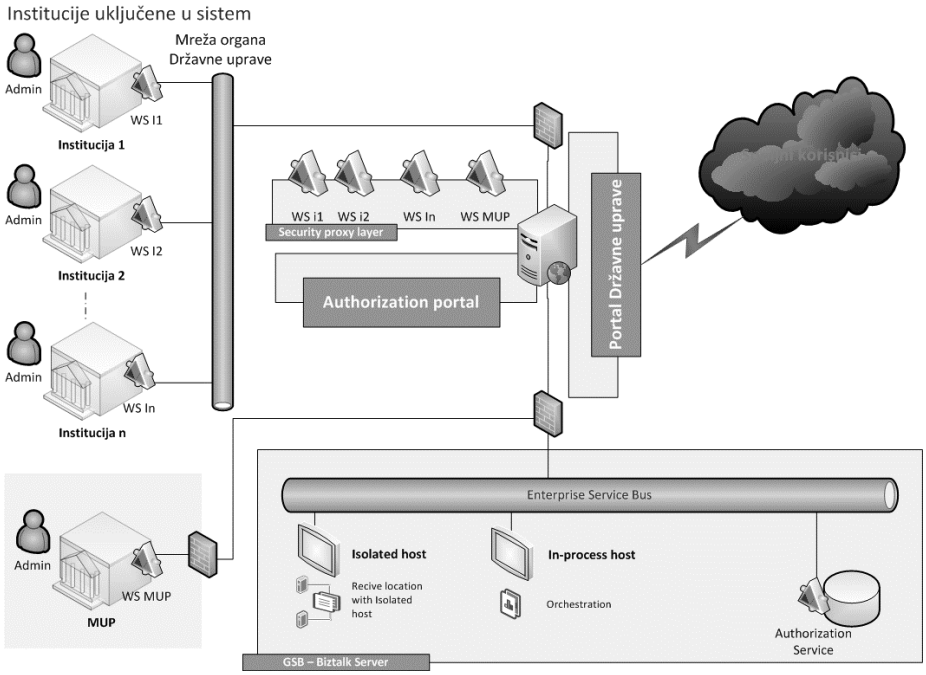
Microsoft BizTalk Server 2010 je platforma koja podržava interoperabilnost e-Uprave i na ovoj platformi je moguće realizovati principe NOI Republike Srbije. Kako bi se ovaj sistem unapredio u smislu razvoja e-Uprave i postizanja većeg stepena interoperabilnosti potrebno je realizovati GSB na bazi Microsoft BizTalk Server 2010 platforme.

Na slici 2.9 je predstavljena realizacija koncepta e-Uprave Republike Srbije [21] i mesto GSB u arhitekturi e-Uprave Republike Srbije.

Realizacija GSB na *Microsoft BizTalk Serveru* 2010 obezbeđuje:

- Platformu za visok stepen interoperabilnosti informacionih sistema javnih organa Republike Srbije;
- Platformu za standardizovanu integraciju javnih organa Republike Srbije;
- Sigurnu razmena podataka između javnih organa Republike Srbije;

- Jednostavnu registraciju servisa na portalu e-Uprava;
- Čvrstu spregu sa modulom za generisanje elektronskih usluga na portalu e-Uprava;



Slika 2.9: Realizacija koncepta e-Uprave Republike Srbije [35].

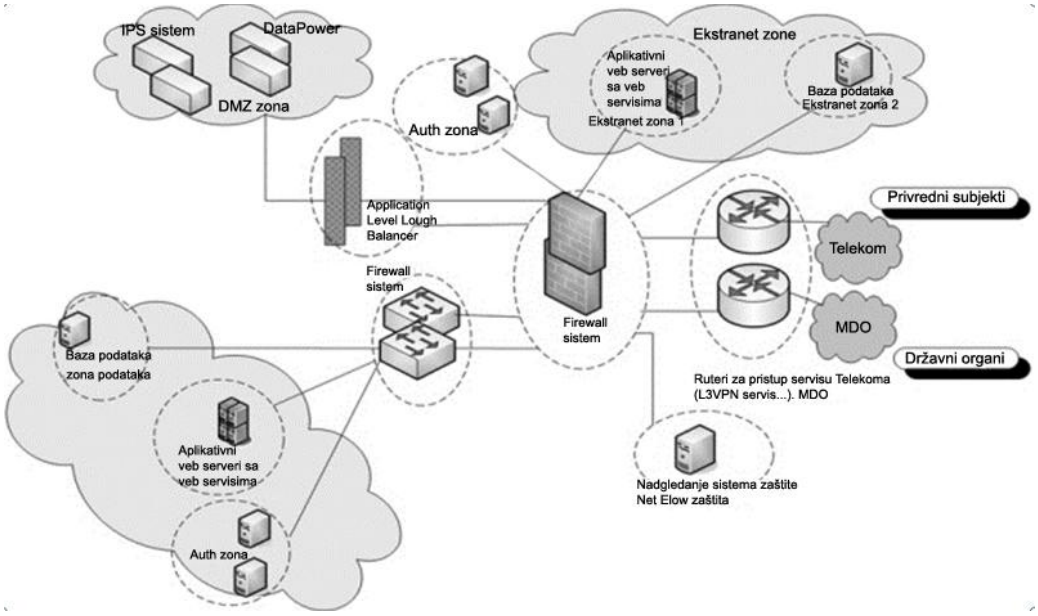
2.3. G2G integracija MUP-a Republike Srbije sa portalom e-Uprava

Kako bi MUP Republike Srbije primenio Akcioni plan za sprovođenje aktivnosti definisane Strategijom razvoja elektronske uprave u Republici Srbiji za period od 2009. do 2013. godine, bilo je neophodno povezivanje sa MDO i nadgradnja IS MUP-a Republike Srbije za poslove e-Uprave. Tako je realizovano rešenje Ekstranet MUP-a Republike Srbije koje je omogućilo povezivanje IS MUP-a Republike Srbije sa MDO i G2G integraciju poslovnih procesa MUP-a Republike Srbije i poslovnih procesa drugih javnih organa. Ekstranet MUP-a Republike Srbije je projektovan tako da se u zavisnosti od poslova Vlade koji se obavljaju biraju načini korišćenja Ekstraneta. Postoje realizovana četiri glavna slučaja korišćenja Ekstraneta [15].

2.3.1. Ekstranet MUP-a Republike Srbije

Arhitektura IS MUP-a Republike Srbije je zasnovana na SOA arhitekturi i veb-servisima. Poslovni procesi koji se automatizuju pomoću SOA metodologije se dele na servise koji se mogu upotrebljavati od strane različitih poslovnih procesa. Računarska mreža intranet MUP-a Republike Srbije je specijalizovana mreža samo za poslove MUP-a Republike Srbije i zatvorena za spoljašnje korisnike. Računarska mreža Ekstranet MUP-a Republike Srbije je zatvorena specijalizovana mreža za povezivanje i razmenu poverljivih i osetljivih podataka sa drugim državnim organima [36]. Arhitektura Ekstraneta MUP-a Republike Srbije je predstavljena na slici 10.

Po ovom konceptu MUP Republike Srbije ostvaruje G2G integraciju sa MDO, a za B2G integraciju sa Telekomom Srbije, gde Telekom dalje organizuje poslovne korisnike. U svakom slučaju radi se o unapred definisanim korisnicima. Prilikom realizacije Ekstraneta MUP-a Republike Srbije, pored standardnog *firewall*-a i *Intrusion prevention system*-a (IPS) implementiran je i uređaj DATA POWER. Kompletna komunikacija se odvija preko ovog uređaja, gde se vrši i autentifikacija i autorizacija zasnovana na poverenju putem sigurne razmene sertifikata. Komunikacija koja se odvija je u potpunosti kriptovana [37]. Ovaj uređaj je specijalizovan za bezbednost veb-servisa.



Slika 2.10: Arhitektura Ekstraneta MUP-a Republike Srbije

Visok stepen funkcionalnosti samog Ekstraneta MUP-a Republike Srbije je obezbeđen implementacijom SOA i određenih komponenti *Oracle Fusion Middleware* (OFM) [38]. Centralni deo Ekstranet *Middleware* MUP-a Republike Srbije predstavlja *Oracle WebLogic Server* (WLS). Pored standardnih komponenti u WLS instalirane su još i:

- *WebLogic* Server, *Enterprise Edition*;
- *Jdeveloper & Application Development Framework* (ADF);
- *Coherence*.

Na WLS su oslonjene još tri komponente:

- Admin server (*host Admin* i *OSB Consoles*);
- OSB server;
- WS server (*Host custom Java EE apps* i *WS*) [14].

OFM se oslanja na DATA BASE sistem i u ovom rešenju je odabran *Oracle Database Enterprise Edition*. Za potrebe OSB u bazi kreirana je DB *Shema* pomoću *Repository Creation Utility* (RCU).

2.3.2. Slučajevi korišćenja Ekstraneta MUP-a Republike Srbije

Primenom metodologija za projektovanje poslovnih procesa koji zahtevaju kolaboraciju i sagledavanjem stanja obavljanja IKT automatizovanih poslovnih procesa u drugim javnim organima, došlo se do zaključka da je potrebno obezbediti četiri glavna slučaja korišćenja Ekstraneta MUP-a Republike Srbije od strane kolaboracionih poslovnih procesa.

Četiri osnovna slučaja korišćenja Ekstraneta MUP-a Republike Srbije su:

- Provera podataka u bazama MUP-a Republike Srbije od strane spoljašnih korisnika;
- Korišćenje spoljašnih izvora podataka od strane ovlašćenog službenika (OS) MUP-a Republike Srbije;
- Korišćenje aplikacija (servisa) drugih informacionih sistema od strane OS MUP-a Republike Srbije;
- Upis podataka u baze intraneta MUP-a Republike Srbije od strane spoljašnih korisnika.

2.3.2.1. Provera podataka u bazama MUP-a Republike Srbije od strane spoljašnih korisnika

MUP Republike Srbije obavljajući poslove iz svoje nadležnosti zadužen je i za formiranje i održavanje velikog broja evidencija. Evidencije koje su nastale u MUP-u Republike Srbije nastale su izvršavanjem IKT automatizovnih poslova koji se takođe odvijaju isključivo u MUP-u Republike Srbije i prilagođene su za druge poslovne procese koji se odvijaju isključivo u MUP-u Republike Srbije.

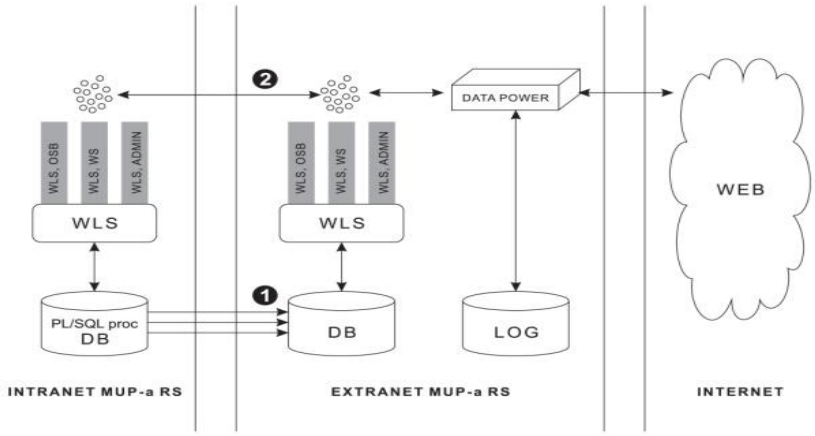
Sa razvojem e-Uprave u Republici Srbiji ukazala se potreba deljenja tih evidencija sa drugim institucijama. Da bi se to obezbedilo u većini slučajeva je bilo neophodno prvo prilagoditi evidencije za potrebe G2G poslovnih procesa i tek onda ih izložiti na Ekstranet MUP-a Republike Srbije. Za specijalne slučajeve omogućeno je i drugim institucijama da određene podatke u okviru obavljanja svojih poslovnih procesa dobijaju direktno iz baza MUP-a Republike Srbije. Slučaj korišćenja Ekstraneta MUP-a Republike Srbije “Provera podataka u bazama MUP-a Republike Srbije od strane spoljašnih korisnika” je realizovan na dva načina:

1. Provera podataka u bazama Ekstraneta MUP-a Republike Srbije od strane spoljašnih korisnika;
2. Provera podataka u bazama intraneta MUP-a RS od strane spoljašnih korisnika;

1. Provera podataka u bazama Ekstranet-a MUP-a RS od strane spoljašnih korisnika

Evidencije koje su nastale u ovom IS MUP-a Republike Srbije su nastale izvršavanjem poslovnih procesa isključivo u IS MUP-a Republike Srbije i prilagođene su tako da podržavaju poslovne procese koji se odvijaju isključivo u MUP-u Republike Srbije. Kako bi ove evidencije bile dostupne za korišćenje i drugim institucijama neophodno je prilagoditi ove evidencije poslovnim procesima drugih institucija. Prilagođavanje evidencija MUP-a Republike Srbije za G2G poslovne procese je realizovano na osnovu principa interoperabilnosti, tako da ove nove prilagođene evidencije može koristiti što veći broj institucija. U nekim slučajevima nije moguće prilagoditi postojeće evidencije MUP-a Republike Srbije za G2G poslovne procese i tada se pristupa formiranju novih evidencija.

Prilagođene i novoformirane evidencije se zatim prebacuju u baze Ekstraneta MUP-a Republike Srbije kako bi bile dostupne drugim institucijama za korišćenje. Pošto se radi o pristupu većeg broja institucija, a iz bezbedonosnih razloga, baze Ekstraneta MUP-a Republike Srbije se pune podacima iz baza intraneta MUP-a Republike Srbije. Obrnut smer toka podataka nije omogućen. Podaci u bazama Ekstraneta MUP-a Republike Srbije se osvežavaju dinamikom koja obezbeđuje da podaci u njima budu uvek aktuelni.



Slika 2.11: Provera podataka u bazama MUP-a RS od strane spoljašnih korisnika

Na slici 2.11 je prikazan slučaj korišćenja Ekstraneta: „Provera podataka u bazama MUP-a Republike Srbije od strane spoljašnih korisnika". Na ovoj slici je predstavljen način korišćenja Ekstranet baza podataka, gde broj 1 označava tok podataka prilikom punjenja ovih baza i način korišćenja intranet baza podataka, gde broj 2 označava tok podataka i korišćenje intranet baza podataka. Spoljašni korisnici dobijaju podatke iz Ekstranet baza podataka isključivo pomoću veb-servisa. Na ovaj način je obezbeđena provera ispravnosti ličnih karata sa čipom građana Republike Srbije.

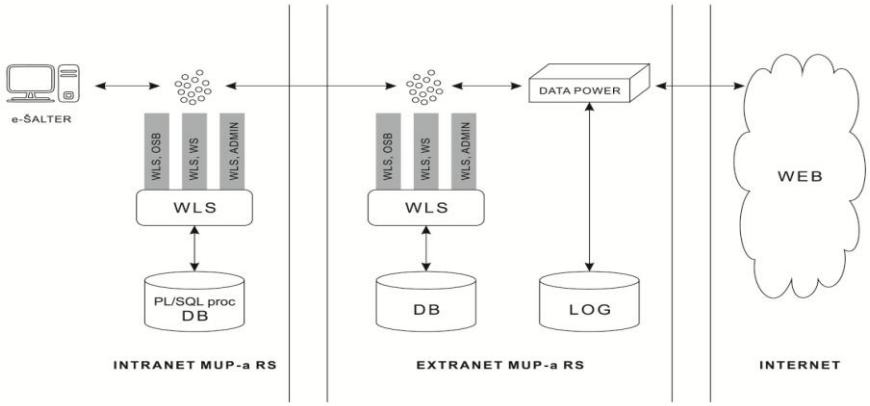
2. Provera podataka u bazama intraneta MUP-a Republike Srbije od strane spoljašnih korisnika

U specijalnim slučajevima obezbeđena je i mogućnost da druge institucije pristupaju direktno bazama intraneta MUP-a Republike Srbije. To se realizuje samo za konkretnog spoljašnjeg korisnika. Glavni razlog za realizaciju ovakvog rešenja jeste složenost poslovnih procesa koji se odvijaju u drugim insticijama i u MUP-u Republike Srbije. Ovde je promena podataka u samim bazama veoma učestala.

Realizacija ovog rešenja je moguća zahvaljujući zaštiti koja je implementirana u Ekstranetu i intranetu MUP-a Republike Srbije, kao i veb-servisima. Ovo rešenje je primenjeno u realizaciji servisa e-Uprave: „Produženje registracije vozila kod ovlašćenog pravnog lica za vršenje tehničkog pregleda vozila“ na portalu e-Uprava [14].

2.3.2.2. Korišćenje spoljašnih izvora podataka od strane OS MUP Republike Srbije;

Decentralizovani razvoj e-Uprave znači i to da je svaka institucija zadužena za prikupljanje i čuvanje podataka u skladu sa zakonskim ovlašćenjima. Ove evidencije druge institucije mogu da koriste samo u skladu sa zakonskim ovlašćenjima. Evidencije se koriste pomoću veb-servisa koje razvijaju institucije u čijem vlasništvu se nalaze evidencije.



Slika 2.12: Korišćenje spoljašnih izvora podataka od strane OS MUP Republike Srbije

SOA arhitektura IS MUP-a Republike Srbije obezbeđuje mogućnost da određena aplikacija koja se odvija u intranetu MUP-a Republike Srbije može u određenom trenutku putem veb-servisa i Ekstraneta MUP-a Republike Srbije da zatraži podatke od veb-servisa drugih institucija. Na slici 2.12 je prikazan slučaj korišćenja Ekstraneta: „Korišćenje spoljašnih izvora podataka od strane ovlašćenog službenika MUP-a Republike Srbije ”. Na ovaj način je realizovano e-Zakazivanje, tj. servis e-Uprave za „Zakazivanje termina za podnošenje zahteva za ličnu kartu i pasoš“ na portalu e-Uprava. Kompletan poslovni proces se odvija na portalu e-Uprava i podaci se smeštaju u bazu podataka portala e-Uprava. Ovlašćeni službenik u MUP-u Republike Srbije kroz

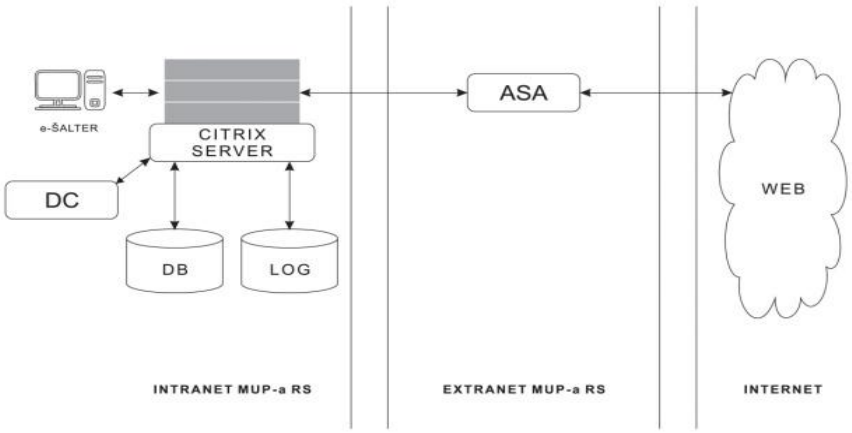
svoju aplikaciju koristi podatke sa portala e-Uprava i realizuje svoj poslovni proces izdavanje dokumenta [36].

2.3.2.3. Korišćenje aplikacija (servisa) drugih informacionih sistema od strane OS MUP Republike Srbije;

Viši nivo integracije koju su omogućili SOA arhitektura i Ekstranet MUP-a Republike Srbije je mogućnost korišćenja aplikacija, tj. kompleksnih i kompozitnih veb-servisa, drugih institucija od strane aplikacija MUP-a Republike Srbije.

Ovo je omogućeno pomoću CITRIX servera koji se nalazi u intranetu MUP-a Republike Srbije na kome se izvršava klijentski deo spoljašne internet aplikacije portala e-Uprave. Na slici 2.13 je prikazan slučaj korišćenja Ekstraneta: „Korišćenje aplikacija (servisa) drugih informacionih sistema od strane OS MUP Republike Srbije ". Na ovoj slici ASA predstavlja *firewall*, a DC domenski konztoler. Proces IKT automatizacije poslovnih procesa i stvaranje servisa e-Uprave olakšava poslove koje obavljaju građani i privredni subjekti sa državom, ali i utiče na promenu navika kod ljudi. Bez obzira što su servisi e-Uprava brži, jednostavniji i jeftiniji, uvek postoje ljudi koji „drže do tradicije" i žele na stari način da obave poslove sa državom. Kada je poslovni process „Produženje registracije motornih i priključnih vozila", koji se u potpunosti odvijao u MUP-u Republike Srbije, prerastao u servis e-Uprave „Produženje registracije vozila kod ovlašćenog pravnog lica za vršenje tehničkog pregleda vozila" i počeo da se odvija na portalu e-Uprave, obezbeđena je mogućnost da građani i ovlašćena službena lica i dalje mogu da produže registraciju vozila na šalterima MUP-a Republike Srbije.

Međutim, na portalu e-Uprava je realizovan servis „objedinjena uplatnica" koji je omogućio da onaj ko produžuje registraciju vozila dobije samo jednu uplatnicu. Za isti ovaj posao na šalterima MUP-a Republike Srbije bilo je potrebno doneti sedam uplatnica.



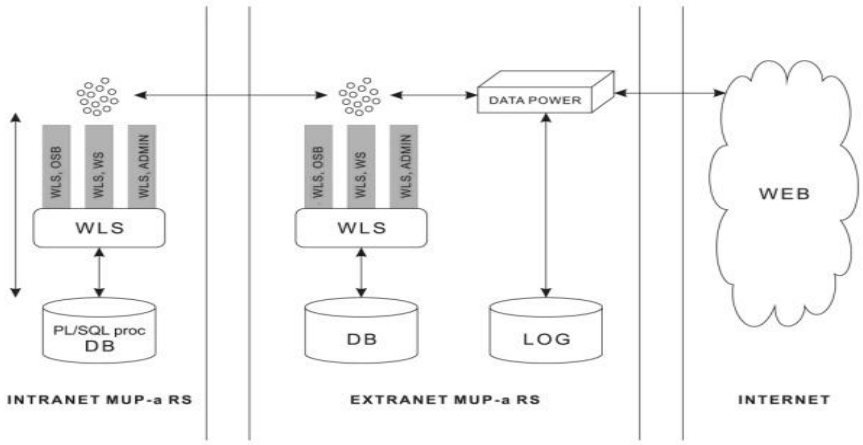
Slika 2.13: Korišćenje aplikacija (servisa) drugih informacionih sistema od strane OS MUP Republike Srbije

Obezbeđeno je da ovlašćeni službenik MUP-a Republike Srbije u momentu izdavanja objedinjene uplatnice pozove veb-servis „objedinjena uplatnica“ koji se nalazi na portalu e-Uprava i direktno na ovom servisu izvrši interaktivni poslovni proces izdavanje objedinjene uplatnice.

2.3.2.4. Upis podataka u baze intraneta MUP-a Republike Srbije od strane spoljašnih korisnika;

Decentralizovani razvoj e-Uprave podrazumeva da je svaka institucija zadužena za evidencije koje su u njenoj nadležnosti. To važi i u slučajevima kada određena institucija zadužena za određene evidencije ne prikuplja sve podatke za tu evidencije već to radi neka druga institucija. U tim slučajevima potrebno je obezbediti mehanizme kako bi sve evidencije imale potpuni skup podataka. Ovo se radi samo u specijanim slučajevima, a za poslove ovakvog tipa u MUP-u Republike Srbije realizovan je slučaj korišćenja Ekstraneta: „Upis podataka u baze Intraneta MUP Republike Srbije od strane spoljašnih korisnika” prikazan na slici 2.14

Prilikom produženja registracije vozila pomoću servisa e-Uprave „Produženje registracije vozila kod ovlašćenog pravnog lica za vršenje tehničkog pregleda vozila“ na portalu e-Uprave, diplomirani pravnici na tehničkim pregledima formiraju PDF dokumente „Registracioni list vozila“, digitalno ih potpisuju i izvršavajući određenu aktivnost na portalu e-Uprava vrše upis ovih dokumenata u bazu MUP-a Republike Srbije [14]. Za realizaciju ovog servisa e-Uprave i održavanje ažurnom ove evidencije podataka primenjen je slučaj korišćenja Ekstraneta: „Upis podataka u baze intraneta MUP Republike Srbije od strane spoljašnih korisnika”.



Slika 2.14: Upis podataka u baze intraneta MUP Republike Srbije od strane spoljašnih korisnika

2.4. Bezbednost e-Uprave Republike Srbije

Kao jedinstveno polazište u cilju uspostavljanja i primene interoperabilnosti predlaže se korišćenje uspostavljene „Liste standarda interoperabilnosti“, koja sadrži spisak standarda, uglavnom otvorenih, sa kompletnom referencom za svaki preporučeni standard. U okviru interoperabilnosti napravljena je klasifikacija za kategorizaciju standarda gde su standardi kategorizovani uz pomoć „modela slojeva“. Bezbednost, kao strukturna komponenta, primenjuje se na svim slojevima „Slojevitog modela za kategorizaciju standarda“ [40].

2.4.1. „Lista standarda interoperabilnosti“ e-Uprave Republike Srbije

Za uspostavljanje interoperabilnosti u e-Upravi Republike Srbije veoma je važno usvajanje „Liste standarda interoperabilnosti“. Verziju 1.0 ovog dokumenta je pripremila međuresorna komisija formirana od strane Direkcije za elektronsku upravu [39].

„Dokument je namenjen svim resursima državne uprave u procesu postizanja tehničke interoperabilnosti. Preporučuje se da organi državne uprave, kada god je to moguće, koriste predložene otvorene standarde, kao i druge standarde koji su u širokoj primeni, s ciljem nezavisnog izbora alternativnih tehnologija kod organa državne uprave, kao i prilagodljivost tehnološkim novinama. Imajući u vidu da se tehnički standardi neprestano razvijaju, potrebno je uspostavljenu Listu standarda neprestano održavati i ažurirati, prema potrebama državnog sektora i tehnologije.“ [40]

Dokument „Lista standarda interoperabilnosti“ daje listu standarda, koja može da se koristi u svim segmentima e-Uprave Republike Srbije, pa čak i u širem krugu javnog sektora, i namenjen je:

- Stručnim licima koja se bave IKT infrastrukturom na nacionalnom i lokalnom nivou;
- Operativnoj podršci IKT sektorima u organima državne uprave RS;
- Svim dobavljačima/pružaocima IKT usluga organima državne uprave RS.

Predloženi standardi temelje se na:

- Najboljim praksama i iskustvima u RS [41];
- Evropskom okviru interoperabilnosti za panevropske usluge elektronske uprave, v 1.0 [24];
- EOI, v 2.0 (Evropska komisija 2011) [18];
- SAGA v.5.1.0 (Standards and Architectures for eGovernment Applications, Savezna Republika Nemačka, 2011.) [42].

„Razmena informacija IKT sistema unutar i sa organima državne uprave je veliki izazov za Vladu Republike Srbije, jer mnogi novi i stariji sistemi imaju sopstvene interfejse koji pružaju ograničene mogućnosti za interoperabilnost. Vlada Republike Srbije je prepoznala značaj standarda za obezbeđivanje interoperabilnosti, jer usvajanje standarda, koji su zasnovani na integraciji rešenja, predstavljaju način da se smanje dugoročni troškovi integracije i olakša fleksibilnost informaciono komunikacione infrastrukture državne uprave.“ [24]

U kontekstu dokumenta „Liste standarda interoperabilnosti“ koriste se sledeće definicije standarda:

- “Standard je objavljen dokument koji određuje specifikaciju i postupke osmišljene da obezbede da dokumenti, materijal, metod ili servis ispunjavaju svoju svrhu i doslednu primenu i namenu.“ [43];
- „Termin standard u oblasti tehničkih standarda i propisa predstavlja tehničku specifikaciju koja je odobrena od strane međunarodnog, evropskog ili nacionalnog tela za standardizaciju.“ [24].

Cilj uspostavljanja „Liste standarda interoperabilnosti“ je „koordinacija i usklađivanje poslovnih procesa i informacionih arhitektura koje premošćavaju unutrašnje i međusobne organizacione granice“ [44]. U okviru interoperabilnosti napravljena je klasifikacija za kategorizaciju standarda gde su standardi kategorizovani uz pomoć „modela slojeva“.

Slika 2.15: Slojeviti model za kategorizaciju standarda [40]

Na slici 2.15 je prikazan slojeviti model koji se koristi za kategorizaciju standarda.

„Modeli slojeva“ su u širokoj primeni i koriste se za klasifikaciju funkcija u okviru IKT sistema radi njihovog pojednostavljenja kroz razdvajanje njihovih funkcija na nivoe. Komponente uobičajeno komuniciraju sa drugima samo na susednim nivoima i na standardizovan način.

Osnovne strukturne komponente „modeli slojeva“ su:

- Mreža;
- Integrisanje podataka;

- Poslovne usluge;
- Pristup i prezentacija;
- Sloj veb servisa.

Strukturne komponente koje se primenjuju na sve slojeve su:

- Bezbednost;
- Rukovođenje i upravljanje.

Bezbednost prožima sve slojeve „modela slojeva“, čime ukazuje na činjenicu da je bezbednost veoma značajna za e-Upravu Republike Srbije i da je neophodan profesionalni pristup u implementaciji bezbednosti, kao i aktivno praćenje i unapređenje bezbednosti u periodu eksploatacije e-Uprave. „Lista standarda interoperabilnosti“ sadrži standarde na različitim nivoima koji su osmišljeni tako da po potrebi nude različite nivoe bezbednosti.

2.4.2. Bezbednost e-Uprave Republike Srbije

Pitanje bezbednosti je veoma važno za e-Upravu Republike Srbije i obrađuje se sa posebnom pažnjom. Veoma je važno da se svi učesnici u kreiranju, razvoju i održavanju eUprave pridržavaju zajedničkih pravila.

Jedno od osnovnih načela NOI Republike Srbije predstavlja „bezbednost i privatnost“. U ovom načelu se kaže: „Korisnici elektronskih usluga moraju biti sigurni da se njihova interakcija sa sistemima elektronske uprave vrši u bezbednom okruženju i u potpunosti u skladu sa odgovarajućim propisima, kao što su propisi o privatnosti i zaštiti podataka. To znači da usluge elektronske uprave moraju garantovati poštovanje privatnosti građana i poverljivosti informacija koje pružaju sami ili koje pružaju privredni subjekti.“ [45]

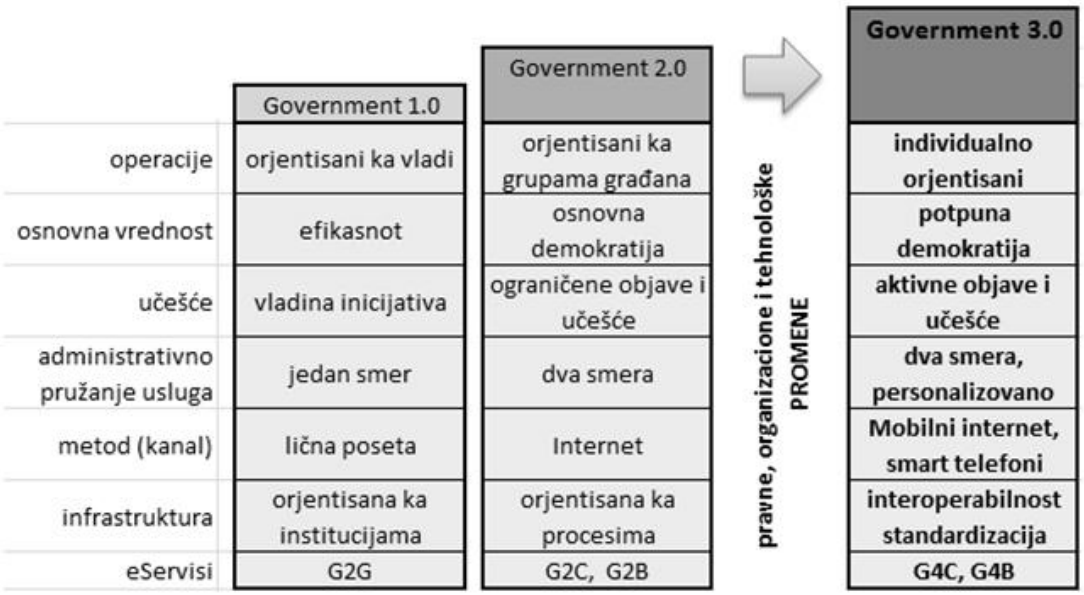
Upravljanje informacionom bezbednošću se posebno odnosi prema uslugama e-Uprave koje se realizuju po raznim kanalima komunikacije sa građanima u okviru grupe servisa iz interakcije Vlada određene zemlje za građane (*eng. Government for Citizens – G4C*). Prema UNITED NATIONS, E-GOVERNMENT SURVEY 2014 [46], Definisani su kanali za komunikacije državnih organa sa građanima:

1. Brojač (licem u lice) servis;
2. Telefon (govorni) servis i call centar;
3. Veb portal;
4. I-mejl;
5. SMS i druge usluge za razmenu poruka;
6. Mobilni portal (*eng. Mobile Website*);
7. *Mobile App*;

8. Društvene mreže;
9. Javni saobraćaj.

Globalni trend ostvarivanja *open data* države svakako zahteva osetljivo postupanje u domenu informacione bezbednosti. Evolucija koncepta e-Uprava sa eGov 2.0 na eGov 3.0. svakako zahteva posebnu relaciju za ostvarivanje transverzale informaciona bezbednost – interoperabilnost (Slika 2.16).

Za uspešnu informacionu bezbednost neophodna je interoperabilnost sistema kako bi se isti princip primenio na svim nivoima. Potrebno je izgraditi najpre sloj servisa/infrastrukture u okviru državnog dela elektronskog poslovanja G2G sa posebnom pažnjom na zaposlene državne službenike G2E. Sve ovo jeste preduslov za uspostavljanje svakog servisa iz domena G4C ili iz domena interakcije Vlade za privredne subjekte (*eng. Government to Businesses – G4B)*. „Pametne“ servise može da pruža samo ona Vlada koja ima odgovarajuću infrastrukturu i edukovane sopstvene kapacitete. Ovo se postiže stalnom težnjom na relaciji interoperabilnost - informaciona bezbednost. Kao što je za uspostavljanje „Pametne“ e-Uprave (eGov 3.0) potrebno imati edukovane državne službenike koji će u realnom okruženju pružati zahtevane servise potrebno je imati i edukovane građane koji će imati kapacitet da koriste „pametne“ servise. Posebna pažnja u ovom novom konceptualnom/tehnološkom okruženju nephodna je u domenu informacione bezbednosti. Uvođenje ISO 27001 standarda samo po sebi nije dovoljno da bi se obezbedio neophodan nivo informacione bezbednosti [47]. Sistem e-Uprava koji na nivou projektovanja interoperabilnosti projektuje i informacionu bezbednost ima šansu da bude realan funcionalan sistem. Sve što se u domenu informacione bezbednosti naknadno projektuje kao iznuđeni, do tog trenutka nepoznati faktor analize rizika, ne može se sistemski jednostavno rešiti [45].



Slika 2.16: . Evolucija koncepta e-Uprave

Usvojena „Lista standarda interoperabilnosti“ je usaglašena sa zahtevima i ciljevima NOI Republike srbije i obuhvata aspekte tehničke interoperabilnosti NOI Republike srbije.

Specifikacija standarda bezbednosti je navedena u tabeli 6. „Liste standarda interoperabilnosti“ [40] sadrži standarde koji su sačinjeni tako da nude različite nivoe bezbednosti u slojevima i obuhvata: naziv standarda, status, verziju, kao i izvor – odnosno URL/veb adresu za pristup detaljnim specifikacijama i podacima/informacijama za preporučeni standard.

U „Listi standarda interoperabilnosti“ obuhvaćeni su sledeći standardi po grupama:

- Opšti bezbedonosni standradi: SRPS ISO/IEC 27001:2014 (sr) i SRPS ISO/IEC 15408-1:2014 (en);
- Mrežni sloj: HTTPS, IP-SEC, ESP i TLS;
- Sloj integrisanja podataka: Zajednički sistem PKI, XML šifrovanje, CadES, PadES, XAdES (Deo 1), PAdES (Deo 2), PAdES (Deo 3), PAdES (Deo 4), PAdES (Deo 5), XML digitalni potpis, DSS OASIS;
- Sloj poslovnih usluga: S/MIME;
- Sloj veb servisa: WS-bezbednost, WS-poverenje, WS-savez, SAML, XACML, ID-WSF WS – Security;
- Infrastruktura javnog ključa (PKI): RFC3467 i ETSI TS 102 176;
- Šifrovanje: AES i RSA;
- Heširanje: SHA-2.

3. *Question answering* sistemi

Zahvaljujući širenju interneta, danas je njihovim korisnicima dostupna velika količina podataka. Ovi podaci mogu zadovoljiti skoro svaku potrebu za informacijama, ali bez odgovarajuće pretraživačke podrške postaju praktično neupotrebljivi. Ovakva situacija je uticala na iznalaženje novih pristupa za pronalaženje informacija, poput *question answering* (QA) sistema.

QA sistem je vrsta sistema za pretraživanje informacija koji obrađuju upite postavljene na prirodnom jeziku i vraćaju ili ekstrahuju odgovor iz struktuiranih (baze podataka) ili nestrukturiranih (tekstualnih) izvora. Ono što ove sisteme razlikuje od sistema za pretraživanje informacija je činjenica da se upiti postavljaju na prirodnom, govornom jeziku, a ne putem ključnih reči. To znači, da sistem mora da prepozna tip odgovora koji korisnik očekuje, kako bi mogao da vrati konkretan odgovor, pasus ili odlomak u kome se odgovor može pronaći. U principu, složenost ovih sistema leži u uspostavljanju podrazumevanih (implicitnih) odnosa između upita i odgovora. Postoje pristupi koji ove sisteme posmatraju kao korak koji vodi ka semantičkom veb-u [49].

Razvoj sistema za pronalaženje odgovora (*eng. answer searching systems*) na upite postavljene prirodnim jezikom, odnosno QA sistema je raznovrstan, što treba uzeti u obzir prilikom njegovog korišćenja, zajedno sa svim njihovim specifičnostima kao što su: različiti korisnički profili, heterogeni izvori podataka, implementacione tehnologije, konkretne oblasti primene i same vrste podataka. QA daju konkretne odgovore na postavljene upite [49]. U ovom trenutku, QA sistemi su fokusirani na davanje kratkih odgovora u vidu polu-informacije, definicije ili vremenske odrednice na postavljeni upit. Ako korisnik postavi upit putem interneta, npr. „Ko je naš najpoznatiji pisac i nobelovac?“, sistem treba da obezbedi sledeći odgovor: „Ivo Andrić“, kao i niz linkova ka stranicama gde je moguće pronaći informacije o nobelovcima, odnosno dodatne informacije o našem nobelovcu i njegovim romanima. Ova operacija se suštinski razlikuje od one koju vrše aktuelni pretraživači kao što su Google ili Yahoo, gde se kao odgovor dobija niz referenci ka sajtovima koje korisnik mora sam da „prečisti“ i pregleda kako bi pronašao traženu informaciju. Ovi pretraživači, takođe, preuzimaju i informacije koje nisu u vezi sa temom pretrage, pa pronalaženje informacija postaje složenije.

Postoji nekoliko pristupa za ekstrahovanje odgovora iz običnog teksta za ovakvu vrstu upita. Većina njih koristi prednost nekih stilskih konvencija koje upravo koriste pisci pri uvođenju novih termina. Ove konvencije obuhvataju neke tipografske elemente koji mogu biti predstavljeni nizom leksičkih obrazaca. U početku, ovi obrasci su bili ručno kreirani [50,51], međutim, usled toga što su složeni za ekstrahovanje i mogu biti domenski zavisni, trenutni pristupi koji se bave ovom problematikom teže njihovom automatskom kreiranju [52,53].

Prvi pristup u arhitekturi razvoja QA sistema imao je četiri glavne komponente:

- **Klasifikator upita** (*eng. Question Classifier*). „Ubacuju“ se upiti na prirodnom jeziku i klasifikator upita odgovoran je za identifikovanje vrste upita (npr. šta, gde), vrste ekstrahovanog odgovora, fokusa odgovora i odgovarajućeg semantičkog značenja.
- **Dokument-Odgovor** (*eng. Answer Document*) je odgovoran za traženje i identifikaciju relevantnih dokumenata u kojima može biti pronađen odgovor; pretraga se vrši u okviru nestruktuiranih izvora podataka koji se mogu naći u različitim formatima.
- **Ekstrahovani kandidat odgovora** (*eng. Extract Candidate Answer*), u okviru koga se identifikuje potencijalni odgovor pronađen u relevantnim dokumentima ili izvorima selektovanim putem gore navedenih komponenti.
- **Selektovanje odgovora** (*eng. Answer Selection*) je možda najvažnija komponenta, jer ona generiše odgovor na upit [52].

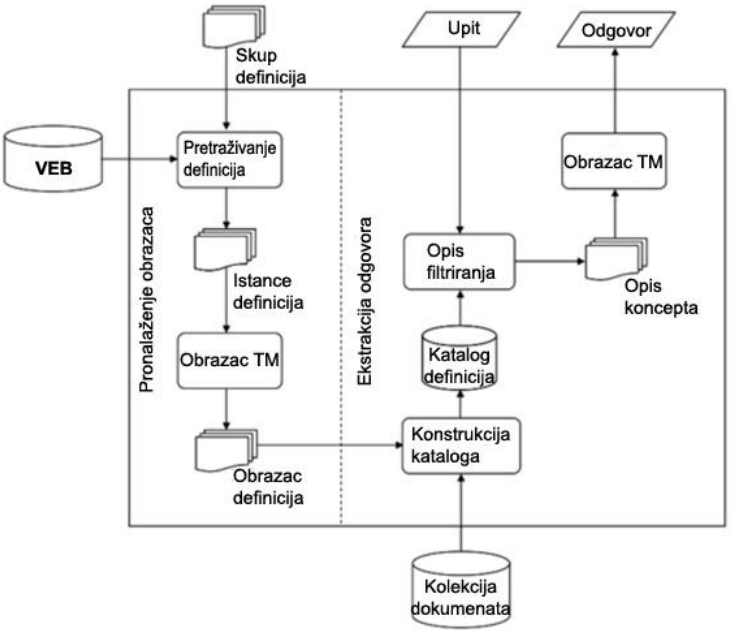
Za razliku od prvog prikazanog, drugi pristup u razvoju QA sistema ima ukupno tri glavne komponente:

- **Komponenta za obradu upita** (*eng. Question Processing Component*) koja vrši tokenizaciju i tagovanje, identifikaciju ključnih reči, gramatičku analizu upita, identifikaciju dvosmislenih reči, identifikaciju tipa očekivanih odgovora i proširivanje značenja ključnih reči;
- **Komponenta za pretragu** (*eng. Search Component*) generiše upite i vrši pretraživanje izvora podataka dostupnih na veb-u;
- **Komponenta za ekstrakciju odgovora** (*eng. Answer Extraction Component*) filtrira podatke koje je pronašla komponenta za pretragu, prepoznaje entitet, identifikikuje odgovor i vrši proveru ispravnosti [54].

Takođe, postoji nekoliko pristupa za davanje odgovora u vidu definicija. Većina njih koristi leksičke obrasce za ekstrahovanje odgovora na zadati upit iz kolekcije metapodataka. U zavisnosti od složenosti tražene definicije, zavisi kompleksnost upotrebe obrazaca. Na primeru prostog slučaja uvođenja nove reference u tekst, stilske konvencije moraju biti jasne.. Kao posledica toga, praktični leksički obrasci moraju biti jednostavni i precizni. Pod ovom pretpostavkom, upiti kao što je „Šta je X ?“ i „Ko je X?“ su lako rešivi.

Postojeći pristupi za davanje odgovora na upite u formi definicije se međusobno razlikuju po načinu na koji određuju obrasce definicija i načinu na koji ih koriste. Postoje neki radovi koji primenjuju obrasce koji su ručno kreirani [55] i oni drugi koji automatski kreiraju obrasce iz skupa gotovih primera [52, 53]. Na primer, metod koji se bavi automatskim kreiranjem obrazaca iz španskog CLEF 2005 [56] foruma sastoji se od dva glavna koraka:

1. U prvom koraku, metod primenjuje algoritam za analizu kako bi otkrio niz tekstualnih obrazaca sa veb-a koji su u vezi sa postavljenom definicijom, tj. opisom. Ovi leksički obrasci omogućavaju povezivanje lica sa njihovim pozicijama (funkcijama) i akronima sa njihovim opisima. Sličan metod ovome je metod koji uzima u obzir sve otkrivene obrasce, npr. ne ocenjuje i ne selektuje analizirane šablone (*eng. mined patterns*). Stoga, glavna razlika je da dok se drugi fokusiraju na izbor malog broja veoma preciznih obrazaca, u ovom slučaju je akcenat na otkrivanju velikog broja međusobno isključivih obrazaca.
2. U drugom koraku, metod primenjuje obrasce kroz kolekciju ciljnih dokumenata, kako bi se odgovorilo na date upite. Način na koji se koriste obrasci za odgovor na upite u formi definicije je sasvim nov. U radovima [53 , 50 ,51] primenjuju se obrasci na skup relevantnih članaka i očekuje se da će najbolji (sa visokim preciznostima) obrasci omogućiti identifikaciju relevantnih odgovora. Nasuprot tome, sličan metod primenjuje sve otkrivene obrasce na celu kolekciju metadokumenata i kreira opšti katalog. Kada stigne upit, analizira se katalog definicija (pretražuje se po katalogu), kako bi se utvrdilo koji je odgovor najprihvatljiviji za postavljeni upit. Na ovaj način, ekstrahovanje odgovora ne zavisi od sistema za pronalaženje pasusa i koristi prednost redundantnosti cele kolekcije.



Slika 3.1: Generalna šema metode

Slika 3.1 prikazuje generalnu šemu navedene metode. Sastoji se od dva glavna modula: jedan se fokusira na otkrivanje definisanih obrazaca, a drugi na ekstrakciju odgovora. Modul za otkrivanje obrazaca koristi mali skup parova po principu termin - opis kako bi sa veb-a sakupio prošireni skup definicija datog slučaja. Potom, primenjuje metodu dubinske analize teksta nad prikupljenim slučajevima kako bi se otkrio skup definicija za osnovne obrazace.

Modul za ekstrakciju odgovora primenjuje TM obrasce preko kolekcije ciljnih dokumenata u cilju stvaranja kataloga definicija koji se sastoji od niza potencijalnih koncept - opis parova. Kasnije, u zavisnosti od upita, on iz kataloga izvlači skup povezanih opisa sve do traženog pojma. Konačno, analizira selektovane opise kako bi pronašao što adekvatniji odgovor na dati upit.

Važno je primetiti da se u ovom procesu otkrivanja obrasca vrši *off-line*, dok ekstrakcija odgovora, osim kada je u pitanju izgradnja kataloga definicija, vrši *on-line*. Takođe, važno je napomenuti da se ovaj pristup razlikuje od klasičnog QA pristupa. Predloženi metod ne razmatra nijedan modul za pronalaženje dokumenta ili pasusa.

3.1. TM i e-Uprava

Informacione i komunikacione tehnologije imaju kapacitet da unaprede proces kojim javna uprava uključuje građane u kreiranje javne politike i javnih projekata. Iako se veliki deo usluga (propisa) republičke Vlade sada može naći u digitalnom obliku (često su na raspolaganju i *on-line*), usled njihove složenosti i raznovrsnosti, nije nimalo jednostavno identifikovati one koji su relevantne za određeni sadržaj. Isto tako, pojavom brojnih elektronskih *on-line* foruma, društvenih sajtova i blogova, povećana je mogućnost sakupljanja peticija građana i stavova zainteresovanih strana o politici Vlade, kao i davanja predloga. Međutim, obim i složenost analize nestrukturiranih podataka ovaj proces čini nimalo jednostavnim. S druge strane, proces analize podataka sadržanih u tekstu je prešao dug put od jednostavne pretrage ključnih reči do discipline koja može da se nosi sa mnogo složenijim zadacima.

3.1.1. TM aplikacije u e-Upravi

Transformisanje konvencionalnih Vladinih servisa u servise e-Uprave najavljuje novu eru u radu javnih službi. Tradicionalni Vladini servisi mogu biti zamenjeni servisima e-Uprave koji su boljeg kvantiteta i kvaliteta i koji mogu da zadovolje potrebe i povećaju zadovoljstvo građana, koristeći IKT. e-Upravljanje ima za cilj da uspostavi interakciju između vlasti i građana (G2C), vladinog sektora i poslovnih subjekata (G2B) i ministarstava u okviru javne uprave kako bi se posao obavljao (G2G) na transparentniji i jeftiniji način [57]. Sve veći broj informativnih tekstova u vezi sa odlukama Vlade, direktivama, pravilima i propisima sada se distribuira na vebu pomoću raznih portala, tako da ih građani mogu pretraživati i pregledati. Ovo podrazumeva, da su oni koji traže informacije u stanju da razreše brojne i kompleksne formalno pravne dokumente [58]. Propisi su obimni, međusobno ne povezani i često dvosmisleni. Informacije vlade su u nestrukturiranom ili polustruktuiranom obliku. Izvori su višestruki (državni organi ili organi lokalne samouprave), a formati su različiti, što stvara ozbiljnu prepreku običnim građanima u njihovom traženju, razumevanju i korišćenju informacija.

U okviru G2G, ministarstva imaju još veću potrebu za sistemom koji je u stanju da omogući pronalaženje informacija, razmenu podataka, homogenost metapodataka i pravilno širenje informacija preko administrativnih kanala nacionalne, regionalne, državne i lokalne vlasti [59]. Sve veća potražnja i složenost državnih propisa o različitim

aspektima ekonomskog, društvenog i političkog života, zahteva uspostavljanje naprednog i baziranog na znanju okvira za prikupljanje, protok i distribuciju informacija. Na primer, ako kreatori politike nameravaju da donesu novi akt, trebalo bi da dobro poznaju sva postojeća akta koji se tiču iste teme, bilo da je sadržaj novog zakona u sukobu sa postojećim ili je već uključen u postojeće akte. Isto tako, propisi se često ažuriraju i zato su potrebni alati koji mogu otkriti nedoumice, nedoslednosti i kontradikcije [60] s obzirom da pravilnici, izmenjene odredbe, pravni presedani i smernice zajedno stvaraju masu polustrukturiranih dokumenata sa potencijalno sličnim sadržajem, ali sa mogućim razlikama u formatu, terminologiji i kontekstu. Informacione infrastrukture koje mogu da konsoliduju i uporede regulatorna dokumenata će u velikoj meri povećati i pomoći u razumevanju postojećih propisa i donošenju novih.

Poželjno je da propisi Vlade budu jednostavni za pronalaženje i razumljivi pravnicima, kreatorima politike, kao i široj javnosti, tj. građanima. Uprkos mnogim pokušajima, smatra se da e-Uprava još uvek nije uspostavila servise usmerene u potpunosti ka građanstvu već uglavnom servise fokusirane na internu efikasnost [57]. Uzimanje u obzir mišljenja građana, dobijenih putem elektronskih medija kroz učešće na forumima i diskusijama, može biti pouzdanije od tradicionalnih metoda zasnovanih na istraživanju javnog mnjenja i pomoći da se izbegnu lažne izjave. Ovo, takođe, drastično menja metode analize trendova mišljenja građana, kao i tačnost procene njihovih mišljenja. To smanjuje troškove, povećava učešće građana i obezbeđuje blagovremeno informisanje. Moguće je da argumenti koji dovode do značajnih promena u mišljenju mogu biti otkriveni. Međutim, obim i složenost analize nestrukturiranih podataka čine ovo daleko težim. TM može da obradi nestrukturirane podatke koji dovode do većeg razumevanja teksta u tumačenju drugih na istu temu. Ovo je posebno važno kada se radi o izražavanju javnog mnjenja, gde su argumenti za i protiv određene pozicije važni za identifikovanje i procenu, ali je izuzetno teško ekstrahovati ih zbog njihovog skladištenja u formatu prirodnog jezika [61].

3.1.2. Dubinska analiza podataka

Dubinska analiza podataka je koncipirana 1990-ih kao sredstvo za rešavanje problema analiziranja ogromne količine podataka koji su dostupni čovečanstvu i koja se povećava u kontinuitetu. S obzirom na činjenicu da se većina podataka (preko 80%) skladišti kao tekst, DM ima još veći potencijal [62]. DM je relativno nova interdisciplinarna oblast koja spaja oblasti statistike, mašinskog učenja, pronalaženja informacija, lingvistike i obrade prirodnog jezika. Kaže se da je računar otkrio nove, do tada nepoznate informacije, tako što je automatski izvlačio podatke iz različitih pisanih izvora [63]. TM se razlikuje od obične pretrage teksta ili veb-pretrage, gde je cilj da se odbaci nebitan materijal kako bi se identifikovalo ono što korisnik traži, u kontekstu pretrage teksta, pri čemu korisnik zna šta traži, a taj (pisani) materijal već postoji. Jedan od ključnih elemenata u TM je da se dođe do nepoznate informacije, povezivanjem postojećih tekstualnih podataka radi kreiranje nove činjenice ili hipoteze. Prema tome, TM veoma podseća na DM, pa ga neki smatraju za širu verziju istog. Suština napuštanja matične discipline DM je u vrsti podataka koji treba da budu analizirani. Dok se DM bavi uglavnom numerički strukturiranim podacima, TM se bavi nestruktuiranim podacima.

Ipak, zadatak Sistema za podršku u odlučivanju (*eng. Decision Support System* – DSS) koji se bazira na TM izgleda kao veći izazov nego analiza strukturiranih podataka, a postojanje ogromne količine informacija u elektronskom tekstualnom obliku je dovelo do intenzivnog istraživanja tehnika TM, pa su mnogi od ovih izazova već prevaziđeni.

Najveći potencijal primene TM je u oblastima u kojima se generiše ili prikuplja velika količina tekstualnih podatka za vreme transakcija. Na primer, poslovi kao što su izdavaštvo, pravo, zdravstvo i farmaceutska istraživanja, kao i područja kao što su upravljanje žalbama potrošača (povratne informacije) i marketing programa sa fokusom na određene grupe biće najveća oblast primene TM. O inovativnim aplikacijama u kontekstu personalizacije u oblasti B2C e-trgovine, konkurentske inteligencije, analize zadovoljstva potrošača i filtriranje mejlova, diskutuje se već duže vreme u brojnim naučnim člancima [64-67].

Tehnologije koje se koriste u TM uključuju: IR, izdvajanje informacija (*eng. Informations Extraction* - IE), klasifikacija, klasterizacija, praćenje teme, generalizacija, davanje odgovora na postavljeni upit i detekcija pravila asocijacije.

Izdvajanje informacija: algoritmi za ekstrakciju informacija identifikuju ključne fraze i odnose unutar teksta. To se radi traženjem unapred definisanih sekvenci u tekstu, putem procesa pod nazivom „podudaranje obrasca". Algoritmi izvode zaključke o odnosima između svih identifikovanih sekvenci da bi se korisniku obezbedio smisleni uvid. Ova tehnologija može biti veoma korisna kada se radi o velikim količinama teksta.

Klasifikacija: Klasifikacija podrazumeva identifikovanje glavnih tema dokumenta smeštanjem dokumenta u unapred definisan skup tema. Klasifikacija ne pokušava da obradi aktuelne informacije kao što je slučaj sa izvlačenjem podataka. Tokom klasifikacije samo se broje reči koje se pojavljuju u tekstu i posle brojanja, identifikuju glavne teme koje taj dokument pokriva. Klasifikacija se često oslanja na leksikon sinonima za koji su teme unapred definisane, a odnosi se identifikuju traženjem šireg i užeg smisla, sinonima i srodnih pojmova.

Klasterizacija: Klasterizacija je tehnika koja se koristi za grupisanje sličnih dokumenata, ali se razlikuje od kategorizacije po tome što se dokumenti grupišu na osnovu međusobne sličnosti umesto kroz upotrebu unapred definisanih tema. Osnovni algoritam klasterizacije kreira vektor termina za svaki od dokumenata i meri koliko se dobro taj dokument uklapa u svaki od unapred definisanih klastera.

Praćenje teme: Sistem praćenja teme funkcioniše održavanjem korisničkih profila i na osnovu dokumenta koje korisnik pregleda predviđa i druge dokumente od interesa za korisnika. Neki od boljih TM alata omogućavaju korisnicima da selektuju određene kategorije od interesa, a mogu čak i automatski da donesu zaključak od interesa za korisnika na osnovu njegove istorije čitanja ili „klikova" na tu informaciju.

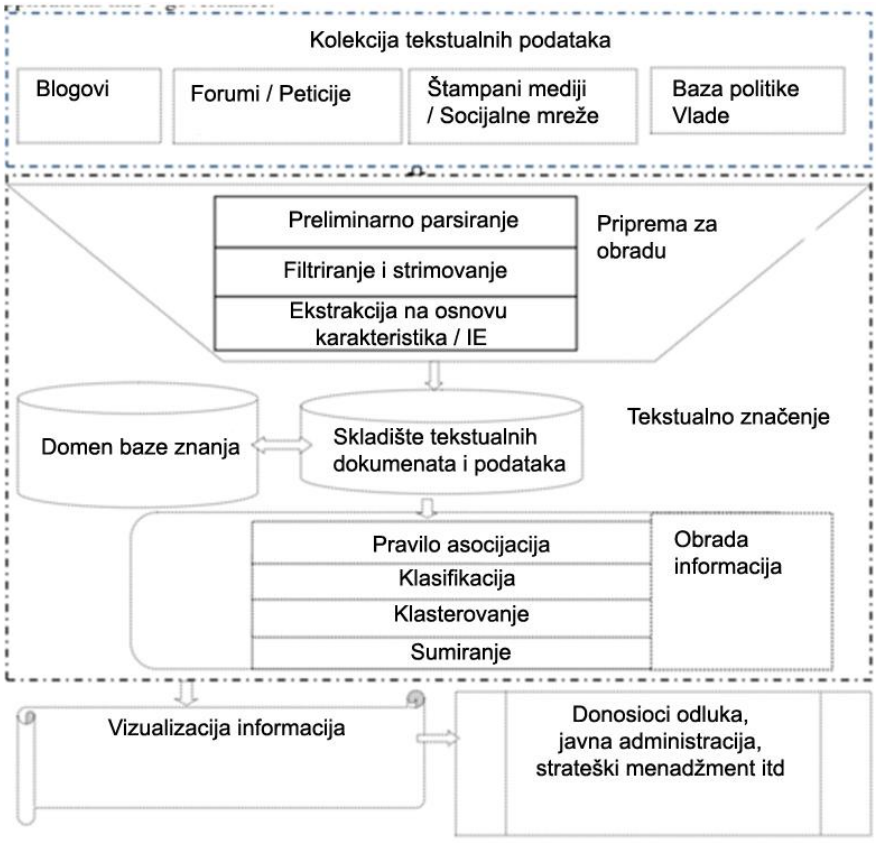
Sumarizacija: Sumarizacija teksta je izuzetno korisna kao pokušaj da se shvati da li obiman dokument zadovoljava potrebe korisnika ili ne i na osnovu toga zaključi da li vredi čitati dokument radi dodatnih informacija. Ključno za sumarizaciju je da smanjuje dužinu i detalje dokumenta zadržavajući njegove glavne delove i ukupan smisao.

Davanje odgovora na postavljeni upit: Još jedna oblast primene TM je davanje odgovora na postavljeni upit, koji se bavi pronalaženjem najadekvatnijeg odgovora na postavljeni upit QA može koristiti više od jedne TM tehnike.

Detekcija pravila asocijacije: U učenju pravila asocijacije (*eng. Association Rules*), fokus je na proučavanju odnosa i veza među temama ili opisnim pojmovima koji se koriste za prezentovanje niza povezanih tekstova. Cilj je pronaći važna pravila u učenju kroz asocijaciju unutar nekog korpusa, tako da prisustvo skupa tema u nekom članku može da podrazumeva prisustvo drugih tema istog korpusa.

3.1.3. Tehnike TM

Iako relativno nove tehnike, smatraju se dovoljno zrelim da budu uključene u skoro sve komercijalne DM softverske pakete. Sagledavanjem karakteristika nekih popularnih DM softvera koji imaju TM module, uočeno je da je TM prešao iz domena istraživanja u domen industrijske tehnologije i može se koristiti u brojnim zahtevnim aplikacijama kao što su one u e-Upravi (slika 3.2).



Slika 3.2: Arhitekturu DSS-a zasnovana na TM-u za e-Upravu

Da bi se implementirao jedan inteligentan sistem uopšteno, prvo treba odrediti potrebne izvore podataka, kao što su baze podataka Vlade neke zemlje, žalbe građana sa relevantnih veb-portala, *on-line* forumi (da bi se građanima omogućio proces diskusije o prestižnim projektima Vlade) i na kraju, ali ne manje značajno, društvene mreže. Društvene mreže imaju ogromnu popularnost u današnje vreme, naročito one iz kojih mogu da se ekstrahuju određeni podaci, na osnovu kojh se može kreirati mišljenje neke zainteresovane strane. Kad se analiziraju nestrukturirani podaci iz različitih izvora i u različitim formatima (PDF, DOC, DOCX, XML, JPG, HTML i drugi), trebalo bi koristiti sistem raščlanjivanja transformisanih dokumenata u format koji ima sposobnost da upravlja nerestruktuiranim i polustruktuiranim podacima. Sledeći zadatak je pretraživanje informacija po ključnim rečima, tj. karakteristikama. To podrazumeva primenu procesa tokenizacije, filtriranja, strimovanja, indeksiranja i prečišćavanja. Međutim, u slučaju kada tradicionalna tehnika, poput ekstrahovanja ključnih reči nije u mogućnosti da bude podržana, onda bi trebalo primeniti drugu tehniku za izdvajanje karakteristika koja podrazumeva generičke karakteristike, domen specifične karakteristike i ekstrahovanje koncepta, što zahteva i preuređivanje same baze podataka. Nakon što karakteristike i informacije budu sačuvane u skladištu tekstova, tj. podataka, može se preći na realizaciju pravila analize asocijacije, klasterizacije, klasifikacije i generalizacije u cilju njene obrade u smislenu informaciju.

Modul za upravljanje ekstrahovanjem odgovora na postavljeni upit kao sastavni deo iteligentnih QA sistemima zasniva se upravo na TM pristupu. Svrha postojanja ovih modula u QA sisteme je da se pronađe što više adekvatnih opisa za traženi pojam iz kataloga definicija na automatizovan način.

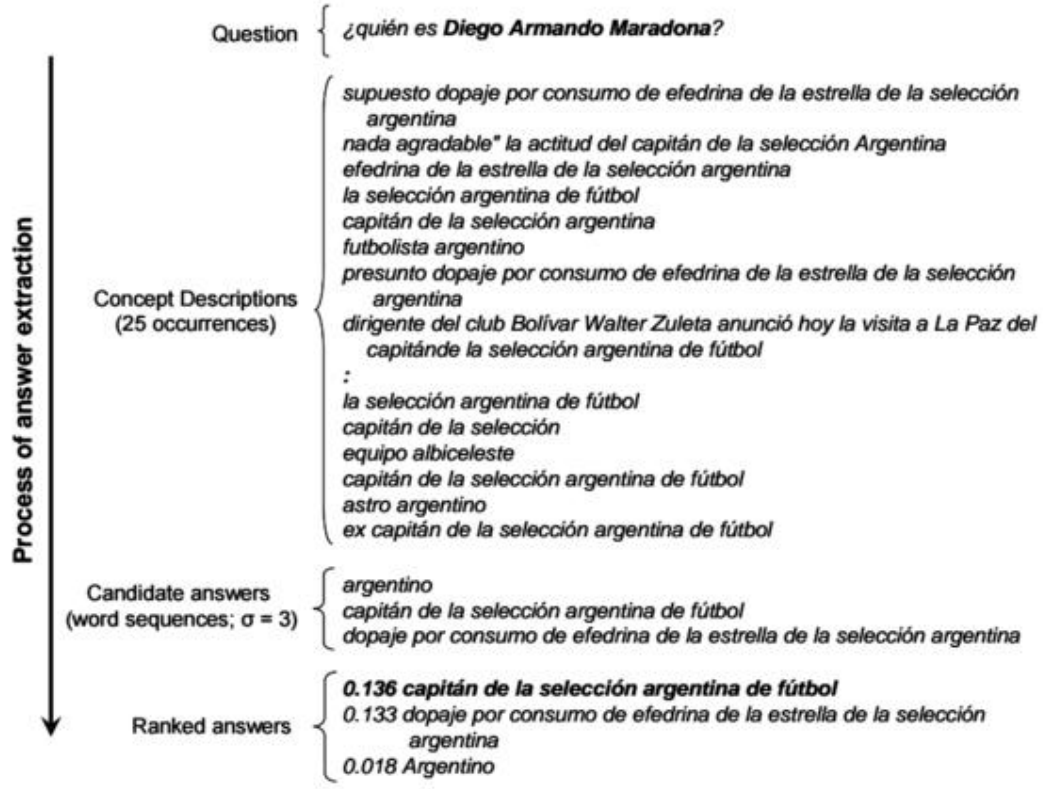
S obzirom da obrasci definicija vode kreiranju kataloga definicija, oni sadrže raznovrsne informacije, uključujući i nepotpune i netačne opise za mnoge termine. Međutim, očekuje se da tačnijih informacija može biti više od pogrešnih. Ovo očekivanje podržava ideju korišćenja TM tehnika kako bi se napravila razlika između adekvatnih i manje verovatnih odgovora na zadati upit.

Primena modul za upravljanje ekstrahovanjem uključuje sledeće korake:

1. **Kreiranje kataloga:** U ovoj fazi, definicije obrazaca otkrivene u prethodnoj fazi (u modelu za otkrivanje obrazaca) primenjuju se nad kolekcijom ciljnih dokumenata. Rezultat je skup podudarnih segmenata za koje se pretpostavlja da sadrže termin i njegov opis. Katalog definicija obrazaca se kreira sakupljanjem svih podudarnih segmenata.
2. **Filtriranje opisa:** Kada je u pitanju konkretno pitanje, ovom se procedurom iz kataloga definicija ekstrahuju svi opisi koji odgovaraju traženom terminu. Kao što je pretpostavljeno, ovi „verovatni" opisi mogu da sadrže nepotpune i netačne informacije. Međutim, očekuje se da mnogi od njih u sebe sadrže, kao podskup, potreban odgovor.

3. **Analiza odgovor**a**:** (*eng. Answer mining*). Ovaj proces ima za cilj pronalaženje samo jednog odgovora na zadati upit iz skupa ekstrahovanih opisa. Čine ga tri glavne faze: priprema podataka, analiza podataka i rangiranje odgovora:
    - Faza pripreme podataka se fokusira na homogenizovanje opisa u skladu sa traženim konceptom. Glavna aktivnost je transformacija ovih opisa u formate za slučajeve korišćenja.
    - Faza analize podataka koristi algoritam za analizu sekvenci za dobijanje svih veoma učestalih sekvenci reči iz skupa opisa. Svaka sekvenca ukazuje na eventualni odgovor na zadati upit.
    - U fazi rangiranja odgovora, procenjuje se svaki eventualni odgovor na osnovu učestalosti pojavljivanja njegovih podsekvenci. Ideja je da potencijalni odgovor koji je sastavljen od čestih podsekvenci ima veću verovatnoću tačnosti nego onaj sastavljen od retkih podsekvenci. Bira se sekvenca sa najviše rangiranim rezultatima za tačan odgovor.

Primer procesa ekstrakcije odgovora u metodi za definisanje odgovora na dati upit preko foruma *Cross-Language Evaluation Forum*–CLEF [68] prikazan je na slici 3.3. Primer demonstrira situaciju kada je postavljen upit na španskom jeziku: „Ko je Dijego Armando Maradona?“.



Slika 3.3: Tok podataka u procesu ekstrakcije odgovora

Važno je razjasniti da upit može imati i više tačnih odgovora. U skladu sa CLEF-om, odgovor je tačan samo ako postoji pasus koji ga podržava. Iz tog razloga postoje i drugi

tačni odgovori poput: „bivši kapiten Argentinskog nacionalnog fudbalskog tima" i „Argentinska zvezda".

3.2. Analiza maksimalne učestalosti sekvenci reči (*eng. Mining Maximal Frequent Word Sequences*)

Pretpostavimo da je D skup tekstova (tekst može predstavljati ceo dokument ili samo jednu rečenicu) i svaki tekst se sastoji od niza reči. Zatim, imamo sledeće definicije [69]:

Definicija 1. Sekvenca $p = a_1 \dots a_k$ je podsekvenca sekvence *q* ukoliko se sve tačke $a_i, 1 \leq i \leq k$, javljaju u *q*, i one se javljaju u istom redosledu kao i u *p*. Ako je sekvenca *p* podsekvenca sekvence *q*, takođe kažemo da se *p* javlja u *q*.

Definicija 2. Sekvenca *p* je učestala sekvenca u *D* ako je *p* podsekvenca od najmanje *σ* tekstova *D*, gde je *σ* dat prag učestalosti.

Definicija 3. Sekvenca *p* je maksimalno učestala sekvenca u *D* ako ne postoji nikakva sekvenca *p'* u *D* tako da je *p* podsekvenca *p'* i *p'* je učestala u *D*.

Nakon uvođenja maksimalno učestalih sekvenci reči, problem analize maksimalno učestalih sekvenci reči može se formalno definisati: datom kolekcijom tekstova D i proizvoljnom vrednošću celog broja σ koji iznosi $1 \leq \sigma \leq |\mathrm{D}|$, moguće je prebrojati sve učestale sekvence reči u skupu D.

Implementacija metode analize sekvenci nije trivijalan zadatak zbog svoje računarske složenosti, a sam algoritam je detaljno opisan u radu [70].

3.3. Rangiranje rezultata pretrage (*eng. ranking score*)

Ova mera ima za cilj da kreira bolji odgovor na zadati upit. Imajući u vidu skup eventualnih odgovora (maksimalno frekfentne sekvence dobijene iz skupa opisa termina), ova mera selektuje konačan jedinstven odgovor, uzimajući u obzir učestalost pojave njegovih podsekvenci. Rezultat rangiranja *R* za sekvencu reči ukazuje na njegovu relevantnu frekvenciju. Izračunava se na sledeći način:

$$R_{p(n)} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n-i+1} \frac{f_{p_j(i)}}{\sum_{\forall\, q \in S_i} f_{q(i)}} \qquad (1)$$

U ovoj formuli uvedena su sledeće oznake radi jednostavnosti. $\mathrm{S_i}$ označava skup sekvenci veličine *i*, $\mathrm{q(i)}$ predstavlja sekvencu *q* veličine *i*, $\mathrm{p_j(i)}$ je *j*-ta podsekvence veličine *i* koja je uključena u sekvencu $\mathrm{p(n)}$, $\mathrm{f_{q(i)}}$ određuje učestalost pojava sekvence *q* u skupu opisa koncepta, tj. termina i na kraju $\mathrm{R_{p(n)}}$ označava relevantnu frekvenciju sekvence *p*.

Ideja ovog rangiranja rezultata je da ponuđen odgovor, sastavljen od učestalih podsekvenci, ima veću verovatnoću tačnosti od onog kreiranog na osnovu retkih

podsekvenci. Učestalost pojave stop-reči ne uzima se u obzir kod izračunavanje rezultata rangiranja.

3.4. Analiza podataka sadržanih u višejezičnim tekstovima (*eng. Multilingual Text Mining* – MLTM)

Može se uočiti da postoji potreba za višejezičkim TM aplikacijama (10, 20 ili više jezika), ali trenutno raspoloživi sistemi raspolažu samo sa malim brojem jezika. Postojeća rešenja mašinskog učenja su posebno obećavajuća kada je u pitanju postizanje višejezičnosti. Uzimajući u obzir i činjenice i više različitih stavova [71], višejezička obrada teksta je svakako korisna, jer je tada sadržaj informacija na različitim jezicima, komplementaran. Za pronalaženje višejezičnih informacija predlažena je TM metoda, koja se koristi kod izdvajanja veza između višejezičnih tekstova. Dokumenti napisani na različitim jezicima se prvo grupišu i organizuju po principu hijerarhije korišćenjem hijerarhijskog modela za samoorganizovanje (*eng. hierarchical self-organizing map model*). Takođe, ističe se da u domenu višejezičnog TM mora više pažnje posvetiti uspostavljanju hijerarhije višejezičnih dokumenta i izvlačenju veza iz takvih hijerarhija višejezičnih dokumenata [72]. Neki od autora TM aplikacija predlažu *Cross-Lingual Text Retrieval* (CLTR) koncept, korišćenjem osnovnog MLTM pristupa i MLTM pristupa za automatsko otkrivanje višejezičnog znanja kroz pretraživanja višejezičnih tekstova u pristupačnim sadržajima koji su alternative skupim ručno izgrađenim jezičkim resursima. Iskorišćavanjem paralelnog korpusa koji pokriva više jezika, postiže se automatski izgradnjom nezavisnog jezičkog koncepta koji „lovi" sve konceptualne odnose između višejezičnih termina [73].

3.5. Softverski paketi za TM

Pored komercijalnih TM paketa, dostupan je i veliki broj softverskih paketa otvorenog koda (*eng. open source*). Većina ovih paketa koji su trenutno dostupni besplatno ili po niskoj ceni, može biti korisna za pilot projekte i može omogućiti početnim korisnicima da odu korak dalje bez preteranog finansijskog troška. U narednoj tabeli (Tabela 3.1), navedeni su neki od paketa otvorenog koda za TM analizu.

(Sistem otvorenog koda)	Opis
Carrot2 http://project.carrot2.org	Carrot2 je internet pretraživač otvorenog koda baziran na mašinama za grupisanje rezultata pretrage. Može automatski da organizuje male zbirke dokumenata, npr. rezultate pretrage u tematske kategorije. Carrot2 nudi gotove komponente za pronalaženje rezultata pretraga iz različitih izvora, uključujući Google API, Bing API, eTools Meta Search, Lucene, SOLR, Google Desktop itd.
GATE http://gate.ac.uk	Softver otvorenog koda koji može da reši skoro svaki problem obrade teksta, sve vrste jezičkih obrada i aplikacija, uključujući glas korisnika: problem istraživanja raka, istraživanja droge, podršku pri odlučivanju, veb-mining, izdvajanje informacija, semantičke napomene. Mnoge obrazovne ustanove su već uključile GATE u svoje TM tehnike.
Natural Language Toolkit (NLTK) http://www.nltk.org	Skup biblioteka i programa za simboličku i statističku obradu NLP pomoću programskog jezika Python. NLTK je praćen brojnim struktuiranim tekstovima, pojednostavljenom gramatikom, obučenim modelima, itd. NLTK je pogodan za kurseve u mnogim oblastima, uključujući obradu prirodnog jezika, računarsku lingvistiku, empirijsku lingvistiku, kognitivne nauke, veštačku inteligenciju, pronalaženje informacija i mašinsko učenje.
RapidMiner http://rapid-i.com/content/view/181/190	Formalno *Yet Another Learning Environment* (YALE) je okruženje za mašinsko učenje, DM, TM, prediktivnu i poslovnu analitiku. *Plug-in* komponenta je specijalno dizajnirana da pripremi tekstualni dokument za analizu, kroz proces tokenizacije, izbacivanje stop-reči i strimovanje. Dodatne komponente *RapidMiner* su *Java* biblioteke koje treba dodatno instalirani u *lib\plugins* direktorijume.
Arhitektura za upravljanje nestruktuiranim informacijama (UIMA) http://uima.apache.org	Prvobitno razvijena od strane IBM-a. To je otvorena, industrijski snažna prilagodljiva i proširiva platforma za kreiranje, integrisanje i primenu rešenja za upravljanje nestrukturiranim informacijama kombinovanjem semantičke analize i komponenata pretrage. Cilj UIMA-a je da obezbedi temelj zajedničke saradnje između industrijske i akademske zajednice širom sveta i da ubrza razvoj onih tehnologija koje su ključne za otkrivanje vitalnog znanja prisutnog u sve obimnijim izvorima informacija.
Text Mining paket http://cran.r-project.org/web/packages/tm/index.html	Ovaj paket nudi funkcionalnost u upravljanju tekstualnim dokumentima, skraćuje proces upravljanja dokumentom i olakšava korišćenje heterogenih tekstualnih formata. Ovaj paket ima pozadinsku podršku zasnovanu na integrisanim podacima kako bi se minimazirali zahtevi za memorisanjem. Unapređeno upravljanje metapodacima se koristi za prikupljanje tekstualnih dokumenata kako bi se lakše koristili veliki (obogaćeni sa metapodacima) skupovi dokumenata.

Tabela 3.1: Paketi otvorenog koda za TM analizu

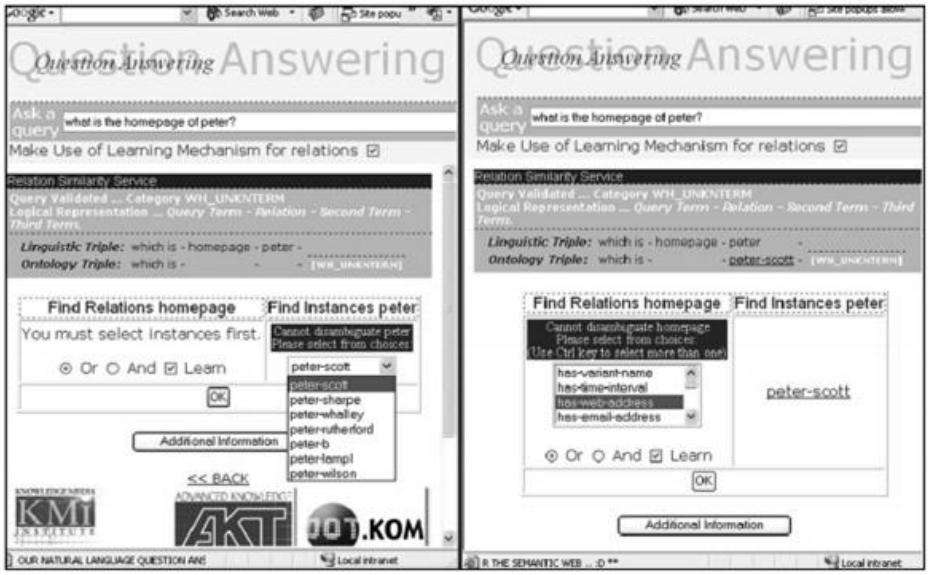
3.6. Postojeći okviri za QA sisteme

Postojeći okviri QA sistema koji su razvijeni i u upotrebi su:

- **QALL-ME.** Obezbeđuje arhitekturu koja se sastoji od tri glavna modula, održiva je i proširiva za izgradnju QA sistema na strukturiranim podacima za domen turizma. Ovaj domen je modeliran pomoću ontologije koja je korišćena kao primarni izvor primene višejezičnosti. Ona takođe koristi prostorno i vremensko zaključivanje u trenutku obrade upita i zaključivanje prililkom klasifikacije odgovora kako bi se utvrdilo koji je odgovor na postavljeni upit najprikladniji za analizu [74]. Izgled QALL-ME QA sistema prikazan je na slici 3.4.

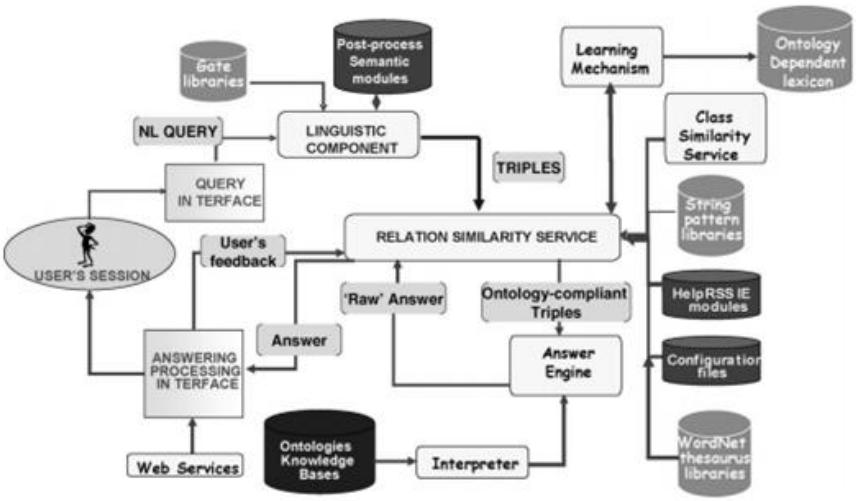
Slika 3.4: Izgled QALL-ME QA sistema [75]

- **AQUALOG**. Koristi model obrade sekvenci. Prvo se ulaz, tj. upit koje je napisan na prirodnom jeziku prevodi pomoću jezičke komponente za klasifikaciju jezika i upita (*eng. Linguistic & Query Classification*) u niz polja definicija (subjekat, predikat i objekat) tzv. triplet. Jezička komponenta koristi skup sintaksičkih napomena koje su povezane sa ulaznim upitom (eng. NL Sentence Input) kako bi klasifikovala upit. Zatim servis za utvrđivanje relacije sličnosti (*eng. Relation Similarity Service*) uzima triplete kao ulaz za postavljeni upit koji je ontološki kompatibilan, tzv. onto-triplet. Kada onto-triplet postane validan, aktivira se sistem za logičko zaključivanje (eng. inference engine) koji pretražuje bazu odgovora, ali u slučaju nevažećeg onto-tripleta, da bi se dobio važeći onto-triplet, potrebno je da korisnik definiše pojedinačno značenje višeznačnih reči s obzirom na kontekst [76]. Polje za dijalog u AQUALOGQA sistemu je prikazan na slici 3.5.



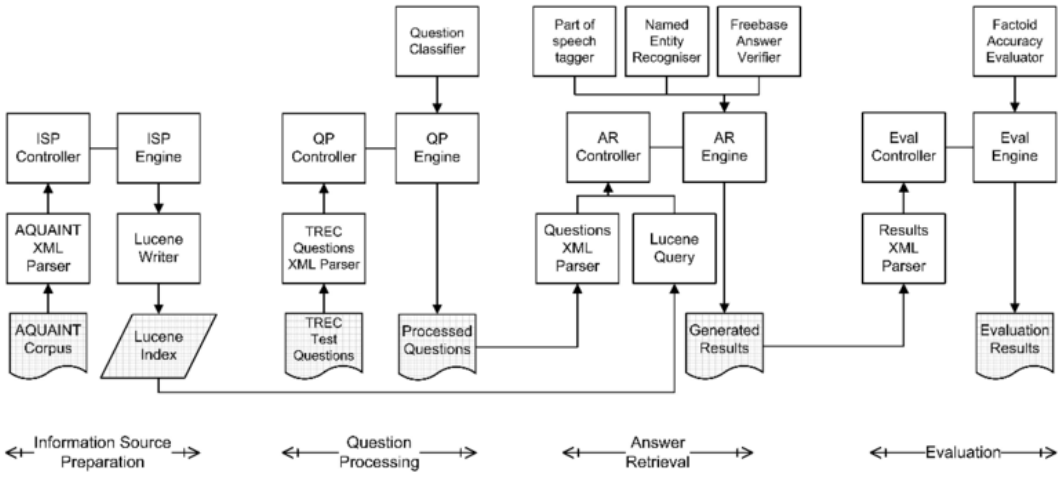
Slika 3.5: Polje za dijalog u AQUALOGQA sistemu

Arhitektura AQUALOG sistema je prikazana na slici 3.6



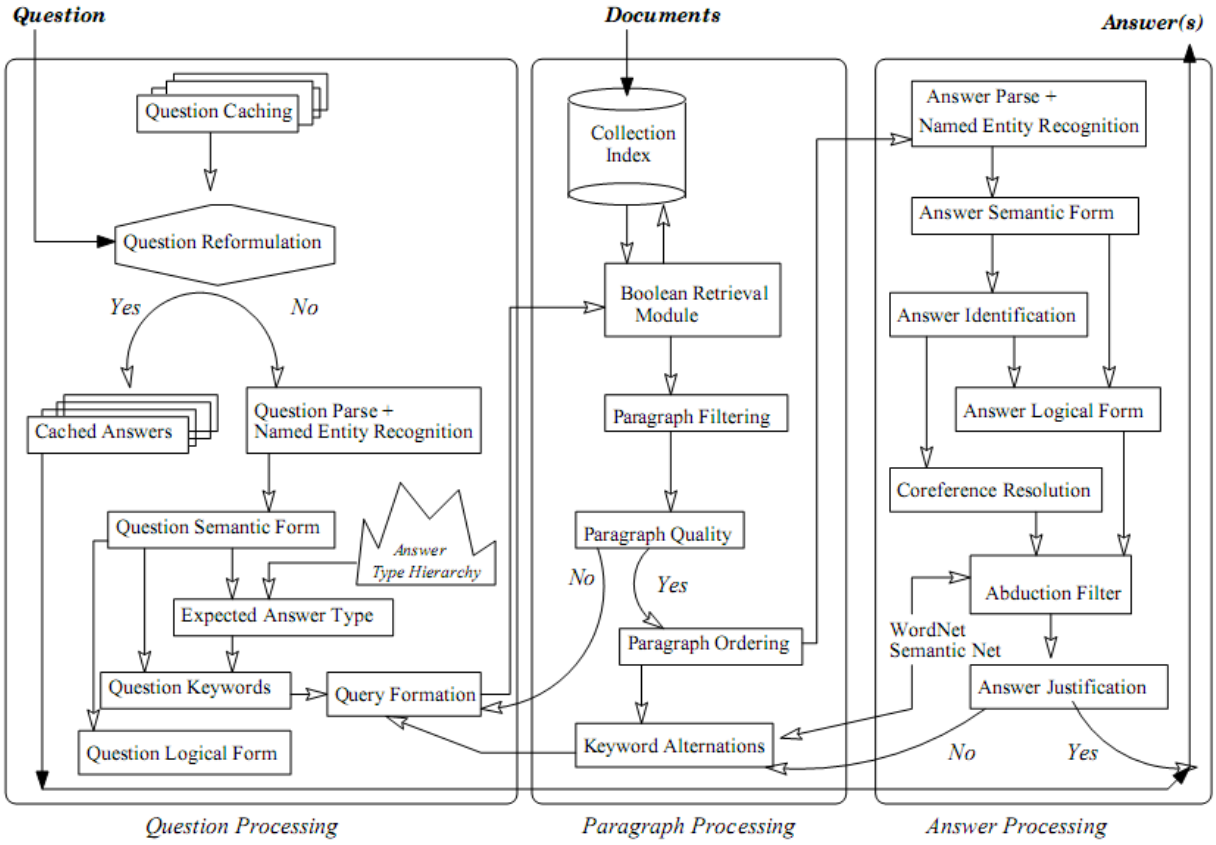
Slika 3.6: Arhitektura AQUALOG [77]

- **QANUS.** Okvir arhitekture QANUS usvaja segmentiran QA pristup, vršeći podelu zadataka kontrole kvaliteta u nekoliko pod zadataka: (1) priprema osnovne informacije, (2) obrada upita, (3) pronalaženje odgovora i (4) evaluacija [78]. Arhitektura QANUS sistema je data na slici 3.7.



Slika 3.7: QANUS sistem [79]

- **FALCON.** Pretraživač fokusiran na otvoreni domen (eng. open domain search engine) se zasniva na dva osnovna stanovišta. Prvo, metodi obrade prirodnog jezika (obrada upita i obrada odlomaka/pasusa) za identifikaciju semantike upita, kako bi se identifikovali eventualni odgovori u okviru kolekcije tekstova. Ove metode su specijalno dizajnirane uz tehnike za pronalaženje informacija sa ciljem preuzimanja svih tekstova iz relevantnih paragrafa. Drugo, da bi se ekstrahovao tačan odgovor (obrada odgovora), pristup velikom broju reči nije dovoljan, pa se koristi metod obrade prirodnog jezika koji je obogaćen pragmatičnim znanjem za filtriranje netačnih odgovora [80] što pomaže u proceni veličine papira i olakšava distribuiranje pre štampanja. Arhitetura FALCON sistema je prikazana na slici 3.8.



Slika 3.8: Arhitetura FALCON sistema [81]

4. Apache Lucene

Za „digitalnu eru" (era u kojoj živimo), je karakteristično da se svakodnevno generišu velike količine informacija i podataka. Najveći deo toga je u tekstualnom obliku: novinski članci, knjige, zvanična dokumenta, internet stranice, elektronska pošta, razni dopisi, istraživačke studije, stručni radovi i slično. Po nekim istraživanjima čak 80% skladištenih informacija se nalazi u tekstualnom obliku. Pored toga, informacije u tekstualnom obliku su najčešće u dokumentima prilagođenim za štampanje [82].

Problem koji se pojavljuje jeste korišćenje informacija koje se nalaze u nestruktuiranim dokumentima. Jedan od načina prevazilaženja ovog problema jeste dubinska analiza teksta. Dubinska analiza teksta se odnosi na pretraživanje potrebnih i netrivijalnih informacija i znanja u nestruktuiranim dokumentima. Postoji veći broj alata koji se bave dubinskom analizom teksta i svi oni se sreću sa problemom obrade prirodnih jezika. Prirodni jezici nisu namenjeni analitičkoj obradi pa ih je s toga potrebno prethodno pripremiti za procesiranje. Jedan od koncepata primene je i NLP koji se može posmatrati kao skup tehnika i metoda za automatsko generisanje tekstova u prirodnom jeziku. Ovaj koncept je primenjiv i podržava mnoge svetske jezike.

Sagledavanje ovih rešenja, široko gledano, se odvija kroz dva procesa: indeksiranje i pretraživanje. U zavisnosti od kvaliteta indeksiranja i pretraživanja pomoću pretraživača zavisi i kvalitet dobijenih informacija. Podaci o dokumentima se stoga čuvaju u indeks fajlu ili bazi podataka, koji se kasnije koriste prilikom pretraživanja kada korisnik izvršava upit. Indeks se može posmatrati kao tradicionalni „back-of-the-book" indeks koji sadrži listu reči, imena ili fraze kojima se ukazuje na materijale koji su na toj poziciji.

Karakteristike indeksiranja i pretraživanja dokumenata pretraživaču obezbeđuju specifične aplikacije koje se nazivaju biblioteke. Jedna od tih aplikacija je Apache *Lucene*. *Apache Lucene* je deo *Apache* fondacije i u skladu sa tim: „*Lucene* je jedna softverska biblioteka za potpuno tekstualno (*eng. full-text*) pretraživanje. To nije aplikacija već tehnologija koja može biti inkorporirana u aplikacijama" [83]. *Lucene* je skalabilna biblioteka za pretraživanje. To je čvrsta osnova na kojoj se mogu razviti aplikacije za pretraživanje. *Lucene* može da analizira i indeksira tekstualni sadržaj, može da obezbedi pretraživanje unutar kreiranih indeksa i prikaže rezultate pretraživanja za određeni upit.

Apache Lucene je specijalizovana biblioteka za složena pretraživanja i sastoji se od veoma kompleksnih funkcija. Ove funkcije se mogu uključiti u razne programske jezike, ali je prevashodno potrebno razumeti njihovu funkcionalnost [82]. Algoritmi i mehanizmi na kojima su funkcije bazirane su veoma složeni i zahteva se poseban pristup u njihovom razumevanju. *Lucene* mehanizam za indeksiranje predstavlja

osnovnu komponentu za indeksiranje tekstualnih dokumenata, dok *Lucene* mehanizam za pretraživanje predstavlja osnovnu komponentu za pretraživanje tekstualnih dokumenata.

4.1 Arhitektura *Apache Lucene*

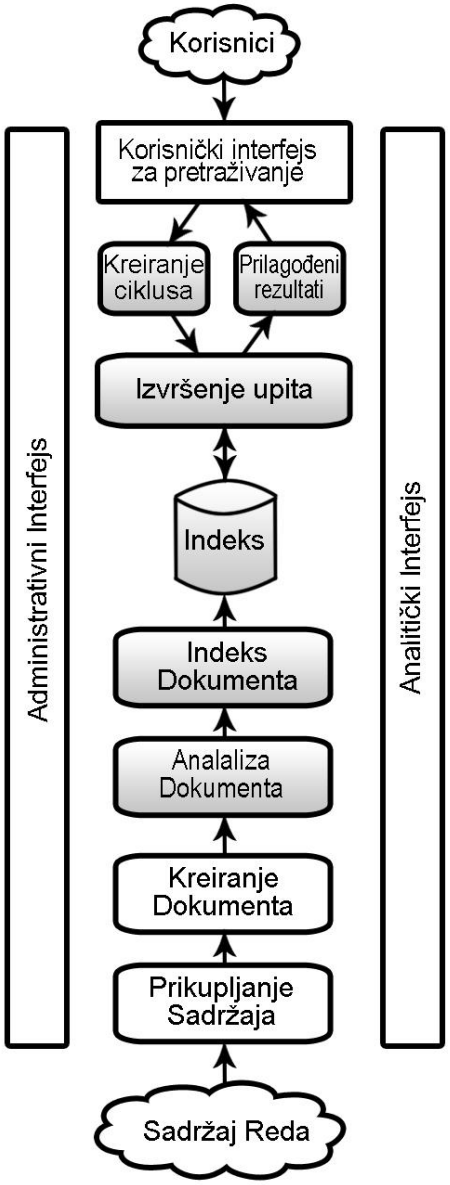
Lucene je visoko performansna i skalabilna IR biblioteka. IR se odnosi na proces traženja dokumenata i informacija u dokumentima ili metapodataka koji opisuju dokumente. *Lucene* omogućava nadogradnju aplikacija sa mogućnostima pretraživanjima. To je zreo projekat (*eng. mature*), slobodan, otvorenog koda implementiran u *Java* okruženju. To je projekat u okviru *Apache Software Foundation*, licenciran kao *Apache Software License*. Kao takav, *Lucene* je trenutno veoma popularna biblioteka koja se često koristi, a i bila je nekoliko godina najpopularnija IR biblioteka otvorenog koda [83].

Aplikacija *Amazon* spadaja u komercijalne aplikacije i ona koristi *Lucene* za indeksiranje i omogućava efikasno pretraživanje. *Lucene* je u stanju da indeksira tekst iz različitih formata kao što su PDF, HTML i *Microsoft Word*, kao i tekst koji je napisan na različitim jezicima [84]. Ključne klase koje se koriste za izgradnju pretraživača su:

- *Document* - *Document* klasa predstavlja klasu za opis dokumenta u *Lucene*. Prvo se indeksira dokument objekat, a kasnije, prilikom pretraživanja, se dobije dokument objekat kao rezultat pretraživanja.
- *Field* - *Field* klasa prdstavlja polje u dokumentu. Field objekat sadrži ime polja i stvarne podatke.
- *Analyzer* - *Analyzer* klasa je jedna apstraktna klasa koja obezbeđuje interfejs koji uzima dokument i pretvara ga u niz tokena koji će biti indeksirani. Postoji nekoliko korisnih implementacije ove klase, a najčešće korišćena je *StandardAnalizer* klasa.
- *IndexWriter* - *IndexWriter* klasa se koristi za kreiranje i održavanje indeksa.
- *IndexSearcher* - *IndexSearcher* klasa se koristi za pretraživanje unutar indeksa.
- *QueryParser* - *QueryParser* klasa se koristi za izgradnju parsera preko kojih se vrši pretraživanje kroz indeks.
- *Query* - *Query* klasa je apstraktna klasa koja sadrži kriterijume pretraživanja *QueryParsera*.
- *Hits* - *Hits* klasa sadrži *Document* objekat koji se vraća prilikom izvršavanja *Query* objekta nad indeksima.

Apache Lucene je softverska biblioteka, skup alata i nije potpuno funkcionalna za aplikacije koje koriste pretraživanje. *Lucene* omogućava dodatne mogućnosti pretrage kako bi se obezbedila puna primena pretraživanja. *Lucene* može da izvrši proces indeksiranja i da obezbedi da bilo koji podatak koji se ekstrahuje iz teksta bude

dostupan za pretraživanje. Da bi se shvatilo kako se tačno *Lucene* implementira u aplikaciju za pretraživanje, uključujući i ono što *Lucene* može, a šta ne može da uradi, neophodnao je razmotriti arhitekturu tipične aplikacije modernog pretraživanja. Glavne komponente aplikacije za pretraživanje su predstavljene na slici 4.1.



Slika 4.1: Glavne komponente aplikacije za pretraživanje

Struktura aplikacije zasnovana na *Apache Lucene* može se sastojati od sledećih komponenti [85]:

- Baza podataka koja sadrži različite vrste dokumenata, na primer PDF, HTML, strane XML dokumenta, običan tekstualni dokument, Word dokument ili druge. Oni mogu biti datoteke u sistemu datoteka ili sadržane u bazi podataka ili generisani od strane određenih aplikacija za vreme pretraživanje veba. *Lucene* očekuje da podaci budu indeksirani i prepoznati kao *Lucene* dokumenti.

- *Lucene* Dokumenti: Lucena biblioteka ne sadrži funkcionalnost za pretvaranje originalnih fajlova u Lucena dokumente i smeštanje u bazu podataka. Aplikacija koristeći *Lucene handler* treba da sprovede dokument na osnovu priloženog *Lucene* dokumenta *handler* interface, kako bi se transformisao sadržaj u *Lucene* dokumenate. Ipak, za ekstrakciju teksta, aplikaciji treba takozvani dokument parser.
- Indeks: *Lucene* dokumenti su analizirani i obrađeni za indeksiranje od strane *IndexWriter*, koji koristi Analizer i jedan ili više filtera, za generisanje stavkog indeksa.
- Implementacija indeks pretraživanja: Aplikacija može pretraživati indeks pružajući *Lucene* zahteve za pretraživanje. Zahtev korisnika se analizira preko *Lucene QueryParser* i formatiran je u *Lucene* jeziku za upite. *Query* parser gradi *Lucene* strukturu podataka upita, što je glavna klauzula koja se realizuje na osnovu zahteva korisnika. *Lucene* upit koji prođe u pretraživanje indeksa postaje *hit* u indeksu. Rezultat pretraživanja mogu se prikazivati na strani korisničke aplikacije.

4.1.1 Koncept Apache Lucene

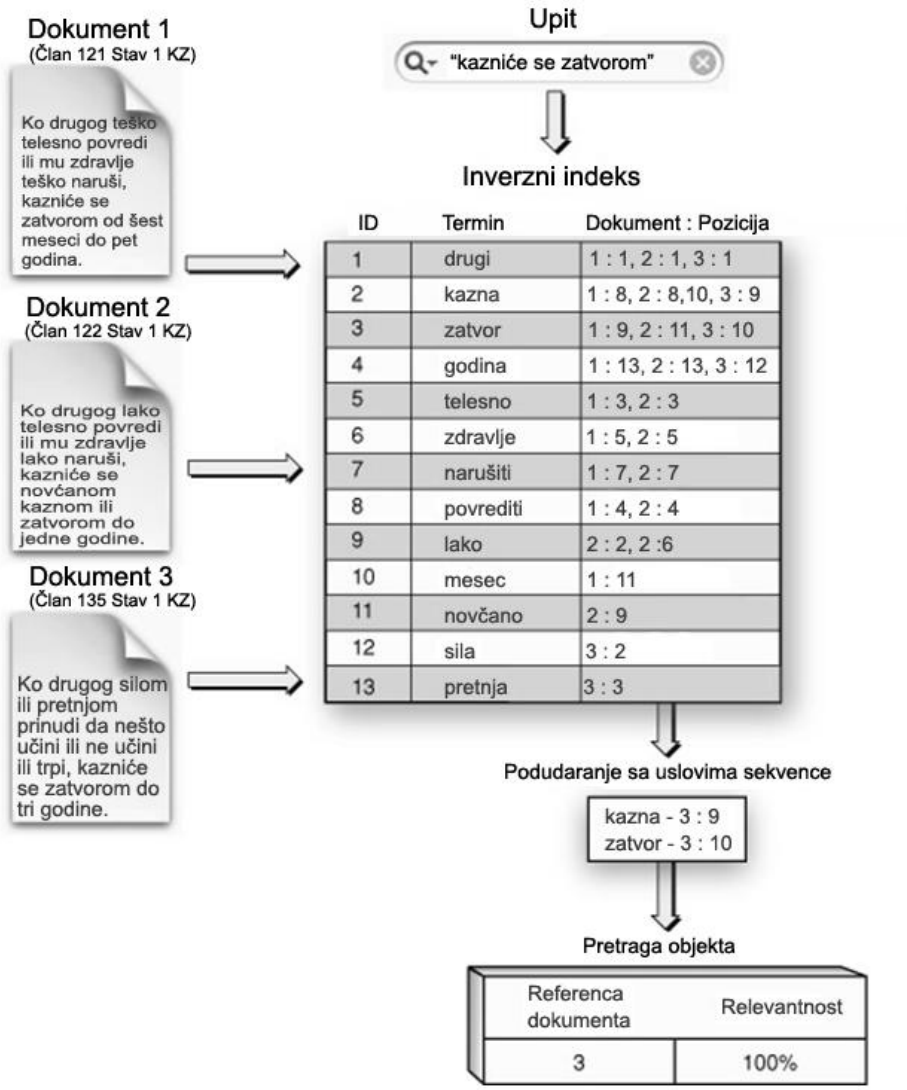
Aplikacije samo koriste Lucenu za indeksiranje i obezbeđivanje efikasne pretrage. *Lucene* je u stanju da indeksira tekst iz različitih formata kao što su PDF, HTML i *Microsoft Word*, kao i u različitim jezicima i kasnije obezbedi proces pretraživanja. Jedan od koncepata *Lucene* prikazan je na slici 4.2.

U poslednje vreme veliki broj aplikacija ima potrebu za naprednim pretraživanjem. Neke od njih rade lokalno nad određenim specifičnim sadržajem podataka i dokumenata. Druge se izvršavaju na udaljenom veb-sajtu sa namenski za to pripremljenom serverskom infrastrukturom, gde krajnji korisnici preko veb-pretraživača ili mobilnog uređaja mogu istovrenemo da pretražuju razne sadržaje koji se nalaze na datim serverima. I u jednom i u drugom slučaju funkcionalnost pretraživanja je, kao softverska komponenta, duboko ugrađeno u kod aplikacije.

Prilikom razvoja aplikacija mora se voditi računa o tome da se samo softverska komponenta za indeksiranje i pretraživanje razvija pomoću funkcija *Lucene* biblioteke. Ostali delovi aplikacija se razvijaju pomoću programskih jezika i tehnologija koje su odabrane za konkretan projekat i oni pozivaju *Lucene* softversku komponentu kada je potrebno indeksiranje i pretraživanje. Pored toga, treba voditi računa o korišćenim veb-pretraživačima zato što moderni veb-pretraživači, a posebno *Google*, imaju prilično veliki skup baznih uslova koji moraju da se uključe u izradi aplikacija kako bi one mogle da se izvršavaju.

Prilikom izrade softverskih komponenti za indeksiranje i pretraživanje koristeći *Lucene* biblioteku neophodno je detaljno poznavanje poslovnih procesa koji se automatizuju i

detaljno poznavanje komponente za indeksiranje i pretraživanje, odnosno funkcionalnosti i mogućnosti *Lucene* biblioteke.



Slika 4.2: Osnovne komponente pretraživača

Prvi deo svih naprednih pretraživanja jeste indeksiranje. Ukoliko je pretraživanje unutar intraneta jedne kompanije onda se indeks za pretraživanje nalazi u kompaniji. Za pretraživanja na internetu koriste se zajednički indeksi koji su unapred pripremljeni za korisnike.

Proces indeksiranja

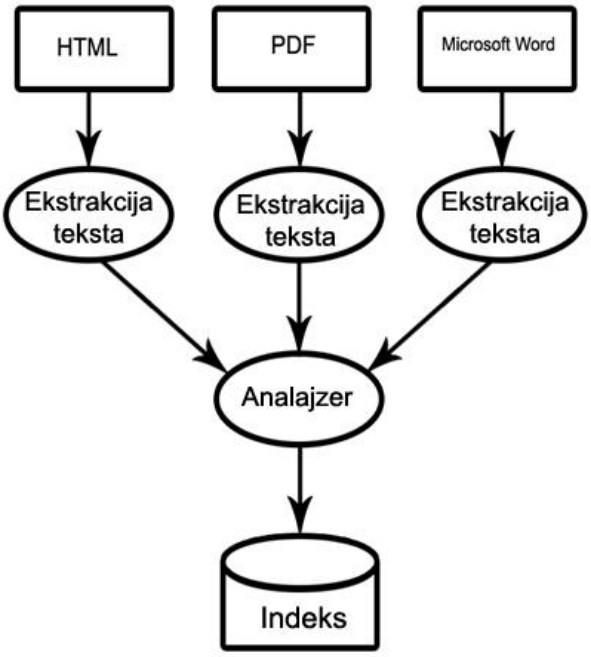
Pretraživanje velike količine nestruktuiranih dokumenata podrazumeva pronalaženje dokumenata koji sadrže određenu reč ili frazu. Sekvencijalni pristup u pretraživanju, pretražuje svaki dokument za datu reč ili frazu, ovde je neprimenjiv, jer zahteva dosta resursa i vremenski dugo traje. Logičan pristup ovom problemu je indeksiranje. Da bi se brzo pretražila velika količina teksta, potrebno je pre samog procesa pretraživanja

formirati indeks koji se odnosi na tekst koji se pretražuje i prevesti indeks u format koji podržava proces pretraživanje. Ovaj proces konverzije se zove indeksiranje, a rezultat koji nastaje po završetku procesa je indeks fajl.

Proces indeksiranja se sastoji od nekoliko postupaka i operacija koje čine *Lucene* metoda indeksiranja [86]. Sve ove operacije su zasebne i diskretne svrstane u tri operativne grupe, koje su prikazane na sledećoj slici 4.3:

- ekstrakcija teksta iz dokumenata,
- analiza,
- dodavanje u indeks.

U suštini svaka od ovih grupa su prilično različite i relativno kompleksne operacije. Prvi korak u indeksiranju je izdvajanje teksta iz sadržaja originalnog dokumenta. Zatim se ekstrahovani tekst koristi za kreiranje dokumenta. Dobijeni dokument se sastoji od polja. Tako dobijena tekst polja se analiziraju i formira se niz tokena. Poslednji korak u indeksiranju tekstualnih dokumenata je kombinovanje tokena sa odgovarajućim indeksima.



Slika 4.3: Indeksiranje pomoću Lucene

Ekstrakcija teksta iz dokumenta

U cilju indeksiranja teksta uz pomoć *Lucene*, prvo se iz običnog teksta izvlači tekst u formatu u kome *Lucene* može da ga obradi, a zatim se kreira *Lucene* dokument. Pretpostavimo da je potrebno u indeks dodati nove indekse dokumenata u PDF formatu. Da bi se kreirali takvi dokumenti u indeksu, prvo se koristi metod za ekstrahovanje informacija u obliku teksta iz dokumenta u PDF formatu, a zatim se ekstrahovani tekst koristi za kreiranje Lucen dokumenata. Isto tako, ako su data XML ili HTML dokumenti koji sadrže obične tekstualne znakove, potrebno je da se na

odgovarajući način pripreme podaci za indeksiranje. Kada se dobije tekst koji bi trebalo da se indeksira i kreira dokument sa poljima, tekst bi trebalo da prođe kroz proces analize.

Analiza

Analiza je pretvaranja tekstualnih podataka u osnovne jedinice koje se zovu tokeni. To je proces pretvaranja sirovog teksta u tokene. *Lucene* to postiže upotrebom Analizera, Tokenizera i token Filter klase. Tokenizer je odgovoran za stvaranje tokena od ulaznih komponenti. Token Filter može dalje da modifikuje tokene koje kreira Tokenizer.

Kada se kreitaju *Lucene* polja u dokumentu pozva se *IndexWriter*. Nakon toga, *Lucene* prvo analizira tekst, a zatim tekstualne podatke podeljene u tokene i onda može da obavlja veliki broj operacija sa njima. Koristeći *Lucene* Filter vrši se potraga za određenim rečima ili skupovima reči koje mogu biti pisane i malim i velikim slovima.

Tokom analize, tekstualni podaci prolaze kroz nekoliko operacija: uklanjanje zajedničkih reči, ignorisanje interpunkcija, svođenje reči na koren reči, pretvaranje reči u reči sa malim slovima i slično. Analiza se odvija neposredno pre indeksiranja i pretraživanja. Analiza pretvara tekstualne podatke u simbole, a ovi simboli se dodaju terminima *Lucene* indeksa.

Lucene biblioteka sadrži različite ugrađene Analizere. Neki od njih, koji se najšećće koriste, su: *SimpleAnalizer*, *StandardAnalizer*, *StopAnalyzer* i *SnowballAnalyzer*. Oni se razlikuju u načinu na koji tretiraju tekst, načinu primene i po tipu filtera koji se koriste. Takva analiza može imati prednosti, uklanjanje pre indeksiranje, smanjivanje veličine indeksa, a to može imati i negativan uticaj na obradu preciznih upita. Primenom *Lucene* je moguće imati veću kontrolu nad analizom procesa koristeći prilagođeni Analizer.

Dodavanje u indeks

Nakon analize unosa teksta, *Lucene* indeks se koriguje. *Lucene* koristi strukturu podataka poznatu kao inverzni indeks. Inverzni indeks koristi i prostor na disku i omogućava brže vreme izvršavanja ukoliko su „podignuti“ ključevi. Njegova struktura je inverzna, jer se tokeni koji se koriste izdvajaju iz ulazne forme dokumenta u oblik „podignutih“ ključeva. Ovaj mehanizam obezbeđuje da se ovaj dokument ne tretira kao centralni entitet. To znači da direktno traži konkretnu reč umesto skeniranja celog dokumenta.

Pretraživanje

Proces pretraživanja je pretraživanje reči u indeksu kako bi se na osnovu toga pronašao dokument u kome se nalazi tražena reč. Kvalitet pretrage se obično opisuje preciznošću i metrikom odziva (*eng. reccal metrics*-RM). RM opisuje koliko dobro sistem za pretragu pronalazi relevantne dokumente, dok preciznost opisuje koliko dobro sistem filtrira irelevantne dokumente. Za *Lucene* postoji poseban okvir za

merenje preciznosti i RM za aplikacije za pretraživanje koji se naziva *Lucene benchmark*. Pored toga, *Lucene* ima još neke faktore koji utiču na pretraživanje: sposobnost za upite sa jednim ili više termina, mogućnost upita sa frazama, korišćenje „džoker“ znakova, *fuzzy* upiti, mogućnost rangiranja rezultata i slično.

Proces pretraživanja se na slici 4.1 logički odvija od vrha na dole. Sa slike se vidi da je prva komponenta: korisnički interfejs za pretraživanje i on predstavlja ono što korisnici stvarno vide u veb-pretaživačima, alatima za pretraživanje na lokalnim mašinama ili mobilnim uređajima. To je ono što korisnici prvo vide kada stupaju u rad, tj. interakciju sa aplikacijom za pretraživanje. Potrebno je da korisnički interfejs za pretraživanje bude vizuelno veoma pregledan i što jednostavniji za upotrebu od strane krajnjih korisnika. Potrebno je da se običan korisnik intuitivno snalazi i da može lako da upiše reči i fraze za koje želi dokumente sa njihovim pojavljivanjem. Isto tako je važan i način kako se prikazuju dokumenti koji su pronađeni i kratak opis iz njihovog sadržaja. *Lucene* ne daje podrazumevani korisnički interfejs za pretraživanje. Prepušteno je programeru da u aplikaciji implementira korisnički interfejs za pretraživanje prema potrebama korisnika. Kada korisnik upiše kriterijume pretraživanja u korisnički interfejs za pretraživanje i da komandu da se izvrši pretraživanje, aplikacija prvo što uradi je da prevede kriterijume pretraživanja u odgovarajući upit za pretraživač.

Upite koje zadaje korisnik mogu biti veoma kompleksni tako da je *Lucene* biblioteka obezbedila veoma snažan paket koji se naziva *QueryParser* koji prevodi tekst korisnika u *Query* objekte koristeći standardnu sintaksu. Zahvaljujući tome upit može sadržati logičke (*eng. boolean*) operacije, fraze pitanja (pitanja pod navodnicima), razna ograničenja za određene korisnike (realizovana preko filtera) i slično. Ukoliko se funkcionalnosti *QueryParser*-a ne uključe u potpunosti, tj. ne obezbede se funkcionalnosti *Lucene* softverskoj komponenti za pretraživanje, onda se upiti moraju prilagođavati kroz aplikaciju, što nije dobro rešenje. Posle kreiranja upita za pretraživač kroz aplikaciju trebalo bi obezbediti njegovo izvršavanje, a to znači obezbediti pretraživanje.

Pretraživanje je proces gde se na osnovu indeksa za pretraživanje pronalaze dokumenti koji se poklapaju sa upitom i sortiraju u odgovarajućem redosledu prema zahtevanom kriterijumu. Ovo je veoma kompleksan unutrašnji proces pretraživača i zahteva se potpuno uključenje mogućnosti *Lucene* kako bi se ovaj proces obavio brzo i precizno. Na ovom mestu je moguće najviše kvalitativno unaprediti pretraživanje implementacijom *Lucene* funkcija.

Postoje tri uobičajena modela pretrage:

- Čist logički model – Za dati upit dokumenti su ili pronađeni ili nisu pronađeni i ne vrši se nikakvo bodovanje. U ovom modelu ne postoji relevantna povezanost između bodovanja dokumenata i samih dokumenata. Upitom se jednostavno identifikuje podskup ukupnog korpusa dokumenata koji odgovaraju upitu.

- Vektorski prostorni model – I upiti i dokumenti se preslikavaju u vektore koji pripadaju prostoru velike dimenzionalnosti, gde je svaki termin pojedinačno predstavlja jednu dimenziju. Relevantnost ili sličnost između upita i dokumenta je mera rastojanja (distance) između dva vektora.
- Probablistički metod – Izračunava verovatnoću da li je dati dokument dobar izbor za dati upit koristeći potpuni probablistički pristup.

Lucene koristi kombinovani pristup vektorskog prostornog modela i čistog logičkog modela. *Lucene* vraća dokumente koje treba pripremiti i prilagoditi (*eng. render*) konkretnim korisnicima.

Prilagođavanje rezultata

Posle izvršavanja upita dobija se „sirov" skup dokumenata koji su ispravno sortirani. Ovaj red dokumenata se mora prilagoditi tako da se pronađena dokumenta predstave korisniku na intuitivan i za korisnika upotrebljiv način. Veoma je važno da korisnički interfejs ponudi jasan put za nastavak pretraživanja ili mogućnost izbora akcije kao što su: prelazak na sledeću stranu, mogućnost korigovanja izvršenog upita, mogućnost pronalaženja dokumenta koji je najsličniji pronađenom i slično, kako korisnik nikada nebi zapao u „ćorsokak". Jezgro *Lucene* ne nudi nikakve komponente koje obezbeđuju potpuno prilagođavanje rezultata korisniku, ali sadrži paket koji proizvodi dinamičke sadržaje i obezbeđuje *hit*.

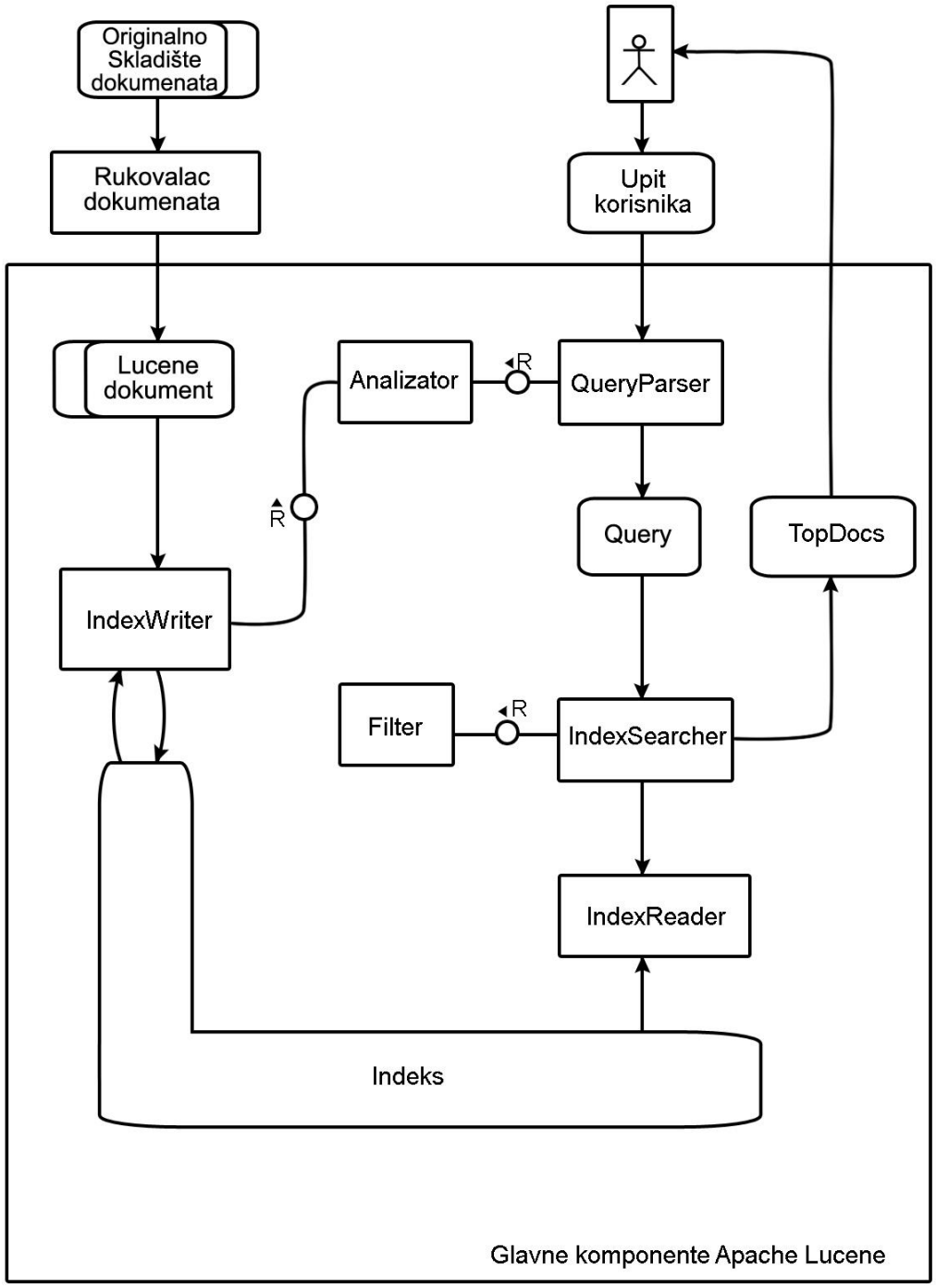
4.1.2 Pregled kompozicione strukture Lucene

Da bi aplikacija koja ima implementirane funkcije *Apach Lucene* mogla da koristi podatke neophodno je da se originalni podaci transformišu u *Lucene* dokumenta. Za tu svrhu se koristi *DocumentHandler* interfejs. On omogućava izdvajanje informacija kao što su tekstualni sadržaj, brojevi i metapodaci iz originalnih dokumenata i formira *Lucene* dokumenata. *Lucene* dokumenti se koriste za dalju obradu u procesu indeksiranja i pretraživanja. Za svaki od uobičajeno korišćenih tipova dokumenata, kao HTML, PDF, XML i tako dalje, potreban je poseban parser dokumenata za ekstrakciju sadržaja. Parseri dokumenata nisu deo *Lucene* jezgra, već se koriste inicijative otvorenog koda koje su u obliku fajlova otvorenog koda. Na primer: JTidi parser za HTML dokumente, Pdfbok za PDF dokumente i Sak za XML dokumente.

Lucene biblioteka sadrži *DocumentHandler* interfejs koji učestvuje u ovom procesu. Svaka klasa koja koristi ovaj interfejs vraća *Lucene* dokumente. To znači da *DocumentHandler* interfejs ekstrahuje sadržaj iz originalnih dokumenata koristeći dokument parser otvorenog koda i nakon toga ekstrahovani sadržaj ubacuje u *Lucene* dokumenta.

Kompoziciona struktura Apache *Lucene* prikazana je na slici 4.4.

Lucene dokument je organizovan kao niz polja. Polje predstavlja par sastavljen od imena i vrednosti. Ime polja je specifično za svaku aplikaciju i određeno je od strane *DocumentHandler-a*. Vrednosti polja su sekvence termina koje su automatski i vrednosti tokena *Lucene* dokumenta. Polja imaju različite atribute i prikazuju kako je izvršen proces indeksiranja.



Slika 4.4: Glavne komponente aplikacije za pretraživanje teksta

IndexWriter analzira i smešta *Lucene* dokumente u indeks u skladu sa atributima polja. Postoje dva osnovna oblika smeštanja dokumenata i to u fajl sistem pomoću *File*

System Directory klase ili u glavnu memoriju pomoću *RAM - Directori* klase. Trebalo bi uzeti u obzir da se novi *Lucene* dokumenti inkrementalno dodaju u indeks, a svaki put kada se to učini, *IndexWriter* pomoću Analizera proizvodi tokene za *Lucene* dokumenta. *IndexWriter* dizajn obrazac može da se definiše kao: *IndexWriter*. On koristi predviđeni Analizer kao strategiju za upisivanje indeksa. Način da se prekine *Lucene* dokument u smislu niza termina, zavisi od izabranog Analizera.

Indeks se sastoji od objavljene (*eng. posting*) liste koja se nalazi i čuva u nestruktuiranom fajlu. Postojanje indeksa je preduslov za proces naprednog pretraživanja. Korisnik bi trebalo da unese čitljiv „*Human-readable Expression called quer y String*“ u aplikaciju baziranoj na *Lucene* pretraživanje, koristeći sintaksu za pisanje ovakvih upita. Ova sintaksa je definisana u okviru jezika za upit. *Query Parser* transformiše korisnički upit napisan *Query String* u objekt tipa *Query*. Postoje različite vrste objekta *Query*, kao što su *WildcardQuery*, *fuzzyQuery*, *booleanQuery* i drugi.

Nakon definisanja *Query* objekta proces analize se nastavlja tako što se prenosi na *IndexSearcher*. *IndexSearcher* pretražuje objavljenu listu prema kriterijumu koji je zadat u *Query* objektu, tj. prema terminu ili terminima koji su zadati. Za pristup indeksu on koristi *IndexReader*, a filter može biti potreban da odredi ograničenja među rezultatima pretrage. Filter obezbeđuje mehanizam za zabranu ili dozvolu da se pronađeni dokument nađe u rezultatima pretrage. Tako, na primer, *spanFilter* označava pojavu neke reči u svim dokumentima, dok *IndexReader* otvara, čita indeks i obezbeđuje poklapanje pronađenog dokumenata u skladu sa terminom ili terminima upita. Na kraju *IndexReader* vraća rezultate pretrage korisniku kao *TopDocs*.

TopDocs je lista koja sadrži brojeve dokumenata, koja se koristi za dobijanje polja smeštenih u indeksu. *Lucene* dokumenti su smešteni u indeksu u formi rečnika termina i objavljene liste. *Lucene* dokument *id* ili broj dokumenta (*docID*) je jedino što se vraća korisniku ili aplikaciji i oni predstavljaju pokazivače (*eng. pointer*) na originalne dokumente. Pored toga, oni mogu sadržati i putanju ili URL adresu gde se nalaze originalni dokumenti.

Na osnovu prikazane povezanosti glavnih komponenti na slici 4.4, može se zaključiti da *Lucene* ne daje celovitost razvoja cele aplikacije za pretraživanje, već samo:

- Obezbeđivanje indeksa iz velike količine različitih tipova podataka.
- Parsiranje korisničkog upita.
- Pronalaženje pojavljivanja termina iz upita u indeksu.
- Računanje statistike na osnovu indeksiranih dokumenata. To obezbeđuje različite informacije, uključujući i to gde se nalaze dokumenti koji sadrže podatke koji se traže upitom, zatim broj koliko puta se oni pojavljuju u dokumentu.

Lucene ne podržava sledeće:

- Upravljanje procesima. Programer mora da odabere određenu komponentu *Lucene* kako bi uradio potrebnu aktivnost u aplikaciji. Tako, na primer ukoliko odabere i koristi *SerbianAnalyzer* u procesu indeksiranja ne može koristiti neki drugi u procesu pretraživanja.
- Izbor dokumenata: Aplikacija, odnosno programer odlučuje o tome koji dokument će biti indeksirani: PDF, XML, HTML ili neki drugi.
- Parsiranje dokumenata: Parsiranje dokumenata nije funkcija *Lucene*. Izvršava se kombinovanjem interfejsa za upravljanje dokumentima sa nekim zajedničkim parserom dokumenata.
- Prikazivanje ekrana za upit korisniku: Ovo se radi na strani korisničke aplikacije.

4.2 *Lucene* dokument

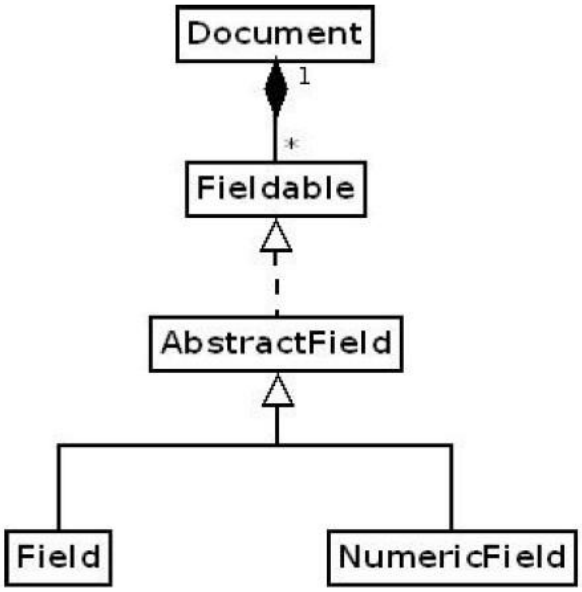
Lucene dokument je *Lucene* logičko predstavljanje ljudima čitljivog sadržaja (teksta) ekstrahovanog iz zajedničkih dokumenata. Ekstrakciju teksta ne vrše funkcije *Lucene*, već parser dokumenata ili interfejs rukovalaca dokumenata koje obezbeđuje aplikacija. *Lucene* podržava ovaj proces pružajući korisniku *DocumentHandler* interfejs za transformaciju tekstualnog sadržaja u *Lucene* dokumente. *Lucene* dokument je instanca *Lucene.document.Document* klase i definisana je kao skup polja. Polje je par ime-vrednost, gde je ime, ime polja i obično se definiše kao konstanta tipa *String*, čija je vrednost jednaka vrednosti polja, tj. sadržaju polja. *Lucene* polje može biti *String*, reč, *Reader* ili *TokenStream* sadrži tokene ekstrahovane iz polja. Funkcionalnosti stvaranja novih polja i njihovo dodavanje u *Lucene* dokument je obezbeđen od strane *Lucene.document.Document* klase. Još jedna funkcionalnost ove klase je da obezbedi korisničke aplikacije sa opcijom polja koja je specificirana i ona određuje kako treba da se rukuje sa *Lucene* dokumentima tokom procesa indeksiranja i pretraživanja.

4.2.1 Package org.apache.lucene.document

Apstraktno gledano, *Lucene* dokument predstavlja dokument koji se sastoji od skupa polja. Na slici 4.5 je predstavljen klas dijagram *Document* klase i njegove komponente. *Lucene* Dokument sadrži jedan ili više *Fieldables*. *Fieldable* interfejs definiše operacije koje se mogu obavljati u polju za upravljanje sadržajem *Lucene Document*. Apstraktna klasa *AbstractField* implementira *Fieldable* metode, koje su zajedničke za *Field* i *NumericField*.

Osnovne operacije iz *Fieldable* interfejsa koje se koriste na poljima za personalizaciju indeksiranja i pretraživanja su:

- Prikazivanje vrednosti *Fields*.
- Promena vrednosti *Fields*.
- Podešavanje indeksnih parametara za *Field*.



Slika 4.5: A Lucene dokument: clas dijagram

Neke od ovih operacija se primenjuju i na *NumericField*. Jedina razlika između *Field* i *NumericField* je u tome što se u prvom koristi tekstualni sadržaj, a u drugom numerički sadržaj.

Korisnička aplikacija počinje proces indeksiranja parsiranjem „sirovih“ dokumenata i kreiranjem *Lucene* dokumenta ekstrahovanjem sadržaja datog dokumenta i organizovanjem tog sadržaja u polja *Lucene* dokumenta. Kada su *Fields* popunjena parom ime-vrednost, onda je dokument spreman za proces indeksiranje. Tokom procesa indeksiranja *Lucene* dokumenti se analiziraju u procesu analiza.

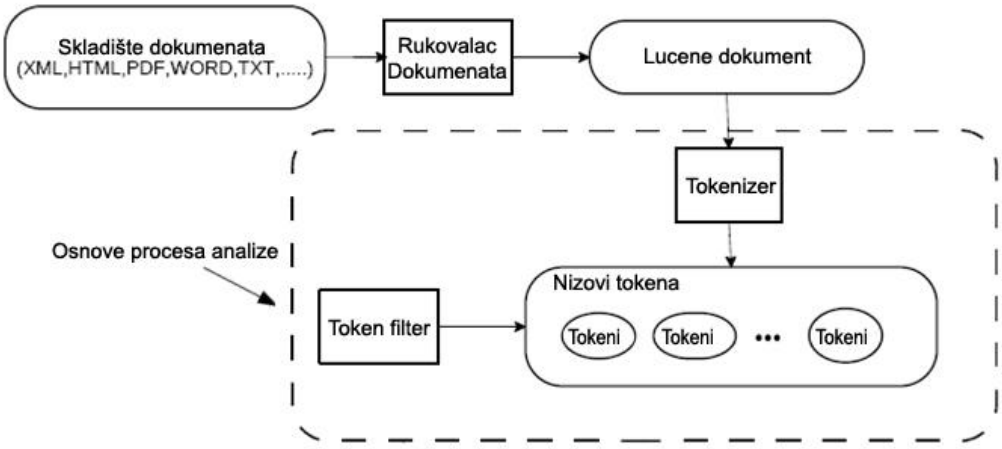
4.3 Lucene *analiza*

Lucene analiza je proces konvertovanja teksta u polju u najosnovniji indeks reprezentacije, tj. termine. Ovi termini se koriste za određivanje šta dokumente povezuje sa upitom tokom procesa pretraživanja. Na primer, kad bi se rečenica indeksirala u polju, u svakom polju bi bio odvojen termin[87].

Proces analize je predstavljen na slici 4.6 i odvija se u nekoliko glavnih koraka:

Korak 1: *Lucene* Dokument se deli na male indeks elemente koji se nazivaju tokenima, pa se ovaj proces naziva tokenizacijom i obavlja se preko Tokenizera. Kada se tokeni formiraju onda se formira niz (*eng. stream*) tokena.

Korak 2: *TokenFilter*, dalje koristi niz token i on filtrira niz token tako što izbacije nepotrebne tokene. Način na koji se dokument analizira zavisi od toga kako je korisnička aplikacija parametrizovana. Na primer, blanko prostori mogu da se uklone iz toka tokena koristeći *WhitespaceAnalyzer* koji su tokenizirani pomoću *WhitespaceTokenizer*, ali ne koristeći *Token Filter*. Suprotno tome, engleske stop-reči se mogu ukloniti pomoću *opAnalizer*, koji koristi *LoverCaseFilter* i *StopFilter* da bi *LetterTokenizer* dao izlazni niz.

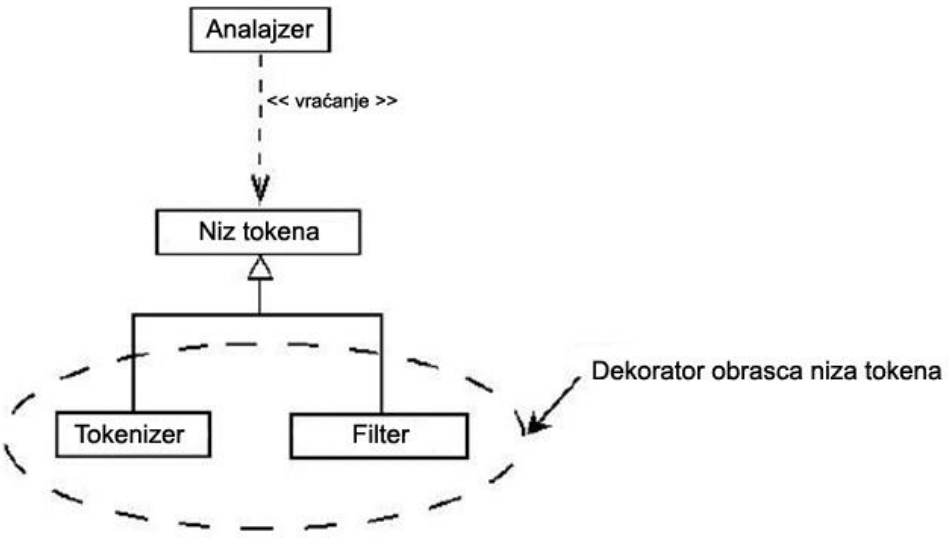


Slika 4.6: Koncept procesa Lucene analize

4.3.1 Package org.apache.lucene.analysis

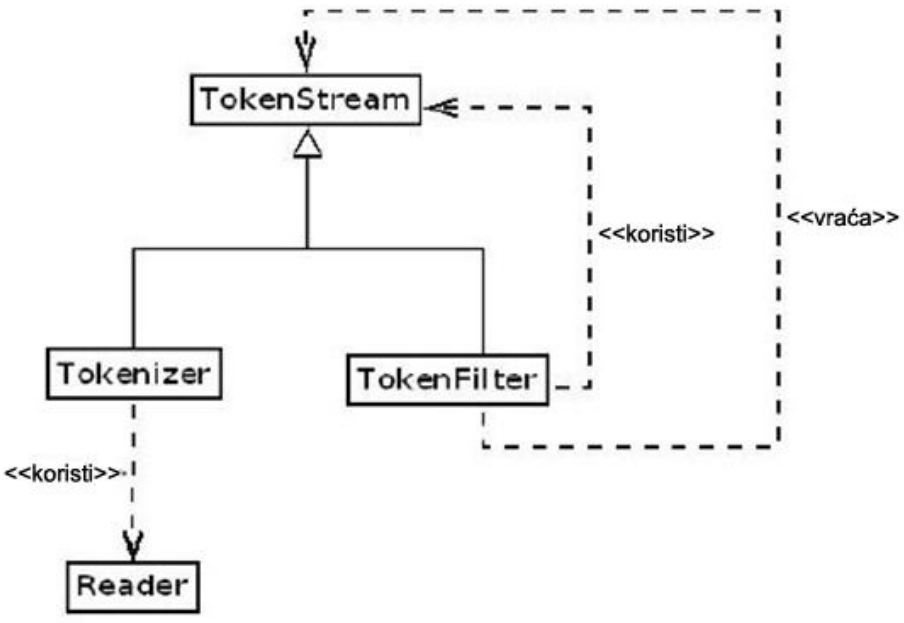
U procesu analize *Lucene* vrši tokenizaciju i filtriranje *Lucene* dokumenata pripremljenih za indeksiranje. Tokom pretraživanja *Lucene* dokumenata analiza upita korisnika vrši se, takođe, preko Analizera i preporučuje se da se koristi isti Analizer koji se koristi u procesu indeksiranja. U *Lucene* jeziku postoje šest osnovnih Analizera: *KeywordAnalyzer*, *PerfieldAnalyzer*, *SimpleAnalyzer*, *StandardAnalyzer*, *StopAnalyzer* i *WhitespaceAnalyzer*. Postoje i dodatni Analizeri koji se uglavnom odnose na određene jezike. Tako na primer, postoje Analizeri za: engleski jezik, francuski jezik, nemački jezik, srpski jezik, i druge jezike [88].

Uloga Analizera je da obezbedi metode za kreiranje niza tokena za *Lucene* dokumenta. Analiziranje *Lucene* dokumenata se realizuje u dva osnovna koraka. Prvo, Analizer stvara Tokenizer koji obezbeđuje čitanje *Lucene* dokumenta i deli polja *Lucene* dokumenata na tokene stvarajući pri tom niz tokena. Niz tokena se zatim propušta kroz *TokenFilter* koji ga čisti od nepotrebnih i nepoželjnih stavki. Na slici 4.7 je prikazana struktura Analizera.



Slika 4.7: Struktura klase Analizer

Dizajn obrasca koristi *TokenStream* koji opisuje obrazac. Apstraktna klasa *Tokenizer* je osnova za Tokenizer i *TokenFilters* koji opisuju *TokenStream*. Tokenizer je *TokenStream* koji „razbija“ ime ili sadržaj polja *Lucene* dokumenata na tokene sa odgovarajućim atributima, koji su bili određeni u procesu indeksiranja. Postoji niz Tokenizera dostupnih u *Lucene*: *CharTokenizer*, *StandardTokenizer*, *LetterTokenizer* i drugi. *TokenFilter* dobija niz tokena iz Tokenizera, filtrira ga i vraća obrađenog *TokenStream*-u. Ovo je mehanizam obrade rezultata jednog Tokenizera od strane *TokenFilter*-a.



Slika 4.8: Dekorator obrazac TokenStream

Upotreba dekorator obrasca u strukturi *TokenStream* je prikazana na slici 4.8.

4.3.2 Analizer

Analizer je enkapsulacija procesa analize. Analizer tokenizuje tekst obavljajući određene aktivnosti, kao što je ekstrakcija reči, odbacivanje interpunkcije, uklanjanje akcenata iz karaktera, normalizacija, uklanjanje uobičajenih reči, korenovanje reči ili promena reči u osnovni oblik (lematizacija). Ovaj proces se naziva tokenizacija, a delovi teksta ekstrahovanih iz niza teksta naziva nizom tokena. Tokeni u kombinaciji sa povezanim imenom polja, su termini. Primarni Analizeri su: V*hitespaceAnalizer*, *SimpleAnalizer*, *StopAnalizer*, *KeivordAnalizer* i *StandardAnalizer*. Oni su dizajnirani da rade sa tekstom na gotovo bilo kom zapadnoevropskom jeziku [87].

Najčešće u je upotrebi *StandardAnalyzer*. On je nešto sofisticiraniji od ostalih. U osnovi je *JFlex-based* gramatika. To je tokenizacija koja koristi sledeće leksičke tipove: alfanumeričke, akronime, nazive kompanija, e-mail adrese, imena računara, brojeve i serijske brojeve, reči sa apostrofom, IP adrese i CJK (*Chinese Japanese Korean*) karaktere.

Pored toga, ovde je uključeno uklanjanje stop-reči i inicijalno je uključena engleska lista osnovnih stop-reči. *StandardAnalyzer* obezbeđuje mogućnost da se uključe i druge stop-reči. Postoje dva načina za to. Prvi način je da se stop-reči definišu u samom kodu aplikacije, a drugi je da se u kodu definiše samo lokacija na kojoj će biti fajl sa stop-rečima. Proširenje funkcionalnosti *StandardAnalyzer* za korišćenje osnovnih stop-reči, kako se najčešće koristi, je prikazano sledećim *Java* kodom:

public final class **StandardAnalyzer**
extends **StopwordAnalyzerBase**

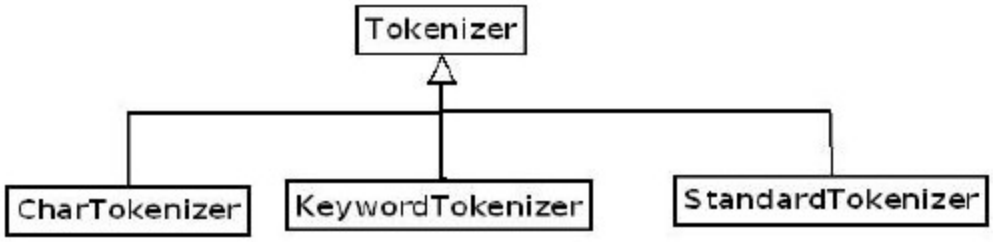
U *Lucene* počev od verzije 5.0 uključen je *SerbianNormalizationFilter*. Ovaj filter normalizuje srpske ćirilične i latinične znakove u osnovne latinične. Ćirilični znakovi se prvo konvertuju u latinične, a onda se latiničnim znakovima uklanjaju dijakritici, sa izuzetkom znaka đ koji se pretvara u dj. Ovde se inicijalno očekuju ulazne reči sa malim slovima.

public class **SerbianNormalizationFilterFactory**
extends **TokenFilterFactory**
implements **MultiTermAwareComponent**
Factory for **SerbianNormalizationFilter**

4.3.3 Tokenizer

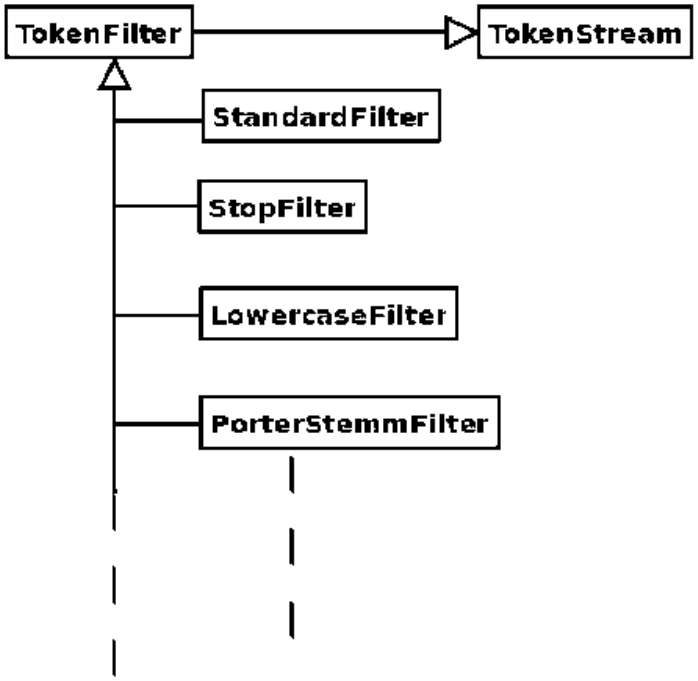
Tokenizer koristi *incrementToken()* metodu za dodavanje tokena u rezultujući *TokenStream*. Tokeni se dodaju jedan za drugim sve dok nema više token atributa dodatih u poslednji token. Na slici 4.9 je prikazan *Lucene* tokenizer koji se sastoji od podklase: *CharTokenizer*, *StandardTokenizer* i *LetterTokenizer*. Oni se i koriste u kombinaciji sa jednim ili više *TokenFilter*. StandardTokenizer tokenizer se najčešće

koristi sa sledećim *TokenAttributes*: *TermAttribute*, *TypeAttribute* i *OffsetAttribute*. *TermAttribute Token* je tekst tokena i on ima definisan početnu i krajnju vrednost. Na disku tokeni se smeštaju u niz.



Slika 4.9: Tokenizer

TokenFilter dobija niz tokena iz Tokenizera, filtrira ga i vraća obrađen niz tokena. Mnoštvo filtera koji se koriste za analizu su dostupni u *Lucene* jezgru kao *Api*. Slika 4.10 daje spisak nekih od raspoloživog *Lucene TokenFilter*.



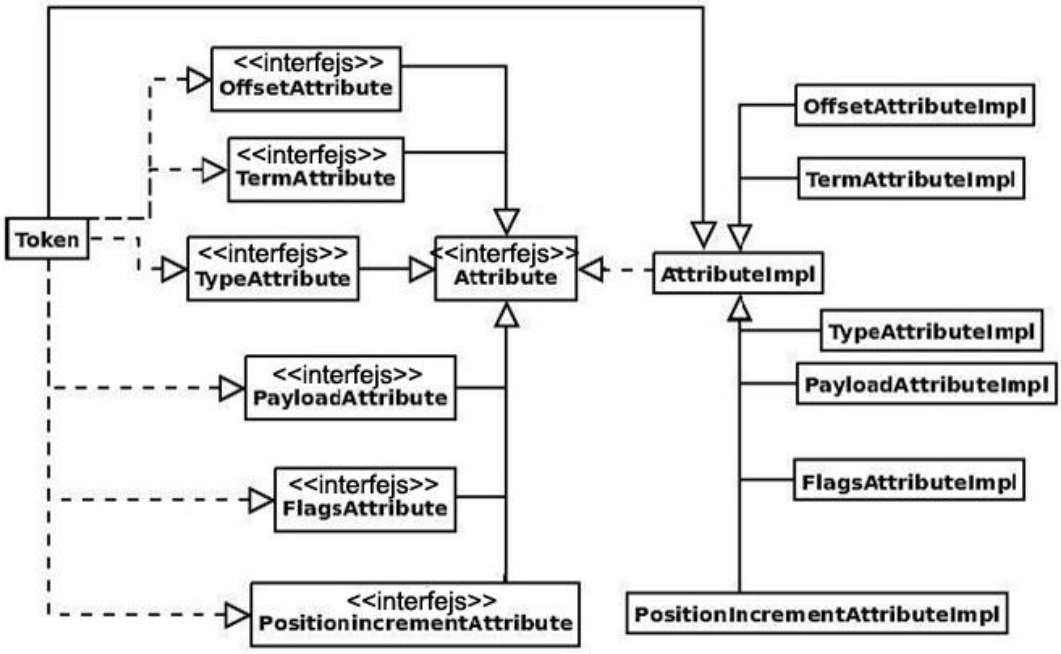
Slika 4.10: Struktura TokenFilter-a

Tokenizer se poziva u skladu sa Api na sledeći način:

Tokenizer (AttributeFactory factory, Reader input)

Ovaj poziv kreira niz tokena procesiranjem ulaznih uslova i korišćenjem date *factory*. Pre procesiranja vrednosti polja svi atributi su postavljeni na inicijalnu vrednost pomoću metode *AttributeSource.clearAttributes()*. Nakon toga se koriste *factory* token atributi. To pokreće mehalizam prikazan na slici 4.11. Koraci koji su uključeni u izgradnju tokena su:

- Tokenizer (ovo je isti za sve *TokenStream*) kreira instancu *factory*. Ovo je objekat tipa *AtributSource.AttributeFactory* ili skraćeno *factory*;
- *Factory* kreira objekte tipa *AttributeImpl*: Ovo se realizuje tako što se uzima ime klase atributa preko atribut interfejsa, a zatim se dodaje *Impl*. Na primer za kreiranje *PayloadAttributeImpl* atribut, *factory* čita ime interfejsa *PayloadAttribute* i dodaje *Impl* na kraju. To nije samo ime koje je preneseno, već metoda specificirana u svaki atribut inerfejs koji su implementirani u odgovarajuću *AttributeImpl* klasu.
- Kasnije, kada se kreiraju atributi, *Token* implementira njihov interfejs i produžava svaku *AttributeImpl* klasu. S obzirom na tako implementiran mehanizam svaki metod u *AttributeImpl* može biti prepisan od strane korisničke aplikacije. Zapravo, *Lucene API* daju mogućnost korišćenja samo onih atributa tokena, koji su potrebni korisničkoj aplikaciji. Iz razloga kompatibilnosti, takođe, moguće je koristiti *Token*-e, koji obezbeđuju pristup svim atributima. To je izbor korisničke aplikacije da koristi *Attributes API* ili *Token* bez da korisnik definiše podešavanje atributa.

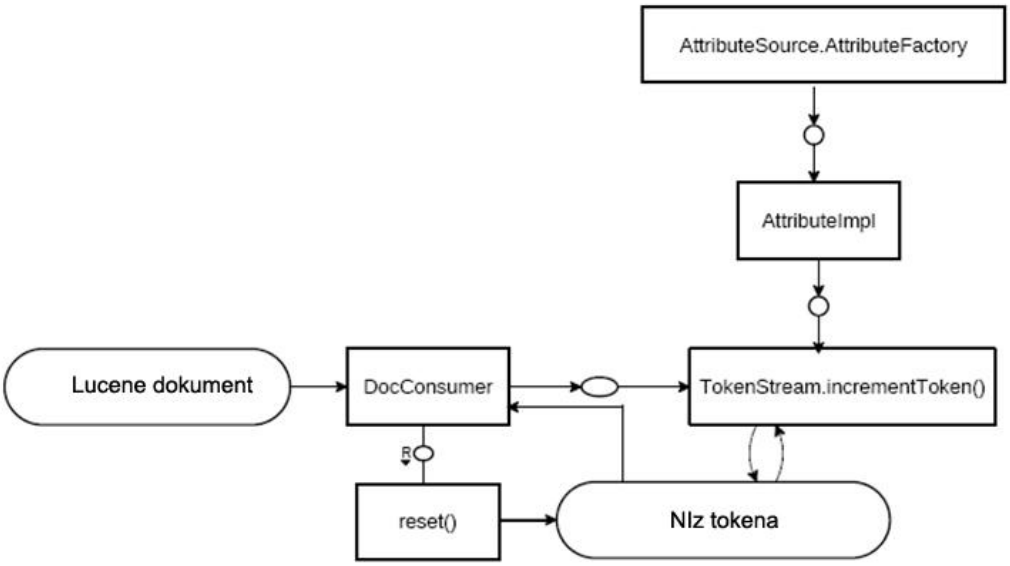


Slika 4.11: Arhitektura factory atributa tokena

Kada su atributi konstruisani, metod *TokenStream.incrementToken()* je pozvan od strane *IndexWriter.DocConsumer* za svaki token. Ovaj metod pomera niz tokena na sledeći token nakon korišćenja atributa prethodnog tokena. Ilustracija na slici 4.12 prikazuje radni tok novog *Lucene TokenStream*-a.

Tokenizer koristi *TokenStream* increment() metodu da bi se pregledali i unapredili svi tokeni u *TokenStream* skladištu, dok nema više atributa za taj token. Ovaj mehanizam Tokenizera je korišćen od strane *DocConsumer*. On koristi svaki token primljen od *TokenStream* skladišta. Tokenizer, takođe, obezbeđuje korisniku atribute za svaki

dodati znak. Atributi su kreirani od strane *AttributeImpl* klase, kao što je već objašnjeno.



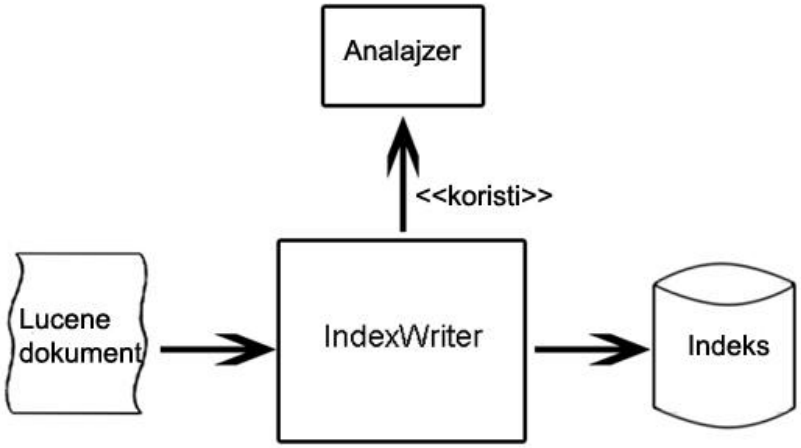
Slika 4.12: Proces tokenizacije (FMC Notation)

4.4 *Lucene* indeksiranje

Indeksiranje je proces koji započinje fundamentalnim jedinicama *Lucene*: dokumenti i polja. Dokument je osnovna jedinica *Lucene* indeksiranja i pretraživanja. Dokument predstavlja kontejner za jedno ili više polja. Svako polje ima ime da bi bilo identifikovano, tekst ili binarnu vrednost i niz detaljnih opcija koje opisuju šta *Lucene* treba da radi sa vrednošću polja, kao i kada se dodaje dokument u indeks. U procesu indeksiranja izvor materijala za pretraživanje je preveden u *Lucene* dokumente i polja.

Indeksiranje u oblasti informacionog pretraživanja i *Lucene* je posebno prikupljanje, analiziranje i čuvanje podataka iz različitih tekstualnih sadržaja kako bi se omogućilo brzo pronalaženje informacija unutar tih podataka [88]. Indeks nije skup dokumenata u kojima se može tražiti, već spisak termina. To su uglavnom ključne reči na osnovu kojih se vrši pretraživanje. Ovi termini se čuvaju kao liste pod nazivom „objavljene liste“.

„Objavljena lista“ je lista *Lucene* identifikacionih brojeva dokumenata (*docID*) tokom dodavanja indeksa u rečnik. Rečnik termina je spisak uslova koji su ekstrahovani iz originalnog i nezavisnog dokumenta. Nakon analize, ti uslovi su sadržani u poljima vrednosti za određeno ime polja. Kada se ta polja analiziraju, *IndexWriter* upisuje za svaki termin polja odgovarajući *docID*. Prema tome, brojevi dokumenata u tom objavljivanju termina se spajaju zajedno u listu koja predstavlja „objavljenu listu“ tog jedinstvenog termina. Koncept indeksa u *Lucene* je prikazan na slici 4.13.



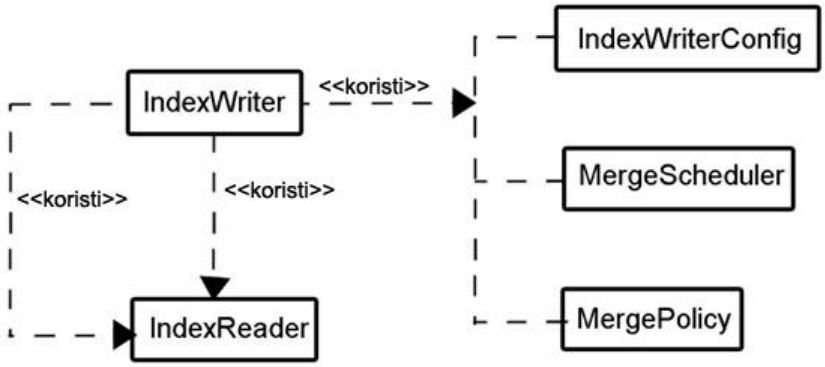
Slika 4.13: Osnovne komponente aplikacije za pretraživanje

Kada se razvijaju aplikacije mora biti kreiran *IndexWriter*, koji koristi *Lucene* dokumente. *IndexWriter* prvo poziva Analizer da sukcesivno prekida polje po polje i filtrira svako *Lucene* polje. Svako polje se analizira i vraća se u indeks putem *IndexWriter*.

4.4.1 Package org.apache.lucene.index.

IndexWriter je odgovoran za kreiranje i održavanje indeksa. On može održavati već postojeći indeks ili napraviti novi. To je u skladu sa *Lucene* osnovnim API.

Na slici 4.14 prikazana je unutrašnja zavisnost indeksa i ilustracija glavne klase paketa *Lucene* indeksa.



Slika 4.14: IndexWriter i interna zavisnost

Kao što je prikazano na slici 4.14, postoje dva slučaja upotrebe:

Slučaj upotrebe 1: *IndexWriter* održava postojeći indeks.

IndexWriter otvara indeks koristeći *IndexReader*. Ukoliko postoji novi ili modifikovani segment spaja se sa indeksom, *MergePolicy* se poziva od strane *IndexWriter* da precizira kako bi se segmenti obrađivali tokom spajanja. *MargeScheduler* upravlja segmentima, kako ne bi postojala konkurencija između dve operacije spajanja. Podrazumevano deset indeksa mogu biti spojena u jedan segment. Ova vrednost može biti modifikovana da bi se poboljšalo indeksiranje ili pretraživanje.

Slučaj upotrebe 2: *IndexWriter* kreira novi indeks.

Potrebno je da IndexReader pristupi indeks direktorijumu. Podrazumevano sve konfiguracije za jedan IndexWriter se čuvaju u IndexWriterConfig objektu. Svrha ovih konfiguracija je da omogući programeru da podesi kako bi trebalo da se pristupi indeksu, kako se termini čuvaju u indeksu i još mnogo toga.

Sam *IndexWriter* koristi za ulazne vrednosti polja *DocConsumer*, koji koristi *incrementToken()* metod *TokenStream* klase, kako bi pripremio niz tokena za sledeći token. Pre toga prvo mora da pretraži atribute niza tokena i da ostavi reference onih kojima želi da pristupi pomoću paketa org.apache.lucene.analysis. *DocConsumer* je jedna od važnih klasa indeksa i ona predstavlja most mehanizma analize i upisa indeksa. *IndexWriter* ga koristi za pristup poljima *Lucene* dokumenata.

DocConsumer pristupa unutar mašine za tokenizaciju i uzima iz polja *Lucene* prvi u nizu dostupan atribut pomoću *TokenStream.incrementToken()* metode. Svaki atribut odgovara delu vrednosti *Lucene* polja. Prikupljanje atributa znači punjenje realne vrednosti atributa u odgovarajući atribut. Vrednost atributa se može modifikovati pomoću korisničke aplikacije:

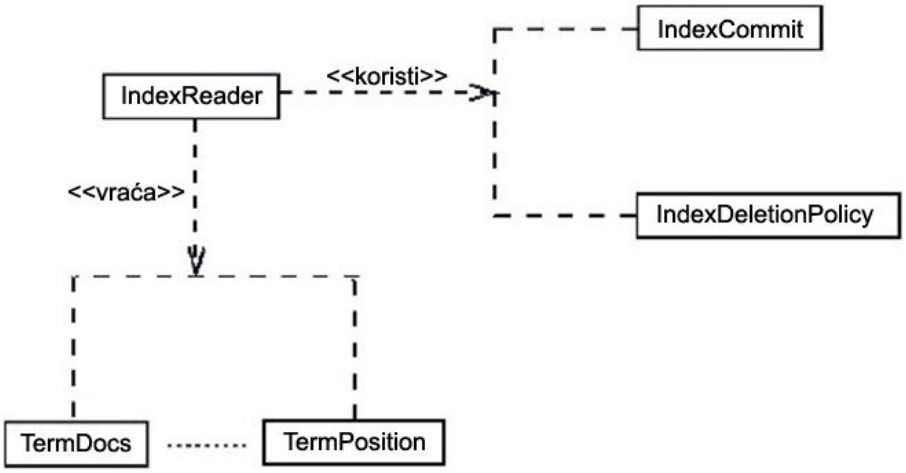
- *TermAttribute* - sadrži tekst tokena i to je reč unutar vrednosti polja;
- *OffsetAttribute* – sadrži početni i krajnji karakter pomeraja tokena;
- *PositionIncrementAttribute* – sadrži broj koji predstavlja poziciju tokena u odnosu na prethodni;
- *TypeAttribute* – je leksički tip tokena,
- *PayloadAttribute* – metapodaci koji se mogu smestiti sa tokenima. Ovo je opciono;
- *FlagsAttribute* – je broj koji može biti postavljen kao zastavica (*eng. fleg*).

Nakon prikupljanja atributa za svaki token *DocConsumer* vraća tokene *IndexWriter* koji ih čuva u indeksu.

Dok *IndexWriter* kreira novi indeks sa *Lucene* dokumentima ili modifikuje postojeće indekse brisanjem polja ili spajanjem indeksa, *IndexReader* čita sadržaj indeksa i vraća informaciju o terminima.

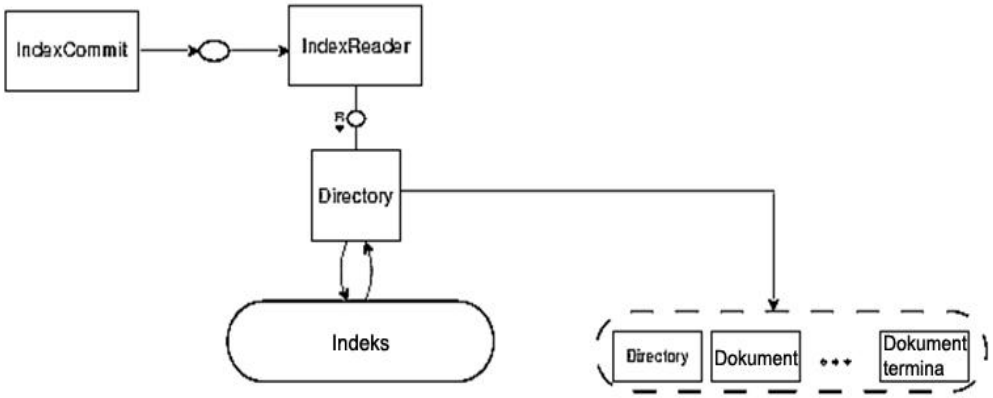
Generalno *IndexReader* obezbeđuje metode za čitanje indeksa. Na slici 4.15 se može videti unutrašnja zavisnost *IndexReader*-a. Klase *IndexCommit* i *IndexDeletionPolicy* se

mogu koristiti za otvaranje indeksa. *IndexCommit* realizuje svaku promenu napravljenu u indeksu uglavnom od *IndexWriter*, prilikom dodavanja ili brisanja segmenta. Operacija *commit* se realizuje kada su segmenti polja smešteni u direktorijum koji sadrže indekse. Kao rezultat operacije *commit IndexReader* pomoću *open()* metode vraća pristup samo za čitanje.



Slika 4.15: Interna zavisnost IndexReader-a

IndexDeletionPolicy osigurava brisanje ranije urađenih indeks *commit* operacija. Stare promene se realizuju pomoću *IndexWriter* i čuvaju se u listi, a prenose se pomoću *Policy* koja ih ujedno i briše. Do brisanja može doći kada se kreira novi *IndexWriter*. To se zove *onInit-deletion* ili može doći svaki put kada *IndexWriter* realizuje operaciju *commit* za novi segment u indeks i naziva se *onCommit-deletion.* Najčešći način da se otvori indeks je pomoću *Directory*. Na slici 4.16 prikazana eksterna zavisnost *IndexReader*. *IndexReader* šalje *Directory* objektu zahtev za pristup indeksu. *Directory* čita indeks i vraća različite vrednosti smeštene u indeksu. Za vreme dok je indeks otvoren za čitanje ili upis neophodno je da *IndexCommit* upravlja promenama koje pravi *IndexWriter*. Za vreme izvršavanja operacije *commit* promene može da napravi *IndexDeletionPolicy*.



Slika 4.16: Pristup indeksu koristeći IndexReader

Kao rezultat vrednosti *IndexReader open()* metode izlazi mogu biti različiti objekti. Sledeći objekti mogu biti uzeti iz indeksa:

- *document (i)* – Upisana polja koja nisu izbrisana iz *Lucene* dokument *id=i* u indeksu. Indeks sadrži objavljenu listu koja sadrži identifikaciju *Lucene* dokumenta, termine, frekvenciju termina i slično. *document ()* metod čita dokument *id* iz objavljene liste i u isto vreme skladiši termine, odnosno polja koja odgovaraju ovom *id*.
- *Document (i, fieldSelector)* – Neobrisani dokument za datu poziciju i sadrži *fieldSelector* za specificirano polje. To znači da kada sva polja ne bi trebalo da se učitaju za pretraživanje, *fieldSelector* pomaže programeru da izaberu one koje treba učitati u *Lucene Document*.
- *Directory()* – Direktorijum u fajl sistemu gde je indeks smešten.
- *Term Frequency Vector* ili lista *Term Frequency Vector* - *Term Frequency* je frekvencija pojavljivanja termina u jednom *Lucene* dokumentu.
- *TermDocs* – Nabraja sve *Lucene* dokumente u kojima se pojavljuje dati termin.
- *numDocs* – broj *Lucene* dokumenata u indeksu koji imaju numeričke vrednosti.
- *numDeletedDocs* – broj obrisanih dokumenata koji su imali numeričku vrednost.
- *TermEnums* – Lista termina smeštenih u indeksu.

4.4.2 Algoritam *Lucene* indeksiranja

Algoritam *Lucene* indeksiranja je realizovan u dva koraka:

a) Osnovni algoritam se sastoji od:
   - Kreiranje indeksa za svaki dokument.
   - Spajanje seta osnovnih indeksa.

b) Inkrementalni algoritam.

Algoritam *Lucene* indeksiranja u skladu sa „Lecture at Pisa university (2004)“ je prikazan na slici 4.17.

```
incremental algorithm:
    maintain a stack of segment indices
    create index for each incoming document
    push new indexes onto the stack
    let b=10 be the merge factor; M=∞
```

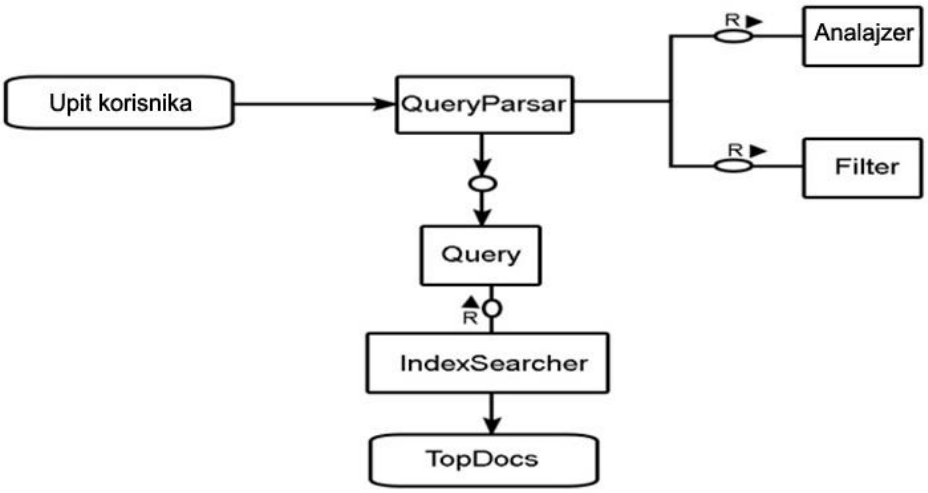
```
for (size = 1; size < M; size *= b) {
   if (there are b indexes with size docs on top of the stack) {
      pop them off the stack;
      merge them into a single index;
      push the merged index onto the stack;
   } else {
      break;
   }
}
```

Slika 4.17: Indeks algoritam Lucene.

4.5 *Lucene* pretraživanje

Lucene obezbeđuje mehanizme i komponente za pretraživanje unutar indeksa i prihvatanje *hit*-a na upit za pretraživanje. *QueryParser* i *IndexSearch* su glavne komponente uključene u osnovne komponente *Lucene* pretraživača. Pretraživanje se može izvršiti tek nakon procesa indeksiranja i kreiranja liste objavljivanja sa terminima - ključne reči po kojima se vrši pretraživanje [89]. Zapravo, pretraživač za korisnikov upit pronalazi indeks. Korisnikov upit koristi Analizer kao u procesu indeksiranja i zatim transformiše korisnikov upit u *Query* objekat u skladu sa *Lucene* jezikom za upite.

Slika 4.18 prikazuje koncept *Lucene* pretraživanja indeksa. Najvažnije komponente koncepta *Lucene* pretraživanja indeksa su *QueryParser* i *IndexSearcher*.

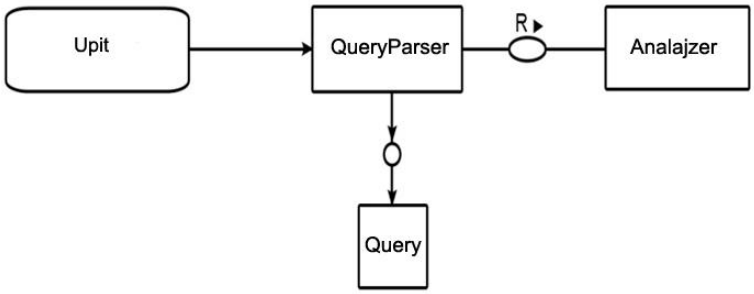


Slika 4.18: Pretraživanje sa korisničkim upitom.

4.5.1 Package org.apache.lucene.QueryParser

Funkcionalnost *QueryParser*-a (Slika 4.19) je definisana u paketu org.apache.lucene.QueryParser. *QueryParser* vrši parsiranje upita korisnika i pretvara ga u *Query* objekat koristeći specifičnu gramatiku za *Lucene* jezik za upite. U procesu transformacije, termini upita su analizirani od strane korisničkog specifičnog Analizera, onog koji je korišćen u procesu indeksiranja.

QueryParser, tj. *QueryParser.java* klasa, procesuira upit uz podršku *Lucene* gramatike za upite. *Lucene* gramatika za upite, kao što je definisano u „*QueryParser.jj*“ klasi, definiše semantičke elemente koje upit treba da ima i interpretaciju tih elemenata. Ova klasa, takođe daje mogućnost sintakse rečenica u upitu. Sa takvom specifikacijom *QueryParser* može da prevede upit korisnika u skladu sa gramatikom definisanom „*QueryParser.jj*“ klasi.



Slika 4.19: Lucene QueryParser

U vremenu izvršenja *QueryParser* može uzeti string upita korisnika i odrediti poklapanje sa gramatikom. Gramatika upita je lista specifikacija koju korisnik definiše pre nego što bude korišćena od strane *Lucene* pretraživača. To znači da metoda parsiranja *parsing()* definisana u dokumentu „*QueryParser.jj*“ je ona koja je odgovorna za parsiranje korisničkog upita u *Lucene* upitu. *Lucene* upit je definisan kao skup rečenica. Rečenica je sastavljena od jednog ili više termina i može biti jedan *Lucene* upit.

Prezentacija upita u *Backus-Naur Form* (BNF) [90] je:

Query::= (Clause)*
Clause::= ["+","-"] [<TERM>":"] ([<TERM>|) "("Query")")

Uloga *QueryParser* je seciranje i prosleđivanje stringa korisničkog upita u *IndexSearch*. Upit korisnika može imati različite forme: jednu reč, rečenicu, veznike, diskjunkcije i slično. *Lucene* definiše gramatiku u zavisnosti od toga da li je upit rečenica ili skup rečenica. Rečenica je skup termina ili simbola (*,?, tilda). Termin je tekstualni deo upita (reč, broj). Uslov između dva termina upita može biti „SHOULD“ ili „MUST“:

- „SHOULD“ - specificira da li termin može biti pronađen (ili ne) u indeksu;

- „MUST“ – znači da termin mora biti obavezno pronađen u indeksu.

QueryParser prevodi:

- Operator AND u upit MUST;
- Operator OR u upit SHOULD;

4.5.2 Package org.apache.lucene.Search

Rezultat *parsing* upita, tj. parsiranja *QueryParser* komponente je rezultat *Lucene Query*. Dalje u procesu pretraživanja, *Lucene Query* se koristi od strane paketa pretraživača za preuzimanje termina i indeksa. Konkretno, to obavlja *IndexSearcher* komponenta (Slika 4.20) koja je ujedno i centralna komponenta. *IndexSearcher* koristi direktorijum ili *IndexReader* za pristup indeksu i prikupljanje informacija iz njega. *IndexSearcher* koristi instancu radnog direktorijuma sa putem koji sadrži indeks i pristup direktorijumu, koji je podrazumevano u režimu samo za čitanje.

Nakon pristupa indeksu, *IndexSearcher* koristi jednu od četiri implementacija *Searcher.search()* metode za pretraživanje *LuceneQuery*. Mogućnosti pretraživanja su predstavljene u strukturi: *Weight*, *Filter*, *Sort*, *Query* i *Term*, kao što se vidi dole:

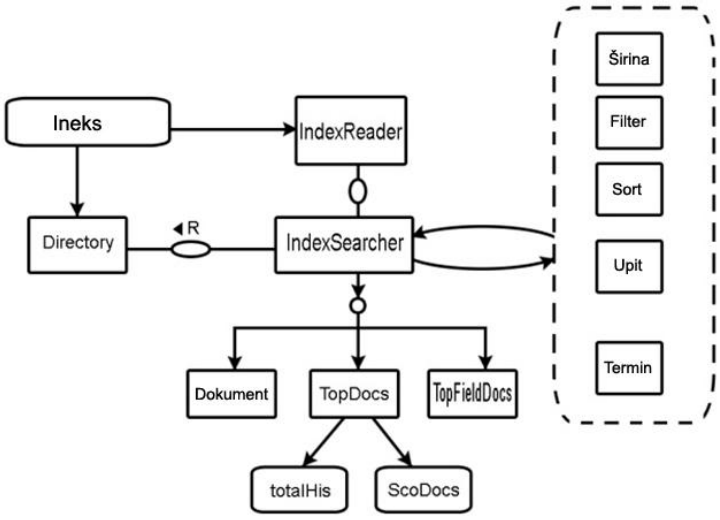
Weight – vraća težinu upita i dokumente kao rezultate pretraživanja;

Filter – koristi se za selekciju dokumenata koji su prikazani u procesu pretraživanja;

Sort – koristi se za prikaz kroz pretrazivanje;

Query – sadrži promenljive strukture zato što mogu biti ponovo napisane;

Term – može biti ekstrahovana iz upita i koristiti se za ponovno vraćanje svih dokumenata pronađenih u indeksu.



Slika 4.20. Lucene IndexSearch.

4.5.3 Algoritam *Lucene* za pretraživanje indeksa

Algoritam za pretraživanje indeksa je opisan u radu koji je napisao Doug Cutting (osnivač *Lucene*) i Jan O. Pedersen [90]. Za dati upit q, *score*-ovi su predstavljeni kao red N koji treba da sadrži rezultat. Napravljeni red je red koji sadrži skup *id* i *score*-ove (scores[*id*]) podudarajućih dokumenata, a *k* je maksimalan broj rezultata koji se prikazuju korisniku. Prikazan je algoritam za pretraživanje indeksa:

```
Inverted_search (query) =
        scores = an array of lengh N initialized to zero
        queue = an emptx quene of (id, score) pairs ordered by
                ascending score
        for (t ϵ q); interate over terms in query
                ps = postings(t); a posting stream for term t
                while (p = nextposting (ps)); iterate over
                postings
                        id = p.id, weight = p.weight
                        scores[id] = scores[id] + qt *
                                weight
                        if  (length(queue)=k + 1)
                                pop(queue)
                        inserte((id,scores[id]), queue)
                end
        end
        pop (queue)
        return (the contents of queue in descending order)
end
```

4.6 LUKE

Luke predstavlja Grafički korisnički interfejs napisan u *Javi* koji omogućava da se pregleda sadržaj *Lucene* indeksa i pojedinačno dokumenti, kao i pokretanje upita preko indeksa.

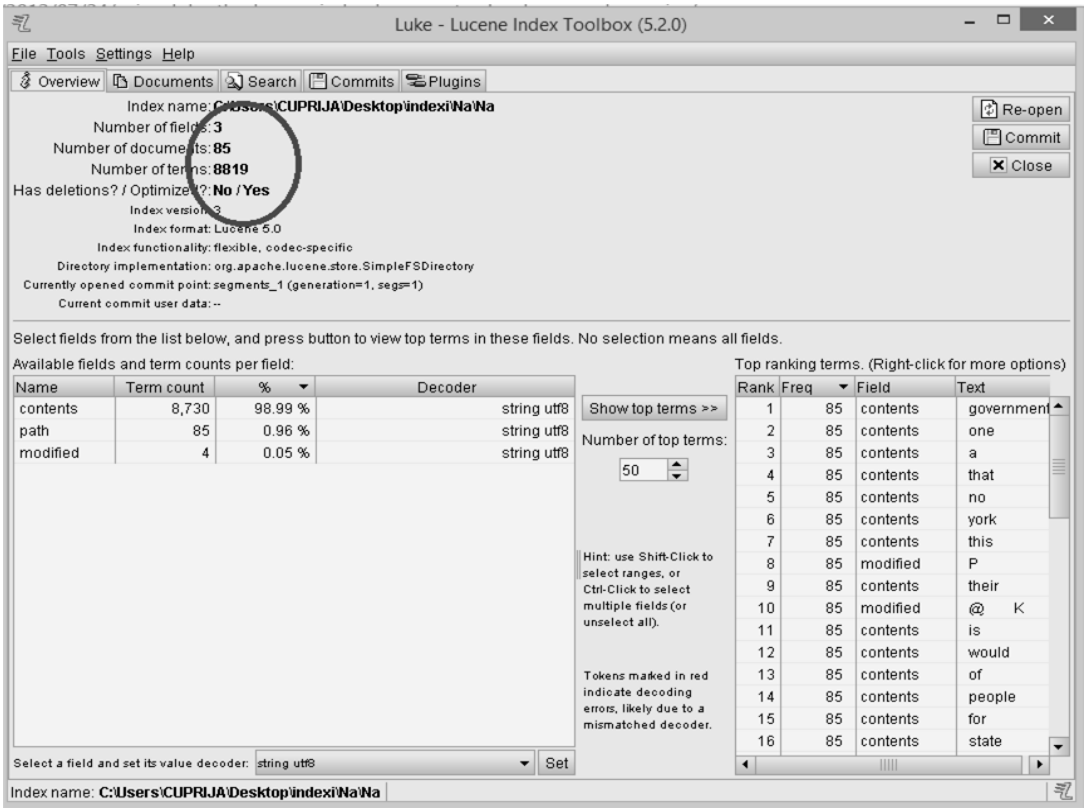
Luke omogućava prikazivanje i izmenu sadržaj na nekoliko načina:

- pretraživanje po broju dokumenata ili termina,
- prikazivanje dokumenta,
- preuzimanje rang liste najčešćih termina,
- pretraživanje i omogućava pregled rezultata,
- analiziranje rezultata pretrage,

- selektivno brisanje dokumenata iz indeksa,
- rekonstrukciju originalnih polja u dokumentu, menja ih i ponovo upisuje u indeks,
- optimizaciju indeksa.

Pretraživanje dokument indeksa

Nakon otvaranja indeksa može se videti na slici 4.21 prikazan GUI Luke. U gornjem levom uglu se nalaze glavne kontrolne funkcije. Zaokruženom linijom na slici su redom označeni: broj polja, broj dokumenta i broj termina. U ovom slučaju to su vrednosti redom: broj polja 3, broj dokumenta 85 i broj termina 8819. Rad u Luki se otvara **sa** *Overview tabom*.

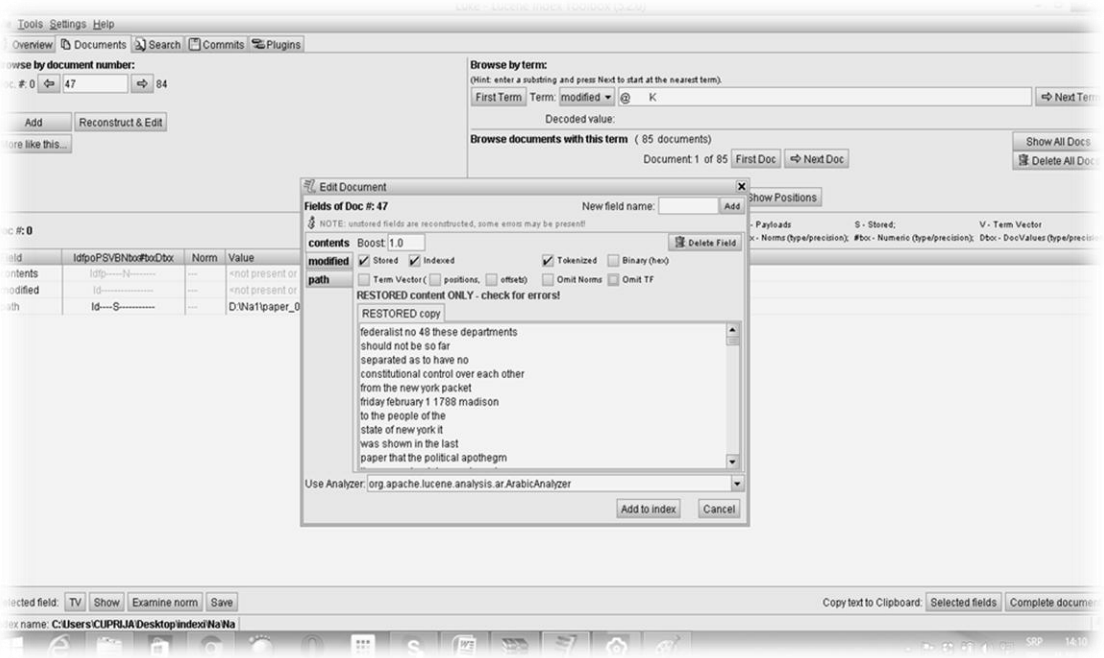


Slika 4.21: GUI Luke

Na tabu *overview*, u donjem desnom uglu, prikazan je rang top 50 termina. Svih 85 dokumenata imaju sličnu strukturu. Termini se ne ponderišu jednako. Termin je ponderisan na osnovu IDF mere, tako da termini koji se javljaju u svega par dokumenata, imaju veću težinu, tj. veći značaj. Manje obimni dokumenti imaju prednost, jer kada dva dokumenta sadrže isti broj slučajeva upita termina, procenat termina za pretraživanje dokumenta manjeg obima je veći i s toga je verovatno i pronalaženje bolje.

Reconstruct & Edit kontrolno dugme, otvara novi prozor koji omogućava da se pregleda sadržaj dokumenta. Na levoj strani vidimo sirov tekst koji je uskladišten. Na desnoj strani vidimo rezultat tokenizacije i indeksiranja preko

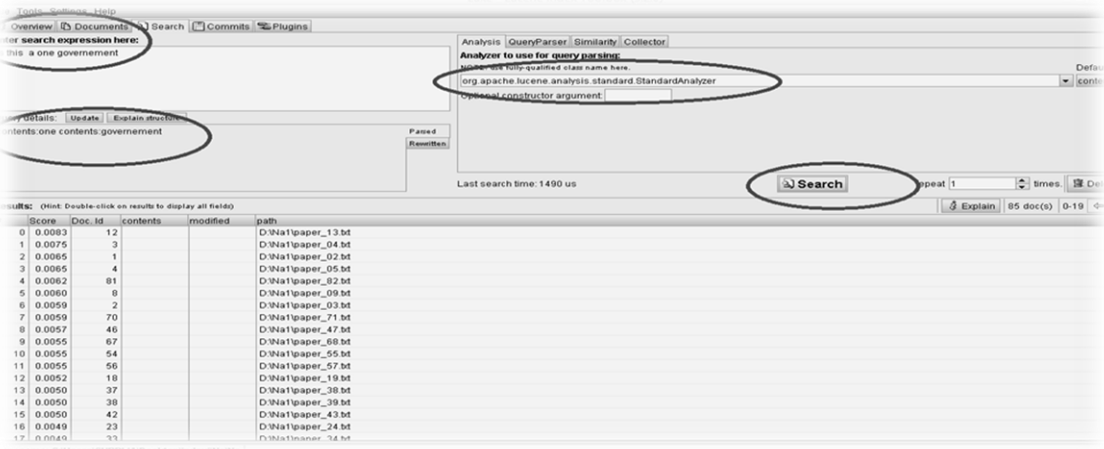
Lucene StandardAnalyzer. *Lucene Standard Analzer* uključuje *StandardTokenizer*, *StandardFilter* i tokenizacija se izvršava prvi put. Uklanjaju se znakovi interpunkcije i dodeljuje svakom tokenu poziciju (slika 4.22).



Slika 4.22: Pregled sadržaja dokumenta

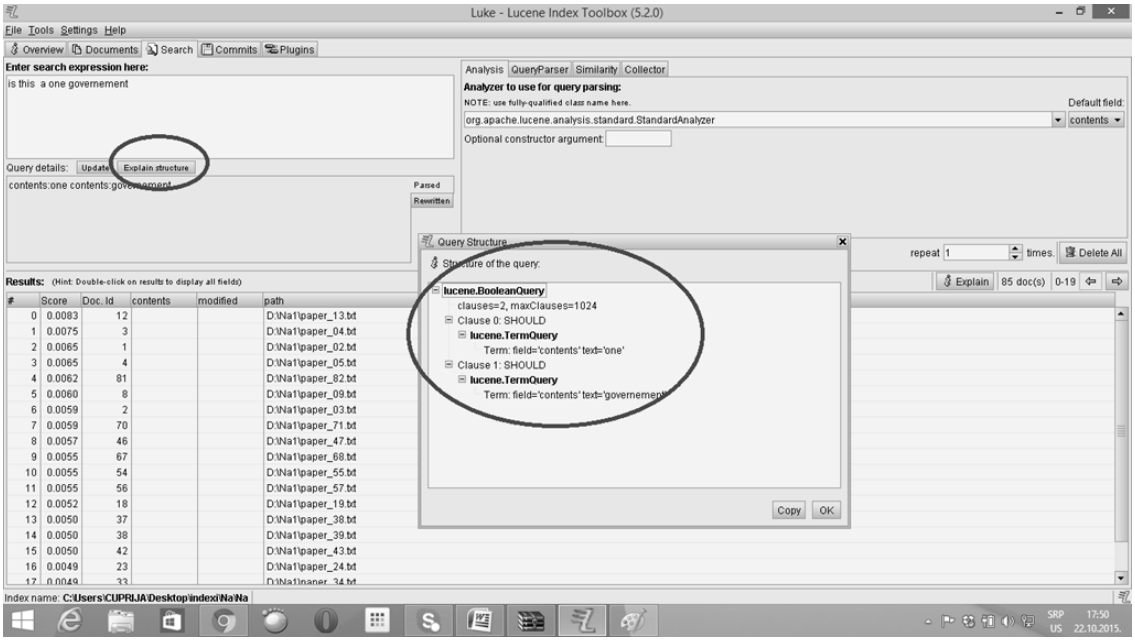
Pretraživanje

Tab *Search* poseduje dosta kontrola. Na slici 4.23 su pokazani rezultati pretrage preko indeksa. U gornjem desnom uglu je ugrađena kartica *Analysis*. Na kartici je izabran *StandardAnalyzer* iz padajućeg menija. Imajući to u vidu, Luka konstruiše *QueryParser* i koristi za razlaganje upita upisanog u okvir za tekst u gornjem levom kvadrantu. Uneta je rečenica: "is this a one governement", u polje predviđeno za unos reči. Odmah ispod ovog polja, nalaze se detalji upita, zaokruženi crvenom bojom. Klikom na *search* pokreće se pretraga indeksa. Donji panel prikazuje rangirane rezultate (Slika 4.23). Rezultat ove pretrage su sva dokumenta koja sadrže tražene reči.



Slika 4.23: Prikaz rangiranih rezultata

Da bi se videla struktura upita koristi se *''explain structure''*. Kada se klikne na ovo dugme, pojavi se novi pop-up prozor, koji prikazuje struktruru upita (slika 4.24).



Slika 4.24: Struktura upita

Izgled strukture upita je sledeći [91]:

```
lucene.BooleanQuery
  clauses=2, maxClauses=1024
  Clause 0: SHOULD
    lucene.TermQuery
      Term: field='contents' text='one'
  Clause 1: SHOULD
    lucene.TermQuery
      Term: field='contents' text='government'
```

5. Modelovanje sistema za dobijanje brzih odgovora za servise e-Uprave Republike Srbije u oblasti Krivičnog zakonika

Obavljajući poslove iz svoje nadležnosti, Vlade raznih zemlja svakodnevno prikupljaju i proizvode velike količine podataka, informacija i dokumenata iz svih oblasti života i rada građana. Oni se gomilaju u IKT sisteme i sa godinama toliko narastu da njihovo korišćenje postaje veliki problem. Podaci, informacije i dokumenti se koriste od strane zaposlenih u Vladinom sektoru, građana i poslovnih korisnika, najčešće kao servisi e-Uprave u tehničkoj realizaciji pomoću veb-servisa. U e-Upravi Republike Srbije postoji znatan broj razvijenih servisa [92]. Međutim, oni se najčešće oslanjaju na evidencije koje su unapred pripremljene za poslovne procese postojećih veb-servisa i nalaze se u bazama podataka [14,15]. Baze podataka predstavljaju sisteme u kojima su podaci organizovani u strukture koje računarski sistemi mogu lako da koriste.

Postoji potreba u e-Upravi Republike Srbije za servisima koji podatke i informacije ekstrahuju iz raznih postojećih tekstualnih dokumenata koji su najčešće u formatu pripremljenom za štampanje [93,94]. Podaci i informacije koji se nalaze u ovakvim dokumentima nisu struktuirani i ne obezbeđuju povezivanje između određenih dokumenata. Generalno, iz ovakvih dokumenata je veoma teško izdvojiti neke forme (znanje) koje postoje u njima. IR predstavlja aktivnost pronalaženja informacija i podataka iz relevantnih kolekcija podataka [95]. Jasno je da sa porastom količine dokumenata i sa prastom potrebe za sistematizovanjem dokumenata i sam IR proces postaje složeniji [96].

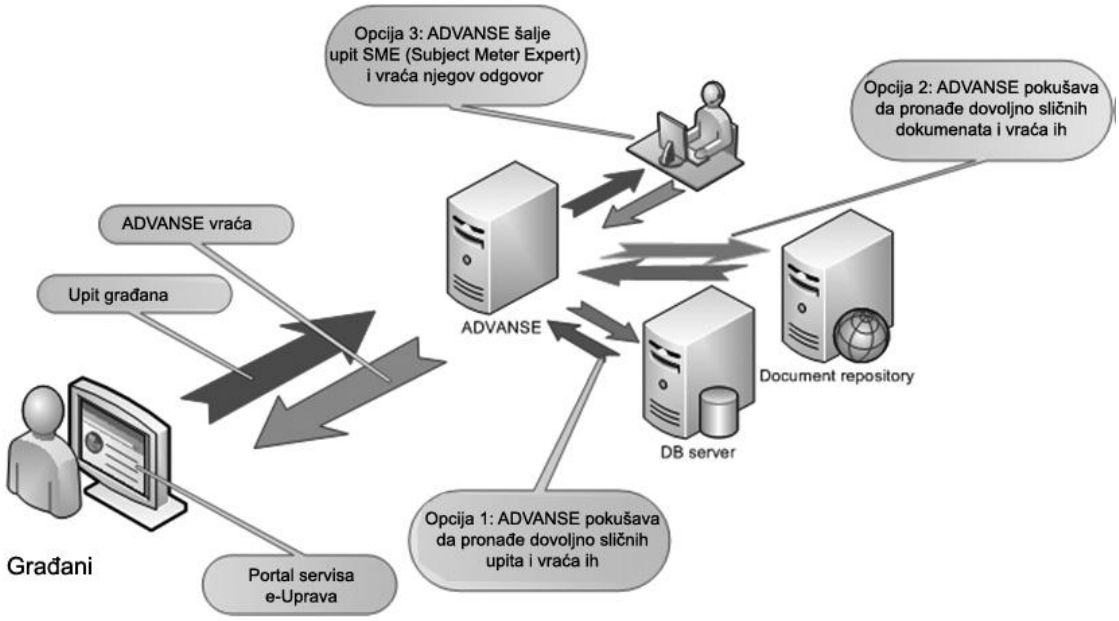
5.1. Princip korišćenja sistema e-Uprava

Interakcija između građana i sistema odvija se u tri faze (slika 5.1). Na početku, sistem nudi grupe ključnih termina (fraze i reči) građanima. Građanin može da izabere jedan (ili ne izabere) termin u skladu sa pitanjem. U sledećem koraku, sistem nudi niz upita koja su usko vezana za odabrane ključne termine. Građanin ima dve opcije u drugoj fazi: da izabere jedan upit sa liste ili da izabere opciju za dalje pretraživanje. Bez obzira na to koja je opcija izabrana, sistem pruža odgovor u sledećoj fazi. Ako je građanin odabrao prvu opciju, sistem nudi odgovor koji se u potpunosti poklapa sa postavljenim upitom. U suprotnom, sistem pronalazi postojeći upit koji je najsličniji postavljenom upitu. Sistem šalje odgovor na upit nazad građaninu. Građanin može oceniti ovaj odgovor. Nakon ova tri koraka interakcije, on može da nastavi ili da završi sesiju sa sistemom. Ako nije zadovoljan odgovorom, građanin može pokušati da pronađe neki drugi postojeći upit ili da unese potpuno novi.

Interakcija između građana i sistema

Slučaj u kome građanin unosi potpuno novi upit predstavlja fokus istraživanja, gde sistem pokušava da pronađe najprikladniji odgovor. Grupe ključnih termina (koji se spominju u prvom koraku interakcije) se automatski generišu u procesu klasterizacije. Upiti i formalna dokumenta se grupišu na osnovu pripadanja klasterima (prema sličnosti termina i reči koje sadrže).

Kada građanin unese novi upit, sistem izračunava sličnost sa upitima u određenom klasteru. Ako je prag sličnosti zadovoljavajući, sistem nudi najbolji postojeći odgovor građaninu. U suprotnom, može da isporuči povezani dokument i/ili obaveštenje. Obično je to poruka sa odgovarajućim obrazloženjem, preporuke ili reference na druge izvore. Građanin može da prati ponuđene korake ili da promeni način interakcije.



Slika 5.1: Interakcija između građana i sistema predstavljena ADVANSE sistemom [97]

Obično postoje tri osnovna komponente u ovakvim sistemima: pitanja građana, Vladina dokumenta i odgovori građanima. One su razdvojene u različitim slojevima. Sloj "odgovori građanima" je između sloja "pitanja građana" i sloja "dokumenti Vlade". Dokumenti Vlade su grupisani po ključnim terminima. Takođe, postoji onoliko klastera koliko i ključnih termina u domenskom rečniku. Pri tome, odgovori se ne grupišu i to zbog dva razloga: oni su isključeni iz procesa pretraživanja i njihov koncept ima dvostruku ulogu. Odgovor može biti ručno kreiran od strane eksperata ili se može automatski generisati u sistemu tako što se pravi veza između pitanja i srodnih dokumenata. Formirane veze mogu biti različitih tipova. Svako pitanje može biti povezano sa jednim ili više odgovora i svaki odgovor može biti povezan sa jednim ili više pitanja.

Isti pristup se primenjuje za povezivanje između odgovora i dokumenata. Upiti i dokumenti mogu da sadrže više od jednog ključnog termina. Ako je sadržaj generalno

veliki, postoje velike šanse da se pronađe adekvatan odgovor na upit. Upit ili dokument onda pripada više nego jednom klasteru. Zato, klasteri mogu biti predstavljeni kao skupovi koji se međusobno presecaju pri čemu upiti i dokumenti koji sadrže više od jedne ključne reči pripadaju ovim presecima.

Značaj odnosa između sadržaja iz različitih slojeva zavisi od potvrde građana o zadovoljavajućem, odnosno nezadovoljavajućem odgovoru. Ova vrednost se izračunava na osnovu dva faktora. Prvi je stepen sličnosti između odgovora i dokumenata, a drugi zavisi od evaluacije odgovora sistema građanima. Stepen sličnosti između odgovora i dokumenata se izračunava od strane sistema i predstavlja objektivnu vrednost (npr. kosinusna sličnost). Stepen sličnosti između upita i odgovora se stalno menja i zavisi od ocene građana. Ovo je neka vrsta evaluacije sadržaja koja zavisi od individualnih očekivanja i stavova građana. Stoga, ova merenja imaju subjektivnu vrednost, ali ona mogu postati objektivna sa većim brojem povratnih informacija od građana. U razvoju e-Uprave, neke zemlje su prevazišle ovaj problem tako što su u servise e-Uprave implementirale algoritme: mašinskog učenja, veštačke inteligencije, ekspertskih sistema, DSS i dr. U tom smislu, mašinsko učenje je disciplina koja se bavi izgradnjom adaptivnih računarskih sistema koji su u stanju da poboljšaju svoje performanse učenjem na osnovu iskustva. Njegova primena u oblasti interpretacije i pretraživanja nestrukturiranih tekstualnih dokumenata pokazala se veoma efikasna.

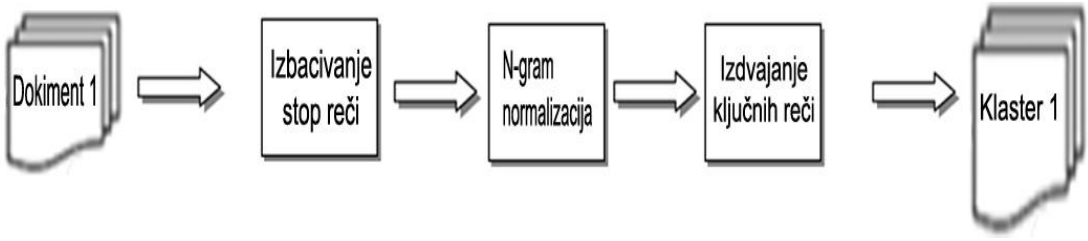
U većini zemalja EU, postoji poseban portal kome građani mogu da pristupe kako bi dobili određene informacije i usluge. Novi trend u mnogim zemljama sveta je uspostavljanje portala koji integrišu veliku količinu informacija i usluga na jednom mestu u skladu sa EIF. Zajednički pristup podrazumeva organizovanje više stavki po temama (naslovima) i/ili određenim grupama. Ti portali uključuju načine pretraživanja u kojima je moguće koristiti sadržaj indeksa drugih Vladinih sajtova.

Savremeni trendovi u razvoju e-Uprave ukazuju na neophodnost primene tehnologije NLP u e-Upravi Republike Srbije [98]. Primena NLP može se posmatrati kao skup tehnika i metoda za automatsko generisanje tekstova na prirodnom jeziku. Ovaj koncept je primenljiv i podržava mnoge jezike. Pored toga, postoji i potreba za rangiranjem dokumenata u IR na srpskom jeziku, gde se izvlače informacije iz nestrukturiranih tekstualnih dokumenata koji se odnose na krivična dela [99].

Kako bi proces pretrage dokumenata sa odgovarajućim sadržajem u velikom skupu podataka bio efikasan, neophodno je dokumente tog skupa grupisati na osnovu sadržaja. Na taj način se formiraju grupe dokumenata sa zajedničkim sadržajem, tj. klasteri. Slika 5.2 prikazuje logički tok formiranja klastera.

Pretraga tekstualnih dokumenata na srpskom jeziku predstavlja kompleksan proces zbog složene gramatike srpskog jezika i zbog upotrebe dva zvanična pisma (ćirilice i latinice). Pored toga, sama morfologija reči srpskog jezika, gde spadaju izgradnja, vrste

i oblici reči, kao i višeznačnost reči i složenost rečeničnih konstrukcija dodatno usložnjavaju procese analize dokumenata. Kako bi se realizovao proces analize dokumenata neophodni su određeni leksički resursi: korpusi srpskog jezika, morfološki rečnik, skup stop-reči, rečnik sinonima, rečnik skraćenica, rečnik pojmova i drugo. Leksički resursi postoje nezavisno od vrste, odnosno od sadržaja dokumenata, pa njihovo pronalaženje, formiranje i održavanje prestavlja dodatni problem. Jezička pravila i odstupanja od njih zahtevaju znatan skup leksičkih resursa kako bi analiza tekstualnih dokumenata bila prihvatljiva.



Slika 5.2: Logički tok formiranja klastera

Probleme u vezi sa leksičkim resursima za srpski jezik moguće je prevazići primenom N-gram analize tekstualnih dokumenata pri čemu semože doći do prihvatljivih rezultata.

5.1.1. Teorijske osnove za razvoj sistema brzih odgovora

U oblasti matematike, statistike, empirijskih nauka, informatike, matematičkom optimizacijom sagledava se izbor najboljeg elementa (u odnosu na neke kriterijume) iz skupa raspoloživih mogućnosti [100]. Optimizacija podrazumeva pronalaženje minimuma i maksimuma realnih funkcija kojima se sistematski menjaju ulazne vrednosti u okviru prethodno definisanog skupa i izračunavaju vrednosti koje se odnose na funkcije. Uopšteno, optimizacija uključuje pronalaženje ekstremnih vrednosti nekih objektivnih funkcija datih definisanim domenom.

Najrasprostranjenije TM tehnike [101] su detaljno analizirane kako bi se omogućilo bolje razumevanje njihove primene u oblasti e-Uprave, učešća građana i e-demokratije [102]. TM u QA aplikaciji bavi se pronalaženjem najboljeg odgovora na dati upit preko servisa e-Uprave, odnosno davanjem odgovora od strane QA. Sama aplikacija može da koristi više od jedne tehnike TM. Tipične TM tehnike uključuju kategorizaciju teksta, klasterovanje teksta, ekstrakciju koncepta entiteta, sumarizaciju dokumenta, i sl. Date tehnike pomažu da se bolje realizuje proces izdvajanja osnovnih pojmova, obrazaca i relacija iz velike količine tekstualnih kolekcija podataka.

Kategorizacija teksta ili klasifikacija tekstualnim dokumenata je zadatak dodeljivanja unapred definisane kategorije ovim dokumentima. Ovo može da omogući konceptualni pogled na kolekcije dokumenata i ima veliku praktičnu primenu [103]. Većina postojećih pristupa za klasifikaciju teksta predstavljaja tekstove kao vektore reči, „vreće reči“ (Bag of Words – BoW). Pristup BoW obično uspostavlja niz značajnih reči i rečenica na osnovu faktora statističke analize kao što su učestalost pojavljivanja

termina i distribucija. Pri tome dokument se posmatra kao skup reči bez obzira na redosled reči i gramatiku.

Da bi se dobili efiksni rezultati pretraživanja potrebno je grupisati dokumente, u zavisnosti od sadržaja dokumenata, po određenom kriterijumu. Dokumenti se grupišu prema ključnim rečima. Ključne reči predstavljaju skup reči koje su glavni nosioci značenja za određenu grupu dokumenata. Izdvajanje ključnih reči je veoma složen i zahtevan proces i zato je važno ukloniti nepotrebne reči i delove reči iz svih tekstualnih dokumenata i to realizovati pre faze analize dokumenata. U fazi analize tekstualnih dokumenata potrebno je da se, među ključnim rečima, svaka reč tretira bez glagolskih oblika i padeža kao jedna reč.

Sličnost između dve reči se određuje na osnovu semantičkog značenja reči. Nosilac semantičkog značenja reči je koren reči. Otežavajuća okolnost za identifikaciju reči su reči koje imaju zajednički koren, jer se u srpskom jeziku pojavljuju složene izvedene reči iz istog korena zbog postojanja prefiksa, infiksa, sufiksa i slično. Iz ovog razloga uvodi se još jedna faza koja se zove normalizacija teksta. Pod normalizacijom teksta se podrazumeva transformacija teksta iz postojećeg oblika u oblik pogodan za računarsku obradu podataka. Svrha normalizacije jeste svođenje reči na osnovni oblik, tako što se date reči oslobađaju promena. Razlog uvođenja normalizacije u ovom slučaju je priprema teksta za pretraživanje.

N-gram analiza određuje frekvenciju različitih N-grama u tekstu. N-grami tekstova se intenzivno koriste u TM i NLP. Mera N-gram preklapanja je preklapanje sekvenci reči između rečenice kandidovanog dokumenta i rečenice upita. Rešenje bazirano na N-gram analizi za rečenicu *P* koristi sledeću formulu [104]:

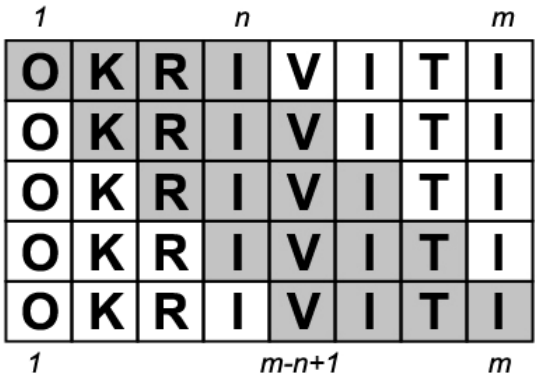
$$NgramScore(P) = max_i(max_j\, Ngram(s_i, q_j)) \quad (2)$$

$$Ngram(S, Q) = \frac{\sum gram_n \in S^{Count_{match}(gram_n)}}{\sum gram_n \in S^{Count(gram_n)}} \quad (3)$$

gde *n* predstavlja dužinu N-grama (n = 1 , 2 , 3 , 4), $Count_{match}$ $(gram_n)$ je broj N-grama za prateći upit i rečenicu kandidata, q_j je *j*-ta rečenica u skupu upita, a s_i je *i*-ta rečenica u skupu rečenica od ukupno *P rečenica*. U principu, N-gram predstavlja podsekvencu (podniz) od *n* vrednosti datog niza. N-gram vrednosti 1, 2 ili 3 se nazivaju uni-gram, bi-gram i tri-gram respektivno.

U srvari, N-gram analiza je skup aktivnosti koje se primenjuju na tekstu kako bi se dobio skup N-grama određene dižine. U ovom postupku prvo se definiše dužina *n* koja može imati vrednosti od *1* do *m-n+1*, gde je *m* dužina reči koja se normalizuje i predstavlja jedan string. Pomeranjem okvira dužine *n* dobija se N-gram.

N-gram analiza se veom uspešno može koristiti za pronalaženje reči koje imaju isti koren. Na slici 5.3 je predstavljena N-gram analiza za reč „okriviti“ gde je n=4. Računarskom obradom reči „krivični“, „okrivljeni“ i „kriva“ koje su naizgled različite imaju zajednički 4-gram „kriv“. Ove tri reči su semantički vezane za imenicu „krivica“ iz koje su nastale.



Slika 5.3: Normalizacija 4-gram

U cilju dobijanja efikasnijih rezultata prilikom izbora broja *n*, sakupljeni su nastavci za oblike reči u srpskom jeziku za reči od važnosti za analizu [105]. Tom prilikom utvrđeno je da najveći procenat (92,70 %) nastavaka ima ispod četiri slova [106]. Takođe, u opširnoj studiji u radu autora Marovca i ostalih [106], korišćena je 4-gram analiza, jer daje najbolje rezultate za srpski jezik. Zato, u ovoj doktorskoj disertaciji za dalja istrazivanja biće vršena 4-gram analiza kao referentnu tehniku za normalizaciju reči u dokumentima na srpskom jeziku.

Slika 5.4: Aplikacija VebRanka [107]

Sa druge strane, jezički resursi za srpski jezik su razvijeni od strane Tehnološke grupe na Matematičkom fakultetu u Beogradu koja se bavi kako informacija može biti

sadržana u bogatim jezičkim resursima, i kao takva, kako se može efikasno koristiti za rešavanje problema tekstualnih dokumenata na srpskom jeziku. Kao rezultat svoga rada, ova grupa je kreirala jedinstven i lako upotrebljiv alat za jezičke resurse, označenen kao *LeXimir*, što u velikoj meri povećava potencijal za izradu svakog drugog pojedinačnog resursa, kao i više resursa istovremeno. Dati alat se već uspešno koristi za različite zadatke u procesu obrade jezika. Deo *LeXimir* sistema je veb-aplikacija VebRanka – radna stanica za upite sa pratećim veb-servisima, u cilju upravljanja ovim složenim zadatkom na internetu (slika 5.4).

Da bi se analizirao sadržaj bilo kog dokumenta, od ponuđenih opcija u aplikaciji, bira se „Bag of Words“. Inače, ovo je alat koji služi za analizu dokumenta na srpskom jeziku, bilo da je dokument pisan na ćirilici ili latinici, pa tako u opciji select language postoji samo srpski jezik: serbian – latin i serbian - cyrilic. Kopira se tekst u text-box-u i u zavisnosti od toga šta je uključeno u pretragu, to se i čekira. Uglavnom su to pridevi (eng. a) , skraćenice (eng. abb), prilozi (eng. adv), lična imena (eng. n), brojevi (eng. num) i glagoli(eng. v). Rezulat analize može da se preuzme u Excel ili XML formatu. Jedan od tekstova sa interneta koji je obrađen u ovom alatu, dobija rezultat kao što je dat u tabeli 5.1. U ovim rezultatima imamo mogućnost da vidimo oblik u kome se reč pojavljuje u tekstu, lemu ili osnovni oblik reči, vrstu reči i njenu frekvenciju.

Oblik	Lema	Vrsta	Frekvencija
u	u	ABB	8
je	jesam	V	5
su	jesam	V	3
pi	piti	V	3
mestu	mesto	N	2
da	dati	V	2
pucao	pucati	V	2
oduzmu	oduzeti	V	1
prostrelnu	prostrelni	A	1
ramena	rameni	A	1
ramena	rame	N	1
levog	levi	A	1
ranu	ran	A	1
ranu	rana	N	1
predelu	predeo	N	1
suprugu	supruga	N	1
suprugu	suprug	N	1
radnom	radan	A	1

radnom	radni	A	1
naneo	naneti	V	1
dok	dok	N	1
bila	biti	V	1
bila	bilo	N	1
uperio	uperiti	V	1
ali	ala	N	1
tu	tu	ADV	1
zadesili	zadesiti	V	1
urlao	urlati	V	1
kancelariju	kancelarija	N	1
ubi	ubiti	V	1
saznaje	saznavati	V	1
Niko	Nika	N	1
Niko	Niko	N	1
nije	jesam	V	1
policijskoj	policijski	A	1
dogodila	dogoditi	V	1
stanici	stanica	N	1
dva	dva	NUM	1
policajca	policajac	N	1
tom	tom	N	1
Op	Op	N	1
Op	Op	ABB	1
bolnice	bolnica	N	1
aba	aba	N	1
ortopedije	ortopedija	N	1
asova	as	N	1
Milo	Milo	N	1
posle	posle	ADV	1
povrede	povrediti	V	1
povrede	povreda	N	1
Milica	Milica	N	1
radnom mestu	radno mesto	N	1
policijskoj stanici	policijska stanica	N	1

Tabela 5.1: Dobijeni rezultati korišćenjem Uniteks platforme

Prikazani alat, kao i ostali servisi ove grupe istraživača realizovani su na bazi Uniteks platforme.

5.1.2. Prikaz stanja u Krivičnom zakoniku Republike Srbije

Skup svih tekstualnih dokumenata koji predstavljaju zakone Republike Srbije

predstavlja ogroman repozitorijum podataka. Predstavnici Vlade Republike Srbije svakodnevno proizvode i koriste veliku količinu tekstualnih dokumenata u vezi sa zakonima. Posebnu grupu zakona predstavljaju zakoni koji se odnose na krivično pravo. Jedan od njih je i Krivični zakonik. Važeći Krivični zakonik je objavljen i ispravljan u sledećim brojevima službenog glasnika: ("Sl. glasnik RS", br. 85/2005, 88/2005 - ispr., 107/2005 - ispr., 72/2009, 111/2009, 121/2012, 104/2013 i 108/2014) [108].

Krivični zakonik Republike Srbije koriste predstavnici Vlade Republike Srbije, građani, predstavnici privrednih subjekata, advokati i drugi. Za tumačenje ovog zakonika je bitno definisati značenja telesnih povreda od strane glavnih korisnika.

Krivični zakonik Republike Srbije u 36 glava sagledava problematiku krivice i odredbe sankcija u vezi sa izvršenim krivičnim delima. Jedan od segmenata kojim se bavi ovaj zakonik jesu i telesne povrede. One se obrađuju u zakoniku sa više aspekata. Za ovo istraživanje, na osnovu mogućnosti nanošenja telesnih povreda, odabrana su tri člana Zakonika:

1. Teška telesna povreda Član 121.
2. Laka telesna povreda Član 122.
3. Prinuda Član 135.

Analizom ovih članova zakona zajedno sa naslovom dokumenata kroz veb-servis VebRanka, dobijen je set reči pri čemu su uključene uključene vrste reči u srpskom jeziku i to: CONJ-konjunkcija, PREP-preparacija, N-imenica, ADV-pridev, PAR-parafraza, PRO-zamenica, ABB-skraćenica, INT-poziv, V-glagol, NUM-broj i A-atribut. Na osnovu analize sadržaja, koju su izvršili glavni korisnici Krivičnog zakonika Republike Srbije, došlo se do zaključka da se iz gornjeg skupa mogućih vrsta reči za dalju analizu treba uključiti isključivo sledeći skup vrsta reči: N, V, A, PREP. Smanjivanjem početnog skupa vrste reči za analizu, početni skup reči celog teksta je smanjen za oko 30%. Posle sagledavanja korisnih informacija iz prethodno dobijenog skupa, a uz pomoć eksperata za ovu vrstu dokumenata (u ovom slučaju advokata), dobijen je konačan skup najbitnijih reči za tekst iz ovog dela Zakonika.

Takođe, gramatički gledano, u srpskom jeziku gradivnim osnovama smatraju se osnove od kojih nastaju nove reči. Nove reči mogu nastati od korena reči ili od gramatičke osnove neke druge reči. Koren reči i gramatička osnova se mogu poklapati ali i ne moraju. Koren reči je nosilac leksičkog značenja i predstavlja jedinicu koja se ne može deliti na sitnije jedinice sa leksičkim značenjem. Gramatička osnova reči služi kao gradivna osnova nove reči. [109]

teška	organ	teško
telesna	prouzrokovana	povredi
povreda	nesposobnost	naruši
telesno	povređenog	prinuda
povredi	kazniti	pretnja
trajno	zatvorom	ubistvom
meri	nehata	teškom
oštećen	napadom	telesnom
oslabljen	laka	povredom
tela	telo	posledice

Tabela 5.2: Najbitnije reči iz Krivičnog zakonika RS za članove 121., 122. i 135.

Tako skup najbitnijih reči iz Tabele 5.2 možemo proširiti ukoliko se sve reči svedu na koren reči ili gramatičke osnove i njima se dodaju odgovarajući nastavci kako bi se dobile reči sa pravim značenjem. Ukoliko se primene prethodna pravila, pored reči teška iz Tabele 5.2 imamo još: teško, teške, tešku i teški; za reč telesna imamo: telesne, telesno, telesnu i telesni; i tako dalje za svaku reč iz Tabele 5.2. Na taj način možemo dobiti proširen skup najbitnijh reči iz Krivičnog zakonika Republike Srbije za pojam telesne povrede:

- kazn/u/a/o/e/i/javati/javan/iti;zatvor;traj-an/no;
- tel-o/esni/esna; povred-/a/iti; napa-d/sti/dati;
- kriv-ica/vac/ičan/ično; jak-a/o; nehat-a; opasnost;
- pro-uzrok-/ovan/ovati; nesposobnost; dove-den/sti; telesna povreda; smrt-an/no/nim.

Za pronalaženje podataka i informacija iz nestruktuiranih tekstualnih dokumenata u e-Upravi Republike Srbije pored primene raznih naučnih postavki i metoda, kao i NLP koji se odnosi na srpski jezik, neophodna je i primena tehnika i tehnologija za praktičnu realizaciju. Inače, za sam proces IR dve komponente su veoma važne: indeksiranje i pretraživanje. Pored toga, neophodno je i usklađivanje aktivnosti i metoda između njih. Tehniku i tehehnologiju za ovaj proces može da obezbedi *Lucene*. *Lucene* se koristi kao jednostavna biblioteka za pretragu i rukovanje komponenti u sklopu drugih aplikacija za pretraživanje: kretanje po internetu (*eng. crawling*), filtriranje dokumenata, *runtime* server, korisnički interfejs, administracija i drugo [110]. To je Java biblioteka otvorenog koda koja podržava procese i tehnike za pronalaženju informacija.

5.1.3. Koncepti poređenja kratkog teksta: poređenje članova Krivičnog zakonika

Klasifikacija teksta BoW pristupom, u mnogim aplikacijama je primenjiva, naročito kada je u pitanju kratak tekst ili jednostavne *on-line* aplikacije koje zahtevaju brže treniranje sistema i adaptaciju istog u slučaju dodavanja novih reči [111]. Pojam „torba koncepata“ - BoC su prvi put predložili Sahlgren i ostali u radu [112]. Tokom realizacije,

potrebno je razvrstati baze znanja kako bi se izvršila transformacija termina u koncepte. Zatim, kratak tekst mora da se klasifikuje, kako bi se razumo njegov sadržaj i neophodno je izvršiti njegovo povezivanje sa relevantnim konceptima (konceptualizacija). Konceptualizacija kratkog teksta ima za cilj da izdvoji skup najreprezentativnijih termina koji ga najbolje opisuju [113,114]. Pošto kratkim tekstovima obično nedostaje značenje, poređenja kratkog teksta i unapred definisanih koncepata mogu bolje pomoći procesu davanja smisla tekstualnim podacima, proširivanju tekstova sa kategorijama ili tematskim informacija i olakšavanju rada mnogih aplikacija. Poređenja kratkih tekstova sa velikim skupom otvorenih domenskih koncepata doprinosi njegovoj uspešnoj primeni. U ovom istraživanju glavni zadatak je da se izvrši prevođenje kratkog teksta koji u Krivičnom zakoniku predstavlja neki član Zakonika u vektor koji najbolje opisuje taj deo Zakonika u oblasti krivičnih dela, ali bez definisanog domena od strane eksperata. Da bi se dobili reprezentativni vektori, početak ovog istraživanja ogleda se u sagledavanju postojećih automatizovanih i računarskih metoda za ekstrahovanje ključnih reči.

Pod ključnim rečima se podrazumevaju termini koji imaju bitne informacije iz tekstualnog dokumenta. Automatsko izdvajanje ključnih reči je postupak pronalaženja malog skupa reči ili fraza koje predstavljaju suštinu tekstualnog dokumenta. Postojeće metode koje se odnose na automatsko izdvajanje ključnih reči iz tekstualnih dokumenata se mogu podeliti u četiri osnovne grupe:

- jednostavan statistički pristup (statističke informacije o rečima zasnovane na učestalosti pojavljivanja reči, TF-IDF, matrici uzajamnog pojavljivanja itd.),
- lingvistički pristup (leksička analiza, sintaksna analiza),
- pristup baziran na mašinskom učenju (model se generiše na bazi skupa dokumenata, gde su izdvojene ključne reči i koristi se za pronalaženje istih u novim dokumentima),
- poziciono-težinski pristup (reči na različitim pozicijama imaju različit stepen značajnosti) [106].

Algoritam za indeksiranje, koji je poznat kao TF-IDF zahteva postojanje skupa tekstualnih dokumenata, kako bi se pomoću njega dobile ključne reči. Ostali navedeni pristupi za automatsko izdvajanje ključnih reči zahtevaju odgovarajuće leksičke resurse ili korišćenje velikog skupa označenih tekstualnih dokumenata. Usled nedostatka korpusa tekstualnih dokumenata, ključne reči se mogu automatski izdvojiti tako što se uzima broja pojavljivanja termina u posmatranom tekstualnom dokumentu. Pre nego što se realizuje proces automatskog izdvajanja ključnih reči iz tekstualnog dokumenta, tekst mora biti normalizovan. Još pre toga tekst treba očistiti od stop-reči i drugih neslovnih oznaka, kao i podeliti na rečenice. Nakon toga sledi proces automatskog izdvajanja ključnih reči iz tekstualnog dokumenta.

Ključne reči mogu da posluže kao rezime za dokument koji obezbeđuje poboljšanje procesa IR ili povezivanje sa kolekcijom dokumenata [115]. Automatizacija ekstrakcije

ključnih reči je proces izvlačenja malih skupova reči i fraza koje opisuje sadržaj tog dokumenta. Mnoge studije o ekstrakciji ključnih reči imaju za cilj da olakšaju pronalaženje informacija. Postoje četiri kategorije metode za ekstrakciju ključnih reči: statističke, lingvističke, metode mašinskog učenja i metode hibridnih pristupa [116]. Učestalost pojavljivanja (*eng. frequency count*)reči u statističkom pristupu se primenjuje na dokument. Četiri vrste učestalosti pojavljivanja se mogu izvesti na nekoj kolekciji dokumenata. To su učestalosti pojavljivanja u svakom poddokumentu sa i bez skraćivanja reči i učestalost pojavljivanja u dokumentu sa i bez skraćivanja reči. TM tehnike obično uspostavljaju niz značajnih reči i rečenica na osnovu faktora statističke analize kao što su termini učestalost i distribucija. Kao deo procesa, moderni alati obično kreiraju matricu termina dokumenta (DTM) i koristite TF-IDF za ponderisanje [117].

Frekfencija termina i inverzna frekvencija dokumenta – TF-IDF

Značaj primene mere TF-IDF ogleda se u čestoj primeni u IR i DM. Ova mera ukazuje na značaj date reči, tj. njenu reprezentativnost u odnosu na dokument ili skup dokumenata, što znači da sa učestalošću njenog pojavljivanja u dokumentu raste i stepen njenog značaja. Varijante ovakvih merenja primenjuju se u pretraživanju interneta pomoću standardnih pretraživača, radi dobijanja adekvatnih rezultata pretraga u odnosu na zadati upit.

TF-IDF mera uglavnom služi za merenje sličnosti između dokumenata, gde se prvo dokumenti opišu u vektorskom prostoru, a zatim se vrši poređenje između vektora. Posebno se primenjuje kod rešavanja problema pronalaženja adekvatnih dokumenata. TF-IDF meri sve elemente dobijenih vektora, od kojih je svaki povezan sa odgovarajućim podacima skupa dokumenata. Prvo se meri učestalost termina u svakom dokumentu, što se predstavlja pojmom frekventnost termina - TF. Ukoliko je TF u dokumentu veći, onda je i posmatrani termin bitniji za dati dokument. Obrnuto, ako je potrebno izraziti manji broj pojavljivanja termina u dokumentu onda je reč o pojmu inverzna frekventnost dokumenta – IDF. Ova merenja važe nad celim skupom dokumenata. Čak i kada se isti termin nalazi u većem broju dokumenata i samim tim je veoma učestao u celom skupu dokumenata, to ne znači da je on reprezenativan. Dobar primer je učestalo pojavljivanje stop-reči u tekstualnim dokumentima, gde one nisu i ne mogu biti reprezentativne, dok neke reči, koje dokument sadrži, mogu biti reprezentativne i ako se pojavljuju sa manjom učestalošću.

Inače, TF-IDF mera je veoma značajna za pretragu informacija nad skupom dokumenata, tj. vektora, tako što poredi vektor upita (tekstualni dokument) sa vektorima skupa dokumenata.

TF označava broj pojavljivanja reči u datom tekstualnom dokumentu. IDF označava stepen značaja date reči. Ukoliko se izvrši deljenje sume svih dokumenata sa sumom dokumenata u kojima je pronađena data reč i primeni logaritamska funkcija, onda se

dobija IDF. Konačno, množenjem ove dve vrednosti dobija se TF-IDF mera i ona predstavlja značaj reči u dokumentu ili skupu dokumenata.

TF-IDF funkciju težine je pronašao Berger i ostali [118] koja glasi:

TF: $$\mathbf{TF(t, d_i) = \frac{n_{t,i}}{\sum_{k=1}^{|T|} n_{k,i}}} \quad \textbf{(4)}$$

gde je $TF(t, d_i)$ = frekvencija termina t u dokumentu d_i,

$n_{t,i}$ = broj pojavljivanja termina t u d_i,

$n_{k,i}$ = broj pojavljivanja svih termina d_i.

IDF: $$IDF_t = \log \frac{M}{m_t + 0.01} \quad (5)$$

gde je M = ukupan broj dokumenata u korpusu,

$\mathrm{m_t}$ = ukupan broj dokumenata u korpusu gde se pojavljuje termin t.

TF-IDF: $$\mathbf{w(t, d_i) = TF(t, d_i) \times IDF_t} \quad (6)$$

gde je $w(t, d_i)$ = težina termina t u dokumentu d_i.

Često korišćeni algoritam za indeksiranje TF-IDF zahteva postojanje korpusa dokumenata, da bi se pomoću njega dobile ključne reči. U nedostatku korpusa ključne reči se mogu izdvojiti na osnovu broja pojavljivanja termina u posmatranom dokumentu. U tabeli 5.3. prikazano je sedam najzastupljenijih termina u tri odvojena dokumenta koji predstavljaju posmatrane članove Krivičnog zakonika Republike Srbije. Pri tome, primenjena je normalizacija reči korišćenjem morfološkog rečnika. Rezultati normalizacije izvršene odsecanjem na N-grame dužine četiri, prikazni su u tabeli 5.4.

R.br.	Termin	Frekvencija
1.	delo	3
2.	učinilac	3
3.	povreda	2
4.	nastupila	2
5.	zdravlje	2
6.	telesna	1
7.	teška	1

Tabela 5.3: Normalizacija pomoću morfološkog rečnika

R.br.	Termin	Frekvencija
1.	delo	3
2.	učin*	3
3.	kazn*	3
4.	zatv*	3
5.	tele*	3
6.	tešk*	3
7.	povr*	3

Tabela 5.4: Normalizacija odsecanjem na N-grame dužine četiri

Skraćivanjem reči na četiri karaktera broj pojmova obuhvaćenih ključnim rečima se povećao, jer, na primer, reči „ kazniti “ i „ kaznom “ su skraćene na istu osnovu, a nova ključna reč je N-gram „kazn“.

Apache Lucene

Lucene indeks obezbeđuje poklapanje termina sa dokumentom. To se zove inverzni indeks, jer „okreće“ uobičajeno poklapanje dokumenata sa terminima koje sadrži. Inverzni indeks obezbeđuje mehanizam za ponderisanje rezultata pretrage: ako je broj pojavljivanja traženih termina prisutan u istom dokumentu, onda je verovatno da taj dokument bude relevantan za pretragu. Polja pod nazivom tekst sadrže reči, tj. izvršena je tokenizacija ulaznih reči, pa su nove reči napisane malim slovima i bez interpunkcije. Stavke inverznog indeksa za uslove koji se sastoje od polja ime teksta i tokene: povredu, teške i povrede, date su u tabeli 5.5.

Termin	Dokument
text: povredu	1, 4
text: teške	5, 9, 3, 6, 7
text: povrede	2, 5, 9, 3, 6, 7, 8

Tabela 5.5: Inverzni indeks dokumenata

Ovde su brojevi dokumenata interne reference *Lucene* na dokument. Ovi *id*-ovi nisu statički. *Lucene* upravlja sa *id* dokumenata kao što upravlja indeksima. Unutrašnje numerisanje *id*-ova se može promeniti kada se dodaju novi dokumenti i kada se neki dokumenti brišu iz indeksa.

Velika prednost *Lucene* je u tome da ne postoji fiksna globalna šema termina, kao što je slučaj sa bazama podataka. Svakom dokumentu se dodaje indeks koji je prazan. I naknadno upisani indeks je potpuno drugačiji od bilo kog prethodno priključenog. Indeksirani dokumenti mogu biti u istom polju, slično kao prethodno dodati

dokumenti, a da pripadaju sasvim različitim oblastima. To omogućava da se odmah dođe do traženog dokumenta bez promene (re-dizajn) šeme.

Pojam i definicija klasterizacije podataka

Klasterizacija je proces kojim se podaci, u odnosu na svoje karakteristike, klasifikuju u određene razrede – klastere, tako da se, u okviru istog klastera nalaze slični podaci, odnosno oni koji imaju zajedničke osobine. To je upravo i razlog što se ovaj proces može pogrešno tumačiti kao klasifikacija podataka i u slučajevima gde je razlika vidna. Klasifikacija predstavlja proces u kome se podaci svrstavaju prema osobinama u skupove koji su već formirani, dok proces klasterizacije omogućava formiranje skupova u odnosu na pronalaženje sličnosti među karakterisitkama samih podataka. To je ono što ovu metodu čini drugačijom, odnosno, u kojoj se ide ka tome da se u okviru jednog klastera nađu najsličniji podaci, tj. oni sa najviše zajedničkih karakteristika, dok će se u različitim klasterima naći podaci koji su najrazličitiji. Sa enormnim i stalnim porastom broja podataka, proces pretraživanja postaje sve složeniji, te je samim tim neophodno pronaći najefikasniji način za pretraživanje podataka i mogućnost da se oni razvrstaju po logičkom kriterijumu. Takođe, u ovom procesu važno je da se ovi ogromni i složeni skupovi podataka urede i predstave u određenim klasterima prema njihovim karakteristikama. Pretragu je moguće olakšati grupisanjem podataka, koja utiče i na brže pretraživanje informacija i dobijanje odgovora.

Klasteri predstavljaju grupe dokumenata koji imaju zajedničke osobine. Osobine klastera se zasnivaju na osobinama dokumenta koje sam dokument klasifikuju u taj klaster. Što je više zajedničkih osobina između dokumenta i klastera, to je i veza dokumenta i klastera jača, odnosno, povećava se pripadnost dokumenta klasteru. Klaster, u tom slučaju, predstavlja deo nekog apstraktnog prostora u kome je gustina zajedničkih tačaka velika. To, zapravo, znači da je, u određenom skupu, masa podataka sa istim karakteristikama veoma čvrsto grupisana, dok je sa različitim karakteristikama izdvojena u različitim klasterima.

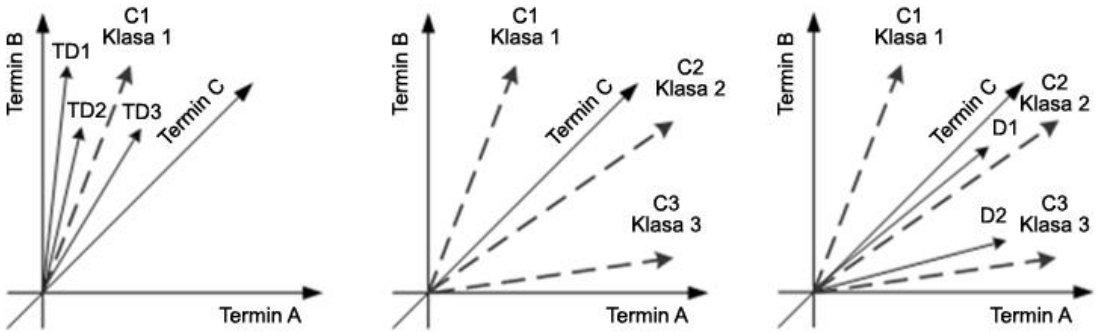
Za svaki dokument ili reprezentaciju dokumenata najsličniji vektor klase određuje najsličniju klasu. Moguće je da se dodeli *n* najsličnijih vektora klasa ili klase.

Proces klasifikacije može biti jednostavno predstavljen:

1) Izračunavanje centroida za svaku klasu.

Skup dokumenata za treniranje sistema za svaku klasu predstavljen u vektorskom prostoru definiše centroid koji predstavlja opis klase u vektorskom prostoru (slika 5.5, A). Za svaki skup dokumenata za treniranje sistema koji definiše određenu klasu, izračunava se odgovarajući vektor centroida (slika 5.5, B).

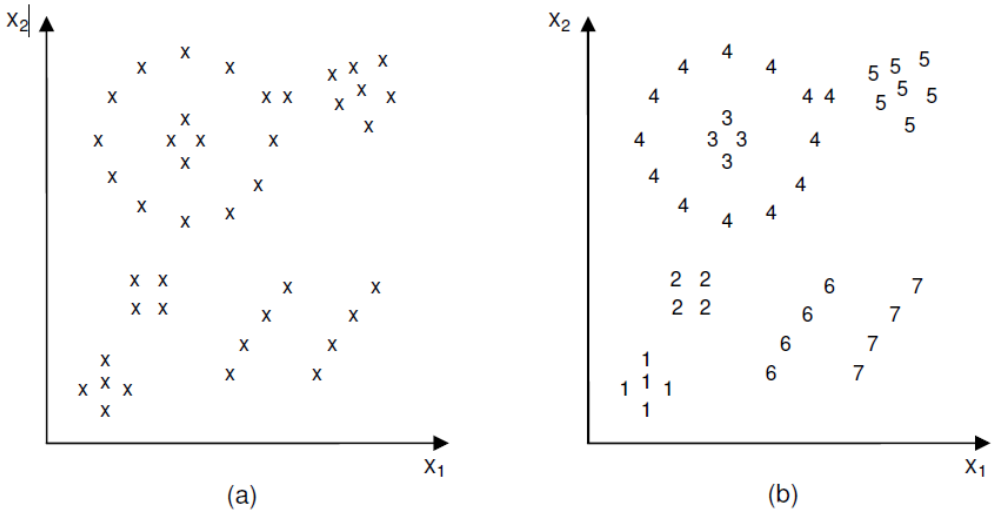
2) Dodeljivanje klasa za svaki dokument.



Slika 5.5: Prikaz procesa klasifikacije u n-dimenzionalnom prostoru

Realizacijom procesa klasterizacije termina omogućava se lakše pronalaženje dokumenata i ima primenu u mnogim delatnostima. Postoje dva osnovna načina klasterizacije podataka, odnosno dokumenata: hijerarhijska i partitivna klasterizacija. Primenom hijerarhijskog algoritma klasterizacije vrši se grupisanje dokumenata u najsličniji klaster, gde su klasteri unapred formirani. Primenom partitivnog algoritma klasterizacije vrši se kreiranje svih klastera odjednom. Takođe, partitivni algoritmi klasterizacije se mogu koristiti kao pomoćni u hijerarhijskoj klasterizaciji.

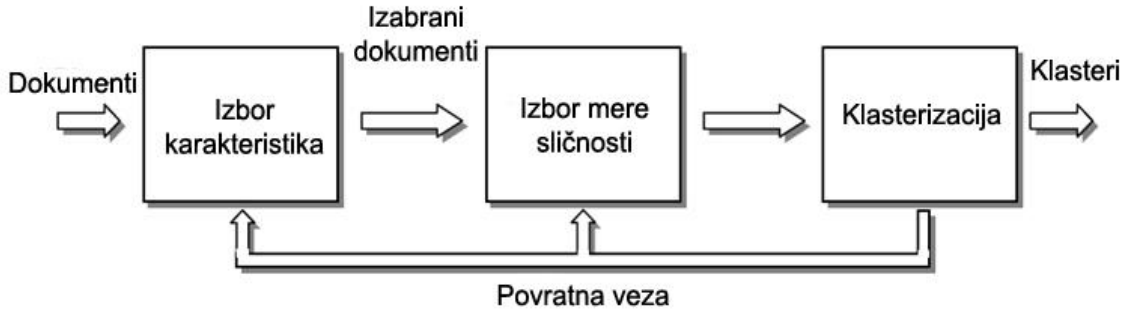
Analiza organizacije klastera je analiza skupa uzoraka, koji su predstavljeni kao vektori u višedimenzionalnom prostoru. Klasteri su bazirani na sličnosti. Uzorci unutar jednog klastera sličniji su jedni drugima nego što su to uzorci koji pripadaju različitim klasterima (slika 5.6). Ulazni skup tačaka je prikazan na slici (a), a željeni klasteri na slici (b). Tačke koje pripadaju istim klasterima imaju postavljene iste brojeve. Postoji velik skup tehnika za prikaz podataka, odnosno dokumenata, mere sličnosti između njih i njihovo pozicioniranje u klasterima.



Slika 5.6: Klasterizacija podataka

Slika 5.7 prikazuje najčešći redosled prva tri koraka klasterovanja, gde je uključena i povratna veza tako da postupak klasterizacije može uticati na izbor karakteristika i izbor mere sličnosti ako se pokaže da rezultati nisu zadovoljavajući. Sličnost uzoraka se

meri funkcijama sličnosti na uzorcima. Postoje različite funkcije sličnosti za različite potrebe.



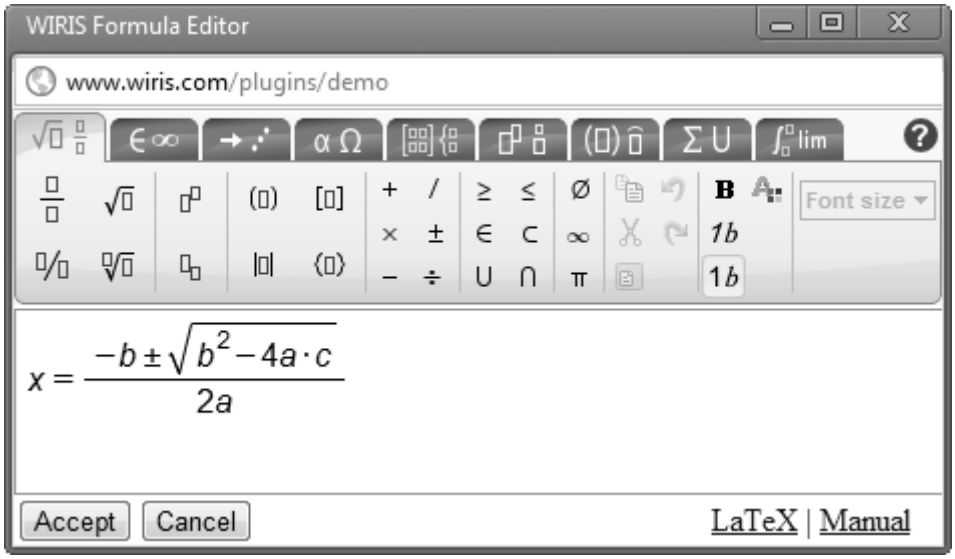
Slika 5.7: Postupak klasterizacije

5.1.4. Princip pretraživanja primenom MathML standarda

Mathematical Markup Language (MathML) prikazuje matematičke formule i izraze u formatu teksta, koji je pogodan za njihovo čuvanje u bazi podataka.

Matematičke formule na web-stranici

Postoje razni načini da se matematičke formule i simboli prikažu na veb-stranici, ali preporuka konzorcija W3C je svakako standard MathML. Većina današnjih veb-pretraživača/pregledača imaju mogućnost podrške za prikaz MathML standarda. Na jednostavan način matematičke formule, izrazi i grafikoni mogu biti uključeni u HTML kod veb-stranice pomoću *plug-in* WIRIS editora kroz softver Ckeditor.



Slika 5.8: plug-in WIRIS editora u softveru Ckeditor

Primer korišćenja WIRIS editora u kreiranju veb-stranice koja sadrži neophodne formule radi pojašnjenja koncepta problema u datom tekstu prikazana je na slici 5.8.

System excitation by voltage short pulses is called the Dirac excitation. For the EEC of the electrochemical system, a short potentiostatic pulse (usually far from equilibrium) and subsequent forced relaxation toward previously registered open-circuit potential.

Curves in Dirac excitation can be described by the following:

$$I_1 = \frac{E_1}{R_0} \quad (1)$$

$$I_2 = \frac{E_2}{R_0 + R_{123}} \quad (2)$$

$$I_{10} = \frac{-U_{C_{10}}}{R_1 + R_{023}} \quad (3)$$

where:

$$R_{123} = \frac{R_1 \times R_2 \times R_3}{R_1 \times R_2 + R_1 \times R_3 + R_2 \times R_3} \quad (4)$$

$$R_{023} = \frac{R_0 \times R_2 \times R_3}{R_0 \times R_2 + R_0 \times R_3 + R_2 \times R_3} \quad (5)$$

In Eq. 3 R_1 is in series with equivalent resistance of R_0, R_2 and R_3 in parallel connection, defined with Eq. 5, because Eq. 3 describes the system immediately after switching off if the Dirac pulse.

Slika 5.9: Izgled veb- strane sa matematičkim formulama u tekstu [119]

Opis slika tekstualnom

Ako na veb-stranici postoje slike, preporučuje se korišćenje atributa ALT koji može da opiše sliku u tekstualnom formatu. Poznati pretraživači, kao što je Google upravo na ovaj način imaju mogućnost da „vide“ sadržaj slike. Oni na osnovu atributa ALT mogu da pretpostave šta se na njoj od sadržaja i nalazi. Osnovna namena postojanja atribut ALT u HTML fajlu jeste da odredi alternativni tekst kao zamenu za sliku u slučaju kada slika ne može biti prikazana u pretraživaču/pregledaču ili usled greške prilikom učitavanja stranice. Vrednost atributa ALT se definiše od strane autora veb-stranice. Atribut ALT se u najvećem broju slučajeva koristi da opiše ono što se na datoj slici nalazi. Ujedno, atribut ALT pomaže pretraživačima u odredjivanju relevantnosti slika, pri čemu je potrebno da opis slike bude kratak i jasan.

Pomenuti softver Ckeditor koristi atribut ALT za opis formula na definisan način radi tumačenja sadržaja formula na jednostavan verbalan način bez opisa njenog značenja. Tako, na primer, za formule predstavljene na slici 5.9, imamo da je za formulu numerisanu sa brojem (1) prikazano sledeće:

„I subscript 1 equals E subscript 1 over R subscript 0“

Predstavljeni tekst ne reprezentuje adekvatno značenje formule pa je neophodno izvršiti njegovu transformaciju u tekst koji adekvatnije opisuje formula, a to je:

„instant charging current“

Nakon izvršenih promena HTML kod veb-stranice za prethodni primer sa MathML oznakama ima sledeći izgled:

```
<p>System excitation by voltage short pulses is called the Dirac excitation. For the EEC of the electrochemical system, a short potentiostatic pulse (usually far from equilibrium) and subsequent forced relaxation toward previously registered open-circuit potential.</p>
<p>Curves in Dirac excitation can be described by the following:</p>
<p style="text-align: center;">
<img align="middle" alt=" instant charging current " class="Wirisformula" data-custom-editor="chemistry" data-mathml="«math xmlns=¨http://www.w3.org/1998/Math/MathML¨»«msub»«mi»I«/mi»«mn»1«/mn»«/msub»«mo»=«/mo»«mfrac»«msub»«mi»E«/mi»«mn»1«/mn»«/msub»«msub»«mi»R«/mi»«mn»0«/mn»«/msub»«/mfrac»«/math»" height="42" role="math" src="/pluginwiris_engine/app/showimage?formula=2ce99201875b83e50362d9c152cd5b2d&cw=62&ch=42&cb=26" style="vertical-align: -16px;" width="62" /><span style="float:right">(1)</span></p>
<p style="text-align: center;">
<img align="middle" alt="quasi-stationary charging current (current plateau on the curve for extremely short pulse duration) " class="Wirisformula" data-custom-editor="chemistry" data-mathml="«math xmlns=¨http://www.w3.org/1998/Math/MathML¨ class=¨wrs_chemistry¨»«msub»«mi»I«/mi»«mn»2«/mn»«/msub»«mo»=«/mo»«mfrac»«msub»«mi»E«/mi»«mn»2«/mn»«/msub»«mrow»«msub»«mi»R«/mi»«mn»0«/mn»«/msub»«mo»+«/mo»«msub»«mi»R«/mi»«mn»123«/mn»«/msub»«/mrow»«/mfrac»«/math»" height="42" role="math" src="/pluginwiris_engine/app/showimage?formula=cb3f8b0fb238edd938b05858c09cd76d&cw=114&ch=42&cb=26" style="vertical-align: -16px;" width="114" /><span style="float:right">(2)</span></p>
<p style="text-align: center;">
<img align="middle" alt=" instant discharging current" class="Wirisformula" data-mathml="«math xmlns=¨http://www.w3.org/1998/Math/MathML¨»«msub»«mi»I«/mi»«mn»10«/mn»«/msub»«mo»=«/mo»«mfrac»«mrow»«mo»-«/mo»«msub»«mi»U«/mi»«msub»«mi»C«/mi»«mn»10«/mn»«/msub»«/msub»«/mrow»«mrow»«msub»«mi»R«/mi»«mn»1«/mn»«/msub»«mo»+«/mo»«msub»«mi»R«/mi»«mn»023«/mn»«/msub»«/mrow»«/mfrac»«/math»" height="46" role="math" src="/pluginwiris_engine/app/showimage?formula=8b8d1cf8c0a56f7b6488b072ce63d679&cw=122&ch=46&cb=30" style="vertical-align: -16px;" width="122" /><span style="float:right">(3)</span></p>
<p>where:</p>
<p>
<img align="middle" alt=" equivalent resistance of R1, R2 and R3 in parallel connection " class="Wirisformula" data-mathml="«math
```

xmlns=¨http://www.w3.org/1998/Math/MathML¨»«msub»«mi»R«/mi»«mn»123«/mn»«/msub»«mo»=«/mo»«mfrac»«mrow»«msub»«mi»R«/mi»«mn»1«/mn»«/msub»«mo»§#215;«/mo»«msub»«mi»R«/mi»«mn»2«/mn»«/msub»«mo»§#215;«/mo»«msub»«mi»R«/mi»«mn»3«/mn»«/msub»«/mrow»«mrow»«msub»«mi»R«/mi»«mn»1«/mn»«/msub»«mo»§#215;«/mo»«msub»«mi»R«/mi»«mn»2«/mn»«/msub»«mo»+«/mo»«msub»«mi»R«/mi»«mn»1«/mn»«/msub»«mo»§#215;«/mo»«msub»«mi»R«/mi»«mn»3«/mn»«/msub»«mo»+«/mo»«msub»«mi»R«/mi»«mn»2«/mn»«/msub»«mo»§#215;«/mo»«msub»«mi»R«/mi»«mn»3«/mn»«/msub»«/mrow»«/mfrac»«/math»" height="42" role="math" src="/pluginwiris_engine/app/showimage?formula=e119eaf25f0974e73698eec97fa3e32a&cw=263&ch=42&cb=26" style="vertical-align: -16px;" width="263" />(4)

(5)

In Eq. 3 R₁ is in series with equivalent resistance of R₀, R₂ and R₃ in parallel connection, defined with Eq. 5, because Eq. 3 describes the system immediately after switching off if the Dirac pulse.

Na osnovu urađenih transformacija sadržaja ALT tagova, primenom MathML standarda stvorena je mogućnost kreiranja koncepta sadržaja veb-stranice korišćenjem matematičkih formula. Pojam koncepta „Dirakov impuls“ se u vektorskom obliku može predstaviti na sledeći način:

Dirac pulse = [parallel, connection, current, quasi-stationary,…]

5.2. Predloženi okvir za realizaciju mogućeg QA veb-servisa

Buduće QA aplikacije trebalo bi da pomognu građanima da dobiju u najkraćem roku odgovore na postavljene upite tj. pitanja u bilo koje vreme. Obično, vrsta sadržaja u takvom sistemu su upiti, formalni dokumenti i stručni odgovori. Dokumenti mogu biti grupisani na osnovu ključnih reči. Kada korisnik postavlja novi upit za neki domen,

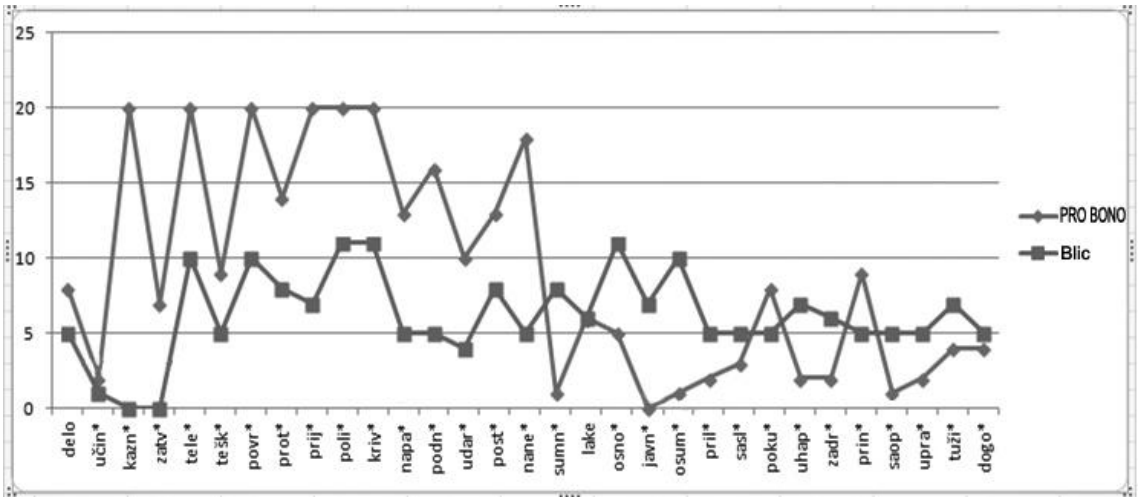
onda se izračunava njegova sličnost sa postojećim dokumentima iz odabrane grupe. U slučaju predloženog sistema, upiti građana se odnose na krivična dela koja su definisana u tri člana Krivičnog zakonika: 121, 122 i 135. Delovi ovog zakonika u predloženom okviru za QA sisteme u ovoj doktorskoj disertaciji, predstavljeni su kao tri različita dokumenta grupisana u celinu, koja prikazuje trenutno raspoloživu bazu znanja (*eng. knowledge base*). Da bi dopunili i popunili ovaj centralizovani repozitorijum, korišćeni su odgovori slučnog lica (eksparta za datu oblast) sa veb-portala za besplatnu pravnu pomoc – PRO BONO (slika 5.10) [121]. Analizirani su postojeći odgovori na pitanja građana koja se odnose na krivična dela iz navedena tri člana Krivičnog zakonika u okviru sekcije „pitanja i odgovori" za oblast krivično pravo. Iz velikog broja postavljenih pitanja izdvojeno je 45 reprezentativnih pitanja (upita) kao moguća grupa kratkih tekstualnih dokumenata koja će pomoći da se bolje predstavi razmatrani BoC.

Za kreiranje metode mapiranja odgovora sa članovima Krivičnog zakonika Republike Srbije, predloženo u ovoj doktorskoj disertaciji, korišćen je BoC pristup ponderisanja vrednosti TF-IDF za tri odvojena dokumenta, članova zakonika. Tehnika ponderisanja koristi kombinaciju termina TF vrednosti i IDF vrednosti u ovim dokumentima. Vrednost TF-IDF ne meri koliko često se pojavljuje ključna reč, ali nudi upoređivanje mere učestalosti pojavljivanja ključne reči u odnosu na očekivano pojavljivanje, prikupljeno iz većeg skupa dokumenata. Za optimizaciju pretraživača (*eng. Search Engine Optimization* - SEO) [120] merenje TF-IDF korelacija sa većim rang listama je, uglavnom, bolje rešenje nego individualna upotrebe ključnih reči. TF-IDF ujedno može da predstavlja BoW za termine iz dokumenata u ovom tekstualnom korpusu. Veća težina se daje terminu sa visokom frekvencijom u datom dokumentu, a manja težina ukoliko je niska frekvencija dokumenta za određeni termin u celom tekstualnom korpusu. Takođe, TF-IDF ima mogućnost da filtrira zajedničke termine iz više dokumenata čime se smanjuje broj relevantnih ključnih reči.

Slika 5.10: Portal PRO BONO

S druge strane, ekstrahovanje podataka iz veb-stranica za popunjavanje baza znanja, zahteva metode koje su pogodne za rad na domenima, ne zahtevaju dodatni napor da

se prilagode novim domenima i integrišu informacije dobijene iz više različitih veb-stranica [122]. *Google,* sa takvim jedinstvenim pristupom, je jedan od najpopularnijih veb-pretraživača. On se koristi da bi se pronašao sadržaj, tj. bilo koji tag sa ključnim rečima iz formiranog BoC. Pretraživači prepoznaju relevantne veb-stranice kao dodatni resurs za bolje objašnjenje termina koji bi mogli da budu značajni prilikom mapiranja odgovarajućeg upita sa adekvatnom grupom odgovora. Najčešće, prilikom postavljanja teme sadržaja članka na veb-stranici, administratori ujedno definišu i povezuju date članke sa nizom ključnih reči kroz opciju „povezani članci“ ili preko meta taga „ključne reči“.



Slika 5.11: Reprezentacija BoC u domenu kriminala za deo Krivičnog zakonika

Da bi se pronašla grupa reči koja se odnosi na dati problem, preko Google pretraživača postavljen je upit pojma “teška telesna povreda” koji je kao termin - osnovna lema, dobijena na bazi elektronskog rečnika za srpski jezik, korišćenjem dostupnog *on-line* veb-jezičkog resursa „vreća reči“ [123]. Tom prilikom repozitorijum koji je dao najbolje rezultate pretrage bio je veb-sajt dnevnih novina Blic. Sa ovog repozitorijuma izdvojeno je 35 tekstova koji su odgovarali navedenom upitu, tj. tagu članaka.

Reprezentacija BoC u domenu kriminala za deo Krivičnog zakonika je prikazana na slici 5.11.

Definisanje skupa frekventnih termina

U ovom primeru postavljena je granična vrednost dva za minimalnu frekvenciju termina. Termin mora da se pojavljuje bar dva puta u dokumentima koji se obrađuju, s obzirom na dužinu dokumenata, kako bi se uzeo u razmatranje. Pretpostavimo da je dužina dokumenta $c_1, c_2, \dots, c_n$ i *f*($c_1, c_2, \dots, c_n$) predstavlja frekvenciju, onda se izdvoji $c_1, c_2, \dots, c_n$ kao termin samo iz teksta ako je f($c_1, c_2, \dots, c_n$) jednak ili veći od dva [124, 125]. Uobičajeno je da se izabere središte svakog intervala: (v_i+v_i+1)/2 kao reprezentativna granična vrednost. Algoritam *C.45* uzima kao prag, manju vrednost v_i za svaki interval {v_i,v_i+1}, a ne srednju vrednost [126]. Polazeći od ove činjenice, u ovoj disertaciji, takođe, se koristi ovakva vrsta računanja praga.

5.2.1. Stop-reči

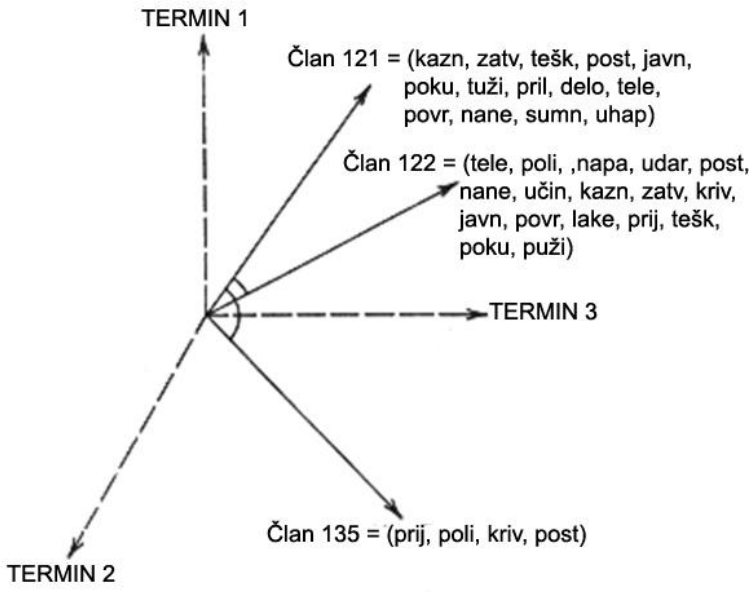
U prethodnim potpoglavljima definisano je da su stop-reči one reči koje ne nose neko relevantno značenje za temu koju razmatramo. Postoje određene vrste reči za koje je to u potpunosti istinito (npr. za veznike). U nekim drugim slučajevima i drugim tipovima reči nije tako. U tim slučajevima izbor stop-reči zavisi od konteksta u kome se razmatraju dokumenti. Ako je potrebno grupisati dokumente koji sadrže podatke o današnjim i prošlim događajima, tada se u listi stop-reči ne uključuju prilozi (npr. danas, juče, sada itd). Ako se isti dokumenti grupišu po značenju, tada bi trebalo navedene priloge uključiti kao stop-reči, jer u tom kontekstu nisu značajni za razmatranje. Imenice i glagoli retko se stavljaju kao stop-reči, ali i to je moguće u zavisnosti od konteksta u kome se dokumenti razmatraju. Opšta lista engleskih stop-reči se sastoji od oko 600 reči, a u SAS-u znatno manje – samo 330. Stop-reči koje su za ovo istraživanje korišćene broje oko 700 najčešće korišćenih reči. Deo te liste je prikazan na slici 5.12.

Slika 5.12: Deo liste stop-reči

Za srpski jezik trenutno ne postoji zvanična lista stop-reči, što se može videti na veb-sajtu https://sites.google.com/site/kevinbouge/stopwords-lists. Zato je ovde korišćen Srpski Lematizator i *PoS* (*eng. Part of speech*) Anotirani Korpus *(eng. Serbian Lemmatized and PoS Annotated Corpus*-SrpLemKor*)*. Kreirana je lista stop-reči od 700 srpskih reči. Osim toga, lista stop-reči se automatski generiše (izvodi) u svakom procesu treniranja sistema, preuzimanjem termina koji imaju najveću učestalost u celom korpusu.

5.2.2. QA sistem baziran na BoC modelu

Šema TF-IDF ponderisanja se često koristi u modelu vektora prostora (VSM). Ovaj model zajedno sa merama sličnosti se često koristi za određivanje sličnosti između dokumenata ili dokumenata i upita predstavljenih u vektorskom prostoru. Dokument je predstavljen terminima u n-dimenzionalnom vektorskom prostoru od $d_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$), gde je "$w_{t,j}$" definisana težina ključnih reči ili termina BoC.



Slika 5.13: Vektori dokumenata (tri člana Krivičnog zakonika) u prostoru

U današnjim veb-pretraživačima kada korisnik ubacuje upit u sistem za pretraživanje, prvo što mora da uradi je da utvrdi koje stranice u indeksu se odnose na upit, a koje ne. QA sistem za brze odgovore u domenu kriminala, koji je predložen u ovoj doktorskoj disertaciji, predstavljen na slici 5.14, bazira se na odsustvu ili prisustvu traženih termina u dokumentu, gde su dokumenti predstavljeni kao BoC.

Ovi koncepti su mapirani na rečniku kriminala za tri člana Zakonika i dati su kao vektori:

- Član 121 [kazn,zatv,tešk,post,javn,poku,tuži,pril,delo,tele,povr, nane, sumn,uhap]
- Član 122 [tele,poli,napa,udar,post,nane,učin,kazn,zatv,kriv,javn,povr,lake,prij, tešk,poku,tuži]
- Član 135 [prij,poli,kriv,post]

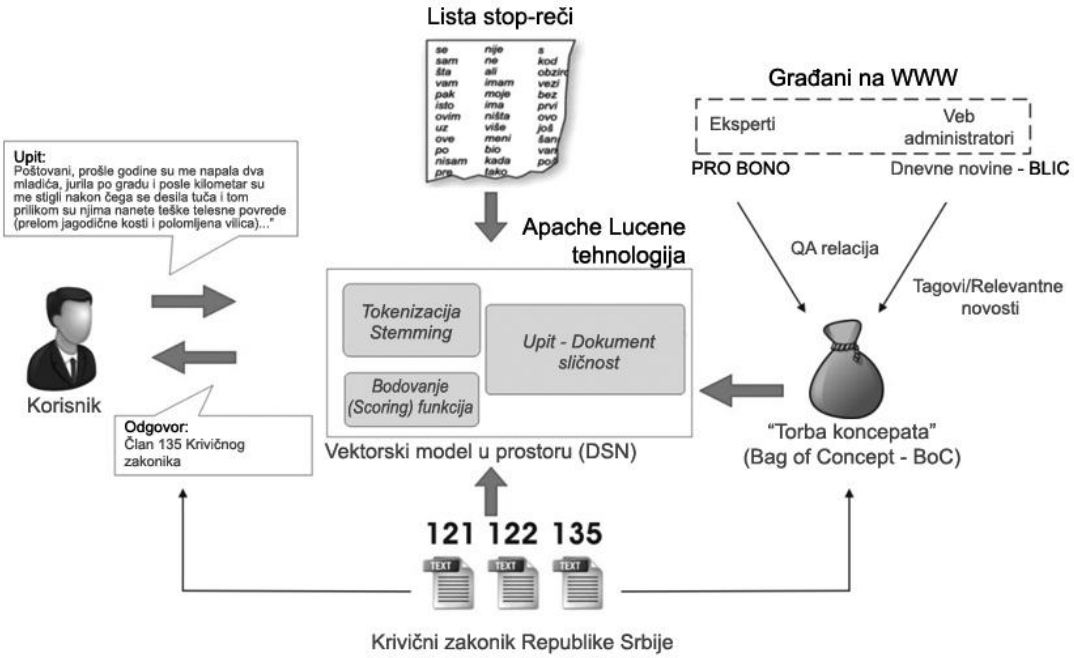
Vektori ovih dokumenata u prostoru je prikazan na slici 5.13.

Predstavljeni okvir u QA sastoji se od u klasifikacije pitanja Krivičnog zakonika korišćenjem posebnog domena baziranog na BoC reprezentaciji korpusa, kao i poređenja performansi sa performansama klasifikatora kada se koristi ekspertska prezentacija istih dokumenata.

U predloženom QA sistemu, prvo, neophodno je bilo da se izvrši mapiranje pitanja sa BoC iz korpusa (reči) definisanog na osnovu 31 termina kao što je prikazano na slici 5.14. Drugi korak u procesu tokenizacije pitanja je filtriranje stop-reči. Sledeći korak je stemovanje, uklanjanjanje zajedničkih afiksa iz reči, kako bi se obavio neki oblik morfološke normalizacije i stvorila više opštih karakteristika reči. U tom smislu,

koristimo 4-gram stemer što je najčešći algoritam za stemovanje za rad sa tekstom na srpskom jeziku. Na kraju je izračunata vrednost sličnosti između upita i BoC reprezentacije tri dokumenta putem funkcija sličnosti. Predloženi okvir u QA sistemu korisniku kao izlaz prikazuje relevantni dokument za postavljeni upit. Poruka koju korisnik kao odgovor dobija jeste: „Pogledajte član *n* Krivicnog zakona.“ Koji broj *n* će biti prikazan korisniku određuje deo sistema softvera koji se naziva Specifični *Annotator* (SA) za stemovanje.

SA je softverski agent koji kao izvor koristi BoC određenog domena iz baze znanja za mapu ekstrahovanja ključnim termina. On dodeljuje odsustvo ili prisustvo svakog ekstrahovanog termina u skladu sa relevantnim kljjučnim rečima u okviru BoC. Prevođenje upita i dokumenta iz „sirovih“ podataka u oblik potreban za poređenje, moguće je uraditi računarskom obradom, što predstavlja prvu prepreku u procesu izračunavanju sličnosti teksta. Kako bi se prevazišao ovaj problem, radi se konverzija reprezentacije tekstualnih informacija u reprezentaciju algebarskog vektora prostora [127].

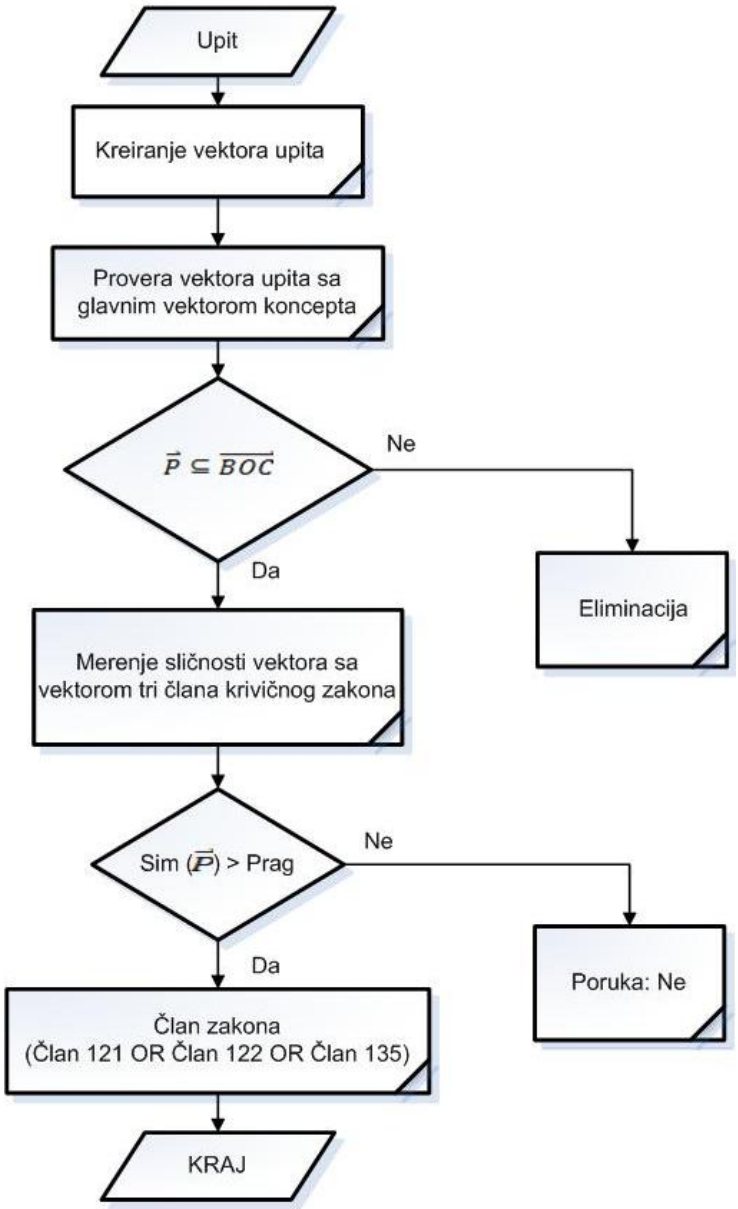


Slika 5.14: QA sistem baziran na BoC modelu

Da bi QA sistem u celini bio uspešan neophodna je uspostaviti dobru korelacija između klasifikacije upita i ekstrakcije tipa odgovora. Cilj SA jeste upravo da se sistem „nauči“ kako da na osnovu upita mapira odgovarajući tip odgovora [128].

Parametri za nadgledano učenje postavljeni su na bazi vrednosti koje su dale dobre rezultate u sličnim procesima klasifikacije teksta. Predloženi, novi, algoritam u sklopu SA ima (slika 5.15) je baziran na BoC pristupu sa zadatakom da automatizuje klasifikaciju. Pri čemu je BoC spisak svih reči rangiranih prema njihovoj deskriptivnoj

vrednosti za tri člana Krivičnog zakonika Republike Srbije (pripadnost klasteru). Mere sličnosti, u ovom slučaju upravo koriste vektorske reprezentacije dokumenata i pitanja za izračunavanje rastojanja između njih.



Slika 5.15: Algoritam za dela sistema za klasifikaciju upita za članove krivičnih dela Krivičnog zakonika

Kada se skup reprezentativnih termina koji je izabran za svaki klaster $c_j \in C(= \{c_1, c_2, \ldots c_n\})$, smatra reprezentativnim terminima (r_{tj}) klastera c_j, onda se upoređuju sličnosti svakog upita q_i mapiranog u BoC (poređenje vektora upita sa svakim vektorom klastera) r_{tj} korišćenjem metrike određene sličnosti za automatsko računanje relevantnosti rezultata koji meri sličnost između upita i člana Krivičnog zakonika: c_j. Da bi se odredio prag sličnosti između upita predstavljen u formi kratkog teksta i članova Zakonika, koji je takođe prikazan kao kratak tekst, korišćena je sledeća formula sličnosti [129]:

$$Similarity = \frac{W(Sa) \cap W(Sb)}{\min(W(Sa), W(SB))} \quad (7)$$

Gde je *w (Usa)* ∩ *w (Sb)* presek skupa broja reči u upitu q_i i broj reči u r_{tj}, a min(*w (Sa), w (Sb)*) je manja vrednost od broja reči u oba dokumenta.

6. Analiza eksperimentalnih rezultata

Pre samog procesa klasterizacije moraju se odrediti mere sličnosti (distance). Mera sličnosti je veoma važna zbog direktnog uticaja na rangiranje dokumenata, zapravo zbog direktnog uticaja na stepen blizine ili udaljenosti od ciljnih dokumenata. Pored toga, merenje sličnosti dokumenata na osnovu karakteristika koje zavise od vrste podataka koji su u kontekstu dokumenata i obrade, dovodi do grupisanja dokumenata i grupisanja dokumenata unutar klastera. Ne postoji mera sličnosti koja je univerzalno najbolja za klasterizaciju svih vrsta dokumenata.

Izbor odgovarajuće mere sličnosti je od ključnog značaja za klaster analizu, naročito za određenu vrstu algoritama za klasterovanje. Istraživanje koje predstavlja predmet ove doktorske disertacije uključuje tri vrste najčešće korišćenih sličnosti, kako bi se odabrala ona koja daje najpreciznije rezultate za oblast kriminala.

Nije svaka mera sličnosti pokazatelj realne situacije. Da bi se ona kvalifikovala kao pokazatelj, *d* bi trebalo da zadovolji sledeća četiri uslova. Neka su *x* i *y* bilo koja dva objekta u skupu i *d(x, y)* rastojanje između *x* i *y*.

1. Rastojanje između bilo koje dve tačke mora biti nenegativna vrednost, tj. $\mathrm{d}(\mathrm{x},\mathrm{y}) \geq 0$.
2. Rastojanje između dve tačke mora biti nula ako i samo ako su dva objekta identična, tj. $d(x, y) = 0$ ako i samo ako je *x* = *y*.
3. Udaljenost mora biti simetrična, tj. udaljenost od *x* do *y* je ista kao i rastojanje od *y* do *x* , tj. $d(x, y) = d(y, x)$.
4. Ova mera mora da zadovolji nejednakost trougla, koji je $d(x, z) \geq d(x, y) + d(y, z)$.

U ovom istraživanju korišćene su sledeće funkcije sličnosti:

1. Kosinusna sličnost (*eng. cosine similarity*)

Dokumenti su predstavljeni kao vektori termina i sličnost dva dokumenta odgovara korelaciji između vektora. Ovo se kvantifikuje kao kosinus ugla između dva vektora, odnosno kosinusna sličnost.

To kvantifikuje korelaciju između vektora $\overrightarrow{\mathrm{t_a}}$ i $\overrightarrow{\mathrm{t_b}}$, kao kosinus ugla između njih u n-dimenzionalnom prostoru. Vrednost kosinusne sličnosti je ograničena između [0,1] i nezavisna je od dužine dokumenta.

$$SIM_C(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{\overrightarrow{t_a}.\overrightarrow{t_b}}{|\overrightarrow{t_a}|^2 \times |\overrightarrow{t_b}|^2} \quad (8)$$

gde su $\overrightarrow{t_a}$ i $\overrightarrow{t_b}$ n-dimenzionalni vektori termina skupa $T = t_1, \dots, t_n$.

2. Džakard korelacioni koeficijent (*eng. Jaccard correlation coefficient*)

Džakard korelacioni koeficijent meri sličnost kao razlomak unije skupova objekata. Za tekst dokumenata, ovaj koeficijent upoređuje sumu težina zajedničkih termina sa sumom težina termina, koji su prisutni u bilo kom od dva dokumenta.

Formula Džakard korelacionog koeficijenta je:

$$SIM_J(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{\overrightarrow{t_a}.\overrightarrow{t_b}}{|\overrightarrow{t_a}|^2 + |\overrightarrow{t_b}|^2 - \overrightarrow{t_a}.\overrightarrow{t_b}} \quad (9)$$

Džakard korelacioni koeficijent je mera sličnost i kreće se između 0 i 1. On je 1 kada je $t_a = t_b$ i 0 kada su t_a i t_b potpuno različiti, gde 1 znači da su dva vektora ista i 0 znači da su ona potpuno različita. Odgovarajuća mera sličnosti je: $D_j = 1 - SIM_j$.

3. Euklidova distanca (*eng. Euclidean distance*)

Euklidova distanca je standardna metrika za geometrijske probleme i to je obično rastojanje između dve tačke. To je merljivo u dvodimenzionalnom ili trodimenzionalnom prostoru.

Data dva dokumenta d_a i d_b su predstavljena vektorima termina t_a i t_b respektivno, i Euklidova distanca se može predstaviti:

$$D_E(\overrightarrow{t_a}, \overrightarrow{t_b}) = \left(\sum_{t=1}^{n} |w_{t,a} - w_{t,b}|^2\right)^{\frac{1}{2}} \quad (10)$$

gde je $T = t_1, \dots, t_n$ skup termina. Euklidova distanca koristi TF-IDF vrednost kao težinu termina, gde je $w_{t,a} = tf - idf(d_a, t)$.

6.1. Prikaz tekstualnih dokumenata u vektorskom obliku

Krivični zakonik Republike Srbije se sastoji od 36 glava koji sadrže 432 člana. Tri člana zakona koja se odnose na telesne povrede data su u trinaestoj glavi koja se zove „Krivična dela protiv života i tela“ numerisani kao: Član 121, Član 122 i u četrnaestoj glavi koja se zove „Krivična dela protiv sloboda i prava čoveka i građanina“ numerisan kao: Član 135. Sadržaj tih članova dat je u Krivičnom zakoniku Republike Srbije na sledeći način:

Teška telesna povreda

Član 121

(1) Ko drugog teško telesno povredi ili mu zdravlje teško naruši, kazniće se zatvorom od šest meseci do pet godina.

(2) Ko drugog teško telesno povredi ili mu zdravlje naruši tako teško da je usled toga doveden u opasnost život povređenog ili je uništen ili trajno i u znatnoj meri oštećen ili oslabljen neki važan deo njegovog tela ili važan organ ili je prouzrokovana trajna nesposobnost za rad povređenog ili trajno i teško narušenje njegovog zdravlja ili unakaženost, kazniće se zatvorom od jedne do osam godina.

(3) Ako je usled dela iz st. 1. i 2. ovog člana nastupila smrt povređenog lica, učinilac će se kazniti zatvorom od dve do dvanaest godina.

(4) Ko delo iz st. 1. i 2. ovog člana učini iz nehata, kazniće se zatvorom do tri godine.

(5) Ko delo iz st. 1. do 3. ovog člana učini na mah, doveden bez svoje krivice u jaku razdraženost napadom, zlostavljanjem ili teškim vređanjem od strane povređenog, kazniće se za delo iz stava 1. zatvorom do tri godine, za delo iz stava 2. zatvorom od tri meseca do četiri godine, a za delo iz stava 3. zatvorom od šest meseci do pet godina.

(6) Ako je delo iz stava 1. ovog člana učinjeno prema maloletnom licu ili bremenitoj ženi ili licu koje obavlja poslove od javnog značaja, učinilac će se kazniti zatvorom od jedne do osam godina, za delo iz stava 2. ovog člana zatvorom od dve do dvanaest godina, a za delo iz stava 3. ovog člana zatvorom od pet do petnaest godina.

Laka telesna povreda

Član 122

(1) Ko drugog lako telesno povredi ili mu zdravlje lako naruši, kazniće se novčanom kaznom ili zatvorom do jedne godine.

(2) Ako je takva povreda nanesena oružjem, opasnim oruđem ili drugim sredstvom podobnim da telo teško povredi ili zdravlje teško naruši, učinilac će se kazniti zatvorom do tri godine.

(3) Sud može učiniocu dela iz stava 2. ovog člana izreći sudsku opomenu, ako je učinilac bio izazvan nepristojnim ili grubim ponašanjem oštećenog.

(4) Gonjenje za delo iz stava 1. ovog člana preduzima se po privatnoj tužbi.

Prinuda

Član 135

(1) Ko drugog silom ili pretnjom prinudi da nešto učini ili ne učini ili trpi, kazniće se zatvorom do tri godine.

(2) Ko delo iz stava 1. ovog člana učini na svirep način ili pretnjom ubistvom ili teškom telesnom povredom ili otmicom, kazniće se zatvorom od šest meseci do pet godina.

(3) Ako je usled dela iz st. 1. i 2. ovog člana nastupila teška telesna povreda ili druge teške posledice,učinilac će se kazniti zatvorom od jedne do deset godina.

(4) Ako je usled dela iz st. 1. i 2. ovog člana nastupila smrt prinuđenog lica ili je delo izvršeno od strane grupe,učinilac će se kazniti zatvorom od tri do dvanaest godina.

(5) Ako je delo iz st. 1. i 2. ovog člana izvršeno od strane organizovane kriminalne grupe, učinilac će se kazniti zatvorom od pet do petnaest godina.

Dati članovi Krivičnog zakonika Republike Srbije u istraživanju koje je predmet doktorske disertacije, su predloženom sistemu predstavljeni kao tri posebna tekstualna dokumenta. Ovi dokumenti su ujedno i osnovni dokumenti za poređenje sa pitanjima građana u vezi s telesnim povredama iz Krivičnog zakonika Republike Srbije. Na osnovu analize najfrekventnijih reči u sva tri dokumenta dobijen je rezultat prikazan u tabeli 6.1.

Termin	tf	idf	tf*idf
delo	8	0.7123	5.6984
učinilac	3	0.7123	2.1369
kazniće	2	0.7123	1.4246
zatvorom	5	0.7123	3.5615
učini	3	1	3
telesno	1	1	1
teško	2	1	2
povreda	1	1.9163	1.9163

Tabela 6.1: Analiza najfrekventnijih reči u tri dokumenta

Nakon procesa normalizacije primenom 4-gram metode od osam najzastupljenih reči po kriterijumu tf*idf dobijen je skup od sedaam reči: delo, učin, kazn, zatv, tele, tešk i povr. Kako ovaj skup reči nije dovoljan za dalju analizu korišćena su pitanja građana sa portala PRO BONO i tekstovi dnevnih novena Blic koji su tagovani kao telesne povrede. Rezultati frekventnosti reči pokazuju da je početni skup proširen sa sedam na 31 reč.

Novi skup reči dat u tabeli 6.2:

R.br.	Reč
1	delo
2	učin*
3	kazn*
4	zatv*
5	tele*
6	tešk*
7	povr*
8	prot*
9	prij*
10	poli*
11	kriv*
12	napa*
13	podn*
14	udar*
15	post*
16	nane*
17	sumn*
18	lake
19	osno*
20	javn*
21	osum*
22	pril*
23	sasl*
24	poku*
25	uhap*
26	zadr*
27	prin*
28	saop*
29	upra*
30	tuži*
31	dogo*

Tabela 6.2: Rezultati frekventnosti reči

U narednom koraku, vršen je proces grupisanja reči po članovima, a na osnovu kriterijuma datog sledećim nizom koraka:

a) Prikupljen je skup od 10 inicijalnih pitanja sa portala PRO BONO čiji su odgovori direktno vezani sa 3 pomenuta člana Krivičnog zakonika Republike Srbije. Za svako pitanje formiran je poseban skup reči iz gore definisanog skupa reči.
b) Prikupljen je skup od 45 pitanja sa istog portala radi uočavanja pojavljivanja reči definisanih u tabeli X (31), a koja su kao rezultat imala ponuđene članove Krivičnog zakonika Republike Srbije.

c) Prikupljen je skup tekstova koje je administrator veb-sajta Blica označio kao telesne povrede.
d) Klasterovanje je urađeno prebrojavanjem pojavljivanja reči u skupu a) i skupu b) i dobijen je reprezentativan skup za date članove prikazane u tabeli 6.3.

Član 121		**Član 122**		**Član 135**
kazn*		tele*		prij*
zatv*		poli*		poli*
tešk*		napa*		kriv*
post*		udar*		post*
javn*		post*		
poku*		nane*		
tuži*		učin*		
pril*		kazn*		
delo*		zatv*		
tele*		kriv*		
povr*		javn*		
nane*		povr*		
sumn*		lake		
uhap*		prij*		
		tešk*		
		poku*		
		tuži*		

Tabela 6.3: Reprezentativni skupovi za tri člana krivičnog zakonika

Konačan skup reči je prikazan u vektorskom obliku, za svaki član kao BoC:

- Član 121 [kazn,zatv,tešk,post,javn,poku,tuži,pril,delo,tele,povr, nane, sumn,uhap]
- Član 122 [tele,poli,napa,udar,post,nane,učin,kazn,zatv,kriv,javn,povr,lake, prij,tešk,poku,tuži]
- Član 135 [prij,poli,kriv,post]

6.2. Provera preciznosti predloženog sistema

Radi provere preciznosti predloženog sistema prikupljeno je 540 upita iz oblasti Krivičnog zakonika Republike Srbije. Nakon obrade ovih upita sistem je odmah eliminisao 130 pitanja koja se ne odnose na članove za koje je sistem urađen. Za dalju analizu i proveru sistema korišćeno je 410 upita.

Na slici 6.4 prikazan je način skladištenja 540 upita u posebnim tekstualnim dokumentima. Upiti dati kao tekstualni dokumenti su transformisani u niz vektora pomoću Apache Lucene. Primer ovog procesa (koraka) dat je kroz prvih 10 upita:

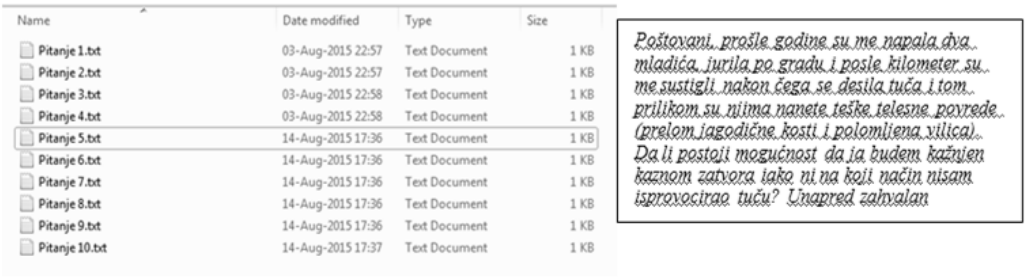


Tabela 6.4: Deset pripremljenih upita

Primer 1:

Upit KP 1: *„Poštovani, zanima me kada i kako zastareva delo urađeno pre više od 10 godina? U pitanju je teška telesna povreda. Ja sam se potukao sa jednim momkom. Pošto sam ga udario vema nezgodno on je pao i onda sam ga ja u besu polomio. Još se vodi postupak i nije rešeno ništa. Koliko je vremena potrebno da sve to zastari i kako?“*

Oznaka upita: Upit KP 1

Vektorski oblik upita KP 1: {delo, tele, tešk, povr, udar, post}

Član 121 {kazn, zatv, tešk, post, javn, poku, tuži, pril, delo, tele, povr, nane, sumn, uhap}

	delo	tele	tešk	povr	udar	post	kazn	zatv	javn	poku	tuži	pril	nane	sumn	uhap
v1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
v2	8	2	6	6	0	0	6	10	1	0	0	0	0	0	0

Tabela 6.5: Vektori upita KP 1 i člana 121 Krivičnog zakonika

Član 122 {tele, poli, napa, udar, post, nane, učin, kazn, zatv, kriv, javn, povr, lake, prij, tešk, poku, tuži}

	delo	tele	tešk	povr	udar	post	poli	napa	nane	učin	kazn	zatv	kriv	javn	lake	prij	poku	tuži
v1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
v2	1	1	2	3	0	0	0	0	1	3	3	2	0	0	0	0	0	0

Tabela 6.6: Vektori upita KP 1 i člana 122 Krivičnog zakonika

Član 135 { prij, poli, kriv, post}

	delo	tele	tešk	povr	udar	post	prij	poli	kriv
v1	1	1	1	1	1	1	0	0	0
v2	3	2	3	2	0	0	0	0	0

Tabela 6.7: Vektori upita KP 1 i člana 135 Krivičnog zakonika

Primer 2:

Upit KP 2: *„Kod nas u Kruševcu svaki advokat priča drugačije, tako da ja za godinu dana nisam uspela da saznam tačnu informaciju, a to mi mnogo znači. Moj muž je osuđen na godinu i osam meseci za delo teška telesna povreda. On nije otišao na izvršenje kazne,*

a trebalo je da ode još pre godinu dana. Naravno, podignuta je poternica, a on je u bekstvu. Zanima me kolika je apsolutna zastarelost za njegovu kaznu teška telesna povreda iz člana 121 krivičnog zakonika i od kada se računa? Hvala puno unapred."

Upit KP 2 {delo, kazn, tele, tešk, povr, kriv}

Član 121 {kazn, zatv, tešk, post, javn, poku, tuži, pril, delo, tele, povr, nane, sumn, uhap}

	delo	kazn	tele	tešk	povr	kriv	zatv	post	javn	poku	tuži	pril	nane	sumn	uhap
v1	1	2	2	2	2	1	0	0	0	0	0	0	0	0	0
v2	8	6	2	6	6	1	10	0	1	0	0	0	0	0	0

Tabela 6.8: Vektori upita KP 2 i člana 121 Krivičnog zakonika

Član 122 {tele, poli, napa, udar, post, nane, učin, kazn, zatv, kriv, javn, povr, lake, prij, tešk, poku, tuži}

	delo	kazn	tele	tešk	povr	kriv	poli	napa	udar	post	nane	učin	zatv	javn	lake	prij	poku	tuži
v1	1	2	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0
v2	1	3	1	2	3	0	0	0	0	1	1	3	2	0	0	0	0	0

Tabela 6.9: Vektori upita KP 2 i člana 122 Krivičnog zakonika

Član 135 { prij, poli, kriv, post}

	delo	kazn	tele	tešk	povr	kriv	prij	poli	post
v1	1	2	2	2	2	1	0	0	0
v2	3	5	2	3	2	0	0	0	0

Tabela 6.10: Vektori upita KP 2 i člana 135 Krivičnog zakonika

Primer 3:

Upit KP 3: *„Imam u toku sudjenje iz cl 122/ krivičnog zakonika znaci lake telesne povrede, imam za koji mesec glavni pretres. Momak sa koji se poznajem onako naneo mi je lake telesne povrede. Zvao sam policiju i oni su ga priveli u stanicu. Dali smo izjave i sada je glavni pretres. E sad molim vas ako mozete recite mi kad to sudjenje zastareva u mom slucaju, posto se vuce negde sigurno vec 4 godine ako ne i vise. Srdacan pozdrav."*

Upit KP 3 {tele, povr, poli, kriv, post, nane, lake}

Član 121 {kazn, zatv, tešk, post, javn, poku, tuži, pril, delo, tele, povr, nane, sumn, uhap}

	tele	povr	poli	kriv	post	nane	lake	kazn	zatv	tešk	javn	poku	tuži	pril	delo	sumn	uhap
v1	2	2	1	1	1	1	2	0	0	1	0	0	0	0	1	0	0
v2	2	6	0	1	0	0	0	6	10	6	1	0	0	0	8	0	0

Tabela 6.11: Vektori upita KP 3 i člana 121 Krivičnog zakonika

Član 122 {tele, poli, napa, udar, post, nane, učin, kazn, zatv, kriv, javn, povr, lake, prij, tešk, poku, tuži}

	tele	povr	poli	kriv	post	nane	lake	napa	udar	učin	kazn	zatv	javn	prij	tešk	poku	tuži
v 1	2	2	1	1	1	1	2	0	0	0	0	0	0	0	1	0	0
v 2	1	3	0	0	0	1	0	0	0	3	3	2	0	0	2	0	0

Tabela 6.12: Vektori upita KP 3 i člana 122 Krivičnog zakonika

Član 135 { prij, poli, kriv, post}

	tele	povr	poli	kriv	post	nane	lake	prij	poli
v1	2	2	1	1	1	1	2	0	1
v2	2	2	0	0	0	0	0	0	0

Tabela 6.13: Vektori upita KP 3 i člana 135 Krivičnog zakonika

Primer 4:

Upit KP 4: *„Poštovani moj komšija mi je na silu i uz pretnju uzeo 1000 evra. Posle toga smo imali više rasprava o tome i nije mi vratio novac. Ovo je delo iz člana 135 Krivičnog zakonika.Pošto se to desilo pre godinu ipo dana, interesuje me da li je ovo zastarelo? Da li treba da odem u policiju i prijavim ga?“*

Upit KP 4 {delo, prij, poli, kriv}

Član 121 {kazn, zatv, tešk, post, javn, poku, tuži, pril, delo, tele, povr, nane, sumn, uhap}

	delo	prij	poli	kriv	kazn	zatv	tešk	post	javn	poku	tuži	pril	tele	povr	nane	sumn	uhap
v 1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
v 2	8	0	0	1	6	10	6	0	1	0	0	0	2	6	0	0	0

Tabela 6.14: Vektori upita KP 4 i člana 121 Krivičnog zakonika

Član 122 {tele, poli, napa, udar, post, nane, učin, kazn, zatv, kriv, javn, povr, lake, prij, tešk, poku, tuži}

	delo	prij	poli	kriv	tele	napa	udar	post	nane	učin	kazn	zatv	javn	povr	lake	tešk	poku	tuži
v 1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v 2	1	0	0	0	1	0	0	0	1	3	3	2	0	3	0	2	0	0

Tabela 6.15: Vektori upita KP 4 i člana 122 Krivičnog zakonika

Član 135 { prij, poli, kriv, post}

	delo	prij	poli	kriv	post
v1	1	1	1	1	0
v2	3	0	0	0	0

Tabela 6.16: Vektori upita KP 4 i člana 135 Krivičnog zakonika

Primer 5:

Upit KP 5: *„Poštovani, već pet godina sudim se u vezi sa teškom telesnom povredom. Pošto se bliži puna peta godina kako se sudimo, a suđenja se odlažu jer uvek neko izostane, u pitanju je više ljudi za isto delo, zanima me da li slučaj može da zastari i da li može da nas osudi u odsustvu, odnosno, da donese presudu bez prisustva nekog od okrivljenih? Dobili smo na zadnjem ročištu advokate po službenoj dužnosti, ali niko im nije potpisao punomoćje. U nadi da ćete mi odgovorti, srdačan pozdrav“*

Upit KP 5 {delo, tele, tešk, povr}

Član 121 {kazn, zatv, tešk, post, javn, poku, tuži, pril, delo, tele, povr, nane, sumn, uhap}

	delo	tele	tešk	povr	kazn	zatv	post	javn	poku	tuži	pril	nane	sumn	uhap
v1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
v2	8	2	6	6	6	10	0	1	0	0	0	0	0	0

Tabela 6.17: Vektori upita KP 5 i člana 121 Krivičnog zakonika

Član 122 {tele, poli, napa, udar, post, nane, učin, kazn, zatv, kriv, javn, povr, lake, prij, tešk, poku, tuži}

	delo	tele	tešk	povr	poli	napa	udar	post	nane	učin	kazn	zatv	kriv	javn	lake	prij	poku	tuži
v1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
v2	1	1	2	3	0	0	0	0	1	3	3	2	0	0	0	0	0	0

Tabela 6.18: Vektori upita KP 5 i člana 122 Krivičnog zakonika

Član 135 { prij, poli, kriv, post}

	delo	tele	tešk	povr	prij	poli	kriv	post
v1	1	1	1	1	0	0	1	0
v2	3	2	3	2	0	0	0	0

Tabela 6.19: Vektori upita KP 5 i člana 135 Krivičnog zakonika

Primer 6:

Upit KP 6: *„Poštovani, po rešenju iz X godine, proglašen sam krivim jer sam udario jednog čoveka. Dobio je lake telesne povrede, a mislim da mu nije ništa bilo. Kazna je izrečena u visini od X dinara i da platim sudske troškove. Od tada je prošlo skoro X godina. Nadam se zastarevanju. Da li je to moguće i kada? Unapred zahvalan.“*

Upit KP 6 {kazn, tele, povr, kriv, udar, lake}

Član 121 {kazn, zatv, tešk, post, javn, poku, tuži, pril, delo, tele, povr, nane, sumn, uhap}

	kazn	tele	povr	kriv	udar	lake	zatv	tešk	post	javn	poku	tuži	pril	delo	nane	sumn	uhap
v1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
v2	6	2	6	1	0	0	10	6	0	1	0	0	0	8	0	0	0

Tabela 6.20: Vektori upita KP 6 i člana 121 Krivičnog zakonika

Član 122 {tele, poli, napa, udar, post, nane, učin, kazn, zatv, kriv, javn, povr, lake, prij, tešk, poku, tuži}

	kazn	tele	povr	kriv	udar	lake	poli	napa	post	nane	učin	zatv	javn	prij	tešk	poku	tuži
v 1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
v 2	3	1	3	0	0	0	0	0	0	1	3	2	0	0	2	0	0

Tabela 6.21: Vektori upita KP 6 i člana 122 Krivičnog zakonika

Član 135 { prij, poli, kriv, post}

	kazn	tele	povr	kriv	udar	lake	prij	poli	post
v1	1	1	1	1	1	1	0	0	0
v2	5	2	2	0	0	0	0	0	0

Tabela 6.22: Vektori upita KP 6 i člana 135 Krivičnog zakonika

Primer 7:

Upit KP 7: *„Poštovani, 2010. godine mi je jedna osoba nanela teške telesne povrede (slomljen nos, jagodična kost), a nisam podneo krivičnu prijavu na insistiranje zajedničkih poznanika pošto je on već bio u zatvoru. Međutim, ispostavilo se da ću najverovatnije morati na operaciju nosa zbog velike devijacije koja mi otežava disanje (nakon koje ću morati po rečima lekara sa kojima sam se savetovao da se oporavljam i do godinu dana), pa sam odlučio da podnesem krivičnu prijavu. Koliki je period zastarevanja ovakvog slučaja po našem trenutnom zakonu i kolika bi bila njegova kazna?“*

Upit KP 7{kazn, zatv, tele, tešk, povr, prij, kriv, podn, nane}

Član 121 {kazn, zatv, tešk, post, javn, poku, tuži, pril, delo, tele, povr, nane, sumn, uhap}

	kazn	zatv	tele	tešk	povr	prij	kriv	podn	nane	post	javn	poku	tuži	pril	delo	sumn	uhap
v 1	1	1	1	1	1	2	2	2	1	1	0	0	0	0	0	0	0
v 2	6	10	2	6	6	0	0	0	0	0	1	0	0	0	8	0	0

Tabela 6.23: Vektori upita KP 7 i člana 121 Krivičnog zakonika

Član 122 {tele, poli, napa, udar, post, nane, učin, kazn, zatv, kriv, javn, povr, lake, prij, tešk, poku, tuži}

	kazn	zatv	tele	tešk	povr	prij	kriv	podn	nane	poli	napa	udar	post	učin	javn	lake	poku	tuži
v1	1	1	1	1	1	2	2	2	1	0	0	0	1	0	0	0	0	0
v2	3	2	1	2	3	0	0	0	1	0	0	0	0	3	0	0	0	0

Tabela 6.24: Vektori upita KP 7 i člana 122 Krivičnog zakonika

Član 135 { prij, poli, kriv, post}

	kazn	zatv	tele	tešk	povr	prij	kriv	podn	nane	poli	post
v1	1	1	1	1	1	2	2	2	1	0	1
v2	5	5	2	3	2	0	0	0	0	0	0

Tabela 6.25: Vektori upita KP 7 i člana 135 Krivičnog zakonika

Primer 8:

Upit KP 8: *„Dobio sam presudu i osudjen sam zajedno sa rodjenim bratom, kao saucesnik, ja na uslovnu kaznu, a on na kaznu zatvora od X meseci, osudjeni smo za lake telesne povrede, To je član 122 Krivičnog zakonika, učinjeno sa opasnim orudjem koje može opasno da povredi. Dogadjaj se dogodio u aprilu 2007. god, a ja sam dobio presudu pre par dana. Da li moze te da mi kazete da li je to apsolutno zastarelo?“*

Upit KP 8 {učin, kazn, zatv, tele, povr, kriv, lake, dogo}

Član 121 {kazn, zatv, tešk, post, javn, poku, tuži, pril, delo, tele, povr, nane, sumn, uhap}

	učin	kazn	zatv	tele	povr	kriv	lake	dogo	tešk	post	javn	poku	tuži	pril	delo	nane	sumn	uhap
v1	1	2	1	1	2	1	1	1	0	0	0	0	0	1	0	0	0	0
v2	5	6	10	2	6	1	0	0	6	0	1	0	0	0	8	0	0	0

Tabela 6.26: Vektori upita KP 8 i člana 121 Krivičnog zakonika

Član 122 {tele, poli, napa, udar, post, nane, učin, kazn, zatv, kriv, javn, povr, lake, prij, tešk, poku, tuži}

	učin	kazn	zatv	tele	povr	kriv	lake	dogo	poli	napa	udar	post	nane	javn	prij	tešk	poku	tuži
v1	1	2	1	1	2	1	1	1	0	0	0	0	0	0	0	0	0	0
v2	3	3	2	1	3	0	0	0	0	0	0	0	1	0	0	2	0	0

Tabela 6.27: Vektori upita KP 8 i člana 122 Krivičnog zakonika

Član 135 { prij, poli, kriv, post}

	učin	kazn	zatv	tele	povr	kriv	lake	dogo	prij	poli	post
v1	1	2	1	1	2	1	1	1	0	0	0
v2	6	5	5	2	2	0	0	0	0	0	0

Tabela 6.28: Vektori upita KP 8 i člana 135 Krivičnog zakonika

Primer 9:

Upit KP 9: *„Moj suprug je u julu 2009. godine imao saobraćajnu nezgodu pod dejstvom alkohola 1,19%. Udario je auto ispred njega. Neposredno posle udesa, izašao je iz auta pretukao vozača tog auta. Njemu je naneo tešku telesnu povredu i čovek je jedva preživeo. Posle saslušanja moj suprug je u decembru 2013. godine otišao u Ameriku tako da se odazvao ni jednom pozivu suda. Interesuje me kolika je apsolutna zastarelost u ovom slučaju? Unapred hvala.“*

Upit KP 9 {tele, tešk, povr, udar, nane, sasl}

Član 121 {kazn, zatv, tešk, post, javn, poku, tuži, pril, delo, tele, povr, nane, sumn, uhap}

	tele	tešk	povr	udar	nane	sasl	kazn	zatv	post	javn	poku	tuži	pril	delo	sumn	uhap
v1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
v2	2	6	6	0	0	0	6	10	0	1	0	0	0	8	0	0

Tabela 6.29: Vektori upita KP 9 i člana 121 Krivičnog zakonika

Član 122 {tele, poli, napa, udar, post, nane, učin, kazn, zatv, kriv, javn, povr, lake, prij, tešk, poku, tuži}

	tele	tešk	povr	udar	nane	sas l	poli	napa	post	učin	kazn	zatv	kriv	javn	lake	prij	poku	tuži
v 1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
v 2	1	2	3	0	1	0	0	0	3	0	3	2	0	0	0	0	0	0

Tabela 6.30: Vektori upita KP 9 i člana 122 Krivičnog zakonika

Član 135 { prij, poli, kriv, post}

	tele	tešk	povr	udar	nane	sasl	prij	poli	kriv	post
v1	1	1	1	1	1	1	0	0	0	0
v2	2	3	2	0	0	0	0	0	0	0

Tabela 6.31: Vektori upita KP 9 i člana 135 Krivičnog zakonika

Primer 10:

Upit KP 10: *„Poštovani prvostepenom presudom sam osuđen na prinudu u pokušaju iz čl. 135 st. 2 KZ-a u vezi člana 30. Pošto je za ovo krivično delo zaprećena kazna od 6 meseci do 5 godina zatvora, a pokušaj krivičnog dela je kažnjiv ako je zaprećena kazna*

zatvora od 5 godina i teža zanima me da li je sud mogao da me kazni u pokušaju s obzirom na zaprećenu kaznu za krivicno delo koje mi se stavlja na teret?“

Upit KP 10 {delo, kazn, zatv, kriv, poku, prin}

Član 121 {kazn, zatv, tešk, post, javn, poku, tuži, pril, delo, tele, povr, nane, sumn, uhap}

	delo	kazn	zatv	kriv	poku	prin	tešk	post	javn	tuži	pril	tele	povr	nane	sumn	uhap
v1	2	4	2	3	3	1	0	0	0	0	0	0	0	0	0	0
v2	8	6	10	1	0	0	6	0	1	0	0	2	6	0	0	0

Tabela 6.32: Vektori upita KP 10 i člana 121 Krivičnog zakonika

Član 122 {tele, poli, napa, udar, post, nane, učin, kazn, zatv, kriv, javn, povr, lake, prij, tešk, poku, tuži}

	delo	kazn	zatv	kriv	poku	prin	tele	poli	napa	udar	post	nane	učin	javn	povr	lake	prij	tešk	tuži
v1	2	4	2	3	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
v2	8	3	2	0	0	0	1	0	0	0	3	1	0	0	3	0	0	2	0

Tabela 6.33: Vektori upita KP 10 i člana 122 Krivičnog zakonika

Član 135 { prij, poli, kriv, post}

	delo	kazn	zatv	kriv	poku	prin	prij	poli	post
v1	2	4	2	3	3	1	0	0	1
v2	3	5	5	0	0	2	0	0	0

Tabela 6.34: Vektori upita KP 10 i člana 135 Krivičnog zakonika

U cilju utvrđivanja odgovarajuće mere sličnosti u našem predloženom sistemu, sproveli smo eksperiment koristeći sličnost sve tri gore navedene mere (tekst tri člana Krivičnog zakonika) sa 10 upita u obliku tekstualnih dokumenata iz oblasti Krivičnog zakonika. Rezultati ove analize su dati u tabeli 6.35.

	Član 121	**Član 122**	**Član 135**	**Ekspert**
Upit KP 1	sim(Cos)= 0.539644 sim(Jacc.)= **1.000000** sim(Eucl.)= 15.459625	sim(Cos)= 0.463586 sim(Jacc.)= 0.800000 sim(Eucl.)= 5.477226	sim(Cos)= **0.800641** sim(Jacc.)= **1.000000** sim(Eucl.)= **3.464102**	Član 121
Upit KP 2	sim(Cos)= 0.692688 sim(Jacc.)= 0.750000 sim(Eucl.)= 14.071247	sim(Cos)= 0.717109 sim(Jacc.)= **0.800000** sim(Eucl.)= 4.358899	sim(Cos)= **0.891133** sim(Jacc.)= 0.666667 sim(Eucl.)= **3.872983**	Član 121
Upit KP 3	sim(Cos)= 0.438231 sim(Jacc.)= 0.833333 sim(Eucl.)= 15.297059	sim(Cos)= 0.438599 sim(Jacc.)= **0.909091** sim(Eucl.)= 5.656854	sim(Cos)= **0.685994** sim(Jacc.)= 0. 750000 sim(Eucl.)= **3.000000**	Član 122
Upit KP 4	sim(Cos)= 0.269892 sim(Jacc.)= 0.900000 sim(Eucl.)= 16.248077	sim(Cos)= **0.081111** sim(Jacc.)= 0.909091 sim(Eucl.)= 6.324555	sim(Cos)= 0.500000 sim(Jacc.)= **1.000000** sim(Eucl.)= **2.645751**	Član 135
Upit KP 5	sim(Cos)= 0.660926 sim(Jacc.)= **1.000000** sim(Eucl.)= 15.394804	sim(Cos)= 0.507833 sim(Jacc.)= 0.777778 sim(Eucl.)= 5.385165	sim(Cos)= **0.877058** sim(Jacc.)= **1.000000** sim(Eucl.)= **3.316625**	Član 121
Upit KP 6	sim(Cos)= 0.367277 sim(Jacc.)= 0.900000 sim(Eucl.)= 15.937377	sim(Cos)= 0.469809 sim(Jacc.)= 0.900000 sim(Eucl.)= 5.385165	sim(Cos)= **0.639602** sim(Jacc.)= **1.000000** sim(Eucl.)= **4.582576**	Član 122
Upit KP 7	sim(Cos)= 0.413528 sim(Jacc.)= **1.000000** sim(Eucl.)= 15.362291	sim(Cos)= 0.452589 sim(Jacc.)= 0.818182 sim(Eucl.)= **5.656854**	sim(Cos)= **0.476469** sim(Jacc.)= **1.000000** sim(Eucl.)= 7.211103	Član 121
Upit KP 8	sim(Cos)= 0.622992 sim(Jacc.)= **0.916667** sim(Eucl.)= 15.297059	sim(Cos)= **0.790875** sim(Jacc.)= 0.900000 sim(Eucl.)= **3.872983**	sim(Cos)= 0.744279 sim(Jacc.)= 0.875000 sim(Eucl.)= 7.348469	Član 122
Upit KP 9	sim(Cos)= 0.368166 sim(Jacc.)= **1.000000** sim(Eucl.)= 15.968719	sim(Cos)= 0.497468 sim(Jacc.)= 0.777778 sim(Eucl.)= 5.385165	sim(Cos)= **0.693103** sim(Jacc.)= **1.000000** sim(Eucl.)= **3.000000**	Član 121
Upit KP 10	sim(Cos)= 0.576214 sim(Jacc.)= **1.000000** sim(Eucl.)= 13.96424	sim(Cos)= 0.485574 sim(Jacc.)= 0.909091 sim(Eucl.)= 8.944272	sim(Cos)= **0.72175** sim(Jacc.)= **1.000000** sim(Eucl.)= **5.567764**	Član 135

Tabela 6.35: Zbirna tabela

Kako bi se odabrala mera sličnosti, dobijeni rezultati sa tri vrednosti sličnosti, upoređene su sa sličnosti koje su date gore (Džakard korelacioni koeficijent). Pokazalo se da upit Upit KP1 ima najveću sličnost sa članom 121, gde je vrednost sličnosti 1.000000. Ako uzmemo u obzir dobijene sličnosti u koloni člana 121 može se uočiti da Džakard korelacioni koeficijent jedini daje tačnu vrednost, pa je zato za ovaj upit on uzet kao referentna mera sličnosti. Na isti način je analizirano preostalih devet upita. Rezultati pokazuju da je referentna mera sličnosti u ovom slučaju Džakard korelacioni koeficijent, pa se zato ovde uzima kao referentna za obračun algoritma u poglavlju 5 na slici 5.15.

Provera ispravnosti (tačnosti) predloženog algoritma u ovoj doktorskoj disertaciji na osnovu usklađivanja sa „zlatnim“ standardom naspram realnog predviđanja:

$$Preciznost\ (eng. Precision) = \frac{Relevantno\ Pronađeno}{Pronađeno} \tag{11}$$

$$Odziv\ (eng. Recall) = \frac{Relevantno\ Pronađeno}{Relevantno} \tag{12}$$

Akcija / Dokumenat	**Pronađeno (eng. Retrieved)**	**Ne Pronađeno (eng. Not Retrieved)**
Relevantno (eng. Relevant)	Relevantno Pronađeno	Relevantno Odbačeno (eng. Rejected)
Ne Relevantno (eng. Not Relevant)	Irelevantno (eng. Irelevant) Pronađeno	Irelevantno Odbačeno

Tabela 6.36: Merenje ocene rezultata istraživanja: Pregled klasifikacija

$$F_{i\ (i=1,n)} = \frac{2*preciznost_i*odziv_i}{preciznost_i+odziv_i} \tag{13}$$

$$F_{Prosečno} = \frac{F_1+F_2+\cdots+F_n}{N} \tag{14}$$

Preciznost, odziv i F_1 zavisi samo od tačnih pozitivnih, odnosno, onih pozitivnih uzoraka koji su u skladu sa „zlatnim“ standardima. U jednojezičnom usklađivanju, pozitivni uzorci su oni tokeni koji se uklapaju, a negativni uzorci su oni tokeni koji se ne uklapaju sa „zlatnim“ standardima. Obično se vodi računa samo o tome da li su pravilno postavljeni oni tokeni koji treba da budu usklađeni sa „zlatnim“ standardima, tako da je mera F_1 za t pozitivnih termova.

Kako bi se realizovalo „ispravno odbacivanje“ (istina negativna), tačnost se izračunava na sledeći način:

$$Tačnost\ (eng. Accuracy) = \frac{TP+TN}{TP+TN+FP+FN} \tag{15}$$

gde je:

TP – Tačno Pozitivno (eng. True Positive);

TN – Tačno Negativno (eng. True Negative);

FP – Netačno Pozitivno (eng. False Positive);

FN – Netačno Negativno (eng. False Negative).

Tačnost je jednaka: Tačno Pozitivno i Tačno Negativno. Klaster prepoznaje i Pozitivne i Negativne uzorke. Konkretno oko usklađivanja, gde većina tokena nije usklađena, vrednost tačnosti će biti vrlo visoka u celini, a razliku je teško odrediti. U ovom slučaju uzimamo u obzir samo F_1 na pozitivne (usklađene) uzorke.

Broj ekstrahovanih rezultata upita

Za testiranje algoritma korišćena su 410 upita. U tabeli 6.37 su predstavljeni rezultati koji su dobijeni na osnovu predstavljenog algoritma i paralelno sa njima rezultate koje je dao ekspert iz oblasti krivičnog prava.

Preciznost	75,71 %
Odziv	57,00 %
$F_{Prosečno}$	0.6936
TP	287
FP	205
FN	123
Tačnost	46,66 %

Tabela 6.37: Rezultati

Eksperiment pokazuje da predstavljeni sistem u doktorskoj disertaciji može mapirati upit sa relevantnim članom Krivičnog zakonika Republike Srbije sa prosečnom preciznošću od 75,71 %. Obično je upii građana u e-Upravi vrlo kratak i teško ga je klasifikovati korišćenjem tradicionalnih BoW tehnika. Neophodno je obraditi dokumente ili delove dokumenata da bi se dobili tipični koncepti, jer upiti mogu biti sličniji naslovima dokumenata. Ovo štedi mnogo vremena prilikom obrade velike količine dugih tekstova, kako bi se omogućio konceptualni model dokumenata prezentovan kao predefinisana klasa da daje brze preporuke predstavljene kao odgovor.

Što se tiče ovakvog koncepta koji koristi podsadržaje i koji je u stanju da proširi svoj dijapazon direktno sa nivoa podnaslova, potrebno je veliko predznanje za konvertovanje termina u koncepte. Da bi se odgovorilo na adekvatan način, gradi se konceptualni model za svaki član Krivičnog zakonika Republike Srbije. Zahteva se korišćenje mnogih izvora, tako da iz ovog istraživanja proizilazi preporuka za korišćenje

postojećih novinskih članaka na veb-sajtovima koji su već grupisani od strane čitalaca i klasifikovani pomoću tag oznaka. Ovaj princip dopune relevantnih termina za postojeće nestruktuirane dokumente je neophodan kako bi što preciznije vektorski opisali dati dokument, u slučaju nepostojanja rečnika za datu oblast. Ovakav okvir je pogodan za mnoge buduće elektronske servise, posebno one koji ne sadrže prethodno definisane odgovore, ponuđene u formi relevantnih dokumenata. Isto tako, tokom istraživanja se došlo do najboljih rezultata korišćenjem Džakard korelacionog koeficijenta, zatim pogodne mere sličnosti za upoređivanje kratkih tekstova, što može biti dobro rešenje kada su dokumenti u obliku kratkog teksta, kao što je proces klasterizacije za korpuse koji sadrže veliku količinu dokumenata sa kratkim tekstom.

7. Zaključak

Građani i predstavnici privrednih subjekata imaju stalnu potrebu za određenim informacijama u vezi usluga koje pružaju državni organi Vlade Republike Srbije. Oni obično i ne znaju, niti je neophodno da znaju ko je nadležan za pružanje konkretne informacije ili usluge, što za posledicu ima veliki broj poziva ka nenadležnim službama i gubljenje vremena. Kancelarije za brze odgovore u okviru Vlade omogućava lakšu i bržu komunikaciju i rešavanje zahteva građana da na jednom mestu dobiju potrebne informacije. Zainteresovani građani i predstavnici privrednih subjekata svoje upite mogu postavljati u elektronskoj formi, popunjavanjem aplikacionog formulara preko veb-portala. Odgovori na postavljena pitanja dobijaju se u roku od tri do pet radnih dana od strane stručnih lica.

Otvaranje Kancelarije za brze odgovore je omogućilo razvoj brojnih uslužnih veb-portala za građane i predstavnike privrednih subjekata. Tako je u Republici Srbiji 2014. godine, Zavod za informatiku i statistiku osmislio veb-aplikaciju pod nazivom Kancelarija za brze informacije privredi - BIC (https://portal.beograd.gov.rs/bic). Funkcije ove aplikacije su obrada pitanja, priprema odgovora, distribuiranje odgovora, kao i pregled podataka o pitanjima i odgovorima. Najveći broj pitanja odnosi se na nadležnosti *Biznis Info Centra* Privredne komore Beograda. *Biznis Info Centru* logistiku pružaju svi centri regionalnih komora i udruženja, kao i resursi Skupštine grada Beograda. Privrednici, preduzetnici i postojeći ili potencijalni investitori, svoja pitanja mogu dostaviti popunjavanjem upitnika na portalu. Dati portal je na usluzi postojećim i potencijalnim privrednicima sa teritorije grada Beograda, kao i onima koji žele da investiraju u razvoj Beograda, bez obzira u kojoj delatnosti posluju ili nameravaju da posluju. Cilj otvaranja ovih kancelarija je da domaći i inostrani privrednici dobijaju informacije potrebne za rešavanje problema vezanih za svoje poslovanje na brz i efikasan način. Oblasti na koje građani i predstavnici privrednih subjekata mogu dobiti odgovore preko ovog portala su: edukacija preduzetnika, zapošljavanje, zaštita intelektualne svojine, izdavanje raznih dozvola, kategorizacija smeštaja, letnje bašte, komunalna deltanost, lokalne takse, oglašavanje van poslovnog prostora, osnivanje privrednih subjekata, poreska politika, postavljanje privremenih objekata, pravna regulativa, uvoz/izvoz, upravljanje otpadom, carinski propisi, finanijska podrška za postojeća mala i srednja preduzeća, kao i za početnike u privatnom poslu.

Ako zaposleni u Kancelariji za brze odgovore ne mogu odmah da daju odgovor na postavljeni upit, tada oni prosleđuju e-mail upit nadležnom državnom organu, kako bi predstavnici državnih organa, nakon prijema i obrade pitanja, vratili odgovor Kancelariji za brze odgovore. Ovaj proces može i te kako da uspori dobijanje odgovora

u predviđenom roku. Zaposleni, pored dobrog poznavanja oblasti moraju da poznaju i sastav, organizaciju i nadležnosti organa Vlade, kao i evidenciju službi u konkretnom nadležnom organu.

U okviru svakog ministarstva, određeni *on-line* servisi mogu se definisati do sledećih nivoa sofisticiranosti [130]:

• nivo 1 – pružanje informacija: *on-line* informacija;

• nivo 2 – jednosmerna interakcija: pružanje informacija i preuzimanje obrazaca;

• nivo 3 – dvosmerna interakcija: *on-line* podnošenje obrazaca (autentifikacija);

• nivo 4 – transakcija: potpuna obrada predmeta, odluka, rešenje uz *on-line* plaćanje usluge (autorizacija).

• nivo 5 – personalizacija (automatsko pokretanje usluge ili upozoravanje korisnika da je vreme da pokrene npr. produženje lične karte ili vozačke dozvole).

Da bi poboljšali nivo 2 bilo kog veb-servisa e-Uprave u ovoj doktorskoj disertaciji data je konceptualna arhitektura modela razvoja sistema za brze odgovore koja uključuje pružanje informacija u vidu preporuka, tj. smernicu za preuzeimanje/pregled odgovarajućeg dokumenta. Jednosmernom interakcijom građani mogu dobiti osnovnu informaciju i smernicu za kompletno sagledavanje mogućih odgovora, a na osnovu postojećih dokumenata (zakona, odredbi i slično) u e-Upravi. Većina dokumenata je u vidu nestruktuiranih tekstualnih fajlova. U okviru automatskog prikazivanja strukture tekstualnih dokumenata, prva faza realizacije obuhvata kreiranje njihove logične strukture. Da bi kreirali logičku strukturu neophodno je sagledati opšta pravila ponašanja, društvene odnose u oblasti kojoj pripada dokument. Konkretnu materiju pravnog sistema čine zakoni, dok se opštiji karakter uređuje zakonikom. Tako na primer, ako dokument sadrži krivična dela on se logicki može svrstati u grupu dokumenata iz oblasti stvarnog, naslednog, porodičnog, obligacionog, radnog i krivičnog prava. S druge strane, niz podzakonskih akata, bliže definišu pojedine članove zakona, pa članove zakona možemo takođe grupisati po područjima. Ako razmatramo pojam fizički napad, on pripada podpodručju krivično delo protiv lica koji pripada području krivičnih dela u okviru područja Krivičnog prava. Tekstove u delu zakona, posmatrane kao članove, na taj način možemo grupisati i po pojmovima, pa tako za pojam telesna povreda možemo vezati tri člana zakona 121, 122 и 135 Krivičnog zakonika Republike Srbije.

U mnogim algoritmima mašinskog učenja tekstualni dokument je kompresovan u formu BoW. U doktorskoj disertaciji ovaj pojam je detaljno objašnjen i kroz tekst pitanja opisan pomoću modela BoW. Reprezentaciji teksta je predstavljena kao skup reči koje opisuju zadati tekst, pri čemu se gubi informacija o rasporedu reči u tekstu.

Ova metoda zasniva se na pridruživanju odabranih koncepata, posmatranog problema pa se u okviru ove doktorske disertacije sagledava tehnika klasifikacije tekstova na osnovu njihovog sadržaja primenom novog modela pod nazivom BoW. Ova metoda zasniva se na pridruživanju izdvojenih termina iz pitanja predefinisanim grupama koncepata koje predstavljaju dokumente, na osnovu kojih se izračunava mera pripadnosti termina i vrši pridruživanje pitanja nekoj grupi koncepata. Na osnovu dobijene liste koncepata, za svaki dokument (deo zakona, skup članova zakona) se formira globalna lista koncepata koja će predstavljati tu grupu/klasu. Ova lista se formira tako što se za reči koje su po svojoj učestalosti i specifičnosti važne za posmatranu klasu, u repozirorijumu (Probone, FAQ, blic,…) pronalaze koncepti koji će obuhvatiti jednu ili više takvih reči. Primer za koncept telesne povrede prikazan je u studiji slučaja. Ovom konceptu pridružen je 31 termin: delo, učin, kazn, zatv, tele, tešk, povr, prot, prij, poli, kriv, napa, podn, udar, post, nane, sumn, lake, osno, javn, osum, pril, sasl, poku, uhap, zadr, prin, saop, upra, tuži, dogo.

Kao rezultat parsiranja pitanja biblioteka tokenizer u okviru Apache *Lucene* tehnologije vraća strukturu koja sadrži sve reči koje se pojavljuju u tekstu sa brojem pojavljivanja svake od njih, a koja odgovara pomenutoj reprezentaciji teksta BoW. U predloženom sistemu za brze odgovore u doktorskoj disertaciji se pitanja građana posmatraju kao tekst, a *Apache Lucene* se koristi za pretraživanje teksta. Velika društvena mreža Twiter svoje servise upravo implementira korišćenjem Apache *Lucene* sistemom visokih performansi, koji služi za pretragu teksta i koji koristi metodu obrnutih indeksa. Poznati Wikia Search engine kombinuje u sebi prednosti softverske tehnologije i ljudske inteligencije. Kao automatizovani deo pretraživanja on se upravo bazira na tehnologiji pretraživačkih mašina otvorenog koda iza kojih stoji fondacija *Apache Software* (apache.org).

Kvalitet dobijenih informacija zavisi od kvaliteta procesa indeksiranja i pretraživanja pomoću pretraživača. U ovoj doktorskoj disertaciji korišćena je *Apache Lucene* biblioteka, kao skalabilna biblioteka za potpuno tekstualno pretraživanje. Ona je korišćena za analizu i indeksiranje tekstualnih sadržaj (članovi Krivičnog zakonika, upiti građana i drugi dokumenti), pretraživanje unutar kreiranih indeksa i prikaz rezultata pretraživanja za određeni upit. Pored toga, u *Lucene* biblioteka je obezbedila Analizer za srpski jezik, kao i izračunavanje TF-IDF vrednost.

Na osnovu prethodno izloženog, u doktorskoj disertaciji su ostvareni sledeći rezultati:

- Analizirana je mogućnost i ograničenja primene postojećih analitičkih metoda u predloženom sistemu. Posebna pažnja u disertaciji je data na TF-IDF meri kojom se izdvaja najbolji skup onih atributa koji su frekventni u individualnim dokumentima ali se retko pojavljuju u ostalom delu kolekcije.

- Pregled tehnologija modelovanja u servisima e-Uprave brojnih zemalja sveta, kao i unapređenje značenja pronađenih informacija kroz algorime bazirane na takozvanom BoW modelu. Zakon ili njegovi delovi su preko ovog modela predstavljeni kao neuređeni skup reči, zanemarujući pri tome njihov redosled.
- Konkretno opisan način kreiranja korpusa za modelovanje znanja kroz primer jedne vrste krivičnih dela. Predstavljen je proces ekstrakcije i prevođenja suštine iz teksta napisanog prirodnim jezikom u jasno definisan format. Pitanja koja su ovde modelovana napisana su ljudski razumljivim jezikom, tzv. prirodnim jezikom. Takođe, prezentovan je sistem za obradu teksta na prirodnom jeziku, pronalaženje i vizuelno predstavljanje konteksta i koncepata. Na taj način izvršeno je unapređenje postojećih tehnika označavanja koje mogu da obezbede odgovor sa ekstrahovanog dela (delova) dokumenta umesto celog tela dokumenta.
- Kako bi se obezbedio adekvatan odgovor iz ekstrahovanih delova dokumenata umesto iz celih tekstova dokumenata u disertaciji su ispitivane različite tehnike merenja sličnosti u tekstualnim dokumentima. Ispitivane su razne funkcije sličnosti, kao što je Džakard korelacioni koeficijent, Euklidova distanca i kosinusna sličnost između dokumenta i kratkih tekstova koji predstavljaju pitanja građana. Džakard korelacioni koeficijent ukazuje na veliku sličnost klaster analize i faktorske analize u kratkim tekstualnim dokumentima.
- Izvršena eksperimentalna provera efikasnosti predloženog sistema primenom najčešće osnovnih mere za efektivnost pronalaženja podataka - preciznost i odziv. Eksperiment je urađen na skupu podataka iz postojećih elektronskih verzija Krivičnog zakonika Republike Srbije s jedne strane i skupom pitanja izdvojenih sa internet portala za besplatnu pravnu pomoć – PRO BONE, s druge strane.

Dalja istraživanja biće bazirana na primeni inteligentnih agenata u svim fazama implementacije procesa automatizacije sistema brzih odgovora. Takođe, primeniće se i tehnike savremenih metoda IR, kao što su konceptualne šeme koje prate trendove razvoja veb 3.0 tehnologija.

Literatura

[1] Commission Staff Working Document (2003) „Linking up Europe: the importance of interoperability for e-Government services“ Brussels, 2003.

[2] MITRE (2004) „Information Interoperability“, THE EDGE, MITRE's Advanced Technologiy Newsleter, Vol.8, num.1

[3] European Commission, European Interoperability Framework or European Public Services (EIF), Version 2.0, 2004.

[4] Radatz, J., Geraci, A., Katki, F., 1990. IEEE Standard Glossary of Software Engineering Terminology. IEEE Standard, pp.1-84. [doi:10.1109/IEEESTD.1990.101064]

[5] Breitfelder, K., Messina, D., 2000. IEEE 100: the Authoritative Dictionary of IEEE Standards Terms (7th Ed.). IEEE Press. [doi:10.1109/IEEESTD.2000.322230].

[6] European Commission, European Interoperability Framework for pan-European eGovernment services, version 1.0, 2004.

[7] European Commission, The Role of eGovernment for Europe's Future, 2003.

[8] United Nations Development Programme, e-Government interoperability: overview, 2007.

[9] Bloomberg and Schmelzer, Service Orient or Be Doomed. Hoboken, NJ: Wiley & Sons, 2006, p. 118.

[10] Franc¸ois B. Vernadat, Technical, semantic and organisational issues of enterprise interoperability and networking, Annual Reviews in Control 34 (2010) 139–144.

[11] (Denmark) Ministry of Science,Technology and Innovation, White Paper on Enterprise Architecture, http://www.oio.dk/files/whitepaper.pdf

[12] Norbert Biebertein, Sanjay Bose,Marc Fiammente, Keith Jones, and Rawn Shah. Service Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap.Upper Saddle, NJ: IBM Press, 2006, p. 4.

[13] Chappell, D. A. (2004). Enterprise Service Bus. USA: O'Reilly Media Inc.

[14] V.Nikolić, P. Đikanović, D. Batoćanin, eGovernment Republike Srbije - Produženje registracije motornih i priključnih vozila, YU INFO 2013.

[15] V.Nikolić, J. Protić, P. Đikanović, G2G integracija MUP-A Republike Srbije sa portalom e-Uprava, ETRAN 2013.

[16] V. Nikolić, J. Protić, P. Djikanović, eGovernment interoperability in the context of European Interoperability Framework (EIF), ICIST 2014

[17] Vlada RS, Nacionalni okvir interoperabilnosti Republike Srbije, 2014.

[18] Strategija reforme državne uprave i Akcioni plan za sprovođenje reforme državne uprave 2009. do 2013. godine („Službeni glasnik RS", br. 55/05, 71/05 – ispravka, 101/07, 65/08, 16/11, 68/12 – US i 72/12).

[19] Strategija i akcioni plan za razvoj elektronske uprave do 2013. godine („Službeni glasnik RS", br. 55/05, 71/05-ispravka, 101/07 i 65/08).

[20] Strategija i akcioni plan za razvoj širokopojasnog pristupa do 2012. godine („Službeni glasnik RS" br. 55/05, 71/05-ispravka, 101/07 i 65/08).

[21] Strategija razvoja elektronskih komunikacija u Republici Srbiji do 2020. godine ("Službeni glasnik RS", broj 44/10).

[22] Strategija razvoja informacionog društva u Republici Srbiji do 2020. godine („Službeni glasnik RS", br. 55/05, 71/05-ispravka, 101/07 i 65/08).

[23] Aneks 2. Saopštenja Komisije Evropskom parlamentu, Savetu, Evropskom ekonomskom i socijalnom komitetu i Komitetu regiona 'U pravcu interoperabilnosti evropskih javnih usluga' COM(2010) 744 konačna verzija http://ec.europa.eu/isa/documents/isa_annex_ii_eif_en.pdf

[24] IDABC 2004. Evropski okvir interoperabilnosti za panevropske usluge elektronske uprave v 1.0 str. 5. http://ec.europa.eu/idabc/servlets/Doca2cd.pdf?id=19528

[25] United Nations Development Programme, eGovernment interoperability: overview, 2007.

[26] Evropska mreža javne uprave, Ključna načela arhitekture interoperabilnosti, str.5.

[27] Evropski okvir interoperabilnosti, v1.0. http://ec.europa.eu/idabc/en/document/3761/5845.html

[28] Okvir tehničke interoperabilnosti vlade Australije (AGTIF) v2. http://www.agimo.gov.au/publications/2005/04/agtifv2

[29] Evropska mreža javne uprave (EPAN), Ključna načela arhitekture interoperabilnosti, str.11.

[30] Evropski okvir interoperabilnosti, v1.0, str.16.

[31] Okvir tehničke interoperabilnosti vlade Australije v2, str. 1a.

[32] World Wide Web Consortium W3C, www.w3.org.

[33] Kreger, H., Web Services Conceptual Architecture (WSCA 1.0), IBM, 2001

[34] P. Djikanović , V. Nikolić, D. Sivčević, Nacionalni okvir interoperabilnosti Republike Srbije i servisno orijentisana arhitektura (SOA), YU INFO 2014

[35] Ivan Lazarević S&T, Interoperabilnost kao obaveza, prezentacija Sinergija 2013.

[36] V.Nikolić, S. Radovanović, Integracija web servisa MUP-a Republike Srbije sa portalom eUprave, INFOTECH 2012, 2012.

[37] Bieber G., J Carpenter, Introduction to Service-Oriented Programing, http://www.openwings.org, 2002.

[38] Shankar Raman, Steve Friedberg, Oracle WebLogic Server 11g: Administration essentials: Volume 1, 2010.

[39] V.Nikolić, R. Dragović, Interoperabilnosti i bezbednost eGovernment Republike Srbije, TELFOR 2014.

[40] Direkcija za elektronsku upravu, Lista standarda interoperabilnosti, Verzija 1.0, 2014

[41] Portal e-Uprava Republike Srbije, www.euprava.gov.rs.

[42] SAGA - Modul Grundlagen v 5.1.0

[43] IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossarie, Institute of Electrical and Electronics Engineers : http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=4683 , https://standards.ieee.org/

[44] Evropska mreža javne uprave (EPAN- European Public Administration Network), Ključna načela arhitekture interoperabilnosti, 2004

[45] R. Dragović, M. Ivković, B. Perović, Đ. Klipa, Dataveillance i data mining kao tehnološka podrška procesu istražnih radnji, TELFOR 2011

[46] E-government survey 2014, UNITED NATIONS

[47] R. Dragović, Data mining sistemi kao podrška istražnim radnjama, YUINFO 2011

[48] M. Konopik, o. Rohlik, Question answering for not yet semantic web, in: Brno, 2010, pp. 125 - 132.

[49] G. Koteswara Rao i Shubhamoy Dey, DECISION SUPPORT FOR E-GOVERNANCE: A TEXT MINING APPROACH, International Journal of Managing Information Technology (IJMIT) Vol.3, No.3, August 2011

[50] Hildebrandt W., Katz B., and Lin J. (2004). Answering Definition Questions Using Mul¬tiple Knowledge Sources. Proceedings of Human Language Technology Conference. Boston, USA, 2004.

[51] Soubbotin M. M., and Soubbotin S. M. (2001). Patterns of Potential Answer Expressions as Clues to the Right Answer. Proceedings of the TREC-10 Conference. Gaithersburg, 2001.

[52] Cui H., Kan M., and Chua T. (2004). Unsupervised Learning of Soft Patterns for Gener¬ating Definitions from Online News. Proceedings International WWW Conference. New York, USA, 2004.

[53] Ravichandran D., and Hovy E. (2002). Learning Surface Text Patterns for a Question Answering System. Proceedings of the ACL-2002 Conference. Philadelphia, USA, 2002.

[54] B. Magnini, M. Speranza, V. Kumar, Towards interactive question answering: an ontology-based approach, in, Berkeley, CA, 2009, pp. 612-617.

[55] Fleischman M., Hovy E. and Echihabi A. (2003). Offline Strategies for Online Question Answering: Answering Question Before they are Asked. Proceedings of the ACL-2003, Sapporo, Japan, 2003.

[56] http://www.clef-initiative.eu/home;jsessionid=C302BA2AD930C7C2FC842552D364C418

[57] S. Bhatnagar, E-Government: From Vision to Implementation, Sage Publications, India, 2004

[58] C. P. Cheng, G. T. Lau, K. H. Law, J. Pan, and A. Jones, "Improving Access to and Understanding of Regulations through Taxonomies," Government Information Quarterly, 26(2): 238-245, 2009.

[59] Prokopiadou, G., Papatheodorou, C., and Moschopoulos, D., Integrating knowledge management tools for government information, Government Information Quarterly, 21, 2, 2004, 170—198.

[60] Stuart W. Shulman, "eRulemaking: Issues in Current Research and Practice," International Journal of Public Administration Vol. 28 (2005), 621-641.

[61] Josh Froelich,Sergei Ananyan, David L. Olson,2008. The Use of Text Mining to Analyze Public Input .White paper.

[62] W. McKnight, "Building Business Intelligence: text data mining in business intelligence", DM Review, pp 21¬22,

[63] M.W. Berry, Survey of Text Mining: Clustering, Classification and Retrieval, Springer Verlag, New York, 2004.

[64] H. Ong, A. Tan, J. Ng, H. Pan, Q. Li., "FOCI : Flexible Organizer for Competitive Intelligence", Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM'01), pp 523-525, Atlanta, USA, 2001.

[65] S. Godbole, S. Roy, "Text to Intelligence: Building and Deploying a Text Mining Solution in the Services Industry for Customer Satisfaction Analysis", IEEE, pp 441-448, 2008.

[66] S. Weng, C. Liu, "Using text classification and multiple concepts to answer e-mails", Expert Systems with Applications, pp 529-543, 2004.

[67] N. Singh, C. Hu, W. S. Roehl, "Text mining a decade of progress in hospitality human resource management research: Identifying emerging thematic development", Hospitality Management, pp 131-147, 2007.

[68] http://www.clef-campaign.org/

[69] Ahonen-Myka H. (2002). Discovery of Frequent Word Sequences in Text Source. Pro-ceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery. London, UK, 2002.

[70] Garcia-Hernandez, R., Martinez-Trinidad F., and Carrasco-Ochoa A. (2006). A New Algorithm for Fast Discovery of Maximal Sequential Patterns in a Document Collection. International Conference on Computational Linguistics and text Processing, CICLing- 2006. Mexico City, Mexico, 2006.

[71] Ralf Steinberger ,Challenges and methods for multilingual text mining

[72] Hsin-Chang Yang, Chung-Hong Lee, and Ding-Wen Chen (2009) "A Method for Multilingual Text Mining and Retrieval Using Growing Hierarchical Self-Organizing Maps." Journal of Information Science, Vol. 35, No. 1, pp. 2-23. (SSCI)

[73] R. Chau and C.H. Yeh, A multilingual text mining approach to web cross-lingual text retrieval, Knowledge-Based Systems 17(5/6) (2004) 219–27.

[74] O. Ferrandez, C. Spurk, M. Kouylekov, I. Dornescu, S. Ferrandez, M. Negri, R. Izquierdo, D. Tomas, C. Orasan, G. Neumann, B. Magnini, J.L. Vicedo, The qall-me framework: a specifiable-domain multilingual question answering architecture, Journal of web semantics, 9 (2011) 137-145.

[75] http://www.slideshare.net/andreajob06/20090611-1809-teindomainoftourist

[76] V. Lopez, V. Uren, R. Motta, M. Pasin, Aqualog: an ontology-driven question answering system for organizational semantic intranets, web semantics: science, services and agents on the world wide web, 5 (2007) 72-105.

[77]http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=D4ED7BCBDA1C7F39D592920D924F784?doi=10.1.1.678.5788&rep=rep1&type=pdf

[78] J.-p. Ng, m.-y. Kan, Qanus: an open-source question-answering platform, in, 2010.

[79] https://www.semanticscholar.org/paper/QANUS-An-Open-source-Question-Answering-Platform-Ng-Kan/0d82201e0819a64f8e7f929cc233bd29d487f721/figure/2

[80] S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus, P. Morarescu, Falcon: boosting knowledge for answer engines, in: Proceedings of trec, 2000.

[81] https://www.cse.iitb.ac.in/~cs626.../group2_QuestionAnswering.ppt

[82] V. Nikolić, M. Ivković, P. Đikanović, S. Nedeljković, Realizacija procesa analize tekstualnih dokumenata pomoću biblioteke otvorenog koda Apache Lucene, DQM 2016

[83] E. Hatcher, O.Gospodnetić, M. McCandless, *Lucene* in action, Manning Publications, 2009

[84] Paul, T. (2004). The *Lucene* Search Engine. http://www.javaranch.com/journal/2004/04/Lucene.html

[85] J. Gamgo, Architecture and implementation of *Apache Lucene*, GIESSEN FRIEDBERG, 2010.

[86] V. Nikolić, S. Nedeljković, P. Djikanović, Information retrieval for unstructured text documents: Lucene indexing, EUROBREND 2015

[87] E. Hatcher, O.Gospodnetic, M. McCandless, *Lucene* in action, Manning Publications, 2009

[88] V. Nikolić, P. Djikanović, S. Nedeljković, Tehniques Of Cyberspace Information Searching In Serbian Text Document: Case Study For Crime Law, Archibald Rais, KPA Beograd 2016

[89] Javacc[tm] grammar files, 2010

[90] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. In the Third Conference on Applied Natural Language Processing (ANLP-92), pages 133--140, 1992.

[91] https://lingpipe-blog.com/2012/07/24/using-luke-the-lucene-index-browser-to-develop-search-queries/

[92] The strategy and action plan for the development of electronic administration until 2013 ("RS Official Gazette", Nos. 55/05, 71/05-correction, 101/07 and 65/08).

[93] D. Randjelovic, B. Popovic, V. Nikolic, S. Nedeljkovic, Intelligent search terms in the case of police services in eGovernment, New information technology for analitycal decision-making in the biological, economic and social systems, (M44), State university in Novi Pazar, 2014

[94] R. Dragović, J. Ivković, D. Dragović, Đ. Klipa, D. Radišić, V Nikolic, Sistem za podršku odlučivanju kao podrška za strateško upravljanje državnom upravom, YU INFO 2015

[95] Gerald Kowalski, Information Retrieval Architecture and Algorithms, The Springer International Series (2011)

[96] Peter Teufl, Udo Payer, Guenter Lackner, From NLP (Natural Language Processing) to MLP (Machine Language Processing), Computer Network Security (2010)

[97] G. Šimić, Z. Jeremić, E. Kajan, D. Randjelović, A. Presnall, A Framework for Delivering e-Government Support, Acta Polytechnica Hungarica, Vol. 11, No. 1, 2014

[98] Peter Teufl, Udo Payer, Guenter Lackner, From NLP (Natural Language Processing) to MLP (Machine Language Processing), Computer Network Security (2010)

[99] Krstev Cvetana, Obradovic Ivan, Utvic Milos, Vitas Dusko M, A system for named entity recognition based on local grammars, JOURNAL OF LOGIC AND COMPUTATION, (2014), vol. 24 br. 2, str. 473-489

[100] "The Nature of Mathematical Programming", Mathematical Programming Glossary, INFORMS Computing Society, http://glossary.computing.society.informs.org

[101] Optimization Algorithm Toolkit, http://optalgtoolkit.sourceforge.net/

[102] Koteswara Rao G., Shubhamoy Dey, DECISION SUPPORT FOR E-GOVERNANCE: A TEXT MINING APPROACH MINING APPROACH, International Journal of Managing Information Technology (IJMIT) Vol.3, No.3, August 2011

[103] http://www.scholarpedia.org/article/Text_categorization

[104] Jiang W., Samanthula B.: N-Gram based Secure Similar Document Detection, the 25th Annual WG 11.3 Conference on Data and Applications Security, Richmond, Virginia, July 11-13, 2011.

[105] D. Subotić, N. Forbes, "Serbo-Croatian language – Grammar", Oxford Clarendon press, str.25-31, 61-64, 101-113

[106] U. Marovac, A. Pljasković, A. Crnišanin, E. Kajan, N-gram analiza tekstualnih dokumenata na srpskom jeziku, Telfor, Beograd, novembar, 2012

[107] http://hlt.rgf.bg.ac.rs/VeBranka/NERanka.aspx

[108] www.paragraf.rs

[109] Ivan Klajn, Gramatika srpskog jezika, Zavod za udžbenike i nastavna sredstva, 2005

[110] *Lucene* (http://lucene.apache.org/)

[111] F. Wang, Z. Wang, Z. Li, Ji-Rong Wen, Concept-based Short Text Classification and Ranking, ACM 978-1-4503-2598-1/14/11, 2014

[112] Magnus Sahlgren and Rickard Coster. Using bag-of-concepts to improve the performance of support vector machines in text categorization. In Proceedings of the 20th International Conference on Computational Linguistics, COLING 2004, pages 487– 493, Geneva, Switzerland, August 2004.

[113] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen. Short text conceptualization using a probabilistic knowledgebase. In IJCAI, pages 2330–2336. AAAI Press, 2011.

[114] Z. Wang, H. Wang, and Z. Hu. Head, modifier, and constraint detection in short texts, Data Engineering (ICDE), 2014 IEEE 30th International Conference on, 280-291

[115] Hulth, A. 2003. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 Conference on Emprical Methods in Natural Language Processing, (Sapporo, Japan, 2003).

[116] Zhang, C., Wang, H., Liu, Y. Wu, D, Liao, Y. and Wang, B. 2008. Automatic Keyword Extraction from Documents Using Conditional Random Fields. In Journal of Computational Information Systems, 4, 3, 1169-1180.

[117] https://blogs.aws.amazon.com/bigdata/post/Tx22THFQ9MI86F9/Applying-Machine-Learning-to-Text-Mining-with-Amazon-S3-and-RapidMiner

[118] Berger, A et al (2000). Bridging the Lexical Chasm: Statistical Approaches to Answer Finding. In Proc. Int. Conf. Research and Development in Information Retrieval, 192-199.

[119] Z. Stevic, M. Rajcic-Vujasinovic, I. Radovanovic, V. Nikolic, Modeling and Sensing of Electrochemical Processes upon Dirac Potentiostatic Excitation of Capacitive Charging/Discharging, Int. J. Electrochem. Sci., 10 (2015) 6020-6029

[120] Andrew B. King, Website Optimization, O'Reilly Media, July 2008

[121] http://www.besplatnapravnapomoc.rs/

[122] Isabelle Augenstein, Diana Maynard, Fabio Ciravegna, Distantly Supervised Web Relation Extraction for Knowledge Base Population, Semantic Web Journal, 2015

[123] http://hlt.rgf.bg.ac.rs/VeBranka/BagOfWords.aspx

[124] Sujian Li, Houfeng Wang, Shiwen Yu, Chengsheng Xin, News-Oriented Keyword Indexing with Maximum Entropy Principle, Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation, 2003

[125] Sujian Li, Houfeng Wang, Shiwen Yu, Chengsheng Xin, News-Oriented Automatic Chinese Keyword Indexing, Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, 2003

[126] J. Ross Quinlan. 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[127] Magerman, Tom, et al. "Assessment of Latent Semantic Analysis (LSA) text mining algorithms for large scale mapping of patent and scientific publication documents." 2011.

[128] Metzler, D. and Croft, W.B., "Analysis of Statistical Question Classification for Fact-based Questions," in Information Retrieval, 8(3), 481-504, 2005.

[129] Shashank, Shailendra Singh, Statistical Measure to Compute the Similarity between Answers in Online Question Answering Portals, International Journal of Computer Applications (0975 – 8887) Volume 103 – No.15, October 2014

[130] Stanje razvoja eUprave u Republici Srbiji za 2008. godinu, Digitalna agenda Republike Srbije, 2008