



Univerzitet „Džemal Bijedić“ u Mostaru
Fakultet informacijskih tehnologija

Završni rad nakon II ciklusa studija

Mogućnosti primjene sistema preporuke u kolaborativnom web okruženju za elektronsko učenje

Mentor:

doc. dr. Nina Bijedić

Kandidat:

Elvis Kadić

Novembar 2014. godine, Mostar

*Zahvaljujem mentoru, doc. dr. Nini Bijedić,
profesoru na Fakultetu informacijskih tehnologija u Mostaru,
na podršci, savjetima i primjedbama tokom izrade ovog rada.*

*Također, želio bih da se zahvalim cijelom osoblju Fakulteta informacijskih tehnologija
u Mostaru, na maksimalnoj podršci tokom studiranja.*

Veliko hvala mojoj porodici na podršci i strpljenju.

Sažetak

Inspiracija za ovaj rad nastala je na osnovu činjenice, da moderno društvo doživljava virtualizaciju u punom smislu. Iako se ne može sa sigurnošću odgovoriti na pitanje da li je to zaista pozitivan smjer kretanja društva, realnost je takva kakva jeste i mora se prihvatiti. U ovom nezaustavljivom procesu, pojavljuju su se mnogobrojna virtualna okruženja, koja nude različite oblike kolaboracije među korisnicima ili grupama korisnika. Koncepti iz „stvarnog svijeta“ dobili su drugu dimenziju i trenutno doživljavaju svoj procvat. Između ostalog, pojavio se i revolucionarni koncept elektronskog učenja, koji je u akademskoj zajednici predmet sadašnjih, ali i budućih istraživanja. Ovaj rad će pokušati napraviti određene pomake u oblasti sistema preporuke, kako bi se korisnicima omogućilo ugodnije virtualno okruženje. U cilju implementacije takve ideje, neophodno je istražiti oblasti, kao što su: kolaboracija i komunikacija, upravljanje znanjem, rudarenje podataka, umjetna inteligencija, sistemi preporuke i sl. Postoji nada da će rezultati istraživanja ostaviti dovoljno prostora za ispitivanje i primjenu novih pristupa.

Ključne riječi: virtualizacija, kolaboracija, elektronsko učenje, rudarenje podataka, umjetna inteligencija

Summary

The inspiration for this work arose from the fact, that modern society is experiencing virtualization in the fullest sense. Although we can not answer with certainty the question of whether this is indeed a positive direction of the society, the reality is the way it is and must be accepted. In this unstoppable process, many virtual environments were being developed, offering various forms of collaboration among users or groups of users. The concepts of "real world" were given another dimension and is currently experiencing its peak. Among other things, there is also the revolutionary concept of e-learning, that is, the subject of current and future researches in the academic community. This paper will try to make some progress in the area of system recommendations, in order to provide users more comfortable virtual environment. To implement such ideas, it is necessary to explore the area, such as collaboration and communication, knowledge management, data mining, artificial intelligence, systems recommendations, etc. There is hope that the results of this study leave enough space for the study and application of new approaches.

Keywords: virtualization, collaboration, e-learning, data mining, artificial intelligence

SADRŽAJ:

1	Uvod.....	1
1.1	Hipotetički okvir istraživanja	3
1.1.1.	Glavna hipoteza	4
1.1.2.	Pomoćne hipoteze.....	4
1.2	Predmet i ciljevi istraživanja	4
1.3	Metode i tok istraživanja	5
2	Pregled relevantnih istraživanja iz oblasti	7
2.1	Kolaborativni sistemi za upravljanje znanjem.....	8
2.1.1	Kolaboracija	9
2.1.2	Komunikacija i upravljanje sadržajem	11
2.1.3	Kolaborativno upravljanje znanjem.....	13
2.1.4	Wiki modul	16
2.1.4.1	Primjena Wiki-ja u kolaborativnom učenju.....	17
2.1.4.2	Primjena Wiki-ja u proučavanju i istraživanju	21
2.1.4.3	Wiki alati	21
2.1.5	Internet forum	22
2.2	Inteligentni programski agenti.....	24
2.2.1	Struktura inteligentnog agenta	25
2.2.2	Vrste inteligentnih programskih agenata	28
2.2.3	Agent-bazirani programski jezici	29
2.2.4	Područje primjene inteligentnih programskih agenata	30
2.3	Sistemi preporuke.....	31
2.3.1	Formalna definicija problema generisanja preporuke	33
2.3.2	Tradicionalni sistemi preporuke	34
2.3.2.1	Sistemi preporuke zasnovane na sadržaju	35
2.3.2.2	Kolaborativno filtriranje	36
2.3.2.3	Agregativni sistemi preporuke.....	38
2.3.2.4	Ograničenja tradicionalnih sistema preporuke	39
2.3.3	Savremeni pristupi u razvoju sistema preporuke.....	42
2.3.3.1	Adaptivni sistem preporuke(ASP).....	42
2.3.3.2	Adaptivna agregacija	46
2.3.4	Prikupljanje podataka za model korisnika.....	52
3	Implementacija FITCKM sistema	55
3.1	FITCKMS.....	55
3.1.1	FITCKMS - Opis sistema	56
3.1.1.1	Moduli sistema	56
3.1.2	FITCKMS – Baza podataka	57
3.1.3	FITCKMS - Pregled mogućnosti aplikacije	59

3.1.4	FITCKMS - Korisnički interfejs	60
3.2	Eksperimentalno okruženje	61
4	Metode	63
4.1	Uloga inteligentnog programskog agenta.....	63
4.2	Priprema podataka.....	67
4.3	Modeli sistema preporuke	69
4.3.1	Sistemi preporuke	70
4.3.1.1	Standardni (tradicionalni) sistemi preporuke	70
4.3.1.2	Adaptivni sistemi preporuke.....	72
4.3.2	Procjena greške.....	72
5	Eksperimenti i rezultati	74
5.1	Eksperiment 1	74
5.2	Eksperiment 2.....	76
5.3	Eksperiment 3.....	78
5.4	Eksperiment 4.....	82
6	Diskusija i zaključak.....	86
6.1	Kolaboracija i upravljanje znanjem.....	86
6.2	Wiki stranica.....	87
6.3	Implikacije i ograničenja sistema preporuke	87
6.4	Kompleksnost adaptivnih sistema preporuke.....	89
6.5	Vremenska ograničenja	90
6.6	Mjere evaluacije	90
6.7	Inteligentni programski agenti.....	91
6.8	Pravci budućih istraživanja.....	92
6.9	Zaključak	93
7	Literatura	95
8	Prilozi.....	103

1 UVOD

Informacione i komunikacijske tehnologije (ICT) su danas, direktno ili indirektno, integrisane u skoro sve aspekte društva. Vrijeme protoka i obrade informacija je svedeno na minimum, te je potreba za razvojem i implementacijom novih i savremenih softverskih rješenja u konstantnom porastu. Između ostalog, uvođenje ICT-a u obrazovne sisteme pokazalo se kao obavezujuće, te je u velikoj mjeri je uticalo na poboljšanje kvaliteta obrazovanja. Dva su osnovna koncepta ovakvog obrazovanja: *učenje na daljinu* (eng. *distance-learning*) i *elektronsko obrazovanje* (eng. *e-learning*) [94]. Spomenuti koncepti su vrlo slični i često se koriste kao sinonimi. Elektronsko obrazovanje se može definisati kao učenje uz pomoć računara, odnosno ICT-a, i podrazumjevano je u konceptu učenja na daljinu. Kombinacija elektronskog obrazovanja sa tradicionalnim konceptom naziva se *miješano obrazovanje* (eng. *blended-learning*) [94].

Pomenuti koncepti su, prije svega, omogućili brži protok informacija, podršku za multimedijalne nastavne materijale, kao i ogromne finansijske uštede obrazovnim ustanovama, osoblju i studentima. Može sa sigurnošću konstatovati da su svojevremeno predstavljali i još uvijek predstavljaju „*revoluciju*“ u obrazovanju.

Međutim, testiranjem u različitim okruženjima, otkriveni su mnogobrojni problemi koje je potrebno prevazići. Jedan od osnovnih problema ovakvih sistema je što se svi korisnici tretiraju na isti način. Oni imaju pristup istim resursima, kao i jednaku sistemsku podršku, bez obzira na njihovo znanje, ciljeve, komunikacijske vještine, interesovanja i slične karakteristike. Ovaj problem je vrlo česta pojava u sistemima današnjice i naziva se *problem latentne subjektivnosti* (eng. *latent subjectivity problem*).

Razmišljanje u smjeru pružanja individualnog pristupa korisnicima bi predstavljalo rješenje za većinu navedenih problema, što bi ovakvim sistemima dalo potpuno novu dimenziju. Slijedeći tu ideju, razvoj modernih sistema je fokusiran na mnogobrojne modifikacije, odnosno različite module, koji omogućavaju prilagodljivu hipermedijsku podršku baziranu na korisničkoj individualnosti. Prije svega, ovaj koncept se odnosi na razvoj metoda i tehnika za adaptivnu navigaciju i prezentaciju sadržaja.

Cilj primjene adaptivne navigacije, odnosno adaptacije hiperveza¹ je navođenje korisnika prema relevantnim i zanimljivim informacijama. Tehnikama poput direktnog navođenja, sortiranja, označavanja, skrivanja i brisanja hiperveza pokušava se pojednostaviti njihova struktura, kako bi se umanjila dezorijentacija u hiperprostoru, a istovremeno sačuvala slobodna navigacija kao glavna karakteristika hipermedijske paradigme [1].

Na osnovu primjera iz stvarnosti, početkom XXI vijeka počinje razvoj sistema preporuke u kontekstu e-učenja, gdje se različitim tehnikama studentima preporučuju online resursi, aktivnosti ili staze pretraživanja na osnovu njihovih karakteristika, kao i na osnovu aktivnosti njima sličnih studenata, čime se uvažava socijalna komponenta procesa učenja.

Iako vrlo slični adaptivnim hipermedijskim aplikacijama, kod kojih se prilikom adaptacije hiperveza u obzir uzimaju različiti aspekti korisnika (npr. predznanje, interesi, ciljevi i sl.), sistemi preporuke se često ograničavaju na jedan aspekt koji se naziva *interes korisnika* [1]. Istraživanja u ovoj oblasti usmjerena su na prevladavanje tog nedostatka koji je vrlo značajan za kontekst obrazovanja, u smislu dostupnosti i usvajanja novog znanja.

Korištenje sistema za preporuke u kontekstu hipermedijskih obrazovnih aplikacija se može direktno povezati i s novijim pristupima u razvoju obrazovnih okruženja na web-u koji ističu važnost *kolaborativnog*² učenja (*eng. collaborative learning*). Kolaborativno učenje bazira se na konstruktivističkoj teoriji, koja studenta stavlja u središte obrazovnog procesa i pretpostavlja da studenti uče jedni od drugih, dok učestvuju u različitim aktivnostima. Jedan od osnovnih ciljeva mnogobrojnih adaptivnih hipermedijskih sistema je i dostavljanje nastavnih materijala, tako da podrška kolaborativnom učenju predstavlja novu priliku za implementaciju adaptacije [2].

¹ Hiperveza ili poveznica (*eng. Hyperlink*) je osobina pojedinih, posebno označenih riječi, slika ili dijelova slika u tekstovima da predstavljaju posrednu vezu ka dodatnim informacijama. One predstavljaju referencu ili navigacioni element ka drugom odjeljku, drugom dokumentu ili odjeljku iz drugog dokumenta.

² Kolaboracija (*lat. co-* = „sa-“, *laborare* = „raditi“) označava saradnju pojedinaca ili grupa radi postizanja određenog cilja.

U ovom radu će se razmatrati mogućnosti primjene različitih modela preporuke u kolaborativnim sistemima za upravljanje znanjem, odnosno u svim sistemima koji imaju potrebu za predviđanje nepoznatih ocjena za određene sadržaje. Osim standardnih sistema preporuke, koji su već nekoliko godina u praktičnoj upotrebi, istraživanje će uglavnom biti fokusirano na njihovom kombinovanju, odnosno na adaptivno-agregativnim pristupima. Detaljno će biti istraženi i objašnjeni postojeći koncepti, problemi i napredak u ovoj oblasti. Također, posebna pažnja će biti posvećena problemima kao što su *latentna subjektivnost* i *preopterećenje informacijama* (eng. *information overload*).

Kao dodatna tehnika za rješavanje spomenutih problema, simuliraće se upotreba inteligentnog programskog agenta, koja bi trebala napraviti određene pomake u cilju preciznosti predviđanja.

U praktičnom dijelu će biti razvijen *kolaborativni sistem za upravljanje znanjem (FITCKMS)*, kao i eksperimentalno okruženje, specijalizovano za testiranje ideja ovog rada. *FITCKM* sistem će se sastojati iz dva modula: *Wiki stranica* (eng. *Wiki page*) i *pitanja i odgovori* (eng. *questions and answers, QA*).

Za potrebe istraživanja teorija i hipoteza rada, u eksperimentalnom okruženju koristiće se stvarni podaci iz *FITCKM* sistema. Kao eksterni izvor podataka u eksperimentima, izabrana je besplatna kolekcija filmova *MovieLens*, koja po svim karakteristikama odgovara planiranim testovima i najčešće koristi za istraživanja u ovoj oblasti.

Iako je tema rada zaista široka i zahtijeva istraživanje iz različitih oblasti ICT-a, navedene kolekcije podataka će biti sasvim dovoljne za izvođenje planiranih eksperimenata i predstavljanje rezultata. Na osnovu rezultata praktičnog testiranja, bit će izvedeni zaključci i predstavljeni pravci budućih istraživanja u ovoj oblasti.

1.1 Hipotetički okvir istraživanja

U skladu sa temom ovog rada i preliminarim istraživanjima informatičkih oblasti kao što su: kolaboracija i komunikacija, upravljanje znanjem, rudarenje podataka, umjetna inteligencija, sistemi preporuke i sl., stvoreni su uslovi za razmišljanje o potpuno novim

konceptima rješavanja problema personalizacije korisnika. Kombinovanjem novih ideja sa modelima, tehnikama i algoritmima, koji su trenutno u upotrebi, postavljene su sljedeće hipoteze:

1.1.1. Glavna hipoteza

H1: Razvojem inteligentnih agenata, u kontekstu razotkrivanja nedavnih/trenutnih korisničkih kretanja, interesovanja i potreba, stvara se pretpostavka da je moguće razviti nove oblike znanja (metapodatke), i pomoću njih unaprijediti modele preporuke usmjerene ka konkretnom entitetu, odnosno postići viši nivo umjetne inteligencije sistema.

1.1.2. Pomoćne hipoteze

H1.1: Praćenjem korisnika kroz sistem moguće je otkriti nove odnose među podacima i stvoriti nove metapodatke, te iskoristiti njihovo značenje

H1.2: Na osnovu novostvorenih metapodataka moguće je preciznije preporučiti resurse kombinovanjem adaptivnih pondera³ određenih sistema preporuke.

1.2 Predmet i ciljevi istraživanja

Predmet istraživanja ovog rada je mogućnost primjene različitih sistema preporuke u kolaborativnim okruženjima, s akcentom na adaptivno-agregativni pristup. Kolaborativna okruženja, između ostalog, mogu poslužiti kao podrška samostalnim virtualnim zajednicama za elektronsko učenje ili tradicionalnim LMS ili VLE sistemima.

Za realizaciju ovakvog koncepta je neophodno je postići slijedeće ciljeve:

- Istražiti koncepte, module i sisteme kolaborativnog upravljanja znanjem
- Istražiti mogućnosti primjene inteligentnog programskog agenta
- Analizirati i definisati postojeće koncepte sistema preporuke, koji će se koristiti u radu

³ Relativni ponder pokazuje udio pojedine vrijednosti u strukturi cjeline

- Definirati metode i testove koji će se koristiti u radu
- Razviti eksperimentalno okruženje za testiranje ideja i hipoteza rada
- Izvršiti eksperimentalna testiranja
- Predstaviti rezultate i zaključke, te definirati pravce budućih istraživanja

Kroz detaljne analize prednosti i nedostataka pristupa korištenih u ovom radu, jedan od osnovnih ciljeva je omogućiti korisniku zanimljivo, korisno i ugodno kolaborativno okruženje.

Postoji iskrena nada da će rezultat, odnosno svrha ovog rada, biti postizanje određenih poboljšanja u oblastima kolaborativnog upravljanja znanjem i sistemima preporuke, te da će predstavljanje novih ideja možda odrediti pravac budućih istraživanja. To se ne odnosi isključivo na domenu elektronskog učenja, već bi se eventualno poboljšani koncepti mogli primijeniti u različitim scenarijima.

1.3 Metode i tok istraživanja

Za testiranje hipoteza rada neophodno je napraviti detaljnu analizu dosadašnjih dostignuća iz oblasti koja se bavi sistemima preporuke. Posebna pažnja bit će usmjerena prije svega na trenutne koncepte korištenja takvih sistema. Osnovna ideja je adaptacija dosadašnjih dostignuća u ovoj oblasti, odnosno unaprijeđeno korištenje provjerenih metoda i algoritama sistema preporuke, usmjereno ka individualnom korisniku.

Pokretačka ideja je veoma jednostavna i bazirana na tome da kvalitet preporuke ne zavisi samo od preciznosti korištenih metoda, već od različitosti skupova, obrade i aspekata podataka. Isti podatak se može koristiti u različitim kontekstima, odnosno posmatrati iz različitih perspektiva. Drugim riječima, značenje podatka se može interpretirati na različite načine, zavisno da li se podatak posmatra individualno ili u korelaciji sa drugim podacima. Također, i različiti koncepti korelacije među podacima predstavljaju specifične vrijednosti i značenja. Ovaj pristup bi mogao omogućiti pronalaženje skrivenih, potencijalno vrijednih informacija, odnosno objekata čija bi se relevantnost dodatno provjeravala.

Za realizaciju ove ideje koristit će se standardne statističke metode koje na različitim mehanizmima istražuju nepoznate veze među podacima. To svakako podrazumijeva korištenje kvalitetnih skupova podataka koji po svojoj strukturi i veličini odgovaraju podacima u stvarnim sistemima. U smislu poboljšanja kvaliteta podataka i predviđanja, glavnu ulogu će imati simulacija inteligentnog programskog agenta opisanog u poglavlju 4.1.

Nakon definisanja metoda i skupova podataka, pristupiće se eksperimentalnoj fazi. Na prvom sloju modelovanja izvršiće se procjena preciznosti svih standardnih metoda i upotrebom greška-modela, rezultati biti predstavljeni kao odstupanje od stvarnog predviđanja. Dobijene vrijednosti će se iskoristiti na drugom sloju modelovanja pri određivanju težina (pondera) svih korištenih standardnih metoda. Ponderi će, uz normalizaciju grešaka, određivati koliko će svaki od definisanih algoritama biti angažovan u adaptivnoj preporuci. Također, i spomenutom adaptivnom pristupu će se procijeniti preciznost. Istraživanje ovog pristupa i njegova primjena detaljno je objašnjena u potpoglavlju 2.3.3 i poglavlju 4.

Komparacijom rezultata dobijenih sa ili bez korištenja inteligentnog programskog agenta izvršiće se dodatna procjena preciznosti predviđanja.

Svim standardnim metodama, odnosno algoritmima koje one koriste, bit će izmjereno vrijeme izvršavanja nad konkretnim podacima. Na osnovu dobijenih rezultata bit će potrebno procijeniti vremensku opravdanost korištenja svake od njih.

2 PREGLED RELEVANTNIH ISTRAŽIVANJA IZ OBLASTI

U ovom poglavlju će biti napravljen presjek pristupa i trenutnih dostignuća iz različitih oblasti vezanih za tematiku ovog rada. Prije svega, to su: kolaboracija, komunikacija, upravljanje znanjem, inteligentni programski agenti i sistemi preporuke. Uz interpretaciju definicija, osnovnih i savremenih pristupa, razmotrit će se mogućnost poboljšanja u ovim oblastima.

Kolaborativna virtualna okruženja su dostigla zavidan nivo upotrebe, koji se teško mogao zamisliti prije samo par godina. Slične karakteristike se mogu pronaći u oblastima upravljanja znanjem i umjetne inteligencije. Iako su se u bliskoj prošlosti ovi koncepti smatrali nedokučivim, napredak je danas očigledan. S druge strane, trenutna dostignuća se mogu posmatrati samo kao početnički pokušaji, u odnosu na širok spektar ideja koji se pojavio u akademskoj zajednici.

Modeli preporuke na određeni način objedinjuju spomenute oblasti i u fokusu su interesovanja IT stručnjaka. Korištenje modela preporuke u modernim softverskim rješenjima je često podrazumjevano i predstavlja okosnicu razvoja proizvoda. Drugim riječima, preporuka je nerijetko najvažniji, odnosno neizostavan segment savremenog softvera. Trenutno su upotrebi različiti napredni modeli preporuke koji su bazirani na veoma komplikovanim algoritmima, a koriste ih popularni pretraživači, socijalne mreže, a u posljednje vrijeme primjena ovakvih modela je sve češća u poslovnim aplikacijama, kao i u aplikacijama koje za osnovu koriste kolaboraciju među korisnicima.

U posljednjih par decenija, na polju umjetne inteligencije posebna pažnja usmjerena je na još uvijek nedovoljno istraženu oblast inteligentnih programskih agenata. Iako je postignut veliki napredak u ovoj oblasti, akademska zajednica vjeruje da njihove potencijalne mogućnosti mogu biti slijedeća velika prekretnica u IT.

Sve spomenute oblasti, modeli i tehnike imaju zajednički cilj koji se može definisati kao omogućavanje što ugodnijeg korisničkog okruženja sa što bržom isporukom interesantnih i značajnih sadržaja.

2.1 Kolaborativni sistemi za upravljanje znanjem

Globalizacija, eksplozivan razvoj interneta kao i drugih komunikacijskih sistema omogućili su visoku dinamiku razmjene informacija i novu vrstu zajednica. Dvije osnovne karakteristike ovakvih zajednica su kolaboracija i upravljanje znanjem. Ove karakteristike imaju binomni odnos, što znači, kolaboracija se ne može ostvariti bez upravljanja znanjem, i obratno, upravljanje znanjem u novom globalnom društvu, se ne može graditi bez kolaboracije.

Problem je dosta kompleksan, jer u kolaboracijskim sistemima znanje je heterogeno i *sistem upravljanja znanjem (KMS)* treba prevesti ogromne količine *tacitnog znanja*⁴, u eksplicitno znanje i obrnuto. Shodno tome, nastala je nova oblast pod nazivom *kolaborativni sistemi upravljanja znanjem (CKMS)* [3].

Paradigme kolaboracije i upravljanja znanjem su u međusobnoj interakciji i stvaraju nove vrijednosti društva kao što su kolaborativne zajednice ili znanje izgrađeno na osnovu društva. Jedan od glavnih izazova u akademskom istraživanju je proučavanje funkcionisanja ovog mehanizama u stvarnom svijetu. U tom smislu, postoji veliki broj studija koje se uglavnom baziraju na istraživanje ekonomije i marketinga, ali su vrlo rijetke u oblasti obrazovanja. Ovaj problem je još izraženiji u zemljama u razvoju, gdje je iskustvo sa primjenom ovakvih sistema na vrlo niskom nivou.

U skladu sa temom rada, u ovom potpoglavlju bit će istražena oblast *kolaborativnih sistema za upravljanje znanjem (CKMS)*, *virtualne zajednice (VC)* i osnovni koncepti *društvenih mreža (SN)*.

⁴ Tacitno ili Prešutno znanje - je znanje koje uključuje razumijevanje, intuiciju i slutnje. Takvo znanje je u našim "glavama" ali nije formalizirano, tj. izraženo brojkama i slovima. Ono zavisi od same individue, njenim iskustvima, emocijama, intuiciji i razumijevanju pojedinih stvari. Michael Polanyi je napisao brojne radove u kojima spominje tacitno znanje. U svojim djelima govori o tacitnom znanju kao "prešutnom znanju", odnosno govori da mi "znamo više nego što možemo reći". Primjer tacitnog znanja je znanje kako voziti biciklo.

2.1.1 Kolaboracija

Kolaboracija se može posmatrati iz različitih perspektiva. Opšta definicija kolaboracije može podrazumjevati sljedeće [4]:

“Kolaborativni sistemi predstavljaju interdisciplinarno polje, i mogu se posmatrati kao presjek informatike, kibernetike, kognitivnih znanosti, psihologije, lingvistike, sistema za podršku u odlučivanju, upravljanja i sl. “

Iz navedene definicije, mogu se izdvojiti određene specifičnosti:

- kolaboracija podrazumijeva vrlo sofisticirane odnose, odnosno zahtijeva velike napore za ujedinjavanje ljudi i organizacija, radi ostvarivanja zajedničkih ciljeva, koji se ne mogu postići samostalnim djelovanjem istih [5]
- kolaboracija je proces u kojem učesnici dijele podatke, resurse i odgovornosti, te zajednički planiraju, implementiraju, i evaluaciju programa aktivnosti za postizanje zajedničkog cilja [6]

Autori su koristili sljedeću definiciju, poznatu kao *3C paradigma* [7]:

„Kolaboracija je kooperacija, komunikacija (umrežavanje) i koordinacija“ [8]-[9].

Kooperacija se posmatra kao zajednički napor za postizanje zajedničkih ciljeva, komunikacija je aktivnost razmjene informacija između učesnika, a koordinacija s aktivnostima osigurava da različiti učesnici zajedno rade na postizanju zajedničkih ciljeva.

Drugi autori insistiraju na kolaborativnoj kreativnosti, kao i na značajnoj ulozi upravljanja sadržajem [10]. Prema njima, upravljanje sadržajem je proces stvaranja, ažuriranja i objavljivanja sadržaja na internetu, te može se tumačiti na različite načine:

- *s aspekta poslovanja*, upravljanje sadržajem predstavlja sredstvo za isporuku poslovnih vrijednosti, koristeći procese i alate koji se nalaze u pozadini

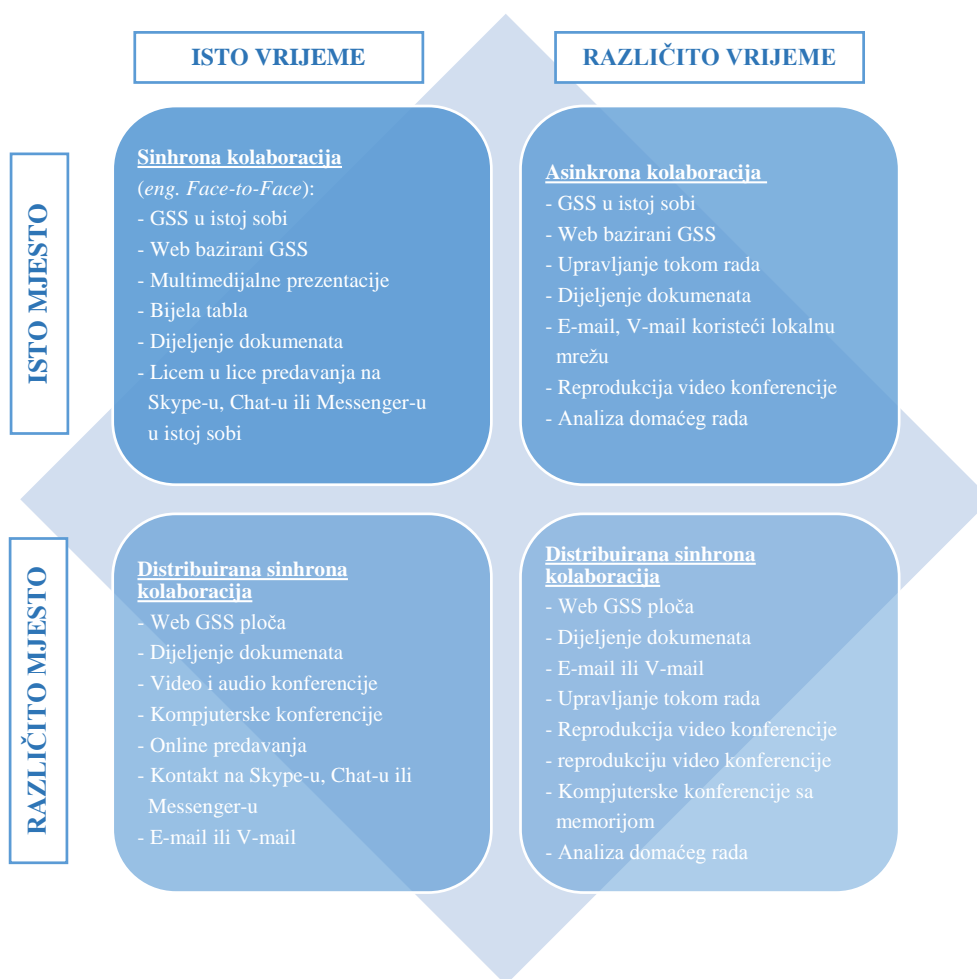
- *sa tehničkog stanovišta*, upravljanje sadržajem predstavlja kombinaciju hardvera i softvera koji čine sistem za upravljanje sadržajem

U navedenom kontekstu, kolaboracija se može posmatrati kao presjek četiri osnovne domene: *kooperacije, komunikacije, koordinacije i upravljanja sadržajem*. Proizvod ovakve teorije se može protumačiti kao oplemenjivanje *3C paradigme*, koja prvenstveno podrazumijeva mogućnost integrisanja metoda upravljanja sadržajem u kolaboraciji. U modernim sistemima, upravljanje sadržajem ima jednu od najvažnijih uloga.

Također, kolaboracija se može posmatrati sa teoretskog, ali i sa praktičnog stanovišta, tako da je i bez detaljnih studija, moguće prepoznati dvije taksonomije:

- a) Pozadina i uloge učesnika u saradnji sa *epistemičkog*⁵ stanovišta [11]:
 - poslodavac / radnik
 - učitelj / učenik
 - slični entiteti
 - različiti entiteti
- b) Drugo stanovište je praktično i predstavlja kombinaciju komunikacijskih modela za vrijeme i mjesto [12], s nekom od kolaborativnih tehnologija (*Slika 1.*) [11].

⁵ Epistemologija (iz grčkog jezika επιστήμη - episteme, "znanje" + λόγος, "logos") ili teorija znanja je grana filozofije koja se bavi prirodom i dosegom znanja. Izraz je prvi uveo škotski filozof James Frederick Ferrier (1808. – 1864.).



Slika 1. Kombinacija vrijeme/mjesto komunikacijskih modela i kolaborativnih tehnologija

Kako bi se iskoristio puni potencijal sticanja, upravljanja, dijeljenja i širenja znanja u dinamičkom okruženju, na polju saradnje moraju biti urađene određene modifikacije sistema [13].

2.1.2 Komunikacija i upravljanje sadržajem

Komunikacija je proces razmjene informacija preko dogovorenog sistema znakova i obično je opisana dimenzijama: sadržaj, forma i cilj.

Komunikacija se može ostvariti na različite načine, putem: direktne komunikacije (*eng. face-to-face*), e-mail poruka, razgovorom (*eng. chat*), e-mail listi (*eng. listservs*), grupa, oglasnih ploča, video konferencija i sistema za elektronske sastanke, portala i sl.

"Bez komunikacije, nema kolaboracije" [11].

Komunikacija predstavlja osnovu, odnosno uslov koji mora biti prethodno ispunjen, kako bi se uopšte moglo razgovarati o kolaboraciji.

„Osnovni zadatak komunikacije je da prenosi informacije od pošiljaoca do primaoca, ali kolaboracija je mnogo dublji pojam. Kolaboracija prenosi značenje ili znanje među članovima grupe“ [11].

U velikim grupama, informacija je heterogena i neophodna za generisanje rezultata za upravljanje sadržajem. *Sistem za upravljanje sadržajem (CMS)* osigurava nabavku, upravljanje i objavljivanje sadržaja. Ovi sistemi mogu biti sastavljeni od različitih komponenti, ali u okvirima ovog rada, najvažniji dijelovi će predstavljati upravljanje internim sadržajima web aplikacije. Osim toga, sistem će na osnovu ključnih riječi korisničke pretrage preporučiti relevantne sadržaje sa originalne *Wikipedije*, kao eksternog izvora informacija.

Tehnički aspekti CMS-a su vrlo zanimljivi, kao na primjer različiti formati podataka i hiperveza (*HTML, RDF ili XML*). Jedan od najinteresantnijih primjera CMS-a je *Wiki povezivanje* (eng. *Wiki linkage*). To je klasični *Wiki* sistem prikupljanja web stranica putem hiperveza [14]. Prednosti ovakvih sistema su:

- ne zahtijevaju tehničke sposobnosti kao što su *HTML*
- posjeduju povratni mehanizam s verzijama
- osiguravaju različite načine povezivanja
- osiguravaju učitavanje različitih sadržaja
- posjeduju funkcije za pretraživanje
- predstavljaju vrlo jaku bazu za CKMS, tj. sadrži heterogene informacije

U posljednje vrijeme, klasični *Wiki* je proširen i podržava različite semantičke oblike koristeći različite semantičke tehnologije kao što su: *RDF, OWL, Topic Maps, Conceptual Graphs* i sl. [14].

2.1.3 Kolaborativno upravljanje znanjem

Kolaborativno upravljanje znanjem je bazirano na dvije, vrlo važne IT tehnologije [15]:

- *sistemi za upravljanje znanjem (KMS)*
- *kolaborativni informacijski sistemi (CIS)*

Osnovni cilj *upravljanja znanjem (KM)* je prikupljanje značajnih informacija iz znanja, kojeg organizacije posjeduju kroz svoje adekvatno i eksplicitno upravljanje [16]. KM uključuje alate, strategije i praktična iskustva za identifikaciju, razvoj, predstavljanje, distribuciju i dostupnost različitih vrsta znanja [17]. Znanje je prihvaćeno kao najvažnija imovina današnjih organizacija [18]. KM također ima različita značenja. Dva gotovo ekvivalentna značenja su :

- upravljanje znanjem je sastavljeno od prikupljanja, kategorizacije i širenja znanja u organizaciji [16]
- upravljanje znanjem je sastavljeno od praktičnog sticanja, snimanja, gradnje, dijeljenja i korištenja znanja [19]

Neki autori sisteme za upravljanje znanjem opisuju kao specijalne, tekst ili hipertekst orijentirane sisteme, jer uglavnom upravljaju dokumentima različitih formata [20].

Sistemi za upravljanje znanjem su fokusirani na neprestano poboljšanje performansi, prednošću nad konkurencijom i inovacijama na dva glavna polja: *organizacijskog učenja (OL)* i *organizacijske memorije (OM)*, kao *spremište znanja (KR)* koje sadrži znanje i najbolje prakse [99]. Spremište znanja predstavlja stratešku imovinu i ima zadatak podsticanje i dijeljenja znanja u organizacijama.

Upravljanje znanjem se teorijski i praktično počelo proučavati krajem prošlog vijeka i trenutno se postepeno kreće do akademske zrelosti. Organizacijskom učenju je posvećena posebna pažnja, prvenstveno zbog pozitivnog efekta u skladištima organizacijskog znanja kod promjenjivih i dinamičkih okruženja [21]. To osigurava razvoj novih znanja, koja mogu biti jako važna u ponašanju organizacije. U virtualnim okruženjima i društvenim mrežama, povećava se potreba za organizacijskim učenjem prvenstveno radi dijeljenja i širenja informacija. Organizacijsko učenje, u pravilu, uvijek započinje s pojedinačnim

učenjem, te prerasta u kolektivno učenje, pri čemu se misli na grupe, timove ili nivo cijele organizacije. Sposobnosti učenja uključuju:

- otvaranje novih perspektiva
- definisanje ličnih sklonosti
- rad sa neobrađenim podacima
- realno razumijevanje

Organizacijsko učenje može ponuditi organizaciona rješenja kod sljedećih pitanja [22]:

- sistemskih rješavanja problema
- kreativnih eksperimenata
- učenja na osnovu ranijeg iskustva
- učenja na osnovu najboljih praksi sličnih sistema
- brze i efikasne transformacija znanja u organizaciji

Organizacijska memorija osigurava prikupljanje, prezentaciju, pohranjivanje i distribuciju znanja u organizaciji . Formirana je od četiri tipa memorije [23]:

- *radna memorija* sadrži eksplicitno referentno znanje, dokumente, korisne programe koji se koriste za poslovanje
- *organizacija memorija* je vezana za aktivnosti organizacije i njenih partnera
- *individualna memorija* sadrži sposobnosti i kompetencije pojedinaca
- *projektna memorija* pohranjuje definicije projekta, aktivnosti, resurse, istoriju projekta i rezultate

Iz perspektive domene sistema organizacijska memorija se dijeli na:

- *internu memoriju*, sadrži znanje i informacije unutar organizacije
- *vanjsku memoriju*, sadrži znanje i informacije iz okoline

Može se zaključiti da je OL i OM zavise manje o tehnologiji, a više o ljudskim resursima. Stvaranje novog znanja podrazumijeva [24]:

- generisanje novih značenja, ideja i ponašanja
- spajanje eksplicitnog i prešutnog znanja

- realizaciju spiralnog znanja pojedinca, grupe ili organizacije

Postoje četiri modela stvaranja znanja [24]:

- *socijalizacija* - pretvaranje prešutnog znanja u novo prešutno znanje pomoću socijalne interakcije i razmjene znanja između članova virtualne organizacije (npr. pomoću instrukcija)
- *eksternalizacija* - prevođenje prešutnog znanja u novo eksplicitno znanje (proizvodnju novog nastavnog materijala u školi)
- *internalizacija* - stvaranje prešutnog od eksplicitnog znanja (stvaranje značenja iz pisanih dokumenata ili upravljanje sadržajem)
- *kombinacija* - stvaranje novih eksplicitnih znanja prema klasifikaciji, kategorizaciji i sistematizaciji (npr. upotreba *rudarenja podataka (DM)* ili statističkih analiza nad različitim skladištima podataka koristeći *upite za otkrivanje znanja (KDQL)* [25])

Prikupljanje znanja podrazumijeva otkrivanje izvora znanja i korištenje internog znanja, te se može organizovati različitim metodama i bez informacionih tehnologija. Jedan od najčešćih je verbalni *diskurs*⁶.

Znanje je moguće prikupiti od pojedinaca ili grupa. Osnovne tehnike pojedinačnog prikupljanja znanja su: intervjui, neposredno opažanje, upitnici, analiza prethodnih slučajeva, introspekcije, simulacije i sl. [28].

Vrlo interesantno pitanje je: "*Mogu li se radne grupe posmatrati kao distributivni sistem za upravljanje znanjem?*" [26]. Taj problem nije dovoljno istražen i zastupljen u literaturi. Upravljanje znanjem je više koncentrisano na integraciju znanja u grupama samo kao aditiv. Također, neki autori smatraju da su distributivni sistemi za upravljanje znanjem jako komplikovani u virtualnim organizacijama [10]. Uopšteno, mreže znanja koriste

⁶ Diskurs (lat. discursus) je imenica koja znači govor, razgovor, odnosno raščlanjivanje, analizu. Danas se često rabi u semantici, jezikoslovnoj disciplini gdje označava lingvističku jedinicu sastavljenu od nekoliko rečenica.

informacione tehnologije za dijeljenje znanja i mudrosti među pojedincima, grupama i jedinicama širom organizacije, pa čak i sa vanjskim organizacijama [15].

Kao asinhrono kolaborativno okruženje, u praktičnom dijelu ovog rada opisana je integracija modula *Wiki stranica* i *Pitanja i odgovori* u *FITCKM* sistemu (potpoglavlje 3.1.1.1). U narednim potpoglavljima predstavljene su mogućnosti ovih modula.

2.1.4 Wiki modul

U bliskoj prošlosti je bilo potrebno ići u biblioteku i satima čitati ili prepisivati tekstove neophodne za školovanje. Poseban značaj pridavao se enciklopedijama, kao vjerodostojnim i sistematizovanim skupovima objašnjenja i tumačenja. Porastom upotrebe ICT-a, porasle su i potrebe korisnika, te su zbog jednostavnosti i lakoće upotrebe, elektronski formati dokumenata zasjenili one štampane. Iz ovog ugla gledano, internet je danas jedna velika biblioteka bogatog sadržaja i skoro sve informacije su udaljene na samo par „klikova“.

Na web stranici američkog programera *Varda Kaningema*, 1995. godine je prvi put predstavljena revolucionarna ideja, da bilo ko može uređivati članke online enciklopedije. Vrlo brzo je stekla ogromnu popularnost. Upravo ovaj autor je i tvorac naziva “*Wiki*”, što znači brzo [117]. Spomenuta web stranica je ustvari preteča, sada već popularne *Wikipedije*, enciklopedije slobodnog sadržaja, koju razvijaju dobrovoljci uz pomoć *Wiki* softvera. Danas, *Wiki* predstavlja dio serverskog softvera koji dozvoljava korisnicima da kreiraju i vrše izmjene nad člancima, uz pomoć bilo kog pretraživača [34].

Primjena *Wiki*-ja u učenju ogleda se u obezbjeđivanju relevantnih tekstova iz velikog broja oblasti. Jedinstvena baza tekstova različitih korisnika, omogućava efikasno pretraživanje, kao i izmjenu članaka [34].

U kontekstu kolaborativne podrške korisnicima, *Wiki* nudi različite alate koji omogućavaju saradničko formiranje baza znanja. Akcent je na aktivnoj participaciji korisnika, koji, uvidom u učinak svojih kolega, stiču nova znanja, istražuju, rješavaju probleme i izrađuju projekte. Svaki od autora može predlagati izmjene rada, koje su vidljive i ostalim autorima pa se mogu dodatno mijenjati.

Wiki predstavlja idealan izbor za naučnike koji žele da objave svoje istraživačke članke na Internetu, a nisu upućeni u osnove HTML kodiranja, odnosno web programiranja. Također, oni žele podijeliti svoja iskustva komentarišući članke koji im se učine zanimljivim. *Wiki* alati povezuju istraživače sa geografski udaljenih lokacija, te je njihova ekspanzivna upotreba u praksi opravdana i logična.

U ovom potpoglavlju, bit će predstavljeni *Wiki* alati i tehnike za kolaborativno učenje, podučavanje i istraživanje.

2.1.4.1 Primjena Wiki-ja u kolaborativnom učenju

Wiki je, kao što je već naglašeno, vrlo moćan sistem koji nudi različite mogućnosti saradnje i razmjene znanja. U kolaborativnom učenju, korisnici mogu da rade u heterogenim grupama za podršku učenju. Kolaborativno učenje omogućava pozitivne međuzavisnosti između članova grupa, pojedinačne odgovornosti, interakcije licem u lice, što rezultira razvojem individualnih kolaborativnih vještina.

Rad u grupi omogućava postizanje boljeg rezultata i članovi imaju pristup velikoj količini različitih informacija, za razliku od korisnika koji individualno obavljaju neki zadatak. Velika podrška za kolaborativno učenje je mogućnost saradnje korisnika iz različitih sredina. Učenje podržano ICT-om, je znatno olakšano uz pomoć *Wiki*-ja, što obrazovanju daje potpuno novu dimenziju. Također, prilikom kolaborativnog rada dolazi do poboljšanja korisničke interakcije, distribucije znanja i razmjene stručnosti u zajednici.

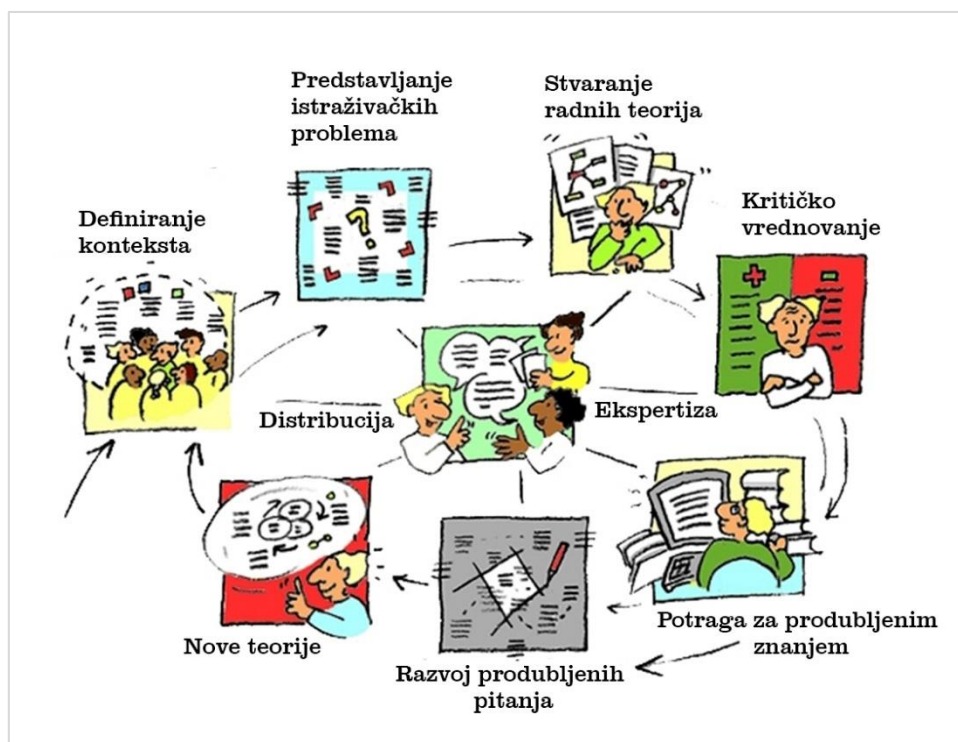
Između ostalog, *Wiki* u procesu kolaborativnog učenja poboljšava asinhronu komunikaciju i kooperativno učenje između korisnika i promovira saradnju umesto konkurencije. Kolaborativno učenje postaje još snažnije kada se odvija u kontekstu zajedničke prakse, koja se sastoji od članova koji rade u kolektivnom učenju određenom domenu i učenje postaje zajednički proces u okviru radne grupe. *Wiki* karakteriziraju i neki elementi od fundamentalnog značaja za uspješnu grupnu realizaciju zadataka, uključujući i virtualno prisustvo, lako i jednostavno učešće u izradi zadataka, vezu sa širokim krugom korisnika, lični i zajednički identitet i interakcija.

Kolaborativno učenje treba da uključi korisnike u učenju smisleno i kritično tako da:

- aktivno i manipulacijski angažuje učenike u interakcijama i istraživanju sa nastavnim materijalima
- konstruktivno i refleksno omogućiti studentima/učenicima da integrišu nove ideje uz prethodna znanja i omogućiti učenje kroz refleksiju
- pruže mogućnost studentima da manipulišu svojim ciljevima učenja i prate napredak dostizanja ciljeva
- autentično i izazovno u realnom svijetu omogućiti razumjevanja i transfer učenja u novim situacijama
- pruže studentima mogućnost da komuniciraju jedni sa drugima, da razjasne i razmjenjuju ideje, traže pomoć i diskutuju rješenja

U posljednjih nekoliko godina, oni koji su uključeni u online učenje imaju veću svijest o prednostima kolaborativnog učenja u virtualnom okruženju korištenjem *Wiki*-ja. Istraživanja pokazuju da veliki broj korisnika, koji su koristili *Wiki*-je za kolaborativno učenje, imaju pozitivno mišljenje o ovakvom načinu učenja.

Online razvoj saradničkih tekstova pruža nove mogućnosti, ne samo kada je reč o usvajanju znanja, već podstiče i takozvano saradničko razmišljanje. Također, sve je više nastavnog osoblja koji koriste ovaj način saradnje i dijeljenja znanja. Kreiranje nastavnog sadržaja postaje jedan kreativan proces, a i dovodi do primjene principa demokratičnosti u obrazovanju. Na ovaj način savremene tehnologije se uključuju u nastavni proces. Često se uspostavlja kulturna i prijateljska saradnja između korisnika [34].



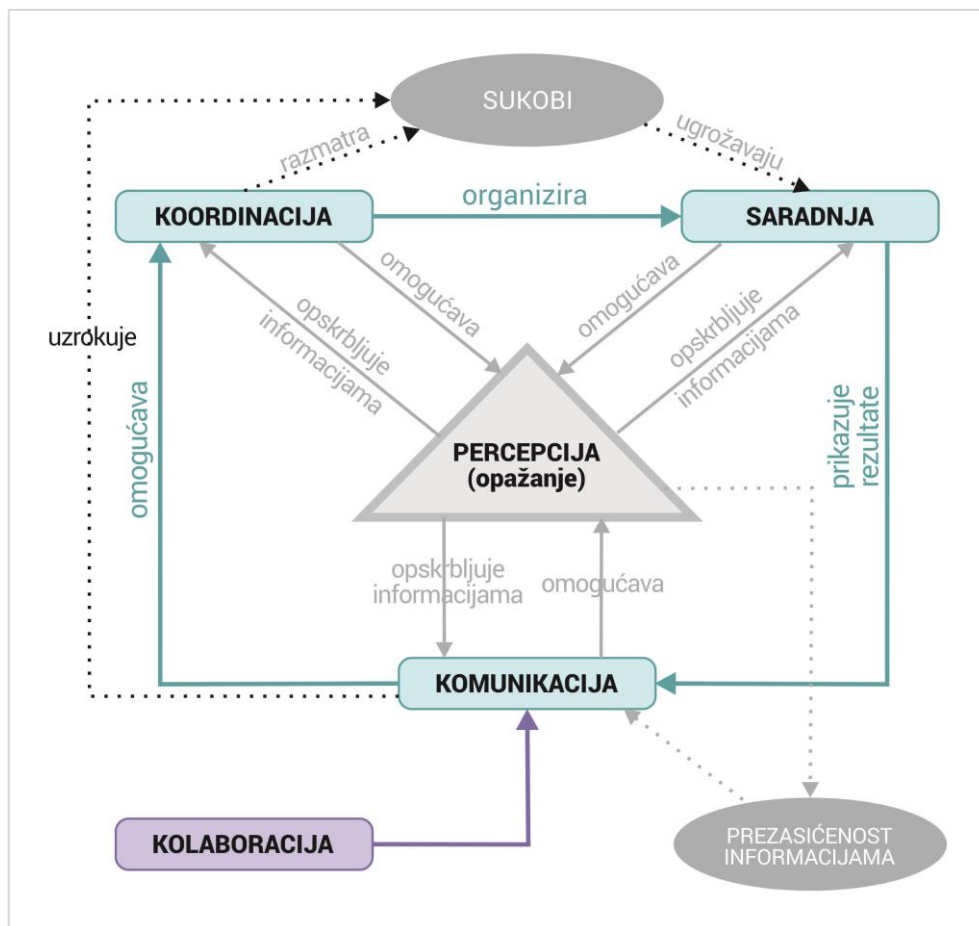
Slika 2. Kolaborativno učenje [118]

U nastavku su predstavljene osnovne karakteristike kolaborativnog učenja (Slika 2.) [35]:

- članovi u grupi, razmjenjuju ideje i misli, zajednički uče i istražuju
- akumulacija znanja o temi, razvijanje jednostavnih vještina, rješavanje problema
- kolaborativno učenje se smatra distribuiranim procesuiranjem ili inteligencijom, što daje veći spektar mogućih kombinacija ideja i stavova, sa manjim posljedicama i puno većom vjerovatnoćom dolaska do najboljeg mogućeg rešenja postavljenog problema
- početna ideja pojedinca se u toku kolaboracije nadograđuje (oplemenjuje) zahvaljujući razmišljanjima ostalih članova; takvim pristupom pojedinac se lakše oslobađa predrasuda, odnosno ograničavajućih stavova koji su prepreka konstruktivnoj promjeni mišljenja kao rezultatu efektivnog učenja
- svaka početna ideja nosi potencijal generisanja novih ideja u kolaborativnoj zajednici i veliku je vjerovatnoću aktiviranja naprednih ideja i rješenja, zbog čega se stvara

pozitivno tumačenje, razvija samopouzdanje i motivacija svakoga člana, a zajednicu čini učinkovitijom

- uz pomoć kolaborativne grupe lakše je uvidjeti pogrešan stav i razmišljanje, te je moguće je pravovremeno djelovati
- korisnici preuzimaju odgovornost za samostalno učenje, kao i za učenje u grupi
- u kolaborativnoj zajednici događaju se lične i društvene preobrazbe; korisnici se ohrabruju da se suprotstave autoritetima, pa kolaborativno učenje vodi razvijanju stvarne “demokratije” u kojoj svako ima pravo glasa
- alternativni način provjere znanja studenta/učenika
- razvijanje socijalnih vještina kolaborativnih partnera



Slika 3. Koncept kolaborativnog učenja [118]

Grafičko predstavljanje osnovnih komponenti i veza između njih (Slika 3.) daje jasniju viziju o konceptu kolaborativnog učenja.

2.1.4.2 Primjena Wiki-ja u proučavanju i istraživanju

Najveća prednost Wiki-ja je u podrška učenju na daljinu i istraživačkom radu, a najveća vrijednost leži u ogromnoj bazi podataka, koja sadrži kvalitativne i kvantitativne podatke [36]. Koncept „otvorenog“ Wiki-ja dopušta da korisnici slobodno postavljaju ili mijenjaju sadržaje, ako je riječ o tekstu ili prezentaciji, što ima svoje prednosti ali i mane.

Prednosti se ogledaju u tome da postoji mogućnost unaprjeđivanja sadržaja i pregleda historije promjena, čime korisnici mogu pratiti napredak u određenoj oblasti ili drugih korisnika. Također, moguće su različite analize korisnika, sadržaja, diskusija i sl. Globalno, gledajući iz perspektive različitosti i količine podataka, moguće je sprovesti vrlo široka istraživanja.

Najveći nedostatak ovog koncepta, obzirom da su sadržaji otvoreni za izmjene, je validnost i kvalitet objavljenih podataka.

2.1.4.3 Wiki alati

Wiki alati predstavljaju web stranice čiji sadržaj može lako da se kreira i da bude mijenjan od strane zajednice, koje najčešće uključuju i posjetioce. Postoje desetine Wiki softvera koje imaju svoje prednosti i mane, ali suština svakog od tih softvera je ista. Radi se o web platformama koje kreiraju web stranice i napravljene su tako da ih bilo ko može mijenjati i uređivati. One služe kao informacije drugim ljudima koji žele da vide sadržaj te stranice. Još jedna od bitnih karakteristika jeste povezivanje sa drugim Wiki stranicama.

Trenutno, neki od najpopularnijih Wiki alata su: *MediaWiki* [37], *DokuWiki* [38], *TermWiki* [39], *WikiEducator* [40], *PmWiki* [41], *FosWiki* [42], *MathWiki* [43].

U ovom radu je razvijen modul koji je po svojim karakteristikama najbliži alatu *MediaWiki*. *MediaWiki* predstavlja softver izdat po GNU⁷ licencom. Ovaj softver je dizajniran za pokretanje na velikom serveru, tako da ga istovremeno može koristiti veliki broj korisnika i može podržati veliki broj različitih stranica. *MediaWiki* koristi Wiki-tekst

⁷ GNU je operativni sistem sastavljen isključivo od slobodnog softvera. Njegovo ime je rekurzivna skraćenica za "GNU Nije Unix" (eng. "GNU's Not Unix").

format, tako da korisnici ne moraju posjedovati programerske vještine, prije svega XHTML i CSS, kako bi uređivali stranicu. Kada korisnik izmjeni određenu stranicu i sačuva promjene, *MediaWiki* automatski tu promjenu upisuje u bazu podataka, ali ne briše prethodnu verziju u slučaju da dođe do zloupotreba [37].



Slika 4 MediaWiki - logo

U prethodnom poglavlju su ukratko predstavljene mogućnosti upotrebe *Wiki*-ja u kolaborativnom okruženju, kao i mogućnosti njene primjene u proučavanju i istraživanju. Osnovne mogućnosti *MediaWiki* implementirane su u praktičnom dijelu ovog rada, odnosno u *FITCKM* sistemu, tačnije u *Wiki* modulu simbolično nazvanom „*FITWiki*“. Više o implementaciji ovog modula nalazi se u potpoglavlju 3.1.1.1.

2.1.5 Internet forum

U srednjem vijeku, svaki grad je imao svoj trg koji se u narodu nazivao forum. Stanovnici su se okupljali u tom dijelu grada da bi razmijenili novosti, čuli najnovije političke vesti, ili obavljali određeni vid trgovine. 1996. godine, forumi u centrima gradova zamijenjeni su modernijim forumima, Internet forumima. Internet forum predstavlja specifičnu web aplikaciju koja omogućuje posjetiocima raspravu o određenim temama. Osnovna razlika između tradicionalnih i Internet foruma je što učesnici Internet foruma ne moraju navesti pravi identitet, odnosno imaju mogućnost obavljanja određene aktivnosti, najčešće iskazivanja svog mišljenja, kao anonimni učesnici. Danas, tradicionalni forumi su vrlo rijetka pojava, te pojam „forum“ u modernom dobu nedvosmisleno predstavlja Internet forum. Po strukturi, Internet forum sadrži veći broj tematskih podforuma u kojima učesnici mogu otvoriti diskusiju na određenu temu. Ta diskusija može biti pitanje, objava neke novosti, konstatacija i sl.

Svaki forum ima svoje osoblje administratore i moderatore, odnosno korisnike koji održavaju rad foruma. Administratori imaju najveće privilegije i jedino oni mogu uređivati podforume, te mogu izvršavati ostala važna podešavanja foruma. Moderatori takođe uređuju forum, ali su njihova prava ograničena. Neki od zadataka moderatora su brisanje tema koje se ponavljaju, prebacivanje teme u odgovarajući podforum, zabranjivanje pristupa forumu članovima sa neadekvatnim ponašanjem i sl. Administratori i moderatori foruma određene teme mogu proglasiti bitnim ili temama koje sadrže važna obavještenja, te se takve obično prikazuju pri vrhu, iznad ostalih.

Na forumu, obično, osim imena, prezimena i drugih informacija o članovima, prikazuje se i mala slika člana, koja se naziva *avatar*. Pojedini forumi sadrže i dodatne opcije, kao što su zahvalnice i reputacije. Ukoliko učesnik smatra nečiju objavu značajnom, vlasniku objave se moguće zahvaliti na korisnoj poruci. Ispod poruke, biće ispisana imena učesnika koji su se zahvalili na korisnoj poruci. Također, učesnicima je moguće dodijeliti i reputaciju koja može biti pozitivna ili negativna, zavisno od ličnog stava prema određenom članu.

Značaj Internet foruma za učesnike ogleda se u mogućnosti kolaboracije, prvenstveno pronalaženja odgovora na različita pitanja putem diskusije sa ostalim učesnicima. Drugi modul *Pitanja i Odgovori*, implementiran u praktičnom dijelu *FITCKM* sistema, je razvijen na principu Internet foruma u nešto izmijenjenom obliku. Logika je vrlo slična, s tim da je za potrebe teme rada, omogućeno ocjenjivanje postavljenih pitanja i odgovora. Prilikom kreiranja pitanja, logirani korisnik (učesnik) foruma dužan je da izabere kategoriju i podkategoriju kojoj pitanje pripada, kao i da unese odgovarajuće oznake. Svi korisnici, objave, ocjene i oznake uključeni su u sistem preporuke, te se logiranom korisniku u vidu hiperveza preporučuju značajni i zanimljivi sadržaji, kao i slični korisnici. Također, korisnicima je omogućeno i napredno *pretraživanje cijelog teksta* (eng. *full-text searches*).

Cilj implementacije modula *Pitanja i Odgovori (FITQ&A)* je podrška modulu *FITWiki*, a svrha poboljšanje udobnosti korisnika u *FITCKM* sistemu. Ovaj modul je razvijen na principu jednog od najpoznatijih IT foruma danas, *Stack Overflow-a* (<http://stackoverflow.com/>) i omogućava većinu originalnih funkcionalnosti.

2.2 Inteligentni programski agenti

Krajem XX vijeka, na polju distribuirane umjetne inteligencije počeo je razvoj inteligentnih programskih agenata kao nove oblasti. U stručnoj literaturi trenutno je moguće pronaći različita tumačenja, klasifikacije, poglede, ideje, načine implementacije i sl., što upućuje na činjenicu da ova oblast obuhvata vrlo široko područje. Postavljene hipoteze na ovu temu, još uvijek su u iščekivanju čvrstih dokaza, odnosno praktičnih rješenja, koja bi ih potvrdila. Ipak, dosadašnja dostignuća su postigla ogroman uspjeh i potencijalne mogućnosti inteligentnih programskih agenata se smatraju budućom velikom prekretnicom u IT.

Vrlo je teško konkretno definisati pojam „agent“⁸, njegovu smisao i korištenje, jer je prisutan u različitim formama:

- *agenti ljudi* (u fizičkom svijetu);
- *robots* (roboti);
- *softbots* (softverski agenti);
- *taskbots* (agenti koji su specijalizovani za određene zadatke);
- *mobilebots* (agenti koji djeluju u dinamičkom i neizvjesnom okruženju);
- *knowbots* (lični asistenti, imaju stručno znanje o specifičnim oblastima);
- i sl.

U nastavku su predstavljena neka od tumačenja inteligentnih programskih agenata:

- „*Kompjuterski program koji je u interakciji sa okruženjem i ima sposobnost da fleksibilno i samostalno reaguje u skladu sa ciljevima koji su mu postavljeni naziva se programski agent. On je ustvari program koji djeluje u ime svog vlasnika, obavljajući pri tom složene informacijske i komunikacijske poslove u okruženju*“ [49]-[50]

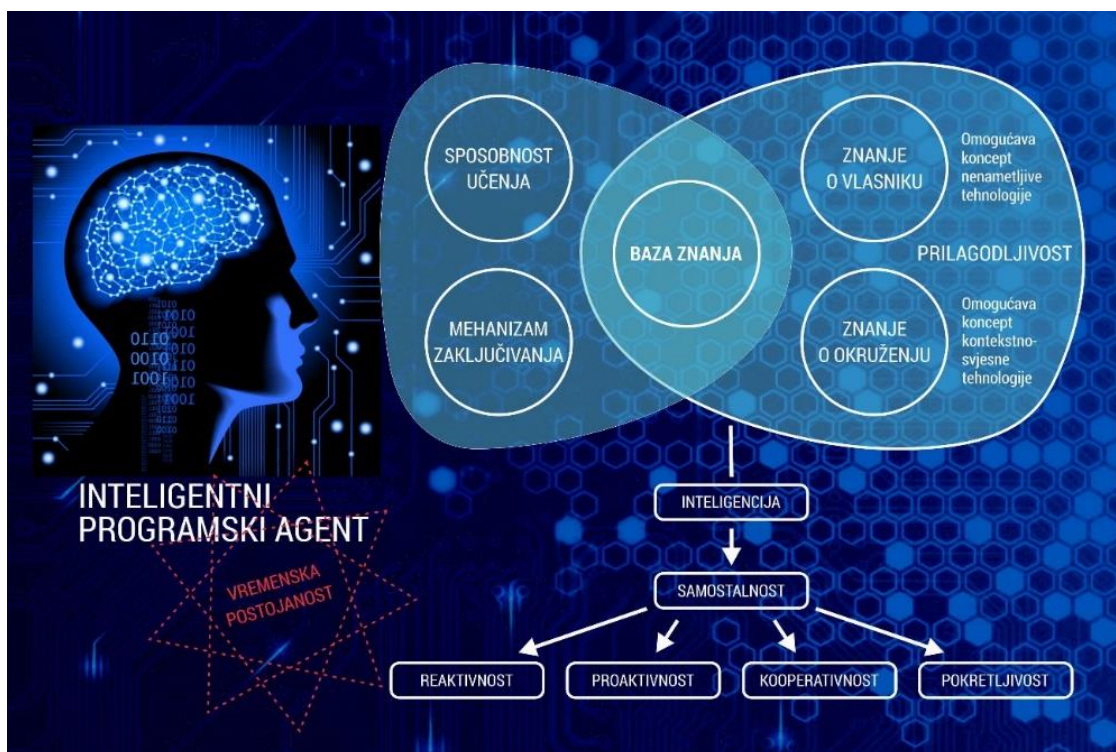
⁸ Agent (lat. agens - koji djeluje, radi) 1. Opšte značenje, posrednik u različitim poslovima. 2. Policijski aagent, osoba koja izviđa, otkriva, sprečava protuzakonito djelovanje. 3. Trgovački agent, osoba koja zaključuje ili zastupa tuđe kupoprodajne poslove.

- „*Inteligentni programski agenti su autonomni kompjuterski programi i njihova okolina dinamički utiče na njihovo ponašanje i strategije za rješavanje problema*“ [100]
- „*Objekt je apstrakcija na visokom nivou, koji opisuje metode i attribute jedne softverske komponente. Agent je, međutim, je na višem nivou apstrakcije softvera koji pruža izrazito pogodan i snažan način za opisivanje složenih softverskih entiteta. Umjesto da se definiše u smislu metoda i atributa, agent se definiše u smislu ponašanja. To je važno saznanje zbog toga što razvoj sistema zasnovanog na inteligentnom programskom agentu predstavlja prvenstveno definisanje ponašanja agenta, umjesto identifikacije klasa, metoda i atributa*“ [101]
- „*Inteligentni agent je softver koji pomaže ljudima i djeluje u njihovo ime. Inteligentni agenti dopuštaju ljudima da poslove koje trebaju uraditi prepuste kompjuterskom programu. Agenti mogu obavljati poslove ponavljanja, podsjećanja na zaboravljene stvari, inteligentnog rezimiranja složenih podataka, učenja od čovjeka, pa čak i davanja preporuka*“ [102]
- „*Agent je sve ono što senzorima opaža svoje okruženje i utiče na to okruženje pomoću izvršnih uređaja (aktuatora)*“ [103]

Navedena tumačenja dokazuju da se ova oblast zaista proteže duboko i široko u različitim sferama života i nauke. Možda je najlakše za shvatiti uopšteno tumačenje da je agent entitet koji nešto radi u ime nekoga. S aspekta softvera, uprošteno tumačenje bi bilo da inteligentni programski agent treba omogućiti samostalnu prilagodljivost softvera, kao odgovor na promjene koje se dešavaju u okruženju.

2.2.1 Struktura inteligentnog agenta

Slično pokušajima definisanja inteligentnog programskog agenta, u literaturi se mogu pronaći i različita tumačenja o njegovoj strukturi. Na *Slici 5.* grafički su predstavljene neke od karakteristika inteligentnih programskih agenata i odnosi između njih [44]-[51]-[52]-[53]-[54]-[55].



Slika 5. Struktura inteligentnog programskog agenta

Trenutno u akademskim krugovima ne postoji jasan zajednički stav ili definicija koja objašnjava strukturu, ali opšte prihvaćena činjenica je da inteligentni programski agenti mogu imati slabi i jaki koncept.

U oblasti IT, termin „agent“ najčešće se koristi za agent-bazirani hardverski ili, mnogo češće, softverski sistem sa slabim konceptom inteligentnog programskog agenta, koji ima sljedeće karakteristike:

- *samostalnost* - agenti djeluju bez direktne intervencije ljudi ili drugih agenata i imaju određenu vrstu kontrole nad svojim postupcima i internim stanjima [104]
- *kooperativnost* – agenti su u interakciji s drugim agentima i (eventualno) ljudima, komunicirajući pomoću jezika razumljivom agentu [105]
- *reaktivnost* - agenti osjete svoje okruženje i blagovremeno odgovaraju na promjene koje se javljaju u njemu; okruženje može biti fizički svijet, korisnik preko grafičkog korisničkog interfejsa, kolekcija drugih agenata, Internet, ili

možda sve navedeno u kombinaciji; ova karakteristika može zahtijevati od agenta da većinu svog vremena provodi u *stanju spavanja* (eng. *sleep state*), iz kojeg će se probuditi (aktivirati) ako osjeti određene promjene u svom okruženju (npr. dolazak e-mail poruke) [106]

- *proaktivnost* - agenti ne djeluju samo kao odgovor na svoje okruženje, oni su u mogućnosti preuzeti inicijativu, kada smatraju da je potrebna takva vrsta akcije
- *vremenski kontinuitet* – agent je kontinuirano pokrenuti proces (aktivan ili pasivan), a ne jednokratni proračun ili skripta koja je mapirana jednim ulazom i jednim izlazom, a zatim prekinuta
- *orijentiranost prema cilju* - agent je sposoban za rukovanje složenim zadacima na visokom nivou; odluke o dijeljenju zadatka na manje zadatke i izbor optimalnog načina rješavanja tih zadataka treba da donese agent samostalno

Za neke istraživače, posebno one koji rade na polju umjetne inteligencije, pomenute karakteristike nisu dovoljne, te u njihovoj interpretaciji termin “agent“ ima jače i konkretnije značenje. Njihovo tumačenje je da agent, osim prethodno navedenih karakteristika, treba biti osmišljen i implementiran koristeći koncepte koji se mnogo više oslanjaju na osobine ljudske inteligencije. Na primjer, danas u umjetnoj inteligenciji uobičajeno je agent ima mentalne karakteristike kao što su znanje, vjerovanje, namjere i obaveze [107]. Neki istraživači su otišli i korak dalje, te razmatraju emocionalnost agenata [108]. Zanimljiv primjer je uloga agenata u vizualnom predstavljanju ljudi, pomoću tehnike kao što je grafička animacija, u kojima animirani likovi oponašaju pokrete tijela ili lica [109].

Inteligentni programski agenti koji odgovaraju jakom konceptu, osim prethodno navedenih, obično imaju jednu ili više od sljedećih karakteristika:

- *pokretljivost* - sposobnost agenta za kretanje kroz elektronsku mrežu [110]
- *dobronamjernost* – pretpostavka je da agenti nemaju konfliktna stanja i ciljeve, te da će svaki agent uvijek pokušati učiniti ono što se traži od njega [111]

- *racionalnost* - je gruba pretpostavka da će agent djelovati kako bi postigao svoje ciljeve, te da neće djelovati na takav način da se spriječi njihovo ostvarenje u uslovima stvorenim na osnovu njegovih uvjerenja [112]
- *adaptivnost* - agent bi trebao biti u mogućnosti da se prilagodi okruženju kao i navikama, radnim metodama i željama svojih korisnika
- *kolaboracija* - agent ne bi trebao bez razmišljanja prihvatiti (ili izvršiti) sve instrukcije imajući u vidu mogućnost pojave korisničkih grešaka (npr. dati nalog koji sadrži suprotne ciljeve), izostavljati važne informacije i davati dvosmislene informacije; agent bi trebao provjeriti stvari postavljanjem pitanja korisniku ili koristiti ugrađeni model korisnika za rješavanje sličnih problema [113]

2.2.2 Vrste inteligentnih programskih agenata

Autori su grupirali agente u pet klasa na osnovu stepena percepcijske inteligencije i mogućnosti [56]:

- *jednostavni refleksivni agenti;*
- *refleksivni agenti bazirani na modelu;*
- *agenti bazirani na cilju;*
- *agenti bazirani na korisnosti;*
- *učeći agenti;*

Međutim, prethodna klasifikacija mora se uzeti sa rezervom, jer, kao što je već rečeno, agenti su prisutni u različitim oblicima i koriste se za različite namjene, te se mogu izdvojiti još neke vrste inteligentnih agenata[56]:

- *odlučujući agenti* (pomažu u donošenju odluka)
- *ulazni agenti* (procesuiraju ulaze, senzori)
- *procesirajući agenti* (rješavaju probleme kao što su prepoznavanje govora)
- *prostorni agenti* (vezani su za fizički, realni svijet)
- *svjetski agenti* (kombinacija ostalih klasa agenata omogućujući autonomno ponašanje)

- *agenti vjerovatnoće*
- *fizički agenti* (entitet koji percipira kroz senzore i djeluje kroz pokretače)
- *vremenski agenti*
- i sl.

Razvoj aplikacija baziranih na inteligentnim programskim agentima uvijek predstavlja novi izazov, te se vrlo često koriste različite kombinacije pomenutih klasa i vrsta agenata.

2.2.3 Agent-bazirani programski jezici

Zbog specifičnih osobina i zahtjeva koji se postavljaju pred agent-orijentirane aplikacije, one se moraju programirati u agent orijentiranim jezicima. U posljednjoj deceniji razvijeno je mnogo agent-baziranih programskih jezika baziranih na popularnom modelu BDI (*eng. Belief, Desire and Intention*). U jezike koji podržavaju inteligentne programske agente integrirane su logička i objektno-orijentirana paradigma programiranja. Logička paradigma programiranja obezbjeđuje deklarativnu bazu znanja i sisteme zaključivanja, dok objektno-orijentirana paradigma obezbjeđuje sposobnost enkapsulacije agenta i olakšava njegovu migraciju. Većina ovih jezika izgrađena je na bazi Jave, jer ona podržava objektno-orijentirano programiranje, višestruke teme, migraciju programskog koda, virtualne mašine koje podržavaju homogenost u heterogenim okruženjima i dovoljan nivo bezbjednosti za objekte [116].

Neki od najpopularnijih istorijskih jezika i agent-baziranih sistema su: *AgentSpeak(L)*, *AgentTalk*, *Aglet*, *APL*, *dMARS*, *Jade*, *Jack*, *CLAIM*, *KABUL* i *Odyssey*. Dizajnirani su uglavnom za jednosprocesorske uređaje. U novije vrijeme razvili su se novi jezici koji podržavaju savremene paradigme u programiranju za mnogo veće razmjere i visoku produktivnost razvoja softvera na masivnim paralelnim klasterima procesora. Ti jezici su nasljednici *High Performance Fortran* i *UPC* (*eng. Unified Parallel C*). Neki od najpoznatijih su: *Chapel*, *Titanium* i *X10*, i bazirani su na *PGAS* (*eng. Partitioned Global Address Space Languages*) [116]. Osim agent-orijentiranih jezika u kojima su pisani agenti (interni jezici), postoje i jezici koji omogućuju komunikaciju između agenata (eksterni jezici) kao npr. *KQML* (*eng. Knowledge Query and Manipulation Language*).

2.2.4 Područje primjene inteligentnih programskih agenata

Napredak IT industrije zahtijeva od modernih aplikacija ne samo da odgovaraju na zahtjeve, nego da i inteligentno učestvuju u aktivnostima u svom okruženju, te se neprestano prilagođavaju trenutnom stanju svog okruženja i na taj način aktivno traže način kako da što bolje ostvare interese entiteta kojeg zastupaju [44]-[45]-[46]-[47]-[48].

Postoji širok spektar aplikacija baziranih na inteligentnim programskim agentima u različitim oblastima kao što su: personalizovano upravljanje informacijama (npr. e-mail filtriranje), elektronsko poslovanje (npr. pronalaženje informacija za nabavku ili kupovinu), upravljanje složenim poslovnim i industrijskim procesima (npr. zakazivanje sastanaka i kontrola zračnog prometa), različite simulacije i sl.

Navedene aplikacije, odnosno zadaci mogu se generalno svrstati u pet kategorija [100]:

- 1) *agenti posmatrači* - traže konkretne informacije
- 2) *učeći agenti* - predviđaju preferencije na osnovu prethodnog znanja o ponašanju entiteta
- 3) *agenti za kupovinu* - pronalaze "najbolju cijenu za određenu stavku"
- 4) *agenti sakupljači informacija* - entitetu obezbjeđuju traženje informacija na inteligentan način
- 5) *agenti pomagači* - obavljaju poslove samostalno bez ljudske interakcije

U skladu sa obimom ovog rada, temu inteligentnih programskih agenata potrebno je iz opšteg usmjeriti na konkretno. Kao što je u uvodnom dijelu spomenuto, posebna pažnja će biti posvećena problemu preopterećenosti informacijama, a u rješavanju tog problema agenti mogu odigrati vrlo važnu ulogu. Ova problematika može se podijeliti na dva generalna problema [100]:

- *filtriranje informacija* – prolaskom kroz ogromnu količinu informacija potrebno je pronaći nekolicinu koja je relevantna u konkretnom slučaju
- *sakupljanje Informacija* – u nedostatku ili nedostupnosti informacija pretraga za relevantnim informacijama je znatno teža i vremenski zahtjevnija

U oblasti sistema preporuke kolaborativnih okruženja za upravljanje znanjem, inteligentni programski agenti se mogu koristiti za više namjena. Konkretno, u *FITCKM* sistemu programski agenti će imati ulogu da samostalno i dinamički izvršavaju kompleksne kalkulacije uređivanja i filtriranja podataka, na osnovu promjena u svom okruženju. Rezultat kalkulacija je zapisan u bazu podataka u ulozi novog znanja koje treba pomoći agentu da „razumije“ novonastalo stanje okruženja, odnosno da usvoji novo znanje, koje će opet predstavljati podlogu za usvajanje budućih znanja. Ovaj agent bi se mogao svrstati u kategorije agenata posmatrača i sakupljača informacija. Više o konkretnoj ulozi i simulaciji programskog agenta u *FITCKM* sistemu se nalazi u potpoglavlju 4.1.

2.3 Sistemi preporuke

Jedan od standardnih problema koji prati razvoj ICT-a, je rapidno povećanje količine informacija, odnosno problem preopterećenja informacijama. Okruženja bogata informacijama, koja ne koriste metode za adekvatno upravljanje njima, mogu biti isto neefikasna kao i okruženja sa nedostatkom informacija. Kako manjak informacija predstavlja očigledan problem, višak dovodi do preopterećenja, tako da prepoznavanje relevantnih informacija u ogromnom rastućem skupu predstavlja vrlo kompleksan zadatak. Najčešći razlog tzv. konfuzije, u okruženjima koja obiluju informacijama, je činjenica da su informacije često vrlo slične, bez opisanih specifičnosti. Optimalnu količinu informacija, u smislu fizičkog brisanja ili dodavanja, je nemoguće postići i kontrolisati, jer takvo nešto zahtijeva kompleksne metode koje bi negativno uticale na performanse sistema. Također, nemogućom se čini i odluka za brisanje određene informacije, jer svaka informacija može biti potencijalno relevantna u određenom kontekstu.

S druge strane, ako se velika količina informacija posmatra kao bogatstvo koje treba adekvatno iskoristiti, javlja se potreba za izdvajanjem, filtriranjem, rangiranjem i sl. Osnovni cilj je preporuka korisnih, značajnih i interesantnih sadržaja koji mogu biti različiti: članci, knjige, muzički albumi, igre, filmovi, vijesti i sl. U ovom radu će se za različite vrste sadržaja koristiti termin „*resurs*“, što je u skladu sa terminologijom koja

se koristi u akademskoj zajednici. Osnovni zadatak sistema za preporuku je da identifikuje resurse potencijalno „važne“ korisniku. U procesu identifikacije sistem bi trebao predvidjeti „važnosti“ resursa, uporediti ih, te korisniku preporučiti „najvažnije“ rezultate. Za takve resurse u radu će se koristiti termin „relevantan“.

Pretpostavka od koje treba krenuti pri razvoju sistema preporuke je ljudska sposobnost u donošenju promišljenih odluka na osnovu dostupnih informacija [57]. Slijedeći tu ideju, kao pomoć u neprestanoj borbi čovjeka sa suvišnim informacijama, počeo je razvoj različitih algoritama u području *umjetne inteligencije* (eng. *artificial intelligence*, AI). U okvirima sistema preporuke, oni imaju veoma važnu ulogu u poboljšavanju performansi pristupa informacijama. Neki autori ovakve koncepte nazivaju: *"nerazumna efikasnost podataka"* [58]. Iako kontradiktorno zvuči, upravo višak informacija, često je glavni adut za efikasnije izvršavanje pojedinih algoritama. Drugi autori pokazuju kako se uobičajeni algoritmi u oblasti umjetne inteligencije mogu znatno poboljšati, dajući im što više podataka za rad [59].

Sistemi za preporuku predstavljaju tehnike za modeliranje objekta korisnika, na osnovu kojeg se vrši procjena relevantnosti resursa. Ovi sistemi koriste podatke kao što su log-ovi pretrage, ocjene sličnih korisnika, društvene veze i sve značajne podatke koji se mogu iskoristiti u predviđanju nepoznatih relevantnosti. Gdje god postoji potreba za individualnim preporukama, prvenstveno u web okruženjima, sistemi preporuke obavljaju vrlo naporan posao iza scene.

Moderni pristupi modeliranja sistema preporuke često koriste „nerazumnu efikasnost podataka“, tako što kombinuju više sistema za preporuku, gdje svaki od njih predviđa relevantnost na različit način. Obzirom na različite afinitete korisnika i različitost resursa, ovi sistemi prilikom predviđanja koriste veoma složene metode, koje se oslanjaju na mnoštvo dokaza.

Ako sistem može precizno predvidjeti kako će korisnici reagovati na određene resurse, učinjen je ogroman korak prema rješavanju problema preopterećenosti informacijama.

2.3.1 Formalna definicija problema generisanja preporuke

Problem generisanja preporuka može se formalno zapisati [60]-[61]:

Neka je $U = (u_1, u_2, \dots, u_m)$ skup korisnika sistema, $I = (i_1, i_2, \dots, i_n)$ skup svih resursa koji se mogu preporučiti, a R potpuno uređeni skup.

Neka je $g: U \times I \rightarrow R$ funkcija korisnosti (*eng. utility function*).

Vrijednost funkcije $g(u_j, i_k)$ predstavlja korisnost resursa i_k za korisnika u_j .

Cilj sistema je da za svakog korisnika $u_j \in U$ odrediti resurs $i^{max, u_j} \in I$ za koji je vrijednost funkcije g maksimalna:

$$\forall u_j \in U, i^{max, u_j} = \arg \max_{i_k \in I} g(u_j, i_k) \quad (1)$$

Korisnost nekog resursa se obično predstavlja brožčanom ocjenom koju određuje direktno korisnik, ili se ona izračunava na osnovu prethodno definisanih pravila sistema. Drugim riječima, ocjena ne mora isključivo predstavljati vrijednost koju su korisnici dodijelili resursu, već se još može izraziti i brojem pregleda svidanja, komentara, kupljenih primjeraka i sl. [114].

Ocjena predstavlja mjeru, odnosno procjenu koliko je određeni resurs relevantan za korisnika. Za provođenje što preciznijeg ocjenjivanja potrebno je prikupiti što više značajnih informacija o učesnicima. Zato je neophodno skupom karakteristika opisati svakog korisnika $u_j \in U$ i svaki resurs $i_k \in I$. Prva faza u procesu generisanja preporuke je izgradnja modela korisnika, koji služi za čuvanje prikupljenih informacija o korisniku. Podaci se mogu prikupljati na dva načina: *eksplicitno* (npr. pomoću upitnika) ili *implicitno*, odnosno automatski (npr. na osnovu interakcije korisnika sa sistemom). U narednoj fazi podaci iz tog modela upoređuju se s karakteristikama resursa i/ili s podacima ostalih korisnika, kako bi se generisale odgovarajuće preporuke [61]-[114]. Osnovni problem sistema za preporuke je što u većini slučajeva funkcija g nije definisana na cijelom skupu $U \times I$, već samo na njegovom podskupu [62]. Inicijalno ocjena je poznata samo za one resurse koje su korisnici prethodno ocijenili, dok se za nepoznate vrijednosti

ona utvrđuje na osnovu predviđanja. Prema tome, jedan od ciljeva sistema preporuke je što preciznije predvidjeti ocjene za neocijenjene resurse na osnovu poznatih ocjena. Nakon predviđanja vrijednosti za sve uređene parove skupa $U \times I$, preporuka za određenog korisnika se određuje prema *formuli* (1). U zavisnosti od načina prezentacije preporučenih resursa, korisniku se može ponuditi jedan ili više rezultata (npr. lista od n preporuka sortiranih prema visini ocjene) [63]- [114].

Većina postojećih sistema preporuke obično koristi samo jedan kriterij za predstavljanje relevantnosti nekog resursa za korisnika u skupu $U \times I$, dok se u novijim istraživanjima preporučuje korištenje više kriterija kako bi se odredile preciznije preporuke. U tom slučaju preporuka se generiše na osnovu proširenog skupa podataka o korisniku i resursima koji se preporučuju. Funkcija g se tada može definisati na sljedeći način: $g: U \times I \rightarrow R_0 \times R_1 \times \dots \times R_l$, gdje je R_0 skup mogućih ukupnih ocjena (relevantnosti) resursa, a skupovi R_c , skupovi mogućih ocjena prema kriterijima $c = 1, \dots, l$. Za razliku od ukupne ocjene iz skupa R_0 , ocjene po kriterijima predstavljaju podatak o tome zašto se određeni resurs korisniku sviđa, a ne samo koliko mu se sviđa. Također, na osnovu ocjena po kriterijima moguće je preciznije odrediti sličnosti među korisnicima sistema [62]- [114].

U ovom radu će se pokušati napraviti korak dalje, te će se određeni kriteriji pridruživati posebnim modelima preporuke, koji će u obliku ponderisanih vrijednosti učestvovati u ukupnim rezultatima predviđanja.

2.3.2 Tradicionalni sistemi preporuke

Postoji mnogo načina za predviđanje relevantnosti resursa korisniku. Istraživači u ovoj oblasti otkrili su različite obrasce za preporuku, koji se u posljednje vrijeme i praktično primjenjuju, te se u ovom radu mogu posmatrati kao tradicionalni.

Obzirom na poznate tehnike, algoritme i njihove kombinacije koje se koriste pri procjeni relevantnosti resursa, tradicionalni sistemi preporuke mogu biti [114]:

- *sistemi preporuke zasnovane na sadržaju (eng. content-based, CB)*
- *kolaborativno filtriranje (eng. collaborative filtering, CF)*

- *agregativni sistemi preporuke*

U ovom poglavlju bit će predstavljeni osnovni koncepti tradicionalnih sistema preporuke sa posebnim osvrtom na njihova ograničenja. Detaljno istraživanje prednosti i nedostataka ovakvih pristupa može biti od velike koristi pri osmišljavanju adaptivnih pristupa preporuci.

2.3.2.1 *Sistemi preporuke zasnovane na sadržaju*

CB metode su bazirane na pretpostavci da će se korisniku svidjeti resursi sa sličnim karakteristikama onim resursima koji su mu se svidjeli u prošlosti. Sistem na osnovu prethodno ocijenjenih ili odabranih resursa pokušava "razumjeti" šta korisnik preferira. Vrijednost funkcije $g(u_j, i_k)$ kojom se predstavlja relevantnost resursa $i_k \in I$ za korisnika $u_j \in U$, u tom slučaju se procjenjuje na osnovu poznatih vrijednosti $g(u_j, i_s)$ za korisnika u_i , pri čemu je resurs $i_s \in I$ "sličan" resursu $i_k \in I$. U literaturi se za ovu vrstu sistema koristi i naziv *individualni sistemi preporuke* [64].

Prilikom generisanja preporuka, preferencije korisnika se utvrđuju na osnovu karakteristika resursa koje je odabrao, odnosno ocijenio. Ti se podaci upoređuju s karakteristikama svih elemenata skupa I te se izdvajaju oni resursi koji se po karakteristikama u najvećoj mjeri podudaraju s preferencijama korisnika. Na taj se način značajno smanjuje broj potencijalnih resursa za preporuku određenom korisniku [114].

Opis resursa $i_n \in I$, u oznaci **Content** (i_n) se obično predstavlja vektorom s realnim vrijednostima, odnosno težinama, pri čemu se svakom komponentom vektora izražava mjera "važnosti" (*eng. importance*) neke karakteristike u opisu [61]:

$$\mathbf{Content}(i_n) = i_n = (i_{n,1}, i_{n,2}, \dots, i_{n,K}) \in \mathbf{R}^k \quad (2)$$

Na primjer, opis može sadržavati *ključne riječi* (*eng. keywords*), odnosno *oznake* (*eng. tags*), kojima se opisuje određeni resurs. Analogno se predstavljaju *preferencije korisnika* (*eng. content-based preferences*) $u_m \in U$ vezane za resurs, u oznaci **ContentBasedUserProfile**(u_m), s težinskim vrijednostima za resurse koje opisuju sklonosti, interese i potrebe korisnika:

$$\text{ContentBasedUserProfile}(u_m) = u_m = (i_{m,1}, i_{m,2}, \dots, i_{m,k}) \in R^K \quad (3)$$

Vrijednost korisnosti resursa $i_n \in I$ za korisnika $u_m \in U$ određuje se funkcijom **score** čija se vrijednost određuje na osnovu kombinacije komponenti opisa resursa i_n i profila korisnika u_m :

$$g(u_m, i_n) = \text{score}(\text{ContentBasedUserProfile}(u_m), \text{Content}(i_n)) \in R \quad (4)$$

Tehnike za određivanje vrijednosti funkcije **score** navedenog izraza klasifikuju se u *heurističke* (eng. *heuristic-based*) i *tehnike zasnovane na modelu* (eng. *model-based*). Kod heurističkih tehnika se predviđanje vrijednosti određuje pomoću heurističkih formula, baziranih na metodama za *pretraživanje informacija* (eng. *information retrieval*). Primjeri takvih metoda su kosinusova sličnost, euklidska udaljenost, *Pearson-Spearmanova* korelacija. Tehnike je bazirane na modelu koriste znanje iz podataka, koristeći mašinsko učenje i statističke modele poput *Bayesovih* i neuronskih mreža, klaster algoritama i stabala odluke [114].

2.3.2.2 Kolaborativno filtriranje

CF metode imaju zadatak predviđanja korisnosti pojedinog resursa, za nekog korisnika, na osnovu vrednovanja resursa od strane ostalih korisnika sistema [114].

Vrijednost funkcije $g(u_j, i_k)$ kojom se predstavlja korisnost resursa $i_k \in I$ za korisnika $u_j \in U$ se u tom slučaju procjenjuje na osnovu poznatih vrijednosti $g(u_s, i_k)$ za resurs $i_k \in I$ pri čemu je korisnik $u_s \in U$ "sličan" korisniku $u_k \in U$. Kako se pri generisanju preporuka u obzir uzimaju podaci o ostalim korisnicima, za ovu vrstu sistema u literaturi se koristi i naziv *kolaborativni sistemi za preporuke*, bez obzira što se i u ovom slučaju preporuke generišu za individualne korisnike [64].

Kod CF metoda, ocjene koje su korisnici dodijelili resursima koriste se kao aproksimativna reprezentacija njihovih interesa i potreba. Za razliku od CB metoda, CF metode ne manipulišu podacima iz resursa [65], već se ocjene dodijeljene od strane ciljnog korisnika upoređuju s ocjenama koje su dodijelili preostali korisnici sistema, te se određuje skup njemu "najbližih komšija". U skladu s tim, uz model ciljnog korisnika

važna je baza podataka s modelima ostalih korisnika sistema. Ciljnom korisniku će među resursima koje još nije ocijenio biti preporučeni oni koje su visoko ocijenili njemu "slični" korisnici iz skupa "najbližih komšija". Pristupi CF metoda se razlikuju upravo po načinu na koji se određuje sličnost među korisnicima [114].

U nastavku je prikazana definicija modela (profila) korisnika, sa oznakom ***CollaborativeUserProfile(u_m)***, kao vektor čije su komponente ocjene koje je korisnik dodijelio dostupnim resursima [61]:

$$\mathbf{CollaborativeUserProfile}(u_m) = \mathbf{r}_m = (r_{m,1}, r_{m,2}, \dots, r_{m,k}) \in \mathbf{R}^N \quad (5)$$

Ocjene koje su određenom resursu dodijelili ostali korisnici predstavljaju se vektorom ***Ratings(i_n)***:

$$\mathbf{Ratings}(i_n) = \mathbf{r}_n = (r_{1,n}, r_{2,n}, \dots, r_{M,n}) \in \mathbf{R}^M \quad (6)$$

U slučaju da korisnik u_i nije ocijenio sadržaj i_j za komponentu $r_{i,j}$ u oba gore navedena vektora vrijedi $r_{i,j} = \emptyset$. Vrijednost relevantnosti resursa $i_n \in I$ za korisnika $u_m \in U$ određuje se funkcijom *score* koja u proračunu kombinuje komponente profila korisnika u_m i ocjene za i_n :

$$g(u_m, i_n) = \text{score}(\mathbf{CollaborativeUserProfile}(u_m), \mathbf{Ratings}(i_n)) \in \mathbf{R} \quad (7)$$

Tehnike za kolaborativno filtriranje dijele se na osnovu različitih formulacija za navedeni izraz u dva osnovna pristupa [61]:

- 1) *filtriranje po korisnicima (eng. user-based)*
- 2) *filtriranje po resursima (eng. item-based)*

Kod pristupa baziranom na filtriranju po korisnicima, vrednovanje resursa ciljnog korisnika se upoređuje s vrednovanjem ostalih korisnika kako bi se odredila grupa sličnih korisnika. Preporučuju se resursi koje su slični korisnici ocijenili najvišim ocjenama. Na taj način se postiže da se ciljnom korisniku preporuče resursi koje preferiraju njemu slični korisnici [114].

S druge strane, kod pristupa baziranog na filtriranju po resursima, za resurse koje je ciljnik korisnik pregledavao ili ocijenio pronalaze se i preporučuju slični sadržaji. Prilikom određivanja sličnosti među resursima upoređuje se vrednovanje ciljnog korisnika s vrednovanjem ostalih korisnika. Dakle, ciljnom korisniku se preporučuju sadržaji koji su ostali korisnici slično vrednovali [114].

2.3.2.3 Agregativni sistemi preporuke

Agregativni sistemi preporuke kombinuju CB i CF modele i nastoje prevazići ograničenja pojedinih pristupa uzimajući u obzir karakteristike i vrednovanje resursa od strane korisnika. Pristupi implementaciji mogu se klasifikovati zavisno od načina kombinovanja različitih tehnika za generisanje preporuka. Neke od mogućnosti su da se na osnovu određenog kriterija vrši izmjena korištenih tehnika u nekoliko faza, da se primjenom jedne tehnike dobiva model koji služi kao ulaz za sljedeću tehniku ili da se zajedno prezentiraju preporuke dobivene na osnovu različitih tehnika [61].

Autor [67] opisuje niz jednostavnih pristupa. *Min*, *Max* i *Sum* modeli kombinuju individualna predviđanja na neki način, ili vrše izbor jednog ili više rezultata kao konačno predviđanje. Drugi modeli koriste prosjek, ili log-prosjek različitih metoda. Modeli linearne kombinacije provjeravaju težine svakog prediktora i tretiraju predviđanja skladu s tim. Nešto složeniji pristup imaju modeli logističke regresije, koji preko provjere skupa podataka pokušavaju pronaći kombinaciju sa minimalnom greškom [66].

U ranijim pristupima eksperimentisano je sa kombinovanjem tradicionalnih CB i CF metoda. Kao što je ranije spomenuto, CB metode baziraju predviđanja na sličnosti resursa i prethodnih akcija određenog korisnika, dok CF pristupi predviđaju rezultate na osnovu prethodnih akcija sličnih korisnika u sistemu. Kombinacijom ova dva pristupa pokušavaju se minimizirati poznati problemi u svakoj od metoda [69]. U CF metodama se obično javljaju problemi rejtinga resursa za nove korisnike, radikalno različite korisnike ili kada se raspolaže vrlo oskudnim podacima. CB metode nemaju takve probleme, ali su manje efikasne nego CF metode na duge staze, jer se CB ne oslanjaju na znanje drugih korisnika u sistemu, odnosno na znanje izvan okvira jednostavne analiza

sadržaja. U obje metode, sistemi preporuke se koriste za stvaranje *jednostavno ponderisanog rezultata* [69].

Agregativni sistem preporuke moguće je realizovati kroz niz pristupa u kombinovanju CB i CF modela, odnosno njihovih rezultata [73].

Generalno, agregativne metode predviđanja u oblasti mašinskog učenja nazivaju se *asembliране metode* (eng. *ensemble methods, EM*) [74]. Iako se najčešće koriste za kombinovanje klasifikatora za klasificiranje resursa diskretnim oznakama (eng. *tags*), ove metode se također koriste za agregaciju numeričkih vrijednosti [72]. Ovi pristupi uključuju *bagging* i *boosting* metode i stvaraju kombinovane softverske agente za kombinovanje prediktora [75]. Autor [76] objašnjava srodne radove i teoriju ovakvih pristupa.

2.3.2.4 Ograničenja tradicionalnih sistema preporuke

Uprkos prividnoj snazi, tradicionalni sistemi preporuke su često ograničeni na jednostavne zadatke kao što su stvaranje kratke liste preporučenih sadržaja ili izračunavanje sličnih sadržaja zanimljivih korisniku. Uobičajeni primjeri su liste preporučenih sadržaja na osnovu pregleda, predlaganje novih društvenih veza ili predlaganje novinskih članaka na osnovu prethodnih čitanja. Vrlo rijetka su okruženja koja iskorištavaju puni potencijal sistema za preporuku i postižu stvaranje potpuno novih, adaptivnih sistema, koji naporno rade kako bi ublažili znakove preopterećenja informacijama.

Gledano iz perspektive adaptacije preporuke za individualnog korisnika, spomenuti tradicionalni sistemi preporuke imaju značajne nedostatke.

Osnovni nedostaci CB metoda su problem automatske analize sadržaja, odnosno nedostatka podataka kojima se opisuju resursi i problem „*hladnog starta*“ (eng. *cold-start problem*), gdje korisnik mora ocijeniti dovoljan broj resursa, kako bi mu sistem mogao predložiti odgovarajuće preporuke. Uspoređivanjem karakteristika pojedinog resursa s profilom korisnika i preporuke samo onih sadržaja za koje je ta vrijednost maksimalna ili vrlo visoka, događa se da se korisniku nikada ne preporučuju resursi

sadržajno različiti od resursa koje je korisnik pregledavao ranije (*eng. content overspecialization problem*), nego uglavnom međusobno vrlo slični resursi (*eng. non diversity problem*) [61]-[115].

Kod CB metoda, osim prevazilaženja problema „hladnog starta“, uspješnost zavisi od poređenja ocijenjenih resursa, te se, obično u okruženjima sa bogatim resursima koje sistem može preporučiti, javlja problem malog broja ocjena (*eng. sparse rating problem*). Resursi koji su vrednovani od strane malog broja korisnika tako neće gotovo nikada biti preporučeni, nezavisno od visine ocjena. CF metode su se pokazale uspješne za one korisnike koji se mogu svrstati u skupove "najbližih komšija" (klastere). Korisnici čije se preferencije razlikuju u poređenju s ostatkom korisničke mreže i za koje je teško naći slične korisnike (*eng. gray sheep problem*) ne mogu od ovakvih sistema očekivati dobre rezultate. Uz navedeno, treba spomenuti i *problem raznovrsnosti sadržaja* (*eng. non diversity problem*). Kako je kod CF metoda znanje sistema o resursima bazirano na tome šta korisnici preferiraju, preporuke su gotovo u potpunosti zavisne o tome koje sadržaje korisnici ocjenjuju, pa se u tom slučaju ne preporučuju najrelevantniji, nego samo najpopularniji resursi [61].

Iako se agregativnim pristupima pokušavaju otkloniti nedostaci CB i CF metoda, kod postojećih sistema mogu se identifikovati i neki opšti nedostaci. Kroz navedene probleme ovih metoda, prožima se i jedan od najvažnijih, a to je manjak subjektivnosti u predviđanju relevantnosti resursa, gdje postoji nesklad između načina izvođenja predviđanja i poželjnog predviđanja, baziranog na korisničkim potrebama ili željama. Većina sistema bazirana je na ograničenom skupu podataka o korisnicima i resursima koje preporučuje, ne uzimajući pritom informacije koje se mogu dobiti na osnovu praćenja korisnika, tehnikama kao što je dubinska analiza podataka (*eng. data mining*). Modeli korisnika su vrlo jednostavni i ne koriste naprednije tehnike modeliranja korisnika. Također, prisutni su problemi nefleksibilnosti i skalabilnosti. Prilikom korištenja algoritama poput nalaženja "najbližih komšija", sistem se ne prilagođava dobro povećanju broja korisnika i sadržaja. Za funkcionalnost sistema u slučaju velikog broja korisnika i sadržaja, implementiraju se tehnike *klasterovanja* (*npr. co-clustering*) ili tehnike za *smanjivanje dimenzije problema* (*npr. singular value decomposition*) [114].

Agregativnim metodama se također pristupa na generalizovanom, globalnom nivou, gdje se u različitim metodama modeliranja pretpostavlja ista važnost individualnih korisnika i resursa [57]-[66]-[68]-[69]-[70]. Ona može poslužiti za minimiziranje grešaka nad skupom podataka, dok problem subjektiviteta ostaje. Kao što je spomenuto, sastavni dio agregacije je generalizacija u kojoj se svi korisnici tretiraju na isti način, što predstavlja veliki problem u pronalasku osnovne ideje pri izgradnji modela korisnika.

Također, svjesno donošenje odluke o izboru odgovarajućeg obrasca preporuke, u fazi dizajniranja sistema preporuke, često se smatra kao ograničavajući faktor [60]. Prije bilo kakvih akcija modeliranja korisnika ili resursa, istraživač ili programer bira jednu ili više metoda, koja najviše odgovara u stvaranju univerzalnog modela. Dakle, modeliranje korisnika ili resursa je bazirano na pretpostavci, što u osnovi subjektivnosti nije poželjno. Ova pojava je klasičan *problem latentne subjektivnosti* [71].

Postoji mnogo primjera u kojima su prisutni ovakvi problemi. Recimo, nekim korisnicima je bitan društveni uticaj pri preporuci, dok kod ostalih to nije slučaj. Na primjer ako se razmatra primjer preporuke filmova, neki korisnici će biti zadovoljni preporukom baziranoj na sličnosti prema naslovu, žanru ili akterima, dok će drugi biti zadovoljniji ako im se preporuči film iz istog vremenskog perioda. Nekim korisnicima mogu odgovarati rezultati koji su globalno visoko ocijenjeni, dok drugima više odgovara iskazano mišljenje sličnih korisnika. Slični problemi se javljaju i kod rezultata, odnosno objekata preporuke. Dok su pojedini resursi možda najrelevantniji po svom sadržaju, drugi imaju svoju vrijednost na osnovu prethodne ocjene od drugih korisnika. Relevantnost nekog resursa može biti usko vezana za period kada je nastala, dok kod ostalih resursa vremenska dimenzija uopšte nije bitna. Precizno definisanje ovih razlika nije toliko bitno, koliko je bitno saznanje da one postoje.

U ovom potpoglavlju predstavljani su tradicionalni sistemi preporuke, kao i njihova ograničenja. Nešto noviji, agregativni modeli, koji na različite načine kombinuju poznate standardne metode (algoritme), već su odavno predmeti istraživanja u ovoj oblasti i ne predstavljaju ništa novo. S druge strane *adaptivni sistemi preporuke (ASP)* imaju zadatak da uklone subjektivno-bazirani izbor u kombinovanju metoda, odnosno da taj izbor, kroz implicitno prikupljanje podataka, „prepuste“ individualnom korisniku.

2.3.3 Savremeni pristupi u razvoju sistema preporuke

U nastavku rada bit će istraženi obrasci kojim se pokušavaju riješiti ili umanjiti navedeni problemi tradicionalnih sistema preporuke. Posebna pažnja će biti usmjerena na savremene pristupe agregativnog predviđanja preporuke, odnosno na kombinaciji više metoda predviđanja, koje koriste različite algoritme.

Prioritet algoritama treba da bude implicitno i automatski baziran na uspješnosti prijašnjeg učinka za individualne korisnike i resurse. Skup korisnika i resursa je često prevelik, te za jednu ili generalizovanu kombinaciju metoda ne bi bilo moguće akumulirati sve nijanse značajnih predviđanja [71].

Za rješavanje ovog problema, neki autori predlažu novu metodu baziranu na prilagodljivoj preporuci (*eng. adaptive recommenders, AR*) [71]. U odnosu na tradicionalne sisteme preporuke, AR predstavlja novi nivo apstrakcije i personalizacije, a odluke se donose implicitno, bez dodatne eksplicitne interakcije. Drugim riječima, u pitanju je novi sloj u sistemima za preporuku, koji je zadužen za preporuku algoritma koji je najučinkovitiji u konkretnom slučaju. To se, prije svega, odnosi na izbor poznatih algoritama. Istraživanje će biti fokusirano na rješavanje problema latentne subjektivnosti, odnosno bit će razmotreno kako bi dodatni sloj apstrakcije mogao pomoći u rješavanju ovog problema.

2.3.3.1 Adaptivni sistem preporuke(ASP)

Adaptivni sistemi preporuke (*eng. adaptive recommenders, AR*) predstavljaju tehnike za kombinovanje različitih sistema preporuke, u cilju optimizacije sistema preporuke za svakog korisnika i svaki resurs. U predviđanju relevantnosti određenog resursa korisniku, koriste se različite metode koje razmatraju disjunktne⁹ obrasce podataka. Kako bi se uspješno provela analiza adaptivnih sistema preporuke (ASP) neophodno je odgovoriti na dva važna pitanja:

⁹ Disjunktne skupovi - Presjek skupova A i B, označen sa $A \cap B$, je skup svih elemenata koji su članovi i skupa A i skupa B. Ako je $A \cap B = \emptyset$, tada za A i B kažemo da su disjunktne.

- 1) Kakav rejting svaka od metoda predviđa?
- 2) Koliko je precizno svako od tih predviđanja?

Prve generacije sistema preporuke pokušavaju odgovoriti uglavnom na prvo pitanje koristeći samo jednu metodu za predviđanje nepoznatog rejtinga. Moderne agregativne tehnike pokušavaju odgovoriti na oba pitanja kombinacijom različitih sistema preporuke, koristeći optimalno-generalizovanu šemu ponderisanja. Na primjer, svaki sistem preporuke korišten u kombinaciji može biti ponderisan, na način da rezultira najnižom prosječnom greškom u odnosima između svih posmatranih korisnika i resursa. Prema tome, adaptacija agregativnih sistema preporuke zavisi isključivo od konkretnih objekata, korisnika ili resursa, koji se trenutno razmatraju.

ASP bi se mogli formalno definisati kao adaptacija skupa sistema preporuke pomoću drugog komplementarnog skupa sistema preporuke (*Slika 6*). Ovo se može posmatrati kao određeni oblik meta-modeliranja, gdje je jedan skup metoda modeliranja prilagođen pomoću drugog skupa metoda modeliranja. Prvi skup stvara standardne rezultate predviđanja i odgovara na prvo pitanje, dok drugi skup predviđa preciznost svake od metoda za trenutnog korisnika i resurs, i odgovara na drugo pitanje. Zanimljivo je da ASP može koristiti skup sistema preporuke za oba navedena zadatka, kao što će biti prikazano u nastavku.

ASP je određen takozvanom 6-torkom $\{ASP = (I; U; R; F; M; A;)\}$, gdje su :

- Resursi ($I - Items$),
- Korisnici ($U - Users$),
- Rejting ($R - Ratings$),
- Okruženje ($F - Framework$),
- Metode ($M - Methods$),
- Adapteri ($A - Adapters$).

Svaki korisnik $u \in U$ može ocijeniti, odnosno proizvesti rejting $r \in R$ za resurs $i \in I$. Tada $r_{u,i}$ predstavlja rejting (skup ocjena) resursa i koju je proizveo korisnik u . Kao što je već spomenuto, resurs može predstavljati različite objekte, a u kontekstu kolaboracije,

resursi mogu biti i sami korisnici, koji implicitno ili eksplicitno proizvode ocjene [77]-[78]-[79]-[80].

Korištenje termina „rejting“ treba uzeti sa rezervom, jer u kontekstu ASP -a predstavlja mnogo šire značenje. Ekvivalentni termini su: relevantnost, korisnost, skor, ocjena, jačina konekcije i sl. Drugim riječima, rejting predstavlja mjeru korisničkog mišljenja ili odnosa prema određenom resursu i na razumljivom jeziku u posmatranom okruženju. U okviru ovog rada termin „rejting“ će predstavljati akumulaciju svih relevantnih informacija koje će biti korištene u različitim prediktabilnim modelima.

Okruženje (F) varijabla određuje način predstavljanja podataka u sistemu. Grafikoni i matrice predstavljaju dva kanonska¹⁰ načina predstavljanja korisnika, resursa i rejtinga [81]. U nastavku je prikazana matrica ukupnog rejtinga, u kojoj prva dimenzija predstavlja korisnike, druga resurse, a svaka prikazana ćelija eksplicitni ili implicitni rejting:

$$R_{u,i} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,i} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ r_{u,1} & r_{u,2} & \cdots & r_{u,i} \end{bmatrix}$$

Za iskorištavanje disjunktnih obrazaca podataka, koristi se skup metoda (M) modeliranja, sa svojim vlastitim načinima procjene nepoznatih rejtinga. Svaki model $m \in M$ se koristi za izračunavanje nezavisnih predviđanja, a svaka metoda predstavlja poseban sistem preporuke. Za pronalaženje, predviđanje i prezentaciju relevantnih informacija koriste se različiti algoritmi preporuke koji, podatke posmatraju iz različitih perspektiva i sa različitih aspekata. Osim konkretno pohranjenih podataka o korisnicima, resursima i ocjenama, vrlo često se analiziraju „skriveni“ podaci, kao što su vrste i jačina veza između poznatih podataka i sl. [62]-[82]-[83]-[84].

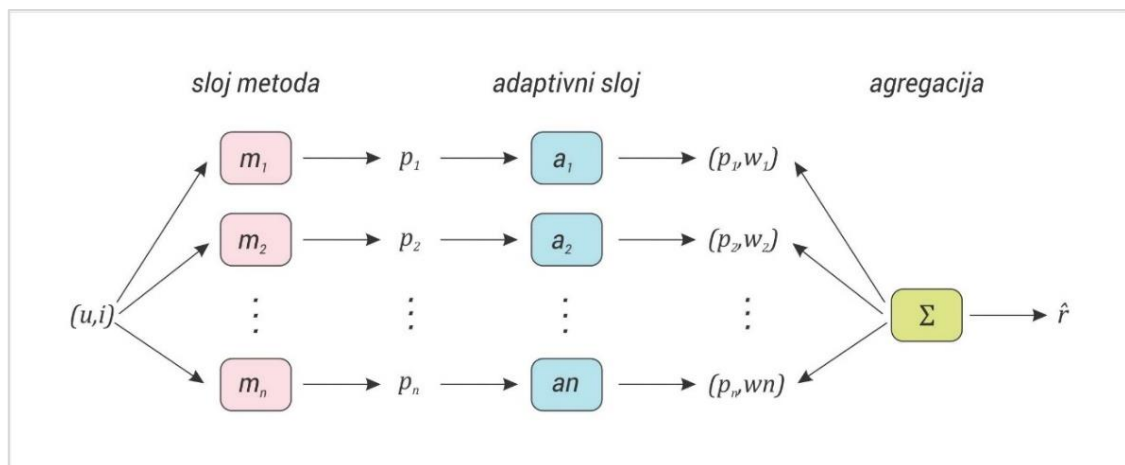
Primjeri takvih sistema preporuke su uključuju „*Slope one*“ familiju algoritama [85], *SVD faktORIZACIJU* (eng. *singular value decomposition factorization*) [57]-[86]-[87] i

¹⁰ Kanon (grč. preko lat.) - pravilo, obrazac, norma; normativni kriterij za bilo koje područje znanja ili djelovanja.

ponderisano predviđanje najbližih komšija [88]. Ove metode predviđaju nepoznate veze između korisnika i resursa na osnovu određenog obrasca nad podacima, na primjer korisnički profil sličnosti, rejting korelacije ili društvenih veza. Kao što je već napomenuto, u postizanju najboljeg mogućeg zajedničkog rezultata, potrebno je koristiti metode koje koriste disjunktne obrasce, odnosno komplementarno-prediktivne dijelove podataka.

Adapteri (A), u pomenutoj 6-torki, se odnose na drugi nivo metoda modeliranja korisnika. U tradicionalnim agregativnim metodama, adapteri predstavljaju jednostavnu linearnu funkciju za kombinovanje različitih predviđanja (npr. proračun sume težinskih vrijednosti, po jedan za svaku metodu).

Preciznost kombinovanih prediktora zavisi više od sposobnosti različitih prediktora u otkrivanju različitih aspekata podataka, nego od pojedinačne preciznosti svakog prediktora [57]. Višestruki rezultati predviđanja se obično kombinuju u konačni, jedinstveni rezultat, baziran na generalizovanoj kombinaciji koja je izvedena minimiziranjem određenih grešaka nad svim korisnicima.



Slika 6. Slojevi sistema preporuke

Adapteri u ASP ustvari, predstavljaju metode modelovanja korisnika, ali umjesto toga, potrebno je modelovati ponašanje, odnosno preciznost sistema preporuke. Metode na drugom sloju koriste se za predviđanje preciznosti svakog od odgovarajućih osnovnih

predviđanja. Dakle, ove metode omogućavaju adaptivnu agregaciju za određenog korisnika i resurs, odnosno postoje dva različita sloja modeliranja korisnika (*Slika 6*).

2.3.3.2 Adaptivna agregacija

Za izvođenje adaptivne agregacije, potrebno je da adapteri predstavljaju stvarne sisteme preporuke. Najjednostavniji generalizovani način predviđanja agregacije je izračunavanje prosjeka svih predviđanja od strane različitih metoda [67]. Većina agregativnih tehnika pokušava izvršiti procjenu svake metode na drugačiji način [69]:

$$\hat{r}_{u,i} = \sum_{m \in M} w_m \times p(m, u, i), \quad \text{gdje je } 0 \leq w_m \leq 1, \sum_{m \in M} (w_m) = 1. \quad (8)$$

U *formuli (8)* w_m je težina w primjenjena na metodu modeliranja m . Težina w može imati vrijednost u rasponu od 0 do 1. Težine se mogu procijeniti kroz različite metode mašinskog učenja. Međutim, to je još uvijek generalizirani rezultat, odnosno prosjek kroz svako predviđanje. Sistem pretpostavlja da najbolji prosjek predstavlja najbolji rezultat za svakog pojedinačnog korisnika i resurs. Čak i uz pomoć ovakvih metoda za specifičnu težinu, problem latentne subjektivnosti nije riješen.

U cilju iskorištavanja što većeg broja obrazaca, ostavljajući po strani latentnu subjektivnost, potrebno je raspolagati adaptivnim težinama koje se izračunavaju specijalno za svaku kombinaciju korisnika i resursa. Ako bi svaka težina bila specifična kombinacija, proračunavanje težine za svaku metodu bi praktično postalo teško izvodljivo, odnosno bilo bi potrebno izračunati sve moguće ocjene za svaku korištenu metodu. Drugim riječima, ove adaptivne težine također moraju biti procjenjene, isto kao i same ocjene:

$$\hat{r}_{u,i} = \sum_{m \in M} p_w(m, u, i) \times p_r(m, u, i), \quad \text{gdje je } \sum_{m \in M} (p_w(m, u, i)) = 1. \quad (9)$$

U ovom primjeru, $p_w(m, u, i)$ je predviđena optimalna težina za metodu m kada se primjenjuje na korisnika u i resurs i . Adaptivna preporuka je jedan od načina za procjenu ovih težina, odnosno jedan od načina da se provede p_w .

Osnovna ideja je da se koriste standardni sistemi preporuke za predviđanje optimalne adaptivne težine. Za sprovođenje ovakve ideje, potrebno je kreirati matricu (grafikon), u kojoj će se čuvati poznate vrijednosti o preciznosti svakog predviđanja rejtinga.

To se može učiniti modelovanjem grešaka za svaku od metoda. Modelovanjem grešaka u standardnim sistemima preporuke, mogu se predvidjeti greške za neprovjerene kombinacije. Predviđanjem grešaka sistema preporuke za korisnika i resurse, također se predviđa i preciznost, što je moguće izvesti kreiranjem *greška-matrice* :

$$E_{u,i} = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,i} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ e_{u,1} & e_{u,2} & \cdots & e_{u,i} \end{bmatrix}$$

Kreiranje greška-matrice za metode modeliranja se vrši razdvajanjem skupa podataka o rejtingu u dva podskupa. Prvi podskup, tzv. trening skup, koriste standardne metode modelovanja, dok se drugi, podskup za procjenu grešaka, koristi za popunjavanje greška-matrice. Svaka standardna metoda modeliranja može proizvesti greška-matricu u kojoj neke ćelije imaju vrijednosti. Vrijednost ćelija u toj matrici odgovara predviđanju greške za kombinaciju korisnika i resursa. Svaka metoda modeliranja koristi dio podataka o rejtingu. Greška-matrica je popunjena od ostatka podataka, računanjem greške svih poznatih ocjena iz standardnih metoda:

$$\forall (u, i, r) \in (d_e - d_m): E(m)_{u,i} = |r - p(m, u, i)| \quad \text{gdje je } d_e, d_m \subset D. \quad (10)$$

U formuli (10) d_m i d_e su podskupovi posmatranog skupa podataka D . m je metoda modeliranja posmatrani podskup d_m . Za popunjavanje greška-matrice za određenu metodu, uzima se svaki rejting koji se ne koristi za stvarne rezultate i izračunava se greška metode u ovoj kombinaciji. Oskudan rezultat greška-matrice može se koristiti za predviđanje nepoznate greške. Također, može se primijetiti velika sličnost između ove dvije matrice, što će na kraju omogućiti standardnim sistemima preporuke obavljanje adaptivne agregacije.

2.3.3.2.1 Faza modelovanja

Za implementaciju novih metoda modelovanja, primjenjuju se sljedeći koraci:

- 1) Podijeliti skup podataka o rejtingu u dva podskupa, trening skup i skup za procjenu grešaka
- 2) Iz trening skupa omogućiti dostupnost specifičnih podataka za svaku metodu modeliranja
- 3) Kreirati greška-matricu koristeći podatke iz skupa za procjenu grešaka
- 4) Podatke iz greška matrice učiniti dostupnim za model greške

Kreiranje podskupova dostupnih podataka je osnovni zadatak u *asebliranom učenju* [75]. Za ovu ideju se koriste „*Bootstrap*“ agregacije, poznate kao *pakovanja* (eng. *bagging*) [72]-[76]. Izvorno, pakovanja su korištena za klasifikacijske metode asebliranog učenja, gdje su višestruki klasifikatori koristili podatke iz uniformno uzorkovanog podskupa dostupnog trening skupa. Svaki model je koristio podatke iz odgovarajućih podskupova, a agregacija modela je bazirana na prosjeku njihovih pojedinačnih predviđanja.

Pakovanja odgovaraju i potrebama adaptivne agregacije u sljedećem:

- pomažu pri stvaranju disjunktnih prediktora, jer svaki prediktor koristi, ili je specijaliziran za, samo podskup dostupnih podataka
- moguće je testiranje osnovne metode modeliranja bez kompleksnog raščlanjivanja podataka.

Pojava nepovezanih prediktora je objašnjena sljedećom izjavom:

"Čvrsto vjerujemo da uspjeh pristupa asebliranog učenja zavisi od sposobnosti različitih prediktora da razotkrije različite, dopunjavajuće aspekte podataka. Iskustvo pokazuje da se takav pristup dosta razlikuje od pristupa koji optimizuje tačnost svakog pojedinog prediktora. Analizirani rezultati većinom ukazuju na to, da je kombinovanje preciznih prediktora manje korisno."

Drugim riječima, prediktori bi u ovom slučaju trebali biti usmjereni na različite aspekte podataka, što se može postići kroz pakovanja slučajnog podskupa dostupnih podataka. Performanse kombinovanih sistema preporuke često ne zavisi od njihovih pojedinačnih preciznosti, nego od njihove vještine predviđanja ocjena za različite dijelove cijelog skupa podataka [89].

Modeli greške su prilagođeni za korištenje standardnih sistema preporuke. Očekivani ulazi i izlazi su isti. Postoje dvije dimenzije, s oskudnim skupom poznatih veza, na osnovu kojih treba predvidjeti nepoznate veze između podataka. Rezultat je skup metoda modeliranja koje mogu predvidjeti greške sistema preporuke za kombinaciju određenog korisnika i resurs.

Primjenom greška-matrice u sistemima preporuke, kao izlaz se generišu greške čija se vrijednost nalazi na istoj skali kao što je i rejting. Rejting koji se posmatra kao buka u matrici rejtinga, odnosno rejting čija vrijednost ne ulazi u skor nekog od osnovnih obrazaca, je pogodan za popunjavanje greška-matrice.

Na primjer, neka metoda može imati veliku vrijednost greške za kombinaciju određenog korisnika i resursa, što može biti korisno za oba elementa, jer predstavlja prepoznatljivu vrstu buke koju sistemi preporuke s lakoćom procesuju, odnosno najčešće odbacuju.

Posebno su interesantne situacije u kojima određena metoda ima stabilne i značajne greške za veliki broj ocjena od korisnika, odnosno veliki broj rejtinga za resurse. U ovom slučaju primjena obrasca greška-matrice nije od velike koristi, jer u sistemima preporuke, ovakvi obrasci su najefikasniji u ulozi identifikacije.

Drugim riječima, iste mogućnosti koje sisteme preporuke na bazi rejting-matrice čine efikasnim, će također omogućiti efikasnost i pri upotrebi greška-matrice. Osobine koje su potrebne za predviđanje rejtinga, također se koriste i za predviđanje preciznosti. Međutim, i slabosti sadržane u rejting-matrici se prenose na greška-matricu. To se prije svega odnosi na: pristrasnost korisnika i resursa, koncepte otklona i varirajuće količine buke. Osim navedenih prisutni su i klasični problemi kao što su hladni start i problem preporuke novim korisnicima, čije preference još uvijek nisu zabilježene u sistemu.

Adaptivni sistemi preporuke bi se trebali suprotstaviti navedenim slabostima, jer u protivnom bi bilo teško pronaći svrhu njihovog korištenja. Korištenjem greška-matrice, će se djelimično pokušati odgovoriti na ovakve izazove, koji su već odavno tema razmatranja u akademskoj zajednici.

Tražeći globalne obrasce među podacima, različiti sistemi preporuke će različito raditi za adaptivni sloj. Fokus interesovanja je na identifikovanju grupe korisnika ili resursa, ili njihovoj kombinaciji, nad kojima će sistemi preporuke raditi vrlo efikasno ili ne. Sistemi preporuke na bazi SVD algoritama su jedna vrsta sistema koji se mogu koristiti za tu svrhu. Smanjivanjem dimenzije za model-greške u posebno kategorizovanu dimenziju greške, može se testirati efikasnost svih dostupnih metoda nad različitim grupama skupova. O ovoj problematici će više biti riječi u diskusiji i zaključcima ovog dokumenta.

2.3.3.2.2 Faza predviđanja

Kada je izgrađen model greške za svaku metodu modeliranja, ove greške se mogu koristiti za procjenu kontekstualne težine. Za adaptivno-agregativno predviđanje primjenjuje se sljedeći algoritam:

$$\hat{r}_{u,i} = \sum_{(m_e, m_r) \in M} \left(1 - \frac{p(m_e, u, i)}{greska(u, i)} \right) \times p(m_r, u, i), \text{ gdje je } error(u, i) = \sum_{m_e \in M} p(m_e, u, i) \quad (11)$$

U nastavku su ukratko opisani koraci korišteni u *formuli (11)*:

- 1) prikupiti predviđanja iz svaku od metoda modeliranja za (u, i)
- 2) prikupiti procijenjene greške za svaku metodu za (u, i)
- 3) Izračunati težine za svaku metodu bazirane na njihovom relativnom predviđanju grešaka
- 4) Sumiranjem ponderisanih predviđanja izračunati adaptivno predviđanje rejtinga

Svako predviđanje rejtinga, odnosno relevantnosti, je ponderisano sopstvenim predviđanjem preciznosti koje je uslovljeno trenutno posmatranim korisnikom i resursom.

U *formuli (11)* prikazano je da svaka metoda sistema preporuke ima dva odgovarajuća modela:

- \mathbf{m}_r je *rejting-model*, koristi za predviđanje ocjena,
- \mathbf{m}_e je *greška-model*, koristiti za predviđanje greške.

$p(\mathbf{m}_e, \mathbf{u}, \mathbf{i})$ je predviđanje za model \mathbf{m} (sistem preporuke) za relevantnost između korisnika \mathbf{u} i resursa \mathbf{i} . Svaka metoda iz skupa \mathbf{M} je ponderisana svojim predviđanjem preciznosti. Ponderi su izračunati uzimajući suprotnu vrijednost od predviđenih grešaka metoda. Greške su normalizovane za sve korisnike i resurse pomoću vrijednosti $\mathbf{error}(\mathbf{u}, \mathbf{i})$, koja predstavlja sumu grešaka svih metoda za trenutnu kombinaciju korisnika \mathbf{u} i resursa \mathbf{i} . Rezultat normalizacije su težine u rasponu od 0 do 1, čime se osigurava da konačna prognoza rejtinga bude na istoj skali kao što je povratna vrijednost standardnih sistema preporuke.

U ovom dijelu se treba istaći da jedina razlika između \mathbf{m}_r i \mathbf{m}_e je njihov način kreiranja. Dok \mathbf{m}_r koristi podatke iz *rejting-matrice*, \mathbf{m}_e koristi podatke iz *greška-matrice*. Iz navedenog slijedi logičan zaključak, da se bilo koji standardni sistem preporuke može koristiti u izvođenju adaptivne agregacije. Takvi sistemi se nazivaju *adaptivni sistemi preporuke*. Skup sekundarnih sistema preporuke na adaptivnom sloju se koristi za prilagođavanje (adaptaciju) skupa standardnih (primarnih) sistema preporuke za određene korisnike i resurse.

Važno je napomenuti da su vrste sistema preporuke koje se koriste na adaptivnom sloju apsolutno nezavisne od standardnih sistema preporuke. Adaptivnom sistemu preporuke su potrebni samo predviđeni rejtinzi od standardnih sistema preporuke, i ne zanima ga koji je algoritam upotrijebljen. Prilikom predviđanja, proračuni u metoda-sloju i adaptivnom sloju su potpuno nezavisni, kao jer oba koriste prethodno izračunate modela. Metoda-sloj koristi *rejting matricu*, ili sopstvene modele stvorene tokom izvođenja, dok adaptivni sloj koristi *greška-matrice* za svaku standardnu metodu.

Rezultat toga je sistem koji ne samo da prikuplja niz predviđanja za svaku nepoznatu kombinaciju korisnika i resursa, već kombinuje spomenute metode na osnovu vjerovatnoće preciznosti svakog predviđanja.

2.3.4 Prikupljanje podataka za model korisnika

U kontekstu sistema za generisanje preporuka podaci za model korisnika mogu se prikupljati eksplicitno, koristeći povratne informacije od korisnika ili implicitno odnosno automatski, prikupljanjem podataka o interakciji korisnika sa sistemom [30]. Navedeni pristupi su nezavisni o tehnikama, odnosno algoritmima koje se koriste za generisanje preporuka [114].

Kod eksplicitnog pristupa, pretpostavlja se da je korisnik voljan pružiti sistemu potrebne informacije i da je spreman za to odvojiti određeni dio svoga vremena, što predstavlja nedostatak ovoj pristupa [90].

Kod implicitnog ili automatskog pristupa, model se izgrađuje i nadograđuje dinamički, bez povratnih informacija od strane korisnika. Na taj način se korisnici ne opterećuju raznim upitnicima i anketama, već se prikupljaju podaci iz njihovih akcija u sistemu [114].

Potrebni podaci mogu se automatski prikupljati na strani servera ili klijenta. Izvor podataka u tom slučaju mogu biti: *keš memorija pretraživača* (eng. *browser cache*), *proksi server* (eng. *proxy servers*), *agenti za pretraživanje* (eng. *browser agents*) ili *interfejs* (eng. *desktop agents*), *web logovi* (eng. *web logs*) i *logovi pretraživanja* (eng. *search logs*) [90].

Na strani klijenta dostupni su samo podaci o individualnom korisniku što znači da je moguće ostvariti samo individualnu personalizaciju, dok su na strani servera dostupni podaci o svim korisnicima sistema te je moguća i personalizacija grupama korisnika. Ipak, podaci na strani poslužitelja odnose se samo na interakciju korisnika s resursima dostupnim u okviru sistema. Kod pristupa prikupljanja podataka na strani klijenta dostupni su i podaci o interakciji s ostalim resursima na web-u, pri čemu se najčešće

automatski prikupljaju podaci o *toku klikova* (eng. *click stream*) koje je izvršio korisnik za vrijeme pregledavanja resursa te *istorije navigacije* (eng. *browsing history*) [91].

Obzirom na količinu tako prikupljenih podataka, za otkrivanje znanja u podacima u svrhu personalizacije koriste se postupci *dubinske analize podataka na web-u* (eng. *Web data mining*). Procesu generisanja preporuka, koji se bazira na otkrivanju znanja u podacima koristeći tehnike dubinske analize podataka, prethode faze *pripreme i transformacije podataka* (eng. *data preparation and transformation*), te *otkrivanje uzoraka* (eng. *Pattern discovery*). U fazi pripreme podaci sakupljaju se podaci iz različitih izvora i transformišu se u oblik pogodan za dalju obradu. U fazi otkrivanja uzoraka koriste se različite tehnike dubinske analize podataka poput *klasterovanja* (eng. *clustering*), *otkrivanja pravila povezivanja atributa* (eng. *association rule mining*), *analize sekvencijalnih uzoraka* (eng. *sequential pattern mining*) i *probabilističkog modeliranja* (eng. *Probabilistic modeling*). Dobiveni rezultat se tada uključuje u model korisnika u obliku pogodnom za fazu generisanja preporuka [65].

Uspoređujući eksplicitno i implicitno prikupljanje povratnih informacija, autori na osnovu evaluacije primjera primjene ovih pristupa zaključuju kako nema jasnog odgovora na pitanje pruža li implicitno prikupljanje podataka osnovu za izgradnju pouzdanijih modela korisnika [90]. Oni navode kako su ranije studije u kojima se pokušavaju uporediti navedeni pristupi, davale prednost eksplicitnom prikupljanju podataka, dok one vršene u posljednje vrijeme predlažu da je pristup implicitnog prikupljanja povratnih informacija bolji. Razlog za to se može potražiti u napretku istraživanja o tome kako automatski prikupiti podatke te kako ih iskoristiti u svrhu izgradnje modela korisnika.

Nedostatak pristupa implicitnog prikupljanja povratnih informacija jeste što ne pruža mogućnost negativnih povratnih informacija. Naime, prilikom automatskog prikupljanja povratnih informacija akcije poput klika na hipervezu koja usmjerava do određenog resursa, pregledavanje neke web stranice i sl., tumače se kao pozitivna povratna informacija - resurs se korisniku "svida". Zato je vrlo upitno može li se zaključiti da za neki sadržaj korisnik nije zainteresovan samo zato što ga nije odabrao, odnosno pregledao.

Ipak većina sistema traži od korisnika eksplicitne povratne informacije, što se uglavnom tretira kao uznemiravanje korisnika, pa se ovakve metode smatraju *nametljivim* (eng. *intrusive*). Ipak u nekim istraživanjima se spominju metode za smanjivanje nametljivosti uz zadržavanje određenog nivoa preciznosti prilikom generisanja preporuka, jer se pokazalo kako ocjene dobivene bez nametljivih tehnika ne mogu u potpunosti zamijeniti eksplicitne povratne informacije dobivene od korisnika [61]-[114].

U *FITCKM* sistemu, će se koristiti kombinacija eksplicitnog i implicitnog prikupljanja podataka. Sistem bi trebao na osnovu eksplicitnih ocjena kojima korisnik ocjenjuje resurse, izvršiti adaptaciju njihovih vrijednosti i kao takve ih pohraniti u bazu podataka. Ovaj koncept će se pokušati realizovati uz pomoć simulacije inteligentnog programskog agenta koji je veoma bitan za provjeru glavne hipoteze rada (potpoglavlje 4.1).

3 IMPLEMENTACIJA FITCKM SISTEMA

Praktični dio rada se sastoji iz dva dijela. U prvom dijelu ovog poglavlja su objašnjeni najvažniji detalji razvoja *FITCKM* sistema, odnosno virtualnog kolaborativnog web okruženja namijenjenog za razmjenu informacija među korisnicima. Podaci o korisničkim aktivnostima u sistemu su pohranjeni u bazu podataka, te će se koristiti u eksperimentalnom okruženju za testiranje različitih sistema preporuke. Eksperimentalni dio je predstavljen u drugom dijelu ovog poglavlja. Osim spomenutih podataka, u eksperimentalnom dijelu koristiće se i eksterna kolekcija podataka, kako bi se što vjerodostojnije testirala mogućnost primjene sistema preporuke.

3.1 FITCKMS

U ovom poglavlju će ukratko biti predstavljen razvoj kolaborativnog sistema za upravljanje znanjem (*FITCKMS*), odnosno web okruženja koje će poslužiti za testiranje i istraživanje ideja, teorija i hipoteza navedenih u radu. Kako bi se ostalo u navedenim okvirima, akcent će biti na modelu korisnika i resursa, odnosno na značajnim podacima koje sistem bilježi o njima. Inteligentni programski agent će biti zadužen za detekciju različitih događaja u sistemu u smislu korisničkog kretanja kroz sistem i na osnovu njih će procesuirati adekvatne odgovore u kontekstu dinamičke izgradnje modela korisnika i poboljšanja kvaliteta sistema preporuke. Podaci će biti pohranjeni u bazu podataka i sistem će, osim fizički pohranjenih, pronalaziti i metapodatke, te ih koristiti kao ulaz za algoritme preporuke. Krajnja namjena aplikacije je prije svega podrška korisnicima u procesu elektronskog učenja, u smislu kolaboracije i adaptivne navigacije usmjerene ka individualnim interesima. Za potrebe rada, odnosno eksperimentalnog okruženja, koristit će se informacije iz spomenute baze podataka.

Za izradu *FITCKM* sistema korištene su sljedeće aplikacije:

- *Microsoft Visual Studio 2013 – Student Version 12.0.21005.1.REL*
- *MicrosoftSQLServer 2014 Management Studio*

3.1.1 FITCKMS - Opis sistema

Fit_CKMS je sastavljen iz dva modula: *Wiki stranica* i *Pitanja i Odgovori*. Iako pojam kolaboracije ima mnogo šire značenje, kao što je navedeno u prethodnim poglavljima, navedeni moduli će biti sasvim dovoljni za testiranje ideja i hipoteza ovog rada. U procesu razvoja sistema je korištena višeslojna arhitektura, te će u budućnosti biti moguće unaprijediti postojeće ili dodati nove module i funkcionalnosti.

3.1.1.1 Moduli sistema

Wiki stranica je modul koji omogućava čitanje, kreiranje, uređivanje, brisanje korisničkih članaka i sl. Osim manipulacije tekстом, ovaj modul će omogućiti i postavljanje različitih multimedijalnih sadržaja. Svi prijavljeni korisnici će biti u mogućnosti da kreiraju novi članak u okviru određene teme i oblasti. Njima je također omogućeno i uređivanje svakog članka, nakon čega će promjena biti predložena autoru članka. Autor članka odlučuje da li je promjena primjerena ili ne, te na osnovu toga ima mogućnost da istu odobri. Ako je autor članka odobrio promjenu, promijenjeni članak postaje finalni, a sve prethodne ili buduće verzije ostaju pohranjene u bazi podataka. Također, će biti omogućeni razgovori o članku, kao i komentari na članak. Svi korisnici će moći eksplicitno ocijeniti članak putem rejtinga ili izraziti se o tome da li im se članak sviđa ili ne (*eng. Like, UnLike*).

Prilikom čitanja određenog članka sistem bi trebao preporučiti listu nepoznatih, korisnih ili zanimljivih članaka logiranom korisniku. Osim preporuke članaka, sistem predlaže i listu potencijalnih korisnika u obliku hiperveze, koji su prema određenom kriterijumu slični logiranom korisniku. Izborom korisnika sa preporučene liste predlaže se lista zanimljivih članaka izabranog korisnika. Paralelno sa spomenutim preporukama sistem preporučuje i listu zanimljivih pitanja i odgovora, odnosno predlože hiperveze za modul *Pitanja i Odgovori*.

Modul Pitanja i odgovori je vrlo sličan modulu *Wiki* stranica i zbog veličine ovog rada neće biti dodatno analiziran. Treba spomenuti samo da je njegova osnovna funkcionalnost omogućavanje akcija za postavljanje pitanja i davanje odgovora korisnicima, a dodatno je omogućeno rangiranje i ocjenjivanje odgovora.

U oba modula implementirano je napredno pretraživanje visokih performansi Apache LuceneTM. Ova tehnologija pogodna je za gotovo sve aplikacije koje zahtijevaju *pretraživanje kompletnog teksta* (eng. *full-text search*). Za predstavljanje traženih pojmova u rezultatima pretrage korištena je klasa *Lucene Highlighter* [95]. *Apache LuceneTM* je *projekt otvorenog koda* (eng. *open source project*) i dostupan za besplatno preuzimanje [96].

Rezultati pretrage interne baze podataka *FTICKM* sistema predstavljeni su kao top 5 članaka i top 5 pitanja ili odgovora. Osim internih rezultata, na osnovu ključnih riječi, prikazani su i eksterni rezultati pretrage, odnosno eksterne hiperveze prema originalnoj *Wikipediji*.

3.1.2 FITCKMS – Baza podataka

Na osnovu prikupljenih i analiziranih zahtijeva osmišljena je relacionala baza podataka *FITCKM* sistema. Za kreiranje baze podataka korištena je aplikacija *MicrosoftSQLServer 2014 Management Studio*. Centralna mjesta u zvjezdastoj strukturi zauzimaju dvije osnovne tabele *Korisnik* i *Post* (Slika 7.). Pomoćne tabele imaju ulogu da čuvaju vrijednosti za različite specifičnosti/generalizacije osnovnih tabela ili da ih na predviđeni način povezuju.

Sve predstavljene tabele imaju specifičnu ulogu u sistemu i u nastavku slijedi kratko objašnjenje:

- Tabela *Post* čuva podatke o svim člancima, pitanjima i odgovorima.

Značajne kolone:

- *PostVrsta* (tip: int)

vrijednosti: 1 - Članak; 2 - Članak izmjena; 5 - pitanje; 6 - odgovor

- *BrojPoena* (tip: double)

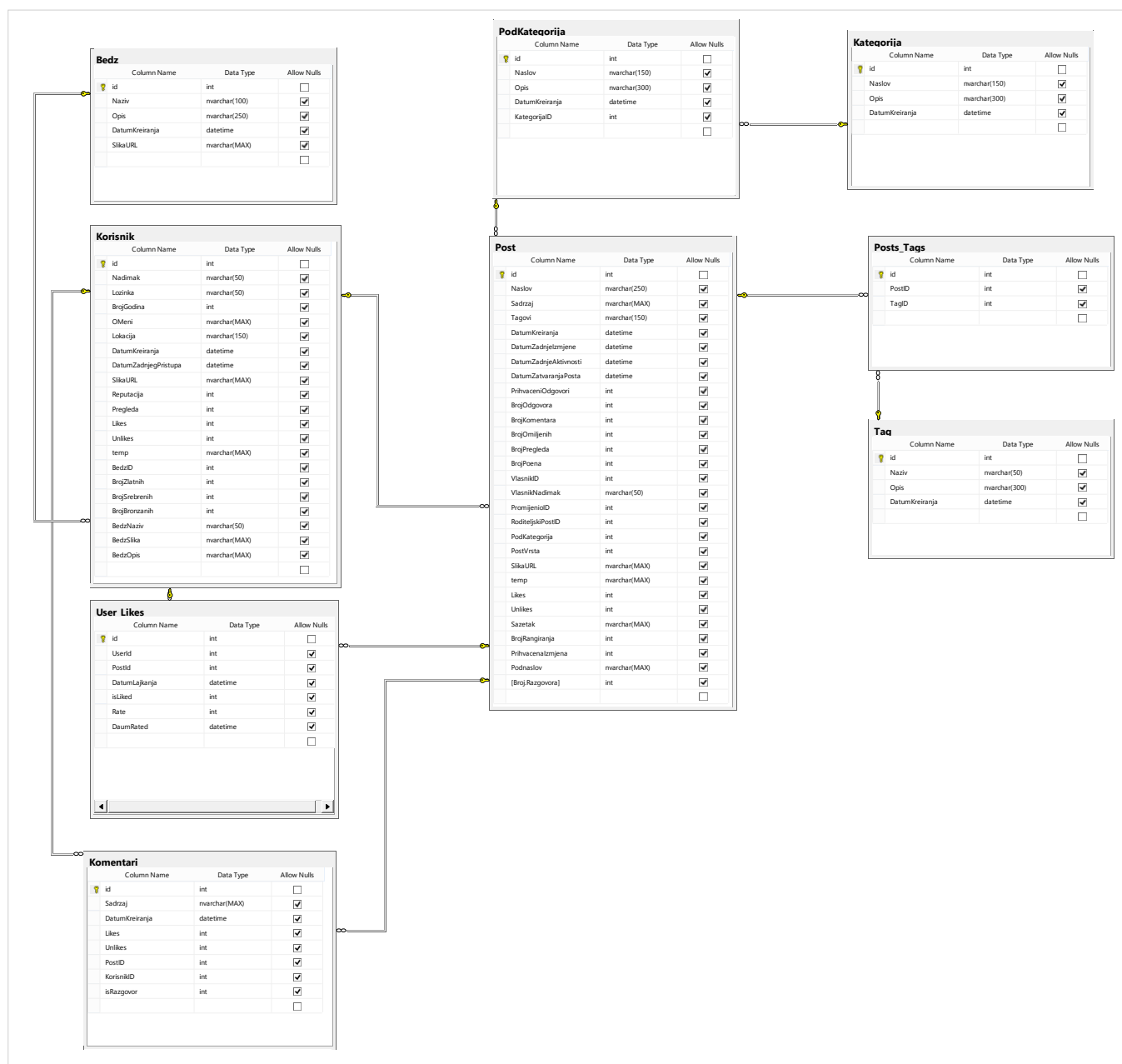
Kompleksni proračun o relevantnosti objekta objekta. U kalkulacije je uključen rejting, sviđanje, nesviđanje, kao i korisnički status koji je

učestvovao u navedenim akcijama. Ova kolona ima ulogu u glavnom proračunu za sistem preporuke.

- *VlasnikID* (tip: int)

Predstavlja vrijednost jedinstvenog identifikatora vlasnika odnosno kreatora objekta. Ova kolona ima ulogu da obezbijedi propisane akcije i vlasnička prava nad objektom. Dokazuje vlasništvo kako bi rejting objekta bio uključen u proračun statusa i ranga korisnika (Bedž) i učestvuje u preporuci sličnih korisnika.

- Tabela *Korisnik* čuva podatke o svim korisnicima u sistemu.
- Tabela *Tag* čuva podatke o svim oznakama u sistemu.
- Tabela *Posts_Tags* čuva podatke o svim oznakama svih objekata iz tabele *Post*.
- Tabela *User_Likes* povezuje tabele *Korisnik* i *Post*. Čuva podatke korisničkim akcijama rangiranja objekata iz tabele *Post* (rejting, svidanje, nesvidanje). Ima važnu ulogu u proračunu sistema preporuke.
- Tabela *Komentari* povezuje tabele *Korisnik* i *Post*. Čuva podatke o svim komentarima i razgovorima.
- Tabela *Bedž* čuva informacije o svim bedževima u sistemu. Proračunom se korisniku automatski dodjeljuje određeni bedž u zavisnosti od njegovog rejtinga u sistemu.
- Tabela *Potkategorija* čuva podatke o svim otvorenim temama u sistemu. Svi objekti tabele *Post* pripadaju jednom od objekata tabele *Potkategorija*.
- Tabela *Kategorija* čuva podatke o svim otvorenim oblastima u sistemu. Svi objekti tabele *Potkategorija* pripadaju jednom od objekata tabele *Kategorija*.



Slika 7. Baza podataka FITKMS

3.1.3 FITCKMS - Pregled mogućnosti aplikacije

U ovom poglavlju će biti navedeni najvažniji dijelovi *FITCKM* sistema koji su potrebni za istraživanje i testiranje hipoteza rada. Akcent je na kolaboraciji i dijeljenju znanja, a osnovna uloga sistema je da osigura validne podatke, koji će se koristiti kao ulazi u algoritme sistema preporuke.

Osnovne funkcionalnosti *FITCKM* sistema su:

- Registracija korisnika
- Prijava korisnika
- Kreiranje/uređivanje članaka
- Komentarisanje članaka
- Ocjenjivanje članaka (rejting, svidanje, nesvidanje)
- Razgovor o člancima
- Kreiranje pitanja /odgovora
- Ocjenjivanje pitanja ili odgovora (rejting, svidanje, nesvidanje)
- Pretraživanje članaka, pitanja ili odgovora
- Preporuka članaka
- Preporuka pitanja ili odgovora
- Preporuka sličnih korisnika
- Preporuka iz eksternog izvora

3.1.4 FITCKMS - Korisnički interfejs

Komunikacija sa korisnikom je jedan od najvažnijih zadataka aplikacije nezavisno od korištenog programskog jezika. *Grafički korisnički interfejs* (eng. *graphical user interface, GUI*) je metoda interakcije sa računarom kroz manipulaciju grafičkim elementima i dodacima uz pomoć tekstualnih poruka i obavještenja. Kod web aplikacija korisnički interfejs je fokusiran na ugodno okruženje, laganu navigaciju i brzo pronalaženje željenih informacija.

FITCKMS korisnički interfejs je prikazan u poglavlju 8, odnosno u prilogu dokumenta (*Slika 11., Slika 12. i Slika 13.*)

U narednom poglavlju bit će objašnjen razvoj eksperimentalnog okruženja, koje će između ostalog, koristiti spomenutu kolekciju podataka i testirati različite sisteme preporuke.

3.2 Eksperimentalno okruženje

Za potrebe istraživanja i testiranja hipoteza rada pristupljeno je razvoju eksperimentalnog okruženja. Izvorni kod je rađen u programskom jeziku *Java*, okruženje *Eclipse* [96], a kao osnovna biblioteka korištena je *Apache MahoutTM* [98] i to u verzijama:

- *Eclipse Standard/SDK, Version: Luna Release (4.4.0)*
- *Biblioteka, Apache MahoutTM, Version 0.9*

U biblioteci *Apache MahoutTM* integrisani su osnovni algoritmi za grupisanje, klasifikaciju i kolaborativno filtriranje. Distribuirana pod komercijalno-prijateljskom *Apache Software* licencom. Trenutno *Apache MahoutTM* podržava tri slučaja korištenja:

- *Preporuka (eng. Recommendation) koja na osnovu korisničkog ponašanja u sistemu koja pokušava pronaći resurse koji bi se potencijalno svidjeli korisniku*
- *Grupisanje (eng. Clustering) tekstualnih dokumenta u grupe lokalno vezanih dokumenata*
- *Klasifikacija (eng. Classification) koja na osnovu prethodnog učenja, odnosno učenja iz postojećih kategorizovanih dokumenata pokušava odrediti ispravnu kategoriju za neoznačene dokumente*

Više detalja o ovoj biblioteci se nalazi na sljedećem linku: <http://mahout.apache.org/>.

Za ovaj rad posebno je zanimljiv prvi slučaj korištenja, koji se odnosi na preporuke. Pakovanja i klase spomenute biblioteke, koristit će se za testiranje različitih sistema preporuke. Pored standardnih u praksi već korištenih metoda, posebna pažnja će biti usmjerena na adaptivno-agregativne metode.

Za testiranje korištena je kolekcija podataka *MovieLens*, koja se najčešće koristi za testiranje performansi sistema preporuke [60]-[85]-[92]-[93]. Sastoji se od skupa korisnika, filmova i ocjena korisnika filmove, na skali od 1 do 5. *MovieLens* kolekcija je dostupna u više formata i veličina, te je za ovaj eksperiment izabran podskup cijele *MovieLens* kolekcije, sa 100.000 ocjena od 943 korisnika i 1.682 filma.

Postoji nekoliko razloga zašto je za testiranje izabrana baš ova kolekcija. Korištenjem ogromnog skupa podataka, kao što cijela *MovieLens* kolekcija, ne postiže se značajna razlika u performansama [71]. Izabrana kolekcija je mnogo manjeg obima što odgovara složenosti izvođenja eksperimenata u ovom radu. Struktura kolekcije je slična podacima *FITCKM* sistema koji će se također koristiti u testiranju.

U prilogu dokumenta se nalazi prikaz programskog koda urađenog u programskom jeziku Java (*Slika 14.*).

4 METODE

U ovom poglavlju je objašnjena uloga inteligentnog programskog agenta u *FITCKM* sistemu i metode korištene u eksperimentalnom okruženju. *FITCKM* sistem je razvijen u okruženju *Visual Studio 2013*, i za te potrebe korišten je univerzalni *.NET* aplikativni okvir, uz programski jezik *C#*. Kompletan programski kod za eksperimente napisan je u programskom jeziku *Java*, koristeći *Eclipse* okruženje. Biblioteka *Apache MahoutTM*, koja je prethodno integrisana u okruženje, sadrži vrlo korisne klase koje su korištene u eksperimentima.

4.1 Uloga inteligentnog programskog agenta

U ovom dijelu rada ukratko je objašnjena simulacija inteligentnog programskog agenta u *FITCKM* sistemu. Metodologija kompletne implementacije agenta izlazi iz okvira ovog rada, te su u nastavku objašnjeni samo dijelovi interesantni i korisni za sisteme preporuke, odnosno za planirane eksperimente.

Kao što je već spomenuto, glavna uloga inteligentnog programskog agenta je dinamička izgradnja modela korisnika, odnosno obrada značajnih podataka u cilju poboljšanja kvaliteta sistema preporuke. Drugim riječima, agent bi neprestano trebao pratiti korisničko kretanje kroz sistem i trenutno stanje dostupnih resursa, odnosno objekata preporuke. Gledano iz perspektive individualnog korisnika, konkretni objekti preporuke su članci iz modula *FITWiki*, pitanja i odgovori iz modula *FITQ&A* i ostali korisnici sistema. Osim dinamičkih notifikacija korisniku (npr. logiranje, novi post-ovi najbližih komšija, novi komentari, novi rejting ili svidanje njegovih post-ova, novi odgovor na pitanje u kojem je korisnik na određeni način učestvovao i sl.), agent ima i neke važnije zadatke. Oni se uprošteno mogu opisati kao dinamičko pohranjivanje značajnih informacija o trenutnom stanju okruženja, zaključivanje i reakcija na osnovu njih u realnom vremenu. Kao primjer za navedeno može poslužiti trenutni rejting posmatranog korisnika i njegovo značenje u budućim aktivnostima u sistemu.

U nastavku ovog potpoglavlja detaljno će biti objašnjen koncept za rješavanje postavljenog zadatka korištenjem inteligentnog programskog agenta.

Na osnovu istorije kretanja korisnika kroz sistem i trenutnog stanja okruženja agent je, između ostalog, zadužen za dinamičku procjenu rejtinga korisnika. Osnova ovog koncepta je da vrednovanje određenog resursa ima različite vrijednosti u odnosu na individualni rejting korisnika. Drugim riječima, mišljenje korisnika sa visokim rejtingom sistem će više cijiniti, nego mišljenje korisnika koji imaju niži rejting. Dakle, dinamička procjena vrijednosti mišljenja (ocjenjivanje post-ova ostalih korisnika) trenutno aktivnog posmatranog korisnika, utiče na dinamičku procjenu mišljenja ostalih korisnika u sistemu. Kako bi se ovaj koncept primijenio u sistemu, određeni procesi agenta moraju „oslušivati“ okruženje i blagovremeno reagovati na određene događaje u njemu.

Konkretni događaji u ovom konceptu, koje bi agent trebao primijetiti i potencijalno reagovati, mogu biti različiti: novi post, komentar, razgovor, verzija članaka, prijava ili odjava korisnika, promjena rejtinga korisnika ili post-a. Znanje o tome se pohranjuje u bazu podataka. Reakcija agenta zavisi od njegove percepcije, odnosno odluke da li je promjena u okruženju je relevantna za određenog korisnika. Ako se posmatraju trenutno aktivni (prijavljeni) korisnici, reakcija bi trebala biti dinamičkog karaktera u vidu notifikacija, promjene individualnog korisničkog okruženja, promjene modela korisnika, promjena u modelu preporuke i sl. Ako je ipak riječ o korisnicima koji nisu aktivni, određene reakcije agenta mogu biti odložene do naredne prijave tih korisnika.

Iako se ovaj koncept može protumačiti kao komplikovan, ideja je vrlo jednostavna.

Pri proračunu rejtinga (Bedž-a) korisnika, ne koristi se samo broj objavljenih post-ova (članaka, razgovora, pitanja ili odgovora), već i njihov kvalitet. Kvalitet post-a određen je brojem poena koje je korisnik dobio na osnovu mišljenja drugih korisnika, putem ocjenjivanja post-a (rangiranje, svidanja, nesvidanja). Na osnovu broja poena, post može dobiti određene medalje. U programskom kodu inteligentnog agenta, medalje su simbolično nazvane imenima: *bronza*, *srebro* i *zlato* i određene brojčano na sljedeći način:

- *bronza*, 10 poena
- *srebro*, 20 poena
- *zlato*, 40 poena

Usljed promjene medalje post-a, pokreće se proračun rejtinga vlasnika post-a i ako je potrebno dodjeljuje mu se odgovarajući bedž. Na primjer, za bedž pionira korisnik mora skupiti najmanje pet bronzanih medalja, a za bedž juniora najmanje tri srebrene. Sličnom logikom bedževi se dodjeljuju sve do zvanja akademika, koji mora skupiti najmanje dvadeset zlatnih medalja. Njegovo mišljenje sistem smatra kao najvrijednije.

Ovakav koncept omogućen je na način da bedž predstavlja ponderisanu vrijednost koja se koristi u proračunu prilikom ocjenjivanja post-ova. U *FITCKM* sistemu korisnici mogu imati 10 vrsta bedževa, kojima su dodijeljeni odgovarajuće težine u rasponu od 0,6 do 1,0 (*Tabela 1.*).

Bedž	Težina
Akademik	1
Doktor	0,92
Profesor	0,85
Magistar	0,79
Inzinjer	0,74
Student	0,69
Senior	0,65
Junior	0,63
Pionir	0,61
Početaknik	0,6

Tabela 1. Težine bedževa

Kalkulacija finalne ocjene može se predstaviti jednostavnom formulom:

$$fri = wb * r, \quad (12)$$

U *formuli (12)* finalna ocjena post-a *fri* predstavlja produkt ocjene *r* na skali od 1 do 5 i težine bedža korisnika *wb*, koji ocjenjuje post. Na primjer, ako korisnik koji ocjenjuje post „nosi bedž profesora“ i ocijeni određeni post ocjenom 4, koristeći prethodnu formulu, finalna ocjena će imati vrijednost 3,4. Ipak, ako se radi o korisniku „nosi bedž studenta“, post ocijenjen ocjenom 4 će finalno imati vrijednost 2,76.

Ocjene, proizvedene upotrebom ovog koncepta, pohranjuju se u bazu podataka i kao takve koriste za preporuku. Potrebno je napomenuti da promjena bedža korisnika utiče samo na buduća ocjenjivanja post-ova, odnosno ocjene koje je korisnik u prošlosti dodijelio post-ovima, noseći neki drugi bedž, ostaju nepromijenjene.

Kakvi se rezultati očekuju primjenom inteligentnih programskih agenata u *FITKCM* sistemu?

Dinamičkim djelovanjem inteligentnog programskog agenta omogućene su notifikacije, proračuni rejtinga objekata i poboljšana preporuka u realnom vremenu za individualnog korisnika. Većina dosadašnjih rješenja, koja se bave problematikom rejtinga korisnika, bazirana su uglavnom na broju post-ova, a ne na njihovom kvalitetu. Implementacijom opisanog koncepta mogu se desiti neke vrlo logične situacije. Na primjer, moguća je situacija da jedan post ima veliki broj pozitivnih ocjena od nisko rangiranih korisnika, a da drugi, ima mali broj pozitivnih ocjena ali od visoko rangiranih korisnika. To znači da je moguće da drugi post ima višu vrijednost rejtinga u odnosu na prvi, iako ima manji broj pozitivnih ocjena. O ukupnom rejtingu korisnika u ovom slučaju odlučuje okolina, odnosno ostali korisnici, čije mišljenje nije na isti način vrijednovano. Dakle, rejting korisnika predstavlja varijablu, koja ne zavisi od toga da li je taj isti korisnik trenutno prijavljen u sistemu.

Težine bedževa mogu se razlikovati od sistema do sistema, zavisno od vrste, namjene, kapaciteta, zajednice koja ga koristi i sl. Na primjer, ako je riječ o akademskom okruženju, ove težine mogu biti blago kategorisane, u suprotnom mogu biti vrlo rigorozne, ali to bi bio predmet nekog drugog istraživanja.

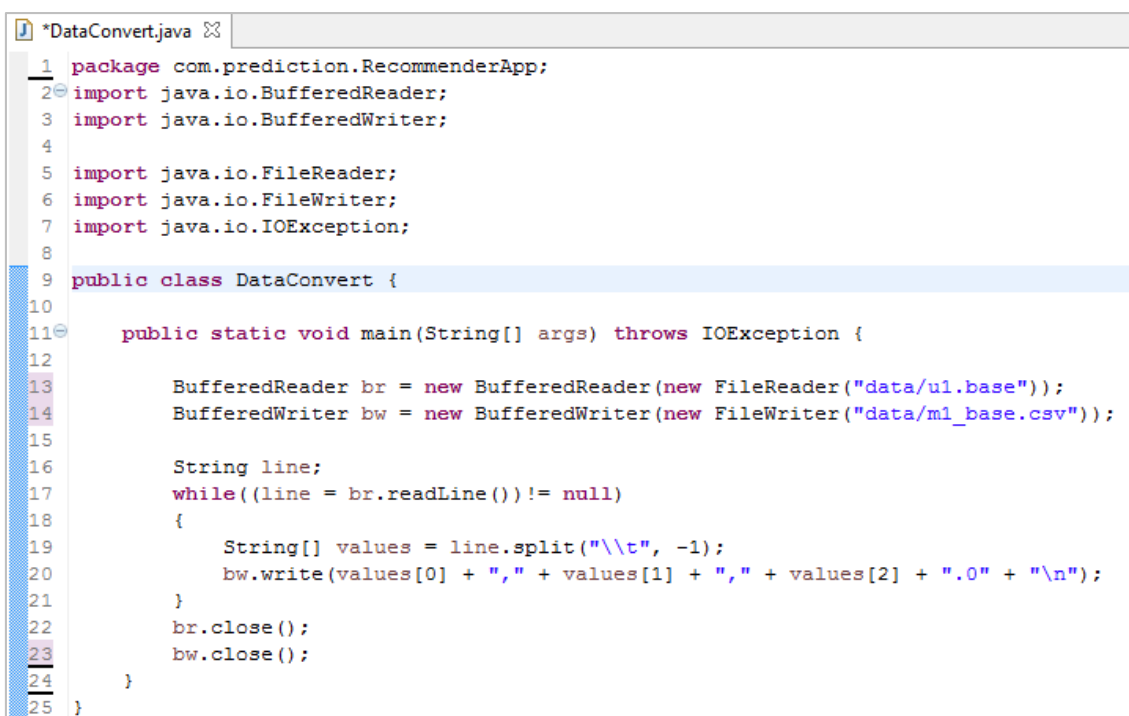
Iz perspektive sistema preporuke primjena ovakvih agenata može poboljšati kvalitet ulaznih podataka u algoritme, odnosno unaprijediti kvalitet sistema preporuke. Također, neocijenjeni postovi bi se mogli posmatrati na drugačiji način u sistemu preporuke, jer bi se ovom slučaju mogli rangirati o odnosu na rejting vlasnika, što opet otvara nove mogućnosti i ideje u budućnosti razvoja ovakvih sistema.

Predstavljeni koncept djelovanja inteligentnog programskog agenta omogućava dinamičko prilagođavanje opšteg okruženja (sistema) sa individualnim preferencijama korisnika i obrnuto.

4.2 Priprema podataka

Kao što je u prethodnim poglavljima spomenuto, u eksperimentima će se koristiti dvije kolekcije podataka, kolekcija *MovieLens*, i kolekcija eksportovana iz baze podataka *FITCKM* sistema. Za potrebe eksperimenata, podaci su prilagođeni u formatu *.csv* (eng. *comma-separated values*), i to određenim redoslijedom: *korisnik*, *resurs*, *ocjena*.

Kolekcija *MovieLens* prilagođena je pomoću klase *DataConvert*:

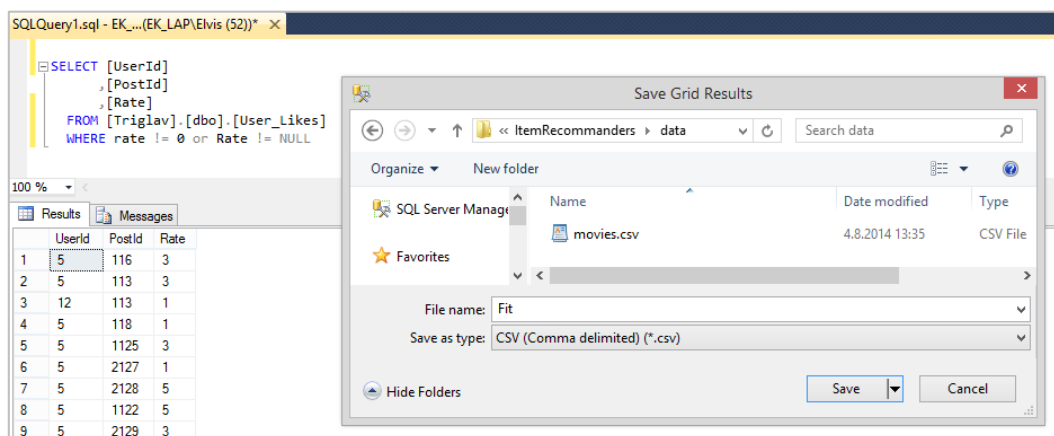


```
1 package com.prediction.RecommenderApp;
2 import java.io.BufferedReader;
3 import java.io.BufferedWriter;
4
5 import java.io.FileReader;
6 import java.io.FileWriter;
7 import java.io.IOException;
8
9 public class DataConvert {
10
11     public static void main(String[] args) throws IOException {
12
13         BufferedReader br = new BufferedReader(new FileReader("data/ul.base"));
14         BufferedWriter bw = new BufferedWriter(new FileWriter("data/ml_base.csv"));
15
16         String line;
17         while((line = br.readLine()) != null)
18         {
19             String[] values = line.split("\\t", -1);
20             bw.write(values[0] + "," + values[1] + "," + values[2] + ".0" + "\n");
21         }
22         br.close();
23         bw.close();
24     }
25 }
```

Slika 8. Klasa *DataConvert*

Kao što je prethodno spomenuto, skup je podijeljen na pet disjunktih podskupova, te je klasa *DataConvert* (Slika 8.) korištena za svaki od njih pojedinačno. Prilagođeni podaci su spremljeni u datoteku „*data*“, kao dokumenti: „*m1_base.csv*“, „*m2_base.csv*“, „*m3_base.csv*“, „*m4_base.csv*“ i „*m5_base.csv*“.

Kolekcija iz baze podataka *FITCKM* sistema kreirana je jednostavnim upitom (Slika 9.) i snimljena u istu datoteku pod nazivom „*Fit.csv*“.



Slika 9. SQL Upit - Spremanje podataka FITCKMS

FITCKM sistem je testnog karaktera, tako da kreirana kolekcija podataka predstavlja mnogo manji skup nego što je to slučaj u sistemima koji su u praktičnoj upotrebi. Kolekcija se sastoji od 230 ocjena, 66 post-ova i 21 korisnika.

```

1020
1021 //funkcija prema formuli fri =wb*r, racuna ponderisane ocjene i zapisuje ih u bazu podataka
1022 public void AgentCalculationRatings()
1023 {
1024     List<User_Likes> UL = new List<User_Likes>();
1025     UL = getUserLikesCorrectRatings();
1026     foreach (var item in UL) {
1027         item.RatePonder = item.Rate * getBedzPonder(item.UserId.Value);
1028         context.SaveChanges();
1029     }
1030 }
1031
1032 // funkcija vraća listu objekata validnih ocijenjenih postova
1033 public List<User_Likes> getUserLikesCorrectRatings()
1034 {
1035     var query = (
1036         from ul in context.User_Likes
1037         join p in context.Post on ul.PostId equals p.id
1038         where (p.PostVrsta == 1 || p.PostVrsta == 5) && (ul.Rate != null && ul.Rate != 0)
1039         select ul).ToList();
1040     return( List<User_Likes>)query;
1041 }
1042
1043 // funkcija vraća vrijednost pondera na osnovu korisnickog bedza
1044 public float getBedzPonder(int UID)
1045 {
1046     Korisnik k = getKorisnikByID(UID);
1047     switch (k.BedzID)
1048     {
1049         case 1: return 1.0f;
1050         case 2: return 0.92f;
1051         case 3: return 0.85f;
1052         case 4: return 0.79f;
1053         case 5: return 0.74f;
1054         case 6: return 0.69f;
1055         case 7: return 0.65f;
1056         case 8: return 0.63f;
1057         case 9: return 0.61f;
1058         case 10: return 0.6f;
1059         default: return 0f;
1060     }
1061 }

```

Slika 10. Simulacija programskog agenta

Radi testiranja uloge inteligentnog programskog agenta, simulacija programskog agenta ima zadatak da na osnovu rejtinga post-ova u *FITCKM* sistemu proračuna bedž vlasnika, te da na osnovu njega napravi pondere budućih ocjenjivanja tog istog vlasnika. Simulacija agenta se zasniva na pokretanju programskog koda koji na osnovu korisničkog bedža i postojećih ocjena, kreira nove ponderisane ocjene i zapisuje ih u bazu podataka. Programski kod je rađen pomoću *ASP.NET tehnologije*, uz korištenje programskog jezika *C#* (*Slika 10.*).

Nakon izvršavanja programskog koda podaci su uspješno spremljeni u bazu podataka i eksportovani na sličan način kao što je prikazano na (*Slika 9.*), pod nazivom „*FitPonder.csv*“. Ovaj postupak je omogućio testiranje sistema preporuke koristeći dva skupa podataka *FITCKM* sistema, sa istim kardinalitetom. Svi elementi, spomenutih skupova, su ostali identični, osim ocijena koje u drugom skupu imaju ponderisanu vrijednost u odnosu na bedževe korisnika, koji su ocjenjivali post-ove.

Na ovaj način svi podaci su pripremljeni za izvođenje planiranih eksperimenata.

4.3 Modeli sistema preporuke

Nakon pripreme podataka, potrebno je definisati skup sistema preporuke koji će se koristiti u eksperimentima. Standardni sistemi preporuke će se koristiti za standardna predviđanja i preciznost procjene, kao što je opisano u *poglavlju 2.3.3.*

U cilju pronalaženja disjunktin obrazaca u podacima, potrebno je izabrati metode koje sa različitih aspekata obrađuju podatke. U *Tabeli 2.* su predstavljeni izabrani sistemi preporuke za eksperimente. Prva kolona odnosi se na vrstu sistema preporuke, gdje oznaka (*S*) predstavlja standardne sisteme, a oznaka (*A*) agregativne. Za sve standardne sisteme izvršiće se vremenska procjena izvođenja, kroz konkretne primjere preporuke. Svi agregativni sistemi će koristiti svaki standardni sistem preporuke.

	<i>Metoda</i>	<i>Algoritam</i>	<i>Opis</i>
<i>S</i>	<i>PCC</i>	<i>Pearson correlation</i>	<i>Slični korisnici</i>
<i>S</i>	<i>cosine</i>	<i>Cosine similarity</i>	<i>Slični resursi</i>
<i>S</i>	<i>baseline</i>	<i>Baseline</i>	<i>Prosjeci korisnika i resursa</i>
<i>S</i>	<i>item_avg</i>	<i>Baseline</i>	<i>Standardni prosjeci resursa</i>
<i>S</i>	<i>slope_one</i>	<i>Slope One</i>	<i>Delta proračun rejtinga</i>
<i>S</i>	<i>svd1</i>	<i>SVD</i>	<i>ALSWR factorizer, 10 funkcija</i>
<i>S</i>	<i>svd2</i>	<i>SVD</i>	<i>ALSWR factorizer, 20 funkcija</i>
<i>S</i>	<i>svd3</i>	<i>SVD</i>	<i>SVDPlusPlusFactorizer, 10 funkcija</i>
<i>S</i>	<i>svd4</i>	<i>SVD</i>	<i>SVDPlusPlusFactorizer, 20 funkcija.</i>
<i>A</i>	<i>median</i>	<i>Aggregation</i>	<i>Medijana agregovanih rejtinga</i>
<i>A</i>	<i>average</i>	<i>Aggregation</i>	<i>Prosjek agregovanih rejtinga</i>
<i>A</i>	<i>adaptive</i>	<i>Adaptive Aggregation</i>	<i>Prilagođeni agregovani rejtingi</i>

Tabela 2. Sistemi preporuke

4.3.1 Sistemi preporuke

Ovo poglavlje daje kratak opis korištenih metoda u eksperimentima. Detaljne informacije o sistemima preporuke se nalaze u poglavlju 2.3. a prikaz implementaciju u prilogu (*Slika 15.*, *Slika 16.*).

4.3.1.1 Standardni (tradicionalni) sistemi preporuke

Kao što je predstavljeno u *Tabeli 2.* koristiće se dvije vrste sistema preporuke. Standardni sistemi preporuke (*S*), koriste različite obrasce među podacima za predviđanje ocjena, što predstavlja osnovni razlog izbora ovako širokog spektra sistema.

Kao što je već objašnjeno, performanse agregativnih metoda (*oznaka A*) više zavise od različitosti standardnih sistema preporuke, nego od njihovog individualnog izvođenja [89]. Radi boljeg razumijevanja različitosti ovih obrazaca, u nastavku su ukratko predstavljeni osnovni principi izvođenja standardnih sistema preporuke, koji su definisani u prethodnom poglavlju.

SVD metode traže globalne obrasce u podacima smanjivanjem rejting-prostora u koncept-prostora. Smanjivanjem ovog prostora, algoritam je u mogućnosti pronaći latentne

odnose, kao što su grupe resursa koji imaju isti ili sličan obrazac rejtinga, ili grupe korisnika koji često ocjenjuju resurse na sličan način. U *SVD metodama*, faktorizacija se odnosi na algoritme koji se koriste za razlaganje podataka (na faktore), te njihovo pohranjivanje u rejting matricu.

Slope One i *Baseline* algoritmi koriste prosječne ocjene resursa i korisnika, te na taj način predviđaju ocjenu. To su jednostavni algoritmi koji se često koriste u kombinaciji u okvirima složenijih pristupa.

Kosinus sličnosti je algoritam koji koristi resurse koji su na sličan način ocijenjeni od strane određenog korisnika, i na osnovu njih predviđa relevantnost za nepoznate resurse. Kosinusova mjera je mjera sličnosti između dva vektora koji se nalaze unutar n dimenzija pronalazeći kosinus ugla između njih. Kosinusova mjera je često korištena za upoređivanje dokumenata u pretraživanju i obradi teksta.

Pearsonov koeficijent korelacije (PCC) algoritam koristi još jedan pristup. Ovaj algoritam, ima sličnu strategiju kao i kosinus sličnosti, traži korisnike sa sličnim obrascima rejtinga. Sličnost se mjeri *Pearsonovim koeficijentom korelacije*. Predviđanje se stvara prikupljanjem ocjena od sličnih korisnika, za posmatrani resurs, ponderisanjem njihovih sličnosti.

Osnovna razlika između agregativnih i standardnih sistema preporuke su obim obrazaca koje koriste. *SVD* i standardne metode su usmjerene na globalne efekte, kao što su latentne kategorije i ukupne prosječne ocjene. *Kosinus sličnosti* i *PCC algoritmi* posmatraju manje klastere sličnih korisnika i resursa, i računaju prosječnu ocjenu ponderisanu na osnovu sličnosti elemenata.

Ovaj široki spektar sistema preporuke bi trebao olakšati posao adaptivno-agregativnim metodama, koje će u ovom radu biti bazirane na disjunktним obrascima za predviđanje rejtinga.

4.3.1.2 Adaptivni sistemi preporuke

Drugi tip sistema preporuke, koji će se koristiti u eksperimentima, su agregativne metode, predstavljene oznakom (A) u *Tabeli 2*. Ove metode kombinuju rezultate svakog od standardnih sistema preporuke. Prve dvije metode imaju vrlo jednostavan pristup agregaciji. *Medijan metoda* agregacije izračunava vrijednost medijane predviđanja, proizvedenu pomoću standardnih sistemima preporuke. Slično, *agregativna metoda prosjeka* (eng. *average*) uzima srednju vrijednost standardnih predviđanja. Navedene agregativne metode nisu kompleksne po svojoj prirodi, te će biti od pomoći u poređenju adaptivno-agregativnih metoda sa jednostavnim tehnikama agregacije.

Posljednji red u *Tabeli 2*. odnosi se na adaptivni pristup, koji je detaljno predstavljen u poglavlju 2.3.3.2. Ovaj pristup se nalazi na drugom sloju i izvršava sekundarnu procjenu preciznosti standardnih sistema preporuke. Procjena preciznosti podrazumijeva proračun adaptivnih težina (pondera) standardnih metoda, koje na kraju određuju koliko će koji algoritam biti angažovan u adaptivnoj preporuci.

Prvi korak je procjena greške u predviđanjima za svaku od standardnih metoda. Procijenjene vrijednosti greške se nalaze na istoj skali kao i predviđanje rejtinga (od 1 do 5), tako da niža vrijednost greške predstavlja preciznije predviđanje rejtinga. Drugi korak je normalizacija grešaka u odnosu na ukupnu grešku svih predviđanja. Nakon normalizacije grešaka moguće je proračunati srednju vrijednost greške za agregativne metode uključujući i adaptivni pristup.

Vrijednosti adaptivnih težina (pondera) predstavljaju ustvari opozitne vrijednosti normalizovanih grešaka za svaki od standardnih sistema preporuke.

4.3.2 Procjena greške

Za ocjenu modela tokom predviđanja, potrebno je odrediti mjeru za računanje ukupne greške u velikom broju predviđanja. Uobičajena mjera za procjenu greške sistema preporuke je *RMSE* (eng. *root mean squared error*) mjera^{[57],[60],[93]}. *RMSE* se obično koristi za predstavljanje devijacija između predviđanja i trenutnog rejtinga i može se predstaviti sljedećom formulom:

$$RMSE(\hat{R}, R) = \sqrt{\frac{\sum_{i=1}^n (\hat{R}_i - R_i)^2}{n}} , \quad (13)$$

U *formuli (13)* n je ukupan broj predviđanja. Prethodna formula kombinuje skup grešaka u jednu kombinovanu grešku na način da skup procijenjenih predviđanja \hat{R} poredi sa skupom trenutnih vrijednosti rejtinga R .

Korisna karakteristika $RMSE$ mjere je da će rezultat greške biti na istoj skali kao i procjene. Na primjer, ako se predviđaju vrijednosti na skali od 1 do 5, izračunate greške će biti na istoj skali. Konkretno, ako greška ima vrijednost 1 onda se može konstatovati da udaljenost od ispravne ocjene u prosjeku iznosi 1.

$RMSE$ predstavlja nelinearni način procjenjivanja greške, što znači da su veće greške oštrije sankcionisane. Zbog kvadratnih razlika u formuli, nekolicina velikih grešaka ima mnogo veći značaj nego mnoštvo manjih. Dakle, u $RMSE$ pristupu prednost imaju metode koje daju stabilnija predviđanja, u odnosu na precizne metode koje, zbog nekoliko korisnika ili resursa, ne daju željeni rezultat.

$RMSE$ će se koristiti za procjenu svih standardnih i agregativnih metoda korištenih u eksperimentima. Rezultati sa dodatnim objašnjenjima predstavljeni su u sljedećem poglavlju.

5 EKSPERIMENTI I REZULTATI

U ovom poglavlju su predstavljeni provedeni eksperimenti i njihovi rezultati, na osnovu kojih bi se trebale potvrditi ili odbaciti hipoteze ovog rada (potpoglavlja 1.1.1 i 1.1.2). U skladu sa temom ovog rada, eksperimenti su uglavnom usmjereni na testiranje standardnih metoda preporuke u službi adaptivno-agregativnog pristupa i na ulozi inteligentnih programskih agenata u obradi podataka za sisteme preporuke.

5.1 Eksperiment 1

U ovom eksperimentu testirane su standardne metode (*S*) nad skupom podataka iz baze *FITCKM* sistema. Top 6 rezultata preporučenih resursa svih standardnih metoda za proizvoljno izabranog konkretnog korisnika (jedinstveni identifikator 5) prikazani su u *Tabeli 3.*:

<i>tip</i>	<i>metoda</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
<i>S</i>	<i>pearson</i>	4205 4,5359	4180 4,4641	4179 4,3489	4181 4,3022	3154 4,2933	4174 4,0000
<i>S</i>	<i>cosine</i>	4204 4,0000	4193 3,6667	4189 3,6643	4187 3,5729	2131 3,5678	4199 3,5508
<i>S</i>	<i>baseline</i>	4178 4,3913	4196 4,3913	4171 4,3913	3142 4,3913	3154 4,2663	4204 4,1413
<i>S</i>	<i>item_avg</i>	4189 3,6651	4193 3,6645	4169 3,5082	4183 3,5069	4179 3,5062	2130 3,5062
<i>S</i>	<i>slope_one</i>	4189 5,5000	4196 4,5000	4171 4,5000	3142 4,2500	4204 4,2500	4190 4,0000
<i>S</i>	<i>SVD_1</i>	4176 4,0709	4208 2,9383	3154 2,4073	4203 2,3584	4199 1,8057	4202 1,7706
<i>S</i>	<i>SVD_2</i>	4196 3,6613	3142 3,6263	4171 3,5420	4178 3,4967	4192 3,4256	3154 3,3725
<i>S</i>	<i>SVD_3</i>	3154 4,0881	4180 3,6764	4174 3,6318	4173 3,6095	4181 3,5928	3145 3,5754
<i>S</i>	<i>SVD_4</i>	3154 4,1458	4180 3,7382	4174 3,6984	4173 3,6414	3145 3,6295	4181 3,6280

Tabela 3. Preporučeni resursi (FITCKMS baza podataka)

U *Tabeli 3.* redovi predstavljaju standardne metode, a kolone top 6 preporučenih resursa. U vrhovima ćelija nalaze se jedinstveni identifikacioni brojevi, a u dnu vrijednost ocjene

za pripadajući resurs. Ne razmatrajući relevantnost preporučenih resursa, različitost svih rezultata ukazuje na to, da korištene standardne metode koriste različite obrasce među podacima, odnosno veze između podataka se u većini metoda posmatraju sa različitih *aspekata*. U koloni 1, koja predstavlja prvi preporučeni resurs, može se primijetiti da su metode *item_avg* i *slope_one* preporučile isti resurs (id: 4189), kao i metode *SVD_3* i *SVD_4*, koje su preporučile resurs (id: 3154). U obzir treba uzeti i činjenicu da je korišten veoma mali skup podataka za testiranje (230 ocjena), te da bi u slučaju korištenja većeg skupa ova razlika bila mnogo izraženija. Kako bi prethodna izjava bila i dokazana, standardne metode su testirane na mnogo većem skupu (*MovieLens*, podskup *d1*), koji se sastoji od oko 20.000 ocjena. Korisnik je također proizvoljno izabran (jedinствени identifikator 5), a rezultati predviđanja su prikazani u *Tabeli 4*.

<i>tip</i>	<i>metoda</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
<i>S</i>	<i>pearson</i>	212 5,0000	515 5,0000	639 5,0000	640 4,5000	86 4,5000	170 4,5000
<i>S</i>	<i>cosine</i>	1654 4,3750	1500 4,2857	1477 4,2500	1417 4,2500	1390 4,2500	1243 4,2500
<i>S</i>	<i>baseline</i>	1599 4,4277	1653 4,4277	1467 4,4277	1122 4,4277	1500 4,4277	1189 4,4277
<i>S</i>	<i>item_avg</i>	1354 5,0000	114 4,4945	515 4,4006	206 4,3866	1654 4,3705	654 4,3049
<i>S</i>	<i>slope_one</i>	1080 5,5000	1368 5,4074	1367 5,1667	1293 5,0000	1233 4,9167	1344 4,6296
<i>S</i>	<i>SVD_1</i>	1367 5,1324	1643 5,1263	1167 4,7713	1385 4,7000	1160 4,6487	320 4,6337
<i>S</i>	<i>SVD_2</i>	851 4,4689	1467 4,3565	1347 4,3392	1642 4,3189	867 4,3018	1524 4,2039
<i>S</i>	<i>SVD_3</i>	318 4,6901	515 4,6313	64 4,6185	483 4,5959	302 4,5897	127 4,5610
<i>S</i>	<i>SVD_4</i>	483 4,7578	64 4,7318	318 4,7071	12 4,7008	603 4,6661	127 4,6085

Tabela 4. Preporučeni resursi (MovieLens podskup d1)

Rezultati iz *Tabele 4*. potvrđuju različitost preporučenih resursa, odnosno da izabrane standardne metode koriste različite obrasce pri generisanju preporuke. Standardne metode i njihov način rada su već opisane u potpoglavljima 2.3.2 i 4.3.1 i ovaj dokaz ne predstavlja ništa novo, ali je vrlo bitan kao podloga za izvođenje narednih eksperimenata, pogotovo za testiranje adaptivno-agregativnog pristupa.

5.2 Eksperiment 2

Drugi eksperiment ima zadatak da procjeni greške za sisteme preporuke *RMSE* metodom u kojoj se skupovi podataka dijele na trening i testne skupove. U tom procesu kreira se novi trening model podataka, te se procijenjene karakteristike sistema preporuke upoređuju sa aktuelnim testnim podacima.

U ovom eksperimentu korištena je kolekcija podataka *MovieLens* podijeljena u 5 disjunktnih skupova u kolonama *d1*, *d2*, *d3*, *d4* i *d5*. Na ovaj način je osigurana pouzdanost rezultata, jer bi se korištenjem jednog skupa rezultati mogli protumačiti kao slučajni. Kolona *d6* predstavlja kolekciju podataka *FITCKM* sistema, koja je, kao što je spomenuto, mnogo manjeg obima. Rezultati standardnih sistema (*S*) su korišteni u procjeni greške za agregativne sisteme (*A*).

<i>tip</i>	<i>metoda</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>	<i>d4</i>	<i>d5</i>	<i>d6</i>
<i>S</i>	<i>pearson</i>	1,2477	1,1683	1,2381	1,2204	1,2375	1,1660
<i>S</i>	<i>cosine</i>	1,0393	1,0483	1,1054	1,0483	1,0520	1,2238
<i>S</i>	<i>baseline</i>	0,9724	0,9740	0,9771	0,9835	0,9822	1,5887
<i>S</i>	<i>item_avg</i>	0,9717	0,9605	0,9833	0,9921	0,9783	1,7255
<i>S</i>	<i>slope_one</i>	0,9471	0,9568	0,9651	0,9624	0,9572	1,7135
<i>S</i>	<i>SVD_1</i>	1,0057	1,0062	1,0040	0,9976	1,0064	2,0414
<i>S</i>	<i>SVD_2</i>	1,0083	1,0189	1,0234	1,0170	1,0282	1,6844
<i>S</i>	<i>SVD_3</i>	1,0903	1,0928	1,0892	1,1075	1,1028	1,2567
<i>S</i>	<i>SVD_4</i>	1,0730	1,0770	1,0759	1,1228	1,1211	1,2599
<i>A</i>	<i>median</i>	0,7969	0,8038	0,8055	0,8016	0,8079	1,3040
<i>A</i>	<i>average</i>	0,8126	0,8077	0,8151	0,8193	0,8232	1,2506
<i>A</i>	<i>adaptive</i>	0,7218	0,7177	0,7241	0,7278	0,7311	1,0951

Tabela 5. *RMSE* vrijednosti sistema preporuke

U ćelijama Tabele 5. nalaze se *RMSE* vrijednosti za sve sisteme preporuke. Niže *RMSE* vrijednosti predstavljaju bolje rezultate. U ovom eksperimentu, rezultati dokazuju da agregativne metode imaju mnogo bolje rezultate u odnosu na standardne metode. Adaptivno-agregativna metoda je ostvarila najbolji rezultat, što bi moglo opravdati njenu primjenu u praksi.

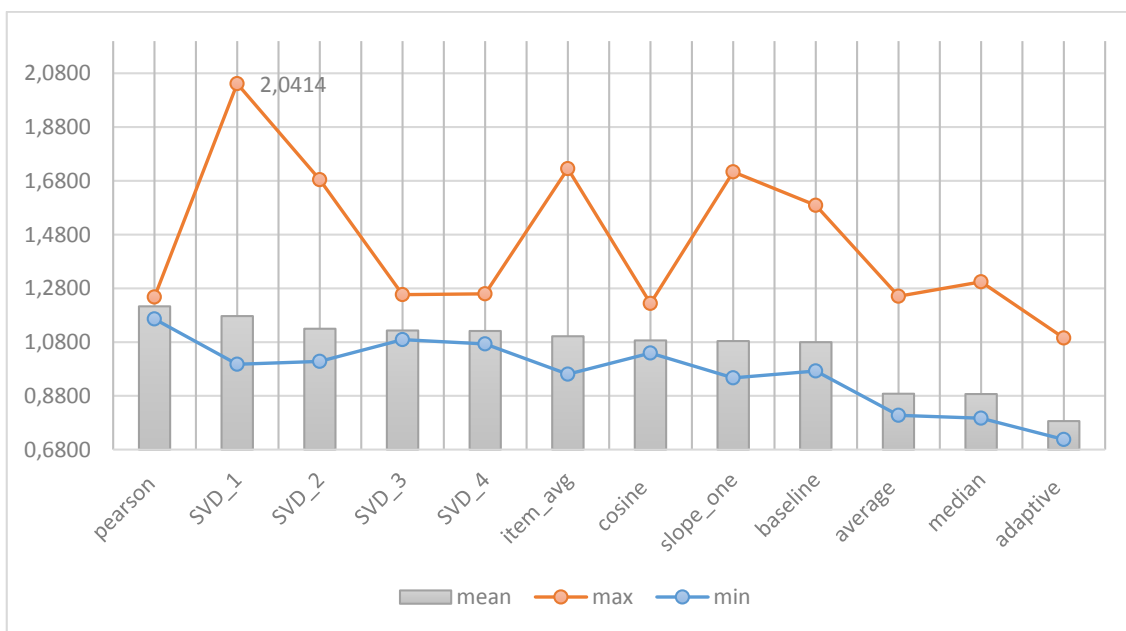
<i>tip</i>	<i>metoda</i>	<i>min</i>	<i>max</i>	<i>mean</i>	<i>stdev</i>
<i>S</i>	<i>pearson</i>	1,1660	1,2477	1,2130	0,0366
<i>S</i>	<i>cosine</i>	1,0393	1,2238	1,0862	0,0715
<i>S</i>	<i>baseline</i>	0,9724	1,5887	1,0797	0,2494
<i>S</i>	<i>item_avg</i>	0,9605	1,7255	1,1019	0,3057
<i>S</i>	<i>slope_one</i>	0,9471	1,7135	1,0837	0,3086
<i>S</i>	<i>SVD_1</i>	0,9976	2,0414	1,1769	0,4235
<i>S</i>	<i>SVD_2</i>	1,0083	1,6844	1,1300	0,2717
<i>S</i>	<i>SVD_3</i>	1,0892	1,2567	1,1232	0,0658
<i>S</i>	<i>SVD_4</i>	1,0730	1,2599	1,1216	0,0715
<i>A</i>	<i>median</i>	0,7969	1,3040	0,8866	0,2045
<i>A</i>	<i>average</i>	0,8077	1,2506	0,8881	0,1777
<i>A</i>	<i>adaptive</i>	0,7177	1,0951	0,7863	0,1514

Tabela 6. Statističke vrijednosti sistema preporuke

U *Tabeli 6.* prikazana je statistika sistema preporuke, koristeći rezultate iz *Tabele 5.* Korištene su statističke vrijednosti: minimum (*min*), maksimum (*max*), srednja vrijednost (*mean*) i standardno odstupanje (*stdev*). Rezultati iz ove tabele potvrđuju rezultate iz *Tabele 5.*, u kojoj je adaptivno-agregativni pristup ostvario najbolje rezultate.

Standardno odstupanje od sredine ima visoke vrijednosti za svaku metodu, što bi predstavljalo veliki problem, ako je ono rezultat nestabilnog predviđanja nad istim skupom podataka. Međutim, u ovom eksperimentu odstupanje je uzrokovano različitim rezultatima nad različitim skupovima podataka, što predstavlja opravdanje za ovako relativno visoke vrijednosti. Maksimalna odstupanja se uglavnom mogu prepisati skupu *d6*, koji se po svim karakteristikama razlikuje od ostalih podskupova koji su uglavnom zaslužni za vrijednosti minimalnih odstupanja. Minimalne vrijednosti su većinom ujednačene, što ide u prilog metodologiji izvođenja eksperimenta.

Na *Grafikonu 1.* predstavljena su minimalna i maksimalna odstupanja od srednje *RMSE* vrijednosti za svaku metodu.



Grafikon 1 Minimalna i maksimalna odstupanja od srednje RMSE vrijednosti

Iako rezultati prikazani na *Grafikonu 1.* na ukazuju na to da je *RMSE* metoda uspješnija korištenjem veće kolekcije podataka, također je dokazano da je istu moguće primijeniti i na manjim kolekcijama, sa nešto višim *RMSE* vrijednostima.

5.3 Eksperiment 3

Treći eksperiment bi trebao procijeniti opravdanost upotrebe i uloge inteligentnog programskog agenta, opisanog u poglavlju 4.1. U *eksperimentu 1* i *2* korišteni su podaci iz „*Fit.csv*“ skupa, *FITKMS* baze podataka bez implementacije inteligentnog programskog agenta. U ovom eksperimentu je korišten spomenuti skup, preuzetih iz *Tabele 3.*, ali i prilagođeni skup „*FitPonder.csv*“, sa ponderisanim vrijednostima ocjena. Za preporuku su korištene standardne metode, kao i u prethodnim eksperimentima.

Cilj ovog eksperimenta je dokazivanje različitosti preporučenih resursa korištenjem istih standardnih sistema preporuke, nad kardinalnim skupovima podataka. Razlika između tih skupova se ogleda samo u ocjenama, odnosno načinu pohranjivanja ocjena, dok su ostali elementi ostali nepromijenjeni. U prvom skupu su pohranjene standardne vrijednosti ocjena, a u drugom ponderisane, upotrebom inteligentnog agenta, što je objašnjeno u poglavlju 4.1.

Postizanje bilo kakve razlike u predviđanjima na ovakav način bi otvorilo razne teme za diskusiju, pogotovo one koje se odnose na upotrebu ovakvih ili sličnih inteligentnih programskih agenata u adaptivno-agregativnim sistemima preporuke.

U *Tabeli 7.* prikazani su rezultati *eksperimenta 3.*, odnosno komparacija top 6 rezultata preporuke, za proizvoljno izabranog konkretnog korisnika (*jedinstveni identifikator 5*), koristeći standardne metode nad spomenutim skupovima.

<i>metoda</i>	<i>rejting</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
<i>pearson</i>	<i>r</i>	4205	4180	4179	4181	3154	4174
	<i>pr</i>	4205	4181	4179	3154	2130	4180
<i>cosine</i>	<i>r</i>	4204	4193	4189	4187	2131	4199
	<i>pr</i>	4204	4193	4189	4187	2131	4199
<i>baseline</i>	<i>r</i>	4178	4196	4171	3142	3154	4204
	<i>pr</i>	3142	4196	3154	4183	4178	4204
<i>item_avg</i>	<i>r</i>	4189	4193	4169	4183	4179	2130
	<i>pr</i>	4189	4193	4169	4183	2130	3158
<i>slope_one</i>	<i>r</i>	4189	4196	4171	3142	4204	4190
	<i>pr</i>	4189	4196	4204	4190	4171	3142
<i>SVD_1</i>	<i>r</i>	4176	4208	3154	4203	4199	4202
	<i>pr</i>	4187	4199	3160	4183	4179	3144
<i>SVD_2</i>	<i>r</i>	4196	3142	4171	4178	4192	3154
	<i>pr</i>	4180	3145	4195	4169	3160	4189
<i>SVD_3</i>	<i>r</i>	3154	4180	4174	4173	4181	3145
	<i>pr</i>	3154	4205	4180	4181	2130	4186
<i>SVD_4</i>	<i>r</i>	3154	4180	4174	4173	3145	4181
	<i>pr</i>	3154	4205	4180	4181	2130	4186

Tabela 7. Komparacija rezultata preporuke standardnih metoda nad različitim skupovima iz baze podataka FITCKMS

U *Tabeli 7.* prva kolona predstavlja standardne metode preporuke testirane na skupu sa standardnim ocjenama sa oznakom *r*, i skupu sa ponderisanim ocjenama sa oznakom *pr* (*kolona 2*). Ostale kolone, sa oznakama od 1 do 6, predstavljaju top 6 preporučenih resursa za oba skupa. Čelije predstavljaju jedinstveni identifikator resursa u bazi podataka. Plavom i zelenom bojom su označene razlike u predviđanjima koristeći iste standardne metode nad spomenutim skupovima. Kao što je već spomenuto, broj elemenata oba skupa je veoma mali (*kardinalitet 230*) i postizanje bilo kakve razlike u

predviđanjima predstavlja odličan rezultat. Realna su očekivanja da bi u većim skupovima ova razlika bi bila još izraženija, što će biti predmet diskusije u narednom poglavlju, ali ta teorija se zbog veličine ovog rada neće provjeravati.

Može se primijetiti da sve metode, osim kosinusa sličnosti, imaju dosta različita predviđanja. Uzrok istih predviđanja za metodu kosinusa sličnosti može da bude metodologija simulacije inteligentnog programskog agenta, u kojem su postojeće ocjene, ponderisane korisničkim rejtinzima, tako da se odnos između resursa nije drastično mijenjao. Ovu teoriju bi također bilo potrebno provjeriti upotrebom većih skupova, ali ona nije toliko bitna za dokazivanje hipoteza ovog rada, te može postati predmet budućih istraživanja. Jedan od mogućih zaključaka može biti da upotreba inteligentnog programskog agenta u kombinaciji sa metodom kosinus sličnosti nad malim skupovima podataka nije opravdana, što bi odbacilo glavnu hipotezu (*H1*) ovog rada. Međutim, pošto je u pitanju samo jedna standardna metoda, koja zavisi od dodatnih istraživanja, hipoteza ne može biti kompletno odbačena.

Pod pretpostavkom da upotreba inteligentnog programskog agenta za metodu *kosinus sličnosti* nije opravdana, potrebno je protumačiti rezultate ostalih standardnih metoda. Razlike u predviđanjima, dokazuju da je upotreba inteligentnog programskog agenta u kombinaciji sa ovim metodama sasvim opravdana. U ovom slučaju, rezultati predviđanja su precizniji u odnosu na rezultate iz *Tabele 3.*, jer su dodatno balansirana i filtrirana prema ocjenama koje su upotrebom ovog agenta ponderisane prema rejtinzima korisnika sistema.

Ako se izuzme neistražena metoda kosinusa sličnosti, standardne metode u kombinaciji sa programskim agentom koje su korištene u ovom radu su unaprijeđene kvalitetom ulaznih podataka i kao rezultat imaju preciznija predviđanja, čime je potvrđena pomoćna hipoteza ovog rada (*H1.1*):

- *H1.1: Praćenjem korisnika kroz sistem moguće je otkriti nove odnose među podacima i stvoriti nove metapodatke, te iskoristiti njihovo značenje*

Kako je razvijen inteligentni programski agent *FITCKM* sistema obučen da djeluje samostalno i stvara novo znanje bez ljudske pomoći, može se zaključiti da je postignut određeni vid umjetne inteligencije, te se i glavna hipoteza rada (*H1*) može smatrati potvrđenom:

- *H1: Razvojem inteligentnih agenata, u kontekstu razotkrivanja nedavnih/trenutnih korisničkih kretanja, interesovanja i potreba, stvara se pretpostavka da je moguće razviti nove oblike znanja (metapodatke), i pomoću njih unaprijediti modele preporuke usmjerene ka konkretnom entitetu, odnosno postići viši nivo umjetne inteligencije sistema.*

Na kraju je potrebno provjeriti šta bi rezultati iz *Tabele 7.* mogli značiti za agregativne metode korištene u ovom radu. Kao što je više puta do sada spomenuto, ove tehnike koriste rezultate standardnih sistema preporuke. Dokazivanjem hipoteza (*H1* i *H1.1*) potvrđeno je da su rezultati predviđanja standardnih sistema preporuke postigli dodatnu preciznost, koja se automatski reflektuje na agregativne metode. U *eksperimentu 2.* je također potvrđeno da je adaptivno-agregativni pristup ostvario najbolje rezultate u odnosu na sve standardne i agregativne metode korištene u ovom radu, u kojem su na osnovu pondera (preciznosti) angažuju i kombinuju standardni modeli, te se i pomoćna hipoteza (*H1.2*) može smatrati potvrđenom :

- *H1.2: Na osnovu novostvorenih metapodataka moguće je preciznije preporučiti resurse kombinovanjem adaptivnih pondera određenih sistema preporuke.*

Iako su sve hipoteze u ovom radu teoretski i praktično dokazane, u radu su se koristila testna okruženja i subjektivno osmišljeni eksperimenti, tako da je ove konstatacije potrebno provjeriti u realnim okruženjima sa mnogo strožijim testovima. Primjena tehnika korištenih u ovom radu bi mogla imati vrlo široke razmjere ako se dodatno provjere mogućnosti i opravdanost istih, što će također biti predmet diskusije.

5.4 Eksperiment 4

U ovom eksperimentu testirana je brzina izvođenja algoritama standardnih metoda, koristeći skup podataka „*FitPonder.csv*“ i podskup *MovieLens* kolekcije „*ml_base.csv*“, odnosno podskup sa oznakom *d1*.

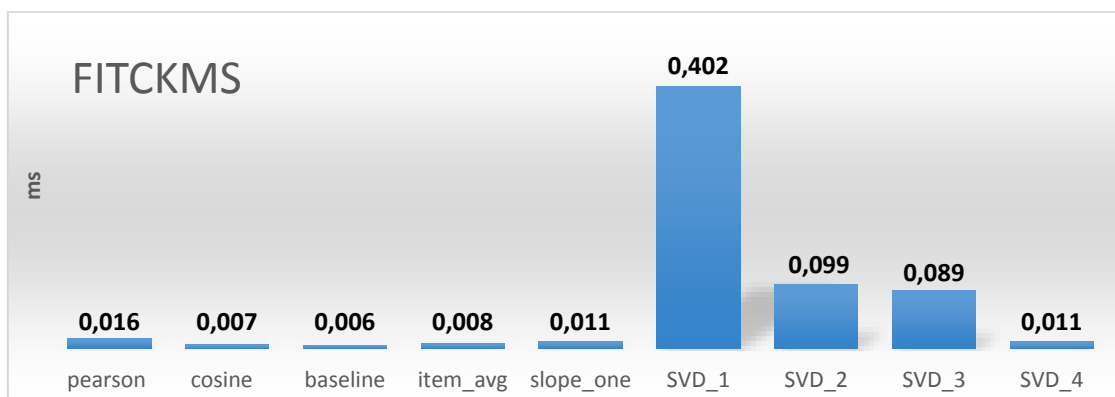
Iako nije planirano istraživanje ove problematike, rezultati ovog eksperimenta mogu postati predmet diskusije i dovesti do određenih zaključaka, jer su usko povezani sa temom ovog rada.

U prvom dijelu ovog eksperimenta testirana je brzina izvođenja standardnih metoda, koristeći skup podataka „*FitPonder.csv*“. Radi pouzdanosti rezultata test je izvršen pet puta, a finalni rezultat predstavlja srednja vrijednost rezultata testova.

<i>tip</i>	<i>metoda</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>srednja vrijednost</i>
<i>S</i>	<i>pearson</i>	0,015	0,019	0,014	0,017	0,015	0,016
<i>S</i>	<i>cosine</i>	0,007	0,006	0,007	0,006	0,007	0,007
<i>S</i>	<i>baseline</i>	0,006	0,005	0,006	0,006	0,006	0,006
<i>S</i>	<i>item_avg</i>	0,010	0,006	0,007	0,009	0,009	0,008
<i>S</i>	<i>slope_one</i>	0,011	0,010	0,012	0,012	0,010	0,011
<i>S</i>	<i>SVD_1</i>	0,363	0,348	0,411	0,414	0,475	0,402
<i>S</i>	<i>SVD_2</i>	0,104	0,091	0,089	0,100	0,110	0,099
<i>S</i>	<i>SVD_3</i>	0,091	0,089	0,094	0,084	0,088	0,089
<i>S</i>	<i>SVD_4</i>	0,009	0,010	0,009	0,012	0,013	0,011

Tabela 8. Brzina izvođenja standardnih metoda (skup "*FitPonder.csv*")

U *Tabeli 8.* kolone od 1 do 5 predstavljaju broj testa a redovi standardne metode. Čelije predstavljaju rezultate brzina izvođenja algoritama, s jedinicom mjere milisekund (*ms*). Posljednja kolona predstavlja srednju vrijednost rezultata testova, što je vizuelno prikazano na *Grafikonu 2.*



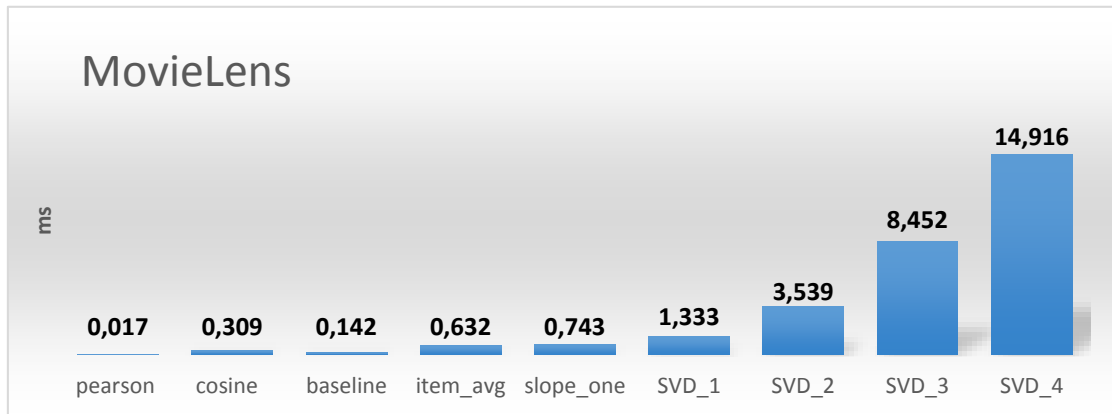
Grafikon 2. Brzina izvođenja standardnih metoda (skup "FitPonder.csv")

Na *Grafikonu 2.* primjeti se da najduže prosječno vrijeme izvođenja algoritma imaju *SVD* metode, s naglaskom na *SVD_1* metodu, za čije izvođenje je potrebno 0,402 milisekunde. Rezultati iz *Tabele 5.* pokazali su da ova metoda ima najvišu *RMSE* vrijednost, koristeći sličan skup podataka, što ujedno predstavlja i najvišu maksimalnu vrijednost standardnog odstupanja od srednje *RMSE* vrijednosti, koja je prikazana na *Grafikonu 1.* Procijenjena brzina izvođenja možda i ne predstavlja veliki problem, jer ipak se radi o 0,4 sekunde, koliko je upitna procijenjena preciznost ove metode u *eksperimentu 2.* Na kraju može se zaključiti da se ova metoda može izostaviti u agregativnim modelima, zbog najviše brzine izvođenja i najniže preciznosti. Ipak, kako su ove metode testirane na malom skupu, ovu teoriju bi valjalo provjeriti na većem skupu, koji je sličniji stvarnim sistemima koji su danas u upotrebi, što je zadatak drugog dijela ovog eksperimenta (*Tabela 9.*).

tip	metoda	d1	d2	d3	d4	d5	srednja vrijednost
S	pearson	0,046	0,012	0,008	0,008	0,012	0,017
S	cosine	0,346	0,293	0,307	0,308	0,291	0,309
S	baseline	0,189	0,134	0,133	0,123	0,133	0,142
S	item_avg	0,800	0,662	0,580	0,547	0,573	0,632
S	slope_one	0,879	0,656	0,584	1,032	0,562	0,743
S	SVD_1	2,139	1,052	1,188	1,173	1,112	1,333
S	SVD_2	4,163	3,272	3,419	3,475	3,366	3,539
S	SVD_3	8,833	8,059	8,707	8,471	8,191	8,452
S	SVD_4	15,730	14,758	14,498	14,679	14,915	14,916

Tabela 9. Brzina izvođenja standardnih metoda (MovieLens)

U drugom dijelu eksperimenta testirane su iste standardne metode ali na mnogo većem skupu. U ovom slučaju izvršen je samo jedan test nad disjunktним skupovima *MovieLens* kolekcije (*dx*). Čelije u *Tabeli 9.* predstavljaju brzine izvođenja standardnih metoda u milisekundama. Posljednja kolona, koja predstavlja srednju vrijednost predstavljena je na *Grafikonu 3.*



Grafikon 3. Brzina izvođenja standardnih metoda (MovieLens)

Kao i u prvom dijelu ovog testa, rezultati srednje vrijednosti brzine izvođenja pokazuju da su najzahtjevnije *SVD* metode. U ovom slučaju najviše vremena za predviđanje je potrebno *SVD_4* metodi, čak oko 15 sekundi. Takvo vrijeme je apsolutno neprihvatljivo pri izvođenju ove metode u realnom vremenu. Ako se u *eksperimentu 2.* posmatraju samo disjunktни podskupovi *MovieLens* kolekcije, ova metoda je imala sasvim solidne *RMSE* vrijednosti, sa prihvatljivim standardnim odstupanjima od sredine. Slične karakteristike imaju i *SVD_2* i *SVD_3* metode, sa nešto nižim vrijednostima brzine izvođenja. Metoda *SVD_1* koja se u prvom dijelu eksperimenta pokazala kao neupotrebljiva, u ovom slučaju je dala relativno prihvatljive rezultate u brzini izvođenja i preciznosti, te je moguća njena upotreba u agregativnim modelima.

Eksperiment 4. je prikazao brzine izvođenja standardnih metoda korištenih u ovom radu. Korištenje *SVD* metoda u agregativnim modelima se pokazalo kao vrlo upitno. Ove metode su vrlo korisne u adaptivno-agregativnim pristupima, zbog načina izvođenja i različitosti obrazaca u odnosu na ostale standardne modele, ali zbog svojih loših rezultata u ovom eksperimentu, mogu se smatrati neupotrebljivim. Ipak u praktičnoj upotrebi bilo

bi neminovno provesti detaljna istraživanja, koristeći stvarne podatke konkretnih sistema i eventualno pronaći alternativna rješenja za njihovo korištenje.

Na kraju ovog poglavlja bitno je istaći da su provedeni eksperimenti potvrdili sve hipoteze ovog rada, ali isto tako i da postoje mnoga otvorena i neodgovorena pitanja na ovom polju. O prednostima i nedostacima tematike ovog rada i pristupa predstavljениh u radu bit će više riječi u narednom poglavlju.

6 DIKUSIJA I ZAKLJUČAK

U ovom poglavlju će se raspravljati o interesantnim dijelovima ovog rada. Tema je veoma široka, što je dodatno otežalo istraživanje usmjereno na ciljnu problematiku. Bez dodatnog istraživanja komplementarnih i opštih tema, bilo bi skoro nemoguće donijeti valjane zaključke. Iz tog razloga, ovo poglavlje je segmentirano na razmatranje samo značajnih dijelova kompletnog rada.

Fokus će svakako biti na implikacijama i ograničenjima dobijenih rezultata. Kroz provedene eksperimente u prethodnom poglavlju, hipoteze rada su potvrđene u teoriji i praksi, ali možda je važnije razjasniti određene zaključke, nova saznanja i njihova ograničenja.

Na kraju će biti sumirani svi doprinosi vezani za temu, koji ukazuju na buduća istraživanja.

6.1 Kolaboracija i upravljanje znanjem

U naslovu rada korišten je izraz „kolaborativno web okruženje za elektronsko učenje“. Simbolički gledano, to može biti bilo koja web aplikacija, koja omogućava određeni vid kolaboracije a koristi se u svrhu e-učenja. U naslovu je također, naveden i izraz „mogućnosti primjene sistema preporuke“, što je i bila osnovna tema istraživanja. Iz perspektive praktične primjene metoda i ideja istraženih u radu, može se zaključiti da je adaptivni pristup preporuci, kao i uloga inteligentnog agenta, sasvim primjenjiva u različitim kolaborativnim okruženjima, koja sadrže komponentu upravljanja znanjem. Upravljanje znanjem je također vrlo široka tematika, vrlo aktuelna i u posljednje vrijeme često istraživana, u akademskoj zajednici.

Iako kvalitet komunikacije i dalje prvenstveno zavisi od socijalno-emocionalnih vještina koje čovjek posjeduje, era informacija i naprednih tehnoloških otkrića dala je komunikaciji jednu potpuno novu i, pokazalo se, izuzetno značajnu dimenziju. Upotreba informacionih tehnologija u kolaboraciji donijela je kao nusproizvod određen broj potpuno novih problema, ali što je daleko značajnije, uslovila je i daleko veći napredak.

Očigledna je činjenica da su ovakvi načini komunikacije višestruko premostili i smanjili udaljenosti, skratili vrijeme, povezali milione ljudi koji sarađuju i na taj način pojačali fokus na rješavanje problema, koji su sve kompleksniji.

Razvojem *FITCKM* sistema, uloga kolaborativnog web okruženje za e učenje u ovom radu je minimizirana, jer su korištena samo dva osnovna modula: *Wiki stranica* i *Pitanja i odgovori*. Za testiranje ideja i hipoteza nije bilo potrebe pristupiti razvoju kompleksnijeg sistema, jer su u eksperimentima korišteni samo podaci iz baze podataka.

Kolaboracija je danas mnogo više od toga, jer je u posljednje vrijeme razvijeno mnoštvo korisnih modula, koji praktično gledano povećavaju jednostavnost razmjene multimedijalnih informacija.

6.2 Wiki stranica

Primjena *Wiki* stranice u kolaborativnom učenju, podučavanju i istraživanju ima svoje prednosti ali i nedostatke. Realnim razmišljanjem i analizama može se doći do permanentnih zaključaka koji idu u korist primjene *Wiki* stranica, kako danas tako i u budućnosti.

Primjenom *Wiki* alata moguće je uspješno organizovati zajednički rad, čime se pozitivno utiče na kolaborativno učenje. Prije svega postiže se pozitivan uticaj na korisnika koji se nalazi u središtu procesa učenja.

Jednostavna upotreba, brzo dodavanje novih stranica, kao i izmjena postojećih sadržaja je ono što utiče na izbor *Wiki*-ja kao jednog od alata koji se često koriste u procesu kolaborativnog učenja, podučavanja i istraživanja. U ovom radu korišten je samo jedan od mnogobrojnih aspekata primjene predstavljenih *Wiki* alata, koji danas imaju veliki doprinos u procesu kolaborativnog elektronskog učenja.

6.3 Implikacije i ograničenja sistema preporuke

Na početku je bitno razjasniti opšte implikacije i ograničenja u adaptivnom pristupu, kao jednom od mogućih rješenja za realizaciju sistema preporuke.

Osnovna pretpostavka je da se moderni sistemi preporuke součavaju sa problemom latentne subjektivnosti. U većini takvih sistema, načini za modelovanje korisnika su predefinisani, na osnovu istraživanja i procjene razvojnog tima. Kako se korisnici razlikuju po svojim karakteristikama, željama, afinitetima i ponašanju pristup modelovanju korisnika ne bi trebao da bude univerzalna, već individualna stvar. Slijedeći tu ideju, modelovanje korisnika je potrebno prepustiti upravo korisnicima putem eksplicitnog ili implicitnog prikupljanja podataka. Iz perspektive sistema preporuke navedeno se može protumačiti kao korisnički izbor sistema preporuke. Kako je u ovom radu izabrano implicitno prikupljanje podataka, po istom modelu je osmišljena i ideja implementacije sistema preporuke. Takvu ideju je moguće sprovesti samo putem adaptivnih metoda. Definisana je kolekcija različitih algoritama za predviđanje, te na osnovu individualnih karakteristika korisnika omogućena je adaptacija. Adaptacija je bazirana na individualnom ponderisanju od kojeg zavisi koliko će se neki od algoritam angažovati za konkretnog korisnika. Provedeni eksperimenti su pokazali da tehnika adaptacije djeluje vrlo obećavajuće, te da se može koristiti kao čvrsta podloga za buduća istraživanja. U isto vrijeme treba se napomenuti da ova tehnika nije testirana u svim mogućim slučajevima korištenja.

Istraživanja provedena u ovom radu, navode na zaključak da bi adaptivni modeli preporuke ostvariti svoj puni potencijal, ako se nalaze u okruženju raznovrsnih algoritama koji koriste različite obrasce u otkrivanju veza između dostupnih podataka, kako bi zaključili koliko je određeni resurs relevantan za individualnog korisnika.

Relevantnost resursa, način i područje pretraživanja informacija je još jedno polje u kojem se javlja problem individualnosti. Postoji mnogo informacija koje određeni resurs čine potencijalno relevantnim. Te informacije, koje se mogu nazvati atributima, predstavljaju ulazne podatke za algoritme, koji mogu biti bazirani na broju atributa, kao što su informacija o vremenu, geografska lokacija, tema, ključne riječi, oznake, rezultati statističkih analiza i sl. Nije teško pretpostaviti da će različiti algoritmi imati različitu preciznost u određivanju relevantnosti za određeni resurs. Ovaj problem bi također bilo moguće riješiti upotrebom adaptivnog pristupa baziranog na karakteristikama individualnog resursa.

Korištenje greška-modela ima veoma važnu ulogu u adaptivnom sistemu preporuke, koji je sastavljen iz dva sloja. Prvi sloj je zadužen za standardna predviđanja rejtinga, dok drugi, adaptivni sloj ima zadatak da na osnovu greška-modela predvidi adaptivne težine uposlenosti pojedenih standardnih modela preporuke. Adaptivni sloj, za ovo predviđanje koristi također standardne modele preporuke, što se može smatrati kao prednost ovog pristupa, jer su ovi modeli već odavno u upotrebi, testirani su na različitim sistemima i njihov način rada je vrlo dobro poznat u akademskoj zajednici.

Međutim, agregativni pristup preporuci osim spomenutih potencijalnih prednosti ima nekoliko vrlo važnih ograničenja kao što su kompleksnost, izbor podataka, brzina izvođenja, mjere preciznosti i generalni problemi sistema preporuke.

6.4 Kompleksnost adaptivnih sistema preporuke

Činjenica da su adaptivni sistemi preporuke mnogo kompleksniji od standardnih, dovodi do pitanja da li su prednosti adaptacije vrijedne u odnosu na dodatnu složenost. To zavisi od više faktora. Jedan su sigurno potrebe okruženja u kojem se ovi sistemi koriste. Ako je riječ o kolaborativnim okruženjima za upravljanje znanjem, razmatranim u ovom radu, implementacija adaptivnog pristupa bi itekako imala smisla, jer tu postoji potreba za posmatranjem korisnika kao centralne tačke u sistemu. Korisnici bi imali različite važnosti i adaptivni sloj bi omogućio potrebnu individualnost i preciznost. Ako bi se razmatrala upotreba ovog pristupa u jednostavnim i zatvorenim sistemima, ona apsolutno ne bi imala smisla, ali na sreću takvi sistemi su sve manje u upotrebi i nemaju svijetlu budućnost.

S druge strane, u kontekstu kompleksnosti, korištenje jednostavnih sistema preporuke koji posmatraju slične obrasce u podacima, standardne agregativne metode bi bile mnogo prihvatljivije. Potvrđivanje hipoteza i eksperimentalno postignuta poboljšanja u ovom radu sigurno nisu dovoljna da se ovakav pristup proglasi nadmoćnijim u odnosu na druge bez rezerve. Iako su eksperimentalni rezultati prikazali veliki potencijal adaptivnog pristupa, bez uspjeha u „stvarnom svijetu“ neosnovano bi bilo zaključiti da je ova tehnika posebna ili bolja od bilo kojeg običnog standardnog sistema preporuke.

Pojam kompleksnosti obično prate negativne konotacije i prirodan izbor pri razvoju sistema bi bili standardni pristupi sistemu preporuke. Ideja ovog rada se ne slaže sa tom činjenicom. Preopterećenje informacijama je vrlo nijansiran problem, i jedino rješenje koje ima perspektivu, je inteligentni adaptivni pristup. Ipak realnost je malo drugačija, većina današnjih okruženja često teži ka jednostavnosti, tako da standardni sistemi preporuke još uvijek predstavljaju optimalno rješenje. Ova previranja bi se mogla opisati kao vječna borba između složenosti i potrebne preciznosti.

Realno se nadati da će u skorije vrijeme razvoj virtualnih kolaboracijskih sistema doživjeti određena poboljšanja i da će u moru različitih scenarija, adaptivni pristupi opisani u ovom radu, imati svoju ulogu.

6.5 Vremenska ograničenja

U *eksperimentu 4.* je procijenjeno vrijeme izvođenja standardnih modela preporuke i rezultati upućuju na određene probleme. Ti problemi se prenose i na agregativne metode koje za svoje izvođenje koriste standardne sisteme preporuke. Kod pojedinih standardnih modela primjećen je problem neprihvatljivog trajanja izvođenja u realnom vremenu.

Iz ove perspektive postoji nekoliko mogućih rješenja. Jedno od njih je vrlo rigorozno i podrazumijeva isključivanje ovih metoda iz agregativnih pristupa, a drugo je optimizacija ovih metoda u kontekstu vremenskog izvođenja. Također, potrebno je ispitati mogućnost i „*offline*“ izvođenja.

Ipak, za rješavanje ovog problema potrebno je provesti dodatna istraživanja i na osnovu njih donijeti odgovarajuće zaključke.

6.6 Mjere evaluacije

Za ocjenjivanje adaptivnog pristupa izabrani su skupovi podataka i evaluacijske mjere koji se najčešće koriste za testiranje sistema preporuke. Izborom tradicionalnih skupova i mjera omogućena je komparacija sa drugim istraživanjima na ovom polju. U testiranje su dodatno uvršteni lokalni skupovi podataka iz testne aplikacije *FITCKMS*, čime su omogućena dodatna istraživanja i otvorena nova pitanja.

6.7 Inteligentni programski agenti

Iako su rezultati adaptivnog pristupa u početnim eksperimentima su bili vrlo obećavajući, potrebno je naglasiti činjenicu da su testovi obavljeni korištenjem svega nekoliko testnih skupova podataka. Testiranje u skupovima iz realnih okruženja bi sigurno dovelo do novih zaključaka. Ipak, ovaj rad se može još uvijek smatrati kao pionirsko istraživanje novih tehnika, koje su dokazano korisne samo u nekolicini scenarija.

Nakon pozitivnih rezultata primjenom adaptivnog pristupa istraživanje u ovom radu je dobilo novu dimenziju. Osim preopterećenja informacijama, jedan od najvećih problema u današnjim sistemima predstavlja kvalitet i obrada podataka. Konkretno, ako je riječ o eksplicitnim ocjenama koje često nastaju „u prolazu“ korisnika kroz sistem, dovodi se u pitanje bilo kakva upotreba sistema preporuke. S druge strane, trenutni pristupi se baziraju uglavnom na statičnim vrijednostima ocjena na određenoj skali. Za prvi problem rješenja su uglavnom nemoguća, bez dodatnog uznemiravanja korisnika, pa čak ni tada se ne bi moglo sa sigurnošću reći da je postignut bilo kakav napredak. Za drugi problem određena poboljšanja su vrlo izvjesna.

Kao što je poznato iz oblasti statistike, na ordinalnoj skali značaj svakog koraka nije nužno jednak. Odnosno, razlika je nepoznata između pojedinih jedinica skale, ali ni ona nije jednaka. Na primjer, ako se koristi skala od 1 do 5, razlika između 2 i 3 možda nije značajna kao razlika između 4 i 5. Ova opaska predstavlja ograničenje većine današnjih sistema u smislu kvaliteta ulaznih podataka u izabrane algoritme sistema preporuke. Upravo iz tog razloga osmišljen je *eksperiment 3*. koji je na tragu poboljšanja ovog problema.

U cilju podizanja kvaliteta ulaznih podataka osmišljen je i integrisan inteligentni programski agent u testni *FITCKM* sistem. Njegova uloga je vrlo jednostavna i odnosi se na individualnu ulogu svih korisnika, resursa i veza između njih u sistemu. Drugim riječima, ako sistem posmatramo kao homogeni prostor, pomijeranje jedne čestice će uzrokovati odgovor prostora pomijeranjem druge ili grupe čestica. Konkretno, ako je riječ o korisnicima, resursima i njihovim vezama, odnosno ocjenama, bilo kakva promjena će uzrokovati neku drugu promjenu.

Rezultati ovog pristupa su vrlo obećavajući, uzimajući u obzir jednostavnu simulaciju inteligentnog programskog agenta i sve potencijalne rizike i probleme. Ozbiljnijim pristupom istraživanju ove oblasti sigurno bi bilo mjesta za mnogo značajnija poboljšanja, posebno na temu adaptivnog pristupa preporuci, odnosno dinamičkoj izgradnji modela korisnika. Također, vrlo je izvjestan i značajan pomak u načinu tretiranja korisnika u sistemu, povećavanju nivoa korisničke individualnosti, kao i rješavanju problematike koja prati sisteme preporuke.

Ipak, kao što je slučaj i u prethodnim razmatranjima, za izvođenje konačnih zaključaka neminovno je svaku ideju provjeriti praktično sa različitim aspektima korištenja.

6.8 Pravci budućih istraživanja

Kroz ovaj rad prikazane su osnovne održivosti adaptivno-agregativnih sistema preporuke i simulacije inteligentnog agenta u ulozi povećanja kvaliteta ulaznih podataka u algoritme tih sistema. Bilo kakav pomak u ovoj oblasti (na bolje ili na gore) bi predstavljao veliki uspjeh za buduća istraživanja. Ipak, primjene i mogućnosti razmatranih pristupa, mogu imati mnogo šire okvire nego što je u ovom radu prikazano.

U toku osmišljavanja teme i postavljanja hipoteza ovog rada, ideja za implementaciju kolaborativnog web okruženja, sa komponentom upravljanja znanjem, je bila mnogo šira, ali i konkretnija. Planiran je razvoj potpuno nove virtualne akademske zajednice, u kojoj bi bile integrisane sve značajne informacije i podaci sistema najpopularnijih visokoškolskih ustanova u regionu. U takvom okruženju bilo bi izuzetno zanimljivo ispitati mogućnost upotrebe adaptivnog sistema preporuke a pogotovo upotrebu inteligentnih programskih agenata i njihovu ulogu u skupu različitih sistema. Ideja se ne čini nemogućom, ali zahtijeva dosta istraživanja i angažman velikog broja stručnjaka iz različitih IT domena. Zbog svoje kompleksnosti, i ova ideja je ostavljena za buduća istraživanja.

Još jedno vrlo interesantno polje za ovu tematiku je kvantitativno izvođenje prilagođenog pretraživanja. Vrlo izvjesna ideja je da bi se adaptivni pristupi preporuci mogli dati određena poboljšanja u adaptivnom pretraživanju, koje se izvodi putem ključnih riječi,

oznaka, log-ova, broja klikova, fraza i sl. To zahtijeva veoma široka istraživanja iz ove oblasti, u smislu preciznosti, trajanja i rada sa ogromnim brojem podataka. Takvi eksperimenti izlaze iz okvira ovog rada, te su također ostavljeni za buduća istraživanja.

Za adaptivni pristup preporuci korištene su standardne metode preporuke. Naravno, ni na tom planu nisu iskorištene sve mogućnosti te bi valjalo istražiti kombinaciju sa drugim metodama. Uopšteno, ideja adaptivne preporuke sigurno predstavlja pravce budućeg istraživanja, bez obzira na pristupe koji se koriste. U ovom radu je istražen samo jedan od njih. Trenutno postoji mnogo raznovrsnih domena gdje bi se ovakvi pristupi mogli testirati.

U pravcu poboljšanja ulaznih podataka u algoritme, predstavljena je uloga inteligentnog programskog agenta, usmjerena samo na određene jednostavne podatke. Ova uloga bi se mogla dodatno proširiti na ostale značajne podatke, važne za izgradnju modela korisnika u smjeru individualizacije. U ovom radu, inteligentni programski agent je dokazano stvorio novo znanje i omogućio sistemu korištenje drugačije dimenzije podataka. Kroz nove uloge i dinamičke reakcije na određene događaje, ovo znanje bi se dodatno moglo proširiti i kombinovati u zaključivanju, odnosno podizanju „svjesnosti“ sistema. Također, i ove ideje su ostavljene čitaocima na razmišljanje.

6.9 Zaključak

Problem latentne subjektivnosti je često zanemaren i sprečava standardnim sistemima preporuke postizanje svog punog potencijala. Osnovna ideja ovih sistema je predviđanje nepoznate ocjene za resurse, na osnovu obrazaca koji koriste ocjene ostalih resursa. Njihov individualni nastup je različit od sistema do sistema i zavisi od korisnika i resursa tog sistema. Moderni pristupi agregativnim sistemima preporuke koriste različite obrasce, većinom na generalizovanom nivou, jer se korisnici i resursi tretiraju na isti način, što narušava osnove subjektivnosti. Pristupi koji koriste prosječne ili predefinisane težine (pondere) modela, pretpostavljaju da je najbolji prosjek ujedno i najbolji rezultat predviđanja za sve pojedinačne korisnike ili resurse. Latentna subjektivnost predstavlja sveobuhvatan problem koji često može biti otkriven u različitim tehnikama mašinskog učenja.

Adaptivni sistemi preporuke mogu predstavljati određene pomake u rješavanju ovog problema. Rezultati eksperimentima dokazuju da adaptivno-agregativni pristup, koji kombinuje predviđanja na osnovu procijenjene tačnosti (pondera), nadmašuje standardne sisteme i jednostavne agregativne pristupe. Dodatna vrijednost ovog pristupa je korištenje standardnih sistema preporuke u procjeni preciznosti, odnosno određivanju pondera. Problem latentne subjektivnosti se u adaptivnom pristupu rješava se stvaranjem boljih rezultata, odnosno boljih kombinacija različitih prediktora za individualnog korisnika. Ako postoji potreba za kombinacijom više algoritama čiji je rezultat skup predviđanja baziran na resursima, adaptivni pristup može procijeniti greške i adaptirati preciznost predviđanja za individualni resurs. To dovodi do zaključka da ovakav pristup može biti primjenjiv i izvan oblasti sistema preporuke, te da postoji mogućnost postizanja mnogo boljih rezultata nego što je to bio slučaj u eksperimentima ovog rada. Zanimljivo bi bilo istraživanje potencijala adaptivnog pristupa u društvenim mrežama ili sistemima koji predviđaju relevantnost veoma različitih resursa.

Uloga inteligentnog programskog agenta je dala dodatni doprinos adaptivnom pristupu preporuke i rješavanju problema latentne subjektivnosti. U *eksperimentu 3*, dokazano je da kvalitet podataka pozitivno utiče na kvalitet preporuke. Dublja istraživanja na polju adaptacije ulaznih podataka ne samo da bi poboljšala rad postojećih algoritama i efikasnost predviđanja, nego bi donekle riješila i problem preopterećenja informacijama. Možda bi ovaj pristup nabolje objasnila izreka: „*kvalitet kreira kvantitet*“.

Na kraju ovog rada treba postaviti još samo jedno pitanje:

„*Koji sistem preporuke primijeniti u kolaborativnom web okruženju za elektronsko učenje*“

?

7 LITERATURA

- [1] P. Brusilovsky, *Adaptive navigation support: From Adaptive Hypermedia to the Adaptive Web and Beyond*, *PsychNology Journal*, Vol. 3, 2004, pp. 7-23.
- [2] A. Paramythis, *Adaptive Support for Collaborative Learning with IMS Learning Design: Are We There Yet? Proceedings of the Workshop on Adaptive Collaboration Support, held in conjunction with the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Hannover, Germany, 2008*, pp. 17-29.
- [3] Mamcenko J., Beleviciute I. (2007), *Data Mining for Knowledge Management in Technology Enhanced Learning. In proceedings of the 6th WSEAS International Conference on Applications of Electrical Engineering, Istanbul, Turkey, May 27-29, 2007*.
- [4] Giboin A., Dieng R., Karsenty L., De Michelis G., *Designing Cooperative Systems, The Use of Theories and Models, Proc. Of 5-th Int. Conf. on the Design of Coop. Syst. (COOP'20000), IOS Perss 2000*.
- [5] Osher D.M., *Creating Comprehensive and Collaborative Systems, Journal of Child & Family Studies*, vol. 11, no. 1, p. 91-99, 2002.
- [6] Camarinha-Matos L.M. and H. Afsarmanesh, *Collaborative Networks: Reference Modeling. Springer US, 2008, ch. Collaboration forms*.
- [7] Nițchi S.I., A.Mihăilă, C. Mihăilă, *Some Remerks on Knowlegdege Management in Virtual Organizations, 1st CEE Symposium on Business Informatics, Viena, februarie 2009, Oestriscche Computer Gessellschaft, Austrian Computer Society, Wi09, ISBN 987-3, 85403, 242-7, p. 207-218*.
- [8] Fuks H., et al. *The 3C Collaboration Model*, 2008, pp. 637-644.
- [9] Grosz B., 1996, *AAAI-94 President Address Collaborative Systems, AAAAI Magazine, summer, 1996, p. 67-85*.
- [10] Alavi, M. and A. Tiwana, A. *Knowledge integration in virtual teams: The potential role of KMS. Journal of the American Society for Information Science and Technology*, 53(12), 2002, 1029-1037.
- [11] Turban E., J.A. Aronson, L. Ting-Peng, *Decisions Support Systems and Intelligent Systems, Seventh Edition, Prentice Hall, 2005*.
- [12] Thagard P., 1997, *Collaborative Knowledge*, Dostupno na:
<http://cogsci.uwaterloo.ca/Articles/Pages/Collab.html>. [04.10.2014.].
- [13] Muntean, M., Tărnăveanu, A. (2009), *Information Technology & Organizational Knowledge Management, Proceedings of the 13th WSEAS International Conference on COMPUTERS, WSEAS Press, ISBN 979- 960-474-099-4, pp. 335-339*
- [14] Scaffert S., *IkeWiki: A SemanticWiki for Collaborative Knowledge Management, Salzburg Research Forschungsgesellschaft/Salzburg New Media Lab*.

- [15] Nunamaker J.F., Romano N.C., Bricks R.O., *A Framework for Collaboration and Knowledge Management, Proceedings of the 34th Hawaii International Conference on System Sciences – 2001.*
- [16] Andrade J., Ares J., García R., Rodríguez S., Suárez S., *A Knowledge-Based System for Knowledge Management Capability Assessment Model Evaluation, WSEAS Transactions on Computers, Issue 5, Volume 9, 2010, pp. 506 – 515.*
- [17] Paladini E.P., de Carvalho F.G., *New Standards for Competitive Distinctions: A Practical Model, WSEAS Transactions on Computers, Issue 2, Volume 10, 2011, pp. 21 – 50.*
- [18] Tučková Z., Strouhal J., *Knowledge-Intensive Services: New Leader of Production Stages?, WSEAS Transactions on Systems, Issue 4, Volume 9, 2010, pp. 432 -441.*
- [19] Akoumiakakis D., *Distributed Knowledge Management in Virtual Organizations: the ‘Social’ Experience Factory, The Electronic Journal of Knowledge Management, Volume 6, Issue 1, 2008, pp.13-32.*
- [20] Jones P.M., *Collaborative Knowledge Management, Social Networks and Organizational Learning, NASA Ames Research Center, Human Factors Research and Technology Division*
- [21] Tomblin M.S., *Group and Organizational Learning Effects from Multiparticipant DSS Usage, in BURSTEIN F and C.W.HOLSAPPLE (Eds.), Handbook on Decision Support Systems, 1, Basic Themes, Springer, 2008, p. 813-133.*
- [22] Garvin D.A., *Building a learning organization, Harvard Business Review, 1993, July/August.*
- [23] Bosch-Sijtsema P.M., *Knowledge Management in Virtual Organizations: Interorganizational and interproject knowledge transfer, Organizational Knowledge, Learning and Capabilities conference 2002.*
<http://www2.warwick.ac.uk/fac/soc/wbs/conf/olkc/archive/oklc3/papers/id191.pdf>
- [24] Nonaka I., *A dynamic Theory of Organizational Knowledge Creation, Organization Science, Vol.5, No.1, February.*
- [25] Zakaria Suliman Zubi. 2008. *Knowledge discovery query language (KDQL). In Proceedings of the 12th WSEAS international conference on Computers (ICCOMP'08), N. E. Mastorakis, V. Mladenov, Z. Bojkovic, D. Simian, S. Kartalopoulos, A. Varonides, C. Udriste, E. Kindler, S. Narayanan, J. Lloret Mauri, H. Parsiani, and Ka Lok Man (Eds.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 497-519*
- [26] Berryman R., *Knowledge management in virtual organizations: A study of a best practices knowledge transfer model, Dissertation Prepared for the Degree of PhD., University of North Texas, May 2005.*
- [27] Massey A.P., *Collaborative Technologies, in BURSTEIN F and C.W.HOLSAPPLE (Eds.), Handbook on Decision Support Systems, 1, Basic Themes, Springer, 2008, p. 341-354.*
- [28] Dieng-Kuntz R. et al., *Methodes et outils pour la gestion des connaissances, Une approche pluridisciplinaire du Knowledge Management, Dunod, Paris, 2-nd Edition, 2001.*

- [29] P. Brusilovsky and N. Henze, *Open Corpus Adaptive Educational Hypermedia*, In *The Adaptive Web*, ser. *Lecture Notes in Computer Science*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4321, ch. 22, pp. 671-696.
- [30] P. Anderson, *What is Web 2.0? Ideas, technologies and implications for education*, JISC Technology and Standards Watch, 2007.
- [31] J. G. Recuenco and D. Bueno, *Balanced Recommenders: A hybrid approach to improve and extend the functionality of traditional Recommenders*. In *Proceedings of International Workshop on Adaptation and Personalization for Web 2.0 (AP-WEB 2.0 2009)*, Trento, Italy, 2009, pp. 88-97.
- [32] E. García, C. Romero, S. Ventura and C. de Castro, *An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering*, *User Model User-Adap Inter* (2009) 19:99–132.
- [33] P. Resnick and H. Varian, *Recommender Systems*. *Communications of the ACM (CACM)*, 40 (3), 56-58, 1997.
- [34] V. Radovic, M. Prodic, a. Trifunovic, A. Stojicic, (12.05.2013.), *Primena Wikija u kolaborativnom učenju i istraživanju* [Online]. Dostupno na: <http://skolskizadatak.wordpress.com/>, [26.08.2014.].
- [35] Antil, L., J. Jenkins, S. Wayne, and P. Vadasy. "Cooperative Learning: Prevalence, Conceptualizations, and the Relationship between Research and Practice." *AMERICAN EDUCATIONAL RESEARCH JOURNAL* 35, no.3 (1997): 419-454.
- [36] Jane E. Brindley and Christine Walti, University of Oldenburg, Germany, Lisa M. Blaschke, University of Maryland University College, USA „The International review of research in open and distance learning“, Dostupno na: <http://www.irrodl.org/index.php/irrodl/article/view/675/1271>, [26.08.2014.].
- [37] MediaWiki, *Slobodna enciklopedija Wikipedia*, (26.06.2014.), Dostupno na: <http://sh.Wikipedia.org/Wiki/MediaWiki>, [26.08.2014.].
- [38] DokuWiki, (26.04.2014.), Dostupno na: <https://www.dokuWiki.org/dokuWiki>, [26.08.2014.].
- [39] TermWiki, *Slobodna enciklopedija Wikipedia*, (21.11.2013.), Dostupno na: <http://en.Wikipedia.org/Wiki/TermWiki>, [26.08.2014.].
- [40] WikiEducator, *Slobodna enciklopedija Wikipedia*, (29.07.2014.), Dostupno na: <http://en.Wikipedia.org/Wiki/WikiEducator>, [26.08.2014.].
- [41] PmWiki, (03.02.2013.), Dostupno na: <http://www.pmWiki.org/>, [26.08.2014.].
- [42] FosWiki, (22.05.2014.), Dostupno na: <http://fosWiki.org/>, [26.08.2014.].
- [43] MathWiki, Dostupno na: <http://mathWiki.ucdavis.edu/>, [26.08.2014.].
- [44] Bradshaw, J.: *Software Agents*. MIT Press, Cambridge (USA) (1997).
- [45] Milicic, T., Podobnik, V., Petric, A., Jezic, G.: *The CrocodileAgent: A Software Agent for SCM Procurement Gaming*. *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science* 5027, 865-875 (2008).

- [46] Podobnik, V., Petric, A., Jezic, G.: *An Agent-Based Solution for Dynamic Supply Chain Management. Journal of Universal Computer Science* 14(7), 1080-1104 (2008).
- [47] Petric, A., Podobnik, V., Grguric, A., Zemljic, M.: *Designing an Effective E-Market: An Overview of the CAT Agent. U: Proceedings of the 2008 AAAI Workshop on Trading Agent Design and Analysis (TADA'08), Chicago (SAD), pp. 62-65 (2008).*
- [48] Sardinha, A., Benisch, M., Sadeh, N., Ravichandran, R., Podobnik, V., Stan, M.: *The 2007 Procurement Challenge: A Competition to Evaluate Mixed Procurement Strategies. Electronic Commerce Research and Applications* 8(2), 106-114 (2009).
- [49] Ceric, V., Varga, M.: *Informacijska tehnologija u poslovanju. Element, Zagreb (2004)*
- [50] Dumic, G., Podobnik, V., Jezic, G., Trzec, K., Petric, A.: *An Agent-Based Optimization of Service Fulfillment in Next-Generation Telecommunication Systems. U: Proceedings of the 9th International Conference on Telecommunications (ConTEL'07), Zagreb (Hrvatska), pp. 57-63 (2007).*
- [51] Chorafas, D.: *Agent Technology Handbook. McGraw-Hill, New York (1998).*
- [52] Cockayne, W., Zyda, M.: *Mobile Agents. Manning Publications, Greenwich (SAD) (1998)*
- [53] Brenner, W., Zarnekow, R., Wittig, H.: *Intelligent Software Agents Foundations and Applications. Springer-Verlag, Berlin (1998).*
- [54] Nwana, H.: *Software Agents: An Overview. Knowledge Engineering Review* 11(3), 205-244 (1996).
- [55] Fasli, M.: *Agent Technology For E-Commerce. John Wiley & Sons, Chichester (2007)*
- [56] *Inteligentni agenti, Slobodna enciklopedija Wikipedia, (14.02.2013.), Dostupno na: http://hr.Wikipedia.org/Wiki/Inteligentni_agenti#CITEREFRussellNorvig2003, , [26.08.2014.].*
- [57] R. Bell, Y. Koren, and C. Volinsky. *The BellKor solution to the Netflix prize. KorBell Team's Report to Netflix, 2007.*
- [58] A. Halevy and P. Norvig. *The unreasonable effectiveness of data. Intelligent Systems, IEEE, 24(2):8-12, Mar. 2009.*
- [59] M. Banko and E. Brill. *Mitigating the paucity-of-data problem: Exploring the effect of training corpus size on classifier performance for natural language processing. In Proceedings of the first international conference on Human language technology research, pages 1-5. Association for Computational Linguistics, 2001.*
- [60] G. Adomavicius and A. Tuzhilin, *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE TKDE: IEEE Transactions on Knowledge and Data Engineering, 17, 2005. pp. 734-749.*
- [61] I. Cantador, A. Bellogín and P. Castells, *A Multilayer Ontology-based Hybrid Recommendation Model, AI Commun. 21, 2-3, 2008, pp. 203-210* [10] G. Adomavicius, Y. Kwon, *New Recommendation Techniques for Multicriteria Rating Systems, IEEE Intelligent Systems, vol. 22, no. 3, pp. 48-55, May/June 2007.*

- [62] G. Adomavicius, Y. Kwon, *New Recommendation Techniques for Multicriteria Rating Systems*, *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 48-55, May/June 2007.
- [63] N. Tintarev and J. Masthoff, *A Survey of Explanations in Recommender Systems*, In *Proceedings of the 2007 ACM Conference on Recommender Systems (Minneapolis, MN, USA, 2007)*. RecSys '07. ACM, New York, NY, pp. 203-206.
- [64] W. Woerndl, A. Helming, and V. Prinz, *Experiences from Implementing Collaborative Filtering in a Web 2.0 Application*. In *Proceedings of International Workshop on Adaptation and Personalization for Web 2.0 (AP-WEB 2.0 2009)*, Trento, Italy, 2009, pp. 120-129.
- [65] B. Mobasher, *Data mining for web personalization*. In *The Adaptive Web*, ser. *Lecture Notes in Computer Science*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4321, ch. 3, pp. 90-135.
- [66] O. Bjørkøy. *User Modeling on The Web: An Exploratory Review*, 2010.
- [67] J. a. Aslam and M. Montague. *Models for metasearch*. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '01*, pages 276–284, 2001.
- [68] D. Carmel, N. Zwerdling, I. Guy, S. Ofek-Koifman, N. Har'el, I. Ronen, E. Uziel, S. Yogev, and S. Chernov. *Personalized social search based on the user's social network*. *Proceeding of the 18th ACM conference on Information and knowledge management – CIKM '09*, page 1227, 2009.
- [69] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. *Combining Content-based and collaborative filters in an online newspaper*. In *Proceedings of ACM SIGIR Workshop on Recommender Systems*, number June, pages 60–64. Citeseer, 1999.
- [70] B. Sergey and P. Lawrence. *The anatomy of a large-scale hypertextual web search engine*. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.
- [71] O. Bjørkøy. „Adaptive Aggregation of Recommender Systems“ Norwegian University of Science and Technology Trondheim, Norway, October 3rd, 2011.
- [72] L. Breiman. *Bagging predictors*. *Machine learning*, 24(2):123–140, 1996.
- [73] R. Burke. *Hybrid web recommender systems*. In *The adaptive web*, pages 377–408. Springer-Verlag, 2007.
- [74] T. Dietterich. *Ensemble methods in machine learning*. *Multiple classifier systems*, pages 1–15, 2000.
- [75] R. Polikar. *Ensemble based systems in decision making*. *Circuits and Systems Magazine, IEEE*, 6(3):21–45, 2006.
- [76] A. Jovic, „Postupci dubinske analize podataka“, *Fakultet elektrotehnike i računarstva, Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave*, Dostupno na: http://www.fer.unizg.hr/_download/repository/Jovic,KDI.pdf, [26.08.2014.].

- [77] F. Liu, C. Yu, and W. Meng. *Personalized web search by mapping user queries to categories. Proceedings of the eleventh international conference on Information and knowledge management - CIKM '02*, page 558, 2002.
- [78] X. Shen, B. Tan, and C. Zhai. *Implicit user modeling for personalized search. Proceedings of the 14th ACM international conference on Information and knowledge management - CIKM '05*, page 824, 2005.
- [79] M. Speretta and S. Gauch. *Personalized Search Based on User Search Histories. The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pages 622–628, 2000.
- [80] K. Sugiyama, K. Hatano, and M. Yoshikawa. *Adaptive web search based on user profile constructed without any effort from users. Proceedings of the 13th conference on World Wide Web - WWW '04*, page 675, 2004.
- [81] B. Mirza and B. Keller. *Studying recommendation algorithms by graph analysis. Journal of Intelligent Information*, 2003.
- [82] M. Pazzani and D. Billsus. *Content-based recommendation systems. In The adaptive web*, pages 325–341. Springer-Verlag, 2007.
- [83] J. Schafer, D. Frankowski, J. Herlocker, and S. Sen. *Collaborative filtering recommender systems. The adaptive web*, pages 291–324, 2007.
- [84] T. Segaran. *Programming collective intelligence. O'Reilly Books*, 1st edition, 2007.
- [85] D. Lemire and A. Maclachlan. *Slope one predictors for online rating-based collaborative filtering. Society for Industrial Mathematics*, 2005.
- [86] D. Billsus and M. Pazzani. *Learning collaborative information filters. In Proceedings of the Fifteenth International Conference on Machine Learning*, volume 54, page 48, 1998.
- [87] J. Sun, H. Zeng, H. Liu, and Y. Lu. *CubeSVD: a novel approach to personalized Web search. on World Wide Web*, pages 382–390, 2005.
- [88] T. Segaran. *Programming collective intelligence. O'Reilly Books*, 1st edition, 2007.
- [89] R. Bell, Y. Koren, and C. Volinsky. *Modeling relationships at multiple scales to improve accuracy of large recommender systems. Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, page 95, 2007.
- [90] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. *User profiles for personalized information access. In The Adaptive Web*, ser. *Lecture Notes in Computer Science*. P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4321, ch. 2, pp. 54-89.
- [91] H. Drachler, H.G.K. Hummel, and R. Koper. *Recommendations for learners are different: Applying memory-based recommender system techniques to lifelong learning. In Proceedings of the Workshop on Social Information Retrieval in Technology Enhanced Learning (SIRTEL 2007)*, Crete, Greece, 2007. pp. 18-26.
- [92] M. Alshamri and K. Bharadwaj. *Fuzzy-genetic approach to recommender systems based on a novel hybrid user model. Expert Systems with Applications*, 35(3):1386–1399, Oct. 2008.

- [93] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, Jan. 2004.
- [94] A. Labus. „Pregled naučno-istraživačke oblasti“. Fakultet organizacionih nauka, Univerzitet u Beogradu,. Dostupno na:
https://www.google.ba/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0CCUQFjAB&url=http%3A%2F%2Fwww.elab.rs%2Fwp-content%2Fuploads%2F2014%2F03%2FNau%25C4%258Dno-istra%25C5%25BEiva%25C4%258Dki-rad_Aleksandra-Labus.doc&ei=_qhQVLKWDM7XPea2gPgN&usg=AFQjCNGamGjUoe9L1TO5U5xWNAoC--xpTA&sig2=FuKaXaYHlU0JrlttgUls&bvm=bv.78597519,d.ZWU. [29.10.2014.].
- [95] Dostupno na: https://lucene.apache.org/core/3_5_0/api/contrib-highlighter/org/apache/lucene/search/highlight/Highlighter.html., [31.10.2014.].
- [96] Dostupno na: <https://lucene.apache.org/>, [29.10.2014.].
- [97] Dostupno na: <https://www.eclipse.org/downloads/>, [29.10.2014.].
- [98] Dostupno na: <http://mahout.apache.org/>, [31.10.2014.].
- [99] M. Jennex and D. Croasdell. Knowledge Management, Organizational Memory & Organizational Learning Cluster. *Proceedings of the 37th Hawaii International Conference on System Sciences – 2004*, Dostupno na:
<http://www.computer.org/csdl/proceedings/hicss/2004/2056/08/205680234.pdf> [31.10.2014.].
- [100] Dostupno na:
<http://groups.engin.umd.umich.edu/CIS/course.des/cis479/projects/FISA.html#intro>,
[31.10.2014.].
- [101] Dostupno na: <http://www.agentbuilder.com/Documentation/whyAgents.html>. [31.10.2014.].
- [102] Dostupno na:
http://groups.engin.umd.umich.edu/CIS/course.des/cis479/projects/agent/Intelligent_agent.html.
[31.10.2014.].
- [103] Stuart J. Russell and Peter Norvig. *Artificial Intelligence A Modern Approach*, Prentice Hall, Englewood Cliffs. New Jersey 07632, 1995.
- [104] Casterfranchi, C (1995). Guarantees for autonomy in cognitive agent architecture. In Woolridge, M. and Jennings, N. R., ed., *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, page 56-70. Springer-Verlag: Heidelberg, Germany.
- [105] Genesereth, M. R. and Ketchpel, S. P. (1994). *Software Agents. Communications of the ACM*, 37(7): page 48-53.
- [106] Analogous to the "sleep" state in a UNIX system, where a process that has no further tasks to be done, or has to wait for another process to finish, goes into a sleep state until another process wakes it up again.
- [107] Shoham, Y. Agent-oriented programming. *Artificial Intelligence*, 60(1): page 51-92, 1993.

- [108] Bates, J. *The role of emotion in believable agents. Communications of the ACM*, 37(7): page 122-125, 1994.
- [109] Maes, P. *Agents that reduce work and information overload. Communications of the ACM*, 37(7): page 31-40, 1994.
- [110] White, J. E. *Telescript technology: The foundation for the electronic marketplace. White paper, General Magic Inc., 1994.*
- [111] Rosenschein, J. S. and Genesereth, M. R. *Deals among rational agents. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85), page 91-99, Los Angeles, United States, 1994.*
- [112] Galliers, J. R. *A Theoretical Framework for Computer Models of Cooperative Dialogue. Acknowledging Multi-Agent Conflict. PhD thesis. page 49-54, Open University, Great Britain, 1994.*
- [113] Eichmann, D. *Ethical Web Agents. Proceedings of the Second International World-Wide Web Conference. Chicago, United States, October 1994.*
- [114] M. Holenko Dlab. *Primjena sustava za preporuke kod hipermedijske programske potpore za učenje. Odjel za informatiku Sveučilišta u Rijeci. Dostupno na:*
https://www.fer.hr/_download/repository/Holenko_Dlab_kvalifikacijski.pdf. [03.11.2014.].
- [115] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor. *Recommender Systems Handbook. Springer New York Dordrecht Heidelberg London. Springer Science+Business Media, LLC 2011. Dostupno na:* http://www.cs.bme.hu/nagyadat/Recommender_systems_handbook.pdf. [03.11.2014.].
- [116] Dostupno na:
<http://books.google.ba/books?id=531cAgAAQBAJ&pg=PA501&lpg=PA501&dq=intelligent+agent+based++programming+languages&source=bl&ots=oWUgFN-MUT&sig=0kyFJXLSZiZsfXijRwZP6NnNzv&hl=en&sa=X&ei=NWJaVOHjH6LA7Aad64DwDQ&ved=0CFUQ6AEwCA#v=onepage&q=intelligent%20agent%20based%20%20programming%20languages&f=false>. [05.11.2014.].
- [117] Dostupno na: <http://c2.com/cgi/wiki?WardCunningham>. [05.11.2014.].
- [118] Dostupno na:
<http://web20alati.pbworks.com/w/page/61915037/Primena%20wikija%20za%20kolaborativno%20u%C4%8Denje,%20pou%C4%8Davanje%20i%20istra%C5%BEivanje>. [05.11.2014.].

8 PRILOZI

RegistracijaLogiranje

FitWIKI

WikiQ&ANedavne promjeneSlučajna stranicaPomoćDonacije

Kreirajte članakPostavite pitanje

ČlanakRazgovorČitajUrediVidi izmjene

Korisnik: blocky | Reputacija: 19 | Članak kreiran: 30.3.2013 1:52:06 | Pregleda: 80 | Odgovora: 1 | Rejting: 0 | Komentara: 1



Lansiran Google Keep

Osim tekstualnih zabilježki, Keep može čuvati i zvukovne snimke, te fotografije. Sučelje podsjeća na pločice s Windows Phonea, koje je moguće reorganizirati u skladu s prioritetom. Uključene su funkcionalnosti pretrage i arhiviranja zabilježki, a za svaku je moguće odabrati i boju. Za uređaje s Androidom 4.2 ili novijim dostupan je i widget koji omogućuje korištenje aplikacije sa zaključanog ekrana uređaja.

Verzija aplikacije na webu dostupna je na adresi drive.google.com/keep, no korisnici zasad prijavljuju velik broj problema pri korištenju. U izdanju za Android aplikacija je dostupna s Play Storea.

UREDENO UREDENO

Osim tekstualnih zabilježki, Keep može čuvati i zvukovne snimke, te fotografije. Sučelje podsjeća na pločice s Windows Phonea, koje je moguće reorganizirati u skladu s prioritetom. Uključene su funkcionalnosti pretrage i arhiviranja zabilježki, a za svaku je moguće odabrati i boju. Za uređaje s Androidom 4.2 ili novijim dostupan je i widget koji omogućuje korištenje aplikacije sa zaključanog ekrana uređaja.

Verzija aplikacije na webu dostupna je na adresi drive.google.com/keep, no korisnici zasad prijavljuju velik broj problema pri korištenju. U izdanju za Android aplikacija je dostupna s Play Storea.

0Bodova

W Like

2 osoba je glasovalo za!

W Unlike

2 osoba je glasovalo protiv!

1

2

3

4

5

Za ocjenjivanje članaka morate se logirati



blocky | Reputacija19 | Sviđanja0 | Nesviđanja0

as

Komentiraj

WIKI preporuke

Google će proizvoditi Google naočale u Kaliforniji
Google Glass uskoro u proizvodnji
Kako Google zarađuje na mobilnoj platformi?
Miliarda korisnika na YouTubeu
iOS donosi četiri puta veće prihode od Androida

Q&A preporuke

Uklanjanje iz Google tražilice
Pozicioniranje web stranice na Google-u
Google AdSense na web stranici

Korisnici

blocky

19

kor12

0

kor14

0

Made by Elvis. Copyright © 2014 by Fit. All rights reserved.

Slika 11. FITCKMS – Modul Wiki (Članak)


103

Registracija
Logiranje


FitQ&A

Wiki
Q&A
Tagovi
Korisnici
Bedževi
Neodgovoreno


Postavite pitanje
Kreirajte članak




Naziv: King
Opis: King
Kreiran: prije prije
godinu dana
Bedz posjeduje: 0
korisnika




Naziv: Akademik
Opis: Akademik
Kreiran: prije prije
godinu dana
Bedz posjeduje: 0
korisnika




Naziv: Doktor
Opis: Doktor
Kreiran: prije prije
godinu dana
Bedz posjeduje: 0
korisnika




Naziv: Magistar
Opis: Magistar
Kreiran: prije prije
godinu dana
Bedz posjeduje: 0
korisnika




Naziv: Profesor
Opis: Profesor
Kreiran: prije prije
godinu dana
Bedz posjeduje: 0
korisnika




Naziv: Student
Opis: Student
Kreiran: prije prije
godinu dana
Bedz posjeduje: 0
korisnika




Naziv: Pionir
Opis: Pionir
Kreiran: prije prije
godinu dana
Bedz posjeduje: 0
korisnika



Naziv: Početnik
Opis: Početnik
Kreiran: prije prije
godinu dana
Bedz posjeduje: 13
korisnika



Naziv: Senior
Opis: Senior
Kreiran: prije prije
godinu dana
Bedz posjeduje: 1
korisnika



Naziv: Junior
Opis: Junior
Kreiran: prije prije
godinu dana
Bedz posjeduje: 1
korisnika

Slika 13. FITCKMS - Korisnički bedževi

```

public class EvaluateRecommender {

    public static void main(String[] args) throws Exception {

        DataModel model6 = new FileDataModel(new File("data/Fit.csv"));
        DataModel model1 = new FileDataModel(new File("data/m1_base.csv"));
        DataModel model2 = new FileDataModel(new File("data/m2_base.csv"));
        DataModel model3 = new FileDataModel(new File("data/m3_base.csv"));
        DataModel model4 = new FileDataModel(new File("data/m4_base.csv"));
        DataModel model5 = new FileDataModel(new File("data/m5_base.csv"));

        //RecommenderEvaluator//////////////////////////////////////
        RecommenderEvaluator evaluator = new RMSRecommenderEvaluator();
        RecommenderEvaluator evaluatara = new AverageAbsoluteDifferenceRecommenderEvaluator();

        //RecommenderBuilder//////////////////////////////////////
        RecommenderBuilder knn_Builder = new KnnPearson();
        Recommender r = knn_Builder.buildRecommender(model1);

        // method: Pearson, algorithm: Pearson Correlation, description: from similar items
        double knn1 = evaluator.evaluate(knn_Builder, null, model1, 0.7, 1.0);
        double knn2 = evaluator.evaluate(knn_Builder, null, model2, 0.7, 1.0);
        double knn3 = evaluator.evaluate(knn_Builder, null, model3, 0.7, 1.0);
        double knn4 = evaluator.evaluate(knn_Builder, null, model4, 0.7, 1.0);
        double knn5 = evaluator.evaluate(knn_Builder, null, model5, 0.7, 1.0);
        double knn6 = evaluator.evaluate(knn_Builder, null, model6, 0.8, 1.0);

        // method: Cosine, algorithm: Cosine similarity, description: from similar items
        RecommenderBuilder cos_Builder = new CosineSimilarity();
        double cos1 = evaluator.evaluate(cos_Builder, null, model1, 0.7, 1.0);
        double cos2 = evaluator.evaluate(cos_Builder, null, model2, 0.7, 1.0);
        double cos3 = evaluator.evaluate(cos_Builder, null, model3, 0.7, 1.0);
        double cos4 = evaluator.evaluate(cos_Builder, null, model4, 0.7, 1.0);
        double cos5 = evaluator.evaluate(cos_Builder, null, model5, 0.7, 1.0);
        double cos6 = evaluator.evaluate(cos_Builder, null, model6, 0.8, 1.0);

        // method: baseline, algorithm: Baseline, description: User and item averages
        RecommenderBuilder bas_Builder = new Baseline();
        double bas1 = evaluator.evaluate(bas_Builder, null, model1, 0.7, 1.0);
        double bas2 = evaluator.evaluate(bas_Builder, null, model2, 0.7, 1.0);
        double bas3 = evaluator.evaluate(bas_Builder, null, model3, 0.7, 1.0);
        double bas4 = evaluator.evaluate(bas_Builder, null, model4, 0.7, 1.0);
        double bas5 = evaluator.evaluate(bas_Builder, null, model5, 0.7, 1.0);
        double bas6 = evaluator.evaluate(bas_Builder, null, model6, 0.8, 1.0);
    }
}

```

Slika 14. Primjer Java koda

```

RecommenderBuilder knn_Builder = new KnnPearson();
Recommender r_knn = knn_Builder.buildRecommender(model7);
List<RecommendedItem> recommendations1 = r_knn.recommend(5, 6);
for (RecommendedItem recommendation : recommendations1) {
    System.out.println(recommendation.getItemID()+"\t"+recommendation.getValue());
}
    System.out.println("\n");

// method: Pearson, algorithm: Pearson Correlation, description: from similar items
double knn1 = evaluator.evaluate(knn_Builder, null, model1, 0.7, 1.0);
double knn2 = evaluator.evaluate(knn_Builder, null, model2, 0.7, 1.0);
double knn3 = evaluator.evaluate(knn_Builder, null, model3, 0.7, 1.0);
double knn4 = evaluator.evaluate(knn_Builder, null, model4, 0.7, 1.0);
double knn5 = evaluator.evaluate(knn_Builder, null, model5, 0.7, 1.0);
double knn6 = evaluator.evaluate(knn_Builder, null, model6, 0.7, 1.0);

// method: Cosine, algorithm: Cosine similarity, description: from similar items

RecommenderBuilder cos_Builder = new CosineSimilarity();
Recommender r_cos = cos_Builder.buildRecommender(model1);
List<RecommendedItem> recommendations2 = r_cos.recommend(5, 6);
for (RecommendedItem recommendation : recommendations2) {
    System.out.println(recommendation.getItemID()+"\t"+recommendation.getValue());
}
    System.out.println("\n");

double cos1 = evaluator.evaluate(cos_Builder, null, model1, 0.7, 1.0);
double cos2 = evaluator.evaluate(cos_Builder, null, model2, 0.7, 1.0);
double cos3 = evaluator.evaluate(cos_Builder, null, model3, 0.7, 1.0);
double cos4 = evaluator.evaluate(cos_Builder, null, model4, 0.7, 1.0);
double cos5 = evaluator.evaluate(cos_Builder, null, model5, 0.7, 1.0);
double cos6 = evaluator.evaluate(cos_Builder, null, model6, 0.7, 1.0);

// method: baseline, algorithm: Baseline, description: User and item averages

RecommenderBuilder bas_Builder = new Baseline();
Recommender r_bas = bas_Builder.buildRecommender(model1);
List<RecommendedItem> recommendations3 = r_bas.recommend(5, 1122);
for (RecommendedItem recommendation : recommendations3) {
    System.out.println(recommendation.getItemID()+"\t"+recommendation.getValue());
}
    System.out.println("\n");
double bas1 = evaluator.evaluate(bas_Builder, null, model1, 0.7, 1.0);
double bas2 = evaluator.evaluate(bas_Builder, null, model2, 0.7, 1.0);

```

Slika 15. Preporuke i procjena (Java code)


```

648
649 // SlopeOne////////////////////////////////////
650 class SlopeOne implements RecommenderBuilder {
651     @SuppressWarnings("deprecation")
652     public Recommender buildRecommender(DataModel dataModel)
653         throws TasteException {
654         DiffStorage diffStorage = new MemoryDiffStorage(dataModel, Weighting.UNWEIGHTED, Long.MAX_VALUE);
655         return new SlopeOneRecommender(dataModel, Weighting.UNWEIGHTED, Weighting.UNWEIGHTED, diffStorage);
656     }
657 }
658
659 // Singular Value Decomposition - ALSWRFactorizer - 10
660 // Features////////////////////////////////////
661 class SVD1 implements RecommenderBuilder {
662     public Recommender buildRecommender(DataModel dataModel)
663         throws TasteException {
664         return new SVDRecommender(dataModel, new ALSWRFactorizer(dataModel, 10,
665             0.05, 10));
666     }
667 }
668
669 // Singular Value Decomposition - ALSWRFactorizer - 20
670 // Features////////////////////////////////////
671 class SVD2 implements RecommenderBuilder {
672     public Recommender buildRecommender(DataModel dataModel)
673         throws TasteException {
674         return new SVDRecommender(dataModel, new ALSWRFactorizer(dataModel, 20,
675             0.08, 10));
676     }
677 }
678
679 // Singular Value Decomposition - SVDPlusPlusFactorizer - 10
680 // Features////////////////////////////////////
681 class SVD3 implements RecommenderBuilder {
682     public Recommender buildRecommender(DataModel dataModel)
683         throws TasteException {
684         return new SVDRecommender(dataModel, new SVDPlusPlusFactorizer(
685             dataModel, 10, 10));
686     }
687 }
688
689 // Singular Value Decomposition - SVDPlusPlusFactorizer - 10
690 // Features////////////////////////////////////

```

Slika 16. Implementacija algoritama (Java code)

POPIS SLIKA:

Slika 1. Kombinacija vrijeme/mjesto komunikacijskih modela i kolaborativnih tehnologija	11
Slika 2. Kolaborativno učenje	19
Slika 3. Koncept kolaborativnog učenja.....	20
Slika 4 MediaWiki - logo	22
Slika 5. Struktura inteligentnog programskog agenta	26
Slika 6. Slojevi sistema preporuke	45
Slika 7. Baza podataka FITKMS.....	59
Slika 8. Klasa DataConvert	67
Slika 9. SQL Upit - Spremanje podataka FITCKMS	68
Slika 10. Simulacija programskog agenta	68
Slika 11. FITCKMS – Modul Wiki (Članak).....	103
Slika 12. FITCKMS - Modul Pitanja i odgovori	104
Slika 13. FITCKMS - Korisnički bedževi	105
Slika 14. Primjer Java koda	106
Slika 15. Preporuke i procjena (Java code)	107
Slika 16. Implementacija algoritama (Java code).....	108

POPIS TABELA:

Tabela 1. Težine bedževa	65
Tabela 2. Sistemi preporuke	70
Tabela 3. Preporučeni resursi (FITCKMS baza podataka).....	74
Tabela 4. Preporučeni resursi (MovieLens podskup d1).....	75
Tabela 5. RMSE vrijednosti sistema preporuke	76
Tabela 6. Statističke vrijednosti sistema preporuke	77
Tabela 7. Komparacija rezultata preporuke standardnih metoda nad različitim skupovima iz baze podataka FITCKMS	79
Tabela 8. Brzina izvođenja standardnih metoda (skup "FitPonder.csv")	82
Tabela 9. Brzina izvođenja standardnih metoda (MovieLens).....	83

POPIS GRAFIKONA:

Grafikon 1. Minimalna i maksimalna odstupanja od srednje RMSE vrijednosti	78
Grafikon 2. Brzina izvođenja standardnih metoda (skup "FitPonder.csv")	83
Grafikon 3. Brzina izvođenja standardnih metoda (MovieLens)	84

POPIS SKRAĆENICA:

LMS – Learning Management System

VLE – Virtual Learning Environment

KMS - Knowledge Management System

CIS - Collaborative Information Systems

CKMS - Collaborative Knowledge Management System

SN – Social Network

VO – Virtual Community

GSS - Group Support Systems

HTML - Hypertext Markup Language

RDF - Resource Description Framework

OWL - Web Ontology Language

XML - Extended Markup Language

CIS - Collaborative Information Systems

OL - Organizational Learning

OM - Organizational Memory

KR - Knowledge Represenatation

KDQL - Knowledge Discovery Query Language

ICT - Information and Communication Technology

CGI - Computer-generated imagery

AR - adaptive recommenders

QA – Questions and Answers

KM - Knowledge Management

DM - Data Mining